International Journal of Pure and Applied Mathematics
Volume 99 No. 2 2015, 153-163
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: http://dx.doi.org/10.12732/ijpam.v99i2.3

SOFTWARE-READY CUBIC SPLINE COLLOCATION OF DIRICHLET BOUNDARY VALUE PROBLEM FOR FIRST ORDER DIFFERENTIAL EQUATIONS WITH IMPULSE EFFECT

V.I. Donev

Department of Mathematics
Technical University of Sliven
Sliven, 8800, BULGARIA

Abstract: A method of cubic spline collocation for construction of approximate solution of first order differential equation with impulse effect under Dirichlet's boundary value conditions is considered. Software-ready realization of the method is proposed.

AMS Subject Classification: 34B37, 65D07, 65L10, 68N19
Key Words: collocation, impulse effect, Dirichlet's boundary value condition

1. Introduction

The collocation method is a powerful universal tool for constructing approximate solutions of a wide class linear and nonlinear two-point boundary value problems for differential equations. Theoretical and practical interest is the possibility of applying this method in solving boundary value problems for ordinary differential equations with impulse effect. Such equations are adequate

Received: September 12, 2014

mathematical apparatus for simulating various processes of biology, medicine, physics, etc.. The impulse effect reflects on the discontinuities of first kind and can be used to model mathematically the short-time disturbances of these processes. The durations of the disturbances are often negligible, compared to the total duration of the process and can be treated in form of impulses, which act instantly. First publications in the area of differential equations with impulse effects are considered to be the works of V.Milman [3] and A.Mishkis [4].

Recently, the interest in the subject of impulsive differential equations is growing because of the numerous applications of these equations in mathematical control theory. Natural interest causes and the development of numerical methods for finding their approximate solutions.

In [6], the authors built a solution of a non-homogeneous boundary value problem by the method of collocation with algebraic polynomials of degree $m, m \quad \to \infty$, using the Green function of a homogeneous linear impulse boundary value problem. Note that the realization of the collocation process with polynomials is very complex and with not fully satisfactory properties of the approximation. Therefore, the method is purely theoretical in nature. In practice, it has shifted from finite difference schemes.

The method of spline-collocation, based on approximation with splines, allows us to build algorithms (ready for software realization), where the numerical computation is not more complicated than the realization of the finite difference schemes.

The invasion of the splines in the approximation theory was due to the interpolation problems and thanks to their good computational and approximation properties. Undoubtedly, the splines have extremely good approximation properties and versatility. They provide an easy implementation of the computational algorithms built on their basis. At the same time algorithms for constructing spline functions coincide with the finite element method, which is the main industrial method of structural analysis in computer-aided design.

In the present paper a method of cubic spline collocation for construction of approximate solution of first order differential equation with impulse effect under Dirichlet's boundary value conditions is considered. The apparatus of B-splines is used. The advantage of this approach is the understanding that approximate solution of the differential equation in the form of a spline can be obtained over the entire area of the definition of the problem, whereas the solution obtained by finite difference schemes is calculated only on a grid of points.

2. Preliminary Notes

Consider the non-homogeneous differential equation

$$L[y(t)] = y'(t) + p(t).y(t) = r(t), \;\; t \in [a,b], \;\; t \neq \tau \in [a,b]\,, \tag{1}$$

whose solution *y(t)* satisfies the Dirichlet boundary value conditions

$$y(a) = \alpha\,, \quad y(b) = \beta \tag{2}$$

and the impulse condition

$$\Delta y(\tau) = \delta, \quad \tau \in [a,b], \tag{3}$$

where $\alpha,\;\beta\,,\;\delta\;\in R$ and $p(t)\,,\;r(t)$ are continuous real functions.

Generally, the relations (1), (2) and (3) will be referred to problem (1). Here, the moment of impulse effect (jump point) is $t = \tau \in [a,b]$, where the unknown solution $y(t)$ reveals its discontinuity of first kind as jump. In order to manifest this jump, we use the notation

$$\Delta y(\tau) = y(\tau+) - y(\tau-),$$

where $y(\tau-) = y(\tau - 0) = \lim\limits_{t\to\tau-0} y(t) = \lim\limits_{\varepsilon\to 0} \frac{y(\tau)-y(\tau-\varepsilon)}{\varepsilon} = y(\tau)$, i.e. the solution $y(t)$ is continuous from the left at the point of the impulse effect $t = \tau \in [a,b]$.

