Aequat. Math.

https://doi.org/10.1007/s00010-023-00978-7

Aequationes Mathematicae

Computer assisted investigation of alienness of linear functional equations

Attila Gilányi and Lan Nhi To

*Dedicated to Professors Maciej Sablik and László Székelyhidi on the occasion of their 70*th *birthday.*

Abstract. In this paper, a computer program developed in the computer algebra system Maple is presented, which investigates alienness and strong alienness of linear functional equations.

Mathematics Subject Classification. Primary 39-04, 39B52, 39B72, Secondary 39A70.

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

Introduction

The concepts of alienness and strong alienness of functional equations were introduced by J. Dhombres in his paper [14] in the following form. Let $E_1(f) = 0$ and $E_2(f) = 0$ be two functional equations for a function f defined on a nonempty set X and mapping to a groupoid Y with an identity element 0. The equations E_1 and E_2 are called alien with respect to X and Y, if each solution $f : X \to Y$ of

$$E_1(f) + E_2(f) = 0$$

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.

Published online: 05 September 2023

is a solution of the system

$$E_1(f) = 0$$
$$E_2(f) = 0.$$

The equations E_1 and E_2 are said to be strongly alien with respect to X and Y, if all solutions $f, g : X \to Y$ of

$$E_1(f) + E_2(g) = 0$$

are solutions of the system

$$E_1(f) = 0$$
$$E_2(g) = 0.$$

These properties of functional equations have been investigated by several authors during the last decades (cf., e.g., [2,3,18,19,21,27,29,30,33]). We note that, in those publications, mostly the word 'alienation' was used for the concept which is called alienness here.

In this paper, we consider the alienness phenomenon for linear functional equations of the type

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

where n is a non-negative integer, $p_0, \dots, p_{n+1}$ and $q_0, \dots, q_{n+1}$ are rational numbers, X, Y are linear spaces and $f_0, \dots, f_{n+1} : X \to Y$ are unknown functions.

Classical results on the class of functional equations above and on its important sub-classes were published by M. Fréchet ([16,17]) and W. H. Wilson ([39]). Their solutions, under general conditions, were determined by L. Székelyhidi ([35,37]; cf., also, [34] and [36]). Recent studies of (1) and its generalizations, were performed, among others, in [22,25,31] and [38]. The alienness property, connected to linear functional equations, was also considered in several papers (e.g. in [1,15,20] and [38]).

Our aim is to present a computer program developed in the computer algebra system Maple,[1] which investigates alienness and strong alienness of linear functional equations belonging to class (1).

In the first part of the paper, we overview some basic properties of linear functional equations which are important for our investigations. In Sect. 2 we define some extensions of the alienness concepts of functional equations. Finally, in the last section we present our program.

[1] Maple is a trademark of Waterloo Maple Inc.

1. Basic properties of linear functional equations

In this section, we present some general theorems on solutions of linear functional equations of type (1). In order to formulate them, we give some basic definitions.

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

$$\Delta_y^0 f(x) = f(x) \qquad (x, y \in X)$$

and, for a non-negative integer n,

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

If n is a non-negative integer, a function $f : X \to Y$ is called a polynomial function of degree n if it satisfies

$$\Delta_y^{n+1} f(x) = 0 \qquad (x, y \in X),$$

f is said to be a monomial function of degree n if it fulfils

$$\Delta_y^n f(x) - n! f(y) = 0 \qquad (x, y \in X).$$

Now, we formulate some fundamental results on solutions of functional equations of type (1) determined by L. Székelyhidi ([35,37]; cf., also, [39]). In these theorems and in the following parts of the paper, we use the convention $0^0 = 1$.

