Contents

Research papers

Methodological papers

ANNALES MATHEMATICAE ET INFORMATICAE 52. (2020)

ANNALES MATHEMATICAE ET INFORMATICAE

TOMUS 52. (2020)

EGER 1774
ESZTERHÁZY KÁROLY UNIVERSITY

HUNGARIA, EGER

ANNALES MATHEMATICAE ET INFORMATICAE

VOLUME 52. (2020)

INSTITUTE OF MATHEMATICS AND INFORMATICS
ESZTERHÁZY KÁROLY UNIVERSITY
HUNGARY, EGER

HU ISSN 1787-6117 (Online)

A kiadásért felelős az
Eszterházy Károly Egyetem rektora
Megjelent a Líceum Kiadó gondozásában
Kiadóvezető: Dr. Nagy Andor
Felelős szerkesztő: Dr. Domonkosi Ágnes
Műszaki szerkesztő: Dr. Tómács Tibor
Megjelent: 2020. december

Research papers

Annales Mathematicae et Informaticae
***52** (2020) pp. 5–18*
DOI: https://doi.org/10.33039/ami.2020.12.003
URL: https://ami.uni-eszterhazy.hu

A static analysis method for safe comparison functors in C++*

Bence Babati[a], Norbert Pataki[b]

[a]Department of Programming Languages and Compilers
Eötvös Loránd University, Budapest, Hungary
babati@caesar.elte.hu

[b]ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
patakino@elte.hu

Submitted: March 17, 2020
Accepted: December 12, 2020
Published online: December 17, 2020

Abstract

The C++ Standard Template Library (STL) is the most well-known and widely used library that is based on the generic programming paradigm. STL takes advantage of C++ templates, so it is an extensible, effective and flexible system. Professional C++ programs cannot miss the usage of the STL because it increases quality, maintainability, understandability and efficacy of the code.

However, the usage of C++ STL does not guarantee perfect, error-free code. Contrarily, incorrect application of the library may introduce new types of problems. Unfortunately, there is still a large number of properties that are tested neither at compilation-time nor at run-time. It is not surprising that in implementations of C++ programs so many STL-related bugs may occur.

It is clearly seen that the compilation validation is not enough to exclude STL-related bugs. For instance, the mathematical properties of user-defined sorting parameters are not validated at compilation phase nor at run-time. Contravention of the strict weak ordering property results in weird behavior

*The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

that is hard to debug. In this paper, we argue for a static analysis tool which finds erroneous implementation of functors regarding the mathematical properties. The primary goal is to support Continuous Integration pipelines, using this tool during development to overcome debugging efforts.

Keywords: C++, static analysis, STL, generic programming, functor

MSC: 68N19 Other programming techniques

1. Introduction

The C++ Standard Template Library (STL) is a widely-used, handy library based on the generic programming paradigm [2]. On one hand, the library provides convenient, suitable containers (e.g. `list`) and algorithms (e.g. `find`) that make easier stock-in-trade [19]. On the other hand, STL introduces many new kinds of bugs which are hard to detect and fix, such as invalid iterators, weird effect of the `remove` algorithm and writing uninitialized memory via `copy` algorithm, etc. [16]

STL provides four standard sorted associative containers, these are `set`, `map`, `multiset` and `multimap` [8]. These containers are able to work together with user-defined orders via functor types [21]. In this case, the user-defined functor has to implement strict weak ordering, but this property is not validated neither at compilation time nor at runtime [15]. If someone uses a functor which does not fulfill the strict weak ordering rules, the container becomes inconsistent because same values are not considered to be equal [14]. Let us consider the following code:

```
struct Comp
{
  bool operator()( int a, int b ) const
  {
    return a >= b;
  }
};
// ...
std::set<int, Comp> s;
s.insert( 3 );
s.insert( 3 );
std::cout << s.size();
  // Prints 2 that is weird because same value inserted twice
  // into the set. Correctly, 1 should be printed.
std::cout << s.count( 3 ); // prints 0 in spite of it is contained
```

This phenonmenon is weird, the root cause is hard to find. Compilers should emit error (or warning at least) diagnostics, but the problem is not detected at all. Strict weak order property should be an *axiom* according to modern generic constraint approach in C++. However, these axioms are not validated by the compiler [22]. Therefore, our aim is to develop a tool based on static analysis that detects problematic functors.

This tool is based on a recently popular software, called Clang. Clang is a standard compliant C/C++/Objective-C compiler, furthermore, it provides a static analyzer, as well. It is open source and based on the LLVM compiler infrastructure. It is mainly developed by the community, there are many contributors, also it is supported by big companies as well [3].

The Clang architecture is well designed and modular which makes it possible to use it as a library [17]. The users can use the end products, like Clang as a compiler or build their own tools on top of its libraries. It provides an API for third-parties to use its internal structures and analyze the source code in a high-level way. Its libraries provide a wide scale of features related to compilation and analysis, for example tokenizer or AST visitor. Many useful static analysis tools have been developed based on Clang (e.g. [1, 4, 10]). Clang's another significant advantage is the evolving approach regarding the C++ standards, so users do not need to take care of parsing of newly introduced language elements and can focus on their actual goal. That makes Clang powerful and very popular recently.

The rest of this paper is organized as follows: the related work is discussed in Section 2, the technical details of our Clang-based solution are presented in Section 3 and decision logic is explained in Section 4. Our approach is evaluated and results are shown in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

A comprehensive description of STL-related bugs can be found in [14] including the ordering functor types' mathematical properties, as well. However, many problems have been presented, but no tool support was proposed to avoid the erroneous situations. Compilation time validation of the STL typically uses two different approaches: template metaprogramming (e.g. [18]) and static analysis (e.g. [4, 9]). These methods do not help to find the problematic ordering functor types in C++ source, the functors' statefulness is analyzed exclusively [10]. Model checking of STL containers also misses the validation of user-defined comparisons [6].

On the other hand, C++ functors are analyzed previously, a limited, lightweight, runtime approach has been developed [15]. This approach has runtime overhead and does not deal with comprehensive evaluation.

Another direction in functors' usage is a transparent version of the functor templates [12]. The paper presents a refactoring tool which makes the usage of functors safer, but this tool does not deal with the mathematical properties.

The *constraints* and *concepts* [22] have been included officially in the *C++20* standard version. These let the users to define compile time expectations on the template parameters. For example, it can be checked pragmatically that a given `T` template parameter type has `operator()` member function or not. However, the beforehand presented STL-related issue is more complex, it requires to check the implementation of the given functions as well.

3. Our Approach

3.1. Technical Background

The previously depicted theoretical problem may appear sometimes. However, the compiler cannot warn about it at all. In order to detect this kind of problem, a brand new tool has been developed. Its purpose is to find misuses of ordered associative containers related to the given issue. Many faulty functor classes can be caught in suspicious context, although, the tool has limitations which are described at the end of this section.

The implementation uses Clang's libraries and framework to analyze the C++ source code. It takes advantage of Clang's architecture including the built-in abstract syntax tree (AST) and its visitors. AST is comprehensively used in our tool to extract information from the source code.

3.2. High-level Overview

This section presents a high-level overview and describes how our tool works in a nutshell [5]. As it was mentioned above, it works on the source code itself and it does not require to execute the binary.

That means, it can only rely on compile time information which are given in the source code. The original compiler arguments are very essential regarding the reproducible compilation process. These arguments or flags may affect the whole compilation process, for instance preprocessor macros often depend on the compilation arguments.

In general, let us see what is the idea behind the analysis and how the workflow looks like. This solid outline will highlight the main points of the analysis and how it is performed to gather the necessary information from the source code.

The main problem is related to the associative containers and the regarding user-defined ordering functors. At the beginning, every instantiation of associated containers has to be found which uses a custom functor for comparing objects. The functor classes only can be identified at usage places, because the instantiated assiative container is the evidence of the given functor must meet certain requirements. The beforehand found instantiations each has a functor whose type is a suspect of misusage.

These marked types are analysed in the next step. The tool retrieves the type of comparison functor and tries to find the proper `operator()` for the given usage. Two cases are possible, the definition is not available, for instance it is defined in another translation unit, it will be skipped. This case is rare because most of comparisons have short implementation, so they are typically inline methods in the class. Another case, when the definition of candidate `operator()` is available, it can be analyzed in order to extract the expressions which are used to compare two objects. From one function, multiple expressions can be collected, for example the return value depends on a condition. The following code snippet presents this case:

```
bool ExampleComp::operator()( int lhs, int rhs ) const
{
  if ( lhs > 0 && rhs > 0 )
  {
    return lhs < rhs;
  }
  else
  {
    return lhs * 2 <= rhs + 1;
  }
}
```

These collected expressions are evaluated later in order to decide whether they meet the requirement of strict weak ordering rules. The details of the proposed analysis method can be seen below.

3.2.1. Analyzing AST

In our tool, Clang libraries are in-use to parse the source code and build internal structures. Clang performs every low level action (tokenizing, parsing, etc.) that lets us to concentrate on our aim by defining a higher level analysis based on the built structures.

The main and worth to mention data structure of them is the abstract syntax tree, AST. It represents the source code in an abstract way, contains all the data about the parsed source files. In Clang, it is a little bit more than a syntax tree, because it contains some semantic information as well.

To collect data from abstract syntax trees, they can be visited by AST visitors. Custom AST visitors need to be implemented in order to use the Clang hierarchy and AST visitor interface. AST visitors can extract the relevant information from the AST and capture any kind of context within the AST, for example, all function declarations can be visited.

The proposed tool is mostly built on AST visitors. These visitors can be used to find container instantiations, types, member functions, expressions and many other source-based constructs. More precisely three different kinds of visitors have been declared. Each has different tasks on different part of the AST. These visitors work together and built on each other.

The following paragraphs detail these AST visitors and the presented order is the same as the order of processing. That means in the analysis logic, the visitor which finds associate container instantiations is used before the visitor which parses the body of member functions.

Usage finder visitor The original issue can occur only when someone uses `std::map`, `std::multimap`, `std::set` or `std::multiset` with custom comparison objects. The first task is to find template instantiations of previously listed

types and inspect them in order to find those which are using custom comparison types other than the default `std::less`.

Although, `std::less` can be specialized for used defined types, in this case the written comparator is user-defined and it should be analysed as well. Other special case, when the default `std::less` is provided without any specialization, in this case the `operator<` is called on the objects. The custom object comparison can sneak into without using custom functors. From the analysis point of view, the only difference is that the `operator<` function should be analysed instead of the `operator()` of the provided functor. However, it is not covered in this paper, focusing only on user-defined functors.

When an instantiation meets the given criteria, it should be analyzed because it can be erroneous, for example `SpecialKeyCmp` class is used here:

```
std::map<SpecialKey, int, SpecialKeyCmp> m;
```

After these usage places are located, the classes of the used functors need to be checked. For this, it is necessary to find the definition of the used functor type and the matching `operator()` member function for the given usage. When the definition of `operator()` is available in this translation unit, it can be used to furthermore processing, but it is done by next visitor.

Function body parser The next AST visitor is responsible for parsing the function implementation. Its input is the function definition in the AST, the usage finder visitor passes the `operator()` member function definitions to this visitor.

The visitor's purpose is to extract one or more expressions from the function body which can be used to compare objects. This kind of visitor can locate and capture every logical or comparison expression which can affect the return value. The outcome is a list of expressions which can define the return value of the given function. The visitor needs to process the job backward, because the root expression which defines the return value, can be identified only at the end of each execution path. These end points are the `return`-statements in the function body.

However, it is not adequate to process only them. It can happen that someone declares a local variable or calls a function to evaluate an expression. This visitor needs to handle variable declarations and assignments, when an expression is bounded to a name which is used in `return`-statement. The names are replaced by the bounded expressions in the `return`-statement.

Nevertheless it tracks function calls which can modify variables or their return values appear in the expressions. In case of function calls, the function body is parsed with another object of this visitor to get the relevant expressions.

An important point here is to manage the currently valid conditions on the given execution path. It is necessary, because the conditions can affect the return value, in some case, they define the comparison implicitly. For example, without analysing the conditions, the following functor cannot be judged well, however, it definitely breaks the strict weak ordering rule.

```
bool CustomComp::operator()( int lhs, int rhs ) const
{
  if ( lhs < rhs )
  {
    return false;
  }
  else
  {
    return true;
  }
}
```

In addition to all of this, they need to be performed recursively in order to dissolve an expression as much as possible at compile time. For instance, when a function calls another one which affects the return value in some way, it is necessary to inspect that function and substitute it with extracted elemental expressions.

This visitor deals with the following code context:

```
bool CustomComp::inRange( int value ) const
{
  return value < 42;
}

bool CustomComp::operator()( int lhs, int rhs ) const
{
  const bool tmp = lhs > 0 && rhs > 0;
  return tmp && inRange( lhs ) && inRange( rhs ) && lhs < rhs;
}
```

In this example, the expression which actually will be evaluated at each `operator()` function call is: `lhs > 0 && rhs > 0 && lhs < 42 && rhs < 42 && lhs < rhs`.

Expression parser This is the lowest level visitor in this implementation. This parser works on a very small part of the AST, the beforehand located expressions are visited by it. Its purpose is parsing the given expressions and convert them to an internal data structure. The advantage of this data structure is that, it is far simpler than Clang's AST and contains only the relevant information.

The internal data structure is a graph which represents logical and comparison expressions. The nodes are typically operators and variables but more constructs are supported. The edges are logical relations between nodes, for example, the `operator<` has two other nodes which are the left and right hand side operands of expression.

The visitor handles binary operators, unary operators, literals, variables and so on. It walks over on that small part of AST and tranforms nodes to proper internal data structures. At the end of the visiting of Clang's AST, the result is a

graph which is identical to the original one without excess. For example, the graph belongs to the `a > 0 && b > 0` code snippet is depicted in Figure 1.

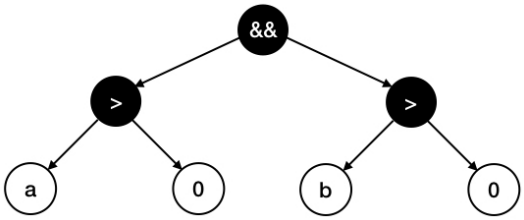


Figure 1: Internal data structure

With this step, the AST processing is mostly done. A list of expressions is extracted for each instantiation which needs to be analyzed later, however, before performing the concrete analysis, some small transformations need to be applied on them. These transformations are detailed in the next subsection.

3.2.2. Transformations

After processing of Clang's AST, an internal graph structure is created for each expression at each functor usage place. They are identical to the original expressions, although most of time they are not that complex. To reduce this complexity, some modifications need to be applied on them. After the transformations, the expressions will be equivalent with the original one, but simpler.

They target to eliminate the obvious complications and keep the expressions plain. There are several well-known replacement rules related to mathematical logic [7]:

- De Morgan's laws: `!(X || Y) -> !X && !Y`
- Double negation: `!!X -> X`
- Tautology: `X && X -> X`

Besides that more transformations can be applied at compile time which comes from the programming language behavior:

- Short-circuit binary operators: at logical `and` and `or`, when the first operand is evaluated, it may define the result of the whole expression, e.g.: `true || (X < 0) -> true`
- Constant evaluation: comparisons may be evaluated at compile time, e.g.: `0 < 42 -> true`

Using these replacement rules, the original expression can be transformed into a new expression which contains less boilerplate. For example, the expression `(x`

`< y) || (0 != 0)` can be converted into `x < y`. The `0 != 0` is not relevant from the analysis point of view since it is always `false` and the outcome of the original expression does not depend on it.

These transformations are applied on each expression when it is possible. This approach results in a new, simplified expression which can be analyzed with more confidence. These newly created expressions will be used later in order to decide the correctness of functors.

3.2.3. Output format

After finding a custom functor suspicious, the tool emits a warning like the compiler does, but it refers to the type that can be seen in the source, not the underlying one [14]. It uses Clang's diagnostic framework to report issues, so they look like a compiler warning at the line of data structure usage, e.g. instantiation of `std::map`.

```
main.cpp:44:10: warning: Strict weak ordering is not fulfilled
                         by comparison type
    std::set<int, Comp> s;
```

4. Decision Logic

The analysis can be executed on cleaned expressions that are prepared to be analyzed whether they meet the requirement of strict weak ordering rules.

Let A be an arbitrary set and relation $R \subseteq A \times A$. It is a strict weak ordering if the following properties are met[20]:

- Asymmetry: $\forall a, b \in A : aRb \Rightarrow \neg(bRa)$.
- Irreflexivity: $\forall a \in A : \neg aRa$.
- Transitivity: $\forall a, b, c \in A : aRb \wedge bRc \Rightarrow aRc$.

On one hand, this analysis is pragmatic and conservative, therefore it minimizes the false positive warnings which is an essential property in static analysis tools, but on the other hand, the tool is not a theorem prover.

The decision logic takes advantage of the previously presented visitors. The pseudocode of the decision logic can be seen in Figure 2, the entry point is the `DecisionLogic` procedure. We omit the proper type information but the informal description helps to comprehend the proposed solution. In this procedure, the first attribute to check whether the comparison uses both arguments because a regular binary relation is required. We use the `ParseNumberOfUtilizedParams` function that is straightforward, therefore we not detailed in Figure 2. If the comparison does not utilize any of its argument, we emit a warning by calling `EmitWarning` that is not detailed in the pseudocode, but presented in Section 3.2.3. However, the functor's `operator()` must have two parameters due to the compilation model of C++ but parameter can be unused [18].

```
procedure CheckExpression(<simplified structure of> expression)
    operator ← ParseOperator(expression)

    if operator is operator== ∨ operator is operator!= then
        EmitWarning
    end if

    if operator is operator<= ∨ operator is operator>= then
        EmitWarning
    end if
end procedure
procedure CheckLiteral(functor)
    literal, expression ← ParseLiteralCondition(functor)

    if ¬(Evaluate(literal)) then
        expression ← ¬(expression)
    end if
    CheckExpression(expression)
end procedure
procedure DecisionLogic(functor)
    params ← ParseNumberOfUtilizedParams(functor)
    if params ≠ 2 then
        EmitWarning
    else
        entity ← ParseReturnEntityType(functor)
        if entity is expression then
            CheckExpression(expression)
        end if
        if entity is literal then
            CheckLiteral(functor)
        end if
        if entity is variable then
            value, success ← ParseVariableValue(functor)

            if success then
                expression ← ParseCondition(functor)

                if ¬ Evaluate(value) then
                    expression ← ¬(expression)
                end if
                CheckExpression(expression)
            end if
        end if
    end if
end procedure
```

Figure 2: Pseudocode for the Decision Logic

If both arguments take part in the comparison, we query what kind of result is specified in the `return`-statement. The potential kinds are expressions, literals (e.g. `true`, or `0`), variables but every kind may depend on function calls that we process by inlining them on the level of AST. However, we do not highlight this fact in Figure 2.

The parameter of the decision logic is the AST representation of the analyzed functor. When we produce the cleaned, simplified expression that we take advantage of transformation steps presented in Section 3.2.2. If this transformed expression contains one of the following operators `<=`, `>=`, `==` or `!=`, we emit a warning, otherwise we consider the comparison meets the requirement conservatively. We query the applied operator with `ParseOperator` method in Figure 2.

When the returned element in the `return`-statement is a literal and the comparison utilizes both parameters the result must depend on a condition. As Section 3 presented, this condition is retrieved by our visitors and the condition is negated when the literal is false or converted to false with the `Evaluate` function. We also showed previously if there are multiple conditional statements, we process all these conditions in the `ParseCondition` function that is not detailed in Figure 2. In case of returned literal is considered to be true by the `Evaluate` method, the condition remains untouched. This condition contains operator to compare the arguments, so we evaluate this processed condition just like the expression previously.

In case of variable is returned, we call the `ParseVariableValue` procedure to recognize its value if we are able to specify it. This recognized value can be used as a literal and evaluate the comparison just like the previous case. We do not emit warning, if the value of the variable cannot be determined. Of course, this can cause false negative cases during the analysis, but it is not a typical use-case.

Briefly, our tool also emits a warning when it detects that the arguments are compared with `operator==` or `operator!=`. If an ordering relation is defined as a C++ comparison functor in an erroneous way, the asymmetry and transitivity requirements are still met. The problematic property is the irreflexivity, therefore our tool focuses on the validation of this requirement that is the most common misuse regarding functors [14]. The possible comparison operators are `<`, `>`, `<=`, `>=`. Although the operators `<` and `>` are considered right, they cannot cause issues regarding to the given problem. The rest of them may cause issues, since the equality is included in all of them, thus we emit warning in these cases.

We also take into consideration whether the arguments are compared with constant values, but they are compared to each other with `<=` or `>=`, therefore this essential expression of functor is incorrect: `lhs > 0 && rhs > 0 && lhs <= rhs`.

5. Limitations and Evaluation

The tool has some limitations, which one should bear in mind. First of them comes from Clang's nature, it handles translation units separately, so if the `operator()` is defined in a different source file (`.cc`) where the container is instantiated with the corresponding functor class, the tool cannot find the operator's definition due

to Clang's limitation [11]. In this case, the given functor will not be analysed.

Another issue is related to compile time behavior, no runtime information is available for the analysis; also if a very tricky comparison expression is written, likely the functor cannot be decided if it is compliant or not.

During the development of the tool, some handmade test cases have been implemented. They are good to cover all the corner cases in theory, however, it would be good to see how the proposed tool performs on real-world projects.

Since the effect of this issue is very well-marked and serious, they usually are eliminated during the development or testing phase of real products.

Nonetheless, in order to ascertain the quality of our approach and solution, the tool was tested and evaluated on well-known open source projects. The user-defined comparison functor usage with associative containers is not used very often, so a limited number of projects could be checked unfortunately. However, even comprehensive profiling does not measure the functors' usage [13].

The methodology of testing was that the tool reported that a functor is being analyzed then the result of the analysis is checked. Each functor which was reportedly analyzed is inspected manually, as well. That makes it possible to verify the result of the tool.

In this testing, four different functors are analyzed from three different projects listed below:

- Flatbuffers - `https://github.com/google/flatbuffers/`
- Thrift - `https://github.com/apache/thrift/`
- Orc - `https://github.com/apache/orc`

All of the analyzed functors are used with `std::map` container. None of them was reported as suspicious by our tool and the manual verification proved the results' correctness. Despite of the limitations of the tool, every functor's properties are evaluated correctly. The limitations do not affect the usage of tool in the source code of real-world applications. The tool does not emit false positive reports at all, so it can be used safely in quality assurance regularly.

6. Conclusion

C++ STL is a widely-used library that is based on the generic programming paradigm. The usage of the library increases the code quality and comprehensibility, however, the incorrect usage of library may result in new kind of errors.

This paper has presented a weird error related to the C++ Standard Template Library that is related to sorted associated containers. The ordering can be customized via functor class, but it should implement strict weak ordering. However, this property is not validated at all. If a functor does not meet this requirement, the container becomes inconsistent.

So in order to detect this kind of defects in the source code, a new approach has been proposed. We have developed a tool for this method. The proposed solution

analyzes source code that means the execution of the program is not required. It is a Clang-based tool that takes advantage of Clang's libraries and framework. Our tool was tested on manually prepared test cases and it was evaluated on open source projects to prove that it works perfectly with real-world applications.

The tool did not find any questionable functor, however, it confirms our tool validity and the fact that is not a very often issue in released projects. Although it does not report unnecessary false positive alarms, so it can be a handy tool in the development process and Continuous Integration servers for quick feedback, as well.

References

[1] M. Arroyo, F. Chiotta, F. Bavera: *An user configurable Clang Static Analyzer taint checker*, in: 2016 35th International Conference of the Chilean Computer Science Society (SCCC), Oct. 2016, pp. 1–12,
doi: 10.1109/SCCC.2016.7835996.

[2] M. H. Austern: *Generic Programming and the STL: Using and Extending the C++ Standard Template Library*, Addison-Wesley, 1999, isbn: 0-201-30956-4.

[3] B. Babati, G. Horváth, N. Pataki, A. Páter-Részeg: *On the Validated Usage of the C++ Standard Template Library*, in: Proceedings of the 9th Balkan Conference on Informatics, BCI'19, Sofia, Bulgaria: ACM, 2019, 23:1–23:8, isbn: 978-1-4503-7193-3,
doi: 10.1145/3351556.3351570,
url: http://doi.acm.org/10.1145/3351556.3351570.

[4] B. Babati, N. Pataki: *Analysis of Include Dependencies in C++ Source Code*, in: Communication Papers of the 2017 Federated Conference on Computer Science and Information Systems, ed. by M. Ganzha, L. Maciaszek, M. Paprzycki, vol. 13, Annals of Computer Science and Information Systems, PTI, 2017, pp. 149–156,
doi: 10.15439/2017F358,
url: http://dx.doi.org/10.15439/2017F358.

[5] B. Babati, N. Pataki: *Static analysis of functors' mathematical properties in C++ source code*, AIP Conference Proceedings 2116.1 (2019), p. 350002,
doi: 10.1063/1.5114355, eprint: https://aip.scitation.org/doi/pdf/10.1063/1.5114355,
url: https://aip.scitation.org/doi/abs/10.1063/1.5114355.

[6] N. Blanc, A. Groce, D. Kroening: *Verifying C++ with STL Containers via Predicate Abstraction*, in: Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering, ASE '07, Atlanta, Georgia, USA: Association for Computing Machinery, 2007, pp. 521–524, isbn: 9781595938824,
doi: 10.1145/1321631.1321724,
url: https://doi.org/10.1145/1321631.1321724.

[7] A. Church: *Introduction to mathematical logic*, Princeton University Press, 1996, isbn: 978-0691029061.

[8] D. Das, M. Valluri, M. Wong, C. Cambly: *Speeding up STL Set/Map Usage in C++ Applications*, in: Performance Evaluation: Metrics, Models and Benchmarks, ed. by S. Kounev, I. Gorton, K. Sachs, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 314–321, isbn: 978-3-540-69814-2.

[9] D. Gregor, S. Schupp: *STLlint: lifting static checking from languages to libraries*, Software: Practice and Experience 36.3 (2006), pp. 225–254,
doi: 10.1002/spe.683, eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.683,
url: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.683.

[10] G. Horváth, N. Pataki: *Clang matchers for verified usage of the C++ Standard Template Library*, Annales Mathematicae et Informaticae 44 (2015), pp. 99–109,
URL: http://ami.ektf.hu/uploads/papers/finalpdf/AMI_44_from99to109.pdf.

[11] G. Horváth, N. Pataki: *Source Language Representation of Function Summaries in Static Analysis*, in: Proceedings of the 11th Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems, ICOOOLPS '16, Rome, Italy: ACM, 2016, 6:1–6:9, ISBN: 978-1-4503-4837-9,
DOI: 10.1145/3012408.3012414,
URL: http://doi.acm.org/10.1145/3012408.3012414.

[12] G. Horváth, N. Pataki: *Transparent functors for the C++ Standard Template Library*, in: Proceedings of the 11th Joint Conference on Mathematics and Computer Science, ed. by E. Vatai, CEUR-WS, 2016, pp. 96–101.

[13] P. Jungblut, R. Kowalewski, K. Fürlinger: *Source-to-Source Instrumentation for Profiling Runtime Behavior of C++ Containers*, in: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), June 2018, pp. 948–953,
DOI: 10.1109/HPCC/SmartCity/DSS.2018.00157.

[14] S. Meyers: *Effective STL*, Addison-Wesley, 2001, ISBN: 0-201-74962-9.

[15] N. Pataki: *Advanced Functor Framework for C++ Standard Template Library*, Studia Universitatis Babeş-Bolyai Informatica LVI (2011), pp. 99–113.

[16] N. Pataki: *C++ Standard Template Library by safe functors*, in: Proc. of 8th Joint Conference on Mathematics and Computer Science, MaCS, 2010, pp. 363–374.

[17] N. Pataki, T. Cséri, Z. Szűgyi: *Task-specific style verification*, AIP Conference Proceedings 1479.1 (2012), pp. 490–493,
DOI: 10.1063/1.4756173, eprint: https://aip.scitation.org/doi/pdf/10.1063/1.4756173,
URL: https://aip.scitation.org/doi/abs/10.1063/1.4756173.

[18] N. Pataki, Z. Porkoláb: *Extension of Iterator Traits in the C++ Standard Template Library*, in: Proceedings of the Federated Conference on Computer Science and Information Systems, ed. by M. Ganzha, L. Maciaszek, M. Paprzycki, Szczecin, Poland: IEEE Computer Society Press, 2011, pp. 911–914.

[19] N. Pataki, Z. Szűgyi, G. Dévai: *Measuring the Overhead of C++ Standard Template Library Safe Variants*, Electronic Notes in Theoretical Computer Science 264.5 (2011), Proceedings of the Second Workshop on Generative Technologies (WGT) 2010, pp. 71–83, ISSN: 1571-0661,
DOI: https://doi.org/10.1016/j.entcs.2011.06.005,
URL: http://www.sciencedirect.com/science/article/pii/S1571066111000764.

[20] F. Roberts, B. Tesman: *Applied combinatorics*, CRC Press, 2009.

[21] B. Stroustrup: *The C++ Programming Language (special edition)*, Addison-Wesley, 2000, ISBN: 0-201-70073-5.

[22] A. Sutton, B. Stroustrup: *Design of Concept Libraries for C++*, in: Software Language Engineering, ed. by A. Sloane, U. Assmann, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 97–118, ISBN: 978-3-642-28830-2.

Annales Mathematicae et Informaticae
***52** (2020) pp. 19–30*
DOI: https://doi.org/10.33039/ami.2020.05.005
URL: https://ami.uni-eszterhazy.hu

Convergence of positive series and ideal convergence*

Vladimír Baláž[a], Kálmán Liptai[b], János T. Tóth[c], Tomáš Visnyai[a]

[a]Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, University of Technology in Bratislava, Radlinského 9, 812 37 Bratislava, Slovakia
vladimir.balaz@stuba.sk
tomas.visnyai@stuba.sk

[b]Department of Applied Mathematics, Eszterházy Károly University, Leányka 4 3300 Eger, Hungary
liptai.kalman@uni-eszterhazy.hu

[c]Department of Mathematics, J. Selye University, P. O. Box 54, 945 01 Komárno, Slovakia
tothj@ujs.sk

Submitted: May 13, 2020
Accepted: May 30, 2020
Published online: June 15, 2020

Abstract

Let $\mathcal{I} \subseteq 2^{\mathbb{N}}$ be an admissible ideal, we say that a sequence (x_n) of real numbers $\mathcal{I}-$converges to a number L, and write $\mathcal{I} - \lim x_n = L$, if for each $\varepsilon > 0$ the set $A_\varepsilon = \{n : |x_n - L| \geq \varepsilon\}$ belongs to the ideal $\mathcal{I}$. In this paper we discuss the relation ship between convergence of positive series and the convergence properties of the summand sequence. Concretely, we study the ideals $\mathcal{I}$ having the following property as well:

$$\sum_{n=1}^{\infty} a_n^{\alpha} < \infty \text{ and } 0 < \inf_n \frac{n}{b_n} \leq \sup_n \frac{n}{b_n} < \infty \Rightarrow \mathcal{I} - \lim a_n b_n^{\beta} = 0,$$

*This contribution was partially supported by The Slovak Research and Development Agency under the grant VEGA No. 2/0109/18.

where $0 < \alpha \leq 1 \leq \beta \leq \frac{1}{\alpha}$ are real numbers and (a_n), (b_n) are sequences of positive real numbers. We characterize $T(\alpha, \beta, a_n, b_n)$ the class of all such admissible ideals $\mathcal{I}$.

This accomplishment generalized and extended results from the papers [4, 7, 12, 16], where it is referred that the monotonicity condition of the summand sequence in so-called Olivier's Theorem (see [13]) can be dropped if the convergence of the sequence (na_n) is weakend. In this paper we will study $\mathcal{I}$-convergence mainly in the case when $\mathcal{I}$ stands for $\mathcal{I}_{<q}$, $\mathcal{I}_c^{(q)}$, $\mathcal{I}_{\leq q}$, respectively.

Keywords: $\mathcal{I}$-convergence, convergence of positive series, Olivier's theorem, admissible ideals, convergence exponent

MSC: 40A05, 40A35

1. Introduction

We recall the basic definitions and conventions that will be used throughout the paper. Let $\mathbb{N}$ be the set of all positive integers. A system $\mathcal{I}$, $\emptyset \neq \mathcal{I} \subseteq 2^{\mathbb{N}}$ is called an ideal, provided $\mathcal{I}$ is additive ($A, B \in \mathcal{I}$ implies $A \cup B \in \mathcal{I}$), and hereditary ($A \in \mathcal{I}$, $B \subset A$ implies $B \in \mathcal{I}$). The ideal is called nontrivial if $\mathcal{I} \neq 2^{\mathbb{N}}$. If $\mathcal{I}$ is a nontrivial ideal, then $\mathcal{I}$ is called admissible if it contains the singletons ($\{n\} \in \mathcal{I}$ for every $n \in \mathbb{N}$). The fundamental notation which we shall use is $\mathcal{I}-$convergence introduced in the paper [11] (see also [3] where $\mathcal{I}-$convergence is defined by means of filter-the dual notion to ideal). The notion $\mathcal{I}-$convergence corresponds to the natural generalization of the notion of statistical convergence (see [5, 17]).

Definition 1.1. Let (x_n) be a sequence of real (complex) numbers. We say that the sequence $\mathcal{I}-$converges to a number L, and write $\mathcal{I} - \lim x_n = L$, if for each $\varepsilon > 0$ the set $A_\varepsilon = \{n : |x_n - L| \geq \varepsilon\}$ belongs to the ideal $\mathcal{I}$.

In the following we suppose that $\mathcal{I}$ is an admissible ideal. Then for every sequence (x_n) we have immediately that $\lim_{n\to\infty} x_n = L$ (classic limit) implies that (x_n) also $\mathcal{I}-$converges to a number L. Let $\mathcal{I}_f$ be the ideal of all finite subsets of $\mathbb{N}$. Then $\mathcal{I}_f-$convergence coincides with the usual convergence. Let $\mathcal{I}_d = \{A \subseteq \mathbb{N} : d(A) = 0\}$, where $d(A)$ is the asymptotic density of $A \subseteq \mathbb{N}$ ($d(A) = \lim_{n\to\infty} \frac{\#\{a \leq n : a \in A\}}{n}$, where $\#M$ denotes the cardinality of the set M). Usual $\mathcal{I}_d-$convergence is called statistical convergence. For $0 < q \leq 1$ the class

$$\mathcal{I}_c^{(q)} = \{A \subset \mathbb{N} : \sum_{a\in A} a^{-q} < \infty\}$$

is an admissible ideal and whenever $0 < q < q' < 1$, we get

$$\mathcal{I}_f \subsetneq \mathcal{I}_c^{(q)} \subsetneq \mathcal{I}_c^{q'} \subsetneq \mathcal{I}_c^{(1)} \subsetneq \mathcal{I}_d.$$

The notions the admissible ideal and $\mathcal{I}-$convergence have been developed in several directions and have been used in various parts of mathematics, in particular in

number theory, mathematical analysis and ergodic theory, for example [1, 2, 5, 6, 9–11, 15, 17–19].

Let λ be the convergence exponent function on the power set of $\mathbb{N}$, thus for $A \subset \mathbb{N}$ put

$$\lambda(A) = \inf\Big\{t > 0 : \sum_{a\in A} a^{-t} < \infty\Big\}.$$

If $q > \lambda(A)$ then $\sum_{a\in A} \frac{1}{a^q} < \infty$, and $\sum_{a\in A} \frac{1}{a^q} = \infty$ when $q < \lambda(A)$; if $q = \lambda(A)$, the convergence of $\sum_{a\in A} \frac{1}{a^q}$ is inconclusive. It follows from [14, p. 26, Examp. 113, 114] that the range of λ is the interval $[0, 1]$, moreover for $A = \{a_1 < a_2 < \cdots < a_n < \dots\} \subseteq \mathbb{N}$ the convergence exponent can be calculate by using the following formula

$$\lambda(A) = \limsup_{n\to\infty} \frac{\log n}{\log a_n}.$$

It is easy to see that λ is monotonic, i.e. $\lambda(A) \le \lambda(B)$ whenever $A \subseteq B \subset \mathbb{N}$, furthermore, $\lambda(A \cup B) = \max\{\lambda(A), \lambda(B)\}$ for all $A, B \subset \mathbb{N}$.

2. Overwiew of known results

In this section we mention known results related to the topic of this paper and some other ones we use in the proofs of our results. Recently in [19] was introduced the following classes of subsets of $\mathbb{N}$:

$$\begin{aligned}
\mathcal{I}_{<q} &= \{A \subset \mathbb{N} : \lambda(A) < q\}, \text{ if } 0 < q \le 1,\\
\mathcal{I}_{\le q} &= \{A \subset \mathbb{N} : \lambda(A) \le q\}, \text{ if } 0 \le q \le 1, \text{ and}\\
\mathcal{I}_0 &= \{A \subset \mathbb{N} : \lambda(A) = 0\}.
\end{aligned}$$

Clearly, $\mathcal{I}_{\le 0} = \mathcal{I}_0$. Since $\lambda(A) = 0$ when $A \subset \mathbb{N}$ is finite, then $\mathcal{I}_f = \{A \subset \mathbb{N} : A \text{ is finite}\} \subset \mathcal{I}_0$, moreover, there is proved [19, Th.2] that each class $\mathcal{I}_0$, $\mathcal{I}_{<q}$, $\mathcal{I}_{\le q}$, respectively forms an admissible ideal, except for $\mathcal{I}_{\le 1} = 2^{\mathbb{N}}$.

Proposition 2.1 ([19, Th.1])**.** *Let $0 < q < q' < 1$. Then we have*

$$\mathcal{I}_0 \subsetneq \mathcal{I}_{<q} \subsetneq \mathcal{I}_c^{(q)} \subsetneq \mathcal{I}_{\le q} \subsetneq \mathcal{I}_{<q'} \subsetneq \mathcal{I}_c^{(q')} \subsetneq \mathcal{I}_{\le q'} \subsetneq \mathcal{I}_{<1} \subsetneq \mathcal{I}_c^{(1)} \subsetneq \mathcal{I}_{\le 1} = 2^{\mathbb{N}},$$

and the difference of successive sets is infinite, so equality does not hold in any of the inclusions.

The claim in the following proposition is a trivial fact about preservation of the limit.

Proposition 2.2 ([11, Lemma])**.** *If $\mathcal{I}_1 \subset \mathcal{I}_2$, then $\mathcal{I}_1 - \lim x_n = L$ implies $\mathcal{I}_2 - \lim x_n = L$.*

In [13] L. Olivier proved results so-called Olivier's Theorem about the speed of convergence to zero of the terms of convergent positive series with nonincreasing

terms. Precisely, if (a_n) is a nonincreasing positive sequence and $\sum_{n=1}^{\infty} a_n < \infty$, then $\lim_{n\to\infty} na_n = 0$ (see also [8]). In [16], T. Šalát and V. Toma made the remark that the monotonicity condition in Olivier's Theorem can be dropped if the convergence the sequence (na_n) is weakened by means of the notion of $\mathcal{I}$-convergence (see also [7]). In [12], there is an extension of results in [16] with very nice historical contexts of the object of our research.

Since $0 = \lim_{n\to\infty} na_n = \mathcal{I}_f - \lim na_n$, then the above mentioned Olivier's Theorem can be formulated in the terms of $\mathcal{I}$-convergence as follows:

$$(a_n) \text{ nonincreasing and } \sum_{n=1}^{\infty} a_n < \infty \;\Rightarrow\; \mathcal{I} - \lim na_n = 0,$$

holds for any admissible ideal $\mathcal{I}$ (this assertion is a direct corollary of the facts $\mathcal{I}_f \subseteq \mathcal{I}$ and Proposition 2.2), and providing (a_n) to be a sequence of positive real numbers.

The following simple example

$$a_n = \begin{cases} \frac{1}{n}, & \text{if } n = k^2, (k = 1, 2, \dots) \\ \frac{1}{2^n}, & \text{otherwise,} \end{cases}$$

shows that monotonicity condition of the positive sequence (a_n) can not be in general omitted. This example shows that $\limsup_{n\to\infty} na_n = 1$, thus the ideal $\mathcal{I}_f$ does not have for positive terms the following property

$$\sum_{n=1}^{\infty} a_n < \infty \;\Rightarrow\; \mathcal{I} - \lim na_n = 0. \tag{2.1}$$

The previous example can be strengthened taking $a_n = \frac{\log n}{n}$ if n is square, in such case the sequence (na_n) is not bounded yet. In [16], T. Šalát and V. Toma characterized the class $S(T)$ of all admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ having the property (2.1), for sequences (a_n) of positive real numbers.

They proved that

$$S(T) = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I} \text{ is an admissible ideal such that } \mathcal{I} \supseteq \mathcal{I}_c^{(1)}\}.$$

J. Gogola, M. Mačaj, T. Visnyai in [7] introduced and characterized the class $S_q(T)$ of all admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ for $0 < q \leq 1$ having the property

$$\sum_{n=1}^{\infty} a_n^q < \infty \;\Rightarrow\; \mathcal{I} - \lim na_n = 0, \tag{2.2}$$

providing (a_n) be a positive real sequence. The stronger condition of convergence of positive series requirest the stronger convergence property of the summands as well. They proved

$$S_q(T) = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I} \text{ is an admissible ideal such that } \mathcal{I} \supseteq \mathcal{I}_c^{(q)}\}.$$

Of course, if $q = 1$ then $S_1(T) = S(T)$.

In [12], C. P. Niculescu, G. T. Prăjitură studied the following implication, which is general as (2.1):

$$\sum_{n=1}^{\infty} a_n < \infty \text{ and } \inf_n \frac{n}{b_n} > 0 \;\Rightarrow\; \mathcal{I} - \lim a_n b_n = 0, \tag{2.3}$$

for sequences (a_n), (b_n) of positive real numbers.

They proved that the ideal $\mathcal{I}_d$ fulfills (2.3). In the next section we are going to show that $\mathcal{I}_c^{(1)}$ is the smallest admissible ideal partially ordered by inclusion which also fulfills (2.3).

3. $\mathcal{I}_c^{(q)}$– convergence and convergence of positive series

In this part we introduce and characterize the class of such ideals that fulfill the following implication (3.1). Obviously this class will generalize the results of (2.2) and (2.3). On the other hand, we define the smallest admissible ideal partially ordered by inclusion which fulfills (3.1).

In the sequel we are going to study the ideals $\mathcal{I}$ having the following property:

$$\sum_{n=1}^{\infty} a_n^{\alpha} < \infty \text{ and } 0 < \inf_n \frac{n}{b_n} \leq \sup_n \frac{n}{b_n} < \infty \Rightarrow \mathcal{I} - \lim a_n b_n^{\beta} = 0, \tag{3.1}$$

where $0 < \alpha \leq 1 \leq \beta \leq \frac{1}{\alpha}$ are real numbers and (a_n), (b_n) are positive sequences of real numbers.

We denote by $T(\alpha, \beta, a_n, b_n)$ the class of all admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ having the property (3.1). Obviously $T(1, 1, a_n, n) = S(T)$ and $T(q, 1, a_n, n) = S_q(T)$.

Theorem 3.1. *Let $0 < \alpha \leq 1 \leq \beta \leq \frac{1}{\alpha}$ be real numbers. Then for every positive real sequences (a_n), (b_n) such that*

$$\sum_{n=1}^{\infty} a_n^{\alpha} < \infty \quad \textit{and} \quad \inf_n \frac{n}{b_n} > 0$$

we have

$$\mathcal{I}_c^{(\alpha\beta)} - \lim a_n b_n^{\beta} = 0.$$

Proof. Let $\varepsilon > 0$, put $A_\varepsilon = \{n \in \mathbb{N} : a_n b_n^{\beta} \geq \varepsilon\}$. We proceed by contradiction. Then there exists such $\varepsilon > 0$ that $A_\varepsilon \notin \mathcal{I}_c^{(\alpha\beta)}$, thus

$$\sum_{n \in A_\varepsilon} \frac{1}{n^{\alpha\beta}} = \infty. \tag{3.2}$$

For $n \in A_\varepsilon$ we have

$$a_n^\alpha \geq \varepsilon^\alpha \frac{1}{b_n^{\alpha\beta}} = \varepsilon^\alpha \Big(\frac{n}{b_n}\Big)^{\alpha\beta} \frac{1}{n^{\alpha\beta}} \geq \varepsilon^\alpha \Big(\inf_n \frac{n}{b_n}\Big)^{\alpha\beta} \frac{1}{n^{\alpha\beta}},$$

and so

$$\sum_{n=1}^{\infty} a_n^\alpha \geq \sum_{n\in A_\varepsilon} a_n^\alpha \geq \varepsilon^\alpha \Big(\inf_n \frac{n}{b_n}\Big)^{\alpha\beta} \sum_{n\in A_\varepsilon} \frac{1}{n^{\alpha\beta}}.$$

Using this and the assumption for a sequence (b_n) and (3.2) we get

$$\sum_{n=1}^{\infty} a_n^\alpha = \infty,$$

which is a contradiction. □

If in Theorem 3.1 we put $\alpha = q$ and $\beta = 1$, we can obtain the following corollary.

Corollary 3.2. *For every positive real sequences (a_n), (b_n) such that*

$$\sum_{n=1}^{\infty} a_n^q < \infty \quad \textit{and} \quad \inf_n \frac{n}{b_n} > 0$$

we have

$$\mathcal{I}_c^{(q)} - \lim a_n b_n = 0.$$

Already in the case when $q = 1$ in Corollary 3.2, we get a stronger assertion than given in [12] for the ideal $\mathcal{I}_d$, because of $\mathcal{I}_c^{(1)} \subsetneq \mathcal{I}_d$.

Remark 3.3. Let (a_n), (b_n) be positive real sequences. For special choices α and (b_n) in Corollary 3.2, we can obtain the following:

i) Putting $\alpha = 1$. Then we get: If $\sum_{n=1}^{\infty} a_n < \infty$ and $\inf_n \frac{n}{b_n} > 0$ then $\mathcal{I}_c^{(1)} - \lim a_n b_n = 0$ (which is stronger result as [12, Theorem 5]).

ii) Putting $\alpha = 1$ and $b_n = n$. Then we get: If $\sum_{n=1}^{\infty} a_n < \infty$ then $\mathcal{I}_c^{(1)} - \lim a_n n = 0$ (see [16, Theorem 2.1]).

iii) Putting $\alpha = q$ and $b_n = n$. Then we get: If $\sum_{n=1}^{\infty} a_n^q < \infty$ then $\mathcal{I}_c^{(q)} - \lim a_n n = 0$ (see [7, Lemma 3.1]).

Theorem 3.4. *Let $0 < \alpha \leq 1 \leq \beta \leq \frac{1}{\alpha}$ be real numbers. If for some admissible ideal $\mathcal{I}$ holds*

$$\mathcal{I} - \lim a_n b_n^\beta = 0$$

for every sequences (a_n), (b_n) of positive numbers such that

$$\sum_{n=1}^{\infty} a_n^\alpha < \infty \quad \textit{and} \quad \sup_n \frac{n}{b_n} < \infty,$$

then

$$\mathcal{I}_c^{(\alpha\beta)} \subseteq \mathcal{I}.$$

Proof. Let us assume that for some admissible ideal $\mathcal{I}$ we have $\mathcal{I}-\lim a_n b_n^\beta = 0$ and take an arbitrary set $M \in \mathcal{I}_c^{(\alpha\beta)}$. It is sufficient to prove that $M \in \mathcal{I}$. Since $\mathcal{I}-\lim a_n b_n^\beta = 0$ we have for each $\varepsilon > 0$ the set $A_\varepsilon = \{n \in \mathbb{N} : a_n b_n^\beta \geq \varepsilon\} \in \mathcal{I}$. Since $M \in \mathcal{I}_c^{(\alpha\beta)}$ we have $\sum_{n\in M} \frac{1}{n^{\alpha\beta}} < \infty$. Now we define the sequence a_n as follows:

$$a_n = \begin{cases} \frac{1}{n^\beta}, & \text{if } n \in M, \\ \frac{1}{2^n}, & \text{if } n \notin M. \end{cases}$$

Obviously the sequence (a_n) fulfills the premises of the theorem as $a_n > 0$ and

$$\sum_{n=1}^{\infty} a_n^\alpha = \sum_{n\in M} \Big(\frac{1}{n^\beta}\Big)^\alpha + \sum_{n\notin M} \Big(\frac{1}{2^n}\Big)^\alpha \leq \sum_{n\in M} \frac{1}{n^{\alpha\beta}} + \sum_{n=1}^{\infty} \Big(\frac{1}{2^\alpha}\Big)^n < \infty.$$

Hence $a_n n^\beta = 1$ for $n \in M$ and so for each $n \in M$ we have

$$a_n b_n^\beta = a_n n^\beta \Big(\frac{b_n}{n}\Big)^\beta = \Big(\frac{b_n}{n}\Big)^\beta \geq \frac{1}{\big(\sup_n \frac{n}{b_n}\big)^\beta} > 0.$$

Denote by $\varepsilon(\beta) = \big(\sup_n \frac{n}{b_n}\big)^{-\beta} > 0$ and preceding considerations give us

$$M \subset A_{\varepsilon(\beta)} \in \mathcal{I}.$$

Thus $M \in \mathcal{I}$, what means $\mathcal{I}_c^{(\alpha\beta)} \subseteq \mathcal{I}$. □

The characterization of the class $T(\alpha, \beta, a_n, b_n)$ is the direct consequence of Theorem 3.1 and Theorem 3.4.

Theorem 3.5. *Let $0 < \alpha \leq 1 \leq \beta \leq \frac{1}{\alpha}$ be real numbers and (a_n), (b_n) be sequences of positive real numbers. Then the class $T(\alpha, \beta, a_n, b_n)$ consists of all admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ such that $\mathcal{I} \supseteq \mathcal{I}_c^{(\alpha\beta)}$.*

For special choices α, β and (b_n) in Theorem 3.5 we can get the following.

Corollary 3.6. *Let $0 < q \leq 1$ be a real number and (a_n) be positive real sequences having the properties*

$$\sum_{n=1}^{\infty} a_n^q < \infty.$$

Then we have

i) $T(q, 1, a_n, n) = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I}$ *is admissible ideal such that* $\mathcal{I} \supseteq \mathcal{I}_c^{(q)}\} = S_q(T)$,

ii) $T(1, 1, a_n, n) = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I}$ *is admissible ideal such that* $\mathcal{I} \supseteq \mathcal{I}_c^{(1)}\} = S(T)$.

4. $\mathcal{I}_{<q}-$ and $\mathcal{I}_{\leq q}-$convergence and convergence of series

In this section we will study the admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ having the special property (4.1) and (4.3), respectively.

$$\sum_{n=1}^{\infty} a_n^{q_k} < \infty \text{ for every } k \text{ and } 0 < \inf_n \frac{n}{b_n} \leq \sup_n \frac{n}{b_n} < \infty \Rightarrow \mathcal{I} - \lim a_n b_n = 0, \quad (4.1)$$

where (q_k) is a strictly decreasing sequence which is convergent to q, $0 \leq q < 1$ and (a_n), (b_n) are sequences of positive real numbers.

Denote by $T_q^{q_k}(a_n, b_n)$ the class of all admissible ideals $\mathcal{I}$ having the property (4.1).

Theorem 4.1. *Let $0 \leq q < 1$ and (q_k) be a strictly decreasing sequence which is convergent to q. Then for positive real sequences (a_n), (b_n) such that holds*

$$\sum_{n=1}^{\infty} a_n^{q_k} < \infty, \text{ for every } k \text{ and } \inf_n \frac{n}{b_n} > 0,$$

we have

$$\mathcal{I}_{\leq q} - \lim a_n b_n = 0.$$

Proof. Again, we proceed by contradiction. Put $A_\varepsilon = \{n \in \mathbb{N} : a_n b_n \geq \varepsilon\}$. Then there exists such $\varepsilon > 0$ that $A_\varepsilon \notin \mathcal{I}_{\leq q}$, thus $\lambda(A_\varepsilon) > q$. Hence there exists such $i \in \mathbb{N}$, that $q < q_{k_i} < \lambda(A_\varepsilon)$, and so we get

$$\sum_{n \in A_\varepsilon} \frac{1}{n^{q_{k_i}}} = \infty. \quad (4.2)$$

For $n \in A_\varepsilon$ we have

$$a_n^{q_{k_i}} \geq \varepsilon^{q_{k_i}} \frac{1}{b_n^{q_{k_i}}} = \varepsilon^{q_{k_i}} \Big(\frac{n}{b_n}\Big)^{q_{k_i}} \frac{1}{n^{q_{k_i}}} \geq \varepsilon^{q_{k_i}} \Big(\inf_n \frac{n}{b_n}\Big)^{q_{k_i}} \frac{1}{n^{q_{k_i}}},$$

therefore

$$\sum_{n=1}^{\infty} a_n^{q_{k_i}} \geq \sum_{n \in A_\varepsilon} a_n^{q_{k_i}} \geq \varepsilon^{q_{k_i}} \Big(\inf_n \frac{n}{b_n}\Big)^{q_{k_i}} \sum_{n \in A_\varepsilon} \frac{1}{n^{q_{k_i}}}.$$

Using this and the assumption for a sequence (b_n) and (4.2) we get

$$\sum_{n=1}^{\infty} a_n^{q_{k_i}} = \infty,$$

what is a contradiction. □

Theorem 4.2. *Let $0 \leq q < 1$ and (q_k) be a strictly decreasing sequence which is convergent to q. If for some admissible ideal $\mathcal{I}$ holds*

$$\mathcal{I} - \lim a_n b_n = 0$$

for every sequences (a_n), (b_n) of positive numbers such that

$$\sum_{n=1}^{\infty} a_n^{q_k} < \infty, \text{ for every } k \text{ and } \sup_n \frac{n}{b_n} < \infty,$$

then

$$\mathcal{I}_{\leq q} \subseteq \mathcal{I}.$$

Proof. Let us assume that for any admissible ideal $\mathcal{I}$ we have $\mathcal{I} - \lim a_n b_n = 0$ and take an arbitrary set $M \in \mathcal{I}_{\leq q}$. It is sufficient to prove that $M \in \mathcal{I}$. Since $M \in \mathcal{I}_{\leq q}$ we have $\lambda(M) \leq q$ and so for each $q_k > q$ we get

$$\sum_{n \in M} \frac{1}{n^{q_k}} < \infty.$$

Moreover $\mathcal{I} - \lim a_n b_n = 0$ and so for each $\varepsilon > 0$ the set $A_\varepsilon = \{n \in \mathbb{N} : a_n b_n \geq \varepsilon\} \in \mathcal{I}$. Define the sequence (a_n) as follows:

$$a_n = \begin{cases} \frac{1}{n}, & \text{if } n \in M, \\ \frac{1}{2^n}, & \text{if } n \notin M. \end{cases}$$

The sequence (a_n) fulfills the premises of the theorem, $a_n > 0$ and for each q_k we obtain

$$\sum_{n=1}^{\infty} a_n^{q_k} = \sum_{n \in M} \frac{1}{n^{q_k}} + \sum_{n \notin M} \left(\frac{1}{2^n}\right)^{q_k} \leq \sum_{n \in M} \frac{1}{n^{q_k}} + \sum_{n=1}^{\infty} \left(\frac{1}{2^{q_k}}\right)^n < \infty.$$

Now $a_n n = 1$ for $n \in M$. Therefore for each $n \in M$ we have

$$a_n b_n = a_n n \Big(\frac{b_n}{n}\Big) = \frac{b_n}{n} \geq \frac{1}{\sup_n \frac{n}{b_n}} > 0.$$

Denote by $\varepsilon = \left(\sup_n \frac{n}{b_n}\right)^{-1} > 0$ we have

$$M \subset A_\varepsilon \in \mathcal{I}.$$

Thus $M \in \mathcal{I}$, what means $\mathcal{I}_{\leq q} \subseteq \mathcal{I}$. □

The above mentioned results (Theorem 4.1 and Theorem 4.2) allow us to give a characterization for the class $T_q^{q_k}(a_n, b_n)$.

Theorem 4.3. *Let $0 \leq q < 1$ and (q_k) be a strictly decreasing sequence which converges to q. Let (a_n), (b_n) be positive real sequences. Then the class $T_q^{q_k}(a_n, b_n)$ consists of all admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ such that $\mathcal{I} \supseteq \mathcal{I}_{\leq q}$.*

Let us consider the following property and pronounce for it analogical results as above.

$$\sum_{n=1}^{\infty} a_n^{q_k} < \infty \text{ for some } k \text{ and } 0 < \inf_n \frac{n}{b_n} \leq \sup_n \frac{n}{b_n} < \infty \Rightarrow \mathcal{I} - \lim a_n b_n = 0, \quad (4.3)$$

where (q_k) is a strictly increasing sequence of positive numbers which is convergent to q, $0 < q \leq 1$ and (a_n), (b_n) are sequences of positive real numbers.

Denote by $T^q_{q_k}(a_n, b_n)$ the class of all admissible ideals $\mathcal{I}$ having the property (4.3).

Theorem 4.4. *Let $0 < q \leq 1$ and (q_k) be a strictly increasing sequence of positive numbers which is convergent to q. Then for positive real sequences (a_n), (b_n) such that holds*

$$\sum_{n=1}^{\infty} a_n^{q_{k_0}} < \infty, \text{ for some } k_0 \in \mathbb{N} \text{ and } \inf_n \frac{n}{b_n} > 0,$$

we have

$$\mathcal{I}_{<q} - \lim a_n b_n = 0.$$

Proof. Again, we proceed by contradiction. Then there exists $\varepsilon > 0$ such that $A_\varepsilon = \{n \in \mathbb{N} : a_n b_n \geq \varepsilon\} \notin \mathcal{I}_{<q}$, thus $\lambda(A_\varepsilon) \geq q$. For each $k \in \mathbb{N}$ (as well for k_0) we have $q_k < q \leq \lambda(A_\varepsilon)$, and so

$$\sum_{n \in A_\varepsilon} \frac{1}{n^{q_k}} = \infty. \quad (4.4)$$

Further the proof continues by the same way as it was outlined in Theorem 4.1. □

Theorem 4.5. *Let $0 < q \leq 1$ and (q_k) be a strictly increasing sequence of positive numbers which is convergent to q. If for some admissible ideal $\mathcal{I}$ holds*

$$\mathcal{I} - \lim a_n b_n = 0$$

for every sequences (a_n), (b_n) of positive numbers such that

$$\sum_{n=1}^{\infty} a_n^{q_{k_0}} < \infty \text{ for some } k_0 \in \mathbb{N} \text{ and } \sup_n \frac{n}{b_n} < \infty,$$

then

$$\mathcal{I}_{<q} \subseteq \mathcal{I}.$$

Proof. Let us assume that for any admissible ideal $\mathcal{I}$ we have $\mathcal{I} - \lim a_n b_n = 0$ and take an arbitrary $M \in \mathcal{I}_{<q}$. It is sufficient to prove that $M \in \mathcal{I}$. Since $M \in \mathcal{I}_{<q}$ we have $\lambda(M) < q$ and so there exists a sufficiently large $k_0 \in \mathbb{N}$ such that $\lambda(M) < q_{k_0} < q$. So

$$\sum_{n \in M} \frac{1}{n^{q_{k_0}}} < \infty.$$

Again, the proof continues by the same way as it was outlined in Theorem 4.2. □

The above results (Theorem 4.4 and Theorem 4.5) allow us to give a characterization for the class $T^q_{q_k}(a_n, b_n)$.

Theorem 4.6. *Let $0 < q \leq 1$ and (q_k) be a strictly increasing sequence of positive numbers which converges to q. Let (a_n), (b_n) be positive real sequences. Then the class $T^q_{q_k}(a_n, b_n)$ consists of all admissible ideals $\mathcal{I} \subset 2^{\mathbb{N}}$ such that $\mathcal{I} \supseteq \mathcal{I}_{<q}$.*

5. Summary and scheme of main results

Let (a_n), (b_n) be fix sequences of positive real numbers having the appropriate property (3.1), (4.1) and (4.3), respectively. Denote in short classes given above $T(\alpha, \beta, a_n, b_n) = T(\alpha, \beta)$, $T^{q_k}_q(a_n, b_n) = T^{q_k}_q$ and $T^q_{q_k}(a_n, b_n) = T^q_{q_k}$. Then we have

i) for $0 < \alpha \leq 1 \leq \beta \leq \frac{1}{\alpha}$,

$$T(\alpha, \beta) = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I} \text{ is admissible ideal such that } \mathcal{I} \supseteq \mathcal{I}_c^{(\alpha\beta)}\},$$

ii) for $1 \geq q_k > q \geq 0$ $(k = 1, 2 \dots)$, $q_k \downarrow q$ as $k \to \infty$,

$$T^{q_k}_q = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I} \text{ is admissible ideal such that } \mathcal{I} \supseteq \mathcal{I}_{\leq q}\},$$

iii) for $0 < q_k < q \leq 1$ $(k = 1, 2 \dots)$, $q_k \uparrow q$ as $k \to \infty$,

$$T^q_{q_k} = \{\mathcal{I} \subset 2^{\mathbb{N}} : \mathcal{I} \text{ is admissible ideal such that } \mathcal{I} \supseteq \mathcal{I}_{<q}\}.$$

For special cases the following scheme shows the smallest(minimal) admissible ideals partially ordered by inclusion which belong to the classes in the second line.

$$\begin{array}{ccccccccccccc}
\mathcal{I}_0 & \subsetneq & \mathcal{I}_c^{(\alpha\beta)} & \subsetneq & \mathcal{I}_{<q} & \subsetneq & \mathcal{I}_c^{(q)} & \subsetneq & \mathcal{I}_{\leq q} & \subsetneq & \mathcal{I}_{<1} & \subsetneq & \mathcal{I}_c^{(1)} \\
\updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\
T_0^{q_k} & \supsetneq & \underset{if\ \alpha\beta<q}{T(\alpha,\beta)} & \supsetneq & T^q_{q_k} & \supsetneq & \underset{if\ \alpha\beta=q}{T(\alpha,\beta)} & \supsetneq & T^{q_k}_q & \supsetneq & T^1_{q_k} & \supsetneq & \underset{if\ \alpha\beta=1}{T(\alpha,\beta)}
\end{array}$$

References

[1] V. Baláž, J. Gogola, T. Visnyai: *$\mathcal{I}_c^{(q)}$-convergence of arithmetical functions*, J. Number Theory 183 (2018), pp. 74–83,
doi: http://dx.doi.org/10.1016/j.jnt.2017.07.006.

[2] V. Baláž, O. Strauch, T. Šalát: *Remarks on several types of convergence of bounded sequences*, Acta Math. Univ. Ostraviensis 14 (2006), pp. 3–12.

[3] N. Burbaki: *Éléments de Mathématique, Topologie Générale Livre III, (Russian translation) Obščaja topologija. Osnovnye struktury*, Moskow: Nauka, 1968.

[4] A. Fasaint, G. Grekos, L. Mišík: *Some generalizations of Olivier's Theorem*, Math. Bohemica 141 (2016), pp. 483–494,
doi: https://doi.org/10.21136/MB.2016.0057-15.

[5] H. Fast: *Sur la convergence statistique*, Colloq. Math. 2 (1951), pp. 241–244.

[6] H. Furstenberg: *Recurrence in Ergodic Theory and Combinatorial Number Theory*, Princeton: Princeton University Press, 1981.

[7] J. Gogola, M. Mačaj, T. Visnyai: *On $\mathcal{I}_c^{(q)}$- convergence*, Ann. Math. Inform. 38 (2011), pp. 27–36.

[8] K. Knopp: *Theorie und Anwendung unendlichen Reisen*, Berlin: Princeton University Press, 1931.

[9] P. Kostyrko, M. Mačaj, T. Šalát, M. Sleziak: *$\mathcal{I}-$convergence and extremal $\mathcal{I}-$limit poits*, Math. Slovaca 55 (2005), pp. 443–464.

[10] P. Kostyrko, M. Mačaj, T. Šalát, O. Strauch: *On statistical limit points*, Proc. Amer. Math. Soc. 129 (2001), pp. 2647–2654,
doi: https://doi.org/10.1090/S0002-9939-00-05891-3.

[11] P. Kostyrko, T. Šalát, W. Wilczyński: *$\mathcal{I}$- convergence*, Real Anal. Exchange 26 (2000), pp. 669–686.

[12] C. P. Niculescu, G. T. Prăjitură: *Some open problems concerning the convergence of positive series*, Ann. Acad. Rom. Sci. Ser. Math. Appl. 6.1 (2014), pp. 85–99.

[13] L. Olivier: *Remarques sur les séries infinies et leur convergence*, J. Reine Angew. Math. 2 (1827), pp. 31–44,
doi: https://doi.org/10.1515/crll.1827.2.31.

[14] G. Pólya, G. Szegő: *Problems and Theorems in Analysis I*. Berlin Heidelberg New York: Springer-Verlag, 1978.

[15] J. Renling: *Applications of nonstandard analysis in additive number theory*, Bull. of Symbolic Logic 6 (2000), pp. 331–341.

[16] T. Šalát, V. Toma: *A classical Olivier's theorem and statistical convergence*, Ann. Math. Blaise Pascal 1 (2001), pp. 305–313.

[17] I. J. Schoenberg: *The Integrability of Certain Functions and Related Summability Methods*, Amer. Math. Monthly 66 (1959), pp. 361–375.

[18] H. Steinhaus: *Sur la convergence ordinaire et la convergence asymptotique*, Colloq. Math. 2 (1951), pp. 73–74.

[19] J. T. Tóth, F. Filip, J. Bukor, L. Zsilinszky: *$\mathcal{I}_{<q}-$ and $\mathcal{I}_{\leq q}-$convergence of arithmetic functions*, Period. Math. Hung. (2020), to appear,
doi: https://doi.org/10.1007/s10998-020-00345-y.

Annales Mathematicae et Informaticae
***52** (2020) pp. 31–37*
DOI: https://doi.org/10.33039/ami.2020.06.002
URL: https://ami.uni-eszterhazy.hu

Integer sequences and ellipse chains inside a hyperbola

Hacène Belbachir[a], László Németh[b], Soumeya Merwa Tebtoub[a]

[a]Department of Mathematics, RECITS Laboratory, USTHB, Algiers, Algeria
hbelbachir@usthb.dz and hacenebelbachir@gmail.com
stebtoub@usthb.dz and tebtoubsoumeya@gmail.com

[b]Institute of Mathematics, University of Sopron, Sopron, Hungary
and associate member of RECITS Laboratory, USTHB
nemeth.laszlo@uni-sopron.hu

Submitted: May 12, 2020
Accepted: June 24, 2020
Published online: June 30, 2020

Abstract

We propose an extension to the work of Lucca [Giovanni Lucca, Integer sequences and circle chains inside a hyperbola, *Forum Geometricorum*, Volume 19. 2019, 11–16]. Our goal is to examine chains of ellipses inside a hyperbola, and we derive recurrence relations of centers and minor (major) axes of the ellipse chains. We also determine conditions for these recurrence sequences to consist of integer numbers.

Keywords: Ellipse chains, circle chains, hyperbola, integer sequences.

MSC: 52C26, 11B37.

1. Introduction

Let us consider the hyperbola $\mathcal{H}$ with the canonical equation

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1, \tag{1.1}$$

and foci $(\pm c, 0)$, where a and b are positive real numbers and $c^2 = a^2 + b^2$. Lucca [1] examined a tangential chain of circles inside the branch $x > 0$ of the hyperbola so that the i-th circle with center $(x_i, 0)$ and radius r_i is tangent to the hyperbola and to the preceding and succeeding circles labelled by indexes $i-1$ and $i+1$, respectively. He showed that in case of certain ratios $\frac{b}{a}$ the sequences $\{\frac{x_i}{x_0}\}_{i=0}^{\infty}$ and $\{\frac{r_i}{r_0}\}_{i=0}^{\infty}$ are integers.

In our article, we extend Lucca's work. We define and examine a special chains of ellipses inside the branch $x > 0$ of the hyperbola, when the ratio of the minor and major axis is fixed. It is a natural extension of Lucca's circle chains. We describe the recurrence relations of sequences of centers, major and minor axes, which determine another type of proof to give integer sequences. Therefore, we are able to provide more integer sequences than in case of Lucca's circle chains.

2. Ellipse chains inside a branch of hyperbola

Let us define a chain of ellipses with the following properties:

- The center of each ellipse lies on the x-axis, inside the branch $x > 0$ of the hyperbola (1.1), the semi-axes are parallel to the coordinate lines. More precisely, the canonical equation of the i-th ellipse centered at point $(u_i, 0)$ $(u_i > 0)$ is

$$\frac{(x-u_i)^2}{\alpha_i^2} + \frac{y^2}{\beta_i^2} = 1, \tag{2.1}$$

  where $2\alpha_i > 0$ is the width and $2\beta_i > 0$ is the height of the ellipse (Figure 1). If $\alpha_i > \beta_i$, then the focal axis of the i-th ellipse is coincident with the x-axis, if $\alpha_i < \beta_i$, then it is parallel to the ordinate axis, and if $\alpha_i = \beta_i$, then the i-th ellipse is a circle. In the figures $\alpha_i < \beta_i$.

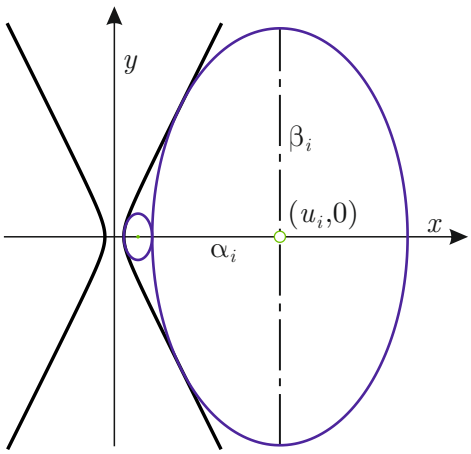


Figure 1: An ellipse chain inside a hyperbola

- The ellipses (2.1) are tangent to the hyperbola (1.1).
- The first ellipse (Figure 2) is tangent (and do not intersect at any points) to the hyperbola at its vertex A having coordinates $(a, 0)$, so:

$$u_0 = a + \alpha_0.$$

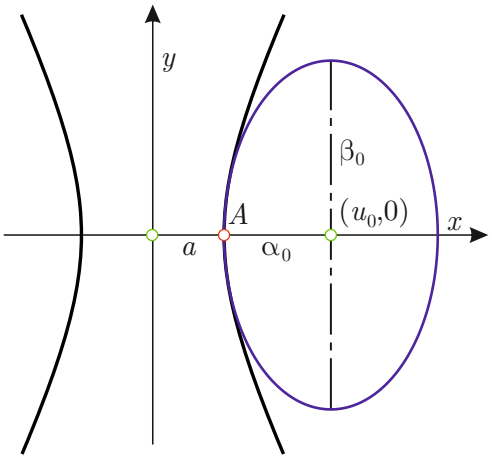


Figure 2: First ellipse of a chain

- The curvature in case of the hyperbola must not be bigger than the curvature in case of the first ellipse at A, otherwise they are not only tangent to each other at A, but also the ellipse intersects the hyperbola at two other points. Thus,

$$\frac{\alpha_0 b^2}{a} \geq \beta_0^2, \tag{2.2}$$

where $\frac{a}{b^2}$ and $\frac{\alpha_0}{\beta_0^2}$ are, respectively, the curvatures of the hyperbola and the first ellipse at point A.

- In order that the first ellipse provides the best touching to the hyperbola at A, we have to require the same curvature of the ellipse and the hyperbola at A. That is why we restrict inequality (2.2) to equation

$$\frac{\alpha_0 b^2}{a} = \beta_0^2.$$

- The ellipses are mutually tangent. It means that the i-th ellipse is tangent to the $(i-1)$-th ellipse and to the $(i+1)$-th ellipse, so we have

$$u_i - u_{i-1} = \alpha_i + \alpha_{i-1}. \tag{2.3}$$

- We pose $\frac{\beta_i}{\alpha_i} = m$, where $m = \frac{\beta_0}{\alpha_0} = \frac{\sqrt{\frac{\alpha_0 b^2}{a}}}{\alpha_0} = \frac{b}{\sqrt{\alpha_0 a}}$. Hence the ellipses are similar to each other.

In order to achieve our goal, we consider the system of equations

$$\begin{cases} \dfrac{x^2}{a^2} - \dfrac{y^2}{b^2} = 1, \\ \dfrac{(x-u_i)^2}{\alpha_i^2} + \dfrac{y^2}{\beta_i^2} = 1. \end{cases} \tag{2.4}$$

Now we give the general relations between u_i and α_i. From (2.4) we have

$$\left(\frac{\beta_i^2}{\alpha_i^2} + \frac{b^2}{a^2}\right) x^2 - 2\frac{\beta_i^2}{\alpha_i^2} u_i x + \frac{\beta_i^2}{\alpha_i^2} u_i^2 = b^2 + \beta_i^2, \tag{2.5}$$

due to the tangency condition the discriminant Δ of (2.5) must be zero so:

$$\frac{\Delta}{4} = \frac{\beta_i^4}{\alpha_i^4} u_i^2 - \left(\frac{\beta_i^2}{\alpha_i^2} + \frac{b^2}{a^2}\right)\left(\frac{\beta_i^2}{\alpha_i^2} u_i^2 - b^2 - \beta_i^2\right) = 0.$$

Then we have

$$u_i^2 = \left(\frac{a^2}{b^2} + \frac{\alpha_i^2}{\beta_i^2}\right)\left(b^2 + \beta_i^2\right).$$

Since $\frac{\beta_i}{\alpha_i} = m$, then

$$u_i^2 = \left(\frac{a^2}{b^2} + \frac{1}{m^2}\right)\left(b^2 + m^2\alpha_i^2\right) \tag{2.6}$$

and

$$u_{i-1}^2 = \left(\frac{a^2}{b^2} + \frac{1}{m^2}\right)\left(b^2 + m^2\alpha_{i-1}^2\right). \tag{2.7}$$

By subtracting (2.7) from (2.6) and by remembering (2.3), we obtain

$$\begin{cases} u_i + u_{i-1} = m^2\left(\dfrac{a^2}{b^2} + \dfrac{1}{m^2}\right)(\alpha_i - \alpha_{i-1}), \\ u_i - u_{i-1} = \alpha_i + \alpha_{i-1}. \end{cases}$$

Since $m^2 = \frac{b^2}{a\alpha_0}$, we gain

$$\begin{cases} u_i + u_{i-1} = \left(1 + \dfrac{a}{\alpha_0}\right)(\alpha_i - \alpha_{i-1}), \\ u_i - u_{i-1} = \alpha_i + \alpha_{i-1}. \end{cases} \tag{2.8}$$

In order to give the expression of u_i and α_i, we have to solve the system (2.8), after some algebraical steps we obtain

$$u_i = \left(2\frac{\alpha_0}{a} + 1\right) u_{i-1} + 2\left(1 + \frac{\alpha_0}{a}\right) \alpha_{i-1}, \tag{2.9}$$

$$\alpha_i = 2\frac{\alpha_0}{a} u_{i-1} + \left(2\frac{\alpha_0}{a} + 1\right) \alpha_{i-1}. \tag{2.10}$$

Since $\beta_i = m\alpha_i$, we have

$$\beta_i = \frac{b}{a\sqrt{\alpha_0 a}}\left(2\alpha_0 u_{i-1} + (2\alpha_0 + a)\, \alpha_{i-1}\right). \tag{2.11}$$

We should notice that the case $m = 1$ (the ellipses are circles) provides $\alpha_0 = \beta_0 = \frac{b^2}{a}$, so we are in [1].

We can represent the expression of u_i and α_i in matrix form as follows:

$$\begin{pmatrix} u_i \\ \alpha_i \end{pmatrix} = \begin{pmatrix} 2\frac{\alpha_0}{a} + 1 & 2\left(1 + \frac{\alpha_0}{a}\right) \\ 2\frac{\alpha_0}{a} & 2\frac{\alpha_0}{a} + 1 \end{pmatrix} \begin{pmatrix} u_{i-1} \\ \alpha_{i-1} \end{pmatrix}.$$

2.1. Sequences

In this paragraph, we give recurrence relations of centers and minor (major) axes of the ellipse chains.

Theorem 2.1. *The sequences $\{u_i\}$, $\{\alpha_i\}$ and $\{\beta_i\}$ are the same second-order linear homogeneous recurrence sequences*

$$\ell_i = 2\left(2\frac{\alpha_0}{a} + 1\right) \ell_{i-1} - \ell_{i-2} \qquad (i \geq 2), \tag{2.12}$$

with initial values $\alpha_0 \in \mathbb{R}^+$, $\beta_0 = \frac{b\alpha_0}{\sqrt{a\alpha_0}}$, $u_0 = a + \alpha_0$, $\alpha_1 = \frac{\alpha_0}{a}(3a + 4\alpha_0)$, $\beta_1 = \frac{b\sqrt{\alpha_0}(3\,a + 4\,\alpha_0)}{a^{3/2}}$ and $u_1 = \frac{a^2 + 5\,\alpha_0 a + 4\,{\alpha_0}^2}{a}$.

Proof. From the sum of (2.8) we have

$$2u_i = \left(2 + \frac{a}{\alpha_0}\right) \alpha_i - \frac{a}{\alpha_0}\alpha_{i-1}$$

and

$$2u_{i-1} = \left(2 + \frac{a}{\alpha_0}\right) \alpha_{i-1} - \frac{a}{\alpha_0}\alpha_{i-2}.$$

The sum of them combined again by (2.8) after some calculation yields

$$\alpha_i = 2\left(2\frac{\alpha_0}{a} + 1\right) \alpha_{i-1} - \alpha_{i-2}.$$

Similarly, $u_i = 2\left(2\frac{\alpha_0}{a} + 1\right) u_{i-1} - u_{i-2}$. Moreover, since $\beta_i = m\alpha_i$, the recurrence is true for β_i. The initial values come from the equations (2.9)–(2.11). □

Let $\overline{u}_i = \frac{u_i}{u_0}$, $\overline{\alpha}_i = \frac{\alpha_i}{\alpha_0}$, and $\overline{\beta}_i = \frac{\beta_i}{\beta_0}$. Then we obtain the following theorem as a corollary of Theorem 2.1.

Theorem 2.2. *The sequences $\{\overline{u}_i\}$, $\{\overline{\alpha}_i\}$ and $\{\overline{\beta}_n\}$ are second-order linear homogeneous recurrence sequences* (2.12) *with initial values $\overline{u}_0 = \overline{\alpha}_0 = \overline{\beta}_0 = 1$, $\overline{u}_1 = 1 + \frac{4\alpha_0}{a}$, $\overline{\alpha}_1 = \overline{\beta}_1 = 3 + \frac{4\alpha_0}{a}$.*

Corollary 2.3. *The sequences $\{\overline{\alpha}_i\}$ and $\{\overline{\beta}_i\}$ are the same.*

Circle chains defined by Lucca [1] are the special cases $\alpha_i = \beta_i = r_i$ of our chains. The following corollary gives the recurrence relation for this unique case.

Corollary 2.4. *If $\alpha_0 = \frac{b^2}{a}$, then the sequences $\{\overline{u}_n\}$, $\{\overline{\alpha}_n\}$ are second-order linear homogeneous recurrence sequences*

$$\ell_i = 2\left(2\frac{b^2}{a^2} + 1\right)\ell_{i-1} - \ell_{i-2} \qquad (i \geq 2), \tag{2.13}$$

with initial values $\overline{u}_0 = \overline{\alpha}_0 = 1$, $\overline{u}_1 = 1 + \frac{4b^2}{a^2}$, $\overline{\alpha}_1 = 3 + \frac{4b^2}{a^2}$.

2.2. Integer sequences

In this paragraph, we determine conditions to relate the ellipse chains with integer sequences.

Theorem 2.5. *In case of any positive integer k, if*

$$\alpha_0 = k\frac{a}{4},$$

then the sequences $\{\overline{u}_i\}_{i\in\mathbb{N}}$ and $\{\overline{\alpha}_i\}_{i\in\mathbb{N}}$ are integer sequences, and their recurrences are

$$\ell_i = (k+2)\,\ell_{i-1} - \ell_{i-2} \qquad (i \geq 2), \tag{2.14}$$

with initial values $\overline{u}_0 = \overline{\alpha}_0 = 1$, $\overline{u}_1 = 1 + k$, $\overline{\alpha}_1 = 3 + k$.

Proof. Theorem 2.2 shows, that the first two items of the sequences are integers if $k = \frac{4\alpha_0}{a}$ is integer. Then the first coefficient of recurrence relation (2.12) is $k + 2$, which guarantees that all the other items of the sequences are integers. □

Example 2.6. We give now some examples of integer sequences that can be obtained for different values of k.

- For $k = 1$;
  The sequence $\{\overline{\alpha}\} = \{1, 4, 11, 29, 76, 199, \ldots\}$ which corresponds to the bisection of Lucas sequence is classified in The On-Line Encyclopedia of Integer Sequences (OEIS [2]) as $A002878$.

- For $k = 2$;
  The sequence $\{\overline{u}\} = \{1, 3, 11, 41, 153, 571, 2131, \ldots\}$ is classified in OEIS as $A001835$.

- For $k = 3$;
  The sequence $\{\overline{u}\} = \{1, 4, 19, 91, 436, 2089, \ldots\}$ appears in OEIS as $A004253$. The sequence $\{\overline{\alpha}\} = \{1, 6, 29, 139, 666, 3191, \ldots\}$ which corresponds to the Chebyshev even index U-polynomials evaluated at $\frac{\sqrt{7}}{2}$ is classified in OEIS as $A030221$.

- For $k = 4$;
  The sequence $\{\overline{\alpha}\} = \{1, 7, 41, 239, 1393, 8119, \ldots\}$ which corresponds to the Newman, Shanks–Williams (NSW) numbers is classified in OEIS as $A002315$.

In case of Lucca's circle chains for integer sequences when $\alpha_0 = \frac{ka}{4} = \frac{b^2}{a}$ we obtain the following corollary.

Corollary 2.7. *If $r_i = \alpha_i = \beta_i$ and the ratio $\frac{b}{a}$ is given by*

$$\frac{\sqrt{k}}{2} = \frac{b}{a}, \qquad k = 1, 2, \ldots,$$

then the sequences $\{\overline{u}_i\}$, $\{\overline{r}_i\}$ are integer sequences. Their recurrences are

$$\ell_n = (k + 4)\,\ell_{n-1} - \ell_{n-2} \qquad (n \geq 2),$$

with initial values $\overline{u}_0 = \overline{\alpha}_0 = 1$, $\overline{u}_1 = 1 + k$, $\overline{\alpha}_1 = 3 + k$.

Comparing Corollary 2.7 and Lucca's similar main theorem, we find that Corollary 2.7 contains more integer sequences. Only if k is a square number, then the sequence appears in Lucca's theorem. We also mention that for relatively small k a huge number of integer sequences are arising in OEIS [2].

Acknowledgements. For H. B. and S. M. T. this work was supported by the grant of DGRSDT, number C0656701. For L. N. this work has been made in the frame of the "EFOP-3.6.1-16-2016-00018 - Improving the role of the research + development + innovation in the higher education through institutional developments assisting intelligent specialization in Sopron and Szombathely".

We would like to thank the anonymous referee for carefully reading the manuscript and for his/her useful suggestions and improvements.

References

[1] G. Lucca: *Integer sequences and circle chains inside a hyperbola*, Forum Geometricorum 19 (2019), pp. 11–16.

[2] N. J. A. Sloane: *The On-Line Encyclopedia of Integer Sequences*, https://oeis.org.

Annales Mathematicae et Informaticae
52 *(2020) pp. 39–55*
DOI: https://doi.org/10.33039/ami.2020.02.001
URL: https://ami.uni-eszterhazy.hu

On two four term arithmetic progressions with equal product

Andrew Bremner

School of Mathematics and Statistical Sciences
Arizona State University
bremner@asu.edu

Submitted: November 6, 2018
Accepted: February 8, 2020
Published online: February 11, 2020

This paper is dedicated to Richard K. Guy in his 104^{th} year in honour of his many and varied contributions to mathematics.

Abstract

We investigate when two four-term arithmetic progressions have an equal product of their terms. This is equivalent to studying the (arithmetic) geometry of a non-singular quartic surface. It turns out that there are many polynomial parametrizations of such progressions, and it is likely that there exist polynomial parametrizations of every positive degree. We find *all* such parametrizations for degrees 1 to 4, and give examples of parametrizations for degrees 5 to 10.

1. Introduction

The problem considered in this paper was first drawn to my attention by Richard Guy and Alex Fink, who asked which n-term arithmetic progressions can have equal product of their terms. For example, when $n = 5$, Fink observed that the two progressions

$$(4 + t^5, 3 + 2t^5, 2 + 3t^5, 1 + 4t^5, 5t^5), \quad (t + 4t^6, 2t + 3t^6, 3t + 2t^6, 4t + t^6, 5t)$$

have equal product. There is some literature on the subject. Gabovich [5] gives infinitely many examples of two such 4-term progressions. For general n, the only

known example of two arithmetic progressions with equal product of terms is given by

$$(n+1)(n+2)\dots(2n) = 2 \cdot 6 \cdot 10 \cdot \dots \cdot (4n-2);$$

in fact, Saradha, Shorey and Tijdeman [9, 10] show that other than this example, solutions in positive integers $x > y$, $n > 2$, to

$$x(x+d_1)...(x+(n-1)d_1) = y(y+d_2)...(y+(n-1)d_2),$$

for fixed integers $0 < d_1 < d_2$, are finite in number, and can be effectively determined. Choudhry [2–4] gives several results, including the construction for a fixed positive integer n of two arithmetic progressions of length n with equal product of terms. Further, he describes infinitely many pairs of 5-term progressions with equal product, and also constructs five 4-term progressions, all having equal product of terms.

Here, we investigate the case $n = 4$. The defining equation is that of a quartic surface, and we study the geometry of this surface. By computing the Néron-Severi group of the surface over $\mathbb{C}$, we can determine infinitely many parametrizations for the problem, and in particular, can determine all parametrizations of a given degree that correspond to curves lying on the surface of arithmetic genus 0. The number of such parametrized curves increases rapidly, with attendant computational difficulties. Here, we simply give all such parametrizations of degrees $1, 2, 3, 4$, and examples of parametrizations for degrees $5, ..., 10$.

2. A quartic surface

Consider two four-term arithmetic progressions with equal products, which by homogeneity we may take in the form $\{a-3d,\ a-d,\ a+d,\ a+3d\}$ and $\{b-3c,\ b-c,\ b+c,\ b+3c\}$. Then

$$V : (a^2-9d^2)(a^2-d^2) = (b^2-9c^2)(b^2-c^2).$$

This equation defines a non-singular quartic surface V. Symmetries of V occur with sign changes of the coordinates, under the mapping $(a,b,c,d) \to (b,a,d,c)$, and under the mapping $(a,b,c,d) \to (3d,3c,b,a)$, generating a symmetry group of order 32. The surface contains the twenty $\mathbb{Q}$-rational straight lines shown in Table 1.

Accordingly, there is a rich geometry of V over the rationals. Denote by $\mathrm{NS}(V(K))$ the Néron-Severi group of the surface V over the field K; then we expect $\mathrm{NS}(V(\mathbb{Q}))$ to be a sizeable subgroup of $\mathrm{NS}(V(\mathbb{C}))$. For reference, the action of the symmetries on the $\mathbb{Q}$-rational straight lines is given in the Appendix.

There are four real lines defined over $\mathbb{Q}(\sqrt{3})$ (see Table 2) and eight imaginary lines (see Table 3).

It is straightforward by considering linear parametrizations to see that this is the full list of lines on the surface V. The intersection matrix $\{(l_i \cdot l_j)\}$ of the 32 lines has rank 19.

l_1:	$a = 3d$ $b = 3c$	l_2:	$a = 3d$ $b = c$	l_3:	$a = 3d$ $b = -c$	l_4:	$a = 3d$ $b = -3c$
l_5:	$a = d$ $b = 3c$	l_6:	$a = d$ $b = c$	l_7:	$a = d$ $b = -c$	l_8:	$a = d$ $b = -3c$
l_9:	$a = -d$ $b = 3c$	l_{10}:	$a = -d$ $b = c$	l_{11}:	$a = -d$ $b = -c$	l_{12}:	$a = -d$ $b = -3c$
l_{13}:	$a = -3d$ $b = 3c$	l_{14}:	$a = -3d$ $b = c$	l_{15}:	$a = -3d$ $b = -c$	l_{16}:	$a = -3d$ $b = -3c$
l_{17}:	$a = b$ $c = d$	l_{18}:	$a = b$ $c = -d$	l_{19}:	$a = -b$ $c = d$	l_{20}:	$a = -b$ $c = -d$

Table 1: Twenty $\mathbb{Q}$-rational straight lines on V

l_{21}:	$a = \sqrt{3}c$ $b = \sqrt{3}d$	l_{22}:	$a = \sqrt{3}c$ $b = -\sqrt{3}d$	l_{23}:	$a = -\sqrt{3}c$ $b = \sqrt{3}d$	l_{24}:	$a = -\sqrt{3}c$ $b = -\sqrt{3}d$

Table 2: Four real straight lines on V

l_{25}:	$a = ib$ $c = id$	l_{26}:	$a = ib$ $c = -id$	l_{27}:	$a = -ib$ $c = id$	l_{28}:	$a = -ib$ $c = -id$
l_{29}:	$a = i\sqrt{3}c$ $b = i\sqrt{3}d$	l_{30}:	$a = i\sqrt{3}c$ $b = -i\sqrt{3}d$	l_{31}:	$a = -i\sqrt{3}c$ $b = i\sqrt{3}d$	l_{32}:	$a = -i\sqrt{3}c$ $b = -i\sqrt{3}d$

Table 3: Eight imaginary straight lines on V

Various conics arise as the residual intersection of V with a plane passing through two of the straight lines. Denote by Π a hyperplane section of the surface V, so that Π has genus 3, and $\Pi^2 = 2 \cdot \text{genus}(\Pi) - 2 = 4$. Then the effective divisor $\Pi - l_i - l_j$ has self-intersection $(\Pi - l_i - l_j)^2 = -4 + 2(l_i \cdot l_j)$, so consequently has genus 0 if and only if $(l_i \cdot l_j) = 1$.

If $\Pi - l_i - l_j$ is irreducible, then its intersection pairing with l_k is non-negative, so $((l_i + l_j) \cdot l_k) \leq 1$. Conversely, if $\Pi - l_i - l_j$ is reducible, then necessarily it is linearly equivalent to $l_m + l_n$ for lines l_m, l_n, and now its intersection pairing with l_n equals $(l_m \cdot l_n) - 2 \leq -1$, that is, $((l_i + l_j) \cdot l_n) \geq 2$. Hence $\Pi - l_i - l_j$ is irreducible if and only if $((l_i + l_j).l_k) \leq 1$ for all lines l_k.

If one of the component lines is $\mathbb{Q}$-rational, then by symmetry we can assume l_i is one of l_1, l_2, l_{17}. Only $\Pi - l_1 - l_j$, for $j = 17, 20, 26, 27$, are acceptable under the above criteria. Only $\Pi - l_2 - l_j$, for $j = 21, 24, 30, 31$, are acceptable. Only $\Pi - l_{17} - l_j$, for $j = 1, 6, 11, 16, 18, 19, 21, 24, 29, 32$, are acceptable.

If no component line is $\mathbb{Q}$-rational, then we have only $\Pi - l_i - l_j$ for $(i, j) =$

$(21,22)$, $(21,23)$, $(21,25)$, $(21,28)$, $(25,26)$, $(25,27)$, $(25,29)$, $(25,32)$, $(29,30)$, $(29,31)$.

It follows that there are precisely two equivalence classes of such $\mathbb{Q}$-rational conics, typified by $\Pi - l_1 - l_{17}$ ($\sim \Pi - l_6 - l_{20}$), and $\Pi - l_{17} - l_{19}$.
The plane $a + b = c + d$ cuts the surface in the two lines l_6, l_{20}, and the residual conic

$$4a^2 + 7ab + 2b^2 - 11ac - 7bc + 9c^2 = 0,$$

with parametrization

$$a : b : c : d \;=\; 3s^2 + s + 2 \;:\; -s^2 - 3s - 8 \;:\; s^2 - 3s - 2 \;:\; s^2 + s - 4. \qquad (2.1)$$

This conic lies in an equivalence class under symmetry of order 16.

