A Note on Trapezoidal Methods for the Solution of Initial Value Problems

By A. R. Gourlay*

Abstract. The trapezoidal rule for the numerical integration of first-order ordinary differential equations is shown to possess, for a certain type of problem, an undesirable property. The removal of this difficulty is shown to be straightforward, resulting in a modified trapezoidal rule. Whilst this latent difficulty is slight (and probably rare in practice), the fact that the proposed modification involves negligible additional programming effort would suggest that it is worthwhile. A corresponding modification for the trapezoidal rule for the Goursat problem is also included.

1. The Trapezoidal Rule. We consider the numerical solution of the initial value problem

$$(1.1) \qquad y' = f(x, y), \qquad y(x_0) = y_0$$

in the region $x_0 \leqq x \leqq X$. The trapezoidal rule for the numerical solution of (1.1) is given by

$$(1.2) \qquad y_{n+1} - y_n = \frac{h}{2} [f(x_{n+1}, y_{n+1}) + f(x_n, y_n)],$$

where $x_{n+1} = x_n + h$, h being the mesh length in the x direction. Equation (1.2) is a one-step implicit finite-difference method which is frequently employed for the numerical solution of (1.1). In fact, it is well known that it is the most accurate A-stable multistep method [1]. A method M is said to be A-stable if all solutions of M tend to zero as $n \to \infty$ when the method is applied with fixed positive h to any differential equation of the form

$$(1.3) \qquad y' = -\lambda y,$$

where λ is a complex constant with positive real part.

For such an equation, (1.2) can be seen to reduce to

$$y_{n+1} = \left[\frac{1 - \dfrac{h}{2} \lambda}{1 + \dfrac{h}{2} \lambda} \right] y_n.$$

The purpose of this note is simply to point out that there are certain problems, only slightly more general than (1.3), whose numerical solution by (1.2) is totally unaccep-

table for certain mesh lengths. We then point out that a simple alternative to the trapezoidal rule does not possess this undesirable property and, in fact, requires no additional computational effort. Since the trapezoidal rule is important with respect to the numerical integration of stiff systems, the proposed modification might be regarded as a useful safeguard.

To demonstrate the defect of (1.3) we consider it applied to the equation

$$(1.4) \qquad y' = -\lambda(x)y, \qquad \lambda(x) \geqq 0.$$

[The restriction of $\lambda(x)$ to be a real function involves no significant loss of generality.] For (1.4) the trapezoidal rule reduces to the recurrence

$$y_{n+1} = \left[\frac{1 - \frac{h}{2}\lambda(x_n)}{1 + \frac{h}{2}\lambda(x_{n+1})} \right] y_n.$$

For the recurrence to be acceptable, we require

$$(1.5) \qquad \left| \frac{1 - \frac{h}{2}\lambda(x_n)}{1 + \frac{h}{2}\lambda(x_{n+1})} \right| \leqq 1.$$

We shall show that, for certain functions $\lambda(x)$, Eq. (1.5) imposes a restriction on the mesh length h.

In fact, condition (1.5) requires

$$(1.6) \qquad h[\lambda(x_n) - \lambda(x_{n+1})] \leqq 4.$$

Condition (1.6) is certainly satisfied if $\lambda(x_n) \leqq \lambda(x_{n+1})$ but, if $\lambda(x_n) > \lambda(x_{n+1})$, then (1.6) restricts the mesh size to lie in the interval

$$(1.7) \qquad 0 < h \leqq 4\{\lambda(x_n) - \lambda(x_{n+1})\}^{-1}.$$

Condition (1.7) is most severe for those functions $\lambda(x)$ which are large and rapidly decaying. Therefore, there may be situations where the trapezoidal rule will not be satisfactory unless a sufficiently small step size is employed.

