JOURNAL OF COMPUTATIONAL PHYSICS 26, 113–116 (1978)

Note

FORTRAN Subprograms for Finite-Difference Formulas

1. Introduction

In this note two FORTRAN subprograms, FINDIF and FINDF, for finding integer finite-difference formulas are described. Such formulas can be found in tables [1] or derived by hand without too much difficulty if combinations of the following are satisfied:

(i) The derivatives approximated are lower order ($\leqslant 4$),
(ii) the approximation is on a uniform mesh,
(iii) the derivative point is "centered,"
(iv) the required order of accuracy is low ($\leqslant 4$),
(v) the dimensionality of the mesh is at most 2.

With the advent of the new parallel and pipeline computers, more complex difference methods requiring formula outside the realm of (i)–(v) above have become feasible. For example:

(i) numerical solutions to PDEs where deferred correction techniques [2] require finite-difference formulas to approximate the higher-order partial derivatives in truncation error expressions;

(ii) number PDE models where mesh refinements (e.g., see [3]) or interactions between coarse and fine mesh geometries (such as a limited area weather model [4]) require finite-difference formulas on nonuniform grids.

These programs were designed for those cases when fixed-order finite-difference formulas are required prior to coding a PDE model. They were not designed to interact dynamically with such a model, as might be required with a variable-order finite-difference package.

2. Program Description

Subroutine FINDIF computes the integer coefficients $j(1), j(2),..., j(n)$, L for the finite-difference approximation to the mth derivative at $x = p \cdot h$ of a function $f(x)$ given at $k(1) \cdot h,..., k(n) \cdot h$. The form of the approximation is

$$d^m f(p \cdot h)/dx^m = \sum_{i=1}^{n} j(i) \cdot f(k(i) \cdot h)/Lh^m + O(h^r). \quad (1)$$

0021-9991/78/0261-0113$02.00/0

The mesh geometry and derivative point can be any integer multiples of some mesh increment h (h is a notational convenience and not an input parameter). Any stencil consisting of rational numbers can be put in this form. The asymptotic error estimate r is also returned by the program. Subroutine FINPDF is the multidimensional extension of FINDIF. Assume now that f is a real-valued function given on a t-dimensional grid. Denote the points where f is given in the ith dimension by $k^i(j) \cdot h(i)$, $j = 1,\ldots, n_i$. No special relationship need exist between $h(1),\ldots, h(t)$ (the basic mesh increments in each dimension). Assume an approximation to the $(m_1,\ldots, m_t)$ mixed partial derivative at $x_1 = p(1) \cdot h(1),\ldots, x_t = p(t) \cdot h(t)$ is required. The increment multiplies $k^i(j)$, $j = 1,\ldots, n(i)$, $p(i)$, $i = 1,\ldots, t$, can be any integer values. FINPDF computes the integers $j(i_1,\ldots, i_t)$, L, $t(i)$ in the difference formula

$$\partial^m f(x_1,\ldots, x_t)/\partial x_1^{m(1)} \cdots x_t^{m(t)} \qquad (m = m(1) + \cdots + m(t))$$

$$= \sum_{i_1=1}^{n(1)} \cdots \sum_{i_t=1}^{n(t)} \frac{j(i_1,\ldots, i_t) \cdot f(k^1(i_1) \cdot h_1,\ldots, k^t(i_t) \cdot h_t)}{L h_1^{m(1)} \cdots h_t^{m(t)}} + \sum_{i=1}^{t} O(h_i^{r(i)}). \tag{2}$$

By differentiating the LaGrange interpolating polynomial [5], one can derive the following expression for the jth rational coefficient in the finite-difference formula as sums and products of rational numbers.

$$\prod_{i=1}^{m} \left(\frac{i}{k(j) - k(\Psi(j, i))}\right) * \sum_{\theta} \prod_{i=1}^{n-m} \left(\frac{p - k(\Psi(j, \theta(i)))}{k(j) - k(\Psi(j, i + m))}\right). \tag{3}$$

