A CLASS OF SELF-STARTING METHODS FOR THE NUMERICAL SOLUTION OF $y'' = f(x, y)$

P. C. CHAKRAVARTI and P. B. WORLAND

1. Introduction.

The differential equation

(1.1) $y'' = f(x, y)$

with initial conditions

(1.2) $y(x_0) = y_0, \quad y'(x_0) = y'_0$

and systems of such equations form an important class which frequently appears e.g. in mechanics and astronomy. It is generally agreed that in solving (1.1) numerically it is better to solve it directly rather than to first reduce it to a system of two first order equations. See [6].

Various special methods have been developed for the solution of (1.1) (see e.g. [3, 4, 5, 8]). In [2] we proposed a more efficient method for (1.1) which proceeds in blocks of $n$ steps rather than one step at a time.

We propose here a modification of the method described in [2]. Both methods are "block" methods, and hence both are self-starting since each block is self-starting: only the values of $y$ and $y'$ at the first point of the block are necessary to get started.

The new method is an improvement in the sense that we can take one extra step with no increase in the number of function evaluations. This is important since in general these evaluations will take up much of the computational work. At the same time we also increase the accuracy of the starting value of $y$ for each block by a factor of the stepsize $h$. Thus for a six-point block we get a method which has a high order of accuracy, $O(h^7)$, but which requires less than two function evaluations per step.

This means that for roughly the same amount of work the block methods will provide more closely spaced solution values than methods requiring more evaluations per step. This is important if one is constructing a table of values or if interpolation at a later stage should be required.

In addition, we will examine the block approach for arbitrary $n$. As we shall see, there are many ways in which a block can be built up. In this paper we will reduce the number of possibilities somewhat by developing a uniform approach to the building process and by comparing various schemes analytically and through testing on examples.

Received Jan. 21, 1971.
2. Development.

Note that although results are given in terms of a single equation, everything here will easily generalize to a system of equations.

If we integrate (1.1) twice we obtain

\[ y(x_0 + kh) = y(x_0) + kh'y(x_0) + \int_{x_0}^{x_0+kh} (x_0 + kh - t)f(t, y(t)) \, dt \]

where \( h \) is the stepsize and \( k \) is an integer.

We obtain a solution by applying interpolatory integration rules which incorporate the weighting function \((x_0 + kh - t)\), and hence we obtain formulas with \( O(h^{n+2}) \) truncation error for all points in an \( n \)-point block.

We assume the existence of a sufficient number of derivatives of the solution for the formulas to be valid.

Let \( x_k = x_0 + kh \), let \( y_k \) be the approximation for the exact solution \( y(x_k) \), and let

\[ y_{kr} = y_0 + k_i h y_0' + \frac{h^2}{d_r} \sum_{i=0}^{n-1} w_i y_i'' , \quad r = 1, 2, \ldots, N - 1 \]

\[ y''_{kr} = f(x_{kr}, y_{kr}) , \quad r = 1, 2, \ldots, N - 1 \]

\[ y'_{kN} = y_0' + \frac{h}{d_N} \sum_{i=0}^{n-1} w_i, N y_i'' \]

represent the formulas used, where \( N \) depends on the length of the block.

3. The building process.

The building process can best be described by Fig. 1.

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \times )</th>
<th>total number of evaluations</th>
<th>truncation error, ( O(h) ), at computed point</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0 )</td>
<td>( x )</td>
<td>( 1 )</td>
<td>( 3 )</td>
</tr>
<tr>
<td>( \times 0 )</td>
<td>( \times )</td>
<td>( 2 )</td>
<td>( 5 )</td>
</tr>
<tr>
<td>( \times \times \times )</td>
<td>( \times )</td>
<td>( 3 )</td>
<td>( 5 )</td>
</tr>
<tr>
<td>( \times \times \times )</td>
<td>( \times )</td>
<td>( 4 )</td>
<td>( 5 )</td>
</tr>
<tr>
<td>( \times \times \times \times )</td>
<td>( \times \times \times \times )</td>
<td>( 5 )</td>
<td>( 7 )</td>
</tr>
<tr>
<td>( \times \times \times \times \times )</td>
<td>( \times \times \times \times \times )</td>
<td>( 6 )</td>
<td>( 7 )</td>
</tr>
<tr>
<td>( \times \times \times \times \times )</td>
<td>( \times \times \times \times \times )</td>
<td>( 7 )</td>
<td>( 7 )</td>
</tr>
<tr>
<td>( \times \times \times \times \times \times )</td>
<td>( \times \times \times \times \times \times )</td>
<td>( 8 )</td>
<td>( 7 )</td>
</tr>
<tr>
<td>( \times \times \times \times \times \times \times )</td>
<td>( \times \times \times \times \times \times \times )</td>
<td>( 9 )</td>
<td>( 7 )</td>
</tr>
</tbody>
</table>

Fig. 1