It is easily seen that the above restriction may be removed if Eq. (1.2) is replaced by

$$(1.8) \qquad y_{n+1} - y_n = \frac{h}{2}[f(x_{n+1/2}, y_{n+1}) + f(x_{n+1/2}, y_n)].$$

Unfortunately, method (1.8) requires twice as many function evaluations as (1.2) to carry out the integration of (1.1) over the interval $x_0 \leqq x \leqq X$. To remove this latter defect, we suggest that (1.2) be replaced by the method

$$(1.9) \qquad y_{n+1} - y_n = hf(x_{n+1/2}, \tfrac{1}{2}(y_{n+1} + y_n)),$$

which has the same order of accuracy as the normal trapezoidal rule. Method (1.9) is not a multistep method but can be regarded as an implicit Runge-Kutta method. In addition, it does not suffer from a restriction such as (1.7) and is just as straight-forward to implement as (1.2). It is worth noting that for problems of the form (1.3),

methods (1.2) and (1.9) are identical, but for quadrature problems $y' = -\lambda(x)$, method (1.9) reduces to the midpoint rule.

Since (1.9) involves no extra effort in implementation over (1.2) and does not suffer from any restriction on the mesh size, we suggest that it should be used in preference to (1.2). Our conclusion is motivated by the fact that (1.4) is a much better *local* model for (1.1) than is (1.3).

A simple example demonstrating the defect of the normal trapezoidal rule is given by the problem

$$y' = -\lambda(x)y, \qquad y(0) = 1,$$

where

$$\lambda(x) = \alpha^2(\beta - x) \qquad \text{for} \quad 0 \le x \le \beta,$$
$$= 0, \qquad x \ge \beta.$$

The solution to this problem is given by

$$y(x) = \exp\{-\alpha^2 x(\beta - x\,2)\}, \qquad 0 \le x \le \beta,$$
$$= \exp\{-\alpha^2\beta^2/2\}, \qquad x \ge \beta.$$

The condition (1.7) in this case requires $h \le 2/\alpha$. Computations were undertaken for the choice of constants

$$\alpha = 10, \qquad \beta = 100$$

and for values of $h = 0.150, 0.198, 0.200, 0.202, 0.250$. In this example, condition (1.7) requires that $h \le 0.2$ for $0 \le x \le \beta$, so that the final three runs should produce results which demonstrate nondamping of the solution of the trapezoidal rule. In this problem, the solution is decaying rapidly to a small quantity. Method (1.9) produced computed solutions which decreased in modulus from the starting value. Scheme (1.2) produced similar results for the first two values of h, though the rate of damping was smaller than that of (1.9). For $h = 0.200$ the computed solution was consistently ± 1 and for $h > 0.2$ the computed solution increased slowly in modulus from the starting value. It follows that the behaviour of (1.2) for $h \ge 0.2$ is not acceptable.

Thus, whereas (1.9) will reproduce the behaviour of the solution for problems of the form $y' = -\lambda(x)y$, the normal trapezoidal rule may fail to do so. Moreover, in the numerical solution of stiff systems, where (1.4) may be regarded as referring to one component of a diagonalized system, the deficiency of the trapezoidal rule will be present if any component corresponds to a $\lambda(x)$ which has $\lambda(x_n) > \lambda(x_{n+1})$. In such a situation, it would appear to be safer to use (1.9).

2. Analogon for the Goursat Problem. The close connection between the solution of an initial value problem for an ordinary differential equation and the solution of the Goursat problem

$$u_{xy} = f(x, y, u, u_x, u_y)$$

(2.1) $\qquad u(x, 0) = \sigma(x), \qquad u(0, y) = \tau(y), \qquad \sigma(0) = \tau(0)$

$$0 \le x \le a, \qquad 0 \le y \le b$$

has often been observed (Stetter and Törnig [4]). The analogon for (2.1) of the trapezoidal rule for (1.1) is given by

$$u_{i+1,j+1} + u_{i,j} - u_{i+1,j} - u_{i,j+1} = \frac{h^2}{4}\,[f_{i+1,j+1} + f_{i,j} + f_{i,j+1} + f_{i+1,j}],$$

where $f_{i,j} = f(x_i, y_j, u_{i,j}, (u_x)_{i,j}, (u_y)_{i,j})$ with suitable difference replacements for $(u_x)_{i,j}$, $(u_y)_{i,j}$. Higher order schemes have also been proposed in various papers, including Day [2], Jain and Sharma [3], Stetter and Törnig [4].

