Alternative and Expanded Version of the Sweep Method for the Numerical Solution of the First Boundary Value Problem for Second-Order Linear Differential Equations

Zinepkali Sisengalievich Utemaganbetov

Caspian State University of Technologies and Engineering named after Sh. Yessenov, Aktau sity,32 mkr, 130003, Mangistau region, Republic of Kazakstan

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 fixed 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 formulae. The results of this article are confirmed by computation data..

[Utemaganbetov Z.S. **Alternative and Expanded Version of the Sweep Method for the Numerical Solution of the First Boundary Value Problem for Second-Order Linear Differential Equations**. *Life Sci J* 2013;10(12s):603-611] (ISSN:1097-8135). http://www.lifesciencesite.com. 99

Keywords: Sweep formulae, forward formulae, backward formulae, forward formulae for a negative “entry”, forward formulae for a positive “entry”, backward formulae for a negative “entry”, backward formulae for a positive “entry”.

1. 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. That is why 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 me 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]). Nevertheless, this type of methods is one of the main tools for computing specialists. This is confirmed by the fact that these methods are described 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], one can find a detailed analysis of sweep formulae and a description of difficulties in the closure of a computational algorithm because, in starting point, forward-sweep formulae act as a quantity inverse to grid step.

In various sources, there are many examples where the sweep method does not work well 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]. 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 formulae similar to sweep formulae but at the same time alternative to classical sweep formulae. Besides, it is desirable that suggested formulae 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 formulae similar to sweep formulae 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 formulae 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

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

with the following boundary conditions

$$y(0) = \beta_0 \tag{2}$$

$$y(1) = \beta_1 \tag{3}$$

where $\beta_0, \beta_1 \in \Re =]-\infty, +\infty[$.

Suppose coefficients $k(t), f(t), q(t) -$ are continuous on segment $[0, \quad 1]$, $k(t) \ge k_0 > 0$.

In order to study the questions of the numerical solution of this boundary value problem, we will divide segment $[0, \quad 1]$ into N equal parts and then we will mark $h = \frac{1}{N}$, $t_n = \frac{n}{N}$, $t_0 = 0$,

$$t_N = 1, \qquad k(t_n) = k_n. \qquad \mu_n = \int_{t_{n-1}}^{t_n} q(t)dt,$$

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

Instead of exact solution of $y(t)$ we find a certain approximation $\tilde{y} = \tilde{y}(t_n) \approx y(t_n)$, $n = 1, 2, \ldots N-1$. It is necessary to derive recurrence sweep formulae and study them for consistency and stability, thereby stating the conditions for the applicability of derived formulae.

3. Recurrence formulae for the numerical solution of boundary value problem (1) – (3) in case when $q(t) \ge 0$. As it is known, in this case there is a unique solution of boundary value problem (1)-(3).

Description of the algorithm

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

Forward formula:

$$a_0 = 0, \qquad a_n = \frac{a_{n-1} + l_n}{1 + a_{n-1}\mu_n}, \tag{1.1}$$

$$v_0 = \beta_0, \qquad v_n = \frac{v_{n-1} - a_n \sigma_n}{1 + a_{n-1}\mu_n}, \tag{1.2}$$

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

Backward formula:

$$y_N = \beta_1,$$

$$y_{n-1} = \left(1 - \frac{h}{\mathrm{k}_n (a_n + l_n)}\right) y_n + \frac{h v_n}{k_n (a_n + l_n)} \tag{1.3}$$

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

Justification for the above formulae.