Consider the interval $[a,b]$ and the grid of nodes on it

(G) $\Delta_N[a,b] : a = t_0 < t_1 < t_2 < ... < t_m < t_{m+1} < ... < t_N = b$, $t_{i+1} - t_i = h\,,\; i = \overline{0,..,N-1}$.

Definition 1. We call $S_3(t)$ a *cubic spline* with defect 1 (class ^{2}) with nodes on the grid (G), if:

1. $S_3(t) = \sum\limits_{k=0}^{3} a_k^i (t-t_i)^k$, $t \in [t_i, t_{i+1}]$, $a_k^i \in R$, $i = 0, N-1$;

2. $S_3(t) \in C^2[a,b]$.

Definition 2. The cubic spline $\tilde{S}_3(t)$ is called a *collocation cubic spline* of class ^{2} for the problem (1) with nodes on the grid (G), if it satisfies (1) at

points $\xi_k = t_k \in [a,b],\ k = 0 \div N$, the boundary value conditions (2) and the impulse condition (3), i.e. if:

$$L[\tilde{S}_3(\xi_k)] = \tilde{S}_3'(\xi_k) + p(\xi_k)\tilde{S}_3(\xi_k) = r(\xi_k), \quad \xi_k \in [a,b],\ k = 0 \div N, \tag{4}$$

$$\tilde{S}_3(a) = \alpha, \quad \tilde{S}_3(b) = \beta, \tag{5}$$

$$\tilde{S}_3(\tau+) - \tilde{S}_3(\tau-) = \delta. \tag{6}$$

Note that the boundary value problems do not have this simple formulation for the conditions of existence and uniqueness of the solution, as well as the relevant conditions at the initial value problem of Cauchy. This even more applies to the equations with impulses.

We present a cubic B-spline collocation procedure to find a numerical solution of problem (1). The application of the method leads to tridiagonal system of linear equations. The numerical solution of this system can be obtained by a variant of the well-known Thomas algorithm.

3. Cubic Spline Collocation Method

In computational mathematics B-spline is a spline function having the smallest interval of definition for a given degree of order of smoothness and domain decomposition. The fundamental theorem states that any spline function to a given degree, smoothness and interval of definition can be represented as a linear combination of B-splines at the same degree and at the same smoothness in the same interval of definition. The term B-spline was introduced by Schoenberg and it is an abbreviation of the phrase "basis spline".

We are looking for a numerical solution of problem (1) in a form of collocation cubic spline $\tilde{S}_3(t)$, in accordance with the Definition 2, where the points of collocation $\xi_i \in [a,b],\ i = 0 \div N,$, are selected to coincide with the nodes of the grid (G), i.e. $\xi_i = t_i$. Such a spline can be presented uniquely as a linear combination of normalized cubic B-splines (see [5] and [7]), i.e.

$$\tilde{S}_3(t) = \sum_{i=-1}^{N+1} c_i B_{3,i}(t), \quad N \geq 4, \tag{7}$$

where

$$B_{3,i}(t) = B_3(t - ih) = \frac{1}{6h^3} \sum_{k=0}^{4} (-1)^k C_4^k . (t - [a + (k + i - 2)h])_+^3$$

$$i = -1, 0, 1, ..., N+1. \quad (8)$$

Here

$$(t - [a + (k + i - 2)h])_+^3 = \max\left\{0,\ (t - [a + (k + i - 2)h])^3\right\}.$$

Observe that the cubic B-splines $B_{3,i}(t)$ are different from zero in the intervals $[t_{i-2}, t_{i+2}]$, $i = -1 \div N + 1$. But in order to be defined there, we need to expand the grid (G) with left nodes $t_{-3} < t_{-2} < t_{-1} < t_0 = a$ and right nodes $b = t_N < t_{N+1} < t_{N+2} < t_{N+3}$, while preserving the same evenly chosen step h, i.e. $t_{i+1} - t_i = h$, $i = \overline{-1, .., N+2}$. Such an extensive grid of nodes we will call (G+).

