Alternative and expanded version of the sweep method for the numerical solution of the second and third boundary value problem for second-order linear differential equations

Zinepkali Sisengalievich Utemaganbetov, Lyazat Dinishevna Diyarova, Gulmira Nagimovna Nigmetova

Caspian State University of Technologies and Engineering named after Sh.Essenov, 32 mkr., Aktau city, 130000, Republic of Kazakhstan

Abstract. In the article, they suggest a new algorithm which is an alternative to the sweep method for numerical solution of second-order linear differential equations with mixed boundary conditions. This algorithm has a wider field of application than the well-known sweep method, and it works both with positive and negative coefficients. Besides, the authors show the consistency and computational stability of difference schemes represented by suggested recurrence formulas. The results of this article are confirmed by computation data.
[Utemaganbetov Z. S., Diyarova L.D., Nigmetova G.N. **Alternative and expanded version of the sweep method for the numerical solution of the second and third boundary value problem for second-order linear differential equations.** *Life Sci J* 2014;11(8s):315-321] (ISSN:1097-8135). http://www.lifesciencesite.com. 70

Keywords: sweep formulas, forward formulas, backward formulas, forward formulas for a negative "entry", forward formulas for a positive "entry", backward formulas for a negative "entry", and backward formulas for a positive "entry"

Introduction

When such widespread methods as the finite-difference method, the grid-projection method and many others are used for numerical solutions of boundary value problems of differential equations [1, 2, 3, 4 and 5], it leads to the use of the sweep method. Thus, the sweep method occupies an important place among the most commonly used numerical methods.

The sweep method is specifically intended for difference equations that appear while writing difference relations for differential equations. The computational stability of the sweep method is guaranteed if there is a diagonal dominant matrix of difference equation system. In turn, for corresponding differential equations, this characteristic means that the coefficient must be positive for a desired solution. When there is a good computational stability, the sweep methods show themselves as a widely applicable way for the numerical solution of boundary value problems of second-order differential equations. Currently, there are various estimates for stability conditions of the sweep method (up to strong criticism [6]), but, nevertheless, this class of methods positively is received and is one of the main tools for computing specialists, as evidenced description of these methods in educational books.

No doubt, the decisive role belongs to the 50-year practice in the application of the sweep methods to specific problems.

Unfortunately, the rigorous substantiation for the application of such methods leaves much to be desired because there is a significant gap in a set of strict results. For example, in [7] (c. 560-565) one can find a detailed analysis of sweep formulas and a description of difficulties in the closure of a computational algorithm, because in starting point, forward-sweep formulas act as a quantity inverse to grid step.

In various sources, there are many examples where the sweep method does not give unsatisfactory results for boundary value problems. In particular, such examples can be found in [8]-[9]. An unsatisfactory result can occur in the case when all conditions of sweep method applicability are met.

Such unfavourable situation can be caused by the accumulation of computational errors. One can ignore the influence of computational errors on the decision in the calculations with relatively large steps *h*. But it is still worth bearing in mind that computational errors can accumulate while using the sweep method for solving boundary value problems of a difference equation system. It is well-know that if $h \to 0$, a computational error can increase in proportion to $1/h^2$. So, a catastrophic loss of accuracy is possible at quite small values of step *h*. Such unacceptable loss of accuracy occurs due to a significant distortion of the desired value at the stage of working out difference equations ([8], p. 499). That is, the situation is caused by lack of the finite-difference method, but not the sweep method. This fully accords with K.I. Babenko's book [6].

From the above we can conclude that, in the toolbox of computational mathematics, it is necessary to have a series of recurrence formulas similar to sweep formulas but at the same time alternative to classical sweep formulas. Besides, it is desirable that suggested formulas are more computationally stable for a wide range of problems than it is for known types of sweep methods.

This paper is aimed at deriving recurrence formulas similar to sweep formulas for the numerical solution of boundary value problems of second-order differential equations when the sweep method can lead to disappointing results. In particular, the most important question is the presence of sweep formulas while the coefficient is negative in the equation, and boundary conditions do not satisfy the stability conditions of widely used sweep method.

