Memoirs on Differential Equations and Mathematical Physics

Volume 91, 2024, 39–50

David Gulua

THE PERTURBATION ALGORITHM FOR THE REALIZATION OF A MULTI-LAYER SEMI-DISCRETE SOLUTION SCHEME FOR AN ABSTRACT EVOLUTIONARY PROBLEM

Abstract. The paper presents the study of the Cauchy problem for an evolutionary equation with a self-adjoint positive definite operator. Using the perturbation algorithm, the purely implicit multi-layer semi-discrete solution scheme is reduced to two-layer schemes. Solutions of the latter schemes are used for constructing an approximate solution of the original problem. The estimate of the approximate solution error is given.

2020 Mathematics Subject Classification. 65J08, 65M12, 65M15, 65M55.

Key words and phrases. Evolutionary problem, semi-discrete scheme, multi-layer scheme, perturbation algorithm.

რეზიუმე. ნაშრომში განხილულია კოშის ამოცანა აბსტრაქტული ევოლუციური განტოლებისთვის თვითშეუღლებული, დადებითად განსაზღვრული ოპერატორით. აღნიშნული ამოცანისთვის, შეშფოთებათა ალგორითმის გამოყენებით, წმინდად არაცხადი მრავალშრიანი ნახევრად-დისკრეტული სქემა დაყვანილია ორშრიან სქემებზე. ორშრიანი სქემების ამონახსნებით იგება საწყისი ამოცანის მიახლოებითი ამონახსნი. მოყვანილია მიახლოებითი ამონახსნის ცდომილების შეფასება.

1 Introduction

The semi-discretization method (the method based on the discretization of a derivative with respect to a time variable) is one of the methods used to solve the Cauchy problem for an abstract evolutionary equation. The semi-discretization method for an evolutionary equation is also known as the Rothe method [27]. The investigation and implementation of multilevel schemes for evolution problems is an important issue. The main difficulty in the implementation of multilevel schemes (especially for multi-dimensional problems) is the use of a large amount of memory, which increases proportionally to the number of levels. One way to overcome this difficulty is to split the multilevel schemes.

The application of the perturbation algorithm to difference schemes for differential equations was considered in [1]. Mention should also be made of the works [24], [11], where a purely implicit three-layer semi-discrete scheme for an evolutionary equation is reduced to two two-layer schemes and the explicit estimates for the approximate solution error are proved in the Banach space under rather general assumptions about the problem data.

Note that in [25, 26], a purely implicit four-layer semi-discrete scheme for an approximate solution of the Cauchy problem for an evolutionary equation is reduced to three two-layer schemes. In that case, the explicit estimates of the approximate solution error are proved in the Hilbert space.

In the present paper, we consider the purely implicit multi-layer semi-discrete scheme for an approximate solution of the Cauchy problem for an evolutionary equation with a self-adjoint positive definite operator in the Hilbert space. Using the perturbation algorithm, the scheme is reduced to two-level schemes. An approximate solution of the original problem is constructed by solving these two-level schemes.

The algorithm proposed in the above-mentioned works is close to the methods discussed in [16, 18, 19].

The perturbation algorithm is widely used for the solution of problems in mathematical physics (see, e.g., [15]). A notable discussion of questions concerning the investigation of semi-discrete schemes for evolutionary equations in general Banach spaces can be found in the works by the authors A. Ashyralyev and P. E. Sobolevskii [2, 3], N. Bakaev [4], S. Piskarev [20], S. Piskarev and H. J. Zwart [21].

Important results on the construction and investigation of approximate solution algorithms of evolutionary problems were in particular considered in the well-known books by S. K. Godunov and V. S. Ryabenki [9], G. I. Marchuk [14], R. Richtmayer and K. Morton [22], A. A. Samarski [28], N. N. Yanenko [12]. Mention should also be made of the works by M. Crouzeix [6], M. Crouzeix and P.-A. Raviart [7].

2 Reduction of a purely implicit multi-layer scheme for an evolutionary problem to two-layer schemes

Let us consider the following evolutionary problem in the Hilbert space H:

$$\frac{du(t)}{dt} + Au(t) = f(t), \quad t \in]0, T], \tag{2.1}$$

$$u(0) = u_0, \tag{2.2}$$

where A is the self-adjoint positive definite operator in H with the domain of definition $D(A)$; $f(t)$ is a continuously differentiable function with values in H; u_0 is a given vector from H; $u(t)$ is an unknown function.

In [11], the realization of a purely implicit three-layer semi-discrete scheme is reduced by the perturbation algorithm to the realization of two two-layer schemes.

In [25], the realization of a purely implicit four-layer semi-discrete scheme is reduced by the perturbation algorithm to the realization of three two-layer schemes. On the interval $[0, T]$, let us introduce the grid $t_k = k\tau$, $k = 0, 1, \ldots, n$, with the step $\tau = T/n$. Using the approximation of the first derivative

$$\frac{du}{dt}\bigg|_{t=t_k} = \frac{\frac{11}{6}\,u(t_k) - 3u(t_{k-1}) + \frac{3}{2}\,u(t_{k-2}) - \frac{1}{3}\,u(t_{k-3})}{\tau} + \tau^3 R_k(\tau,u), \quad R_k(\tau,u) \in H,$$

equation (2.1) can be represented at the point $t = t_k$ as

$$\frac{\frac{11}{6}u(t_k) - 3u(t_{k-1}) + \frac{3}{2}u(t_{k-2}) - \frac{1}{3}u(t_{k-3})}{\tau} + Au(t_k) = f(t_k) - \tau^3 R_k(\tau,u), \quad k = 3,\dots,n. \tag{2.3}$$