The plane $c = d$ cuts V in l_{17}, l_{19}, and the conic

$$a^2 + b^2 = 10c^2,$$

with parametrization

$$a : b : c : d \;=\; 3s^2 - 2s - 3 \;:\; s^2 + 6s - 1 \;:\; s^2 + 1 \;:\; s^2 + 1, \qquad (2.2)$$

lying in an equivalence class of order 4. In this manner we recognise twenty $\mathbb{Q}$-rational conics on V, the residual intersections of the following planes:

Q_1:	$a + b = c + d$	Q_2:	$a + b = c - d$
Q_3:	$a + b = -c + d$	Q_4:	$a + b = -c - d$
Q_5:	$a - b = c + d$	Q_6:	$a - b = c - d$
Q_7:	$a - b = -c + d$	Q_8:	$a - b = -c - d$
Q_9:	$a - b = 3(c - d)$	Q_{10}:	$a - b = 3(c + d)$
Q_{11}:	$a - b = -3(c + d)$	Q_{12}:	$a - b = 3(-c + d)$
Q_{13}:	$a + b = 3(c - d)$	Q_{14}:	$a + b = 3(c + d)$
Q_{15}:	$a + b = -3(c + d)$	Q_{16}:	$a + b = 3(-c + d)$
Q_{17}:	$a = b$	Q_{18}:	$a = -b$
Q_{19}:	$c = d$	Q_{20}:	$c = -d$

Table 4: Twenty $\mathbb{Q}$-rational conics on V

A plane intersection does not of course necessarily contain a straight line, but may give rise to two conics. A straightforward (machine) computation shows that plane intersections delivering two conics arise precisely for the planes (writing $i = \sqrt{-1}$, $r = \sqrt{3}$):

$$a - (1 - i)c + rd = 0, \quad \text{and} \quad a + 2(1 - i)c - ird = 0,$$

together with symmetries and conjugates. The first plane intersection here comprises the two conics

$$Q_0 : \qquad a - (1 - i)c + rd = 0, \quad b^2 + (2r - 5)c^2 + (2i + 2)cd - 2rid^2 = 0;$$

$$Q_0': \qquad a-(1-i)c+rd=0, \quad b^2+(-2r-5)c^2+(-2i-2)cd+2rid^2=0;$$

and Q_0 has parametrization

$$(a,b,c,d)=((-1+r)(3u^2-(3+r)uv-v^2),(1+i)(ru^2+(-4+2r)uv+v^2),\\ (1+i)(ru^2-v^2),(-1+r)(u^2+(1+r)uv-v^2)).$$

Further, the surface V is fibred by curves of genus 1. Consider the intersection of V with the family of planes

$$a-d=t(b-c). \tag{2.3}$$

The intersection contains the line $l_6:\{a=d,\ b=c\}$, together with residual cubic curve

$$b^3(-1+9t^4)+b^2c(-1-27t^4)+9bc^2(1+3t^4)+\\ 9c^3(1-t^4)-36a(b-c)^2t^3+44a^2(b-c)t^2-16a^3t=0.$$

This cubic contains points such as $\mathcal{O}_t(a,b,c,d)=(t,1,-1,-t)$, the point where (2.3) meets the skew line $\{a+d=0=b+c\}$, and so is an elliptic curve over $\mathbb{Q}(t)$. The locus of $\mathcal{O}_t$ as t varies is the line l_{11}. A cubic model of the above curve is

$$E_t: V^2=U^3+67t^2U^2+1440t^4U+36t^2(1+277t^4+t^8), \tag{2.4}$$

with mappings

$$(U,V)=\big(-4t(-2a+7bt-7at^4+2bt^5)/(b+c-2at^3+bt^4-ct^4), \tag{2.5}$$
$$2t(t^4-1)(-b^2-10bc-9c^2-40a^2t^2+82abt^3-82act^3-42b^2t^4+82bct^4$$
$$+20a^2t^6-28abt^7+28act^7+9b^2t^8-18bct^8+9c^2t^8)/(b+c-2at^3+bt^4-ct^4)^2\big),$$

and

$$a:b:c:d=-36t^2(1+t^4)(7+2t^4)-2(4+59t^4)U-5t^2U^2+2t(7+2t^4)V:\\ -36t(1+t^4)(2+7t^4)-2t^3(59+4t^4)U-5tU^2+2(2+7t^4)V:\\ 4t(2+509t^4-43t^8)+2t^3(101-4t^4)U+5tU^2+2(-2+3t^4)V:\\ 4t^2(-43+509t^4+2t^8)+2(-4+101t^4)U+5t^2U^2+2t(3-2t^4)V. \tag{2.6}$$

We note that the torsion subgroup of $E(\mathbb{C}(t))$ is trivial. The curve E_t at (2.4) is singular at $t=0,\infty,\pm1,\pm i$, and at the eight roots of $243t^8+1711t^4+243=0$. The discriminant of (2.4) is

$$-144(t-1)^2t^4(t+1)^2(t^2+1)^2(243t^8+1711t^4+243),$$

and we have the following Kodaira classification types, with the corresponding decomposition of the intersection (see Table 5) together with type I_1 nodal cubics at each root of $243t^8+1711t^4+243=0$. Shioda's fundamental formula [11] results in

$$20\geq \operatorname{rank}\mathrm{NS}(V(\mathbb{C}))=\operatorname{rank}E_t(\mathbb{C}(t))+2+2(3-1)+4(2-1)+8(1-1),$$

whence rank $E_t(\mathbb{C}(t)) \leq 10$.

t	$=$	0	IV	l_5	$+$	l_7	$+$	l_8
t	$=$	∞	IV	l_2	$+$	l_{10}	$+$	l_{14}
t	$=$	1	I_2			l_{17}	$+$	Q_7
t	$=$	-1	I_2			l_{20}	$+$	Q_1
t	$=$	i	I_2			l_{26}	$+$	conic
t	$=$	$-i$	I_2			l_{27}	$+$	conic

Table 5: Singular decompositions of E_t

Theorem 2.1. *NS$(V(\mathbb{C}))$ is a $\mathbb{Z}$-module of rank* 19*, with basis the divisor classes of the 18 lines* l_1, l_2, l_3, l_4, l_5, l_7, l_8, l_{10}, l_{11}, l_{16}, l_{17}, l_{18}, l_{20}, l_{21}, l_{22}, l_{25}, l_{26}, l_{29}, *and the conic* Q_0.

We prove Theorem 2.1 in several steps. It is known that $\mathrm{NS}(V(\mathbb{C}))$ is generated over $\mathbb{Z}$ by (i) a fibre of E_t, the zero section, the fibre components that do not meet the zero section; and (ii) sections that form a basis of $E_t(\mathbb{C}(t))$. For (i), we have the ten generators l_2, l_5, l_7, l_8, l_{10}, l_{11}, l_{17}, l_{20}, l_{26}, l_{27}. For (ii), we shall show $E_t(\mathbb{C}(t))$ has rank 9, so that indeed rank $\mathrm{NS}(V(\mathbb{C})) = 19$. It will then remain to determine an explicit basis.

The straight lines and conic Q_0 provide us with the following 9 independent points in $E_t(\mathbb{C}(t))$:

pullback	point on $E_t(\mathbb{C}(t))$
l_1	$J_1 = (-15t^2,\ 6t^5 + 6t)$;
l_4	$J_2 = (-18t^2,\ 6t^5 - 6t)$;
l_{16}	$J_3 = (-30t^2,\ -6t^5 - 6t)$;
l_{18}	$J_4 = (4t^4 - 10t^3 - 10t^2 - 10t + 4,$ $-8t^6 + 30t^5 - 58t^4 + 60t^3 - 58t^2 + 30t - 8)$;
l_{21}	$J_5 = (2rt^3 - 18t^2 + 2rt,\ 6t^5 + 2rt^4 + 12t^3 + 2rt^2 + 6t)$;
l_{22}	$J_6 = (4rt^3 - 18t^2 - 4rt,\ -6t^5 - 16rt^4 + 12t^3 + 16rt^2 - 6t)$;
l_{25}	$J_7 = (-4t^4 + 10it^3 - 10t^2 - 10it - 4,$ $8it^6 + 30t^5 - 58it^4 - 60t^3 + 58it^2 + 30t - 8i)$;
l_{29}	$J_8 = (-4rit^3 - 18t^2 - 4rit, -6t^5 - 16rit^4 - 12t^3 - 16rit^2 - 6t)$;
Q_0	$J_9 = ((r+3)(i+1)t^3 - 2(r+10)t^2 + (3r+5)(i-1)t + 4(r+2)i,$ $6t^5 + (5r+9)(i-1)t^4 + 2(5r+11)it^3 - 7(r+1)(i+1)t^2$ $-6(4r+7)t + 4(3r+5)(i-1))$

Table 6: Points on $E_t(\mathbb{C}(t))$

That the points J_i, $i = 1, \ldots, 9$, are linearly independent on E_t follows from the height-pairing matrix

$$M = \begin{pmatrix}
\frac{8}{3} & 0 & \frac{4}{3} & 2 & \frac{2}{3} & \frac{4}{3} & 2 & \frac{4}{3} & \frac{4}{3} \\
0 & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\
\frac{4}{3} & 0 & \frac{8}{3} & 2 & \frac{4}{3} & \frac{2}{3} & 2 & \frac{2}{3} & \frac{2}{3} \\
2 & 0 & 2 & 3 & 1 & 1 & 2 & 1 & \frac{3}{2} \\
\frac{2}{3} & 0 & \frac{4}{3} & 1 & \frac{5}{3} & \frac{1}{3} & 1 & \frac{1}{3} & -\frac{1}{6} \\
\frac{4}{3} & 0 & \frac{2}{3} & 1 & \frac{1}{3} & \frac{5}{3} & 1 & \frac{2}{3} & \frac{1}{6} \\
2 & 0 & 2 & 2 & 1 & 1 & 3 & 1 & \frac{1}{2} \\
\frac{4}{3} & 0 & \frac{2}{3} & 1 & \frac{1}{3} & \frac{2}{3} & 1 & \frac{5}{3} & \frac{7}{6} \\
\frac{4}{3} & \frac{1}{3} & \frac{2}{3} & \frac{3}{2} & -\frac{1}{6} & \frac{1}{6} & \frac{1}{2} & \frac{7}{6} & \frac{7}{3}
\end{pmatrix}$$

of determinant $\frac{8}{9}$. It follows that rank $E_t(\mathbb{C}(t)) \geq 9$.

We now have that the divisor classes of the following 19 curves are independent in the Néron-Severi group $\mathrm{NS}(V, \mathbb{C})$:

$$l_1, l_2, l_3, l_4, l_5, l_7, l_8, l_{10}, l_{11}, l_{16}, l_{17}, l_{18}, l_{20}, l_{21}, l_{22}, l_{25}, l_{26}, l_{29}, Q_0. \tag{2.7}$$

(Note: the conic $ac = bd$ cuts V in the divisor

$$l_1 + l_6 + l_{11} + l_{16} + l_{17} + l_{20} + l_{26} + l_{27} \sim 2\,\Pi \sim l_1 + l_2 + l_3 + l_4 + l_5 + l_6 + l_7 + l_8,$$

which allows us up to linear equivalence to replace l_{27} by l_3.)

Lemma 2.2. *NS*$(V(\mathbb{C}))$ *has rank* 19.

Proof. We follow closely the exposition of Kloosterman [6] to which the reader is referred for full details.

Let Y be a smooth projective surface defined over $\mathbb{Q}$, with Néron-Severi group $\mathrm{NS}(Y)$. Suppose that p is a prime of good reduction, and denote by $\overline{Y}$ the reduction of Y modulo p. It is known that $\mathrm{NS}(Y)$ modulo torsion together with the intersection pairing on $\mathrm{NS}(Y)$ forms a lattice. Denote by $\Delta(\mathrm{NS}(Y_K))$ the discriminant of a Gram matrix of the Néron-Severi lattice $\mathrm{NS}(Y_K)$ of Y over K with respect to the pairing. Proposition 4.2 of Kloosterman tells us that $\Delta(\mathrm{NS}(Y_{\overline{\mathbb{Q}}}))$ and $\Delta(\mathrm{NS}(\overline{Y}_{\overline{\mathbb{F}_p}}))$ differ by a square.

The idea therefore (originally suggested by van Luijk) is to find two distinct primes p_1, p_2 of good reduction for which the rank of the Néron-Severi lattices is the same, but for which the discriminants of the lattices differ by a non-square. It will follow that the rank of $\mathrm{NS}(Y_{\overline{\mathbb{Q}}})$ is at least one less than the rank of $\mathrm{NS}(\overline{Y}_{\overline{\mathbb{F}_{p_1}}})$.

We quote two further results from Kloosterman. Here, q is a prime power, and l a prime with $(l, q) = 1$.

Conjecture 4.3 (Tate Conjecture).
Let $Y/\mathbb{F}_q$ be a smooth surface with Néron-Severi rank $\rho(Y)$. Let F_q be the automorphism of $H^2_{\acute{e}t}(Y, \mathbb{Q}_l))$ induced by the Frobenius automorphism of $\mathbb{F}_q$. Let $Q(t)$ be $\det(I - tF_q | H^2_{\acute{e}t}(Y, \mathbb{Q}_l))$. Then $\rho(Y)$ equals the number of reciprocal zeroes of $Q(t)$ of the form $q\zeta$, with ζ a root of unity.

Conjecture 4.6 (Artin-Tate Conjecture).
Let $Y/\mathbb{F}_q$ be a smooth surface with Néron-Severi rank $\rho(Y)$. Let F_q be the automorphism of $H^2_{\acute{e}t}(Y,\mathbb{Q}_l))$ induced by the Frobenius automorphism of $\mathbb{F}_q$. Let $Q_q(t)$ be $\det(I - tF_q|H^2_{\acute{e}t}(Y,\mathbb{Q}_l))$. Then

$$\lim_{s\to 1}\frac{Q_q(q^{-s})}{(1-q^{1-s})^{\rho'(Y)}} = \frac{(-1)^{\rho'(Y)-1}\#\mathrm{Br}(Y)\Delta(\mathrm{NS}(Y_{\mathbb{F}_q}))}{q^{\alpha(Y)}(\#NS(Y_{\mathbb{F}_q})_{\mathrm{tor}})^2},$$

where $\alpha(Y) = \chi(Y,O_Y) - 1 + \dim \mathrm{Pic}^0(Y)$, $\mathrm{Br}(Y)$ is the Brauer group of Y, $\mathrm{NS}(Y_{\mathbb{F}_q})$ is the subgroup of $\mathrm{NS}(Y_{\overline{\mathbb{F}_q}})$ generated by $\mathbb{F}_q$-rational divisors, and $\rho'(Y) = \mathrm{rank}\ \mathrm{NS}(Y_{\overline{\mathbb{F}_q}})$.

These Conjectures are known to be true when $(q,6) = 1$ and $Y/\mathbb{F}_q$ is an elliptic $K3$ surface, as in the case we are considering.

Again from Kloosterman, Proposition 4.7, the order of $\mathrm{Br}(Y)$ is a square, and with the hypothesis that $\rho(Y) = \rho'(Y)$, then the Artin-Tate Conjecture gives the following:

$$\Delta(\mathrm{NS}(Y_{\overline{\mathbb{F}_q}})) \equiv (-1)^{\rho'(Y)-1}q^{\alpha(Y)}\lim_{s\to 1}\frac{Q_q(q^{-s})}{(1-q^{1-s})^{\rho'(Y)}} \bmod \mathbb{Q}^{*2}.$$

In our case, at the primes of good reduction $p = 37, 61$, the known 19 independent divisor classes are defined over $\mathbb{F}_p$. By counting the points on V over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ we compute

$$Q_{37}(x) = (1-37x)^{20}(1+38x+1369x^2), \quad Q_{61}(x) = (1-61x)^{20}(1+118x+3721x^2).$$

We have $\rho(Y) = \rho'(Y) = 20$. We thus get

$$\Delta(\mathrm{NS}(Y_{\overline{\mathbb{F}_p}})) \equiv -p^{\alpha(Y)}\lim_{s\to 1}\frac{Q_p(p^{-s})}{(1-p^{1-s})^{20}} \bmod \mathbb{Q}^{*2}.$$

Hence

$$\Delta(\mathrm{NS}(Y_{\overline{\mathbb{F}_{37}}})) \equiv -37^{\alpha(Y)}(1+\frac{38}{37}+1) \equiv -7\cdot 37^{\alpha(Y)-1} \bmod \mathbb{Q}^{*2};$$
$$\Delta(\mathrm{NS}(Y_{\overline{\mathbb{F}_{61}}})) \equiv -61^{\alpha(Y)}(1+\frac{118}{61}+1) \equiv -3\cdot 5\cdot 61^{\alpha(Y)-1} \bmod \mathbb{Q}^{*2}.$$

Consequently, the two discriminants do not differ by a perfect square, and it follows that the rank of $\mathrm{NS}(Y_{\overline{\mathbb{Q}}})$ is at least one less than the rank of $\mathrm{NS}(Y_{\overline{\mathbb{F}_{37}}})$, so must equal 19. □

Corollary 2.3. *The group $E_t(\mathbb{C}(t))$ has rank nine, and the points $J_1,\ldots,J_9$ listed in Table 6 form a basis.*

Proof. The previous computation implies the rank is 9. That the $\{J_i\}$ form a basis follows from Lemma 2.5 of Kuwata [7]. The first criterion in the Lemma implies

that the index of the subgroup in $E_t(\mathbb{C}(t))$ generated by the J_i can be divisible only by 2 or 3. It is a straightforward computation to determine that for $\epsilon_i = 0, 1$, not all zero, none of the points $\sum_{i=1}^{9} \epsilon_i J_i$ can lie in $2E_t(\mathbb{C}(t))$; and for $\epsilon_i = 0, \pm 1$, not all zero, none of the points $\sum_{i=1}^{9} \epsilon_i J_i$ can lie in $3E_t(\mathbb{C}(t))$. □

It remains to determine a $\mathbb{Z}$-basis for $\mathrm{NS}(V, \mathbb{C})$.

The divisors at (2.7) form a basis over $\mathbb{Q}$. Let $D \sim c_1 l_1 + c_2 l_2 + \cdots + c_{26} l_{26} + c_{29} l_{29} + c_0 Q_0$, which notationally we abbreviate to $(c_1, c_2, \ldots, c_{26}, c_{29}, c_0)$, lie in $\mathrm{NS}(V, \mathbb{C})$ for $c_i \in \mathbb{Q}$. Demanding integer intersection with each of the 32 straight lines and Q_0 gives a system of equations for the coefficients c_i that implies D is a $\mathbb{Z}$-linear combination of the following divisors:

$$l_1, l_2, l_3, l_4, l_5, l_7, l_{10}, l_{17}, l_{18}, l_{20}, l_{21}, l_{22}, l_{25}, l_{26}, l_{29}, Q_0, \tag{2.8}$$

and

$$\begin{aligned}
D_1 &\sim \frac{1}{4}(0, 0, 1, -1, 0, -1, 1, 0, 0, 0, 0, -2, 0, 0, 2, 2, 0, -2, 0), \\
D_2 &\sim \frac{1}{4}(1, -3, 2, 0, -1, 1, 0, -1, 1, 0, 2, 0, 0, 0, -2, 0, -2, 2, 0), \\
D_3 &\sim \frac{1}{8}(0, 1, 1, 3, 3, -5, -1, 2, -1, 1, -2, 0, -2, -4, 4, -4, 4, 0, 0).
\end{aligned}$$

The divisor $\Delta \sim aD_1 + bD_2 + cD_3$ for $a, b, c \in \mathbb{Z}$ satisfies

$$\Delta^2 = -4a^2 + \frac{5}{2}ab - \frac{7}{2}b^2 + \frac{3}{2}ac + \frac{7}{2}bc - \frac{33}{8}c^2,$$

which, being equal to $2 \cdot \mathrm{genus}(\Delta) - 2$, lies in $2\mathbb{Z}$. Thus c is even, and D is a $\mathbb{Z}$-linear combination of the divisors at (2.8) and of $(d_1, d_2, d_3) = (D_1, D_2, 2D_3 + l_2 - l_{26})$. Now

$$\begin{aligned}
4d_1 &\sim -2l_9 + 2l_{13} + 2l_{15} + 2l_{16} + 2l_{19} + 2l_{22} + l_{25} - l_{28} - 5l_{29} - 3l_{32}, \\
4d_2 &\sim -2l_3 + 4l_4 - 6l_9 + 4l_{12} + 4l_{15} + 4l_{16} - 2l_{19} - 8l_{22} - 4l_{23} + 2l_{24} \\
&\quad + l_{25} + 3l_{28} + 2l_{29} + 5l_{30} + 3l_{31} - 2l_{32} - 4Q_0, \\
4d_3 &\sim -2l_3 + 10l_4 - 8l_9 + 8l_{13} + 6l_{15} + 14l_{16} + 3l_{22} - l_{23} + 4l_{24} + 4l_{28} \\
&\quad - 9l_{29} - 10l_{30} - 10l_{31} - 9l_{32},
\end{aligned}$$

linear equivalences which express the divisors $4d_i$ of degree 0 in terms of divisors which meet E_t. Each induces a divisor of points $(4d_i.E_t)$ on E_t of degree 0, and we can compute the image of these divisors under the Jacobian mapping jac from the group of divisors on E_t of degree 0, to E_t.

We first identify the following intersections on E_t.

l	$(l.E_t)$	l	$(l.E_t)$
l_1	J_1	l_{21}	J_5
l_3	$-J_2+J_3$	l_{22}	J_6
l_4	J_2	l_{23}	J_1-J_6
l_9	J_2+J_3	l_{24}	J_3-J_5
l_{11}	$\mathcal{O}$	l_{25}	J_7
l_{12}	J_1-J_2	l_{28}	$J_1+J_3-J_7$
l_{13}	$-J_2$	l_{29}	J_8
l_{15}	J_1+J_2	l_{30}	$-J_1-J_2-J_4+J_5+J_6+J_7-J_8+2J_9$
l_{16}	J_3	l_{31}	$J_1+J_2+J_3+J_4-J_5-J_6-J_7+J_8-2J_9$
l_{18}	J_4	l_{32}	J_1-J_8
l_{19}	$J_1+J_3-J_4$	Q_0	J_9

Table 7: Intersections on E_t

Using the above table,

$$\begin{aligned}\operatorname{jac}(4d_1.E_t) &= -2J_2+J_3-2J_4+2J_6+2J_7-2J_8,\\ \operatorname{jac}(4d_2.E_t) &= J_1-2J_2+2J_3-2J_6+2J_8,\\ \operatorname{jac}(4d_3.E_t) &= 2(J_2+J_3-2J_5+2J_6-2J_7).\end{aligned} \tag{2.9}$$

The assumption that $ad_1+bd_2+cd_3$, $a,b,c\in\mathbb{Z}$, exists as divisor implies that $\operatorname{jac}((a\,4d_1+b\,4d_2+c\,4d_3).E_t)=4\operatorname{jac}((ad_1+bd_2+cd_3).E_t)\in 4E_t(\mathbb{C}(t))$, that is

$$\begin{aligned}bJ_1-2(a+b-c)J_2+(a+2b+2c)J_3-2aJ_4-4cJ_5+2(a-b+2c)J_6\\ +2(a-2c)J_7-2(a-b)J_8\in 4E_t(\mathbb{C}(t)).\end{aligned}$$

The deduction is that $a,b\equiv 0 \bmod 4$, $c\equiv 0 \bmod 2$. A set of $\mathbb{Z}$-generators is now the divisors at (2.8) and $4d_1$, $4d_2$, $2d_3$; equivalently, the divisors

$$l_1,l_2,l_3,l_4,l_5,l_7,l_8,l_{10},l_{11},l_{17},l_{18},l_{20},l_{21},l_{22},l_{25},l_{26},l_{29},Q_0,$$

and

$$d_4=2d_3\sim\frac{1}{2}(0,5,1,3,3,-5,-1,2,-1,1,-2,0,-2,-4,4,-4,0,0,0).$$

Assume that d_4 exists as a divisor in $\mathrm{NS}(V,\mathbb{C})$. From (2.9), we have $\operatorname{jac}(2d_4.E_t)=\operatorname{jac}(4d_3.E_t)=2(J_2+J_3-2J_5+2J_6-2J_7)$, so that the divisor $d_5=d_4-l_9+l_{21}-l_{22}+l_{25}$ of degree 0 satisfies $\operatorname{jac}(2d_5.E_t)=0$. Since E has trivial torsion, it follows that $\operatorname{jac}(d_5.E_t)=0$. Hence from properties of the Jacobian mapping, $d_5.E_t\sim 0$ on E_t. Thus there exists a function f_t on E_t having divisor $d_5.E_t$, and induced by a function f on V. Then $(f)-d_5$ is a divisor not meeting E_t, which therefore is a sum of the singular components of E_t; equivalently, a sum of the singular straight line components of E_t. We deduce

$$d_5\sim c_2l_2+c_5l_5+c_7l_7+c_8l_8+c_{10}l_{10}+c_{14}l_{14}+c_{17}l_{17}+c_{20}l_{20}+c_{26}l_{26}+c_{27}l_{27}.$$

However $1 = d_5.l_{17} = -2c_{17}$, impossible. Thus d_5 cannot exist as divisor, and $\mathrm{NS}(V, \mathbb{C})$ has $\mathbb{Z}$-basis as required. This completes the proof of Theorem 2.1.

In the Appendix, we give a matrix expressing the divisor classes of the 32 lines as linear combinations of this generating set.

3. Rational parametrizations

That part of the Néron-Severi Group defined over $\mathbb{Q}$ is seen to be generated by the divisor classes of

$$l_1, l_2, l_3, l_4, l_5, l_7, l_8, l_{10}, l_{11}, l_{16}, l_{17}, l_{18}, l_{20},$$

which set we denote by $\{C_i\}$, $i = 1, \ldots, 13$, with

$$\begin{aligned}
l_{21} + l_{21}^{\mathrm{conj}} &\sim l_3 + l_4 + l_7 + l_8 - l_{17} - l_{20},\\
l_{22} + l_{22}^{\mathrm{conj}} &\sim l_1 + l_2 - l_5 - l_7 - 2l_8 + l_{10} + l_{11} + l_{17} + l_{20},\\
l_{25} + l_{25}^{\mathrm{conj}} &\sim l_1 - l_7 - l_{10} + l_{16} + l_{17} + l_{20},\\
l_{26} + l_{26}^{\mathrm{conj}} &\sim l_2 + l_3 + l_4 + l_5 + l_7 + l_8 - l_{11} - l_{16} - l_{17} - l_{20},\\
l_{29} + l_{29}^{\mathrm{conj}} &\sim l_1 + 2l_2 + l_3 + l_4 - l_8 + l_{10} - l_{16} - l_{17} - l_{20},\\
l_{30} + l_{30}^{\mathrm{conj}} &\sim -l_2 - l_5 + l_{11} + l_{16} + l_{17} + l_{20}.
\end{aligned}$$

The associated intersection matrix is

	l_1	l_2	l_3	l_4	l_5	l_7	l_8	l_{10}	l_{11}	l_{16}	l_{17}	l_{18}	l_{20}
l_1	-2	1	1	1	1	0	0	0	0	0	1	0	1
l_2	1	-2	1	1	0	0	0	1	0	0	0	0	0
l_3	1	1	-2	1	0	1	0	0	1	0	0	0	0
l_4	1	1	1	-2	0	0	1	0	0	1	0	1	0
l_5	1	0	0	0	-2	1	1	0	0	0	0	0	0
l_7	0	0	1	0	1	-2	1	0	1	0	0	1	0
l_8	0	0	0	1	1	1	-2	0	0	1	0	0	0
l_{10}	0	1	0	0	0	0	0	-2	1	0	0	1	0
l_{11}	0	0	1	0	0	1	0	1	-2	0	1	0	1
l_{16}	0	0	0	1	0	0	1	0	0	-2	1	0	1
l_{17}	1	0	0	0	0	0	0	0	1	1	-2	1	0
l_{18}	0	0	0	1	0	1	0	1	0	0	1	-2	1
l_{20}	1	0	0	0	0	0	0	0	1	1	0	1	-2

Putting $\Gamma \sim x_1C_1 + x_2C_2 + ... + x_{13}C_{13}$, we have

$$\begin{aligned}
&\deg(\Gamma)^2 - 4(\Gamma.\Gamma) = \deg(\Gamma)^2 - 8(\mathrm{genus}(\Gamma) - 1) =\\
&(x_1 - x_2 - x_3 + x_4 - x_5 + x_6 - x_7 + x_8 + x_9 + x_{10} - x_{11} - x_{12} - x_{13})^2\\
&+ 2(x_1 - x_4 - x_6 - x_8 + x_9 + x_{10} - x_{11} + x_{12} - x_{13})^2
\end{aligned}$$

$$
\begin{aligned}
&+2(x_1-x_4+x_6+x_8-x_9+x_{10})^2\\
&+2(x_1-x_2-x_5-x_9-x_{10})^2+2(x_1-x_3+x_7+x_9-x_{10})^2\\
&+2(x_2-x_4-x_5+x_6-x_8)^2+2(x_3-x_4-x_6+x_7+x_8)^2\\
&+2(x_{11}-x_{12}+x_{13})^2+4(x_{11}-x_{13})^2+4(x_5-x_7)^2+4(x_2-x_3)^2+4x_{12}^2
\end{aligned}
$$

which is in a machine computable form if we wish to determine (via the coefficients x_i) the curves Γ of genus 0 and given degree $\deg(\Gamma)$. Putting

$$
\begin{aligned}
m_1 &= x_1-x_2-x_3+x_4-x_5+x_6-x_7+x_8+x_9+x_{10}-x_{11}-x_{12}-x_{13},\\
m_2 &= x_1-x_2-x_5-x_9-x_{10},\\
m_3 &= x_2-x_4-x_5+x_6-x_8,\\
m_4 &= x_1-x_3+x_7+x_9-x_{10},\\
m_5 &= x_3-x_4-x_6+x_7+x_8,\\
m_6 &= x_1-x_4+x_6+x_8-x_9+x_{10},\\
m_7 &= x_1-x_4-x_6-x_8+x_9+x_{10}-x_{11}+x_{12}-x_{13},\\
m_8 &= x_{11}-x_{12}+x_{13},\\
m_9 &= x_2-x_3,\\
m_{10} &= x_5-x_7,\\
m_{11} &= x_{11}-x_{13},\\
m_{12} &= x_{12},\\
m_{13} &= \deg(\Gamma),
\end{aligned}
$$

we have to tabulate the finitely many solutions to the equation

$$
m_1^2+2\sum_{i=2}^{8}m_i^2+4\sum_{i=9}^{12}m_i^2=\deg(\Gamma)^2-4(\Gamma.\Gamma) \tag{3.1}
$$

and then determine $(x_1,\ldots,x_{13})=\mathbf{x}$ from $(m_1,\ldots,m_{13})=\mathbf{m}$ by means of

$$
\mathbf{x}=\frac{1}{4}\begin{pmatrix}
0 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & -2 & 1\\
0 & 1 & 1 & -1 & 1 & -1 & 0 & -1 & 3 & 1 & 0 & -2 & 1\\
0 & 1 & 1 & -1 & 1 & -1 & 0 & -1 & -1 & 1 & 0 & -2 & 1\\
0 & 1 & -1 & -1 & -1 & -1 & 0 & -1 & 1 & -1 & 0 & -2 & 1\\
-1 & -1 & -1 & 1 & -1 & 1 & 0 & -1 & -1 & 1 & 0 & -2 & 0\\
0 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & -2 & 0 & 0 & 0 & 0\\
-1 & -1 & -1 & 1 & -1 & 1 & 0 & -1 & -1 & -3 & 0 & -2 & 0\\
1 & 0 & 0 & 0 & 2 & 0 & -1 & 0 & 1 & 1 & 0 & 2 & 0\\
1 & -1 & 1 & 1 & 1 & -1 & 0 & 1 & -1 & 1 & 0 & 2 & 0\\
0 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 0 & -2 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 2 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -2 & 2 & 0
\end{pmatrix}\mathbf{m}^t
$$

This imposes congruence conditions on the m_i at (3.1), namely:

$$\begin{aligned} m_1 + m_{13} &\equiv 0 \bmod 2, \\ m_2 + m_3 + m_6 &\equiv 0 \bmod 2, \\ m_4 + m_5 + m_6 &\equiv 0 \bmod 2, \\ m_6 + m_7 + m_8 &\equiv 0 \bmod 2, \\ m_8 + m_{11} + m_{12} &\equiv 0 \bmod 2, \\ m_1 + m_3 + m_4 + m_8 &\equiv 0 \bmod 2, \\ m_1 + m_7 + m_9 + m_{10} &\equiv 0 \bmod 2, \end{aligned}$$

and

$$\begin{aligned} m_1 + 2m_6 + m_{13} &\equiv 0 \bmod 4, \\ m_1 - m_2 + m_3 + m_4 + m_5 - m_6 + m_8 - m_9 + m_{10} + 2m_{12} &\equiv 0 \bmod 4, \\ m_2 - m_3 - m_4 + m_5 - m_6 + m_7 + m_8 + 2m_9 &\equiv 0 \bmod 4. \end{aligned}$$

For $\mathbb{Q}$-rational curves of degree 1, we find (as expected) exactly the 20 known $\mathbb{Q}$-rational lines, falling into three equivalence classes under symmetry, with representatives l_1 (8 symmetries), l_2 (8 symmetries), and l_{17} (4 symmetries).

For $\mathbb{Q}$-rational curves of degree 2 we find the known conics, falling into the two equivalence classes $\Pi - l_1 - l_{17}$ (16 symmetries) and $\Pi - l_{17} - l_{18}$ (4 symmetries). Their parametrizations are given at (2.1) and (2.2).

There are 24 $\mathbb{Q}$-rational irreducible cubics, in three equivalence classes up to symmetry, with representatives $2\Pi - l_5 - l_{12} - l_{19} - l_{30} - l_{31}$, $2\Pi - l_{11} - l_{16} - l_{17} - l_{18} - l_{20}$, and $2\Pi - l_1 - l_{11} - l_{17} - l_{18} - l_{20}$ (8 symmetries each).

Equivalence class	Parametrization $(a : b : c : d)$
$2\Pi - l_5 - l_{12} - l_{19} - l_{30} - l_{31}$	$-5 + 21s^2$
	$5 + 3s^2$
	$-7s + 15s^3$
	$s + 15s^3$
$2\Pi - l_{11} - l_{16} - l_{17} - l_{18} - l_{20}$	$4 + s + 7s^2 + 6s^3$
	$6 + 7s + s^2 + 4s^3$
	$-2 + 3s + 7s^2 + 4s^3$
	$4 + 7s + 3s^2 - 2s^3$
$2\Pi - l_1 - l_{11} - l_{17} - l_{18} - l_{20}$	$3 + 7s + 7s^2 + s^3$
	$1 + 7s + 7s^2 + 3s^3$
	$1 + s + 3s^2 + s^3$
	$1 + 3s + s^2 + s^3$

Table 8: Rational cubics on V

There are 176 $\mathbb{Q}$-rational quartics in eight equivalence classes:

Equivalence class	Parametrization $(a : b : c : d)$
$\{0,0,0,-1,0,1,-1,1,2,-1,1,1,1\}$	$6-5s-11s^2-7s^3-s^4$ $-12-21s-15s^2-5s^3-s^4$ $4+s-3s^2-3s^3-s^4$ $6+11s+11s^2+5s^3+s^4$
$\{0,0,0,1,1,1,2,-1,0,1,0,-1,0\}$	$3-7s-2s^2-20s^3+8s^4$ $-3+3s-24s^2+16s^3-8s^4$ $-1+7s-8s^4$ $3-5s+2s^2+4s^3-8s^4$
$\{1,0,1,1,-1,0,0,0,0,0,0,1,1\}$	$12+27s+42s^2+23s^3+2s^4$ $18+37s+18s^2+9s^3+4s^4$ $6+7s-7s^3-4s^4$ $4-9s-12s^2-7s^3-2s^4$
$\{0,-1,0,1,0,0,1,0,0,1,0,1,1\}$	$-3-18s-6s^2-4s^3-s^4$ $9-4s-6s^2-6s^3-s^4$ $-3+2s+12s^2+4s^3+s^4$ $1-12s+2s^3+s^4$
$\{0,0,-1,1,0,0,1,0,0,1,0,1,1\}$	$12+27s-21s^2-149s^3-65s^4$ $6+41s+27s^2+33s^3+65s^4$ $6+25s+81s^2+41s^3-13s^4$ $-4-15s-9s^2-59s^3-13s^4$
$\{1,0,1,0,-1,0,-2,1,1,0,1,1,1\}$	$-1+11s+3s^2+49s^3+10s^4$ $3-s+9s^2+21s^3+40s^4$ $1-s+13s^2-27s^3-10s^4$ $-1+5s+s^2-s^3+20s^4$

Table 9: Rational quartics on V

The divisor $\{0,-1,0,0,0,1,1,0,0,1,0,1,1\}$ represents a $\mathbb{Q}$-rational quartic curve defined over $\mathbb{Q}$, but possessing no rational (indeed real) points; its parametrization may be given as

$$\begin{aligned} a : b : c : d = i\sqrt{3}(1+s^2)(1-s-s^2) &: \\ i\sqrt{3}(1+s^2)(1+s-s^2) &: \\ 1-s+4s^2+s^3+s^4 &: \\ 1+s+4s^2-s^3+s^4. & \end{aligned}$$

Similarly, the divisor $\{2,3,2,2,0,-1,-1,0,0,-1,-1,-1,0\}$ is represented by

$$\begin{aligned} a : b : c : d = 3+7s-8s^2-7s^3+3s^4 &: \\ 3-7s-8s^2+7s^3+3s^4 &: \\ \sqrt{7/3}(1+s-s^2)(1+s^2) &: \\ \sqrt{7/3}(1-s-s^2)(1+s^2). & \end{aligned}$$

The number of rationally parametrizable curves increases rapidly, and it seems likely that there are such curves of every positive degree. We content ourselves with listing just one rational parametrization for degrees 5 to 10.

$$(a,b,c,d) = (3s^5+5s,\ 5s^4+3,\ s^4-1,\ s^5-s);$$

$$\begin{aligned}(a,b,c,d) = (&27s^6+27s^5+19s^2+17s+6,\\ &27s^6+45s^5+36s^4-18s^3-39s^2-23s-4,\\ &9s^6-3s^5+12s^4+30s^3+35s^2+17s+4,\\ &9s^6-9s^5-36s^4-48s^3-31s^2-11s-2);\end{aligned}$$

$$\begin{aligned}(a,b,c,d) = (&s^7+16s^6+56s^5+85s^4+44s^3+s^2-11s-3,\\ &3s^7+11s^6-s^5-44s^4-85s^3-56s^2-16s-1,\\ &s^7+5s^6+9s^5+20s^4+25s^3+16s^2+4s+1,\\ &s^7+4s^6+16s^5+25s^4+20s^3+9s^2+5s+1);\end{aligned}$$

$$\begin{aligned}(a,b,c,d) = (&s^8-5s^7+26s^6-76s^5+137s^4-115s^3+16s^2+64s-24,\\ &s^8-3s^7-2s^6+46s^5-153s^4+277s^3-282s^2+156s-24,\\ &s^8-5s^7+10s^6-6s^5-17s^4+35s^3-30s^2+4s-8,\\ &s^8-7s^7+26s^6-60s^5+105s^4-137s^3+136s^2-80s+24);\end{aligned}$$

$$(a,b,c,d) = (s^9-33s^5-184s,\ s^8+47s^4+96,\ 3s^8+21s^4-32,\ s^9+7s^5+56s);$$

$$\begin{aligned}&(a,b,c,d) =\\ &(4s^{10}-25s^9+123s^8-355s^7+653s^6-610s^5+56s^4+720s^3-976s^2+640s-192,\\ &6s^{10}-31s^9+61s^8-15s^7-233s^6+538s^5-728s^4+760s^3-864s^2+544s-64,\\ &2s^{10}-5s^9-19s^8+155s^7-481s^6+930s^5-1208s^4+1080s^3-608s^2+160s-64,\\ &4s^{10}-31s^9+119s^8-285s^7+533s^6-762s^5+808s^4-560s^3+304s^2-256s+64).\end{aligned}$$

4. Appendix

For reference, we give here (in terms of subscript) the action of the sign-change symmetries on the $\mathbb{Q}$-rational lines, together with the action of the further two symmetries:

(a,b,c,d)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(a,b,c,-d)	13	14	15	16	9	10	11	12	5	6	7	8	1	2	3	4	18	17	20	19
(a,b,-c,d)	4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13	18	17	20	19
(a,b,-c,-d)	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	17	18	19	20
(a,-b,c,d)	4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13	19	20	17	18
(a,-b,c,-d)	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	20	19	18	17
(a,-b,-c,d)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	20	19	18	17
(-a,b,c,d)	13	14	15	16	9	10	11	12	5	6	7	8	1	2	3	4	19	20	17	18
(b,a,d,c)	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16	17	18	19	20
(3d,3c,b,a)	6	5	8	7	2	1	4	3	14	13	16	15	10	9	12	11	17	19	18	20

Table 10: Action of the symmetries on the $\mathbb{Q}$-rational straight lines

The following matrix expresses the linear equivalence classes of the 32 straight lines on V in terms of the set of $\mathbb{Z}$-generators of Theorem 2.1.

	l_1	l_2	l_3	l_4	l_5	l_7	l_8	l_{10}	l_{11}	l_{16}	l_{17}	l_{18}	l_{20}	l_{21}	l_{22}	l_{25}	l_{26}	l_{29}	Q_0
l_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l_2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l_3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l_4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l_5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l_6	1	1	1	1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0
l_7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
l_8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
l_9	0	0	0	1	0	0	1	-1	-1	1	0	0	0	0	0	0	0	0	0
l_{10}	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
l_{11}	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
l_{12}	1	1	1	0	0	0	-1	0	0	-1	0	0	0	0	0	0	0	0	0
l_{13}	0	1	1	0	-1	0	-1	1	1	-1	0	0	0	0	0	0	0	0	0
l_{14}	0	-1	0	0	1	1	1	-1	0	0	0	0	0	0	0	0	0	0	0
l_{15}	1	1	0	1	0	-1	0	0	-1	0	0	0	0	0	0	0	0	0	0
l_{16}	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
l_{17}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
l_{18}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
l_{19}	1	1	1	1	1	0	1	-1	-1	0	-1	-1	-1	0	0	0	0	0	0
l_{20}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
l_{21}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
l_{22}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
l_{23}	1	1	0	0	-1	-1	-2	1	1	0	1	0	1	0	-1	0	0	0	0
l_{24}	0	0	1	1	0	1	1	0	0	0	-1	0	-1	-1	0	0	0	0	0
l_{25}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
l_{26}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
l_{27}	0	1	1	1	1	1	1	0	-1	-1	-1	0	-1	0	0	0	-1	0	0
l_{28}	1	0	0	0	0	-1	0	-1	0	1	1	0	1	0	0	-1	0	0	0
l_{29}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
l_{30}	-1	1	1	0	0	1	0	0	-1	-1	-1	-1	-2	1	1	1	-1	-1	2
l_{31}	1	-2	-1	0	-1	-1	0	0	2	2	2	1	3	-1	-1	-1	1	1	-2
l_{32}	1	2	1	1	0	0	-1	1	0	-1	-1	0	-1	0	0	0	0	-1	0

Table 11: Linear equivalence classes of the lines in terms of the $\mathbb{Z}$-generators of Theorem 2.1

Acknowledgements. All computations for this paper were performed using Magma [1]. It is a pleasure for me to acknowledge here the warm hospitality provided by the Harish-Chandra Research Institute, Allahabad, where this paper was completed.

References

[1] W. Bosma, J. Cannon, C. Playoust: *The Magma algebra system. I. The user language*, J. Symbolic Comput. 24.3-4 (1997), pp. 235–265,
DOI: https://doi.org/10.1006/jsco.1996.0125.

[2] A. Choudhry: *On arithmetic progressions of equal lengths and equal products of terms*, Acta Arith. LXXXII.I (1997), pp. 95–97,
doi: https://doi.org/10.4064/aa-82-1-95-97.

[3] A. Choudhry: *Several arithmetic progressions of equal lengths and equal products of terms*, L'Enseignement Math. 53 (2007), pp. 87–95.

[4] A. Choudhry: *Symmetric Diophantine equations*, Rocky Mountain Math. J. 34.4 (2004), pp. 1281–1298,
doi: https://doi.org/10.1216/rmjm/1181069800.

[5] Y. Gabovich: *On arithmetic progressions with equal product of terms*, Colloq. Math. 15 (1966), pp. 45–48,
doi: https://doi.org/10.4064/cm-15-1-45-48.

[6] R. Kloosterman: *Elliptic $K3$ surfaces with geometric Mordell-Weil rank 15*, Canad. Math. Bull. 50.2 (2007), pp. 215–226,
doi: https://doi.org/10.4153/CMB-2007-023-2.

[7] M. Kuwata: *The canonical height and elliptic surfaces*, J. Number Theory 36.2 (1990), pp. 201–211,
doi: https://doi.org/10.1016/0022-314X(90)90073-Z.

[8] K. Oguiso, T. Shioda: *The Mordell-Weil lattice of a rational elliptic surface*, Commentarii Mathematici Universitatis Sancti Pauli 40.1 (1991), pp. 83–99.

[9] N. Saradha, T. N. Shorey, R. Tijdeman: *On arithmetic progressions of equal lengths with equal products*, Math. Proc. Cambridge Phil. Soc. 117 (1995), pp. 193–201,
doi: https://doi.org/10.1017/S0305004100073047.

[10] N. Saradha, T. N. Shorey, R. Tijdeman: *On the equation $x(x+1)\dots(x+k-1) = y(y+d)\dots(y+(mk-1)d)$, $m = 1, 2$*, Acta Arith. 71 (1995), pp. 181–196,
doi: https://doi.org/10.4064/aa-71-2-181-196.

[11] T. Shioda: *On elliptic modular surfaces*, J. Math. Soc. Japan 24 (1972), pp. 20–59,
doi: https://doi.org/10.2969/jmsj/02410020.

Annales Mathematicae et Informaticae
***52** (2020) pp. 57–70*
DOI: https://doi.org/10.33039/ami.2020.05.003
URL: https://ami.uni-eszterhazy.hu

A note on dual third-order Jacobsthal vectors

Gamaliel Cerda-Morales

Instituto de Matemáticas, Pontificia Universidad Católica de Valparaíso
Blanco Viel 596, Valparaíso, Chile
gamaliel.cerda.m@mail.pucv.cl

Submitted: January 10, 2018
Accepted: May 4, 2020
Published online: May 8, 2020

Dedicated to my daughter Julieta

Abstract

Third-order Jacobsthal quaternions are first defined by [5]. In this study, dual third-order Jacobsthal and dual third-order Jacobsthal–Lucas numbers are defined. Furthermore, we work on these dual numbers and we obtain the properties e.g. linear and quadratic identities, summation, norm, negative dual third-order Jacobsthal identities, Binet formulas and relations of them. We also define new vectors which are called dual third-order Jacobsthal vectors and dual third-order Jacobsthal–Lucas vectors. We give properties of these vectors to exert in geometry of dual space.

Keywords: Dual numbers, Jacobsthal numbers, Recurrences, Third-order Jacobsthal numbers, Third-order Jacobsthal–Lucas numbers.

MSC: Primary 11B39; Secondary 11R52, 05A15.

1. Introduction

Dual numbers which have lots of applications to modelling plane joint, to screw systems and to mechanics, were first invented by W. K. Clifford in 1873. The dual numbers extend to the real numbers has the form $d = a + \varepsilon b$, where ε is the dual unit and $\varepsilon^2 = 0$, $\varepsilon \neq 0$. The set $\mathbb{D} = \mathbb{R}[\varepsilon] = \{a+\varepsilon b : a, b \in \mathbb{R}\}$ is called dual number

system and forms two dimensional commutative associative algebra over the real numbers. The algebra of dual numbers is a ring with the following addition and multiplication operations

$$\begin{aligned}(a_1+\varepsilon b_1)\pm(a_2+\varepsilon b_2)&=(a_1\pm a_2)+\varepsilon(b_1\pm b_2),\\ (a_1+\varepsilon b_1)\cdot(a_2+\varepsilon b_2)&=a_1a_2+\varepsilon(a_1b_2+a_2b_2).\end{aligned}\tag{1.1}$$

The equality of two dual numbers $d_1=a_1+\varepsilon b_1$ and $d_2=a_2+\varepsilon b_2$ is defined as, $d_1=d_2$ if and only if $a_1=a_2$ and $b_1=b_2$. The division of two dual numbers provided $a_2\neq 0$ is given by

$$\frac{d_1}{d_2}=\frac{a_1}{a_2}+\varepsilon\left(\frac{b_1a_2-a_1b_2}{a_2^2}\right).$$

The conjugate of the dual number $d=a+\varepsilon b$ is $\overline{d}=a-\varepsilon b$.

Vectors are used to study the analytic geometry of space, where they give simple ways to describe lines, planes, surfaces and curves in space. In this work we will speak on vectors of dual space using third order Jacobsthal numbers.

Now, the set $\mathbb{D}^3=\{\overrightarrow{a}+\varepsilon\overrightarrow{b}:\ \overrightarrow{a},\overrightarrow{b}\in\mathbb{R}^3\}$ is a module on the ring $\mathbb{D}$ which is called $\mathbb{D}$-Module and the members of $\mathbb{D}^3$ are called dual vectors consisting of two real vectors. Also a dual vector $\overrightarrow{d}=\overrightarrow{a}+\varepsilon\overrightarrow{b}$ has another expression of the form

$$\overrightarrow{d}=(a_1+\varepsilon b_1,a_2+\varepsilon b_2,a_3+\varepsilon b_3)=(d_1,d_2,d_3),$$

where d_1, d_2, d_3 are dual numbers and $\overrightarrow{a}=(a_1,a_2,a_3)$, $\overrightarrow{b}=(b_1,b_2,b_3)$.

The norm of the dual vector $\overrightarrow{d}$ is given by

$$\left\|\overrightarrow{d}\right\|=\|\overrightarrow{a}\|+\varepsilon\frac{\langle\overrightarrow{a},\overrightarrow{b}\rangle}{\|\overrightarrow{a}\|},\tag{1.2}$$

where $\langle\overrightarrow{a},\overrightarrow{b}\rangle=a_1b_1+a_2b_2+a_3b_3$. Furthermore, $\overrightarrow{d}=\overrightarrow{a}+\varepsilon\overrightarrow{b}$ is dual unit vector (e.g. $\left\|\overrightarrow{d}\right\|=1$) if and only if $\|\overrightarrow{a}\|=1$ and $\langle\overrightarrow{a},\overrightarrow{b}\rangle=0$.

The dual unit vectors are related with oriented lines, found by E. Study, which is called Study mapping: The oriented lines in $\mathbb{R}^3$ are in one-to-one correspondence with the points of dual unit sphere in $\mathbb{D}^3$.

On the other hand, the Jacobsthal numbers have many interesting properties and applications in many fields of science (see, e.g., [2]). The Jacobsthal numbers J_n are defined by the recurrence relation

$$J_0=0,\ J_1=1,\ J_{n+2}=J_{n+1}+2J_n,\ n\geq 0.\tag{1.3}$$

Another important sequence is the Jacobsthal-Lucas sequence. This sequence is defined by the recurrence relation $j_0=2,\ j_1=1,\ j_{n+1}=j_n+2j_{n-1},\ n\geq 1$ (see [12]).

In [7] the Jacobsthal recurrence relation (1.3) is extended to higher order recurrence relations and the basic list of identities provided by A. F. Horadam [12] is

expanded and extended to several identities for some of the higher order cases. In fact, third-order Jacobsthal numbers, $\{J_n^{(3)}\}_{n\geq 0}$, and third-order Jacobsthal–Lucas numbers, $\{j_n^{(3)}\}_{n\geq 0}$, are defined by

$$J_{n+3}^{(3)} = J_{n+2}^{(3)} + J_{n+1}^{(3)} + 2J_n^{(3)},\ J_0^{(3)} = 0,\ J_1^{(3)} = J_2^{(3)} = 1,\ n \geq 0, \tag{1.4}$$

and

$$j_{n+3}^{(3)} = j_{n+2}^{(3)} + j_{n+1}^{(3)} + 2j_n^{(3)},\ j_0^{(3)} = 2,\ j_1^{(3)} = 1,\ j_2^{(3)} = 5,\ n \geq 0, \tag{1.5}$$

respectively.

Some of the following properties given for third-order Jacobsthal numbers and third-order Jacobsthal–Lucas numbers are used in this paper (for more details, see [5–7]). Note that Eqs. (1.9) and (1.12) have been corrected in this paper, since they have been wrongly described in [7]:

$$3J_n^{(3)} + j_n^{(3)} = 2^{n+1},$$

$$j_n^{(3)} - 3J_n^{(3)} = 2j_{n-3}^{(3)}, \tag{1.6}$$

$$J_{n+2}^{(3)} - 4J_n^{(3)} = \begin{cases} -2 & \text{if} \quad n \equiv 1 \pmod 3 \\ 1 & \text{if} \quad n \not\equiv 1 \pmod 3 \end{cases}, \tag{1.7}$$

$$j_n^{(3)} - 4J_n^{(3)} = \begin{cases} 2 & \text{if} \quad n \equiv 0 \pmod 3 \\ -3 & \text{if} \quad n \equiv 1 \pmod 3 \\ 1 & \text{if} \quad n \equiv 2 \pmod 3 \end{cases}, \tag{1.8}$$

$$j_{n+1}^{(3)} + j_n^{(3)} = 3J_{n+2}^{(3)}, \tag{1.9}$$

$$j_n^{(3)} - J_{n+2}^{(3)} = \begin{cases} 1 & \text{if} \quad n \equiv 0 \pmod 3 \\ -1 & \text{if} \quad n \equiv 1 \pmod 3 \\ 0 & \text{if} \quad n \equiv 2 \pmod 3 \end{cases}, \tag{1.10}$$

$$\left(j_{n-3}^{(3)}\right)^2 + 3J_n^{(3)} j_n^{(3)} = 4^n,$$

$$\sum_{k=0}^{n} J_k^{(3)} = \begin{cases} J_{n+1}^{(3)} & \text{if} \quad n \not\equiv 0 \pmod 3 \\ J_{n+1}^{(3)} - 1 & \text{if} \quad n \equiv 0 \pmod 3 \end{cases} \tag{1.11}$$

and

$$\left(j_n^{(3)}\right)^2 - 9\left(J_n^{(3)}\right)^2 = 2^{n+2} j_{n-3}^{(3)}. \tag{1.12}$$

Using standard techniques for solving recurrence relations, the auxiliary equation, and its roots are given by

$$x^3 - x^2 - x - 2 = 0;\ x_1 = 2,\ x_2 = \frac{-1 + i\sqrt{3}}{2} \text{ and } x_3 = \frac{-1 - i\sqrt{3}}{2}.$$

Note that the latter two are the complex conjugate cube roots of unity. Call them $x_1 = \omega_1$ and $x_2 = \omega_2$, respectively. Thus the Binet formulas can be written as

$$J_n^{(3)} = \frac{2}{7} \cdot 2^n - \left(\frac{3 + 2i\sqrt{3}}{21} \right) \omega_1^n - \left(\frac{3 - 2i\sqrt{3}}{21} \right) \omega_2^n \tag{1.13}$$

and

$$j_n^{(3)} = \frac{8}{7} \cdot 2^n + \left(\frac{3 + 2i\sqrt{3}}{7} \right) \omega_1^n + \left(\frac{3 - 2i\sqrt{3}}{7} \right) \omega_2^n, \tag{1.14}$$

respectively.

A variety of new results on Fibonacci-like quaternion and octonion numbers can be found in several papers [4–6, 10, 11, 13, 14]. The origin of the topic of number sequences in division algebra can be traced back to the works by Horadam in [11] and by Iyer in [14]. Horadam [11] defined the quaternions with the classic Fibonacci and Lucas number components as

$$QF_n = F_n + F_{n+1}\mathbf{i} + F_{n+2}\mathbf{j} + F_{n+3}\mathbf{k}$$

and

$$QL_n = L_n + L_{n+1}\mathbf{i} + L_{n+2}\mathbf{j} + L_{n+3}\mathbf{k},$$

respectively, where F_n and L_n are the n-th classic Fibonacci and Lucas numbers, respectively, and the author studied the properties of these quaternions. Several interesting and useful extensions of many of the familiar quaternion numbers (such as the Fibonacci and Lucas quaternions [1, 10, 11] have been considered by several authors.

There has been an increasing interest on quaternions and octonions that play an important role in various areas such as computer sciences, physics, differential geometry, quantum physics, signal, color image processing and geostatics (for more, see [3, 8, 15]). For example, in [5, 6] the author studied the third-order Jacobsthal quaternions and give some interesting properties of this numbers.

In this paper, we give some properties and relations of dual third-order Jacobsthal and dual third-order Jacobsthal–Lucas numbers. Then, we define dual third-order Jacobsthal vectors and investigate geometric notions which are created by using dual third-order Jacobsthal vectors.

2. Dual third-order Jacobsthal numbers

In this section, we define new kinds of sequences of dual number called as dual third-order Jacobsthal numbers and dual third-order Jacobsthal–Lucas numbers. We study some properties of these numbers. We obtain various results for these classes of dual numbers included recurrence relations, summation formulas, Binet's formulas and generating functions.

In [9], the authors introduced the so-called dual Fibonacci numbers, which are a new class of dual numbers. They are defined by

$$FD_n = F_n + \varepsilon F_{n+1},\ (n \geq 0) \tag{2.1}$$

where F_n is the n-th Fibonacci number, $\varepsilon^2 = 0$ and $\varepsilon \neq 0$.

We now consider the usual third-order Jacobsthal and third-order Jacobsthal–Lucas numbers, and based on the definition (2.1) we give definitions of new kinds of dual numbers, which we call the dual third-order Jacobsthal numbers and dual third-order Jacobsthal–Lucas numbers. In this paper, we define the n-th dual third-order Jacobsthal number and dual third-order Jacobsthal–Lucas number, respectively, by the following recurrence relations

$$JD_n^{(3)} = J_n^{(3)} + \varepsilon J_{n+1}^{(3)},\ n \geq 0 \tag{2.2}$$

and

$$jD_n^{(3)} = j_n^{(3)} + \varepsilon j_{n+1}^{(3)},\ n \geq 0, \tag{2.3}$$

where $J_n^{(3)}$ and $j_n^{(3)}$ are the n-th third-order Jacobsthal number and third-order Jacobsthal–Lucas number, respectively.

The equalities in (1.1) gives

$$JD_n^{(3)} \pm jD_n^{(3)} = (J_n^{(3)} \pm j_n^{(3)}) + \varepsilon(J_{n+1}^{(3)} \pm j_{n+1}^{(3)}). \tag{2.4}$$

From the conjugate of a dual number, (2.2) and (2.3) an easy computation gives

$$\overline{JD_n^{(3)}} = J_n^{(3)} - \varepsilon J_{n+1}^{(3)},\ \overline{jD_n^{(3)}} = j_n^{(3)} - \varepsilon j_{n+1}^{(3)}.$$

By some elementary calculations we find the following recurrence relations for the dual third-order Jacobsthal and dual third-order Jacobsthal–Lucas numbers from (2.2), (2.3), (2.4), (1.1), (1.4) and (1.5):

$$\begin{aligned} JD_{n+1}^{(3)} + JD_n^{(3)} + 2JD_{n-1}^{(3)} &= (J_{n+1}^{(3)} + J_n^{(3)} + 2J_{n-1}^{(3)}) + \varepsilon(J_{n+2}^{(3)} + J_{n+1}^{(3)} + 2J_n^{(3)}) \\ &= J_{n+2}^{(3)} + \varepsilon J_{n+3}^{(3)} \\ &= JD_{n+2}^{(3)} \end{aligned} \tag{2.5}$$

and similarly $jD_{n+2}^{(3)} = jD_{n+1}^{(3)} + jD_n^{(3)} + 2jD_{n-1}^{(3)}$, for $n \geq 1$.

Now, we will state Binet's formulas for the dual third-order Jacobsthal and dual third-order Jacobsthal–Lucas numbers. Repeated use of (1.13) in (2.2) enables one to write for $\underline{\alpha} = 1 + 2\varepsilon$, $\underline{\omega_1} = 1 + \omega_1\varepsilon$ and $\underline{\omega_2} = 1 + \omega_2\varepsilon$

$$\begin{aligned} JD_n^{(3)} &= J_n^{(3)} + \varepsilon J_{n+1}^{(3)} \\ &= \frac{1}{7}2^{n+1} - \frac{3+2i\sqrt{3}}{21}\omega_1^n - \frac{3-2i\sqrt{3}}{21}\omega_2^n \\ &\quad + \varepsilon\left(\frac{1}{7}2^{n+2} - \frac{3+2i\sqrt{3}}{21}\omega_1^{n+1} - \frac{3-2i\sqrt{3}}{21}\omega_2^{n+1}\right) \\ &= \frac{1}{7}\underline{\alpha}2^{n+1} - \frac{3+2i\sqrt{3}}{21}\underline{\omega_1}\omega_1^n - \frac{3-2i\sqrt{3}}{21}\underline{\omega_2}\omega_2^n \end{aligned} \tag{2.6}$$

and similarly making use of (1.14) in (2.3) yields

$$\begin{aligned} jD_n^{(3)} &= j_n^{(3)} + \varepsilon j_{n+1}^{(3)} \\ &= \frac{1}{7}2^{n+3} + \frac{3+2i\sqrt{3}}{7}\omega_1^n + \frac{3-2i\sqrt{3}}{7}\omega_2^n \\ &\quad + \varepsilon\left(\frac{1}{7}2^{n+4} + \frac{3+2i\sqrt{3}}{7}\omega_1^{n+1} + \frac{3-2i\sqrt{3}}{7}\omega_2^{n+1}\right) \\ &= \frac{1}{7}\underline{\alpha}2^{n+3} + \frac{3+2i\sqrt{3}}{7}\underline{\omega_1}\omega_1^n + \frac{3-2i\sqrt{3}}{7}\underline{\omega_2}\omega_2^n. \end{aligned} \tag{2.7}$$

The formulas in (2.6) and (2.7) are called as Binet's formulas for the dual third-order Jacobsthal and dual third-order Jacobsthal–Lucas numbers, respectively. The recurrence relations for the n-th dual third-order Jacobsthal number are expressed in the following theorem.

Theorem 2.1. *For $n, m \geq 0$, we have the following identities:*

$$JD_{n+2}^{(3)} + JD_{n+1}^{(3)} + JD_n^{(3)} = 2^{n+1}(1+2\varepsilon),$$

$$JD_{n+2}^{(3)} - 4JD_n^{(3)} = \begin{cases} 1-2\varepsilon & \text{if} \quad n \equiv 0 \pmod 3 \\ -2+\varepsilon & \text{if} \quad n \equiv 1 \pmod 3 \\ 1+\varepsilon & \text{if} \quad n \equiv 2 \pmod 3 \end{cases}, \tag{2.8}$$

$$JD_n^{(3)}JD_{m+1}^{(3)} + T_{n-1}^{(3)}JD_m^{(3)} + 2JD_{n-1}^{(3)}JD_{m-1}^{(3)} = JD_{n+m}^{(3)} + \varepsilon J_{n+m+1}^{(3)},$$

$$\left(JD_{n+1}^{(3)}\right)^2 + \left(JD_n^{(3)}\right)^2 + 4JD_n^{(3)}JD_{n-1}^{(3)} = JD_{2n+1}^{(3)} + \varepsilon J_{2n+2}^{(3)}, \tag{2.9}$$

where $T_n^{(3)} = JD_n^{(3)} + 2JD_{n-1}^{(3)}$.

Proof. Consider (2.2) and (2.4) we can write

$$JD_{n+2}^{(3)} + JD_{n+1}^{(3)} + JD_n^{(3)} = J_{n+2}^{(3)} + J_{n+1}^{(3)} + J_n^{(3)} + \varepsilon(J_{n+3}^{(3)} + J_{n+2}^{(3)} + J_{n+1}^{(3)}).$$

Using the identity $J_{n+2}^{(3)} + J_{n+1}^{(3)} + J_n^{(3)} = 2^{n+1}$, the above sum can be calculated as

$$JD_{n+2}^{(3)} + JD_{n+1}^{(3)} + JD_n^{(3)} = 2^{n+1} + 2^{n+2}\varepsilon,$$

which can be simplified as $JD_{n+2}^{(3)} + JD_{n+1}^{(3)} + JD_n^{(3)} = 2^{n+1}(1+2\varepsilon)$. Now, using (1.7) and (2.2) we can write $JD_{n+2}^{(3)} - 4JD_n^{(3)} = 1-2\varepsilon$ if $n \equiv 1 (\text{mod } 3)$ and similarly in the other cases, this proves (2.8). Now, from the definition of third order Jacobsthal number, dual third order Jacobsthal number in Eq. (2.2), the equations

$$\left(J_{n+1}^{(3)}\right)^2 + \left(J_n^{(3)}\right)^2 + 4J_n^{(3)}J_{n-1}^{(3)} = J_{2n+1}^{(3)}$$

and $J_n^{(3)}J_{m+1}^{(3)} + (J_{n-1}^{(3)} + 2J_{n-2}^{(3)})J_m^{(3)} + 2J_{n-1}^{(3)}J_{m-1}^{(3)} = J_{n+m}^{(3)}$ (see Waddill and Sacks [16]), we get

$$\begin{aligned}
JD_n^{(3)}JD_{m+1}^{(3)} &+ (JD_{n-1}^{(3)} + 2JD_{n-2}^{(3)})JD_m^{(3)} + 2JD_{n-1}^{(3)}JD_{m-1}^{(3)} \\
&= (J_n^{(3)} + \varepsilon J_{n+1}^{(3)})(J_{m+1}^{(3)} + \varepsilon J_{m+2}^{(3)}) \\
&\quad + ((J_{n-1}^{(3)} + 2J_{n-2}^{(3)}) + \varepsilon(J_n^{(3)} + 2J_{n-1}^{(3)}))(J_m^{(3)} + \varepsilon J_{m+1}^{(3)}) \\
&\quad + 2(J_{n-1}^{(3)} + \varepsilon J_n^{(3)})(J_{m-1}^{(3)} + \varepsilon J_m^{(3)}) \\
&= (J_n^{(3)}J_{m+1}^{(3)} + (J_{n-1}^{(3)} + 2J_{n-2}^{(3)})J_m^{(3)} + 2J_{n-1}^{(3)}J_{m-1}^{(3)}) \\
&\quad + \varepsilon(J_n^{(3)}J_{m+2}^{(3)} + (J_{n-1}^{(3)} + 2J_{n-2}^{(3)})J_{m+1}^{(3)} + 2J_{n-1}^{(3)}J_m^{(3)}) \\
&\quad + \varepsilon(J_{n+1}^{(3)}J_{m+1}^{(3)} + (J_n^{(3)} + 2J_{n-1}^{(3)})J_m^{(3)} + 2J_n^{(3)}J_{m-1}^{(3)}) \\
&= (J_{n+m}^{(3)} + \varepsilon J_{n+m+1}^{(3)}) + \varepsilon J_{n+m+1}^{(3)} \\
&= JD_{n+m}^{(3)} + \varepsilon J_{n+m+1}^{(3)}.
\end{aligned} \tag{2.10}$$

Finally, if we consider first $n = n + 1$ and $m = n$ in above result (2.10), we obtain

$$\left(JD_{n+1}^{(3)}\right)^2 + \left(JD_n^{(3)}\right)^2 + 4JD_n^{(3)}JD_{n-1}^{(3)} = JD_{2n+1}^{(3)} + \varepsilon J_{2n+2}^{(3)},$$

which is the assertion (2.9) of theorem. □

The following theorem deals with two relations between the dual third-order Jacobsthal and dual third-order Jacobsthal–Lucas numbers.

Theorem 2.2. *Let $n \geq 0$ be integer. Then,*

$$jD_{n+3}^{(3)} - 3JD_{n+3}^{(3)} = 2jD_n^{(3)}, \tag{2.11}$$

$$jD_n^{(3)} + jD_{n+1}^{(3)} = 3JD_{n+2}^{(3)}, \tag{2.12}$$

$$jD_n^{(3)} - JD_{n+2}^{(3)} = \begin{cases} 1-\varepsilon & \text{if} \quad n \equiv 0 \pmod 3 \\ -1 & \text{if} \quad n \equiv 1 \pmod 3 \\ \varepsilon & \text{if} \quad n \equiv 2 \pmod 3 \end{cases}, \tag{2.13}$$

$$jD_n^{(3)} - 4JD_n^{(3)} = \begin{cases} 2-3\varepsilon & \text{if} \quad n \equiv 0 \pmod 3 \\ -3+\varepsilon & \text{if} \quad n \equiv 1 \pmod 3 \\ 1+2\varepsilon & \text{if} \quad n \equiv 2 \pmod 3 \end{cases}. \tag{2.14}$$

Proof. The following recurrence relation

$$jD_{n+3}^{(3)} - 3JD_{n+3}^{(3)} = (j_{n+3}^{(3)} - 3J_{n+3}^{(3)}) + \varepsilon(j_{n+4}^{(3)} - 3J_{n+4}^{(3)}) \tag{2.15}$$

can be readily written considering that $JD_n^{(3)} = J_n^{(3)} + \varepsilon J_{n+1}^{(3)}$ and $jD_n^{(3)} = j_n^{(3)} + \varepsilon j_{n+1}^{(3)}$. Notice that $j_{n+3}^{(3)} - 3J_{n+3}^{(3)} = 2j_n^{(3)}$ from (1.6) (see [7]), whence it follows that

(2.15) can be rewritten as $jD^{(3)}_{n+3}-3JD^{(3)}_{n+3}=2jD^{(3)}_n$ from which the desired result (2.11) of Theorem 2.2. In a similar way we can show the second equality. By using the identity $j^{(3)}_n+j^{(3)}_{n+1}=3J^{(3)}_{n+2}$ we have

$$jD^{(3)}_n+jD^{(3)}_{n+1}=3(J^{(3)}_{n+2}+\varepsilon J^{(3)}_{n+3}),$$

which is the assertion (2.12) of theorem.

By using the identity $j^{(3)}_n-J^{(3)}_{n+2}=1$ from (1.10) (see [7]) we have

$$jD^{(3)}_n-JD^{(3)}_{n+2}=(j^{(3)}_n-J^{(3)}_{n+2})+\varepsilon(j^{(3)}_{n+1}-J^{(3)}_{n+3})=1-\varepsilon$$

if $n\equiv 0(\text{mod } 3)$, the other identities are clear from equation (1.10). Finally, the proof of Eq. (2.14) is similar to (2.13) by using (1.8). □

Now, we use the notation

$$H_n(a,b)=\frac{A\omega_1^n-B\omega_2^n}{\omega_1-\omega_2}=\begin{cases} a & \text{if } n\equiv 0 \pmod 3\\ b & \text{if } n\equiv 1 \pmod 3\\ -(a+b) & \text{if } n\equiv 2 \pmod 3\end{cases}, \tag{2.16}$$

where $A=b-a\omega_2$ and $B=b-a\omega_1$, in which ω_1 and ω_2 are the complex conjugate cube roots of unity (i.e. $\omega_1^3=\omega_2^3=1$). Furthermore, note that for all $n\geq 0$ we have

$$H_{n+2}(a,b)=-H_{n+1}(a,b)-H_n(a,b),$$

where $H_0(a,b)=a$ and $H_1(a,b)=b$.

From the Binet formulas (1.13), (1.14) and Eq. (2.16), we have

$$J^{(3)}_n=\frac{1}{7}\left(2^{n+1}-V_n\right) \text{ and } j^{(3)}_n=\frac{1}{7}\left(2^{n+3}+3V_n\right),$$

where $V_n=H_n(2,-3)$. Then, for $m\geq n$:

$$\begin{aligned} J^{(3)}_mJ^{(3)}_{n+1}-J^{(3)}_{m+1}J^{(3)}_n&=\frac{1}{49}\begin{pmatrix}(2^{m+1}-V_m)(2^{n+2}-V_{n+1})\\ -(2^{m+2}-V_{m+1})(2^{n+1}-V_n)\end{pmatrix}\\ &=\frac{1}{49}\begin{pmatrix}-2^{m+1}V_{n+1}-2^{n+2}V_m+2^{m+2}V_n+2^{n+1}V_{m+1}\\ +V_mV_{n+1}-V_{m+1}V_n\end{pmatrix}\\ &=\frac{1}{7}\left(\ 2^{m+1}U_{n+1}-2^{n+1}U_{m+1}+U_{m-n}\ \right), \end{aligned}\tag{2.17}$$

where $U_{n+1}=\frac{1}{7}(2V_n-V_{n+1})=H_{n+1}(0,1)$ and $V_n=H_n(2,-3)$. Furthermore, if $m=n+1$ in Eq. (2.17), we obtain for $n\geq 0$,

$$J^{(3)}_{n+2}J^{(3)}_n-\left(J^{(3)}_{n+1}\right)^2=\frac{1}{7}\left(2^{n+1}V_{-(n+2)}-1\right), \tag{2.18}$$

where $V_{-n}=U_n-2U_{n+2}=H_n(2,1)$.

Using the above notation, the following theorem investigate a type of Cassini identity for this numbers.

Theorem 2.3. *For $n \geq 0$, the Cassini-like identity for dual third-order Jacobsthal number $JD_n^{(3)}$ is given by*

$$\begin{aligned} &JD_{n+2}^{(3)} JD_n^{(3)} - \left(JD_{n+1}^{(3)}\right)^2 \\ &= \frac{1}{7}\left(2^{n+1} VD_{-(n+2)} + (-1 + \varepsilon(2^{n+2} V_{-(n+2)} + 1))\right), \end{aligned} \tag{2.19}$$

where $V_{-n} = H_n(2,1)$ and $VD_{-n} = V_{-n} + \varepsilon V_{-(n+1)}$.

Proof. From Eqs. (2.2) and (2.4), the identity (2.18) for third-order Jacobsthal numbers and $n = m + 2$ in Eq. (2.17), we get

$$\begin{aligned} &JD_{n+2}^{(3)} JD_n^{(3)} - \left(JD_{n+1}^{(3)}\right)^2 \\ &= \left(J_{n+2}^{(3)} + \varepsilon J_{n+3}^{(3)}\right)\left(J_n^{(3)} + \varepsilon J_{n+1}^{(3)}\right) - \left(J_{n+1}^{(3)} + \varepsilon J_{n+2}^{(3)}\right)^2 \\ &= \left(J_{n+2}^{(3)} J_n^{(3)} - \left(J_{n+1}^{(3)}\right)^2\right) + \varepsilon\left(J_{n+3}^{(3)} J_n^{(3)} - J_{n+1}^{(3)} J_{n+2}^{(3)}\right) \\ &= \frac{1}{7}\left(2^{n+1} V_{-(n+2)} - 1\right) + \frac{\varepsilon}{7}\left(2^{n+1}(V_{-n} + 2V_{-(n+2)}) + 1\right), \end{aligned}$$

where $U_n - 4U_{n+1} = V_{-n} + 2V_{-(n+2)} = H_n(-4,5)$.

Furthermore, using $VD_{-(n+2)} = V_{-(n+2)} + \varepsilon V_{-n}$, we obtain the next result

$$\begin{aligned} JD_{n+2}^{(3)} JD_n^{(3)} - \left(JD_{n+1}^{(3)}\right)^2 &= \frac{1}{7}\left(2^{n+1} V_{-(n+2)} - 1 + 2^{n+1}\varepsilon(V_{-n} + 2V_{-(n+2)}) + \varepsilon\right) \\ &= \frac{1}{7}\left(2^{n+1} VD_{-(n+2)} + (-1 + \varepsilon(2^{n+2} V_{-(n+2)} + 1))\right). \end{aligned}$$

we reach (2.19). □

Theorem 2.4. *If $JD_n^{(3)}$ is a dual third-order Jacobsthal number, then the limit of consecutive quotients of this numbers is*

$$\lim_{n\to\infty} \frac{JD_{n+1}^{(3)}}{JD_n^{(3)}} = \lim_{n\to\infty}\left(\frac{J_{n+1}^{(3)} + \varepsilon J_{n+2}^{(3)}}{J_n^{(3)} + \varepsilon J_{n+1}^{(3)}}\right) = 2. \tag{2.20}$$

Proof. The limit of consecutive quotients of third order Jacosbthal numbers approaches to the radio $\frac{J_{n+1}^{(3)}}{J_n^{(3)}} \to 2$ if $n \to \infty$ (See [7]). From the previous limit, Eqs. (2.2) and (2.18), we have

$$\begin{aligned} \lim_{n\to\infty} \frac{J_{n+1}^{(3)} + \varepsilon J_{n+2}^{(3)}}{J_n^{(3)} + \varepsilon J_{n+1}^{(3)}} &= \lim_{n\to\infty}\left(\frac{J_n^{(3)} J_{n+1}^{(3)} + \varepsilon\left(J_{n+2}^{(3)} J_n^{(3)} - \left(J_{n+1}^{(3)}\right)^2\right)}{\left(J_n^{(3)}\right)^2}\right) \\ &= \lim_{n\to\infty} \frac{J_{n+1}^{(3)}}{J_n^{(3)}} + \varepsilon \lim_{n\to\infty}\left(\frac{2^{n+1} V_{-(n+2)} - 1}{7\left(J_n^{(3)}\right)^2}\right), \end{aligned} \tag{2.21}$$

where $V_{-n} = H_n(2,1)$. In last equality of (2.21), by using $V_{-n} = H_n(2,1)$ (see Eq. (2.16)), $\lim_{n\to\infty} \frac{2^{n+1}}{7J_n^{(3)}} = 1$ and

$$\lim_{n\to\infty}\left(\frac{J_{n+2}^{(3)} - 2J_{n+1}^{(3)}}{J_n^{(3)}}\right) = \lim_{n\to\infty}\left(\frac{J_{n+2}^{(3)}}{J_n^{(3)}} - 4\frac{J_{n+1}^{(3)}}{J_n^{(3)}}\right) = 0,$$

we find zero for the second limit. Thus, the result (2.20) is true. □

3. Dual third-order Jacobsthal vectors

A dual vector in $\mathbb{D}^3$ is given by $\overrightarrow{d} = \overrightarrow{a} + \varepsilon\overrightarrow{b}$ where $\overrightarrow{a}$, $\overrightarrow{b} \in \mathbb{R}^3$. Now, we will give dual third-order Jacobsthal vectors and geometric properties of them.

A dual third-order Jacobsthal vector is defined by

$$\overrightarrow{JD_n^{(3)}} = \overrightarrow{J_n^{(3)}} + \varepsilon\overrightarrow{J_{n+1}^{(3)}},\ n \geq 0, \tag{3.1}$$

where $\overrightarrow{J_n^{(3)}} = \left(J_n^{(3)}, J_{n+1}^{(3)}, J_{n+2}^{(3)}\right)$ and $\overrightarrow{J_{n+1}^{(3)}} = \left(J_{n+1}^{(3)}, J_{n+2}^{(3)}, J_{n+3}^{(3)}\right)$ are real vectors in $\mathbb{R}^3$ with n-th third-order Jacobsthal number $J_n^{(3)}$.

The dual third-order Jacobsthal vector $\overrightarrow{JD_n^{(3)}}$ is also can be expressed as

$$\overrightarrow{JD_n^{(3)}} = \left(JD_n^{(3)}, JD_{n+1}^{(3)}, JD_{n+2}^{(3)}\right),$$

where $JD_n^{(3)}$ is the n-th dual third-order Jacobsthal number. For example, the first three dual third-order Jacobsthal vectors can be given easily as

$$\begin{aligned}
\overrightarrow{JD_0^{(3)}} &= (\varepsilon, 1+\varepsilon, 1+2\varepsilon),\\
\overrightarrow{JD_1^{(3)}} &= (1+\varepsilon, 1+2\varepsilon, 2+5\varepsilon),\\
\overrightarrow{JD_2^{(3)}} &= (1+2\varepsilon, 2+5\varepsilon, 5+9\varepsilon).
\end{aligned}$$

Let $\overrightarrow{JD_n^{(3)}}$ and $\overrightarrow{JD_m^{(3)}}$ be two dual third-order Jacobsthal vectors and $\lambda \in \mathbb{R}[\varepsilon]$ be a dual number. Then the product of the dual third-order Jacobsthal vector and the scalar λ is given by

$$\lambda \cdot \overrightarrow{JD_n^{(3)}} = \lambda\overrightarrow{J_n^{(3)}} + \varepsilon\lambda\overrightarrow{J_{n+1}^{(3)}}.$$

Furthermore, $\overrightarrow{JD_n^{(3)}} = \overrightarrow{JD_m^{(3)}}$ if and only if $JD_n^{(3)} = JD_m^{(3)}$, $JD_{n+1}^{(3)} = JD_{m+1}^{(3)}$ and $JD_{n+2}^{(3)} = JD_{m+2}^{(3)}$, where $JD_n^{(3)} = J_n^{(3)} + \varepsilon J_{n+1}^{(3)}$.

Theorem 3.1. *The dual third-order Jacobsthal vector* $\overrightarrow{JD_n^{(3)}}$ *is a dual unit vector if and only if*

$$3 \cdot 2^{2(n+1)} - 2^{n+2}U_n = 5 \text{ and } 3 \cdot 2^{2n+3} - 2^{n+1}(U_n - U_{n+2}) = 1, \tag{3.2}$$

where $U_n = H_n(0,1)$.

Proof. By using the definitions of third-order Jacobsthal numbers, Eq. (3.1) and the identities $V_nV_{n+1} + V_{n+1}V_{n+2} + V_{n+2}V_{n+3} = -7$ and $V_n^2 + V_{n+1}^2 + V_{n+2}^2 = 14$ (see Eq. (2.16)) we get the following statements

$$\begin{aligned}
\left\|\overrightarrow{J_n^{(3)}}\right\|^2 &= \left(J_n^{(3)}\right)^2 + \left(J_{n+1}^{(3)}\right)^2 + \left(J_{n+2}^{(3)}\right)^2 \\
&= \frac{1}{49}\left(\left(2^{n+1} - V_n\right)^2 + \left(2^{n+2} - V_{n+1}\right)^2 + \left(2^{n+3} - V_{n+2}\right)^2\right) \\
&= \frac{1}{49}\left(21 \cdot 2^{2(n+1)} - 2^{n+2}(V_n + 2V_{n+1} + 4V_{n+2}) + 14\right) \\
&= \frac{1}{7}\left(3 \cdot 2^{2(n+1)} - 2^{n+2}U_n + 2\right)
\end{aligned}$$

and

$$\begin{aligned}
&J_n^{(3)}J_{n+1}^{(3)} + J_{n+1}^{(3)}J_{n+2}^{(3)} + J_{n+2}^{(3)}J_{n+3}^{(3)} \\
&\quad= \frac{1}{49}\left(\begin{array}{c}(2^{n+1} - V_n)(2^{n+2} - V_{n+1}) + (2^{n+2} - V_{n+1})(2^{n+3} - V_{n+2}) \\ +(2^{n+3} - V_{n+2})(2^{n+4} - V_{n+3})\end{array}\right) \\
&\quad= \frac{1}{49}\left(\begin{array}{c}21 \cdot 2^{2n+3} - 2^{n+1}(4V_{n+3} + 10V_{n+2} + 5V_{n+1} + 2V_n) \\ +V_nV_{n+1} + V_{n+1}V_{n+2} + V_{n+2}V_{n+3}\end{array}\right) \\
&\quad= \frac{1}{7}\left(3 \cdot 2^{2n+3} - 2^{n+1}(U_n - U_{n+2}) - 1\right),
\end{aligned}$$

where $7U_n = 3V_{n+2} + V_{n+1}$, $V_n + 5V_{n+2} = U_n - U_{n+2}$ and $U_n = H_n(0,1)$.

Using that $\left\|\overrightarrow{JD_n^{(3)}}\right\| = 1$ if and only if $\left\|\overrightarrow{J_n^{(3)}}\right\| = 1$ and $\left\langle \overrightarrow{J_n^{(3)}}, \overrightarrow{J_{n+1}^{(3)}} \right\rangle = 0$ (see Eq. (1.2)) and above calculations, we easily reach the result (3.2). □

Now, if $\overrightarrow{d_1} = \overrightarrow{a_1} + \varepsilon\overrightarrow{b_1}$ and $\overrightarrow{d_2} = \overrightarrow{a_2} + \varepsilon\overrightarrow{b_2}$ are two dual vectors, then the dot product and cross product of them are given respectively by

$$\begin{aligned}
\left\langle \overrightarrow{d_1}, \overrightarrow{d_2} \right\rangle &= \langle \overrightarrow{a_1}, \overrightarrow{a_2} \rangle + \varepsilon\left(\left\langle \overrightarrow{a_1}, \overrightarrow{b_2} \right\rangle + \left\langle \overrightarrow{b_1}, \overrightarrow{a_2} \right\rangle\right), \\
\overrightarrow{d_1} \times \overrightarrow{d_2} &= \overrightarrow{a_1} \times \overrightarrow{a_2} + \varepsilon\left(\overrightarrow{a_1} \times \overrightarrow{b_2} + \overrightarrow{b_1} \times \overrightarrow{a_2}\right).
\end{aligned} \tag{3.3}$$

(For more details, see [9]).

Theorem 3.2. *Let* $\overrightarrow{JD_n^{(3)}}$ *and* $\overrightarrow{JD_m^{(3)}}$ *be two dual third-order Jacobsthal vectors. The dot product of these two vectors is given by*

$$\left\langle \overrightarrow{JD_n^{(3)}}, \overrightarrow{JD_m^{(3)}} \right\rangle = \frac{1}{7}\left(\begin{array}{c}3 \cdot 2^{n+m+2}(1 + 4\varepsilon) - 2^{n+1}(UD_m + 2\varepsilon U_m) \\ -2^{m+1}(UD_n + \varepsilon U_n) + W_{n-m}(1 - \varepsilon)\end{array}\right), \tag{3.4}$$

where $U_n = H_n(0,1)$, $W_n = H_n(2,-1)$ *and* $UD_n = U_n + \varepsilon U_{n+1}$.

Proof. If $\overrightarrow{JD_n^{(3)}} = \overrightarrow{J_n^{(3)}} + \varepsilon\overrightarrow{J_{n+1}^{(3)}}$ and $\overrightarrow{JD_m^{(3)}} = \overrightarrow{J_m^{(3)}} + \varepsilon\overrightarrow{J_{m+1}^{(3)}}$ are two dual vectors, then the dot product of them are given respectively by

$$\begin{aligned}\left\langle \overrightarrow{JD_n^{(3)}}, \overrightarrow{JD_m^{(3)}} \right\rangle &= \left\langle \overrightarrow{J_n^{(3)}}, \overrightarrow{J_m^{(3)}} \right\rangle + \varepsilon\left(\left\langle \overrightarrow{J_n^{(3)}}, \overrightarrow{J_{m+1}^{(3)}} \right\rangle + \left\langle \overrightarrow{J_{n+1}^{(3)}}, \overrightarrow{J_m^{(3)}} \right\rangle \right) \\ &= J_n^{(3)}J_m^{(3)} + J_{n+1}^{(3)}J_{m+1}^{(3)} + J_{n+2}^{(3)}J_{m+2}^{(3)} \\ &\quad + \varepsilon \begin{pmatrix} J_n^{(3)}J_{m+1}^{(3)} + J_{n+1}^{(3)}J_{m+2}^{(3)} + J_{n+2}^{(3)}J_{m+3}^{(3)} \\ +J_{n+1}^{(3)}J_m^{(3)} + J_{n+2}^{(3)}J_{m+1}^{(3)} + J_{n+3}^{(3)}J_{m+2}^{(3)} \end{pmatrix}.\end{aligned}$$

By using the definition of third-order Jacobsthal number (1.13), the equations (2.16) and (3.1), we have

$$\begin{aligned}&J_n^{(3)}J_m^{(3)} + J_{n+1}^{(3)}J_{m+1}^{(3)} + J_{n+2}^{(3)}J_{m+2}^{(3)} \\ &\quad= \frac{1}{49}\begin{pmatrix} \left(2^{n+1} - V_n\right)\left(2^{m+1} - V_m\right) + \left(2^{n+2} - V_{n+1}\right)\left(2^{m+2} - V_{m+1}\right) \\ + \left(2^{n+3} - V_{n+2}\right)\left(2^{m+3} - V_{m+2}\right) \end{pmatrix} \\ &\quad= \frac{1}{49}\begin{pmatrix} 21\cdot 2^{n+m+2} - 2^{n+1}(V_m + 2V_{m+1} + 4V_{m+2}) \\ -2^{m+1}(V_n + 2V_{n+1} + 4V_{n+2}) + V_nV_m + V_{n+1}V_{m+1} + V_{n+2}V_{m+2} \end{pmatrix} \\ &\quad= \frac{1}{7}\left(3\cdot 2^{n+m+2} - 2^{n+1}U_m - 2^{m+1}U_n + W_{n-m}\right),\end{aligned}$$

where $V_{n+1} + 3V_{n+2} = 7U_n$ and $W_n = H_n(2,-1) = \omega_1^n + \omega_2^n$. Then,

$$\begin{aligned}\left\langle \overrightarrow{JD_n^{(3)}}, \overrightarrow{JD_m^{(3)}} \right\rangle &= \frac{1}{7}\left(3\cdot 2^{n+m+2} - 2^{n+1}U_m - 2^{m+1}U_n + W_{n-m}\right) \\ &\quad + \frac{\varepsilon}{7}\left(3\cdot 2^{n+m+4} + 2^{n+1}W_{m+1} + 2^{m+1}W_{n+1} - W_{n-m}\right) \\ &= \frac{1}{7}\begin{pmatrix} 3\cdot 2^{n+m+2}(1+4\varepsilon) - 2^{n+1}(U_m - \varepsilon W_{m+1}) \\ -2^{m+1}(U_n - \varepsilon W_{n+1}) + W_{n-m}(1-\varepsilon) \end{pmatrix},\end{aligned}$$

with $U_{n+1} + 2U_n = -W_{n+1}$, $W_n + W_{n+2} = -W_{n+1}$ and $UD_n = U_n + \varepsilon U_{n+1}$, we easily reach the result (3.4). □

Theorem 3.3. *For* $n, m \geq 0$. *Let* $\overrightarrow{JD_n^{(3)}}$ *and* $\overrightarrow{JD_m^{(3)}}$ *be two dual third-order Jacobsthal vectors. The cross product of* $\overrightarrow{JD_n^{(3)}}$ *and* $\overrightarrow{JD_m^{(3)}}$ *is given by*

$$\overrightarrow{JD_n^{(3)}} \times \overrightarrow{JD_m^{(3)}} = \frac{1}{7}\begin{pmatrix} 2^{n+1}(ZD_{m+1} + 2\varepsilon Z_{m+1}) - 2^{m+1}(ZD_{n+1} + 2\varepsilon Z_{n+1}) \\ +U_{n-m}(1-\varepsilon)(\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k}) \end{pmatrix},$$

where $Z_n = 2U_{n+1}\boldsymbol{i} + W_{n+1}\boldsymbol{j} + U_n\boldsymbol{k}$, $U_n = H_n(0,1)$, $W_n = H_n(2,-1)$, $\boldsymbol{i} = (1,0,0)$, $\boldsymbol{j} = (0,1,0)$, $\boldsymbol{k} = (0,0,1)$ *and* $ZD_n = Z_n + \varepsilon Z_{n+1}$.

Proof. From the equations (3.1) and (3.3), we get

$$\overrightarrow{JD_n^{(3)}} \times \overrightarrow{JD_m^{(3)}} = \overrightarrow{J_n^{(3)}} \times \overrightarrow{J_m^{(3)}} + \varepsilon \left(\overrightarrow{J_n^{(3)}} \times \overrightarrow{J_{m+1}^{(3)}} + \overrightarrow{J_{n+1}^{(3)}} \times \overrightarrow{J_m^{(3)}} \right). \tag{3.5}$$

First, let us compute $\overrightarrow{J_n^{(3)}} \times \overrightarrow{J_m^{(3)}}$, if we use the properties of determinant to calculate the cross product of two vectors, the equality

$$J_n^{(3)} J_{m+1}^{(3)} - J_{n+1}^{(3)} J_m^{(3)} = \frac{1}{7} \left(\ 2^{n+1} U_{m+1} - 2^{m+1} U_{n+1} + U_{n-m} \ \right)$$

(see (2.17)), $U_n = H_n(0,1)$ and simplify the statements, we find that

$$\begin{aligned} \overrightarrow{J_n^{(3)}} \times \overrightarrow{J_m^{(3)}} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ J_n^{(3)} & J_{n+1}^{(3)} & J_{n+2}^{(3)} \\ J_m^{(3)} & J_{m+1}^{(3)} & J_{m+2}^{(3)} \end{vmatrix} \\ &= \mathbf{i} \begin{vmatrix} J_{n+1}^{(3)} & J_{n+2}^{(3)} \\ J_{m+1}^{(3)} & J_{m+2}^{(3)} \end{vmatrix} - \mathbf{j} \begin{vmatrix} J_n^{(3)} & J_{n+2}^{(3)} \\ J_m^{(3)} & J_{m+2}^{(3)} \end{vmatrix} + \mathbf{k} \begin{vmatrix} J_n^{(3)} & J_{n+1}^{(3)} \\ J_m^{(3)} & J_{m+1}^{(3)} \end{vmatrix} \\ &= \frac{1}{7} \begin{pmatrix} \mathbf{i}(2^{n+2} U_{m+2} - 2^{m+2} U_{n+2} + U_{n-m}) \\ -\mathbf{j}(-2^{n+1} W_{m+2} + 2^{m+1} W_{n+2} - U_{n-m}) \\ +\mathbf{k}(2^{n+1} U_{m+1} - 2^{m+1} U_{n+1} + U_{n-m}) \end{pmatrix} \\ &= \frac{1}{7} \left(\ 2^{n+1} Z_{m+1} - 2^{m+1} Z_{n+1} + U_{n-m}(\mathbf{i} + \mathbf{j} + \mathbf{k}) \ \right) \end{aligned} \tag{3.6}$$

where $Z_n = 2U_{n+1}\mathbf{i} + W_{n+1}\mathbf{j} + U_n\mathbf{k}$, $U_n = H_n(0,1)$, $W_n = H_n(2,-1)$, $\mathbf{i} = (1,0,0)$, $\mathbf{j} = (0,1,0)$ and $\mathbf{k} = (0,0,1)$. Putting the equation (3.6) in (3.5), and using the definition of third-order Jacobsthal numbers, we obtain the result as

$$\begin{aligned} \overrightarrow{JD_n^{(3)}} \times \overrightarrow{JD_m^{(3)}} &= \frac{1}{7} \left(\ 2^{n+1} Z_{m+1} - 2^{m+1} Z_{n+1} + U_{n-m}(\mathbf{i} + \mathbf{j} + \mathbf{k}) \ \right) \\ &\quad + \frac{\varepsilon}{7} \begin{pmatrix} 2^{n+1} Z_{m+2} - 2^{m+2} Z_{n+1} + U_{n-m-1}(\mathbf{i} + \mathbf{j} + \mathbf{k}) \\ +2^{n+2} Z_{m+1} - 2^{m+1} Z_{n+2} + U_{n-m+1}(\mathbf{i} + \mathbf{j} + \mathbf{k}) \end{pmatrix} \\ &= \frac{1}{7} \begin{pmatrix} 2^{n+1}(ZD_{m+1} + 2\varepsilon Z_{m+1}) - 2^{m+1}(ZD_{n+1} + 2\varepsilon Z_{n+1}) \\ +U_{n-m}(1-\varepsilon)(\mathbf{i} + \mathbf{j} + \mathbf{k}) \end{pmatrix}, \end{aligned}$$

where $ZD_m = Z_m + \varepsilon Z_{m+1}$. □

Acknowledgements. The author also thanks the suggestions sent by the reviewer, which have improved the final version of this article.

References

[1] M. Akyiğit, H. H. Kösal, M. Tosun: *Split Fibonacci Quaternions.* Adv. Appl. Clifford Algebr. 23 (2013), pp. 535–545,
doi: https://doi.org/10.1007/s00006-013-0401-9.

[2] P. Barry: *Triangle Geometry and Jacobsthal Numbers.* Irish Math. Soc. Bulletin 51.1 (2003), pp. 45–57.

[3] K. Carmody: *Circular and Hyperbolic quaternions, octonions and sedenions.* Appl. Math. comput. 28 (1988), pp. 47–72.

[4] G. Cerda-Morales: *Dual third-order Jacobsthal quaternions.* Proyecciones Journal of Mathematics 37.4 (2018), pp. 731–747.