2. Problem definition

Let us consider a second-order differential equation

$$(k(t)y'(t))' - q(t)y(t) = f(t),\ 0 \le t \le 1 \quad (1)$$

with the following boundary conditions

$$k(0)y'(0) - \alpha_0 y(0) = \beta_0 \quad (2)$$

$$k(1)y'(1) - \alpha_1 y(1) = \beta_1 \quad (3)$$

where $\alpha_0, \alpha_1, \beta_0, \beta_1$ – real numbers.

Where $\beta_0, \beta_1 \in \Re = \left]-\infty, +\infty\right[$. Suppose coefficients of equation $f(t), q(t)$ – are continuous on segment $[0,1]$, coefficient $k(t)$ continuously differentiable on $[0,1]$, and $k(t) \ge k_0 > 0$.

In order to study the questions of the numerical solution of this boundary value problem, we will divide segment $[0,1]$ into N equal parts by introduction of nodal points $0 = t_0 < t_1 < \ldots < t_N = 1$. If we mark h as the distance between nodes (the grid spacing), then $h = \frac{1}{N}$, $t_n = \frac{n}{N}$, ($n = 0, 2, \ldots, N$), where N is integer number of segments of the partition (grid spacing might be uneven). Later on we denote by $y(t_n)$ value of exact solution of the boundary value problem (1) - (3) at t_n, and by y_n and y'_n - corresponding to the approximate solution and its derivative constructed by the numerical method under consideration. Also, for convenience, we shall use the notation of the form

$$k(t_n) = k_n,\quad \mu_n = \int_{t_{n-1}}^{t_n} q(t)dt,\ \ \sigma_n = \int_{t_{n-1}}^{t_n} f(t)dt,$$

$$l_n = \int_{t_{n-1}}^{t_n} \frac{dt}{k(t)},\ \ n = 1, 2, \ldots N.$$

It is necessary to derive recurrence sweep formulas of numerical solution of the boundary value problem (1)-(3) and study them for consistency and stability, thereby stating the conditions for the applicability of derived formulas.

3. Recurrence formulas for the numerical solution of boundary value problem (1) – (3), in case when $q(t) \ge 0$, $\alpha_0 > 0$.

Description of the algorithm

In case when $q(t) \ge 0$, $\alpha_0 > 0$, the following recurrence formulas can be used for the numerical solution of boundary value problem (1) – (3):

Forward formula:

$$a_0 = \alpha_0,\ \ a_n = \frac{a_{n-1} + \mu_n}{1 + a_{n-1} l_n}, \quad (4)$$

$$v_0 = \beta_0,\ \ v_n = \frac{v_{n-1} + \sigma_n}{1 + a_{n-1} l_n} \quad (5)$$

For all $n = 1, 2, \ldots, N$.

Backward formula:

$$y_N = \frac{\beta_1 - v_N}{a_N - \alpha_1},$$

$$y_{n-1} = \left(1 - \frac{h a_n}{k_n(1 + a_n l_n)}\right) y_n - \frac{h v_n}{k_n(1 + a_n l_n)}, \quad (6)$$

For all $n = N, N-1, \ldots, 1$, provided that $a_N \ne \alpha_1$.

Consistency proof

In order to prove consistency, we will show that if $h \to 0$, then, from above recurrence formulas (4) - (6) we can get a Cauchy problem for three first-order differential equations. This problem, in its turn, is equal to original boundary value problem (1) – (3).