We rewrite system (2.3) in the form

$$\begin{aligned}\frac{u(t_k) - u(t_{k-1})}{\tau} &+ Au(t_k) \\ &+ \frac{\tau}{2}\Big(\frac{u(t_k) - 2u(t_{k-1}) + u(t_{k-2})}{\tau^2}\Big) + \frac{\tau^2}{3}\Big(\frac{u(t_k) - 3u(t_{k-1}) + 3u(t_{k-2}) - u(t_{k-3})}{\tau^3}\Big) \\ &= f(t_k) - \tau^3 R_k(\tau,u).\end{aligned} \tag{2.4}$$

It is obvious that the expression in brackets in case $\frac{\tau}{2}$ is $u''(t_{k-1}) + \tau^2 R_{1,k-1}$, and the expression in brackets in case $\frac{\tau^2}{3}$ is $u'''(t_k) + \tau R_{2,k}$, $R_{1,k}, R_{2,k} \in H$.

By analogy with the above system, let us consider, in the space H, the one-parametric family of equations

$$\begin{aligned}\frac{u_k - u_{k-1}}{\tau} + Au_k + \frac{\varepsilon}{2}\Big(\frac{u_k - 2u_{k-1} + u_{k-2}}{\tau^2}\Big) + \frac{\varepsilon^2}{3}\Big(\frac{u_k - 3u_{k-1} + 3u_{k-2} - u_{k-3}}{\tau^3}\Big) \\ = f_k + \varepsilon^3 R_k, \quad R_k \in H,\end{aligned} \tag{2.5}$$

where $f_k = f(t_k)$.

Assume that for u_k in H, the expansion in series

$$u_k = \sum_{j=0}^{\infty} \varepsilon^j u_k^{(j)} \tag{2.6}$$

is true. Substituting (2.6) into (2.5) and equating the members of identical powers ε, we obtain the following system of equations:

$$\frac{u_k^{(0)} - u_{k-1}^{(0)}}{\tau} + Au_k^{(0)} = f_k, \quad u_0^{(0)} = u_0, \quad k = 1,\dots,n, \tag{2.7}$$

$$\frac{u_k^{(1)} - u_{k-1}^{(1)}}{\tau} + Au_k^{(1)} = -\frac{1}{2}\,\frac{\Delta^2 u_{k-2}^{(0)}}{\tau^2}\,, \quad k = 2,\dots,n, \tag{2.8}$$

$$\frac{u_k^{(2)} - u_{k-1}^{(2)}}{\tau} + Au_k^{(2)} = -\frac{1}{2}\,\frac{\Delta^2 u_{k-2}^{(1)}}{\tau^2} - \frac{1}{3}\,\frac{\Delta^3 u_{k-3}^{(0)}}{\tau^3}\,, \quad k = 3,\dots,n, \tag{2.9}$$

$$\frac{u_k^{(3)} - u_{k-1}^{(3)}}{\tau} + Au_k^{(3)} = -\frac{1}{2}\,\frac{\Delta^2 u_{k-2}^{(2)}}{\tau^2} - \frac{1}{3}\,\frac{\Delta^3 u_{k-3}^{(1)}}{\tau^3} + R_k,$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

where $\Delta u_{k-1} = u_k - u_{k-1}$.

We introduce the notation

$$v_k = u_k^{(0)} + \tau u_k^{(1)} + \tau^2 u_k^{(2)}, \quad k = 3,\dots,n. \tag{2.10}$$

Let the vector v_k be an approximate value of the exact solution of problem (2.1), (2.2) for $t = t_k$, $u(t_k) \approx v_k$.

Note that in scheme (2.8) the starting vector $u_1^{(1)}$ is defined from the equality $v_1 = u_1^{(0)} + \tau u_1^{(1)}$, where $u_1^{(0)}$ is found by scheme (2.7), and v_1 is an approximate value of $u(t_1)$ with accuracy of $O(\tau^3)$.

In a similar way, the starting vector $u_2^{(2)}$ is defined from the equality $v_2 = u_2^{(0)} + \tau u_2^{(1)} + \tau^2 u_2^{(2)}$, where $u_2^{(0)}$ and $u_2^{(1)}$ are found by schemes (2.7), (2.8), respectively, and v_2 is an approximate value $u(t_2)$ with accuracy of $O(\tau^3)$.

The following theorem is proved in [25].

Theorem 2.1. *Let A be a self-adjoint positive definite operator in H and let the solution of problem (2.1), (2.2) be a smooth enough function. Then, if $\|u(t_k) - v_k\| = O(\tau^3)$, $k = 1, 2$, then the estimate*

$$\|u(t_k) - v_k\| = O(\tau^3), \quad k = 3, \dots, n, \tag{2.11}$$

is true.

Let us proceed to deriving the algorithm of an approximate solution of problem (2.1), (2.2) for the general case, namely, when for the approximation of the first derivative we use the $(m+1)$-layer scheme.