Theorem 1. *Let n be a non-negative integer, let X and Y be linear spaces over the field of rationals, let $p_0, p_1, \dots, p_{n+1}$ and $q_0, q_1, \dots, q_{n+1}$ be rational numbers and assume that*

$$M_k^{(i)} : X \to Y$$

are monomial functions of degree k for $i = 0, \dots, n+1$ and $k = 0, \dots, n$. The functions $f_0, \dots, f_{n+1} : X \to Y$,

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

solve functional Eq. (1) if and only if the monomial functions $M_k^{(i)} : X \to Y$ given above fulfil

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

Theorem 2. *Let n be a non-negative integer, let X and Y be linear spaces over the field of rationals, let $p_0, p_1, \dots, p_{n+1}$ and $q_0, q_1, \dots, q_{n+1}$ be rational numbers that satisfy*

$$p_i q_j \neq p_j q_i \quad (i, j = 0, \dots, n+1,\ i \neq j). \tag{2}$$

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

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

where

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

are monomial functions of degree k *satisfying the equations*

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

2. Extensions of the concepts of alienness and strong alienness

With the help of computers, one can consider alienness and strong alienness properties for more than two functional equations and for more than one (or two) unknown functions as well. In order to perform such investigations, we extend the corresponding definitions of J. Dhombres ([14]) in the following way.

Let m, n be positive integers ($m \geq 2$) and let

$$\begin{aligned} E_1(f_1, \dots, f_n) &= 0 \\ &\vdots \\ E_m(f_1, \dots, f_n) &= 0 \end{aligned}$$

be functional equations for the unknown functions $f_1, \dots, f_n$ defined on a nonempty set X and mapping to a groupoid Y with an identity element 0.

The equations $E_1, \dots, E_m$ are called alien with respect to X and Y, if all solutions $f_i : X \to Y$, $(i = 1, \dots, n)$, of

$$E_1(f_1, \dots, f_n) + \cdots + E_m(f_1, \dots, f_n) = 0 \tag{3}$$

are solutions of the system

$$\begin{aligned} E_1(f_1, \dots, f_n) &= 0 \\ &\vdots \\ E_m(f_1, \dots, f_n) &= 0, \end{aligned} \tag{4}$$

too.

The equations $E_1, \dots, E_m$ are said to be strongly alien with respect to X and Y, if all solutions $f_{ij} : X \to Y$, $(i = 1, \dots, m,\ j = 1, \dots, n)$ of

$$E_1(f_{11}, \dots, f_{1n}) + \cdots + E_m(f_{m1}, \dots, f_{mn}) = 0 \tag{5}$$

are solutions of the system

$$\begin{aligned} E_1(f_{11},\dots,f_{1n}) &= 0 \\ &\vdots \\ E_m(f_{m1},\dots,f_{mn}) &= 0 \end{aligned} \tag{6}$$

as well.

3. Description of the program isalien

In this section, we describe the computer program `isalien`, which investigates alienness and strong alienness of linear functional equations of type (1).

The program was developed in the computer algebra system Maple. (It can be used in Maple 17 and in newer versions of Maple. A copy of the program can be requested from the first author of the paper.) It operates with formal computations (without any numerical approximations). This means that it always makes an exact decision about the alienness or strong alienness of the equations considered. By this property, using the terminology of 'mathability' ([7]), it 'increases the level of mathability' (or 'improves the mathability') of the underlying system (cf., also, [11] and [12]).

The program extensively uses another Maple program `lfesolve`, which determines solutions of functional equations belonging to class (1). (Cf. [10], furthermore, [8,9,23] and [24].) Just like `lfesolve`, `isalien` is contained in the Maple package `FunctionalEquations`. It enriches the field of computer algebraic investigations of functional equations and inequalities described, among others, in the papers [4–6,13,26,28] and [32].

Analogously to other Maple programs, `isalien` can be accessed by the forms

```
> FunctionalEquations[isalien](arguments);
```

or

```
> FunctionalEquations:-isalien(arguments);
```

or, if the command

```
> with(FunctionalEquations);
```

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

```
> isalien(arguments);.
```

To run the program, users have to give at least two input parameters: the list of linear functional equations called `e` and the list of the unknown functions involved in the given equations `f`. In addition, optional parameters can be used. Thus, the input of the program should be of the form

```
> isalien(e,f,options);.
```

The list of functional equations `e` have to be given as a Maple list structure between square brackets, all equations in the list are separated by a comma. Moreover, all of the input equations have to belong to the class (1) and the variables of all the unknown functions of the equations in `e` should be represented by 'x' and 'y'. An important property of `e` is that all of the functional equations in `e` must have the same number of unknown functions.