From formula (4) we get

$$a_n + a_n a_{n-1} l_n = a_{n-1} + \mu_n \text{ or}$$

$$a_n - a_{n-1} = \mu_n - a_n a_{n-1} l_n \ .$$

If we divide both parts of this expression by h and pass to the limit while $h \to 0$, we can get differential equation which is called Ricatti

$$a'(t) + \frac{1}{k(t)} a^2(t) = q(t),\ a(0) = \alpha_0 \quad (7)$$

Reasoning quite similarly, we can become convinced that the following differential equations are the differential analogues for respective recurrence formulas (5) - (6):

$$v'(t)+\frac{1}{k(t)}a(t)v(t)=f(t) \qquad v(0)=\beta_0, \tag{8}$$

$$k(t)y'(t)-a(t)y(t)=v(t) \qquad y(1)=\frac{v(1)-\beta_1}{\alpha_1-a(1)} \tag{9}$$

Provided that $a(1)\neq\alpha_1$, where the latter equation of the system is integrated from right to left.

Justification that the solution $y(t)$ of the resulting differential system is also a solution of boundary value problem (1) - (3) that can be found in the book [10]. There's also conducted some analysis of this system, however, corresponding to them discrete formulas for the numerical solution are not given. This paragraph of this article, in a sense fills this gap.

Stability proof

Now let us receive evidences that the above recurrence formulas are computationally stable. It should be noted that by condition

$$\mu_n=\int_{t_{n-1}}^{t_n} q(t)dt\geq 0,\ \alpha_0>0$$

and hence, as seen from the formula (4) it follows that $a_n\geq 0$, so, inequality $\frac{1}{1+a_{n-1}l_n}\leq 1$ is satisfied for all $n=1,2,\dots N$. This fact ensures the stability of calculation by formulas (4) – (5). In formula (6) if y_n , the factor can be rearranged in the form

$$1-\frac{ha_n}{k_n(1+a_nl_n)}=\frac{k_n(1+a_nl_n)-ha_n}{k_n(1+a_nl_n)}=\frac{k_n\left(1+\frac{ha_n}{k_n}+O(h^2)\right)-ha_n}{k_n(1_n+a_nl_n)}=\frac{1}{1+a_nl_n}+O\left(h^2\right)$$

Since by condition $l_n=\int_{t_{n-1}}^{t_n}\frac{dt}{k(t)}dt\geq 0$ and $a_n\geq 0$, then inequality $\frac{1}{1+a_nl_n}\leq 1$ will be satisfied for all $n=N,N-1,\dots,1$. This guarantees the stability of calculation by backward formula (6). It is notable that the above recurrence formulas (4) - (6) approximate the original boundary value problem with the first order of accuracy. If necessary, recurrence formulas similar to (4) - (6) can be written. These recurrence formulas provide a higher accuracy than the above ones, but this paragraph is aimed at justifying the correctness of formulas (4) - (6) that form a basis for the algorithm of the numerical solution of problem (1) - (3), in case when $q(t)\leq 0$.

The reduction of boundary value problem (1) - (3) to Cauchy problem (7) – (9) and its subsequent solution is called a differential sweep method or simple factorization method. When in equation (1) $q(t)\geq 0$, this method was studied by many authors, such as Gelfand, Lokutsievsky, Marchuk, Ridley etc. Many distinguished mathematicians contributed into the development of the sweep method in relation to different problems. They include: A.A. Abramov, N.S. Bakhvalov, V.S. Vladimirov, A.F. Voyevodin, S.K. Godunov, L.M. Degtyarev, I.D. Safronov etc.

As a result, today there are many modifications of the sweep method, such as: classical, flow, cyclical, orthogonal and non-monotonic modifications. All of them are designed for solving equation systems that appear in the course of the approximation of boundary value problems. Besides, they are modifications of classical sweep method. Each of them can be chosen to solve a specific class of problems.

Numerical examples

As a numerical example, we will consider boundary value problem

$$y''(t)-25y(t)=0,$$
$$0\leq t\leq 1,\ y'(0)-y(0)=1,\ y'(1)-y(1)=1.$$

In the conditions of this example $k(t)\equiv 1$, $q(t)\equiv 25$, $f(t)\equiv 0$ $\alpha_0=\alpha_1=\beta_0=\beta_1=1$. In numerical calculation with step $N=100$, by formulas (4)-(6), the greatest absolute error is $\delta=0.003$.

