Письма в ЭЧАЯ. 2011. Т. 8, № 5(168). С. 823–827

AN IMPLEMENTATION OF THE HEAVISIDE ALGORITHM

I. H. Dimovski [1], *M. N. Spiridonova* [2]

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia

The so-called Heaviside algorithm based on the operational calculus approach is intended for solving initial value problems for linear ordinary differential equations with constant coefficients. We use it in the framework of Mikusiński's operational calculus.

A description and implementation of the Heaviside algorithm using a computer algebra system are considered. Special attention is paid to the features making this implementation efficient. Illustrative examples are included.

Алгоритм, связанный с именем Хевисайда и основанный на подходе операционного исчисления, предназначен для решения задач с начальными условиями для линейных обыкновенных дифференциальных уравнений с постоянными коэффициентами. Мы базируем алгоритм на операционном исчислении Микусинского.

Рассматривается описание алгоритма Хевисайда и его программная реализация с использованием системы компьютерной алгебры. Специальное внимание уделяется тем особенностям реализации, которые определяют его эффективность. Приводятся примеры решения задач с использованием реализованного алгоритма Хевисайда.

PACS: 02.30.-f

INTRODUCTION

The operational calculus of Oliver Heaviside (1850–1925) had been developed for practical applications. Working on problems of electromagnetic theory, Heaviside aimed to solve initial value problems for ordinary linear differential equations with constant coefficients and his method had to facilitate that by means of an algebraic algorithm. Unfortunately, the practical applicability and convenience of the suggested method were not sufficient for its good acceptance. A long period of rejections and justifications had to pass. In the mid-20th century the Polish mathematician Jan Mikusiński (1913–1987) developed a direct algebraic approach to the Heaviside operational calculus and changed the viewpoint of many mathematicians to it. His calculus is known as Mikusiński's operational calculus.

Further we describe the Heaviside algorithm for solving initial value problems for linear ordinary differential equations with constant coefficients in the framework of Mikusiński's operational calculus and our program implementation of this algorithm.

[1]E-mail: dimovski@math.bas.bg

[2]E-mail: mspirid@math.bas.bg

1. ABOUT THE APPROACH OF MIKUSIŃSKI

The most important role in Mikusiński's operational calculus is played by the classical Duhamel convolution (see [1]):

$$(f * g) = \int_0^t f(t-\tau)g(\tau)\,d\tau, \tag{1}$$

in the space $\mathcal{C}[0,\infty)$ of the continuous functions on $[0,\infty)$. Mikusiński considers this space as a ring on $\mathbb{R}$ or $\mathbb{C}$. He uses the fact that due to a famous theorem of Titchmarsh the operation (1) has no divisors of zero and hence $(\mathcal{C}[0,\infty), *)$ is an integrity domain. In the same way as the ring $\mathbb{Z}$ of the integers is extended to the field $\mathbb{Q}$ of the rational numbers, he extends the ring $(\mathcal{C}[0,\infty), *)$ to the smallest field $\mathcal{M}$ containing the initial ring. This field is called Mikusiński's field. The elements of $\mathcal{M}$ are convolution fractions $\dfrac{f}{g} = \dfrac{\{f(t)\}}{\{g(t)\}}$, called «operators».

In Mikusiński's calculus each function $f : [0,\infty) \to \mathbb{R}$ is considered as an algebraic object and the notation $f = \{f(x)\}$ is used.

Basic operators in the Mikusiński approach are the integration operator l: $lf(t) = \int_0^t f(\tau)\,d\tau$, and the algebraic analog on $s = 1/l$ of the differentiation operator d/dt.

The relation between the derivative $f'(t)$ and the product $s\{f(t)\}$ is presented by the basic formula of the Mikusiński operational calculus

$$\{f'(t)\} = s\{f(t)\} - f(0), \tag{2}$$

where $f \in C^1[0,\infty)$ and $f(0)$ is considered as a «numerical operator». If a function $f = \{f(t)\}$ has continuous derivatives to nth order for $0 \leqslant t < \infty$, a more general formula can be derived:

$$f^{(n)} = s^n f - \sum_{i=0}^{n-1} s^i f^{(n-1-i)}(0), \quad n = 1, 2, 3, \ldots \tag{3}$$

2. SOLVING INITIAL VALUE PROBLEMS FOR LINEAR ORDINARY DIFFERENTIAL EQUATION USING THE HEAVISIDE ALGORITHM

Let $P(\lambda) = a_0\lambda^n + a_1\lambda^{n-1} + \ldots + a_{n-1}\lambda + a_n$ be a nonzero polynomial of nth degree with real or complex coefficients.

