A GENERAL SCHEME FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS UNDER TWO-POINT BOUNDARY CONDITIONS

By

MANAS CHANDA

(Department of Chemical Engineering, Indian Institute of Science, Bangalore-12, India)

AND

SYAMAL KUMAR SEN

(Central Instruments and Services Laboratory, Indian Institute of Science, Bangalore-12, India)

[Received: January 12, 1972]

ABSTRACT

A procedure for rewriting ordinary differential equations (linear or nonlinear, eoupled or noncoupled) under two-point boundary conditions into a system of first order equations with a set of general boundary conditions is given. A scheme suitable for solving the latter general system numerically is discussed.

I. INTRODUCTION

A general numerical procedure (and hence a general subprogram) for solving arbitrary differential equations (ODE), linear or nonlinear, coupled or noncoupled, is available so long as it is an initial value problem (IVP). In fact, nonlinearity of ODE and its coupled form do not pose any extra problem over its linearity and noncoupled form. This is however not true for ODE under two-point boundary conditions. Linearity of ODE here makes the problem much simpler and in most cases it can be directly solved; non-linear ODE, on the other hand, has to be solved iteratively in general. Secondly, the two-point boundary value problem (BVP) has not been posed[1] in a general form as IVP, probably due to the fact that the conditions on the two boundaries may have many possible combinations instead of a single form as in IVP.

It is the object of this paper to describe a general form into which a wide variety of two-point BVP (linear or nonlinear, coupled or noncoupled) can be fitted; then an iterative scheme (for numerical solution) suited to this general form is discussed.

2. GENERAL FORM OF TWO-POINT BVP

A general two-point BVP (ordinary differential equations) can be written as a system of n first order differential equations, as defined below:

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, \ldots, y_n)$$

(I) *Boundary conditions* :

(a) $x = x_o$, $y_i = y_{io}$, $i = 1(1)p$, $p \leq n$

(b) $x = x_l$, $y_i = y_{il}$, $i = j(k)$, $k = 1(1)q$ where $q = n - p$

Note that $j(1)$, $j(2)$, ..., $j(q)$ may have any value between 1 and n. We have to, in fact, feed these values in j-locations.

This may be illustrated by taking an example. Let the equation be

$$\phi_1(x, y)\frac{d^3y}{dx^3} + \phi_2(x, y)\frac{d^2y}{dx^2} + \phi_3(x, y)\frac{dy}{dx} + \phi_4(x, y)\, y + \phi_5(x, y) = 0$$

(II)

with the boundary conditions :

$$x = x_o,\ y = \alpha,\ \frac{d^2y}{dx^2} = \beta$$

$$x = x_l,\ \frac{dy}{dx} = \gamma$$

Using the symbolic transformation $y \to z_1$, $\frac{dy}{dx} \to z_2$, $\frac{d^2y}{dx^2} \to z_3$, the system (II) can be written as

$$\frac{dz_1}{dx} = \theta_1(x, z_1, z_2, z_3) = z_2$$

$$\frac{dz_2}{dx} = \theta_2(x, z_1, z_2, z_3) = z_3$$

$$\frac{dz_3}{dx} = \theta_3(x, z_1, z_2, z_3) = -(\phi_2 z_3 + \phi_3 z_2 + \phi_4 z_1 + \phi_5)/\phi_1$$

(III)

with the boundary conditions :

$$x = x_o,\ z_1 = \alpha,\ z_3 = \beta$$

$$x = x_l,\ z_2 = \gamma$$

Renaming the symbols z_1, z_2, z_3 as y_1, y_3, y_2 respectively, system (III) can be rewritten as

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, y_3) = y_3$$

$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, y_3) = -(\phi_2 y_2 + \phi_3 y_3 + \phi_4 y_1 + \phi_5)/\phi_1$$

$$\text{(IV)} \quad \frac{dy_3}{dx} = f_3(x, y_1, y_2, y_3) = y_2$$

having the boundary conditions

$$x = x_o, \; y_1 = \alpha, \; y_2 = \beta$$

$$x = x_l, \; y_3 = \gamma$$

System (IV) is the required particular form of (I) where $j(1) = 3$.

3. Scheme of Solution

Given a boundary value problem (ODE), linear or nonlinear, coupled or noncoupled, we feed at the start the values of $j(1)$, $j(2)$, $\cdots$, $j(q)$ which are the values of i in y_i in the second boundary. In the above example, $j(1)$ is 3. Note that when $p = n$, the problem is an initial value problem.