If we multiply both parts of the equation (1) by function

$$g(t) = 1 + \frac{1}{a_n} \int_{t_{n-1}}^{t} \frac{d\tau}{k(\tau)} + \int_{t_{n-1}}^{t} \frac{1}{k(\tau)} \left[\int_{t_{n-1}}^{\tau} q(x)g(x)dx \right] d\tau$$

(numbers a_n, will be found later) and if we integrate by parts on one of segments $[t_{n-1}, t_n]$, we will obtain the following expression:

$$k(t)[y'(t)g(t) - y(t)g'(t)] \Big|_{t_{n-1}}^{t_n} + \int_{t_{n-1}}^{t_n} \left[(k(t)g'(t))' - q(t)g(t) \right] y(t)dt = \int_{t_{n-1}}^{t_n} f(t)g(t)dt$$

By direct inspection, on can become convinced that function $g(t)$ - is a solution for homogeneous equation, that is relation $((k(t)g'(t))' - q(t)g(t) = 0$ is correct on segment $[t_{n-1}, t_n]$. Therefore, there is equality

$$k(t)[y'(t)g(t) - y(t)g'(t)] \Big|_{t_{n-1}}^{t_n} = \int_{t_{n-1}}^{t_n} f(t)g(t)dt$$

Hence, after some arithmetic conversions, we obtain

$$\left[k_n y_n - k_n \frac{g(t_n)}{g'(t_n)} y''(t_n) \right] - \frac{g'(t_{n-1})}{g'(t_n)} \left[k_{n-1} y(t_{n-1}) - k_{n-1} \frac{g(t_{n-1})}{g'(t_{n-1})} y'(t_{n-1}) \right] = \frac{\int_{t_{n-1}}^{t_n} f(t)g(t)dt}{-g'(t_n)}$$

Easy to see that

$$g'(t_n) = \frac{1}{k_n} \left(\frac{1}{a_n} + \int_{t_{n-1}}^{t_n} q(t)dt \right) + O(h^2),$$

$$g(t_n) = 1 + \frac{1}{a_n} \int_{t_{n-1}}^{t_n} \frac{dt}{k(t)} + O(h^2)$$

$$g'(t_{n-1}) = \frac{1}{a_n k_{n-1}}, \quad g(t_{n-1}) = 1.$$

Plugging these values in the previous expression we get:

$$\left[k_n y_n - k_n^2 \frac{1 + \frac{1}{a_n}\int_{t_{n-1}}^{t_n} \frac{dt}{k(t)}}{\frac{1}{a_n} + \int_{t_{n-1}}^{t_n} q(t)dt} y'_n\right] - \frac{k_n}{k_{n-1}\left(1 + a_n \int_{t_{n-1}}^{t_n} q(t)dt\right)}\left[k_{n-1}y_{n-1} - a_n k_{n-1}^2 y'_{n-1}\right] =$$

$$= \frac{\int_{t_{n-1}}^{t_n} f(t)g(t)dt}{-\frac{1}{k_n}\left(\frac{1}{a_n} + \int_{t_{n-1}}^{t_n} q(t)dt\right)} + O(h^2).$$

For convenience, we will denote

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

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

$$y(t_n) = y_n, \quad y'(t_n) = y'_n, \quad n = 1, 2, \ldots N.$$

With the help of these denotations, after reducing both parts by k_n, the previous expression will be rewritten in the following more visual form: $\left[y_n - k_n \frac{a_n + l_n}{1 + a_n\mu_n} y'_n\right] - \frac{1}{1 + a_n}\left[y_{n-1} - a_n k_{n-1} y'_{n-1}\right] = -\frac{a_n \sigma_n}{1 + a_n \mu_n} + O(h^2)$. Then for all $n = 1, 2, \ldots N$, we will require that the following conditions are met: $a_n = \frac{a_{n-1} + l_n}{1 + a_{n-1}\mu_n}$.

Then, in view of these recurrence equalities, we will have:

$$\left[y_n - k_n \frac{a_n + l_n}{1 + a_n \mu_n} y'_n\right] - \frac{1}{1 + a_n}\left[y_{n-1} - k_{n-1}\frac{a_{n-1} + l_{n-1}}{1 + a_{n-1}\mu_{n-1}} y'_{n-1}\right] = -\frac{\sigma_n a_n}{1 + a_n \mu_n} + O(h^2)$$