4. Recurrence formulas for the numerical solution of boundary value problem (1) – (3) in case when $q(t)\leq 0$, $\alpha_0<0$.

Description of the algorithm

Forward algorithm organization.

We will begin calculations with the following formulas calling them forward formulas for a negative "entry"

$$b_n=\frac{b_{n-1}+l_n}{1+b_{n-1}\mu_n},\ b_0=\frac{1}{\alpha_0};\ d_n=\frac{d_{n-1}+b_{n-1}\sigma_n}{1+b_{n-1}\mu_n},$$
$$d_0=\frac{\beta_0}{\alpha_0}; \qquad n=1,\dots\theta_1. \tag{10}$$

where θ_1 is such a number that for all $n=1,\dots,\theta_1-1$, values $b_n\leq 0$, and $b_{\theta_1}>0$. That is, here the value of number n, for which $b_{\theta_1} > 0$, for the first time, is denoted by θ_1 (if there is not such a number θ_1,, then the calculation by these formulas will be conducted to the right end of the segment).

As formulas (10) are designed for a negative "entry", thus we will take $a_{\vartheta_1} = \dfrac{1}{b_{\theta_1}} > 0$, $v_{\theta_1} = \dfrac{d_{\theta_1}}{b_{\vartheta_1}}$, and continue the calculations with the following formulas calling them forward formulas for a positive "entry":

$$a_n = \frac{a_{n-1} + \mu_n}{1 + a_{n-1} l_n},\ a_{\vartheta_1} = \frac{1}{b_{\theta_1}};\ v_n = \frac{v_{n-1} + \sigma_n}{1 + a_{n-1} l_n},$$

$$v_{\theta_1} = \frac{d_{\theta_1}}{b_{\vartheta_1}}; \qquad n = \theta_1 + 1, \ldots, \theta_2. \quad (11)$$

where θ_2, is such a number that for all, values $a_n \geq 0$, and $a_{\theta_2} < 0$. If the input value a_n, for formulas (11) on a number θ_2 are becoming negative, using the relations $b_{\theta_2} = \dfrac{1}{a_{\theta_2}}$, $d_{\theta_2} = \dfrac{v_{\theta_2}}{a_{\theta_2}}$ then we will move to formulas (10). That is to say the calculation will continue by forward formulas for a negative "entry":

$$b_n = \frac{b_{n-1} + l_n}{1 + b_{n-1}\mu_n}, \qquad b_{\theta_2} = \frac{1}{a_{\theta_2}};$$

$$d_n = \frac{d_{n-1} + b_{n-1}\sigma_n}{1 + b_{n-1}\mu_n}, \qquad d_{\theta_2} = \frac{v_{\theta_2}}{a_{\theta_2}};$$

$$n = \theta_2 + 1, \ldots, \theta_3.$$

where θ_3 is such a number for all, values $b_n \leq 0$, и $b_{\theta_3} > 0$ (if there is not such a number, then calculations by formulas θ_3, then the calculation by these formulas will be conducted to the right end of the segment). Then, if needed, the above procedure is repeated in the next possible transition points. Thereby, this method is suitable for many transitions between forward formulas of positive and negative "entries". The number of such transitions depends on the value of function $q(t)$.

If we denote θ_k is the number on which the latter transition from formulas (10) to (11) was performed, or vice versa, then $\{\theta_1, \theta_2, \ldots \theta_k\}$ will be a set of indexes that are "transition step numbers". And accordingly, a set of indexes from 1 to N, is divided into subintervals; $[0, \theta_1]$, $[\theta_1 + 1, \theta_2]$, $[\theta_2 + 1, \theta_3]$, ..., $[\theta_{k-2} + 1, \theta_{k-1}]$, $[\theta_{k-1} + 1, \theta_k]$, $[\theta_k + 1, N]$. In terms of this notation can be argued, that transition from (10) to (11) and back, is done by means of relations $a_{\theta_j} = \dfrac{1}{b_{\theta_j}}$, $v_{\theta_j} = \dfrac{d_{\theta_j}}{b_{\theta_j}}$, where θ_j is the index number from which transition ($j = 1, 2, \ldots k.$) is performed, and j is the number of transition.

