Mathematica Pannonica
9/1 (1998), 57–70

SOLVING LINEAR FUNCTIONAL EQUATIONS WITH COMPUTER

Attila **Gilányi**

Institute of Mathematics and Informatics, Kossuth Lajos University, H-4010 Debrecen, P.O.Box 12, Hungary

Received: December 1996

MSC 1991: 39 B 99

Keywords: Linear functional equations.

Abstract: In this paper we are going to give a computer-program which solves linear functional equations.

In the present work we deal with the linear functional equation

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

where n is a positive integer, $p_0, p_1, \dots, p_{n+1}$, $q_0, q_1, \dots, q_{n+1}$ are rational numbers, L, M are linear spaces over the rationals and $f_0, f_1, \dots$ $\dots, f_{n+1} : L \to M$ are unknown functions. Our aim is to give a computer-program which solves functional equations of this type. The description of this program can be found in Part 3 of the paper. The theoretical background of the program, discussed in Part 4, is based on the results of L. Székelyhidi ([2], [3]). The problem of writing such a program is also due to L. Székelyhidi.

Some parts of the results published here were achieved during the author's stay at the University of Karlsruhe, funded by the Deutscher Akademischer Austauschdienst (DAAD).

1. Notation and terminology

In the whole paper L and M denote linear spaces over the rationals. For a function $f : L \to M$, for $x, y \in L$ and $n \in \mathbb{N} = \{1, 2, 3, \dots\}$ define

$$\Delta_y^1 f(x) = \Delta_y f(x) = f(x + y) - f(x)$$

and

$$\Delta_y^{n+1} f(x) = \Delta_y(\Delta_y^n f(x)).$$

We call a function $f : L \to M$ a *polynomial function* of degree n, ($n \in \in \mathbb{N} \cup \{0\}$) if $\Delta_y^{n+1} f(x) = 0$ holds for all $x, y \in L$. A function $f : L \to M$ is called a *monomial function* of degree $n \in \mathbb{N}$ if $\Delta_y^n = n! f(y)$ holds for all $x, y \in L$. *Constant functions* are called monomial functions of degree 0. For a positive integer n a function $A : L^n \to M$ is said to be n-additive (for $n = 1$ additive) if for every $k = 1, 2, \dots, n$ and $x_1, x_2, \dots, x_k, y_k, \dots, x_n \in L$

$$\begin{aligned} &A(x_1, x_2, \dots, x_{k-1}, x_k + y_k, x_{k+1}, \dots, x_n) = \\ &= A(x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n) + \\ &+ A(x_1, x_2, \dots, x_{k-1}, y_k, x_{k+1}, \dots, x_n) \end{aligned}$$

holds; a function $A : L^n \to M$ is called *symmetrical* if for every permutation $i_1, i_2, \dots, i_n$ of the integers $1, 2, \dots, n$ and for all $x_1, x_2, \dots, x_n \in \in L$

$$A(x_1, x_2, \dots, x_n) = A(x_{i_1}, x_{i_2}, \dots, x_{i_n})$$

holds. We define the *diagonal* of a function $A : L^n \to M$ as follows: $A^* : L \to M$, $A^*(x) = A(x, x, \dots, x)$. In the case when $n = 0$ we take $L^0 = \{0\}$ with the zero-element of L, and we call the function $A : L^0 \to M$ *0-additive*. The diagonal of an $A : L^0 \to M$ is defined by $A^*(x) = A(0)$, $(x \in L)$.

It is known that for a nonnegative integer n if $A : L \to M$ is an n-additive symmetrical function then A^* is a monomial function of degree n, and conversely, if $f : L \to M$ is a monomial function of degree n then there exists an n-additive, symmetrical function $A : L^n \to M$ for which $A^* = f$ holds. (See [2], Th. 3.2.)

