Serdica J. Computing **1** (2007), 387–402

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

SOLVING DIFFERENTIAL EQUATIONS BY PARALLEL LAPLACE METHOD WITH ASSURED ACCURACY*

Natasha Malaschonok

Abstract. We produce a parallel algorithm realizing the Laplace transform method for the symbolic solving of differential equations.

In this paper we consider systems of ordinary linear differential equations with constant coefficients, nonzero initial conditions and right-hand parts reduced to sums of exponents with polynomial coefficients.

1. Introduction.

We produce a parallel algorithm applying the Laplace transform method to the symbolic solving of differential equations.

An application of Laplace transform in differential equations theory in spite of its long history is actual. It has been very useful in classical or modified forms for solving ordinary or partial differential equations ([2], [3], [8], [19], [21]). It is frequently applied to problems of fractional order equations ([6], [20]).

ACM Computing Classification System (1998): I.1.

Key words: Laplace transform, parallel algorithm, systems of differential equations, polynomials of exponents, partial fractions, systems of linear equations.

*The paper has been presented at the 12th International Conference on Applications of Computer Algebra, Varna, Bulgaria, June, 2006.

In this paper we consider systems of ordinary linear differential equations with constant coefficients, nonzero initial conditions and right-hand parts as composite functions, reducible to sums of exponents with polynomial coefficients. We stress the symbolic character of computations. Efficient algorithmizaton of symbolic solving is achieved at several stages.

At the first stage a preparation of data functions for the formal Laplace transform is performed (Section 2). It is achieved by application of Heaviside function and moving the obtained functions into the bounds of smoothness intervals. The parallelization of computations is realized as a multilevel tree, in the paper this is evident from the numeration of algorithm blocks.

The second stage is the parallel solving of the algebraic system with polynomial coefficients and a right-hand part obtained after the Laplace transform of the data system (Section 3). There are parallel algorithms which are very efficient for solving this type of equations, and are different for various types of such systems.

At the third stage the obtained solution of the algebraic system is prepared to the inverse Laplace transform. It is reduced to the sum of partial fractions with exponential coefficients. One of the problems is calculation of roots of a polynomial. In [17], [18] the algorithm to determine the error of the roots, sufficient for the required accuracy of the data system solution is obtained. The solving of the algebraic system for reducing into the sum of partial fractions is performed by means of parallel algorithms, cited in the paper.

At the last stage the solution of the data system is produced (Section 4). It is obtained as the real part of the inverse Laplace transform image of the algebraic system solution, prepared previously.

In the last section an example is considered.

2. Input data. Denote x_j, $j = 1, \dots, n$, unknown functions of argument t, $t \geq 0$, x_j^k – the order k derivative of the function x_j, $k = 0, \dots, N$. As the right-hand members of equations we consider here composite functions f_l, $l = 1, \dots, n$, whose components are represented as finite sums of exponents with polynomial coefficients. So we have to solve the system

$$\sum_{j=1}^{n} \sum_{k=0}^{N} a_{kj}^l x_j^k = f_l, \quad l = 1, \dots, n, \quad a_{kj}^l \in \mathbb{R}, \tag{1}$$

of n differential equations of order N with initial conditions $x_j^k(0) = x_{0j}^k$, $k = 0, \dots, N-1$ with functions f_l reduced to the form

(2) $$f_l(t) = f_l^i(t),\ t_l^i < t < t_l^{i+1},\ i = 1, \ldots, I_l, t_l^1 = 0, t_l^{I_l+1} = \infty,$$

where

$$f_l^i(t) = \sum_{s=1}^{S_l^i} P_{ls}^i(t) e^{b_{ls}^i t}, \quad i = 1, \ldots, I_l, \quad l = 1, \ldots, n,$$

and $P_{ls}^i(t) = \sum_{m=0}^{M_{ls}^i} c_{sm}^{li} t^m$.

2.1. Note. The algorithm tree is exposed by multi-index numerating of blocks. For example, **Block** $42jk$**,** $k = 1, \ldots, K_\Theta$, denotes the vertex $42jk$, which is an entrance of the k-th tree-edge, outgoing from the vertex $42j$ – **Block** $42j$, and there are K_Θ such edges. As k is the fourth index in the multi-index $42jk$, **Block** $42jk$ is the vertex of the fourth level. All the blocks **Block** $42jk$**,** $k = 1, \ldots, K_\Theta$ are performed independently and in parallel.