By considering the properties of B-splines in conjunction with collocation conditions from Definition 2, we can determine the key values of collocation spline $\tilde{S}_3(t)$ and its first derivative (see Table 3.9, [5]), i.e.

$$\tilde{S}_3(t) = \frac{c_{i-1} + 4c_i + c_{i+1}}{6}, \quad \tilde{S}_3'(t) = \frac{c_{i+1} - c_i}{2h}. \quad (9)$$

Therefore, if we substitute (9) in (2.1), we obtain a system of $N + 1$ equations with $N + 3$ unknowns c_i , $i = -1$, 0, 1..., $N + 1$, i.e.

$$A_i c_{i-1} + B_i c_i + C_i c_{i+1} = r_i \ , \ \ i = 0 \div N, \quad (10)$$

where $A_i = \frac{p_i}{6} - \frac{1}{2h}$, $B_i = \frac{2p_i}{3}$, $C_i = \frac{p_i}{6} + \frac{1}{2h}$, $p_i = p(t_i)$, $r_i = r(t_i)$.

From the left boundary condition $y(a) = \alpha = \tilde{S}_3(a)$ in (2.2), using (9), we have the equation

$$c_{-1} + 4c_0 + c_1 = 6\alpha \quad (11)$$

Now, we can exclude the unknown variablec_{-1}, using (11) and the first equation of (10), when $i = 0$. In such a way, the first equation of (10) takes the form

$$2c_0 + c_1 = 3\alpha + h(r_0 - p_0\alpha) \ = R_0. \quad (12)$$

Hence, the system (10) can be introduced as a system of $N + 1$ equations with $N + 2$ unknowns c_i , $i = 0$, 1 , ..., $N + 1$, i.e.

$$A_i c_{i-1} + B_i c_i + C_i c_{i+1} = R_i \ , \ \ i = 0 \div N, \quad (13)$$

where $A_0 = 0$, $B_0 = 2$, $C_0 = 1$, $R_0 = 3\alpha + h(r_0 - p_0\alpha)$, (9a)

$$A_i = \frac{p_i}{6} - \frac{1}{2h}, \quad B_i = \frac{2p_i}{3}, \quad C_i = \frac{p_i}{6} + \frac{1}{2h}, \quad p_i = p(t_i) \ ,$$

$$R_i = r_i = r(t_i)\ ,\ i = 1 \div N. \quad (14)$$

Further, we have to take into account the impulse condition (3) in the problem (1). Without loss of generality we can evenly choose the step h in such a way, that the moment of impulse $t = \tau \in [a, b]$ to coincide with one of the nodes of the spline collocation, for example $\tau = t_m\,,\ 1 < m < N\,.$ In order to process the jump of the unknown solution of (1) from the left hand side at $t = \tau \in [a, b]$ we introduce one artificial unknown$\hat{\alpha}$, for which it is satisfied

$$\tilde{S}_3(\tau-) = \tilde{S}_3(\tau) = \hat{\alpha}. \quad (15)$$

It does mean, by (9) and (15), that.

$$c_{m-1} + 4c_m + c_{m+1} - 6\hat{\alpha} = 0. \quad (16)$$

The equation (16) can be coupled with the *m-th* equation of the system (13), which will give us an updated version of the *m-th* equation of (13), namely

$$c_{m-1} + 2c_m = (3 + hp_m)\hat{\alpha} + hR_m. \quad (17)$$

So, the equations of the system (13) stay again $N + 1$, whereas the unknowns became $N + 3$. Now, in order to process the jump of the unknown solution of (1) from the right hand side at $t = \tau \in [a, b]$ we consider the special issue of (2.1), when $t = \tau+$, i.e.

$$\tilde{S}_3'(\tau+) + p(\tau+)\tilde{S}_3(\tau+) = r(\tau+). \quad (18)$$

From (2.3) and (9), we have $\tilde{S}_3(\tau+) = \tilde{S}_3(\tau-) + \delta \quad \Rightarrow \quad \tilde{S}_3(\tau+) = \hat{\alpha} + \delta.$ Hence, $\left(\tilde{S}_3(\tau+)\right)' = \left(\tilde{S}_3(\tau) + \delta\right)' = \left(\tilde{S}_3(\tau)\right)'$ and from (18) and (9) we obtain $\frac{c_{m+1} - c_{m-1}}{2h} + p_m(\hat{\alpha} + \delta) = R_m$, or more precisely

$$-c_{m-1} + c_{m+1} = -2hp_m\hat{\alpha} + 2h(R_m - p_m\delta). \quad (19)$$

The equation (19) can be seen as the necessary equation, whereby the system (13) becomes a system of $N + 2$ equations with $N + 3$ unknowns $c_i\ ,\ i = 0,\ 1\,, ..., N + 1$ and $\hat{\alpha}$.