We call a function $f : L \to M$ by [2] a *function of degree* $n \in \mathbb{N} \cup \cup \{0\}$ if for $i = 1, 2, \dots, n + 1$ there exist functions $f_i : L \to M$ and additive functions $\varphi_i, \psi_i : L \to L$, such that for all $i = 1, 2, \dots, n + 1$ we have $\operatorname{Rg} \varphi_i \subseteq \operatorname{Rg} \psi_i$ and

$$f(x) + \sum_{i=1}^{n+1} f_i(\varphi_i(x) + \psi_i(y)) = 0 \qquad (x, y \in L).$$

(Here Rg means the range of the functions above.)

L. Székelyhidi has proved that for a function $f : L \to M$ of degree $n \in \mathbb{N} \cup \{0\}$ there exist k-additive symmetrical functions $A_k : L^k \to M$, $(k = 0, 1, \ldots, n)$ such that $f = \sum_{k=0}^{n} A^*{}_k$, and f uniquely determines the functions $A_0, A_1, \ldots, A_n$. (See [2], Th. 3.6 and Cor. 3.3.)

2. Theoretical background

Lemma 2.1. *Let n be a nonnegative integer. If for the functions $f_0, f_1, \ldots, f_{n+1} : L \to M$ functional equation (1) holds where for the rational numbers $p_0, p_1, \ldots, p_{n+1}$, and $q_0, q_1, \ldots, q_{n+1}$ we have*

(2) $$p_i q_j \neq p_j q_i \qquad (i, j = 0, 1, \ldots, n+1, i \neq j)$$

then the functions $f_0, f_1, \ldots, f_{n+1}$ are functions of degree n.

Proof. For an arbitrary rational number c the function $h : L \to L$, $h(x) = cx$ is an additive function, and for $c \neq 0$ we have $\operatorname{Rg} h = L$. Moreover for the rational numbers $p_0, p_1, \ldots, p_{n+1}, q_0, q_1, \ldots, q_{n+1}$ which satisfy condition (2) one and only one of the following properties is valid:

— $p_i q_i \neq 0$ for $i = 0, 1, \ldots, n+1$,
— there exists a $\lambda \in \{0, 1, \ldots, n+1\}$ for which $p_\lambda = 0$, $q_\lambda \neq 0$ and $p_i q_i \neq 0$ for $i = 0, 1, \ldots, n+1, i \neq \lambda$,
— there exists a $\lambda \in \{0, 1, \ldots, n+1\}$ for which $q_\lambda = 0$, $p_\lambda \neq 0$ and $p_i q_i \neq 0$ for $i = 0, 1, \ldots, n+1$, $i \neq \lambda$,
— there exist integers $\lambda, \mu \in \{0, 1, \ldots, n+1\}$, $\lambda \neq \mu$ such that $p_\lambda = 0$, $q_\lambda \neq 0$, $q_\mu = 0$, $p_\mu \neq 0$ and $p_i q_i \neq 0$ for $i = 0, 1, \ldots, n+1$, $i \neq \lambda$, $i \neq \mu$.

In each of these cases it can be shown by a simple variable transformation, that the functions $f_0, f_1, \ldots, f_{n+1}$ are functions of degree n. $\Diamond$

Theorem 2.2. *For a nonnegative integer n let $p_0, p_1, \ldots, p_{n+1}$, $q_0, q_1, \ldots, q_{n+1}$ be rational numbers and for $i = 0, 1, \ldots, n+1$, $k = 0, 1, \ldots, n$ let the functions $A_k^{(i)} : L^k \to M$ be k-additive symmetrical functions. The functions*

$$f_i = \sum_{k=0}^{n} A*_k^{(i)} \qquad (i = 0, 1, \dots, n+1)$$

solve functional equation (1) *if and only if*

$$\sum_{i=0}^{n+1} p_i^j q_i^{k-j} A*_k^{(i)}(x) = 0 \qquad x \in L)$$

holds for all $k = 0, 1, \dots, n$ *and* $j = 0, 1, \dots, k$. *(Here* $0^0 = 1$.*)*