[5] G. Cerda-Morales: *Identities for Third Order Jacobsthal Quaternions.* Adv. Appl. Clifford Algebr. 27.2 (2017), pp. 1043–1053,
doi: https://doi.org/10.1007/s00006-016-0654-1.

[6] G. Cerda-Morales: *On the third-order Jacobsthal and third-order Jacobsthal-Lucas sequences and their matrix representations.* Mediterr. J. Math. 16.2 (2019), pp. 1–12,
doi: https://doi.org/10.1007/s00009-019-1319-9.

[7] C. K. Cook, M. R. Bacon: *Some identities for Jacobsthal and Jacobsthal-Lucas numbers satisfying higher order recurrence relations.* Ann. Math. Inform 41.1 (2013), pp. 27–39.

[8] M. Gogberashvili: *Octonionic Geometry*, Adv. Appl. Clifford Algebr. 15 (2005), pp. 55–66,
doi: https://doi.org/10.1007/s00006-005-0003-2.

[9] İ. A. Güven, S. K. Nurkan: *A New Approach To Fibonacci, Lucas Numbers and Dual Vectors*, Adv. Appl. Clifford Algebr. 25 (2015), pp. 577–590,
doi: https://doi.org/10.1007/s00006-014-0516-7.

[10] S. Halici: *On Fibonacci quaternions*, Adv. Appl. Clifford Algebr. 22 (2012), pp. 321–327,
doi: https://doi.org/10.1007/s00006-011-0317-1.

[11] A. F. Horadam: *Complex Fibonacci numbers and Fibonacci quaternions.* Am. Math. Month. 70.1 (1963), pp. 289–291.

[12] A. F. Horadam: *Jacobsthal representation numbers.* Fibonacci Quart. 34 (1996), pp. 40–54.

[13] A. F. Horadam: *Quaternion recurrence relations.* Ulam Quarterly 2 (1993), pp. 23–33.

[14] M. Iyer: *A Note On Fibonacci Quaternions.* Fibonacci Quart. 3.1 (1969), pp. 225–229.

[15] J. Köplinger: *Hypernumbers and relativity*, Appl. Math. Computation 188 (2007), pp. 954–969,
doi: https://doi.org/10.1016/j.amc.2006.10.051.

[16] M. E. Waddill, L. Sacks: *Another generalized Fibonacci sequence*, Fibonacci Quart. 5 (1967), pp. 209–222.

Annales Mathematicae et Informaticae
***52** (2020) pp. 71–84*
DOI: https://doi.org/10.33039/ami.2020.02.002
URL: https://ami.uni-eszterhazy.hu

Binary quadratic forms and sums of powers of integers

José Luis Cereceda

Collado Villalba 28400 – Madrid, Spain
jl.cereceda@movistar.es

Submitted: May 16, 2019
Accepted: February 8, 2020
Published online: February 15, 2020

Abstract

In this methodological paper, we first review the classic cubic Diophantine equation $a^3 + b^3 + c^3 = d^3$, and consider the specific class of solutions $q_1^3 + q_2^3 + q_3^3 = q_4^3$ with each q_i being a binary quadratic form. Next we turn our attention to the familiar sums of powers of the first n positive integers, $S_k = 1^k + 2^k + \cdots + n^k$, and express the squares S_k^2, S_m^2, and the product $S_k S_m$ as a linear combination of power sums. These expressions, along with the above quadratic-form solution for the cubic equation, allows one to generate an infinite number of relations of the form $Q_1^3 + Q_2^3 + Q_3^3 = Q_4^3$, with each Q_i being a linear combination of power sums. Also, we briefly consider the quadratic Diophantine equations $a^2 + b^2 + c^2 = d^2$ and $a^2 + b^2 = c^2$, and give a family of corresponding solutions $Q_1^2 + Q_2^2 + Q_3^2 = Q_4^2$ and $Q_1^2 + Q_2^2 = Q_3^2$ in terms of sums of powers of integers.

Keywords: Diophantine equation, binary quadratic form, algebraic identity, sums of powers of integers, product of power sums, Pythagorean quadruples

MSC: 11D25, 11B57

1. Introduction

Our starting point is the cubic Diophantine equation

$$a^3 + b^3 + c^3 = d^3, \qquad abcd \neq 0. \tag{1.1}$$

(Note that $abcd \neq 0$ as, by Fermat's Last Theorem, we cannot have $a^3 + b^3 = c^3$.) As pointed out by Dickson in his comprehensive *History of the Theory of Numbers*, the problem of finding the rational or integer (positive or negative) solutions to Equation (1.1) can be traced back to Diophantus [11, p. 550]. A first parametric solution was given by Vieta in 1591 [11, p. 551] and, in 1754, Euler found the most general family of rational solutions to (1.1) (see [11, p. 552] and [9]). Much more recently, Choudhry [8] obtained a *complete* solution of (1.1) in positive integers.

It is to be noted that there are several different formulations equivalent to the general solution discovered by Euler. In his third notebook, Ramanujan provided a family of solutions equivalent to Euler's general solution that appears to be the simplest of all [4, 7]. In addition to this general solution, Ramanujan also gave some further families of parametric solutions to (1.1) as well as several numerical examples. Specifically, in a problem submitted to the *Journal of the Indian Mathematical Society*, (Question 441, JIMS 5, p. 39, 1913), Ramanujan put forward the following two-parameter solution to Equation (1.1) [5]:

$$(3u^2+5uv-5v^2)^3+(4u^2-4uv+6v^2)^3+(5u^2-5uv-3v^2)^3 = (6u^2-4uv+4v^2)^3. \quad (1.2)$$

Relation (1.2) constitutes an algebraic identity and, as such, is satisfied by any real or complex values of the parameters u and v. As we are dealing with Diophantine equations, however, it will be assumed that u and v take only rational or integer values. In particular, putting $u = 1$ and $v = 0$ in (1.2) gives us the smallest positive solution to (1.1), namely $3^3 + 4^3 + 5^3 = 6^3$.

In this methodological paper, we search for solutions of the kind shown in Equation (1.2), that is, solutions $q_1^3 + q_2^3 + q_3^3 = q_4^3$ for which each of the q_i's ($i = 1, 2, 3, 4$) adopts the form of a quadratic polynomial of two variables, say u and v (a binary quadratic form): $q_i = \alpha_i u^2 + \beta_i uv + \gamma_i v^2$, where α_i, β_i, and γ_i take integer (positive or negative) values. Our interest in this type of solutions stems from the fact that, as explained in [12], by using the above identity $3^3+4^3+5^3 = 6^3$ as a seed, one can generate quadratic-form formulas to Equation (1.1). Expanding on this point, and borrowing a theorem of Sándor [22, Theorem 1], in Section 2 we show that, indeed, it is possible to construct quadratic-form representations for the cubic equation (1.1) starting from *any* particular nontrivial solution to (1.1) (see below for the definition of a trivial solution). The proof of this result given by Sándor (which is essentially reproduced in Section 2) is particularly suitable for our purpose since it utilizes only precalculus tools. Furthermore, Sándor's theorem allows one to readily produce a wealth of algebraic identities like that in Equation (1.2) by simply adding and multiplying the integers a, b, c, and d constituting a particular (nontrivial) solution of (1.1).

In Section 3, we consider the familiar sums of powers of the first n positive integers, $S_k = 1^k + 2^k + \cdots + n^k$ (with k being a nonnegative integer), and express the squares S_k^2, S_m^2, and the product $S_k S_m$ as a linear combination of power sums. In this way, using such expressions for S_k^2, S_m^2, and $S_k S_m$, the following generic quadratic form

$$Q_i(k, m, n) = \alpha_i S_k^2 + \beta_i S_k S_m + \gamma_i S_m^2, \quad (1.3)$$

can be equally expressed as a linear combination of power sums. (Note that $Q_i(k,m,n)$ depends explicitly on n through the power sums S_k and S_m.) Therefore, using the quadratic-form solutions obtained in Section 2, one can construct relationships of the type $Q_1(k,m,n)^3 + Q_2(k,m,n)^3 + Q_3(k,m,n)^3 = Q_4(k,m,n)^3$, with each $Q_i(k,m,n)$ being a linear combination of power sums. Moreover, substituting each of the power sums in $Q_i(k,m,n)$ for its polynomial representation yields (for fixed k and m) algebraic identities of the form $Q_1(u)^3+Q_2(u)^3+Q_3(u)^3 = Q_4(u)^3$, where each $Q_i(u)$ is itself a polynomial in the real or complex variable u (see, for instance, Equation (3.8) below).

Finally, in Section 4 we briefly consider the quadratic Diophantine equation $a^2 + b^2 + c^2 = d^2$. Using a particularly simple quadratic-form solution for this equation, we give a corresponding solution in terms of S_k and S_k^2 (see Equation (4.4) below). On the other hand, starting from an almost trivial identity, we give a family of Pythagorean triangles whose side lengths are given by $|S_k^2 - S_m^2|$, $2S_kS_m$, and $S_k^2 + S_m^2$. As a by-product, we also obtain a family of solutions for the Diophantine equation $a^2 + b^2 = c^2 + d^2$.

From a pedagogical point of view, this methodological paper could be of interest to both high school and college students for the following reasons. On the one hand, it shows in an elementary way how to obtain systematically quadratic-form solutions for the cubic equation (1.1). In this regard, as we shall see, Sándor's theorem proves to be extremely useful to this end since it provides a fairly elementary yet powerful method to generate quadratic-form formulas $q_1^3 + q_2^3 + q_3^3 = q_4^3$ for Equation (1.1). On the other hand, we introduce some well-known formulas (though rarely found in the current literature) involving sums of powers of integers, in particular that expressing the product S_kS_m as a linear combination of S_j's. Using these formulas, and with the aid of a computer algebra system, students ought reliably compute the quadratic form in Equation (1.3) for a variety of values of the parameters. Last, but not least, equipped with the given formulas for S_kS_m, S_k^2, S_1^k, and $S_2S_1^k$, students might want to explore other low degree Diophantine equations (see, in this respect, [3, Chapter 2]) and recast some of their solutions in terms of sums of powers of integers.

2. Quadratic solutions for the cubic equation

As was anticipated in the introduction, we shall make use of a theorem of Sándor (see [22, Theorem 1]) in order to construct two-parameter quadratic solutions for the cubic equation (1.1). Following Sándor, we say that a solution of (1.1) is trivial if $d = a$ or $d = b$ or $d = c$. The said theorem, adapted to our notation, is as follows.

Theorem 2.1. *If* (a,b,c,d) *is a nontrivial integer solution of* (1.1) *then for any*

integer values of u and v

$$\begin{aligned} q_1 &= a(a+c)u^2 + (d-b)(d+b)uv - c(d-b)v^2, \\ q_2 &= b(a+c)u^2 - (c-a)(c+a)uv + d(d-b)v^2, \\ q_3 &= c(a+c)u^2 - (d-b)(d+b)uv - a(d-b)v^2, \\ q_4 &= d(a+c)u^2 - (c-a)(c+a)uv + b(d-b)v^2, \end{aligned} \tag{2.1}$$

satisfy $q_1^3 + q_2^3 + q_3^3 = q_4^3$.

Proof. As noted by Sándor, relations (2.1) can be obtained by generalizing Ramanujan's quadratic solution (1.2) but, following [22], we will give a simpler proof of Theorem 2.1 employing a technique devised by Nicholson [18]. Thus, let $a^3 + b^3 + c^3 = d^3$ be a nontrivial solution of (1.1), and consider Nicholson's parametric equation [22]

$$\big(ux - cy\big)^3 + \big(-ux - ay\big)^3 + \big(vx - by\big)^3 = \big(vx - dy\big)^3. \tag{2.2}$$

Expanding in (2.2) and using the constraint $a^3+b^3+c^3 = d^3$, the cubic terms vanish and we are left with an equation involving only quadratic and linear exponents, namely

$$\begin{aligned} -3u^2x^2cy + 3uxc^2y^2 - 3u^2x^2ay - 3uxa^2y^2 - 3v^2x^2by \\ + 3vxb^2y^2 = -3v^2x^2dy + 3vxd^2y^2. \end{aligned}$$

Dividing throughout by the common factor $3xy$ gives

$$x\big(dv^2 - bv^2 - cu^2 - au^2\big) = y\big(d^2v - b^2v + a^2u - c^2u\big).$$

Clearly, the values $x = d^2v - b^2v + a^2u - c^2u$ and $y = dv^2 - bv^2 - cu^2 - au^2$ satisfy the equation, and then we obtain

$$\begin{aligned} ux - cy &= u\big(d^2v - b^2v + a^2u - c^2u\big) - c\big(dv^2 - bv^2 - cu^2 - au^2\big) \\ &= a(a+c)u^2 + (d-b)(d+b)uv - c(d-b)v^2, \\ -ux - ay &= -u\big(d^2v - b^2v + a^2u - c^2u\big) - a\big(dv^2 - bv^2 - cu^2 - au^2\big) \\ &= b(a+c)u^2 - (c-a)(c+a)uv + d(d-b)v^2, \\ vx - by &= v\big(d^2v - b^2v + a^2u - c^2u\big) - b\big(dv^2 - bv^2 - cu^2 - au^2\big) \\ &= c(a+c)u^2 - (d-b)(d+b)uv - a(d-b)v^2, \\ vx - dy &= v\big(d^2v - b^2v + a^2u - c^2u\big) - d\big(dv^2 - bv^2 - cu^2 - au^2\big) \\ &= d(a+c)u^2 - (c-a)(c+a)uv + b(d-b)v^2. \end{aligned}$$

Nicholson's parametric equation (2.2) then guarantees that $q_1^3 + q_2^3 + q_3^3 = q_4^3$, with the q_i's being given by Equations (2.1). □

Let us consider a few examples illustrating the application of Theorem 2.1. In each case, starting from a given (nontrivial) integer solution (a, b, c, d) to (1.1), it generates a two-parameter family of solutions (q_1, q_2, q_3, q_4) satisfying (1.1):

1. Substituting $(a, b, c, d) = (3, 4, 5, 6)$ into Equations (2.1), dividing by 2, and using the linear transformation $u \to u/2$, we get Ramanujan's solution (1.2) [22].

2. Using $(a, b, c, d) = (1, 6, 8, 9)$ into Equations (2.1) and dividing by 3 yields

$$(3u^2 + 15uv - 8v^2)^3 + (18u^2 - 21uv + 9v^2)^3 + (24u^2 - 15uv - v^2)^3 = (27u^2 - 21uv + 6v^2)^3. \quad (2.3)$$

   Now, putting $u = 1$ and $v = 2$ in (2.3) gives $1^3 + 12^3 + (-10)^3 = 9^3$ or, equivalently,

$$1^3 + 12^3 = 9^3 + 10^3 = 1729. \quad (2.4)$$

   Readers will recognize 1729 as the famous Hardy-Ramanujan number, having the distinctive property that it is the smallest positive integer that can be written as the sum of two positive cubes in more than one way [16, 23, 24]. On the other hand, for $u = 6$ and $v = -1$, we obtain

$$10^3 + 783^3 + 953^3 = 1104^3. \quad (2.5)$$

3. Likewise, using $(a, b, c, d) = (7, 14, 17, 20)$ into Equations (2.1) and dividing by 6 yields

$$(28u^2 + 34uv - 17v^2)^3 + (56u^2 - 40uv + 20v^2)^3 + (68u^2 - 34uv - 7v^2)^3 = (80u^2 - 40uv + 14v^2)^3. \quad (2.6)$$

   Putting $u = -2$ and $v = -3$ in (2.6) we find

$$163^3 + 164^3 + 5^3 = 206^3. \quad (2.7)$$

   And, for $u = 10$ and $v = 3$, we obtain

$$3667^3 + 4580^3 + 5717^3 = 6926^3. \quad (2.8)$$

Rather interestingly, it can be shown (see [22, Theorem 2]) that if (a, b, c, d) is a nontrivial integer solution to (1.1) and (q_1, q_2, q_3, q_4) is a nontrivial integer solution obtained via Equations (2.1), then necessarily

$$\frac{a + c}{d - b} = \frac{q_1 + q_3}{q_4 - q_2}. \quad (2.9)$$

Note that the denominators in Equation (2.9) are well defined since both (a, b, c, d) and (q_1, q_2, q_3, q_4) are nontrivial solutions. Note further that, for given a, b, c, and d,

relation (2.9) holds irrespective of the (integer) values taken by u and v in Equations (2.1). Conversely (see [22, Theorem 3]), if (a,b,c,d) and (q_1,q_2,q_3,q_4) are two integer nontrivial solutions to (1.1) and $\frac{a+c}{d-b} = \frac{q_1+q_3}{q_4-q_2}$, then there exist integers u and v such that substituting these into Equations (2.1) yields (q_1,q_2,q_3,q_4) or a multiple of it. Thus, the solutions (q_1,q_2,q_3,q_4) obtained from a given solution (a,b,c,d) via Equations (2.1) can be essentially characterized through the condition stated in Equation (2.9).

We can readily check that the said condition is indeed satisfied by the examples given above. So, for the example given in (2.3), the condition (2.9) reads as

$$\frac{1+8}{9-6} = \frac{3u^2+15uv-8v^2+24u^2-15uv-v^2}{27u^2-21uv+6v^2-18u^2+21uv-9v^2} = 3,$$

which is fulfilled for all integer values of u and v (discarding the trivial solution $u=v=0$). In particular, it holds for the numerical examples (2.4) and (2.5), as $\frac{1-10}{9-12} = \frac{10+953}{1104-783} = 3$. Similarly, regarding the example given in (2.6), the condition (2.9) reads as

$$\frac{7+17}{20-14} = \frac{28u^2+34uv-17v^2+68u^2-34uv-7v^2}{80u^2-40uv+14v^2-56u^2+40uv-20v^2} = 4,$$

which is equally fulfilled for any choice of integers u and v (excluding the case $u=v=0$). In particular, it holds for the numerical examples (2.7) and (2.8), as $\frac{163+5}{206-164} = \frac{3667+5717}{6926-4580} = 4$.

We conclude this section by noting that, naturally, given an initial solution (a,b,c,d) to (1.1), we can use as well either of its permutations (a,c,b,d), (b,a,c,d), (b,c,a,d), (c,a,b,d), or (c,b,a,d) as inputs to Equations (2.1). For instance, instead of the initial solution $(1,6,8,9)$ used above, we can plug $(1,8,6,9)$ into Equations (2.1) to obtain

$$(7u^2+17uv-6v^2)^3 + (56u^2-35uv+9v^2)^3 + (42u^2-17uv-v^2)^3 = (63u^2-35uv+8v^2)^3. \quad (2.10)$$

Incidentally, we observe that setting $u=-1$ and $v=-3$ in (2.10) yields (after dividing by 2): $2^3+16^3 = 9^3+15^3 = 4104$, which is the next Hardy-Ramanujan number after 1729. We encourage the students to search for their own solutions to the cubic equation (1.1) by means of Theorem 2.1, and to verify that they comply with relation (2.9).

3. Quadratic forms of sums of powers of integers

Consider now the power sums $S_k = 1^k+2^k+\cdots+n^k$ and $S_m = 1^m+2^m+\cdots+n^m$. Their product is given by

$$S_k S_m = \frac{1}{k+1}\sum_{j=0}^{k/2} B_{2j}\binom{k+1}{2j} S_{k+m+1-2j}$$

$$+\frac{1}{m+1}\sum_{j=0}^{m/2} B_{2j}\binom{m+1}{2j} S_{k+m+1-2j}, \tag{3.1}$$

where $B_0 = 1$, $B_1 = -1/2$, $B_2 = 1/6$, $B_3 = 0$, $B_4 = -1/30$, etc., are the Bernoulli numbers (which fulfill the property that $B_{2j+1} = 0$ for all $j \geq 1$) [2, 10]; $\binom{k}{m}$ are the familiar binomial coefficients; and where the upper summation limit $k/2$ denotes the greatest integer lesser than or equal to $k/2$. Formula (3.1) is not commonly encountered across the abound literature on sums of powers of integers. A notable exception being the paper [15], where formula (3.1) is stated as a theorem. As noted in [15], formula (3.1) was known to Lucas by 1891. For the case that $k = m$, formula (3.1) reduces to

$$S_k^2 = \frac{2}{k+1}\sum_{j=0}^{k/2} B_{2j}\binom{k+1}{2j} S_{2k+1-2j}. \tag{3.2}$$

For later reference, we also quote the formula for the k-th power of S_1 expressed as a linear combination of power sums

$$S_1^k = \frac{1}{2^{k-1}}\sum_{j=0}^{\frac{k-1}{2}} \binom{k}{2j+1} S_{2k-1-2j}, \tag{3.3}$$

as well as the formula for the product

$$S_2 S_1^k = \frac{1}{3\cdot 2^k}\sum_{j=0}^{\frac{k+1}{2}} \frac{2k+3-2j}{2j+1}\binom{k+1}{2j} S_{2k+2-2j}. \tag{3.4}$$

Note that the right-hand side of Equations (3.2) and (3.3) involves only power sums S_j with j odd, whereas that of Equation (3.4) involves only power sums S_j with j even. Formula (3.3) (written in a slightly different form) appears as a theorem in [15], where it is further noted that it was known as far back as 1877 (Lampe) and 1878 (Stern). Regarding formula (3.4), it looks somewhat more exotic, although it is by no means new. An equivalent formulation of both Equations (3.3) and (3.4) can be found in, respectively, formulas (17) and (22) of the review paper by Kotiah [14]. It is worth pointing out, on the other hand, that the right-hand side of Equation (3.3) [(3.4)] can be interpreted as a sort of average of sums of powers of integers as the total number $\sum_{j=0}^{\frac{k-1}{2}} \binom{k}{2j+1}$ $[\sum_{j=0}^{\frac{k+1}{2}} \frac{2k+3-2j}{2j+1}\binom{k+1}{2j}]$ of power sums appearing on the right-hand side of (3.3) [(3.4)] is just 2^{k-1} $[3\cdot 2^k]$ (see [6, 21]).

Provided with Equations (3.1) and (3.2), we can thus write the quadratic form (1.3) as the following linear combination of power sums:

$$Q_i(k,m,n) = \frac{2\alpha_i}{k+1}\sum_{j=0}^{k/2} B_{2j}\binom{k+1}{2j} S_{2k+1-2j} + \frac{\beta_i}{k+1}\sum_{j=0}^{k/2} B_{2j}\binom{k+1}{2j} S_{k+m+1-2j}$$

$$+\frac{\beta_i}{m+1}\sum_{j=0}^{m/2}B_{2j}\binom{m+1}{2j}S_{k+m+1-2j}+\frac{2\gamma_i}{m+1}\sum_{j=0}^{m/2}B_{2j}\binom{m+1}{2j}S_{2m+1-2j}. \quad (3.5)$$

Regarding the coefficients α_i, β_i, and γ_i, we must choose them so that the quadratic forms $q_i=\alpha_i u^2+\beta_i uv+\gamma_i v^2$ satisfy the relation $q_1^3+q_2^3+q_3^3=q_4^3$ for any integer values of u and v. This in turn ensures that the quadratic forms $Q_i(k,m,n)$ in Equation (3.5) will satisfy $Q_1(k,m,n)^3+Q_2(k,m,n)^3+Q_3(k,m,n)^3=Q_4(k,m,n)^3$ as well.

At this point, it is obviously most useful to run a computer algebra system such as *Mathematica* to quickly compute the quadratic form $Q_i(k,m,n)$ for concrete values of α_i, β_i, γ_i, k, m, and n. As a simple but illustrative example, let us first take $k=1$ and $m=2$ to obtain

$$Q_i(1,2,n)=\frac{1}{6}\big(\beta_i S_2+\big(6\alpha_i+2\gamma_i\big)S_3+5\beta_i S_4+4\gamma_i S_5\big).$$

Then, choosing for example the coefficients α_i, β_i, and γ_i appearing in Equation (2.3) (namely, $\alpha_1=3$, $\beta_1=15$, $\gamma_1=-8$, $\alpha_2=18$, $\beta_2=-21$, $\gamma_2=9$, $\alpha_3=24$, $\beta_3=-15$, $\gamma_3=-1$, $\alpha_4=27$, $\beta_4=-21$, and $\gamma_4=6$), we get (after removing the common factor $1/6$) the following relationship among the power sums S_2, S_3, S_4, and S_5:

$$\begin{aligned}\big(15S_2+2S_3+75S_4-32S_5\big)^3+\big(-21S_2+126S_3-105S_4+36S_5\big)^3\\ +\big(-15S_2+142S_3-75S_4-4S_5\big)^3\\ =\big(-21S_2+174S_3-105S_4+24S_5\big)^3. \quad (3.6)\end{aligned}$$

Equation (3.6) can in turn be written explicitly as a function of the variable n by expressing each of the involved power sums in terms of n. It is a well-known result that S_k can be expressed as a polynomial in n of degree $k+1$ with zero constant term according to the formula (see, for instance, [14, 26–28]):

$$S_k=\frac{1}{k+1}\sum_{j=1}^{k+1}\binom{k+1}{j}(-1)^{k+1-j}B_{k+1-j}n^j, \qquad k\geq 0. \quad (3.7)$$

$$\begin{aligned}S_2&=\tfrac{1}{3}n^3+\tfrac{1}{2}n^2+\tfrac{1}{6}n\\ S_3&=\tfrac{1}{4}n^4+\tfrac{1}{2}n^3+\tfrac{1}{4}n^2\\ S_4&=\tfrac{1}{5}n^5+\tfrac{1}{2}n^4+\tfrac{1}{3}n^3-\tfrac{1}{30}n\\ S_5&=\tfrac{1}{6}n^6+\tfrac{1}{2}n^5+\tfrac{5}{12}n^4-\tfrac{1}{12}n^2\\ S_6&=\tfrac{1}{7}n^7+\tfrac{1}{2}n^6+\tfrac{1}{2}n^5-\tfrac{1}{6}n^3+\tfrac{1}{42}n\\ S_7&=\tfrac{1}{8}n^8+\tfrac{1}{2}n^7+\tfrac{7}{12}n^6-\tfrac{7}{24}n^4+\tfrac{1}{12}n^2\end{aligned}$$

Table 1: The power sums $S_2, S_3, \ldots, S_7$ expressed as polynomials in n

This formula, which was first established by Jacob Bernoulli in his masterpiece *Ars Conjectandi* (published posthumously in 1713 [1]), provides an efficient way to compute the power sums S_k. Table 1 shows the polynomials for $S_2, S_3, \ldots, S_7$ as obtained from Bernoulli's formula (3.7).[1] Thus, substituting the power sums S_2, S_3, S_4, and S_5 in Equation (3.6) by the corresponding polynomial in Table 1, and renaming the variable n as a generic variable u, we get (after multiplying by an overall factor of 3) the algebraic identity

$$\begin{aligned}\big(32u^2+93u^3+74u^4-3u^5-16u^6\big)^3+\big(54u^2+63u^3-18u^4-9u^5+18u^6\big)^3\\ +\big(85u^2+123u^3-11u^4-51u^5-2u^6\big)^3\\ =\big(93u^2+135u^3+3u^4-27u^5+12u^6\big)^3,\end{aligned}\tag{3.8}$$

which is true for all real or complex values of u. For example, for $u=1$, relation (3.8) gives us (after dividing by 36): $5^3+3^3+4^3=6^3$.

On the other hand, taking $u=S_2$ and $v=S_1^k$ in the quadratic form $q_i=\alpha_i u^2+\beta_i uv+\gamma_i v^2$ yields

$$F_i(k,n)=\alpha_i S_2^2+\beta_i S_2 S_1^k+\gamma_i S_1^{2k}.$$

Utilizing Equations (3.3) and (3.4), and noting that $S_2^2=\frac{1}{3}S_3+\frac{2}{3}S_5$, we can then write $F_i(k,n)$ as the linear combination of power sums

$$\begin{aligned}F_i(k,n)=\frac{\alpha_i}{3}S_3+\frac{2\alpha_i}{3}S_5+\frac{\beta_i}{3\cdot 2^k}\sum_{j=0}^{\frac{k+1}{2}}\frac{2k+3-2j}{2j+1}\binom{k+1}{2j}S_{2k+2-2j}\\ +\frac{\gamma_i}{2^{2k-1}}\sum_{j=0}^{\frac{2k-1}{2}}\binom{2k}{2j+1}S_{4k-1-2j}.\end{aligned}\tag{3.9}$$

As before, in order to derive relations of the type $F_1(k,n)^3+F_2(k,n)^3+F_3(k,n)^3=F_4(k,n)^3$, we must choose the coefficients α_i, β_i, and γ_i such that the quadratic forms $q_i=\alpha_i u^2+\beta_i uv+\gamma_i v^2$ satisfy $q_1^3+q_2^3+q_3^3=q_4^3$ for any integer values of u and v. As a concrete example, let us first take $k=2$ in Equation (3.9) to get

$$F_i(2,n)=\frac{1}{12}\big(4\alpha_i S_3+5\beta_i S_4+(8\alpha_i+6\gamma_i)S_5+7\beta_i S_6+6\gamma_i S_7\big).$$

Then, using the coefficients α_i, β_i, and γ_i appearing in Equation (2.10) (namely, $\alpha_1=7$, $\beta_1=17$, $\gamma_1=-6$, $\alpha_2=56$, $\beta_2=-35$, $\gamma_2=9$, $\alpha_3=42$, $\beta_3=-17$, $\gamma_3=-1$, $\alpha_4=63$, $\beta_4=-35$, and $\gamma_4=8$), we obtain (omitting the common factor

[1] Equation (3.7) is often referred to in the literature as Faulhaber's formula after the German engineer and mathematician Johann Faulhaber (1580-1635). In our view, however, it is more accurate to name Equation (3.7) as Bernoulli's formula or Bernoulli's identity.

1/12) the following relationship among the power sums S_3, S_4, S_5, S_6, and S_7:

$$\begin{aligned}&\big(28S_3+85S_4+20S_5+119S_6-36S_7\big)^3\\&\quad+\big(224S_3-175S_4+502S_5-245S_6+54S_7\big)^3\\&\quad+\big(168S_3-85S_4+330S_5-119S_6-6S_7\big)^3\\&\qquad=\big(252S_3-175S_4+552S_5-245S_6+48S_7\big)^3.\end{aligned}\tag{3.10}$$

Likewise, replacing each of the power sums in (3.10) by its respective polynomial in Table 1 yields (after multiplying by an overall factor of 12) the algebraic identity

$$\begin{aligned}&\big(28u^2+270u^3+820u^4+1038u^5+502u^6-12u^7-54u^8\big)^3\\&\quad+\big(224u^2+1134u^3+1943u^4+1122u^5-88u^6-96u^7+81u^8\big)^3\\&\quad+\big(168u^2+906u^3+1665u^4+1062u^5-96u^6-240u^7-9u^8\big)^3\\&\quad=\big(252u^2+1302u^3+2298u^4+1422u^5-30u^6-132u^7+72u^8\big)^3,\end{aligned}\tag{3.11}$$

which has been written in terms of the generic (complex or real) variable u. Moreover, it is to be noted that each of the four summands in Equation (3.11) can be factorized as $u^2(u+1)^2$ times a polynomial in u of degree 4, so that the identity in (3.11) can be neatly simplified to

$$\begin{aligned}&\big(28+214u+364u^2+96u^3-54u^4\big)^3+\big(224+686u+347u^2-258u^3+81u^4\big)^3\\&+\big(168+570u+357u^2-222u^3-9u^4\big)^3=\big(252+798u+450u^2-276u^3+72u^4\big)^3.\end{aligned}$$

In particular, for $u=0$, we obtain $28^3+224^3+168^3=252^3$ or, after dividing each term by 28, $1^3+8^3+6^3=9^3$.

Trivially, for $u=0$, all four summands in either of relations (3.8) or (3.11) vanish. Less obvious is the fact that the same happens for $u=-1$. To see why, we need to extend the domain of definition of $S_k=1^k+2^k+\cdots+n^k$ to negative values of n. As explained in [13], this can be achieved simply by subtracting successively the k-th power of 0, -1, -2, etc. In this way, it is not difficult to show (see Table 1 of [13]) that, for all $k\geq 1$, the polynomial S_k is symmetric about the point $-\frac{1}{2}$ (see also [17, Theorem 10] for a rigorous proof of this assertion). Thus, as $S_k=0$ for $n=0$, this means that S_k equally vanishes for $n=-1$.[2] (It is readily verified that the polynomials in Table 1 indeed satisfy $S_j(-1)=0$ for each $j=2,3,\ldots,7$.[3]) As a consequence, the quadratic forms $Q_i(k,m,n)$ and $F_i(k,n)$ (defined in Equation

[2]It is left as an exercise to the reader to show the following basic recurrence formula for the Bernoulli numbers

$$B_k=-\frac{1}{k+1}\sum_{j=0}^{k-1}\binom{k+1}{j}B_j,\quad\text{for all } k\geq 1,$$

employing Bernoulli's formula (3.7), and using that $S_k(-1)=0$ for all $k\geq 1$.

[3]That $S_k(-1)=0$ also follows directly from the well-known fact that $S_1=\frac{1}{2}n(n+1)$ is a factor of S_k for all $k\geq 1$.

(3.5) and (3.9), respectively) are zero for $n = -1$, regardless of the values that α_i, β_i, γ_i, k, and m may take (provided that $k, m \geq 1$).

Again, we encourage the students to construct their own algebraic identities like those in Equations (3.8) and (3.11) by making use of the quadratic forms (3.5) and (3.9), and Bernoulli's formula (3.7).

4. Concluding remarks

In what follows, we briefly consider the Diophantine quadratic equation

$$a^2 + b^2 + c^2 = d^2. \tag{4.1}$$

Quadruples of positive integers (a, b, c, d) such as $(2, 3, 6, 7)$ satisfying (4.1) are called Pythagorean *quadruples*, in analogy with the Pythagorean *triples* (a, b, c) satisfying $a^2 + b^2 = c^2$. A full account of Equation (4.1), including its most general solution, can be found in, for instance, [19, 25]. A partial, quadratic-form solution to (4.1) was given by Titus Piezas III in [20]

$$\left(au^2 - 2duv + av^2\right)^2 + \left(bu^2 - bv^2\right)^2 + \left(cu^2 - cv^2\right)^2 = \left(du^2 - 2auv + dv^2\right)^2, \tag{4.2}$$

where (a, b, c, d) is a Pythagorean quadruple and u and v are integer variables.[4]

On the other hand, setting $u = S_k$ in the algebraic identity[5]

$$u^2 + (1 + u)^2 + (u + u^2)^2 = (1 + u + u^2)^2, \tag{4.3}$$

and utilizing the formula (3.2), we obtain the following solution to Equation (4.1) in terms of sums of powers of integers:

$$\begin{aligned}\left(S_k\right)^2 + \left(1 + S_k\right)^2 + \left(S_k + \frac{2}{k+1}\sum_{j=0}^{k/2} B_{2j}\binom{k+1}{2j} S_{2k+1-2j}\right)^2 \\ = \left(1 + S_k + \frac{2}{k+1}\sum_{j=0}^{k/2} B_{2j}\binom{k+1}{2j} S_{2k+1-2j}\right)^2.\end{aligned} \tag{4.4}$$

[4]It is to be noted that, for the specific case in which $b^2 + c^2$ happens to be a perfect square, say e^2, Equation (4.2) becomes

$$\left(au^2 - 2duv + av^2\right)^2 + \left(eu^2 - ev^2\right)^2 = \left(du^2 - 2auv + dv^2\right)^2,$$

which constitutes a two-parameter solution to the Pythagorean equation $r^2 + s^2 = t^2$. For example, for $(a, b, c, d) = (8, 9, 12, 17)$, where $9^2 + 12^2 = 15^2$, we have

$$\left(8u^2 - 34uv + 8v^2\right)^2 + \left(15u^2 - 15v^2\right)^2 = \left(17u^2 - 16uv + 17v^2\right)^2.$$

[5]Clearly, Equation (4.3) is of the form $q_1^2 + q_2^2 + q_3^2 = q_4^2$, with each q_i being a quadratic form $q_i = \alpha_i u^2 + \beta_i uv + \gamma_i v^2$. For example, taking $v = 1$, $\alpha_1 = \gamma_1 = 0$, and $\beta_1 = 1$, we have $q_1 = u$.

For example, for $k = 2$, from Equation (4.4) we find (after multiplying by an overall factor of 3)

$$\big(3S_2\big)^2 + \big(3 + 3S_2\big)^2 + \big(3S_2 + S_3 + 2S_5\big)^2 = \big(3 + 3S_2 + S_3 + 2S_5\big)^2.$$

Now, replacing S_2, S_3, and S_5 by its respective polynomial in Table 1, and multiplying by an overall factor of 12, we arrive at the following identity

$$a^2 + (a + 18)^2 + b^2 = (b + 18)^2,$$

where

$$a = 3u + 9u^2 + 6u^3, \quad \text{and} \quad b = 3u + \frac{19}{2}u^2 + 9u^3 + \frac{13}{2}u^4 + 6u^5 + 2u^6,$$

with u taking any real or complex value. (It is easily seen that b is integer whenever u so is.)

Let us finally mention that, by replacing u with S_k^2 and v with S_m^2 in the basic identity $(u - v)^2 + (2\sqrt{uv})^2 = (u + v)^2$, one can generate infinite Pythagorean triangles through the relation

$$\big(S_k^2 - S_m^2\big)^2 + \big(2S_kS_m\big)^2 = \big(S_k^2 + S_m^2\big)^2. \tag{4.5}$$

Using Equations (3.1) and (3.2), the side lengths of the triangle can furthermore be written as a linear combination of power sums. For example, for $k = 1$ and $m = 3$, from Equation (4.5) we get (after multiplying by a global factor of 2) the relation

$$\big(S_5 + S_7 - 2S_3\big)^2 + \big(S_3 + 3S_5\big)^2 = \big(2S_3 + S_5 + S_7\big)^2.$$

This is to be compared with the following relation

$$\big(2S_5 + 2S_7 - S_3\big)^2 + \big(S_3 + 3S_5\big)^2 = \big(S_3 + 2S_5 + 2S_7\big)^2,$$

which was derived by Piza [21] using the algebraic identity $(y^4-y^2)^2/4+(2y^3)^2/4 = (y^4+y^2)^2/4$ and then taking $y = 2S_1$. Now, from the last two relations, we readily obtain

$$\big(S_5 + S_7 - 2S_3\big)^2 + \big(S_3 + 2S_5 + 2S_7\big)^2 = \big(2S_3 + S_5 + S_7\big)^2 + \big(2S_5 + 2S_7 - S_3\big)^2.$$

Taking into account that $S_5 + S_7 = 2S_3^2$, this relation can be simplified to (after dividing by the common factor S_3):

$$\big(2u - 2\big)^2 + \big(4u + 1\big)^2 = \big(2u + 2\big)^2 + \big(4u - 1\big)^2, \tag{4.6}$$

with $u = S_3$. The identity in Equation (4.6), which actually holds for arbitrary values of u, gives us a family of solutions to the Diophantine equation $a^2+b^2 = c^2+d^2$. For example, for $u = 17$, from Equation (4.6) we find that $32^2+69^2 = 36^2+67^2$.

Acknowledgment. The author thanks the referee for carefully going through the manuscript and for valuable comments.

References

[1] G. L. Alexanderson: *An anniversary for Bernoulli's Ars Conjectandi*, Math. Mag. 86.5 (2013), pp. 319–322,
doi: https://doi.org/10.4169/math.mag.86.5.319.

[2] T. M. Apostol: *A primer on Bernoulli numbers and polynomials*, Math. Mag. 81.3 (2008), pp. 178–190,
doi: https://doi.org/10.1080/0025570X.2008.11953547.

[3] E. J. Barbeau: *Power Play*, Washington, DC: The Mathematical Association of America, 1997.

[4] B. C. Berndt, S. Bhargava: *Ramanujan—for lowbrows*, Amer. Math. Monthly 100.7 (1993), pp. 644–656,
doi: https://doi.org/10.1080/00029890.1993.11990465.

[5] B. C. Berndt, Y. S. Choi, S. Y. Kang: *The problems submitted by Ramanujan to the Journal of the Indian Mathematical Society*, in: Ramanujan: essays and surveys, History of Mathematics, Vol. 22, Providence (RI): American Mathematical Society, 2001, pp. 215–258.

[6] J. L. Cereceda: *Averaging sums of powers of integers and Faulhaber polynomials*, Ann. Math. Inform. 42 (2013), pp. 105–117.

[7] M. Chamberland: *Families of solutions of a cubic Diophantine equation*, Fibonacci Quart. 38.3 (2000), pp. 250–254.

[8] A. Choudhry: *On equal sums of cubes*, Rocky Mountain J. Math. 28.4 (1998), pp. 1251–1257,
doi: https://doi.org/10.1216/rmjm/1181071714.

[9] R. Cook: *Sums of powers*, Math. Spectrum 32.1 (1999/2000), pp. 7–10.

[10] E. Y. Deeba, D. M. Rodriguez: *Bernoulli numbers and trigonometric functions*, Internat. J. Math. Ed. Sci. Tech. 21.2 (1990), pp. 275–282,
doi: https://doi.org/10.1080/0020739900210214.

[11] L. E. Dickson: *History of the Theory of Numbers, Volume II: Diophantine Analysis*, New York: Dover Publications, 2005.

[12] J. D. Harper: *Ramanujan, quadratic forms, and the sum of three cubes*, Math. Mag. 86.4 (2013), pp. 275–279,
doi: https://doi.org/10.4169/math.mag.86.4.275.

[13] R. Hersh: *Why the Faulhaber polynomials are sums of even or odd powers of* $(n + 1/2)$, College Math. J. 43.4 (2012), pp. 322–324,
doi: https://doi.org/10.4169/college.math.j.43.4.322.

[14] T. C. T. Kotiah: *Sums of powers of integers—A review*, Internat. J. Math. Ed. Sci. Tech. 24.6 (1993), pp. 863–874,
doi: https://doi.org/10.1080/0020739930240611.

[15] J. A. MacDougall: *Identities relating sums of powers of integers – An exercise in generalization*, Australian Senior Math. J. 2.1 (1988), pp. 53–62.

[16] U. Mukhopadhyay: *Ramanujan and some elementary mathematical problems*, At Right Angles (March 2018), pp. 13–21.

[17] N. J. Newsome, M. S. Nogin, A. H. Sabuwala: *A proof of symmetry of the power sum polynomials using a novel Bernoulli number identity*, J. Integer Seq. 20 (2017), Article 17.6.6.

[18] J. W. Nicholson: *A simple solution of the Diophantine equation* $U^3 = V^3 + X^3 + Y^3$, Amer. Math. Monthly 22.7 (1915), pp. 224–225,
doi: https://doi.org/10.1080/00029890.1915.11998120.

[19] P. Oliverio: *Self-generating Pythagorean quadruples and N-tuples*, Fibonacci Quart. 34.2 (1996), pp. 98–101.

[20] T. Piezas III: *A Collection of Algebraic Identities: Sums of Three Squares*,
url: https://sites.google.com/site/tpiezas/004.

[21] P. A. Piza: *Powers of sums and sums of powers*, Math. Mag. 25.3 (1952), pp. 137–142,
doi: https://doi.org/10.2307/3029445.

[22] C. Sándor: *On the equation* $a^3+b^3+c^3 = d^3$, Period. Math. Hungar. 33.2 (1996), pp. 121–134,
doi: https://doi.org/10.1007/BF02093510.

[23] P. Schumer: *Sum of two cubes in two different ways*, Math. Horizons 16.2 (2008), pp. 8–9,
doi: https://doi.org/10.1080/10724117.2008.11974795.

[24] J. H. Silverman: *Taxicabs and sums of two cubes*, Amer. Math. Monthly 100.4 (1993), pp. 331–340,
doi: https://doi.org/10.1080/00029890.1993.11990409.

[25] R. Spira: *The Diophantine equation* $x^2 + y^2 + z^2 = m^2$, Amer. Math. Monthly 69.5 (1962), pp. 360–365,
doi: https://doi.org/10.1080/00029890.1962.11989898.

[26] J. Tanton: *Sums of powers*, Math. Horizons 11.1 (2003), pp. 15–20,
doi: https://doi.org/10.1080/10724117.2003.12021732.

[27] K. S. Williams: *Bernoulli's identity without calculus*, Math. Mag. 70.1 (1997), pp. 47–50,
doi: https://doi.org/10.1080/0025570X.1997.11996499.

[28] D. W. Wu: *Bernoulli numbers and sums of powers*, Internat. J. Math. Ed. Sci. Tech. 32.3 (2001), pp. 440–443,
doi: https://doi.org/10.1080/00207390117519.

Annales Mathematicae et Informaticae
***52** (2020) pp. 85–95*
DOI: https://doi.org/10.33039/ami.2020.08.001
URL: https://ami.uni-eszterhazy.hu

A sum of negative degrees of the gaps values in 2 and 3-generated numerical semigroups

Leonid G. Fel[a*], **Takao Komatsu**[b],
Ade Irma Suriajaya[c]

[a]Department of Civil Engineering, Technion,
Haifa 32000, Israel
lfel@technion.ac.il

[b]Department of Mathematical Sciences, School of Science,
Zhejiang Sci-Tech University,
Hangzhou 310018, China
komatsu@zstu.edu.cn

[c]Faculty of Mathematics, Kyushu University,
744 Motooka, Nishi-ku,
Fukuoka 819-0395, Japan
adeirmasuriajaya@math.kyushu-u.ac.jp

Submitted: January 25, 2020
Accepted: August 27, 2020
Published online: September 2, 2020

Abstract

We show explicit expressions for an inverse power series over the gaps values of numerical semigroups generated by two and three integers. As an application, a set of identities of the Hurwitz zeta functions is derived.

Keywords: numerical semigroups, gaps and non-gaps, the Hurwitz zeta function

MSC: Primary 20M14; Secondary 11P81

*The research of LGF was supported in part by the Kamea Fellowship.

1. Introduction

A sum of integer powers of gaps values in numerical semigroups $S_m = \langle d_1, \ldots, d_m \rangle$ with $\gcd(d_1, \ldots, d_m) = 1$, is referred often as the semigroup series

$$g_n(S_m) = \sum_{s \in \mathbb{N} \setminus S_m} s^n, \quad n \in \mathbb{Z},$$

where $\mathbb{N} \setminus S_m$ is known as the set of gaps of S_m and $g_0(S_m)$ is called the genus of S_m. The semigroup series $g_n(S_m)$ has been attractive by many researchers for $n \geq 0$. In particular, an explicit expression of $g_n(S_2)$ and implicit expression of $g_n(S_3)$ were given in [6] and [4], respectively. However, the series $g_n(S_m)$ for negative integers n has not seemingly treated so often. In this paper we derive a formula for semigroup series $g_{-n}(S_2) = \sum_{s \in \mathbb{N} \setminus S_2} s^{-n}$ and $g_{-n}(S_3) = \sum_{s \in \mathbb{N} \setminus S_3} s^{-n}$ $(n \geq 1)$. In fact, it will be known that such series are related with zeta functions in Number theory.

Consider a numerical semigroup $S_2 = \langle d_1, d_2 \rangle$, generated by two integers $d_1, d_2 \geq 2$ with $\gcd(d_1, d_2) = 1$. Here, the Hilbert series $H(z; S_2)$ and the gaps generating function $\Phi(z; S_2)$ are given as

$$H(z; S_2) = \sum_{s \in S_2} z^s \quad \text{and} \quad \Phi(z; S_2) = \sum_{s \in \mathbb{N} \setminus S_2} z^s,$$

respectively, satisfying

$$H(z; S_2) + \Phi(z; S_2) = \frac{1}{1-z} \quad (z < 1), \tag{1.1}$$

where $\min\{\mathbb{N} \setminus S_2\} = 1$, and $\max\{\mathbb{N} \setminus S_2\} = d_1 d_2 - d_1 - d_2$ is called the Frobenius number and is denoted by F_2. A rational representation (Rep) of $H(z; S_2)$ is given by

$$H(z; S_2) = \frac{1 - z^{d_1 d_2}}{(1 - z^{d_1})(1 - z^{d_2})}. \tag{1.2}$$

We introduce a new generating function $\Psi_1(z; S_2)$, defined by

$$\Psi_1(z; S_2) = \int_0^z \frac{\Phi(t; S_2)}{t} \, \mathrm{d}t = \sum_{s \in \mathbb{N} \setminus S_2} \frac{z^s}{s} \quad \text{with} \quad \Psi_1(1; S_2) = g_{-1}(S_2). \tag{1.3}$$

Substituting (1.1) into (1.3), we obtain

$$\Psi_1(z; S_2) = \int_0^z \left(\frac{1}{1-t} - H(t; S_2) \right) \frac{\mathrm{d}t}{t}. \tag{1.4}$$

Since $(1 - t^{d_i})^{-1} = \sum_{k_i=0}^{\infty} t^{k_i d_i}$, by substituting (1.4) into (1.2), we obtain

$$H(t; S_2) = \sum_{k_1, k_2 = 0}^{\infty} t^{k_1 d_1 + k_2 d_2} - \sum_{k_1, k_2 = 0}^{\infty} t^{k_1 d_1 + k_2 d_2 + d_1 d_2}. \tag{1.5}$$

Indeed, an expression (1.5) is an infinite series with degrees $s = k_1 d_1 + k_2 d_2$ running over all nodes in the following sublattice $\mathbb{K}$ of the integer lattice $\mathbb{Z}_2$.

$$\mathbb{K} = \{0,0\} \cup \mathbb{K}_1 \cup \mathbb{K}_2, \qquad \begin{cases} \mathbb{K}_1 = \{1 \le k_1 \le d_2 - 1, \;\; k_2 = 0\}, \\ \mathbb{K}_2 = \{0 \le k_1 \le d_2 - 1, \;\; 1 \le k_2 \le \infty\}. \end{cases} \tag{1.6}$$

In Figure 1, as an example, we present a part of the integer lattice $\mathbb{K}$ for the numerical semigroup

$$\langle 5, 8\rangle = \{0, 5, 8, 10, 13, 15, 16, 18, 20, 21, 23, 24, 25, 26, 28, \longmapsto\},$$

where the symbol $\longmapsto$ denotes an infinite set of positive integers exceeding 28.

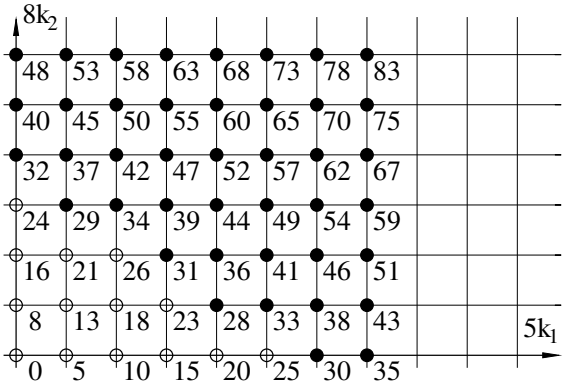


Figure 1: A part of the integer lattice $\mathbb{K} \subset \mathbb{Z}_2$ for the numerical semigroup $\langle 5, 8\rangle$. The nodes mark the non-gaps of semigroup: the values, assigned to the black and white nodes, exceed and precede $F_2 = 27$, respectively.

Proposition 1.1. *There exists a bijection between the infinite set of nodes in the integer lattice* $\mathbb{K}$ *and an infinite set of non-gaps of the semigroup* $\langle d_1, d_2\rangle$.

Proof. We have to prove two statements of existence and uniqueness:

1) Every $s \in \langle d_1, d_2\rangle$ has its Rep node in $\mathbb{K}$,

2) All $s \in \langle d_1, d_2\rangle$ have their Rep nodes in $\mathbb{K}$ only once.

1) Let $s \in \langle d_1, d_2\rangle$ be given. Then by definition of $\langle d_1, d_2\rangle$ an integer s has Rep,

$$s = k_1 d_1 + k_2 d_2, \quad k_1, k_2 \in \mathbb{Z}, \quad k_1, k_2 \ge 0. \tag{1.7}$$

Choose s such that $k_1 = p d_2 + q$, where $p = \lfloor k_1/d_2 \rfloor$, namely, $0 \le q \le d_2 - 1$, and $\lfloor x \rfloor$ denotes the integer part of a real number x. Then Rep (1.7) is expressed as

$$s = q d_1 + (k_2 + p d_1) d_2,$$

and s has its Rep node in $\mathbb{K}$.

2) By way of contradiction, assume that there exist two nodes $\{k_1, k_2\} \in \mathbb{K}$ and $\{l_1, l_2\} \in \mathbb{K}$ such that

$$\begin{aligned} &k_1 d_1 + k_2 d_2 = l_1 d_1 + l_2 d_2, \\ &0 \le k_1, l_1 \le d_2 - 1, \quad 0 \le k_2, l_2 \le \infty, \quad k_1 > l_1, \ \ k_2 < l_2, \end{aligned} \tag{1.8}$$

namely, that there exists such $s \in \langle d_1, d_2 \rangle$ which has two different Rep nodes in $\mathbb{K}$. Rewrite equality (1.8) as follows.

$$(k_1 - l_1) d_1 = (l_2 - k_2) d_2. \tag{1.9}$$

Since $\gcd(d_1, d_2) = 1$, the equality (1.9) implies that

$$k_1 - l_1 = b d_2 \quad (b \ge 1) \quad \Longrightarrow \quad k_1 = l_1 + b d_2 \quad \Longrightarrow \quad k_1 \ge d_2,$$

contradicting the assumption $\{k_1, k_2\} \in \mathbb{K}$. □

2. A sum of the inverse gaps values $g_{-1}(S_2)$

Rewrite the integral in (1.4) as follows.

$$\Psi_1(z; S_2) = \int_0^z \left(\sum_{k=0}^{\infty} t^{k-1} - \frac{H(t; S_2)}{t} \right) \mathrm{d}t, \tag{2.1}$$

where

$$\begin{aligned} &\frac{H(t; S_2)}{t} = \sum_{j=0}^{2} h_j(t; S_2), && h_0(t; S_2) = \frac{1}{t}, \\ &h_1(t; S_2) = \sum_{k_1=1}^{d_2-1} t^{k_1 d_1 - 1}, && h_2(t; S_2) = \sum_{k_1, k_2 \in \mathbb{K}_2} t^{k_1 d_1 + k_2 d_2 - 1}. \end{aligned}$$

By integration we obtain from (2.1),

$$\Psi_1(z; S_2) = \sum_{k=1}^{\infty} \frac{z^k}{k} - \frac{1}{d_1} \sum_{k_1=1}^{d_2-1} \frac{z^{k_1 d_1}}{k_1} - \sum_{k_1, k_2 \in \mathbb{K}_2} \frac{z^{k_1 d_1 + k_2 d_2}}{k_1 d_1 + k_2 d_2}, \tag{2.2}$$

and deduce by (1.3) and (1.6),

$$g_{-1}(S_2) = \sum_{k=1}^{\infty} \frac{1}{k} - \sum_{k_1, k_2 \in \mathbb{K}_2} \frac{1}{k_1 d_1 + k_2 d_2} - \frac{1}{d_1} \sum_{k_1=1}^{d_2-1} \frac{1}{k_1}. \tag{2.3}$$

By Proposition 1.1, after subtraction in (2.3) there is a finite number of terms left, since all terms, which exceed F_2 in the two first infinite series in (2.3), are cancelled. To emphasize that fact, we represent formula (2.3) as follows.

$$g_{-1}(S_2) = \sum_{k=1}^{c_2} \frac{1}{k} - \sum_{k_1,k_2\in\mathbb{K}_2}^{k_1d_1+k_2d_2\le c_2} \frac{1}{k_1d_1+k_2d_2} - \frac{1}{d_1}\sum_{k_1=1}^{d_2-1}\frac{1}{k_1},$$

where $c_2 = F_2 + 1$ is called the conductor of semigroup S_2.

3. A sum of the negative degrees of gaps values $g_{-n}(S_2)$

We generalize formula (2.2) and introduce a new generating function $\Psi_n(z; S_2)$ $(n \ge 2)$

$$\Psi_n(z;S_2) = \int_0^z \frac{\mathrm{d}t_1}{t_1}\int_0^{t_1}\frac{\mathrm{d}t_2}{t_2}\dots\int_0^{t_{n-1}}\Phi(t_n;S_2)\frac{\mathrm{d}t_n}{t_n} = \sum_{s\in\mathbb{N}\setminus S_2}\frac{z^s}{s^n}, \tag{3.1}$$

where $\Psi_n(1; S_2) = g_{-n}(S_2)$ and satisfies the following recursive relation.

$$\Psi_{k+1}(t_{n-k-1};S_2) = \int_0^{t_{n-k-1}}\frac{\mathrm{d}t_{n-k}}{t_{n-k}}\Psi_k(t_{n-k};S_2), \quad k\ge 0,$$
$$\Psi_0(t_n;S_2) = \Phi(t_{n-1};S_2), \quad t_0 = z.$$

Namely,

$$\Psi_1(t_{n-1};S_2) = \int_0^{t_{n-1}}\frac{\mathrm{d}t_n}{t_n}\Psi_0(t_n;S_2),$$
$$\Psi_2(t_{n-2};S_2) = \int_0^{t_{n-2}}\frac{\mathrm{d}t_{n-1}}{t_{n-1}}\Psi_1(t_{n-1};S_2), \quad \dots.$$

By integration in (3.1), we obtain

$$\Psi_n(z;S_2) = \sum_{k=1}^{\infty}\frac{z^k}{k^n} - \frac{1}{d_1^n}\sum_{k_1=1}^{d_2-1}\frac{z^{k_1d_1}}{k_1^n} - \sum_{k_1,k_2\in\mathbb{K}_2}\frac{z^{k_1d_1+k_2d_2}}{(k_1d_1+k_2d_2)^n}.$$

Thus, for $z = 1$ we have

$$g_{-n}(S_2) = \sum_{k=1}^{\infty}\frac{1}{k^n} - \sum_{k_1=0}^{d_2-1}\sum_{k_2=1}^{\infty}\frac{1}{(k_1d_1+k_2d_2)^n} - \frac{1}{d_1^n}\sum_{k_1=1}^{d_2-1}\frac{1}{k_1^n}, \quad n\ge 2. \tag{3.2}$$

Denoting the ratio d_1/d_2 by δ, we can rewrite (3.2) as

$$g_{-n}(S_2) = \sum_{k=1}^{\infty} \frac{1}{k^n} - \frac{1}{d_2^n} \sum_{k_2=1}^{\infty} \frac{1}{k_2^n} - \frac{1}{d_2^n} \sum_{k_1=1}^{d_2-1} \sum_{k_2=1}^{\infty} \frac{1}{(k_1\delta + k_2)^n} - \frac{1}{d_1^n} \sum_{k_1=1}^{d_2-1} \frac{1}{k_1^n}.$$

Making use of the Hurwitz $\zeta(n,q) = \sum_{k=0}^{\infty}(k+q)^{-n}$ and Riemann zeta functions $\zeta(n) = \zeta(n,1)$, we represent the last formula as follows.

$$g_{-n}(S_2) = \left(1 - \frac{1}{d_2^n}\right)\zeta(n) - \frac{1}{d_2^n} \sum_{k_1=1}^{d_2-1} \zeta(n, k_1\delta), \quad n \geq 2. \tag{3.3}$$

On interchanging the generators d_1 and d_2 in (3.3), we obtain an alternative expression for $g_{-n}(S_2)$:

$$g_{-n}(S_2) = \left(1 - \frac{1}{d_1^n}\right)\zeta(n) - \frac{1}{d_1^n} \sum_{k_2=1}^{d_1-1} \zeta\left(n, \frac{k_2}{\delta}\right). \tag{3.4}$$

4. Symmetric 3-generated numerical semigroup

We deal with symmetric numerical semigroup $S_3 = \langle d_1, d_2, d_3\rangle$ generated by three integers with the Hilbert series $H(z; S_3)$, satisfying minimal relations,

$$H(z; S_3) = \frac{\left(1 - z^{a_{22}d_2}\right)\left(1 - z^{a_{33}d_3}\right)}{(1 - z^{d_1})(1 - z^{d_2})(1 - z^{d_3})} \quad (a_{22}, a_{33} \geq 2), \tag{4.1}$$

with $a_{11}d_1 = a_{22}d_2$, $a_{33}d_3 = a_{31}d_1 + a_{32}d_2$ (see [3]). In this section, we prove a statement which is necessary to establish the convergence for $g_1(z, S_3)$, namely, the difference between two divergent infinite series is convergent

$$g_1(z, S_3) = \sum_{k=1}^{\infty} \frac{1}{k} - \sum_{k_2=0}^{a_{22}-1} \sum_{k_3=0}^{a_{33}-1} \sum_{k_1=0}^{\infty} \frac{1}{k_1d_1 + k_2d_2 + k_3d_3}, \quad \sum_{j=1}^{3} k_j \geq 1. \tag{4.2}$$

The idea is to prove that after cancellation of identical terms, a finite number of terms is left in (4.2).

We consider the sublattice $\widetilde{\mathbb{L}} = \mathbb{L} \cup \{0,0,0\}$ of the integer lattice $\mathbb{Z}_3$, where

$$\mathbb{L} = \bigcup_{\substack{k_1=0 \\ k_1+k_2+k_3 \geq 1}}^{\infty} \mathbb{L}_{k_1}, \qquad \mathbb{L}_{k_1} = \bigcup_{k_2,k_3}^{\infty} \{k_1, k_2, k_3\},$$

with $0 \leq k_2 < a_{22}$ and $0 \leq k_3 < a_{33}$. In Figure 2, we present a part of the integer lattice $\widetilde{\mathbb{L}}$ for the numerical semigroup $\langle 4, 7, 10\rangle$.

Proposition 4.1. *There exists a bijection between the infinite set of nodes in the integer lattice $\widetilde{\mathbb{L}}$ and an infinite set of non-gaps of the semigroup $\langle d_1, d_2, d_3\rangle$.*

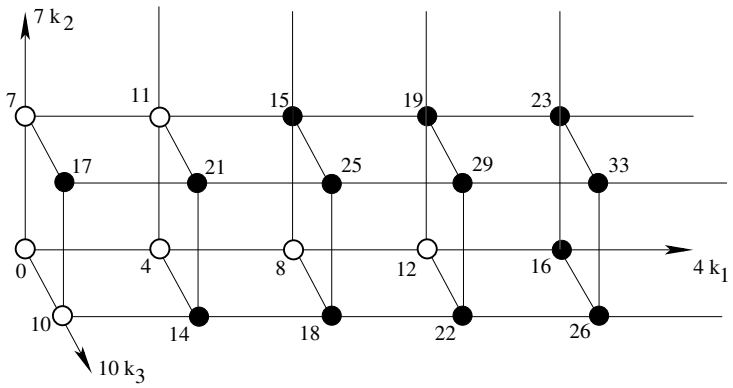


Figure 2: A part of the integer lattice $\widetilde{\mathbb{L}} \subset \mathbb{Z}_3$ for $\langle 4, 7, 10\rangle$. The nodes mark the non-gaps of semigroup: the values, assigned to the black and white nodes, exceed and precede the Frobenius number $F_3 = 13$.

Proof. We have to prove both existence and uniqueness.

1) Every $s \in \langle d_1, d_2, d_3\rangle$ has its representative node in $\widetilde{\mathbb{L}}$.

2) All $s \in \langle d_1, d_2, d_3\rangle$ have their representative nodes in $\widetilde{\mathbb{L}}$ only once.

1) Let $s \in \langle d_1, d_2, d_3\rangle$ be given. Then by definition of $\langle d_1, d_2, d_3\rangle$ an integer s has a representation,

$$s = k_1 d_1 + k_2 d_2 + k_3 d_3, \quad 0 \le k_1, k_2, k_3 < \infty. \tag{4.3}$$

Choose s such that

$$k_2 = p_2 a_{22} + q_2, \quad k_3 = p_3 a_{33} + q_3, \quad \text{namely,} \quad p_2 = \left\lfloor \frac{k_2}{a_{22}} \right\rfloor, \; p_3 = \left\lfloor \frac{k_3}{a_{33}} \right\rfloor, \tag{4.4}$$
$$p_2, p_3, q_2, q_3 \in \mathbb{Z}, \quad p_2, p_3 \ge 0, \quad 0 \le q_2 < a_{22}, \quad 0 \le q_3 < a_{33}.$$

By substituting (4.4) into (4.3), we get

$$s = k_1 d_1 + (p_2 a_{22} + q_2) d_2 + (p_3 a_{33} + q_3) d_3. \tag{4.5}$$

Combining (4.5) with minimal relations (4.1), we obtain

$$\begin{aligned} s &= (k_1 + p_2 a_{11}) d_1 + p_3 (a_{31} d_1 + a_{32} d_2) + q_2 d_2 + q_3 d_3 \\ &= (k_1 + p_2 a_{11} + p_3 a_{31}) d_1 + (p_3 a_{32} + q_2) d_2 + q_3 d_3. \end{aligned} \tag{4.6}$$

If $p_3 a_{32} + q_2 < a_{22}$, then s has its representative node in $\widetilde{\mathbb{L}}$. But, if $p_3 a_{32} + q_2 \ge a_{22}$, let us write

$$p_3 a_{32} + q_2 = p_4 a_{22} + q_4, \quad p_4 \ge 0, \quad 0 \le q_4 < a_{22}, \quad p_4 = \left\lfloor \frac{p_3 a_{32} + q_2}{a_{22}} \right\rfloor. \tag{4.7}$$

Substitute (4.7) into (4.6) and get

$$s = (k_1 + p_2 a_{11} + p_3 a_{31} + p_4 a_{11})d_1 + q_4 d_2 + q_3 d_3,$$

and s still has its representative node in $\widetilde{\mathbb{L}}$.
2) By way of contradiction, assume that there exist two nodes $\{k_1, k_2, k_3\} \in \widetilde{\mathbb{L}}$ and $\{l_1, l_2, l_3\} \in \widetilde{\mathbb{L}}$ such that

$$k_1 d_1 + k_2 d_2 + k_3 d_3 = l_1 d_1 + l_2 d_2 + l_3 d_3, \tag{4.8}$$
$$0 \le k_1 \ne l_1 < \infty, \quad 0 \le k_2 \ne l_2 < a_{22}, \quad 0 \le k_3 \ne l_3 < a_{33}. \tag{4.9}$$

The case, when one of the differences $k_j - l_j$ vanishes, will be considered later. Suppose that $k_1 - l_1 > 0$, and $k_2 - l_2 < 0$, $k_3 - l_3 < 0$. In fact, due to (4.9) we also have to include the upper bound

$$0 < l_2 - k_2 < a_{22}, \quad 0 < l_3 - k_3 < a_{33}. \tag{4.10}$$

Rewrite (4.8) as

$$(k_1 - l_1)d_1 = (l_2 - k_2)d_2 + (l_3 - k_3)d_3,$$

where $k_1 - l_1 \ge a_{11}$, otherwise (due to minimal relations) equation (4.8) would have trivial solution $k_j = l_j$ $(j = 1, 2, 3)$. But the last contradicts (4.9), namely, $k_1 \ne l_1, k_2 \ne l_2, k_3 \ne l_3$.

If so, represent $k_1 - l_1 = u_1 a_{11} + v_1$ with $u_1 \ge 1$, $0 \le v_1 < a_{11}$, then

$$(u_1 a_{11} + v_1)d_1 = u_1 a_{22} d_2 + v_1 d_1 = (l_2 - k_2)d_2 + (l_3 - k_3)d_3. \tag{4.11}$$

Rewrite (4.11) as

$$(l_3 - k_3)d_3 = v_1 d_1 + (u_1 a_{22} - (l_2 - k_2))d_2, \tag{4.12}$$

and note that the both terms on the right-hand side in (4.12) are positive by (4.10),

$$0 < l_2 - k_2 < a_{22} < u_1 a_{22}. \tag{4.13}$$

However, $0 < l_3 - k_3 < a_{33}$ by (4.10), and (due to minimal relations) equation (4.12) has only a trivial solution, $l_3 = k_3$, $v_1 = 0$, $l_2 = k_2 + u_1 a_{22}$. But the last contradicts an inequality (4.13).

Now, consider the case when

$$a_{33} > k_3 - l_3 > 0, \quad 0 < l_1 - k_1, \quad 0 < l_2 - k_2 < a_{22},$$

and write

$$(k_3 - l_3)d_3 = (l_1 - k_1)d_1 + (l_2 - k_2)d_2. \tag{4.14}$$

But (due to minimal relations) equation (4.14) has only trivial solution $k_j = l_j$ $(j = 1, 2, 3)$, that contradicts (4.9), namely, $k_1 \ne l_1, k_2 \ne l_2, k_3 \ne l_3$.

Next, consider the case when

$$l_1 - k_1 = 0, \quad 0 < l_2 - k_2 < a_{22}, \quad a_{33} > k_3 - l_3 > 0, \tag{4.15}$$

and write

$$(k_3 - l_3)d_3 = (l_2 - k_2)d_2. \tag{4.16}$$

But (due to minimal relations) equation (4.16) has only a trivial solution, $l_3 = k_3$, $l_2 = k_2$, that contradicts (4.15). For similar reasons the case

$$k_3 - l_3 = 0, \quad 0 < k_1 - l_1 < a_{11}, \quad 0 < l_2 - k_2 < a_{22}, \tag{4.17}$$

leads to an equality

$$(k_1 - l_1)d_1 = (l_2 - k_2)d_2,$$

which also has only a trivial solution, $l_1 = k_1$, $l_2 = k_2$, that contradicts (4.17). Thus, what is left

$$l_1 = k_1, \quad l_2 = k_2, \quad l_3 = k_3,$$

and the result is proven. □

5. Identities for the Hurwitz zeta function

As an application, our argument can be deduced to the multiplication theorem in Hurwitz zeta functions. Indeed, combining formulas (3.3) and (3.4), we get an identity

$$\delta^n \sum_{k=1}^{d_2-1} \zeta(n, k\delta) = (1 - \delta^n)\,\zeta(n) + \sum_{k=1}^{d_1-1} \zeta\left(n, \frac{k}{\delta}\right).$$

Another spinoff of formulas (3.3) and (3.4) is a set of identities for Hurwitz zeta functions. For example, consider the numerical semigroup $\langle 3, 4\rangle$ with three gaps $\mathbb{N} \setminus \langle 3, 4\rangle = \{1, 2, 5\}$. Substituting it into (3.3) and (3.4), we have

$$\zeta\left(n, \frac{3}{4}\right) + \zeta\left(n, \frac{6}{4}\right) + \zeta\left(n, \frac{9}{4}\right) = (4^n - 1)\zeta(n) - \left(4^n + 2^n + \left(\frac{4}{5}\right)^n\right)$$

and

$$\zeta\left(n, \frac{4}{3}\right) + \zeta\left(n, \frac{8}{3}\right) = (3^n - 1)\zeta(n) - \left(3^n + \left(\frac{3}{2}\right)^n + \left(\frac{3}{5}\right)^n\right),$$

respectively.

We shall show that the identity (3.3) can be deduced to the multiplication theorem in Hurwitz zeta functions (see, e.g., [1, p.249], [2, (16), p.71]). It is similar for (3.4).

Since $\gcd(d_1, d_2) = 1$, if $k_1 d_1 \equiv k_2 d_1 \pmod{d_2}$ then $k_1 \equiv k_2 \pmod{d_2}$. Therefore,

$$\zeta\left(n, \left\{\frac{d_1}{d_2}\right\}\right) + \zeta\left(n, \left\{\frac{2d_1}{d_2}\right\}\right) + \cdots + \zeta\left(n, \left\{\frac{(d_2 - 1)d_1}{d_2}\right\}\right)$$

$$= \zeta\left(n, \frac{1}{d_2}\right) + \zeta\left(n, \frac{2}{d_2}\right) + \cdots + \zeta\left(n, \frac{d_2-1}{d_2}\right),$$

where $\{x\}$ denotes the fractional part of a real number x. There exists a nonnegative integer a such that

$$\frac{ad_1}{d_2} < 1 < \frac{(a+1)d_1}{d_2}.$$

Then for any integer k' with $a < k' \le d_2 - 1$ there exists a positive integer l' such that $1 \le k'd_1 - l'd_2 < d_2$, and

$$\zeta\left(n, \frac{k'd_1}{d_2}\right) = \zeta\left(n, \frac{k'd_1 - l'd_2}{d_2}\right) - \left(\frac{d_2}{k'd_1 - l'd_2}\right)^n - \left(\frac{d_2}{k'd_1 - (l'-1)d_2}\right)^n - \cdots - \left(\frac{d_2}{k'd_1 - d_2}\right)^n, \quad (5.1)$$

where

$$\frac{k'd_1 - l'd_2}{d_2} = \left\{\frac{k'd_1}{d_2}\right\}.$$

For any positive integer r, there exist integers x and y such that $r = xd_1 + yd_2$. If $0 \le x < d_2$, then r can be expressed uniquely. Thus, if $y \ge 0$, then $r \in S_2$. If $y < 0$, then $r \notin S_2$. The largest integer is given by $(d_2 - 1)d_1 - d_2$, that is exactly the same as the Frobenius number $F(d_1, d_2)$. Thus, $k'd_1 - l''d_2 \notin S_2$ for all l'' with $1 \le l'' \le l'$ in (5.1). In addition, if $k_1d_1 - l_1d_2 = k_2d_1 - l_2d_2$, then by $\gcd(d_1, d_2) = 1$ we have $d_1|(k_1 - k_2)$ and $d_2|(l_1 - l_2)$. As $0 < k_1, k_2 < d_2$ and $0 < l_1, l_2 < d_1$, we get $k_1 = k_2$ and $l_1 = l_2$. Thus, all such numbers of the form $kd_1 - ld_2 \notin S_2$ are different.

In [5, (3.32)] for a real ξ and $d = \gcd(d_1, d_2)$

$$\sum_{k=0}^{d_2-1} \left\lfloor \frac{kd_1 + \xi}{d_2} \right\rfloor = d\left\lfloor \frac{\xi}{d} \right\rfloor + \frac{(d_1-1)(d_2-1)}{2} + \frac{d-1}{2}. \quad (5.2)$$

Hence, by (5.2) with $d = 1$ and $\xi = 0$, the total number of non-representable positive integers of the form $kd_1 - ld_2$ ($a < k < d_2$, $l = 1, 2, \ldots, \lfloor kd_1/d_2 \rfloor - 1$) is

$$\sum_{k=1}^{d_2-1} \left\lfloor \frac{kd_1}{d_2} \right\rfloor = \frac{(d_1-1)(d_2-1)}{2},$$

which is exactly the same as the number of integers without non-negative integer representations by d_1 and d_2, that was given by Sylvester in 1882. Therefore, the right-hand side of (3.3) is

$$\left(1 - \frac{1}{d_2^n}\right)\zeta(n) - \frac{1}{d_2^n}\sum_{k_1=1}^{d_2-1} \zeta\left(n, \frac{k_1d_1}{d_2}\right)$$

$$= \left(1 - \frac{1}{d_2^n}\right)\zeta(n) - \frac{1}{d_2^n}\left(\sum_{k_1=1}^{d_2-1}\zeta\left(n, \left\{\frac{k_1 d_1}{d_2}\right\}\right) - d_2^n \sum_{s\in\mathbb{N}\setminus S_2} s^{-n}\right)$$

$$= \left(1 - \frac{1}{d_2^n}\right)\zeta(n) - \frac{1}{d_2^n}\sum_{k_1=1}^{d_2-1}\zeta\left(n, \frac{k}{d_2}\right) + \sum_{s\in\mathbb{N}\setminus S_2} s^{-n}.$$

On the other hand, the left-hand side of (3.3) is

$$g_{-n}(S_2) = \sum_{s\in\mathbb{N}\setminus S_2} s^{-n}.$$

Therefore, we obtain that

$$\sum_{k=1}^{d_2}\zeta\left(n, \frac{k}{d_2}\right) = d_2^n\zeta(n),$$

which is the multiplication theorem in Hurwitz zeta functions.

Acknowledgements. This work was partly done when the first author visited the second author's institute and the second author visited the third author's institute. We would like to thank both researchers for their hospitality and discussion. We thank the anonymous referee for careful reading of our manuscript and many insightful comments and suggestions.

References

[1] T. M. Apostol: *Introduction to analytic number theory.* English, Springer, Cham, 1976, doi: https://doi.org/10.1007/978-1-4757-5579-4.

[2] H. Davenport: *Multiplicative number theory. 2nd ed. Rev. by Hugh L. Montgomery.* English, vol. 74, Springer, New York, NY, 1980, doi: https://doi.org/10.1007/978-1-4757-5927-3.

[3] L. G. Fel: *Frobenius problem for semigroups* $\mathbf{S}(d_1, d_2, d_3)$, English, Funct. Anal. Other Math. 1.2 (2006), pp. 119–157, issn: 1991-0061; 1863-7914/e, doi: https://doi.org/10.1007/s11853-007-0009-5.

[4] L. G. Fel, B. Y. Rubinstein: *Power sums related to semigroups* $S(d_1, d_2, d_3)$. English, Semigroup Forum 74.1 (2007), pp. 93–98, issn: 0037-1912; 1432-2137/e, doi: https://doi.org/10.1007/s00233-006-0658-6.

[5] R. L. Graham, D. E. Knuth, O. Patashnik: *Concrete mathematics: a foundation for computer science. 2nd ed.* English, 2nd ed., Amsterdam: Addison-Wesley Publishing Group, 1994, pp. xiii + 657, isbn: 0-201-55802-5/hbk.

[6] Ö. J. Rödseth: *A note on Brown and Shiue's paper on a remark related to the Frobenius problem.* English, Fibonacci Q. 32.5 (1994), pp. 407–408, issn: 0015-0517.

Annales Mathematicae et Informaticae
***52** (2020) pp. 97–105*
DOI: https://doi.org/10.33039/ami.2020.10.002
URL: https://ami.uni-eszterhazy.hu

Combinatorial sums associated with balancing and Lucas-balancing polynomials

Robert Frontczak[a*], **Taras Goy**[b]

[a]Landesbank Baden-Württemberg (LBBW), Stuttgart, Germany
robert.frontczak@lbbw.de

[b]Vasyl Stefanyk Precarpathian National University,
Faculty of Mathematics and Computer Science, Ivano-Frankivsk, Ukraine
taras.goy@pnu.edu.ua

Submitted: April 17, 2020
Accepted: October 20, 2020
Published online: October 29, 2020

Abstract

The aim of the paper is to use some identities involving binomial coefficients to derive new combinatorial identities for balancing and Lucas-balancing polynomials. Evaluating these identities at specific points, we can also establish some combinatorial expressions for Fibonacci and Lucas numbers.

Keywords: Balancing polynomial, Lucas-balancing polynomial, balancing number, Fibonacci number, Lucas number.

MSC: 11B39, 11B83, 05A10

1. Introduction

Balancing polynomials $(B_n(x))_{n\geq 0}$ and Lucas-balancing polynomials $(C_n(x))_{n\geq 0}$ are defined for $x \in \mathbb{C}$ by the recurrences [17]

$$B_n(x) = 6xB_{n-1}(x) - B_{n-2}(x), \quad n \geq 2,$$

*Statements and conclusions made in this article are entirely those of the author. They do not necessarily reflect the views of LBBW.

with $B_0(x) = 0$, $B_1(x) = 1$ and

$$C_n(x) = 6xC_{n-1}(x) - C_{n-2}(x), \quad n \geq 2,$$

with $C_0(x) = 1$, $C_1(x) = 3x$.

(Lucas-) Balancing numbers and (Lucas-) balancing polynomials are related by $B_n = B_n(1)$ and $C_n = C_n(1)$. Sequences $(B_n)_{n\geq 0}$ and $(C_n)_{n\geq 0}$ are indexed in On-Line Encyclopedia of Integer Sequences [19] (see entries A001109 and A001541, respectively). The polynomials are interesting also due to their direct connection to Fibonacci numbers, Lucas numbers and Chebyshev and Legendre polynomials [7].

These polynomials have been introduced recently as an extension of the popular balancing and Lucas-balancing numbers B_n and C_n, respectively, as presented by Behera and Panda in [2].

Balancing polynomials (numbers) are members the Lucas sequence of the first kind defined by the recurrence relation $U_0 = 0$, $U_1 = 1$, $U_n = pU_{n-1} + qU_{n-2}$ $(n \geq 2)$. Lucas-balancing polynomials (numbers) can also be defined using initial values $C_0(x) = 2$ and $C_1(x) = 6x$. In this case, Lucas-balancing polynomials will belong to the Lucas sequence of the second kind defined by $V_0 = 2$, $V_1 = p$, $V_n = pV_{n-1} + qV_{n-2}$ $(n \geq 2)$. Such an approach would allow us to simplify some formulas, but would complicate our comparative analysis with articles where these polynomials are defined by initial values $C_0(x) = 1$ and $C_1(x) = 3x$.

Solving the recurrences routinely we get the following closed forms for polynomials $B_n(x)$ and $C_n(x)$ known as Binet formulas:

$$B_n(x) = \frac{\lambda^n(x) - \lambda^{-n}(x)}{\lambda(x) - \lambda^{-1}(x)}, \quad C_n(x) = \frac{\lambda^n(x) + \lambda^{-n}(x)}{2}, \tag{1.1}$$

where $\lambda(x) = 3x + \sqrt{9x^2 - 1}$.

Using (1.1), it is easy to see that

$$B_{2n}(x) = 2B_n(x)C_n(x), \quad n \geq 0. \tag{1.2}$$

Combinatorial expressions for balancing and Lucas-balancing polynomials are [3, 15]

$$B_n(x) = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^k \binom{n-1-k}{k} (6x)^{n-1-2k}, \quad n \geq 1, \tag{1.3}$$

$$C_n(x) = \frac{n}{2} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{n-k} \binom{n-k}{k} (6x)^{n-2k}, \quad n \geq 1. \tag{1.4}$$

The relations $B_n(-x) = (-1)^{n+1}B_n(x)$ and $C_n(-x) = (-1)^n C_n(x)$ follow from $\lambda(\pm x) = -\lambda^{-1}(\mp x)$.

Some examples of recent works involving balancing and Lucas-balancing polynomials conclude [7–9, 16].

The aim of the paper is to derive new combinatorial identities for polynomials $B_n(x)$ and $C_n(x)$. Evaluating these identities at specific points, we can also establish some interesting combinatorial identities as special cases, especially those with Fibonacci and Lucas numbers.

2. Combinatorial identities using Waring's formulas

Our first result provides two combinatorial identities for balancing and Lucas-balancing polynomials involving binomial coefficients.

Theorem 2.1. *Let $m \geq 0$. Then*

$$B_{(n+1)m}(x) = B_m(x) \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \binom{n-k}{k} (2C_m(x))^{n-2k}, \quad n \geq 0, \tag{2.1}$$

$$C_{nm}(x) = \frac{n}{2} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{n-k} \binom{n-k}{k} (2C_m(x))^{n-2k}, \quad n \geq 1. \tag{2.2}$$

Proof. We combine the Binet formulas (1.1) with the following combinatorial formulas

$$\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \binom{n-k}{k} (XY)^k (X+Y)^{n-2k} = \frac{X^{n+1} - Y^{n+1}}{X - Y} \tag{2.3}$$

and

$$\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{n}{n-k} \binom{n-k}{k} (XY)^k (X+Y)^{n-2k} = X^n + Y^n. \tag{2.4}$$

To get (2.1), set $X = \lambda^m(x)$ and $Y = \lambda^{-m}(x)$ in (2.3). Formula (2.1) is the immediate result when replacing n by $n-1$. To get (2.2) apply the same argument to the formula (2.4). □

Remark 2.2. Formulas (2.3) and (2.4) are well-known in combinatorics and called Waring's (sometimes Girard-Waring's) formulas. In [12] the reader will find some interesting remarks about the history and the use of these formulas and their generalizations. The proof of these formulas can be seen, for example, in [4].

In view of (1.2), formulas (2.1) and (2.2) can be written entirely in terms of balancing polynomials $B_n(x)$. Special cases of (2.1) and (2.1) for $m = 1$ are formulas (1.3) and (1.4), respectively.

Setting $x = 1$ in (2.1), we immediately get

$$B_{mn} = B_m \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^k \binom{n-1-k}{k} (2C_m)^{n-1-2k}.$$

This result appears as Theorem 3.2 in [18]. Similarly, setting $x = 1$ in (2.2) yields

$$C_{mn} = \frac{n}{2}\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{n-k}\binom{n-k}{k}(2C_m)^{n-2k}. \tag{2.5}$$

Special cases of (2.5) are

$$C_n = \frac{n}{2}\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{n-k}\binom{n-k}{k}6^{n-2k}, \tag{2.6}$$

$$C_{2n} = \frac{n}{2}\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{n-k}\binom{n-k}{k}34^{n-2k},$$

and so on. Formula (2.6) may be found in [15]. More expressions of this kind can be found in [10].

Next we are going to present some consequences of the above results to combinatorial sums involving Fibonacci numbers F_n and Lucas numbers L_n. Recall that both sequences satisfy the same recurrence relation $u_n = u_{n-1} + u_{n-2}$ for $n \geq 2$, but with initial conditions $F_0 = 0$, $F_1 = 1$ and $L_0 = 2$, $L_1 = 1$ (sequences A000045 and A000032 in [19], respectively).

Balancing and Lucas-balancing polynomials are linked to Fibonacci and Lucas numbers via

$$B_n\left(\frac{L_{2q}}{6}\right) = \frac{F_{2qn}}{F_{2q}}, \quad C_n\left(\frac{L_{2q}}{6}\right) = \frac{L_{2qn}}{2}, \tag{2.7}$$

and

$$B_n\left(\frac{L_{2q+1}}{6}i\right) = \frac{F_{(2q+1)n}}{F_{2q+1}}i^{n-1}, \quad C_n\left(\frac{L_{2q+1}}{6}i\right) = \frac{L_{(2q+1)n}}{2}i^{n}, \tag{2.8}$$

where q is an integer and i is the imaginary unit; see [7].

Formulas (2.7) and (2.8), coupled with Theorem 2.1 above, yield the following results, which are known.

Corollary 2.3. *Let $m \geq 0$. Then*

$$F_{2m(n+1)} = F_{2m}\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor}(-1)^k\binom{n-k}{k}L_{2m}^{n-2k}, \quad n \geq 0, \tag{2.9}$$

$$F_{(2m+1)(n+1)} = F_{2m+1}\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor}\binom{n-k}{k}L_{2m+1}^{n-2k}, \quad n \geq 0, \tag{2.10}$$

$$L_{2mn} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor}(-1)^k\frac{n}{n-k}\binom{n-k}{k}L_{2m}^{n-2k}, \quad n \geq 1, \tag{2.11}$$

$$L_{(2m+1)n} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n}{n-k} \binom{n-k}{k} L_{2m+1}^{n-2k}, \quad n \geq 1. \tag{2.12}$$

The above results are rediscoveries of known identities. Formulas (2.9) and (2.10) we can united as a single formula [13]

$$F_{m(n+1)} = F_m \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^{k(m+1)} \binom{n-k}{k} L_m^{n-2k}, \quad n, m \geq 0. \tag{2.13}$$

Also, formulas (2.11) and (2.12) may be written in the same manner as follows [13]

$$L_{mn} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^{k(m+1)} \frac{n}{n-k} \binom{n-k}{k} L_m^{n-2k}, \quad n \geq 1,\ m \geq 0. \tag{2.14}$$

Since $L_s = F_{2s}/F_s$, formulas (2.13) and (2.14) can be written entirely in terms of Fibonacci numbers.

Specific examples of (2.13) and (2.14) include the following combinatorial Fibonacci and Lucas identities:

$$F_n = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-1-k}{k}, \tag{2.15}$$

$$F_{2n} = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^k \binom{n-1-k}{k} 3^{n-2k-1}, \tag{2.16}$$

$$F_{3n} = 2 \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-1-k}{k} 4^{n-2k-1}, \tag{2.17}$$

$$L_n = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n}{n-k} \binom{n-k}{k},$$

$$L_{2n} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{n}{n-k} \binom{n-k}{k} 3^{n-2k},$$

$$L_{3n} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n}{n-k} \binom{n-k}{k} 4^{n-2k},$$

and so on. All identities in our list are know. For instance, identity (2.15) appears as equation (1) in [11] and again as equation (5.1) in [5]. Identity (2.16) is equation (2) in [11] and stated slightly differently equation (5.10) in [5].

3. Combinatorial identities using Jennings' formulas

Theorem 3.1. *For $m, n \geq 0$, we have*

$$\frac{B_{(2n+1)m}(x)}{2n+1} = \sum_{k=0}^{n} \binom{n+k}{2k} \frac{(36x^2-4)^k}{2k+1} B_m^{2k+1}(x), \tag{3.1}$$

$$\frac{C_{(2n+1)m}(x)}{2n+1} = \sum_{k=0}^{n} (-1)^{n-k} \binom{n+k}{2k} \frac{4^k}{2k+1} C_m^{2k+1}(x). \tag{3.2}$$

Proof. The following identities are from Jennings [14, Lemmas (i) and (ii)]:

$$\sum_{k=0}^{n} \frac{2n+1}{2k+1} \binom{n+k}{2k} \left(\frac{z^2-1}{z}\right)^{2k} = \frac{z^{2(n+1)} - z^{-2n}}{z^2-1}, \tag{3.3}$$

$$\sum_{k=0}^{n} (-1)^{n-k} \frac{2n+1}{2k+1} \binom{n+k}{2k} \left(\frac{z^2+1}{z}\right)^{2k} = \frac{z^{2(n+1)} + z^{-2n}}{z^2+1}. \tag{3.4}$$

To get (3.1), set $z = X/Y$ in (3.3) to derive at

$$\sum_{k=0}^{n} \frac{2n+1}{2k+1} \binom{n+k}{2k} (XY)^{n-k} (X-Y)^{2k+1} = X^{2n+1} - Y^{2n+1}.$$

Now, we can insert $X = \lambda^m(x)$ and $Y = \lambda^{-m}(x)$, and the statement follows. To get (3.2) apply the same argument to formula (3.4). □

Note that identity (3.3) also appears in [1] to prove some Fibonacci identities.

Corollary 3.2. *For $n \geq 0$,*

$$\sum_{k=0}^{n} \binom{n+k}{2k} \frac{(-4)^k}{2k+1} = \frac{(-1)^n}{2n+1}.$$

Proof. Set $x = 0$ in (3.1) and use

$$B_n(0) = \begin{cases} 0, & \text{if } n \text{ is even,} \\ (-1)^{\frac{n-1}{2}}, & \text{if } n \text{ is odd.} \end{cases}$$

□

Corollary 3.3. *For $n, m \geq 0$,*

$$B_{(2n+1)m} = (2n+1) \sum_{k=0}^{n} \binom{n+k}{2k} \frac{32^k}{2k+1} B_m^{2k+1},$$

$$C_{(2n+1)m} = (2n+1) \sum_{k=0}^{n} (-1)^{n-k} \binom{n+k}{2k} \frac{4^k}{2k+1} C_m^{2k+1}.$$

Proof. Set $x = 1$ in (3.1) and (3.2), respectively. □

Corollary 3.4. *For $n, m \geq 0$,*

$$F_{2m(2n+1)} = (2n+1)\sum_{k=0}^{n} \binom{n+k}{2k} \frac{5^k}{2k+1} F_{2m}^{2k+1}, \tag{3.5}$$

$$F_{(2m+1)(2n+1)} = (2n+1)(-1)^n \sum_{k=0}^{n} \binom{n+k}{2k} \frac{(-5)^k}{2k+1} F_{2m+1}^{2k+1}. \tag{3.6}$$

Proof. Insert $x = L_{2q}/6$ and $x = iL_{2q+1}/6$ in (3.1), use (2.7) and (2.8), and simplify using $L_n^2 = 5F_n^2 + (-1)^n 4$. □

Remark 3.5. Equations (3.5) and (3.6) are rediscoveries of Theorem 1 in [14].

4. Combinatorial identities using Toscano's identity

Theorem 4.1. *For $n \geq 1$ and $m \geq 0$, we have the following combinatorial identity:*

$$2^{2n-1} C_m^{2n}(x) = \sum_{k=1}^{n} \binom{2n-k-1}{n-1} 2^k C_m^k(x) C_{mk}(x). \tag{4.1}$$

Proof. Combine the Binet formula for $C_n(x)$ with combinatorial identity

$$\sum_{k=1}^{n} \binom{2n-k-1}{n-1} (X^k + Y^k) \left(\frac{XY}{X+Y}\right)^{n-k} = (X+Y)^n,$$

which have been proved in [20] by Toscano. □

Setting $x = 1$ in (4.1) immediately gives the next relation.

Corollary 4.2. *For $n \geq 1$ and $m \geq 0$,*

$$2^{2n-1} C_m^{2n} = \sum_{k=1}^{n} \binom{2n-k-1}{n-1} 2^k C_m^k C_{mk}.$$

The next two identities are special instances of the previous corollary for $m = 0$ and $m = 1$, respectively:

$$\sum_{k=1}^{n} \binom{2n-k-1}{n-1} 2^k = 2^{2n-1}$$

and

$$2\sum_{k=1}^{n} \binom{2n-k-1}{n-1} 6^k C_k = 36^n.$$

Focusing on Lucas numbers we obtain the following known combinatorial identities [6].

Corollary 4.3. *For $n \geq 1$ and $m \geq 0$, Lucas numbers satisfy*

$$L_{2m}^{2n} = \sum_{k=1}^{n} \binom{2n-k-1}{n-1} L_{2m}^{k} L_{2mk},$$

and

$$L_{2m+1}^{2n} = \sum_{k=1}^{n} (-1)^{n-k} \binom{2n-k-1}{n-1} L_{2m+1}^{k} L_{(2m+1)k}.$$

The next evaluation are consequences of Corollary 4.3:

$$\sum_{k=1}^{n} (-1)^{n-k} \binom{2n-k-1}{n-1} L_k = 1,$$

$$\sum_{k=1}^{n} \binom{2n-k-1}{n-1} \frac{L_{2k}}{3^{2n-k}} = 1,$$

$$\sum_{k=1}^{n} (-1)^{n-k} \binom{2n-k-1}{n-1} \frac{L_{3k}}{4^{2n-k}} = 1,$$

$$\sum_{k=1}^{n} \binom{2n-k-1}{n-1} \frac{L_{4k}}{7^{2n-k}} = 1.$$

References

[1] K. Adegoke: *Fibonacci and Lucas identities the Golden Way*, Preprint, arXiv:1810.12115v1 (2018).

[2] A. Behera, G. K. Panda: *On the square roots of triangular numbers*, Fibonacci Quart. 37.2 (1999), pp. 98–105.

[3] H. Belbachir, T. Komatsu, L. Szalay: *Linear recurrence associated to rays in Pascal's triangle and combinatorial identities*, Math. Slovaca 64.2 (2014), pp. 287–300, doi: 10.2478/s12175-014-0203-0.

