UNIVERSITAS COMENIANA
ACTA MATHEMATICA UNIVERSITATIS COMENIANAE
LIV—LV—1988

AN ALGORITHM FOR SOLVING THE THREE-POINT BOUNDARY VALUE PROBLEM FOR O. D. E.

JÁN PEKÁR, Bratislava

1 Introduction

In this paper we construct and test an algorithm for solving the three-point boundary value problem, which is from the class of such algorithms as the one described for the two-point boundary value problem by Tewarson and Huslak [3]. The exact formulation of the problem and the used notation are given in §2. The proposed method is described in detail in §3. The numerical aspects of the method are considered in §4. Here are also given some results obtained from computational experiments.

2 The formulation of the problem

We consider the numerical solution of the three-point boundary value problem for the system of s non-linear differential equations

$$y'(x) - f(x, y(x)) = 0 \tag{1}$$

with linear boundary value conditions

$$g(y(a), y(b), y(c)) = 0, \tag{2}$$

where y, f and g are functions with values in R^s, and $a \leqq x \leqq c$, $a < b < c$. We assume that the problem (1), (2) has an isolated solution and the function f is sufficiently smooth.

Let us subdivide the x range $[a, b]$ into m equal parts and the x range $[b, c]$ into n equal parts, such that $h = (b - a)/m$ and $k = (c - b)/n$. Then the interval $[a, c]$ is subdivided into $m + n$ subintervals $[x_i, x_{i+1}]$, $i = 0, 1, 2, \ldots, m + n - 1$, where

$$x_i = a + i . h \qquad \text{for } i = 0, 1, 2, \ldots, m - 1$$

and

$$x_i = b + (i - m) . k \quad \text{for } i = m, m + 1, m + 2, \ldots, m + n - 1.$$

Let us denote the exact value of the solution of the problem (1), (2) at point x_i as $y(x_i)$, the approximation of $y(x_i)$ as y_i, and $f_i = f(x_i, y_i)$. Now Y is a vector with $m + n + 1$ components y_i. Every component y_i is a vector containing s components. If we integrate the equation (1) in the interval $[x_i, x_{i+1}]$, then we have

$$y(x_{i+1}) - y(x_i) - \int_{x_i}^{x_{i+1}} f(t, y(t))\,\mathrm{d}t = 0. \tag{3}$$

Let us evaluate the integral in (3) by a numerical rule H_i We obtain the following integration formulae:

$$y_{i+1} - y_i - h . H_i = 0, \quad i = 0, 1, \ldots, m - 1, \tag{4.a}$$

$$y_{i+1} - y_i - k . H_i = 0, \quad i = m, m + 1, \ldots, m + n - 1. \tag{4.b}$$

If the integral in (3) is evaluated by the trapezoidal rule with the local order of accuracy 3, then we have

$$H_i^T = (f_i + f_{i+1})/2. \tag{5}$$

If the considered integral is evaluated by either the Simpson rule (with the local order of accuracy 5) or the Newton-Cotes rule (with the local order of accuracy 7), then analogously to (5) we get

$$H_i^S = (f_i + 4f_{i+1/2} + f_{i+1})/6 \tag{6}$$

and

$$H_i^N = (f_i + 32f_{i+1/4} + 12f_{i+1/2} + 32f_{i+3/4} + f_{i+1})/90. \tag{7}$$

In formulae (6) and (7) we have used the values of the function f at the added points $x_{i+1/4}, x_{i+1/2}, x_{i+3/4}$ (i.e. values $f_{i+1/4}, f_{i+1/2}, f_{i+3/4}$), where

$$x_{i+q} = x_i + q . h \quad \text{for } i = 0, 1, \ldots, m - 1,$$

and

$$x_{i+q} = x_i + q . k \quad \text{for } i = m, m + 1, \ldots, m + n - 1.$$

Now we use the trapezoidal and the Simpson formula on the subintervals $[x_i, x_{i+1/2}]$ and $[x_{i+1/2}, x_i]$. On the whole interval $[x_i, x_{i+1}]$ we get the composite trapezoidal formula

$$H_i^{CT} = (f_i + 2f_{i+1/2} + f_{i+1})/4 \tag{8}$$

and the composite Simpson formula

$$H_i^{CS} = (f_i + 4f_{i+1/4} + 2f_{i+1/2} + 4f_{i+3/4} + f_{i+1})/12 \tag{9}$$

of orders 3 and 5 respectively.

Now we can consider our method in detail.

