



Computer assisted solution of systems of two variable linear functional equations

GERGŐ GYULA BORUS AND ATTILA GILÁNYI 

Dedicated to Professor János Aczél on the occasion of his 95th birthday.

Abstract. In the present paper, a general class of linear functional equations is considered and a computer program is described, which determines the exact solutions of systems of equations belonging to this class.

Mathematics Subject Classification. Primary 39B52, 39B72; Secondary 39A70.

Keywords. Linear functional equations, Systems of functional equations, Polynomials, Monomials, Computer assisted methods, Computational mathematics.

1. Introduction

In this paper, linear functional equations of the type

$$\sum_{i=0}^{n+1} f_i(p_i x + q_i y) = 0 \quad (x, y \in X) \quad (1)$$

are considered, where n is a positive integer, p_0, \dots, p_{n+1} and q_0, \dots, q_{n+1} are rational numbers, X, Y are linear spaces over the rationals and $f_0, \dots, f_{n+1} : X \rightarrow Y$ are unknown functions. Our aim is to present a computer program developed in the computer algebra system Maple,¹ which determines the solutions of functional equations and also solutions of systems of functional equations belonging to the class above.

This research has been supported by the Hungarian Scientific Research Fund (OTKA) Grant K-111651. This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was co-financed by the Hungarian Government and the European Social Fund.

¹Maple is a trademark of Waterloo Maple Inc.

It is easy to see that the class above contains several well-known functional equations, like the Cauchy, the square-norm (or quadratic, or parallelogram or Jordan-von Neumann), the Jensen and the Pexider equations as well as the so called polynomial and monomial equations as a special case (cf., among others, the books [1–3, 26, 27, 30]). Early studies of polynomial equations were made by Maurice Fréchet ([14, 15]), while the class above was already considered by W. Harold Wilson ([32]) in the first decades of the previous century. For recent investigations, we refer to the papers [20, 23–25, 31] and the references therein. Solutions of (1), in the general case when X and Y are certain Abelian groups and the products $p_i x$ and $q_i y$ in the arguments of the unknown functions are replaced by homomorphisms of X , were determined by László Székelyhidi in [29] (cf., also, [30]).

In the 1980s, János Aczél raised the question, whether it is possible to develop a computer program which determines the solutions of functional equations belonging to class (1) based on Székelyhidi's method. With the application of such a program, the arduous (in several cases 'hopeless') calculations needed if someone applies the known theoretical results for solving equations of the type above can be avoided (more precisely, can be done by a computer).

The first programs dealing with the solution of the equations considered, were developed more than 20 years ago and were described in the Thesis [18] and in the paper [19]. Their improvement presented here is much more effective than those. Additionally, the new version is capable of treating not only single functional equations but also systems of them. Its structure fits the architecture of modern ('module-based') Maple packages. Due to this fact, the entire program is more transparent and, besides several further advantages, it can serve as the first part of a larger class of programs handling functional equations and inequalities. The next part of such a system could be a properly modified version of Attila Háyzy's program also written in the computer algebra system Maple and described in [22]. Concerning further studies of functional equations, inequalities and related topics using computer algebra systems, we refer to the papers [4–6, 13, 16, 17] and [28].

We point out that our program operates with symbolic calculations provided by Maple. This means that it does not contain any numerical or approximate methods and it yields the 'exact' solutions of the equations considered. Since it determines solutions of equations, using the concept of 'mathability' introduced in [8], it 'increases the level of mathability' (or using another terminology, it 'improves the mathability') of the underlying system. (For further investigations connected to mathability we refer to the book [7] and to the papers [11, 12]).

We note that a previous version of the program was presented during the 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom) in 2013 (cf. [10]) and also a related Demo Presentation was performed at the same conference (cf. [9]).

In the first part of the paper, we present the theoretical basis of our program, while, in Sect. 3, we describe the program itself.

2. Theoretical background

In this section, we present the theoretical background of our program. First we give some basic definitions.