Proof. See at L. Székelyhidi [2], Cor. 3.8. ◊

Theorem 2.3. *For a nonnegative integer* n *let* $p_0, p_1, \dots, p_{n+1}$, $q_0, q_1, \dots, q_{n+1}$ *be rational numbers which satisfy property* (2). *The functions* $f_0, f_1, \dots, f_{n+1} : L \to M$ *solve functional equation* (1) *if and only if they have the form*

$$f_i = \sum_{k=0}^{n} A*_k^{(i)} \qquad (i = 0, 1, \dots, n+1) \tag{3}$$

where $A_k^{(i)} : L^k \to M$, $(i = 0, 1, \dots, n+1,\ k = 0, 1, \dots, n)$ *are* k*-additive symmetrical functions for which*

$$\sum_{i=0}^{n+1} p_i^j q_i^{k-j} A*_k^{(i)}(x) = 0 \qquad (x \in L,\ k = 0, 1, \dots, n,\ j = 0, 1, \dots, k) \tag{4}$$

holds. (Here $0^0 = 1$.*)*

Proof. It follows from Th. 3.2 that the functions $f_0, f_1, \dots, f_{n+1}$ which have properties (3) and (4) solve equation (1).

The functions $f_0, f_1, \dots, f_{n+1}$ which solve (1) are, by Lemma 3.1, functions of degree n, so they have the form (3). And then, by Th. 3.2 we get that (4) also holds. ◊

Remark. In the last two theorems, because of the connection between monomial functions and multi-additive symmetrical functions, we may write monomial functions of degree k instead of k-additive symmetrical functions.

Example. *The functions* $f, g, h : L \to M$ *solve the so called Pexider equation (see among others* [1], 316*)*

$$f(x+y) - g(x) - h(y) = 0 \qquad (x, y \in L) \tag{5}$$

if and only if they have the form

$$(6)\qquad \begin{aligned} f(x) &= M_0^{(0)} + M_1^{(0)}(x) && (x \in L)\\ g(x) &= M_0^{(1)} + M_1^{(0)}(x) && (x \in L)\\ h(x) &= M_0^{(0)} - M_0^{(1)} + M_1^{(0)}(x) && (x \in L) \end{aligned}$$

where $M_k^{(i)} : L \to M$ *are for* $i = 0, 1$ *and* $k = 0, 1$ *monomial functions of degree* k.

Proof. Because of Lemma 3.1 the functions f, g, h are functions of degree 1, so, by 3.3 and 3.4, they solve (5) if and only if they have the form

$$\begin{aligned} f(x) &= M_0^{(0)} + M_1^{(0)}(x) && (x \in L)\\ g(x) &= M_0^{(1)} + M_1^{(1)}(x) && (x \in L)\\ h(x) &= M_0^{(2)} + M_1^{(2)}(x) && (x \in L), \end{aligned}$$

where $M_k^{(i)} : L \to M$, $(i = 0, 1, 2,\ k = 0, 1)$ are monomial functions of degree k, for which

$$(7)\qquad \begin{aligned} M_0^{(2)} &= M_0^{(0)} - M_0^{(1)} &&\\ M_1^{(2)}(x) &= M_1^{(0)}(x) && (x \in L)\\ M_1^{(1)}(x) &= M_1^{(0)}(x) && (x \in L) \end{aligned}$$

hold, which implies our statement. ◊

3. The computer-program *lfesolve*

3.1. General properties

The computer-program *lfesolve* (*l*inear *f*unctional *e*quations *solve*) was written in the Computeralgebra-System MAPLE[1] V, Release 3. It can be used for solving functional equations of type (1), where n is a positive integer, $p_0, p_1, \dots, p_{n+1}$, $q_0, q_1, \dots, q_{n+1}$ are rational numbers for which $p_i^2 + q_i^2 > 0$, $(i = 0, 1, \dots, n+1)$ holds, and $f_0, f_1, \dots, f_{n+1} : L \to M$ are unknown functions. For some $i \in$ $\in \{0, 1, \dots, n+1\}$ and a $c_i \in \mathbb{Q}$, $c_i \neq 0$ and a $j \in \{0, 1, \dots, n+1\}$ we might have the connection $f_i = c_i f_j$ between the unknown functions $f_0, f_1, \dots, f_{n+1}$ in (1). If for the rational numbers $p_0, p_1, \dots, p_{n+1}$ and

[1]MAPLE is a registered trademark of Waterloo Maple software.