2.2. Block 1**: Block** 10**, Block** $1l$**,** $l = 1, \ldots, n$**.**

Data file. The data file contains the coefficients a_{kj}^l, the initial conditions x_{0j}^k, $k = 0, \ldots, N-1$, $j = 1, \ldots, n$, and the right-hand members f_l, $l = 1, \ldots, n$.

The data for functions f_l consists of the polynomial coefficients c_{sm}^{li}, parameters b_{ls}^i of exponents, the bounds t_i of smoothness intervals. Here $m = 0, \ldots, M_{ls}^i$, $s = 1, \ldots, S_l^i$, $i = 1, \ldots, I_l$. The numbers M_{ls}^i are degrees of corresponding polynomials, S_l^i are amounts of exponents in the expressions for f_l.

3. Laplace transform.

Denote the Laplace image of the function $x_j(t)$ by $X_j(p)$, of $f_l(t)$ by $F_l(p)$.

The Laplace transform of the left-hand part of the system (1) with respect to the initial conditions is performed by formal writing the expression

$$\sum_{j=1}^{n} \sum_{k=0}^{N} a_{kj}^l p^k X_j(p) - \sum_{j=1}^{n} \sum_{k=0}^{N-1} d_{jk}^l(p) x_{0j}^k,$$

where

(3) $$d_{jk}^l(p) = \sum_{i=k}^{N-1} a_{i+1,j}^l p^{i-k},$$

starting directly from input data.

3.1. Block $21l$**,** $l = 1, \dots, n$**.**

Preparation of right-hand functions $f_l(t)$ *to the Laplace transform.* The functions $f_l(t)$, $l = 1, \dots, n$, are composite and reduced to the form (2).

We use the Heaviside function $\eta(t)$ and represent $f_l(t)$ as a sum

$$f_l(t) = \sum_{i=2}^{I_l-1} [f_l^i(t) - f_l^{i-1}(t)]\eta(t - t_l^i) + f_l^1(t)\eta(t).$$

3.2. Block $21li$**,** $i = 1, \dots, I_l$**.**

Transform into the function of $t - t_l^i$*.* Transform $f_l^i(t) - f_l^{i-1}(t)$ into the function of $t - t_l^i$:

$$f_l^i(t) - f_l^{i-1}(t) = \phi_l^i(t - t_l^i).$$

Generally, the functions $f_l^i(t) - f_l^{i-1}(t)$ are decomposed into power series at the point t_l^i.

In our case the function $\phi_l^i(t - t_l^i)$ is represented as a finite sum

$$\phi_l^i(t - t_l^i) = \sum_{s=1}^{S_l^i} \psi_{ls}^i(t - t_l^i) e^{b_{ls}^i t_l^i} e^{b_{ls}^i (t - t_l^i)} - \sum_{s=1}^{S_l^{i-1}} \psi_{ls}^{i-1}(t - t_l^i) e^{b_{ls}^{i-1} t_l^i} e^{b_{ls}^{i-1}(t - t_l^i)}.$$

Here $\psi_{ls}^k(t - t_l^i) = P_{ls}^k(t)$ and $\psi_{ls}^k(t - t_l^i) = \sum_{m=0}^{M_{ls}^k} \gamma_{lsm}^{ki}(t - t_l^i)^m$. The coefficients γ_{lsm}^{ki} are calculated by the formula

$$\gamma_{lsm}^{ki} = \sum_{j=0}^{M_{ls}^k - m} c_{s,m+j}^{lk} \binom{m+j}{j} (t_l^i)^j.$$

Finally the function $f_l(t)$ is reduced to the form

$$f_l(t) = \sum_{i=2}^{I_l-1} \phi_l^i(t - t_l^i)\eta(t - t_l^i) + \sum_{s=1}^{S_l^1} P_{ls}^1(t) e^{b_{ls}^1 t}\eta(t).$$

3.3. Block $22l$**,** $l = 1, \dots, n$**.**

The parallel Laplace transform of the functions $f_l(t)$. Since the Laplace image of $(t-t^*)^n e^{\alpha(t-t^*)}\eta(t-t^*)$ is $\dfrac{n!}{(p-\alpha)^{n+1}}e^{-t^*p}$ the Laplace transform of $\phi_l^i(t-t_l^i)\eta(t-t_l^i)$ equals

$\Phi_l^i(p) =$

$$= \left[\sum_{s=1}^{S_l^i}\sum_{m=0}^{M_{ls}^i}\gamma_{lsm}^{ii}e^{b_{ls}^i t_l^i}\frac{m!}{(p-b_{ls}^i)^{m+1}} - \sum_{s=1}^{S^{i-1}i_l}\sum_{m=0}^{M_{ls}^{i-1}}\gamma_{lsm}^{i,\ i-1}e^{b_{ls}^{i-1}t_l^i}\frac{m!}{(p-b_{ls}^{i-1})^{m+1}}\right]e^{-t_l^i p}.$$