Let X and Y be linear spaces, and let $f : X \rightarrow Y$ be a function. Let, furthermore,

$$\Delta_y^1 f(x) = f(x + y) - f(x) \quad (x, y \in X)$$

and, for a positive integer n ,

$$\Delta_y^{n+1} f(x) = \Delta_y^1 \Delta_y^n f(x) \quad (x, y \in X).$$

If n is a non-negative integer, a function $f : X \rightarrow Y$ is said to be a *polynomial function of degree n* (with another terminology a *polynomial function of degree at most n*) if it fulfils

$$\Delta_y^{n+1} f(x) = 0 \quad (x, y \in X). \tag{2}$$

It is well-known that, in the class of real functions (i.e., if $X = Y = \mathbb{R}$), functions of the form

$$f(x) = a_n x^n + \dots + a_0 \quad (x \in \mathbb{R}) \tag{3}$$

with (fixed) real coefficients a_0, \dots, a_n solve (2). It is also known, that, in the case when n is a positive integer, functional equation (2) has solutions which are different from (3). Thus, polynomial functions can be considered as generalized algebraic polynomials.

If n is a positive integer, a function $f : X \rightarrow Y$ is called a *monomial function of degree n* if it satisfies

$$\Delta_y^n f(x) - n!f(y) = 0 \quad (x, y \in X). \tag{4}$$

Similarly to the situation above, in the class of real functions, algebraic monomials, i. e. functions of the form

$$f(x) = cx^n \quad (x \in \mathbb{R}) \tag{5}$$

with a (fixed) real constant c are solutions of functional equation (4). Here it is also valid that, in the case when $n > 1$, there exist functions which satisfy (4) but are not of the form given in (5), therefore, monomial functions are generalized algebraic monomials. (For real polynomial and monomial functions which are different from (3) and (5), respectively, we refer to Remark 1 below.)

There is an important connection between polynomial and monomial functions. If $f_k : X \rightarrow Y$, ($k = 0, \dots, n$) are monomial functions of degree k , then their sum is a polynomial function of degree n . On the other hand, if $f : X \rightarrow Y$ is a polynomial function of degree n , then there exist monomial

functions $f_k : X \rightarrow Y$ of degree k , ($k = 0, \dots, n$) such that their sum is f . This means that (analogously to algebraic monomials and algebraic polynomials) monomial functions serve as ‘building blocks’ for polynomial functions (cf., e.g. [26, 27] and, in a general setting, [30]).

A similar property is valid for the solutions of general linear functional equations considered in this paper, too. According to L. Székelyhidi’s results ([29], cf., also, [30]), under some natural conditions, the solutions of equations of class (1) are also sums of certain monomial functions. Now, we formulate his corresponding theorems, which serve as a background for our program. In these statements and in the following parts of the paper, we will use the convention $0^0 = 1$.

Theorem 1. *Let X and Y be linear spaces over the rationals, let $p_0, \dots, p_{n+1}, q_0, \dots, q_{n+1}$ be rational numbers and assume that*

$$M_k^{(i)} : X \rightarrow Y \quad (i = 0, \dots, n + 1, k = 0, \dots, n) \tag{6}$$

are monomial functions of degree k . The functions $f_0, \dots, f_{n+1} : X \rightarrow Y$,

$$f_i(x) = \sum_{k=0}^n M_k^{(i)}(x) \quad (x \in X, i = 0, \dots, n + 1) \tag{7}$$

solve functional equation (1) if and only if the monomials given in (6) fulfil

$$\sum_{i=0}^{n+1} p_i^j q_i^{k-j} M_k^{(i)}(x) = 0 \quad (x \in X, k = 0, \dots, n, j = 0, \dots, k). \tag{8}$$

