JOURNAL OF COMPUTATIONAL PHYSICS 29, 141–144 (1978)

Note

Automated Computation of Modified Equations*

The calculation of the modified equation has proved to be a valuable aid in assessing the accuracy and stability of a given difference equation. The technique consists of expanding the difference equation in a Taylor series about some suitably chosen expansion center. The result is a differential equation, the so-called modified equation, and consists of the original differential equation that is modeled by the difference equation plus truncation error terms (for example, see Ref. [1]). For some purposes it is necessary to eliminate time derivatives from the truncation error terms by differentiating and substituting the modified equation into itself. The same procedure may be applied to systems of equations, making it suitable for areas of research such as numerical fluid mechanics.

For complicated problems, an algebraic computer system is almost essential for the computation of modified equations. Massive amounts of algebra are required, the manipulations are quite tedious, and there is always the possibility of blunders. Warming and Hyett [2] have used FORMAC to treat the linear case. In this note, we describe two ALTRAN programs for nonlinear systems [3]. These programs were written with numerical fluid dynamics applications in mind, but are by no means limited to that application. The first program performs the Taylor series expansion for a single difference equation in two independent variables, and the modified equation is produced as input for the second program. The second program eliminates time derivatives from the truncation error terms. The resulting equation can be examined for undesirable properties, such as errors of negative order. Modified equations for several finite difference approximations to a given differential equation may also be analyzed to help choose the optimum numerical scheme for a particular application. Another use is the prediction of numerical stability conditions, which can then be used as the basis for techniques for stabilizing finite difference algorithms.

As a simple example of a problem that can be run with these programs, consider the one-dimensional continuity equation for a fluid dynamics problem:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = \frac{\partial}{\partial x}\left(\xi \frac{\partial \rho}{\partial x}\right), \tag{1}$$

where ρ is the density and u is the velocity. The diffusivity ξ may represent turbulence

* Work supported by U.S. Energy Research and Development Administration Contract W-7405-ENG 36.

0021-9991/78/0291–0141$02.00/0

effects or may be used as an artificial mass diffusion for numerical stability of the finite difference scheme, such as in the ICE algorithm [4]. One simple finite difference approximation to Eq. (1) is

$$
\begin{aligned}
&\frac{\rho_i^{n+1} - \rho_i^n}{\delta t} + \frac{\theta}{2\delta x}\left[(\rho_{i+1}^{n+1} + \rho_i^{n+1})\, u_{i+1/2}^{n+1} - (\rho_i^{n+1} + \rho_{i-1}^{n+1})\, u_{i-1/2}^{n+1}\right] \\
&\quad + \frac{(1-\theta)}{2\delta x}\left[(\rho_{i+1}^n + \rho_i^n)\, u_{i+1/2}^n - (\rho_i^n + \rho_{i-1}^n)\, u_{i-1/2}^n\right] \\
&\quad = \frac{1}{\delta x^2}\left[\xi_{i+1/2}^n(\rho_{i+1}^n - \rho_i^n) - \xi_{i-1/2}^n(\rho_i^n - \rho_{i-1}^n)\right],
\end{aligned}
\tag{2}
$$

where θ, $0 \leqslant \theta \leqslant 1$, is a time-centering parameter and ρ_i^n is the density at grid point i and time level n. The velocity $u_{i+1/2}$ and diffusivity $\xi_{i+1/2}$ are defined at a point halfway between grid points i and $i + 1$, as in the ICE method. Expansion of Eq. (2) yields the modified equation

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= \frac{\partial}{\partial x}\left(\xi \frac{\partial \rho}{\partial x}\right) - \frac{\delta t}{2}\left[\frac{\partial^2 \rho}{\partial t^2} + 2\theta\left(u \frac{\partial^2 \rho}{\partial t\, \partial x} + \frac{\partial u}{\partial t}\frac{\partial \rho}{\partial x} + \frac{\partial u}{\partial x}\frac{\partial \rho}{\partial t} + \rho \frac{\partial^2 u}{\partial t\, \partial x}\right)\right] \\
&\quad - \frac{\partial x^2}{24}\left[4u \frac{\partial^3 \rho}{\partial x^3} + 6 \frac{\partial u}{\partial x}\frac{\partial^2 \rho}{\partial x^2} + 3 \frac{\partial^2 u}{\partial x^2}\frac{\partial \rho}{\partial x}\right. \\
&\quad \left. + \rho \frac{\partial^3 u}{\partial x^3} - 2\xi \frac{\partial^4 \rho}{\partial x^4} - 4 \frac{\partial \xi}{\partial x}\frac{\partial^3 \rho}{\partial x^3} - 3 \frac{\partial^2 \xi}{\partial x^2}\frac{\partial^2 \rho}{\partial x^2} - \frac{\partial^3 \xi}{\partial x^3}\frac{\partial \rho}{\partial x}\right].
\end{aligned}
\tag{3}
$$