Taking l intervals between x_0 and x_l, we obtain the step size as $h = (x_l - x_o)/l$. We then apply the Gill method[2] or Gill and Hamming's Predictor-Modifier-Corrector (PMC) method[3] to obtain y_1, y_2, $\ldots$, y_n at all l points using the equations [I] with initial conditions:

$$x = x_o, \; y_i = y_{io}, \; i = 1(1)\, p$$

$$\text{(V)} \qquad y_{p+m} = \alpha_m, \; m = 1(1)q, \text{ where } \alpha_m, \; m = 1(1)q \text{ are trial values.}$$

The step by step integration can be tabulated as in Table 1

In general, the assumed α is not correct ; hence the given $y_{j(1), l}$, $\cdots$, $y_{j(q), l}$ (see the second boundary conditions) do not agree with the corresponding values $y_{j(1)}(x_l, \alpha)$, $\ldots$, $y_{j(q)}(x_l, \alpha)$ obtained through the step by step integration shown in the last row of Table 1. Store the last row of this Table.

Let the corrections to α be given by k_i, so that

$$\alpha_i = \alpha_i + k_i, \; i = 1(1)q$$

TABLE 1

α stands for $(\alpha_1, \alpha_2, \cdots \alpha_q)$

x	y_1	y_2	$\cdots$	y_p	y_{p+1}	$\cdots$	y_{n-1}	y_n
x_0	y_{10} ←Given initial conditions→	y_{20}	$\cdots$	y_{p0}	α_1 ←Assumed initial condtions→	$\cdots$	α_{q-1}	α
x_0+h								
x_0+2h								
$\cdot$								
$\cdot$								
$\cdot$								
$x+(l-1)\,h$								
$x=x_0+lh$	$y_1(x_l, \alpha)$	$y_2(x_l, \alpha)$	$\cdots$	$y_p(x_l, \alpha)$	$y_{p+1}(x_l, \alpha)$	$\cdots$	$y_{n-1}(x_l, \alpha)$	$y_n(x_l, \alpha)$

We choose k_i such that

$$\begin{aligned} y_{j(1),\, l} &= y_{j(1)}(x_l,\ \alpha_1+k_1,\ \alpha_2+k_2,\ \ldots,\ \alpha_q+k_q) \\ &= y_{j(1)}(x_l,\ \alpha)+\left[k_1 \frac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_1}+k_2 \frac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_2}+\ldots\right. \\ &\quad \left.+\,k_q \frac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_q}\right]+O(k_1^2,\ k_2^2,\ \ldots,\ k_q^2) \end{aligned}$$

$$\text{(VI)} \quad \begin{aligned} y_{j(2),\, l} &= y_{j(2)}(x_l,\ \alpha)+\left[k_1 \frac{\partial y_{j(2)}(x_l,\ \alpha)}{\partial \alpha_1}+k_2 \frac{\partial y_{j(2)}(x_l,\ \alpha)}{\partial \alpha_2}+\ldots\right. \\ &\quad \left.+\,k_q \frac{\partial y_{j(2)}(x_l,\ \alpha)}{\partial \alpha_q}\right]+O(k_1^2,\ k_2^2,\ \ldots,\ k_q^2) \end{aligned}$$

$$\ldots \qquad \ldots \qquad \ldots \qquad \ldots$$

$$\begin{aligned} y_{j(q),\, l} &= y_{j(q)}(x_l,\ \alpha)+\left[k_1 \frac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_1}+k_2 \frac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_2}+\ldots\right. \\ &\quad \left.+\,k_q \frac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_q}\right]+O(k_1^2,\ k_2^2,\ \ldots,\ k_q^2) \end{aligned}$$

Neglecting second and higher order terms, system (VI) can be written as:

$$
\text{(VII)}\quad
\begin{bmatrix}
\dfrac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_1} & \dfrac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_2} & \cdots & \dfrac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_q} \\
\dfrac{\partial y_{j(2)}(x_l,\ \alpha)}{\partial \alpha_1} & \dfrac{\partial y_{j(2)}(x_l,\ \alpha)}{\partial \alpha_2} & \cdots & \dfrac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_q} \\
\cdots & \cdots & \cdots & \cdots \\
\dfrac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_1} & \dfrac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_2} & \cdots & \dfrac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_q}
\end{bmatrix}
\begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_q \end{bmatrix}
=
\begin{bmatrix} y_{j(1),\,l}-y_{j(1)}(x_l,\ \alpha) \\ y_{j(2),\,l}-y_{j(2)}(x_l,\ \alpha) \\ \vdots \\ y_{j(q),\,l}-y_{j(q)}(x_l,\ \alpha) \end{bmatrix}
$$

Let the correction vector $(k_1,\ k_2,\ \ldots,\ k_q)'$ be denoted by k ($'$ indicates transpose) and the coefficient matrix be denoted by $Z(=z_{ij})$. The *r. h. s.* of (VII) is known from the step by step integration (see Table 1) and given conditions at x_l (see I b). Call it b. Thus (VII) is written as

$$\text{(VIII)}\quad Zk=b$$

The correction vector k may be obtained if Z is known.