Theorem 2. *Let X and Y be linear spaces over the rationals and let $p_0, \dots, p_{n+1}, q_0, \dots, q_{n+1}$ be rational numbers such that*

$$p_i q_j \neq p_j q_i \quad (i, j = 0, \dots, n + 1, i \neq j). \tag{9}$$

The functions $f_0, \dots, f_{n+1} : X \rightarrow Y$ solve functional equation (1) if and only if they are of the form

$$f_i(x) = \sum_{k=0}^n M_k^{(i)}(x) \quad (x \in X, i = 0, \dots, n + 1) \tag{10}$$

where

$$M_k^{(i)} : X \rightarrow Y, \quad (i = 0, \dots, n + 1, k = 0, \dots, n)$$

are monomial functions of degree k satisfying the equations

$$\sum_{i=0}^{n+1} p_i^j q_i^{k-j} M_k^{(i)}(x) = 0 \quad (x \in X, k = 0, \dots, n, j = 0, \dots, k). \tag{11}$$

Remark 1. According to a celebrated result of Georg Hamel presented in [21], there exist monomial functions $f : \mathbb{R} \rightarrow \mathbb{R}$ of degree 1, such that the graph of f , i.e., the set

$$\{(x, y) \in \mathbb{R}^2 \mid x \in \mathbb{R}, y = f(x)\}$$

is dense in \mathbb{R}^2 . A similar property is valid for monomial functions of higher order, too (cf., e.g., [26,27]). As a consequence of the Theorems above, we obtain that equations belonging to class (1) can also have such ‘irregular’ solutions.

Remark 2. We note that, in the paper [29], L. Székelyhidi investigated a more general situation (cf., also, [30]). He considered the class of functional equations

$$f(x_0) + \sum_{i=1}^{n+1} f(\varphi_i(x) + \psi_i(y)) = 0 \quad (x, y \in G), \tag{12}$$

where G is a divisible, S is a torsion free Abelian group, $\varphi_i, \psi_i : G \rightarrow G$, ($i = 1, \dots, n + 1$), are homomorphisms and $f_i : G \rightarrow S$, ($i = 0, \dots, n + 1$), are unknown functions. He proved that, in the case when φ_i, ψ_i , ($i = 0, \dots, n + 1$), are homomorphisms of G onto itself with the property

$$\text{Rg}(\psi_i \circ \psi_j^{-1} - \varphi_i \circ \varphi_j^{-1}) = G \quad (i = 1, \dots, n + 1, i \neq j) \tag{13}$$

then the solutions $f_i : G \rightarrow S$, ($i = 0, \dots, n + 1$) of (12) are of the form

$$f_i = \sum_{k=0}^n D(A_k^{(i)}) \quad (i = 0, \dots, n + 1) \tag{14}$$

where $D(A_k^{(i)})$ denote the diagonals

$$D(A_k^{(i)})(x) = A_k^{(i)}(x, \dots, x) \quad (x \in G, i = 0, \dots, n + 1, k = 0, \dots, n)$$

of the k -additive symmetric functions $A_k^{(i)} : G^k \rightarrow S$, ($i = 0, \dots, n + 1, k = 0, \dots, n$), which satisfy the equations

$$A_k^{(0)}(x \dots, x, 0, \dots, 0) + \sum_{i=0}^{n+1} A_k^{(i)}(\varphi_i(x), \dots, \varphi_i(x), \psi_i(y), \dots, \psi_i(y)) = 0 \tag{15}$$

for each $x, y \in G$, $j = 0, \dots, n$ and $k = j, \dots, n$. He also showed that, if condition (13) does not hold, then functions $f_i : G \rightarrow S$, ($i = 0, \dots, n + 1$) of the form (14) solve (12) if and only if the k -additive symmetric functions $A_k^{(i)} : G^k \rightarrow S$ fulfil the equations given in (15).

3. Description of the program

In this section, we describe the computer program which determines solutions of systems of functional equations belonging to class (1). The program was developed in the computer algebra system Maple. (It can be used in all versions of Maple between Maple 16 and Maple 2020. A copy of the program can be requested from the second author of the paper.) It uses formal calculations and provides the exact solutions of the equations considered. Its algorithm is based on the results presented in Sect. 2.