Finally, from the right boundary condition $y(b) = \beta\ = \tilde{S}_3(b)$ in (2.2), using (9), we receive another one equation, i.e.

$$c_{N-1} + 4c_N + c_{N+1} = 6\beta\,. \quad (20)$$

Here, we can exclude and the unknown variablec_{N+1}, using (20) and the last equation of (13), when $i = N$. In such a way, the last equation of (13) takes the form

$$c_{N-1} + 2c_N = (6 + 2hp_N)\beta - 2hR_N. \tag{21}$$

Thus, we obtain the tridiagonal system (22) of $N + 2$ equations with $N + 2$ unknowns c_i , $i = 0, 1, ..., N$ and $\hat{\alpha}$.

$$\left|\begin{array}{rl}
& 2c_0 + c_1 = R_0, \\
& A_1c_0 + B_1c_1 + C_1c_2 = R_1, \\
& A_2c_1 + B_2c_2 + C_2c_3 = R_2, \\
& \vdots \\
& A_{m-1}c_{m-2} + B_{m-1}c_{m-1} + C_{m-1}c_m = R_{m-1}, \\
& c_{m-1} + 2c_m = (3 + hp_m)\hat{\alpha} + hR_m, \\
& \vdots \\
& -c_{m-1} + c_{m+1} = -2hp_m\hat{\alpha} + 2h(R_m - p_m\delta), \\
& A_{m+1}c_m + B_{m+1}c_{m+1} + C_{m+1}c_{m+2} = R_{m+1}, \\
& \vdots \\
& A_{N-1}c_{N-2} + B_{N-1}c_{N-1} + C_{N-1}c_N = R_{N-1}, \\
& c_{N-1} + 2c_N = (6 + 2hp_N)\beta - 2hR_N.
\end{array}\right. \tag{22}$$

The solution of the system can be obtained by a variant of the well-known Thomas algorithm.

Forward movement of the algorithm: It is easy to be seen, that the moment of impulse effect at the *m-th* point $\tau = t_m$, is introduced by two equations, which divide the system (22) into two partitions.

Processing of the first partition: From the first m equations of (22) we introduce temporarily the unknownsc_i, $i = 1 \div m$, as

$$c_{i-1} = \tilde{A}_i c_i + \tilde{R}_i \quad , \quad i = 1 \div m, \tag{23}$$

where

$$\tilde{A}_1 = \frac{-1}{2} \ , \ \tilde{R}_1 = \frac{R_0}{2},$$

$$\tilde{A}_i = \frac{-C_{i-1}}{A_{i-1}\tilde{A}_{i-1} + B_{i-1}} \ , \quad \tilde{R}_i = \frac{r_{i-1} - A_{i-1}\tilde{R}_{i-1}}{A_{i-1}\tilde{A}_{i-1} + B_{i-1}} \ , \quad i = 2 \div m.$$

(18a)

At the end of the partition, from the *(m+1)-st* equation of (22) and (23), when $i = m$, we obtain

$$c_m = \bar{A}_m\hat{\alpha} + \bar{R}_m \quad , \tag{24}$$

where

$$\bar{A}_m = \frac{3 + hp_m}{2 + \tilde{A}_m} \quad , \quad \bar{R}_m = \frac{hR_m - \tilde{R}_m}{2 + \tilde{A}_m}.$$

(19a)

Processing of the second partition: For our needs in this partition we introduce the unknown c_{m-1} in another way : from (23) when $i = m$ and from (24), that is

$$c_{m-1} = \bar{A}_{m-1}\hat{\alpha} + \bar{R}_{m-1} \quad , \tag{25}$$

where $\bar{A}_{m-1} = \frac{\tilde{A}_m(3+hp_m)}{2+\tilde{A}_m}$, $\bar{R}_{m-1} = \frac{\tilde{A}_m hR_m + 2\tilde{R}_m}{2+\tilde{A}_m}$. (20a)