Finally, the Laplace transform of $f_l(t)$ is the following:

$$F_l(p) = \sum_{i=2}^{I_l-1}\Phi_l^i(p) + \sum_{s=1}^{S_l^1}\sum_{m=0}^{M_{s_l}^1}c_{sm}^{l1}\frac{m!}{(p-b_{ls}^1)^{m+1}}. \tag{4}$$

In the case when the right-hand part of the given system is exposed in the form

$$f_l(t) = \sum_{s=1}^{S_l}\sum_{m=0}^{M_{s_l}}c_{sm}^l t^m e^{b_{ls}t}, \quad l = 1,\dots,n,$$

the Laplace transform is performed formally – according to input data we write the expression for $F_l(p)$:

$$F_l(p) = \sum_{s=1}^{S_l}\sum_{m=0}^{M_{s_l}}c_{sm}^l\frac{m!}{(p-b_{ls})^{m+1}}, l = 1,\dots,n. \tag{5}$$

For each $l = 1,\dots,n$ we reduce (4)(or (5)) to the common denominator. The common denominator is left factorized. At that the nominator is the sum of exponents with polynomial coefficients.

In the case of a periodic function $f_l(t)$ with the period T the respective denominator contains the expression $1-e^{-pT}$. Then such a fraction is expanded into power series.

4. The parallel solving of the algebraic system.

4.1. Block 31.

The construction of the algebraic system. As a result of the Laplace transform of the system (1) we obtain an algebraic system relative to $X_j, j = 1, \dots, n$:

$$\sum_{j=1}^{n}\sum_{k=0}^{N} a_{kj}^{l} p^k X_j(p) = \sum_{j=1}^{n}\sum_{k=0}^{N-1} d_{jk}^{l}(p) x_{0j}^{k} + F_l(p), \quad l = 1, \dots, n, \tag{6}$$

where $d_{jk}^{l}(p)$ is defined by (3).

Denote

$$\sum_{j=1}^{n}\sum_{k=0}^{N-1} d_{jk}^{l}(p) x_{0j}^{k} = S_l(p).$$

We obtain the system

$$\sum_{j=1}^{n}\sum_{k=0}^{N} a_{kj}^{l} p^k X_j(p) = S_l(p) + F_l(p), \; l = 1, \dots, n. \tag{7}$$

For each $l = 1, \dots, n$ the expressions on the right-hand of (6) are reduced to a common denominator. The calculations are executed in parallel.

4.2. Block 32.

The parallel solving of the algebraic system. The system (7) may be solved by any possible classical method, for example Cramer's. But now there have been developed new effective procedures for parallel computations, for example, p-addic method ([5], [16], [15]), modula method ([15] – [11]), the method based on determinant identities ([15] – [10]). The fastest method for solving such systems is the p-addic method. But its code parallelization is not rather effective. The best one for parallelization is the modula method, based on Chinese Remainder Theorem.

5. Inverse Laplace transform.

5.1. Block 41j, $j = 1, \dots, n$.

Preparation of $X_j(p)$ to the inverse Laplace transform. Finally the solution of (7), i.e. each desired function $X_j(p)$, $j = 1, \dots, n$, is represented as a fraction with polynomial denominator. This denominator is partially factorized – it contains the multipliers of $F_l(p)$ denominators and the determinant $D(p)$ of system (7). The nominator is the sum of exponents with polynomial coefficients.

We reduce the function $X_j(p)$, $j = 1, \dots, n$, to the sum of exponents with fractional coefficients. The nominators and denominators of these coefficients are polynomials.

The next step is the decomposition of each fraction in the $X_j(p)$ expansion into the sum of partial fractions $A/(p - p^*)^v$, $p^* \in \mathbb{C}$. The first action here is the determination of the $D(p)$ roots.

5.2. Block $42j$**,** $j = 1, \dots, n$**.**

Computation of the denominator roots. As it was pointed out, the denominator of $X_j(p)$ is already represented as a product of partial multipliers and the polynomial $D(p)$. So we have to find the roots of $D(p)$.

The accuracy of these calculations is determined first of all. Its value must be sufficient for the preassigned precision of system solution. An algorithm to compute such accuracy is described in §5.

5.3. Block $42jk$**,** $k = 1, \dots, K_\Theta$**.**

Decomposition into a sum of partialfractions. We decompose rational fractions or fractional coefficients of exponents into sums of partial fractions $A/(p - p^*)^v$, $p^* \in \mathbb{C}$. The calculations for all fractions are performed in parallel, the number of blocks is formally denoted by K_Θ. It depends upon the parameters, which we do not describe here in detail.