The name of the program is `lfesolve` and it is contained in the Maple package `FunctionalEquations`. Like Maple programs, it can be accessed by using either the long or the short form of the command name, i.e., one of the (long) forms

```
FunctionalEquations[lfesolve](arguments);
```

or

```
FunctionalEquations:-lfesolve(arguments);
```

or if the command

```
with(FunctionalEquations);
```

has already been applied in the Maple session considered, then simply

```
lfesolve(arguments);.
```

During the presentation of the program, we will use the last version only.

To use the program, we have to give at least two input parameters: the system of equations to be solved as a (Maple) list \mathbf{e} and the list \mathbf{f} of the unknown functions contained in the system. In addition, we may use some optional parameters. Thus, the input of the program should be of the form

```
lfesolve(e,f,options);.
```

Obviously, the equations in \mathbf{e} have to belong to class (1). In order for the program to work properly, the variables of the unknown functions of the system should be denoted by ‘ x ’ and ‘ y ’.

If any of the equations is given in a form with 0 on the right hand side—similarly to other `solve` functions in Maple—it is enough to give its left hand side in the input. If any of \mathbf{e} or \mathbf{f} contains one element only, then, analogously to other appearances of lists in Maple, it can be given without the brackets `[]`. For the users’ convenience, if some functions contained in the system \mathbf{e} are not given in \mathbf{f} , then the program sends a warning message of this fact with the names of the missing functions, adds these functions automatically to the list \mathbf{f} and continues the solution of the problem. (This ‘convenience service’ cannot cause any problems, since the use of parameter functions is not allowed in the system of equations considered.) On the other hand, if some functions appear

in \mathbf{f} which are not contained in the system \mathbf{e} , then the program—without any warning or error message—simply omits them from \mathbf{f} and solves the system without them.

As the output of the program, provided that the input is given in a correct form, we obtain the solutions of the system of functional equations given in parameter \mathbf{e} . The solutions are presented, according to Theorems 1 and 2, as linear combinations of monomial functions. The monomial functions $M_k^{(i)}$, ($i = 0, \dots, n + 1, k = 0, \dots, n$) contained in the Theorems above appear in the output in the form

$$M(i, k) = M_k^{(i)} \quad (i = 0, \dots, n + 1, k = 0, \dots, n).$$

The program gives the solution functions in a set.

Let us consider, as an example, the well-known square-norm functional equation

$$f(x + y) + f(x - y) = 2f(x) + 2f(y) \quad (x, y \in X) \tag{16}$$

(cf., e.g., the books [1,2] and the references therein) and let now and in the following part of the paper X and Y denote linear spaces over the rationals. It is an easy consequence of Theorem 2 that a function $f : X \rightarrow Y$ solves this functional equation if and only if

$$f(x) = M_2(x) \quad (x \in X),$$

where $M_2 : X \rightarrow Y$ is a monomial function of degree 2.

If we would like to solve this equation with our program, we have to use the input

```
> lfsolve(f(x+y)+f(x-y)=2*f(x)+2*f(y), f); .
```

We obtain the output

```
{f = M(0, 2)}.
```

If property (9) is not valid in any of the equations in \mathbf{e} , then we receive a warning message.

Let us consider, for example, the simple functional equation

$$f(x) - f(y) = g(x) - g(y) \quad (x, y \in X), \tag{17}$$

where $f : X \rightarrow Y$ and $g : X \rightarrow Y$ are unknown functions.

