საქართველოს მეცნიერებათა ეროვნული აკადემიის მოამბე, ტ. 18, №3, 2024
BULLETIN OF THE GEORGIAN NATIONAL ACADEMY OF SCIENCES, vol. 18, no. 3, 2024

Mathematics

On One Approach for Constructing Difference Formulas of Any Order of Accuracy for Calculating the Derivative

David Gulua

Department of Computational Mathematics, Georgian Technical University, Tbilisi, Georgia

(Presented by Academy Member Elizbar Nadaraya)

Constructing high-order accurate difference formulas is a crucial problem for obtaining high-precision approximations to continuous problems in mathematical physics. The perturbation algorithm, which is used to implement finite-dimensional approximations of such problems, is designed to achieve certain level of accuracy. To formulate the algorithm in its general form, it is necessary to have a difference formulas in the general representation. This paper examines the method of undetermined coefficients, which provides difference formulas of arbitrary accuracy for the calculation of derivatives. A general form of the formula for multilayer schemes is presented, and some relationships between the coefficients of these formulas are also discussed. *© 2024 Bull. Georg. Natl. Acad. Sci.*

difference formulas, coefficients of difference formulas, perturbation algorithm

The Rothe method [1] (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. In the work [2], the application of the perturbation algorithm to difference schemes for differential equations was considered. Using the perturbation algorithm, we solve the semi-discrete scheme for an approximate solution of the Cauchy problem for an evolutionary equation. By this algorithm, the considered scheme is reduced to two-layer schemes. An approximate solution to the original problem is constructed by means of the solutions of these schemes. Note that the first two-layer scheme gives an approximate solution to an accuracy of first order, whereas the solution of each subsequent scheme refines the preceding solution by one order. In the works [3,4] where a purely implicit three-layer [3] and four-layer [4] semi-discrete scheme for an evolutionary equation is reduced to two- layer [3] and three [4] two-layer schemes and the explicit estimates for the approximate solution error are proved in the Banach [3] and Hilbert [4] spaces under rather general assumptions about the problem data. An approximate solution to the original problem is constructed by solving these two-layer schemes.

The idea of generalizing the perturbation algorithm for multi-layer schemes leads us to the need to obtain a general form of a multi-layer difference scheme for the first derivative.

We should agree with [5] that previously published methods to generate finite difference coefficients (e.g., references [6-10]) have been of considerable complexity. In the paper [5] the simple recursions are derived for calculating coefficients in compact finite difference formulas for any order of derivative and any order of accuracy on one-dimensional grids with arbitrary spacing. However, these algorithms were also quite complex for our purposes.

This paper discusses one of the ways to obtain the coefficients of a multi-layer difference scheme for approximating the first derivative. The type of difference scheme obtained by proposed method will make it possible in the future to obtain a perturbation algorithm for a multi-layer semi-discrete scheme for the approximate the solution to the Cauchy problem for the evolution equation.

Let us assume that $u(t)$ is a sufficiently smooth function. Using the method of undetermined coefficients (see, e.g., [11]), we write the first derivative in the following form

$$\left.\frac{du}{dt}\right|_{t=t_k} = c_0 u\left(t_k\right) + c_1 u\left(t_{k-1}\right) + c_2 u\left(t_{k-2}\right) + \ldots + c_m u\left(t_{k-m}\right) + R_k\left(\tau, u\right), \tag{1}$$

where $R_k(\tau,u) = O(\tau^m)$, τ is the grid step $(\tau = t_k - t_{k-1})$.

The coefficients c_k are to be determined from the assumption that formula (1) is exact when $u(t)$ is a polynomial of not higher than m-th degree, and this formula must be exact in each of the following cases:

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

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

$$\begin{cases} 1\cdot c_0 + 1\cdot c_1 + 1\cdot c_2 + \ldots + 1\cdot c_m = 0 \\ 0\cdot c_0 + \tau\cdot c_1 + 2\tau\cdot c_2 + \ldots + m\tau\cdot c_m = -1 \\ 0\cdot c_0 + \tau^2\cdot c_1 + (2\tau)^2\cdot c_2 + \ldots + (m\tau)^2\cdot c_m = 0 \\ 0\cdot c_0 + \tau^3\cdot c_1 + (2\tau)^3\cdot c_2 + \ldots + (m\tau)^3\cdot c_m = 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0\cdot c_0 + \tau^m\cdot c_1 + (2\tau)^m\cdot c_2 + \ldots + (m\tau)^m\cdot c_m = 0 \end{cases}.$$

Rewrite the obtained system in the matrix-vector form

$$Ac=b,$$

in which

$$A = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & \tau & 2\tau & \ldots & m\tau \\ 0 & \tau^2 & (2\tau)^2 & \ldots & (m\tau)^2 \\ 0 & \tau^3 & (2\tau)^3 & \ldots & (m\tau)^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \tau^m & (2\tau)^m & \ldots & (m\tau)^m \end{pmatrix}, \qquad c = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_m \end{pmatrix}, \qquad b = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

It can be shown that

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

where $F\left(m\right) = m!\left(m-1\right)!\left(m-2\right)!\ldots 2!1!$.