Assume $u(t)$ to be a scalar function. Using the method of undetermined coefficients (see, e.g., [17]), it can be shown that

$$u'(t_k) = c_0 u(t_k) + c_1 u(t_{k-1}) + c_2 u(t_{k-2}) + \dots + c_m u(t_{k-m}) + R_k(\tau, u), \tag{2.12}$$

where $R_k(\tau, u) = O(\tau^m)$,

$$c_0 = \tau^{-1} \sum_{k=1}^{m} (-1)^{k+1} \frac{1}{k} C_m^k, \tag{2.13}$$

$$c_{k-1} = (-1)^{k+1} \frac{m!}{(k-1)(m-k+1)!(k-1)!} \tau^{-1}, \quad k = 2, \dots, m+1. \tag{2.14}$$

We will prove formula (2.12).

The coefficients c_k are to be determined from the assumption that formula (2.12) is exact when $u(t)$ is a polynomial of not higher than m-th degree, and it is this formula that must be exact in each of the following cases:

$$u(t) = 1, \;\; u(t) = (t - t_k), \;\; u(t) = (t - t_k)^2, \;\; u(t) = (t - t_k)^3, \; \dots \;, u(t) = (t - t_k)^m.$$

After inserting these values of $u(t)$ in equation (2.12), we obtain a system of $m+1$ linear equations with respect to c_k $(k = 0, \dots, m)$:

$$\begin{cases} 1 \cdot c_0 + 1 \cdot c_1 + 1 \cdot c_2 + \dots + 1 \cdot c_m = 0 \\ 0 \cdot c_0 + \tau \cdot c_1 + 2\tau \cdot c_2 + \dots + n\tau \cdot c_m = -1 \\ 0 \cdot c_0 + \tau^2 \cdot c_1 + (2\tau)^2 \cdot c_2 + \dots + (n\tau)^2 \cdot c_m = 0 \\ 0 \cdot c_0 + \tau^3 \cdot c_1 + (2\tau)^3 \cdot c_2 + \dots + (n\tau)^3 \cdot c_m = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0 \cdot c_0 + \tau^n \cdot c_1 + (2\tau)^n \cdot c_2 + \dots + (n\tau)^n \cdot c_m = 0, \end{cases}$$

where τ is the grid step $(\tau = t_k - t_{k-1})$.

Rewrite the obtained system in the matrix form

$$Ac = b, \tag{2.15}$$

where

$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & \tau & 2\tau & \dots & m\tau \\ 0 & \tau^2 & (2\tau)^2 & \dots & (m\tau)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \tau^m & (2\tau)^m & \dots & (m\tau)^m \end{pmatrix}, \quad c = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Let us calculate the determinant of the matrix A.

It is easy to prove that

$$|A|=\tau^{\frac{m(m+1)}{2}}F(m), \tag{2.16}$$

where $F(m)=m!\,(m-1)!\,\cdots\,3!\,2!\,1!$.

Indeed, we have that the following equality is valid:

$$|A|=\begin{vmatrix}1 & 1 & 1 & \dots & 1\\ 0 & \tau & 2\tau & \dots & m\tau\\ 0 & \tau^2 & (2\tau)^2 & \dots & (m\tau)^2\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & \tau^m & (2\tau)^m & \dots & (m\tau)^m\end{vmatrix}=\tau^{\frac{m(m+1)}{2}}m!\cdot\begin{vmatrix}1 & 1 & \dots & 1\\ 1 & 2 & \dots & m\\ \vdots & \vdots & \vdots & \vdots\\ 1 & 2^{m-1} & \dots & m^{m-1}\end{vmatrix}.$$

From this equality follows formula (2.16), since the last determinant is the Vandermonde determinant (see, e.g., [13]).

From (2.15), we have $c=A^{-1}b$. In view of the structure of the vector b, the formula

$$c_{k-1}=-\frac{A_{2k}}{|A|}\,,\quad k=1,\dots,m+1, \tag{2.17}$$

is valid, where A_{2k} are algebraic complements of the matrix A.

Thus for determining elements of the vector c, there is no need to determine all the elements of the adjoint matrix A^*, it is enough to determine the values of elements of its second column, i.e., it is enough to calculate the values of the algebraic complements A_{2k}, $k=1,\dots,m+1$.

Let us determine A_{2k}, $k=1,\dots,m$. It is easy to prove the formula

$$A_{21}=-\sum_{k=2}^{m+1}A_{2k}. \tag{2.18}$$

By virtue of formula (2.18), the calculation of the algebraic complement A_{21} reduces to the calculation of the algebraic complements A_{2k}, $k=2,\dots,m+1$.

Note that the structure of the algebraic complements A_{2k}, $k=2,\dots,m$, is the same.

The following formula is valid:

$$A_{2k}=(-1)^k\tau^{\frac{(m+2)(m-1)}{2}}\Big(\frac{m!F(m)}{(k-1)(m-k+1)!(k-1)}\Big)^2\,\frac{F(m-1)}{(k-2)!}\,,\quad k=2,\dots,m+1. \tag{2.19}$$