It is easy to see that this equation belongs to class (1) but does not satisfy condition (9). In such situations, Theorem 2 cannot be applied and, by Theorem 1, we can determine those solutions of the equation which can be written as a sum of certain monomial functions. Thus, assuming that

$$M_k^{(i)} : X \rightarrow Y \quad (i = 0, 1, k = 0, 1, 2)$$

are monomial functions of degree k , the functions $f, g : X \rightarrow Y$

$$\begin{aligned} f(x) &= M_2^0(x) + M_1^0(x) + M_0^0(x) \quad (x \in X) \\ g(x) &= M_2^1(x) + M_1^1(x) + M_0^1(x) \quad (x \in X), \end{aligned}$$

solve functional equation (17) if and only if they are of the form

$$\begin{aligned} f(x) &= M_2^0(x) + M_1^0(x) + M_0^0(x) & (x \in X) \\ g(x) &= M_2^0(x) + M_1^0(x) + M_0^1(x) & (x \in X). \end{aligned} \quad (18)$$

Obviously, functional equation (17) has solutions which are not contained in (18). For example, if $f : X \rightarrow Y$ is an arbitrary function and $g : X \rightarrow Y$ has the property $g(x) = f(x)$, ($x \in X$), then the pair (f, g) satisfies (17).

Using our program, the input

```
> lfsolve(f(x)-f(y)=g(x)-g(y), [f, g]);
```

yields

```
Warning, for functional equations of this type
the program may not provide all solutions.
```

```
{f = M(0, 0) + M(1, 1) + M(1, 2),
g = M(1, 0) + M(1, 1) + M(1, 2)}.
```

The output of the program can be modified by the application of two different types of options. The first type makes it possible to obtain the solutions in a more detailed and more informative form. If we use its default version **brief**, we get the solutions in the same form as it was described above. E.g., in the case of the square-norm equation (16), the input

```
> lfsolve(f(x+y)+f(x-y)=2*f(x)+2*f(y), f, brief);
```

yields

```
{f = M(0, 2)}.
```

This output contains the solutions in an exact form, and they can also be used by other programs for further investigations.

However, in several cases, it is useful to get more information about the solutions in the output. This is possible with the application of the option **verbose**. If we use this, the program presents the unknown functions in the system considered as sums of monomial functions (similarly to (7) and (10) in Theorems 1 and 2, respectively), it yields the connections between these monomials (solving the systems of equations (8) or (11)), finally, it also gives the solutions as above. The solutions of the square-norm equation (16) with this option will be presented in the following form:

```
> lfsolve(f(x+y)+f(x-y)=2*f(x)+2*f(y), f, verbose);
```

The form of the solutions is

```
f = M(0, 0) + M(0, 1) + M(0, 2)
```

where $M(.,k)$ are monomial functions of degree k , for which

```
M(0, 0) = 0
```

```
M(0, 1) = 0
```

Thus,

```
f = M(0, 2)
```

```
{f = M(0, 2)}.
```

Considering the same thing for the linear functional equation ('chosen randomly')

$$f(x+y) + f(x-3y) = g(x-2y) - 2f(y) + f(x+2y) \quad (x, y \in X),$$

for the unknown functions $f, g: X \rightarrow Y$, using the input

```
> lfsolve(f(x+y)+f(x-3*y)=g(x-2*y)-2*f(y)+f(x+2*y), [f,g]);
```

we obtain

```
{f = M(0, 0)+M(1, 1), g = 3*M(0, 0)+M(1, 1)},
```

while

```
> lfsolve([f(x+y)+f(x-3*y)=g(x-2*y)-2*f(y)+f(x+2*y)], [f,g], 'verbose');
```

gives

The form of the solutions is

```
f = M(0, 0) + M(0, 1) + M(0, 2) + M(0, 3)
```

```
g = M(1, 0) + M(1, 1) + M(1, 2) + M(1, 3)
```

where $M(.,k)$ are monomial functions of degree k , for which

```
M(0, 1) = M(1, 1)
```

```
M(0, 2) = 0
```

```
M(0, 3) = 0
```

```
M(1, 0) = 3 M(0, 0)
```

```
M(1, 2) = 0
```

```
M(1, 3) = 0
```

Thus,

```
f = M(0, 0) + M(1, 1)
```

```
g = 3 M(0, 0) + M(1, 1)
```

```
{f = M(0, 0) + M(1, 1), g = 3 M(0, 0) + M(1, 1)}.
```

The program can also determine solutions of systems of functional equations of type (1). Performing calculations 'by hand', in the case of non-trivial linear equations, the solution of such systems could require lengthy or even infeasible computations. However, using computers, such situations can also be handled.