The sum on the right is over all injections $\theta\colon \{1, 2,\ldots, n - m\} \to \{1,\ldots, n\}$ (i.e., over all $(n - m)$-order subsets of $\{1,\ldots, n\}$), and the index function Ψ is defined by

$$\begin{aligned} \Psi(j, i) &= i && \text{if } i < j \\ &= i + 1 && \text{if } i \geqslant j \end{aligned} \qquad (j = 1,\ldots, n). \tag{4}$$

This expression is used to compute each coefficient recursively by using only integer (fixed-point) aritmetic. This approach is not computationally optimal but eliminates roundoff error considerations and allows numerator–denominator cancellation prior to multiplication of each pair of rational numbers. This minimizes the likelihood of integer overflow (an error condition that is flagged). Runs on the Control Data 7600 with $m = 1$, $i = (n + 1)/2$ and uniform meshes first overflowed when $n = 33$; runs with $m = n - 1$, $i = 1$ first overflowed when $n = 52$.

As noted, the mesh geometry inputs to FINDIF, FINPDF are represented (modulo increments h_i) by integer tuples. The formulas generated are invariant under integer coefficient linear transformations and cancellations of common factors in the integer tuples. For example, $k(1) = 8400$, $k(2) = 9600$, $k(3) = 12{,}000$, $k(4) = 14{,}400$, $p = 7200$ yield (for any $m \leqslant 3$) the same difference formula as $k(1) = 2$, $k(2) = -1$, $k(3) = 1$, $k(4) = 3$, $p = 0$. Such reductions should be performed prior to calling the subprograms since they reduce the likelihood of integer overflow.

The time required to compute a one-dimensional difference formula is proportional to

$$T(n, m) = n \cdot m \cdot (n - m) \cdot \binom{n}{m}. \tag{5}$$

The last factor is dominant when n is large and m is in the middle portion of the integer interval $[1, n]$. In these cases (which should never occur in practice) FINDIF is prohibitively slow. For example, $T(30,15)/T(30,1)$ is approximately 10^8.

3. Examples

Example 1. The truncation error for the Crank–Nicholson scheme [6] applied to the heat equation $U_t = U_{xx}$ (on a uniform mesh in space and time, $x_i = i\Delta x$, $i = 0, 1, 2, \ldots$; $t_n = n\Delta t$, $n = 0, 1, 2, \ldots$) is

$$T_j^n = (\Delta x^2 \cdot \partial^4 U_j^{n+1/2}/\partial x^4 + \Delta t^2 \cdot \partial^6 U_j^{n+1/2}/\partial x^6)/12. \tag{6}$$

Here $U_j^{(n+1)/2}$ is U at x_j, $t_{n+1/2}$. A second-order approximation to T_j^n would require, in particular, the difference formula for $\partial^6 U(x)/\partial x^6$ at the left boundary $x_0 = 0$. The eight points $x_0, x_1, \ldots, x_7$ are needed for an $O(\Delta x^2)$ approximation. If we take $n = 8$, $k(1) = 0$, $k(2) = 1$, $k(3) = 2$, $k(4) = 3$, $k(5) = 4$, $k(6) = 5$, $k(7) = 6$, $k(8) = 7$, $m = 6$, and $p = 0$, FINDIF gives

$$(4U_1 - 27U_2 + 78U_3 - 125U_4 + 120U_5 - 69U_6 + 22U_7 - 3U_8)/\Delta x^6 \tag{7}$$

and $r = 2$. Setting $p = 1, 2, 3$ will yield the other left-boundary formulas.

