ARTUR KROWIAK*

DETERMINATION OF THE WEIGHTING COEFFICIENTS FOR DIFFERENTIAL QUADRATURE METHOD BASED ON SPLINE INTERPOLATION

WYZNACZANIE WSPÓŁCZYNNIKÓW WAGOWYCH W METODZIE KWADRATÓR RÓŻNICZKOWYCH BAZUJĄCEJ NA FUNKCJACH SKLEJANYCH

Abstract

The paper deals with the methodology of the determination of the weighting coefficients for differential quadrature method based on spline interpolation. Appropriate formulas are derived and two practical approaches to determine mentioned coefficients are proposed, one – pure numeric, the other that uses symbolic-numeric programming. Both approaches are analyzed on account of efficiency, conditioning of the problem and easiness of the implementation.

Keywords: *differential quadrature method*, *spline interpolation*, *weighting coefficients*

Streszczenie

W artykule omówiono metodykę wyznaczania współczynników wagowych dla metody kwadratur różniczkowych bazującej na interpolacji funkcjami sklejanymi. Wyprowadzono związki na wspomniane współczynniki oraz zaproponowano dwa sposoby praktycznej realizacji tego zagadnienia, jeden – czysto numeryczny, drugi – wykorzystujący programowanie symboliczno-numeryczne. Oba sposoby przeanalizowano pod kątem efektywności obliczeń, uwarunkowania zagadnienia oraz łatwości implementacji.

Słowa kluczowe: *metoda kwadratur różniczkowych*, *funkcje sklejane*, *współczynniki wagowe*

* PhD. Eng. Artur Krowiak, Institute of Applied Informatics, Faculty of Mechanical Engineering, Cracow University of Technology.

1. Introduction

The differential quadrature method (DQM) is an efficient numerical tool for solving differential equations. One can achieve very accurate results of the considered problem with a little computational effort. The DQM is characterized by simple formulation, which allows to implement even complex mathematical problems without having much programming skills. These features make the method very useful in solving various scientific or engineering problems [1].

The main idea of the DQM is to descritize spatial derivatives in the governing equation by a linear weighted sum of all function values $f(x_i)$ from the domain, which can be put as

$$\frac{d^r f(x)}{dx^r}\Big|_{x=x_i} = \sum_{j=1}^{N} a_j^{(r)}(x_i) f(x_j) = \sum_{j=1}^{N} a_{ij}^{(r)} f_j \quad i = 1,\dots,N \tag{1}$$

where N denotes the number of nodes x_i and $a_{ij}^{(r)}$ are the weighting coefficients for the rth order derivative. In such a way the differential equation and associated boundary or initial conditions can be written in the form of algebraic equations. The key stage of the method is to determine the weighting coefficients. To this end some algebraic expressions, based on the Lagrange interpolation formula, have been derived [2, 3]. This approach turns out to be very efficient since no system of equations has to be solved to obtain these coefficients. However, the use of these coefficients to problem discretization may lead to an unstable solution, especially in dynamics problems [4, 5]. To overcome this drawback, the spline interpolation has been introduced instead of Lagrange polynomial [6]. In this approach the function sought is approximated in the following way

$$f(x) \approx \{ s_i(x), \, x \in [x_i, x_{i+1}], \; i = 1,\dots,N-1 \tag{2}$$

where the $s_i(x)$ spline segment of n-th degree is defined by the formula

$$s_i(x) = \sum_{j=0}^{n} c_{ij}(x - x_i)^j \tag{3}$$

In Equation (3) c_{ij} denotes the spline coefficients, which are calculated using the interpolation conditions (4) as well as derivative continuity conditions (5) and the so called natural end conditions.

$$s_i(x_i) = f_i, \quad s_i(x_{i+1}) = f_{i+1}, \quad i = 1,\dots,N-1 \tag{4}$$

$$s_i^{(k)}(x_{i+1}) = s_{i+1}^{(k)}(x_{i+1}), \quad i = 1,\dots,N-2, \quad k = 1,\dots,n-1 \tag{5}$$

$$s_1^{(k)}(x_1) = 0, \quad s_{N-1}^{(k)}(x_N) = 0, \quad k = \frac{n+1}{2},\dots,n-1 \tag{6}$$

Although one can use an odd or even degree of the spline function, the formulas presented in this paper are limited to odd spline degrees. Similar formulas can be easily derived for an even spline degree.

In the present paper the details of the calculation of the weighting coefficients for spline-based DQM are presented. Two algorithms are shown and compared with respect to their efficiency, conditioning of the problem and easiness of the implementation.