Estimation of Z-matrix

To evaluate the q^2 elements of Z or, in other words, $\dfrac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_1}$, $\dfrac{\partial y_{j(1)}(x_l,\ \alpha)}{\partial \alpha_2}$, $\ldots$, $\dfrac{\partial y_{j(q)}(x_l,\ \alpha)}{\partial \alpha_q}$, we repeat the above step by step integration process q times as follows:

To obtain the s-th column of Z we solve (I) with boundary conditions

(IX) (a) $x=x_o,\ y_i=y_{io},\ i=1(1)p$

(b) $y_{p+m}=\alpha_m,\ m=1(1)q,\ m\neq s;\ y_{p+s}=\alpha_s+\Delta\alpha_s$

and store the values of $y_{j(k)}(x_l,\ \alpha_1,\ \ldots,\ \alpha_s+\Delta\alpha_s,\ \ldots,\ \alpha_q)$, $k=1(1)q$ (*i.e.*, the last row of Table 1). Using two-term Taylor's series

$$\text{(X)}\quad \frac{\partial y_{j(k)}(x_l,\ \alpha)}{\partial \alpha_s}=\frac{y_{j(k)}(x_l,\ \alpha_1,\ \ldots,\ \alpha_s+\Delta\alpha_s,\ \ldots,\ \alpha_q)-y_{j(k)}(x_l,\ \alpha)}{\Delta\alpha_s}\qquad k=1(1)q$$

Thus, when s runs from 1 to q (*i.e.*, $s=1(1)q$), we obtain the complete Z matrix, Note that the values of $y_1(x_l, \alpha)$, $y_2(x_l, \alpha)$, $\ldots$, $y_n(x_l, \alpha)$, *i. e.*, the last row of Table 1, in the first step by step integration process have to be retained over the entire range of s as they are required for obtaining each colum of Z.

Estimation of correction vector k

Z in general turns out to be nonsingular in each iteration as is our experience with the computation of many physical problems. If, however, Z, by chance, turns out to be singular we may change any one of $\Delta\alpha$'s and reestimate Z. System (VIII) can be solved by methods like partial pivoting[4] that does not explicitly require interchange of rows (or columns) as well as explicit inversion or any other suitable methods[5, 6].

New initial conditions

Now the old α's are replaced by $\alpha_m + k_m$, $m=1(1)q$. The new initial conditions for the second iteration are

$$x = x_o, \; y_i = y_{io}, \; i = 1(1)p$$

$$y_{p+m} = \alpha_m + k_m = \text{new } \alpha_m, \; m = 1(1)\,q$$

In a few iterations (3 or 4) we obtain thus the required intial conditions (n conditions) which satisfy the conditions at the second boundary through step by step integration process.

4. REMARKS

(a) Preparing a general computer program into which a wide variety of two-point BVP (ODE) can be fitted is made possible by this scheme.

(b) This scheme aims at transforming a two-point BVP into an IVP.

(c) Though a linear ODE (BVP) in most of the cases can be solved directly by using the property that the linear combination of the solution at a point is also a solution of linear ODE, the present iterative scheme may have better performance not only for linear ODE (BVP) which can be solved directly but also for those which cannot be solved directly.

(d) The main point to note is the condition (b) of (I) which makes the computer program a general one. A nonsubscripted variable instead of a subscripted one (like $j(k)$) cannot explicitly achieve the generalisation of a program that can tackle a large variety of two-point BVP's. Depending on a given problem it is only required to read the $j(k)$'s.

(e) Experiments on many physical problems (BVP) in computer shows that even if the assumed α's are quite far from the actual ones, in a few iterations (say 3 or 4) we are sufficiently near to the actual α's.

(f) It is more economical to consider that boundary as the first boundary (*i.e.*, initial boundary) where the number of given conditions are more than that on the other, since in that case the correction vector k is smaller in dimension.

References

1. Tikekar, V. G. .. .. "Some Flow and Heat Transfer Problems", Ph D Tcesis, I. I. Sc.. 1969.
2. Romanelli, M. J. .. .. "Math. Methods for Digital Computers", editee by A. Ralston and H. S. Wilf, John Wiley & Sons, 1965, 114.
3. Ralston, A. .. .. .. *Ibid*, 100.
4. Fox, L. .. .. .. "An Introduction to Numerical Linear Algebra" Clarendon Press, Oxford, 1964, 64.
5. Wilkinson, J. H. .. .. "The Algebraic Eigen-Value Problem", Claredon Press, Oxford, 1965, 189.
6. Sen, S. K. .. .. .. "Overcoming Failures of Triangular Decomposition Schemes for solving Linear Equations", to be published.