So, the alternate use of forward formulas (10) and (11) for negative and positive "entries" allows us to calculate to the right end of the segment and thereby to complete the "forward stroke". At that, on last segment, where it is conducted the calculation of forward formulas, that is on $[\theta_k + 1, N]$, two mutually exclusive cases are possible:

1) Calculations by forward formulas (10) for a negative "entry";
2) Calculations by forward formulas (11) for a positive "entry".

Backward algorithm organization

In the first case, we will set $z_N = \dfrac{\beta_1 - \alpha_1 d_N}{1 - \alpha_1 b_N}$ (provided that $b_N \neq \dfrac{1}{\alpha_1}$) and begin backward calculations by the following formulas calling them backward formulas for a negative "entry".

$$z_{n-1} = \frac{z_n + \mu_n d_{n-1} - \sigma_n}{1 + b_{n-1}\mu_n},\ y_{n-1} = b_{n-1} z_{n-1} - d_{n-1};$$

$$n = N, N-1, \ldots \theta_k + 2,\ \theta_k + 1. \quad (12)$$

Then, beginning with step θ_k, we continue calculations by the following recurrence formulas that can be called a backward formulas for a positive "entry".

$$y_{n-1} = \left(1 - \frac{h a_n}{k_n (1 + a_n l_n)}\right) y_n - \frac{h v_n}{k_n (1 + a_n l_n)};$$

$$n = \theta_k, \ldots \theta_{k-1} + 2,\ \theta_{k-1} + 1. \quad (13)$$

In order to continue calculations for index set $[\theta_{k-1}, \theta_{k-2} + 1]$ $[\theta_{k-1},\ \theta_{k-2} + 1]$, backwards, we need a turn to backward formulas for a negative "entry" (12). The latter two values $y_{\theta_{k-1}+1}$ and $y_{\theta_{k-1}}$ calculated by formula (13), make it possible to find $z_{\theta_{k-1}}$ by formula $z_{\theta_{k-1}} = \dfrac{y_{\theta_{k-1}+1} - y_{\theta_{k-1}}}{hk_{\theta_{k-1}}}$.

Then, for all indexes from $n = \theta_{k-1}$ to $n = \theta_{k-2} + 1$, calculations are conducted by formulas (12). At the next interval $[\theta_{k-2}, \theta_{k-3} + 1]$ calculations are performed from right to left by formulas (13). Thus, by alternating backward formulas for a negative "entry" (12) and backward formulas for a positive "entry" (13), we can find all desired values y_n, $(n = N-1, ..., 1)$. Besides, if it is necessary to turn from (13) to (12), it can be made by formula $z_{\theta_j} = \dfrac{y_{\theta_j+1} - y_{\theta_j}}{hk_{\theta_j}}$, where is the number of index from which transition $(j = k, k-1, ...0.)$, and j is the number of transition.

In the second case, we assume that $y_N = \dfrac{\beta_1 - v_N}{a_N - \alpha_1}$, (provided $a_N \neq \alpha_1$) and calculations continue by formulas (13) that is by backward formulas for a positive "entry" from index N to $\theta_k + 1$. On index θ_k, when it is necessary to turn to (12), we calculate by formula; $z_{\theta_k} = \dfrac{y_{\theta_k+1} - y_{\theta_k}}{hk_{\theta_k}}$, and calculations will continue by the formulas (12) for all indexes of interval $[\theta_k, \theta_{k-1} + 1]$, from right to left. Then we organize this backward numerical process in perfect analogy with the previous case that is we alternate backward formulas for a negative and a positive "entry". This can let us obtain all desired values y_{n-1}, $(n = N, N-1, ..., 1)$.