2. Weighting coefficients

Spline coefficients from Equation are linear combinations of the function sought

$$c_{ij} = \sum_{k=1}^{N} C_{ijk} \cdot f_k, \quad i = 1,...,N-1, j = 0,...,n \tag{7}$$

where the values of C_{ijk} depend on the node distribution. In order to obtain the weighting coefficients for the DQM, an appropriate order derivative of the interpolation function (2) has to be calculated and evaluated at each node of the domain, what can be expressed by the formula

$$f^{(r)}(x_i) \approx \left\{ s_i^{(r)}(x_i), \, i = 1,\ldots,N-1, s_{N-1}^{(r)}(x_N) \right\} \tag{8}$$

where

$$s_i^{(r)}(x_i) = c_{ir} \cdot r!, \quad i = 1,...,N-1, \; s_{N-1}^{(r)}(x_N) = \sum_{j=r}^{n} \left(c_{N-1j} \cdot (x_N - x_{N-1})^{j-r} \cdot \frac{j!}{(j-r)!} \right) \tag{9}$$

Introducing Equation (7) into Equation (9), after simple algebraic transformations one obtains

$$s_i^{(r)}(x_i) = \sum_{k=1}^{N} \left[C_{irk} \cdot r! \right] f_k, \quad i = 1,...,N-1,$$

$$s_{N-1}^{(r)}(x_N) = \sum_{k=1}^{N} \left[\sum_{j=r}^{n} \left(C_{N-1jk} \cdot (x_N - x_{N-1})^{j-r} \cdot \frac{j!}{(j-r)!} \right) \right] f_k \tag{10}$$

Comparing Equation (10) with the main formula of the DQM (1) it is easy to notice that the expressions enclosed in square brackets are the weighting coefficients for the DQM based on spline interpolation

$$a_{ik}^{(r)} = C_{irk} \cdot r!, \quad i = 1,...,N-1, \, k = 1,...,N$$

$$a_{Nk}^{(r)} = \sum_{j=r}^{n} \left(C_{N-1jk} \cdot (x_N - x_{N-1})^{j-r} \cdot \frac{j!}{(j-r)!} \right), \quad k = 1,...,N \tag{11}$$

Due to the assumed approximation (2), the determination of the weighting coefficients is possible, when the coefficients C_{ijk} are computed. Unfortunately, this process requires to solve the system of equations mentioned in section 1 what is an inconvenience compared to classical DQM. However, this inconvenience is compensated by more versatility of the method based on spline functions.

3. Practical approaches to determining the weighting coefficients

In order to compute the weighting coefficients given by Equation (11) the set of equations for coefficients C_{ijk} has to be formulated and solved. Two approaches are proposed to this end.

3.1. Numeric approach

Introducing Equation (7) into interpolation conditions (4) and comparing the terms standing next to the appropriate values of f_i one obtains

$$C_{i0k} = \delta_{ik}, \quad i = 1,...,N-1, \, k = 1,...,N$$

$$\sum_{j=0}^{n} C_{ijk} \cdot (x_{i+1} - x_i)^j = \delta_{ik-1}, \quad i = 1,...,N-1, \, k = 1,...,N \tag{12}$$

where δ_{ik} is the Kronecker symbol.

Taking into account Equation (7) in derivative continuity conditions (5), the following equations can be derived

$$\sum_{j=r}^{n} C_{ijk} \cdot (x_{i+1} - x_i)^{j-r} \cdot \frac{j!}{(j-r)!} = r! \cdot C_{i+1rk}, \quad i = 1,...,N-2, \, k = 1,...,N, r = 1,...,n-1 \tag{13}$$

To complete the set of equations, expression (7) should be introduced into the natural end conditions (6), what yields

$$C_{1rk} = 0, \quad r = \frac{n+1}{2},...,n-1, \, k = 1,...,N$$

$$\sum_{j=r}^{n} C_{N-1jk} \cdot (x_N - x_{N-1})^{j-r} \cdot \frac{j!}{(j-r)!} = 0, \quad r = \frac{n+1}{2},...,n-1, \, k = 1,...,N \tag{14}$$

Expressions (12)–(14) constitute the system of $NE = (N-1) \cdot (n+1) \cdot N$ equations for C_{ijk}. Once this system is solved, the weighting coefficients can be computed using formula(11). Although this system is characterized by a sparse matrix and some coefficients C_{ijk} in (12)–(14) are directly defined, the solution of the system can be troublesome due to weak conditioning. This problem is illustrated in Fig. 1. To overcome the inconvenience, another approach that takes advantage of symbolic-numeric programing is proposed.