Hence, denoting

$$y_n - k_n \frac{a_n + l_n}{1 + a_n \mu_n} = v_n, \ (n = 1, 2, \ldots N.) \quad (1.4)$$

and dropping second-order remainder terms, we obtain recurrence formula $v_n = \frac{v_{n-1} - a_n \sigma_n}{1 + a_{n-1}\mu_n}$, $n = 1, 2, \ldots N.$. Then, if in formula (1.4) we use approximation $y'_n = \frac{y_n - y_{n-1}}{h} + O(h)$ for First Backward Finite-Divided-Difference at t_n, then, ignoring quantities of order $O(h^2)$, after several arithmetic computations, we obtain the following recurrence formula for the evaluation of desired solution in grid nodes with the first order of approximation

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

$n = N, N-1, \ldots 1.$

By taking $a_0 = 0$, $v_0 = \beta_0$ we will satisfy boundary condition (2) at the left end. And, accordingly, if we take $y_N = \beta_1$, then the boundary condition at the right end will be met automatically. Thereby, we obtained all recurrence formulae (1.1) - (1.3).

Consistency proof. In order to prove consistency, we will show that if $h \to 0$, then, from above recurrence formulae (1.1) - (1.3) 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 (1.1) we get $a_n + a_n a_{n-1}\mu_n = a_{n-1} + l_n$ or $a_n - a_{n_{-1}} = l_n - a_n a_{n-1}\mu_n$. If we divide both parts of this expression by h and pass to the limit while $h \to 0$, we can get differential equation

$a'(t) + q(t)a^2(t) = \frac{1}{k(t)}$, with initial value

$$a(0) = 0 \quad (1.5)$$

Reasoning quite similarly, we can become convinced that the following differential equations are the differential analogues for respective recurrence formulae (1.2) - (1.3):

$$v'(t) + q(t)a(t)v(t) = -a(t)f(t) \quad v(0) = \beta_0, \ (1.6)$$

$$y(t) - a(t)k(t)y'(t) = v(t) \qquad y(1) = \beta_1. (1.7)$$

where the latter equation of the system is integrated from right to left.

The equivalence of received system to boundary value problem (1)-(3) is checked by differentiating the latter equation considering two previous equations. At that, noted initial conditions for $a(t), v(t), y(t)$ ensure the fulfillment of boundary conditions (2)-(3).

Stability proof.

Now let us receive evidences that the above recurrence formulae are computationally stable. It should be noted that by condition $\mu_n = \int_{t_{n-1}}^{t_n} q(t)dt,$ $a_0 = 0$, and it follows that $a_n \geq 0$. That is why, inequality $\frac{1}{1+a_{n-1}\mu_n} \le 1$ is satisfied for all $n = 1,2,\ldots N$. This fact ensures the stability of calculation by formulae (1.1) – (1.2). In formula (1.3), if y_n , the factor can be rearranged in the form

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

Since by condition $l_n = \int_{t_{n-1}}^{t_n} \frac{dt}{k(t)} dt \ge 0$ and $a_n \ge 0$, then inequality $\frac{a_n}{a_n+l_n} \le 1$ will be satisfied for all $n = N, N-1,\ldots 1$. This guarantees the stability of calculation by backward formula (1.3). It is notable that the above recurrence formulae (1.1) - (1.3) approximate the original boundary value problem with the first order of accuracy. If necessary, recurrence formulae similar to (1.1)-(1.3) can be written. These recurrence formulae provide a higher accuracy than the above ones, but this paragraph is aimed at justifying the correctness of formulae (1.1)-(1.3) that form a basis for the algorithm of the numerical solution of problem (1)-(3), in case when $q(t) \le 0$. Moreover, if necessary, some well-known methods, such as the Runge-Kutta method etc., can be used to improve the accuracy of the desired solution. In book [10] one can find a system of differential equations (1.5) – (1.7) and a certain analysis of this system. Nevertheless, the book does not contain corresponding discrete formulae for the numerical solution. This paragraph of the present paper meets this lack.