3 The description of the method

The evaluation of the integral in (3) by (4. a—b) leads to a system of $s.(m+n)$ equations in $s.(m+n+1)$ variables. From (2) we get

$$g(y_0, y_m, y_{m+n}) = 0, \tag{10}$$

which is now a system of s functions in $3.s$ variables. Because g is a linear function, we can express s variables from (10) in terms of the other $2.s$ variables and then utilize them to eliminate s variables from the system obtained by using (4.a—4.b). The resulting system of $s.(m+n)$ non-linear equations in $s.(m+n)$ variables can be written as

$$F(Y) = 0. \tag{11}$$

If the function F satisfies the sufficient conditions for the convergence (see e.g. in [2], Chapter 10), then we can use the Newton Iterations Method in such a modified form:

To solve (11), we select a starting vector $Y^{(0)}$ and proceed iteratively, obtaining refined approximations $Y^{(k)}$ to the solution vector. An iteration consists of three steps:

a. The choice of the numerical formula which will be employed in (3).

b. A repeated bisection of certain subintervals until the local error tolerance is met (only in the case of precluding the selection of the integration formulae by the estimated error).

c. A Newton step to update the current approximation $Y^{(k)}$.

Now we describe this algorithm in greater mathematical detail. Let ε be a given local absolute error tolerance and $Y^{(k-1)}$ an approximate solution vector for (1), (2). First we calculate the values of the formulae H_i^{CT} and H_i^S for all i. If the difference

$$|H_i^{CT}(Y^{(k-1)}) - H_i^S(Y^{(k-1)})|, \tag{12}$$

which is an estimate of the local error and is of the order 3, is less than ε for any i, then we use $H_i^S(Y^{(k-1)})$ in (11); otherwise the estimate of the order 5

$$|H_i^{CS}(Y^{(k-1)}) - H_i^N(Y^{(k-1)})| \tag{13}$$

is calculated and compared with ε. If it is less than ε, then the value of $H_i^N(Y^{(k-1)})$ is used as the value of H_i in (11); otherwise the use of the step b is necessery.

Thus we can write the calculation of the value H_i (actually the step a of the iteration) in the form

$$H_i = \begin{cases} H_i^S(Y^{(k-1)}) & \text{if} \, |H_i^{CT}(Y^{(k-1)}) - H_i^S(Y^{(k-1)})| < \varepsilon \\ H_i^N(Y^{(k-1)}) & \text{if} \, |H_i^{CS}(Y^{(k-1)}) - H_i^N(Y^{(k-1)})| < \varepsilon \\ \text{see step b} & \text{otherwise} \end{cases} \tag{14}$$

for $i = 0, 1, 2, \ldots, m + n - 1$, where H_i^S is given by (6), H_i^N is given by (7), H_i^{CT} is given by (8), and H_i^{CS} is given by (9).

If the estimate (13) is greater than or equal to ε, then we start the step b of the iteration, else we go to the step c.

If the use of step b is necessery, we add point $x_{i+1/2}$ to the original mesh and instead of the interval $[x_i, x_{i+1}]$ we consider the subintervals $[x_i, x_{i+1/2}]$ and $[x_{i+1/2}, x_{i+1}]$. We apply the step a of the iteration on both subintervals. We repeatedly bisect all such intervals which do not satisfy the local error tolerance until the above-mentioned tolerance is met or too many bisections take place. In the first case we give the sum of the partial values of H obtained by a repeated bisection at the place of H_i in (11):

$$H_i = \sum_j H_{i_j}(Y^{(k-1)}) \tag{15}$$

for all such j that $i \leqq i_j < i + 1$ and on $[x_{i_j}, x_{i_{j+1}}]$ the local error tolerance is met. In the latter case the use of this method does not lead to the successful finish.

The both above-described steps give the local error control of computations.

The final step, step c, is the Newton step to update the current approximation $Y^{(k)}$:

$$Y^{(k)} = Y^{(k-1)} - [J(F(Y^{(k-1)}))]^{-1} F(Y^{(k-1)}), \tag{16}$$

where $J(F(Y^{(k-1)}))$ denotes the Jacobian of F with respect to Y evaluated at $Y^{(k-1)}$. The Jacobian in (16) can be computed by the consistent discrete approximation [2]:

$$J(F(Y^{(k-1)}))e_j \sim (F(Y^{(k-1)} + \delta e_j) - F(Y^{(k-1)}))/\delta, \tag{17}$$

where e_j is the j-th column of the identity matrix of the order $s \, . \, (m + n)$ and δ is a suitable small number. By the Discrete Newton Theorem [2] the convergence of (16) is not affected by the use of the approximation (17).

4 Numerical aspects and experiments

The last problem we have not touched yet in the description of our method is the calculation of the approximate values of the solution inside the mesh-points and the appropriate values of f. Various interpolation formulae are available to satisfy the error criterion. The used formula must be at least of the order 7, because the use of lower order formulae does not lead to significiantly better results.