The list of unknown functions `f` also has to be given as a Maple list. In the general case, each equation in `e` can have more than one unknown function, so these unknown functions can be denoted by a nested list (i.e. by a list in list structure). (In the case when only one unknown function appears in each equation in `e`, the outside brackets can be omitted.)

Some examples of the correct form of `e` and `f`

$$[f_1(x+y)-f_1(x)-f_2(y)=0, g_1(x-2y)+3g_2(x)=0], [[f_1,f_2],[g_1,g_2]] \tag{7}$$

$$[f(x+y)-f(x)-f(y)=0, g(x-2y)+3g(x)=0], [[f],[g]]$$

or simply

$$[f(x+y)-f(x)-f(y), g(x-2y)+3g(x)], [f,g].$$

We note that the order of the unknown functions in list `f` is also important, since it will affect the final solution (cf. the definitions in Sect. 2). For example, the solutions for the input (7) and the input

$$[f_1(x+y)-f_1(x)-f_2(y)=0, g_1(x-2y)+3g_2(x)=0], [[f_1,f_2],[g_2,g_1]]$$

are different.

Furthermore, according to the definition of strong alienness of functional equations and its extension, when users want to check the strong alienness of functional equations given in `e`, the unknown functions appearing in the equations in `e` have to be pairwise different.

The program has three types of options:

- CHECK = {'`alien`', '`strongalien`'}
- OUTPUT = {'`brief`', '`verbose`', '`full`'}
- ErrorWarn_STYLE = {'`ewYes`', '`ewNo`'}

The default options are: ['`alien`, '`brief`', '`ewYes`'].

As it was mentioned above, `isalien` uses the Maple function `lfesolve`, which determines the exact solutions of systems of linear functional equations containing equations of type (1). It has turned out, that warning and error messages of the program `lfesolve` could be misleading for the users in the output of `isalien`. For this reason, we decided to modify `lfesolve` accordingly. With the new version, we can manage the warning and error messages directly in our main program and any other applications in the future.

If the input is given correctly, the program will provide an answer about the alienness or the strong alienness of the functional equations given in `e`. In the following, some examples with different options and their meanings will be shown.

When users give values for the parameters `e` and `f` only, the default values of the options will be used. This means that the program will investigate the alienness of the given equations and return the answer `True` or `False`. For example:

```
> isalien([f(x+y)-f(x)-f(y)=0,2*g(x-3*y)-g(3*x-8*y)-g(-x+2*y) = 0],
            [f, g]);
```

yields `True`.

For more information about the solutions of the functional equations considered, the OUTPUT option has to be modified into '`verbose`' or '`full`'. E.g.,

```
> isalien([f(x+y)-f(x)-f(y)=0,2*g(x-3*y)-g(3*x-8*y)-g(-x+2*y) = 0],
            [f, g], verbose);
```

gives

```
The functional equations
E1 :
f(x + y) − f(x) − f(y) = 0
E2 :
2 ∗ f(x − 3 ∗ y) − f(3 ∗ x − 8 ∗ y) − f(−x + 2 ∗ y) = 0
are alien
```

With the option '`full`', the program will present an output which contains the solutions of the functional equation $E_1 + \cdots + E_m = 0$ and the solutions of the system $[E_1 = 0, \ldots, E_m = 0]$. According to Theorems 1 and 2, the solutions are presented in the form of linear combinations of monomial functions. (Monomial functions of degree k, $M_k^i, (i = 0, \ldots, n+1,\ k = 0, \ldots, n)$ are denoted by `M(i,k)` in Maple.)

```
> isalien([f(x+y)-f(x)-f(y)=0,2*g(x-3*y)-g(3*x-8*y)-g(-x+2*y) = 0],
            [f, g], full);
```

yields

The functional equations
E1 :
$f(x+y)-f(x)-f(y)=0$
E2 :
$2*f(x-3*y)-f(3*x-8*y)-f(-x+2*y)=0$
are alien
In details,
The solution of function equation $E1+E2=0$ and the solution of the system $[E1=0, E2=0]$ is
$\{f=M(0,1)\}$

