MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0363959 7

ornl

ORNL/TM-13135

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

A Fast Multipole Transformation for Global Climate Calculations

J. A. Holmes
Z. Wang
J. B. Drake
B. F. Lyon
W.-T. Chen

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.
MCN-7969 (3 9-77)

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-13135

402

Computational Physics and Engineering Division

A FAST MULTIPOLE TRANSFORMATION FOR GLOBAL CLIMATE CALCULATIONS

J. A. Holmes
Z. Wang[1]
J. B. Drake
B. F. Lyon
Wen-Tao Chen[1]

1. GLCA/ACM Student, program administered by Oak Ridge Institute for Science and Energy

DATE PUBLISHED - January 1996

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6418
managed by
LOCKHEED MARTIN ENERGY RESEARCH CORP.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-96OR22464

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0363959 7

CONTENTS

ABSTRACT v

1. INTRODUCTION 1

2. MOTIVATION 3

3. FAST MULTIPOLE TRANSFORMATION IN ONE DIMENSION 8

4. RESULTS 15

5. CONCLUSIONS 21

6. REFERENCES 22

A Fast Multipole Transformation for Global Climate Calculations

J. A. Holmes, Z. Wang, J. B. Drake, B. F. Lyon, and Wen-Tao Chen

ABSTRACT

A fast multipole transformation is adapted to the evaluation of summations that occur in global climate calculations when transforming between spatial and spherical harmonic representations. For each summation, the timing of the fast multipole transformation scales linearly with the number of latitude gridpoints, but the timing for direct evaluations scales quadratically. In spite of a larger computational overhead, this scaling advantage renders the fast multipole method faster than direct evaluation for transformations involving greater than approximately 300 to 500 gridpoints. Convergence of the fast multipole transformation is accurate to machine precision. As the resolution in global climate calculations continues to increase, an increasingly large fraction of the computational work involves the transformation between spatial and spherical harmonic representations. The fast multipole transformation offers a significant reduction in computational time for these high-resolution cases.

1. INTRODUCTION

Numerical global climate calculations involve the evolution of nonlinear partial differential equations in space and time.[1-3] Typically, the dynamical equations describe the evolution of physical fields such as temperature, fluid velocity, pressure, density, or moisture content. These are solved in a spatial domain that consists of a spherical shell defined to include the earth's atmosphere, ocean, land surface, and/or ice pack. The description of this domain is conveniently given in terms a coordinate system (η,ϕ,λ), where η is some chosen monotone function of the height or depth in the medium (or spherical radius), $\phi(-\pi/2 \le \phi \le \pi/2)$ is the angle of latitude, and $\lambda(0 \le \lambda \le 2\pi)$ is the angle of longitude. An alternative description can be achieved by using a complete set of functions,[4,5] such as the spherical harmonics, to replace the spherical surface coordinates ϕ and λ, while retaining the vertical coordinate η. Both representations prove useful for solving the dynamical equations. The nonlinear terms, such as advection, are most conveniently calculated by direct evaluation at the gridpoints in the (η,ϕ,λ) coordinate system. However, these terms take the form of unwieldy convolutions when evaluated in terms of a function space expansion, such as spherical harmonics. Linear terms, such as diffusion, are compactly represented in function space. Derivatives in the ϕ and λ coordinates can be evaluated exactly using recursion relations between the spherical harmonics, but an evaluation of derivative terms in the (η,ϕ,λ) coordinate system introduces discretization errors.[6] Linear eigenfunctions also tend to be localized in terms of a small number of spherical harmonics; their calculation in the (η,ϕ,λ) coordinate system requires the full functional representation in the coordinate domain. Apart from computational considerations, the coordinate and function space representations provide useful and complementary diagnostic information. Visualization and intuitive understanding of many phenomena are best seen in coordinate space; however spectral and turbulent dynamical questions are best studied in function space.

In order to utilize the advantages of both representations discussed above, computational schemes have been developed to solve the nonlinear pieces of the dynamics in coordinate space and the linearized pieces in function space.[1,3] To couple the nonlinear and linear pieces together, it is necessary to transform between coordinate and function space representations. The spherical harmonics consist of products of complex exponential functions of the longitude $e^{im\lambda}$ and associated Legendre functions $P_n^m(\mu)$ of the latitude $(\mu \equiv \sin\phi)$, where m and n are integers. Note that on a spherical surface the complex exponentials are periodic functions of longitude with period $2\pi/m$, and hence are well defined. Thus the function space representation factors into products of functions of longitude and functions of latitude, and the transformation between coordinate and function space can also be factored into separate longitude and latitude transformations.