Consider the following initial value problem:

$$P\left(\frac{d}{dt}\right) y = f(t), \quad y(0) = \gamma_0, \quad y'(0) = \gamma_1, \ldots, \quad y^{(n-1)}(0) = \gamma_{n-1}. \tag{4}$$

Using the main formulae (2), (3) of the operational calculus of Mikusiński, an «algebraization» of the problem can be made. The problem (7) reduces to the following single algebraic equation of first degree:

$$P(s)y = f + Q(s), \tag{5}$$

with $P(s) = \sum_{j=1}^{n} a_j\, s^j, \quad Q(s) = \sum_{j=1}^{n} \left(\sum_{k=j}^{n} a_{n-k}\, \gamma_{k-j} \right) s^{j-1}, \quad \deg Q < \deg P.$

The formal solution has the form

$$y = \frac{1}{P(s)} f + \frac{Q(s)}{P(s)}. \tag{6}$$

It can be interpreted as a functional solution if we decompose $1/P(s)$ and $Q(s)/P(s)$ in elementary fractions and interpret these fractions as functions using the formula (see [1]):

$$\frac{1}{(s-\alpha)^n} = \left\{ \frac{t^{n-1}}{(n-1)!} \mathrm{e}^{\alpha\, t} \right\}, \quad n = 1, 2, \ldots \tag{7}$$

Thus, we represent $1/P(s)$ and $Q(s)/P(s)$ as functions:

$$G(t) = 1/P(s), \quad R(t) = Q(s)/P(s) \tag{8}$$

and the solution takes the form

$$y(t) = G(t) * f(t) + R(t). \tag{9}$$

At last the computation of the convolution product denoted by $*$ in (9) has to be performed.

This is an outline of the main steps of the Heaviside algorithm for solving initial value problems for linear ordinary differential equations with constant coefficients.

The solution of an initial value problem for simultaneous ordinary linear differential equations with constant coefficients can be performed in a similar way: algebraization of the problem and reducing it to a system of linear algebraic equations; using linear algebra methods for solving the obtained system and functional interpretation of the solution.

3. IMPLEMENTATION OF THE HEAVISIDE ALGORITHM

3.1. General Remarks. A program implementation of the Heaviside algorithm would allow it to be used by means of computer. Having in mind the kind of the operations of this algorithm and the capabilities of the computer algebra system *Mathematica*, we decided to use it and to develop a program package implementing the Heaviside algorithm.

In the paper [2], published in 1984, the use of the computer algebra system Macsyma is mentioned for solving the algebraic equation at the corresponding step of the Heaviside algorithm. One of the authors of the same paper considers in [3] use of the operational calculus approach with *Mathematica* system for obtaining solutions of linear ordinary differential equations with polynomial coefficients and polynomial right-hand sides, in the form of power series. We did not find any information about a full implementation of the Heaviside algorithm.

3.2. Program Implementation of the Successive Steps of the Algorithm. Formulating once again the steps of the Heaviside algorithm, we will describe briefly the features of their program implementation. The case of one equation will be considered more elaborately.

Step 1. *Algebraization of the problem.* The language tools of *Mathematica* allow the transformation of (4) into (5) to be described in a convenient way using appropriate rules for formulae (2), (3) and for the initial conditions of (4).

Step 2. *Solution of the algebraic equation (5).* We have to solve a polynomial equation (or a system of such equations) and *Mathematica* provides this capability. The result is obtained in the form (6).

Step 3. *Factorization of the polynomial $P(s)$ and partial fraction decomposition of $1/P(s)$ and $Q(s)/P(s)$.* There is a built-in function (named $Factor$) in *Mathematica*, making factorization of a polynomial over the integers. The syntax of this function allows specifying an appropriate extension field, but, in general, it is difficult to do that. That is why we combine the use of the *Mathematica* functions $Solve$ and $NSolve$ (for solving the equation $P(\lambda) = 0$) and the function $Factor$, thus obtaining presentation of $P(s)$ as a product of factors, each of which is a polynomial of first or second degree, raised to an integer positive number. This process may not finish with success if some of the coefficients of $P(s)$ are parameters and in the same time $\deg P > 4$. In this case the solution of problem (4) is aborted. If the factorization of P is finished successfully, the *Mathematica* function $Apart$ represents the rational expressions $1/P(s)$ and $Q(s)/P(s)$ as sums of terms with minimal denominators of minimal degrees.