For a decision about strong alienness of functional equations, users can use the CHECK option with value '`strongalien`' combined with the appropriate OUTPUT option as mentioned above. For example,

```
> isalien([f(x+y)-f(x)-f(y)=0,2*g(x-3*y)-g(3*x-8*y)-g(-x+2*y)
= 0], [f, g], strongalien);
```

gives

```
True.
```

```
> isalien([f(x+y)-f(x)-f(y)=0,2*g(x-3*y)-g(3*x-8*y)-g(-x+2*y)
= 0],
              [f, g], strongalien, full);
```

returns

The functional equations
E1 :
$f(x+y)-f(x)-f(y)=0$
E2 :
$2*g(x-3*y)-g(3*x-8*y)-g(-x+2*y)=0$
are strongly alien
In details,
The solution of function equation $E1+E2=0$ and the solution of the system $[E1=0, E2=0]$ are
$\{f=M(0,1), g=M(1,0)+M(1,1)\}$

```
> isalien([f(x+y)-f(x)-f(y)=0,g(2*x+3*y)-2*g(x-y)=0], [f, g],

              strongalien, full);
```

gives

The functional equations
E1 :
$f(x+y)-f(x)-f(y)=0$
E2 :
$g(2*x+3*y)-2*g(x-y)=0$
are not strongly alien
In details,
The solution of function equation $E1+E2=0$ are
$\{f=-M(1,0)+M(0,1), g=M(1,0)\}$
while the solution of the system $[E1=0, E2=0]$ are
$\{f=M(0,1), g=0\}$

As it was mentioned in Chapter 2, our program is able to handle the situations when the input contains more than two functional equations and each equation has more than one unknown functions. In the following example, the program works with four functional equations:

```
> isalien([f(x+y)-f(x)-f(y)=0, g(x+5*y)+2*g(2*x-y)-2*g(x-y)=0,
           h(2*x-3*y)-h(2*x+7*y)=0, k(x+3*y)+2*k(x-y)=0],
          [f,g,h,k], verbose);
```

yields

The functional equations
E1 :
$f(x+y)-f(x)-f(y)=0$
E2 :
$f(x+5y)+2f(2x-y)-2f(x-y)=0$
E3 :
$f(2x-3y)-f(2x+7y)=0$
E4 :
$f(x+3y)+2f(x-y)=0$
are alien

In case the equations contain two unknown functions,

```
> isalien([f_1(x+y)-f_1(x)-f_1(y)+f_2(2*x+5*y)-f_2(x-6*y)= 0,
          g_1(2*x+y)-g_1(x-3*y)-g_1(x-2*y)+g_2(x-y)-g_2(x+2*y)= 0],

          [[f_1,f_2],[g_1,g_2]], full);
```

has the output

```
The functional equations
E1 :
        f_1(x + y) − f_1(x) − f_1(y)
+f_2(2x + 5y) − f_2(x − 6y) = 0
E2 :
        f_1(2x + y) − f_1(x − 3y) − f_1(x − 2y)
+f_2(x − y) − f_2(x + 2y) = 0
are alien
In details,
The solution of function equation E1 + E2 = 0 and the
solution of the system [E1 = 0, E2 = 0] are
{f_1 = 0, f_2 = M(1, 0)}
```

As mentioned before, our program decides about the alienness or strong alienness of linear functional equations by a comparison of the solutions of Eqs. (3) or (5) and the solutions of the systems of Eqs. (4) or (6), respectively. According to Theorem 2, we can find all solutions of functional equations which belong to class (1) in case these functional equations fulfill condition (2). On the other hand, if condition (2) is not valid, by Theorem 1, we are still able to determine all polynomial solutions of (1). This means that, in case condition (2) is not valid in (3) or (5), we can only find its polynomial solutions and cannot decide whether the remaining solutions are also solutions of (4) or (6), respectively, or not. Therefore, the final conclusion about the alienness and strong alienness might be affected.