[4] L. Comtet: *Advanced Combinatorics: The Art of Finite and Infinite Expansions*, Dordrecht: D. Reidel, 1974.

[5] K. Dilcher: *Hypergeometric functions and Fibonacci numbers*, Fibonacci Quart. 38.4 (2000), pp. 342–363.

[6] P. Filipponi: *Some binomial Fibonacci identities*, Fibonacci Quart. 33.3 (1995), pp. 251–257.

[7] R. Frontczak: *On balancing polynomials*, Appl. Math. Sci. 13.2 (2019), pp. 57–66, doi: https://doi.org/10.12988/ams.2019.812183.

[8] R. Frontczak: *Powers of balancing polynomials and some consequences for Fibonacci sums*, Int. J. Math. Anal. 13.3 (2019), pp. 109–115, doi: https://doi.org/10.12988/ijma.2019.9211.

[9] R. Frontczak: *Relating Fibonacci numbers to Bernoulli numbers via balancing polynomials*, J. Integer Seq. 22 (2019), Article 19.5.3.

[10] R. Frontczak: *Sums of balancing and Lucas-balancing numbers with binomial coefficients*, Int. J. Math. Anal. 12.12 (2018), pp. 585–594,
doi: https://doi.org/10.12988/ijma.2018.81067.

[11] H. W. Gould: *A Fibonacci formula of Lucas and its subsequent manifestations and rediscoveries*, Fibonacci Quart. 15.1 (1977), pp. 25–29.

[12] H. W. Gould: *The Girard-Waring power sums formulas for symmetric functions and Fibonacci sequences*, Fibonacci Quart. 37.2 (1999), pp. 135–139.

[13] V. E. Hoggatt, D. A. Lind: *Composition and Fibonacci numbers*, Fibonacci Quart. 7.3 (1969), pp. 253–266.

[14] D. Jennings: *Some polynomial identities for the Fibonacci and Lucas numbers*, Fibonacci Quart. 31.2 (1993), pp. 134–137.

[15] B. K. Patel, N. Irmak, P. K. Ray: *Incomplete balancing and Lucas-balancing numbers*, Math. Reports 20(70).1 (2018), pp. 59–72.

[16] P. K. Ray: *Balancing polynomials and their derivatives*, Ukrainian Math. J. 69.4 (2017), pp. 646–663,
doi: https://doi.org/10.1007/s11253-017-1386-7.

[17] P. K. Ray: *On the properties of k-balancing numbers*, Ain Shams Eng. J. 9 (2018), pp. 395–402,
doi: https://doi.org/10.1016/j.asej.2016.01.014.

[18] P. K. Ray, S. Patel, M. K. Mandal: *Identities for balancing numbers using generating function and some new congruence relations*, Notes Number Theory Discrete Math. 22.4 (2016), pp. 41–48.

[19] N. J. A. Sloane: *The On-Line Encyclopedia of Integer Sequences*, Published electronically at https://oeis.org.

[20] L. Toscano: *Su due sviluppi della Potenza di un Binomio, q-coefficienti di Eulero*, Boll. S. M. Calabrese 16 (1965), pp. 1–8.

Annales Mathematicae et Informaticae
***52** (2020) pp. 107–115*
DOI: https://doi.org/10.33039/ami.2020.06.001
URL: https://ami.uni-eszterhazy.hu

Markov triples with k-generalized Fibonacci components

Carlos A. Gómez[a*], Jhonny C. Gómez[b], Florian Luca[c†]

[a]Departamento de Matemáticas, Universidad del Valle
carlos.a.gomez@correounivalle.edu.co

[b]Departamento de Matemáticas, Universidad del Valle
jhonny.gomez@correounivalle.edu.co

[c]School of Mathematics, University of the Witwatersrand
Johannesburg, South Africa

Research Group in Algebraic Structures and Applications
King Abdulaziz University
Jeddah, Saudi Arabia

Centro de Ciencias Matemáticas UNAM
Morelia, Mexico
florian.luca@wits.ac.za

Submitted: February 13, 2020
Accepted: June 4, 2020
Published online: June 5, 2020

Abstract

We find all triples (x, y, z) of k-Fibonacci numbers which satisfy the Markov equation $x^2 + y^2 + z^2 = 3xyz$. This paper continues and extends previous work by Luca and Srinivasan [6].

Keywords: Markov equation, Markov triples, k-generalized Fibonacci numbers, k-Fibonacci numbers.

MSC: 11B39, 11D61, 11J86

*The first author was supported in part by Project 71228 (Universidad del Valle).
†The third author was supported in part by grant CPRR160325161141 from the NRF of South Africa and the Focus Area Number Theory grant RTNUM19 from CoEMaSS Wits.

1. Introduction

A positive integer x is known as a Markov number if there are positive integers y, z, such that the triple (x, y, z) satisfies the equation

$$x^2 + y^2 + z^2 = 3xyz. \tag{1.1}$$

Some Markov numbers (sequence A002559 in the OEIS [7]) are

$$1, 2, 5, 13, 29, 34, 89, 169, 194, 233, 433, 610, 985, \ldots.$$

Note that, if (x, y, z) satisfies (1.1), then y and z are also Markov numbers, hence (x, y, z) is called a Markov triple. Clearly, one can permute the order of the three components and assume that $0 < x \leq y \leq z$.

It is known that $(1, F_{2n-1}, F_{2n+1})$ is a Markov triple for all $n \geq 0$, where F_r denotes the rth Fibonacci number. Luca and Srinivasan [6] showed these are the only Markov triples whose components are all Fibonacci numbers.

For $k \geq 2$, let $\{F_r^{(k)}\}_{r \geq -(2-k)}$ denote the k-generalized Fibonacci sequence given by the recurrence

$$F_r^{(k)} = F_{r-1}^{(k)} + \cdots + F_{r-k}^{(k)}, \quad \text{for all} \quad r \geq 2,$$

with $F_j^{(k)} = 0$ for $j = 2 - k, \ldots, 0$ and $F_1^{(k)} = 1$.

We determine all Markov triples of the form $(F_s^{(k)}, F_m^{(k)}, F_n^{(k)})$, where s, m, n are positive integers. That is, we find all the solutions of the Diophantine equation

$$\left(F_s^{(k)}\right)^2 + \left(F_m^{(k)}\right)^2 + \left(F_n^{(k)}\right)^2 = 3F_s^{(k)}F_m^{(k)}F_n^{(k)}. \tag{1.2}$$

By symmetry and since $F_1^{(k)} = F_2^{(k)} = 1$, we assume that $2 \leq s \leq m \leq n$. Many arithmetic properties have recently been studied for the k-generalized Fibonacci sequences. Some Diophantine equations similar to the one discussed in this paper can be found in [1] and [4].

Here is our main result.

Main Theorem. *The only solutinos* (k, s, m, n) *of equation* (1.2) *with* $k \geq 2$ *and* $2 \leq s \leq m \leq n$ *are the trivial solutions* $(k, 2, 2, 2)$ *and* $(k, 2, 2, 3)$ *and the parametric one* $(2, 2, 2l - 1, 2l + 1)$ *for some integer* $l \geq 2$.

In particular, there are no non-trivial Markov triples of k-generalized Fibonacci numbers for any $k \geq 3$.

2. Preliminaries

To start, let us assume that (x, y, z) is a Markov triple with $x \leq y \leq z$. Suppose that $x = y$. Then

$$2x^2 + z^2 = 3x^2z,$$

which implies $(z/x)^2 = 3z - 2 \in \mathbb{Z}$. Therefore, $z = rx$ where r is some positive integer. We thus get

$$2 + r^2 = 3xr. \tag{2.1}$$

Hence, $r|2$, so $r = 1, 2$ and we obtain the triples $(x, y, z) = (1, 1, 1), (1, 1, 2)$.

Suppose next that $y = z$. Then,

$$x^2 + 2z^2 = 3z^2x,$$

which implies $(x/z)^2 = 3x - 2 \in \mathbb{Z}$. Hence, $z \mid x$, but since $x \leq z$, we get $x = y = z$, and again the only possibility is $(x, y, z) = (1, 1, 1)$. The previous observation shows that aside from the triples $(1, 1, 1)$ and $(1, 1, 2)$, each Markov triple consists of different integers. Thus, we obtained for the Diophantine equation (1.2) the trivial solutions (k, s, m, n) given by $(k, 2, 2, 2)$ and $(k, 2, 2, 3)$. From now on, we assume that $1 \leq x < y < z$, so $2 \leq s < m < n$.

We need some facts about k-generalized Fibonacci numbers. For $k \geq 2$ fixed, by [3] we have the following Binet-like formula for the rth k-generalized Fibonacci number

$$F_r^{(k)} = \sum_{i=1}^{k} f_k(\alpha_i)\alpha_i^{r-1}, \tag{2.2}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_k$ are the roots of the characteristic polynomial

$$\Phi_k(x) = x^k - x^{k-1} - \cdots - 1,$$

and

$$f_k(x) := \frac{x - 1}{2 + (k + 1)(x - 2)}.$$

It is known that this polynomial has only one real root larger than 1, let's denote it by $\alpha(= \alpha_1)$. It is in the interval $(2(1 - 2^{-k}), 2)$, see [5, Lemma 2.3] or [8, Lemma 3.6]. The remaining roots $\alpha_2, \ldots, \alpha_k$ are all smaller than 1 in absolute value. Furthermore, powers of α can be used to bound $F_r^{(k)}$ (see [2]) from above and below as in the inequality

$$\alpha^{r-2} < F_r^{(k)} < \alpha^{r-1}, \quad \text{which holds for all} \quad r \geq 1. \tag{2.3}$$

It is known from [4] that the coefficient $f_k(\alpha)$ in the Binet formula (2.2) satisfies the inequalities

$$\frac{1}{2} \leq f_k(\alpha) \leq \frac{3}{4}, \quad \text{for all} \quad k \geq 2. \tag{2.4}$$

It is also known (see [3]) that

$$F_r^{(k)} = f_k(\alpha)\alpha^{r-1} + e_k(r), \quad \text{for all} \quad r \geq 1, \quad \text{with} \quad |e_k(r)| < 1/2, \tag{2.5}$$

and it follows from the recurrence formula that

$$F_r^{(k)} = 2^{r-2} \quad \text{for all} \quad 2 \leq r \leq k + 1. \tag{2.6}$$

Sometimes we write $\alpha(k) := \alpha$ in order to emphasize the dependence of α on k. It is easy to check that $\alpha(k)$ is increasing as a function of k. In particular, the inequality

$$\phi := \frac{1+\sqrt{5}}{2} = \alpha(2) \leq \alpha(k) < \alpha(k+1) < 2 \tag{2.7}$$

holds for all $k \geq 2$

By (1.2) and (2.3), we have the following relations between our variables:

$$\alpha^{2(n-2)} < (F_n^{(k)})^2 < 3F_s^{(k)}F_m^{(k)}F_n^{(k)} < \alpha^{s+m+n}$$

and

$$3\alpha^{s+m+n-6} < 3F_s^{(k)}F_m^{(k)}F_n^{(k)} < (3F_n^{(k)})^2 < 3\alpha^{2(n-1)},$$

which imply $n \leq s+m+3$ and $s+m \leq n+3$, respectively. We record this intermediate result.

Lemma 2.1. *Assume that (k,s,m,n) is a solution of equation (1.2) with $k \geq 2$ and $2 \leq s < m < n$. Then*

$$|n-(s+m)| \leq 3. \tag{2.8}$$

3. The proof of the Main Theorem

To avoid notational clutter, we omit the superscript (k), so we write F_r instead of $F_r^{(k)}$ but understand that we are working with the k-generalized Fibonacci numbers. We use (2.5) to rewrite (1.2), as

$$\begin{aligned} F_s^2 + F_m^2 + f_k^2\alpha^{2(n-1)} + 2e_kf_k\alpha^{n-1} + e_k^2 \\ = 3(f_k\alpha^{s-1} + e_k'')(f_k\alpha^{m-1} + e_k')(f_k\alpha^{n-1} + e_k). \end{aligned} \tag{3.1}$$

Here, for simplicity, we wrote $f_k := f_k(\alpha)$, $e_k := e_k(n)$, $e_k' := e_k(m)$, $e_k'' := e_k(s)$. Therefore, after some calculations, we get

$$|f_k^2\alpha^{2(n-1)} - 3f_k^3\alpha^{s+m+n-3}| \leq |G_1(k,s,m,n,\alpha)| + F_s^2 + F_m^2, \tag{3.2}$$

where $G_1(k,s,m,n,\alpha)$ is the contributions of those terms in the right-hand side expansion of (3.1). Therefore,

$$\begin{aligned} |G_1(k,s,m,n,\alpha)| \leq \frac{27}{32}\alpha^{s+m-2} + \frac{27}{32}\alpha^{s+n-2} + \frac{9}{16}\alpha^{s-1} \\ + \frac{27}{32}\alpha^{m+n-2} + \frac{9}{16}\alpha^{m-1} + \frac{21}{16}\alpha^{n-1} + \frac{5}{8}. \end{aligned}$$

Now, we divide both sides of (3.2) by $3f_k^3\alpha^{s+m+n-3}$. By (2.3) and (2.4), we get

$$|1-(3f_k)^{-1}\alpha^{n-(m+s)+1}| \leq \frac{8}{3}\left(\frac{27\alpha}{32\alpha^n} + \frac{27\alpha}{32\alpha^m} + \frac{9\alpha^2}{16\alpha^{m+n}}\right.$$

$$+\frac{27\alpha}{32\alpha^s}+\frac{9\alpha^2}{16\alpha^{s+n}}+\frac{21\alpha^2}{16\alpha^{s+m}}$$
$$+\frac{5\alpha^3}{8\alpha^{s+m+n}}+\frac{\alpha}{\alpha^{m+n-s}}+\frac{\alpha}{\alpha^{s+n-m}}\Bigg).$$

Since $2 \le s < m < n$, we have $m \ge 3$, $n \ge 4$, $m \ge s+1$ and $n \ge s+2$. Therefore, after some calculations, we arrive at

$$|1-(3f_k)^{-1}\alpha^{n-(m+s)+1}| < \frac{15.2}{\alpha^s}. \tag{3.3}$$

We put $t := n-(m+s)$. By (2.8), we have that $t \in \{\pm 3, \pm 2, \pm 1, 0\}$. We proceed by cases. If $t+1 \le 0$, then

$$\frac{1}{3} \le 1-(3f_k)^{-1}\alpha^{t+1} \le 1-\frac{2^{t+3}}{9},$$

which implies

$$1/3 < |1-(3f_k)^{-1}\alpha^{t+1}|. \tag{3.4}$$

Now, if $t+1 \ge 2$, then $\phi^2 \le \alpha^{t+1} \le 2^{t+1}$. Thus, we obtain

$$1-\frac{2}{3}2^{t+1} \le 1-(3f_k)^{-1}\alpha^{t+1} \le 1-\frac{4}{9}\phi^2.$$

Since $1-4\phi^2/9 < -0.16$ and $1-2^{t+2}/3 < -1.6$, we get

$$0.16 < |1-(3f_k)^{-1}\alpha^{t+1}|. \tag{3.5}$$

Finally, we treat the case $t=0$. Let us consider, for $k \ge 2$, the function

$$g(x,k) = \frac{2x+(k+1)(x^2-2x)}{3(x-1)}.$$

Clearly, for $x > \sqrt{2}$ fixed, the function $g(x,k)$ is increasing as a function of k. On the other hand,

$$\frac{\partial}{\partial x}g(k,x)\Big|_{x=x_k} = 0, \quad \text{where} \quad x_k := \frac{1+k\pm\sqrt{1-k^2}}{k+1}.$$

Assume first that $k \ge 4$ fixed. Then $g(x,k)$ is increasing for $x \in (1,2)$ and $1.93 < \alpha(4) \le \alpha(k)$. Therefore,

$$1.14 < g(1.93,4) \le g(\alpha,k) = (3f_k)^{-1}\alpha.$$

Thus, we conclude that

$$0.14 < |1-(3f_k)^{-1}\alpha|. \tag{3.6}$$

Now, for $k=2$ and $k=3$, we get

$$(3f_2)^{-1}\phi < 0.75 \quad \text{and} \quad (3f_3)^{-1}\alpha(3) < 0.992, \tag{3.7}$$

respectively. By (3.4), (3.5), (3.6) and (3.7), we conclude that the inequality

$$0.008 < |1 - (3f_k)^{-1}\alpha^{n-(m+s)+1}|, \tag{3.8}$$

holds in all the cases when $k \geq 2$ and $|n - (m + s)| \leq 3$. Thus, by the previous estimate (3.8) together with (3.3) and the inequality (2.8), we get

$$2 \leq s \leq 15 \quad \text{and} \quad 1 \leq n - m \leq 18.$$

Now, we rewrite equation (1.2) as

$$\begin{aligned} F_s^2 + f_k^2\alpha^{2(m-1)} + 2e'_k f_k \alpha^{m-1} + (e'_k)^2 + f_k^2\alpha^{2(n-1)} + 2e_k f_k \alpha^{n-1} + e_k^2 \\ = 3F_s(f_k\alpha^{m-1} + e'_k)(f_k\alpha^{n-1} + e_k). \end{aligned} \tag{3.9}$$

After some calculations, we obtain

$$|f_k^2\alpha^{2(n-1)} + f_k^2\alpha^{2(m-1)} - 3F_s f_k^2\alpha^{n+m-2}| \leq |G_2(k, s, m, n, \alpha)| + F_s^2, \tag{3.10}$$

where $G_2(k, s, m, n, \alpha)$ correspond to those terms in the right-hand side expansion of (3.9). Therefore,

$$|G_2(k, s, m, n, \alpha)| \leq \left(\frac{9\alpha^{13}}{8} + \frac{3}{4\alpha}\right)(\alpha^m + \alpha^n) + \frac{3\alpha^{14}}{4} + \frac{1}{2}. \tag{3.11}$$

Now, we divide both sides of (3.10) by $3F_s f_k^2\alpha^{n+m-2}$ and use the previous estimate (3.11) together with the fact that the inequality $F_s^2 < \alpha^{28}$ holds for all $2 \leq s \leq 15$, to get

$$|(3F_s)^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) - 1| < \frac{1.37 \times 10^8}{\alpha^m} \tag{3.12}$$

Let us assume that $k \geq 14$. By (2.6), we have that $F_s = 2^{s-2}$ for $2 \leq s \leq 15$. We now put $t := n - m$ and we study the function

$$h(s, t, x) = \frac{1}{3 \cdot 2^{s-2}}\left(\frac{x^{2t} + 1}{x^t}\right),$$

where $(s, t) \in [2, 15] \times [1, 18]$ and $x \in (\alpha(14), 2)$. Clearly this function is increasing in terms of x, therefore

$$h(s, t, \alpha(14)) \leq (3F_s)^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) \leq h(s, t, 2).$$

We check computationally that

$$h(s, t, 2) < 0.9 \quad \text{and} \quad 1.1 < h(s, t, \alpha(14)),$$

hold in the entire range of our variables $(s, t) \in [2, 15] \times [1, 18] \cap (\mathbb{Z} \times \mathbb{Z})$. Therefore, for $k \geq 14$, $2 \leq s \leq 15$ and $1 \leq n - m \leq 18$, we get

$$0.1 < |(3F_s)^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) - 1|.$$

On the other hand, for $3 \leq k \leq 13$, $2 \leq s \leq 15$ and $1 \leq n-m \leq 18$, we find computationally that

$$0.004 < \min |(3F_s)^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) - 1|. \tag{3.13}$$

Therefore, comparing the above lower bound (3.13) with (3.12), we get that for $k \geq 3$,

$$2 \leq s \leq 15, \quad 3 \leq m \leq 50 \quad \text{and} \quad 4 \leq n \leq 68. \tag{3.14}$$

The remaining case $k = 2$ has already been treated but we can include it in our analysis nevertheless. We start noting that for $3 \leq s \leq 15$ and $1 \leq n-m \leq 18$, we have

$$0.16 < \min |(3F_s)^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) - 1|. \tag{3.15}$$

If $s = 2$, we have that $3^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) = 1$ when $n - m = 2$. Therefore, for $1 \leq n-m \leq 18$ with $n \neq m+2$,

$$0.25 < \min |3^{-1}(\alpha^{n-m} + \alpha^{-(n-m)}) - 1|.$$

Thus, comparing the above lower bound (3.15) with (3.12), for $k = 2$ and $n \neq m+2$, we get,

$$2 \leq s \leq 15, \quad 3 \leq m \leq 42 \quad \text{and} \quad 4 \leq n \leq 50. \tag{3.16}$$

By (3.14) and (3.16), we conclude that:

Lemma 3.1. *If (k, s, m, n) is a solution of equation (1.2) with $2 \leq s < m < n$ and $k \geq 2$, then either $k = 2$ and $n = m + 2$ or $k \geq 3$,*

$$2 \leq s \leq 15, \quad 3 \leq m \leq 50 \quad \textit{and} \quad 4 \leq n \leq 68.$$

Now, we need to bound the variable k. Let us assume first that $k > 67$. Then, by Lemma 3.1, we have

$$n \leq 68 < k + 1.$$

Thus, the formula $F_r = 2^{r-2}$ holds for all three $r \in \{s, m, n\}$. Hence, equation (1.2) may be rewritten as

$$2^{2(s-2)} + 2^{2(m-2)} + 2^{2(n-2)} = 3 \cdot 2^{n+m+s-6}.$$

Dividing both sides of this equality by $2^{2(s-2)}$, we get

$$1 + 2^{2(m-s)} + 2^{2(n-s)} = 3 \cdot 2^{n+m-s-2}. \tag{3.17}$$

Since $m - s \geq 1$ and $n - s \geq 2$, the left-hand side of (3.17) is an odd integer greater than or equal to 21. If $n + m > s + 2$, the right-hand side is an even number, if $n + m < s + 2$ the right-hand side is not an integer and if $n + m = s + 2$ the right-hand side is 3 and none of these situations is possible. Thus, $k \leq 67$.

Assume next that $k = 2$ and $n = m + 2$ for some $m \geq 3$. Recall that the case $k = 2$ was treated in [6], so, the following has already been done and we present it here just to end our analysis. By their Lemma 3.2, we have $s = 2$. Thus,

$$1 + F_m^2 + F_{m+2}^2 = 3F_m F_{m+2}. \tag{3.18}$$

If m is an even number, then one of m or $m + 2$ is a multiple of 4, so one of F_m or F_{m+2} is a multiple of 3, which leads to

$$1 + F_j^2 \equiv 0 \bmod 3,$$

for some $j \in \{m, m + 2\}$, which is not possible. Therefore, $m = 2l - 1$ for some $l \geq 2$. Thus, equation (3.18) may rewritten as

$$1 + F_{2l-1}^2 + F_{2l+1}^2 = 3F_{2l-1}F_{2l+1} \quad \text{for} \quad l \geq 2,$$

which holds since it is equivalent to $1 + F_{2l}^2 = F_{2l-1}F_{2l+1}$, which is a particular case of Cassini's formula.

In summary, we have the following result:

Lemma 3.2. *If (k, s, m, n) is a solution of (1.2) with $2 \leq s < m < n$, then either $k = 2$, $s = 2$, $m = 2l - 1$ and $n = 2l + 1$ for some $l \geq 2$ or*

$$2 \leq k \leq 67, \quad 2 \leq s \leq 15, \quad 3 \leq m \leq 50 \quad \textit{and} \quad 4 \leq n \leq 68.$$

Finally, a brute force search for solutions (k, s, m, n) of the equation (1.2), using the respective range given by the previous lemma, finishes the proof of our Main Theorem. Here, we used

$$F[r\_, k\_] := \text{SeriesCoefficient}[\text{Series}[x/(1 - \text{Sum}[x^j, \{j, 1, k\}]), \{x, 0, 1400\}], r],$$

to create the rth k-Fibonacci number.

Acknowledgements. We thank the referee for the valuable comments. J. C. G. also thanks the Universidad del Valle for its support during his Ph.D. studies. Part of this work was done while F. L. was visiting the Max Planck Institute for Mathematics in Bonn, Germany. This author thanks the people of that Institute for their hospitality and support.

References

[1] J. J. Bravo, C. A. Gómez, F. Luca: *Power of two as sums of two k-Fibonacci numbers*, Miskolc Math. Notes 17 (2016), pp. 85–100, doi: https://doi.org/10.18514/MMN.2016.1505.

[2] J. J. Bravo, F. Luca: *Powers of two in generalized Fibonacci sequences*, Rev. Colombiana Mat. 46.1 (2012), pp. 67–79.

[3] G. P. B. Dresden, Z. Du: *A simplified Binet formula for k-generalized Fibonacci numbers*, J. Integer Sequences 17 (2014), Article 14.4.7.

[4] C. A. Gómez, F. Luca: *An exponential Diophantine equation related to the sum of powers of two consecutive k-generalized Fibonacci numbers*, Colloquium Mathematicum 137.2 (2014), pp. 171–188,
doi: https://doi.org/10.4064/cm137-2-3.

[5] L. K. Hua, Y. Wang: *Applications of number theory to numerical analysis*, Translated from Chinese. Springer-Verlag, Berlin-New York; Kexue Chubanshe (Science Press), Beijing (1981).

[6] F. Luca, A. Srinivasan: *Markov equation with Fibonacci components*, Fibonacci Quart. 56.2 (2018), pp. 126–169.

[7] N. J. A. Sloane: *The OnLine Encyclopedia of Integer Sequences*, http://oeis.org.

[8] D. A. Wolfram: *Solving generalized Fibonacci recurrences*, Fibonacci Quart. 36.2 (1998), pp. 129–145.

Annales Mathematicae et Informaticae
***52** (2020) pp. 117–135*
DOI: https://doi.org/10.33039/ami.2020.10.004
URL: https://ami.uni-eszterhazy.hu

Thickness distribution of Boolean functions in 4 and 5 variables and a comparison with other cryptographic properties

Mathias Hopp[a*], Pål Ellingsen[a], Constanza Riera[a], Pantelimon Stănică[b†]

[a]Department of Computer Science,
Electrical Engineering and Mathematical Sciences,
Western Norway University of Applied Sciences, 5020 Bergen, Norway
mathias.hopp@spv.no,{pel,csr}@hvl.no

[b]Applied Mathematics Department,
Naval Postgraduate School, Monterey, USA
pstanica@nps.edu

Submitted: September 23, 2020
Accepted: October 20, 2020
Published online: October 29, 2020

Abstract

This paper explores the distribution of algebraic thickness of Boolean functions (that is, the minimum number of terms in the ANF of the functions in the orbit of a Boolean function, through all affine transformations), in four and five variables, and the complete distribution is presented. Additionally, a complete analysis of some complexity properties (e.g., nonlinearity, balancedness, etc.) of all relevant orbits of Boolean functions is presented. Some properties of our notion of rigid function (which enabled us to reduce significantly the computation) are shown and some open questions are proposed, providing some further explanation of one of these questions.

Keywords: Boolean function, algebraic normal form, thickness, nonlinearity, affine equivalence

MSC: 06E30, 11T06, 94A60, 94D10.

*Currently with Sparebanken Vest, Bergen, Norway
†Corresponding author

1. Introduction

In this paper, we deal with the concept of algebraic thickness, defined by Carlet in [3, 4] as the minimum number of terms of all Boolean functions in the affine equivalence orbit of a Boolean function – and aim to reveal the distribution of algebraic thickness of all Boolean functions in four and five variables.

As will be discussed in the coming sections, by using an exhaustive search, the calculation of this distribution for $n \leq 4$ variables is at best a straightforward, and at worst, a lengthy – but manageable – endeavor. There are 2^{2^n} Boolean functions in n variables, which, for $n = 4$, equals 65536. Since there are 322560 different affine transformations needed to be checked for *each* Boolean function, the calculation of the algebraic thickness for all Boolean functions in four variables is a time consuming task, albeit doable.

However, in moving from four to five variables, this number grows significantly. The total number of unique Boolean functions is 4 294 967 296, and the number of different affine transformations is 319 979 520. One of the sub-goals of the paper was to find an efficient method able to handle the magnitude of the computation, and another was to effectively handle and analyze the resulting data set for $n = 5$.

Additionally, throughout the paper, when discussing functions $n \leq 5$, we omit the trivial cases $n = 0, 1$, unless specified. We used SageMath [9] for all computations in this paper.

A Boolean function f in n variables, where n is any positive integer, is a function from the vector space $\mathbb{F}_2^n$ to the finite field $\mathbb{F}_2$, i.e. $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$. The set of all Boolean functions in n variables is denoted by $\mathcal{B}_n$, and the symbol $\oplus$ denotes addition modulo 2, in $\mathbb{F}_2$, $\mathbb{F}_2^n$, and $\mathcal{B}_n$.

Every Boolean function f has a unique representation called its *algebraic normal form* (ANF) as a polynomial over $\mathbb{F}_2$ in n variables:

$$f(\mathbf{x}) = \bigoplus_{\mathbf{u}\in\mathbb{F}_2^n} c_{\mathbf{u}} \left(\prod_{i=1}^{n} x_i^{u_i} \right) = \bigoplus_{\mathbf{u}\in\mathbb{F}_2^n} c_{\mathbf{u}} \mathbf{x}^{\mathbf{u}},$$

where each $c_{\mathbf{u}} \in \mathbb{F}_2$, $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{x} = (x_1, \ldots, x_n)$. The algebraic *degree* of f is the largest weight of $\mathbf{u}$ such that $c_{\mathbf{u}} \neq 0$. A *homogeneous* function is a sum of monomials of the same degree.

An *affine function* $\ell_{\mathbf{u},c}$ is a function with algebraic degree at most 1, which takes the form

$$\ell_{\mathbf{u},c}(\mathbf{x}) = \mathbf{u} \cdot \mathbf{x} \oplus c = u_1 x_1 \oplus \cdots \oplus u_n x_n \oplus c, \tag{1.1}$$

where $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ and $c \in \mathbb{F}_2$. If $c = 0$, such that $\ell_{\mathbf{u},0}$ only consists of monomials of algebraic degree 1, and no constant, then it is a *linear* function. The *Hamming weight* of a vector $\mathbf{x} \in \mathbb{F}_2^n$ is denoted by $wt(\mathbf{x})$ and is equal to the number of 1's in the vector $\mathbf{x}$. For a Boolean function f on $\mathbb{F}_2^n$, let $\Omega_f = \{\mathbf{x} \in \mathbb{F}_2^n \mid f(\mathbf{x}) = 1\}$ be the *support* of f. The Hamming weight of f is then $|\Omega_f|$, or equivalently, the weight of the vector of its truth table. The *Hamming distance* between two

functions $f, g\colon \mathbb{F}_2^n \to \mathbb{F}_2$, denoted by $d(f,g)$, is defined as $d(f,g) = wt(f \oplus g)$. A *balanced* function on n variables has weight exactly 2^{n-1}. For $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ we define the *Walsh-Hadamard transform* to be the integer-valued function

$$\mathcal{W}_f(\mathbf{u}) = \sum_{x \in \mathbb{F}_2{}^n} (-1)^{f(\mathbf{x})+\mathbf{ux}},\ \mathbf{u} \in \mathbb{F}_2^n.$$

The *nonlinearity* $\mathcal{N}_f$ of a function f is defined as

$$\mathcal{N}_f = \min_{\phi \in \mathcal{A}_n} d(f, \phi)$$

where $\mathcal{A}_n$ is the class of all affine functions on $\mathbb{F}_2^n$. The largest nonlinearity, namely $2^{n-1} - 2^{\frac{n}{2}-1}$ is achieved by *bent* functions (they exist for even dimension n) and they have only two values in their Walsh spectrum (the multiset of Walsh coefficients), namely $\pm 2^{\frac{n}{2}}$. The *semi-bent* functions will have three values in their Walsh spectrum, namely, $\{0, \pm 2^{\frac{n+2}{2}}\}$, $\{0, \pm 2^{\frac{n+1}{2}}\}$, for n even, respectively, odd, and they can be balanced, as opposed to bent functions, whose weight can only be $2^{n-1} \pm 2^{\frac{n}{2}-1}$.

For these definitions and to know more on Boolean functions, and their cryptographic properties, the reader can consult [2, 5].

2. Algebraic thickness

Carlet, in [3], defined algebraic thickness, and discussed lower and upper bounds. His paper also includes further discussion on the relation that algebraic thickness has with other complexity criteria (e.g., nonlinearity, algebraic degree, etc.). In [4], Carlet improved some of the prior results, and further expanded on the properties of algebraic thickness.

Definition 2.1 ([4])**.** *The* algebraic thickness $\mathcal{T}(f)$ *of a Boolean function f is the minimum number of monomials with non-zero coefficients in the ANF of the functions $f \circ \mathcal{A}$, where $\mathcal{A} \in \mathrm{GL}(n, \mathbb{F}_2)$ (the general affine group). When we want to emphasize the number of variables, we shall write $\mathcal{T}_n(f)$.*

Surely, the algebraic thickness of affine functions is at most 1 [1, 4]. The quadratic functions are also well understood, due to the well-known Dickson's theorem (see MacWilliams and Sloane [8], or the simpler version below taken from Boyar and Find [1]).

Theorem 2.2 (Dickson's Theorem)**.** *If $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ is a quadratic Boolean function, then there exist an invertible $n \times n$ matrix A, $\mathbf{b} \in \mathbb{F}_2^n$, $t \leq \frac{n}{2}$, and $c \in \mathbb{F}_2$ such that for $\mathbf{y} = A\mathbf{x} + \mathbf{b}$ one of the following two equations holds:*

$$f(x) = y_1y_2 + y_3y_4 + \cdots + y_{t-1}y_t + c, \text{ or}$$
$$f(x) = y_1y_2 + y_3y_4 + \cdots + y_{t-1}y_t + y_{t+1}.$$

Furthermore A, $\mathbf{b}$, and c can be found efficiently.

We also mention that we re-computed (see Table 4) the distribution of nonlinearities of all functions in $2 \leq n \leq 5$ variables, confirming known results (see, for instance, the paper by Sertkaya and Doğanaksoy [10]).

For Boolean functions in n variables, it is of interest to determine the maximum value possible for the thickness, namely, $\tau_n = \max_{f \in \mathcal{B}_n}(\mathcal{T}(f))$, and specifically, its growth. Surely, we have the trivial upper bound $\tau_n \leq 2^n$, since the maximum number of terms in the ANF of a function in n variables is $\leq 2^n$.

Regarding the lower bound of the thickness, Carlet showed in [3] that, for every $\lambda < \frac{1}{2}$ and positive integer n, the density in $\mathcal{B}_n$ of the subset

$$\{f \in \mathcal{B}_n \,|\, \mathcal{T}(f) \geq \lambda 2^n\}$$

is greater than $1 - 2^{2^n H_2(\lambda) - 2^n + n^2 + n}$, where $H_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the *entropy function*, and therefore *almost* all Boolean functions have algebraic thickness greater than $\lambda 2^n$. This was improved in [4], showing that *almost* all Boolean functions have algebraic thickness greater than $2^{n-1} - n2^{\frac{n-1}{2}}$. The best upper bound on algebraic thickness is still the one in [3], namely,

$$\mathcal{T}(f) \leq \frac{2}{3}2^n,$$

which is believed to be improvable.

3. Some theoretical results on thickness

Brute force computation is still possible for $n = 4$, but for $n = 5$ we need to find some techniques to reduce the computational time, as it would take thousands of years on a personal computer. The idea is that this new technique may be useful in approaching the thickness distribution computation for $n = 6$ (or at least for some subclass of $\mathcal{B}_6$).

For any Boolean function f, we define its *orbit* or *equivalence class* as the set of functions $\{f \circ \mathcal{A} : \mathcal{A} \in \mathrm{GL}(n, \mathbb{F}_2)\}$.

Given a Boolean function f, if f_{min} is an element (not necessarily unique) of its equivalence class with minimum number of terms, then the algebraic thickness of f_{min} is the number of terms in its ANF.

Definition 3.1 (Rigid Boolean functions)**.** *We call a Boolean function f with $\mathcal{T}(f)$ monomials in its ANF, a* rigid function*. The set of all rigid functions will be denoted by $\mathcal{S}_n$.*

Thus, a rigid Boolean function cannot be mapped to a function with lower monomial count, through any affine transformation. Furthermore, any Boolean function can be mapped to a rigid function. The reason for this should be clear, but for completion, we state it as a lemma.

Proposition 3.2. *Any Boolean function can be mapped to a rigid function, by an affine transformation.*

Proof. Given a Boolean function $f \in \mathcal{B}_n$, let g be a function in the orbit of f (through affine transformations), where the monomial count of g is equal to $\mathcal{T}(f)$. If g is not a rigid function, then g does not have the minimum monomial count in its orbit. Suppose h is in the orbit of g, and has lower monomial count than g. Since f maps to g and g maps to h, then by composition of transformations, f maps to h as well. Thus, we reach a contradiction. □

Experimentally, it was found that $\mathcal{S}_n \subset \mathcal{S}_{n+1}$, for small values of n, suggesting that perhaps this is true in general, and will be shown next.

Theorem 3.3. *All rigid functions in n variables are also rigid functions in $(n+1)$ variables, that is, $\mathcal{S}_n \subset \mathcal{S}_{n+1}$.*

Remark 3.4. As is customary in this area (for easy writing), in the following proof, we disregard the usual linear algebra convention of matrix-vector multiplication and regard $\mathbf{x}$ and $\mathbf{b}$ both as a row- and a column vector, when there is no danger of confusion.

Proof. Let $f \in \mathcal{S}_n$ with $\mathcal{T}(f) = t$. We embed f in $n+1$ variables, and we denote its embedding by $\tilde{f}$, such that $\tilde{f}(x_1, \ldots, x_n, x_{n+1}) = f(x_1, \ldots, x_n)$. Let a non-zero affine transformation of the input of $\tilde{f}$ be given by $\mathbf{x} \mapsto \tilde{A}\tilde{\mathbf{x}} + \mathbf{b}$, where $\tilde{A}$ is an $(n+1) \times (n+1)$ matrix and $\mathbf{b} = (b_1 \ldots, b_n)$, and $\tilde{\mathbf{x}} = (x_1, \ldots, x_n, x_{n+1}), \mathbf{x} = (x_1, \ldots, x_n)$. We label the first n rows and n columns in $\tilde{A}$ by A and so,

$$\tilde{A} = \begin{pmatrix} & & a_{1,n+1} \\ & A & \vdots \\ a_{n+1,1} & \cdots & a_{n+1,n+1} \end{pmatrix}.$$

Thus,

$$\tilde{A}\tilde{\mathbf{x}} + \mathbf{b} = \begin{pmatrix} A\mathbf{x} + x_{n+1} \begin{pmatrix} a_{1,n+1} \\ \vdots \\ a_{n,n+1} \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \\ a_{n+1,1}x_1 + \cdots + a_{n+1,n+1}x_{n+1} + b_{n+1} \end{pmatrix},$$

and so

$$\tilde{f}(\tilde{A}\tilde{\mathbf{x}} + \mathbf{b}) = f\left(A\mathbf{x} + x_{n+1} \begin{pmatrix} a_{1,n+1} \\ \vdots \\ a_{n,n+1} \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \right),$$

from which our claim is inferred. □

In summary, the introduction of x_{n+1} does not induce any further monomial eliminations not already possible in n variables. Therefore, for a rigid function f with monomial count t,

$$\mathcal{T}_n(f) = t = \mathcal{T}_{n+1}(\tilde{f}).$$

Corollary 3.5. *For any Boolean function f in n variables,*

$$\mathcal{T}_n(f) = \mathcal{T}_{n+1}(\tilde{f}),$$

where $\tilde{f}$ is the embedding of f in $\mathcal{B}_{n+1}$, such that

$$\tilde{f}(x_1, \ldots, x_n, x_{n+1}) = f(x_1, \ldots, x_n).$$

Proof. Given a rigid Boolean function f in n variables, let $\mathcal{A}_n(f)$ be the orbit of f through all nonzero affine transformations, and let $\mathcal{T}_n(f) = t$. As we know, from the definition of algebraic thickness, any Boolean function $g \in \mathcal{A}_n(f)$ satisfies $\mathcal{T}_n(g) = t$, as well. Since f is rigid, $\mathcal{T}_{n+1}(f) = t$, by Theorem 3.3. Clearly, then, $\mathcal{A}_n(f) \subseteq \mathcal{A}_{n+1}(\tilde{f})$, by the very same affine transformations as in n variables (leaving the new variable x_{n+1} mapped to itself), and therefore all functions in $\mathcal{A}_n(f)$ have thickness t in $n+1$ variables, as well. □

3.1. Multiplication by a new variable may conserve thickness

We showed in Theorem 3.3 that all rigid functions in $\mathcal{B}_n$ are also rigid functions in $\mathcal{B}_{n+1}$. Moreover, $f \in \mathcal{B}_n$, $\mathcal{T}_n(f) = \mathcal{T}_{n+1}(\tilde{f})$, as well. These properties give insight into the distribution of algebraic thickness in $(n+1)$ variables, when the distribution for n variables is known. Surely, we cannot expect an inductive procedure for the computation of thickness, but as observed already in Theorem 3.3, a connection does exist that may decrease the complexity even further.

Proposition 3.6. *Let $f \in \mathcal{B}_n$ be a Boolean function in variables $\mathbf{x} = (x_1, ..., x_n)$ vector, and let x_{n+1} be a new variable. Then*:

$$\mathcal{T}_{n+1}(f(x_1, \ldots, x_n) \cdot x_{n+1}) \leq \mathcal{T}_n(f).$$

Proof. Given a Boolean function $f \in \mathcal{B}_n$, with known algebraic thickness $\mathcal{T}_n(f) = t$, on the variables $(x_1, \ldots, x_n)$, we let $f_{\min} \in \mathcal{B}_n$ be the representative function with monomial count t of the orbit of f, and let π denote the affine transformation such that $\pi(f) = f_{\min}$. As before, x_{n+1} is the new variable introduced in $\mathcal{B}_{n+1}$.

In $\mathcal{B}_{n+1}$, then, $\pi'(f(x_1, \ldots, x_n) \cdot x_{n+1}) = f_{\min}(x_1, \ldots, x_n) \cdot x_{n+1}$, by the transformation $\pi'(x_j) = \pi(x_j)$, for $j < (n+1)$, and $\pi'(x_{n+1}) = x_{n+1}$. Since $f_{\min}$ has monomial count t, $f_{\min}(x_1, \ldots, x_n) \cdot x_{n+1}$ also has monomial count t, and therefore $\mathcal{T}_{n+1}(f(x_1, \ldots, x_n) \cdot x_{n+1}) \leq \mathcal{T}_n(f)$. □

Based upon extensive computations (exhaustive for lower dimensions and random for higher dimensions) and the previous proposition, we propose the following question.

Open question 3.7 (Thickness conservation)**.** *Let $f \in \mathcal{B}_n$ be a Boolean function in variables $\mathbf{x} = (x_1, ..., x_n)$ vector, and let x_{n+1} be a new variable. Is it true that*

$$\mathcal{T}_{n+1}(f(x_1, \ldots, x_n) \cdot x_{n+1}) = \mathcal{T}_n(f)?$$

While this is not necessarily the goal of the paper, and we cannot provide an answer to this question, we attempt to explain it further. Assume that there exists a function f in n variables such that $\mathcal{T}_n(f) > \mathcal{T}_{n+1}(f \cdot x_{n+1}) = t$. Take an affine transformation that brings $f(x_1, \ldots, x_n) \cdot x_{n+1}$ to its minimal thickness form, transformation determined by the vector $\mathbf{b} = (b_1, \ldots, b_{n+1})$, and the matrix $\tilde{A}$ of the form

$$\tilde{A} = \begin{pmatrix} & & a_{1,n+1} \\ & A & \vdots \\ a_{n+1,1} & \cdots & a_{n+1,n+1} \end{pmatrix},$$

where A is an $n \times n$ matrix, and so,

$$\tilde{A}\tilde{\mathbf{x}} + \mathbf{b} = \begin{pmatrix} A\mathbf{x} + x_{n+1} \begin{pmatrix} a_{1,n+1} \\ \vdots \\ a_{n,n+1} \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \\ a_{n+1,1}x_1 + \cdots + a_{n+1,n+1}x_{n+1} + b_{n+1} \end{pmatrix},$$

as in Theorem 3.3. We label $r_{i,\tilde{A}}$, $r_{i,A}$, the ith row of $\tilde{A}$, respectively A, and $\tilde{\mathbf{x}} = (x_1, \ldots, x_n, x_{n+1}), \mathbf{x} = (x_1, \ldots, x_n)$. Thus, using "$\cdot$" to denote the usual scalar product,

$$\begin{aligned} &(f(x_1, \ldots, x_n) \cdot x_{n+1}) \circ (\tilde{A}\tilde{\mathbf{x}} + \mathbf{b}) \\ &= f(r_{1,A} \cdot \mathbf{x} + a_{1,n+1}x_{n+1} + b_1, \ldots, r_{n,A} \cdot \mathbf{x} + a_{n,n+1}x_{n+1} + b_n) \\ &\qquad \cdot (a_{n+1,1}x_1 + \cdots + a_{n+1,n+1}x_{n+1} + b_{n+1}). \end{aligned} \tag{3.1}$$

We let $b'_i = b_i + a_{i,n+1}x_{n+1}$, $1 \le i \le n+1$ and $\mathbf{b}' = (b'_1, \ldots, b'_n)$. Since the first factor is simply $f(A\mathbf{x} + \mathbf{b}')$ (we regard its coefficients in $\mathbb{F}_2[x_{n+1}]$, and assume that A is invertible; again, it may happen that it is not), it must have more than $\mathcal{T}_{n+1}(f(x_1, \ldots, x_n) \cdot x_{n+1}) = t$ terms (call them $T_i(x_1, \ldots, x_n)$, of degrees $\deg T_i = d_i$, $1 \le i \le s$, with $d_1 \le d_2 \le \cdots \le d_s$), given our assumption. We thus write its algebraic normal form as

$$\begin{aligned} f(A\mathbf{x} + \mathbf{b}') = (\alpha_1 x_{n+1} + \beta_1)T_1(x_1, \ldots, x_n) + \cdots \\ + (\alpha_s x_{n+1} + \beta_s)T_s(x_1, \ldots, x_n), s > t, \end{aligned}$$

(α_i, β_i are not zero simultaneously, since we need to have $s > t$ terms in $f(A\mathbf{x}+\mathbf{b}')$), and therefore Equation (3.1) becomes (for easy writing, we denote the $(n+1)$st row of $\tilde{A}$ by $(\gamma_1, \ldots, \gamma_{n+1})$ and we will not write the input $(x_1, \ldots, x_n)$ for T_i),

$$\begin{aligned} &\sum_{i=1}^{s} (\alpha_i x_{n+1} + \beta_i) T_i \left(\sum_{j=1}^{n} \gamma_j x_j + \gamma_{n+1}x_{n+1} + b_{n+1} \right) \\ &= \sum_{j=1}^{n} \sum_{i=1}^{s} (\alpha_i x_{n+1} + \beta_i)\gamma_j x_j T_i(x_1, \ldots, x_n) \end{aligned}$$

$$
\begin{aligned}
&+\sum_{i=1}^{s}\gamma_{n+1}(\alpha_i+\beta_i)x_{n+1}T_i+\sum_{i=1}^{s}(\alpha_i x_{n+1}+\beta_i)b_{n+1}T_i\\
&=\sum_{i=1}^{s}x_{n+1}T_i\left(\alpha_i b_{n+1}+\gamma_{n+1}(\alpha_i+\beta_i)+\alpha_i\sum_{j=1}^{n}\gamma_j x_j\right)\\
&\quad+\sum_{j=1}^{n}\sum_{i=1}^{s}\beta_i\gamma_j x_j T_i+\sum_{i=1}^{s}\beta_i b_{n+1}T_i.\\
&=\sum_{i=1}^{s}\left(\alpha_i b_{n+1}+\gamma_{n+1}(\alpha_i+\beta_i)+\alpha_i\sum_{j=1}^{n}\gamma_j x_j\right)x_{n+1}T_i\\
&\quad+\sum_{i=1}^{s}\beta_i\left(b_{n+1}+\sum_{j=1}^{n}\gamma_j x_j\right)T_i.
\end{aligned}
\tag{3.2}
$$

We thus get

$$
\begin{aligned}
&(f(x_1,\dots,x_n)\cdot x_{n+1})\circ(\tilde{A}\tilde{\mathbf{x}}+\mathbf{b})\\
&=x_{n+1}\sum_{i=1}^{s}\left(\alpha_i b_{n+1}+\gamma_{n+1}(\alpha_i+\beta_i)+\alpha_i\sum_{j=1}^{n}\gamma_j x_j\right)T_i\\
&\quad+\sum_{i=1}^{s}\beta_i\left(b_{n+1}+\sum_{j=1}^{n}\gamma_j x_j\right)T_i.
\end{aligned}
$$

For the inequality to hold, we need to have enough cancellations in both sums

$$
\begin{aligned}
S_1&=\sum_{i=1}^{s}\left(\alpha_i b_{n+1}+\gamma_{n+1}(\alpha_i+\beta_i)+\alpha_i\sum_{j=1}^{n}\gamma_j x_j\right)T_i\\
S_2&=\sum_{i=1}^{s}\beta_i\left(b_{n+1}+\sum_{j=1}^{n}\gamma_j x_j\right)T_i,
\end{aligned}
$$

for a total of more than $(s-t)$ terms. We let A_i be the index support for T_i (that is, if $T_i(x_1,\dots,x_n)=x_{i_1}\cdots x_{i_\ell}$, then $A_i=\{i_1,\dots,i_\ell\}$). Therefore, the above sums can be written as (we let $|J|_2=|J| \pmod 2$, where $J=\{j|\gamma_j\neq 0\}$, and $|J_i|_2=|J_i| \pmod 2$, where $J_i=\{j\in A_i|\gamma_j\neq 0\}$),

$$
\begin{aligned}
S_1&=\sum_{i=1}^{s}(\alpha_i b_{n+1}+\gamma_{n+1}(\alpha_i+\beta_i))T_i+\sum_{i=1}^{s}\alpha_i\left(|J_i|_2+\sum_{j\in J\setminus J_i}x_j\right)T_i\\
&=\sum_{i=1}^{s}\left(\alpha_i b_{n+1}+\gamma_{n+1}(\alpha_i+\beta_i)+\alpha_i|J_i|_2\right)T_i+\sum_{i=1}^{s}\alpha_i T_i\sum_{j\in J\setminus J_i}x_j
\end{aligned}
$$

$$= \sum_{i=1}^{s} \alpha_i \left(b_{n+1} + |J_i|_2\right) T_i + \sum_{i=1}^{s} \alpha_i T_i \sum_{j \in J \setminus J_i} x_j + \gamma_{n+1} \sum_{i=1}^{s} (\alpha_i + \beta_i) T_i,$$

$$S_2 = \sum_{i=1}^{s} \beta_i b_{n+1} T_i + \sum_{i=1}^{s} \beta_i \left(\sum_{j \in A_i} \gamma_j + \sum_{j \notin A_i} \gamma_j x_j \right) T_i$$

$$= \sum_{i=1}^{s} \beta_i \left(b_{n+1} + |J_i|_2\right) T_i + \sum_{i=1}^{s} \beta_i T_i \sum_{j \in J \setminus J_i} x_j.$$

If $\gamma_{n+1} = 0$ then, for i such that $b_{n+1} + |J_i|_2 \neq 0$, then either $\alpha_i \left(b_{n+1} + |J_i|_2\right) T_i$, or $\beta_i \left(b_{n+1} + |J_i|_2\right) T_i$ survives. Similarly, assuming that for an i, $J \setminus J_i \neq \emptyset$, then either $\alpha_i T_i \sum_{j \in J \setminus J_i} x_j$, or $\beta_i T_i \sum_{j \in J \setminus J_i} x_j$ survives. If it were true that for all i, $J \setminus J_i \neq \emptyset$, then the inequality would be false and the conjecture would "hold" in this case. However, at least $J \setminus J_s = \emptyset$, since otherwise our affinely equivalent function would have degree higher than $d_s + 2$ (recall that S_1 is multiplied by x_{n+1}), and that is impossible. If one would attempt to find a counterexample for a negative answer to our open question, then one could take a matrix $\tilde{A}$ where the last row is rather very sparse, along with b such that $A\mathbf{x} + b'$ has most of the $\beta_i = 0$. Can that be achieved? We do not know the answer to this question.

3.2. Gaps in thickness distribution

Noting the algebraic thickness distributions listed in Table 3, it is easy to see that, for $n \leq 5$ and $m > 0$, if there exists a representative with $\mathcal{T}_n = m$, then there exists a representative with $\mathcal{T}_n = m - 1$, and conversely: if there are no representatives with $\mathcal{T}_n = m - 1$, then there are no representatives with $\mathcal{T}_n = m$. The following conjecture is an extension of Lemma 3.9.

Conjecture 3.8. *For any n, in any given monomial count $m \leq 2^n$, if there are no rigid functions with m monomials, then for any $f \in \mathcal{B}_n$,*

$$\mathcal{T}_n(f) < m.$$

The idea here is that if there are no rigid functions in a set monomial count m, then there are no rigid functions in any monomial count M, where $M > m$. Proving this would have implications for further attempts at determining maximum algebraic thickness (and the following thickness distribution) using the methods described in this paper, as finding no rigid functions in n variables with monomial count (e.g.) 2^{n-1} would imply there are no rigid functions with monomial count greater than 2^{n-1}, thus eliminating half of the set of functions to search through.

The definition for rigid functions is closely related to Carlet's definition for algebraic thickness. We record that below.

Proposition 3.9. *Given all Boolean functions in n variables with monomial count k in their ANF, if there are no rigid functions with k monomials, then there are no functions f in n variables with $\mathcal{T}_n(f) = k$.*

This simple proposition was the inception of the program described later to find the thickness distribution.

n	Number of rigid functions
0	2
1	3
2	6
3	28
4	588
5	211 259

Table 1: Number of rigid functions in $n \leq 5$ variables

The distribution of the number of rigid functions in $n \leq 5$ variables is listed in Table 1, where: for $n \leq 4$ variables, these numbers were collected from analysis of the data sets calculated by brute-force, and for $n = 5$, the number was (along with double-checking values for $n < 5$) collected from analysis of the data sets calculated by the program described later.

We hope that our methods will prove useful for $n > 5$, as well, since an iterative approach is impossible by modern computing standards for these dimensions. Searching for rigid functions and – most importantly – disregarding non-rigid functions, should improve the efficiency of any program (at the very least, it improves the program given later).

Determining which functions are rigid functions in n variables yields information regarding the thickness distribution in $n + 1$ variables as well, by Theorem 3.3. Furthermore, by Corollary 3.5, unveiling the distribution of all functions in $\mathcal{B}_n$ immediately gives the distribution of 2^{2^n} functions in $\mathcal{B}_{n+1}$ – which may be a small portion compared to $2^{2^{n+1}}$, but is nonetheless a start.

The functions in $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2$ (i.e., the rigid functions in $n = 0, 1, 2$ variables) are listed below:

$$\begin{aligned}
\mathcal{S}_0 &= \{0,\, 1\} \\
\mathcal{S}_1 &= \{0,\, 1,\, x_1\} \\
\mathcal{S}_2 &= \{0,\, 1,\, x_1,\, x_2,\, x_1x_2,\, x_1x_2 + 1\}
\end{aligned}$$

Since the sets $\mathcal{S}_3, \mathcal{S}_4$ are of rather large sizes (28 and 588, respectively), they will not be listed here (but the data can be found in [7]).

4. Representatives

By uncovering one function ϕ for each of these orbits, every function in n variables can be generated from a corresponding ϕ, by iteration through all affine transformations for each one. Calculating the algebraic thickness of each ϕ yields the thickness distribution for all functions in $\mathcal{B}_n$, as $\mathcal{T}$ is (trivially) an affine invariant. Since these ϕ would be representing their orbits, the name representative function

n	Number of equivalence classes
1	3
2	5
3	10
4	32
5	382
6	15 768 919

Table 2: Number of affine equivalence classes of Boolean functions [6]

was chosen. As the rigid functions are the functions with the minimum number of monomials in their ANF, these representative functions were chosen to be the *smallest* rigid functions in their orbit (we call smallest, a function with a minimal sum of the degrees of each monomial in its ANF, with lowest indexed variables, in lexicographical order, in descending order by degree of monomials).

We give an example below.

Example 4.1. For $n = 3, \mathcal{T}_3 = 3$, and there is a single orbit with maximum thickness, containing 9 rigid functions, namely: $x_1x_2x_3 + x_3 + 1$, $x_1x_2x_3 + x_2 + 1$, $x_1x_2x_3 + x_1 + 1$, $x_1x_2x_3 + x_2 + x_3$, $x_1x_2x_3 + x_1 + x_3$, $x_1x_2x_3 + x_1 + x_2$, $x_1x_2x_3 + x_2x_3 + x_1$, $x_1x_2x_3 + x_1x_3 + x_2$, $x_1x_2x_3 + x_1x_2 + x_3$. In the first three functions, the sum of the monomial degrees for each function is 4, the next three functions have this sum 5, and the last three, 6. We therefore, look at the first three functions, and going through from the highest to the lowest degree monomials in the three functions, and observing that x_1 is smaller (lexicographically), we therefore choose $x_1x_2x_3 + x_1 + 1$ as a representative.

It is clear that the choice of a representative in any orbit is purely implementation specific and will not affect any properties related to algebraic thickness. As with rigid functions, the set of all representative functions will be denoted as $\mathcal{R}_n \subseteq \mathcal{S}_n$ for representatives in n variables.

The number of Boolean functions in n variables that have exactly m monomials in their ANF is $\binom{2^n}{m}$, and so, the number of Boolean functions with at least m monomials is the sum of the binomial coefficients $\binom{2^n}{i}$, where $i \geq m$, that is, $\sum_{i=m}^{2^n} \binom{2^n}{i}$.

Using Carlet's upper-bound for algebraic thickness in n variables, $\mathcal{T} \leq \left\lfloor \frac{2}{3}2^n \right\rfloor$, it follows that no rigid function will have more than $\left\lfloor \frac{2}{3}2^5 \right\rfloor = 21$ monomials in its ANF. We checked and ultimately, the first monomial count where a rigid function could be found, was $m = 8$ (i.e., first, in descending order). Thus, the maximum thickness of $n = 5$ is 8, by Proposition 3.9.

Our code takes advantage of various "quality-of-life" method calls for printing out current positions – and saving the positions of the iterations, in case of power failure. Surely, the "bottleneck" of finding representatives of functions in five variables is the number of affine transformations to go through for each function – but also the fact that the number of affine transformations is much larger than the number of functions in any orbit (by the pigeonhole principle). This means

that there are several affine transformations that, for each f, maps f to the same function. However, since there is no way of predicting, as far as we know, *which* transformations will do this, it cannot be avoided. The final program used for finding all 382 representatives in $n = 5$ variables (and lower dimensions) can be found in [7], which also includes a listing of these.

5. Distribution of thickness

The full distribution of algebraic thickness of the representative functions in $n \leq 5$ variables is given in Table 3, summarizing the results of the data collection conducted by our program. The distribution for number of functions within each thickness value is further detailed and described later.

$\mathcal{T}$	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
0	1	1	1	1	1	1
1	1	2	3	4	5	6
2	-	-	1	4	10	19
3	-	-	-	1	10	46
4	-	-	-	-	5	81
5	-	-	-	-	1	111
6	-	-	-	-	-	81
7	-	-	-	-	-	33
8	-	-	-	-	-	4
Sum	2	3	5	10	32	382
$\max(\mathcal{T}_n)$	1	1	2	3	5	8

Table 3: Distribution of representatives within each thickness value

While this is known, we re-checked the distribution of functions with a specific nonlinearity $\mathcal{N}$ for $n \leq 5$, confirming the results listed in [10]. Columns for $n = 2, 3$ are not strictly relevant to the following property analysis, but are included for completeness.

$\mathcal{N}$	$n=2$	$n=3$	$n=4$	$n=5$
0	8	16	32	64
1	8	128	512	2048
2	-	112	3840	31 744
3	-	-	17 920	317 440
4	-	-	28 000	2 301 440
5	-	-	14 336	12 888 064
6	-	-	896	57 996 288
7	-	-	-	215 414 784
8	-	-	-	647 666 880
9	-	-	-	1 362 452 480
10	-	-	-	1 412 100 096
11	-	-	-	556 408 832
12	-	-	-	27 387 136
$\max(\mathcal{N})$	1	2	6	12

Table 4: Distribution of number of $f \in \mathcal{B}_n$ with given $\mathcal{N}$-value, $n \leq 5$

Furthermore, the distribution of the number of orbits within each possible $\mathcal{N}$-

value (i.e., the distribution of nonlinearity of the representatives) is shown in Table 5 – recall that nonlinearity is an affine invariant. Note that there are 16 representatives (and therefore orbits) in $n = 5$ variables where $\mathcal{N} = 5$. Further, we can see that there are two orbits with maximum nonlinearity in $n = 4$ (and therefore two orbits that contain all bent functions in $n = 4$), and 14 orbits with maximum nonlinearity in $n = 5$ ($\mathcal{N} = 6$ and $\mathcal{N} = 12$, respectively; recall that the maximum nonlinearity for $n = 5$ is $2^{n-1} - 2^{\frac{n-1}{2}} = 12$, the well known bent concatenation bound).

$\mathcal{N}$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
0	3	3	3	3
1	2	4	4	4
2	-	3	5	5
3	-	-	6	6
4	-	-	8	12
5	-	-	4	16
6	-	-	2	31
7	-	-	-	46
8	-	-	-	68
9	-	-	-	72
10	-	-	-	73
11	-	-	-	32
12	-	-	-	14

Table 5: Distribution of number of orbits with given $\mathcal{N}$-value, $n \leq 5$

6. Conclusions

Table 3 summarizes the outcome of our computation to find the number of orbits in each thickness class, for $n \leq 5$ variables, with the number of orbits and maximum thickness listed. As a double check, the number of equivalence classes (orbits) in $\mathcal{B}_n$ matches the one of Harrison [6].

By using the concepts of rigid and representative functions defined in Sections 3 and 4, the thickness distribution of $n \leq 5$ can be calculated in significantly less time than the time estimation of a brute-force application, by (roughly) $2 \cdot 10^6$ years. The case of $n = 4$ took little time compared to $n = 5$ (we display in Table 6 the time our computation took; iterations stand for the number of parallel sessions we ran).

Mon. count	Functions/Iterations	Min. time	Max. time	Total time (add.)
2	28 / 3	4h	4h	12h
3	134 / 4	6h	12h	1d 12h
4	625 / 4	1d 3h	1d 7h	4d 21h
5	2674 / 8	4d 10h	5d 5h	38d 14h
6	10 195 / 14	1d 14h	3d 19h	39d 17h
7	34 230 / 15	1d 4h	3d 15h	36d 16h
8	100 577 / 20	24s	5d 1h	11d 20h
Total			19d 15h	131d 16h

Table 6: Execution time of the iterations completed by our program

We display in Appendices A and B, the distribution of various cryptographic properties (bentness and semi-bentness, balancedness, etc.) as they relate to thickness, for $n = 4$, respectively, $n = 5$. Three physical computers were used for these computations (which took about 35 days): 1) a dedicated Windows server with Intel(R) Xeon(R) E5-2690 v2 3.00 GHz CPU, 20 cores, and 128 GiB RAM, responsible for the bulk of the calculations, 2) a desktop running Ubuntu with Intel(R) Core(TM) i7-6800K 3.40 GHz CPU, 8 cores, and 32 GiB RAM, and finally 3) a desktop running Windows 10 with Intel(R) Core(TM) i5-4460 3.20 GHz CPU, 4 cores, and 16 GiB RAM. The program iterations referenced in Table 6 were run simultaneously and each program was continually updated whenever new representatives were found.

Acknowledgements. The authors would like to thank the referee for the comments and the editors for the prompt handling of our paper.

References

[1] J. Boyar, M. G. Find: *Constructive Relationships Between Algebraic Thickness and Normality*, in: In Fundamentals of Computation Theory, LNCS 9210, Springer, Cham, 2015, pp. 106–117,
doi: https://doi.org/10.1007/978-3-319-22177-9_9.

[2] C. Carlet: *Boolean Functions*, in: van Tilborg H, ed. by J. S. C. A., Springer, Boston, MA: Encyclopedia of Cryptography and Security, 2011, pp. 162–164,
doi: https://doi.org/10.1007/978-1-4419-5906-5_336.

[3] C. Carlet: *On Cryptographic Complexity of Boolean Functions*, in: Proc. 6th Conf. Finite Fields With Applications to Coding Theory, Springer, 2002, pp. 53–69,
doi: https://doi.org/10.1007/978-3-642-59435-9_4.

[4] C. Carlet: *On the Degree, Nonlinearity, Algebraic Thickness, and Nonnormality of Boolean Functions, With Developments on Symmetric Functions*, IEEE Trans. Inf. Theory 50.9 (2004), pp. 2178–2185,
doi: https://doi.org/10.1109/TIT.2004.833361.

[5] T. W. Cusick, P. Stănică: *Cryptographic Boolean Functions and Applications (2nd ed.)* Elsevier-Academic Press, 2017,
doi: https://doi.org/10.1016/C2016-0-00852-5.

[6] M. A. Harrison: *On the classification of Boolean functions by the general linear and affine groups*, Journal of the Society for Industrial and Applied Mathematics 12.2 (1964), pp. 285–299,
doi: https://doi.org/10.1137/0112026.

[7] M. Hopp: *Thickness Distribution of Boolean Functions in* 4 *and* 5 *Variables*, Master Thesis, Department of Computing, Mathematics and Physics, Western Norway University of Applied Sciences (2020).

[8] F. J. MacWilliams, N. J. A. Sloane: *The theory of error correcting codes*, Amsterdam: North-Holland, 1977.

[9] SageMath: *Open-Source Mathematical Software System*, Last Accessed: 2020-09-10,
url: http://www.sagemath.org.

[10] I. Sertkaya, A. Doğanaksoy: *On the Affine Equivalence and Nonlinearity Preserving Bijective Mappings of* $\mathbb{F}_2$, International Workshop on the Arithmetic of Finite Fields, WAIFI Arithmetic of Finite Fields, (2014), pp. 121–136,
doi: https://doi.org/10.1007/978-3-319-16277-57.

Appendix A: Property distribution in $n = 4$, sorted by thickness

We include here the comparison between various cryptographic properties (homogeneous, rigid, balanced, bentness, nonlinearity, degree) of Boolean functions as related to thickness for $n = 4$ variables. Table 7 is a summary of all the property distributions of Tables 8–12 (independent on algebraic thickness).

Properties	Total number
Number of functions	65536
Homogeneous functions	96
Rigid functions	588
Balanced functions	12 870
Bent functions	896
Orbits	32
Bent orbits	2
Balanced orbits	4

Table 7: Summary of the property distribution of $n = 4$

Properties		Nonlinearity		Degrees	
Number of functions	307	0	31	0	1
Homogeneous functions	52	1	16	1	30
Rigid functions	16	2	120	2	140
Balanced functions	30	3	0	3	120
Bent functions	0	4	140	4	16
Orbits	5	5	0		
Bent orbits	0	6	0		
Balanced orbits	1				

Table 8: Property distribution of functions in $\mathcal{B}_4$ with $\mathcal{T}_4 = 1$

Properties		Nonlinearity		Degrees	
Number of functions	6804	0	0	0	0
Homogeneous functions	42	1	256	1	0
Rigid functions	64	2	2880	2	1428
Balanced functions	2760	3	560	3	4560
Bent functions	448	4	2660	4	816
Orbits	10	5	0		
Bent orbits	1	6	448		
Balanced orbits	2				

Table 9: Property distribution of functions in $\mathcal{B}_4$ with $\mathcal{T}_4 = 2$

Properties		Nonlinearity		Degrees	
Number of functions	33 448	0	0	0	0
Homogeneous functions	1	1	240	1	0
Rigid functions	188	2	840	2	448
Balanced functions	10 080	3	8960	3	19 320
Bent functions	448	4	18 480	4	13 680
Orbits	10	5	4480		
Bent orbits	1	6	448		
Balanced orbits	1				

Table 10: Property distribution of functions in $\mathcal{B}_4$ with $\mathcal{T}_4 = 3$

Properties		Nonlinearity		Degrees	
Number of functions	22 288	0	0	0	0
Homogeneous functions	0	1	0	1	0
Rigid functions	271	2	0	2	0
Balanced functions	0	3	8400	3	6720
Bent functions	0	4	6720	4	15 568
Orbits	5	5	7168		
Bent orbits	0	6	0		
Balanced orbits	0				

Table 11: Property distribution of functions in $\mathcal{B}_4$ with $\mathcal{T}_4 = 4$

Properties		Nonlinearity		Degrees	
Number of functions	2688	0	0	0	0
Homogeneous functions	0	1	0	1	0
Rigid functions	48	2	0	2	0
Balanced functions	0	3	0	3	0
Bent functions	0	4	0	4	2688
Orbits	1	5	2688		
Bent orbits	0	6	0		
Balanced orbits	0				

Table 12: Property distribution of functions in $\mathcal{B}_4$ with $\mathcal{T}_4 = 5$

Appendix B: Property distribution in $n = 5$, sorted by thickness

The cryptographic properties dealt with and the goals of the comparison for $n = 5$ are the same as for $n = 4$.

Properties	Total amount
Number of functions	4 294 967 296
Homogeneous functions	2111
Rigid functions	211 259
Balanced functions	601 080 390
Semi-Bent functions	14 054 656
Number of orbits	382
Semi-Bent orbits	9
Balanced orbits	38

Table 13: Summary of the property distribution of $n = 5$

Properties		Nonlinearity		Degrees	
Number of f	2451	0	63	0	1
Homogeneous f	203	1	32	1	62
Rigid f	32	2	496	2	620
Balanced f	62	3	0	3	1240
Semi-Bent f	0	4	1240	4	496
Orbits	6	5	0	5	32
Semi-Bent orbits	0	6	0		
Balanced orbits	1	7	0		
		8	620		
		9	0		
		10	0		
		11	0		
		12	0		

Table 14: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 1$

Properties		Nonlinearity		Degrees	
Number of f	695 796	0	0	0	0
Homogeneous f	987	1	1024	1	0
Rigid f	336	2	23 808	2	23 188
Balanced f	84 072	3	4960	3	466 736
Semi-Bent f	13 888	4	104 160	4	194 928
Orbits	19	5	0	5	10 944
Semi-Bent orbits	1	6	45 136		
Balanced orbits	3	7	4960		
		8	180 420		
		9	0		
		10	317 440		
		11	0		
		12	13 888		

Table 15: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 2$

Properties		Nonlinearity		Degrees	
Number of f	31 424 328	0	0	0	0
Homogeneous f	859	1	992	1	0
Rigid f	2480	2	7440	2	41 664
Balanced f	4 228 896	3	158 720	3	7620792
Semi-Bent f	874 944	4	1 536 360	4	22 119 120
Orbits	46	5	34 720	5	1 642 752
Semi-Bent orbits	3	6	2 138 752		
Balanced orbits	6	7	853 120		
		8	15 323 920		
		9	317 440		
		10	9 900 160		
		11	277 760		
		12	874 944		

Table 16: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 3$

Properties		Nonlinearity		Degrees	
Number of f	240 101 200	0	0	0	0
Homogeneous f	61	1	0	1	0
Rigid f	11 520	2	0	2	0
Balanced f	15 582 336	3	153 760	3	23 290 176
Semi-Bent f	2 499 840	4	659 680	4	168 597 840
Orbits	81	5	1 416 576	5	48 213 184
Semi-Bent orbits	2	6	10 731 952		
Balanced orbits	6	7	17 541 536		
		8	112 334 080		
		9	18 213 120		
		10	63 162 624		
		11	10 888 192		
		12	4 999 680		

Table 17: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 4$

Properties		Nonlinearity		Degrees	
Number of f	1 086 598 112	0	0	0	0
Homogeneous f	0	1	0	1	0
Rigid f	47 220	2	0	2	0
Balanced f	187 210 240	3	0	3	27 664 896
Semi-Bent f	2 666 496	4	0	4	763 701 120
Orbits	111	5	7 936 992	5	295 232 096
Semi-Bent orbits	1	6	42 413 952		
Balanced orbits	11	7	53 524 352		
		8	364 837 760		
		9	193 162 240		
		10	375 614 848		
		11	40 608 512		
		12	8 499 456		

Table 18: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 5$

Properties		Nonlinearity		Degrees	
Number of f	1 842 215 424	0	0	0	0
Homogeneous f	0	1	0	1	0
Rigid f	59 760	2	0	2	0
Balanced f	308 646 912	3	0	3	7 999 488
Semi-Bent f	7 999 488	4	0	4	951 105 792
Orbits	81	5	3 499 776	5	883 110 144
Semi-Bent orbits	2	6	2 666 496		
Balanced orbits	9	7	96 827 136		
		8	154 990 080		
		9	694 122 240		
		10	788 449 536		
		11	88 660 992		
		12	12 999 168		

Table 19: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 6$

Properties		Nonlinearity		Degrees	
Number of f	935 273 472	0	0	0	0
Homogeneous f	0	1	0	1	0
Rigid f	64 470	2	0	2	0
Balanced f	85 327 872	3	0	3	0
Semi-Bent f	0	4	0	4	174 655 488
Orbits	33	5	0	5	760 617 984
Semi-Bent orbits	0	6	0		
Balanced orbits	2	7	46 663 680		
		8	0		
		9	436 638 720		
		10	174 655 488		
		11	277 315 584		
		12	0		

Table 20: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 7$

Properties		Nonlinearity		Degrees	
Number of f	158 656 512	0	0	0	0
Homogeneous f	0	1	0	1	0
Rigid f	25 440	2	0	2	0
Balanced f	0	3	0	3	0
Semi-Bent f	0	4	0	4	0
Orbits	4	5	0	5	158 656 512
Semi-Bent orbits	0	6	0		
Balanced orbits	0	7	0		
		8	0		
		9	19 998 720		
		10	0		
		11	138 657 792		
		12	0		

Table 21: Property distribution of functions in $\mathcal{B}_5$ with $\mathcal{T}_5 = 8$

Annales Mathematicae et Informaticae
***52** (2020) pp. 137–145*
DOI: https://doi.org/10.33039/ami.2020.09.002
URL: https://ami.uni-eszterhazy.hu

Pentagonal and heptagonal repdigits

Bir Kafle[a], Florian Luca[b], Alain Togbé[a]

[a]Department of Mathematics and Statistics
Purdue University Northwest
Westville, USA
bkafle@pnw.edu
atogbe@pnw.edu

[b]School of Mathematics
University of the Witwatersrand
Wits, South Africa
florian.luca@wits.ac.za

Submitted: July 25, 2019
Accepted: September 24, 2020
Published online: October 8, 2020

Abstract

In this paper, we prove a finiteness theorem concerning repdigits represented by a fixed quadratic polynomial. We also show that the only pentagonal numbers which are also repdigits are 1, 5 and 22. Similarly, the only heptagonal numbers which are repdigits are 1, 7 and 55.

Keywords: Pentagonal numbers, heptagonal numbers, repdigits.

MSC: 11A25, 11B39, 11J86

1. Introduction

It is well known that the polygonal numbers of the forms $n(3n-1)/2$ and $n(5n-3)/2$ are called pentagonal number (OEIS [14] A000326) and heptagonal numbers (OEIS [14] A000566), respectively, where n is any positive integer. Many authors have studied the problems of searching for these numbers in some *interesting* sequence of positive integers.

In 1996, M. Luo [6] has proved that 1 and 5 are the only pentagonal numbers in the Fibonacci sequence and later identified in [7] that 1 is the only pentagonal

number in the Lucas sequence. The so-called generalized pentagonal numbers are given by $n(3n-1)/2$ with n integral, not necessarily positive. In [7], again M. Luo showed that $2, 1$ and 7 are the only generalized pentagonal numbers which are also Lucas numbers. In [10], V. S. Rama Prasad and B. Rao proved that 1 and 7 are the only generalized pentagonal numbers in the associated Pell sequence and subsequently in [11], they identified that the only Pell numbers which are also pentagonal are 1, 5, 12 and 70.

In 2002, B. Rao [13] proved that $1, 4, 7$ and 18 are the only generalized heptagonal numbers (where n is any integer) in the Lucas sequence. Furthermore in [12], B. Rao identified that $0, 1, 13, 34$ and 55 are the only generalized heptagonal numbers in the sequence of Fibonacci numbers.

A positive integer is called a *repdigit* (OEIS [14] A010785), if it has only one distinct digit in its decimal expansion. Repdigits have the form

$$\ell\left(\frac{10^m-1}{9}\right), \quad \text{for some } m \geq 1 \text{ and } \ell \in \{1, 2, \ldots, 9\}.$$

The first few repdigits are

$$1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 11,\ 22,\ 33,\ 44, \ldots, 111,\ \ldots$$

In addition, repunits are particular instances of repdigits, obtained when the repeating digit has the value 1. Earlier in 2000, F. Luca [5] proved that 55 is the largest repdigit in the Fibonacci sequence, and 11 is the largest member of the Lucas sequence which is also the repdigit. In 2012, Marques and Togbé [8] studied the repdigits that are products of consecutive Fibonacci numbers.

According to Ballew and Weger [1], E. B. Escott in 1905 proved that $1, 3, 6, 55, 66$ and 666 are the only triangular numbers of less than 30 digits that consist of a single repeated digit. And in 1975, they [1] proved that, in fact these are the only triangular repdigits. Recently, J. H. Jaroma [3], proved that 1 is the only integer that is both triangular and repunit.

In this paper, we first establish the finiteness of the solutions of some of the equations that involve repdigits, and consequently, we identify the petangonal and heptagonal numbers that are also repdigits.

2. Main results

The following result is a restatement of Theorem 1 in [4].

Theorem 2.1. *Let A, B, C be fixed rational numbers with $A \neq 0$. Then the Diophantine equation*

$$\ell\left(\frac{10^m-1}{9}\right) = An^2 + Bn + C, \tag{2.1}$$

has only finite number of solutions, in integers $m, n \geq 1$ and $\ell \in \{1, 2, \ldots, 9\}$ provided $9B^2 - 36AC - 4A\ell \neq 0$.

Proof. We multiply both sides of equation (2.1) by $4A$, and rearrange some terms, which gives us

$$4A\ell\left(\frac{10^m-1}{9}\right)+B^2-4AC=(2An+B)^2.$$

Further, we can rewrite the last equation as

$$4A\ell 10^{3m_1+r}+\left(9\left(B^2-4AC\right)-4A\ell\right)=9(2An+B)^2, \tag{2.2}$$

where we let $m=3m_1+r$ with $r\in\{0,1,2\}$. We again multiply both sides of equation (2.2) by $16\ell^2 10^{2r}$, thus we get

$$Y^2=X^3+A, \tag{2.3}$$

where

$$X:=4\ell 10^{m_1+r},\quad Y:=12\ell 10^r(2An+B),$$

and

$$A:=16\ell^2 10^{2r}\left(9(B^2-4AC)-4A\ell\right).$$

By the hypothesis, we have $A\neq 0$. Thus, we obtain an elliptic curve over $\mathbb{Q}$ given by (2.3). By a theorem of Siegel (see [9], p. 313), this curve has a finite number of integer points. As a consequence, equation (2.1) has only a finite number of positive integer solutions. □

The result of Ballew and Weger [1] is the case when $A=B=\frac{1}{2}$ and $C=0$ in equation (2.1), though their method of proof is different. Now, we establish some further applications of Theorem 2.1. First, we identify all the pentagonal repdigits. Our result is the following, which comes as a corollary of Theorem 2.1.

Corollary 2.2. *The complete list of pentagonal repdigits is* 1, 5 *and* 22.

Proof. In order to prove our result, we study the equation

$$\ell\left(\frac{10^m-1}{9}\right)=\frac{n(3n-1)}{2}, \tag{2.4}$$

in integers $m,n\geq 1$ and $\ell\in\{1,2,\ldots,9\}$, which is the case when $A=\frac{3}{2}$, $B=-\frac{1}{2}$ and $C=0$ in equation (2.1). Further, working as in the proof of Theorem 2.1, equation (2.4) can be written as

$$y_1^2=x_1^3+a_1, \tag{2.5}$$

where $x_1:=6\ell 10^{m_1+r}$, $y_1:=9\ell 10^r(6n-1)$, and $a_1:=27(3-8\ell)\ell^2 10^{2r}$. We note that a_1 is nonzero, otherwise this would lead to $\ell=3/8$, which is not true. By Theorem 1, the equation (2.4) has only a finite number of solutions in $m,n\geq 1$ and $1\leq\ell\leq 9$. Since $\ell\in\{1,\ldots,9\}$ and $r\in\{0,1,2\}$, we obtain twenty-seven elliptic curves given by (2.5). Now, we determine the integer points (x_1,y_1) on each these elliptic curves. For this, we used MAGMA [2].

The following table displays all[1] the integer points (x_1,y_1)[2], described above

[1] Equation (2.5) has no integer points for $(\ell,r)=(1,2),(3,1),(4,1),(4,2),(5,2),(6,1),(8,1),(8,2),(9,2)$.

[2] (x_1,y_1)'s in **bold** correspond to the integer solutions of the equation (2.4) in the third column.

and corresponding integer solutions (m, n) of the equation (2.4), whenever they exist.

ℓ, r	(x_1, y_1)	(m, n)
$\ell = 1$, $r = 0$	$(6, \pm 9), (19, \pm 82), (24, \pm 117)$	
$\ell = 1$, $r = 1$	$(24, \pm 18), (\mathbf{60}, \pm \mathbf{450}), (85, \pm 775), (2256, \pm 107154)$	(1, 1)
$\ell = 2$, $r = 0$	$(12, \pm 18), (120, \pm 1314)$	
$\ell = 2$, $r = 1$	$(120, \pm 1260)$	
$\ell = 2$, $r = 2$	$(264, \pm 2088), (300, \pm 3600), (1000, \pm 31400),$ $(\mathbf{1200}, \pm \mathbf{41400}), (24400, \pm 3811400),$ $(130296, \pm 47032344)$	(2, 4)
$\ell = 3$, $r = 0$	$(18, \pm 27), (288, \pm 4887)$	
$\ell = 3$, $r = 2$	$(856, \pm 24004)$	
$\ell = 4$, $r = 0$	$(24, \pm 36), (33, \pm 153), (112, \pm 1180), (384, \pm 7524),$ $(528, \pm 12132)$	
$\ell = 5$, $r = 0$	$(30, \pm 45), (46, \pm 269), (64, \pm 487), (75, \pm 630),$ $(120, \pm 1305), (480, \pm 24345), (1654, \pm 67267)$	
$\ell = 5$, $r = 1$	$(136, \pm 134), (\mathbf{300}, \pm \mathbf{4950}), (525, \pm 11925),$ $(4800, \pm 332550)$	(1, 2)
$\ell = 6$, $r = 0$	$(36, \pm 54), (1224, \pm 42822)$	
$\ell = 6$, $r = 2$	$(1224, \pm 37368)$	
$\ell = 7$, $r = 0$	$(42, \pm 63), (1680, \pm 68859)$	
$\ell = 7$, $r = 1$	$(240, \pm 2610), (301, \pm 4501), (420, \pm 8190),$ $(2940, \pm 159390)$	
$\ell = 7$, $r = 2$	$(4200, \pm 270900)$	
$\ell = 8$, $r = 0$	$(48, \pm 72), (2208, \pm 103752)$	
$\ell = 9$, $r = 0$	$(54, \pm 81), (108, \pm 1053), (162, \pm 2025), (279, \pm 4644),$ $(2808, \pm 148797), (2979, \pm 162594),$ $(3310254, \pm 6022710369)$	
$\ell = 9$, $r = 1$	$(1296, \pm 46494)$	

Table 1: Integer solutions (x_1, y_1)

The list of ordered pair (m, n) in third column of Table 1 above, together with the corresponding values of ℓ in the first column give us the complete list of the solutions (m, n, ℓ) in positive integers for equation (2.4). From this, we can deduce that the only pentagonal numbers in the sequence of repdigits are given by the statement of Corollary 2.2. This completes the proof of Corollary 2.2. □

Next, we identify all the heptagonal numbers in the sequence of the repdigits. Our result is the following.

Corollary 2.3. *The complete list of heptagonal repdigits is* 1, 7 *and* 55.

Proof. We let $A = \frac{5}{2}, B = -\frac{3}{2}$ and $C = 0$ in equation (2.1), which allows us to study the following equation (finite number of solutions, by Theorem 2.1),

$$\ell\left(\frac{10^m - 3}{9}\right) = \frac{n(5n-1)}{2}, \tag{2.6}$$

in integers $m, n \geq 1$ and $\ell \in \{1, 2, \ldots, 9\}$. As before, last equation can be reduced to

$$y_2^2 = x_2^3 + a_2, \tag{2.7}$$

where $x_2 := 10\ell 10^{m_1+r}$, $y_2 := 15\ell 10^r(10n-3)$, and $a_2 := 25\ell^2 10^{2r}(81 - 40\ell)$. We note that a_2 is nonzero, otherwise we get $\ell = 81/40$, which is not true. Now, we use MAGMA [2], to determine the integer points (x_2, y_2) on the elliptic curves given by (2.7).

The following table shows all[3] the integer points (x_2, y_2)[4], described above and corresponding integer solutions (m, n) of the equation (2.6), whenever they exist.

ℓ, r	(x_2, y_2)	(m, n)
$\ell = 1$, $r = 0$	$(10, \pm 5), (5, \pm 30), (4, \pm 31), (1, \pm 32), (4, \pm 33)$, $(10, \pm 45), (20, \pm 95), (40, \pm 225), (50, \pm 355), (64, \pm 513)$, $(155, \pm 1930), (166, \pm 2139), (446, \pm 9419), (920, \pm 27905)$, $(3631, \pm 218796), (3730, \pm 227805)$	
$\ell = 1$, $r = 1$	$(\mathbf{100}, \pm\mathbf{1050})$	(1, 1)
$\ell = 1$, $r = 2$	$(200, \pm 1500), (2000, \pm 89500)$	
$\ell = 2$, $r = 0$	$(4, \pm 6), (0, \pm 10), (5, \pm 15), (20, \pm 90), (24, \pm 118)$, $(2660, \pm 137190)$	
$\ell = 2$, $r = 1$	$(0, \pm 100)$	
$\ell = 2$, $r = 2$	$(100, \pm 0), (0, \pm 1000), (200, \pm 3000)$	

[3]Equation (2.7) has no integer points for $(\ell, r) = (6, 1), (6, 2), (8, 1), (9, 1)$.

[4](x_2, y_2)'s in **bold** correspond to the integer solutions of the equation (2.6) in the third column.

$\ell = 3$, $r = 0$	$(30, \pm 135), (40, \pm 235), (1299, \pm 46818)$	
$\ell = 3$, $r = 1$	$(100, \pm 350)$	
$\ell = 3$, $r = 2$	$(1200, \pm 40500)$	
$\ell = 4$, $r = 0$	$(40, \pm 180)$	
$\ell = 4$, $r = 1$	$(184, \pm 1752), (200, \pm 2200), (400, \pm 7800), (1900, \pm 82800),$ $(60625, \pm 14927175)$	
$\ell = 4$, $r = 2$	$(800, \pm 14000)$	
$\ell = 5$, $r = 0$	$(50, \pm 225), (134, \pm 1527), (7550, \pm 656025)$	
$\ell = 5$, $r = 1$	$(200, \pm 750), (6000, \pm 464750)$	
$\ell = 5$, $r = 2$	$(\mathbf{5000}, \pm \mathbf{352500})$	(2, 5)
$\ell = 6$, $r = 0$	$(60, \pm 270), (280, \pm 4670)$	
$\ell = 7$, $r = 0$	$(70, \pm 315), (91, \pm 714), (200, \pm 2785), (2240, \pm 106015)$	
$\ell = 7$, $r = 1$	$(301, \pm 1701), (\mathbf{700}, \pm \mathbf{17850}), (1400, \pm 52150),$ $(7900, \pm 702150)$	(1, 2)
$\ell = 7$, $r = 2$	$(1400, \pm 17500), (25424, \pm 4053532), (49000, \pm 10846500),$ $(325000, \pm 185278500)$	
$\ell = 8$, $r = 0$	$(80, \pm 360), (120, \pm 1160), (200, \pm 2760), (396, \pm 7856),$ $(1244, \pm 43872), (2081, \pm 94929)$	
$c = 8$, $r = 2$	$(1700, \pm 33000), (2400, \pm 100000), (32000, 5724000)$	
$\ell = 9$, $r = 0$	$(90, \pm 405), (171, \pm 2106), (180, \pm 2295), (630, \pm 15795),$ $(700, \pm 18505), (720, \pm 19305),$ $(150750, \pm 58531005), (238770, \pm 116672805)$	
$\ell = 9$, $r = 2$	$(1800, \pm 13500), (2016, \pm 50436), (3600, \pm 202500),$ $(5625, \pm 415125), (9000, \pm 850500), (25425, \pm 4053375),$ $(83800, \pm 24258500), (126000, \pm 44725500)$	

Table 2: Integer solutions (x_2, y_2)

In Table 2, as in the proof of Corollary 2.2, the list of ordered pair (m, n) in third column together with the corresponding values of ℓ in the first column give us the complete list of the solutions (m, n, ℓ) in positive integers with $1 \leq \ell \leq 9$ for the equation (2.4), which are the only pentagonal numbers in the sequence of repdigits. This completes the proof of Corollary 2.3. □

Recently in [4], authors of this paper studied the triangular numbers that are also repeated blocks of two digits, which we call the *repblocks of two digits*. Such numbers have the form

$$\ell\left(\frac{10^{2m}-1}{99}\right), \quad \text{for some } m \geq 1 \text{ and } \ell \in \{10, 11, \ldots, 99\}.$$

Additionally in this paper, we extend and complement the results obtained in [4] by finding all the pentagonal repblocks of two digits. Our results are the following.

Corollary 2.4. *The complete list of pentagonal numbers which are also repblocks of two digits is*

$$12,\ 22,\ 35,\ 51,\ 70,\ 92,\ 1717.$$

Proof. To prove our result, in equation (2.1), we replace the left hand side by $\ell\left(\frac{10^{2m}-1}{99}\right)$, with $\ell \in \{10, 11, \ldots, 99\}$ and the right hand side of it by $A = \frac{3}{2}$, $B = -\frac{1}{2}$ and $C = 0$. As before, the resulting equation can be written as

$$y_3^2 = x_3^3 + a_3, \tag{2.8}$$

where $x_3 := 66\ell 10^{2m_1+2r}$, $y_3 := 1089\ell 10^{2r}(6n-1)$ and $a_3 := 11979\ell^2 10^{4r}(3-8\ell)$. We note that a_3 is nonzero. Now, we use MAGMA [2], to determine the integer points (x_3, y_3) on the two hundred forty-three elliptic curves given by (2.8).

The following table displays all the integer points (x_3, y_3)[5] of (2.8), which produce the corresponding integer solutions (m, n) of equation (2.4). There are only seven such elliptic curves. The other two hundred thirty-six equations either do not have any integer points (x_3, y_3), or do not produce relevant solutions (m, n) and thus, we omit those equations.

ℓ, r	(x_3, y_3)	(m, n)
$\ell = 12$, $r = 1$	$(25524, \pm 3656232)$, $(\mathbf{79200}, \pm\mathbf{22215600})$, $(127600, \pm 45544400)$, $(1753200, \pm 2321384400)$	(1, 3)
$\ell = 17$, $r = 2$	$(1734000, \pm 2259810000)$, $(3706000, \pm 7126910000)$, $(4686000, \pm 10138590000)$, $(\mathbf{11220000}, \pm\mathbf{37581390000})$, $(17217600, \pm 71442126000)$, $(20476500, \pm 92657565000)$, $(166268400, \pm 2143949598000)$	(2, 34)
$\ell = 22$, $r = 1$	$(31944, \pm 2779128)$, $(36300, \pm 4791600)$, $(121000, \pm 41793400)$, $(\mathbf{145200}, \pm\mathbf{55103400})$, $(2952400, \pm 5072973400)$, $(15765816, \pm 62600049864)$	(2, 4)
$\ell = 35$, $r = 1$	$(67200, \pm 13954500)$, $(\mathbf{231000}, \pm\mathbf{110533500})$, $(279400, \pm 147317500)$	(2, 5)
$\ell = 51$, $r = 1$	$(\mathbf{336600}, \pm\mathbf{194386500})$	(1, 6)

[5] (x_3, y_3)'s in **bold** correspond to the integer solutions in the third column.

$\ell = 70$, $r = 1$	$(\mathbf{462000}, \pm\mathbf{312543000})$	(1, 7)
$\ell = 92$, $r = 1$	$(\mathbf{607200}, \pm\mathbf{470883600})$	(1, 8)

Table 3: Integer solutions (x_3, y_3)

The ordered pairs (m, n) in the third column of Table 3, together with the corresponding values of ℓ in the first column give us the complete list of pentagonal numbers which are also the repblock of two digits. This completes the proof of Corollary 2.4. □

In the same fashion, one can show that 18, 34, 55, 81 and 4141 are the only heptagonal numbers which are also repblocks of two digits.

Acknowledgements. B. K. and A. T. are partially supported by Purdue University Northwest, IN.

References

[1] D. W. Ballew, R. C. Weger: *Repdigit triangular numbers*, J. Recreational Math. 8.2 (1975-1976), pp. 96–97.

[2] J. Cannon, C. Playoust: *MAGMA: a new computer algebra system*, Euromath Bull. 2.1 (1996), pp. 113–144.

[3] J. H. Jaroma: *Triangular repunit - there is but* 1, Czech. Math. J. 60 (2010), pp. 1075–1077, doi: https://doi.org/10.1007/s10587-010-0072-9.

[4] B. Kafle, F. Luca, A. Togbé: *Triangular repblocks*, The Fibonacci Quart. 56.4 (2018), pp. 325–328.

[5] F. Luca: *Fibonacci and Lucas numbers with only one distinct digit*, Portugaliae Math. 57.2 (2000), pp. 243–254.

[6] M. Luo: *Pentagonal numbers in the Fibonacci sequence*, Appl. Fibonacci Numbers, Eds. G. E. Bergum et al., Kluwer 6 (1996), pp. 349–354, doi: https://doi.org/10.1007/978-94-009-0223-7_29.

[7] M. Luo: *Pentagonal numbers in the Lucas sequence*, Portugaliae Math. 53.3 (1996), pp. 325–329.