One step of the algorithm is solving a system of linear equations with constant coefficients. Depending upon the size of system matrix we use one or another fast parallel algorithm, for example modular the method ([4] – [7], [15] – [10]).

If the roots of $D(p)$ have been found exactly, then we obtain the exact solution of the system (7) – the functions $X_j(p)$. Each of them is represented as a sum

$$X_j(p) = \sum_m \sum_k \frac{A_{mk}}{(p - p_{jk})^{\beta_{mk}}} e^{-\alpha_m p}. \tag{8}$$

Denote by $\Xi_j(p)$ the expression that represents $X_j(p)$ after its reduction to the partial fractions form in the case when the roots of $D(p)$ are not calculated exactly. Each $\Xi_j(p)$ is also written in the form (8).

5.4. Block $43j$, $j = 1, \ldots, n$.

Inverse Laplace transform. The Laplace originals of functions $X_j(p)$ are obtained formally – by writing the expressions

$$x_j(t) = \sum_m \sum_k \frac{A_{mk}}{(\beta_{mk} - 1)!}(t - \alpha_m)^{\beta_{mk}-1} e^{p_{ik}(t-\alpha_m)} \eta(t - \alpha_m), \quad j = 1, \ldots, n.$$

In the case when the roots of $D(p)$ are not calculated exactly denote $\xi_j(t)$ the Laplace original of $\Xi_j(p)$. It is also written in the form (8). In general, the functions $\xi_j(t)$ are complex valued. We take the real part of $\xi_j(t)$ for each $j = 1, \ldots, n$. The functions $\mathrm{Re}\xi_j(t)$ may be taken as the solution of the system (1), i.e., the required functions $x_j(t)$. It is easy to show that the error would not exceed the established precision, ensured by the calculated accuracy of roots of $D(p)$.

6. On the estimation of accuracy. We shall consider all functions and make calculations on the interval $[0, T]$, where T is sufficiently high for the input problem. In Section 4 we denoted by $\xi_l(t)$ the approximative solution of (1). We require the following accuracy for the solutions on the interval T:

$$\max_{t \in [0,T]} |x_l(t) - \xi_l(t)(t)| < \varepsilon, \; l = 1, \ldots, n. \tag{9}$$

We must determine an error Δ of the $D(p)$ roots, sufficient for the required accuracy ε for $x_l(t)$. An algorithm for computation of Δ is produced in ([17] – [18]).

Here we consider the case when the Laplace transform of the right-hand parts of (1) is expressed in the form (5). In this case the solution $x_l(t)$, $l = 1, \ldots, n$ of the given system may be written in the form

$$x_l(t) = \sum_{r=1}^{R_l} \left(\sum_{\mu=1}^{\mu_{lr}} \mathcal{B}^l_{r\mu} \frac{t^{\mu_{lr}-\mu}}{(\mu_{lr} - \mu)!)} \right) e^{p_{lr}t}, l = 1, \ldots, n. \tag{10}$$

Here p_{lr} denotes the pole of $X_l(p)$, μ_{lr} the order of this pole. The coefficients $\mathcal{B}^l_{r\mu}$ may be calculated as the Taylor coefficients of the functions $(p - p_{lr})^{\mu_l r} X_l(p)$ at the point p_{lr}.

Denote by $T_i^l(p)$ the i, l minor of the matrix of the system (1). The solution $X_l(p)$ of the system (6) may be expressed in the following way:

$$X_l(p) = \frac{D_l(p)}{D(P)},$$

where

$$D_l(p) = \sum_{i=1}^{N} [F_i(p) + S_i(p)]\, T_i^l(p).$$

Denote by p_r roots, and by $p_r^*, r = 1, \ldots, n$, approximate roots of the polynomial $D(p)$. If the error of a root is less than Δ, then $|p_r - p_r^*| < \Delta$. Let us consider the polynomial $D(p + \Delta e^{i\alpha})$, $\alpha \in [0, 2\pi]$. Denote $\tilde{X}_l(p) = \dfrac{D_l(p)}{D(p + \Delta e^{I\alpha})}$, and let the Laplace original of $\tilde{X}_l(p)$ be $\tilde{x}_l(t)$. We require the following estimation for the originals:

$$\max_{t \in [0,T]} |x_l(t) - \tilde{x}_l(t)| < \varepsilon,\ l = 1, \ldots, n. \tag{11}$$

We must find Δ which produces such estimation.