Indeed, we have

$$A_{2k}=(-1)^k\begin{vmatrix}1 & 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1\\ 0 & \tau^2 & (2\tau)^2 & \dots & ((k-2)\tau)^2 & (k\tau)^2 & ((k+1)\tau)^2 & \dots & (m\tau)^2\\ 0 & \tau^3 & (2\tau)^3 & \dots & ((k-2)\tau)^3 & (k\tau)^3 & ((k+1)\tau)^3 & \dots & (m\tau)^3\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & \tau^m & (2\tau)^m & \dots & ((k-2)\tau)^m & (k\tau)^m & ((k+1)\tau)^m & \dots & (m\tau)^m\end{vmatrix}$$

$$=(-1)^k\begin{vmatrix}\tau^2 & (2\tau)^2 & \dots & ((k-2)\tau)^2 & (k\tau)^2 & ((k+1)\tau)^2 & \dots & (m\tau)^2\\ \tau^3 & (2\tau)^3 & \dots & ((k-2)\tau)^3 & (k\tau)^3 & ((k+1)\tau)^3 & \dots & (m\tau)^3\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \tau^m & (2\tau)^m & \dots & ((k-2)\tau)^m & (k\tau)^m & ((k+1)\tau)^m & \dots & (m\tau)^m\end{vmatrix}$$

$$=(-1)^k\tau^{\frac{(m+2)(m-1)}{2}}\Big(\frac{m!}{k-1}\Big)^2\begin{vmatrix}1 & 1 & \dots & 1 & 1 & 1 & \dots & 1\\ 1 & 2 & \dots & k-2 & k & k+1 & \dots & m\\ 1 & 2^2 & \dots & (k-2)^2 & k^2 & (k+1)^2 & \dots & m^2\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & 2^{m-2} & \dots & (k-2)^{m-2} & k^{m-2} & (k+1)^{m-2} & \dots & m^{m-2}\end{vmatrix}.$$

The last determinant we have obtained is the Vantermonde determinant. Therefore, we finally obtain formula (2.19).

From (2.18) and (2.19), we get

$$A_{21} = -\tau^{\frac{(m+2)(m-1)}{2}} m! \times F(m)\Big(\frac{1}{(m-1)!} - \frac{1}{2^2}\cdot\frac{1}{1!\,(m-2)!} + \frac{1}{3^2}\cdot\frac{1}{2!\,(m-3)!} + \cdots + (-1)^{m+1}\frac{1}{m^2}\cdot\frac{1}{(m-1)!}\Big). \quad (2.20)$$

Formulas (2.20), (2.17), (2.18) and (2.19) yield (2.13), (2.14).

It can be shown that if $u(t)$ is an abstract function and we expand $u(t-\tau), u(t-2\tau), \dots, u(t-m\tau)$ by the Taylor formula, then, applying the method of undetermined coefficients, we obtain a system whose solutions enable us to obtain the same values of coefficients that are given by formulas (2.13), (2.14).

As a result, equation (2.1) takes at the point $t = t_k$ the following form:

$$\frac{a_0 u_k + a_1 u_{k-1} + a_2 u_{k-2} + \cdots + a_m u_{k-m}}{\tau} + Au(t_k) = f(t_k) - \tau^m R_k(\tau, u), \quad (2.21)$$

where $k = m, \dots, n$, $a_i = \tau c_i$, $i = 0, 1, \dots, m$, $R_k(\tau, u) \in H$.

From (2.12), applying Newton's second interpolation formula, we obtain

$$u'(t_k) = \frac{\Delta u_{k-1}}{\tau} + \frac{\tau}{2}\,\frac{\Delta^2 u_{k-2}}{\tau^2} + \frac{\tau^2}{3}\,\frac{\Delta^3 u_{k-3}}{\tau^3} + \cdots + \frac{\tau^{m-1}}{m}\,\frac{\Delta^m u_{k-m}}{\tau^m} + \tau^m R_k(\tau, u), \quad R_k(\tau, u) \in H. \quad (2.22)$$

By virtue of (2.22), equality (2.21) can be rewritten as follows:

$$\frac{u_k - u_{k-1}}{\tau} + Au_k + \sum_{i=2}^{m} \frac{\tau^{i-1}}{i}\,\frac{\Delta^i u_{k-i}}{\tau^i} = f_k - \tau^m R_k(\tau, u). \quad (2.23)$$

Now, applying the perturbation method, from (2.23) we obtain the following algorithm for an approximate solution of problem (2.1), (2.2):

$$\frac{u_k^{(0)} - u_{k-1}^{(0)}}{\tau} + Au_k^{(0)} = f_k, \quad u_0^{(0)} = u_0, \quad k = 1, \dots, n, \quad (2.24)$$

$$\frac{u_k^{(1)} - u_{k-1}^{(1)}}{\tau} + Au_k^{(1)} = -\frac{1}{2}\,\frac{\Delta^2 u_{k-2}^{(0)}}{\tau^2}, \quad k = 2, \dots, n, \quad (2.25)$$

$$\frac{u_k^{(2)} - u_{k-1}^{(2)}}{\tau} + Au_k^{(2)} = -\sum_{i=2}^{3} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(3-i)}}{\tau^i}, \quad k = 3, \dots, n, \quad (2.26)$$

. . . .

$$\frac{u_k^{(p)} - u_{k-1}^{(p)}}{\tau} + Au_k^{(p)} = -\sum_{i=2}^{p+1} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(p+1-i)}}{\tau^i}, \quad k = p+1 \dots, n, \quad (2.27)$$

. . . .

$$\frac{u_k^{(m-1)} - u_{k-1}^{(m-1)}}{\tau} + Au_k^{(m-1)} = -\sum_{i=2}^{m} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(m-i)}}{\tau^i}, \quad k = m, \dots, n, \quad (2.28)$$

$$\frac{u_k^{(m)} - u_{k-1}^{(m)}}{\tau} + Au_k^{(m)} = -\frac{1}{2}\,\frac{\Delta^2 u_{k-2}^{(m-1)}}{\tau^2} - \frac{1}{3}\,\frac{\Delta^3 u_{k-3}^{(m-2)}}{\tau^3} - \cdots - \frac{1}{m}\,\frac{\Delta^m u_{k-m}^{(1)}}{\tau^m} + R_k.$$