The algorithm was tested on some three-point boundary value problems. All computations were performed in double precision arithmetics on a SM 4/20 using FORTRAN IV. In *Table 1* we have given the results obtained by solving the problem

$$y_1' = y_2, \quad y_2' = y_3, \quad y_3' = y_2 + 2y_3, \tag{18}$$

$$\begin{cases} y_3(0) = 1, \\ y_3(1) = e^{-1}, \\ y_3(1{,}5) = e^{-1{,}5}. \end{cases} \tag{19}$$

Table 1

Number of mesh-points	$m = 100,\ n = 50$				$m = 1000,\ n = 500$			
Absolute accuracy	10^{-8}	10^{-10}	10^{-12}	10^{-14}	10^{-8}	10^{-10}	10^{-12}	10^{-14}
Numer of iterations	9	12	14	16	6	10	10	12
Number of added points	23	23	23	23	16	18	19	19

In *Table 2* we have given the results which are obtained by solving the problem

$$\begin{gathered} y_1' = y_2, \quad y_2' = y_3, \\ y_3' = xy_1 + y_2 - \left(\frac{x^4}{12} + \frac{x^2}{6}\right)y_3 + x, \end{gathered} \tag{20}$$

$$\begin{cases} y_3(-1) = 1/2, \\ y_3(0) = 0, \\ y_3(1) = 1/2. \end{cases} \tag{21}$$

Table 2

Number of mesh-points	$m = n = 100$				$m = n = 1000$			
Absolute accuracy	10^{-8}	10^{-10}	10^{-12}	10^{-14}	10^{-8}	10^{-10}	10^{-12}	10^{-14}
Number of iterations	8	11	15	18	6	9	11	15
Number of added points	17	19	23	26	11	15	16	18

Sufficient conditions for the existence and uniqueness of the solution of our three-point boundary value problem for a differential equation of the third order are given in [1]. We can formulate them as follows:

Let the function f satisfy the Lipschitz condition

$$|f(x, y_1, z_1, w_1) - f(x, y_2, z_2, w_2)| \leqq$$
$$\leqq L_0|y_1 - y_2| + L_1|z_1 - z_2| + L_2|w_1 - w_2|$$

and the conditions (i), and (ii):

(i) $$y_1 \geqq y_2,\ z_1 < z_2 \text{ implies}$$
$$f(x, y_1, z_1, w) < f(x, y_2, z_2, w) \qquad \text{on } (a, b]$$

and

(ii) $$y_1 \leqq y_2,\ z_1 < z_2 \text{ implies}$$
$$f(x, y_1, z_1, w) < f(x, y_2, z_2, w) \qquad \text{on } [b, c)$$

at point b in (a, c).

If $h_1 = b - a$ and $h_2 = c - b$ satisfy the condition

$$\frac{1}{60} L_0 h_i^3 + \frac{1}{6} L_1 h_i^2 + \frac{2}{3} L_2 h_i < 1, \quad i = 1, 2,$$

then the boundary value problem

$$y''' = f(x, y, y', y'')$$
$$y(a) = y_a, \quad y(b) = y_b, \quad y(c) = y_c$$

has a unique solution.

Since the functions on the right-hand side of the equations of the third order corresponding to the systems (18) and (20) satisfy these conditions, both problems (18), (19) and (20), (21) have a unique solution.

In both cases a starting vector was used which was obtained by choosing all y-values equal to the relevant boundary values. For all the needed interpolations the Newton forward and the Newton backward rules of the 7-th order were used.

REFERENCES

1. Das, K. M.—Lalli, B. S.: Boundary value problems for $y''' = f(x, y, y', y'')$. J. Math. Anal. Appl. 81 (1981), Nr. 2, pp. 300—307.

2. Ortega, J.—Rheinboldt, W. C.: Iterative solution of non-linear equations in several variables. Academic Press, New York 1970.
3. Tewarson, R. P.—Huslak, N. S.: An adaptive implementation of interpolation methods for boundary value ordinary differential equations. BIT 23 (1983), pp. 382—387.

Author's address:

Ján Pekár
MFF UK, Katedra numerických a optimalizačných metód
Matematický pavilón
Mlynská dolina
842 15 Bratislava

Received: 9. 2. 1987

SÚHRN

ALGORITMUS NA RIEŠENIE TROJBODOVEJ OKRAJOVEJ ÚLOHY PRE OBYČAJNÉ DIFERENCIÁLNE ROVNICE

Ján Pekár, Bratislava

V článku je skonštruovaný a otestovaný algoritmus na riešenie trojbodovej okrajovej úlohy pre systém obyčajných diferenciálnych rovníc $y' = f(x, y)$ s lineárnymi okrajovými podmienkami, ktorý je založený na newtonovských iteráciách.

РЕЗЮМЕ

АЛГОРИТМ ДЛЯ РЕШЕНИЯ ТРЕХТОЧЕЧНОЙ КРАЕВОЙ ЗАДАЧИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Ян Пекар, Братислава

В работе построен и на примерах проверается алгоритм решения трехточечной краевой задачи для обыкновенных дифференциальных уравнений, основанный на идее итераций Ньютона.