For (11) we must estimate the original of

$$\frac{D_l(p)}{D(p)} - \frac{D_l(p)}{D(p + \Delta e^{i\alpha})}.$$

According to the linearity of Laplace and inverse Laplace transforms we estimate separately the Laplace originals of

$$\sum_{i=1}^{n} F_i(p) \frac{T_i^l(p)}{D(p)} - \sum_{i=1}^{n} F_i(p) \frac{T_i^l(p)}{D(p + \Delta e^{i\alpha})} \tag{12}$$

and

$$\sum_{i=1}^{n} S_i(p) \frac{T_i^l(p)}{D(p)} - \sum_{i=1}^{n} S_i(p) \frac{T_i^l(p)}{D(p + \Delta e^{i\alpha})}. \tag{13}$$

Both functions $F_i(p)$ and $\dfrac{T_i^l(p)}{D(p)}$ have the Laplace originals. The original $Q_l(t)$ of (12) is calculated as the linear combination of convolutions of $f_i(t)$ and the originals of $\dfrac{T_i^l(p)}{D(p)}$ or $\dfrac{T_i^l(p)}{D(p+\Delta e^{i\alpha})}$, correspondingly. The original of $\dfrac{T_i^l(p)}{D(p)}$ is estimated starting from its representation in the form (10) using the Cauchy evaluations for the coefficients of the Tailor decomposition for the functions $\dfrac{T_i^l(p)}{D(p)}(p-p_r)^{m_r}$, $i=1,\dots,n$ in the δ-neighbourhood of each polo p_r, m_r is its order. Considering the correspondent integrals we obtain the estimation:

$$\max\nolimits_{t\in[0,T]}|Q_l(t)| \le e^{\sigma T}\left(\frac{T}{\delta}\right)^m M_l\left(e^{\Delta T}-1\right). \tag{14}$$

Here $\sigma>0$ is such that $D(p)$ has no zeros in the half plane Re $\ge \sigma$, $\delta < \min_{r,s}|p_r-p_s|$, M_l depends upon $\mathcal{M}(f_i)=\max_{t\in[0,T]}|f_i(t)|, i=1,\dots,n$, and the coefficients of the system (1).

The polynomial $S_i(p)$ has no an original. Denote $\sum_{i=1}^n S_i(p)T_i^l(p) = K_l(p)$. Then (13) may be written as following:

$$\frac{K_l(p)}{D(p)}-\frac{K_l(p)}{D(p+\Delta e^{i\alpha})}=\frac{K_l(p)}{D(p)}-\frac{K_l(p+\Delta e^{i\alpha})}{D(p+\Delta e^{i\alpha})}+\frac{\Delta e^{i\alpha}\cdot K_l'(p+\Delta e^{i\alpha})}{D(p+\Delta e^{i\alpha})}. \tag{15}$$

For estimation of the original $R_l(t)$ for (15) we obtain

$$\max_{t\in[0,T]}|R_l(t)| \le e^{\sigma T}\left(\frac{T}{\delta}\right)^m B_l\left(e^{\Delta T}-1\right). \tag{16}$$

Here B_l depends upon the elements of the coefficients of the system (1) and the initial conditions.

Starting from (14) and (16) we require

$$e^{\sigma T}\left(\frac{T}{\delta}\right)^m M_l\left(e^{\Delta T}-1\right)+e^{\sigma T}\left(\frac{T}{\delta}\right)^m B_l\left(e^{\Delta T}-1\right)<\varepsilon. \tag{17}$$

From (17) we find the estimation for Δ for each l:

$$\Delta_l < \frac{1}{T}\ln W(T,\sigma,\delta,M_l,B_l).$$

We choose the minimum with respect to l and obtain the final value of Δ.

Consider the case of two unknown functions ($n = 2$) and the third order of differential equations ($m = 3$). For this case the estimations (14) and (16) look like

$$\max_{t\in[0,T]} |Q_l(t)| \leq e^{\sigma T} \left(\frac{T}{\delta}\right)^3 \left(e^{\Delta T} - 1\right) \left(\mathcal{M}(f_1)a_l^2 + \mathcal{M}(f_2)a_l^1\right)$$

and

$$\max_{t\in[0,T]} |R_l(t)| \leq e^{\sigma T} \left(\frac{T}{\delta}\right)^3 \left(e^{\Delta T} - 1\right) \bar{a}_l^1 \bar{a}_l^2 \sum_{j=1}^{3} (|x_{01}^{j-1}| + |x_{02}^{j-1}|),$$

correspondingly. Here $a_l^s = \sum_{k=0}^{3} |a_{kl}^s|\sigma^k$, $\bar{a}_l^s = \sum_{k=1}^{3} |k a_{kl}^s|\sigma^k$, $l = 1, 2, s = 1, 2,$ a_{kl}^s are the coefficients of (1).