The reduction of boundary value problem (1) - (3) to Cauchy problem (1.5) – (1.7) and its subsequent solution is called a differential sweep method or simple factorization method. When in equation (1) $q(t) \ge 0$, 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

1. As a numerical example, we will consider boundary value problem $y''(t) - 25y(t) = 0$, $0 \le t \le 1$, $y(0) = 1$, $y(1) = 1$. In the conditions of this example $k(t) \equiv 1$, $q(t) \equiv 25$, $f(t) \equiv 0$ $\beta_0 = 1$, $\beta_1 = 1$. In numerical calculation with step $N = 1000$ by formulae (1.1)-(1.3), the greatest absolute error is $\delta = 0.005$.

2. As the next example, we consider $y''(t) - 100y(t) = 0$, $0 \le t \le 1$, $y(0) = 1$, $y(1) = 1$. Here $k(t) \equiv 1$, $q(t) \equiv 100$, $f(t) \equiv 0$ $\beta_0 = 1$, $\beta_1 = 1$. In numerical calculation with the same step $N = 1000$ by formulae (1.1)-(1.3), the greatest absolute error reaches value $\delta = 0.01$.

3. As the third numerical example, we consider $y''(t) - 10000y(t) = 0$, $0 \le t \le 1$, $y(0) = 1$, $y(1) = 1$. Here $k(t) \equiv 1$, $q(t) \equiv 10\,000$, $f(t) \equiv 0$ $\beta_0 = 1$, $\beta_1 = 1$. In numerical calculation with step $N = 1000$ by formulae (1.1)-(1.3), the greatest absolute error reaches value $\delta = 0.089$.

These and other examples show that if equation coefficient $q(t)$ and/or values β_0, β_1 of end parameters are high, it is necessary to reduce the step for the best accuracy. Here we express the fact that, if values $q(t)$, β_0, β_1 are high, the original problem becomes harder; at the same time, the solution of a problem around ends changes very quickly and forms a “boundary layer” or a “boundary effect”. Inside the segment, the solution changes very slowly, that is it goes to quasi-stationary mode. In similar cases, under this method, it is possible to specify the points where the integration step reduces and increases. But in this paper we will not study this question in depth (it can become the object for further research) because the main goal of this paper is to study the questions of the numerical solution of problem (1)-(3) in case when $q(t) \le 0$.

4. Recurrence formulae for the numerical solution of boundary value problem (1) – (3) in case when $q(t) \le 0$.

Algorithm description

Forward algorithm organization.

We will begin calculations with the following formulae calling them **forward formulae for negative "entry"**

$$b_n = \frac{b_{n-1} + l_n}{1 + b_{n-1}\mu_n}, \quad b_0 = 0; \quad d_n = \frac{d_{n-1} + b_{n-1}\sigma_n}{1 + b_{n-1}\mu_n},$$
$$d_0 = -\beta_0; \qquad (2.8)$$

We obtain $b_1 = l_1 > 0$, and $d_1 = d_0$. (The need to start calculations using these formulae is caused by the fact that the left-end boundary condition is satisfied automatically if there are the above initial values of recurrence formulae).

As formulae (2.8) are designed for negative "entry", we will take $a_1 = \frac{1}{b_1} > 0$, $v_1 = \frac{d_1}{b_1}$, then we begin calculations with the following formulae calling them **forward formulae for positive "entry"**

$$a_n = \frac{a_{n-1} + \mu_n}{1 + a_{n-1}l_n}, \quad a_1 = \frac{1}{b_1}; \quad v_n = \frac{v_{n-1} + \sigma_n}{1 + a_{n-1}l_n},$$
$$v_1 = \frac{d_1}{b_1}; \qquad (2.9) \qquad n = 2,\dots\theta_1.$$

where θ_1, is such a number that for all $n = 2,\dots\theta_1 - 1$, values $a_n \ge 0$, and $a_{\theta_1} < 0$. That is, here the value of number n, for which $a_n < 0$, for the first time, is denoted by θ_1. If there is not such a number, then calculations by formulae (1.9) will be conducted to the right end of the segment.