\Consistency proof

Now we will study the following system on the first-order differential equations:

$$b'(t) + q(t)b^2(t) = \frac{1}{k(t)},\quad b(0) = \frac{1}{\alpha_0} \qquad (14)$$

$$d'(t) + q(t)b(t)d(t) = b(t)f(t),\ d(0) = \frac{\beta_0}{\alpha_0};. \qquad (15)$$

$$z'(t) - q(t)b(t)z(t) = f(t) - q(t)d(t),$$

$$z(1) = \frac{\beta_1 - \alpha_1 d(1)}{1 - \alpha_1 b(1)}. \qquad (16)$$

when $b(1) \neq \dfrac{1}{\alpha_1}$. Here the latter equation of the system is integrated from right to left. If we know the solution for this system, we can write the solution for original boundary value problem in the following form:

$$y(t) = b(t)z(t) - d(t) \qquad (17)$$

It is true that if we differentiate this expression and use equations of the system (14)–(16), we will get $y'(t) = \dfrac{z(t)}{k(t)}$ or $k(t)y'(t) = z(t)$. Thus, we obtain an original equation:

$$(k(t)y'(t))' = z'(t) = f(t) - q(t)d(t) + q(t)b(t)z(t) = f(t) + q(t)\big(b(t)z(t) - d(t)\big) = f(t) + q(t)y(t)$$

Now if we take $b(0) = \dfrac{1}{\alpha_0}$, $d(0) = \dfrac{\beta_0}{\alpha_0}$; then the boundary condition at the left end of the segment is satisfied automatically. In order to determine the initial value for $z(t)$, when $t = 1$ we have $y(t) = b(1)z(1) - d(1)$ и $y'(1) = \dfrac{z(1)}{k(1)}$. Hence, with the boundary conditions (3), if $b(1) \neq \dfrac{1}{\alpha_1}$, we have $z(1) = \dfrac{\beta_1 - \alpha_1 d(1)}{1 - \alpha_1 b(1)}$. Thus, we show that function $y(t) = b(t)z(t) - d(t)$ is a solution for (1)-(3), where $b(t)$, $d(t)$, $z(t)$ are the solutions of differential system (14) – (16). Backwards, from boundary value problem (1) – (3), we get system (14) – (16), as follows.

The desired solution will be sought in the form $y(t) = b(t)z(t) - d(t)$, where $b(t)$, $d(t)$ - yet unknown functions (the sweep coefficients) for which it is necessary to obtain the differential equation. So, if in this equation (1) we take $k(t)y'(t) = z(t)$, then we will get the equation (16). Further, considering just introduced relations and equations (16) we have

$$k(t)y'(t) = k(t)[(b(t)\,z(t) - d(t)]' = k(t)[b'(t)z(t) + b(t)z'(t) - d'(t)] =$$
$$k(t)[k(t)b'(t) + k(t)q(t)b^2(t)]y'(t) + q(t)b(t)d(t) - b(t)f(t) + d'(t)$$

After leading to similar terms we have the equality
$[k(t)b'(t)+k(t)q(t)b^2(t)-1]y'(t)+q(t)b(t)d(t)-b(t)f(t)+d'(t)=0.$

Equating the coefficients to zero by $y'(t)$, and to unity, we obtain two differential equations for the sweep coefficients, i.e. equation (14)-(15). From expression (17) if $t=0$ and left-end boundary condition we get $z(0)(1-\alpha_0 b(0))+\alpha_0 d(0)=\beta_0$. Here, considering $b(0)=\frac{1}{\alpha_0}$, we get $d(0)=\frac{\beta_0}{\alpha_0}$;. The initial value for $z(t)$ is obtained by analogy. Thus, it was shown that the boundary value problem (1) - (3) and a system of differential equations (14) - (16) have the same solution. In system (14) – (16) we can make the following substitution in points where function $b(t)$, do not become zero.