[8] D. Marques, A. Togbé: *On repdigits as product of consecutive Fibonacci numbers*, Rend. Istit. Mat. Univ. Trieste 44 (2012), pp. 393–397.

[9] R. A. Mollin: *Advanced Number Theory with Applications*, Boca Raton, FL: CRC Press, 2010, doi: https://doi.org/10.1201/b12331.

[10] V. S. R. Prasad, B. S. Rao: *Pentagonal numbers in the Associated Pell sequence and Diophantine equations* $x^2(3x-1)^2 = 8y^2 \pm 2$, The Fibonacci Quart. 39.4 (2001), pp. 299–303.

[11] V. S. R. Prasad, B. S. Rao: *Pentagonal numbers in the Pell sequence and Diophantine equations* $2x^2 = y^2(3y-1)^2 \pm 2$, The Fibonacci Quart. 40.3 (2002), pp. 233–241.

[12] B. S. Rao: *Heptagonal numbers in the Fibonacci sequence and Diophantine equations* $4x^2 = 5y^2(5y-3)^2 \pm 16$, The Fibonacci Quart. 41.5 (2003), pp. 414–420.

[13] B. S. Rao: *Heptagonal numbers in the Lucas sequence and Diophantine equations* $x^2(5x-3)^2 = 20y^2 \pm 16$, The Fibonacci Quart. 40.4 (2002), pp. 319–322.

[14] N. J. A. Sloane: *The Online Encyclopedia of Integer Sequences*, OEIS Foundation Inc., electronically at `http://oeis.org` (2020).

Annales Mathematicae et Informaticae
***52** (2020) pp. 147–158*
DOI: https://doi.org/10.33039/ami.2020.02.003
URL: https://ami.uni-eszterhazy.hu

Generalisation of the rainbow neighbourhood number and k-jump colouring of a graph

Johan Kok[a], Sudev Naduvath[a], Eunice Gogo Mphako-Banda[b]

[a]Department of Mathematics, CHRIST (Deemed to be University)
Bangalore-560029, India
johan.kok@christuniversity.in
sudev.nk@christuniversity.in

[b]School of Mathematical Sciences, University of Witswatersrand
Johannesburg, South Africa
eunice.mphako-banda@wits.ac.za

Submitted: July 11, 2019
Accepted: February 13, 2020
Published online: March 23, 2020

Abstract

In this paper, the notions of rainbow neighbourhood and rainbow neighbourhood number of a graph are generalised and further to these generalisations, the notion of a proper k-jump colouring of a graph is also introduced. The generalisations follow from the understanding that a closed k-neighbourhood of a vertex $v \in V(G)$ denoted, $N_k[v]$ is the set, $N_k[v] = \{u : d(v,u) \leq k, k \in \mathbb{N}$ and $k \leq diam(G)\}$. If the closed k-neighbourhood $N_k[v]$ contains at least one of each colour of the chromatic colour set, we say that v yields a k-rainbow neighbourhood.

Keywords: k-rainbow neighbourhood, k-rainbow neighbourhood number, k-jump colouring.

MSC: 05C15, 05C38, 05C75, 05C85.

1. Introduction

For general notation and concepts in graphs and digraphs we refer to [1, 2, 8]. Unless mentioned otherwise all graphs G are simple, connected and finite graphs. For corresponding results of disconnected graphs, see [3].

Recall that a *vertex colouring* of a graph G is an assignment $\varphi : V(G) \mapsto \mathcal{C}$, where $\mathcal{C} = \{c_1, c_2, c_3, \ldots, c_\ell\}$ is a set of distinct colours. A vertex colouring is said to be a *proper vertex colouring* of a graph G if no two distinct adjacent vertices have the same colour. The cardinality of a minimum set of colours in a proper vertex colouring of G is called the *chromatic number* of G and is denoted $\chi(G)$. A colouring of G consisting of exactly $\chi(G)$ colours may be called a *χ-colouring* or a *chromatic colouring* of G.

When the cardinality of the set of colours $\mathcal{C}$ is bound by conditions such as minimum, maximum or others and since $c(V(G)) = \mathcal{C}$, it can be agreed that $c(G)$ means $c(V(G))$ and hence $c(G) \Rightarrow \mathcal{C}$ and $|c(G)| = |\mathcal{C}|$.

Index labelling the elements of a graph such as the vertices say, $v_1, v_2, v_3, \ldots, v_n$ or written as v_i; $1 \leq i \leq n$ or as v_i; $i = 1, 2, 3, \ldots, n$, is called a *minimum parameter indexing*. Similarly, a *minimum parameter colouring* of a graph G is a proper colouring of G which consists of the colours c_i; $1 \leq i \leq \ell$. The set of vertices of G having the colour c_i is said to be the *colour class* of c_i in G and is denoted by $\mathcal{C}_i$. Unless stated otherwise, we consider minimum parameter colouring throughout this paper.

Note that the closed neighbourhood $N[v]$ of a vertex $v \in V(G)$ which contains at least one vertex from each colour class of G in the chromatic colouring, is called *a rainbow neighbourhood* (see [4–7] for further results on rainbow neighbourhoods of different graphs). The number of vertices in G which yield rainbow neighbourhoods, denoted by $r_\chi(G)$, is called the *rainbow neighbourhood number* of G corresponding to the chromatic colouring. Note that $r_\chi^-(G)$ and $r_\chi^+(G)$ respectively denote the minimum value and maximum value of $r_\chi(G)$ over all minimum proper colourings (see [4]).

Rainbow neighbourhood convention ([4]): The rainbow neighbourhood convention is a colouring protocol as described below:

Let X_1 be a maximal independent set in G. Let $G_1 = G - X_1$. Let X_2 be a maximal independent set in G_1 and $G_2 = G_1 - X_2$. Proceed like this, until after a finite number of iterations, say k, the induced graph $\langle X_k \rangle$ is a trivial or empty graph. Clearly, we have $|X_1| \geq |X_2| \geq \ldots |X_{k-1}| \geq |X_k|$. Now, consider a set $\mathbb{C} = \{c_1, c_2, \ldots, c_k\}$ of k colours and we assign the colour c_i to all vertices in X_i for $1 \leq i \leq k$.

Unless mentioned otherwise the rainbow neighbourhood convention together with a minimum parameter colouring will be used for all graph colourings.

2. *k*-rainbow neighbourhood number of a graph

In this section, we generalise the notion of a rainbow neighbourhood of a graph. A *closed k-neighbourhood* of a vertex $v \in V(G)$, denoted by $N_k[v]$, is the set, $N_k[v] = \{u : d(v,u) \leq k, k \in \mathbb{N}\}$ (Note that $k \leq diam(G)$).

Definition 2.1. If the closed k-neighbourhood $N_k[v]$; $v \in V(G)$ contains at least one of each colour from the chromatic colour class, we say that v yields a k-rainbow neighbourhood.

In this context, a rainbow neighbourhood defined in [5] is indeed a 1-rainbow neighbourhood.

Definition 2.2. For a chromatic colouring of a graph G, the number of distinct vertices which yield a k-rainbow neighbourhood is called the *k-rainbow neighbourhood number* of G and is denoted by $r_{\chi,k}(G)$.

Definition 2.3. The *k^--rainbow neighbourhood number* of a graph G, denoted by $r^-_{\chi,k}(G)$, is defined as the minimum number of distinct vertices which yield a k-rainbow neighbourhood. That is,

$$r^-_{\chi,k}(G) = \min\{r_{\chi,k}(G) : \text{over all chromatic colourings of } G\}.$$

Definition 2.4. The *k^+-rainbow neighbourhood number* of a graph G, denoted by $r^+_{\chi,k}(G)$, is defined as the maximum number of distinct vertices which yield a k-rainbow neighbourhood. That is,

$$r^+_{\chi,k}(G) = \max\{r_{\chi,k}(G) : \text{over all chromatic colourings of } G\}.$$

Note that $r^-_{\chi,k}(G)$ necessarily corresponds to a chromatic colouring in accordance with the rainbow neighbourhood convention. Note that if vertex v yields a k-rainbow neighbourhood it does not imply that v yields a $(k-1)$-rainbow neighbourhood. The aforesaid is true because $N_{(k-1)}[v] \subseteq N_k[v]$ and hence for any colouring, $|N_k[v]| \geq |N_{(k-1)}[v]|$. However, all vertices yield a $diam(G)$-rainbow neighbourhood. Hence, for a graph G of order n we have, $r_{\chi,diam(G)}(G) = n$. Also, if the vertex v yields a 1-rainbow neighbourhood, it yields a k-rainbow neighbourhood, where $2 \leq k \leq diam(G)$.

We now present a fundamental recursive lemma.

Lemma 2.5. *If the vertex $v \in V(G)$ yields a t-rainbow neighbourhood in graph G, it yields a k-rainbow neighbourhood for $t+1 \leq k \leq diam(G)$.*

Proof. Because $N_t[v] \subseteq N_k[v]$, $t+1 \leq k \leq diam(G)$, the result immediately follows by mathematical induction. □

Lemma 2.5 implies that $r_{\chi,k}(G) \geq r_\chi(G)$, because for a vertex v that yields a rainbow neighbourhood all $u \in N[v]$ yields a 2-rainbow neighbourhood if $N_2[u]$ exists. For now our interest lies in understanding the invariant $r_{\chi,2}(G)$ and determining $r^-_{\chi,2}(G)$.

Proposition 2.6. *The minimum 2-rainbow neighbourhood number for the following graphs, all of order n are:*

(i) For 2-colourable graphs G, $r^-_{\chi,2}(G) = n$.

(ii) For cycle C_3, $r^-_{\chi,2}(C_3) = 3$, for C_n, n is odd and $n \geq 5$ we have: $r^-_{\chi,2}(C_n) = 5$.

(iii) For wheels $W_n = K_1 + C_n$, $n \geq 3$:

$$r^-_{\chi,2}(W_n) = \begin{cases} 4, & \text{if } n = 3; \\ 6, & \text{if } n \geq 5, \ n \text{ is odd}; \\ n+1, & \text{if } n \text{ is even.} \end{cases}$$

Proof. (i) Because $r^-_{\chi}(G) = n$ for 2-colourable graphs and $r^-_{\chi}(G) = r^-_{\chi,1}(G)$ it follows that, $r^-_{\chi,2}(G) = n$.

(ii) The first part, which states that $r^-_{\chi,2}(C_3) = 3$, is straight forward. Furthermore, because $r^-_{\chi,2}(G)$ corresponds to a chromatic colouring in accordance with the rainbow neighbourhood convention, such chromatic colouring of a cycle C_n, n is odd and $n \geq 5$ permits a single vertex to have colour c_3. The result follows immediately from the aforesaid.

(iii) Part (1) and Part(2) of (iii) are direct consequence of (ii). Furthermore, since an even cycle is 2-colourable, result (i) read together with the fact that the central vertex is adjacent to all cycle vertices implies that, $r^-_{\chi,2}(C_n) = n + 1$ if n is even. □

The results for many other cycle related graphs such as sun graphs, sunlet graphs, helm graphs and so on, can be derived easily through similar reasoning.

2.1. *k*-rainbow neighbourhood number of certain graph operations

Generally, graph operations are distinguished between *operations on a graph G* such as the complement graph, the line graph, the total graph, the power graph and so on. It results in a new graph or a derivative graph of the given graph G. Then there are those which are *operations between* graphs G and H. In this subsection the join and the corona of graphs G and H will be considered.

Theorem 2.7. *Let two graphs G and H of order n_1, n_2 respectively. Let $G + H$ and $G \circ H$ be the join and the corona of G and H. Then,*

(i) $r^-_{\chi,2}(G + H) = n_1 + n_2$.

(ii) (a) $r^-_{\chi,2}(G \circ H) = n_1 \cdot n_2$, if $\chi(H) \geq \chi(G) - 1$; else,

(b) $r^-_{\chi,2}(G \circ H) = r^-_{\chi,2}(G)$.

Proof. (i) Since, for any two vertices $v, u \in V(G + H)$ the distance is, $d(v, u) \leq 2$, the result is immediate.

(ii)(a): For $\chi(H) \geq \chi(G) - 1$ each $v \in V(G)$ yields a rainbow neighbourhood. Also for $u \in V(H)$, $d(v, u) \leq 2$, and therefore, the result is immediate.

(b): For the second part it is clear that all the vertices $v \in V(G)$ that yield a 2-rainbow neighbourhood in G will yield a 2-rainbow neighbourhood in $G \circ H$. Therefore, $r^-_{\chi,2}(G \circ H) \geq r^-_{\chi,2}(G)$.

It also follows that no vertex $w \in V(H)$ can yield a 2-rainbow neighbourhood in $G \circ H$. To show the aforesaid, assume that the vertex $w \in V(H)$ of the t-th copy of H joined to $v \in V(G)$ is a vertex yielding a 2-rainbow neighbourhood in $G \circ H$. It means that vertex w has at least one 2-reach neighbour for each colour c_i, $1 \leq i \leq \chi(H) < \chi(G) - 1$ as well as the neighbour v with, without loss of generality the colour $c(v) = c_{\chi(H)+1}$. Since, $c_{\chi(H)+1}$ can at best be the colour $c_{\chi(G)-1}$, the colour $c_{\chi(G)} \notin N[w]$ in $r^-_{\chi,2}(G \circ H)$ which is a contradiction. Therefore, $r^-_{\chi,2}(G \circ H) = r^-_{\chi,2}(G)$. □

3. On *k*-jump colouring

In this section, we introduce the main concept of study and the main results of this paper.

A path of length k also called a *k-path* is a path on $k+1$ vertices. Similarly, a cycle of length (or circumference) k, also called a *k-cycle* is a cycle on k vertices. If a graph G has $diam(G) = \ell$, then clearly it is possible for each vertex $v \in V(G)$ to find a vertex u which is at maximum distance $d(v,u) = \ell' \leq \ell$ and hence furthest away from v in G. We say u is a *ℓ'-jump away* from v. Consider a graph G for which $X \cup Y = V(G)$ and for which the vertices in set $X \subseteq V(G)$ are uncoloured and the vertices in set $Y \subseteq V(G)$ are coloured. We say G is partially coloured.

Definition 3.1. Consider a partially coloured graph G and let the set of uncoloured vertices be $X \subseteq V(G)$. A *k-jump colouring* in G with respect to v is the colouring in G such that of vertex $v \in X$ together with all vertices $u \in X$ for which $d(v,u) = k$ have the same colour.

The rainbow neighbourhood convention can naturally be extended to vertices at distance k. The derivative is called the rainbow k-neighbourhood convention. It is also clear that since G is finite, that colouring v say, c_1 and then colouring all vertices u_i at jump ℓ' from v also c_1 followed by repeating the colouring procedure for all ℓ'-jumps from vertices u_i and so on will exhaust in finite number of iterations and either, colour all vertices in G the colour c_1 or result in some vertices remaining uncoloured. It means that no vertex which remains uncoloured is at distance ℓ from any vertex coloured c_1. The aforesaid implies that the procedure is possible for a k-jump, $k \leq \ell$. For a graph G with $diam(G) = \ell$ and $0 \leq k \leq \ell$, consider the k-jump colouring procedure (k-JCP) as explained below:

k-JCP for a graph

Step-0: Let $\mathcal{V}_0 = \emptyset$.
Step-1: For $0 \leq k \leq diam(G)$, choose an arbitrary vertex $v_1 \in V(G)$. Let $\mathcal{V}_1 = \mathcal{V}_0 \cup \{v_1\}$ and colour v_1 and all uncoloured vertices $u_{1,i} \in V(G)$ at distance k

(k-jump) from v_1 if such vertices exist, the colour c_1. Repeat the procedure for all vertices $u_{1,i}$ to obtain all vertices $w_{1,i}$ to be coloured c_1 and so on. When this procedure is exhausted proceed to Step 2.
Step 2: If any uncoloured vertices exist, choose an arbitrary vertex v_2. Let $\mathcal{V}_2 = \mathcal{V}_1 \cup \{v_2\}$ and colour v_2 and all uncoloured vertices $u_{2,i}$ at distance k (k-jump) from v_2 if such vertices exist, the colour c_2. Repeat the procedure similar to that in Step 1 for all vertices $u_{2,i}$ to obtain all vertices $w_{2,i}$ to be coloured c_2, if such vertices exist and so on. When this procedure is exhausted proceed to Step 3.
Step-3: If possible proceed iteratively through the arbitrary choice of an uncoloured v_3 and update $\mathcal{V}_3 = \mathcal{V}_2 \cup \{v_3\}$ and colour corresponding k-jump vertices c_3, and so on, until the graph has a k-jump colouring which might not be proper.
Step-4: When this iterative procedure is exhausted, delete all edges between vertices u and v for which $c(u) = c(v)$.

On conclusion of Step-4, a proper colouring is obtained. Call the concluding set of vertices say, $\mathcal{V}_i$, a *k-string*. Note that it means that the graph permits a maximum of i colours in respect of the k-string $\mathcal{V}_i$. For the corresponding set of colours $\mathcal{C}$, we call the mapping $f_{\mathcal{V}_i} : V(G) \mapsto \mathcal{C}$, a *$k$-jump colouring* of G in respect of $\mathcal{V}_i$. The *k-jump colouring number* of G, with respect to the rainbow k-neighbourhood convention, is defined to be, $\chi_{J(k)}(G) = j = |\mathcal{V}_j| = \max\{|\mathcal{V}_i| : f_{\mathcal{V}_i}(G); \text{a } k\text{-jump colouring of } G \text{ in respect of } \mathcal{V}_i\}$. It is easy to verify that $\chi_{J(2)}(C_9) = 1$, $\chi_{J(3)}(C_9) = 3$ and $\chi_{J(4)}(C_9) = 1$. Hence, in general there is no relation between $\chi_{J(k)}(G)$ and k per se. Also, there is no relation between the chromatic number $\chi(G)$ and the jump colouring number, $\chi_{J(k)}(G)$.

For $k = 0$ we have the jump string $\mathcal{V}_n = V(G)$ and $c(v) \neq c(u) \Leftrightarrow v \neq u$. It is called the *Type I primitive jump colouring*. For $k = 1$ the we have the k-string, $\mathcal{V}_1 = \{v\}$, $v \in V(G)$, $c(G) = c_1$. It is called the *Type II primitive jump colouring* which returns a null graph in Step 4 of the k-JCP.

Further throughout this section the bounds for a k-jump colouring, $2 \leq k \leq diam(G)$ will apply. A complete graph K_n, $n \geq 3$ only permits a k-jump colouring for $k = 0, 1$ and the 1-jump colouring always returns a null graph. It is easy to verify that a path P_n, $n \geq 3$ has $\chi_{J(k)}(P_n) = k$, $1 \leq k \leq n-1$. Because acyclic graphs are bipartite and hence 2-colourable, such graphs permit a 2-jump colouring without the deletion of any edges. It implies that the 2-jump colouring returns a chromatic 2-colouring. For 2-colourable graphs G, $\chi_{J(2)}(G) = \chi(G)$. It is easy to see that a 2-jump colouring returns a null graph for an odd cycle graph, meaning that all vertices are coloured c_1. We say that an odd cycle permits a *Type II primitive jump colouring* or *returns a null graph* in respect of a 2-jump colouring. We are now in a position to state and prove two of the main results of this study.

Theorem 3.2. *A non-trivial graph G returns a null graph in respect of a 2-jump colouring if and only if G contains an odd cycle (not necessarily an induced odd cycle).*

Proof. Say that for an odd cycle $C_m \subseteq G$ and $u, v \in V(C_m)$, $m \leq n$, a 2-path from u to v, if it exists, is *within* C_m. Similarly, say that a 2-path from u to v, $u \notin V(C_m)$, $v \in V(C_m)$ if it exists, is *into* C_m. Also, say that a 2-path from u to v,

$u \in V(C_m)$, $v \notin V(C_m)$ if it exists, is out of C_m. Consider a graph which contains an odd cycle, C_m, $m \leq n$. Here are two sub-cases to be considered.

(a) Assume that G has odd cycle C_m and the arbitrary vertex $v_1 \notin V(C_m)$. For any vertex $u \in V(C_m)$ a vu-path exists because G is connected. If the vu-path is odd then $c(v_1) = c(u) = c_1$. Without loss of generality, 2-jump colour the cycle to exhaustion, followed by 2-jump colouring the vu-path. It follows that $c(V(vu\text{-path}) \cup V(C_m))) = c_1$.

(b) If the vu-path is even then a vertex w which is adjacent to u exists and which does not lie on the vu-path. Extend to the vw-path which is odd and 2-jump colour similar to (a). It follows that $c(v_1) = c(w) = c_1$. Without loss of generality, 2-jump colour the cycle to exhaustion, followed by 2-jump colouring the vu-path. It follows that $c(V(vu - path) \cup V(C_m))) = c_1$.

Invoking the sub-cases (a), (b) together, the result follows by mathematical induction.

If a non-trivial graph G returns a null graph with respect to a 2-jump colouring, the result follows by logical deduction in that, from say v_j, the 2-jump colouring iteration must be along a combination of paths or even cycles (not necessarily induced even cycles). □

The proof of Theorem 3.2 makes a generalized result for cycles possible. Note that for the discussion of cycles and chorded cycles and certain cycle related graphs the bounds on k are relaxed for convenience to, $2 \leq k \leq n$. For graphs in general a similar relaxation is possible by substituting modulo bounds on $diam(G)$.

Theorem 3.3. *Let $k \geq 3$. A cycle C_n, returns a null graph in respect of a k-jump colouring if and only if $n \neq t \cdot k$ where $t \in \mathbb{N}$.*

Proof. For a cycle C_n, $n \geq 3$ and by relaxed convention, $2 \leq k \leq n$, all paths from vertices u to v are within C_n. Also, for any n-path from u to v we have $u = v$. Similarly, for any k for which n is divisible by k, a $(k \cdot \frac{n}{k})$-path from u to v implies $u = v$. Therefore, for any k for which n is not divisible by k, Step 1 will exhaust all vertices with colouring c_1. Hence, the result. □

The following two corollaries are direct consequences of Theorem 3.3.

Corollary 3.4. *For $k_1, k_2, k_3, \ldots, k_s$ and $k_i \geq 3$, let the least common multiple, $LCM(k_1, k_2, k_3, \ldots, k_s) = \ell$. A cycle C_n, returns a null graph in respect of a k_i-jump colouring if and only if $n \neq t \cdot \ell$ where $t \in \mathbb{N}$.*

Corollary 3.5. *For $k_1, k_2, k_3, \ldots, k_s$ and $k_i \geq 3$, let the least common multiple, $LCM(k_1, k_2, k_3, \ldots, k_s) = \ell$. A cycle C_n, has $\chi_{J(k_i)}(C_n) = 1$ or k_i in respect of a k_i-jump colouring.*

It is observed that cycles has the extremal edge deletion properties i.e. either all edges are deleted for a k-jump colouring or no edges are deleted.

3.1. Investigating chorded cycles, slings graphs and p-sling graphs

From Corollary 3.4 a general result for chorded cycles follows.

Theorem 3.6. *For $k_1, k_2, k_3, \ldots, k_s$ and $k_i \geq 3$ let the least common multiple, $LCM(k_1, k_2, k_3, \ldots, k_s) = \ell$. A chorded cycle $C_n^{\circledast}$, $n \geq 4$ returns a null graph in respect of a k_i-jump colouring.*

Proof. From Corollary 3.4, a cycle C_{m_1} and C_{m_2} must both have $m_1 = t_1 \cdot \ell$, $t_1 \in \mathbb{N}$ and $m_2 = t_2 \cdot \ell$, $t_2 \in \mathbb{N}$ for each to permit a k_i-jump colouring, $1 \leq i \leq s$. Obtain a chorded cycle $C_n^{\circledast}$ by merging two edges, one each from C_{m_1} and C_{m_2}. It is easy to verify that $n = m_1 + m_2$ is not divisible by at least one k_i, $1 \leq i \leq s$. From Corollary 3.3 it then follows that $C_n^{\circledast}$ will return a null graph. Though immediate induction the resut follows for any chorded graph $C_n^{\circledast}$, $n \geq 4$. □

An immediate consequence of Theorem 3.6 is that Theorem 3.2 cannot be generalized for k-jump colouring for $k \geq 3$. Hence, for k-jump colouring, $k \geq 3$ only graphs with edge-disjoint holes (induced cycles) can be investigated.

Consider a cycle C_n, $n \geq 3$ and a path P_{m+1}, $m \geq 1$ (also called a m-path). The graph obtained by merging an end vertex of the path with a vertex of C_n is called a *sling graph* and is denoted by $S_{n,m+1}$. We begin with an important lemma.

Lemma 3.7. *Let the vertices of a m-path be labeled, $v_1, v_2, v_3, \ldots, v_{m+1}$. For the cycle C_n, $n = t \cdot \ell$, $t = 1, 2, \ldots$, and $\ell = LCM(1, 2, 3, \ldots, m)$, construct the sling graph $S_{n,m+1}$ by merging v_1 with a vertex on C_n. For $2 \leq k \leq m$ initiate (Step 1 of the k-JCP) a k-jump colouring from vertex v_{k+1}. The sling graph $S_{n,m+1}$ permits such k-jump colouring.*

Proof. Initiating a k-jump colouring from vertex v_{k+1} in accordance with the conditions set, clearly colours vertex v_1 to be, $c(v_1) = c_1$. Proceeding along the cycle without returning a null graph follows from Corollary 3.4. □

A p-sling graph has paths, P_{m_i+1}, $1 \leq i \leq p$, each linked to a common cycle in accordance to the construction of a sling graph. It is denoted, $S_{n,m_i+1}^{1 \leq i \leq p}$. In this sense a sling graph is a 1-sling graph.

Assume without loss of generality that $m_1 \leq m_2 \leq m_3 \leq \cdots \leq m_p$. Label the vertices of the respective paths to be, $v_{i,1}, v_{i,2}, v_{i,3}, \ldots, v_{i,m_i}$, $1 \leq i \leq p$. The next lemma generalizes Lemma 3.7.

Lemma 3.8. *For a cycle C_n, $n = t \cdot \ell$, $t \in \mathbb{N}$, and $\ell = LCM(1, 2, 3, \ldots, m_p)$, construct the p-sling graph $S_{n,m_i+1}^{1 \leq i \leq p}$ by merging $v_{i,1}$ with some vertex on C_n. For $2 \leq k \leq m_p$ initiate (Step-1 of the k-JCP), a k-jump colouring from any vertex $v_{i,k+1}$. The p-sling graph $S_{n,m_i+1}^{1 \leq i \leq p}$ permits such k-jump colouring if all paths P_{m_j+1}, $j \neq i$ are merged with some vertex on C_n which is coloured c_1.*

Proof. Note that ℓ is divisible by m_i, $1 \leq i \leq p$. The result follows trivially from Lemma 3.7 by induction on the number of paths. □

A trivial illustration of Lemma 3.8 is the observation that a thorny cycle $C_n^\star$, n is even, permits a 2-jump colouring.

Theorem 3.9. *If a graph G which permits a k-jump colouring then $v \in V(G)$ yields a $(k-1)$-rainbow neighbourhood.*

Proof. Consider any vertex v and any $(k-1)$-path P_k leading from v. Label the vertices on P_k to be, $v_1, v_2, v_3, \ldots, v_k$. Since for any pair of distinct vertices say, v_i, v_j the distance, $d(v_i, v_j) \leq k-1$ it follows that $c(v_i) \neq c(v_j)$. Therefore, all $c(P_k) = \mathcal{C}$. Hence, the result. □

Theorem 3.9 implies that if G permits a k-jump colouring, then $r^-_{\chi,(k-1)}(G) = |V(G)|$.

Theorem 3.10. *For $2 \leq k \leq diam(G)$, the k-jump colouring of G returns a null graph if G contains a cycle C_m (not necessarily induced) of length, $m \neq t \cdot k; t = 1, 2, 3 \ldots$.*

Proof. The result follows by similar reasoning to that found in the proof of Theorem 3.2. □

3.2. On acyclic graphs

With some understanding of the importance of path, cycles and chorded cycles two general results can be stated. We begin with two important lemmas.

Lemma 3.11. *If an acyclic graph G with $diam(G) = \ell$, permits a k-jump colouring for $2 \leq k \leq diam(G)$ such colouring is unique (up to isomorphism).*

Proof. Note that for an acyclic graph a path from v to v in G exists and is unique. Hence, Theorem 3.9 read together with with any injective mapping $f : \mathcal{C} \mapsto \mathcal{C}$ implies up to isomorphism that the k-jump colouring is unique. □

Lemma 3.11 implies that a k-jump colouring may initiate from any $v \in V(G)$.

Lemma 3.12. *If an acyclic graph G is k-jump colourable, $2 \leq k \leq diam(G)$ then G is tk-jump colourable for $2 \leq tk \leq diam(G)$.*

Proof. Let G be k-jump colourable, $2 \leq k \leq diam(G)$. Note that for an acyclic graph G a path from v to v in G exists and is unique. Consider a vertices v, u, w such that $d(v,u) = k$ and $d(u,w) = (t-1)k$. Clearly $c(v) = c(u) = c(w)$. Hence, in a t·-jump colouring, $c(v) = c(w) \neq c(u)$. The aforesaid holds for all vu-paths and all uw-paths in G. Therefore, the result follows through immediate induction. □

Theorem 3.13. *An acyclic graph G with $diam(G) = \ell$, permits a k-jump colouring for $k = 2, 3, 4, \ldots, \ell$.*

Proof. If G is acyclic the result for $k = 2, 3, 5, 7, \ldots, p \leq diam(G)$, p is prime follows by the same reasoning as for $d(v, u) = k$ and $d(u, w) = (t-1)k$ in the proof of Lemma 3.12. For the multiples of the corresponding prime jumps, the result is a direct consequence of Lemma 3.12. □

We can now state and prove results for the elementary graph operations, join and corona. First, the result for the corona $P_n \circ H$ will be stated.

Remark 3.14. Heuristic reasoning suggests that in Step i of the k-JCP the vertex v_i should be such that an uncoloured vertex u at maximum distance from v_i (furthest away) exists. So for such v_1 such u always exists at distance $d(v_1, u) = diam(G)$.

Theorem 3.15. *The join $G+H$ of two graphs G and H returns a Type II primitive jump colouring.*

Proof. Since $diam(G + H) = 2$ we only consider $k = 2$. Without loss of generality consider vertices $v, u \in V(G)$ and vertex $w \in V(H)$. Since $d(v, u) \geq 2$ in G there exists a cycle from v to u to w to v in $G+H$ with length (circumference) at least 4. If the cycle length is odd the result follows from Theorem 3.2. If the cycle length is even then since there exists a vertex v' adjacent to v on a vu-path in G, there exists an odd cycle from v' to u to w to v' in $G + H$. Similarly the result follows from Theorem 3.2. □

For the corona of graphs some special graph classes will be discussed.

Proposition 3.16. *(i) For a path P_n, $n \geq 4$ and graph H of order m, the corona $P_n \circ G$ is k-jump colourable, if $2 \leq k \leq n + 1$ and $k \neq 3$. A 3-jump colouring returns a Type-II trivial jump colouring.*
(ii) For P_n, $n = 1, 2, 3$, 2-jump colourings are returned.

Proof. (i) Consider any path P_n, $n \geq 4$ and any graph H of order m. Two sub-cases must be considered.

(a) Let $k = 3$. In accordance with the rainbow k-neighbourhood convention and without loss of generality begin Step 1 of the k-JCP by selecting any $u \in V(H_1)$. The first iteration results in $c(u) = c(v_3) = c(V(H_2) = c_1$. The second iteration results in $c(v_4) = c(V(H_3) = c_1$ followed by, $c(v_1) = c_1$. Immediate iterative exhaustion shows that a Type II trivial jump colouring returns.

(b) Begin by considering the case of maximum k-jump. Clearly $diam(P_n \circ H) = n + 1$. Let the path vertices be $v_1, v_2, v_3, \ldots, v_n$ and the corresponding corona'd copies of H be labeled $H_1, H_2, H_3, \ldots, H_n$. In accordance with the rainbow k-neighbourhood convention and without loss of generality begin Step 1 of the k-JCP by selecting any $u \in V(H_1)$. Step 1 results in $c(u) = c(V(H_n) = c_1$. Similarly, Step 2 results in $c(V(H_1) = c_1$. Hereafter, for $1 \leq i, j \leq n$ and $2 \leq j' \leq n - 2$, all pairs of vertices v_i, v_j, $v_i u_{j'}$, $u_{j'} \in V(H_{j'}$ and pair $u_{i'} u_{j'}$ all distances are at most, $n - 1$. Hence, k-JCP results in each vertex in $\{v_i : 1 \leq i \leq n\} \bigcup_{j=2}^{n-1} V(H_j)$ to be distinctly coloured. The result follows for $k = n + 1$. By immediate inverse induction the result follows for $k \neq 3$.

(ii)(a) For $k = 2$, and applying k-JCP to $P_1 \circ H_1$ returns a 2-jump colouring. $P_2 \circ H$ returns a 2-jump colouring. Also, $P_3 \circ H$ returns a 2-colouring.

(b) For $k = 3$, and applying k-JCP to $P_2 \circ H$ returns a 2-jump colouring with 3 colours needed. $P_3 \circ H$ returns a 2-jump colouring with all vertices except v_2 coloured c_1 and $c(v_2) = c_2$. □

Theorem 3.17. *Consider a cycle C_n, $n \geq 3$. For all graphs H, of order m the k-colourability of the corona, $C_n \circ H$ is equivalent to the k-colourability of the thorny graph $C_N^\star$ with m thorns (pendant vertices) attached to each vertex, $v \in V(C_n)$.*

Proof. The adjacency properties of H are irrelevant in $C_n \circ H$ in that for $v, u \in V(H)$ the distance reduces to $d(v, u) \leq 2$. So for the direct application of Lemma 3.8, $C_n \circ H$ can be treated as if, equivalent to a thorny cycle. □

3.3. On modified *k*-jump colouring

Consider a cycle C_n, $n \geq 3$ which for some $2 \leq k \leq n-1$ is not k-jump colourable. Certainly P_n is k-jump colourable. Now allocate any colour $c_i \in \mathcal{V}_k$, $c_i \neq c(v_n)$ or a new colour c_{k+1} to vertex v_n in accordance to a proper colouring. If colour c_{k+1} is needed, then update, $\mathcal{V}_{k+1} = \mathcal{V}_k \cup \{c_{k+1}\}$. The $(k+1)$-string colouring of C_n is called a *modified k-jump colouring* of C_n. Now similarly for P_n which has been k-jump coloured, it is possible to recolour a vertex v_i with $c_j \in \mathcal{V}_k$ or with c_{k+1} to add the edge $v_i v_j$. From Theorem 3.12 it follows that for a graph G and $2 \leq k \leq diam(G)$, any spanning tree T of G is k-jump colourable. Therefore it is possible to obtain a modified k-jump colouring of G by iteratively applying the colouring principles set out. Clearly the modified modified k-jump colouring obtained in respect of a particular spanning tree is minimal. The minimum colours in a modified k-jump colouring over all distinct spanning trees is the *optimal modified k-jump colouring* of G.

Theorem 3.18. *For any graph G and $2 \leq k \leq diam(G)$, an optimal modified k-jump colouring exists.*

Proof. For any graph G and any spanning tree T we have, $diam(G) \leq diam(T)$. Hence, $2 \leq k \leq diam(G) \Rightarrow 2 \leq k \leq diam(T)$. Therefore, from Theorem 3.15, it follows that all possible distinct spanning trees are k-jump colourable and therefore permits a corresponding modified k-jump colouring. By the principle of well-ordering of integers a minimum number of colours exists over all minimal modified k-jump colourings of G. □

4. Conclusion

In this paper, we introduced the notion of the k-rainbow neighbourhood number of a graph G. There is a wide scope for determining the minimum and maximum k-rainbow neighbourhood numbers for many other classes of graphs. In terms of

graph operations on and between graphs, investigations in respect of the complement of a graph, the line graph, the jump graph, the total graph etc. seem to be promising. Studies in this area on graph products such as the Cartesian product, the tensor product, the strong product and the lexicographical product of various graph classes also seem to be worthy research directions.

In this article, we also introduced a new notion of a k-jump colouring of graphs. Further studies on various aspects of k-jump colouring remains open. Note from Proposition 3.16 that for the $(n+1)$-jump colouring, where $n \geq 4$, $\chi_{J(n+1)}(P_n \circ H) = (n+1)+m(n-2)$. Determining the values of $\chi_{J(k)}(P_n \circ H)$, $0 \leq k \leq diam(P_n \circ H)$ is another open problem in this area.

Complexity analysis with respect to the optimal modified k-jump colouring of a graph G is considered to be worthy research. There are good algorithms to find the spanning trees such as Prim's algorithm for edge weighted graphs and Kruskal's algorithm. It is also well-known that the number of distinct spanning trees of a graph denoted by, $t(G)$ can be calculated by using the Kirchhoff matrix-tree theorem.

All the above mentioned facts show that there is a wide scope for further investigations in this direction.

Acknowledgements. Authors would like to acknowledge the positive and critical comments of the referee(s), which helped improving the content and presentation style of the paper significantly.

References

[1] J. A. Bondy, U. S. R. Murty: *Graph theory with applications*, Macmillan London, 1976, doi: https://doi.org/10.1007/978-1-349-03521-2.

[2] F. Harary: *Graph theory*, Narosa, New Delhi, 2001.

[3] J. Kok, S. Naduvath: *On compônentă analysis of graphs*, arXiv preprint arXiv:1709.00261 (2017).

[4] J. Kok, S. Naduvath, O. Buelban: *Reflection on rainbow neighbourhood numbers of graphs*, arXiv preprint arXiv:1710.00383 (2017).

[5] J. Kok, S. Naduvath, M. K. Jamil: *Rainbow neighbourhood number of graphs*, Proyecciones (Antofagasta) 38.3 (2019), pp. 469–484, doi: https://doi.org/10.22199/issn.0717-6279-2019-03-0030.

[6] S. Naduvath, S. Chandoor, S. J. Kalayathankal, J. Kok: *A Note on the Rainbow Neighbourhood Number of Certain Graph Classes*, National Academy Science Letters 42.2 (2019), pp. 135–138, doi: https://doi.org/10.1007/s40009-018-0702-6.

[7] S. Naduvath, S. Chandoor, S. J. Kalayathankal, J. Kok: *Some new results on the rainbow neighbourhood number of graphs*, National Academy Science Letters 42.3 (2019), pp. 249–252, doi: https://doi.org/10.1007/s40009-018-0740-0.

[8] D. B. West: *Introduction to graph theory*, vol. 2, Prentice hall Upper Saddle River, NJ, 1996.

Annales Mathematicae et Informaticae
***52** (2020) pp. 159–166*
DOI: https://doi.org/10.33039/ami.2020.09.001
URL: https://ami.uni-eszterhazy.hu

Ellipse chains inscribed inside a parabola and integer sequences

Giovanni Lucca

Piacenza, Italy
vanni_lucca@inwind.it

Submitted: July 11, 2020
Accepted: September 25, 2020
Published online: September 26, 2020

Abstract

The paper presents formulas and conditions relevant to the construction of chains of mutually tangent ellipses inscribed inside a parabola. Moreover, some connections with certain integer sequences and Pythagorean triplets are shown.

Keywords: Ellipse chains, parabola, integer sequences, Pythagorean triplets.

MSC: 51M04, 51M15

1. Introduction

In the previous paper [2], we studied the problem of inscribing a chain of mutually tangent circles inside a parabola; here we want to generalise it by considering the case of ellipses instead of circles.

We also mention that a cognate problem has been presented in [1] by considering a hyperbola instead of a parabola.

Let us consider a parabola in its simplest form that is:

$$y = ax^2, \quad a > 0.$$

This is not a limitation because, as known, the shape of the parabola depends only on the coefficient of the second order term; moreover, the main results presented in this paper do not change in the case when $a < 0$. The advantage in considering only the case $a > 0$ consists in obtaining simpler formulas.

Inside this parabola, we want to inscribe an infinite chain of ellipses where the generic i-th ellipse is tangent to the preceding and succeeding ones; see an example in Figure 1.

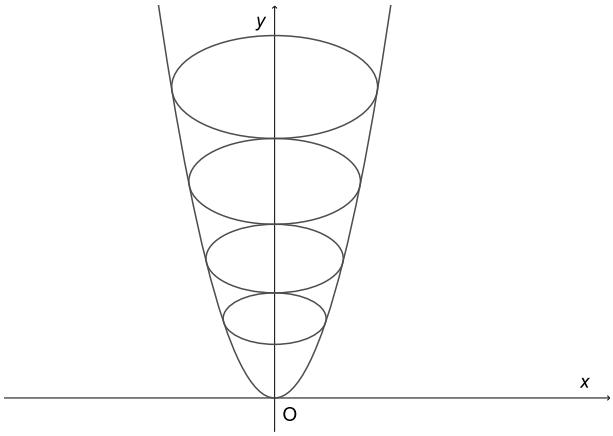


Figure 1: Example of ellipse chain inscribed inside a parabola

2. Construction of the ellipse chain

For symmetry reasons, the centre of each ellipse must be placed on the ordinate axis; thus, the centre of the generic i-th ellipse of the chain has coordinates $(0, Y_i)$.

Moreover, we define respectively by α_i and β_i the horizontal and vertical semi-axes of the generic i-th ellipse.

In the next subsections, we introduce the hypotheses adopted and the basic conditions needed to build up the ellipse chain.

2.1. Similarity of the ellipses

The first basic assumption we make is that all the ellipses forming the chain are similar that is

$$\lambda = \frac{\alpha_i}{\beta_i}, \quad \lambda \in \mathbb{R}^+, \ i = 0, 1, \dots \tag{2.1}$$

Note that it could be $\lambda < 1$; in that case, the major axis of the ellipses of the chain is the vertical one.

2.2. Tangency condition between to consecutive ellipses

By considering two consecutive ellipses of the chain, we have that the difference between the ordinate centres is equal to the sum of the vertical semi-axis that is

$$Y_i - Y_{i-1} = \beta_i + \beta_{i-1}, \quad i = 1, 2, \dots \tag{2.2}$$

2.3. Tangency condition between parabola and ellipses

In order to find the intersections between the parabola and the generic i-th ellipse, we have to consider the following equation system

$$\begin{cases} y = ax^2, \\ \dfrac{x^2}{\alpha_i^2} + \dfrac{(y - Y_i)^2}{\beta_i^2} = 1. \end{cases}$$

By solving with respect to y, one obtains

$$y = \frac{-\beta_i^2 + 2a\alpha_i^2 Y_i \pm \beta_i \sqrt{\beta_i^2 - 4a\alpha_i^2 Y_i + 4a^2\alpha_i^4}}{2a\alpha_i^2}. \tag{2.3}$$

In order that the ellipses of the chain are tangent to the parabola, we have, from equation (2.3), that the discriminant $\Delta = \beta_i^2 - 4a\alpha_i^2 Y_i + 4a^2\alpha_i^4$ must be zero; therefore the tangency condition is

$$\beta_i^2 - 4a\alpha_i^2 Y_i + 4a^2\alpha_i^4 = 0. \tag{2.4}$$

2.4. Condition relating λ, a and β_0

Even if we are considering only the case with $a > 0$, it is necessary to remark that by looking at equation (2.3), one has that the sign of the ordinates y_{Ti} of the tangency points (just given by equation (2.3) when equation (2.4) holds) between ellipses and parabola must be consistent with the sign of a; i.e., they must be positive when a is positive and vice-versa. Therefore, we must have

$$\begin{cases} y_{Ti} = \dfrac{-\beta_i^2 + 2a\alpha_i^2 Y_i}{2a\alpha_i^2} \geq 0 \text{ if } a > 0, \\ y_{Ti} = \dfrac{-\beta_i^2 + 2a\alpha_i^2 Y_i}{2a\alpha_i^2} \leq 0 \text{ if } a < 0. \end{cases} \tag{2.5}$$

In the case when $a > 0$, equation (2.5) is verified if

$$Y_i \geq \frac{1}{2a\lambda^2}, \quad i = 0, 1, \ldots \tag{2.6}$$

Clearly, if the following relationship holds

$$Y_0 \geq \frac{1}{2a\lambda^2}. \tag{2.7}$$

then also (2.6) is verified because the relation $Y_i \geq Y_0$ is always fullfilled. Neverthless, it must also be $Y_0 \geq \beta_0$ because, in order to have no intersections between the first ellipse and the parabola, the ordinate of the centre of the first ellipse cannot be smaller than its vertical semi-axis length; so, we can write the following relation

$$\min(Y_0) = \beta_0.$$

Thus, by considering the case $Y_0 = \beta_0$, from relation (2.7) we finally obtain

$$\frac{1}{\beta_0 a\lambda^2} \leq 2. \tag{2.8}$$

Condition (2.8) or equivalently

$$\frac{1}{\alpha_0 a\lambda} \leq 2$$

are the basic relationships, relating the parameters of the parabola and of the ellipse chain, that must be fulfilled in order to be able to construct the ellipse chain itself.

2.5. Recursive formulas

Let us consider equation (2.4); by means of (2.1) it can be written as

$$\beta_i^2 - 4a\lambda^2\beta_i^2 Y_i + 4a^2\lambda^4\beta_i^4 = 0.$$

Being $\beta_i \neq 0$, it can be simplified into

$$1 - 4a\lambda^2 Y_i + 4a^2\lambda^4\beta_i^2 = 0.$$

We also have

$$1 - 4a\lambda^2 Y_{i-1} + 4a^2\lambda^4\beta_{i-1}^2 = 0.$$

By subtracting the corresponding members of the two above equations, by means of equation (2.2) one gets

$$\beta_i = \beta_{i-1} + \frac{1}{a\lambda^2}, \quad i = 1, 2, \ldots \tag{2.9}$$

By substituting (2.9) into (2.2) one finally has

$$Y_i = Y_{i-1} + 2\beta_{i-1} + \frac{1}{a\lambda^2}, \quad i = 1, 2, \ldots \tag{2.10}$$

Equation (2.9) together equation (2.10) form a system of non homogeneous linear recursive relations that allow us to built the ellipse chain starting from the pair of initial values (β_0, Y_0) where β_0 must full-fill relation (2.8) and Y_0 is given by

$$Y_0 = a\lambda^2\beta_0^2 + \frac{1}{4a\lambda^2}$$

as one can deduce from (2.4) when $i = 0$.

Clearly, the values of α_i can be determined by remembering (2.1).

3. Some integer sequences associated to the ellipse chains

In this paragraph, we focus our attention on the particular chains characterised by the following relationship

$$Y_0 = \beta_0. \tag{3.1}$$

All these chains have in common the characteristic that the first ellipse is tangent to the parabola at its vertex (see Figure 2).

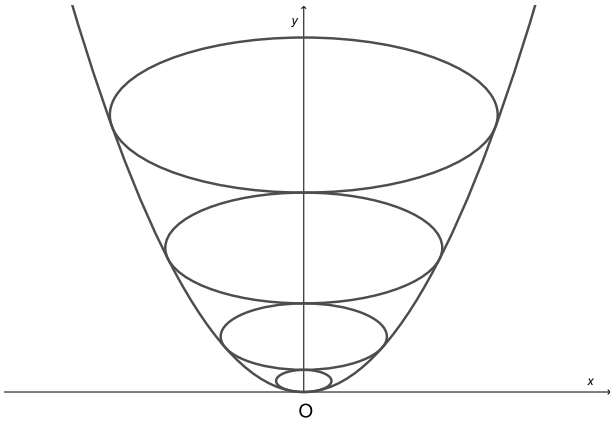


Figure 2: Example of ellipse chain with tangency point at the parabola vertex

Remark 3.1. In this case we have that

$$\frac{1}{\beta_0 a \lambda^2} = 2. \tag{3.2}$$

This kind of ellipse chains, as we shall see in the following, are in relation with certain integer sequences that do not depend neither on a, that is the shape of the parabola, nor on λ, that is the ratio between the ellipse semi-axes, but, on the contrary, they can be considered as common and invariant sequences to be related to the set of all parabolas with inscribed ellipse chains disposed as in Figure 2.

Let us introduce the following sequences $\{\overline{Y}_i\}$, $\{\overline{\alpha}_i\}$, $\{\overline{\beta}_i\}$ respectively defined as

$$\overline{Y}_i = \frac{Y_i}{Y_0}, \quad \overline{\alpha}_i = \frac{\alpha_i}{\alpha_0}, \quad \overline{\beta}_i = \frac{\beta_i}{\beta_0}.$$

Remark 3.2. By remembering equation (2.1) and from the definitions of $\{\overline{\alpha}_i\}$ and $\{\overline{\beta}_i\}$, one has:

$$\{\overline{\alpha}_i\} = \{\overline{\beta}_i\}. \tag{3.3}$$

Thus, in the following, we focus only on sequence $\{\overline{\beta}_i\}$.

We now derive some theorems related to the above introduced sequences.

Theorem 3.3. *Sequence* $\{\overline{\beta}_i\}$ *is the sequence of the odd numbers.*

Proof. By dividing both the members of equation (2.12) by β_0 and by taking into account equation (3.2) one gets

$$\overline{\beta}_i = \overline{\beta}_{i-1} + 2, \quad i = 1, 2, \ldots \tag{3.4}$$

By remembering that $\overline{\beta}_0 = 1$, from equation (3.4), it follows, by induction, that $\{\overline{\beta}_i\}$ is the sequence of the odd numbers. □

Sequence $\{\overline{\beta}_i\}$ is classified in the On-Line Encyclopedia of Integer Sequences OEIS [3] as A005408.

As far as sequence $\{\overline{Y}_i\}$ is concerned, the following theorem holds:

Theorem 3.4. *Sequence* $\{\overline{Y}_i\}$ *is the integer sequence* $\{2i^2 + 2i + 1\}$.

Proof. From equations (2.4) and (2.1) one obtains

$$Y_i = a\lambda^2\beta_i^2 + \frac{1}{4a\lambda^2}. \tag{3.5}$$

By dividing both the members of equation (3.5) by β_0 and by taking into account of equations (3.1) and (3.2) and of Theorem 3.3 one has

$$\overline{Y}_i = \frac{1}{2}(2i+1)^2 + \frac{1}{2} = 2i^2 + 2i + 1, \quad i = 0, 1, \ldots \tag{3.6}$$

which was to be proved. □

This sequence is classified in OEIS as A046092.

Let us consider now, the ordinates of the tangency points y_{Ti} of the ellipses to the parabola given by equation (2.5). From this equation, we have that y_{Ti} is given by

$$y_{Ti} = Y_i - \frac{1}{2a\lambda^2}, \quad i = 1, 2, \ldots \tag{3.7}$$

Then, we can define a further sequence $\{\overline{y}_{Ti}\}$ as follows

$$\overline{y}_{Ti} = \frac{y_{Ti}}{\beta_0}, \quad i = 1, 2, \ldots \tag{3.8}$$

and the following theorem holds:

Theorem 3.5. *Sequence* $\{\overline{y}_{Ti}\}$ *is the integer sequence* $\{2i^2 + 2i\}$.

Proof. From equations (3.7) and (3.8) we have

$$\overline{y}_{Ti} = \frac{Y_i}{\beta_0} - \frac{1}{2\beta_0 a\lambda^2}, \quad i = 1, 2, \ldots \tag{3.9}$$

By remembering equations (3.6) and (3.2), one finally has:

$$\overline{y}_{Ti} = 2i^2 + 2i, \quad i = 1, 2, \ldots \tag{3.10}$$

which was to be proved. □

The sequence $\{\overline{y}_{Ti}\}$ can be found in OEIS as well. It is classified as: A001844.

If we consider the area A_i of the i-th ellipse, it is given by

$$A_i = \pi\alpha_i\beta_i.$$

Thus, we can introduce another sequence $\left\{\overline{A}_i\right\}$ defined as

$$\overline{A}_i = \frac{A_i}{A_0}, \quad i = 0, 1, \ldots$$

By considering this sequence, we have the following theorem:

Theorem 3.6. *The sequence $\left\{\overline{A}_i\right\}$ is the integer sequence given by the square of the odd numbers.*

Proof. We have that $\overline{A}_i$ is given by

$$\overline{A}_i = \frac{\alpha_i}{\alpha_0}\frac{\beta_i}{\beta_0}, \quad i = 0, 1, \ldots$$

and from Theorem 3.3 and equation (3.3) it follows that

$$\overline{A}_i = (2i+1)^2, \quad i = 0, 1, \ldots \tag{3.11}$$

which was to be proved. □

This sequence is classified in OEIS as A016754.

The results here found, relevant to the integer sequences, are consistent with the ones appearing in [2] which are a particular case of the work here presented when $\alpha_i = \beta_i$, i.e., the ellipses degenerate into circles.

4. Relation with Pythagorean triplets

By looking at the sequences $\left\{\overline{\beta}_i\right\}$, $\{\overline{y}_{Ti}\}$ and $\left\{\overline{Y}_i\right\}$ for $i = 1, 2, \ldots$, they have a particular characteristic that puts them in relation with the primitive Pythagorean triplets.

In fact, the following theorem holds:

Theorem 4.1. *The sequences $\left\{\overline{\beta}_i\right\}$, $\{\overline{y}_{Ti}\}$ and $\left\{\overline{Y}_i\right\}$ for $i = 1, 2, \ldots$ form an infinite set of primitive Pythagorean triplets.*

Proof. By remembering that $\beta_i = 2i + 1$ and by using equations (3.6) and (3.10), one can immediately verify that:

$$\overline{\beta}_i^2 + \overline{y}_{Ti}^2 = \overline{Y}_i^2, \quad i = 1, 2, \ldots$$

so meaning that the corresponding terms of these sequences form a Pythagorean triplet; in particular, these Pythagorean triplets are also primitive.

In fact, we have that, for each i with $(i = 1, 2, \ldots)$, $\overline{y}_{Ti} = \frac{\overline{\beta}_i^2 - 1}{2}$ and $\overline{Y}_i = \frac{\overline{\beta}_i^2 + 1}{2}$. On the other hand, a well known algorithm, attributed to Pythagoras himself, allows to generate a primitive Pythagorean triplet starting from any odd integer number $2i + 1$; according to it, the primitive triplet is given by

$$\left(2i+1, \frac{(2i+1)^2 - 1}{2}, \frac{(2i+1)^2 + 1}{2}\right).$$

Being $\overline{\beta_i}$ an odd integer, we have that the triplet

$$\left(2i+1, \frac{(2i+1)^2 - 1}{2}, \frac{(2i+1)^2 + 1}{2}\right)$$

is identical to the triplet $\left(\overline{\beta_i}, \overline{y}_{Ti}, \overline{Y}_i\right)$ so deducing that it is primitive. □

Remark 4.2. Notice that for $i = 1$, the corresponding first three terms of the three above sequences form the basic primitive Pythagorean triplet $(3, 4, 5)$.

Acknowledgements. The author would like to thank the anonymous reviewer for the useful comments and suggestions that allowed to improve the paper.

References

[1] H. Belbachir, L. Németh, S. M. Tebtoub: *Integer sequences and ellipse chains inside a hyperbola*, Annales Mathematicae et Informaticae 52 (2020), doi: https://doi.org/10.33039/ami.2020.06.002.

[2] G. Lucca: *Integer sequences, Pythagorean triplets and circle chains inscribed inside a parabola*, International Journal of Geometry 8.1 (2019), pp. 22–31.

[3] N. J. A. Sloane: *The On-Line Encyclopedia of Integer Sequences*, url: https://oeis.org.

Annales Mathematicae et Informaticae
***52** (2020) pp. 167–181*
DOI: https://doi.org/10.33039/ami.2020.05.002
URL: https://ami.uni-eszterhazy.hu

Efficiently parallelised algorithm to find isoptic surface of polyhedral meshes

Ferenc Nagy

Faculty of Informatics, University of Debrecen, Hungary
Doctoral School of Informatics, University of Debrecen, Hungary
nagy.ferenc@inf.unideb.hu

Submitted: February 8, 2020
Accepted: May 1, 2020
Published online: May 14, 2019

Abstract

The isoptic surface of a three-dimensional shape is defined in [1] as the generalization of isoptics of curves. The authors of the paper also presented an algorithm to determine isoptic surfaces of convex meshes. In [9] new searching algorithms are provided to find points of the isoptic surface of a triangulated model in $\mathbb{E}^3$. The new algorithms work for concave shapes as well.

In this paper, we present a faster, simpler, and efficiently parallelised version of the algorithm of [9] that can be used to search for the points of the isoptic surface of a given closed polyhedral mesh, taking advantage of the computing capabilities of the high-performance graphics cards and using the benefits of nested parallelism. For the simultaneous computations, the NVIDIA's Compute Unified Device Architecture (CUDA) was used. Our experiments show speedups up to 100 times using the new parallel algorithm.

Keywords: Isoptic surface, CUDA, Parallel algorithm, Nested parallelism

MSC: 65D17, 68U07

1. Introduction

The isoptic curves in the Euclidean plane $\mathbb{E}^2$ have been widely studied since centuries. It is defined as the locus of points, from where a given curve can be seen under a predefined angle (of less than π). There are well-known results of several

classical curves [13]. However, the exact calculation of the isoptic curve may be a complicated task. For example, using direct computations, it is only possible to obtain it for low degree Bézier curves [6]. In such difficult cases, the points of the isoptic curve are determined by the appropriate tangents of the given curve, which meet at the given angle.

The isoptics in the three-dimensional space, besides the theoretical results, can also be of great interest in certain applications, which are concerned with the quality or quantity of visibility. However, the extension to $\mathbb{E}^3$ is not straightforward and the calculations are also getting more complicated. In [8] an algorithm was presented to find the isoptic curve of a Bézier surface to be used as a camera path. Despite the specific case, the exact equations seemed too difficult to solve, even for computer algebra systems. Only the numerical methods can determine the isoptic curve.

In [1], the isoptic in $\mathbb{E}^3$ is defined as a surface by substituting the two-dimensional viewing angle for the appropriate three-dimensional measure of visibility (solid angle). The authors are also provided a formula and algorithm for convex shapes, but it is possible to solve and plot the isoptic surface only using computer algebra systems. Moreover, it takes around 20–40 minutes to display it, even for simple regular polyhedra. In [9], a faster, general algorithm was presented to determine the isoptic surface of a given polyhedral mesh. These results, including the precise definitions, will be briefly summarized in Section 2.

The latter algorithm is able to find and render the isoptic surface in case of concave objects as well independently of computer algebra systems, but for a mesh with a few hundred polygons, the process still takes several minutes. Our aim is to accelerate the algorithm of [9] to find the isoptic surface within a reasonable time, using general-purpose computing on graphics processing units (GPGPU).

In the following sections, we present the simpler and parallel version of the algorithm. The different levels of parallelism will be discussed separately, in Section 3 and in Section 4. The running times of the new GPU-based methods will be compared with the original version of the algorithm presented in [9] in Section 5.

2. Previous results

In this section, we recall the notion of the isoptic surface, defined in [1] and briefly describe the earlier sequential algorithm, presented in [9] that obtains the isoptic surface of a closed polyhedral mesh.

The 3D generalization of the isoptics is based on the extension of the two-dimensional measure of angles. The angle at vertex A can be measured by the arc length on the unit circle around A. An appropriate substitution of the arc length in the Euclidean space can be the solid angle [2]:

Definition 2.1. The solid angle $\Omega_{\mathcal{S}}(P)$ subtended by a surface $\mathcal{S}$ is defined as the surface area of the projection of $\mathcal{S}$ onto the unit sphere around P.

Based on this notion the isoptic surface is defined in [1] as follows:

Definition 2.2. The isoptic surface $\mathcal{S}_{\mathcal{D}}^{\alpha}$ in $\mathbb{E}^3$ of an arbitrary 3-dimensional compact domain $\mathcal{D}$ is the locus of points P where the measure of the projection of $\mathcal{D}$ onto the unit sphere around P is equal to a given fixed solid angle value α, where $0<\alpha<2\pi$.

The algorithm, presented in [9] searches for spatial points around the mesh, where the solid angle is equal to a given value α. The solid angle is calculated at each point P as the area of the projection of the given model on the unit sphere, centered at P. The projection of a polyhedral mesh covers a spherical polygon on the unit sphere, the area of which can be calculated by the following formula:

$$\Omega(P) = \theta - (n-2)\pi, \tag{2.1}$$

where n is the number of the containing vertices and θ is the sum of the angles of the spherical polygon.

After calculating the solid angle, the isoptic surface of the model can be determined by finding the appropriate three-dimensional points, where the solid angle is equal to the given value α. The search for these points is done using the following methods, regarding [9]:

1. brute-force: one can scan the space around the model with a given increment and select the appropriate 3-dimensional points, where the solid angle differs from the given α with a suitable small (error) value.

2. flood-fill: in this search, we test the neighboring positions of a previously found isoptic point. The first point can be determined by shooting a ray outwards from the barycenter of the mesh.

3. spherical: it is based on the search of the first point of the flood-fill method, by shooting rays outwards from the barycenter of the mesh into many directions.

The result of the above algorithms is the point cloud of the isoptic surface of the given mesh. The comparison of the search methods is described in [9]. From the found points the isoptic surface can be constructed as a polygon model using mesh reconstruction algorithms (see Figure 1).

The running time of the algorithm highly depends on the speed of obtaining the spherical contour (and calculating the solid angle) and the swiftness of the used searching method. In Section 3, we present an alternative method to accelerate the computation of the solid angle at point P, using the graphical processing unit (GPU). The steps of the algorithm call special functions that run on the GPU. In CUDA they are called kernels. Each execution launches a specified number of thread blocks (a group of threads) and each thread performs the operation specified by the kernel function.

Besides the procedure that calculates the solid angle, the used search method can also be accelerated by parallel processing. Therefore, the algorithm to find the isoptic surface requires embedded kernel launches. This solution leads us to use multiple levels of parallelism. In CUDA it is called dynamic parallelism [3]. It enables the threads to create and synchronize new nested work. The new parallel versions of the searching methods will be discussed in Section 4.

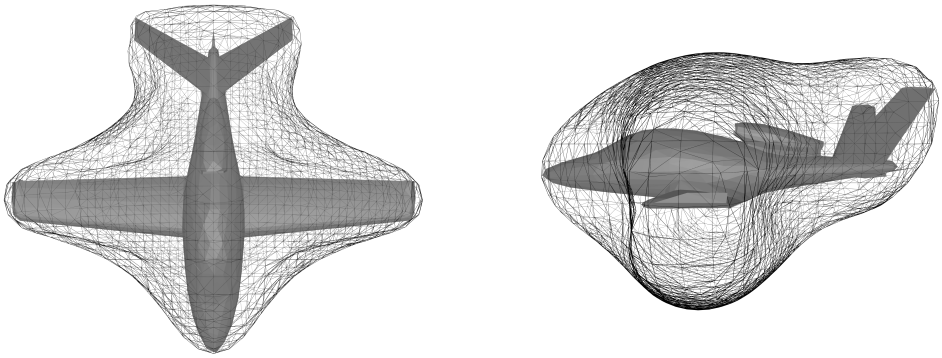


Figure 1: Isoptic surface of an airplane model, constructed from point cloud ($\alpha = \frac{\pi}{2}$, model source: `www.cadnav.com`)

3. Solid angle computation in parallel mode

Let $\mathcal{M}$ be a closed polyhedral mesh given in a half-edge data structure, in which each facet $\mathcal{F}$ is represented by a list of directed edges $\mathcal{E}$. The vertices of the edges belonging to the same facet are required to be in counterclockwise order to calculate the proper solid angle. Beside the endpoints (`V_1[3]`, `V_2[3]`), to determine faster the visible edges from a point P the normal vector (`other_normal[3]`) of the other facet that also contains the edge is stored as well:

```
edge = {
  V_1[3]          : float ;
  V_2[3]          : float ;
  other_normal[3] : float ;
}
```

Listing 1: Edge data structure

To make the computations easier, one can apply a coordinate transformation to place the origin into the center of the unit sphere (i.e. P). In this case, during the calculation of the solid angle of $\mathcal{M}$ at a point P we project the model onto an origin centered unit sphere.

The algorithm of [9] first determines the spherical boundary of the projection and then computes the spherical area of it. The new method focuses on the calculation of the solid angle, rather than obtaining the spherical contour. The computation requires four steps. At first, one needs to project the edges of $\mathcal{M}$ onto the unit sphere, then calculate the spherical angles at all the intersections. After, to summarize the appropriate spherical angles, it is required to find a proper starting point. The last step is the traversing of the consecutive (counterclockwise ordered)

edges at their intersections. It is done by selecting always that edge, which intersects the given edge closer to its first vertex or which meet at the same position but has a higher spherical angle at the intersection. Finally, the solid angle is obtained using Eq. (2.1).

The following subsections will describe the steps in detail and specify the technique and data structures needed for the parallelisation. The final pseudo-code of the algorithm is shown in Listing 4.

3.1. Projecting edges

To calculate the solid angle on the unit sphere it is necessary to project the edges of the facets visible from P. To avoid projecting the edges containing the same endpoints with the opposite direction we project the edges which belong to one facet visible from P and to another that hidden from P. The selection of the appropriate edges is done efficiently, using the `other_normal[3]` data of Listing 1. In this step, we create a new array $\mathcal{S}$ that consists of the projected edges, which are great arc segments on the origin centered unit sphere. The elements of $\mathcal{S}$ have the following structure:

```
spherical_edge = {
  A[3]                      : float ;
  B[3]                      : float ;
  dist_from_center          : float ;
  intersection_id[max_int] : integer ;
  n_int                     : integer ;
}
```

Listing 2: Spherical edge data structure

The projected endpoints `A[3]` and `B[3]` remain stored as points of $\mathbb{E}^3$. The `dist_from_center` is the distance between `B[3]` and the projection of the barycenter of $\mathcal{M}$. It is required for the subsequent step of the algorithm but the values are computed in this stage. It can be calculated as spherical or Euclidean distance. To avoid using trigonometric functions, the latter is preferred. The rest of Listing 2 will be explained in the further steps of the algorithm.

It is possible to fill $\mathcal{S}$ in parallel since the faces can be processed independently. However, for the preceding memory allocation, it is necessary to approximate the expected size of $\mathcal{S}$ (see `max_S` in Listing 4). For simplicity, one can multiply the number of facets of $\mathcal{M}$ by the number of edges of a facet (which is three in case of triangulated meshes) to obtain a limit of $\mathcal{S}$. However, a better approximation is recommended to preserve the memory. The authors of [5] have made a probabilistic analysis of the expected number of the contour edges with respect to a random viewing direction, which can be used to approximate the size of $\mathcal{S}$. Therefore, the expected number of the contour edges is calculated by the following formula [5]:

$$\sum_{e\in\mathcal{E}} 1-2\phi_e, \quad \text{where } \phi_e = \frac{1}{2\pi}\arccos\frac{-\vec{n_{f_i}}\cdot\vec{n_{f_j}}}{|-\vec{n_{f_i}}||\vec{n_{f_j}}|}. \tag{3.1}$$

The $\vec{n_{f_i}}$ and $\vec{n_{f_j}}$ are the normal vectors of faces f_i and $f_j \in \mathcal{F}$ and the ϕ_e is the probability that the facets incident to $e \in \mathcal{E}$ are both front or both back facets.

In addition, the further steps of the algorithm require the exact size $n_{\mathcal{S}}$ of $\mathcal{S}$, therefore it needs to be counted during the filling. It can be is done using atomic increment operation that reads the value at a specified address of the memory, adds a number to it (in this case one), and writes the result back to the same address. The atomic means that, it is guaranteed to be performed without interference from other threads [11].

3.2. Calculating intersections

In this step one needs to calculate the spherical angles of all the overlapping edge pairs e and f of $\mathcal{S}$ at the intersections and when they meet at an opposite endpoint ($\texttt{A[3]}_e = \texttt{B[3]}_f$ or $\texttt{B[3]}_e = \texttt{A[3]}_f$). This stage is done in a brute-force manner, in parallel (consider all pairs of spherical segments and test each pair for intersection). The intersection point of two projected edge is computed using the formula described in [9]. However, there are faster line segment intersection algorithms for GPU (such as [12]), considering the nested parallelisation, the much simpler brute-force manner is recommended.

Besides the spherical angles, the algorithm requires other data as well. If there is an intersection between e and f, we set two elements in the intersection array $\mathcal{I}$, using the following structure:

```
intersection = {
  angle       : float ;
  other_edge  : integer ;
  dist_from_A : float ;
}
```

Listing 3: Intersection data structure

One element, which corresponds to edge e stores the spherical angle between e and f. To calculate it (up to 2π), the formula presented in [9] was used. The `other_edge` is the index j of edge f in $\mathcal{S}$. The `dist_from_A` is the distance between the intersection point and the first endpoint (`A[3]`) of edge e. It can also be calculated as spherical or Euclidean distance. The other element that corresponds to edge f stores the 2π - `angle`, the index i of e, and the distance between the intersection point and the first endpoint of f.

The above calculations are processed in parallel, using one thread for each e and f pairs. The $\mathcal{I}$ is stored in the global memory as a one-dimensional array. The size of it should be the number of the expected size of the projected edges on the square to avoid array access conflicts. The appropriate indices in $\mathcal{I}$ of the element e and f is calculated using their indices i, j and the size $n_{\mathcal{S}}$. The position of e is $n_{\mathcal{S}} \cdot i + j$ and f is $n_{\mathcal{S}} \cdot j + i$.

The element indices of $\mathcal{I}$ are also needed to be stored locally in the corresponding edge structure (see `intersection_id[max_int]` in Listing 2). Since one spherical

segment can cross multiple others, it also needs to be stored as an array. The size `max_int` of it should be estimated previously for the memory allocation. In [5], there is a formula also for the expected number of edge intersections. The detailed calculation of the probability that two edges are crossing with respect to a random viewing direction is described in Section 3. of [5]. To obtain `max_int`, one needs to find that edge pair that has the highest likelihood. In addition, the coincident endpoints of the edges also need to be considered, since they are also stored as intersections. To take it into account, one needs to find that vertex position, where the most facets meet. The number of the incident facets at this vertex should be added twice to consider the intersection at both endpoints of an edge.

The actual number of the intersections, i.e. the size `n_int` of the local array (in Listing 2) is counted similarly as in the case of $\mathcal{S}$, using atomic increment operation.

3.3. Finding the first edge

To begin the traversal of $\mathcal{S}$ one needs to determine a proper starting spherical segment e that is a contour edge of $\mathcal{S}$. It is selected by its first endpoint $\texttt{A[3]}_e$, which should be the farthest away from the projection C of the barycenter of $\mathcal{M}$. However, it is also necessary that the vertex of the mesh that corresponds to $\texttt{A[3]}_e$ is visible from P and not covered by any facet of $\mathcal{M}$. It can be seen, if the spherical arc segment $\overset{\frown}{C'_{\texttt{A[3]}_e}}$, formed by the antipode of C and $\texttt{A[3]}_e$, does not intersect with any edge of $\mathcal{S}$. The above conditions can also be satisfied by the first endpoint of an interior silhouette of $\mathcal{S}$. Therefore, to obtain a contour spherical segment e, one has to select the edge that has the highest spherical angle $\sphericalangle C'_{\texttt{B[3]}_e\texttt{A[3]}_e}$.

In $\mathcal{I}$, besides the intersections, the coincident opposite endpoints are also stored. Therefore, let us find the farthest endpoint $\texttt{B[3]}_f$ from C, using `dist_from_center` of Listing 2. In this way, the starting edge e with the highest spherical angle $\sphericalangle C'_{\texttt{B[3]}_e\texttt{A[3]}_e}$ is found faster since $\mathcal{S}$ is not processed again because the indices of the possible starting edges are obtained using the local `intersection_id[max_int]` array of f.

In this step, the visibility test is processed simultaneously. In the iteration of obtaining the farthest endpoint $\texttt{B[3]}_f$ from C' the spherical arc segment $\overset{\frown}{C'_{\texttt{B[3]}_f}}$ is tested for intersection with the other edges of $\mathcal{S}$ in parallel.

3.4. Calculating the sum of the spherical angles

The final task is to summarize the appropriate spherical angles by traversing $\mathcal{S}$ and calculate the solid angle using Eq. (2.1). It can only be done by an iterative loop, that begins with the selected starting arc segment.

One has to go along its intersections using the local array `intersection_id[max_int]` of Listing 2 and select the subsequent edge by comparing the distances between its intersections and the first endpoint. The distance values are computed in `dist_from_A` of Listing 3. The following edge is that, which has the closest intersection. Its index is stored in `other_edge`. If a spherical arc segment has more

than one intersection at the same position (in cases when the `dist_from_A` values are equal) the one with the highest spherical angle needs to be considered.

The search of the minimum `dist_from_A` of an edge f does not necessarily start from zero but from a minimum value, based on where the earlier edge e is connected. In case of an intersection between e and f two elements are added to $\mathcal{I}$. Each stores the distance from the first endpoint and the cross point. Therefore, the minimum value can be obtained by finding the corresponding members in $\mathcal{I}$. If the index of the intersection element of the edge e in $\mathcal{I}$ is i, then the corresponding element index j of $\mathcal{I}$ that belongs to the next edge f is calculated as $n_{\mathcal{S}}(i \bmod n_e) + (i/n_{\mathcal{S}})$, where $n_{\mathcal{S}}$ is the size of $\mathcal{S}$.

The iteration ends when the loop reaches again the first edge. During the traversal, the appropriate spherical angles are added directly to θ of Eq. (2.1) since the values are already calculated.

Regarding [5], one contour edge has only a few numbers of intersections (`n_int`), therefore, it is not necessary to sort them since it can be traversed fast enough iteratively. The complete pseudo-code of the solid angle computation is shown in Listing 4.

4. Search for the isoptic points in parallel mode

In the preceding sections, the new parallel method is described to compute the solid angle and decide when the point P in $\mathbb{E}^3$ is a point of the isoptic surface of $\mathcal{M}$. Besides the speed, the simplicity of the algorithm is also important, since we intend to calculate it for multiple points at the same time, using the new parallel search methods. In this case, all the computations to obtain the point cloud of the isoptic surface are embedded into one kernel function call, which entirely handled by the graphical processor. The procedures to calculate the solid angle at the specific 3-dimensional points are nested parallel works. The new approach is efficient because the process is not interrupted by memory management operations. All the required space can be allocated and all the required data can be loaded into the memory previously.

To search for the isoptic points in parallel the following modifications need to be performed:

1. brute-force: one has to divide the space around $\mathcal{M}$, where the points are searched into a discrete set of cubes (see marching cubes in [7]) and process the containing points of the cubes at the same time.

2. spherical: in this case, one can do the search in many directions simultaneously from the barycenter of the vertices of $\mathcal{M}$. Each thread can process one direction.

Unfortunately, the flood-fill search can not be accelerated using the GPU. In the case of the above methods, the parallelisation was feasible and straightforward. The main difficulty of the flood-fill algorithm, even in the sequential CPU case is

to keep track of the previously visited positions. In the CPU version, to speed up the process a binary search tree can be used. However, there are also algorithms to build search trees parallel on the GPU (e.g. [10]) along with the new solid angle computation the isoptic surface determination is slower than the CPU version algorithm.

The algorithms running on the GPU are producing the same isoptic surfaces as the CPU versions since the base stages of the algorithms are the same (calculating the solid angle and the search for the three-dimensional points).

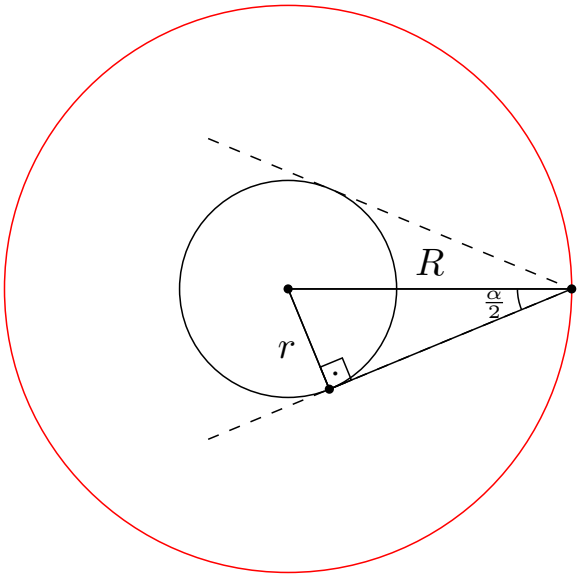


Figure 2: Radius R of the isoptic of the circle

In the case of the brute-force method the space around the model which needs to be traversed can be defined using a minimum bounding sphere of $\mathcal{M}$. The isoptic surface of this enclosing sphere is also a sphere. Its radius R is calculated similarly as the radius of the two-dimensional isoptic of the circle (see Figure 2):

$$R = r / \sin \frac{\alpha}{2},$$

where r is the radius of the bounding sphere. The radius R defines the maximum distance from the mesh, where the isoptic points of $\mathcal{M}$ are (See Figure 3). This maximum distance can also be used in case of the spherical method to limit the search.

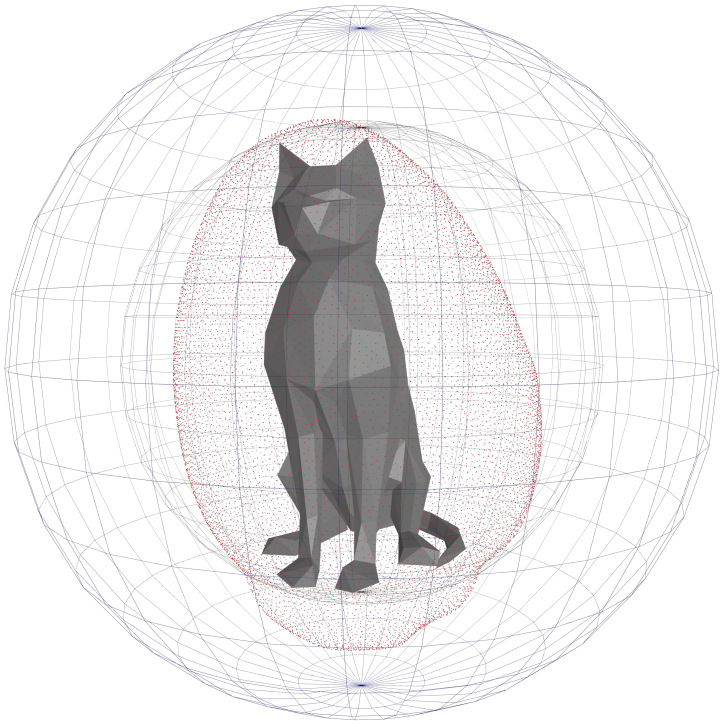


Figure 3: The points (red dots) of the isoptic surface of the cat model are inside the isoptic (blue sphere) of the enclosing sphere (model source: `www.turbosquid.com`)

5. Performance analysis

Table 1 shows the execution time of the original CPU-based algorithms of [9] and the new GPU-based methods, using the following parameters: $\alpha = \frac{\pi}{2}$, the size of the traversal step is 0.1, and a point P is accepted if the difference between the calculated solid angle at point P and α is less than 2×10^{-3} (error). All the tested models are scaled to have radius $r = 5$ of the bounding sphere, which is calculated as the distance of the farthest vertex from the barycenter of the mesh. Therefore the same radius $R = 5/\sin(\pi/4)$ was used for all the meshes. The experiments were run on an Intel Core i7-7700HQ and Geforce GTX 1050 Ti with CUDA version 10.1. The algorithms were using single-precision arithmetic. The results can be seen in Figure 4. The isoptic surfaces around the tested objects are displayed as wireframe models created from the found point cloud using mesh reconstruction algorithm [4].

The execution times are generally increasing according to the complexity of the

Model	**Brute-force**		**Spherical**	
	Sequential	Parallel	Sequential	Parallel
Stanford Bunny F: 128, V: 66	499.1	10.6	24.2	0.9
Cat F: 428, V: 216	2279.9	36.5	117.4	2.9
Moose F: 747, V: 376	4233.7	92.2	228.9	8.8
Airplane F: 910, V: 529	6449.4	63.8	418.3	5.1
Elephant F: 1492, V: 779	9185.6	163.5	453.5	13.6

Table 1: Execution times (in seconds) of the previous sequential and the new GPU-based parallel searching algorithms ($\alpha = \frac{\pi}{2}$, step size $=$ 0.1, solid angle error $= 2 \times 10^{-3}$, F and V are the numbers of the faces and vertices of the models)

meshes. However, in the case of the airplane model, which consists of more faces than the moose model, the parallel algorithm finds the isoptic surface within a shorter time. The reason behind is the number of the threads running in parallel. The expected size of $\mathcal{S}$ is estimated using Eq. (3.1). This calculation is based on the probability that both facets sharing the same edge are front faces with respect to a random viewing direction. On the wings of the airplane model, there are numerous coincident front facets from many positions, which imply the small number of the expected contour edges. Therefore, more threads can search in parallel because the fewer number of contour edges indicates the larger size of the marching cubes. It causes lower execution times for the airplane model in case of the GPU algorithms.

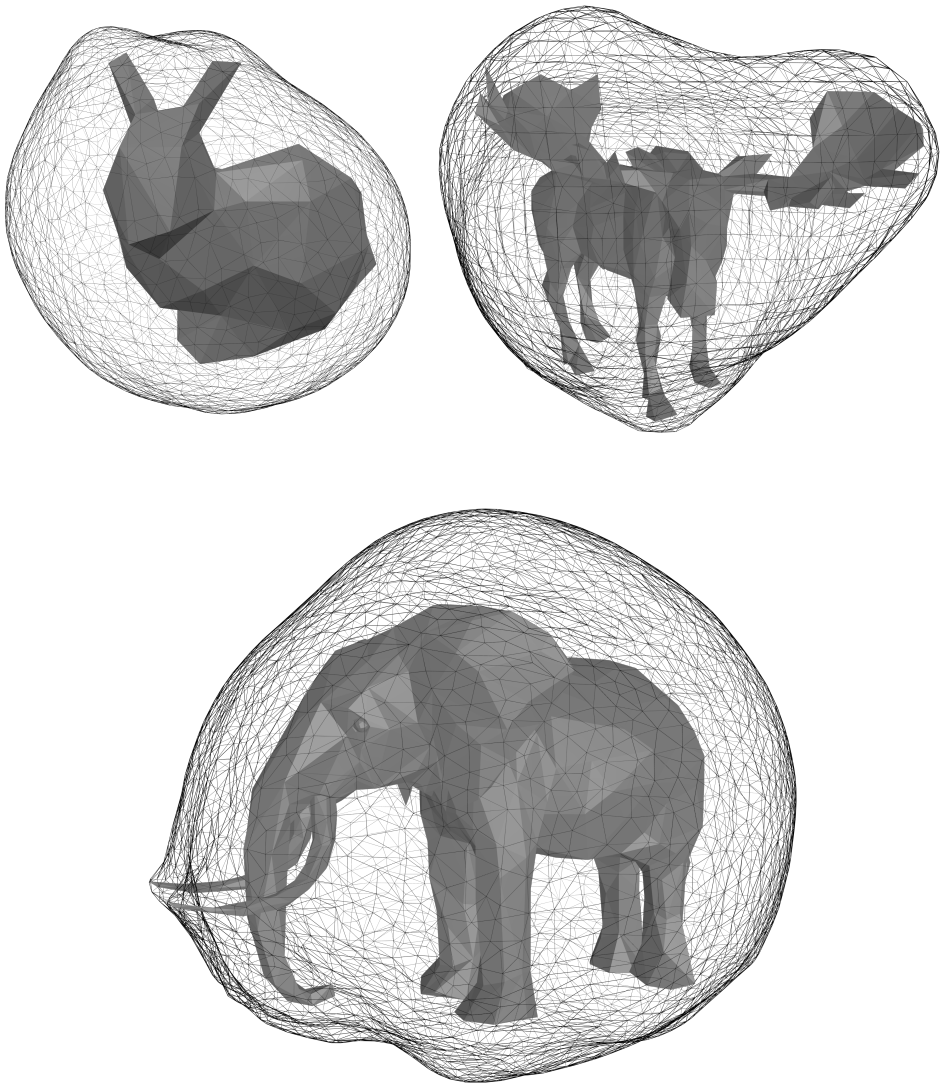


Figure 4: Isoptic surfaces of the tested models (Stanford Bunny, a moose and an elephant), constructed from point cloud ($\alpha = \frac{\pi}{2}$, model sources: `graphics.stanford.edu/data/3Dscanrep`, `www.cadnav.com`)

6. Summary

As can be seen from Table 1, the search of the isoptic surface is highly accelerated using the new parallel algorithms. In case of simple meshes, it effects greatly for the whole isoptic surface obtaining process. However, in case of complex meshes, the parallelism of the searching methods is limited, because the solid angle computation consumes more GPU resources (memory space and threads as well). Therefore, fewer points are searched in parallel.

To render the isoptic surface of a highly detailed polyhedral mesh (with thousands of facets) using the new GPU algorithms can still be time-consuming. A simple way to increase the speed is to search for the spatial isoptic points after the decimation of the given model, because even significant polygon reduction of the mesh can cause only slight decrease of the area of its projection, which is negligible, regarding the acceptance error between the given α and the computed solid angle.

Acknowledgements. This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

References

[1] G. Csima, J. Szirmai: *Isoptic surfaces of polyhedra*, Computer Aided Geometric Design 47 (2016), pp. 55–60,
doi: https://doi.org/10.1016/j.cagd.2016.03.001.

[2] R. Gardner, K. Verghese: *On the solid angle subtended by a circular disc*, Nuclear Instruments and Methods 93.1 (1971), pp. 163–167,
doi: https://doi.org/10.1016/0029-554X(71)90155-8.

[3] S. Jones: *Introduction to dynamic parallelism*, Nvidia GPU Technology Conference (GTC), May 2012.

[4] M. Kazhdan, H. Hoppe: *Screened Poisson Surface Reconstruction*, ACM Transactions on Graphics (TOG) 32.3 (2013), p. 29,
doi: https://doi.org/10.1145/2487228.2487237.

[5] L. Kettner, E. Welzl: *Contour edge analysis for polyhedron projections*, in: Geometric Modeling: Theory and Practice, ed. by W. Strasser, R. Klein, R. Rau, Springer, 1997, pp. 379–394,
doi: https://doi.org/10.1007/978-3-642-60607-6_25.

[6] R. Kunkli, I. Papp, M. Hoffmann: *Isoptics of Bézier curves*, Computer Aided Geometric Design 30.1 (2013), pp. 78–84,
doi: https://doi.org/10.1016/j.cagd.2012.05.002.

[7] W. E. Lorensen, H. E. Cline: *Marching cubes: A high resolution 3D surface construction algorithm*, ACM SIGGRAPH Computer Graphics 21.4 (1987), pp. 163–169,
doi: https://doi.org/10.1145/37402.37422.

[8] F. Nagy, R. Kunkli: *Method for computing angle constrained isoptic curves for surfaces*, Annales Mathematicae et Informaticae 42 (2013), pp. 65–70.

[9] F. Nagy, R. Kunkli, M. Hoffmann: *New algorithm to find isoptic surfaces of polyhedral meshes*, Computer Aided Geometric Design 64 (2018), pp. 90–99,
doi: https://doi.org/10.1016/j.cagd.2018.04.001.

[10] N. Nakasato: *Implementation of a parallel tree method on a GPU*, Journal of Computational Science 3.3 (2012), pp. 132–141,
doi: https://doi.org/10.1016/j.jocs.2011.01.006.

[11] NVIDIA: *NVIDIA CUDA C programming guide*, 2019.

[12] C. Rüb: *Line-segment intersection reporting in parallel*, Algorithmica 8.1-6 (1992), pp. 119–144,
doi: https://doi.org/10.1007/BF01758839.

[13] R. C. Yates: *A Handbook on Curves and their Properties*, Ann Arbor, J.W. Edwards, 1947, pp. 138–140.

Appendix

```
float CALCULATE_SOLID_ANGLE(P[3], faces)
  spherical_edge S[max_S] // max_S approximated
  n_S = PROJECT_EDGES(*S, faces)
  CALCULATE_INTERSECTIONS(S, n_S)
  float min = float_max
  spherical_edge f, first
  foreach spherical_edge e of S do
    if (e.dist_from_center < min AND IS_VISIBLE(e))
      f = e
      min = e.dist_from_center
    endif
  endforeach
  min = 0
  spherical_edge C_fB // from the antipode of C to f.B
  foreach index i of f.intersection_id do
    if ((I[i].dist_from_a=length(f)) AND // it is at f.B
        (spherical_angle(Cf_B, S[I[i].other_edge])>min))
      first = S[I[i].other_edge]
      min = spherical_angle(Cf_B, S[I[i].other_edge])
    endif
  endforeach
  spherical_edge current_edge = first
  integer n = 0 // number of traversed edges
  float theta, min_dist = 0
  repeat
    integer int_id
    float min_angle = 0, max_dist = 2 // Euclidean distance
    foreach index i of current_edge.intersection_id do
      if (((I[i].dist_from_a > min_dist) OR
           ((I[i].dist_from_a = min_dist) AND
            (I[i].angle > min_angle))) AND
          (I[i].dist_from_a < max_dist))
        int_id = i
        max_dist = I[i].dist_from_a
        min_angle = I[i].angle
      endif
    endforeach
    theta = theta + I[i].angle
    min_dist = I[n_S * (int_id%n_S) + (int_id/n_S)].dist
    current_edge = S[I[int_id].other_edge]
```

```
    n = n + 1
  until (current_edge != first)
  return (theta - ((n - 2) * PI))
end CALCULATE_SOLID_ANGLE
```

Listing 4: Solid angle computation method

Annales Mathematicae et Informaticae
***52** (2020) pp. 183–197*
DOI: https://doi.org/10.33039/ami.2020.01.001
URL: https://ami.uni-eszterhazy.hu

Analysing the vegetation of energy plants by processing UAV images*

Melinda Pap, Sándor Király, Sándor Molják

Eszterházy Károly University
(pap.melinda,kiraly.sandor,moljak.sandor)@uni-eszterhazy.hu

Submitted: June 21, 2019
Accepted: January 4, 2020
Published online: January 9, 2020

Abstract

Bioenergy plants are widely used as a form of renewable energy. It is important to monitor the vegetation and accurately estimate the yield before harvest in order to maximize the profit and reduce the costs of production. The automatic tracking of plant development by traditional methods is quite difficult and labor intensive. Nowadays, the application of Unmanned Aerial Vehicles (UAV) became more and more popular in precision agriculture. Detailed, precise, three-dimensional (3D) representations of energy forestry are required as a prior condition for an accurate assessment of crop growth. Using a small UAV equipped with a multispectral camera, we collected imagery of 1051 pictures of a study area in Kompolt, Hungary, then the Pix4D software was used to create a 3D model of the forest canopy. Remotely sensed data was processed with the aid of Pix4Dmapper to create the orthophotos and the digital surface model. The calculated Normalized Difference Vegetation Index (NDVI) values were also calculated. The aim of this case study was to do the first step towards yield estimation, and segment the created orthophoto, based on tree species. This is required, since different type of trees have different characteristics, thus, their yield calculations may differ. However, the trees in the study area are versatile, there are also hybrids of the same species present. This paper presents the results of several segmentation algorithms, such as those that the widely used eCognition provides and other Matlab implementations of segmentation algorithms.

Keywords: photogrammetry, 3D reconstruction, segmentation, NDVI.

*The research was supported by the grant EFOP-3.6.1-16-2016-00001 ("Complex improvement of research capacities and services at Eszterházy Károly University").

1. Introduction

Energy plants are important for preserving the Earth's ecology and as alternative energy sources like bio-fuel. They play an important role both in producing biofuel and heating electricity-generating power stations. There are plenty of tree species (woody plants) that can be planted as energy plants, for example Gray Poplar (Populus canescens), White Poplar (Populus alba) or Red oak (Quercus rubra).

Precise and detailed three-dimensional (3D) representations of the forestry area are very important for an accurate assessment of their volume and growth. Until recently, measuring the volume, spatial arrangement and shape of trees with precision has been constrained by logistical and technological limitations and cost. Traditional methods of plant biometrics provide merely partial measurements and these methods are labor intensive. Advances in Unmanned Aerial Vehicle (UAV) technology has made it feasible to obtain high-resolution imagery and three dimensional (3D) data using lightweight and inexpensive cameras. These are esential for energy plants monitoring and assessing tree attributes automatically [18].

In order to monitor and control the vegetation of plants and to measure their volume, it is necessary to create a 3D model from the UAV recorded 2D images. For processing huge amounts of imaging data, there are two frequently applied approaches: structure-from-motion (SfM) and multi-view stereopsis (MVS). Both can operate without information on the 3D position of the camera or the 3D location of control points.

SfM is a cost-effective method for extracting the 3D model of a scene from multiple overlapping images using bundle adjustment procedures [17]. It can generate high quality 3D point clouds for characterizing forest structures and can be used to generate accurate Digital Surface Models (DSMs) from the 3D point clouds. The 3D representation of the surface of a terrain and DEM (Digital Elevation Model) is a subset, and the most fundamental component of DSM [4, 15, 21]. The DSM is essential in creating an orthophoto of the whole scanned area as a geometrically corrected uniform-scale photograph, it is possible to use it for measurements. The success of SfM is controlled by image resolution, the degree of image overlap, as well as the relative motion of the camera with respect to the scene [25]. Photos created by an UAV are ideal for SfM since UAV fly only a few tens of meters above the ground, providing data with high spatial resolution that is better than spaceborne sensors. Therefore, UAVs have the potential of resolving the measurement of individual trees and plants for biomass estimation[24].

Image segmentation is the process of separating or grouping an image into different image objects. An image object is a group of connected pixels in a picture, where the objects are homogeneous with respect to specific features. These features can be represented by the RGB values, textures or gray-levels, each encoding similarities between the pixels of a region. Other segmentation methods focus on finding boundaries between regions. There are many different ways of performing image segmentation, ranging from the simple thresholding method to different colour image segmentation algorithms. In this paper, we aimed to find the best seg-

mentation algorithm for the purpose of segmenting an energy forest where hybrids of the same species are present. The goal is to find a method that is efficient but also robust in the sense that it is not strongly dependent on its input parameters.

Hossain and Chen [16] conducted an extensive state-of-the-art survey on OBIA (Segmentation for Object-Based Image Analysis) techniques, discussed different segmentation techniques and their applicability to OBIA. This article shows, that Ming et al. [12] implemented MeanShift algorithm for QuickBird (high-resolution remote sensing imagery) and panchromatic images, Maurer [13] for cropland and Michel et al. [8] for multiple objects but none of them targeted energy plants. Li et al. [14] implemented SRM segmentation method for QuickBird imagery as well.