Then we take $b_{\theta_1} = \frac{1}{a_{\theta_1}} 0$, $d_{\theta_1} = \frac{v_{\theta_1}}{a_{\theta_1}}$ and continue calculations by forward formulae for negative "entry" $b_n = \frac{b_{n-1} + l_n}{1 + b_{n-1}\mu_n}$, $b_{\theta_1} = \frac{1}{a_{\theta_1}}$;

$$d_n = \frac{d_{n-1} + b_{n-1}\sigma_n}{1 + b_{n-1}\mu_n}, \qquad d_{\theta_1} = \frac{v_{\theta_1}}{a_{\theta_1}};$$
$$n = \theta_1 + 1,\dots\theta_2.$$

where θ_2, is such a number that for all $n = \theta_1 + 1,\dots\theta_2 - 1$. values $b_n \le 0$, and $b_{\theta_2} > 0$. If there is not such a number, then the calculations by these formulae will be conducted to the right end of the segment. Then, if needed, the above procedure is repeated in the next possible transition points. This method is suitable for many transitions between forward formulae of positive and negative "entries". The number of such transitions depends on the value of function $q(t)$.

If we denote $\theta_0 = 1$, and θ_k is the number on which the latter transition from formula (2.8) to (2.9) was performed, or vice versa, then $\{\theta_0, \theta_1, \theta_2, \dots \theta_k\}$ will be a set of indexes that are "transition step numbers". And accordingly, a set of indexes from 0 to N, is divided into subintervals in the amount of $k+1$ pieces; $[0, \theta_0]$, $[\theta_0 + 1, \theta_1]$, $[\theta_1 + 1, \theta_2]$,..... $[\theta_{k-2} + 1, \theta_{k-1}]$, $[\theta_{k-1} + 1, \theta_k]$, $[\theta_k + 1, N]$. Then if needed it is possible to turn to forward formulae for positive "entry" (2.9) by substituting $b_{\theta_j} = \frac{1}{a_{\theta_j}}$, $d_{\theta_j} = \frac{v_{\theta_j}}{a_{\theta_j}}$, and back to forward formulae for negative "entry" (2.8) by substituting $a_{\theta_j} = \frac{1}{b_{\theta_j}}$, $v_{\theta_j} = \frac{d_{\theta_j}}{b_{\theta_j}}$, where θ_j is the index number from which transition $j = 0, 1, 2, \dots k$. is performed, and j is the number of transition.

So, the alternate use of forward formulae (2.8) and (2.9) 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 segment $[\theta_k + 1, N]$ two mutually exclusive cases are possible:

1) Calculations by forward formulae (2.8) for negative "entry";

2) Calculations by forward formulae (2.9) for positive "entry".

Backward algorithm organization

In the first case, we will begin backward calculations by the following formulae calling them **backward formulae for negative "entry".**

$$z_N = \frac{\beta_1 + d_N}{b_N}, \quad z_{n-1} = \frac{z_n + \mu_n d_n - \sigma_n}{1 + b_{n-1}\mu_n},$$

$$y_{n-1} = b_{n-1} z_{n-1} - d_{n-1};$$

$$n = N, N-1, \ldots \theta_k + 2, \theta_k + 1. \quad (2.10)$$

if $b_N \neq 0$.

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

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

$$n = \theta_k, \ldots \theta_{k-1} + 2, \theta_{k-1} + 1. \quad (2.11)$$

In order to continue calculations for index set $[\theta_{k-2} + 1, \theta_{k-1}]$ backwards, we need a turn to backward formulae for negative "entry" (2.10). The latter two values $y_{\theta_{k-1}+1}$ and $y_{\theta_{k-1}}$ calculated by formula (2.11) make it possible to find $z_{\theta_{k-1}}$ by formula $z_{\theta_{k-1}} = \frac{y_{\theta_{k-1}+1} - y_{\theta_{k-1}}}{hk_{\theta_{k-1}}}$. Then, for all indexes from θ_{k-1} to $\theta_{k-2} + 1$, calculations are conducted by formulae (2.10). At the next interval $[\theta_{k-3} + 1, \theta_{k-2}]$ calculations are performed from right to left by formulae (2.11). Thus, by alternating backward formula for negative "entry" (2.10) and backward formula for positive "entry" (2.11), we can find all desired values y_n, $(n = N-1, \ldots 1.)$. Besides, if it is necessary to turn from (2.11) to (2.10), it can be made by formula $z_{\theta_j} = \frac{y_{\theta_j+1} - y_{\theta_j}}{hk_{\theta_j}}$, where θ_j is the number of index from which transition $(j = k, k-1, \ldots 0.)$. begins, and j is the number of transition.