An approximate solution of problem (2.1), (2.2) is defined by the formula

$$v_k = \sum_{i=0}^{m-1} \tau^i u_k^{(i)}, \quad k = m, \dots, n. \quad (2.29)$$

Note that in scheme (2.25), the starting vector $u_1^{(1)}$ is defined from the equality $v_1 = u_1^{(0)} + \tau u_1^{(1)}$, where $u_1^{(0)}$ is found by scheme (2.24), and v_1 is an approximate value of $u(t_1)$ with an accuracy of $O(\tau^m)$. In a similar way, the starting vector $u_2^{(2)}$ is defined from the equality $v_2 = u_2^{(0)} + \tau u_2^{(1)} + \tau^2 u_2^{(2)}$, where $u_2^{(0)}$ and $u_2^{(1)}$ are found, respectively, by schemes (2.24), (2.25), while v_2 is an approximate value of $u(t_2)$ with an accuracy of $O(\tau^m)$. Continuing this process of defining the starting vectors for schemes (2.25)–(2.28), the starting vector $u_{m-1}^{(m-1)}$ is defined in a similar way from the equality

$$v_{m-1} = u_{m-1}^{(0)} + \tau u_{m-1}^{(1)} + \tau^2 u_{m-1}^{(2)} + \cdots + \tau^{m-1} u_{m-1}^{(m-1)},$$

where $u_{m-1}^{(0)}, u_{m-1}^{(1)}, \dots, u_{m-1}^{(m-2)}$ are found, respectively, by schemes (2.24)–(2.27), while v_{m-1} is an approximate value of $u(t_{m-1})$ with an accuracy of $O(\tau^m)$.

3 A priori estimate for the approximate solution error

Let us first estimate the residual of the purely implicit $(m+1)$-layer scheme for solutions of schemes (2.24)–(2.29).

Equations (2.25), (2.26), …, (2.27), …, (2.28) are multiplied, respectively, by $\tau, \tau^2, \dots, \tau^p, \dots, \tau^{m-1}$ and the results are added to (2.24). So, we get that v_k is a solution of the following system of equations:

$$\begin{aligned}\frac{v_k - v_{k-1}}{\tau} + Av_k = f_k - \frac{\tau}{2}\,\frac{\Delta^2 u_{k-2}^{(0)}}{\tau^2} - \tau^2 \sum_{i=2}^{3} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(3-i)}}{\tau^i} - \tau^3 \sum_{i=2}^{4} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(4-i)}}{\tau^i} - \cdots \\ - \tau^p \sum_{i=2}^{p+1} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(p+1-i)}}{\tau^i} - \cdots - \tau^{m-1} \sum_{i=2}^{m} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(m-i)}}{\tau^i}, \quad k = m, \dots, n. \qquad (3.1)\end{aligned}$$

This system can be rewritten as follows:

$$\frac{v_k - v_{k-1}}{\tau} + Av_k + \sum_{i=2}^{m} \frac{\tau^{i-1}}{i}\,\frac{\Delta^i v_{k-i}}{\tau^i} = f_k + \widetilde{R}_k(\tau), \qquad (3.2)$$

where

$$\begin{aligned}\widetilde{R}_k(\tau) = \sum_{i=2}^{m} \frac{\tau^{i-1}}{i}\,\frac{\Delta^i v_{k-i}}{\tau^i} - \frac{\tau}{2}\,\frac{\Delta^2 u_{k-2}^{(0)}}{\tau^2} - \tau^2 \sum_{i=2}^{3} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(3-i)}}{\tau^i} \\ - \tau^3 \sum_{i=2}^{4} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(4-i)}}{\tau^i} - \cdots - \tau^p \sum_{i=2}^{p+1} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(p+1-i)}}{\tau^i} - \cdots - \tau^{m-1} \sum_{i=2}^{m} \frac{1}{i}\,\frac{\Delta^i u_{k-i}^{(m-i)}}{\tau^i}.\end{aligned}$$

By virtue of (2.23), equality (3.2) can be represented as

$$\frac{a_0 v_k + a_1 v_{k-1} + a_2 v_{k-2} + \cdots + a_m v_{k-m}}{\tau} + Av_k = f_k + \widetilde{R}_k(\tau), \quad k = m+2, \dots, n. \qquad (3.3)$$

Therefore, $\widetilde{R}_k(\tau)$ is the residual of the purely implicit $(m+1)$ -layer scheme for solutions of schemes (2.24)–(2.29) (see (2.21)).

Taking (2.29) into account, the following is obvious:

$$\begin{aligned}\widetilde{R}_k(\tau) = \sum_{i=0}^{m-1} \frac{\tau^{i+1}}{2}\,\frac{\Delta^2 u_{k-2}^{(i)}}{\tau^2} + \sum_{i=0}^{m-1} \frac{\tau^{i+2}}{3}\,\frac{\Delta^3 u_{k-3}^{(i)}}{\tau^3} + \cdots \\ + \sum_{i=0}^{m-1} \frac{\tau^{i+m-1}}{m}\,\frac{\Delta^m u_{k-m}^{(i)}}{\tau^m} - \frac{\tau}{2}\,\frac{\Delta^2 u_{k-2}^{(0)}}{\tau^2} - \sum_{i=2}^{3} \frac{\tau^2}{i}\,\frac{\Delta^i u_{k-i}^{(3-i)}}{\tau^i} - \sum_{i=2}^{4} \frac{\tau^3}{i}\,\frac{\Delta^i u_{k-i}^{(4-i)}}{\tau^i} - \cdots \\ - \sum_{i=2}^{p+1} \frac{\tau^p}{i}\,\frac{\Delta^i u_{k-i}^{(p+1-i)}}{\tau^i} - \cdots - \sum_{i=2}^{m} \frac{\tau^{m-1}}{i}\,\frac{\Delta^i u_{k-i}^{(m-i)}}{\tau^i},\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=0}^{m-1} \frac{\tau^{i+1}}{2} \frac{\Delta^2 u_{k-2}^{(i)}}{\tau^2} + \sum_{i=0}^{m-1} \frac{\tau^{i+2}}{3} \frac{\Delta^3 u_{k-3}^{(i)}}{\tau^3} + \cdots + \sum_{i=0}^{m-1} \frac{\tau^{i+m-1}}{m} \frac{\Delta^m u_{k-m}^{(i)}}{\tau^m} \\
&\quad - \sum_{j=1}^{m-1} \frac{\tau^j}{2} \frac{\Delta^2 u_{k-2}^{(j-1)}}{\tau^2} - \sum_{j=2}^{m-1} \frac{\tau^j}{3} \frac{\Delta^3 u_{k-3}^{(j-2)}}{\tau^3} - \sum_{j=3}^{m-1} \frac{\tau^j}{4} \frac{\Delta^4 u_{k-4}^{(j-3)}}{\tau^4} - \cdots - \frac{\tau^{m-1}}{m} \frac{\Delta^m u_{k-m}^0}{\tau^m} \\
&= \Big(\sum_{j=1}^{m} \frac{\tau^j}{2} \frac{\Delta^2 u_{k-2}^{(j-1)}}{\tau^2} - \sum_{j=1}^{m-1} \frac{\tau^j}{2} \frac{\Delta^2 u_{k-2}^{(j-1)}}{\tau^2} \Big) + \Big(\sum_{j=2}^{m+1} \frac{\tau^j}{3} \frac{\Delta^3 u_{k-3}^{(j-2)}}{\tau^3} - \sum_{j=2}^{m-1} \frac{\tau^j}{3} \frac{\Delta^3 u_{k-3}^{(j-2)}}{\tau^3} \Big) \\
&\quad + \Big(\sum_{j=3}^{m+2} \frac{\tau^j}{4} \frac{\Delta^4 u_{k-4}^{(j-3)}}{\tau^4} - \sum_{j=3}^{m-1} \frac{\tau^j}{4} \frac{\Delta^4 u_{k-4}^{(j-3)}}{\tau^4} \Big) + \cdots \\
&\qquad + \Big(\sum_{j=m-1}^{2m-2} \frac{\tau^j}{m} \frac{\Delta^m u_{k-m}^{(j-(m-1))}}{\tau^m} - \frac{\tau^{m-1}}{m} \frac{\Delta^m u_{k-m}^{(0)}}{\tau^m} \Big) \\
&= \tau^m \Big(\frac{1}{2} \frac{\Delta^2 u_{k-2}^{(m-1)}}{\tau^2} + \frac{1}{3} \frac{\Delta^3 u_{k-3}^{(m-2)}}{\tau^3} + \frac{\tau}{3} \frac{\Delta^3 u_{k-3}^{(m-1)}}{\tau^3} + \frac{1}{4} \frac{\Delta^4 u_{k-4}^{(m-3)}}{\tau^4} \\
&\quad + \frac{\tau}{4} \frac{\Delta^4 u_{k-4}^{(m-2)}}{\tau^4} + \frac{\tau^2}{4} \frac{\Delta^4 u_{k-4}^{(m-1)}}{\tau^4} + \cdots + \sum_{j=m}^{2m-2} \frac{\tau^{j-m}}{m} \frac{\Delta^m u_{k-m}^{(j-(m-1))}}{\tau^m} \Big).
\end{aligned} \tag{3.4}
$$

Analogously to the proof given in [25], we prove the validity of the estimate

$$\|\widetilde{R}_k(\tau)\| \le c\tau^m, \quad c = const > 0, \quad k = m+2, \ldots, n. \tag{3.5}$$

Furthermore, for the error $z_k = u(t_k) - v_k$, from (2.21) and (3.3) we obtain

$$\frac{a_0 z_k + a_1 z_{k-1} + a_2 z_{k-2} + \cdots + a_m z_{k-m}}{\tau} + A z_k = r_k(\tau), \quad k = m+2, \ldots, n, \tag{3.6}$$

where

$$r_k(\tau) = -(\tau^m R_k(\tau, u) + \widetilde{R}_k(\tau)).$$

Remark 3.1. With (3.5) taken into account, we conclude that if the solution of problem (2.1), (2.2) is smooth enough, then $\|r_k(\tau)\| = O(\tau^m)$.

Since A is a self-adjoint positive-definite operator and $a_0 > 0$, we have that the operator $a_0 I + \tau A$ is continuously invertible and (3.6) implies

$$z_k = -a_1 L z_{k-1} - a_2 L z_{k-2} - \cdots - a_m L z_{k-m} + \tau L r_k(\tau), \quad k = m+2, \ldots, n, \tag{3.7}$$

where $L = (a_0 I + \tau A)^{-1}$.