Csillik [3] proposed a segmentation workflow where MRS algorithm started from superpixels instead of individual pixels. He also used the quickBird dataset and reached accuracy above 90%. Instead of using a single scale for the entire image, Fonseca-Luengo et al. [6] offered a hierarchical multiscale segmentation using superpixels (SLIC) which allowed users to detect objects at different scales. They used a satellite image that was collected by the Pléiades Satellite, in the central irrigated valley of Chile. Several researchers applied Region growing/merging, Region splitting and merging, Hybrid method (HM) and Semantic methods for different targets: road and agricultural land, fozen oil, sand ore, mixed vegetation, etc. but none for hybrid energy plants.

Tsouros et al. [23] reviewed the UAV-Based Applications for Precision Agriculture. This article shows that Zhao et al. [26] implemented segmentation method for targeting canopy pixels of pomegranate trees by using a fully convolutional neural network. Their tests on validation set showed that its precision reached above 90% and it was robust to changes in camera settings, lighting condition, canopy development and changing background. They worked with LIDAR imagery. Hassain et al. [10] has introduced a new vegetation segmentation approach which aims to generate vegetation binary images from RGB images acquired by a lowcost UAV system. They reached 87.29% with standard deviation 12.5%, in the detection of any type of vegetation in an area. Their study did not involve distinguishing the types of vegetation. Parraga et al. [22] used an algorithm to segment wheat plots for two kinds of Brazilians wheat cultivates.

2. Methods

2.1. Study area

The study area is located in Kompolt, at the Rudolf Fleischman Research Institute of the Eszterhazy Karoly University (47.735889 N, 20.224807 W, see Fig. 1).

A variety of three species (willows, acacias, poplars) and six hybrids of poplar are present in this study area.

Figure 1: Satellite image of the study area from Google Earth

2.2. Data Acquisition and Processing

For this study, the aerial survey was conducted on 6 September 2018 using a DJI Inspire 2 quadcopter. Besides the built-in high-precision satellite positioning system, ultrasonic, infrared and optical sensors help the machine to navigate and operate autonomously. An IMU (Inertial Measurement Unit), compass and barometer also improved the navigation. The true colour sensor was a Zenmus X5S. Using a 4/3 inch CMOS sensor, the camera can produce images of a resolution of 20.8 megapixels. The camera was equipped with a 15 mm F/1.7 lens that has a FoV value of 72°. Multispectral images were taken with a Parrot Sequoia camera. The sensor has 4 multispectral sensors, capable of a resolution of 1.2 MP, and a 16 MP RGB sensor. In addition, the Sequoia has a sun sensor that eliminates changes in ambient light intensity. The camera and the sun sensor have also a built-in IMU and magnetometer, and the sun sensor also contains an integrated GPS. We applied the Altizure flight planning software to create multispectral subsides, since the application also supports individual cameras that are not directly connected to the drone. The Altizre was adapted to the unique viewing angle of the multispectral sensor, so the overlaps between the lines were accurate. The ground switch points were measured with a 216 channel GPS + Glonass signaling dual frequency Javad Triumph 2 GNSS rover. The GNSS receiver was controlled by a Carlson SurvPC installed on a Juniper Systems Mesa2 tablet computer. The RTK correction service was provided by `www.gnssnet.hu`.

2.2.1. RGB images

The flight altitude was 100 meters and the total flight time was 13 minutes 21 seconds. The overlap in the flight range was 90%, while the overlap between the two lines was 75%. The drone flew the hand-selected area of approximately 326 × 433 meters at the set height and overlap in 7 flight lines. During the flight 263 images were taken in orthogonal camera positions. The georeference was specified with 7 ground connection points (see Fig. 2).

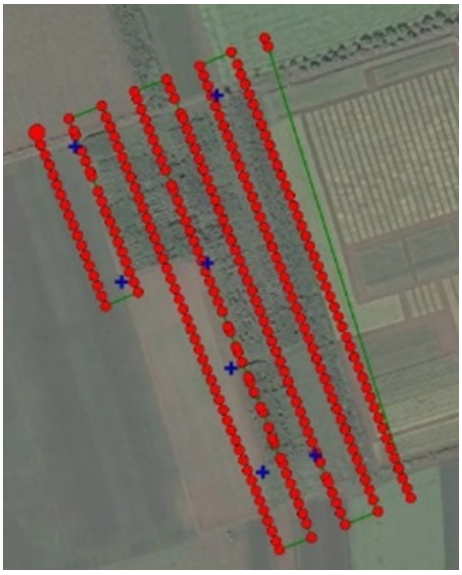


Figure 2: The flight path of the above study area and the Ground Control Points (GCPs)

2.2.2. Multispectral images

The multispectral investigation of the study area was performed at 70 meters and with a 70% overlap. The drone flew 8 flight lines at 6 m/s. The georeference was refined with 7 ground connection points. During the flight 788 images were taken from 197 positions and in 4 channels (Green, Red, Red edge, NIR) in an orthogonal camera position (see Fig. 3).

2.2.3. 3D reconstruction

The essence of 3D reconstruction is the assumption that the 3D point corresponding to a specific image point is constrained to the line of sight. Taking two images, we know that a 3D point that is present on both of the images is located at the intersection of the two projection rays. This process is also known as triangula-

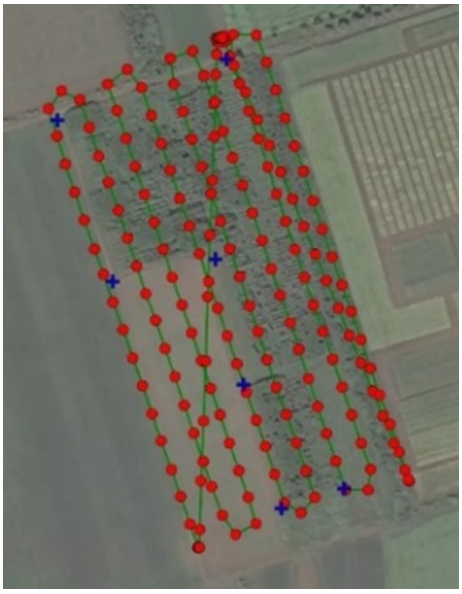


Figure 3: The flight path above the study area and Ground Control Points (GCPs) – multispectral images

tion. Furthermore, we can conclude that corresponding sets of points must have a relationship that is related to the positions and the calibration of the camera.

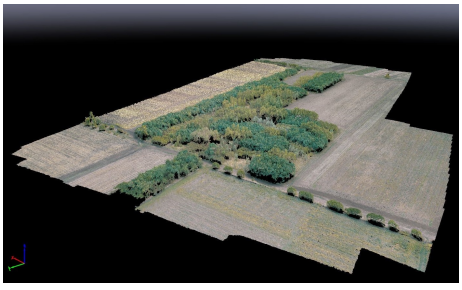


Figure 4: The 3D point cloud obtained from the RGB images

As a result of the photogrammetric processing, the average field resolution was 2.3 cm/pixel in the case of the RGB 3D model and the total processed area was 12.8 hectares. The number of the average key points was 72423. The average RMS error was 0.013 m. The finished 3D point cloud consists of over 34 million points (Figure 4), averaging 2614 points m^2. After the photogrammetric processing of the images taken by multispectral camera the average field resolution was 7.7 cm/pixel, the total processed area was 12.3 hectares. The average RMS error was 0.023 m. The finished 3D point cloud consists of over 412,000 points, averaging

2.21 points/m^2 (see Figure 5).

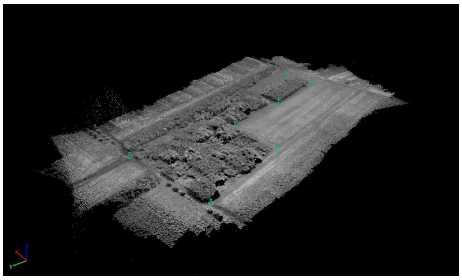


Figure 5: The 3D point cloud obtained from the multispectral images

Using Pix4Dmapper photogrammetry software, the digital surface model was created in the WGS 1984 UTM Zone 34N coordinate system. The achieved spatial resolution was 2 cm/pixel at an average height of 95.5 meters. The difference between the lowest and the highest point is 21.5 meters (see Figure 4). Applying the Pix4Dmapper software, we also created the NDVI map of the study area in the WGS 1984 UTM Zone 34N coordinate system. The spatial resolution of the map is 6.9 cm/pixel and the average NDVI is 0.78. The lowest value was 0.28 and the highest one was 0.95 (see Figure 6a).

2.2.4. Segmentation Methods

The difficulty of the segmentation of species arises from the presence of hybrids. They have similar characteristics in height and colour; therefore they are barely distinguishable even to the human eye. There are a vast amount of segmentation methods already in the literature. We aimed at selecting the ones that are based on colour rather than edges or shapes present in images. We used the most widespread segmentation methods in the field of agriculture such as Multiresolution segmentation provided by the eCognition software and the Mean shift segmentation implemented in Matlab. In addition to these algorithms we also investigated the usability of the Statistical Region Merging method.

2.2.5. Multiresolution segmentation (MRS)

We investigated the eCognition's hierarchical, multiresolution (MRS) algorithm. This algorithm combines pixels or objects, so it is based on region growing. Also this is an optimization method that minimizes the average heterogeneity with a given number of objects and maximizes the homogeneity of the object. It merges objects that best fit each other. The steps of the algorithm:

- Step 1: each pixel is an independent object. These are combined in several steps into larger objects until they reach a certain homogeneity threshold.

This threshold is derived from spectral and formal homogeneity values that can be specified in the parameter.

- Step 2: Find the best matched neighbor for each core object that is created in the first step.
- Step 3: If the best fit is not mutual, the object in the comparison will be the next object tested.
- Step 4: If the best fit is mutual, the two objects are merged.
- Step 5: In each iteration, each object is tested once.
- Step 6: Iteration stops if there is no additional merge option.

The following parameters can be set:

- Layer weights: selection and weighting of the layers we want to apply during segmentation. It increases the weighting of the layer when calculating the heterogeneity measure used to decide whether pixels/objects are merged. Zero ignores the layer.
- Scale parameter(r): maximum allowed heterogeneity within an object. It controls the amount of spectral variation within objects and therefore their resultant size. It can be any positive, integer number.
- Shape(s): the degree of spectral and geometric homogeneity (colour = 1 - shape). A weighting between the objects shape and its spectral colour whereby if 0, only the colour is considered whereas if > 0, the objects shape along with the colour are considered and therefore less fractal boundaries are produced. The higher the value, the more that shape is considered.
- Compactness(c): compactness of objects. A weighting for representing the compactness of the objects formed during the segmentation [1]. It can be a value between 0 and 1.

By changing these parameters and the input layers the size and shape of image objects are almost endlessly modifiable. The ability to perform other types of segmentation such as conditional or classification-based segmentation makes limitless modifications to the results of multiresolution segmentation possible. Sadly, it is a semiautomatic approach, there is no generally applicable formula for assigning layer weights, setting the parameters, and implementing segmentation - ultimately, trial and error and experience are the best guides [9].

2.2.6. Statistical Region Merging (SRM)

The Statistical Region Merging (SRM) Segmentation algorithm proposed by the authors of [20] is a time efficient method that operates as follows. It defines a real-valued function of similarity $f(p_0; p_1)$, where the p_0 and p_1 are two different

points in the image. It takes each pair of points and sorts the pairs based on their similarity. In the next step, it iterates through the sorted pairs that are not yet in the same region and merges their two regions if a predefined probabilistic function returns true. The value of the function f is based on the between-pixel local gradients, and their maximal perchannel variation.

The algorithm takes one argument, the scale parameter Q that defines the sizes of expected regions relative to the size of the original image. By choosing the value of Q for smaller results in larger segments, while choosing it for greater results in small segments.

2.2.7. Mean shift

The Mean Shift Segmentation (MSS), proposed in [7], is based on the assumption that the feature space is a probabilistic density function. The dense regions in the feature space correspond to local maximas. So for each data point, the algorithm performs a gradient ascent on the local estimated density until convergence. The stationary points obtained through gradient ascent represent the local maximas of the density function. All points associated with the same stationary point belong to the same cluster. The MSS only requires one parameter: the spatial radius r_s.

3. Results and discussion

3.1. NDVI and DEM

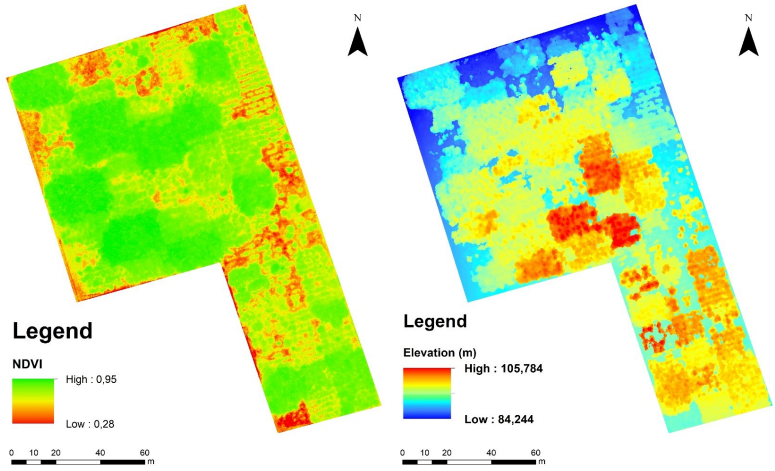
The created maps (NDVI and DEM) are suitable to track the vegetation of plants. The NDVI (Normalized Difference Vegetation Index) was developed to give an insight of plant presence and health. It is calculated as follows:

$$\mathrm{NDVI} = \frac{a_{\mathrm{nir}} - a_{\mathrm{vis}}}{a_{\mathrm{nir}} + a_{\mathrm{vis}}},$$

where a_{nir} is the surface reflectance in the near infrared channel, and a_{vis} is the reflectance in the visible red channel [2]. The NDVI map indicates where an area has healthy vegetation (green areas) and also the segments where the vegetation is low, i.e. the NDVI values are below 0.6 (yellow and red areas). With the incorporation of the DEM it is possible to locate areas where the canopy is low, as well as detect the lack of trees. As it is visible in Figure 6, compared to the NDVI map it can be seen that the hight of the plants correlates with low vegetation.

3.2. Preprocessing methods

When analyzing the orthophotos, it was found that the leaves of the same species did not have the same intensity value. Furthermore, the canopy of trees always have small gaps between leaves, branches and crowns. To eliminate the differences, two types of blur filters were used: Gauss filter and Median filter. The latter is a



(a) NDVI map (b) DEM map

Figure 6: The calculated NDVI and DEM images

nonlinear filter being used frequently to remove the "salt and pepper" image noise while preserving edges. The effect of Gaussian smoothing is also to blur the image. The degree of smoothing is determined by the standard deviation of the Gaussian and it outputs a "weighted average" of each pixel's neighborhood. Both methods were used with versatile kernels.

3.3. Evaluation method

We evaluated the results of the segmentation methods by an external cluster validity index, the Sorensen-Dice similarity coefficient (D) [5]. Based on the conclusions stated in [19], the D is a suitable measure for evaluation in the field of biogeography since it is less sensitive to outliers than the other coefficients. The coefficient D is calculated as follows:

$$D = \frac{2a}{2a + b + c},$$

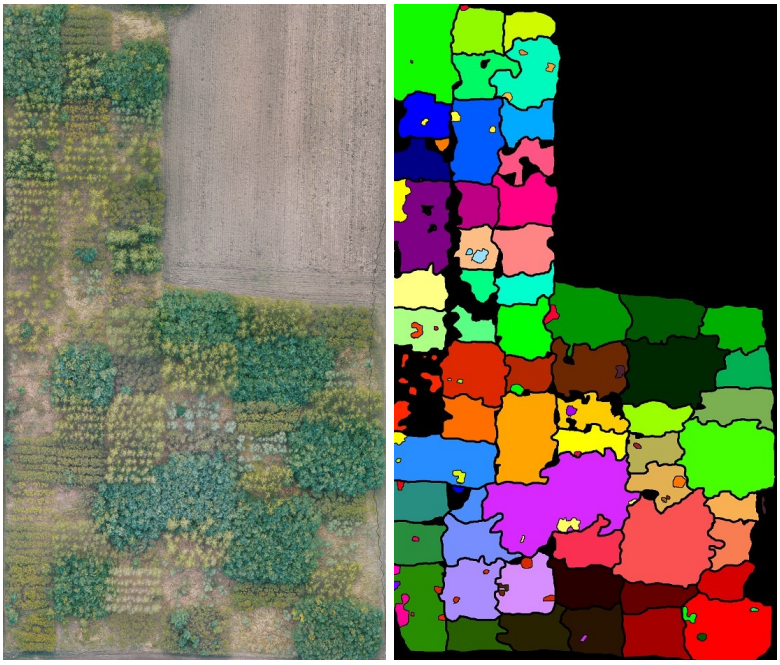
where a is the number of point pairs that belong to the same segment in the ground truth as well as in the segmentation result, b is the number of point pairs that belong to the same segment in the ground truth, but to different ones in the segmentation result and c is the number of point pairs that are in different segments in the ground truth, but in the same segment in the segmentation result.

3.4. Segmentation

We used the aforementioned segmentation algorithms: MRS, SRM and MSS. In order to evaluate the results of segmentations, we first had to create the ground

truth image for the area. This images is presented in Figure 7b.

The goal was to find a method that is efficient but also robust in the sense that it is not strongly dependent on its input parameters. First we evaluated the eCognition's MRS segmentation algorithm that is the state of the art currently in this field of application. This is a semi-automatic process, it requires the users supervision to achieve the best results. Therefore, to reach the accuracy of this method was the goal for the other, unsupervised methods. The different segmentation methods were used with varying preprocessing methods. The segmentation methods also differ in their inputs.



(a) The original orthophoto (b) The ground truth image

Figure 7: The original orthophoto and the corresponding ground truth image

In the MRS of eCognition, the user can select the RGB channels and can add the DEM and the NDVI of the original image as a new layer. The above mentioned parameters were selected empirically in our study. The image layer weights were set for RGB channels (1,5,10 and 1,10,5), for DEM (between 1 and 10) and for NDVI (between 1 and 10). The scale parameter was tested between 100 and 240. Our shape parameter was set between 0.01 and 0.4, the compactness parameter was between 0.6 and 1.0. For the original size images, the application of filters resulted in worse results. For the resampled (reduced to 75%) images, we got slightly worse results (68–71% accuracy) and the applied filters also resulted in weaker results.

Reducing the size of the original orthophoto (and both DEM and NDVI images) to 50% led to results below 50%. The best results were obtained by performing this algorithm on the image reduced to one quarter of the original size with the layers: 1,5,10 of the RGB channels respectively, 2 for the DEM layer and 0 to the NDVI layer, the other parameters were set as follows: $r = 240$, $s = 0.4$, $c = 0.9$. After the first run of the algorithm, further hierarchical segmentation was applied by selecting regions that we wished to further split and the segmentation method was rerun on that region only. After performing such steps repeatedly, we reached the highest accuracy: 73.15% (Fig. 8).

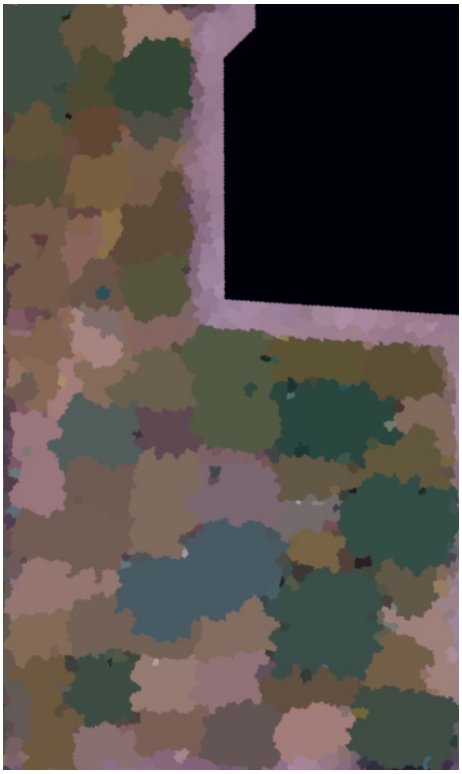


Figure 8: The segmented objects after performing multiresolution segmentation for the study area

The Matlab implementation of the MSS only required one parameter: the spatial radius r_s. The rest of the settings of the algorithm are calculated from this parameter. The parameter r_s was tested with the values $0.05, 0.06, \ldots, 0.1$.

For preprocessing, the Median filter was tested with two kernel sizes: 9 and 12. The used kernel sizes of the Gauss filter were 2 and 3, the application of preprocessing methods improves the segmentation accuracy significantly regardless of the type of smoothing operator. The best accuracy, 80.70% was reached on the

RGB image resampled to 10% of the original size and filtered by a Gauss filter of kernel size 3 and the rs set to 0.07.

We have used the Matlab implementation of the SRM algorithm provided by the authors of [20]. We tested this algorithm with the RGB images downsampled to 10% of the original size. The Median and Gauss smoothing filters were also tested as preprocessing steps before the application of the SRM. The values for the scale parameter Q were selected from the range $[100, 3000]$ on a logarithmic basis as it was proposed by the authors of [20]. The Median filter was tested with two kernel sizes: 9 and 12. The used kernel sizes of the Gauss filter were 2 and 3. The achieved best accuracy was 62.45%.

The summary of the best accuracies of the used segmentation methods is shown in Table 1.

Segmentation	Accuracy
eCognition (MRS)	73.15%
MSS	**80.70%**
SRM	62.45%

Table 1: Results of segmentations

4. Conclusions

One aim of this case study was to investigate the applicability of a lowcost UAV in the field of precision agriculture. On other goal was to find a suitable segmentation method that is able to operate on an image that contains several hybrids of the same tree species, a task which is hard even for the human eye. Many studies were solving segmentation problems on areal imagery, however, these mostly aimed at detecting vegetation and distinguishing it from the surrounding landmarks [6, 8, 10, 12]. Our goal was to perform the task of segmenting tree species apart, that is an even more challenging task with hybrids present.

The created precise 3D model is suitable for agriculture experts to examine energy plantation, the NDVI and DEM maps can be used to observe vegetation in the study area and to give a mass estimate. The orthophoto obtained from the 3D model can be used for segmentation. The eCognition's MRS reached 73.15% accuracy when the DEM was added to the RGB orthophoto as a layer. With the help of the NDVI map added to the RGB image as a layer we got worse segmentation results. The Matlab implementation of the MSS algorithm was the parameter insensitive method that reached the highest accuracy with 80.70%.

References

[1] J. van Aardt, R. Wynne: *A Multiresolution Approach to Forest Segmentation as a Precursor to Estimation of Volume and Biomass by Species*, in: Jan. 2004.

[2] A. Bannari, D. Morin, F. Bonn, A. R. Huete: *A review of vegetation indices*, Remote Sensing Reviews 13.1-2 (1995), pp. 95–120,
doi: https://doi.org/0.1080/02757259509532298.

[3] O. Csillik: *Fast segmentation and classification of very high resolution remote sensing data using*, Remote Sens 9.3 (2016), p. 243,
doi: https://doi.org/10.3390/rs9030243.

[4] A. Cunliffe, Brazier: *Ultrafine grain landscape-scale quantification of dryland vegetation structure with drone-acquired structure-from-motion photogrammetry*, Remote Sensing of Environment 183.1 (2016), pp. 129–143,
doi: https://doi.org/10.1016/j.rse.2016.05.019.

[5] L. R. Dice: *Measures of the amount of ecologic association between species*, Ecology 26.3 (1945), pp. 297–302.

[6] D. Fonseca-Luengo, A. García-Pedrero, M. Lillo-Saavedra, et al.: *Optimal scale in a hierarchical segmentation method for satellite images*, in: International Conference on Rough Sets and Intelligent Systems Paradigms, New York, NY, USA: Springer, Cham, 2014, pp. 351–358,
doi: https://doi.org/10.1007/978-3-319-08729-0_36.

[7] K. Fukunaga, L. Hostetler: *The estimation of the gradient of a density function, with applications in pattern recognition*, in: IEEE Transactions on information theory, New York, NY, USA: IEEE, 1975, pp. 32–40,
doi: https://doi.org/10.1109/TIT.1975.1055330.

[8] M. Grizonnet, J. Michel, V. Poughon, et al.: *Orfeo ToolBox: open source processing of remote sensing images*, Open geospatial data, softw. stand 2 (2017), Article number: 15,
doi: https://doi.org/10.1186/s40965-017-0031-6.

[9] R. Hamilton, K. Megown, T. Mellin, I. Fox: *Guide to automated stand delineation using image segmentation*, Oct. 2007.

[10] M. Hassanein, N. El-Sheimy: *An efficient weed detection procedure using low-cost UAV imagery system for precision agriculture applications*, in: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. Karsluhe, Germany: ISPRS rs, 2018, pp. 181–187,
doi: https://doi.org/10.5194/isprs-archives-XLII-1-181-2018.

[11] G. J. Haya, T. Blaschkeb, D. J. Marceaua, A. Bouchard: *A comparison of three image-object methods for the multiscale analysis of landscape structure*, ISPRS Journal of Photogrammetry and Remote Sensing 57.5-6 (2003), pp. 327–345,
doi: https://doi.org/10.1016/S0924-2716(02)00162-4.

[12] M. Hossain, D. Chen: *Segmentation for Object-Based Image Analysis (OBIA): A review of algorithms and challenges from remote sensing perspective*, ISPRS Journal of Photogrammetry and Remote Sensing 150.6 (2019), pp. 115–134,
doi: https://doi.org/10.1016/j.isprsjprs.2019.02.009.

[13] J. W. Karl, B. A. Maurer: *Spatial dependence of predictions from image segmentation: A variogram-based method to determine appropriate scales for producing landmanagement information*, Ecological Informatics 5 (2010), pp. 194–202,
doi: https://doi.org/10.1016/j.ecoinf.2010.02.004.

[14] H. Li, H. Gu, Y. Han, J. Yang: *An efficient multiscale SRMMHR (Statistical Region Merging and Minimum Heterogeneity Rule) segmentation method for high-resolution remote sensing imagery*, Open geospatial data, softw. stand 2.2 (2009), pp. 67–73,
doi: https://doi.org/10.1109/JSTARS.2009.2022047.

[15] S. Messinger, Asner: *Rapid assessments of Amazon forest structure and biomass using small unmanned aerial systems*, Forests 8.8 (2016), p. 615,
doi: https://doi.org/10.3390/rs8080615.

[16] D. Ming, J. Li, J. Wang, M. Zhang: *Scale parameter selection by spatial statistics for GeOBIA:Using mean-shift based multi-scale segmentation as an example*, ISPRS Journal of Photogrammetry and Remote Sensing 106 (2015), pp. 28–41,
doi: https://doi.org/10.1016/j.isprsjprs.2015.04.010.

[17] R. Mlambo, I. H. Woodhouse, F. Gerard, K. Anderson: *Structure from motion (sfm) photogrammetry with drone data: a low cost method for monitoring greenhouse gas emissions from forests in developing countries*, Forests 8.3 (2017), p. 68,
doi: https://doi.org/10.3390/f8030068.

[18] M. Mohan, C. A. Silva, C. Klauberg, et al.: *Individual tree detection from unmanned aerial vehicle (uav) derived canopy height model in an open canopy mixed conifer forest*, Forests 8.9 (2017), p. 340,
doi: https://doi.org/10.3390/f8090340.

[19] M. Murguía, J. L. Villaseñor: *Estimating the effect of the similarity coefficient and the cluster algorithm on biogeographic classifications*, Annales Botanici Fennici. JSTOR 40.6 (2003), pp. 415–421.

[20] R. Nock, F. Nielsen: *Statistical region merging*, in: IEEE Transactions on pattern analysis and machine intelligence, New York, NY, USA: IEEE, 2004, pp. 1452–1458,
doi: https://doi.org/10.1109/TPAMI.2004.110.

[21] B. St-Onge, J. Jumelet, M. Cobello, C. Véga: *Measuring individual tree height using a combination of stereophotogrammetry and lidar*, Canadian Journal of Forest Research 34.1 (2004), pp. 2122–2130,
doi: https://doi.org/10.1139/x04-093.

[22] A. Parraga, D. Doering, J. G. Atkinson, et al.: *Wheat Plots Segmentation for Experimental Agricultural Field from Visible and Multispectral UAV Imaging*, in: SAI Intelligent Systems Conference, London, UK: Springer Cham, 2018, pp. 388–399,
doi: https://doi.org/10.1007/978-3-030-01054-6_28.

[23] D. C. Tsouros, S. Bibi, P. G. Sarigiannidis: *A Review on UAV-Based Applications for Precision Agriculture*, MDPI 10.11 (2019), p. 349,
doi: https://doi.org/10.3390/info10110349.

[24] H. Watts, Ambrosia: *Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use*, Forests 4 (2012), pp. 1671–1692,
doi: https://doi.org/10.3390/rs4061671.

[25] G. H. Westoby, Brasington, Reynolds: *Structure-from-motion photogrammetry: A low-cost, effective tool for geoscience applications*, Geomorphology 4.6 (2012), pp. 300–314,
doi: https://doi.org/10.1016/j.geomorph.2012.08.021.

[26] T. ZHao, H. Niu, E. de la Rosa, et al.: *Tree canopy differentiation using instance-aware semantic segmentation*, in: Proceedings of the 2018 ASABE Annual International Meeting, NSt. Joseph, MI, USA: American Society of Agricultural and Biological Engineers, 2018, p. 1.

Annales Mathematicae et Informaticae
***52** (2020) pp. 199–215*
DOI: https://doi.org/10.33039/ami.2020.05.001
URL: https://ami.uni-eszterhazy.hu

Optimized line and line segment clipping in E2 and Geometric Algebra*

Vaclav Skala

University of West Bohemia, Pilsen, Czech Republic
www.VaclavSkala.eu

Submitted: November 28, 2019
Accepted: May 1, 2020
Published online: May 11, 2020

Abstract

Algorithms for line and line segment clipping are well known algorithms especially in the field of computer graphics. They are formulated for the Euclidean space representation. However, computer graphics uses the projective extension of the Euclidean space and homogeneous coordinates for representation geometric transformations with points in the E^2 or E^3 space. The projection operation from the E^3 to the E^2 space leads to the necessity to convert coordinates to the Euclidean space if the clipping operation is to be used.

In this contribution, an optimized simple algorithm for line and line segment clipping in the E^2 space, which works directly with homogeneous representation and not requiring the conversion to the Euclidean space, is described. It is based on Geometric Algebra (GA) formulation for projective representation.

The proposed algorithm is simple, efficient and easy to implement. The algorithm can be efficiently modified for the SSE4 instruction use or the GPU application, too.

Keywords: line clipping, line segment clipping, homogeneous coordinates, projective space, geometric algebra, principle of duality, GPU, SSE4 instruction.

MSC: 65D18, 68U05

*The research was partially supported by the Czech Science Foundation, Czech Republic, project GACR No. GA17-05534S

1. Introduction

The line and line segment clipping are fundamental and critical operations in the computer graphics pipeline as all the processed primitives have to be clipped out of the drawing area to decrease computational requirements and also respect the physical restrictions of the hardware. The clipping operations are mostly connected with the Window-Viewport and projection operations. There are many algorithms developed recently with many modifications, see Andreev [1], Day [4], Dörr [5], Duvalenko [8], Kaijian [12], Krammer [14], Liang [16], Sobkow [29].

However, those algorithms have been developed for the Euclidean space representation in spite of the fact, that geometric transformations, i.e. projection, translation, rotation, scaling and Window-Viewport etc., use homogeneous coordinates, i.e. projective representation. This results in the necessity to convert the results of the geometric transformations to the Euclidean space using division operation.

The conversion of a point $\mathbf{x} = [x, y : w]^T$ from homogeneous coordinates to the Euclidean representation $\mathbf{X} = (X, Y)$ is given as:

$$X = x/w, \quad Y = y/w, \quad w \neq 0,$$

where w is the homogeneous coordinate. It means, that a point $\mathbf{X} \in E^2$ is represented by a line in the projective space $x, y : w$ without the origin, which represents a point in the infinity, see Figure 1.

The extension to the E^3 case is straightforward, e.g. Foley [9].

$$X = x/w, \quad Y = y/w, \quad Z = z/w, \quad w \neq 0,$$

where $\mathbf{x} = [x, y, z : w]^T$. The use of the projective extension of the Euclidean space is convenient not only for geometric transformations, as it replaces addition by multiplication in the case of translation operation, but also it enables to represent a point in infinity. Also, it enables to express some geometric entities in more compact form, e.g. a line in the E^2 case as:

$$aX + bY + c = 0, \quad ax + by + cw = 0, \quad \mathbf{a}^T\mathbf{x} = 0, \tag{1.1}$$

where $\mathbf{a} = [a, b : c]^T$. It is necessary to note, that (a, b) represents the normal vector of a line, while c is related to the distance of a line from the origin of the Euclidean coordinate system. Similarly, a plane in the E^3 case is defined as:

$$aX + bY + cZ + d = 0, \quad ax + by + cz + dw = 0, \quad \mathbf{a}^T\mathbf{x} = 0, \tag{1.2}$$

where $\mathbf{a} = [a, b, c : d]^T$.

However, it is necessary to distinguish vectors, as "movable" entities, from "frames", which have the origin as the reference point. It is necessary to note, that metric is not defined in the projective space. In many cases, the principle of duality can be used to derive a solution of a dual problem and have only one programming sequence for both problems, i.e. the primary one and the dual. Unfortunately, the principle of duality is not usually part of the standard computer science curricula.

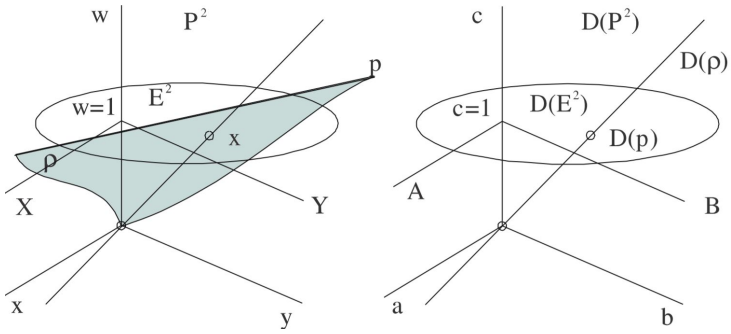


Figure 1: Projective space and its dual

2. Principle of duality

The principle of duality is one of the most important principles in mathematics. In our case of geometric problems described by linear equations, see (1.1) and (1.2), the principle of duality states that any theorem remains true when we interchange the words

- "point" and "line" in the E^2 case, resp. "point" and "plane" in the E^3 case,
- "lie on" and "pass through", "join" and "intersection" and so on.

Once the theorem has been established, the dual theorem is obtained as described by Johnson [11].

In other words, the principle of duality in the E^2 case says, that in all theorems it is possible to substitute the term "point" by the term "line" and term "line" by the term "point" and the given theorem remains valid. This helps a lot in the solution of some geometrical problems, similarly in the E^3 case. It means, that the intersection computation of two lines is dual to the computation of a line given by two points in the E^2 case.

Similarly, the intersection computation of three planes is dual to the computation of a plane given by three points in the E^3 case.

It is strange as the usual solution in the first case leads to formulation $\mathbf{Ax} = \mathbf{b}$, while in the second case, the parameters of a line are determined as $\mathbf{Ax} = \mathbf{0}$. However, if the projective representation is used, both cases are solved as $\mathbf{Ax} = \mathbf{0}$, Skala [23].

This is the direct impact of the fact, that the point must lie on a line in the E^2 case, resp. on a plane in the E^3 case, (2.1). Also, two lines in the E^2 case, respectively three planes in the E^3 case must not be collinear, i.e.:

$$\mathbf{a}^T\mathbf{x} = 0 \tag{2.1}$$

where $\mathbf{a} = [a, b : c]^T$ in E^2, resp. $\mathbf{a} = [a, b, c : d]^T$ in E^3. It can be seen that the meaning of the term $\mathbf{a}$ and $\mathbf{x}$ can be interchanged due to the principle of duality.

Let us consider the intersection of two lines in the E^2 case. Both lines must not be collinear, the conditions (2.1) for each line must be orthogonal to other, therefore the result of the outer product (cross product) must be zero. Similarly, for planes which must be non-collinear Lengyel [15], Skala [22–24].

Let us consider two lines $\mathbf{a}_1 = [a_1, b_1 : c_1]^T$ and $\mathbf{a}_2 = [a_2, b_2 : c_2]^T$ in the E^2 case (using the cross-product notation extended to the $x, y : w$ coordinate system). Then the intersection point $\mathbf{x} = [x, y : w]^T$ is given as:

$$\mathbf{a}_1^T\mathbf{x} = 0, \quad \mathbf{a}_2^T\mathbf{x} = 0, \quad (\mathbf{a}_1 \times \mathbf{a}_2)^T\mathbf{x} = 0.$$

Using the matrix notation:

$$\det \begin{bmatrix} x & y & w \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} = 0.$$

It means that a point given as the intersection of two lines is given as:

$$\mathbf{x} = \mathbf{a}_1 \times \mathbf{a}_2 \quad \text{i.e.} \quad \mathbf{x} = \mathbf{a}_1 \wedge \mathbf{a}_2,$$

where $\mathbf{x} = [x, y : w]^T$ and $\wedge$ means the outer product.

As a direct consequence of the principle of duality a line $\mathbf{a} = [a, b : c]^T$ given by two points $\mathbf{x}_1 = [x_1, y_1 : w_1]^T$ and $\mathbf{x}_2 = [x_2, y_2 : w_2]^T$ is given as:

$$\mathbf{a} = \mathbf{x}_1 \times \mathbf{x}_2 \quad \text{i.e.} \quad \mathbf{a} = \mathbf{x}_1 \wedge \mathbf{x}_2. \tag{2.2}$$

It should be noted that the operator $\times$ is the equivalent specific symbol used in the E^3 case, while $\wedge$ is defined for the n-dimensional space, in general.

Extension to the E^3 dimensional case is quite simple due to multi-linearity. It means, that the intersection point of three planes $\mathbf{a}_i$, $i = 1, 2, 3$ is given as:

$$\det \begin{bmatrix} x & y & z & w \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix} = 0.$$

It means that the point given as an intersection of three planes is given as:

$$\mathbf{x} = \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_3,$$

where $\mathbf{x} = [x, y, z : w]^T$.

As a direct consequence of the principle of duality, a plane $\mathbf{a} = [a, b, c : d]^T$ given by three points $\mathbf{x}_i = [x_i, y_i, z_i : w_i]^T$, $i = 1, 2, 3$ is given as:

$$\mathbf{a} = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \mathbf{x}_2.$$

3. Geometric algebra

Linear algebra is used for formulation and solution of many engineering problems, including a solution of geometrically oriented problems, e.g. in computer vision or computer graphics. Usually, vectors or matrices are used to represent one-dimensional or two-dimensional (data) structure and standard operations are defined. For vectors in the mathematical sense, basic mathematical operations are defined, e.g. addition, multiplication (dot-product, cross-product), etc. This "standard" vector algebra framework enables basic operation with geometric entities.

3.1. Geometric product

However, there is another framework called Geometric Algebra (GA), which comes from the William Kingdom Clifford formulation and which enables to define the product of union and intersection operations with points, lines, areas, volumes and hyper-volumes in general, see Vince [30], Kanatani [13]. The GA is an alternative formalism for describing geometrical entities and operations in n-dimensional space. It uses only one product (multiplication) called *geometric product* defined as:

$$\mathbf{ab} = \mathbf{a} \bullet \mathbf{b} + \mathbf{a} \wedge \mathbf{b}$$

where $\mathbf{a} \bullet \mathbf{b}$ is the *dot* (scalar) product and $\mathbf{a} \wedge \mathbf{b}$ is the *outer product* (equivalent to the *cross-product* in the E^3 case).

It can be seen that the geometric product is "strange" as the result consists of a scalar value and a bivector (usually called as a vector, but having different properties and representation from a vector), which is the result of the outer product.

The geometric product can be easily computed as the non-commutative tensor product as $\mathbf{a} \otimes \mathbf{b}$, see Skala [28], as:

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix}.$$

The diagonal elements represent the *dot product* part, while the upper and the lower triangular matrices represent the *outer product* part. It should be noted, that the geometric product computation using the non-commutative tensor product can be used for the n-dimensional space, too.

Nowadays, the geometric algebra is widely used across many fields, but mostly in connection with vector oriented operations in the Euclidean space. The applications of GA can be found in Physics (Hestenes [10]), Computer Science (Dorst [6]), Computer Graphics (Vince [30], Lengyel [15]), and in other engineering fields as well (Dorst [7]).

3.2. Geometric product and projective space

It should be noted, that it is possible to extend the GA for the projective extension of the Euclidean space as well. In the case of computer graphics, points are not

represented by vectors in the mathematical sense, as they are represented by a vector data structure, which represents a frame fixed to the origin of the coordinate system.

As mentioned in Chapter 2, computation of a line $\mathbf{p}$ by given two points and an intersection point $\mathbf{x}$ given as an intersection of two lines is given by the outer product as:

$$\mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2, \quad \mathbf{x} = \mathbf{p}_1 \wedge \mathbf{p}_2.$$

Using the determinant notation:

$$\det \begin{bmatrix} a & b & c \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{bmatrix} = 0, \quad \det \begin{bmatrix} x & y & w \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} = 0.$$

It can be seen, there is no need to convert homogeneous coordinates of the points to the Euclidean space, since the determinant is multi-linear.

Extension to the E^3 case is straightforward, i.e. a plane ρ given by three points and an intersection point $\mathbf{x}$ of three planes are given as:

$$\rho = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \mathbf{x}_3, \quad \mathbf{x} = \rho_1 \wedge \rho_2 \wedge \rho_3.$$

Using the determinant notation, the intersection point of three planes, respectively the plane given by three points is are given as:

$$\det \begin{bmatrix} a & b & c & d \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_1 & y_3 & z_3 & w_3 \end{bmatrix} = 0, \quad \det \begin{bmatrix} x & y & z & w \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_3 & c_3 & d_3 \end{bmatrix} = 0.$$

It can be seen, that there is no problem with singular cases, like collinear lines, respectively planes, as the intersection is in infinity. In this case, the homogeneous coordinate of the result is $w = 0$ or $w \mapsto 0$. This is to be evaluated after outer product computation.

4. Line clipping

The line clipping operation in E^2 space is a fundamental problem in Computer Graphics. It was already deeply analyzed and many algorithms have been developed. The Cohen-Sutherland (CS) [9] for a line segment clipping against the rectangular window, the Liang-Barsky (LB) [16] and Cyrus-Beck (CB) [3] (extensible to the E^3 case) algorithms for clipping a line against a convex polygon, the Nichol-Lee-Nichol (LNL) [17] (modified by Skala [27]) are the most used algorithms.

However, some more sophisticated algorithms or modification of the recent ones have been developed recently, e.g. line clipping against a rectangular window, see Bui [2], Skala [20, 21], line clipping by a convex polygon with $O(\lg N)$ complexity,

see Skala [26] (based on Rappaport [18]), or algorithm with $O_{\text{expected}}(1)$ complexity, see Skala [25], etc. In the case of E^3, the algorithms have computational complexity $O(N)$ as there is no ordering in E^3 case, however, the algorithm with $O_{\text{expected}}(\text{sqrt}(N))$ have been developed by Skala [19, 20, 25].

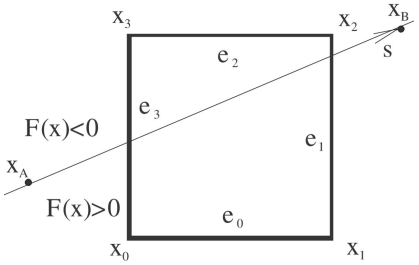


Figure 2: Clipping against the rectangular window in E^2

The line and line segment clipping algorithms against a rectangular window in E^2 are probably the most used algorithms and any improvements or speed up can have a high influence on the efficiency of the whole graphics pipeline.

Let us consider a typical example of a line clipping by a rectangular clipping window, see Figure 2, and a line p given in the implicit form using projective notation.

$$p\colon ax + by + cw = 0 \quad \text{i.e.} \quad \mathbf{a}^T\mathbf{x} = 0,$$

where $\mathbf{a} = [a, b : c]^T$ are coefficients of the given line p, $\mathbf{x} = [x, y : w]^T$ is a point on this line using projective notation.

In the following, a version of the line clipping algorithm for the general case will be described, which can be easily extended to a line clipping and line segment clipping by a convex polygon, and the optimization for the use in the Normalized Device Coordinate (NDC) system.

4.1. S-L-Clip algorithm

Let us consider an implicit function $F(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$. The clipping operation should determine intersection points $\mathbf{x}_i = [x_i, y_i : w_i]^T$, $i = 1, 2$ of the given line with the window, if any. The line splits the plane into two parts, see Fig. 2. The corners of the window are split into two groups according to the sign of the $F(\mathbf{x})$ value. This results into Smart-Line-Clip (S-L-Clip) algorithm, see Algorithm 1.

It means that each corner can be classified by a bit value c_i as:

$$c_i = \begin{cases} 1 & \text{if } F(\mathbf{x}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad i = 0, 1, 2, 3,$$

where $\mathbf{a} = [a, b : c]^T$ are coefficients of the given line p, $\mathbf{x} = [x, y : w]^T$ means a point on this line.

c	**c**	TAB1	TAB2	MASK
0	0000	None	None	None
1	0001	0	3	0100
2	0010	0	1	0100
3	0011	1	3	0010
4	0100	1	2	0010
5	0101	N/A	N/A	N/A
6	0110	0	2	0100
7	0111	2	3	1000

c	**c**	TAB1	TAB2	MASK
15	1111	None	None	None
14	1110	3	0	None
13	1101	1	01	0100
12	1100	3	1	0010
11	1011	2	1	0010
10	1010	N/A	N/A	N/A
9	1001	2	0	0100
8	1000	3	2	1000

Table 1: All cases; N/A - Non-Applicable (impossible) cases

Table 1 shows the codes for all situations (some of those are not possible). The **TAB1** and **TAB2** contain indices of edges of the window intersected by the given line (values in the **MASK** will be used in the line segment algorithm).

It can be seen, that the S-L-Clip algorithm (see Algorithm 1) is quite simple and easily extensible for the convex polygon clipping case as well (Table 1 can be generated synthetically). It is significantly simpler than the Liang-Barsky algorithm [16]. It also supports SSE4 and GPU use directly and leads to simple implementations, as the cross-product and dot-product operations, are supported in hardware. It should be noted, that the algorithm is designed for a very general case, as the window corners and the points defining the line, are generally in the projective representation, i.e. $w \neq 1$. Therefore, the S-L-Clip algorithm has further potential for optimization, especially for the case, when the corner points of the window are given in the Euclidean coordinates, i.e. $w = 1$, and clipping is made in the Normalized Device Coordinate (NDC) system.

Algorithm 1 S-L-Clip - Line clipping algorithm by the rectangular window

```
1: procedure S-L-Clip(x_A, x_B);                    ▷ line is given by two points
2:     p := x_A ∧ x_B;                              ▷ computation of the line coefficients
3:     for i := 0 to 3 do
4:         if p^T x_i ≥ 0 then c_i := 1 else c_i := 0;   ▷ codes computation
5:     end for
6:     if c ≠ [0000]^T and c ≠ [1111]^T then        ▷ line intersects the window
7:         i := TAB1[c];   x_A := p ∧ e_i;          ▷ first intersection point
8:         j := TAB2[c];   x_B := p ∧ e_j;          ▷ second intersection point
9:         output(x_A, x_B);
10:    end if
11: end procedure
```

4.2. S-L-Clip-Opt - Optimization of the S-L-Clip

The S-L-Clip algorithm can be optimized for the use in the E^2 case, as the corners of the window and points defining the line are in the Euclidean coordinates, i.e. $w = 1$, and the edges of the window are vertical or horizontal only. Also, it is necessary to consider the computer graphics pipeline, where all primitives passing the clipping operations are transformed from the World Coordinates (WC) to the Normalized Device Coordinates(NDC) and then to the Device Coordinates (DC), where NDC coordinates are $\langle 0, 1\rangle \times \langle 0, 1\rangle$ or $\langle -1, 1\rangle \times \langle -1, 1\rangle$, which simplifies the outer-product (cross-product) computation significantly.

These computational transformations can be described as:

$$\mathbf{x}' = \mathbf{T}_{NDC \mapsto DC}\ \mathbf{CLIP}\ (\mathbf{T}_{WC \mapsto NDC}\ \mathbf{x}).$$

Let us consider a line coefficients determination first, using (2.2), and setting $w = 1$. Then the coefficients of the line are given by (4.1).

$$a = y_1 - y_2, \quad b = x_2 - x_1, \quad c = x_1 * y_2 - x_2 * y_1. \tag{4.1}$$

It leads to a significant reduction of a number of the floating point operations $(\pm, *)$ from $(6, 3)$ to $(3, 2)$. Also, the outer product is used for computation of the intersection points, i.e. $\mathbf{x}_A$ and $\mathbf{x}_B$, can be simplified.

As the edges of the window are vertical or horizontal only and clipping is done in the normalized space NDC, the codes of the corners and related intersection point computation can be simplified significantly. It means, that for each edge of the window the intersection computation with the line can be simplified as:

$\begin{bmatrix} x & y & w \\ a & b & c \\ 0 & 1 & 0 \end{bmatrix} = 0$	$\begin{bmatrix} x & y & w \\ a & b & c \\ 1 & 0 & -1 \end{bmatrix} = 0$	$\begin{bmatrix} x & y & w \\ a & b & c \\ 0 & 1 & -1 \end{bmatrix} = 0$	$\begin{bmatrix} x & y & w \\ a & b & c \\ 1 & 0 & 0 \end{bmatrix} = 0$
edge e_0	edge e_1	edge e_2	edge e_3

Table 2: Explicit evaluation of an intersection point for each edge

It means that the outer product sequence for the line intersection with an edge can be replaced by direct computing of all cases shown in Table 1. Rewriting those conditions at Table 2, the intersection for each edge is given as:

edge e_0:	$x = -c$,	$y := 0$,	$w := a$,
edge e_1:	$x = -b$,	$y := a + c$,	$w := -b$,
edge e_2:	$x = -b - c$,	$y := a$,	$w := a$,
edge e_3:	$x = 0$,	$y := c$,	$w := -b$.

It leads to another significant reduction of the number of the floating point operations $(\pm, *)$ from $(6, 3)$ to $(1, 0)$ in the most of cases. If the line does not intersect the window, there is no computation at all.

The optimized line clipping algorithm for the E^2 case is represented by the algorithm S-L-Clip-Opt, see Algorithm 2.

From Algorithm 2, it can be seen that also the evaluation of the window corners were simplified as instead of $4 * (2, 3)$ with floating point operations, only $(4, 0)$ operations are needed. This results in an additional speedup of the proposed optimization.

Algorithm 2 Optimized S-L-Clip-Opt line clipping algorithm in E^2

```
1: procedure S-L-Clip-Opt(x_A, x_B);                    ▷ line is given by two points
2:     a = y_1 − y_2;   b = x_2 − x_1;
3:     c = x_1 * y_2 − x_2 * y_1;                                  ▷ line coefficients
4:     c_0 := sign(c);   c_1 := sign(a + c);                ▷ corner's codes computation
5:     c_2 := sign(a + b + c);   c_3 := sign(b + c);                ▷ c = [c_3, c_2, c_1, c_0]^T
6:     if c ≠ [0000]^T and c ≠ [1111]^T then                ▷ line intersects the window
7:         i := TAB1[c];                                              ▷ x_A := [x_A, y_A : w_A]^T
8:         switch i do                                          ▷ equivalent of x_A := p ∧ e_i;
9:             case 0: x_A := −c;   y_A := 0;   w_A := a;
10:            case 1: x_A := −b;   y_A := a + c;   w_A := b;
11:            case 2: x_A := −b − c;   y_A := −a;   w_A := a;
12:            case 3: x_A := 0;   y_A := c;   w_A := −b;
13:            default: ERROR                                  ▷ actually the N/A case
14:        end switch
15:        j := TAB2[c];                                              ▷ x_B := [x_B, y_B : w_B]^T
16:        switch j do                                          ▷ equivalent of x_B := p ∧ e_j;
17:            case 0: x_B := −c;   y_B := 0;   w_B := a;
18:            case 1: x_B := −b;   y_B := a + c;   w_B := b;
19:            case 2: x_B := −b − c;   y_B := −a;   w_B := a;
20:            case 3: x_B := 0;   y_B := c;   w_B := −b;
21:            default: ERROR                                  ▷ actually the N/A case
22:        end switch
23:        output(x_A, x_B);                  ▷ output with the intersection points
24:    end if
25: end procedure
```

Now, the proposed optimized algorithm is to be modified for the line segment clipping case, which is used nearly exclusively in computer graphics.

5. Line segment clipping

In computer graphics, geometric elements like points, line segments, triangles, etc. are processed. Therefore, the proposed algorithm is to be modified for the

line segment clipping case, see Figure 3.

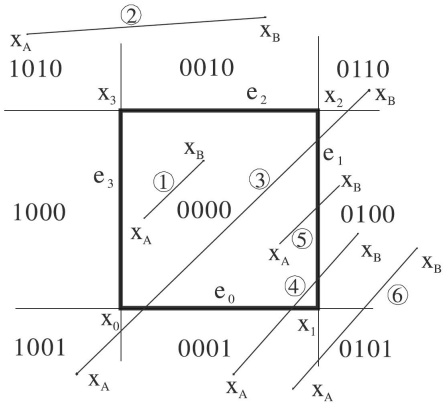


Figure 3: The codes of line segment end-points

It can be seen that there are some special line positions, which lead to direct acceptance or rejection of the whole line segment, while other cases have to be processed.

5.1. End-points coding

A line segment is defined by its end-points $\mathbf{x}_A$ and $\mathbf{x}_B$. The classification of the line segment end-points and the corners of the window mutual positions enables faster processing, see Algorithm 3. The end-point classification was used in the CS algorithms developed by Cohen-Sutherland [9]. Some additional coding for speedup were introduced in Bui [2]. It enables simple rejection of line segments not intersecting the window and direct acceptance of segments totally inside of the window. If $\mathbf{c}_A$ and $\mathbf{c}_B$ are codes of the end-points then the sequence catching those cases can be expressed as:

if $(\mathbf{c}_A$ **or** $\mathbf{c}_B) = [0000]$ **then** the line segment is totally inside;

if$(\mathbf{c}_A$ **and** $\mathbf{c}_B) \neq [0000]$ **then** the line segment is outside;

If the end-points of a line are given in the Euclidean space, i.e. $w = 1$, then the codes of the end-points are determined as in Algorithm 3. In the general case, i.e. when $w \neq 1$ and $w > 1$ the conditions must be modified using multiplication as $xw_{\min} < x_{\min}w$, etc. and therefore no division operation is needed.

It can be seen, that other cases, see Figure 3, cannot be directly distinguished by the CS algorithm coding and intersection points are to be computed, including the invalid ones. It is necessary to note, that the CS algorithm uses division operations in the floating point.

The S-LS-Clip algorithm (Algorithm 4) is derived from the S-L-Clip algorithm (Algorithm 1), which uses the gained information on positions of the line segment end-points.

Algorithm 3 End-point code computation

```
1: procedure CODE (c, x);                    ▷ code c for the position x = [x, y : 1]^T
2:     c := [0000]^T;                                              ▷ initial setting
3:     if x < x_min then c := [1000]^T          ▷ setting according to x coordinate
4:         if x > x_max then c := [0100]^T;
5:     if y < y_min then c := c lor [0001]^T  ▷ setting according to y coordinate
6:         if y > y_max then c := c lor [0010]^T;
7:                                  ▷ lor represents or operation on all bits
8: end procedure
```

Algorithm 4 Smart-line segment clipping algorithm by the rectangular window

```
1: procedure S-LS-CLIP(x_A, x_B);                    ▷ two line segment end-points
2:     CODE (c_A, x_A); CODE (c_B, x_B);     ▷ code for the end-points x_A and x_B
3:     if (c_A lor c_B) = [0000]^T then output (x_A, x_B); EXIT
4:                                                  ▷ the whole segment is inside
5:     if (c_A land c_B) ≠ [0000]^T then EXIT     ▷ the whole segment is outside
6:     p := x_A ∧ x_B;                          ▷ computation of the line coefficients
7:     for i := 0 to 3 do
8:         if p^T x_i ≥ 0 then c_i := 1 else c_i := 0;          ▷ codes computation
9:     end for
10:    if c = [0000]^T or c = [1111]^T then EXIT          ▷ line does not intersect
11:    if c_A ≠ 0 and c_B ≠ 0 then                        ▷ two intersection points
12:        x_A := p ∧ e_i; x_B := p ∧ e_j;
13:        output (x_A, x_B); EXIT
14:    i := TAB1[c];   j := TAB2[c];                ▷ only one end-point is inside
15:                                                    ▷ end-points handling
16:    if c_A = 0 then                                  ▷ point x_A is inside
17:        if (c_B land MASK[c]) ≠ 0 then
18:            x_B := p ∧ e_i;                          ▷ new position of x_B
19:        else
20:            x_B := p ∧ e_j;
21:    else                                             ▷ point x_B is inside
22:        if (c_A land MASK[c]) ≠ 0 then               ▷ new position of x_A
23:            x_A := p ∧ e_i;
24:        else
25:            x_A := p ∧ e_j;
26:    end if
27:    output (x_A, x_B);
28: end procedure
```

5.2. Optimized Line Segment Clipping S-LS-Clip-Opt

For clipping line segments, the line segment S-L-Clip algorithm (see Algorithm 2) is to be modified to take into account positions of the end-points of the given line segment using the **MASK** part of Table 1. The modification uses the end-points codes to determine the case, how the line segment intersects the window. However, computation of the line segment intersection points with the window is needed and the **MASK** determines the appropriate end-point, which is to be replaced by the computed intersection point. It can be seen, that the modification is quite simple, see Algorithm 4.

If the end-points of the line segments are given in homogeneous coordinates, i.e. $w \neq 1$ and $w > 0$, the algorithm Algorithm 3 needs a simple modification, i.e. the conditions are to be changed to $x < x_{\min} * w$, $y < y_{\min} * w$ and similarly for all other cases. It should be noted that in the NDC coordinate system, the conditions are even more simplified, see Algorithm 5.

Algorithm 5 End-point code computation for NDC coordinate system

```
1: procedure CODE (c, x);     ▷ code c for the position x = [x, y : w]^T   w > 0
2:     c := [0000]^T;                                          ▷ initial setting
3:     if  x < 0  then  c := [1000]^T                ▷ setting according to x coordinate
4:         if  x > w  then  c := [0100]^T;     ▷ as x_max = 1  w > 0 then w * 1 = w
5:     if  y < 0  then  c := c lor [0001]^T      ▷ setting according to y coordinate
6:         if  y > w  then  c := c lor [0010]^T;
7:                                  ▷ lor represents or operation on all bits
8: end procedure
```

The end-points classification was simplified for the NDC coordinate system above. The algorithm Smart Line Segment Clip (S-LS-Clip), see Algorithm 4, was designed for the general case, when the end-points of the given line and the corners of the window are given in the projective space, i.e. $w \neq 1$ and $w > 0$.

It means, that the S-LS-Clip algorithm can be further optimized for the case, when clipping is done in the NDC coordinate system. After clipping in the NDC coordinates, the *window-viewport* transformation is applied according to the output device resolution. The transformation can be made in homogeneous coordinates as it uses matrix multiplication, therefore the conversions of the line segment end-points are not needed.

In the NDC case, the CODE computation is to be modified, as the line segments end-points might be given in homogeneous coordinates (see Algorithm 5) and the algorithm for the line segment clipping can be simplified as well. Algorithm 6 is optimized line segment clipping algorithm for the case, when the end-points are given in the Euclidean space. If the end-points are given in homogeneous coordinates, the outer product for the line coefficients is to be used, however, it is one-clock instruction on GPU (cross-product).

Algorithm 6 Optimized S-LS-Clip-Opt line clipping algorithm in E^2

procedure S-L-CLIP-OPT($\mathbf{x}_A, \mathbf{x}_B$); ▷ line is given by two points
  CODE ($\mathbf{c}_A, \mathbf{x}_A$); CODE ($\mathbf{c}_B, \mathbf{x}_B$); ▷ **lor** represents **or** operation on all bits
  if ($\mathbf{c}_A$ **lor** $\mathbf{c}_B$) $= [0000]^T$ **then** ▷ code for the end-points $\mathbf{x}_A$ and $\mathbf{x}_B$
    output ($\mathbf{x}_A$, $\mathbf{x}_B$); **EXIT** ▷ the whole segment is inside
  if ($\mathbf{c}_A$ **land** $\mathbf{c}_B$) $\neq [0000]^T$ **then EXIT** ▷ the whole segment is outside
  $a = y_1 - y_2; \quad b = x_2 - x_1;$ ▷ line coefficients computation
  $c = x_1 * y_2 - x_2 * y_1;$ ▷ if $w \neq 1$ use $[a, b : c]^T := \mathbf{x}_A \wedge \mathbf{x}_B;$
  $c_0 := sign(c); \quad c_1 := sign(a + c);$ ▷ corner's codes computation
  $c_2 := sign(a + b + c); \quad c_3 := sign(b + c);$ ▷ $\mathbf{c} = [c_3, c_2, c_1, c_0]^T$
  if $\mathbf{c} = [0000]^T$ **or** $\mathbf{c} = [1111]^T$ **then EXIT**; ▷ no intersection
  ▷ line segment intersects the window
  $i := TAB1[\mathbf{c}];$ ▷ $\mathbf{x}_A := [x_A, y_A, w_A]^T$
  switch i **do** ▷ equivalent of $\mathbf{x}_A := \mathbf{p} \wedge \mathbf{e}_i;$
    case 0: $x_A := -c; \quad y_A := 0; \quad w_A := a;$
    case 1: $x_A := -b; \quad y_A := a + c; \quad w_A := b;$
    case 2: $x_A := -b - c; \quad y_A := -a; \quad w_A := a;$
    case 3: $x_A := 0; \quad y_A := c; \quad w_A := -b;$
    default: ERROR ▷ actually the N/A case
  end switch
  $j := TAB2[\mathbf{c}];$ ▷ $\mathbf{x}_B := [x_B, y_B, w_B]^T$
  switch j **do** ▷ equivalent of $\mathbf{x}_B := \mathbf{p} \wedge \mathbf{e}_j;$
    case 0: $x_B := -c; \quad y_B := 0; \quad w_B := a;$
    case 1: $x_B := -b; \quad y_B := a + c; \quad w_B := b;$
    case 2: $x_B := -b - c; \quad y_B := -a; \quad w_B := a;$
    case 3: $x_B := 0; \quad y_B := c; \quad w_B := -b;$
    default: ERROR ▷ actually the N/A case
  end switch
  ▷ evaluation of the end-points $\mathbf{x}_i = [x_A, y_i : w_i]^T, i = 1, 2$
  if $\mathbf{c}_A = \mathbf{0}$ **then** ▷ point $\mathbf{x}_A$ is inside
    if ($\mathbf{c}_B$ **land** $MASK[\mathbf{c}]$) $\neq 0$ **then**
      $\mathbf{x}_B := \mathbf{x}_A;$ ▷ new position of $\mathbf{x}_B$
    else
      $\mathbf{x}_B := \mathbf{x}_B;$
  else ▷ point $\mathbf{x}_B$ is inside
    if ($\mathbf{c}_A$ **land** $MASK[\mathbf{c}]$) $\neq 0$ **then** ▷ new position of $\mathbf{x}_A$
      $\mathbf{x}_A := \mathbf{x}_A;$
    else
      $\mathbf{x}_A := \mathbf{x}_B;$
  end if
  output ($\mathbf{x}_A, \mathbf{x}_B$);
end procedure

The algorithm S-L-Clip (see Algorithm 1) and S-LS-Clip (see Algorithm 4) can be easily modified for the line and line segment clipping by a convex polygon, as Table 1 can be generated synthetically for the given number of the convex polygon vertices and the cycle **for** and codes **c** computation must be modified accordingly.

6. Conclusion

Algorithms for the line clipping and line segment clipping by a rectangular window have been deeply studied for a long time and many algorithms and their modifications have been described.

This contribution describes a new line and line segment clipping algorithms with their optimization using principles of geometric algebra extended for the projective extension of the Euclidean space. The algorithms process line and line segments given by end-points in the homogeneous coordinates and use the outer product applied in the projective space. Also, a simple geometric product computation using tensor multiplication is presented.

It should be noted, that the S-L-Clip and S-LS-Clip algorithms can be easily modified for the line and line segment clipping by a convex polygon.

Acknowledgements The author would like to thank to colleagues and students at the University of West Bohemia (Czech Republic), Shandong University and Zhejiang University (China) for their critical comments and constructive suggestions, and to anonymous reviewers for their valuable comments and hints provided.

References

[1] R. Andreev, E. Sofianska: *New algorithm for two-dimensional line clipping*, Computers and Graphics 15.4 (1991), pp. 519–526, ISSN: 00978493, DOI: 10.1016/0097-8493(91)90051-I.

[2] D. Bui, V. Skala: *Fast algorithms for clipping lines and line segments in E2*, Visual Computer 14.1 (1998), pp. 31–37, DOI: 10.1007/s003710050121.

[3] M. Cyrus, J. Beck: *Generalized two- and three-dimensional clipping*, Computers and Graphics 3.1 (1978), pp. 23–28, DOI: 10.1016/0097-8493(78)90021-3.

[4] J. Day: *A New Two Dimensional Line Clipping Algorithm for Small Windows*, Computer Graphics Forum 11.4 (1992), pp. 241–245, ISSN: 01677055, DOI: 10.1111/1467-8659.1140241.

[5] M. Dörr: *A new approach to parametric line clipping*, Computers and Graphics 14.3-4 (1990), pp. 449–464, ISSN: 00978493, DOI: 10.1016/0097-8493(90)90067-8.

[6] L. Dorst, D. Fontijne, S. Mann: *Geometric Algebra for Computer Science (Revised Edition)*, 2009, ISBN: 9780123749420, DOI: 10.1016/B978-0-12-374942-0.X0000-0.

[7] L. J. Dorst L.: *Guide to Geometric Algebra in Practice*, London, Springer, 2011, isbn: 978-0-85729-810-2,
doi: 10.1007/978-0-85729-811-9.

[8] V. Duvanenko, W. Robbins, R. Gyurcsik: *Line-segment clipping revisited*, Dr. Dobb's Journal 21.1 (1996), pp. 107–110, issn: 1044789X.

[9] D. Foley, A. van Dam, S. Feiner, J. Hughes: *Computer graphics: principles and practice*, Boston, MA, USA: Addison-Wesley, 1990, isbn: 0-201-12110-7.

[10] D. Hestenes: *Tutorial on Geometric Calculus*, Advances in Applied Clifford Algebras 24.2 (2014), pp. 257–273, issn: 01887009,
doi: 10.1007/s00006-013-0418-0.

[11] M. Johnson: *Proof by Duality: or the Discovery of New Theorems*, Mathematics Today (1996).

[12] S. Kaijian, J. Edwards, D. Cooper: *An efficient line clipping algorithm*, Computers and Graphics 14.2 (1990), pp. 297–301, issn: 00978493,
doi: 10.1016/0097-8493(90)90041-U.

[13] K. Kanatani: *Understanding Geometric Algebra*, CRC Press, Japan, 2015, isbn: 9780429157127,
doi: 10.1016/B978-0-12-374942-0.X0000-0.

[14] G. Krammer: *A line clipping algorithm and its analysis*, Computer Graphics Forum 11.3 (1992), pp. 253–266, issn: 01677055,
doi: 10.1111/1467-8659.1130253.

[15] E. Lengyel: *Mathematics for 3D Game Programming and Computer Graphics*, Cengage Learning, USA, 2011, isbn: 978-1-4354-5886-4.

[16] Y.-D. Liang, B. Barsky: *A New Concept and Method for Line Clipping*, ACM Transactions on Graphics (TOG) 3.1 (1984), pp. 1–22,
doi: 10.1145/357332.357333.

[17] T. M. Nicholl, D. Lee, R. A. Nicholl: *Efficient New Algorithm for 2-D Line Clipping: Its Development and Analysis*, Computer Graphics (ACM) 21.4 (1987), pp. 253–262,
doi: 10.1145/37402.37432.

[18] A. Rappoport: *An efficient algorithm for line and polygon clipping*, The Visual Computer 7.1 (1991), pp. 19–28,
doi: 10.1007/BF01994114.

[19] V. Skala: *A fast algorithm for line clipping by convex polyhedron in E3*, Computers and Graphics (Pergamon) 21.2 (1997), pp. 209–214,
doi: 10.1016/S0097-8493(96)00084-2.

[20] V. Skala: *Algorithm for 2D line clipping*, New Advances in Computer Graphics, NATO ASI (1989), pp. 121–128,
doi: /10.1007/978-4-431-68093-2_7.

[21] V. Skala: *An efficient algorithm for line clipping by convex polygon*, Computers and Graphics 17.4 (1993), pp. 417–421,
doi: 10.1016/0097-8493(93)90030-D.

[22] V. Skala: *Barycentric coordinates computation in homogeneous coordinates*, Computers and Graphics (Pergamon) 32.1 (2008), pp. 120–127,
doi: 10.1016/j.cag.2007.09.007.

[23] V. Skala: *Intersection Computation in Projective Space Using Homogeneous Coordinates*, Int. Journal of Image and Graphics 8.4 (2008), pp. 615–628,
doi: 10.1142/S021946780800326X.

[24] V. Skala: *Length, Area and Volme Computation in Homogeneous Coordinates*, Int. Journal of Image and Graphics 6.4 (2006), pp. 625–639,
doi: 10.1142/S0219467806002422.

[25] V. Skala: *Line clipping in E2 with O(1) processing complexity*, Computers and Graphics (Pergamon) 20.4 (1996), pp. 523–530,
doi: 10.1016/0097-8493(96)00024-6.

[26] V. Skala: *O(lg N) line clipping algorithm in E2*, Computers and Graphics 18.4 (1994), pp. 517–524,
doi: 10.1016/0097-8493(94)90064-7.

[27] V. Skala, D. Bui: *Extension of the Nicholls-Lee-Nichols algorithm to three dimensions*, Visual Computer 17.4 (2001), pp. 236–242,
doi: 10.1007/s003710000094.

[28] V. Skala, M. Smolik: *A New Formulation of Plücker Coordinates Using Projective Representation*, in: 5th Int. Conf. on Mathematics and Computers in Sciences and Industry (MCSI 2018), IEEE, 2018, pp. 52–56,
doi: 10.1109/MCSI.2018.00020.

[29] M. Sobkow, P. Pospisil, Y.-H. Yang: *A fast two-dimensional line clipping algorithm via line encoding*, Computers and Graphics 11.4 (1987), pp. 459–467, issn: 00978493,
doi: 10.1016/0097-8493(87)90061-6.

[30] J. Vince: *Geometric algebra for computer graphics*, 2008, pp. 1–252, isbn: 9781846289965,
doi: 10.1007/978-1-84628-997-2.

Annales Mathematicae et Informaticae
***52** (2020) pp. 217–228*
DOI: https://doi.org/10.33039/ami.2020.10.001
URL: https://ami.uni-eszterhazy.hu

Fuzzification of training data class membership binary values for neural network algorithms

Tibor Tajti

Eszterházy Károly University
tajti.tibor@uni-eszterhazy.hu

Submitted: August 16, 2020
Accepted: October 21, 2020
Published online: October 21, 2020

Abstract

We propose an algorithm improvement for classifying machine learning algorithms with the fuzzification of training data binary class membership values. This method can possibly be used to correct the training data output values during the training. The proposed modification can be used for algorithms running individual learners and also as an ensemble method for multiple learners for better performance. For this purpose, we define the single and the ensemble variants of the algorithm. Our experiment was done using convolutional neural network (CNN) classifiers for the base of our proposed method, however, these techniques might be used for other machine learning classifiers as well, which produce fuzzy output values. This fuzzification starts with using the original binary class membership values given in the dataset. During training these values are modified with the current knowledge of the machine learning algorithm.

Keywords: Machine learning, neural networks, fuzzification

MSC: 92B20, 03B70, 03B52

1. Introduction

The increasing performance of computers enables the wide use of artificial intelligence and machine learning technologies. These technologies come into our daily

lives, with image recognition, automatic translation, AI assistants, chatbots, autonomous cars, etc. One of the most widely used machine learning algorithms is the Artificial Neural Network and its Deep and Convolutional variants [8, 17]. Neural network algorithms are supervised machine learning algorithms, their major applications include classification, regression, pattern recognition, function approximation, intelligent control, learning from data. The neural network is basically a set of interconnected artificial neurons and the appropriate algorithms working on them [8].

A variation of the multi-layer perceptron model is the convolutional neural network. LeNet was one of the very first convolutional neural networks creating an area of deep learning. Yann LeCun's pioneering work has been named LeNet-5, after many successful iterations [11]. Convolutional networks have shown to be very effective e.g. in image classification [4, 5], natural language processing [6] and time series forecasting [3]. CNNs have a convolution operator, hence the name convolutional network. This convolution operator does feature extraction, e.g. when learning to classify a 2D image, smaller (e.g. 3×3 or 5×5 pixels) parts of the image will be processed as a sliding window over the whole image, so the network learns such smaller-scale features of the images. Committee machines and ensemble methods have been shown to improve the accuracy of neural networks and other machine learning algorithms.

One of the most widely used public datasets is the Modified National Institute of Standards and Technology database (MNIST) [12], which contains 60,000 handwritten numbers in the training set and 10,000 handwritten numbers in the test set. Different classifiers, like K-Nearest Neighbors, SVMs, Neural Nets, Convolutional Neural Nets, proved on this database had shown fail rate down to about 0.2% (20 failures from 10000 test samples) [12]. We have used this dataset for our research.

State-of-the-art architecture as of the time writing this paper is the squeeze-and-excitation network[1] [9].

Modification of training data is often useful for regularization. This can be done by e.g. making distortion, adding noise, using data augmentation [21] or adversarial training [19]. Changing the class membership values of the training data can be considered as one such method.

Usual classification is done providing binary class membership values in the training data, although even for the input patterns for which the classification could be considered uncertain. Fuzzy logic has advantages compared to binary logic having values between false and true as well [1, 2, 10, 15, 22]. Fuzzy logic can be used in machine learning as well, e.g. combining with neural network [7], even with ensemble methods [17]. Using fuzzy class membership values can have performance improvement and this method can also be considered to provide a kind of confidence, which can be an additional advantage in cases where the confidence of the outputs is also required [13].

[1] MNIST classifier with average 0.17% error, 25 February 2020, `https://github.com/Matuzas77/MNIST-0.17/blob/master/MNIST_final_solution.ipynb`

2. Improvements for neural network classifiers

We propose the fuzzification of training data output class membership values. This can be used with standalone learners and with multiple (ensemble) learners as well for better result.

One common problem is that training data usually has binary output values, even when the train samples may belong to more than one class at a certain fuzzy level. [7, 23] These data come usually labeled so that each sample has one or more labels, each of which means the crisp True membership in the class behind that label, and crisp False membership value for the other classes in the same category. There can be cases where these crisp class membership values can be considered misleading, so the correction of these values can lead to reducing the confounding effect of them.

The proposed fuzzification technique might be applied to other classifying algorithms as well, in case they are able to give fuzzy membership values in their output. Research on other algorithms in order to apply the fuzzification technique on them can be a future research, in the current research we conducted our measurements with convolutional neural network algorithms.

We define simple methods which can be used to modify the target output values given for train patterns during the training process to get fuzzy output values from the crisp (binary) values of the training data set. This class membership fuzzification is done so that the knowledge gained during the learning process will be used to correct the inaccurate or incorrect output class membership values of the train patterns. In the following, we will show and describe the proposed algorithm variants. The performance of these algorithm variations will be analysed and shown in Section 3.2.

Three versions of the algorithm will be presented below. The first of them (Algorithm1) is for single learners, the second version (Algorithm2) is for multiple learners the result of which can be used with committee machine voting functions, the third variant (Algorithm3) is a simple modification to handle the parameters of the fuzzification for multiple learners.

```
function FuzzyTraining(model, train_X, train_Y, a, b, c):
    epoch = 0
    fuzzy_Y = train_Y
    while epoch < MAX_EPOCHS and CheckEarlyStopCondition() == False:
      model.fit(train_X, fuzzy_Y, epochs=1)
      out = model.predict(train_X)
      if epoch > START\_FUZZY:
        fuzzy_Y =  a*fuzzy_Y + b*out + c*train_Y
    epoch = epoch + 1
```

Algorithm 1: Fuzzy Training

Algorithm1 must be called with the training data inputs and outputs, and the parameters for the fuzzification for the training of a learning model. The parameter a is the coefficient for the momentum which means the importance of the actual (current) class membership values, which are in the fuzzy_Y vector. This affects

the change from original values towards the desired values. The parameter b is the coefficient for the current knowledge (the out vector), that means the courage to change. The parameter c is the coefficient used for the train_Y vector, which means the importance of the original target output data. The sum of the parameters a, b and c must be 1.0 . Of course the condition when to start the correction must be also considered. In the algorithm presented above, it is a simple condition to have a number of epochs before the first correction. This of course can be changed to an adaptive condition to achieve better performance, however, for our measurement it is more important to know the number of correction operations. When the learning starts, the initial values in the fuzzy_Y vector are the same as given in the train_Y vector.

As it can be seen in Algorithm1, the defined algorithm can work with individual learner algorithm.

We give an extended variant of the algorithm as well to enable to use the combined knowledge of multiple (ensemble) learners. The usage of multiple learners of similar level usually gives better result compared to the individual learners, well-known methods are the committee machines and the ensemble methods [14, 16, 20]. In this version of the algorithm, all the learners will modify the same fuzzy_Y corrected output values, so their combined opinion will have an effect on the subsequent training epochs.

```
function FuzzyTrainingEnsemble(models, train_X, train_Y, a, b, c):
  epoch = 0
  fuzzy_Y = train_Y
  while epoch < MAX_EPOCHS and CheckEarlyStopCondition() == False:
    for model in models:
      model.fit(train_X, fuzzy_Y, epochs=1)
      out = model.predict(train_X)
      if epoch > START_FUZZY:
        fuzzy_Y = a*fuzzy_Y + b*out + c*train_Y
    epoch = epoch + 1
```

Algorithm 2: Fuzzy Training Ensemble

In the case of this new variant of the proposed algorithm (Algorithm2) the correction of the training data outputs will be better, because the combined knowledge of the learners has a better performance compared to the individual results. The correction will also be faster, because after every learning epoch of each individual learner a correction of the training data outputs will be done. In case of multiple learners, parameter a affects the change from original values towards the desired values and it affects the averaging effect on the outputs of multiple learners too. In a future development, it might be useful to change the algorithm with an additional parameter to separately control these two effects.

In this case, the number of times the correction statement will be run is the number of (epochs - START_FUZZY) multiplied by the number of the learners. This can be taken into account when setting the parameters for the training data output value fuzzification. We provide a modification to Algorithm2 with a simple normalization with respect to the number of learners.

Let M be the number of learners, a, b and c the weights for the train output class membership value fuzzification as described for the algorithm. We can calculate the normalized a', b' and c' weights as follows:

- $a' = \sqrt[M]{a}$
- $b' = \frac{(1-a')b}{b+c}$
- $c' = 1 - (a' - b')$

The parameters a', b', c' now correspond to the parameters a, b, c so that the speed of convergence with M learners giving the same output will be the same as the speed of convergence would be using the parameters a, b and c with one learner. Now we apply the above formulae to get the new version of this algorithm.

```
function FuzzyTrainingEnsemble2(models, train_X, train_Y, a, b, c):
  epoch = 0
  fuzzy_Y = train_Y
  a = power(a, 1/len(models))
  b = (1-a)*b/(b+c)
  c = 1-(a+b)
  while epoch < MAX_EPOCHS and CheckEarlyStopCondition() == False:
    for model in models:
      model.fit(train_X, fuzzy_Y, epochs=1)
      out = model.predict(train_X)
      if epoch > START_FUZZY:
        fuzzy_Y = a*fuzzy_Y + b*out + c*train_Y
    epoch = epoch + 1
```

Algorithm 3: Fuzzy Training Ensemble 2

In Algorithm3 the normalization of the parameters has to be done only once, before the training loop. Certainly it might be possible to adaptively change the parameters during the training process, the research of this can be conducted in the future. Note that we have overwritten the original values of the parameters a, b and c. If this is not the desired behavior then these values can be preserved. Since with given number of learners and given (not changing) a, b and c parameters the difference between the Algorithm2 and Algorithm3 variants lies only on changing the parameters, we have not conducted any separate measurements on Algorithm3.

3. Performance evaluation of fuzzification of training data binary class membership values

3.1. Performance evaluation framework

The experiments ran on personal computers equipped with NVIDIA and AMD GPUs using Tensorflow from Python programs. Our simple framework was based on file interface which enables to run the machine learning on multiple machines, and then later collected and processed the output files generated by the learners.

For the research, two convolutional neural network learning algorithms with different strength have been chosen as the basis of the modifications.

The problem set given to the learning algorithms was the well known MNIST database of handwritten digits [12].

The results may vary given the stochastic nature of the algorithms, so thousands of experiments with different parameters were performed, and average results were analysed. For the analyses we first measured the performance of the individual learners on the test dataset with different parameters for fuzzification. Since multiple learners have proven to be more successful when we combine their results through voting, we might expect better results by setting the fuzzified class membership together as well. We will show the results of this research in Section 3.2. In these experiments we have measured the standalone test results of the learners, as well as the results of the fuzzy average voting of multiple learners, as a committee machine. When we talk about committee machine voting we can choose from many voting functions, e.g. fuzzy averaging, plurality (or majority) voting, etc. In our research we have used the well-known fuzzy voting [18].

For the analyses we used the Python Numpy and Pandas frameworks. The algorithms run with different epoch counts to see the behavior of our proposed algorithm variations not only with the statistically best settings. In the following sub-section we will show the performance of the proposed fuzzification of training data binary class membership values. For the evaluation we run about one million learning sessions with convolutional neural network algorithms modified according to our proposed methods. The first algorithm variant was built from the algorithm introduced in[2]. The second algorithm variant was based on the algorithm which uses the Squeeze-and-Excitation Network method. We have added to both algorithms the proposed fuzzification of binary class membership values of training data.

3.2. Performance evaluation of training data class membership value fuzzification

We have executed several experiments with two algorithms of different strengths. The algorithm variations were executed with different parameters, e.g. the number of epochs to run, the number of instances in the ensembles and the parameters for the fuzzification of binary class membership values of training data, including parameters which keep the original class membership values. We note that we have executed many learning sessions without fuzzification as well in order to have more reliable results for comparison.

3.2.1. Fuzzification experiment 1

The first experiment ran using the modified variant of. Thousands of learners learned with different epoch counts and different parameters for fuzzification, in-

[2] https://www.kaggle.com/cdeotte/25-million-images-0-99757-mnist

cluding parameters which keep the original class membership values($a = 0$, $b = 0$, $c = 1$). We will first show the average results in function of the $c/(b + c)$ ratio.

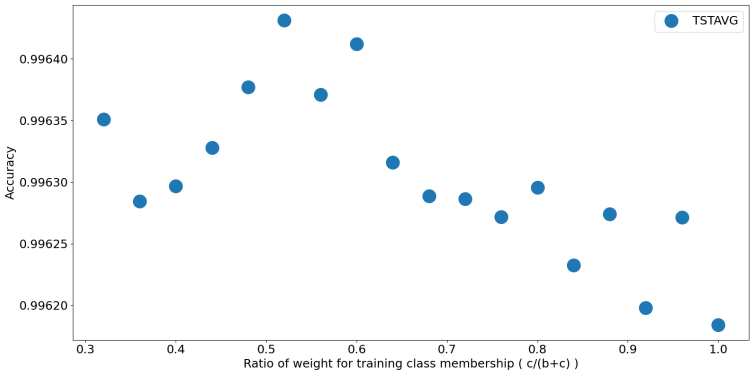


Figure 1: The average accuracy results of our algorithm on test data using different parameters for the fuzzification of the training data class membership values.

Figure 1 shows the average accuracy of the individual learners with different parameters used for the fuzzification. The ratio $c/(b + c)$ of the parameters of Algorithm1 has the meaning of how important the original binary class membership values provided in the training data are. If the ratio is 1.0 , then no fuzzification will happen. As it can be seen, the accuracy achieved was better when the algorithm was used with fuzzification. We note that when the ratio goes below 50% then the performance gets again lower. In that case the fuzzification can change the class membership values to have a big difference from the original values. We will also show the performance using the fuzzy average voting function when using multiple learners.

Figure 2 shows the accuracy of the V1 fuzzy average voting on the same experiment. As we can see, the results using the fuzzy average voting are similar, the fuzzification helps to achieve better performance, i.e. higher accuracy on the training dataset. As the ratio of $c/(b + c)$ increases, i.e., the possibility of fuzzification decreases, so the accuracy achieved tends to decrease as well. We note, that although the shown results are mean values of several measurements, the random behavior of the algorithms can result in fluctuation in performance, some values can be the effect of that. We also show a 3D diagram to better understand the results for different parameters. Since the sum of the parameters a, b and c must be 1.0 we can choose two of these parameters for the X and Y axes of the diagram, and the Z axis can show the average accuracy values. We have chosen the a and b parameters for that, the c parameter for every measurement is $1 - (a + b)$.

Figure 3 shows the average individual accuracy on test data for the a and b parameters. The value with $a = 0$, $b = 0$ coordinates shows the average result without fuzzification. We can see that with values of parameter b around 0.4

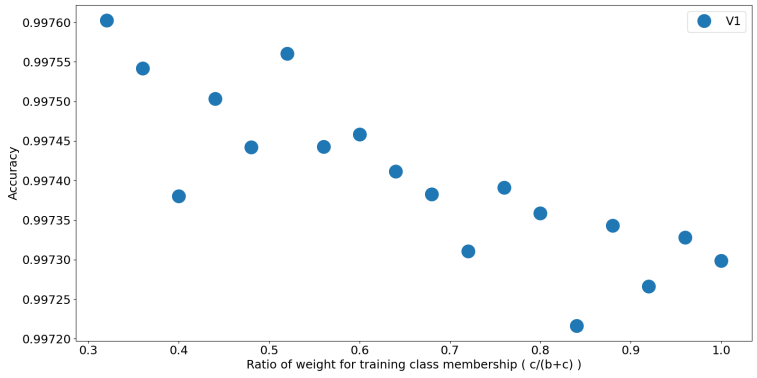


Figure 2: The performance results of our algorithms V1 fuzzy average voting function by 6-20 voters on test data using different parameters for the fuzzification of the training data class membership values.

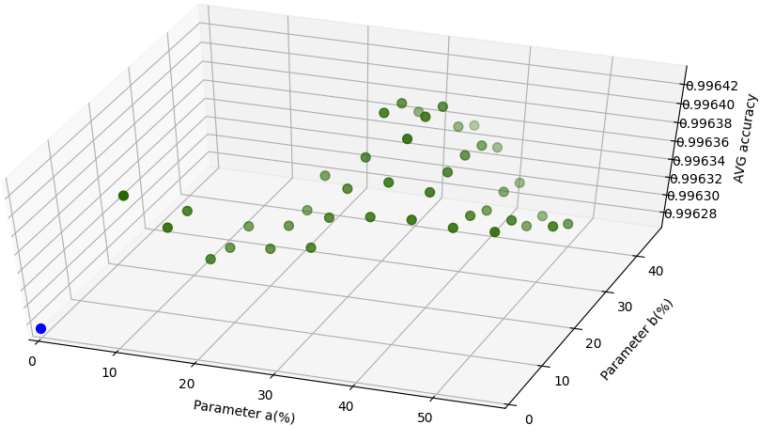


Figure 3: The average performance results of our algorithms on test data using different parameters for the fuzzification of the training data class membership values.

(40%) we had better accuracy, especially when the value of parameter a was close to 0.3 (30%).

3.2.2. Fuzzification experiment 2

The next experiment ran using a modified algorithm of , using the parameters for the fuzzification. The number of epochs we had run the algorithm was from 15 to 20.

Figure 4 shows the average accuracy of the individual learners with different

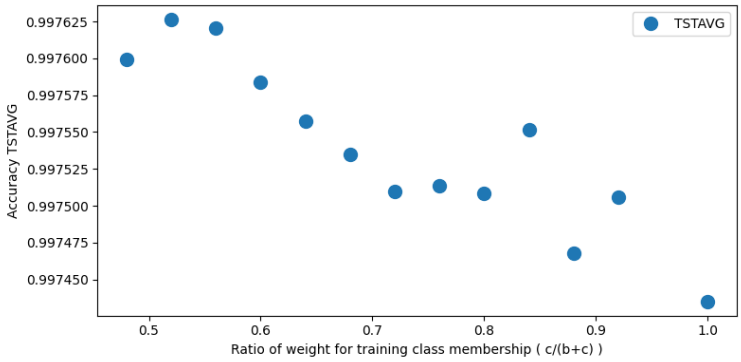


Figure 4: The individual accuracy results of the algorithm on test data using different parameters for the fuzzification of the training data class membership values.

parameters used for the fuzzification. As described for Figure 1 the ratio $c/(b+c)$ tells the importance of the original class membership values of the training data, fuzzification can be done only if the ratio is below 1.0 . As we can see, the accuracy can be better with modest fuzzification. We also note that if the ratio of $c/(b+c)$ decreases to below 0.5 then the accuracy seems to decrease as well. This can be the effect of too much freedom of the algorithm to change the class membership values. For this experiment, too, we have measured the performance using the well-known fuzzy average voting function (V1) when using multiple (6–20) learners.

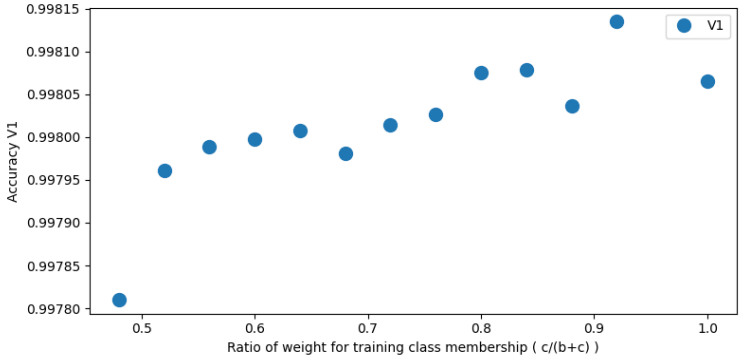


Figure 5: The individual accuracy results of algorithm on test data using different parameters for the fuzzification of the training data class membership values.

Figure 5 shows the results of the V1 fuzzy mean vote in this experiment. The results are different in this case. The accuracy averages using fuzzified training

data class membership values were lower for most parameters than the accuracy using only the original training data. However, there is a promising range what we can look from another perspective as well. Below we show the average accuracy of the learners for different a, b and c parameters on a 3D figure using the fuzzy average (V1) voting function with 6–20 voters. We show the results in function of a and b parameters, while parameter c is dependent on them.

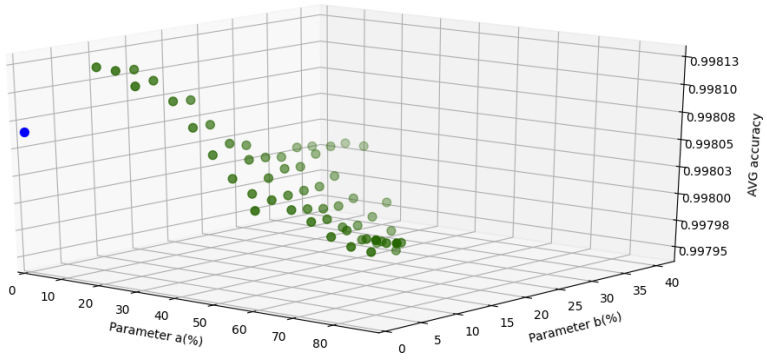


Figure 6: The results of our algorithms' average accuracy on test data using different parameters for the fuzzification of the training data class membership values.

Figure 6 shows the results of thousands of learning sessions which were executed with different a, b and c parameters. The point with $a = 0$, $b = 0$ coordinates shows the average result when class membership values of training data were not corrected. We can see that the results were higher with lower a and b parameter values. For such parameters the c parameter is higher, so only minor corrections on the training data class membership values can be made.

We note that this is a strongly different behavior compared to the performance of fuzzification with the first (weaker) algorithm variant. This is probably because the Squeeze-and-Excitation Network has much higher accuracy on this dataset, and this might mean that it can handle misclassified train samples better, so fuzzification may result only in minor improvement.

For a range of parameter values where parameter a and b are not zero but both have low values the accuracy was better using the proposed fuzzification. That means that fuzzification in a lower rate had an improvement even for this strong algorithm.

4. Conclusion

From the results of our fuzzification experiments we can conclude that the fuzzification of the training data binary class membership values can improve the accuracy of the prediction of class membership values.

The results show that the parameters of our proposed fuzzification algorithm highly affect the accuracy of the predictions of the learners. Their effect was different depending on the basic algorithm to which we added it. The performance improvement of individual test accuracy was significant for both algorithms we used for the evaluation. When we compared the accuracy of the fuzzy average voting function for different parameters of the fuzzifying algorithm we had also significant improvement for the weaker algorithm with wider range of the parameters of the fuzzification algorithm, but in case of the stronger algorithm only a minor improvement was observed for a narrow range of these parameters. Further measurements will be performed to analyze this behavior with the same dataset and with other datasets as well.

References

[1] R. Basbous, B. Nagy, T. Tajti: *Short Circuit Evaluations in Gödel Type Logic*, Proc. of FANCCO 2015: 5th International Conference on Fuzzy and Neuro Computing, Advances in Intelligent Systems and Computing 415 (2015), pp. 119–138, doi: https://doi.org/10.1007/978-3-319-27212-2_10.

[2] R. Basbous, T. Tajti, B. Nagy: *Fast Evaluations in Product Logic: Various Pruning Techniques*, in: FUZZ-IEEE 2016 - the 2016 IEEE International Conference on Fuzzy Systems, Vancouver, Canada: IEEE, 2016, pp. 140–147, doi: 10.1109/FUZZ-IEEE.2016.7737680.

[3] A. Borovykh, S. Bohte, C. W. Oosterlee: *Conditional time series forecasting with convolutional neural networks*, arXiv preprint arXiv:1703.04691 (2017).

[4] D. Ciresan, U. Meier, J. Schmidhuber: *Multi-column deep neural networks for image classification*, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 3642–3649.

[5] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber: *Flexible, high performance convolutional neural networks for image classification*, in: Twenty-second international joint conference on artificial intelligence, 2011.