$q_0, q_1, \ldots, q_{n+1}$ in (1) property (2) holds, we get the general solution of the equation; if (2) does not hold, we get only those solutions which can be written as a sum of monomial functions of degree at most n.

3.2. The input and the output of the program

The program can be started with the command

$lfesolve(f_0(p_0*x+q_0*y)+\ldots+f_{n+1}(p_{n+1}*x+q_{n+1}*y), [f_0, \ldots, f_{n+1}]);$

where the first parameter of $lfesolve$ is the functional equation to be solved (which must be of type (1), and must be given in the form above), and the second parameter is the list of the unknown functions in the functional equation.

If the parameters are not of this form, we get an error-message, otherwise the program gives the solutions of the functional equation in two forms: first we get the connections between the monomial functions the sum of which provides the solutions (like in (7)); after that the solutions themselves (like in (6)). The program also tells us if it does not give the general solution of the considered equation.

3.3. The list of the program[2]

```
lfesolve:=proc(e,f)
    local aa,ii,i,j,a,b,b1,b2,n,s,ss,hlp,P,Q,PP,QQ,
          Equat,Equat1,Fcts;
    global c,M;
restart;
with(linalg);
read ('ausgabe');
read ('loesung');
if nops(convert(f,set))<nops(f)
then ERROR('you have given wrong data.');
else
  s:=1;
  c:=subs('op(i,f)=0'$'i'=1..nops(f),e);
  if not c=0
  then ERROR('you have given wrong data.');
  else
```