Let us consider the characteristic equation associated with the differential equation (3.7)

$$\lambda^m - {}_1(s)\lambda^{m-1} - \cdots - {}_{m-1}(s)\lambda - {}_m(s) = 0, \tag{3.8}$$

where $s \in [0, +\infty)$,

$${}_i(s) = a_i(a_0 + s)^{-1}, \quad i = 1, \ldots, m.$$

The following theorem is valid (see [23]).

Theorem 3.1. *Let for any $s \in [0, +\infty)$, the roots of equation (3.8), with the exception perhaps of one, belong to the same circle which lies inside the unit circle, and the excepted root belong to the unit circle. Then the estimate*

$$\|z_{k+2}\| \le c\Big(\sum_{i=2}^{m+1} \|z_i\| + \tau \sum_{i=m}^{k} \|r_{i+2}(\tau)\| \Big), \quad c = const > 0, \tag{3.9}$$

is valid, where $k = m, \ldots, n-2$.

Based on the a priori estimate (3.9), the following theorem is formulated.

Theorem 3.2. *Let A be a self-adjoint positive-definite operator in the space H and let the solution of problem (2.1), (2.2) be a smooth enough function. Then, if $\|u(t_k)-v_k\| = O(\tau^m)$, $k=1,\dots,m+1$, the estimate*

$$\|u(t_k)-v_k\| = O(\tau^m), \quad k=m+2,\dots,n, \tag{3.10}$$

is true.

4 Realization of the perturbation algorithm in the parallel mode

From the standpoint of realization, one of the positive properties of the perturbation algorithm is that it allows us to execute computations in the parallel mode. We mean a possibility to organize the computation process in such a way that in order to obtain the outcome of algorithm realization, certain computations might be performed in the parallel mode, which essentially saves the processing time and speeds up to the output delivery.

In the scientific literature, topics of parallel processing were discussed long before the appearance of appropriate computational technologies. Today, when the existence of cluster computers has become the reality, the simultaneous execution of different parts of the algorithm has become quite a topical issue [5, 8, 10, 29].

Let us consider algorithm (2.24)–(2.29).

Note that to define $u_2^{(1)}$ from (2.25), together with the initial value of $u_1^{(1)}$, we must have the values of $u_0^{(0)}$, $u_1^{(0)}$, $u_2^{(0)}$ (but not the values of all $u_k^{(0)}$, $k=1,\dots,n$). Analogously, to define $u_3^{(1)}$, we must have the values of $u_1^{(0)}$, $u_2^{(0)}$, $u_3^{(0)}$ (but not the values of all $u_k^{(0)}$, $k=1,\dots,n$), and so on. To define $u_n^{(1)}$, we must have the values of $u_{n-2}^{(0)}$, $u_{n-1}^{(0)}$, $u_n^{(0)}$ (but not the values of all $u_k^{(0)}$, $k=1,\dots,n$).

Hence we conclude that after having calculated the value of $u_2^{(0)}$ from (2.24), we can, in a parallel mode with (2.24), start the realization of formula (2.25).

Further, note that to define $u_3^{(2)}$ from (2.26), together with the initial value of $u_2^{(2)}$, we must have the values of $u_1^{(1)}$, $u_2^{(1)}$, $u_3^{(1)}$ and $u_0^{(0)}$, $u_1^{(0)}$, $u_2^{(0)}$, $u_3^{(0)}$ (but not the values of all $u_k^{(1)}$, $k=2,\dots,n$ or $u_k^{(0)}$, $k=1,\dots,n$). Analogously, to define $u_4^{(2)}$, we need to have the values of $u_2^{(1)}$, $u_3^{(1)}$, $u_4^{(1)}$ and $u_1^{(0)}$, $u_2^{(0)}$, $u_3^{(0)}$, $u_4^{(0)}$ (but not the values of all $u_k^{(1)}$, $k=2,\dots,n$ or $u_k^{(0)}$, $k=1,\dots,n$), and so on. To define $u_n^{(2)}$, we need to have the values of $u_{n-2}^{(1)}$, $u_{n-1}^{(1)}$, $u_n^{(1)}$ and $u_{n-3}^{(0)}$, $u_{n-2}^{(0)}$, $u_{n-1}^{(0)}$, $u_n^{(0)}$ (but not the values of all $u_k^{(1)}$, $k=2,\dots,n$ or $u_k^{(0)}$, $k=1,\dots,n$).

Hence we conclude that after calculating the value of $u_3^{(1)}$ from (2.25), we can, in the parallel mode with (2.25), start the realization of formula (2.26).

Continuing an analogous reasoning, we conclude that formulas (2.24)–(2.28) can be realized in the parallel mode (with a delay). Also, note that formula (2.29) is likewise involved in the process of parallelization.

An easy analysis shows that the parallel execution of the perturbation algorithm (2.24)–(2.29) essentially reduces the time of realization of the entire algorithm.

References

[1] V. I. Agoshkov and D. V. Gulua, *A Perturbation Algorithm for the Realization of Finite-Dimensional Approximations of Problems.* (Russian) Preprint, 253. Akad. Nauk SSSR, Otdel Vychisl. Mat., Moscow, 1990.

[2] A. Ashyralyev and P. E. Sobolevskiĭ, *Well-Posedness of Parabolic Difference Equations.* Translated from the Russian by A. Iacob. Operator Theory: Advances and Applications, 69. Birkhäuser Verlag, Basel, 1994.

[3] A. Ashyralyev and P. E. Sobolevskiĭ, *New Difference Schemes for Partial Differential Equations.* Operator Theory: Advances and Applications, 148. Birkhäuser Verlag, Basel, 2004.