Taking into account that $c = A^{-1}b$ and view of the structure of the vector b, the formula

$$c_{k-1} = -\frac{A_{2k}}{\det(A)}, \qquad k = 1, \ldots, m+1 \tag{3}$$

is valid, where A_{2k} denote algebraic complements of the matrix A. Note that 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 in elements of its second column.

Further it is easy to prove that

$$A_{21} = -\sum_{k=2}^{m+1} A_{2k}. \tag{4}$$

The structure of the algebraic complements A_{2k} $(k = 2, \ldots, m+1)$ is the same and we can show

$$A_{2k} = (-1)^k \tau^{\frac{(m+2)(m-1)}{2}} \frac{F(m)}{(k-1)} C_m^{k-1}, \qquad k = 2, \ldots, m+1, \tag{5}$$

were $C_j^i = \dfrac{j!}{i!(j-i)!}$.

From (4), (5) we have

$$A_{21} = \tau^{\frac{(m+2)(m-1)}{2}} F(m) \sum_{k=2}^{m+1} (-1)^{k+1} \frac{1}{k-1} C_m^{k-1}. \tag{6}$$

Taking into account (2), (3), (5) and (6), for coefficients from (1) we obtain

$$\begin{aligned} c_0 &= \tau^{-1} \sum_{k=1}^{m} (-1)^{k+1} \frac{1}{k} C_m^k, \\ c_k &= (-1)^k \tau^{-1} \frac{1}{k} C_m^k, \qquad k = 1, \ldots, m. \end{aligned} \tag{7}$$

As a result, equation (1) takes the following form

$$\left.\frac{du}{dt}\right|_{t=t_k} = \frac{a_0 u_k + a_1 u_{k-1} + a_2 u_{k-2} + \ldots + a_m u_{k-m}}{\tau} + R_k(\tau, u), \tag{8}$$

were $a_i = \tau c_i$, $u_i = u(t_i)$, $i = 0, \ldots, m.$

As we already mentioned, having equality (8), we can formulate a perturbation algorithm for a general $(m+1)$-layer difference scheme.

Below, we provide the Table of a_i coefficients for some l-layer schemes calculated by (7).

At the end we prove one proposition.

Proposition. Assume a_0^m to be a coefficient of u_k (see (8)) for $(m+1)$-layer difference scheme. Then the equation

$$a_0^{m+1} - a_0^m = \frac{1}{m+1}$$

is valid.

Table. The coefficients of l -layer schemes

l-layer	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
3	$\frac{3}{2}$	-2	$\frac{1}{2}$						
4	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$					
5	$\frac{25}{12}$	-4	1	$-\frac{4}{3}$	$\frac{1}{4}$				
6	$\frac{137}{60}$	-5	5	$-\frac{10}{3}$	$\frac{5}{4}$	$-\frac{1}{5}$			
7	$\frac{49}{20}$	-6	$\frac{15}{2}$	$-\frac{20}{3}$	$\frac{15}{4}$	$-\frac{6}{5}$	$\frac{1}{6}$		
8	$\frac{25929}{10000}$	-7	$\frac{21}{2}$	$-\frac{116667}{10000}$	$\frac{35}{4}$	$-\frac{21}{5}$	$\frac{11667}{10000}$	$-\frac{1429}{10000}$	
9	$\frac{27179}{10000}$	-8	14	$-\frac{186667}{10000}$	$\frac{35}{2}$	$-\frac{56}{5}$	$\frac{46667}{10000}$	$-\frac{11429}{10000}$	$\frac{1}{8}$

Actually from (7) we have

$$\begin{aligned} a_0^{m+1} - a_0^m &= \sum_{k=1}^{m+1} (-1)^{k+1} \frac{1}{k} C_{m+1}^k - \sum_{k=1}^{m} (-1)^{k+1} \frac{1}{k} C_m^k \\ &= \sum_{k=1}^{m} (-1)^{k+1} \frac{1}{k} (C_{m+1}^k - C_m^k) + (-1)^{m+2} \frac{1}{m+1} C_{m+1}^{m+1} \\ &= \sum_{k=1}^{m} (-1)^{k+1} \frac{1}{k} C_m^{k-1} + (-1)^{m+2} \frac{1}{m+1} = \sum_{k=1}^{m+1} (-1)^{k+1} \frac{1}{k} C_m^{k-1}. \end{aligned} \tag{9}$$