If there exists a polynomial solution of (3) or (5) which does not fulfil (4) or (6), respectively, we can return that the given functional equations are not alien or not strongly alien, respectively. E.g.,

```
> isalien([f(x+y)-f(2*x+y)+f(x)=0, g(x+2*y)-g(y)-g(x)=0,
           h(2*x-3*y)-h(2*x+7*y)=0], [f,g,h], strongalien, full);
```

gives

```
The functional equations
E1 :
f(x + y) − f(2x + y) + f(x) = 0
E2 :
g(x + 2y) − g(y) − g(x) = 0
E3 :
h(2x − 3y) − h(2x + 7y) = 0
are not strongly alien
In details,
The polynomial solution of function equation E1 + E2 + E3 = 0 are
{f = M(1, 0) + M(0, 1), g = M(1, 0) + 10 ∗ M(2, 1),
                          h = M(2, 0) + M(2, 1)}
while the solution of the system [E1 = 0, E2 = 0, E3 = 0] are
{f = M(0, 1), g = 0, h = M(2, 0)}
```

However, if all polynomial solutions of (3) or (5) satisfy (4) or (6), respectively, the program is not able to decide about the alienness or strong alienness of the functional equations given. In this case, the program will provide a warning message announcing the problem and will determine a result if polynomial solutions are considered only.

```
> isalien([f_1(2*x-y)-2*f_1(x)+f_1(y)+f_2(x+5*y)+2*f_2(2*x-7*y)
             -2*f_2(x)-f_2(y)+f_3(2*x-3*y)-f_3(y) = 0,
           g_1(x+y)-g_1(x)-g_1(y)+g_2(x+y)+2*g_2(x-y)-3*g_2(x)
             -g_2(y)+g_3(2*x-y)-2*g_3(y) = 0],
           [[f_1, f_2, f_3], [g_1, g_2, g_3]], full);
```

gives

Warning, for functional equations of this type, isalien may not be able to decide about their alienness or strong alienness.
The functional equations
E1 :

$$\mathtt{f\_1}(2x-y)-2\mathtt{f\_1}(x)+\mathtt{f\_1}(y)+\mathtt{f\_2}(x+5y)+2\mathtt{f\_2}(2x-7y)$$
$$-2\mathtt{f\_2}(x)-\mathtt{f\_2}(y)+\mathtt{f\_3}(2x-3y)-\mathtt{f\_3}(y)=0$$

E2 :

$$\mathtt{f\_1}(x+y)-\mathtt{f\_1}(x)-\mathtt{f\_1}(y)+\mathtt{f\_2}(x+y)+2\mathtt{f\_2}(x-y)$$
$$-3\mathtt{f\_2}(x)-\mathtt{f\_2}(y)+\mathtt{f\_3}(2x-y)-2\mathtt{f\_3}(y)=0$$

are alien if polynomial solutions are considered only.
In details,
The polynomial solution of function equation $\mathtt{E1}+\mathtt{E2}=0$ and the polynomial solution of the system $[\mathtt{E1}=0,\mathtt{E2}=0]$ are

$$\{\mathtt{f\_1}=-\mathtt{M}(1,0)-\mathtt{M}(2,0)+\mathtt{M}(0,1),\mathtt{f\_2}=\mathtt{M}(1,0)$$
$$\mathtt{f\_3}=\mathtt{M}(2,0)\}$$

In the following example, the functional equation $E1+E2=0$ yields $0=0$, it means an arbitrary f fulfills this equation. Comparing this with the results of the system of functional equations $[E1=0, E2=0]$, the program will give the conclusion that the given functional equations $E1$ and $E2$ are not alien.

```
> isalien([f(x+y)-f(x)-f(y)=0,-g(x+y)+g(x)+g(y)=0], [f, g], full);
```

we obtain

The functional equations
E1 :

$$\mathtt{f}(x+y)-\mathtt{f}(x)-\mathtt{f}(y)=0$$

E2 :

$$-\mathtt{f}(x+y)+\mathtt{f}(x)+\mathtt{f}(y)=0$$

are not alien
In details,
The solution of function equation $\mathtt{E1}+\mathtt{E2}=0$ are arbitrary|||[f]
while the solution of the system $[\mathtt{E1}=0,\mathtt{E2}=0]$ is