[4] N. Yu. Bakaev, *Linear Discrete Parabolic Problems.* North-Holland Mathematics Studies, 203. Elsevier Science B.V., Amsterdam, 2006.

[5] K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations.* Numerical Mathematics and Scientific Computation. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1995.

[6] M. Crouzeix, Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques. (French) *Numer. Math.* **35** (1980), no. 3, 257–276.

[7] M. Crouzeix and P.-A. Raviart, *Approximation des Équations d'évolution Linéaires Par des méthodes à Pas Multiples.* (French) C. R. Acad. Sci. Paris Sér. A-B 28 (1976), no. 6, Aiv, A367–A370.

[8] T. L. Freeman and C. Phillips, *Parallel Numerical Algorithms.* Prentice Hall International Series in Computer Science. Prentice Hall International, New York, 1992.

[9] S. Godounov, *Équations de la Physique Mathématique.* (French) Traduit de russe par Édouard Gloukhian. Éditions Mir, Moscow, 1973.

[10] G. Golub and J. M. Ortega, *Scientific Computing. An Introduction with Parallel Computing.* Academic Press, Inc., Boston, MA, 1993.

[11] D. V. Gulua and J. L. Rogava, On the perturbation algorithm for the semidiscrete scheme for the evolution equation and estimation of the approximate solution error using semigroups. *Comput. Math. Math. Phys.* **56** (2016), no. 7, 1269–1292.

[12] N. N. Janenko, *The Method of Fractional Steps for Solving Multidimensional Problems of Mathematical Physics.* (Russian) Academy of Sciences of the USSR. Siberian Branch. Computing Center Izdat. "Nauka" Sibirsk. Otdel., Novosibirsk, 1967.

[13] A. G. Kurosh, Course in Higher Algebra. (Russian) Eighth edition, reprinted Gosudarstv. Izdat. Fiz.-Mat. Lit., Moscow, 1963.

[14] G. I. Marchuk, *Methods of Computational Mathematics.* (Russian) Izdat. "Nauka" Sibirsk. Otdel., Novosibirsk, 1973.

[15] G. I. Marchuk, V. I. Agoshkov and V. P. Shutyaev, *Adjoint Equations and Perturbation Methods in Nonlinear Problems of Mathematical Physic*s. (Russian) Nauka, Moscow, 1993.

[16] G. I. Marchuk and V. V. Shaĭdurov, *Difference Methods and Their Extrapolations.* Translated from the Russian. Applications of Mathematics (New York), 19. Springer-Verlag, New York, 1983.

[17] W. E. Milne, *Numerical Calculus. Approximations, Interpolation, Finite Differences, Numerical Integration, and Curve Fitting.* Princeton University Press, Princeton, N.J., 1949.

[18] V. Pereyra, On improving an approximate solution of a functional equation by deferred corrections. *Numer. Math.* **8** (1966), 376–391.

[19] V. Pereyra, Accelerating the convergence of discretization algorithms. *SIAM J. Numer. Anal.* **4** (1967), 508–533.

[20] S. Piskarev, *Differential Equations in Banach Space and Their Approximation.* (Russian) Moscow State University Publish House, Moscow, 2005.

[21] S. Piskarev and H. Zwart, Crank–Nicolson scheme for abstract linear systems. *Numer. Funct. Anal. Optim.* **28** (2007), no. 5-6, 717–736.

[22] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems.* Übersetzung der zweiten englischen Auflage von B. M. Budak, A. D. Gorbunov, V. E. Kondrasov und V. E. Trosciev. Herausgegeben von B. M. Budak und A. D. Gorbunov. (Russian) Mir, Moscow, 1972.

[23] Dzh. L. Rogava, *Semidiscrete Schemes for Operator-Differential Equations.* (Russian) Izdatel'stvo "Tekhnicheskogo Universitet", Tbilisi, 1995.

[24] Dzh. L. Rogava and D. V. Gulua, A perturbation algorithm for realizing finite-difference approximation of an abstract evolution problem, and an explicit error estimate of the approximate solution. (Russian) *Dokl. Akad. Nauk* **456** (2014), no. 4, 405–407; translation in *Dokl. Math.* **89** (2014), no. 3, 335–337.

[25] J. Rogava and D. Gulua, The perturbation algorithm for the realization of a four-layer semi-discrete solution scheme of an abstract evolutionary problem. *Georgian Math. J.* **25** (2018), no. 1, 77–92.

[26] J. Rogava and D. Gulua, Reduction of a four-layer semi-discrete scheme for an abstract evolution equation to two-layer schemes and estimation of the approximate solution errors by using associated polynomials. *Bull. Georgian Natl. Acad. Sci. (N.S.)* **15** (2021), no. 2, 23–30.

[27] E. Rothe, Über die Wärmeleitungsgleichung mit nichtkonstanten Koeffizienten im räumlichen Falle. (German) *Math. Ann.* **104** (1931), no. 1, 340–354.

[28] A. A. Samarskiĭ, *Theory of Difference Schemes.* (Russian) Izdat. "Nauka", Moscow, 1977.

[29] V. V. Voevodin and Vl. V. Voevodin, *Parallel Computing.* (Russian) BHV-Petersburg, St. Petersburg, 2002.

(Received 11.05.2022; revised 23.07.2022; accepted 07.09.2022)

Author's address:

Department of Computational Mathematics, Georgian Technical University, 77 M. Kostava Str., Tbilisi 0175, Georgia.

E-mail: d_gulua@gtu.ge