Step 4. *Interpretation of the rational expressions $1/P(s)$ and $Q(s)/P(s)$.* Each fraction in these expressions has to be interpreted as a function by means of formulae, such as (7). We use the main part of the Mikusiński table, excluding the special functions (see [1]). The formulae are presented using *Mathematica* rules and appropriate pattern matching. We achieved uniform interpretation of all fractions. As a result, the presentations (8) and (9) are obtained.

Step 5. *Computation of the Duhamel convolution in the final form of the solution.* Since this convolution is defined by a definite integral (see (1)), we use the *Mathematica* integrator for its computation.

Step 6. *Showing the result: solution or a message that the problem cannot be solved.* We mentioned above (in the description of Step 3) when the problem will not be solved in case of one equation. In case of solving initial value problem for a system of equations, a similar situation may occur, but, in addition, the problem will not be solved if on Step 2 the algebraic system has no solution.

3.3. Program Package for the Heaviside Algorithm. The implementation of the Heaviside algorithm with the features we just described is developed as a *Mathematica* program package. Its main function $DSolveOC$ defines the performance of all steps of the Heaviside algorithm. The call of this function is similar to the call of the *Mathematica* function $DSolve$. The output also has similar form. The solution is presented as a rule or as a list of rules in case of several solutions. The use of options for visualization of the solution and for some additional capabilities is provided.

Illustrative Examples. With the following two examples we illustrate the use of the main function $DSolve$ of our package. Two initial value problems, represented as first arguments of $DSolve$, are solved — for one linear ordinary differential equation and for a system of two linear ordinary differential equations

$$DSolveOC[\{x^{(4)}(t) + 2\,\alpha^2\,x''(t) + \alpha^4\,x(t) = \cosh(\alpha\,t),$$
$$x(0) = 1, x'(0) = 1,\ x''(0) = 1 - 2\,\alpha^2,\ x^{(3)}(0) = -2\,\alpha^2\},\ x(t), t],$$

$$x(t) \to \frac{1}{4\alpha^4}((-1 + 2\,(2+t)\,\alpha^4)\cos(t\alpha) +$$
$$+ \cosh(t\alpha) - \alpha\left(t - 2\alpha^2 - 2t\alpha^2 + 2t\alpha^4\right)\sin(t\alpha)),$$

$$DSolveOC[\{y''(t) - 2\,z(t) = 5\,e^t, y'(t) + z'(t) = t^2,$$
$$y(0) = a,\ y'(0) = b,\ z(0) = 1\},\ \{y(t), z(t)\},\ t],$$

$$\left\{y(t) \to \frac{1}{6}\left(2\left(3 + 3a + 5e^t - 3t + t^3\right) - 16\cos(\sqrt{2}t) + \sqrt{2}\,(-2+3b)\sin(\sqrt{2}t)\right),\right.$$
$$\left. z(t) \to \frac{1}{6}\left(10\,e^t + 6\,t - 4\,\cos(\sqrt{2}\,t) - \sqrt{2}\ (8 + 3\,b)\ \sin(\sqrt{2}\,t)\right)\right\}.$$

3.4. Conclusion about the Heaviside Algorithm and Its Implementation. We tried to compare the Heaviside algorithm and its implementation with classical algorithms and their implementation in *Mathematica*. The following notes can be outlined: (a) the Heaviside algorithm gives a closed form solution of an initial value problem for a linear ordinary differential equation or a system of such equations in a direct way, without trying to find partial and general solution; (b) a uniform approach is used for homogeneous and for non-homogeneous equations; (c) no special requirements to the right-hand side function are posed (as in the case of Laplace transformation); (d) the algorithm is convenient to be implemented and used in the environment of a computer algebra system; (e) during the experimental use of the package many examples were run using the Heaviside algorithm, the *Mathematica* function $DSolve$ and the Laplace transformation and in many cases advantages of our package were discovered.

The presented results are considered in more detail in [4].

REFERENCES

1. *Mikusiński J.* Operational Calculus. Oxford; Warszawa, 1959.
2. *Glinos N., Saunders B. D.* Operational Calculus Techniques for Solving Differential Equations // Proc. of EUROSAM 1984. Lecture Notes in Comp. Sci. V. 174. Springer-Verlag, 1984. P. 23–34.
3. *Glinos N.* Designing an Algorithm for the Series Solution of nth Order Linear ODEs with Polynomial Coefficients // Innovation in Mathematics: Proc. of the Second Intern. Mathematica Symp., Rovaniemi, Finland, 1997. P. 161–168.
4. *Spiridonova M.* Direct Operational Methods in the Environment of a Computer Algebra System. PhD Thesis. Inst. of Math. and Informatics, Bulg. Acad. of Sci. Sofia, 2008 (in Bulgarian).