$$\{\mathtt{f}=\mathtt{M}(0,1)\}$$

Obviously, besides checking alienness and strong alienness of linear functional equations 'directly', our program can also be used by other applications. In such cases, it might be confusing, if the the error or warning messages appear in the form mentioned above (e.g., in the output of the application using our program). In order to be able to manage such situations, we added one more parameter in the options part of the program called ErrorWarn_STYLE which accepts two values: '`ewYes`' and '`ewNo`'. This option can be used to switch on and switch off the warning and error messages in the output. In the following part, some examples relating to that option will be given.

```
> isalien([f(x+y)-g(x+2*y)+g(x) = 0, h(2*x+2*y)+k(y)-k(2*x-3*y) = 0],
                [[f, g], [h, k]], ewNo);
```

yields

```
True(If polynomial solutions are considered only)
```

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

```
> isalien([0 = 0, g(x+y)+2*g(x)+2*g(y) = 0], [f, g,], full);
```

has the output

```
Error, (in error_management) [07] invalid input : 0
```

the input

```
> isalien([f(x+y)-f(x)-f(y) = 0, g(x+y)-g(x)-h(y) = 0], [f, g], full);
```

gives

```
Error, (in error_management) [12] invalid input
in list of unknown functions
```

and

```
> isalien([f(2*x)-f(x)+f(y) = 0, 2*g(x)-g(y)-g(x+5*y)=0],
            [f,g], full, abc);
```

yields

```
Error, (in error_management) invalid arguments
in the input, [full, abc]
```

moreover

```
> isalien([f(2*x*y)-2*f(x)+f(y) = 0, 2*g(x)-g(y)-g(x+5*y) = 0],
              [f, g], full);
```

gives

```
Error, (in error_management) [06] invalid argument :
2 * x * yinf(2 * x * y)
```

The option ErrorWarn_STYLE can be used to switch off error messages and to return error codes (which can be used by other applications):

```
> isalien([f(2*x*y)-2*f(x)+f(y) = 0, 2*g(x)-g(y)-g(x+5*y) = 0],
              [f, g], full, ewNo);
```

yields

$$[\text{`'E06''}, [\texttt{2xy}, \texttt{f(2xy)}]]$$

Funding Open access funding provided by University of Debrecen.

Declarations

Conflict of interest I have no conflict of interest.

Ethical approval Not applicable.

References

[1] Adam, M.: Alienation of the quadratic and additive functional equations. Anal. Math. **45**, 449–460 (2019)
[2] Aissi, Y., Zeglami, D., Fadli, B.: Alienation of Drygas' and Cauchy's functional equations. In: Annales Mathematicae Silesianae, vol. 35, pp. 131–148 (2021)
[3] Bahyrycz, A., Sikorska, J.: On a general bilinear functional equation. Aequ. Math. **95**(6), 1257–1279 (2021)
[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., Gilányi, A.: Mathability: emulating and enhancing human mathematical capabilities. In 4^{th} International Conference on Cognitive Infocommunications (CogInfoCom), pp. 555–558. IEEE (2013)
[8] 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)
[9] Borus G. Gy., Gilányi, A.: Solving systems of linear functional equations with computer. In 4^{th} International Conference on Cognitive Infocommunications (CogInfoCom), pp. 559–562. IEEE (2013)
[10] Borus G. Gy., Gilányi, A.: Computer assisted solution of systems of two variable linear functional equations. Aequ. Math. **94**, 723–736 (2020)