In the second case, we assume that $y_N = \beta_1$, and calculations continue by formulae (2.11), that is by backward formulae for positive "entry" from index N to $\theta_k + 1$. On index θ_k, when it is necessary to turn to (2.10), we calculate by formula $z_{\theta_k} = \frac{y_{\theta_k+1} - y_{\theta_k}}{hk_{\theta_k}}$, and calculations will continue by the formula for z_{n-1} from (2.10) for all indexes of interval $[\theta_{k-1} + 1, \theta_k]$ from right to left. Then we organize this backward numerical process in perfect analogy with the previous case, that is we alternate backward formulae for negative and positive "entry". This can let us obtain all desired values y_{n-1}, $(n = N-1, \ldots 1.)$.

Justification for the above recurrence formulae

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

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

(2.12)

$$d'(t) + q(t)b(t)d(t) = b(t)f(t),$$

$$d(0) = -\beta_0.$$

(2.13)

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

$$z(1) = \frac{\beta_1 + d(1)}{b(1)}. \quad (2.14)$$

when $b(1) \neq 0$. 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:

$$y(t) = b(t)z(t) - d(t) \quad (2.15)$$

It is true that if we differentiate this expression and use equations of the system (2.12)–(2.14) we will get $y'(t) = \frac{z(t)}{k(t)}$ or $k(t)y'(t) = z(t)$.

After we differentiate this expression, we get

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

That is we obtain an original equation. Now if we take $b(0) = 0$, $d(0) = -\beta_0$, then the boundary condition at the left end of the segment is satisfied automatically. In order to determine the missing initial value for $z(t)$, we will put down (2.15), when $t = 1$ and taking into account boundary condition $y(1) = \beta_1$. We have $y(1) = b(1)z(1) - d(1) = \beta_1$. Hence, if

$b(1) \neq 0$, we have $z(1) = \dfrac{\beta_1 + d(1)}{b(1)}$. 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 (2.12) – (2.14). Backwards, from boundary value problem (1)-(3), we get system (2,12)-(2,14) as follows. In equation (1) we take $k(t)y'(t) = z(t)$ and $y(t) = b(t)z(t) - d(t)$. After that equation (1) will be rewritten in form (2.14). Considering these equalities and relation

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

, we have

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

.

When $y'(t)$ we make terms of this equation equal and get (2.12), and if we make free terms equal we get (2.13). From expression (2.15), if $t = 0$, and left-end boundary condition we get $y(0) = b(0)z(0) - d(0) = \beta_0$. Here, if we take $b(0) = 0$, we get $d(0) = -\beta_0$. The initial value for $z(t)$ is obtained by analogy. Thus, one can see that boundary value problems (1)-(3) and (2. 12)-(2.15) are equal.

In system (2.12) – (2.14) we can make the following substitution in points where function $b(t)$, do not become zero.

$$a(t) = \frac{1}{b(t)}, \; v(t) = \frac{d(t)}{b(t)}, \; y(t) = b(t)z(t) - d(t). \tag{2.16}$$

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

$$a'(t) + \frac{1}{k(t)}a^2(t) = q(t) \tag{2.17}$$

$$v'(t) + \frac{1}{k(t)}a(t)v(t) = f(t) \tag{2.18}$$

$$k(t)y'(t) - a(t)y(t) = v(t) \tag{2.19}$$

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 (2.12)-(2.14). Initial values for differential equation system (2.17)-(2.19) are determined from relations (2.16). As it follows from (2.16), if necessary we can perform a backward transition from system (2.17)-(2.19) to (2.12)-(2.14), using relations

$$b(t) = \frac{1}{a(t)}, \; d(t) = \frac{v(t)}{a(t)}, \; z(t) = a(t)y(t) + v(t). \tag{2.20}$$

Now let us develop recurrence formulae for approximating the solutions of systems (2.12) – (2.14) and (2.17) – (2.19).