In this section we propose the analogon of (1.9) for the solution of (2.1). This is given by

$$(2.2) \qquad u_{i+1,j+1} + u_{i,j} - u_{i+1,j} - u_{i,j+1} = h^2 f(x_{i+1/2}, y_{j+1/2}, \hat{u}, \hat{p}, \hat{q}),$$

where

$$\hat{u} = \tfrac{1}{4}[u_{i+1,j+1} + u_{i,j} + u_{i+1,j} + u_{i,j+1}],$$

$$(2.3) \qquad \hat{p} = \frac{1}{2h}\,[u_{i+1,j+1} - u_{i,j+1} + u_{i+1,j} - u_{i,j}],$$

$$\hat{q} = \frac{1}{2h}\,[u_{i+1,j+1} - u_{i+1,j} + u_{i,j+1} - u_{i,j}],$$

and the truncation error is of order h^4. In fact, (2.2) is implicit and is solved at each point by an iterative process which uses the starting values

$$\hat{u}^{(0)} = \tfrac{1}{2}[u_{i+1,j} + u_{i,j+1}],$$

$$\hat{p}^{(0)} = \frac{1}{h}\,[u_{i+1,j} - u_{i,j}],$$

$$\hat{q}^{(0)} = \frac{1}{h}\,[u_{i,j+1} - u_{i,j}],$$

and iterates with

$$u_{i+1,j+1}^{(m+1)} + u_{i,j} - u_{i+1,j} - u_{i,j+1} = h^2 f(x_{i+1/2}, y_{j+1/2}, \hat{u}^{(m)}, \hat{p}^{(m)}, \hat{q}^{(m)}),$$

where

$$\hat{u}^{(m)} = \tfrac{1}{4}[u_{i+1,j+1}^{(m)} + u_{i,j} + u_{i+1,j} + u_{i,j+1}],$$

and where similar expressions for $\hat{p}^{(m)}$, $\hat{q}^{(m)}$ are derived from (2.3).

It may be shown that the above process is convergent for sufficiently small h and in practice the convergence is rapid, occurring after one or two iterations. A comparison was undertaken with the results given in [2]. These experiments showed that the above technique was as good as the algorithm of Day and in certain examples considerably better. In fact, for the three problems considered in [2], the present algorithm produced results whose relative errors were respectively 2, 1/16, 4/5 times the errors given in [2]. An additional advantage of the present algorithm is its simplicity.

3. Conclusion.   A defect of the trapezoidal rule which, though rare in practice, actually exists (and indeed may readily exist in strongly stiff problems) has been

exhibited. A modified trapezoidal rule has been proposed. An analogous method for the Goursat problem has also been given and shown to be simple to use and accurate in practice.

Scientific Centre
IBM (U.K.) Limited
Peterlee
Co. Durham, England

1. G. G. Dahlquist, "A special stability problem for linear multistep methods," *Nordisk Tidskr. Informations-Behandling*, v. 3, 1963, pp. 27–43. MR **30** #715.
2. J. T. Day, "A Runge-Kutta method for the numerical solution of the Goursat problem in hyperbolic partial differential equations," *Comput. J.*, v. 9, 1966, pp. 81–83. MR **33** #890.
3. M. K. Jain & K. D. Sharma, "Cubature method for the numerical solution of the characteristic initial value problem $u_{xy} = f(x, y, u, u_x, u_y)$," *J. Austral. Math. Soc.*, v. 8, 1968, pp. 355–368.
4. H. J. Stetter & W. Törnig, "General multistep finite difference methods for the solution of $u_{xy} = f(x, y, u, u_x, u_y)$," *Rend. Circ. Mat. Palermo*, v. 12, 1963, pp. 281–298. MR **29** #6644.