3.2. Symbolic-numeric approach

This approach requires the following steps:
- Definition of the interpolation conditions as well as derivative continuity conditions and the end conditions (4)–(6), where unknown function values f_i are noted as symbols.
- Solution of the defined set of equations for the interpolation coefficients c_{ij}. The coefficients obtained are in the form of symbolic-numeric terms given by (7).
- Determination of the numeric values C_{ijk} in expression (7) by separating these numbers from appropriate symbols f_i. This step is easy to implement using some tools provided by Computer Algebra Systems (CAS).
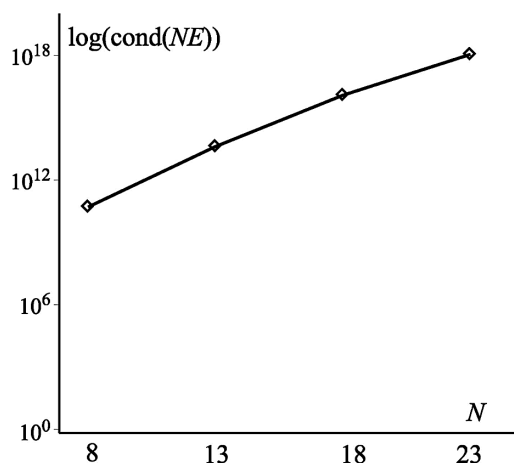
Fig. 1. Condition number of the system of equations (spline degree $n = 9$) versus number of nodes
Rys. 1. Wskaźnik uwarunkowania układu równań (stopień funkcji sklejanej $n = 9$) w zależności od liczby węzłów

Once the C_{ijk} coefficients are obtained, the weighting coefficients can be computed using formula (11). It is well known that the symbolic manipulations are less efficient than numeric ones, but the algorithm presented above requires far fewer equations to solve then the approach described in section 2.1. Moreover, the possibilities of using exact arithmetic provided by CAS eliminate the ill-conditioning problem that arises in the pure numeric approach. Therefore the algorithm presented here seems to be a more convenient way to determine the weighting coefficients for the DQM based on spline interpolation.

3.3. Some notes on efficiency

In order to show the efficiency of the algorithms presented in the previous subsections some numerical experiments have been carried out. The algorithms have been implemented in the Maple programming language. The reason of the choice of this environment is the broad possibilities of the symbolic and numeric manipulations. The CPU (AMD Phenom II X4 810, 2.6 GHz) time of the computation of the C_{ijk} coefficients, which are crucial for the calculation of the weighting coefficients $a_{ij}^{(r)}$, has been determined. The results are presented in Table 1.

Results in table 1 indicate that the symbolic-numeric approach is quite good alternative for the determination of the weighting coefficients for the DQM based on spline functions. When the system of equations is large enough, this algorithm works faster than the pure numeric one. The reason lies in the fact that this algorithm requires far fewer equations to solve then pure numeric one. Moreover, as mentioned earlier the use of the exact arithmetic in the symbolic-numeric approach overcomes the problem of the ill-conditioning that arises during pure numeric computations.

CPU time of the computation of the C_{ijk} coefficients

Spline degree (n), Number of nodes (N)	CPU Time [s]	
	Numeric approach	Symbolic-numeric approach
$n = 7$		
$N = 8$	0.078	0.156
$N = 13$	0.577	0.500
$N = 18$	3.541	0.936
$n = 9$		
$N = 8$	0.125	0.234
$N = 13$	1.076	0.827
$N = 18$	6.630	2.574
$n = 11$		
$N = 8$	0.156	0.327
$N = 13$	1.638	1.498
$N = 18$	11.060	4.274

4. Conclusions

The DQM based on spline interpolation is an alternative to the classical DQM, especially in the case of dynamics problems, where the latter is not a reliable tool due to its instability. In the paper two algorithms to determine the weighting coefficients for spline-based DQM have been presented. The discussion on the efficiency and simplicity of these approaches has been carried out. One can conclude that the symbolic-numeric algorithm is better way to determine of the weighting coefficients than pure numeric one.

R e f e r e n c e s

[1] Bert C.W., Malik M., *Differential quadrature method in computational mechanics*, Applied Mechanics Review, 49, 1996, 1-28.
[2] Shu C., Richards B.E., *Application of generalized differential quadrature to solve two-dimensional incompressible Nanier-Stokes equations*, International Journal for Numerical Methods in Fluids, 15, 1992, 791-798.
[3] Shu C., *Differential quadrature and its application in engineering*, Springer-Verlag, London, 2000.
[4] Krowiak A., *The application of the differential quadrature method based on a piecewise polynomial to the vibration analysis of geometrically nonlinear beams*, Computer Assisted Mechanics and Engineering Sciences, 15, 2008, 1-13.
[5] Zong Z., *A variable order approach to improve differential quadrature accuracy in dynamic analysis*, Journal of Sound and Vibration, 266, 2003, 307-323.
[6] Krowiak A., *Symbolic computing in spline-based differential quadrature method*, Communications in Numerical Methods in Engineering, 22, 2006, 1097-1107.