[6] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun: *Very deep convolutional networks for text classification*, arXiv preprint arXiv:1606.01781 (2016).

[7] R. Fullér: *Fuzzy systems*, in: Introduction to Neuro-Fuzzy Systems, Springer, 2000, pp. 1–131.

[8] S. Haykin: *Neural Networks: A Comprehensive Foundation*, 2nd, USA: Prentice Hall PTR, 1998, isbn: 0132733501.

[9] J. Hu, L. Shen, G. Sun: *Squeeze-and-excitation networks*, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[10] G. Kovásznai, C. Biró, B. Erdélyi: *Puli–A Problem-Specific OMT solver*, in: Proc. 16th International Workshop on Satisfiability Modulo Theories (SMT 2018), 371, 2018.

[11] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner: *Gradient-based learning applied to document recognition*, Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

[12] Y. LeCun, C. Cortes, C. J. Burges: *The MNIST database of handwritten digits, 1998*, URL http://yann. lecun. com/exdb/mnist 10.34 (1998), p. 14.

[13] L. Li, Q. Hu, X. Wu, D. Yu: *Exploration of classification confidence in ensemble learning*, Pattern recognition 47.9 (2014), pp. 3120–3131.

[14] U. Naftaly, N. Intrator, D. Horn: *Optimal ensemble averaging of neural networks*, Network: Computation in Neural Systems 8.3 (1997), pp. 283–296,
doi: 10.1088/0954-898X_8_3_004, eprint: https://doi.org/10.1088/0954-898X_8_3_004,
url: https://doi.org/10.1088/0954-898X_8_3_004.

[15] B. Nagy, R. Basbous, T. Tajti: *Lazy evaluations in Łukasiewicz type fuzzy logic*, Fuzzy Sets and Systems 376 (2019), Theme: Computer Science, pp. 127–151, issn: 0165-0114,
doi: https://doi.org/10.1016/j.fss.2018.11.014,
url: http://www.sciencedirect.com/science/article/pii/S0165011418309357.

[16] D. Opitz, R. Maclin: *Popular ensemble methods: An empirical study*, Journal of artificial intelligence research 11 (1999), pp. 169–198.

[17] S. Russell, P. Norvig: *Artificial intelligence: a modern approach* (2002).

[18] C. Sammut, G. I. Webb: *Encyclopedia of machine learning*, Springer Science & Business Media, 2011.

[19] F. Tramèr, A. Kurakin, N. Papernot, et al.: *Ensemble adversarial training: Attacks and defenses*, arXiv preprint arXiv:1705.07204 (2017).

[20] S. Wan, H. Yang: *Comparison among Methods of Ensemble Learning*, 2013 International Symposium on Biometrics and Security Technologies (2013), pp. 286–290.

[21] S. C. Wong, A. Gatt, V. Stamatescu, M. D. McDonnell: *Understanding data augmentation for classification: when to warp?*, in: 2016 international conference on digital image computing: techniques and applications (DICTA), IEEE, 2016, pp. 1–6.

[22] L. A. Zadeh, G. J. Klir, B. Yuan: *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*, WORLD SCIENTIFIC, 1996,
doi: 10.1142/2895, eprint: https://www.worldscientific.com/doi/pdf/10.1142/2895,
url: https://www.worldscientific.com/doi/abs/10.1142/2895.

[23] L. A. Zadeh: *Fuzzy logic—a personal perspective*, Fuzzy Sets and Systems 281 (2015), Special Issue Celebrating the 50th Anniversary of Fuzzy Sets, pp. 4–20, issn: 0165-0114,
doi: https://doi.org/10.1016/j.fss.2015.05.009,
url: http://www.sciencedirect.com/science/article/pii/S0165011415002377.

Annales Mathematicae et Informaticae
***52** (2020) pp. 229–242*
DOI: https://doi.org/10.33039/ami.2020.10.003
URL: https://ami.uni-eszterhazy.hu

New voting functions for neural network algorithms

Tibor Tajti

Eszterházy Károly University
tajti.tibor@uni-eszterhazy.hu

Submitted: August 16, 2020
Accepted: October 21, 2020
Published online: October 21, 2020

Abstract

Neural Network and Convolutional Neural Network algorithms are among the best performing machine learning algorithms. However, the performance of the algorithms may vary between multiple runs because of the stochastic nature of these algorithms. This stochastic behavior can result in weaker accuracy for a single run, and in many cases, it is hard to tell whether we should repeat the learning giving a chance to have a better result. Among the useful techniques to solve this problem, we can use the committee machine and the ensemble methods, which in many cases give better than average or even better than the best individual result. We defined new voting function variants for ensemble learner committee machine algorithms which can be used as competitors of the well-known voting functions. Some belong to the locally weighted average voting functions, others are meta voting functions calculated from the output of the previous voting functions functions called with the results of the individual learners. The performance evaluation of these methods was done from numerous learning sessions.

Keywords: Machine learning, neural networks, committee machines, ensemble methods

MSC: 92B20, 03B70, 03B52

1. Introduction

One of the most widely used machine learning algorithms is the Artificial Neural Network or its Deep and Convolutional variants [7, 12, 19]. Neural network

algorithms are supervised machine learning algorithms, widely used in machine learning. Its major applications include classification, regression, pattern recognition, function approximation, intelligent control, learning from data. The neural network is a set of interconnected artificial neurons and the appropriate algorithms working on them [7].

A variation of the multi-layer perceptron model is the convolutional neural network. LeNet was one of the very first convolutional neural networks creating an area of deep learning. Yann LeCun's pioneering work has been named LeNet-5, after many successful iterations [12]. CNNs have a convolution operator, hence the name convolutional network. This convolution operator does feature extraction, e.g. when learning to classify a 2D image, smaller (e.g. 3×3 or 5×5 pixels) parts of the image will be processed as a sliding window over the whole image, so the network learns such smaller-scale features of the images.

The knowledge of experts can be very useful in machine learning as well. When several learner algorithms learn the same problem or parts of the problem their knowledge can be combined in numerous ways [5, 17, 23]. This can be used both for getting satisfactory results from weak learners and for reaching top performance when using strong learners. Since multiple learners have proven to be more successful when we combine their results through voting, we defined new voting functions and measured their performance with ensemble learners in different group sizes. Committee machine algorithms and ensemble methods use multiple neural networks or other machine learning algorithms to make predictions and combine their results [22]. This can work with multiple instances of the same algorithm (e.g. [4]) or different algorithms or models (e.g. [9]) as well. Several simple committee machine variants are used efficiently with committees voting on the same problem and combining their results with voting functions.

Note that many voting functions are available, e.g. minimum, maximum, median voting [10]. We use the most well-known voting functions: fuzzy average, weighted fuzzy average, plurality, borda and product voting.

Ensemble methods have been very successful in setting record performance on challenging data sets [17]. Ensemble learners can also be used combined with other methods that can be used with machine learning algorithms, e.g. the fuzzification of training data binary class membership values [21], to have the advantage of using fuzzy truth values instead of the binary truth values [2, 3, 6, 8, 16, 24].

The most well-known committee machine voting functions are described in the followings. For each voting function, first let o_i be the actual output vector of class membership values predicted by learner i for the actual sample given as input.

We note that training data can be changed dynamically, e.g. for time series prediction often we get new training data periodically.

1.1. Voting functions

1.1.1. Fuzzy average voting

Averaging is one of the most simple linear combiner voting schemes having the $1/N$ weight for the outputs of each learner [20]. Calculate the average of the individual predictions: $o[j] = \frac{1}{N}\sum_{i=1}^{n} o_i[j]$ for each j output class, where N is the number of learners, $o_i[j]$ is the jth element (class membership value) in the output vector of the prediction. Then find for each sample the class with the highest membership value as the chosen class for the given sample ($l = \text{argmax}(o)$).

1.1.2. Plurality voting [18]

Find for each learner i, the class with the highest membership value from the prediction o_i. If it is at index h_i ($h_i = \text{argmax}(o_i)$), then then let

$$c_i[j] = \begin{cases} 1, & \text{if } j = h_i, \\ 0, & \text{otherwise,} \end{cases}$$

for all j classes.

Then calculate the sum $c[j] = \frac{1}{N}\sum_{i=1}^{n} c_i[j]$ for each j classes, where N is the number of learners. The winner of the voting for the sample is a class with the maximum value $l = \text{argmax}(c)$. We note, that sometimes this method is called majority voting, although majority voting means choosing the winner only if more than 50% of the learners have voted on it. When using majority voting it is recommended to use an odd number of voters.

1.1.3. Borda voting [1]

For each individual learner i, calculate the index $s_i[j]$ in order of the membership values from the prediction $o_i[j]$. Let $s_i[j]$ be n if $o_i[j]$ has the nth smallest value, for each j class for each i learner. Then calculate the sum $s[j] = \frac{1}{N}\sum_{i=1}^{n} s_i[j]$ for each j classes, where N is the number of learners used for the prediction. The winner of the voting is a class with the maximum values $l = \text{argmax}(s)$.

1.1.4. Nash (product) voting [1]

For each class j evaluate the product of the predictions of all of the i individual learners: $o[j] = \prod_{i=j}^{N} o_i[j]$ Then find for each sample the class with the highest membership value ($l = \text{argmax}(o)$).

We note that the fuzzy voting and the product voting can be used for regression as well, while plurality voting and borda voting are suitable for classification only. These voting functions can be applied simply on the predictions of the individual learners which have learned either sequentially or in parallel.

2. New voting functions for neural network classifiers

We propose the addition of new variants for committee machine voting functions which in some cases might have better performance compared to the well-known voting functions. We note that our experiment was done using convolutional neural network classifiers, however, these voting functions might be used for every classifier which can produce fuzzy output values, as well. The good performance and the variety of the well-known committee machines motivated us to develop our new ones. We defined the following new committee machine voting functions which we will compare with some of the well-known voting functions. Some of the proposed new voting functions belong to the locally weighted average voting functions [22], others are meta voters using the previous ones.

2.1. Fuzzy average voting weighted by the confidence

Fuzzy average voting can be weighted by confidence [14]. Here we propose a simple function with getting a confidence from the class membership values. This method obviously needs less performance compared to other more advanced methods. Class membership values closer to 0 or 1 will have stronger weight, we transform the output of the individual learners before calculating the fuzzy average, so that the values which are considered uncertain (not close to 0 or 1) values will be less important by multiplying with a smaller weight. Given the network output $o_i[j]$ for each i learners for each j classes we calculate the combined result with the following formula:

$$o[j] = \frac{1}{N}\sum_{i=1}^{N}((o_i[j] - 0.5)(2o_i[j] - 1)^2 + 0.5).$$

Then we get the winner class from this weighted average: $l = \text{argmax}(o)$.

2.2. Fuzzy average voting weighted by 1-difference from the combined output

Knowing the outputs of the learners we can base another weighted average method based on the better performance of the fuzzy average compared to the individual learners. Starting with the calculation of the fuzzy average, individual predictions will be multiplied by a weight that is the difference from the ensemble prediction subtracted from 1. Let $o[j]$ be calculated as defined for the fuzzy voting in Section 1. Then we calculate the new variant as follows:

$$o'[j] = \frac{1}{N}\sum_{i=1}^{N}((o_i[j] - 0.5)(1 - |o_i[j] - o[j]|) + 0.5).$$

We can find the winner class from the weighted average: $l = \text{argmax}(o')$.

2.3. Fuzzy average voting weighted by the reciprocal value of the number of failed training samples

Let f_i be the number of failed (misclassified) samples for each learner i, of the training dataset. The reciprocal value of f_i will be used as the weight for the learner i if f_i is not equal to 0, otherwise we use a maximal weight, e.g. 2.

$$o[j] = \frac{1}{N} \sum_{i=1}^{N} \frac{o_i[j]}{f_i}$$

From this weighted average we get the winner class: $l = \text{argmax}(o)$.

2.4. Geometric mean (Nash voting with Nth root)

We create a variant of the Nash (product) vote function for using in meta voting function as well. Since with higher number of voters (N) the product of many values from the interval $[0, 1]$ can be a very small number, much smaller than e.g. the fuzzy average, so we take the Nth root of the product, getting the geometric mean of the output values. We note that the geometric mean will choose the same winner as the Nash (product) voting, since the Nth root function is strictly monotonically increasing over the interval $[0, 1]$. For each class j evaluate the product sum of the predictions of all of the i individual learners:

$$o[j] = \sqrt[N]{\prod_{i=1}^{N} o_i[j]}.$$

Then find the class with the highest membership value ($l = \text{argmax}(o)$).

2.5. Meta-voting variants

Fuzzy average or plurality vote by combining selected voting functions by calculating the fuzzy average or the plurality of votes on the classes of the results of the selected voting functions. For analysis purposes, we define three meta voter variants.

- V8: Plurality voting from the results of V1, V2, V3, V4, V5, V6, V7
- V9: Plurality voting from the results of V1, V2, V3, V4, V7
- V10: Fuzzy average voting from the results of V1, V2, V3, V4, V7

For the above three meta voting functions, we calculate the results of the needed voting functions first, then we combine them as it was described above for the voting functions calculated from the results of the individual learners.

We note that any data used to calculate the weights certainly can only be part of the training data or result of the learning process, without any knowledge

about test data or performance on test data. We also note that plurality vote and borda vote functions do not give fuzzy class membership values, so they cannot be combined well by fuzzy average with the fuzzy results of other voters. So for the performance evaluation, we will use three meta voter functions described above (V8, V9, V10) for the better understanding and comparison possibility.

3. Performance evaluation of voter functions

3.1. Performance evaluation framework

We performed our evaluation using NVIDIA and AMD GPUs with the Tensorflow framework. Our simple system was based on a file interface allowing to run on multiple machines. For the experiments, we have used two convolutional neural network learning algorithms with different strength. They were built as modified variants of [15].

We used the MNIST database of handwritten digits [13] to perform our research. The accuracy results may vary because of the stochastic nature of the algorithms, so many learning sessions were executed, and their average results were analysed. We can choose from many voting functions, e.g. fuzzy averaging, plurality(or majority) voting, etc. In our research, we have compared the results of some of the most well-known voting functions with our newly defined ones.

For the analyses, we used the Python Numpy and Pandas frameworks. The algorithms run with different epoch counts to see the behavior of our proposed algorithm variations not only with the statistically best settings. In the following subsection we will show the performance of the proposed voting functions. For the evaluation we run about one million learning sessions with three convolutional neural network algorithms modified according to our proposed methods.

We have executed several experiments with two algorithms of different strengths. The first algorithm variant was built from the algorithm introduced in, the second variant was developed based on the algorithm. The algorithm variations were executed with different parameters, e.g. number of epochs to run, number of instances in the ensembles and parameters for the fuzzification of binary class membership values of training data, including parameters which keep the original class membership values. We note that we have executed many learning sessions without fuzzification in order to have more reliable results for comparison.

3.2. Performance of voting functions

For the evaluation, we have included the well-known voter schemes and our new variants as well. We have implemented the following voting functions:

- V1: fuzzy voting, i.e. averaging
- V2: fuzzy variant – average of individual predictions weighted by a confidence estimation of the class membership values

- V3: fuzzy variant – average of individual predictions weighted by 1-difference from V1 results, predictions will be multiplied by a weight which is the difference from the ensemble prediction subtracted from the value 1.0
- V4: fuzzy variant – average of individual predictions weighted by 1/training failures
- V5: plurality voting
- V6: borda voting
- V7: geometric mean voting (instead of product voting)
- V8: meta-voter: plurality vote by using all the above voting functions
- V9: meta-voter: plurality meta vote of voters without the plurality and borda voting (V1–V4, V7)
- V10: meta voter: fuzzy average meta vote of voters without the plurality and borda voting (V1–V4, V7)

We note that variations of the plurality vote also can be applied [11] however plurality and borda votes are not among the best performing voting functions according to our measurements, we included them for reference and comparison purposes.

3.2.1. Voting experiment 1 with algorithm based on [14]

Our first experiment on voting schemes has been run 1000 times. In each turn 6–20 voters voted with the voting functions (V1–V10) described above.

Voting function	**MIN**	**AVG**	**MAX**
max(accuracy)	0.995600	0.997043	0.998000
avg(accuracy)	0.995188	0.996306	0.997260
min(accuracy)	0.992700	0.995422	0.996900
V1 – fuzzy average	0.996000	0.997351	0.998400
V2 – weighted by confidence	0.995900	0.997360	0.998300
V3 – weighted by diff from V1	0.996000	0.997346	0.998300
V4 – weighted by 1/failures	0.995500	0.997321	0.998300
V5 – plurality voting	0.995500	0.997280	0.998400
V6 – borda voting	0.995600	0.997296	0.998400
V7 – geometric mean voting	0.996000	0.997367	0.998300
V8 – meta – plurality (V1–V7)	0.995900	0.997353	0.998400
V9 – meta – plurality (V1–V4, V7)	0.996000	0.997356	0.998400
V10 – meta – fuzzy avg (V1–V4, V7)	0.996100	0.997350	0.998300

Table 1

Table 1 shows the accumulated results of the tested voting functions with the minimum, average and maximum number of the failed samples of the individual learners included. The best individual result was 20 fails of 10000 test samples, the worst was 73 failed samples and on average they performed as low as 36.94 fails from 10000 samples as individual learners. The well-known fuzzy voting performed 26.49 fails on average. There were no big differences among the voting functions, the best result came from the product voting (V7) from committee results on training failures. For some of the best performing voting functions (V1, V2, V7) we also show the accuracy achieved by them with different number of voters.

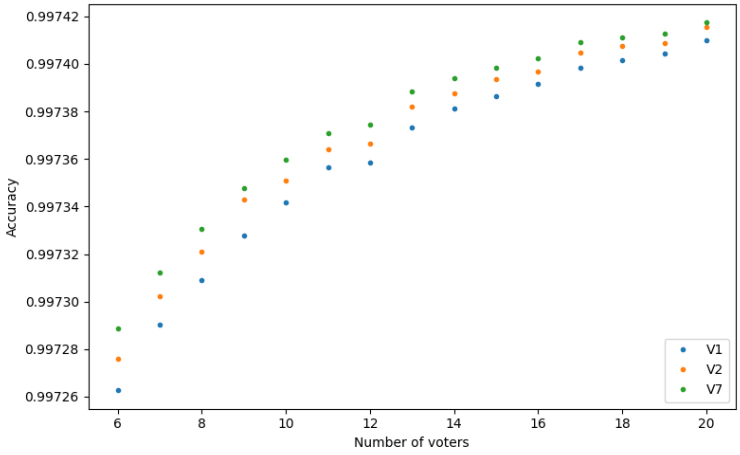


Figure 1: The performance results of our algorithm with V1 fuzzy average voting function by 6-20 voters on average on test data using different epoch counts (15, 17, 19, 20).

As we can see on Figure 1 the three voting functions show similar behavior. All of them were performing better with more voters.

3.2.2. Voting experiment 2 with algorithm based on [15]

The second experiment ran 1000 training sessions on a slightly better algorithm, a modified version of [15]. It was executed with different epoch counts (15, 17, 19, 20) to eliminate the effect of a possibly statistically optimized epoch count for a specific dataset.

Table 2 shows the results where in each turn 6–20 voters cast their votes which were then combined using the voter functions defined above. The best individual result was 15 fails from 10000 test samples, the worst individual result was 44 failed samples and on average they performed only 27.55 fails as individual learners. The well-known fuzzy voting performed 21.41 fails on average. There were no big differences among the voting functions, the best average result (21.28 fails on

Voting function	**MIN**	**AVG**	**MAX**
max(accuracy)	0.996700	0.997795	0.998500
avg(accuracy)	0.996580	0.997245	0.997880
min(accuracy)	0.995600	0.996662	0.997600
V1 – fuzzy average	0.996700	0.997859	0.998700
V2 – weighted by confidence	0.996600	0.997863	0.998600
V3 – weighted by diff from V1	0.996700	0.997857	0.998700
V4 – weighted by 1/training failures	0.996800	0.997872	0.998700
V5 – plurality voting	0.996500	0.997786	0.998600
V6 – borda voting	0.996600	0.997791	0.998600
V7 – geometric mean voting	0.996700	0.997861	0.998700
V8 – meta – plurality (V1-V7)	0.996700	0.997860	0.998700
V9 – meta – plurality (V1-V4,V7)	0.996700	0.997862	0.998700
V10 – meta – fuzzy avg (V1-V4,V7)	0.996700	0.997857	0.998700

Table 2

average) came from the fuzzy voting weighted by the reciprocal value of training failures (V4).

Also, we can check whether the difference between the voting functions depends on the number of voters. On the next figure, we can check that for three of the best performing voting functions.

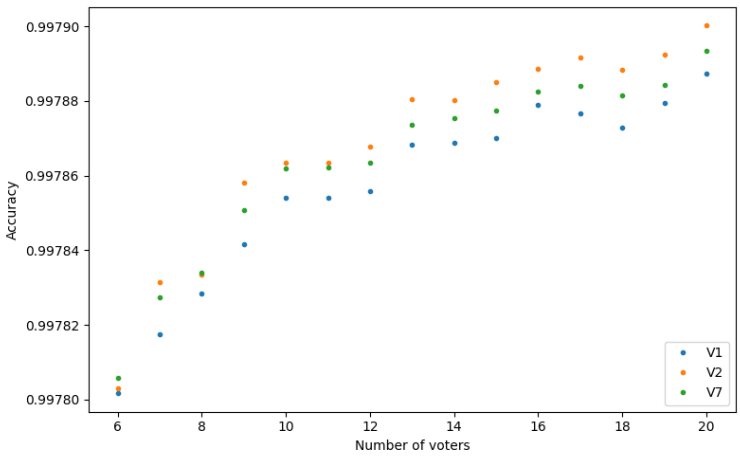


Figure 2: The performance results of our algorithm with V1 fuzzy average voting function by 6-20 voters on average on test data using different parameters for the fuzzification of the training data class membership values.

3.2.3. Voting experiment 3 with algorithm based on [15]

The third experiment ran also 1000 times on a modified version of [15]. It was executed with 20 epochs for each learner.

Voting function	MIN	AVG	MAX
max(accuracy)	0.997100	0.997914	0.998500
avg(accuracy)	0.996840	0.997418	0.997886
min(accuracy)	0.996200	0.996878	0.997700
V1 – fuzzy average	0.997300	0.998126	0.998700
V2 – weighted by confidence	0.997300	0.998122	0.998700
V3 – weighted by diff from V1	0.997200	0.998128	0.998700
V4 – weighted by 1/failures	0.997300	0.998126	0.998700
V5 – plurality voting	0.997000	0.998044	0.998700
V6 – borda voting	0.997000	0.998044	0.998700
V7 – geometric mean voting	0.997100	0.998126	0.998700
V8 – meta – plurality (V1–V7)	0.997200	0.998125	0.998700
V9 – meta – plurality (V1–V4, V7)	0.997200	0.998128	0.998700
V10 – meta – fuzzy avg (V1–V4, V7)	0.997200	0.998129	0.998700

Table 3

Table 3 shows the results where in each turn 6-20 voters voted using the above-defined voter functions. The best individual result was 15 fails from 10000 test samples, the worst individual result was 38 failed samples and on average they performed only 25.82 fails as individual learners. The well-known fuzzy voting performed 18.74 on average. There were no big differences among the voting functions, the best result (18.71 fails) came from our meta fuzzy voter function (V10).

3.2.4. Voting experiment 4 with algorithm based on [15]

Our last experiment to compare voting functions also ran 1000 training sessions on a similar modified version of [15]. This time we also added a 0.2 dropout to the algorithm. Dropout is a useful regularization method, which helps to eliminate the overfitting in general, as well as in our case is useful for the fuzzification of the training data class membership values.

Table 4 shows the results of the experiment where in each turn 6–20 voters voted using the voting functions V1–V10. The best individual result was 17 fails from 10000 test samples, the worst individual result was 42 failed samples and on average they performed only 26.35 fails as individual learners. The well-known fuzzy voting performed 21.76 fails on average. There were no big differences among the voting functions, the best result came from our fuzzy voting weighted by the reciprocal value of training failures (V4).

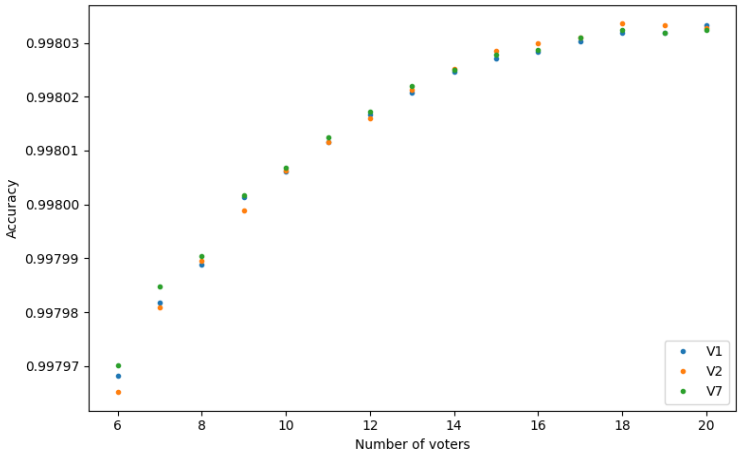


Figure 3: The performance results of our algorithm with V1 fuzzy average voting function by 6-20 voters on average on test data using different parameters for the fuzzification of the training data class membership values.

Voting function	**MIN**	**AVG**	**MAX**
max(accuracy)	0.996900	0.997805	0.998300
avg(accuracy)	0.996740	0.997365	0.997980
min(accuracy)	0.995800	0.996841	0.997700
V1 – fuzzy average	0.996800	0.997824	0.998600
V2 – weighted by confidence	0.996900	0.997809	0.998600
V3 – weighted by diff from V1	0.996900	0.997833	0.998600
V4 – weighted by 1/failures	0.996800	0.997835	0.998600
V5 – plurality voting	0.996600	0.997721	0.998500
V6 – borda voting	0.996600	0.997733	0.998500
V7 – geometric mean voting	0.996800	0.997826	0.998600
V8 – meta – plurality (V1–V7)	0.996800	0.997829	0.998600
V9 – meta – plurality (V1–V4, V7)	0.996800	0.997828	0.998600
V10 – meta – fuzzy avg (V1–V4, V7)	0.996800	0.997829	0.998600

Table 4

3.2.5. Combined statistics from experiments with different learners

We also show combined statistics from the collected results performed by different learners to see a more comprehensive comparison between the voting functions. We collected all the results of our experiments which had all the variables presented in the above tables: individual test results and the results of the V1–V10 voting functions.

Voting function	MIN	AVG	MAX
max(accuracy)	0.994600	0.997751	0.998500
avg(accuracy)	0.994400	0.997193	0.998250
min(accuracy)	0.992600	0.996563	0.998200
V1 – fuzzy average	0.995200	0.997819	0.998800
V2 – weighted by confidence	0.995300	0.997825	0.998700
V3 – weighted by diff from V1	0.995200	0.997820	0.998800
V4 – weighted by 1/failures	0.994800	0.997813	0.998800
V5 – plurality voting	0.993900	0.997749	0.998700
V6 – borda voting	0.993900	0.997758	0.998700
V7 – geometric mean voting	0.995200	0.997824	0.998700
V8 – meta – plurality (V1-V7)	0.995300	0.997819	0.998800
V9 – meta – plurality (V1-V4,V7)	0.995200	0.997821	0.998800
V10 – meta – fuzzy avg (V1-V4,V7)	0.995200	0.997821	0.998800

Table 5

Table 5 shows the combined statistics of about 1 million votings where in each turn 2–40 voters voted using the voting functions V1–V10. This statistics can differ from what we can see from the above tables, since the results of yet more learning sessions are included and the number of conducted tests and the number of learners participated in the tests were not the same in the experiments. The best individual result was 15 fails from 10000 test samples, the worst individual result was 74 failed samples and on average they performed only 28.07 fails as individual learners. The well-known fuzzy voting performed 21.81 fails on average. There were small differences among the voting functions, the best result came from our fuzzy average voting variant (V2) with 21.75 fails from 10000 test samples. V3 and V7 voting functions and V9 and V10 meta voting functions have also outperformed the V1 fuzzy average voting function. The voting performance of V7 had the lowest standard deviation among the voting functions.

4. Conclusion

From the experiments, which were performed to compare the new voting functions with some of the well-known ones, we can conclude that the accuracy of the examined voting functions have a stochastic behavior. We discovered that there is no voting function that is always the winner. The availability of multiple voting functions can, however, lead to better performance, if the best performer function will be chosen for a specific problem set. Some of the proposed voting functions had better accuracy, in all our experiments, compared to the most frequently used well-known fuzzy average and plurality voting functions (V2, V7, V9, V10). This is results are very promising, although further research and analysis must be done to discover their behavior.

References

[1] G. Auda, M. Kamel, H. Raafat: *Voting schemes for cooperative neural network classifiers*, in: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 3, IEEE, 1995, pp. 1240–1243.

[2] R. Basbous, B. Nagy, T. Tajti: *Short Circuit Evaluations in Gödel Type Logic*, Proc. of FANCCO 2015: 5th International Conference on Fuzzy and Neuro Computing, Advances in Intelligent Systems and Computing 415 (2015), pp. 119–138, doi: https://doi.org/10.1007/978-3-319-27212-2_10.

[3] R. Basbous, T. Tajti, B. Nagy: *Fast Evaluations in Product Logic: Various Pruning Techniques*, in: FUZZ-IEEE 2016 - the 2016 IEEE International Conference on Fuzzy Systems, Vancouver, Canada: IEEE, 2016, pp. 140–147, doi: https://doi.org/10.1109/FUZZ-IEEE.2016.7737680.

[4] D. Ciresan, U. Meier, J. Schmidhuber: *Multi-column deep neural networks for image classification*, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 3642–3649.

[5] Y. Freund: *Boosting a weak learning algorithm by majority*, Information and computation 121.2 (1995), pp. 256–285.

[6] R. Fullér: *Fuzzy systems*, in: Introduction to Neuro-Fuzzy Systems, Springer, 2000, pp. 1–131.

[7] S. Haykin: *Neural Networks: A Comprehensive Foundation*, 2nd, USA: Prentice Hall PTR, 1998, isbn: 0132733501.

[8] G. Kovásznai, C. Biró, B. Erdélyi: *Puli–A Problem-Specific OMT solver*, in: Proc. 16th International Workshop on Satisfiability Modulo Theories (SMT 2018), 371, 2018.

[9] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, L. E. Barnes: *Rmdl: Random multimodel deep learning for classification*, in: Proceedings of the 2nd International Conference on Information System and Data Mining, 2018, pp. 19–28.

[10] L. I. Kuncheva: *A theoretical study on six classifier fusion strategies*, IEEE Transactions on pattern analysis and machine intelligence 24.2 (2002), pp. 281–286.

[11] L. Lam, C. Y. Suen: *Optimal combinations of pattern classifiers*, Pattern Recognition Letters 16.9 (1995), pp. 945–954.

[12] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner: *Gradient-based learning applied to document recognition*, Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

[13] Y. LeCun, C. Cortes, C. J. Burges: *The MNIST database of handwritten digits, 1998*, 10.34 (1998), p. 14, url: http://yann.lecun.com/exdb/mnist.

[14] L. Li, Q. Hu, X. Wu, D. Yu: *Exploration of classification confidence in ensemble learning*, Pattern recognition 47.9 (2014), pp. 3120–3131.

[15] Matuzas77: *MNIST classifier with average 0.17% error*, github.com (2020), url: https://github.com/Matuzas77/MNIST-0.17/blob/master/MNIST_final_solution.ipynb.

[16] B. Nagy, R. Basbous, T. Tajti: *Lazy evaluations in Łukasiewicz type fuzzy logic*, Fuzzy Sets and Systems 376 (2019), Theme: Computer Science, pp. 127–151, issn: 0165-0114, doi: https://doi.org/10.1016/j.fss.2018.11.014, url: http://www.sciencedirect.com/science/article/pii/S0165011418309357.

[17] D. Opitz, R. Maclin: *Popular ensemble methods: An empirical study*, Journal of artificial intelligence research 11 (1999), pp. 169–198.

[18] W. Richards, H. S. Seung, G. Pickard: *Neural voting machines*, Neural Networks 19.8 (2006), pp. 1161–1167.

[19] S. Russell, P. Norvig: *Artificial intelligence: a modern approach* (2002).

[20] C. Sammut, G. I. Webb: *Encyclopedia of machine learning*, Springer Science & Business Media, 2011.

[21] T. Tajti: *Fuzzification of training data class membership binary values for neural network algorithms*, Annales Mathematicae et Informaticae 52 (2020), to be approved,
doi: https://doi.org/10.33039/ami.2020.10.001.

[22] V. Tresp: *Committee machines*, Handbook for neural network signal processing (2001), pp. 1–18.

[23] S. Wan, H. Yang: *Comparison among Methods of Ensemble Learning*, 2013 International Symposium on Biometrics and Security Technologies (2013), pp. 286–290.

[24] L. A. Zadeh, G. J. Klir, B. Yuan: *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*, WORLD SCIENTIFIC, 1996,
doi: https://doi.org/10.1142/2895,
url: https://www.worldscientific.com/doi/abs/10.1142/2895.

Annales Mathematicae et Informaticae
***52** (2020) pp. 243–253*
DOI: https://doi.org/10.33039/ami.2020.01.003
URL: https://ami.uni-eszterhazy.hu

On the exponential Diophantine equation $(4m^2+1)^x+(21m^2-1)^y=(5m)^z$

Nobuhiro Terai*

Division of Mathematical Sciences, Department of Integrated Science and Technology
Faculty of Science and Technology, Oita University, Oita, Japan
terai-nobuhiro@oita-u.ac.jp

Submitted: July 18, 2019
Accepted: January 19, 2020
Published online: February 3, 2020

Abstract

Let m be a positive integer. Then we show that the exponential Diophantine equation $(4m^2+1)^x+(21m^2-1)^y=(5m)^z$ has only the positive integer solution $(x,y,z)=(1,1,2)$ under some conditions. The proof is based on elementary methods and Baker's method.

Keywords: Exponential Diophantine equation, integer solution, lower bound for linear forms in two logarithms.

MSC: 11D61

1. Introduction

Let a,b,c be fixed relatively prime positive integers greater than one. The exponential Diophantine equation

$$a^x+b^y=c^z \tag{1.1}$$

in positive integers x,y,z has been actively studied by a number of authors. It is known that the number of solutions (x,y,z) of equation (1.1) is finite, and all solutions can be effectively determined by means of Baker's method of linear forms in logarithms.

*The author is supported by JSPS KAKENHI Grant (No.18K03247).

Equation (1.1) has been investigated in detail for Pythagorean numbers a, b, c, too. Jeśmanowicz [8] conjectured that if a, b, c are Pythagorean numbers, i.e., positive integers satisfying $a^2+b^2 = c^2$, then (1.1) has only the positive integer solution $(x, y, z) = (2, 2, 2)$ (cf. [14, 17, 22]). As an analogue of Jeśmanowicz' conjecture, the author proposed that if a, b, c, p, q, r are fixed positive integers satisfying $a^p+b^q = c^r$ with $a, b, c, p, q, r \geq 2$ and $\gcd(a, b) = 1$, then (1.1) has only the positive integer solution $(x, y, z) = (p, q, r)$ except for a handful of triples (a, b, c) (cf. [6, 12, 13, 15, 21, 24]). This conjecture has been proved to be true in many special cases. This conjecture, however, is still unsolved.

In Terai [23], the author showed that if m is a positive integer such that $1 \leq m \leq 20$ or $m \not\equiv 3 \pmod 6$, then the Diophantine equation

$$(4m^2 + 1)^x + (5m^2 - 1)^y = (3m)^z \tag{1.2}$$

has only the positive integer solution $(x, y, z) = (1, 1, 2)$. The proof is based on elementary methods and Baker's method. Suy-Li [20] proved that if $m \geq 90$ and $3 \mid m$, then equation (1.2) has only the positive integer solution $(x, y, z) = (1, 1, 2)$ by means of the result of Bilu-Hanrot-Voutier [3] concerning the existence of primitive prime divisors in Lucas-numbers. Finally, Bertók [1] has completely solved equation (1.2) including the remaining cases $20 < m < 90$. His proof can be done by the help of exponential congruences. This is a nice application of Bertók and Hajdu [2].

More generally, several authors have studied the Diophantine equation

$$(pm^2 + 1)^x + (qm^2 - 1)^y = (rm)^z \tag{1.3}$$

under some conditions, where p, q, r are positive integers satisfying $p + q = r^2$:

- (Miyazaki-Terai [16], 2014) $(m^2 + 1)^x + (qm^2 - 1)^y = (rm)^z$, $1 + q = r^2$,
- (Terai-Hibino [25], 2015) $(12m^2 + 1)^x + (13m^2 - 1)^y = (5m)^z$,
- (Terai-Hibino [26], 2017) $(3pm^2 - 1)^x + (p(p - 3)m^2 + 1)^y = (pm)^z$,
- (Fu-Yang [7], 2017) $(pm^2 + 1)^x + (qm^2 - 1)^y = (rm)^z$, $r \mid m$,
- (Pan [19], 2017) $(pm^2 + 1)^x + (qm^2 - 1)^y = (rm)^z$, $m \equiv \pm 1 \pmod r$,
- (Murat [18], 2018) $(18m^2 + 1)^x + (7m^2 - 1)^y = (5m)^z$,
- (Kizildere et al. [10], 2018) $((q+1)m^2+1)^x+(qm^2-1)^y = (rm)^z$, $2q+1 = r^2$.

We note that equation (1.2), which was completely resolved by Terai, Suy-Li and Bertók, is the first equation shown that equation (1.3) has only the trivial solution $(x, y, z) = (1, 1, 2)$ without any assumption on m. All known results for the above-mentioned equations need congruence relations or inequalities on m.

In this paper, we consider the exponential Diophantine equation

$$(4m^2 + 1)^x + (21m^2 - 1)^y = (5m)^z \tag{1.4}$$

with m positive integer. Denote $v_p(n)$ by the exponent of p in the factorization of a positive integer n. Our main result is the following:

Theorem 1.1. *Let m be a positive integer. Suppose that $v_5(4m^2+1)=v_5(21m^2-1)=1$ only if $m\equiv\pm1 \pmod{10}$. Then equation (1.4) has only the positive integer solution $(x,y,z)=(1,1,2)$.*

This paper is organized as follows. When m is even or m is odd in (1.4) with $y\geq 2$, we show Theorem 1.1 by using elementary methods such as congruence methods and the quadratic reciprocity law. When m is odd in (1.4) with $m\equiv\pm2 \pmod 5$ and $y=1$, we show Theorem 1.1 by applying a lower bound for linear forms in two logarithms due to Laurent [11]. The proof of the case $m\equiv\pm1 \pmod 5$ uses the Primitive Divisor Theorem due to Zsigmondy [27]. That of the case $m\equiv 0 \pmod 5$ is based on a result on linear forms in p-adic logarithms due to Bugeaud [5].

2. Preliminaries

In order to obtain an upper bound for a solution of Pillai's equation, we need a result on lower bounds for linear forms in the logarithms of two algebraic numbers. We will introduce here some notations. Let α_1 and α_2 be real algebraic numbers with $|\alpha_1|\geq 1$ and $|\alpha_2|\geq 1$. We consider the linear form

$$\Lambda = b_2\log\alpha_2 - b_1\log\alpha_1,$$

where b_1 and b_2 are positive integers. As usual, the *logarithmic height* of an algebraic number α of degree n is defined as

$$h(\alpha)=\frac{1}{n}\left(\log|a_0|+\sum_{j=1}^{n}\log\max\left\{1,\left|\alpha^{(j)}\right|\right\}\right),$$

where a_0 is the leading coefficient of the minimal polynomial of α (over $\mathbb{Z}$) and $(\alpha^{(j)})_{1\leq j\leq n}$ are the conjugates of α. Let A_1 and A_2 be real numbers greater than 1 with

$$\log A_i\geq\max\left\{h(\alpha_i),\frac{|\log\alpha_i|}{D},\frac{1}{D}\right\},$$

for $i\in\{1,2\}$, where D is the degree of the number field $\mathbb{Q}(\alpha_1,\alpha_2)$ over $\mathbb{Q}$. Define

$$b'=\frac{b_1}{D\log A_2}+\frac{b_2}{D\log A_1}.$$

We choose to use a result due to Laurent [11, Corollary 2], with $m=10$ and $C_2=25.2$.

Proposition 2.1 (Laurent [11]). *Let Λ be given as above, with $\alpha_1 > 1$ and $\alpha_2 > 1$. Suppose that α_1 and α_2 are multiplicatively independent. Then*

$$\log|\Lambda| \geq -25.2D^4 \left(\max\left\{\log b' + 0.38, \frac{10}{D}\right\}\right)^2 \log A_1 \log A_2.$$

Next, we shall quote a result on linear forms in p-adic logarithms due to Bugeaud [5]. Here we consider the case where $y_1 = y_2 = 1$ in the notation from [5, p. 375].

Let p be an odd prime. Let a_1 and a_2 be non-zero integers prime to p. Let g be the least positive integer such that

$$\mathrm{ord}_p(a_1^g - 1) \geq 1, \quad \mathrm{ord}_p(a_2^g - 1) \geq 1,$$

where we denote the p-adic valuation by $\mathrm{ord}_p(\,\cdot\,)$. Assume that there exists a real number E such that

$$1/(p-1) < E \leq \mathrm{ord}_p(a_1^g - 1).$$

We consider the integer

$$\Lambda = a_1^{b_1} - a_2^{b_2},$$

where b_1 and b_2 are positive integers. We let A_1 and A_2 be real numbers greater than 1 with

$$\log A_i \geq \max\{\log|a_i|,\, E\log p\} \quad (i = 1, 2),$$

and we put $b' = b_1/\log A_2 + b_2/\log A_1$.

Proposition 2.2 (Bugeaud [5]). *With the above notation, if a_1 and a_2 are multiplicatively independent, then we have the upper estimate*

$$\mathrm{ord}_p(\Lambda) \leq \frac{36.1g}{E^3(\log p)^4}\big(\max\{\log b' + \log(E\log p) + 0.4,\, 6E\log p,\, 5\}\big)^2 \log A_1 \log A_2.$$

The following is a direct consequence of an old version of the Primitive Divisor Theorem due to Zsigmondy [27]:

Proposition 2.3 (Zsigmondy [27]). *Let A and B be relatively prime integers with $A > B \geqslant 1$. Let $\{a_k\}_{k\geqslant 1}$ be the sequence defined as*

$$a_k = A^k + B^k.$$

If $k > 1$, then a_k has a prime factor not dividing $a_1 a_2 \cdots a_{k-1}$, whenever $(A, B, k) \neq (2, 1, 3)$.

3. Proof of Theorem 1.1

In this section, we give a proof of Theorem 1.1.

3.1. The case where m is odd and $m \equiv \pm 1 \pmod 5$

Lemma 3.1. *Let m be a positive integer such that m is odd and $m \equiv \pm 1 \pmod 5$. Suppose that $v_5(4m^2+1) = v_5(21m^2-1) = 1$. Then equation (1.4) has only the positive integer solution $(x,y,z) = (1,1,2)$.*

Proof. If $v_5(4m^2+1) = v_5(21m^2-1) = 1$, then $\gcd(4m^2+1, 21m^2-1) = 5$. Put $A = (4m^2+1)/5$ and $B = (21m^2-1)/5$. Then $\gcd(A,B) = 1$ and $AB \not\equiv 0 \pmod 5$. In view of $(5m)^x < (4m^2+1)^x < (5m)^z$ from (1.4), it follows that the inequality $z > x$ holds. Equation (1.4) can be written as

$$5^y B^y = 5^x(5^{z-x}m^z - A^x)$$

with $AB \not\equiv 0 \pmod 5$. This implies that $x = y$. Then equation (1.4) becomes

$$a_x = A^x + B^x = 5^{z-x}m^z.$$

Apply Proposition 2.3 with $A = (4m^2+1)/5$ and $B = (21m^2-1)/5$. Note that $\gcd(A,B) = 1$. Since $a_1 = 5m^2$, it follows that $x = 1$, which yields $(y,z) = (1,2)$. □

Lemma 3.2. *In (1.4), y is odd.*

Proof. When $m = 1$, we see that $v_5(4m^2+1) = v_5(21m^2-1) = 1$. By Lemma 3.1, we may suppose that $m \geq 2$. It follows that $z \geq 2$ from (1.4). Taking (1.4) modulo m^2 implies that $1 + (-1)^y \equiv 0 \pmod{m^2}$ and hence y is odd. □

3.2. The case where m is even

Lemma 3.3. *If m is even, then equation (1.4) has only the positive integer solution $(x,y,z) = (1,1,2)$.*

Proof. If $z \leq 2$, then $(x,y,z) = (1,1,2)$ from (1.4). Hence we may suppose that $z \geq 3$. Taking (1.4) modulo m^3 implies that

$$1 + 4m^2x - 1 + 21m^2y \equiv 0 \pmod{m^3},$$

so

$$4x + 21y \equiv 0 \pmod m,$$

which is impossible, since y is odd and m is even. We therefore conclude that if m is even, then equation (1.4) has only the positive integer solution $(x,y,z) = (1,1,2)$. □

3.3. The case where m is odd and $m \equiv \pm 2 \pmod 5$

By Lemma 3.3, we may suppose that m is odd with $m \geq 3$. Let (x,y,z) be a solution of (1.4).

Lemma 3.4. *If m is odd and $m \equiv \pm 2 \pmod 5$, then $y = 1$ and x is odd.*

Proof. Suppose that $m \equiv \pm 2 \pmod 5$, i.e., $m^2 \equiv -1 \pmod 5$. Then $\left(\frac{21m^2-1}{4m^2+1}\right) = 1$ and $\left(\frac{5m}{4m^2+1}\right) = -1$, where $\left(\frac{*}{*}\right)$ denotes the Jacobi symbol. Indeed,

$$\left(\frac{21m^2-1}{4m^2+1}\right) = \left(\frac{m^2-6}{4m^2+1}\right) = \left(\frac{4m^2+1}{m^2-6}\right) = \left(\frac{25}{m^2-6}\right) = 1$$

and

$$\left(\frac{5m}{4m^2+1}\right) = \left(\frac{5}{4m^2+1}\right)\left(\frac{m}{4m^2+1}\right) = \left(\frac{4m^2+1}{5}\right)\left(\frac{4m^2+1}{m}\right) =$$

$\left(\frac{-3}{5}\right)\left(\frac{1}{m}\right) = (-1)\cdot 1 = -1$, since $m^2 \equiv -1 \pmod 5$. In view of these, z is even from (1.4).

Suppose that $y \geq 2$. Taking (1.4) modulo 8 implies that

$$5^x \equiv (5m)^z \equiv 1 \pmod 8,$$

so x is even.

On the other hand, since $m^2 \equiv -1 \pmod 5$, taking (1.4) modulo 5 implies that

$$2^x + 3^y \equiv 0 \pmod 5,$$

which contradicts the fact that x is even and y is odd. Hence we obtain $y = 1$. Then, taking (1.4) modulo 8 implies that $5^x + 4 \equiv (5m)^z \equiv 1 \pmod 8$, so x is odd. □

From Lemma 3.4, it follows that $y = 1$ and x is odd. If $x = 1$, then we obtain $z = 2$ from (1.4). From now on, we may suppose that $x \geq 3$. Hence our theorem is reduced to solving Pillai's equation

$$c^z - a^x = b \tag{3.1}$$

with $x \geq 3$, where $a = 4m^2 + 1$, $b = 21m^2 - 1$ and $c = 5m$.

We now want to obtain a lower bound for x.

Lemma 3.5. $x \geq \frac{1}{4}(m^2 - 21)$.

Proof. Since $x \geq 3$, equation (3.1) yields the following inequality:

$$(5m)^z = (4m^2+1)^x + 21m^2 - 1 \geq (4m^2+1)^3 + 21m^2 - 1 > (5m)^3.$$

Hence $z \geq 4$. Taking (3.1) modulo m^4 implies that

$$1 + 4m^2x + 21m^2 - 1 \equiv 0 \pmod{m^4},$$

so $4x + 21 \equiv 0 \pmod{m^2}$. Hence we obtain our assertion. □

We next want to obtain an upper bound for x.

Lemma 3.6. $x < 2521 \log c$.

Proof. From (3.1), we now consider the following linear form in two logarithms:

$$\Lambda = z\log c - x\log a \quad (>0).$$

Using the inequality $\log(1+t) < t$ for $t > 0$, we have

$$0 < \Lambda = \log(\frac{c^z}{a^x}) = \log(1+\frac{b}{a^x}) < \frac{b}{a^x}. \tag{3.2}$$

Hence we obtain

$$\log\Lambda < \log b - x\log a. \tag{3.3}$$

On the other hand, we use Proposition 2.1 to obtain a lower bound for Λ. It follows from Proposition 2.1 that

$$\log\Lambda \geq -25.2\left(\max\{\log b' + 0.38, 10\}\right)^2 (\log a)(\log c), \tag{3.4}$$

where $b' = \frac{x}{\log c} + \frac{z}{\log a}$.

We note that $a^{x+1} > c^z$. Indeed,

$$a^{x+1} - c^z = a(c^z - b) - c^z = (a-1)c^z - ab \geq 4m^2 \cdot 25m^2 - (4m^2+1)(21m^2-1) > 0.$$

Hence $b' < \frac{2x+1}{\log c}$.

Put $M = \frac{x}{\log c}$. Combining (3.3) and (3.4) leads to

$$x\log a < \log b + 25.2\left(\max\left\{\log\left(2M + \frac{1}{\log c}\right) + 0.38, 10\right\}\right)^2 (\log a)(\log c),$$

so

$$M < 1 + 25.2\left(\max\left\{\log\left(2M + \frac{1}{2}\right) + 0.38, 10\right\}\right)^2,$$

since $\log c = \log(5m) \geq \log 15 > 2$. We therefore obtain $M < 2521$. This completes the proof of Lemma 3.6. □

We are now in a position to prove Theorem 1.1. It follows from Lemmas 3.5, 3.6 that

$$\frac{1}{4}(m^2 - 21) < 2521 \log 5m.$$

Hence we obtain $m \leq 269$. From (3.2), we have the inequality

$$\left|\frac{\log a}{\log c} - \frac{z}{x}\right| < \frac{b}{xa^x \log c},$$

which implies that $\left|\frac{\log a}{\log c} - \frac{z}{x}\right| < \frac{1}{2x^2}$, since $x \geq 3$. Thus $\frac{z}{x}$ is a convergent in the simple continued fraction expansion to $\frac{\log a}{\log c}$.

On the other hand, if $\frac{p_r}{q_r}$ is the r-th such convergent, then

$$\left|\frac{\log a}{\log c} - \frac{p_r}{q_r}\right| > \frac{1}{(a_{r+1}+2)q_r^2},$$

where a_{r+1} is the $(r+1)$-st partial quotient to $\frac{\log a}{\log c}$ (see e.g. Khinchin [9]). Put $\frac{z}{x} = \frac{p_r}{q_r}$. Note that $q_r \le x$. It follows, then, that

$$a_{r+1} > \frac{a^x \log c}{bx} - 2 \ge \frac{a^{q_r} \log c}{bq_r} - 2. \tag{3.5}$$

Finally, we checked by Magma [4] that inequality (3.5) does not hold for any r with $q_r < 2521 \log(5m)$ in the range $3 \le m \le 269$.

3.4. The case $m \equiv 0 \pmod 5$

Let m be a positive integer with $m \equiv 0 \pmod 5$. Let (x, y, z) be a solution of (1.4). Taking (1.4) modulo $m(\ge 5)$ implies that y is odd. Here, we apply Proposition 2.2. For this we set $p := 5, a_1 := 4m^2+1, a_2 := 1-21m^2, b_1 := x, b_2 := y$, and

$$\Lambda := (4m^2+1)^x - (1-21m^2)^y.$$

Then we may take $g = 1, E = 2, A_1 = 4m^2+1, A_2 := 21m^2-1$. Hence we have

$$2z \le \frac{36.1}{8(\log 5)^4}\left(\max\{\log b' + \log(2\log 5) + 0.4,\ 12\log 5\}\right)^2 \log(4m^2+1)\log(21m^2-1),$$

where $b' := \frac{x}{\log(21m^2-1)} + \frac{y}{\log(4m^2+1)}$. Suppose that $z \ge 4$. We will observe that this leads to a contradiction. Taking (1.4) modulo m^4 implies that

$$4x + 21y \equiv 0 \pmod{m^2}.$$

In particular, we see that $M := \max\{x, y\} \ge m^2/25$. Therefore, since $z \ge M$ and $b' \le \frac{M}{\log m}$, we obtain

$$2M \le \frac{36.1}{8(\log 5)^4}\left(\max\left\{\log\left(\frac{M}{\log m}\right) + \log(2\log 5) + 0.4,\ 12\log 5\right\}\right)^2 \times \log(4m^2+1)\log(21m^2-1). \tag{3.6}$$

If $m \ge 122009$, then

$$2M \le \frac{36.1}{8(\log 5)^4}\left(\log\left(\frac{M}{\log m}\right) + \log(2\log 5) + 0.4\right)^2 \log(4m^2+1)\log(21m^2-1).$$

Since $m^2 \le 25M$, the above inequality gives

$$2M \le 0.7\left(\log M - \log(\log 122009) + 1.6\right)^2 \log(100M+1)\log(525M-1).$$

We therefore obtain $M \leq 3386$, which contradicts the fact that $M \geq m^2/25 \geq 595447844$.

If $m < 122009$, then inequality (3.6) gives

$$\frac{2}{25}m^2 \leq 251 \log(4m^2+1)\log(21m^2-1).$$

This implies that $m \leq 882$. Hence all x, y and z are also bounded. It is not hard to verify by Magma [4] that there is no (m, x, y, z) under consideration satisfying (1.4). We conclude that $z \leq 3$. In this case, we can easily show that $(x, y, z) = (1, 1, 2)$. This completes the proof of Theorem 1.1.

Remark 3.7. The values of m, a, b, c satisfying the condition of Theorem 1.1 with $1 \leq m < 100$ are given in the table below.

m	a	b	c
1	5	$2^2 \cdot 5$	5
11	$5 \cdot 97$	$2^2 \cdot 5 \cdot 127$	$5 \cdot 11$
19	$5 \cdot 17^2$	$2^2 \cdot 5 \cdot 379$	$5 \cdot 19$
21	$5 \cdot 353$	$2^2 \cdot 5 \cdot 463$	$3 \cdot 5 \cdot 7$
29	$5 \cdot 673$	$2^2 \cdot 5 \cdot 883$	$5 \cdot 29$
31	$5 \cdot 769$	$2^2 \cdot 5 \cdot 1009$	$5 \cdot 31$
39	$5 \cdot 1217$	$2^2 \cdot 5 \cdot 1597$	$3 \cdot 5 \cdot 13$
49	$5 \cdot 17 \cdot 113$	$2^2 \cdot 5 \cdot 2521$	$5 \cdot 7^2$
51	$5 \cdot 2081$	$2^2 \cdot 5 \cdot 2731$	$3 \cdot 5 \cdot 17$
61	$5 \cdot 13 \cdot 229$	$2^2 \cdot 5 \cdot 3907$	$5 \cdot 61$
69	$5 \cdot 13 \cdot 293$	$2^2 \cdot 5 \cdot 4999$	$3 \cdot 5 \cdot 23$
71	$5 \cdot 37 \cdot 109$	$2^2 \cdot 5 \cdot 67 \cdot 79$	$5 \cdot 71$
79	$5 \cdot 4993$	$2^2 \cdot 5 \cdot 6553$	$5 \cdot 79$
81	$5 \cdot 29 \cdot 181$	$2^2 \cdot 5 \cdot 83^2$	$3^4 \cdot 5$
89	$5 \cdot 6337$	$2^2 \cdot 5 \cdot 8317$	$5 \cdot 89$
99	$5 \cdot 7841$	$2^2 \cdot 5 \cdot 41 \cdot 251$	$3^2 \cdot 5 \cdot 11$

Let m be a positive integer with $m \equiv \pm 1 \pmod{10}$. Suppose that $v_5(4m^2+1) = v_5(21m^2-1)$. Since $(4m^2+1)+(21m^2-1) = 25m^2$, we see that $\gcd(4m^2+1, 21m^2-1) = 5$ or 25 according as $v_5(4m^2+1) = v_5(21m^2-1) = 1$ or 2. Put $A = (4m^2+1)/5^e$ and $B = (21m^2-1)/5^e$ with $e = 1, 2$ according as $v_5(4m^2+1) = v_5(21m^2-1) = 1$ or 2. Then $\gcd(A, B) = 1$ and $AB \not\equiv 0 \pmod{5}$. Though we apply Proposition 2.3 to the case $v_5(4m^2+1) = v_5(21m^2-1) = 2$, e.g., $m = 9, 41, 59, 191, 209$, etc., we can not obtain $x = 1$ unlike Theorem 1.1. Indeed, $a_x = A^x + B^x = 5^{z-2x}m^z$ and $a_1 = m^2$.

References

[1] C. Bertók: *The complete solution of the Diophantine equation* $(4m^2+1)^x+(5m^2-1)^y = (3m)^z$, Period. Math. Hung. 72 (2016), pp. 37–42,
DOI: https://doi.org/10.1007/s10998-016-0111-x.

[2] C. Bertók, L. Hajdu: *A Hasse-type principle for exponential Diophantine equations and its applications*, Math. Comp. 85 (2016), pp. 849–860,
doi: https://doi.org/10.1090/mcom/3002.

[3] Y. Bilu, G. Hanrot, P. M. Voutier: *Existence of primitive divisors of Lucas and Lehmer numbers*, Journal für die Reine und Angewandte Mathematik 539 (2001), pp. 75–122,
doi: https://doi.org/10.1515/crll.2001.080.

[4] W. Bosma, J. Cannon: *Handbook of magma functions*, Department of Math., University of Sydney,
url: http://magma.maths.usyd.edu.au/magma/.

[5] Y. Bugeaud: *Linear forms in p-adic logarithms and the Diophantine equation* $(x^n-1)/(x-1) = y^q$, Math. Proc. Cambridge Phil. Soc. 127 (1999), pp. 373–381,
doi: https://doi.org/10.1017/S0305004199003692.

[6] Z. Cao: *A note on the Diophantine equation* $a^x + b^y = c^z$, Acta Arith. 91 (1999), pp. 85–93,
doi: https://doi.org/10.4064/aa-91-1-85-93.

[7] R. Fu, H. Yang: *On the exponential Diophantine equation* $(am^2+1)^x+(bm^2-1)^y = (cm)^z$ *with* $c \mid m$, Period. Math. Hung. 75 (2017), pp. 143–149,
doi: https://doi.org/10.1007/s10998-016-0170-z.

[8] L. Jeśmanowicz: *Some remarks on Pythagorean numbers*, Wiadom. Mat. 1 (1955/1956), pp. 196–202.

[9] A. Y. Khinchin: *Continued Fractions*, 3rd ed., Groningen: P. Noordhoff Ltd., 1963.

[10] E. Kizildere, T. Miyazaki, G. Soydan: *On the Diophantine equation* $((c+1)m^2+1)^x + (cm^2-1)^y = (am)^z$, Turk. J. Math. 42 (2018), pp. 2690–2698,
doi: https://doi.org/10.3906/mat-1803-14.

[11] M. Laurent: *Linear forms in two logarithms and interpolation determinants II*, Acta Arith. 133 (2008), pp. 325–348,
doi: https://doi.org/10.4064/aa133-4-3.

[12] M. Le: *A conjecture concerning the exponential diophantine equation* $a^x + b^y = c^z$, Acta Arith. 106 (2003), pp. 345–353,
doi: https://doi.org/10.4064/aa106-4-2.

[13] T. Miyazaki: *Exceptional cases of Terai's conjecture on Diophantine equations*, Arch. Math. (Basel) 95 (2010), pp. 519–527,
doi: https://doi.org/10.1007/s00013-010-0201-6.

[14] T. Miyazaki: *Generalizations of classical results on Jeśmanowicz' conjecture concerning primitive Pythagorean triples*, J. Number Theory 133 (2013), pp. 583–595,
doi: https://doi.org/10.1016/j.jnt.2012.08.018.

[15] T. Miyazaki: *Terai's conjecture on exponential Diophantine equations*, Int. J. Number Theory 7 (2011), pp. 981–999,
doi: https://doi.org/10.1142/S1793042111004496.

[16] T. Miyazaki, N. Terai: *On the exponential Diophantine equation* $(m^2+1)^x+(cm^2-1)^y = (am)^z$, Bull. Australian Math. Soc. 90 (2014), pp. 9–19,
doi: https://doi.org/10.1017/S0004972713000956.

[17] T. Miyazaki, P. Yuan, D. Wu: *Generalizations of classical results on Jeśmanowicz' conjecture concerning Pythagorean triples II*, J. Number Theory 141 (2014), pp. 184–201,
doi: https://doi.org/10.1016/j.jnt.2014.01.011.

[18] A. Murat: *On the exponential Diophantine equation* $(18m^2+1)^x + (7m^2-1)^y = (5m)^z$, Turk. J. Math. 42 (2018), pp. 1990–1999,
doi: https://doi.org/10.3906/mat-1801-76.

[19] X. Pan: *A note on the exponential Diophantine equation* $(am^2+1)^x+(bm^2-1)^y = (cm)^z$, Colloq. Math. 149 (2017), pp. 265–273,
doi: https://doi.org/10.4064/cm6878-10-2016.

[20] J. Su, X. Li: *The exponential Diophantine equation* $(4m^2+1)^x+(5m^2-1)^y=(3m)^z$, Abstract and Applied Analysis 2014 (2014), pp. 1–5,
doi: https://doi.org/10.1155/2014/670175.

[21] N. Terai: *Applications of a lower bound for linear forms in two logarithms to exponential Diophantine equations*, Acta Arith. 90 (1999), pp. 17–35,
doi: https://doi.org/10.4064/aa-90-1-17-35.

[22] N. Terai: *On Jeśmanowicz' conjecture concerning primitive Pythagorean triples*, J. Number Theory 141 (2014), pp. 316–323,
doi: https://doi.org/10.1016/j.jnt.2014.02.009.

[23] N. Terai: *On the exponential Diophantine equation* $(4m^2+1)^x+(5m^2-1)^y=(3m)^z$, Int. J. Algebra 6 (2012), pp. 1135–1146.

[24] N. Terai: *The Diophantine equation* $a^x+b^y=c^z$, Proc. Japan Acad. Ser. A Math. Sci. 70 (1994), pp. 22–26,
doi: https://doi.org/10.3792/pjaa.70.22.

[25] N. Terai, T. Hibino: *On the exponential Diophantine equation* $(12m^2+1)^x+(13m^2-1)^y=(5m)^z$, Int. J. Algebra 9 (2015), pp. 261–272,
doi: https://doi.org/10.12988/ija.2015.5529.

[26] N. Terai, T. Hibino: *On the exponential Diophantine equation* $(3pm^2-1)^x+(p(p-3)m^2+1)^y=(pm)^z$, Period. Math. Hung. 74 (2017), pp. 227–234,
doi: https://doi.org/10.1007/s10998-016-0162-z.

[27] K. Zsigmondy: *Zur Theorie der Potenzreste*, Monatsh. Math. 3 (1892), pp. 265–284,
doi: https://doi.org/10.1007/BF01692444.

Annales Mathematicae et Informaticae
***52** (2020) pp. 255–265*
DOI: https://doi.org/10.33039/ami.2020.01.002
URL: https://ami.uni-eszterhazy.hu

Simulation of the performance of Cognitive Radio Networks with unreliable servers*

Mohamed Hedi Zaghouani[a], János Sztrik[a], Arban Uka[b]

[a]School of Informatics, Faculty of Informatics
University of Debrecen, Debrecen, Hungary
zaghouani.hedi,sztrik.janos@inf.unideb.hu

[b]EPOKA University, Tirana, Albania
auka@epoka.edu.al

Submitted: September 17, 2019
Accepted: January 19, 2020
Published online: January 31, 2020

Abstract

This paper deals with a Cognitive Radio Network (CRN) which is modeled using a retrial queuing system with two finite-sources. This network includes two non-independent service units treating two types of users: Primary Users (PU) and Secondary Users (SU). The primary unit has priority queue (FIFO) and a second service unit contains an orbit both units are dedicated for the Primary Users and Secondary Users, respectively.

The current work highlights the unreliability of the servers as we are assuming that both servers of this network are subject to random breakdowns and repairs. All the inter-event times in this CRN are either exponentially or non-exponentially distributed. The novelty of our investigation is to analyze the effect of several distributions (Gamma, Pareto, Log-normal, Hypo-Exponential and Hyper-Exponential) of the failure and repair times on the main performance measure of the system. By the help of simulation we show some interesting results concerning to sensitivity problems.

Keywords: Finite source queuing systems, Simulation, Cognitive Radio Networks, Performance and Reliability measures, Non-reliable servers.

*The work of Dr. János Sztrik is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-financed by the European Union and the European Social Fund. The work of Mohamed Hedi Zaghouani is financed by the Stipendium Hungaricum Scholarship.

1. Introduction

Recent years have seen a significant increase in the demand for radio spectrum. As this kind of network provides a maximum use rate for customers, by allowing unlicensed (Secondary) users to process their services, while there is no licensed (Primary) user in the spectrum. Cognitive Radio (CR) is an intelligent technology that can sense automatically the available channels in a wireless spectrum and modify the transmission parameters allowing more communications to be established, additionally improves the network's behavior. The CRN's ultimate goal is to exploit the free sections of the primary frequency bands for the benefit of unlicensed customers, without any disadvantage for the licensed users, more explanations can be found in [2, 3, 14]. As the idea of the cognitive radio was introduced to the research community of wireless communications, the establishment of CRNs becomes more realistic day by day, since so many researchers consider this innovation a big benefit for the network field. There are two types of CRN, the first is known as (underlay network) in which unlicensed users are entitled at the same time to use the primary channels with the PUs, depending on some predefined conditions. The second type is called (overlay networks) where the secondary customers are allowed to use the Primary Service Unit if this unit does not contain licensed customers, more explanation was introduced by the authors of [10, 13, 16]. Theoretically, the current paper treats the second mentioned type of CRN (overlay), by modeling a CRN that uses two finite-source subsystems with non-reliable servers (Primary and Secondary) exposed to breakdowns and repairs.

We are taking into consideration two subsystems in this queuing system. The first sub-system is built for the primary users (PU) requests. The number of sources is finite, moreover, in exponentially distributed time each source generates a primary request for the PU, these tasks should be sent with a preemptive discipline to a single server which is called the Primary Channel Service (PCS), to start the service based on an exponentially distributed time as well. The second part of the model is dedicated to the secondary unit requests arriving from a finite-source as well, knowing that the service and the source times of the secondary customers are exponentially distributed. All the generated primary requests are headed to the primary server in order to check its accessibility. If the service unit is free the service starts instantly. However, if the primary unit is already busy with another primary request this last packet joins a FIFO queue. Nevertheless , if the primary unit is busy by treating a secondary user service, as consequence this packet disconnects right away and will be sent back to the Secondary Channel Service (SCS). Based on the availability of the secondary server this postponed task either starts the service again or joins the orbit. In the other hand, the secondary requests are sent to the secondary server to verify its availability. If the aimed server is available the service of request starts instantly, otherwise these unlicensed requests will try to join the Primary Service Unit (PSU). If it is free the service of the low priority task begins. If not, they must join the orbit automatically. Canceled requests in the orbit retry to be served after an exponentially distributed random interval,

more details can be found in [2, 3, 10, 13, 14, 16]. The servers used in our network are subject to some random breakdowns the interrupted requests are sent to the queue or to the orbit, respectively.

In our case, we assume that the servers failure and repair times are non-exponentially distributed (Hypo-Exponential, Hyper-Exponential, Gamma, Pareto and Log-normal). All the random times concerned in this model construction are supposed to be independent of each other.

In a similar work [6] authors considered that the network has a single server which is subject to breakdowns and repairs. This type of network suffers from difficulty with processing the requests as the breakdown of the only server effects the whole system, if the server is down then the whole network is down. Some other papers investigated further the retrial queuing model by modeling a cognitive radio network using two service channels (Primary and Secondary) both are subject to breakdowns and repair. For example, the authors of [9] assumed that both servers are unreliable and used different distributions for the inter-event times, Hypo and Hyper Exponential were used for the failure and repair times and Exponential distribution was assumed for the rest of the inter-event times (arrival, service and retrial). As extended work authors of [15] have added Gamma distribution to the above mentioned above distributions.

The main aim of this work is to study the effect of distributions for the failure and repair time on the main performance measures of the system. By the help of simulation we show some interesting results concerning to sensitivity problems.

2. System Model

As shown in Figure 1 our system model is a finite source queuing system with retrials which contains two sub-systems for the PUs and SUs knowing that these two subsystems are connected to each other. The model's first subsystem will be dedicated for PU requests, in which $N1$ is the finite number of sources. Each inter-arrival time is exponentially distributed with rate λ_1 thus a primary request will be created and sent to a preemptive priority queue (FIFO). If the target server is idle the service starts instantly and last for an exponentially distributed time with parameter μ_1. Otherwise, the new created request will have to wait in the queue. The calls of the SUs will be generated randomly as well as, thus every inter-request time is assumed to be exponentially distributed with parameter λ_2 and will be served according to an exponentially distributed random variable with parameter μ_2. The number of sources in this second subsystem is $N2$.

It should be noted that if a high priority request joins the primary server and finds it busy with an unlicensed (secondary) request the latter request will be interrupted and sent back either to the SSU (Secondary Service Unit) or to the orbit depending on the accessibility of the secondary channel. However, if the primary server is processing a licensed request the new customer will have to wait in the queue.

In case of secondary users they can process their services immediately if the

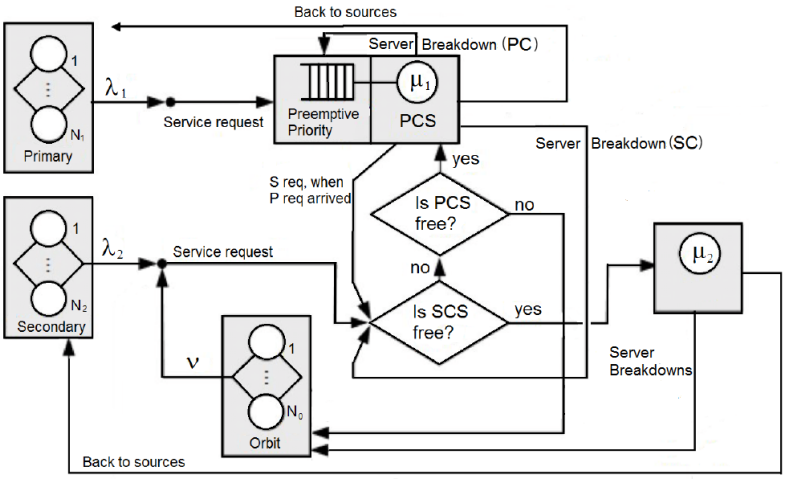


Figure 1: Cognitive Radio Network with finite-source retrial queuing system

dedicated channel is free. If it is busy they will check the availability of the primary channel hoping to start the service, furthermore, if the primary channel is busy too the involved task will be forwarded to the orbit. These postponed packets will try to get served after an exponentially distributed time with parameter ν.

As mentioned before the two subsystems of our network will be subject to breakdowns and repairs the failures of the service units can occur both in busy and idle status.

Failure and repair times will appear randomly for primary and secondary servers according to Hyper-Exponential, Hypo-Exponential, Gamma, Log-normal and Pareto distributions with given parameters. The corresponding intensities which are the inverse of the mean are denoted by γ_1, γ_2 and σ_1, σ_2, respectively.

Using the following stochastic model we can identify our system through the notations:

- $k1(t)$: represents the number of licensed sources at given time t;
- $k2(t)$: refers to the number of unlicensed at time given t;
- $q(t)$: is the number of primary requests in the queue at certain time t;
- $o(t)$: denotes the number of tasks in the orbit at time t;
- $y(t) = 0$, if the primary channel is idle, $y(t) = 1$, if the primary channel is processing (busy) a high-priority request and $y(t) = 2$, if the primary service unit is processing (busy) a low-priority request at time t;
- $c(t) = 0$, if the secondary service unit is idle(free) and $c(t) = 1$, if the secondary service unit is busy at given time t.

As consequence we can see that:

$$k1(t) = \begin{cases} N1 - q(t), & y(t) = 0, 2, \\ N1 - q(t) - 1, & y(t) = 1, \end{cases}$$

$$k2(t) = \begin{cases} N2 - o(t) - c(t), & y(t) = 0, 1, \\ N2 - o(t) - c(t) - 1, & y(t) = 2. \end{cases}$$

Beside these we have to know if the server is operational or failed. We assume that the random variables involved in the model construction are either exponentially or non-exponentially distributed. Due to the non-exponential distributions the determination of stationary distribution of the system is too difficult so we decided to use a stochastic simulation using C coding language with GSL stochastic library.

Several previous works were analyzing and investigating the behavior and the performance measure of CRN, authors of [11] have dealt with two reliable servers in which all the inter-event times (arrival, service and retrial) were exponentially distributed.

In papers [8, 12] authors took into consideration that the servers are subject to breakdowns. In these works all the inter-event times including the failure and repair times were exponentially distributed. Same authors of papers [8, 12] have investigated further the CRN with breakdowns, in paper [9] the failure and repair time of the servers are non-exponential distributed (Hypo-exponentially and Hyper-exponentially).

In the present work we add Gamma, Pareto and Log-normal distributions to the Hypo-Exponential and Hyper-Exponential for the failure and repair times and we provide different parameters for these distributions, in order to investigate and show the impact of the distributions and their parameters on the behavior of the system.

Parameters	Value at moment t	Maximum Value
Primary sources	$k1(t)$	N1
Secondary Sources	$k2(t)$	N2
Primary arrival rate	λ_1	
Secondary arrival rate	λ_2	
Number of requests at the queue (FIFO)	$q(t)$	N1-1
Number of requests at the orbit	$o(t)$	N2-1
Primary service rate	μ_1	
Secondary service rate	μ_2	
Failure rate of the primary server	γ_1	
Failure rate of the secondary server	γ_2	
Repair rate of the primary server	σ_1	
Repair rate of the secondary server	σ_2	

Table 1: Parameters of the simulation

The set of parameters used in the simulation are shown in Table 1. Table 1 presents all the values needed for the simulation and their maximums (if exists), we can see that the primary number of sources is $k1$ at moment t, however the Maximum number for this values is $N1$, similarly for the second server has $k2$ a number of sources and $N2$ is the max number of secondary sources. As the Maximum number of primary sources in the system is $N1$, the Maximum of the requests in the queue will be $N1 - 1$ since the server deals with one user in the same time, likewise for the orbit the maximum number of requests at the orbit will be $N2 - 1$.

3. Simulation Results

We used the batch-mean method to estimate the mean response times of each request. This method is one of the most common confidence interval techniques which is used for steady-state simulation output analysis. See for example [1, 7, 12].

This method aims to obtain a series of independent samples(batches) by accumulating a number of contiguous observations of the simulation in order to produce point and interval estimators. Each batch size needs to be enough large so that the sample averages will not be highly correlated. Then we take the average of the data points in each batch in order to get the final mean or variance.

A confidence interval for the mentioned above technique can be obtained using the corresponding theorem as can be seen in [5].

The distribution's confidence intervals are displayed in Table 2.

$$\widehat{\mu}_N \pm t_{N,1-\frac{\beta}{2}} \frac{S}{\sqrt{N}} \quad \text{with confidence level } 1-\beta.$$

- $\widehat{\mu}_N$: Estimator for the mean response time
- N: Number of Batches
- $t_{N,1-\frac{\beta}{2}}$: The $1-\frac{\beta}{2}$ critical value of the Student t distribution with N degrees of freedom
- S: Sample standard deviation.

Using our simulation program we could display different figures for several case combinations in which we focused on the effect of the distribution of the failure and repair times on the mean response time of secondary users using different distributions.

Both Figure 2 and 3 show the mean response time of secondary users in function of the primary repair intensity σ_1 using different distributions (Hypo-Exponential, Hyper-Exponential, Gamma, Pareto and Log-normal) for the primary operating time, knowing that the Exponential distribution was used for the rest of the inter-event times (arrival, service, retrial and failure).

Fig.	**Obs. Point**	**Distribution**	**N**	$t_{N,1-\frac{\beta}{2}}$	**95% Confidence Interval**	
					LB	**UB**
2	0,05	Pareto	68	1.995	51.00341	60.68059
		Gamma	70	1.994	50.36767	60.85413
		Hypo	65	1.997	52.49875	61.91125
		Log-normal	60	2.000	50.85023	63.76017
3	0,06	Pareto	86	1.988	39.51273	45.94387
		Gamma	85	1.988	20.90805	29.10195
		Hyper	79	1.990	28.46989	39.66451
		Log-normal	90	1.987	27.29105	38.18235
4	5	Pareto	105	1.960	47.26031	58.73689
		Gamma	95	1.985	66.61442	80.87038
		Hyper	83	1.989	59.83858	68.57142
		Log-normal	89	1.987	56.9076	68.7024
5	6	Pareto	115	1.960	69.67006	84.29174
		Gamma	97	1.985	68.84348	82.95172
		Hypo	92	1.986	68.33082	84.06698
		Log-normal	87	1.988	70.94546	83.30734

Table 2: Confidence intervals of the figures

Figure No.	$N1$	$N2$	λ_1	λ_2	μ_1	μ_2	σ_1	σ_2	γ_1	γ_2
Figure 2,3	6	10	0.6	0.1	1.5	1	x-axis	0.5	5	4
Figure 4,5	6	10	0.6	0.1	1.5	1	0.5	0.5	5	x-axis

Table 3: Numerical values of model parameters

Distribution		**Hyper**	**Hypo**	**Gamma**	**Pareto**	**Lognormal**
Figure 2,5	**Mean**	N/A	0.2	0.2	0.2	0.2
	Variance	N/A	0.03	0.03	0.03	0.03
	Parameters	N/A	$\lambda_1 = 0.0292$ $\lambda_2 = 0.1707$	$\alpha = 1.333$ $\beta = 6.667$	$\alpha = 2.5275$ k=0.6043	m=-1.889 $\sigma = 0.74807$
Figure 3,4	**Mean**	0.2	N/A	0.2	0.2	0.2
	Variance	0.4	N/A	0.4	0.4	0.4
	Parameters	$\lambda_1 = 0.2$ $\lambda_2 = 0.632$	N/A	$\alpha = 0.1$ $\beta = 0.5$	$\alpha = 2.04880$ k=0.51191	m=-1.657 $\sigma = 1.5485$

Table 4: Values of the distribution parameters

As expected the mean response time of the users decreases with the increment of the repair intensity. In Figure 2 we can observe the insensitivity of the distributions where the squared coefficient of variation was less than one, as the difference between the distributions was almost negligible. However, the difference between distributions is very significant in Figure 3 as the squared coefficient of variation is greater than one.

The last two results are related to the effect of the failure intensity for the

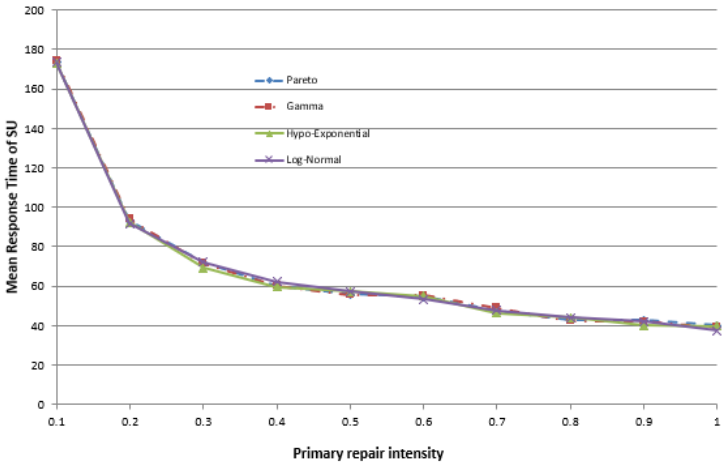


Figure 2: The effect of the Primary repair intensity on the mean response time of the Secondary Users

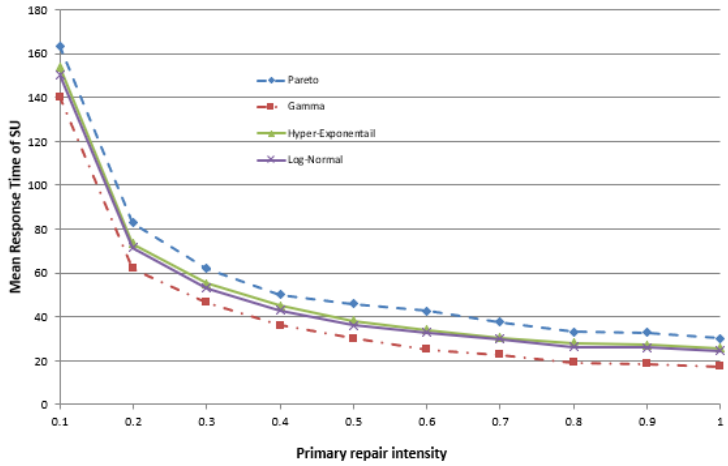


Figure 3: The effect of the Primary repair intensity on the mean response time of the Secondary Users

secondary server γ_2 versus the mean response time of secondary users. It should be noted that in all the figures we assumed that both Primary and Secondary servers of our network are non-reliable.

Figure 4 shows the mean response time of SU in function with the secondary failure intensity, knowing that all the means and variances of the different distributions were equals and their squared coefficient of variation was greater than one as shown in Table 3.

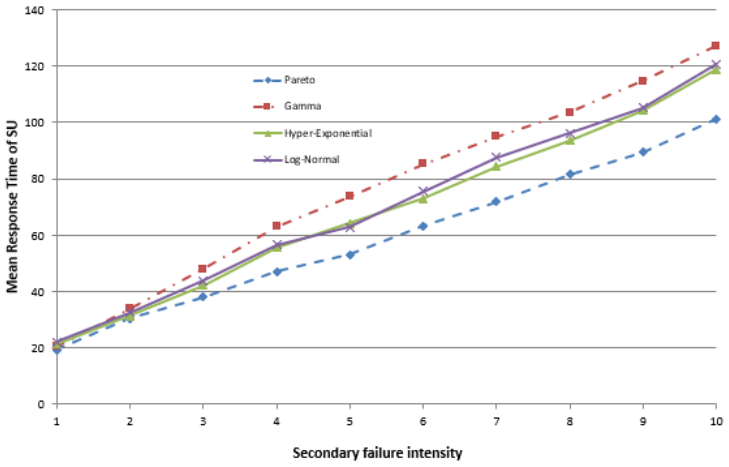


Figure 4: The effect of the Secondary failure intensity on the mean response time of the Secondary Users

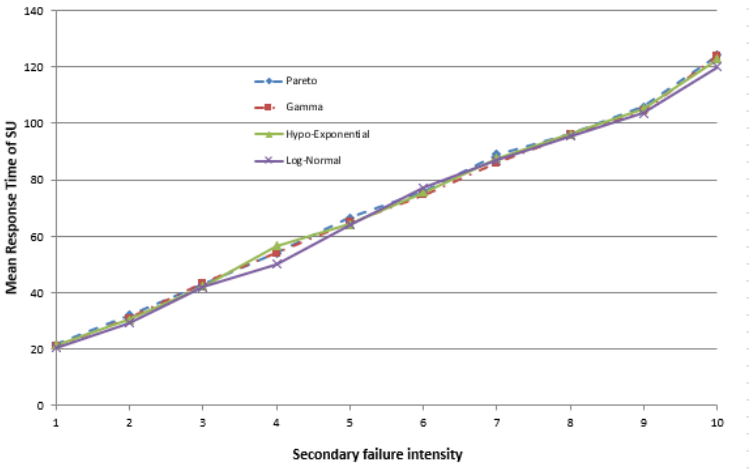


Figure 5: The effect of the Secondary failure intensity on the mean response time of the Secondary Users

Even though the mean and the variance of the distributions were equals, a significant difference can be seen between the values of mean response time of the SUs, mainly for Gamma and Pareto distribution. The effect of the used distributions can be obviously observed in this figure and it shows sensitivity to the involved distributions.

In Figure 5 where the squared coefficient of variation was less than one we

can see how the the mean response time of the secondary users increases with the increment of the failure intensity. By examining closely the figure no effect can be seen regardless of the different distributions. Furthermore, all the values of mean response time were nearly similar which means insensitivity to the type of distributions.

4. Conclusion

In this paper a finite-source retrial queuing system is presented with a non-reliable server in each subsystem. We showed the effect of several distributions concerning the failure and repair times of the servers on the mean response time of secondary users. A significant effect of these distributions was seen when the squared coefficient of variation was greater than one having the same mean and variance, however the impact was almost negligible when it was less than one. Lastly, as future works we would like deal with more distributions, in order to investigate further their influence on the cognitive radio networks.

Acknowledgments. The authors are very grateful to the reviewers for their valuable comments and suggestions which improved the quality and the presentation of the paper.

References

[1] E. Carlstein, D. Goldsman: *The use of subseries values for estimating the variance of a general statistic from a stationary sequence*, Ann. Stat. 14 (1986), pp. 1171–1179, doi: https://doi.org/10.1214/aos/1176350057.

[2] N. Devroye, M. Vu, V. Tarock: *Cognitive radio networks*, IEEE Signal Process. Mag. 25 (2008), pp. 12–23, doi: https://doi.org/10.1109/MSP.2008.929286.

[3] S. Gunawardena, W. Zhuang: *Modeling and Analysis of Voice and Data in Cognitive Radio Networks*, Berlin 2191–5768: Springer, 2014, doi: https://doi.org/10.1007/978-3-319-04645-7.

[4] C. Henk, A. Tijms: *First Course in Stochastic Models*, Jhon Wiley and sons LTD, 0–471–49880–7, 2003, doi: https://doi.org/10.1002/047001363X.

[5] E. Jack Chen, W. David Kelton: *A Procedure for Generating Batch-Means Confidence Intervals for Simulation: Checking Independence and Normality*, SIMULATION 83.10 (2007), pp. 683–94, doi: https://doi.org/10.1177/0037549707086039.

[6] A. Kuki, T. Berczes, B. Almasi, J. Sztrik: *Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs*, J. Math. Sci. 132.2006, pp. 677–685, doi: https://doi.org/10.1007/s10958-006-0014-0.

[7] A. Law, W. Kelton: *Simulation Modeling and Analysis, 2nd edn.* New York: McGrawHill College, 1991.

[8] H. Nemouchi, J. Sztrik: *Performance evaluation of finite-source cognitive radio networks with collision using simulation*, IEEE Inter. Conf. on Cognitive Infocommuni. 49 (2018), pp. 109–122,
doi: https://doi.org/10.33039/ami.2018.12.001.

[9] H. Nemouchi, J. Sztrik: *Performance simulation of finite source cognitive radio with servers subjects to breakdowns and repairs*, Journal of Mathematical Sciences 37 (2019), pp. 1072–3374.

[10] F. Paluncic, A. Alfa, B. Mahraj, H. Simba: *Queueing Models for Cognitive Radio Networks: A survey*, IEE Access 6 (2018), pp. 10–1109,
doi: https://doi.org/10.1109/ACCESS.2018.2867034.

[11] J. Sztrik, T. Berczes, B. Almasi, A. Kuki, J. Wang: *Performance modeling of finite-source cognitive radio networks*, J. Acta Cybern. 22.3 (2016), pp. 617–631,
doi: https://doi.org/10.14232/actacyb.22.3.2016.5.

[12] J. Sztrik, T. Berczes, H. Nemouchi, A. Z. Melikov: *Performance modeling of finite-source cognitive radio networks using simulation*, Com. Comput. Inform. Sci. 678 (2016), pp. 64–73,
doi: https://doi.org/10.1007/978-3-319-51917-3_7.

[13] T. Van Do, H. Nam, A. Horvath, W. Jinting: *Modelling opportunistic spectrum renting in mobile cellular networks*, Journal of Network and Computer Applications Elsevier 52 (2015), pp. 129–138,
doi: https://doi.org/10.1016/j.jnca.2015.02.007.

[14] E. Wong, C. Foch, F. Adachi: *Analysis of cognitive radio spectrum desicion for cognitive radio networks*, IEEE J. Select. Areas Common. 29 (2011), pp. 757–769,
doi: https://doi.org/10.1109/JSAC.2011.110408.

[15] M. H. Zaghouani, J. Sztrik, A. Z. Melikov: *Reliability Analysis of Cognitive Radio Networks*, (IDT) (pp. 557–562). IEEE (2019),
doi: https://doi.org/10.1109/DT.2019.8813383.

[16] Y. Zhao, L. Bai: *Performance Analysis and Optimization for Cognitive Radio Networks with Classified Secondary Users and Impatient Packets*, Mobile Information Systems, Hindawi (2017), Article ID 3613496,
doi: https://doi.org/10.1155/2017/3613496.

Methodological papers

Annales Mathematicae et Informaticae
***52** (2020) pp. 269–279*
DOI: https://doi.org/10.33039/ami.2020.12.002
URL: https://ami.uni-eszterhazy.hu

Cooperative learning methods in mathematics education – 1.5 year experience from teachers' perspective

Tünde Berta[a], **Miklós Hoffmann**[b]

[a]Selye János University, Komarno, Slovakia
bertat@ujs.sk

[b]Eszterházy Károly University, Eger, Hungary
hoffmann.miklos@uni-eszterhazy.hu

Submitted: November 15, 2020
Accepted: December 9, 2020
Published online: December 17, 2020

Abstract

Inclusive education and inclusive learning environment have become a major issue in Hungarian schools in Slovakia in the last decade. The number of underprivileged students with marginalised social background has risen tremendously. The primary questions of education including the provision of a proper learning environment in heterogeneous classes, the tackling of status problems among students have become inevitable. The introduction of the Complex Instruction Methodology (CIP) into the education process created an opportunity for teachers to work with heterogeneous groups in classes. Mathematics by definition creates heterogeneous groups in schools as due to large knowledge gaps among students the differentiation of work and the cooperation of students is extremely difficult. The CIP method is a special group work designed for heterogeneous groups which could provide enormous help for the work of the teachers. In the current work we would like to present the adaptation process of the CIP method in two Hungarian schools in Slovakia. By conducting a series of interviews with teachers we are going to analyse the efficiency of the CIP method in cooperative mathematics classes.

Keywords: Complex Instruction Program, problem solving, cooperative math-

ematics, inclusive teaching, group work, heterogeneous student group
MSC: 97D20

1. Introduction

Teaching even in its simplest form is one the most complex tasks. The teacher has to adapt to classes numbering 20 to 30 people, to class syllabuses and to individual need of students as well. The usual teaching methods are not in compliance with current expectations of the society, they have changed a lot in the past 15 years. The labour market presents a demand for young people who are innovative and adaptive to the rapidly changing world. There is a widespread need for new basic skills affordable to everyone. There are new skills to be acquired: to solve problems, to look for sources needed for the task, to cooperate with each other in groups, to express their opinion, to focus on a given problem. It is an outstanding obstacle for Hungarian schools in Slovakia as the composition of the students is extremely heterogeneous. The work with heterogeneous groups of students according to knowledge and social status presents a highly demanding challenge for the teachers. The international and Slovakian studies on teaching has revealed that the unchanged system of education has not been successful. Critique has been oriented both towards the content and the methodology of teaching. A substantial part of students do not have sufficient level of development (Monitor5, Monitor9, centralised school leaving exams). The students lost their motivation, studying has become a painful experience for them. Traditional teaching methods are not sufficient for success.

There are new promising methods available, but their infiltration to the Hungarian education system in Slovakia is very poor. Due to the Slovak official state language there is a time shift until they can reach Hungarian students in Slovakia. It takes years of preparation until a new textbook or a new development reaches its target audience. The introduction of new teaching methods is a slow procedure. In the past years project methodology and cooperative learning has come to the forefront. Its adaptation Hungarian schools is similar to its adaptation in Hungary, most Hungarian teachers in Slovakia use the actual available literature in Hungary. In the study we shall focus on the adaptation of the Complex Instruction Methodology into the mathematics teaching in Slovakia: the process itself and the experiences of teachers.

2. CIP – a special cooperative learning process

Cooperative learning has become the most dynamic teaching method in Western Europe and North America. Large number of studies unveiled its efficiency in various fields of education. Cooperative learning has become a summarizing term for various methods focusing on groups, classes and schools ([1, 8, 10, 15]). Cooperative learning is a management of learning in which the acquisition of knowledge,

cognitive knowledge, social motives and skills, learning motives are parallel in time and equal in their status. During cooperative learning the tasks are organised in such a way that students cannot cope with them individually, there is an essential need for constructive cooperation. The society building is not achieved by separate tools, but it is inherently built into the process [7].

The provision of equal chance to each student has become more important. The teaching method should take into account both the needs of elite students and of students with slower and lower achievements, they should have a chance to converge to students with higher achievements. It has become a major obstacle to teachers. In Hungarian schools in Slovakia this problem has become even more eminent, the share of students with disadvantageous social background has been rising in the past decades. The number of parents choosing Slovak schools for their children has been rising as well due to the fact that majority schools with Slovak language of education have better social ratios, thus enhancing the assimilation of minorities in Slovakia. When looking for plausible solutions our attention has been drawn to the Complex Instruction Programme, which has been successfully adapted in Hungary since 2000, first at the school IV. Béla Elementary School by Emese K. Nagy and her colleagues [8]. Based on their experience, a complex system called Complex Basic Programme has been developed in Eszterházy Károly University in 2016, and has been introduced to several Hungarian primary schools since 2018 [9, 11, 12].

The Complex Instruction Programme developed at the Stanford University provides an excellent opportunity for schools and teachers to create an inclusive space for students. The founding scientists of the programme assumed that the social structure of a given class could be transformed by the teachers implying the change in the environment of education which could lead to a modified learning environment for students which is most adaptable to their individual skills and needs and provides an environment for tackling status problems in classes.

The main goal of the programme was to create an equal opportunity learning environment in which access to learning and to teaching materials is equally provided for students. The programme has been designed so that teachers would have tools and methods for tackling the heterogeneity of linguistic and cultural background of students. The primary idea has been formulated that each student is capable of acquiring a higher level of knowledge once the proper conditions have been set for learning. It has been a guiding principle that the cooperation among students should be based on equal status, each student should participate in the solution of selected problems and tasks. The new method is independent of special subjects and teaching materials. In addition to cooperative learning the development of an individual student has been taken into account as well ([2, 8]).