For Δ_l, $l = 1, 2$, we obtain the estimation

$$\Delta_l <$$

$$\frac{1}{T} \ln \left\{ e^{-\sigma T} \left(\frac{\delta}{T}\right)^m \left[\bar{a}_l^1 \bar{a}_l^2 \sum_{j=1}^{3} (|x_{01}^{j-1}| + |x_{02}^{j-1}|) + \mathcal{M}(f_1)a_l^2 + \mathcal{M}(f_2)a_l^1 \right]^{-1} \varepsilon + 1 \right\}.$$

Then

$$\Delta = \min_{l=1,2} \{\Delta_l\}.$$

7. Example.

7.1. Block 10.

$$(a_{kj}^1) = \begin{pmatrix} 1 & 0 & -1 & -2 \\ 3 & 1 & -2 & 0 \end{pmatrix}; \quad (a_{kj}^2) = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix};$$

$x_{01}^0 = 5$, $x_{01}^1 = 10$, $x_{01}^2 = 30$, $x_{02}^0 = 4$, $x_{02}^1 = 14$, $x_{02}^2 = 20$;

7.2. Block 11. $f_1^1 = e^t$, $f_1^2 = t^2 e^{2t}$, $t_1^1 = 0$, $t_1^2 = 1$;

7.3. Block 12. $f_2^1 = te^t, \quad f_2^2 = e^{2t}, \quad t_2^1 = 0, \quad t_2^2 = 1;$

7.4. Block 211. $f_1 = (f_1^2 - f_1^1)\eta(t - t_1^2) + f_1^1\eta(t);$

7.5. Block 212. $f_2 = (f_2^2 - f_2^1)\eta(t - t_2^2) + f_2^1\eta(t);$

7.6. Block 221. $F_1 = \dfrac{1}{-1+p} - \dfrac{e^{1-p}}{-1+p} + \dfrac{e^{2-p}(p^2-2p+2)}{(-2+p)^3};$

7.7. Block 222. $F_2 = \dfrac{e^{2-p}}{-2+p} + \dfrac{1}{(-1+p)^2} - \dfrac{e^{1-p}p}{(-1+p)^2};$

7.8. Block 31.

$$\left\{\begin{array}{rcl}
-2X_1 - pX_1 + p^3X_1 + X_2 - p^3 & = & 10 - 4p - 4p^2 + 5(-1+p^2)+ \\
& & +\dfrac{1}{-1+p} - \dfrac{e^{1-p}}{-1+p}+ \\
& & +\dfrac{e^{2-p}(p^2-2p+2)}{(-2+p)^3} \\
-2pX_1 + p^2X_1 + 3p^3X_1 + X_2 + p^3X_2 & = & 110 + 14p + 4p^2 + 10(1+3p)+ \\
& & +(-2+p+3p^2) + \dfrac{e^{2-p}}{-2+p}+ \\
& & +\dfrac{1}{(-1+p)^2} - \dfrac{e^{1-p}p}{(-1+p)^2;}
\end{array}\right.$$

$D(p) = -2 + p - p^2 - 4p^3 - 3p^4 + p^5 + 4p^6;$

7.9. Block 411.

$$\begin{aligned}
X_1(p) \;=\; & e^{-p}\left(\frac{8e + 2e^2 - 4ep - 4e^2p + 2ep^2 + 2e^2p^2 + 9ep^3 - 6e^2p^3 - 19ep^4}{(-2+p)^3(-1+p)(-2+p-p^2-4p^3-3p^4+p^5+4p^6)}+\right.\\
& \left. + \frac{12e^2p^4 + 11ep^5 - 8e^2p^5 - 2ep^6 + 2e^2p^6}{(-2+p)^3(-1+p)(-2+p-p^2-4p^3-3p^4+p^5+4p^6)}\right)+\\
& +\frac{-856 + 1692p - 982p^2 + 1061p^3 - 1991p^4 + 1398p^5 - 412p^6}{(-2+p)^3(-1+p)(-2+p-p^2-4p^3-3p^4+p^5+4p^6)+}\\
& +\frac{160p^7 - 95p^8 + 20p^9}{(-2+p)^3(-1+p)(-2+p-p^2-4p^3-3p^4+p^5+4p^6)};
\end{aligned}$$