It follows from the first part of this paper (when $q(t) \geq 0$) that recurrence formula for b_n from series (2.8) corresponds to equation (2.12). We will develop the recurrence formula for equation (2.13) as follows: we will multiply equation (2.13) by certain function $g(t) = 1 + \gamma_n(t - t_{n-1})$ (number γ_n will be found later) and integrate by parts on one of segments $[t_{n-1}, t_n]$, ($n = 1,2,\ldots N$). As a result, we obtain

$$d(t)g(t)\Big\|_{t_{n-1}}^{t_n} - \int_{t_{n-1}}^{t_n} d(t)g'(t)dt + \int_{t_{n-1}}^{t_n} g(t)b(t)d(t)q(t)dt = \int_{t_{n-1}}^{t_n} b(t)f(t)g(t)dt$$

It is notable that $g'(t_n) = g'(t_{n-1}) = \gamma_n$, $g(t_n) = 1 + h\gamma_n$. $g(t_{n-1}) = 1$. Then, if we denote $d(t_n) = d_n$, $b(t_n) = b_n$, $\mu_n = \int_{t_{n-1}}^{t_n} q(t)dt$

$\sigma_n = \int_{t_{n-1}}^{t_n} f(t)dt, \quad n = 1,2,\ldots N.$, and take into account a Taylor series expansion in point t_{n-1}, functions under integrals, after some transformations we obtain

$$d_n(1 + h\gamma_n) - d_{n-1}(1 + h\gamma_{n-1} - b_{n-1}\mu_n) = b_{n-1}\sigma_n + O(h^2)$$

Here we will require for all $n = 1,2,\ldots N$ the fulfillment of equalities. $1 + h\gamma_n - b_{n-1}\mu_n = 1 + h\gamma_{n-1}$. Hence we find all values of γ_n in form $\gamma_n = \gamma_{n-1} + b_{n-1}\dfrac{\mu_n}{h}$. Then

$$d_n(1 + h\gamma_n) - d_{n-1}(1 + h\gamma_{n-1}) = b_{n-1}\sigma_n + O(h^2)$$

or

$$d_n = \frac{d_{n-1}(1 + h\gamma_{n-1}) + b_{n-1}\sigma_n}{1 + h\gamma_n}$$

, here we drop the remainder of order $O(h^2)$. Then, if in formula for γ_n we choose $\gamma_{n-1} = 0$, with all $n = 1,2,\ldots N$., we will have $\gamma_n = b_{n-1}\dfrac{\mu_n}{h}$, and considering this expression formula for d_n will look like in (2.8).

In order to develop a recurrence formula corresponding with equation (2.14), we will multiply both parts of this equation by function $g(t) = 1 + \xi_{n-1}(t - t_{n-1})$, (number ξ_n will be found lately) by parts on one of segments $[t_{n-1}, t_n]$. Then we act similarly to how we did above while developing the formula for d_n , and we obtain $z_{n-1} = \frac{z_n + \mu_n d_n - \sigma_n}{1 + b_{n-1}\mu_n}$, which was shown in (2.10). As it follows from relations (2.16), recurrence formulae corresponding with equation of system (2.17)-(2.19) are obtained from formulae $\{b_n, d_n, z_n, y_n\}$ by substituting

$$a_n = \frac{1}{b_n}, \quad v(t) = \frac{d_n}{b_n}, \quad y_n = b_n z_n - d_n ,$$

and they coincide with (2.9) and (2.11). This can be proved by a direct verification.

Consistency proof.

Passing to the limit when $h \to 0$, in recurrence formulae $\{b_n, d_n\}$ from (2.8) we obtain differential equations (2.12)-(2.13). Similarly, in the limit when $h \to 0$, in (2.10) we obtain equation (2.14). In the same way, we can become convinced that differential equations represented by system (2.17) - (2.19) are the analogues for corresponding recurrence formulae (2.9), (2.11). The equivalence of each of systems (2.12)-(2.14) and (2.17)-(2.19) to original boundary value problem (1)-(3) was shown above.