[11] Chmielewska, K., Gilányi, A.: Mathability and computer aided mathematical education. In 6^{th} IEEE Conference on Cognitive Infocommunications (CogInfoCom), pp. 473–477. IEEE (2015)
[12] Chmielewska, K., Gilányi, A.: Educational context of mathability. Acta Polytech. Hung. **15**, 223–237 (2018)
[13] Czirbusz, S.: Testing regularity of functional equations with computer. Aequ. Math. **84**(3), 271–283 (2012)
[14] Dhombres, J.: Relations de dépendance entre les équations fonctionnelles de Cauchy. Aequ. Math. **35**(2–3), 186–212 (1988)
[15] Fadli, B.: Alienation of Cauchy's and the quadratic functional equations on semigroups. Aequ. Math. **95**(3), 527–534 (2021)
[16] Fréchet, M.: Une définition fonctionnelle des polynômes. Nouv. Ann. **49**, 145–162 (1909)
[17] Fréchet, M.: Les polynômes abstraits. Journal de Math. Pures et Appl. Sér. IX **8**, 71–92 (1929)
[18] Ger, R.: Additivity and exponentiality are alien to each other. Aequ. Math. **80**(1–2), 111–118 (2010)
[19] Ger, R.: Alienation of additive and logarithmic equations. Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Computatorica **40**, 269–274 (2013)
[20] Ger, R.: On alienation of two functional equations of quadratic type. Aequ. Math. **95**, 1169–1180 (2021)
[21] Ger, R., Sablik, M.: Alien functional equations: a selective survey of results. In: Developments in Functional Equations and Related Topics, Volume 124 of Springer Optim. Appl. pp. 107–147. Springer, Cham (2017)
[22] Gilányi, A., Lewicka, A.: On linear functional equations modulo $\mathbb{Z}$. Aequ. Math. **95**(6), 1301–1311 (2021)
[23] Gilányi, A.: Charakterisierung von monomialen Funktionen und Lösung von Funktionalgleichungen mit Computern. Universität Karlsruhe, Karlsruhe, Germany, Diss. (1995)
[24] Gilányi, A.: Solving linear functional equations with computer. Math. Pannon. **9**(1), 57–70 (1998)
[25] Gselmann, E., Kiss, G., Vincze, Cs.: On a class of linear functional equations without range condition. Aequ. Math. **94**(3), 473–509 (2020)
[26] Házy, A.: Solving linear two variable functional equations with computer. Aequ. Math. **67**(1–2), 47–62 (2004)
[27] Kominek, Z., Sikorska, J.: Alienation of the logarithmic and exponential functional equations. Aequ. Math. 90 (2016)
[28] Lundberg, A.: Some methods for a Sutô-Aczél project. Aequ. Math. **89**, 957–980 (2015)
[29] Maksa, Gy.: On the alienation of the logarithmic and exponential Cauchy equations. Aequ. Math. **92**(3), 543–547 (2018)
[30] Maksa, G., Sablik, M.: On the alienation of the exponential Cauchy equation and the Hosszú equation. Aequ. Math. **90**(1), 57–66 (2016)
[31] Nadhomi, T., Okeke, C.P., Sablik, M., Szostok, T.: On a new class of functional equations satisfied by polynomial functions. Aequ. Math. **95**(6), 1095–1117 (2021)
[32] Okeke, C.P., Sablik, M.: Functional equation characterizing polynomial functions and an algorithm. Results Math. **77**(3), 125 (2022)
[33] Sobek, B.: Alienation of the Jensen, Cauchy and D'alembert equations. Ann. Math. Silesianae **30**, 01 (2016)
[34] Székelyhidi, L.: The stability of linear functional equations. C. R. Math. Rep. Acad. Sci. Can. **3**(2), 63–67 (1981)
[35] Székelyhidi, L.: On a class of linear functional equations. Publ. Math. Debr. **29**(1–2), 19–28 (1982)
[36] Székelyhidi, L.: On a linear functional equation. Aequ. Math. **38**(2–3), 113–122 (1989)
[37] Székelyhidi, L.: Convolution Type Functional Equations on Topological Abelian Groups. World Scientific Publishing Co., Inc., Teaneck, NJ (1991)

[38] Szostok, T.: Alienation of two general linear functional equations. Aequ. Math. **94**(2), 287–301 (2020)
[39] Wilson, W.H.: On a certain general class of functional equations. Am. J. Math. **40**(3), 263–282 (1918)

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

Lan Nhi To
Doctoral School of Mathematical and Computational Sciences
University of Debrecen
Egyetem tér 1
Debrecen 4032
Hungary
e-mail: lan.nhi.to@science.unideb.hu

Received: February 9, 2023
Revised: February 9, 2023
Accepted: July 13, 2023