From (25) and the *(m+2)-nd* equation of (22), we have

$$-\frac{\tilde{A}_m\,(3 + hp_m)}{2 + \tilde{A}_m}\hat{\alpha} - \frac{\tilde{A}_m hR_m + 2\tilde{R}_m}{2 + \tilde{A}_m} + c_{m+1} = -2hp_m\hat{\alpha} + 2h(R_m - p_m\delta),$$

which can help us to introduce

$$c_{m+1} = \bar{A}_{m+1}\hat{\alpha} + \bar{R}_{m+1}, \tag{26}$$

where

$$\bar{A}_{m+1} = \frac{\tilde{A}_m\,(3 - hp_m) - 4hp_m}{2 + \tilde{A}_m},$$

$$\bar{R}_{m+1} = \frac{3\tilde{A}_m hR_m + 2\tilde{R}_m + 4hR_m - 2hp_m\delta(2 + \tilde{A}_m)}{2 + \tilde{A}_m}.$$

(21a)

Further, from the *(m+3)-rd* to the *(N+1)-st* equation of the system (22), we obtain

$$c_j = \bar{A}_j\hat{\alpha} + \bar{R}_j,\ j = m + 2 \div N, \tag{27}$$

where $\bar{A}_j = -\frac{A_{j-\mathbf{1}}\bar{A}_{j-\mathbf{2}} + B_{j-\mathbf{1}}\bar{A}_{j-\mathbf{1}}}{C_{j-\mathbf{1}}}$, $\bar{R}_j = \frac{R_{j-\mathbf{1}} - (A_{j-\mathbf{1}}\bar{R}_{j-\mathbf{2}} + B_{j-\mathbf{1}}\bar{R}_{j-\mathbf{1}})}{C_{j-\mathbf{1}}}$. (22a)

Finally, from the *(N+2)-nd* equation of the system (22) , using (27) for $j = N - 1$ and for $j = N$, we are able to determine how the variable $\hat{\alpha}$ is equal, that is

$$\hat{\alpha} = \frac{(6 + 2hp_N)\beta - 2hR_N - (\bar{R}_{N-1} + \bar{R}_N)}{\bar{A}_{N-1} + 2\bar{A}_N}. \tag{28}$$

Backward movement of the algorithm: Tracing the way back through formulas (27), (26), (24) and (23) determines the unknowns$c_i,\ i = 0 \div N$, which completely define the spline (7) we are looking for.

4. Software Implementation of the Algorithm

The software implementation of the above mentioned algorithm may be performed by means of the formulas (23), (24), (26) and (27), based on (13). Briefly, with elements of pseudo code, we can say the following (by using C/C++):

Define the operands needed to initialize the algorithm:

float a, b, alpha, alphaH, beta, delta, tau, h; //

for $a,\ b,\ \alpha,\ \hat{\alpha}\,,\ \beta,\ \delta,\ \tau,\ h$;

integer N, m; // for the length of the grid (G+) and the distance to τ;

as well as the base functions

float funcP(float t)

{*// to calculate* $p_i = p(t_i)$ *for fixed i;* }

float funcR(float t)

{*// to calculate* $r_i = r(t_i)$ *for fixed i;* }.

Calculate or input the relevant initial values:

BaseVal ( )

{*// input a,b for interval [a,b]; input* $\alpha,\ \beta$ *for boundary conditions;*

// input δ *for the impulse condition ;*

// input the point of jump τ*; calculate step h, accordingly a, b and tau, so that* $\tau = t_m\,,\ 1 < m < N\,;$

// calculate N in order to know the nodes of the grid (G+) : $t_{i+1} - t_i = h\,,\ i = \overline{-3, .., N+2}$;

// calculate m in accordance to $\tau = t_m\,,\ 1 < m < N$ }

Callculate the coefficients of the forward movement of the algorithm:

main( )