$$a(t)=\frac{1}{b(t)},\quad v(t)=\frac{d(t)}{b(t)},$$

$$y(t)=b(t)z(t)-d(t). \qquad (18)$$

As a result, we come to another system of first-order differential equations:

$$a'(t)+\frac{1}{k(t)}a^2(t)=q(t) \qquad (19)$$

$$v'(t)+\frac{1}{k(t)}a(t)v(t)=f(t) \qquad (20)$$

$$k(t)y'(t)-a(t)y(t)=v(t) \qquad (21)$$

As the previous one, this system is equal to the original boundary value problem (this can be shown similarly to how it was done with system (14)-(16)). Initial values for differential equation system (19)-(21) are determined from relations (18). As it follows from (18), if necessary, we can perform a backward transition from system (19)- (21) to (14)-(16), using relations

$$b(t)=\frac{1}{a(t)},\quad d(t)=\frac{v(t)}{a(t)},\quad z(t)=a(t)y(t)+v(t).$$

Passing to the limit when $h\to 0$, in recurrence formulas $\{b_n, d_n\}$ from (10) we obtain differential equations (14)-(15). Similarly, in the limit when $h\to 0$ in recurrence formulas (12) we obtain equation (16). In the same way, we can become convinced that differential equations represented by system (19) - (21) are the analogues for corresponding recurrence formulas (11), (13). The equivalence that the decision $y(t)$ of each of systems (14)-(16) and (19)-(21) to original boundary value problem (1)-(3) was shown above.

Stability proof

According to the condition $\mu_n=\int_{t_{n-1}}^{t_n} q(t)dt\le 0,$ and to the construction of the algorithm, in formulas for $\{b_n, d_n, z_n\}$ which means that inequality $\frac{1}{1+b_{n-1}\mu_n}\le 1$ is fulfilled for all $b_{n-1}\le 0$. This fact ensures the stability of calculations by formulas (10) and (12). Similarly, the stability of calculations by formula (11) guarantees inequality $\frac{1}{1+a_{n-1}l_n}\le 1$, which is always which is always satisfied by virtue of the fact that according to the problem we have the inequality $l_n=\int_{t_{n-1}}^{t_n}\frac{dt}{k(t)}>0$ and the condition of the algorithm $a_{n-1}\ge 0$. Thus, the condition of stability can be seen in all the formulas directly except formula (13). In formula (13), the factor at y_n can be converted to the form:

$$1-\frac{ha_n}{k_n(a_n+l_n)}=\frac{k_n(1+a_nl_n)-ha_n}{k_n(1+a_nl_n)}=\frac{k_n\left(1+\frac{ha_n}{k_n}\right)-ha_n}{k_n(1+a_nl_n)}+O(h^2)=\frac{1}{1+a_nl_n}+O(h^2)$$

Since, in the construction of the algorithm $a_n\ge 0$, then the inequality $\frac{1}{1+a_nl_n}\le 1$ is performed.

From the above considerations it follows that the above algorithm is correct, if the condition is $b_N\ne\frac{1}{\alpha_1}$ under the first of these possible cases. But the condition $a_N\ne\alpha_1$, guarantees the correctness of the algorithm in the implementation of the second of the cases at the end of the billing interval. If these conditions are not met, then we can start the calculation from the right end of the segment that is to organize the process of "backward sweep".

5. Recurrence formulas for the numerical solution of boundary value problem (1) – (3) in case when $q(t)\le 0$, $\alpha_0>0$.

In this case, the calculation begins by formulas (11) with initial values $a_0=\alpha_0$, $v_0=\beta_0$

and then the process of numerical solution is organized similarly to the previous case, that is when $q(t) \le 0$, $\alpha_0 < 0$, which was described in detail above.