If a system of equations belonging to class (1) is given, where each equation contained in the system satisfies property (9) then, according to Theorem 2, the set of solutions of the system can be determined as the intersection of the sets of solutions of the single equations.

The situation is more problematic if there are some equations in the system which do not satisfy condition (9). In this case, Theorem 2 cannot be applied to these equations. If we determine their solutions (partly based on Theorem 1), we can 'lose' some solutions of the system which can be written as sums of monomial functions.

Let us illustrate this case with a simple example. Consider the system

$$\begin{aligned} f(x+y) + f(x-y) &= 2f(x) + 2f(y) & (x, y \in X) \\ f(x) &= g(x) & (x \in X) \end{aligned} \quad (19)$$

for unknown functions $f, g : X \rightarrow Y$. If we apply Theorem 1 to determine the (polynomial) solutions of the second equation, we obtain that such solutions are of the form

$$\begin{aligned} f(x) &= M_0^0(x) & (x \in X) \\ g(x) &= M_0^0(x) & (x \in X), \end{aligned}$$

where $M_0^0 : X \rightarrow Y$ is a monomial function of degree 0. On the other hand, according to Theorem 2 (as it was also determined in connection with equation (16)), all solutions of the first equation in (19) are

$$f(x) = M_2(x) \quad (x \in \mathbb{R}),$$

where $M_2 : \mathbb{R} \rightarrow \mathbb{R}$ is a monomial function of degree 2. If we determine the polynomial solutions of system (19) based on these facts, we obtain the functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = g(x) = 0$ for all $x \in \mathbb{R}$. Thus, we easily obtain that functions $f, g : X \rightarrow Y$ solve the system of functional equation given in (19) if and only if they are of the form

$$\begin{aligned} f(x) &= M_2(x) & (x \in X), \\ g(x) &= M_2(x) & (x \in X), \end{aligned}$$

where $M_2 : X \rightarrow Y$ is a monomial function of degree 2.

This means that, based on the theoretical results described in Sect. 2, we can determine the polynomial solutions of such systems in two different ways with possibly different results. This fact has to be taken into consideration when developing a computer program which determines solutions of systems of linear functional equations of type (1).

The program can manage such situations completely. Using the (default) option `strict` it will apply Theorem 1 ‘literally’ for those equations which do not satisfy condition 9 and will determine the solutions of the system considered accordingly. With the option `permissive`, it will investigate polynomials of higher order than Theorem 1 would allow and will determine the (polynomial) solutions of the system among them. It is important to point out that, with the application of these options, the user may decide which class of solutions will be determined.

Now, we present the solution of system (19) provided by our program in the two different ways described above. In order to make the examples more informative, we consider them in `verbose` form. If the option `strict` is used

```
> lfsolve([f(x+y)+f(x-y)=2*f(x)+2*f(y), f(x)=g(x)], [f,g],
verbose,strict);
```

we get

Warning, for functional equations of this type
the program may not provide all solutions.
The form of the polynomial solutions of degree at most 0 is
 $f = M(0, 0)$
 $g = M(1, 0)$
where $M(.,k)$ are monomial functions of degree k , for which
 $M(0, 0) = 0$
 $M(0, 1) = 0$
 $M(0, 2) = M(1, 2)$
 $M(1, 0) = 0$
 $M(1, 1) = 0$
Thus,
 $f = 0$
 $g = 0$
 $\{f = 0, g = 0\}$.