If we denote $n = k-1$ then from (9) we have

$$a_0^{m+1} - a_0^m = \sum_{n=0}^{m} (-1)^n \frac{1}{n+1} C_m^n. \tag{10}$$

It is obvious that

$$\frac{1}{n+1} = \int_0^1 x^n dx.$$

Then from (10) we have

$$a_0^{m+1} - a_0^m = \sum_{n=0}^{m} (-1)^n C_m^n \int_0^1 x^n dx = \int_0^1 \sum_{n=0}^{m} (-1)^n C_m^n x^n dx = \int_0^1 (1-x)^m dx = \frac{1}{m+1}.$$

The Proposition is proved.

Consequence. If a_0^m is a coefficient of u_k (see (8)) for $(m+1)$-layer difference scheme. Then a_0^m can be presented as the following sum

$$a_0^m = \sum_{i=0}^{m} \frac{1}{i+1}.$$

მათემატიკა

წარმოებულის გამოსათვლელი ნებისმიერი რიგის სიზუსტის სხვაობიანი ფორმულების აგების ერთი მიდგომის შესახებ

დ. გულუა

საქართველოს ტექნიკური უნივერსიტეტი, გამოთვლითი მათემატიკის დეპარტამენტი, თბილისი, საქართველო

(წარმოდგენილია აკადემიის წევრის ე. ნადარაიას მიერ)

მაღალი რიგის სიზუსტის სხვაობიანი ფორმულების აგება მნიშვნელოვანი ამოცანაა მათემატიკური ფიზიკის უწყვეტი ამოცანების მაღალი სიზუსტის მიახლოების მისაღებად. შეშფოთებათა ალგორითმი, მათემატიკური ფიზიკის ამოცანების სასრულგანზომილებიანი აპროქსიმაციის რეალიზაციისათვის, გვაძლევს ამოცანის აპროქსიმაციას გარკვეული სიზუსტით. ალგორითმის ზოგადი სახით ფორმულირებისათვის საჭიროა სხვაობიანი ფორმულების წარმოდგენა ზოგადი სახით. ნაშრომში განხილულია უცნობ კოეფიციენტთა მეთოდი, რომელიც იძლევა წარმოებულისთვის ნებისმიერი სიზუსტის სხვაობიანი ფორმულის მიღების საშუალებას. მრავალშრიანი სქემისთვის ჩაწერილია ზოგადი ფორმულა, აგრეთვე მოყვანილია ზოგიერთი ტოლობა ფორმულის კოეფიციენტებს შორის.

REFERENCES

1. Rothe E. (1931) Über die Wärmeleitungsgleichung mit nichtkonstanten Koeffizienten im räumlichen Falle. *Math. Ann*., **104(**1): 340–354 (German).
2. Agoshkov V. I., Gulua D. V. (1990) Algoritm vozmushenii dlia realizatsii konechnomernykh approksimatsii zadach. Preprint, 253. Akad. Nauk SSSR, otdel vychisl. mat.: 35p. (in Russian).
3. Gulua D. V., Rogava J. L. (2016) 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(**7): 1269–1292.
4. Rogava J., Gulua D. (2018) The perturbation algorithm for the realization of a four-layer semidiscrete solution scheme of an abstract evolutionary problem. *Georgian Math. J*., **25**(1): 77-92.
5. Fornberg B. (1988) Generation of finite difference Fformulas on arbitrarily spaced grids, *Mathematics of Computation*, **51**(184): 699-706.
6. Abramowitz M., Stegun I. A. (1964) Handbook of mathematical functions, National Bureau of Standards, *Appl. Math. Series*, 55: 470p. Washington, D.C.
7. Cynar S. J. (1987) Using Gaussian elimination for computation of the central difference equation coefficients, *Signum News* (USA), 22: 12-19.
8. Fornberg B. (1975) On a Fourier method for the integration of hyperbolic equations, *SIAM J. Numer. Anal.*, 12: 509-528.
9. Keller H. B., Pereyra V. (1978) Symbolic generation of finite difference formulas, *Math. Comp.,* 32: 955-971.
10. Lakin W. D. (1986) Differentiating matrices for arbitrarily spaced grid points, *Internat. J. Numer. Methods Engrg.*, 23: 209-218.
11. Milne W. E. (1949) Numerical calculus. approximations, interpolation, finite differences, numerical integration, and curve fitting. Princeton University Press, Princeton, 408p.

Received August, 2024