Stability proof.

According to the condition, $\mu_n = \int_{t_{n-1}}^{t_n} q(t)dt \geq 0,$ which means that inequality $\frac{1}{1 + b_{n-1}\mu_n} \leq 1$ is fulfilled for all $b_{n-1} \leq 0$. $n = 1,2,\ldots N$. This fact ensures the stability of calculations by backward formulae for $\{b_n, d_n, z_{n-1}\}$ for negative "entry". And the stability of calculations by backward and forward formulae for positive "entry" - $\{a_n, v_n\}$ ensures inequality $\frac{1}{1 + a_{n-1}l_n} \leq 1$, when $a_{n-1} \geq 0$. So, one can directly see that the condition of stability is met in all formulae except formula for $\{y_{n-1}\}$. In formula for $\{y_{n-1}\}$, if y_n , the factor can be transformed to the form

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

As by condition $l_n = \int_{t_{n-1}}^{t_n} \frac{dt}{k(t)} > 0,$ and $a_n \geq 0$, then inequality $\frac{1}{1 + a_n l_n} \leq 1$ is fulfilled for all $n = N, N-1, \ldots 1$. This guarantees the stability of calculations by the backward formula for positive "entry". As it follows from the above reasoning, **the above algorithm is correct if $b_N \neq 1$.**

Numerical examples

1. For a numerical example, let us study boundary value problem $y''(t) + 49y(t) = 0,$ $0 \leq t \leq 1$, $y(0) = -1$, $y(1) = 0$. In the conditions of this example, $k(t) \equiv 1$, $q(t) \equiv -49$, $f(t) \equiv 0$ $\beta_0 = -1$, $\beta_1 = 0$. If we perform numerical calculations with step $N = 100$, then, according to the above algorithm, the greatest absolute value is $\delta = 0.302$. 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 $\delta = 0.06$.

2. For the next numerical example, let us study boundary value problem $y''(t) + 100y(t) = 0,$ $0 \leq t \leq 1$, $y(0) = -1$, $y(1) = 0$. In the conditions of this example, $k(t) \equiv 1$, $q(t) \equiv -100$, $f(t) \equiv 0$ $\beta_0 = -1$, $\beta_1 = 0$. If we perform numerical calculations with step $N = 100$, then, according to the above algorithm, the greatest absolute error is $\delta = 0.724$. And if we calculate with step $N = 1000$, the same error is $\delta = 0.09$.

5. Conclusion.

In this paper authors suggest recurrence formulae for the numerical solution of boundary value problem (1)-(3). These formulae 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 formulae can be used with discontinuous coefficients of equations. The results obtained in this article are proved by computational data. These results can be generalized for numerical solutions in case $q(t)$ is an alternating function, and other kinds of boundary conditions for higher-order differential equations.

Drawbacks and advantages of the method represented by formulae (1.1)-(1.3) and a series of formulae (2.8)-(2.11) can be clarified on the basis of practical application of this method by specialists in computational mathematics.

Corresponding Author:
Dr.Utemaganbetov
Caspian State University of Technologies and Engineering named after Sh. Yessenov, Aktau sity,32 mkr, 130003, Mangistau region, Republic of Kazakstan

References

1. Earl A. Coddington, 2012. An Introduction to Ordinary Differential Equations. Dover Publications, pp: 320.
2. Earl A. Coddington and Norman Levinson, 1955. Theory of Ordinary Differential Equations. McGraw-Hill, pp: 429.
3. Fred Brauer and John A. Nohel, 1989. The Qualitative Theory of Ordinary Differential Equations: An Introduction. Dover, pp: 314.
4. G. Wanner and E. Hairer, 2010. Solving Ordinary Differential Equations II. Springer Berlin Heidelberg, pp: 626.
5. Kendall Atkinson, Weimin Han and David 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.

12/12/2013