7.10. Block 412.

$$X_2(p) = e^{-p}\left(\frac{-8e^2 - 32ep + 24e^2p + 64ep^2 - 28e^2p^2 - 32ep^3 + 17e^2p^3}{(-2+p)^3(-1+p)^2(-2+p-p^2-4p^3-3p^4+p^5+4p^6)} + \right.$$
$$\left. + \frac{-24ep^4 - 5e^2p^4 + 34ep^5 - 3e^2p^5 - 14ep^6 + 5e^2p^6 + 2ep^7 - 2e^2p^7}{(-2+p)^3(-1+p)^2(-2+p-p^2-4p^3-3p^4+p^5+4p^6)}\right) +$$
$$+ \frac{1776 - 4576p + 3568p^2 - 1404p^3 + 2465p^4 - 2751p^5 + 841p^6}{(-2+p)^3(-1+p)^2(-2+p-p^2-4p^3-3p^4+p^5+4p^6)};$$
$$+ \frac{+133p^7 + 2p^8 - 68p^9 + 16p^{10}}{(-2+p)^3(-1+p)^2(-2+p-p^2-4p^3-3p^4+p^5+4p^6)};$$

7.11. Block 421. $p^1_{x_1} = 1$, $p^2_{x_1} = -0.5949378 - 0.830714i$, $p^3_{x_1} = -0.5949378 + 0.830714i$, $p^4_{x_1} = 0.355937 - 0.513128i$, $p^5_{x_1} = 0.355937 + 0.513128i$, $p^6_{x_1} = 1$, $p^7_{x_1} = 1.228$, $p^8_{x_1} = 2$, $p^9_{x_1} = 2$, $p^{10}_{x_1} = 2$.

7.12. Block 422. $p^1_{x_2} = -1$, $p^2_{x_2} = -0.5949378 - 0.830714i$, $p^3_{x_2} = -0.5949378 + 0.830714i$, $p^4_{x_2} = 0.355937 - 0.513128i$, $p^5_{x_2} = 0.355937 + 0.513128i$, $p^6_{x_2} = 1$, $p^7_{x_2} = 1$, $p^8_{x_2} = 1.228$, $p^9_{x_2} = 2$, $p^{10}_{x_2} = 2$, $p^{11}_{x_2} = 2$.

7.13. Block 431.

$$x_1(t) = \begin{cases} \quad 10.031249e^{-t} - 1.25e^{t} + 5.538602e^{1.228001t} + \\ + 2e^{0.355937t}(-3.735568\cos[0.513128t] + 15.529795\sin[0.513128t]) + \\ + 2e^{-0.594937t}(-0.924357\cos[0.830713t] + 0.061193\sin[0.830713t]), \\ \quad 0 < t < 1; \\ \quad 9.425260e^{-t} + 4.378584e^{1.228001t} + \\ + 0.322878e^{2t} - 0.185554e^{2t}t + 0.0441176e^{2t}t^2 + \\ + 2e^{0.355937t}(-3.708886\cos[0.513128t] + 15.104078\sin[0.513128t]) + \\ + 2e^{-0.594938t}(-0.953417\cos[0.830713t] + 0.057591\sin[0.830713t]), \\ \quad t > 1; \end{cases}$$

7.14. Block 432.

$$x_2(t) = \begin{cases} \quad 10.03125e^{-t} + 0.5e^{t} - 8.948223e^{1.228001t} + 0.5e^{t}t + \\ + 2e^{0.355937t}(-0.493116\cos[0.513128t] + 33.959275\sin[0.513128t]) + \\ + 2e^{-0.594938t}(1.701602\cos[0.830713t] + 0.929609\sin[0.830713t]), \\ \quad 0 < t < 1; \\ \quad 9.42526e^{-t} - 7.074087e^{1.228001t} - \\ - 0.577176e^{2t} + 0.435986e^{2t}t - 0.117647e^{2t}t^2 + \\ + 2e^{0.355937t}(-0.636666\cos[0.513128t] + 33.06373\sin[0.513128t]) + \\ + 2e^{-0.594938t}(1.748891\cos[0.830714t] + 0.968596\sin[0.830714t]), \\ \quad t > 1; \end{cases}$$

The table gives the values of Δ for three values of ε and three values of T.

$T\backslash\varepsilon$	$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
$T = 2$	$1.37 \cdot 10^{-10}$	$1.37 \cdot 10^{-11}$	$1.37 \cdot 10^{-12}$
$T = 3$	$1.25 \cdot 10^{-12}$	$1.25 \cdot 10^{-13}$	$1.25 \cdot 10^{-14}$
$T = 4$	$5.93 \cdot 10^{-15}$	$6.10 \cdot 10^{-16}$	$5.55 \cdot 10^{-17}$

8. Conclusion. We produce the application of Laplace method in symbolic computation. This is a great reason for significant advantages in solving differential equations. The parallel algorithm permits to increase essentially the velocity of the calculations, that allows to solve systems of great size in real time.