[2]For the resource code of the program please write to the author's E-mail address:gil@math.klte.hu

```
for i to nops(f) do
  if not has(e,op(i,f)) then s:=0;
  fi;
od;
if s=0
then ERROR(`you have given wrong data.`);
elif not whattype(e)=`+` and nops(f)=1
then
  for i to nops(e)-1 do
    if hastype(op(i,e),function)
    then
      ERROR(`you have given wrong data.`);
      s:=0;
    fi;
  od;
  if s=1
  then print(`Equations like this will not be solved.`);
  fi;
elif not whattype(e)=`+`
then ERROR(`you have given wrong data.`);
else
  P:=array(1..nops(e));
  Q:=array(1..nops(e));
  ss:=0;
  for i to nops(e) do
  for j to nops(op(i,e))-1 do
    if (hastype(op(j,op(i,e)),function) or
        has(op(j,op(i,e)),f)) and ss=0
    then
      print(`Sorry, this program is not meant
              for solving equations like this.`);
      ss:=1;
    fi;
  od;
  od;
  if ss=0
  then
    for i to nops(e) do
      aa:=op(i,e);
```

```
        while not (type(aa,function) or whattype(aa)='**'
1 or type(aa,string)) do
          aa:=op(nops(aa),aa);
        od;
        if whattype(aa)='**' and ss=0
        then
          print('Sorry, this program is not meant
                 for solving equations like this.');
          ss:=1;
        elif whattype(aa)=string and ss=0
        then
          ERROR('you have given wrong data.');
          ss:=1;
        else
          b:=op(aa);
          if (not(type(b,polynom(rational,{x,y})))
              or not(degree(b,{x,y})=1)) and ss=0
          then
            print('Sorry, this program is not meant
1 for solving equations like this.');
            ss:=1;
          else
            P[i]:=coeff(b,x);
            Q[i]:=coeff(b,y);
          fi;
        fi;
      od;
      if ss=0
      then
        P:=convert(P,list);
        Q:=convert(Q,list);
        hlp:=0;
        for i to nops(P) do
        for j to nops(P) do
          if (not i=j) and P[i]*Q[j]=Q[i]*P[j]
          then
            hlp:=1
          fi;
        od;
```

```
od;
if hlp=1
then
  print(`Warning:  for functional equations
          of this type the program will`);
  print(`not provide the general solution.`);
fi;
M:='M';
Equat:=matrix(nops(e),nops(e)-1);
Equat1:=matrix(nops(e),nops(e)-1);
Fcts:=matrix(nops(e),nops(e)-1);
for i to nops(e) do
for j to nops(e)-1 do
  Equat1[i,j]:=M(i-1,j-1);
  Equat[i,j]:=M(i-1,j-1);
  Fcts[i,j]:=M(i-1,j-1);
od;
od;
for i to nops(e) do
for j to nops(e)-1 do
  Equat1[i,j]:=subs
  ('op(ii,f)=Equat[ii,j]'$'ii'=1..nops(f),op(i,e))
od;
od;
for i to nops(e) do
for j to nops(e)-1 do
  if nops(Equat1[i,j])>1
  then
    Equat1[i,j]:=
    product ('op(ii,Equat1[i,j])',
    'ii'=1..nops(Equat1[i,j])-1)
    *op(0,op(nops(Equat1[i,j]),Equat1[i,j]))
  else
    Equat1[i,j]:=op(0,Equat1[i,j])
  fi;
od;
od;
if nops(e)>nops(f)
then
```

```
            for i to nops(e)-1 do
            for j from nops(f)+1 to nops(e) do
            Fcts[j,i]:=Fcts[1,i];
            od;
            od;
          fi;
          ausgabe(e,f);
          loesung(P,Q,Equat1,Fcts,f);
        fi;
      fi;
    fi;
  fi;
fi;
end;


ausgabe:= proc(e,f)
    local i,j;
print(‘ The form of the solutions:‘ );
for i to nops(f) do
  print(op(i,f)=sum('M(i-1,j)','j'=0..nops(e)-2));
od;
print(‘where M(.,k) are monomial functions of degre k,‘);
print(‘for which:‘);
end;


loesung:=proc(P,Q,Mat,B,f)
    local i,j,k,hlp,g,lt,ult,sult,esult,result,BB,Ph,Qh;
lt:=matrix(nops(P)-1,nops(P)-1,0);
for i from 1 to nops(P) do
for j from 0 to nops(P)-2 do
  if P[i]=0 and j=0 then Ph[i,j]:=1 else Ph[i,j]:=P[i]**j
  fi;
  if Q[i]=0 and j=0 then Qh[i,j]:=1 else Qh[i,j]:=Q[i]**j
  fi;
od;
od;
for k from 0 to (nops(P)-2) do
for j from 0 to k do
```

```
  lt[j+1,k+1]:=
      sum('Ph[i,j]*Qh[i,k-j]*Mat[i,k+1]','i'=1..nops(P))=0:
od;
od;
esult:={};
for i to nops(P)-1 do
  ult:=solve(convert(col(lt,i),set),convert(col(B,i),set)):
  sult:={};
  for j to nops(ult) do
    if not op(2,(op(j,ult)))=op(1,(op(j,ult)))
    then
      sult:=sult union {op(j,ult)};
    fi;
  od;
  print(sult);
  esult:=esult union sult;
od;
for i to nops(f) do
  g[i]:=op(i,f)=sum('M(i-1,j)','j'=0..nops(P)-2)
od;
BB:='op(1,op(i,esult))'$'i'=1..nops(esult);
result:=solve(convert(g,set) union esult,
             convert(f,set) union {BB});
print(`That is:`);
for i to nops(result) do
  if has(f,op(1,op(i,result)))
  then print(op(i,result));
  fi;
od;
end;
```

3.4. Some testing results of the program

— The solution of the Pexider equation (see 3.5)

INPUT: $lfesolve(f(x+y)-g(x)-h(y),[f,g,h]);$

OUTPUT:

The form of the solutions:

$$f = M(0,0) + M(0,1)$$
$$g = M(1,0) + M(1,1)$$
$$h = M(2,0) + M(2,1)$$

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

$$\{M(0,0) = M(1,0) + M(2,0)\}$$
$$\{M(1,1) = M(0,1), M(2,1) = M(0,1)\}$$

So the solutions are:

$$f = M(1,0) + M(2,0) + M(0,1)$$
$$h = M(2,0) + M(0,1)$$
$$g = M(1,0) + M(0,1)$$

— The square-norm-equation

INPUT: $lfesolve(f(x+y) + f(x-y) - 2 * f(x) - 2 * f(y), f);$
OUTPUT:

The form of the solutions:

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

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

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

So the solutions are:

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

— A generalization of the square-norm-equation

INPUT: $lfesolve(f(x+y) + g(x-y) - h(x) - m(y), [f, g, h, m]);$

OUTPUT:

The form of the solutions:

$$f = M(0,0) + M(0,1) + M(0,2)$$
$$g = M(1,0) + M(1,1) + M(1,2)$$
$$h = M(2,0) + M(2,1) + M(2,2)$$
$$m = M(3,0) + M(3,1) + M(3,2)$$

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

for which:

$$\{M(3,0) = M(0,0) + M(1,0) - M(2,0)\}$$
$$\{M(0,1) = M(1,1) + M(3,1), M(2,1) = 2M(1,1) + M(3,1)\}$$
$$\{M(0,2) = M(1,2), M(2,2) = 2M(1,2), M(3,2) = 2M(1,2)\}$$

So the solutions are:

$$h = 2M(1,2) + M(2,0) + 2M(1,1) + M(3,1)$$
$$g = M(1,0) + M(1,1) + M(1,2)$$
$$m = M(0,0) + M(1,0) - M(2,0) + M(3,1) + 2M(1,2)$$
$$f = M(1,2) + M(0,0) + M(1,1) + M(3,1)$$

— The polynomial equation for $n = 5$

INPUT: $lfesolve(f(x+6*y) - 6*f(x+5*y) + 15*f(x+4*y) - 20*$
$*f(x+3*y) + 15*f(x+2*y) - 6*f(x+y) + f(x), f);$

OUTPUT:

The form of the solutions:

$$f = M(0,0) + M(0,1) + M(0,2) + M(0,3) + M(0,4) + M(0,5)$$

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

for which:

$$\{\}$$
$$\{\}$$
$$\{\}$$
$$\{\}$$
$$\{\}$$
$$\{\}$$

So the solutions are:

$$f = M(0,0) + M(0,1) + M(0,2) + M(0,3) + M(0,4) + M(0,5)$$

— The functional equation $f(x+3y) - f(x-y) + g(x) - 2h(x) = 0$, $(x, y \in L)$, for which we do not get the general solution

INPUT: $lfesolve(f(x+3*y) - f(x-y) + g(x) - 2*h(x), [f,g,h]);$
OUTPUT:

Warning: for functional equations of this type the program will not provide the general solution.

The form of the solutions:

$$f = M(0,0) + M(0,1) + M(0,2)$$
$$g = M(1,0) + M(1,1) + M(1,2)$$
$$h = M(2,0) + M(2,1) + M(2,2)$$

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

$$\{M(1,0) = 2M(2,0)\}$$
$$\{M(0,1) = 0, M(1,1) = 2M(2,1)\}$$
$$\{M(0,2) = 0, M(1,2) = 2M(2,2)\}$$

So the solutions are:

$$h = M(2,0) + M(2,1) + M(2,2)$$
$$f = M(0,0)$$
$$g = 2M(2,0) + 2M(2,1) + 2M(2,2)$$

References

[1] KUCZMA, M.: An Introduction to the Theory of Functional Equations and Inequalities, Państwowe Wydawnictwo Naukowe, Uniwersytet Śląski, Warszawa, 1985.

[2] SZÉKELYHIDI, L.: On a class of linear functional equations, *Publ. Math. Debrecen* **29** (1982), 19–28.

[3] SZÉKELYHIDI, L.: Convolution Type Functional Equations on Topological Abelian Groups, World Sci. Publ. Co., Singapore, 1991.