Applying the option permissive

```
> lfsolve([f(x+y)+f(x-y)=2*f(x)+2*f(y),f(x)=g(x)], [f,g],
verbose,permissive);,
```

we obtain

Warning, for functional equations of this type
the program may not provide all solutions.
The form of the polynomial solutions of degree at most 2 is
 $f = M(0, 0) + M(0, 1) + M(0, 2)$
 $g = M(1, 0) + M(1, 1) + M(1, 2)$
where $M(.,k)$ are monomial functions of degree k , for which
 $M(0, 0) = 0$
 $M(0, 1) = 0$
 $M(0, 2) = M(1, 2)$
 $M(1, 0) = 0$
 $M(1, 1) = 0$
Thus,
 $f = M(1, 2)$
 $g = M(1, 2)$
 $\{f = M(1, 2), g = M(1, 2)\}$.

The order of the options is given in the program: first **brief** or **verbose** and after that **strict** or **permissive** can be used. They can also be omitted (in this case always the default option is applied). Obviously, simultaneous application of the same type of options is not allowed.

In the case when the input is incorrect, the program yields an appropriate error message with a short hint at the problem (similarly to other error messages in Maple). For example,

```
>lfsolve(f(sqrt(2)*x+y)-f(x)-f(y),f);
```

gives

```
Error, (in solver) [05] invalid input: 2^(1/2)*x+y
the input
```

```
> lfsolve(f(x*y)-f(x)-f(y),f);
```

yields

```
Error, (in solver) [06] invalid argument: x*y in f(x*y).
```

and

```
> lfsolve(f(x+y)*f(x)-f(y),f);
```

has the output

```
Error, (in solver) [02] invalid input: f(x+y)*f(x),
```

(We note that the functional equations considered in the last two examples above make sense if f is a real function. However, since the equations are not of the form (1), they can not be solved with our program.) Finally,

```
> lfsolve(f(x+y)-f(x)-f(y),f,brief,verbose);
```

results in

```
Error, (in lfsolve) invalid input, brief, verbose.
```

Acknowledgements

Open access funding provided by University of Debrecen.

Open Access. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- [1] Aczél, J.: Vorlesungen über Funktionalgleichungen und ihre Anwendungen. Birkhäuser, Basel, Lehrbücher und Monographien aus dem Gebiete der exakten Wissenschaften, Mathematische Reihe, Bd. 25 (1961)
- [2] Aczél, J.: Lectures on Functional Equations and Their Applications, volume 19 of Mathematics in Science and Engineering. Academic Press, New York (1966)