Cooperative learning requires a different approach to the teaching planning process as well a different attitude of the teachers. The classic, in our region overruling frontal teaching method puts the teacher in the forefront, his knowledge and methods are superior. In classes using cooperative learning methods the activities of students are prior, the teacher's role is modified: helping and organising is required

from teachers. The CIP method is independent of specific teaching subjects, in addition to its ability to develop competencies it could be effectively used in individual development and work with talented students as well.

3. The introduction of CIP method in Hungarian schools in Slovakia

The cross-border project called KIP ON LEARNING – Schools in a changing world – Inclusive, innovative and reflexive teaching and learning – cross border exchange of know-how was launched in September 2017 supported by Interreg V-A Slovakia Hungary Cross Border Cooperation Programme.[1] The main objective of the project has been the introduction of CIP method in two Hungarian elementary schools in Slovakia: Feszty Elementary School in Hurbanovo (Ógyallai Feszty Árpád Magyar tanítási Nyelvű Alapiskola) and in Helmeczy Mihály Elementary School in Královsky Chlmec (Királyhelmeci Helmeczy Mihály Magyar Tanítási Nyelvű Alapiskola). The two schools are similar in number of students and in social status of students, one is in Western Slovakia the other in Eastern Slovakia. The number of students is approximately 300 with a large share of students from socially marginalized groups (30–35%) and Roma pupils (appr. 25%).

The CIP method has been adapted in these two elementary schools. The preparation process included purchase of equipment and surveys in both schools. The introduction of CIP methodology started in January 2018. All the teachers from both schools (60 teachers) participated at the CIP training course (60 hour training) which was followed by tutoring until the end of the project. The teachers started developing their own CIP materials after the training. The students were prepared for the new methodology, sociometric analysis was made in order to divide students into groups within classes. The teachers were acquainted with the CIP methodology in the first semester, with the help of tutors they prepared its adaptation to the education system of Slovakia. The teachers started using the method in their classes starting from the 2018/19 academic year.

The sustainability period for the project is 5 years, CIP classes were held until the closures due to the COVID-19 pandemics in 2020. The effects and effectiveness of the new methodology are analysed by continuous surveys, where several indicators are monitored: social and affective factors, number of absences, change in the status of students etc.

4. CIP learning process and the social competences

Research focusing on innovative teaching methods including cooperative learning has shown numerous accomplishments. The primary goal of the new method is the

[1]Project number: SKHU/1601/4.1/172.

elevation of the knowledge level of students as well as experiencing success in the school environment. Acceptance of others and respect have been key elements as well [8].

Cohen, Lotan describe group work as: „Students are working in groups which are small enough for each member to participate in solving tasks. It is a requirement that students are able to work autonomously without direct control of the teacher." ([2], p. 22.)

The CIP methodology enhances the convergence of less developed students while advanced students do not need to slow down in their work. Studies have revealed that group work enhances the understanding of the theoretical part of the subject, they are less distracted, and are able to focus more when compared to classic frontal teaching ([4, 13, 14]). Students working in groups are more tolerant in accepting their fellow classmates of different origin or social status.

CIP could be successfully used in classes with different levels of knowledge, origin and language. Students can learn from each other, they serve as role models to each other, they are interconnected, dependent on each other. Joint effort brings the experience of success to the students, authentic intellectual pride could be achieved, the outcome is on a higher level when compared to individual work. Hereby comes the main slogan of the programme: We are smarter together than alone [2].

Group work with tasks requiring different levels of knowledge and skills brings students closer to cooperation, enabling the unravelling of talents in students. Students with different levels of knowledge and skill could present different problem solving strategies, which develops their strengths while acquiring new skills. The proper compilation of tasks creates an opportunity for each student to show their understanding and abilities thus students with different social backgrounds can solve tasks successfully teaching [8].

The aim of the teacher through the organisation of group work is to provide equal opportunities to each student to participate and to ensure that everyone has a skill that leads to successful solution to the problem. The teachers need to learn to tackle the differences among students in order to enhance the development of each student. The development of the teacher's skill to organise groups is a key element to the success of the new method. When compared to classical methods where teachers are more eager to get into direct involvement and control in the new method there is no need for it. It is necessary that teachers stay in the background thus enabling the individual work of students based on the previously set roles.

It is essential that teachers are able to assign their controlling role to students. The teacher stays in the background ready to get involved if necessary thus creating a space for students to make their decisions and to develop their personalities. Students acquire skills needed for individual planning and organising their learning process [6].

Each member of the group benefits from the work if tasks are not based on routine work, when it needs to be discussed, when the outcome is not visible from the very beginning [3]. Learning from more advanced students is a key element in

cognitive development as VYGOTSKY stated:"Learning initiates various inner processes, which can only work if children get into connection with each other and they work together ([17] p.90). Joint work, common discussions, interactions provide opportunity for students to cooperate irrespectively of their level of knowledge and their rank in the class.

5. Cooperativ learning and mathematics – mathematics in CIP

There are many benefits coming from collaboration on math lessons in the classroom. Education is not equitably accessible to all students. Collaborative learning could contribute to closing the achievement gap among students and to reach greater success in mathematics. Learning by cooperative and problem-based approach students at elementary schools could get a more detailed impression on higher-level math. Students often believe that success in math is based on memorization. In addition to basic mechanics of solving problems it is of high importance that students are able to formulate and interpret more complex problems, and are able to work in groups while managing problem solving strategies. This process makes mathematics such an interesting subject. Cooperative problem solving motivates students more and could introduce them into further careers in mathematics [5].

Cooperative problem-based learning enhances the use of different abilities and could lead to mathematical growth. The cooperation develops interpersonal skills, makes conflict resolution easier and gives students some leadership experience. Working on well designed tasks provides students with common goals, sharing ideas connects people. Students can immerse the beauty and the fun in mathematics. Teachers can use cooperative learning activities in order to help students find connections between the concrete and abstract level of instruction through peer interactions and properly designed activities. Cooperative learning can contribute to the promotion of classroom discourse and oral language development [5, 7].

In Hungary a complex mathematical programme based on CIP methodology has been elaborated, called Logical Basic Program. The main goal of the basic program focuses on the change of the approach to mathematics teaching including motivation, experience based learning and development. The skills and knowledge of students should be impacted by by playing games and using game-structures, new approaches, promoting enactive and visual representation, positive reassuring environment enhancing creativity taking into consideration the level of students' progress in mathematics. Theoretical works of Zoltán Dienes, Tamás Varga and Jerome Bruner and problem and research based theories have supported this approach.

Why does the CIP method work on math lessons? First, students are more willing to solve challenging problems as a group. Second, students are often able to explain things to each other in ways that make more sense than the teacher's origi-

nal explanation. Third, students are more willing to ask questions and take risks in small groups. Fourth, students learn more when they invest in math discourse. It is of high importance teachers use flexible grouping throughout the school year so that each student is challenged appropriately and the rotation of the roles is supported. When classrooms achieve this balance, all students have the opportunity to learn within their zone of proximal development [17].

6. The experiences of teachers with CIP – analysis of interviews with teachers

We have conducted deep interviews with teachers to survey their experiences while using the new CIP method. The participating schools are mid-size involving 2 mathematics teachers, all 4 teachers have been interviewed.

Mapping the experiences of teachers with the introduction of the new teaching method was the main goal of the interviews. The interviews included the teachers's perception of the method as well as its potential to their professional development.

The interviews focused on 4 topics.

1. The first topic included the teacher's motivation, what was their motive for enlarging the scale of used methods beyond classical frontal teaching. The main motivation described by teachers promoted the involvement of those students into mathematics learning who were less interested in the subject. Bringing new motivation for students has been mentioned as well. This has been the hardest part of the education process. Students were described as having low levels of motivation, curriculum described as badly designed, useful textbooks missing from the market, deepening the gap of the level of knowledge among students as well as of their social status. Hungarian schools in Slovakia have a higher share of Roma students which is characterised by strong backlash in the level of knowledge as well as in social status of families. The involvement of Roma students into mathematics learning has been a painful experience. The new CIP method presented an emerging possibility to involve these students into education.

The improvement of the attitude of students during the classes has been mentioned often. The new method to be adapted in these schools should have been able to be inserted into the educational system. International experience with the CIP method helped the teachers' orientation and decision.

2. The second topic included the teachers' experiences on application of CIP into their classes. The questions focused on the behaviour of students during the classes, on changes in social interactions and in mathematics learning. Which competencies in mathematics could be developed most and what change could be induced in the approach of students.

All teachers involved used CIP in classes for exercising and for repetition, the new method was not used for the acquisition of new knowledge. During the interviews the teachers revealed their fears that the essence of mathematics knowledge could be lost during the cooperative classes. An assumption that less developed students would refrain from works was highlighted by the teachers. The teachers assumed that in heterogeneous groups only talented students would follow the course of mathematics classes while all the other students would emerge only as observers.

The perception of the new method has changed during the course of its adaptation. In the first period when both teachers and students were getting familiar with the new method the original assumptions were fulfilled but later the perception has changed. It is still evident that new tasks are solved by more talented students, but with the acquisition of new roles used in the CIP method all students could contribute to the outcome. The most imminent change perceived by the teachers was the enhanced active communication within the groups.

The ability of students to understand texts and solve problems was on a low level (textbooks used on the first stage of education in elementary schools do not encourage the development of these skills. Common analysis of problems and their common solution contributed to the development of these skills in a visible manner. There has been a perceivable change.

The most visible change in the long run while using the new method was the fact that students became more open in classical classes as well, they asked more questions and felt encouraged to participate in debates, the behaviour of students and their attention to classes improved as well. There were issues which could not be successfully implemented, the cyclic rotation of roles within the groups posed difficulties. Presenting the outcomes of the works has always been difficult for students and not only for those with lower level of knowledge and skills. There were students refusing to take that part, teachers were not pushing hard to do so in order to maintain the functionality of the groups. Difference in the knowledge of mathematics was not perceived with use of the CIP method, at least it could not be attached to the use of CIP. One year of experience was not enough for its perception. There has been a positive change however in the attitude of students towards problem solving, mathematical text analysing which could be connected to the use of CIP.

3. The third topic analysed was oriented towards the impact of CIP on the work of mathematics teachers in participating schools. The first thing mentioned by the teachers was the change of used routines, the shift from their comfort zone. The importance of CIP in their education was characterised by the process of the acquisition of a new method by the use of which they could depict the importance and the impact of mathematics in everyday life.The tasks selected by the teachers presented an everyday life case, formulated as an open question. The existing mathematical knowledge was made to be used in a non-conventional environment for the students. The attention paid by the teachers, the evaluation process and the self esteem of students has been outlined as well. There were teachers admitting

that observation and evaluation was missing from their previous work, it was the use of CIP that drew their attention to the importance of these elements. The method was perceived as successful for less developed students whose self esteem and motivation skyrocketed with the use of CIP.

The formulation of groups was questioned by the teachers. There were teachers using the proper method in heterogeneous group formation, but there were several teachers who disagreed, claiming that more homogeneous groups would enhance the solution of more complex mathematical problems.

Teachers who used the proper CIP method argued that cooperative learning often works best if the team members are not of the same level in mathematics. According to their argumentation the more capable students are advancing by teaching the concept while others are advancing by engaging with the problem and wrestling with the solution.

None of the teachers said that CIP could be used for effective talent management in heterogeneous groups, not even with the use of individual tasks. Individual tasks at the end of classes due to time pressure were poorly performed, often even neglected thus their function was completely lost.

The long preparation of CIP classes has been reported as the harshest drawback of the CIP method. It is hard to create innovative tasks for group work. Social sciences provide a more fruitful environment for the use of CIP classes. Drama pedagogy, music, arts are not applicable to mathematics. The types of tasks for group work start repeating after a certain amount of time. The most important factor for the use of CIP in mathematics teaching is the enhanced motivation of students for the subject. It brings new variety into monotony.

4. The last area of questions focused on the training and tutoring. Teachers expressed their satisfaction with the initial training, its content and source as well. The importance of tutoring was strengthened by the responses of the teachers. They did not perceive it as a burden, rather as a collegial help. It created a positive environment among teachers to share their experiences, difficulties and successes. The visiting of classes by the mentors was highly appreciated especially with the following consultations. The teachers would like to be in contact with their mentors in the long run. More class syllabuses are needed for the sustainability of the method in schools which are in line with the requirements in Slovakia [16]. Altogether 600 class syllabuses were prepared during the 24 months of the project, but the share of mathematics syllabuses is low. A collection of Great ideas for group work for each grade would be highly appreciated. In the management the teachers proposed that in class management double classes using CIP methodology would contribute to the success and efficiency of the new method.

7. Conclusion – experiences from the interviews

The introduction of CIP methodology into the teaching system in two Hungarian schools in Slovakia was successful. According to the teachers the method could

and should be used in education. The new method enhances better understanding of mathematics, special vocabulary is acquired by students and used in active debates. The development of problem solving and logical thinking is visible even on classical frontal classes. The most important improvement could be perceived in the behaviour of students during classes, their activity level has been visibly rising after one year of experience. Continuous mentoring is a key factor to the efficiency of teachers' work. The mentoring method should be considered in the adaptation process of other methods, starting freshman teachers should be involved in tutoring on a larger systematic level. CIP found its place in the education system of Slovakia, the penetration of the method into the teaching process of other schools is highly recommended.

Acknowledgements. The authors are thankful for the anonymous reviewers for their valuable comments and support.

References

[1] J. Benda: *A kooperatív pedagógia szocializációs sikerei és lehetőségei Magyarországon I [Success in socialization and opportunities of cooperative pedagogy in Hungary]*, Új Pedagógiai Szemle 9 (2002), pp. 26–37.

[2] E. G. Cohen, R. A. Lotan: *Designing groupwork: Strategies for the heterogeneous classroom Third Edition*, Teachers College Press, 2014.

[3] E. G. Cohen: *Restructuring the classroom: conditions for productive smallgroups*, Review of Educational Research 64.1 (1994), pp. 1–35,
doi: https://doi.org/10.3102/00346543064001001.

[4] E. G. Cohen, R. A. Lotan, C. Leechor: *Can classrooms learn?*, Sociology of education (1989), pp. 75–94,
doi: https://doi.org/10.2307/2112841.

[5] N. Davidson: *Cooperative Learning in Mathematics: A Handbook for Teachers*, Addison-Wesley, 1990.

[6] N. A. Flanders: *Analyzing teaching behavior*, Addison-Wesley, 1970.

[7] K. Józsa, G. Székely: *Kísérlet a kooperatív tanulás alkalmazására a matematika tanítása során.[Experiment for Using Cooperative Learning in Teaching Mathematics]*, Magyar pedagógia 104.3 (2004), pp. 339–362.

[8] E. K. Nagy: *KIP Könyv I-II. [KIP Book I-II.]* Miskolci Egyetemi Kiadó, 1978.

[9] E. K. Nagy, L. Révész: *Differenciált fejlesztés heterogén tanulócsoportokban – metódus, mint a Komplex Alapprogram tanítási-tanulási stratégiája, fókuszban a tanulók státuszkezelése [Differentiated development in heterogeneous groups of students – a method such as the teaching-learning strategy of the Complex Basic Program, focusing on the status management of students]*, Líceum Kiadó, 2019.

[10] S. Kagan: *Cooperative learning*, Kagan Cooperative Learning Publisher, San Juan Capistrano, CA, 1992.

[11] I. Oláhné Téglási: *A logika alapú alprogram koncepciója [The concept of a logic-based programme]*, Líceum Kiadó, 2018.

[12] I. Oláhné Téglási: *Megalapozó tanulmány a logika alapú iskolai programok fejlesztéséhez [Groundbreaking study for the development of logic-based school programmes]*, Líceum Kiadó, 2018.

[13] S. Sharan: *Cooperative learning in the classroom: Research in desegregated schools*, Lawrence Erlbaum Assoc Inc, Hillsdale, NJ., 1984.

[14] R. E. Slavin: *Cooperative learning and intergroup relations*, in: James A. Banks and Cherry A. McGee Banks (Eds.): Handbook of research on multicultural education, Jossey-Bass, San Francisco, 2001.

[15] R. E. Slavin: *Cooperative Learning. Research on Teaching Monograph Series*, Longman, NY, 1983.

[16] *Slovakian State Educational Programme*, `https://www.minedu.sk/data/att/7500.pdf`, [Online; accessed 13-11-2020], 2015.

[17] L. S. Vygotsky: *Mind in society: the development of higher psychological processes*, Harvard University Press, Harvard, MA, 1978.

Annales Mathematicae et Informaticae
***52** (2020) pp. 281–298*
DOI: https://doi.org/10.33039/ami.2020.05.004
URL: https://ami.uni-eszterhazy.hu

A survey on the global optimization problem using Kruskal–Wallis test

Viliam Ďuriš, Anna Tirpáková

Department of Mathematics
Constantine The Philosopher University in Nitra
Tr. A. Hlinku 1, 949 74 Nitra, Slovakia
vduris@ukf.sk
atirpakova@ukf.sk

Submitted: April 1, 2020
Accepted: May 26, 2020
Published online: June 3, 2020

Abstract

The article deals with experimental comparison and verification of stochastic algorithms for global optimization while searching the global optimum in dimensions 3 and 4 of selected testing functions in Matlab computing environment. To draw a comparison, we took the algorithms Controlled Random Search, Differential Evolution that we created for this test and implemented in Matlab, and fminsearch function which is directly built in Matlab. The basic quantities to compare algorithms were time complexity while searching the considered area and reliability of finding the global optimum of the 1st De Jong function, Rosenbrock's saddle, Ackley's function and Griewangk's function. The time complexity of the algorithms was determined by the number of test function evaluations during the global optimum search and we analysed the results of the experiment using the "Kruskal–Wallis test" non-parametric method.

Keywords: global optimization; test functions; simplex; population; Controlled Random Search; Differential Evolution; fminsearch; Matlab, Kruskal–Wallis test

MSC: 90C26, 62G09

1. Introduction

In mathematics, we often solve a problem that we characterize as finding a minimum value of an examined function (so-called global optimization) $f\colon D \to R$ on a specific set $D \subseteq R^d$, $d \in N$. This minimum value (global minimum or global optimum) is one or more points from the smallest functional value set D, that is a set $\{x' \in D : f(x') \leq f(x) \ \forall x \in D\}$ [8]. From mathematical analysis, we know the procedure for finding the extreme of a function when $d = 2$ and there are the first and second function derivatives. However, finding a general solution to the problem formulated this way is very difficult (or even impossible) for any d or if the function considered is multimodal or not differentiable [3]. Any deterministic algorithm addressing the generally formulated problem of global optimization is exponentially complex [2]. That is why we use the so-called *randomly working (stochastic) algorithms* to find a solution to this task, which, although not capable of finding a solution, are capable of finding a satisfactory solution to the problem within a reasonable time. Thus, for the same input problem, such an algorithm performs several different calculations and we aim to create conditions for the algorithm so that we reduce the probability of incorrect calculation as much as possible. Today, the use of stochastic algorithms, especially of the evolutionary type, is very successful in seeking global optimization functions [4]. Those are simple models of Darwin's evolutionary theory of populations development using *selection* (the strongest individuals are more likely to survive), *crossing* (from two or more individuals new individuals with combined parental properties will emerge) and *mutation* (accidental modification of information that an individual bears); to create a population with better properties. For some classes of evolution algorithms, the truly best "individuals" of the population are approaching the global optimum.

Experimental verification and comparison of algorithms on test functions offer us an insight into the performance and behavior of the used global optimization algorithms. Based on this, we can then decide which algorithm is most efficient and usable under the given conditions when solving practical tasks. The most basic test functions include the 1st De Jong function, Rosenbrock's saddle (2nd De Jong function), Ackley's function and Griewangk's function. Matlab source code of these functions can be found in [12].

In the Matlab environment, there are several implemented optimization functions, of which the `fminsearch` function [6] is very important. The `fminsearch` function serves for finding a global minimum of the function of multiple variables. The variables of the function, the global minimum of which we are looking for, are entered into the vector and, also, we specify the so-called start vector x_0, from which the search for the minimum will start. The start vector x_0 must be sufficiently close to the global minimum (not necessarily in unimodal functions only), because different estimates of the start vector may result in different local minima being found instead of the global one. Since the algorithm fminsearch is based on the so-called simplex method [9], it may happen that the solution will not be found at all. Otherwise, the `fminsearch` function returns the vector x, that is the point

at which the global minimum of the given function is located.

Some optimization parameters for global minimum search can be specified in `options` structure using the function `optimset` [7]. Then the generic call command of the `fminsearch` function has the form

$$[\text{x, fval, exitflag, output}] = \text{fminsearch(fun, x0, options)},$$

where `fun` is a string that records a given mathematical function, $x0$ start vector, search setup `options, x` is the resulting vector of the global minimum, `fval` functional value at x, `exitflag` is a value, which specifies the type of search termination and output is a structure that contains the necessary optimization information (algorithm used, number of function evaluations, number of iterations).

For each algorithm, we need to distinguish four types of search termination. Type 1 is the correct completion of the algorithm when it is found (sufficiently close) to the global minimum, type 2 means that the algorithm is completed by reaching the maximum allowed number of iterations (although it converts to a global minimum), type 3 is an early convergence (the algorithm has completed searches in the local minimum) and type 4 means that browsing is completed by reaching the maximum allowed number of iterations, but no close point to the minimum has been found.

In order to determine the type of algorithm termination for the `fminsearch` function, it is possible to use the nested `funcCount` element of the `output` structure (which gives the number of function evaluations). You can also find out how to complete by using the `exitflag` element.

The *Controlled Random Search* (CRS) algorithm [11] works with a population of N points in space D, from which a new point y is generated with the so-called *simplex reflex*. *Simplex* $S = \{x_1, x_2, \ldots, x_{d+1}\}$ is a set of randomly selected $d+1$ space D points. In simplex, we find the point $x_h = \max_{x \in S} f(x)$ with the highest functional value and, as the worst of simplex, we remove it. To the remaining d points, we find their center of gravity

$$g = \frac{1}{2} \sum_{x \in S} (x - x_h).$$

Reflexion means a point overturning x_h around the center of gravity g to obtain a point

$$y = g + (g - x_h) = 2g - x_h.$$

The simplest variant of the reflection algorithm can then be entered as a function in Matlab as follows: [12]

Reflection algorithm

```
1: function [y] = reflex(P)
2: N = length(P(:, 1))
3: d = length(P(1, :)) - 1
4: v = random_simplex(N, d + 1)
5: S = P(v, :)
```

```
6: [x, id] = max(S(:, d + 1))
7: x = S(id, 1:d)
8: S(id, :) = [ ]
9: S(:, d + 1) = [ ]
10: g = mean(S)
11: y = 2*g − x
```

where a random selection of a set S is provided by the function

Random selection algorithm

```
1: function [res] = random_simplex(N, j);
2: v = 1:N;
3: res = [ ];
4: for i = 1:j
5: index = fix(rand(1) * length(v)) + 1;
6: res(end + 1) = v(index);
7: v(index) = [ ];
8: end
```

If the $f(y) < f(x_h)$, the point y of the population replaces the point x_h and we continue to do so. In case that $f(y) \geq f(x_h)$, simplex is reduced. By replacing the worst points of the population, this is concentrated around the lowest functional point being sought. However, the reflection does not guarantee that the newly generated point y will be in the searched area D. Then, we flip all coordinates $y_i \notin \langle a_i, b_i \rangle, i = 1, \ldots, d$, and inside of the searched area D around the relevant side of the $d-$dimensional rectangular parallelepiped D. The algorithm of the so-called mirroring can be entered as a function in Matlab:

Mirroring algorithm

```
1: function [res] = mirror(y, a, b);
2: f = find(y < a | y > b);
3: for i = f
4: while(y(i) < a(i) | y(i) > b(i))
5: if y(i) > b(i)
6: y(i) = 2 * b(i) - y(i);
7: elseif(y(i) < a(i))
8: y(i) = 2 * a(i) - y(i);
9: end
10: end
11: end
12: res = y;
```

The algorithm's source text itself, *Controlled Random Search*, can be then written as an m-file `crs.m` in Matlab.

CRS algorithm

```
1: function [FunEvals, fval, ResType] = crs(N, d, a, b, TolFun, MaxIter, fnear, fname);
2: P = zeros(N, d + 1);
3: for i = 1:N
4: P(i, 1:d) = a + (b - a).* rand(1, d);
5: P(i, d + 1) = feval(fname,(P(i, 1:d)));
6: end
```

```
7: [fmax, indmax] = max(P(:, d + 1));
8: [fval, indmin] = min(P(:, d + 1));
9: FunEvals = N;
10: while (fmax - fval > TolFun) & (FunEvals < d * MaxIter)
11: y = reflex(P);
12: y = mirror(y, a, b);
13: fy = feval(fname, y);
14: FunEvals = FunEvals + 1;
15: if(fy < fmax)
16: P(indmax, :) = [y fy];
17: [fmax, indmax] = max(P(:, d + 1));
18: [fval, indmin] = min(P(:, d + 1));
19:  end
20: end
21: if fval <= fnear
22:  if (fmax - fval) <= TolFun
23: ResType = 1;
24:  else
25: ResType = 2;
26:  end
27: elseif (fmax - fval) <= TolFun
28: ResType = 3;
29: else
30: ResType = 4;
31: end
```

The `FunEvals` variable is the counter of the number of algorithms' function evaluations. As the previous selection N of population points results in N function evaluation, it must be preset to the value N. Line 10 represents a search termination condition $(f_{max} - f_{min} < \epsilon) \vee (FunEvals > MaxIter * d)$ where d is the dimension of the searched area, f_{max} is the largest functional value that is located in the searched population, f_{min} is the smallest functional value, ϵ is the tolerance of the distance of the largest and smallest functional value, `MaxIter*d` is the limitation of the maximum number of permitted function evaluations during the execution of the algorithm. For test functions, where the solution to the problem is known in advance, it is sufficient that the best point of the population has a value less than `f_near`, a value close enough to the global minimum that we pre-set. At the end of the algorithm, we find the type of search termination.

Differential Evolution is a stochastic algorithm for a heuristic search for a global minimum using evolutionary operators [10], [1]. The *Differential Evolution* algorithm creates a new population Q by gradually creating a point y for each point $x_i, i = 1, \ldots, N$ of the old population P, and assigning a point with a lower functional value to the population Q from that pair. The point y is created by crossing the vector v, where the point v is generated from three different points, r_1, r_2, r_3 which are randomly selected from the population P and different from the point x_i of the relationship $v = r_1 + F\,(r_2 - r_3)$, where $F > 0$ is the input parameter which can be determined according to different rules and the vector x_i so that any of its elements $x_{ij}, i = 1, \ldots, N, j = 1, \ldots, d$, is replaced by a value v_j with probability $C \in \langle 0, 1\rangle$. If no change occurs for x_{ij} or for $C = 0$ one randomly selected vector x_i element is replaced. We can see that, compared to the algorithm

Controlled Random Search, Differential Evolution does not replace the worst point in a population but only the worse of a pair of points and thus the *Differential Evolution* algorithm tends to end searches in a local minimum. On the other hand, however, it converges more slowly with the same end condition. The algorithm for generating a point y can be entered as a function in Matlab [12]:

Algorithm for generating a point y

```
1: function [y] = gen(P, F, C, v);
2: N = length(P(:, 1));
3: d = length(P(1, :)) - 1;
4: y = P(v(1), 1:d);
5: re = rand_elem(N, 3, v);
6: r1 = P(re(1), 1:d);
7: r2 = P(re(2), 1:d);
8: r3 = P(re(3), 1:d);
9: v = r1 + F * (r2 - r3);
10: prob = find(rand(1, d) < C);
11: if(length(prob) == 0)
12:   prob = 1 + fix(d * rand(1));
13: end
14: y(prob) = v(prob);
```

Selecting points r_1, r_2, r_3 from the population P provides a function

Algorithm for selecting points r_1, r_2, r_3

```
1: function [res] = rand_elem(N, k, v);
2: c = 1 N;
3: c(v)=[ ];
4: res = zeros(1, k);
5: for i = 1:k
6:  index = 1 + fix(rand(1) * length(c));
7:  res(i) = c(index);
8:  c(index) = [ ];
9: end
```

We construct the source text of the algorithm *Differential Evolution* in the same way as with the algorithm C*ontrolled Random Search* as the m-file `difevol.m`.

Differential Evolution algorithm

```
1: function [FunEvals, fval, ResType]
= difevol(N, d, a, b, TolFun, MaxIter, fnear, fname, F, C);
2: P = zeros(N, d + 1);
3: for i = 1:N
4: P(i, 1:d) = a + (b - a) .* rand(1, d);
5: P(i, d + 1) = feval(fname, (P(i, 1:d)));
6: end
7: fmax = max(P(:, d + 1));
8: [fval, imin] = min(P(:, d + 1));
9: FunEvals = N;
10: Q = P;
11: while (fmax - fval > TolFun) (FunEvals < d * MaxIter)
12: for i = 1:N
13: y = gen(P, F, C, i);
```

```
14: fy = feval(fname, y);
15: FunEvals = FunEvals + 1;
16: if(fy < P(i, d + 1))
17: Q(i, :) = [y fy];
18: end
19: end
20: P = Q;
21: fmax = max(P(:, d + 1));
22: [fval, imin] = min(P(:, d + 1));
23: end
24: if fval <= fnear
25: if (fmax - fval) <= TolFun
26: ResType = 1;
27: else
28: ResType = 2;
29: end
30: elseif (fmax - fval) <= TolFun
31: ResType = 3;
32: else
33: ResType = 4;
34: end
```

2. Methodology of research and algorithms verification and comparison

The experiment was carried out at the Faculty of Natural Sciences of Constantine the Philosopher University in Nitra during the academic years 2018/2019 and 2019/2020. A total of 42 students of single branch study of mathematics and students of teaching combined with maths, who selected the subject numerical mathematics, participated in the experiment. The group of students was taught a selected part "global optimization" of the mathematics curriculum with use of Matlab.

The aim of our research was to verify the behavior and efficiency of three selected algorithms in the global optimization problem. We used four known test functions to test the efficiency of each of the three selected algorithms. In the experiment while searching a global optimum, we recorded the number of evaluations of each algorithm used for each of the selected function. When algorithms are compared, tests must be performed under the same conditions. The experiment was realized for two different dimensions $d = 3$ and $d = 4$, the number of times the minimum search is repeated to 100, tolerance ϵ to the value `seteps = 1e−7`; and the default value `MaxIter=`10000 for one dimension. The search algorithm ends when the minimum and maximum distance are at the selected value or the maximum number of function evaluations has been reached. Especially for the function `fminsearch`, the above end condition is maintained by the code sequence:

Termination condition for `fminsearch`

```
1: while func_evals < maxfun && itercount < maxiter
2: if max(abs(fv(1)-fv(two2np1))) <= max(tolf,10*eps(fv(1))) &&
3: max(max(abs(v(:,two2np1)-v(:,onesn)))) <= max(tolx,10*eps(max(v(:,1))))
4: break
5: end
```

in the part Main algorithm of the source text fminsearch [6]. The `func_evals` variable is in the role of the variable FunEvals, the maxfun constant in the role of the expression MaxIter*d.

The required limit to the number of algorithm iterations and the tolerance of any point in the population from the local minimum point, but also the necessary tolerance ϵ of functional values and the limit to the maximum number of test function evaluations can be adjusted by setting the appropriate parameters of the structure options for the `fminsearch` function.

optimset('MaxFunEvals', MaxIter*d, 'MaxIter', MaxIter*d, 'TolX', seteps, 'TolFun', seteps));

The dimension of space to be searched, the space limitations, the number of searches repeated and ensuring that the correct function is linked to the algorithm and the correct file name for storing the necessary records are entered through the formal `run_test` parameters that can be run for each global optimization algorithm (depending on the `altype` parameter). While searching for the global optimum, the function writes the repeat number, the number of function evaluations, the type of algorithm termination, the function value of the minimum found into the `filenamesaveres` text file.

Algorithm for running the test

```
1: function run_test(fname, filenamesaveres, boundary_interval, repcount, d, algtype);
2: N = 10 * d;
3: a = boundary_interval * ones(1, d);
4: b = -a;
5: MaxIter = 10000;
6: seteps = 1e-7;
7: fnear = 1e-6;
8: fid = fopen(filenamesaveres, 'a');
9: if(algtype == 3) %fminsearch
10: setval = optimset('MaxFunEvals', MaxIter*d, 'MaxIter', MaxIter*d,
'TolX', seteps, 'TolFun', seteps);
11: end
12: for i = 1:repcount
13: i
14: switch(algtype)
15: case 1 %crs
16: [FunEvals, fmin, ResType] = crs(N, d, a, b, seteps, MaxIter, fnear, fname);
17: case 2 %difevol
18: F = 0.8;
19: C = 0.5;
20: [FunEvals, fmin, ResType] = difevol(N, d, a, b, seteps, MaxIter, fnear, fname, F, C);
21:   case 3 %fminsearch
22: x = a + (b - a).* rand(1, d);
23: [x, fmin, exitflag, output] = fminsearch(fname, x, setval);
24: FunEvals = output.funcCount;
25: if fmin <= fnear
26: if FunEvals < MaxIter*d
27: ResType = 1;
28: else
29: ResType = 2;
30: end
31: elseif FunEvals < MaxIter*d
```

```
32: ResType = 3;
33: else
34: ResType = 4;
35: end
36: end
37: fprintf(fid, '%5.0f ', i);
38: fprintf(fid, '%10.0f', FunEvals);
39: fprintf(fid, ' %1.0f', ResType);
40: fprintf(fid, ' %15.4e', fmin);
41: fprintf(fid, '%1s\r\n', ' ');
42: end
43: fclose(fid);
```

The `fminsearch` algorithm has 4 output parameters `(x, fmin, exitflag, output)`. Thus, in determining the type of algorithm termination for the `fminsearch` function, it is not possible to use the non-existent variable `fmax`. The disadvantage in the `run_test` function above is solved by using the nested `funcCount` element of the output structure, which indicates the number of function evaluations.

We ran the `run_test` function for each dimension, for each algorithm, and for each test function, so we each time got 100 evaluations of the test function by selected algorithm in selected dimension.

Example of calling the run_test function for the 1st De Jong function in dimension 4

```
1: fname = 'dejong';
2: filenamesaveres = 'dejong.txt';
3: boundary_interval = -5.12;
4: repcount = 100;
5: d = 4;
6: algtype = 1;
7: run_test(fname, filenamesaveres, boundary_interval, repcount, d, algtype);
```

For example, program code creates a `dejong.txt` file with 100 rows and 4 columns `i, FunEvals, ResType, fmin` for the 1st De Jong function in dimension 4, with the CRS algorithm used. Thus, for all combinations of the algorithm and the test function, we get 12 files for each dimension, each with 100 evaluations.

3. Results of the experiment and their statistical analysis

Based on the results of the experiment, we can compare the search time complexity [2] and reliability of the algorithms used. The time complexity of the algorithm is determined by the number of test function evaluations during the search, which ensures comparability of results regardless of the speed of the computer used. We analysed the results of the experiment using selected statistical methods. Since we have been following the influence of two factors − the algorithm and function on the assessment numbers, the possibility to use a two-factor variance analysis in

addition to descriptive statistics was offered for the assessment of the results of the experiment. However, we can only use the variance analysis if the following conditions are met: The sample files come from the basic files with normal distributions, the sample files are independent of each other and the variances of the basic files are equal. Given that the observed feature assumptions described above were not met, we used the non-parametric method of Kruskal–Wallis test [5] for the analysis. Since the Kruskal–Wallis test is a non-parametric analogue to a one-factor analysis, all combinations of the levels of the original two factors were a factor: algorithms and types of functions. In our case, we tested 3 algorithms in combination with 4 types of functions, so we gained 12 independent selections (sub-groups) or 12 levels of the factor "algorithm type + function type" in each of the two dimensions.

In the experiment, 100 measurements of the assessment numbers were performed in dimension 3 and 4 in each of the 12 selections (so-called sub-groups), i.e. altogether 1200 measurements. The tested problem is formulated as follows. We test a null hypothesis H_0: the numbers of evaluations in the 12 sub-groups created according to the factor levels "algorithm type + function type" are identical as in the alternative hypothesis H_1: The numbers of evaluations in the 12 sub−groups created according to the factor levels indicated are not identical (or, at least at a level, they are different). As we have already stated, since the condition of the normal distribution of observed features was not met, we used the Kruskal–Wallis test to test the null hypothesis.

The Kruskal–Wallis test is a non-parametric analogue to one-factor variance analysis, i.e. it allows testing the hypothesis H_0 that $k\,(k \geq 3)$ independent files originate from the same distribution. It is a direct generalization of Wilcoxon signed-rank test in the case k of independent selection files $(k \geq 3)$.

Let's mark $n_1, n_2, \ldots, n_k$ the ranges of individual selection files. Let's pose, $n = n_1 + n_2 + \cdots + n_k$. Let's line all n elements into a non-decreasing sequence and let's assign its rank to each element. Let's mark T_i the sum of the elements ranks of the ith selection file $(i = 1, 2, \ldots, k)$. Since $T_1 + T_2 + \cdots + T_k = \frac{n(n+1)}{2}$ must hold, we can use this relationship to check the calculation of the values of the characteristics $T_i\,(i = 1, 2, \ldots, k)$. The test statistics is the statistics

$$K = \frac{12}{n\,(n+1)} \sum_{i=1}^{k} \frac{T_i^2}{n_i} - 3\,(n+1)$$

which has asymptotically the χ^2-distribution with $k-1$ degrees of freedom. We reject the null hypothesis H_0 at significance level α if $K \geq \chi^2\,(k-1)$, where $\chi^2\,(k-1)$ is the critical value of the χ^2-distribution with $k-1$ degrees of freedom. As the statistics K has an asymptotic χ^2-distribution, we can only use the above relationship if the selections have a large range $(n_i \geq 5, i = 1, 2, \ldots, k)$, and if $k \geq 4$. For some i is $n_i < 5$, or if $k = 3$, we compare the test criteria value K with the critical value K_α of the Kruskal–Wallis test. Critical values K_α are listed in the critical values table. The tested hypothesis H_0 is rejected at significance level α if $K \geq K_\alpha$.

If identical values occur in the obtained sequence data, that are assigned the average rank, it is necessary to divide the value of the testing criterion K by the so−called correction factor. Its value is calculated by the following formula:

$$f = 1 - \frac{\sum_{i=1}^{p} \left(t_i^3 - t_i\right)}{n^3 - n}$$

where p is the number of classes with the same rank, t_i the number of ranks in the $i-$th class. The testing statistics will then have the form

$$K_2 = \frac{K}{f}.$$

If we reject the tested hypothesis H_0 in favour of the alternative hypothesis H_1, which means that the selections do not come from the same distribution, a question remains unanswered: which selections differ statistically significantly from each other. In the analysis of variance, Duncan test, Tukey method, Scheffe method or Neményi test are used to answer this question. In the Kruskal–Wallis test, Tukey method is most frequently used to test contrasts, which we also briefly describe below.

In the Tukey method, we compare the $i-$th and the $j-$th file for each i, j, where $i, j = 1, 2, \ldots, k$ and $i \neq j$, according to the following procedure. For each pair of compared files, we calculate average ranks

$$\bar{T}_i = \frac{T_i}{n_i}, \quad \bar{T}_j = \frac{T_j}{n_j}.$$

The testing criterion of the null hypothesis H_0, that the distributions of the files i and j are identical, is the absolute value of the difference in their average rank

$$D = \left|\bar{T}_i - \bar{T}_j\right|.$$

The tested hypothesis H_0 is rejected at significance level α, if $D > C$, where

$$C = \sqrt{\chi_\alpha^2 \left(k-1\right) \frac{n\left(n+1\right)}{12} \left(\frac{1}{n_i} + \frac{1}{n_j}\right)}$$

$\chi_\alpha^2 \left(k-1\right)$ is the critical value of the χ^2-distribution with $k-1$ degrees of freedom, k is the number of compared files. In our case, we verified by the Kruskal–Wallis test whether the 12 sub-groups produced by the level of the factor "type of algorithm + type of function" statistically significantly differ in the observed feature "the numbers of evaluations". Therefore $k = 12$, while $n_1 = n_2 = \cdots = n_{12} = 100$, $n = n_1 + n_1 + \cdots + n_1 2 = 1200$ are measured numbers of evaluations. We implemented the Kruskal–Wallis test in program STATISTICA. After entering the input data in the computer output reports, we get the following results for the selected Kruskal–Wallis test: the testing criterion value H and the probability value p.

Dimension 3

We have used the Kruskal–Wallis test to test the null hypothesis H_0: the numbers of evaluations in the 12 sub-groups created according to the factor levels "algorithm type + function type" are identical as in the alternative hypothesis H_1: the numbers of evaluations in the 12 sub-groups created according to the factor levels indicated are not identical (or, at least at a level, they are different).

First, we calculated arithmetic averages and standard deviations of the assessment numbers (Table 1) and also presented it graphically in Figure 1 in each of the 12 sub-groups.

Evaluations count			
Groups	Means	N	Std. Dev.
1	22709,21	100	7098,061
2	2249,60	100	114,419
3	27863,82	100	3143,617
4	2703,36	100	191,001
5	5980,20	100	328,916
6	2711,70	100	117,233
7	10827,00	100	1327,715
8	16384,20	100	1092,761
9	192,89	100	25,877
10	234,79	100	14,931
11	272,89	100	28,379
12	376,47	100	69,821
All Grps.	7708,84	1200	9504,885

Table 1: Numbers of evaluations in each subgroup in dimension 3

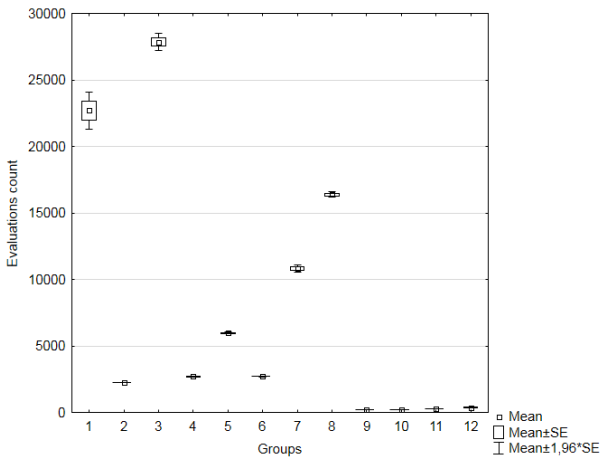


Figure 1: Numbers of evaluations (average values) in each subgroup in dimension 3

We have calculated the rank sums Table 2, the test criterion value $K = 1172.02$

and the value $p = 0.000$ by the Kruskal–Wallis test in dimension 3 for the assessment numbers. As the calculated probability value p is less than 0.01, we reject the null hypothesis at the significance level $\alpha = 0.01$, i.e. the difference between the 12 sub-groups in dimension 3 with respect to the observed feature of the "number of evaluations" is statistically significant.

Groups	Evaluation count	Sum of Ranks
1	100	105274,0
2	100	45221,0
3	100	111855,0
4	100	59602,0
5	100	75050,0
6	100	60327,0
7	100	85271,5
8	100	97799,5
9	100	6294,5
10	100	15348,0
11	100	24279,5
12	100	34278,0

Table 2: Kruskal–Wallis test results

The test confirmed that the individual sub-groups in dimension 3 differ statistically significantly from each other in relation to the assessment numbers. In the same way, as in dimension 2, we have been able to find out by multiple comparisons which groups are statistically significantly different from each other Table 3 in this case.

	Groups										
	2	3	4	5	6	7	8	9	10	11	12
1	0,00*	1,00	0,00*	0,00*	0,00*	0,00*	1,00	0,00*	0,00*	0,00*	0,00*
2		0,00*	0,22	0,00*	0,14	0,00*	0,00*	0,00*	0,00*	0,00*	1,00
3			0,00*	0,00*	0,00*	0,00*	0,27	0,00*	0,00*	0,00*	0,00*
4				0,11	1,00	0,00*	0,00*	0,00*	0,00*	0,00*	0,00*
5					0,18	1,00	0,00*	0,00*	0,00*	0,00*	0,00*
6						0,00*	0,00*	0,00*	0,00*	0,00*	0,00*
7							0,70	0,00*	0,00*	0,00*	0,00*
8								0,00*	0,00*	0,00*	0,00*
9									1,00	0,02*	0,00*
10										1,00	0,01*
11											1,00

Table 3: Results of Kruskal–Wallis multiple comparison test (p-values)

The Table 3 shows that there is a statistically significant difference in the numbers of evaluations in dimension 3 between sub-group 1 and sub-group 2, between sub-group 1 and sub-groups 4 to 7 and between the sub-group 1 and sub-groups 9 to 12 (probability value $p = 0.000$). This means that the measured assessment

numbers in sub-group 1 are statistically significantly different as measured in sub-group 2 and sub-groups 4 to 7 and 9 to 12 (or the assessment numbers between sub-groups 1st and 2nd and between the 1st sub-group and sub-groups 4–7 a 9–12 are significantly different respectively). In the same way, we can interpret all results in Table 3 marked with a *. We also illustrated the situation graphically (Figure 2).

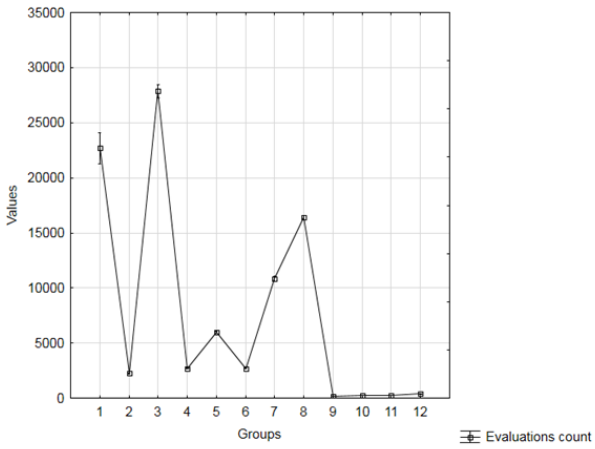


Figure 2: Numbers of evaluations (average values) in each subgroup in dimension 3

Dimension 4

As in previous dimensions, we also tested the statistical significance of differences in the number of evaluations in 12 sub-groups by the Kruskal–Wallis test in dimension 4. In each of the 12 sub-groups of dimension 4, we calculated arithmetic averages and standard deviations of the number of evaluations (Table 4) and we also presented the situation graphically in Figure 3.

We have calculated the rank sums (Table 5) and the test criterion value $K = 1180.46$ and the value $p = 0.000$ by the Kruskal–Wallis test. As the calculated probability value p is as well less than 0.01 in this case, we reject the null hypothesis at the significance level $\alpha = 0.01$, i.e. the difference between the 12 sub-groups in dimension 4 with respect to the observed feature "numbers of evaluations" is statistically significant. The Kruskal–Wallis test confirmed that the individual sub-groups in dimension 4 differ statistically significantly from each other in relation to the assessment numbers. Subsequently, we have identified by multiple comparisons (Table 6) which groups are statistically significantly different from each other. Table 6 shows that there is a statistically significant difference in the numbers of evaluations in dimension 4 between sub-group 1and sub-group 2. and sub-group 1 and sub-groups 4 to 6 and between the sub-group 1 and sub-groups

Evaluations count			
Groups	Means	N	Std. Dev.
1	34741,03	100	3119,75
2	4666,28	100	186,43
3	40000,00	100	0,00
4	5618,71	100	237,65
5	11676,80	100	518,89
6	4992,00	100	160,60
7	22800,00	100	2404,70
8	40000,00	100	0,00
9	298,57	100	57,56
10	379,91	100	44,35
11	439,19	100	61,26
12	609,18	100	129,52
All Grps	13851,81	1200	15432,29

Table 4: Numbers of evaluations in each subgroup in dimension 4

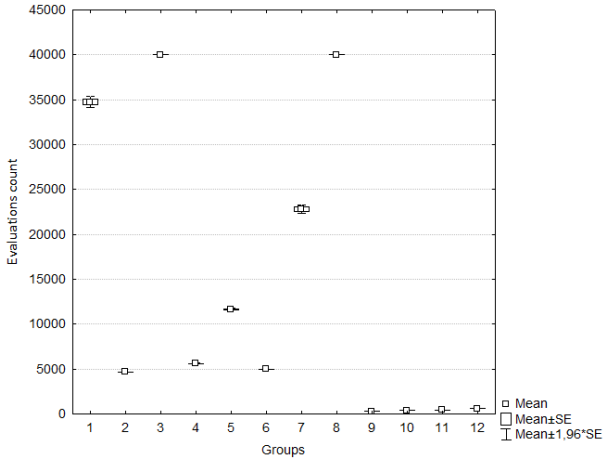


Figure 3: Numbers of evaluations (average values) in each subgroup in dimension 4

9 to 12 (probability value $p = 0.000$). This means that the measured assessment numbers in sub-group 1 in dimension 4 are statistically significantly different as measured in this dimension in sub-groups 2, 4 to 6 as well as 9 to 12 (or the assessment numbers between sub-groups 1 and sub-groups $4-7$ and $9-12$ in dimension 4 are significantly different). In the same way, we can interpret all results in Table 6 marked with a *. We also illustrated the situation graphically (Figure 4).

Groups	Evaluation count	Sum of Ranks
1	100	105000,0
2	100	49972,5
3	100	105000,0
4	100	65050,0
5	100	75050,0
6	100	50127,5
7	100	85200,0
8	100	105000,0
9	100	5884,5
10	100	16658,5
11	100	23892,5
12	100	33764,5

Table 5: Kruskal–Wallis test results

	Groups										
	2	3	4	5	6	7	8	9	10	11	12
1	0,00*	0,24	0,00*	0,00*	0,00*	1,00	0,24	0,00*	0,00*	0,00*	0,00*
2		0,00*	0,01*	0,00*	1,00	0,00*	0,00*	0,00*	0,00*	0,00*	0,83
3			0,00*	0,00*	0,00*	0,00*	1,00	0,00*	0,00*	0,00*	0,00*
4				1,00	1,00	0,00*	0,00*	0,00*	0,00*	0,00*	0,00*
5					0,00*	1,00	0,00*	0,00*	0,00*	0,00*	0,00*
6						0,00*	0,00*	0,00*	0,00*	0,00*	0,00*
7							0,00*	0,00*	0,00*	0,00*	0,00*
8								0,00*	0,00*	0,00*	0,00*
9									1,00	0,04*	0,00*
10										1,00	0,02*
11											1,00

Table 6: Results of Kruskal–Wallis multiple comparison test (p-values)

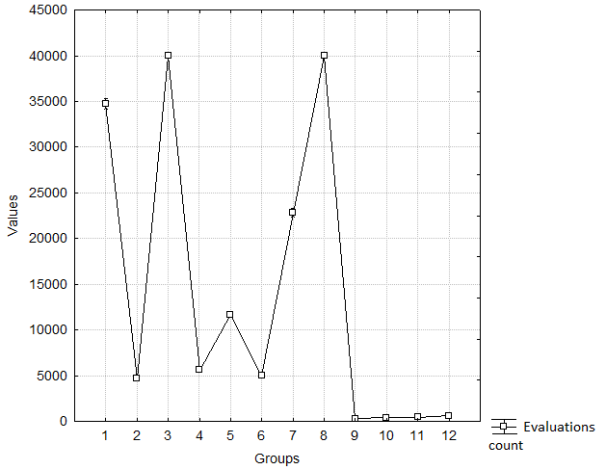


Figure 4: Numbers of evaluations (average values) in each subgroup in dimension 4

4. Conclusion

In conclusion, we can summarize based on the results of the experiment that the `fminsearch` function is the fastest algorithm and, on the other hand, the *Differential Evolution* algorithm is the slowest one. Only type 1 search is considered successful. Then the reliability of finding the global minimum can be characterized as the relative number of type 1 termination, that is $R = \frac{n_1}{n}$, where n_1 is the number of type 1 terminations and n the number of repetitions. The algorithms based on the experiment determine this reliability (in percent) of the global minimum finding (Table 7).

De Jong 1	Rosen	Ackley	Griewangk	Average
CRS				
100%	100%	53%	3%	64%
fminsearch				
100%	87%	0%	0%	47%
difevol				
100%	35%	99%	94%	82%

Table 7: Reliability of algorithms id dimensions 3 and 4

We can see a considerable difference in the reliability of the *Differential Evolution* algorithm (which is very slow) and algorithms *Controlled Random Search* and *fminsearch*. We can say that evolutionary operations mutation, cross-breeding, and selection are a major benefit of reliability for the algorithm. The difference in reliability of the algorithm *Controlled Random Search* and *Fminsearch* can also be considered significant given the number of times the global minimum search and the test functions used are repeated. Furthermore, based on the results of the test, we can say that finding a global minimum of the *First De Jong function* is simple and almost certain for any algorithm. Finding a global minimum for *Rosenbrock's saddle* is not easy just for an algorithm *Differential Evolution* that searches unreliably. From the above-mentioned reliabilities, it can be said that finding the global minimum of the *Ackley function* and *Griewangk's function* is difficult for the truly fast *Fminsearch* algorithm implemented in the Matlab environment, which produces great results for the *Second De Jong function* reliably and quickly. In general, the finding of the global minimum of the *Griewangk's function* is least likely in dimensions 3 and 4.

Based on the results of the experiment, we can conclude that by involving mathematical software to solve global optimization problems, a higher level of knowledge was achieved, a better understanding of various principles and algorithms, and, thus, students better mastered the issue. It is therefore effective and necessary to pay sufficient attention to these methods. Thanks to the use of computer techniques in the pedagogical process, everyone can draw into mathematics secrets of global optimization problems. On the basis of the results and theoretical starting points of the work, we have arrived at the following recommendations:

1. lead the students in solving mathematical application tasks in order to best

understand the theoretical starting points of the subject topic

2. use computer technology to increase students' activity and to provide a successful motivation to work
3. create suitable, modern and pregnant study material that will enhance the knowledge of students
4. within the cross-subject relations, extend the students' knowledge from the computer algebra systems
5. involve the use of computer algebra systems into maths teaching for achieving better results
6. effectively use the subject matter from another field within the framework of cross-subject relationships (such as mathematics and informatics).

References

[1] Y. Gao, K. Wang, C. Gao, Y. Shen, T. Li: *Application of Differential Evolution Algorithm Based on Mixed Penalty Function Screening Criterion in Imbalanced Data Integration Classification*, Mathematics 7.12 (2019), p. 1237,
doi: https://doi.org/10.3390/math7121237.

[2] M. R. Garey, D. S. Johnson: *Computers and intractability*, vol. 174, Freeman San Francisco, 1979.

[3] K. N. Kaipa, D. Ghose: *Glowworm swarm optimization: theory, algorithms, and applications*, vol. 698, Springer, 2017,
doi: https://doi.org/10.1007/978-3-319-51595-3.

[4] V. Kvasnicka, J. Pospíchal, P. Tino: *Evolutionary algorithms*, STU Bratislava (2000).

[5] D. Markechová, B. Stehlíková, A. Tirpáková: *Statistical Methods and their Applications. FPV UKF in Nitra, 534 p*, 2011.

[6] Mathworks: *Online documentation*, accessed 6th March, 2020, 2020,
url: https://www.mathworks.com/help/matlab/ref/fminsearch.html.

[7] Mathworks: *Online documentation*, accessed 16th March, 2020, 2020,
url: https://www.mathworks.com/help/matlab/ref/optimset.html.

[8] S. Míka: *Mathematical optimizatio*, Plzeň: ZCU Plzeň, 1997.

[9] J. A. Nelder: *A Simplex Method for Function Minimization*, Computer Journal 7.1 (1964), pp. 308–313.

[10] K. Price, R. M. Storn, J. A. Lampinen: *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media, 2006.

[11] W. L. Price: *A Controlled Random Search Procedure for Global Optimization*, Computer Journal 20.4 (1977), pp. 367–370.

[12] J. Tvrdík: *Evolutionary algorithms*, Ostrava: University of Ostrava, 2010.

Annales Mathematicae et Informaticae
***52** (2020) pp. 299–307*
DOI: https://doi.org/10.33039/ami.2020.12.001
URL: https://ami.uni-eszterhazy.hu

The education and development of mathematical space concept and space representation through fine arts

Rita Kiss-György

Doctoral School of Mathematical and Computational Sciences
University of Debrecen, Debrecen, Hungary
kgyrita@gmail.com

Submitted: October 20, 2020
Accepted: December 1, 2020
Published online: December 17, 2020

Abstract

The aim of this paper is to present potential connection points and interrelations between the development of spatial abilities and concepts and fine arts in high school (K9–12) education. The evaluation and comparison results of the assessment of spatial abilities and freehand drawing abilities show strong correlation, which fact further supports this potential educational interconnection.

Keywords: spatial abilities, teaching of fine arts, visualisation of space

MSC: 97D40, 00A35, 97D60

1. Introduction

Development of spatial abilities and concepts is of utmost importance in school education. Many papers studied the importance of spatial visualisation skills in various countries and in various school levels (for a good overview see e.g. [1, 6]). Spatial skills and spatial visualisation abilities have no unique definition – here we apply the following concept: by spatial ability we mean the ability of perception of two and three dimensional shapes, and the ability of application of the perception of these shapes and their relationships in spatial reasoning and solving spatial problems [10].

Spatial abilities or some of their aspects can be assessed through various tests, among which one can find international standards (such as Mental Cutting Test, Mental Rotation Test, see [3, 4, 12, 13]) and national versions as well. Most of them are specified for a certain aspect of school period/age. Several studies, including Hungarian surveys, observed gender differences in these abilities, which can and must be considered during the educational process [6–8].

It is also known that spatial ability is not the result of a person's ability to be born with it, but is the result of a long-term learning process([10, pp. 40–45]). The lack, or rather deficiency, of spatial ability yields problems only in a rather narrow part of the mathematics curriculum (spatial geometry), but in practical terms, in everyday life, correct spatial approach is extremely important and should be improved.

According to psychological studies, the development of visual thinking is completed relatively early. Therefore, neglecting spatial geometric problems and developing spatial attitudes in elementary school at the age of 10–14 can lead to an imminent disadvantage [9]. Literature data (see [10] and references therein) prove that the spatial concept of students can still be improved at the age of 12–16, but only to a certain extent. That is, in high school, spatial awareness is more difficult to develop, especially if a pupil comes from the primary school already possessing spatial deficiencies.

Therefore, we must grab all the tools, fields, subjects and lessons that can help us further develop spatial abilities. Beside mathematics lessons, which are natural milieu of development, other study fields can also support this development.

In this paper we discuss the correlation between art history and spatial visualization in order to show potential connection points in terms of development of space concept and space representation through fine arts. We intend to specify those periods or styles in the history of art, which can be assigned to well defined spatial visualisation problems. This is discussed in Section 2. In Section 3 we present outcomes of a test period, where spatial ability tests and drawing tasks have been tested and their correlation has been studied. This study supports our initial hypothesis, that there is a strong correlation between spatial abilities and (spatial) drawing abilities, which makes sense the above mentioned interdisciplinary approach.

2. Periods of art history and spatial visualisation

Problem solving on the plane of a drawing sheet is evidently not sufficient to develop spatial concept, to improve the spatial abilities. It is also important to act and construct in space for better understanding and more effective development. Kárpáti et al. found that the most effective developmental procedures are real-world operations: making sculptures and installations, modeling, object creation. ([5, p. 103]).

Where else can students find and study spatial constructions and their representation? Evidently in lessons on fine arts. Since people exists in space and time,

from the beginning of (art) history man is deeply interested in the visible and tactile space, and its visual expression, the planar representation of space. At first we briefly review the development of spatial approach and spatial representation, as well as its development in different eras of art from this point of view.

Period/style of art	Topics to study	Spatial visualisation
Prehistoric and Egyptian Art	Cave paintings (side view), Egyptian paintings (principle of largest surface view)	Similarity to orthogonal mappings
Art of the Roman Empire	wall-paintings (Pompei) and mosaics	Analogue of axonometric mapping
Medieval Art	Byzantine icons, codex illustrations	Reverse perspective
Late Medieval Paintings	Giotto frescos	Similarity to axonometric mapping
Renaissance	Development of correct perspective drawing through many artworks of various artists	Perspective mapping
Baroque, Classicist and Romantic Paintings	Illusionistic ceiling paintings, flourishing of perspectivity	Perspective mapping
Impressionism, Post-impressionism, Cubism	Monet, Cézanne, Gauguin, Picasso, Braque, emphasising the cubist approach (Braque, Picasso), where spatial relations and structure of objects have been studied, with manifold unified views and mappings in the same figure. Mondrian's geometric compositions	Different parallel projections
Op-art, Contemporary Graphic Art	Art of geometric compositions (e.g. Victor Vasarely and Maurits Escher)	Non-linear mappings

Table 1: Embedding spatial visualisation methods into teaching of history of fine arts

Here we only review and list a few important elements of potential areas of drawing lessons in the National Core Curriculum, where spatial ability can be

intentionally developed. For a more detailed overview see [10, pp. 135–148].

2.1. Primary school

Class 5: Making apparent drawings depicting spatial situations after a sight and based on imagination. View and derivation of objects from simple geometric shapes. Output: draws a fictional object based on memory, imagination. Knows how to display spatial situations. Class 6: Preparation of shape analysis structural drawings, sections, reductions. Projection representation. Output: Apply the familiar representation modes as appropriate. Class 7: Means of plastic expression (degree of spatial extent, directions, articulation, place in the environment). Structural, perspective representation of larger artificial and natural forms, Monge projection, (one-dimensional) editing of axonometry. Designing a utility object by making an appearance and projection drawing. Output: Its ability to abstraction is manifested in the emphasis on substance, in geometric simplification. He knows the basics of Monge projection and axonometric representation, he/she solves such an editing task with more or less independence. Class 8: Observation of a built spatial unit (building, street detail), experience-perspective representation. Edit a one- or two-way perspective image. Use of longitudinal and cross-sections to make illustrative diagrams. Reconstructions based on projection and axonometric drawings. Observation of the representation that creates the illusion of space, apparent shortenings, point of view. The perspective representation. Spatial representation modes, mixed perspective representation modes in different ages. Output: Has the necessary spatial basis for representation conventions, is able to edit simple projection, axonometric and perspective figures.

In grade 7–8. it is no accident that the curriculum related to the spatial approach swells in the classroom. These are the two school years when students who are no longer in secondary school must be taught all this. Another question is how much the number of hours shrunk to one drawing lesson per week is enough to master the diverse curriculum [10].

2.2. Secondary school

Prior to the new framework curriculum introduced from the 2014/15 school year, in secondary school (in vocational high school from the 2016/17 school year), it was possible to learn more about spatial representations and apply them to students as part of a drawing and visual culture class. There was an opportunity for freehand drawing and even occasionally even editing. The development of spatial representation can be traced throughout art history, which could be interestingly approached and skilfully emphasized in drawing lessons on art history topics. (As a drawing teacher, I also used it in secondary school for as long as I could – in the framework of the Drawing and Visual Culture subject, before the new framework curriculum introduced from the 2014/15 school year.)

The way in which space is depicted is a central issue in painting, as it also expresses the painter's relationship to reality, and the worldview of the ages can

also be read from it. (Thus, from prehistory to the renaissance, where painters had perfected the experiential perspective – which mathematicians only wrote much later – we could arrive at isms, modern endeavors, where they again began to ignore perspective.)

3. Experimental study of the relation between spatial abilities and art

In order to apply the above principles, we need to test that drawing skill and spatial ability are related. Therefore we organised an experimental study in this regard. 9th and 12th grade students participated in the survey. The hypotheses are that the above mentioned correlation exists, that final year students score higher (perform better) on the spatial test than 9th graders, meaning students increase their spatial performance with age, and there is a clear gender difference in these abilities. Continuing the previous studies in gender differences, and applying their methods ([6–8, 10]) we also studied this latter aspect. While previous studies mainly focused on university students, here we prove through experimental data that there is a difference in the spatial and performance of boys and girls already in high school. According to the results of an experiment([2]), the development of the spatial ability is significantly reflected by the students marks in maths and drawing. Our main focus is on the correlation between drawing abilities and spatial abilities.

We present the evaluation of the first three tasks of the written test series, which require mathematical and geometric knowledge in addition to the appropriate spatial approach, in terms of the two age groups (grades 9 and 12), and gender (girls and boys). We also examine the relationship between test scores and students' representation of space from memory. Drawing from memory is more difficult than drawing from sight, because from memory one draws what one knows about objects – based on schemes. In the case of a room and an interior, on the other hand, the task is to display the space, to place and draw the objects in space (abstracting from schemes, application of a representation system is necessary).

So this work raises the following questions:

- How did students of different ages pass the spatial test?
- How did students of different ages perform in the part of the test that also required knowledge of mathematics?
- Is there a significant relationship between students' performance in the test and the spatial quality of the room drawings made from memory?

4. Test results and correlations

The survey was conducted in two vocational grammar schools of a Transdanubian county seat, among "incoming" (9th grade), starting the vocational grammar school, and graduating (12th grade) students. Regarding the qualification in drawing, it should be mentioned that the graduates had a Visual Culture class in the 10th grade, 1 hour per week, and one class (21 people – 12 girls and 9 boys) had a Technical Representation class in the 9th grade, for one semester, 1 hour a week. Incoming 9th grade students are equipped with the drawing knowledge and skills learned in primary school (they will not have a Visual Culture, or Technical Representation, or any other "drawing" class during their high school years). The test is based on the tasks from the online database [11].

The tests were written at the beginning of the school year – in the second half of September. The drawing assignment was completed a few months later, in November and December. Due to this time delay, a few drawings may not have been drawn, or conversely, some drawings may not be assigned to test. The number of completed drawings became less than the number of test writers, so we take the number of written tests as a basis, and we have associated the drawings with these tests. For those tests no drawing was assigned, the category "none (drawing)" is created.

The test was written by: 295 people (241 girls and 54 boys); the drawing task was completed by 262 people (214 girls and 48 boys) in 12 classes. (no drawing: 27 girls, 6 boys) In grade 9, the test was written by 182 people (153 girls and 29 boys); the drawing task was completed by 163 people (137 girls and 26 boys) in 6 classes. (no drawing: 16 girls, 3 boys). In grade 12, 113 people (88 girls and 25 boys) completed the test; the drawing task was completed by 99 people (77 girls and 22 boys) in 6 classes (no drawing: 11 girls, 3 boys).

Tasks of the test include standard spatial problems about a cube. For example, we have marked some points on the edges of a cube (see Fig. 1).

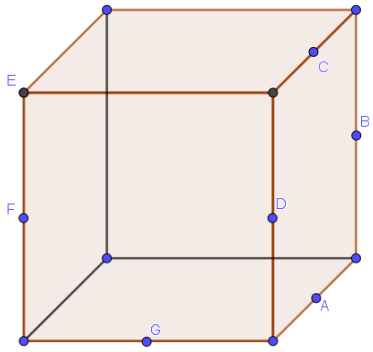


Figure 1: A cube with edge mid-points for spatial tasks

Students have to find a) four-point sets that are coplanar, and b) four-point sets that are not coplanar. A total of 10 points can be obtained in the task, from which 4 points could be obtained with four good solutions in part a). There are many good solutions in part b), where 1 point is awarded for a good answer, but we maximized the points that can be obtained in 6 points.

Another task is to determine what type of triangle are defined by the point triples BCD; BFD; FGC; ACE; GBE. A total of 10 points can be obtained in the task. Since triangles can be grouped according to their sides and angles (6th grade curriculum), the triangles listed by the points (their vertices) must be given in both ways.

In the spatial tasks the usual gender difference is observed, as one can see in the Table 2.

	9th grade average score	12th grade average score
overall	3.63	4.64
girls	3.48	4.28
boys	4.38	5.88

Table 2

We can also observe a significant improvement from 9th to 12th grade. More precisely, testing the hypothesis that this improvement is significant, the result of the t-probe is 0.0024 overall, 0.0291 for girls, and 0.0432 for boys, all less than 0.05.

What is even more interesting is our second hypothesis: that the spatial abilities measured by the test and the freehand drawing abilities measured by a drawing task (see below) are in correlation.

The drawing task had to be done freehand, that is without the use of a ruler, compass or other tool. The time allotted for the drawing task was 20 minutes – not much, but enough to sketch the main fixtures (optimally). Task text: "Draw your room from memory, as if you were standing in the doorway or sitting on your bed! You have 20 minutes for the task, you don't have to tone or color, line drawing is enough. (But whoever has the time and ambition, and of course the means, can also tone and color.) Write your name, year of birth and class on the back of the drawing! Use paper of size A4, preferably draw with a pencil (so that you can adjust or correct it)."

For the evaluation and classification of the drawings, we intended to define the categories in a similar way as the development of spatial representation can be traced in art history. It is surpirsing that axonometric drawings were not used, that is the expected, typically axonometric representation was not typical at all.

The established categories according to the spatial representation and the number (and proportion) of drawings corresponding to the categories by grades:

category	9th grade	12th grade
perspective (at least 90%)	4 (2.20%)	7 (6.20%)
perspective-axonometric (around 50%–50%)	22 (12.09%)	25 (22.12%)
space-like with planar elements	32 (17.58%)	25 (22.12%)
planar-like with a single view	43 (23.63%)	8 (7.08%)
mixed view with "unfolded" parts	44 (24.18%)	19 (16.81%)
simple layout (floor plan)	18 (9.90%)	15 (13.27%)
no drawing	19 (10.44%)	14 (12.39%)

Table 3

What we observed through calculating the significance is that the quality (category) of the freehand drawing and the scoring level of the test are in strong correlation in both grades. The correlation coefficient is 0.3994 for 9th grade students and 0.2937 for 12th grade students, both are well above the signifcance level of 98%, which is 0.2301.

5. Conclusion and future work

Our aim was to test a hypothesis in terms of spatial abilities of secondary school students, namely that the spatial ability, tested by standard tasks, and the freehand drawing ability, tested by a drawing task, are in strong correlation. This hypothesis has been proved, and, as a side effect, we also observed a gender difference, reported by many other publication, in spatial ability.

Our further goal is to prove the assumption that due to changes in curricula (visual culture instead of drawing and visual culture) and omission of subjects (cessation of drawing lessons), tests written in later school years show worse results, and spatial representation reaches a lower level. Further tests and drawings are needed to investigate this issue.

References

[1] B. Babály, A. Kárpáti: *The impact of creative construction tasks on visuospatial information processing and problem solving*, Acta Polytechnica Hungarica 13 (2016), pp. 159–180, doi: https://doi.org/10.12700/APH.13.7.2016.7.9.

[2] B. Csapó, Z. Varsányi: *Examining the development of drawing skills in high school students*, Development of Visual Abilities, ed. by A. Kárpáti, (in Hungarian), Nemzeti Tankönyvkiadó, 1995, pp. 659–693.

[3] R. Gorska, S. Sorby, C. Leopold: *Gender differences in visualization skills – an international perspective*, The Engineering Design Graphics Journal 62 (1998), pp. 9–18.

[4] Z. Juscaková, R. Gorska: *A pilot study of a new testing method for spatial abilities evaluation*, Journal for Geometry and Graphics 7 (2003), pp. 237–247.

[5] A. Kárpáti: *Measure the unmeasurable (Mérni a mérhetetlent)*, Iskolakultúra 13 (2003), pp. 95–106.

[6] R. Nagy-Kondor: *Importance of spatial visualization skills in Hungary and Turkey: Comparative Studies*, Annales Mathematicae et Informaticae 43 (2014), pp. 171–181.

[7] B. Németh, M. Hoffmann: *Gender differences in spatial visualization among engineering students*, Annales Mathematicae et Informaticae 33 (2006), pp. 169–174.

[8] B. Németh, C. Sörös, M. Hoffmann: *Typical mistakes in Mental Cutting Test and their consequences in gender differences*, Teaching Mathematics and Computer Science 5.2 (2007), pp. 385–392,
doi: https://doi.org/10.5485/TMCS.2007.0169.

[9] M. Niss: *Mathematical Competencies and the Learning of Mathematics: The Danish KOM Poject*, in: Gagatsis, A. and Papastavrides, S (eds): 3rd Mediteranean Conference on Mathematical Education Athen, Hellas 3–5 January 2003. Athens: Hellenic Mathematical Society, 2003, pp. 115–124.

[10] L. Séra, A. Kárpáti, J. Gulyás: *Spatial ability*, Comenius, Pécs (in Hungarian), 2002.

[11] G. Széplaki: *Can you see what I see?*, online database,
url: http://www.kooperativ.hu/matematika/4_modszertani%20segedletek/3_hattertan-felsotag-kozepiskola/terszemleletfejl_Te_is_latod_II.pdf.

[12] E. Tsutsumi, K. Shiina, A. Suzaki, et al.: *A Mental Cutting Test on female students using a stereographic system*, Journal for Geometry and Graphics 3 (1999), pp. 111–119.

[13] S. G. Vanderberg, A. R. Kuse: *Mental Rotations, a group test of three dimensional spatial visualization*, Perceptual and Motor Skills 47 (1978), pp. 599–604,
doi: https://doi.org/10.2466/pms.1978.47.2.599.

Annales Mathematicae et Informaticae
***52** (2020) pp. 309–319*
DOI: https://doi.org/10.33039/ami.2020.12.004
URL: https://ami.uni-eszterhazy.hu

Students' non-development in high school geometry*

Csaba Szabó†, Csilla Bereczky-Zámbó‡, Anna Muzsnay, Janka Szeibert

Eötvös Loránd University
csaba@cs.elte.hu
csilla95@gmail.com
annamuzsnay@gmail.com
szeibert.janka@gmail.com

Submitted: February 24, 2020
Accepted: December 9, 2020
Published online: December 17, 2020

Abstract

Earlier findings suggest that there is a gap between the knowledge of students entering the university and the expectations and prerequisites of the universities' curriculum. In this paper we investigate this gap in the Hungarian mathematics education. In particular we concentrate on the geometrical knowledge of the Hungarian high school students. For measuring their levels of geometrical understanding we used the Usiskin test for the framework of the Van Hieles. The test was filled in by 342 students from five different high schools. The results show that there is no improvement during the high school years, the average score of the Usiskin test is between 2.03 and 2.17 on all grades.

Keywords: van Hiele levels, math education, high shool, understanding geometry, development

MSC: AMS classification numbers

*This research was supported by the ÚNKP-19-2 New National Excellence Program of the Ministry for Innovation and Technology and by the ELTE Tehetséggondozási Tanács.

†The research of the first autoher was supported by the National Research, Development and Innovation Fund of Hungary, financed under the *FK* 124814 funding scheme.

‡The second author thanks the fund Mészáros Alapítvány for their support.

1. Introduction and problem statement

Earlier findings show that there is a gap between the knowledge of students entering the university and the expectations and prerequisites of the universities' curriculum [8]. These prerequisites are based on the National Core Curriculum (NCC) [20] for high schools. The admission process to the university is strongly based on a final exam that every student has to take at the end of secondary school from five subjects: mathematics is compulsory. They get a grade 1–5 for the exam and this grade is put into their transcripts. At the same time their score in percentages counts at the admission points to the university. Students can choose between two levels, medium and raised. The tasks in the final exam are mainly standard tasks and can be anticipated, even on the raised level, hence, can be practised. Thus a student can practice to the final exam without gaining deeper understanding. This final exam has a high impact on secondary education and on the transition from secondary to tertiary education in Hungary. Not only students are ranked and can get admission to universities based on their final exam-results, high schools are also ranked based on the average scores of their students on final exams [19]. Most students successfully pass their final exam in mathematics [7], however, it is a general observation that the knowledge of students entering university is deficient. This suggests that there is a gap between the final exams and the NCC [8].

We would like to argue that the strong external influence of the final exam distorts the original conceptions of the NCC. German universities struggled with a similar problem [3]: there was a big difference between the knowledge of the students entering the university and the knowledge required by the university. After several conciliations between the universities and each province's secondary schools, this problem seems to be being solved in Germany. In this paper we investigate this gap in the Hungarian math education.

Understandably, achieving good results on the final exam becomes a crucial aspect in high school mathematics education – sometimes even more important than aspects set up by the NCC. This implies that teachers will concentrate more on the topics and the tasks which occur in the final exam than on other topics that are in the curriculum. The entry system allows students to enter the university in the absence of the required knowledge [8]. We chose to examine this problem focusing on students' geometrical thinking due to the great proportion of geometry in the curriculum and the final exam. Geometry holds a central role in science, has several applications in everyday life, and in arts as well [5, 12]. Geometry itself is a separate high school subject in Greece, for example. In Hungary usually thirty percent of the final exam tasks are geometric flavoured. This is a significant proportion. Geometry is a substantial part of secondary mathematics education as well. It occupies approximately thirty-five percent of the high school mathematics material, similarly to its proportion in the final exam.

Hence it is natural to consider to investigate the geometrical understanding of Hungarian high school students. The aim of this research was to investigate students' Van Hiele levels to follow their development, especially to see whether or

not this development is parallel to the requirements of the NCC. In particular, we were interested if students from grade 12 have achieved level 4, the level of proofs. In our study we use the Usiskin test for the framework of the Van Hieles [15]. The test was filled in by 342 students from five different high schools. The results show that there is no improvement during the high school years, the average score of the Usiskin test is between 2.03 and 2.17 on all grades.

2. Description of geometrical understanding in the National Core Curriculum

There are several ways of thinking about geometry, there are different ways people think about it and there are several ways to structure geometry and how to teach geometry. The Van Hieles elaborated one possible way of structuring and describing people's understanding of geometry: focusing on understanding of geometrical shapes and structures, they distinguished five different levels of geometrical understanding. These levels are: visualization, analysation, abstraction, deduction and rigor (they are explained down below). According to the van Hiele theory, a student moves sequentially from the initial level (Visualization) to the highest level (Rigor). Students cannot achieve one level of thinking successfully without having passed through the previous levels [15].

The van Hieles' theory has been applied to clarify students' difficulties with the higher order cognitive processes. In order to succeed in high school geometry, higher order cognitive processes are indispensable. [20] According to the theory if students are not taught at the proper Van Hiele level, then they will face difficulties and they cannot understand geometry. This makes measuring students' Van Hiele level necessary. A possible validated tool for this measurement is the test elaborated by Usiskin in 1982.

2.1. Level 1: Visualization

At this initial stage, students recognize figures only by appearance and they usually think about space only as something that exists around them. Geometric concepts are viewed as undivided, whole entities rather than as having components or attributes. For example, geometric figures are recognized by their whole physical appearance, not by their parts or properties, so the properties of a figure are not detected. A person functioning at this level makes decisions based on perception, not reasoning. On the other hand, they can learn geometric vocabulary, identify specified shapes, reproduce a given figure. However, a person at this stage would not recognize the part of the figures, thus, they cannot identify the properties of these parts.

2.2. Level 2: Analysation

At this level an analysis of geometric concepts begins. For example, students can connect a collection of properties to figures, but at this point they see no relationship between these properties. Figures are recognized as having parts and are recognized by their parts. Usually they know a list of properties, but they cannot decide which properties are necessary and which are sufficient to describe the object. Interrelationships between figures are still not seen, and definitions are not yet understood at this level.

2.3. Level 3: Abstraction

At level 2 students perceive relationships between properties and between figures, they are able to establish the interrelationships of properties both within figures (e.g., in a quadrilateral, opposite angels being equal necessitates opposite sides being equal) and among figures (a rectangle is a parallelogram because it has all the properties of a parallelogram). So, at this level, class inclusion is understood, and definitions are meaningful. They are also able to give informal arguments to justify their reasoning. However, a student at this level does not understand the role and significance of formal deduction.

2.4. Level 4: Deduction

The 4th level is the level of deduction: students can construct smaller proofs (not just memorize them), understand the role of axioms, theorems, postulates and definitions, and recognize the meaning of necessary and sufficient conditions. The possibility of developing a proof in more than one way is also seen and distinctions between a statement and its converse can be made at this level.

2.5. Level 5: Rigor

This level is the most abstract of all. A person at this stage can think and construct proofs in different kind of geometric axiomatic systems. So, students at this level can understand the use of indirect proof and proof by contra-positive and can understand non-Euclidean systems.

The existence of Level 0 – the level of pre- recognition is also proposed [6]. Students at this level notice only a subset of the visual characteristics of a shape. As a result, they are not able to distinguish between certain figures. Progress from one level to the next is more dependent on educational experiences, than on age or maturation. Some experiences can facilitate progress within a level or to a higher level.There is some logic behind this kind of structuring. Although there might be other levelings, but these levels should be achieved by everybody independently of the manner in which they learned geometry.

The logic of this structure is also confirmed by the observation that the Van Hiele levels can be recognized in the Hungarian National Core Curriculum [20] step

by step. The following sentences and requirements connecting to different grades are from the NCC.

- Grade 1–4: "The creation, recognition and characteristics of triangles, squares, rectangles, polygons and circles."
- Grade 5–8 "Triangles and their categories. Quadrilaterals, special quadrilateral (trapezoids, parallelograms, kites, rhombuses). Polygons, regular polygons. The circle and its parts. Sets of points that meet given criteria."
- Grade 9–12.: "The classification of triangles and quadrilaterals. Altitudes, centroid, incircle and circumcircle of triangles. The incircle and circumcircle of regular polygons. Thales' theorem."
  "Remembering argumentation, refutations, deductions, trains of thought; applying them in new situations, remembering proof methods is important."
  "Generalization, concretization, finding examples and counterexamples (confirming general statements by deduction; proving, disproving: demonstrating errors by supplying a counterexample); declaring theorems and proving them (directly and indirectly) is also necessary."

The levels correspond to age groups: a 4^{th} grader (10 years old) has to reach level 1, a 6^{th} grader (12 years old) should reach level 2, an 8^{th} grader (14 years old) should be on at least level 3, and finally at grade 12 students (18 years old) have to reach level 4, which means they have to reach the level of deductions – students have to be able to construct smaller proofs, understand the role of axioms, theorems, postulates and definitions.

3. The survey

A survey of high-school students was held during the 2015/2016 academic year. Participants were 342 students from five different high-schools: one from Budapest and four from Miskolc. The schools were selected from a list that either had an agreement with our university or showed earlier a willingness to participate in research experiments. We omitted the schools with a special math program and schools founded by our university. Among the schools there was one music conservatory, and four standard high-schools such that three of them is considered as an average high-school, and one of them is in the top forty by the official ranking of the Hungarian Ministry of National Resources [19]. Four schools are founded by the government, one by the church. The data from Miskolc was collected by two colleagues from Miskolc: Csenge Edőcsény and Ákos Győry. There was 62 students from the music conservatory and the other 280 students followed the normal curriculum. Also there were 32 pupils who belonged to the Arany János Tehetséggondozó Program (AJTP) which is a talent care program for pupils coming from socially handicapped families, mostly from small villages. Out of the 280 students there were 91 from grade 9, 103 from grade 10, 27 from grade 11, and 59

from grade 12. Among the 62 music conservatory students 18 9^{th} graders, 17 10^{th} graders, 15 11^{th} graders, and 59 12^{th} graders participated in our research.

The measuring of the levels was carried out by means of the Usiskin-test [15], which is a 25 item multiple-choice test with 5 foils per item. The test contains five questions per level and to fulfill correctly a level one has to answer correctly to at least 3 or 4 questions – depending on which scoring system is used – out of the five questions. We distinguish two kinds of scoring system: we called them "strong" version and a "weak" version. In the "weak" version one has to answer correctly to at least 3 questions from the five to fulfill correctly a level, while in the "strong" version at least 4 good answer is needed. To reach a level one has to fulfill correctly all the previous levels, too. That means if a person completed correctly level 1, 2, 3, 5 but not level 4, then this person is on level 3 according to the test. In general if a person met the criteria of passing each level up to and including level n, but not level $n+1$, then the person is assigned to level n. This test was used in more than forty countries [1, 2, 4, 9–11, 13, 14, 16–18, 22], and this test is tested and used continuously from 1982.

There are 35 minutes for the test independently of age and grade. In our experiment the students had to complete the test either on paper or online decided by the teacher of the class.

4. The results

The following table shows the results of the high-school students. On the table A, B, C, D, E letters denote the schools. The abbreviation n.o.p. denote the number of participants. By strong version we mean that the text filler has to answer four questions correctly out of the five to fill correctly a certain level and by weak version we mean that the text filler has to answer only three questions correctly out of the five.

	A	**B**	**C**	**D**	**E**	**total**
mean	1,42	1,26	1,40	1,00	0,67	1,21
dev.	1,35	1,38	1,16	1,05	0,97	1,23
n.o.p.	24	27	30	10	18	109

Table 1: Grade 9 – strong version

	A	**B**	**C**	**D**	**E**	**total**
mean	2,29	2,22	2,13	2,10	1,17	2,03
dev.	0,95	1,69	1,22	1,20	1,34	1,35
n.o.p.	24	27	30	10	18	109

Table 2: Grade 9 – weak version

	A	**B**	**C**	**D**	**E**	**total**
mean	1,18	1,13	1,54	1,80	1,00	1,31
dev.	1,26	1,18	1,14	1,32	1,12	1,19
n.o.p.	22	32	39	10	17	120

Table 3: Grade 10 – strong version

	A	**B**	**C**	**D**	**E**	**total**
mean	1,18	2,16	2,21	2,40	1,59	2,05
dev.	1,10	1,30	1,10	0,97	1,12	1,16
n.o.p.	22	32	39	10	17	120

Table 4: Grade 10 – weak version

	A	**B**	**C**	**D**	**E**	**total**
mean	2,86	-	1,38	1,14	0,47	1,26
dev.	1,21	-	1,39	0,90	0,74	1,33
n.o.p.	7	0	13	7	15	42

Table 5: Grade 11 – strong version

	A	**B**	**C**	**D**	**E**	**total**
mean	4,00	-	2,23	2,43	1,13	2,17
dev.	1,29	-	1,48	1,27	0,92	1,54
n.o.p.	7	0	13	7	15	42

Table 6: Grade 11 – weak version

	A	B	C	D	E	total
mean	0,70	1,75	2,11	0,87	1,00	1,14
dev.	1,02	1,36	1,05	1,19	1,04	1,21
n.o.p.	23	12	9	15	12	71

Table 7: Grade 12 – strong version

	A	B	C	D	E	total
mean	2,47	2,75	2,89	1,00	1,17	2,04
dev.	1,04	1,42	0,60	1,95	1,11	1,34
n.o.p.	23	12	9	15	12	71

Table 8: Grade 12 – weak version

	n.o.p.	mean (strong version)	mean (weak version)
grade 9	109	1,21	2,03
grade 10	120	1,31	2,05
grade 11	42	1,26	2,17
grade 12	71	1,14	2,04

Table 9: cumulative results

Although the results are from different schools, the performances of the schools are similar and based on these results we can estimate the Van Hiele levels of students attending to other schools in the country. Based on this estimation most of the Hungarian high-school students are on the level of a primary school student in geometry. This raises the question how students can be successful on the final exam. This question requires a deeper investigation, a part of it could be the analysis of the geometry problems and their sample solutions in the final test. Reading through the past fifteen years' final exams it is reasonable to question the amount of geometrical proving skills needed to solve the tasks. Typical geometry flavoured tasks are the following ones [21]:

Problem 4.1 (A tipical final exam task – a "less difficult" one)**.** *The ending point of a straight line that closes at* $6.5°$ *to the horizontal is* 124 *meters higher than its starting point. How long is the road? Justify your answer!*

Problem 4.2 (A tipical final exam task – a "difficult" one)**.** *A motion sensor is on the top of a* 4 *m high vertical pole. The lamp connected to the sensor illuminates vertically downwards at a rotational cone of* $140°$.

a) *Make a sketch with the details.*
b) *How far is the farthest illuminated point from the lamp?*
c) *Does the sensor lamp illuminate an object on the ground* $15\,m$ *from the bottom of the pole?*
d) *There are hooks on the pole, one per meter, in order to hang the motion detector lamp. Which hook should we use in order that the lamp illuminates at most* $100\,m^2$ *on the horizontal ground? (Numbering of the hooks starts from the bottom of the pole.)*

The first task is from May 2003 and the second one was from May 2006. On the first task students could reach 3 points, while with the second one they could get 17 out of 115 points on the exam. In the latter case pupils get only 2 points for noticing that the flat section is a triangle and they get 15 points for the calculations. So a possible answer could be obtained answering the quesiton: Does the final exam require the 4^{th} van Hiele level at all?

5. Discussion

The tables show the results of the van-Hiele tests from five Hungarian high-schools. The sample naturally does not cover the whole country. It involves three schools that are average in the Hungarian rankings, one vocational music school and a non-elite, but fairly top ranked school. With these limitations we made the following observations. It can be read from the tables, that even considering the weaker criteria, in each grade the average performance of the students is around level 2 – which should be the level of a 6^{th} grader. There is no development in the level of understanding geometry from grade 9 to grade 12. Most of the students do not even reach the 3^{rd} level which should be the level of an 8^{th} grader according to the NCC. In the weak version 40.37% of the students reached the 3^{rd} Van Hiele level at grade 9, which is the level that the NCC suggests. In grade 12 45.07% of the students reached the 3^{rd} Van Hiele level and only 8.45% of these students reached the 4^{th} Van Hiele level, which is the level that a 12 grader should reach according to the NCC. Although there exist students who reach the required level, the geometrical thinking of the vast majority did not improve during their high-school years. Still, both groups passed the final exam with relatively good results. One can see that mathematical education in Hungary is in a controversial situation. On the one hand, students achieve a certain, well defined level, namely, they perform well on final exam. On the other hand, they do not reach the level of geometric understanding required by the NCC.

To look for the possible reasons it is worth examining the geometry content of the final exam. By its nature the final exam is predictable and a has a high impact on the curriculum and teacher activities in class.

The gap between the final exam's requirement and the NCC's requirement indicates further problems for higher education. Since the university education is built on the National Core Curriculum, not on the final exam, this gap results

in a big difference between the knowledge of the students entering the university and the knowledge required by the universities. We see three kind of solution to this problem. The first one is to change the requirement system of the NCC and make it consistent with the requirement system of the final exam. It follows that universities would adopt to the new NCC and the standard of higher education would fall. The second solution is to change the entry system of the universities concerned, and make it obligatory to take the mathematics final exam on advanced level or reintroduce an entry exam for the universities. The third solution we imagine is to introduce bridging courses at the universities specialized to different topics and levels depending on their needs. We think that the latter solution would not work. It would be difficult and nearly impossible to bring the students from such a low level to the level where they understand the need for proof and where they can also construct easier ones in a half a year course.

References

[1] A. H. Abdullah, E. Zakaria: *Enhancing Students' Level of Geometric Thinking through Van Hiele's Phase-based Learning*, Indian Journal of Science and Technology 6.5 (2013), pp. 4432–4446.

[2] R. Astuti, D. Suryadi, T. Turmudi: *Analysis on geometry skills of junior high school students on the concept congruence based on Van Hiele's geometric thinking level*, Journal of Physics Conference Series (2018), pp. 1–5, doi: https://doi.org/10.1088/1742-6596/1132/1/012036.

[3] I. Braun, J. E. Schröder: *Cooperation schule hochschule*, Baden-Württembergs: Hochschulen Baden-Württembergs, 2014.

[4] W. F. Burger, J. M. Shaughnessy: *Characterizing the van Hiele Levels of Development in Geometry*, Journal for Research in Mathematics Education 17.1 (1986), pp. 31–48, doi: https://doi.org/10.2307/749317.

[5] P. Bursill-Hall: *Why do we study geometry? Answers through the ages*, Cambridge: Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, 2002.

[6] D. Clements, M. T. Battista: *Geometry and spatial reasoning*, in: Jan. 1992, pp. 420–464.

[7] C. Csapodi, L. Koncz: *The efficiency of written final exam questions in mathematics based on voluntary data reports, 2012–2015*, Teaching Mathematics and Computer Science 14.1 (2016), pp. 63–81, doi: https://doi.org/10.5485/TMCS.2016.0417.

[8] É. Erdélyi, A. Dukán, C. Szabó: *The transition problem in Hungary: curricular approach*, Teaching Mathematics and Computer Science 17.1 (2019), pp. 1–16, doi: https://doi.org/10.5485/TMCS.2019.0454.

[9] T. Erdogan, S. Durmus: *The effect of the instruction based on Van Hiele model on the geometrical thinking levels of preservice elementary school teachers*, Procedia - Social and Behavioral Sciences 1.1 (2009), pp. 154–159, doi: https://doi.org/10.1016/j.sbspro.2009.01.029.

[10] K. Jones: *Issues in the teaching and learning of geometry*, in: Aspects of Teaching Secondary Mathematics: perspectives on practice. London, GB: Routledge, 2002, pp. 121–139, doi: https://doi.org/10.4324/9780203165874.

[11] G. Kospentaris, P. Spyrou: *Assessing the development of geometrical thinking from the visual towards the analytic-descriptive level*, Annales de didactique et de sciences cognitives 13.5 (2008), pp. 133–157.

[12] C. Mammana, V. V.: *Perspectives on the Teaching of Geometry for the 21st Century*, Dordrecht: Springer, 1998,
doi: https://doi.org/10.1007/978-94-011-5226-6.

[13] S. Senk: *Van Hiele levels and achievement in writing geometry proofs*, Journal for Research in Mathematics Education 20.3 (1989), pp. 309–321,
doi: https://doi.org/10.2307/749519.

[14] A. Ural: *Investigating 11th Grade Students' Van-Hiele Level 2 Geometrical Thinking*, Journal Of Humanities And Social Science 21.12 (2016), pp. 13–19,
doi: https://doi.org/10.9790/0837-2112061319.

[15] Z. Usiskin: *Van Hiele Levels and Achievement in Secondary School Geometry. CDASSG Project.* Chicago: Chicago Univ, IL., 1982.

[16] M. D. de Villers: *Some reflections on the Van Hiele theory*, in: June 2010.

[17] M. D. de Villers: *The role and function of a hierarchical classification of the quadrilaterals*, For the Learning of Mathematics 14.1 (1994), pp. 11–18.

[18] I. Vojkuvkova, J. Haviger: *The van Hiele Levels at Czech Secondary Schools*, Procedia - Social and Behavioral Sciences 171 (2015), pp. 912–918,
doi: https://doi.org/10.1016/j.sbspro.2015.01.209.

[19] www.eduline.hu: *A száz legjobb vidéki gimnázium és szakközépiskola - itt a teljes lista*, Eduline (2014).

[20] www.ofi.hu: *The Hungarian National Core Curriculum*, Teaching Mathematics and Computer Science (2012).

[21] www.oktatas.hu: *Központi írásbeli feladatsorok, javítási útmutatók*, www.oktatas.hu.

[22] I. Zachos: *Register of Educational Research in the United Kingdom – Problem Solving in Euclidean Geometry in Greek Schools*, in: vol. 10, 0615, London and New York: Routledge, 1995.