Assume that we are interested in the criterion for diffusional stability. It is necessary to eliminate all time derivatives from the truncation error terms of Eq. (3) by substituting derivatives of Eq. (3) and the modified equation for u into Eq. (3). One must also approximate the pressure gradient in the momentum equation by the density gradient times the square of the sound speed, c^2. The lowest order diffusional truncation errors are found to be

$$
\zeta \frac{\partial^2 \rho}{\partial x^2} = \left[(2\theta - 1)\frac{\delta t}{2}(u^2 + c^2) - \frac{\delta x^2}{4}\frac{\partial u}{\partial x}\right]\frac{\partial^2 \rho}{\partial x^2}\,. \tag{4}
$$

If the effective diffusion coefficient ζ is negative, the difference equation (2) is unstable for $\xi = 0$. The original ICE technique was stabilized by providing a constant ξ such that $\xi + \zeta > 0$ in each cell at each time step. A more accurate, less diffusive stabilization technique is to compute an explicit finite difference estimate of ζ at the half integer grid points and set

$$\begin{aligned} \xi^n_{i+1/2} &= -(1+\beta)\,\zeta^n_{i+1/2} \qquad \text{if} \quad \zeta^n_{i+1/2} < 0 \\ &= -(1-\beta)\,\zeta^n_{i+1/2} \qquad \text{if} \quad \zeta^n_{i+1/2} \geqslant 0, \end{aligned} \tag{5}$$

where β, $0 \leqslant \beta \leqslant 1$, is a free parameter that is normally set to unity. This algorithm is described in detail for all the equations in a two-dimensional fluid dynamics program in Ref. [5], where numerical examples are also presented. Although this example can be done by hand, it is tedious enough to make one appreciate the great convenience of automating the Taylor series expansions and time derivative eliminations. We emphasize that this simple example was chosen to preseve the simplicity of our presentation and that the programs have been applied to more complex problems. And even for simple cases, the lack of blunders is an advantage for those of us who do not enjoy doing mechanical algebraic manipulations.

The ALTRAN programs represent a compromise among generality, efficiency, and computer resource requirements that allows them to be used on modest computers. For example, the basic time derivative elimination program makes a single differentiation and substitution of the modified equation into itself, so several runs may be necessary to eliminate all of the time derivatives. Reference [3] contains program modifications to allow as many passes through the algorithm as needed in a single run although running time increases. The user can also lift the restriction to two independent variables if the necessary workspace and computer time are available. Reference [3] moreover contains modifications to make the elimination code run faster and use less workspace, although there is a small penalty in the generality of the program. These possibilities illustrate the fact that the basic codes have the flexibility to be modified to fit a user's requirements and computing facilities.

The sample problem in this note was run on a CDC 7600. The expansion code used 32 sec of central processor time and 36,513 words of workspace. The elimination code needed 3 runs for the system of two modified equations, which used a total of 200 sec and a maximum of 85,053 words of workspace. Although the run times and memory requirements are problem dependent, these values are typical of problems of the same complexity as the example. The workspace and running time requirements increase at least linearly with the number of terms and the rate of increase can even be exponential for some types of problems.

Reference [3] contains a more detailed description, listings, test problems, and flow charts for these programs, as well as several examples of simple applications. This report is available from the authors and from the National Technical Information Service.

References

1. C. W. Hirt, *J. Computational Physics* **2** (1968), 339.
2. R. F. Warming and B. J. Hyett, *J. Computational Physics* **14** (1974), 159.
3. L. D. Cloutman and L. W. Fullerton, "Automated Heuristic Stability Analysis for Nonlinear Equations," Los Alamos Scientific Laboratory Report LA-6885-MS, 1977.

4. F. H. HARLOW AND A. A. AMSDEN, *J. Computational Physics* **8** (1971), 197.
5. W. C. RIVARD, O. A. FARMER, T. D. BUTLER, AND P. J. O'ROURKE, "A Method for Increased Accuracy in Eulerian Fluid Dynamics Calculations," Los Alamos Scientific Laboratory Report LA-5426-MS, 1973.

RECEIVED: September 7, 1977; REVISED: December 21, 1977

L. D. CLOUTMAN

Theoretical Division, MS-216
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 81545

L. W. FULLERTON

Computer Science and Services Division, MS-265
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545