Numerical examples

1. For a numerical example, let us study boundary value problem $y''(t) + 81y(t) = 0$, $0 \le t \le 1$, $y'(0) + 10y(0) = 10$, $y'(1) - y(1) = 1$. In the conditions of this example, $k(t) \equiv 1$, $q(t) \equiv -81$, $f(t) \equiv 0$ $\alpha_0 = -10$, $\alpha_1 = 1$ $\beta_0 = 10$, $\beta_1 = 1$. If we perform numerical calculations with step $N = 100$, then, according to the above algorithm, the greatest absolute value is $\delta = 0.436$. Such low accuracy is caused by the fact that function $q(t)$ and the number of steps N are the values of the same order in this example. Nevertheless, such accuracy does not contradict the first accuracy order guaranteed by the stated method. And if we calculate with step $N = 1000$, the same error is $\delta = 0.058$.

2. For the next numerical example, let us study boundary value problem $y''(t) + 81y(t) = 0$, $0 \le t \le 1$, $y'(0) + 100y(0) = 10$, $y'(1) - y(1) = 1$. In the conditions of this example; $k(t) \equiv 1$, $q(t) \equiv -81$, $f(t) \equiv 0$ $\alpha_0 = 100$, $\alpha_1 = 1$ $\beta_0 = 10$, $\beta_1 = 1$. If we perform numerical calculations with step $N = 100$, then, according to the above algorithm, the greatest absolute error is $\delta = 0.085$. And if we calculate with step $N = 1000$, the same error is $\delta = 0.021$.

6. Conclusion

In this paper authors suggest recurrence formulas for the numerical solution of boundary value problem (1) - (3). These formulas have a wider field of application in solving boundary value problems of second-order differential equations. They work both with positive and negative coefficients $q(t)$. Besides, these formulas can be used with discontinuous coefficients of equations. The results obtained in this article are proved by computational data. These results and ideas of this paper can be generalized for numerical solutions in case, where $q(t)$ is an alternating function, and other kinds of boundary conditions for higher-order differential equations.

After a slight modification method presented here can be used for solving linear partial differential equations. The above algorithm does not directly work in the case, when $q(t) \ge 0$, $\alpha_0 < 0$. In this case, if $\alpha_1 > 0$, then numerical solution process can begin at the right end of the segment using formula (4)-(6). But if condition $\alpha_1 > 0$ is not performed, then foregoing algorithm becomes numerically unstable, and thus its use can lead to disappointing results.

Drawbacks and benefits of this represented method can be clarified on the basis of practical application of this method by specialists in computational mathematics.

Corresponding Author:
Dr. Utemaganbetov Zinepkali Sisengalievich
Caspian State University of Technologies and Engineering named after Sh.Essenov
32 mkr., Aktau city, 130000, Republic of Kazakhstan

References

1. Coddington, E.A., 2012. An Introduction to Ordinary Differential Equations. Dover Publications, pp: 320.
2. Coddington, E.A. and N. Levinson, 1955. Theory of Ordinary Differential Equations. McGraw-Hill, pp: 429.
3. Brauer, F. and J.A. Nohel, 1989. The Qualitative Theory of Ordinary Differential Equations: An Introduction. Dover, pp: 314.
4. Wanner, G. and E. Hairer, 2010. Solving Ordinary Differential Equations II. Springer Berlin Heidelberg, pp: 626.
5. Atkinson, K., W. Han and D. E. Stewart, 2009. Numerical Solution of Ordinary Differential Equations. John Wiley & Sons, pp: 252.
6. Babenko, K.I., 2002. The Fundamentals of Numerical Analysis. Moscow-Izhevsk: “Regular and Chaotic Dynamics”, pp: 586, 588 and 590.
7. Bakhvalov, N.S., 1973. Numerical Methods. Moscow: “Nauka”, pp: 654.
8. Amosov, A.A., Y.A. Dubinskiy and N.V. Kopchenova, 1994. Numerical Methods for Engineers. Moscow: “Vysshaya Shkola”.
9. Ilyin, V.P. and Y.I. Kuznetsov, 1985. Tridiagonal Matrixes and Their Applications. Moscow: “Nauka”.
10. Babushka, I., E. Vitasek and M. Prager, 1969. Numerical Processes of the Solutions of Differential Equations. Moscow: “Mir”, pp: 123-129.

520/2014