Example 2. Suppose we are simulating a limited-area weather model using sixth-order finite differencing. The simple hyperbolic equation $U_t = -U_x$ is solved simultaneously for a coarse mesh function $U^c(x)$ (given at..., $-8h$, $-4h$, 0, $4h$, $8h$,...) and a fine mesh function $U^F(x)$ (given at h, $3h$, $7h$,...). To compute $\partial U^F(x)/\partial x$ at $x = h$ (the fine mesh left boundary), we choose the seven closest points from the two meshes to base the difference formula on. Taking $n = 7$, $k(1) = -4$, $k(2) = 0$, $k(3) = 1$, $k(4) = 3$, $k(5) = 4$, $k(6) = 5$, $k(7) = 7$, $m = 1$, and $p = 1$, FINDIF returns the coefficients below.

$$\frac{\left[\begin{array}{l}6 \cdot U_{-4h}^c - 1980 \cdot U_0^c - 231 \cdot U_h^F + 4950 \cdot U_{3h}^F \\ \qquad - 3850 \cdot U_{4h}^c + 1155 \cdot U_{5h}^F - 50 \cdot U_{7h}^c\end{array}\right]}{4620 \cdot h} + O(h^6). \tag{8}$$

Example 3. The truncation error, resulting from the usual second-order finite-difference approximation to the elliptic equation with cross derivative

$$a(x, y)U_{xx} + b(x, y)U_{xy} + c(x, y)U_{yy} = g(x, y) \tag{9}$$

on a grid $x_i = i\Delta x(i = 0,\ldots, N)$, $y_j = j \cdot \Delta y(j = 0,\ldots, M)$, includes the term $\Delta x^2 \partial^4 U(x, y)/\partial x^3 \partial y$. Applying deferred corrections, to increase the accuracy to fourth order, requires second-our difference approximations to this term at all nonspecified grid points. At $(\Delta x, 0)$ we take $n(1) = 5$, $k^1(1) = 0$, $k^1(2) = 1$, $k^1(3) = 2$, $k^1(4) = 3$, $k^1(5) = 4$, $m(1) = 3$, $p(1) = 1$, and $n(2) = 3$, $k^2(1) = 0$, $k^2(2) = 1$, $k^2(3) = 2$, $m(2) = 1$, $p(2) = 0$. With this choice FINPDF yields

$$\begin{aligned} &\partial^4 U(\Delta x, 0)/\partial x^3 \partial y \\ &\quad = [-27 \cdot U_{1,1} - 108 \cdot U_{2,1} - 9 \cdot U_{3,1} + 90 \cdot U_{4,1} + 54 \cdot U_{5,1} \\ &\qquad + 3 \cdot U_{1,2} + 12 \cdot U_{2,2} + 1 \cdot U_{3,2} - 10 \cdot U_{4,2} - 6 \cdot U_{5,2} \\ &\qquad + 24 \cdot U_{1,3} + 96 \cdot U_{2,3} + 8 \cdot U_{3,3} - 80 \cdot U_{4,3} - 48 \cdot U_{5,3}] \\ &\quad \div \Delta x^3 \Delta y + O(\Delta x^2 + \Delta y^2) \end{aligned} \tag{10}$$

The taks performed by FINDIF and FINPDF might be handled with more ease by using a symbolic manipulation language. The implementation described in this note makes them immediately available to anyone with a rudimentary knowledge of FORTRAN. The programs and more details can be obtained by writing directly to the author at NCAR.

References

1. "Handbook of Mathematical Functions," U.S. Dept. of Commerce (National Bureau of Standards), Applied Mathematics Series, Vol. 55, Aug. 1966.
2. V. Pereyera, *Math. Comp.* **24** (1970), 771.
3. M. Ciment and R. Sweet, *J. Computational Phys.* **12** (1973), 513.
4. G. Browning, H. O. Kreiss, and J. Oliger, *Math. Comp.* **27** (1973), 29.
5. E. Isaacson and H. Keller, "Analysis of Numerical Methods," Wiley, New York, 1966.
6. R. D. Richtmyer and K. W. Morton, *in* "Difference Methods for Initial-Value Problems," (R. D. Richtmeyer, Ed.), 2nd ed., Wiley-Interscience, New York, 1967.

Received: November 29, 1976; Revised February 25, 1977

John C. Adams,

*National Center for Atmospheric Research,**
Boulder, Colorado 80307

* The National Center for Atmospheric Research is sponsored by the National Science Foundation.