Estimation of accuracy for solution of differential equations in numerical computations is a natural part of each algorithm. It is of high importance to assure the necessary accuracy in symbolic computations as well, especially because elements of numerical computations are contained in every analytical algorithm.

The method presented in the paper gives not only an efficient algorithm for solving differential equations, but guarantees a necessary accuracy of a solution.

The work is partially supported by the grant RFBR 04-07-90268b, RFBR, No.05-01-00074a, the Sci. Program "Devel. Sci. Potent. High. School", RNP.2.1.1.351.

REFERENCES

[1] Akritas A. G. Elements of Computer Algebra with Applications. J. Wiley Interscience, New York, 1989.

[2] Burghelea D., S. Haller. Laplace transform, dynamics and spectral geometry. arXiv:math.DG/0405037v2, 17 Jan., 2005.

[3] Dahiya R. S., Jabar Saberi-Nadjafi. Theorems on n-dimensional Laplace transforms and their applications. 15th Annual Conf. of Applied Math., Univ. of Central Oklahoma, Electr. Journ. of Differential Equations, Conf.02, (1999) 61–74.

[4] Davenport J., Y. Siret, E. Tournier. Calcul formel. Systemes et algorithmes de manipulations algebriques. MASSON, Paris, 1987.

[5] Dixon J. Exact solution of linear equations using p-adic expansions. *Numer. Math.* **40** (1982), 137–141.

[6] Felber F. S. New exact solutions of differential equations derived by fractional calculus. arXiv:math/0508157v1, 9 Aug., 2005.

[7] Von zur Gathen J., J. Gerhard. Modern Computer Algebra. Cambridge University Press, 1999.

[8] Goursat E. Sur les équations linéaires et la méthode de Laplace. *Amer. J. Math.* **18** (1896), 347–385.

[9] Malaschonok G. I. Solution of a system of linear equations in an integral domain. *USSR J. of Comput. Math. and Math. Phys.* **23**, No. 6 (1983), 497–500.

[10] Malaschonok G. I. The solution of a system of linear equations over a commutative ring. *Math. Notes* **42**, No. 3–4 (1987), 801–804.

[11] Malaschonok G. I. Algorithms for the solution of systems of linear equations in commutative rings. In: Effective methods in Algebraic Geometry. Progr. Math. vol. **94**, Birkhauser, Boston, 1991, 289–298.

[12] Malaschonok G. I. Algorithms for computing determinants in commutative rings. *Discrete Math. Appl.* **5**, No. 6 (1996), 557–566.

[13] Malaschonok G. I. Recursive Method for the Solution of systems of Linear Equations. In: Computational Mathematics (15th IMACS World Congress Vol. I, Berlin, August 1997), Wissenschaft and Technik Verlag, Berlin, 1997, 475–480.

[14] Malaschonok G. I. Effective Matrix Methods in Commutative Domains. In: Formal Power Series and Algebraic Combinatorics, Springer, Berlin, 2000, 506–517.

[15] Malaschonok G. I. Matrix computational methods in commutative rings. Tambov, TSU, 2002.

[16] Malaschonok G. I. Solution of systems of linear equations by the p-adic method. *Programming and Computer Software* **29**, No. 2 (2003), 59–71.

[17] Malaschonok N. An algorithm to settle the necessary exactness in Laplace transform method. In: Computer science and information technologies. Yerevan, 2005, 453–456.

[18] Malaschonok N. Estimations in Laplace transform method for solutions of differential equations in symbolic computations. In: Differential equations and computer algebra systems. Minsk, 2005, 195–199.

[19] Mizutani N., Yamada H. Modified Laplace transformation method and its applications to anharmonic oscillator. arXiv:physics/9702013, 15 Feb., 1997.

[20] Podlubny I. The Laplace transform method for linear differential equations of the fractional order. arXiv:funct-an/9710005v1, 30 Oct., 1997.

[21] Roux J. Le. Extensions de la methods de Laplace aux équatons linéaires aux deriveées partielles dórde supérieur du second. *Bull. Soc. Math. de France* **27** (1899), 237–262

Tambov State University
Internatsionalnaya 33
392622 Tambov, Russia
e-mail: `malaschonok@narod.ru`

Received September 25, 2006
Final Accepted October 18, 2007