- [3] Aczél, J., Dhombres, J.: *Functional Equations in Several Variables*. Cambridge University Press, Cambridge (1989)
- [4] Baják, Sz, Páles, Zs: Computer aided solution of the invariance equation for two-variable Gini means. *Comput. Math. Appl.* **58**, 334–340 (2009)
- [5] Baják, Sz, Páles, Zs: Computer aided solution of the invariance equation for two-variable Stolarsky means. *Appl. Math. Comput.* **216**(11), 3219–3227 (2010)
- [6] Baják, Sz, Páles, Zs: Solving invariance equations involving homogeneous means with the help of computer. *Appl. Math. Comput.* **219**(11), 6297–6315 (2013)
- [7] Baranyi, P., Csapo, A., Sallai, Gy: *Cognitive Infocommunications (CogInfoCom)*. Springer, Berlin (2015)
- [8] Baranyi, P., Gilányi, A.: Mathability: emulating and enhancing human mathematical capabilities. In: 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), pp. 555–558. IEEE (2013)
- [9] Borus, G.Gy., Gilányi, A.: On a computer program for solving systems of functional equations. In: 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), pp. 939. IEEE (2013)
- [10] Borus, G.Gy., Gilányi, A.: Solving systems of linear functional equations with computer. In: 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), pp. 559–562. IEEE (2013)
- [11] Chmielewska, K., Gilányi, A.: Mathability and computer aided mathematical education. In: 6th IEEE Conference on Cognitive Infocommunications (CogInfoCom), pp. 473–477. IEEE (2015)
- [12] Chmielewska, K., Gilányi, A.: Educational context of mathability. *Acta Polytechnica Hungarica* **15**, 223–237 (2018)
- [13] Czirbusz, S.: Testing regularity of functional equations with computer. *Aequationes Math.* **84**(3), 271–283 (2012)
- [14] Fréchet, M.: Une définition fonctionnelle des polynômes. *Nouv. Ann.* **49**, 145–162 (1909)
- [15] Fréchet, M.: Les polynômes abstraits. *J. Math. Pures et Appl. Sér. IX* **8**, 71–92 (1929)
- [16] Gilányi, A., Merentes, N., Quintero, R.: Mathability and an animation related to a convex-like property. In: 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom), pp. 227–232. IEEE (2016)
- [17] Gilányi, A., Merentes, N., Quintero, R.: Presentation of an animation of the m -convex hull of sets. In: 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom), pp. 307–308. IEEE (2016)
- [18] Gilányi, A.: *Charakterisierung von monomialen Funktionen und Lösung von Funktionalgleichungen mit Computern*. Diss. Universität Karlsruhe, Karlsruhe, Germany (1995)
- [19] Gilányi, A.: Solving linear functional equations with computer. *Math. Pannon.* **9**(1), 57–70 (1998)
- [20] Gselmann, E., Kiss, G., Vincze, Cs: On a class of linear functional equations without range condition. *Aequationes Math.* **94**(3), 473–509 (2020)
- [21] Hamel, G.: Eine Basis aller Zahlen und die unstetigen Lösungen der Funktionalgleichung $f(x + y) = f(x) + f(y)$. *Math. Ann.* **60**, 459–462 (1905)
- [22] Házy, A.: Solving linear two variable functional equations with computer. *Aequationes Math.* **67**(1–2), 47–62 (2004)
- [23] Kiss, G., Laczkovich, M.: Linear functional equations, differential operators and spectral synthesis. *Aequationes Math.* **89**, 301–328 (2015)
- [24] Kiss, G., Vincze, Cs: On spectral analysis in varieties containing the solutions of inhomogeneous linear functional equations. *Aequationes Math.* **91**(4), 663–690 (2017)
- [25] Kiss, G., Vincze, Cs: On spectral synthesis in varieties containing the solutions of inhomogeneous linear functional equations. *Aequationes Math.* **91**(4), 691–723 (2017)
- [26] Kuczma, M.: *An Introduction to the Theory of Functional Equations and Inequalities*, volume 489 of *Prace Naukowe Uniwersytetu Śląskiego w Katowicach*. Państwowe Wydawnictwo Naukowe—Uniwersytet Śląsk, Warszawa–Kraków–Katowice (1985)

- [27] Kuczma, M.: An introduction to the theory of functional equations and inequalities. Cauchy's equation and Jensen's inequality, 2nd edn. Birkhäuser, New York (2009)
- [28] Lundberg, A.: Some methods for a Sütő-Aczél project. *Aequationes Math.* **89**, 957–980 (2015)
- [29] Székelyhidi, L.: On a class of linear functional equations. *Publ. Math. Debrecen* **29**(1–2), 19–28 (1982)
- [30] Székelyhidi, L.: Convolution Type Functional Equations on Topological Abelian Groups. World Scientific Publishing Co., Inc., Teaneck (1991)
- [31] Szostok, T.: Alienation of two general linear functional equations. *Aequationes Math.* **94**(2), 287–301 (2020)
- [32] Wilson, W.H.: On a certain general class of functional equations. *Am. J. Math.* **40**(3), 263–282 (1918)

Gergő Gyula Borus
Institute of Mathematics
University of Debrecen
Egyetem tér 1
Debrecen 4032
Hungary
e-mail: gergo@borus.hu

Attila Gilányi
Faculty of Informatics
University of Debrecen
Egyetem tér 1
Debrecen 4032
Hungary
e-mail: gilanyi@inf.unideb.hu

Received: July 3, 2019

Revised: June 24, 2020