AN ADAPTIVE IMPLEMENTATION OF INTERPOLATION METHODS FOR BOUNDARY VALUE ORDINARY DIFFERENTIAL EQUATIONS*

R. P. TEWARSON and N. S. HUSLAK

Applied Mathematics and Statistics Department, State University of New York, Stony Brook, N.Y. 11794, U.S.A.

Abstract.

Various interpolation-based schemes are used to construct a variable order algorithm with local error control for the numerical solution of two-point boundary value problems. Results of computational experiments are presented to demonstrate the empirical relationship between prescribed local error and the resultant global error in the computed solution.

1. Introduction.

We consider the numerical solution of the system of non-linear differential equations

\[ y'(x) - f(x, y(x)) = 0 \]

with the two-point linear boundary conditions

\[ g(y(0), y(1)) = 0, \]

where \( y, f \) and \( g \) are functions with values in \( \mathbb{R}^n \), and \( 0 \leq x \leq 1 \). We assume that (1.1) and (1.2) have an isolated solution and \( f \) is smooth so that all the involved derivatives of this solution exist.

A number of algorithms have appeared in the literature that perform automatic mesh refinement and local truncation error approximation (see e.g. [1–3]). In the method proposed here, the original mesh (upon which the approx-

* Research supported by NIH Grant No. AM17593.
Received October 15, 1982, Revised January 25, 1983.
imate solution is ultimately given) is not at any time permanently refined. Instead, "pseudo-points" are added as needed to improve the accuracy of the solution approximation via interpolation techniques. These pseudo-points are then dropped from consideration as the algorithm proceeds into the next iteration. This feature is proposed as a principal advantage of the method over existing software, since adding points permanently to a mesh increases the dimensionality of the system with an attendant rise in computational effort and storage requirements.

Let us subdivide the $x$ range $[0, 1]$ into $n$ equal parts, such that $h = 1/n$ and $x_i = ih$, $i = 0, 1, .., n$. If we integrate equation (1.1) in the interval $[x_{i-1}, x_i]$, then we have

$$y(x_i) - y(x_{i-1}) - \int_{x_{i-1}}^{x_i} f(x, y(x)) \, dx = 0. \tag{1.3}$$

The integral in (1.3) can be evaluated by the various numerical schemes. For example, the well known trapezoidal rule (box scheme) which has local order of accuracy 3 leads to

$$\phi_i^{TR}(y) \equiv y_i - y_{i-1} - \frac{h}{2} (f_i + f_{i-1}) = 0, \quad i = 1, \ldots, n, \tag{1.4}$$

where $f_i = f(x_i, y_i)$, $y$ is now a vector with the components $y_i$ (the approximation at $x_i$) which in turn are vectors with $m$ components. From (1.2) we get

$$g(y_0, y_n) = 0, \tag{1.5}$$

which is now a system of $m$ equations in $2m$ variables. We have assumed that $g$ is linear and, therefore, half of its variables can be expressed in terms of the other half and then utilized to eliminate $m$ variables from (1.4). The resulting system of $mn$ equations and variables can be written as

$$\phi(y) = 0. \tag{1.6}$$

To solve (1.6), we select a starting vector $y^{(0)}$ and proceed iteratively, obtaining refined approximations $y^{(k)}$ to the solution vector. An iteration consists of three parts. First, as described in Section 2, we show how the Trapezoidal (TR), Simpson (SR) and Newton-Côtes (NC) methods can be used selectively to first estimate the local error and, based on this estimate, to choose the appropriate numerical integration formula to be employed in (1.3). The second part of the iteration is invoked in any subinterval in which the use of our selection of integration formulas is precluded by the estimated error. In Section 3 we