{BaseVal( );

float a[N+1], b[N+1], c[N+1], r[N+1]; // for coefficients in (9a) and (14)

float aW[m], rW[m]; // for coefficients in (18a);

aW[0]=0; rW[0]=0;

float aL[m], rL[m]; // for coefficients in (19a);

float cS[N+1]; // for coefficients of the Spline

float t, $ta = a - h; t = ta;$

$a[0] = 0$, $b[0] = 2$, $c[0] = 1$, $r[0] = 3{*}alpha{+}h{*}(a{-}funcP(a){*}alpha)$; // calculation of (9a)

//calculation of (14);

$$for(i = 0; i < N + 1; i + +)$$

{$t{=}\ t + h$;

$$a[i] = (funcP(t)/6) - 1/2 * h\,; \quad b[i] = 2 * funcP(t)/3;$$
$$c[i] = (funcP(t)/6) + 1/2 * h; \quad r[i] = funcR(t);\}$$

//calculation of (18a);

$t = ta;\ aw[1] = -1/2,\ rw[1] = r[0]/2;$

$$for(i = 2\,; i < m + 1; i + +)$$

{$t{=}\ t + h$;

$aW[i] = -c[i-1]/(a[i-1] * aW[i-1] + b[i-1])$;
$rW[i] = (r[i-1] - a[i-1] * rW[i-1])/(a[i-1] * aW[i-1] + b[i-1]);\}$

//calculation of (20a);

$$aL[m-1] = aw[m] * (3 + h * funcP(a + m * h))/(2 + aW(m);$$
$$rL[m-1] = (aw[m] * h * r[m] + 2 * rW[m])/(2 + aW[m]);$$

//calculation of (19a);

$$aL[m] = (3 + h * funcP(a + m * h))/(2 + aW(m);$$
$$rL[m] = (h * r[m] - rW[m])/2 + aW[m]);$$

// calculation of aL[m+1] and rL[m+1] – the code follows (21a) and is omitted for convenience

//calculation of (22a);

$t = a + h * (m{+}1);$

$$for(i = m + 2\,; i < N + 1; i + +)$$

{$t{=}\ t + h$;

$aL[i] = (a[i-1] * aL[i-2] + b[i-1] * aW[i-1])/c[i-1]$;
$rL[i] = (r[i-1] - (a[i-1] * rL[i-2] + b[i-1] * rL[i-1]))/c[i-1];\}$

//calculation of $\hat{\alpha}$, *accordingly (28), i.e. we know yet alphaH;*

$$alphaH = ((6 + 2 * h * funcP(a + N * h)) * beta - 2 * h * r[N] - (rL[N-1] + rL[N]))/(aL[N-1] + 2 * aL[N])$$

Callculate the coefficients of the spline (backward movement of the algorithm):

/ calculation of coefficients in (24), (26) and (27);

$$for(i = m;\ i < N + 1;\ i++)$$
$$cS[i] = alphaH * aL[i] + rL[i];$$

// calculation of coefficients in (23);

$$for(i = m;\ i > 0;\ i--)$$
$$cS[i] = alphaH * aW[i] + rW[i];$$

References

[1] A. Andreev, On interpolation by cubic splines of a function, possessing discontinuities, *Publ. House of Bulg. Acad. of Sciences*, **27**, No. 8 (1974), 881-884.

[2] *Inequalities*. Proceedings of a Symposium Held at Wright-Paterson Air Force Base, Ohio, August, 1965.

[3] V.D. Milman, A.D. Myshkis, On stability of motion in the presence of impulses, *Sibirsk Math. J.*, **1**, No. 2 (1960), 233-237, In Russian.

[4] V.D. Milman, A.D. Myshkis, Random impulses in linear dynamical systems, In: *Approximate Methods of Solutions of Differential Equations*, Publ. House of Acad. of Sci. USSR, Kiev (1963), 64-81, In Russian.

[5] U. Zavyalov, B. Kvasov, V. Miroshnichenko, *Spline Functions Methods*, Moskow, Nauka (1980), In Russian.

[6] A. Samojlenko, N. Ronto, O. Kurbanbaev, *Collocation Method of Boundary Value Problems for Ordinary Differential Equations with Impulse Effects*, Publ. house Acad. of Sci. USSR, Kiev (1989), In Russian.

[7] S. Stechkin, Yu. Subotin, *Splines in Computational Mathematics*, Moskow, Nauka (1976), In Russian.