The longitude transformation relates the fields expressed as functions of λ to their expressions as Fourier series in $e^{im\lambda}$. This transformation can be carried out very efficiently using the method of fast Fourier transforms (FFTs).[7,8] For each field these FFTs reduce the computational work from $O(N \cdot M^2)$ to $O(N \cdot M \cdot \ln M)$, where N represents the number of latitude gridpoints, M represents the number of longitude gridpoints, and O denotes that the computational work scales as its argument.

The latitude transformation relates the fields expressed as functions of ϕ to their expressions as series in $P_n^m(\mu)$. Numerically this transformation can be carried out using the

properties of the associated Legendre functions,[4,5] together with Gaussian integration techniques,[8] which will be discussed later. With straightforward evaluation of the resulting expressions, the scaling of computational work with the number of gridpoints for the transformation of each field is $O(M \cdot N^2)$, so that for higher resolutions this transformation increasingly dominates the computational work. Although it is not in general possible to derive fast versions of transformations between function and coordinate representations, it will be shown in this report that a fast transformation can be derived between functions of ϕ and $P_n^m(\mu)$. Such a transformation will be obtained by combining the transformation just described with a fast multipole algorithm developed for particle simulations.[9] This method combines partitioning of the domain of μ with Taylor series to obtain a transformation that scales as $O(M \cdot N)$ in computational work. Because of the convergence of the Taylor expansions used here, the fast multipole algorithm can be made to converge to the direct summation, limited only by machine precision. Because of the computational overhead associated with the fast multipole method, a direct evaluation of the summations is most efficient for transforming between coordinate and function space representations in low-resolution calculations. However, the favorable scaling of the fast multipole technique leads to a substantial reduction in computing time for high-resolution calculations. In addition to demonstrating the above scalings, and other properties of the fast multipole transformation that will be derived later, this work shows that the resolution at which the fast multipole method becomes more efficient than the direct summation is at approximately 300 latitude gridpoints. Such resolutions are now being used in grand challenge global climate calculations.

The remainder of this discussion is organized as follows: the occurrence and form of the latitude transformation in climate calculations will be presented in Sect. 2 as a motivation for the fast multipole algorithm; the details of the fast multipole algorithm will be considered in Sect. 3; numerical calculations will be presented in Sect. 4 to demonstrate and compare the properties of the fast multipole algorithm with those of direct summation; and Sect. 5 will highlight our conclusions.

2. MOTIVATION

The dynamical equations in global climate calculations involve the transport of various physical quantities, such as mass, momentum (or velocity), pressure, temperature, moisture content, or salinity, to name a few, through various media, including the atmosphere, ocean, or cryosphere (ice). These equations are derived and discussed in great detail by various authors,[1-3] and a detailed treatment will not be given here. For the purpose of motivating the fast multipole transform, let us consider a 'generic' transport equation, one that embodies the basic characteristics present in all the transport equations, as pertains to the fast multipole transform. Assuming a physical field $f(\eta,\mu,\lambda,t)$, where t is the time, the generic dynamical equation takes the form

$$\partial f / \partial t = -\partial f / \partial \eta \cdot d\eta / dt - 1 / \{a(1-\mu^2)\} \cdot \partial (Uf) / \partial \lambda - 1 / a \cdot \partial (Vf) / \partial \mu - \nu \cdot \nabla^2 s + q \,, \quad (1)$$

where a is the radius of the earth, U is the longitudinal velocity, V is the latitudinal velocity, ν is a constant, s is a potential function for f, and q is a source function for f. Hence on the right-hand side of Eq. (1), the potential and source terms are linear, but the advective terms (the first term in the vertical velocity component and the second and third terms in the horizontal velocity components U and V) are nonlinear.

The vertical, or radial, dynamics is superfluous to the present consideration, namely, the transformation between coordinate space and function space on spherical surfaces. For this reason we drop the radial terms (η dependence) from Eq. (1) and consider the simpler "shallow water equation"

$$\partial f / \partial t = -1 / \{a(1-\mu^2)\} \cdot \partial (Uf) / \partial \lambda - 1 / a \cdot \partial (Vf) / \partial \mu - \nu \cdot \nabla^2 s + q \,, \quad (2)$$

where now the operator $\nabla^2 = 1/a^2 \cdot [\partial / \partial \mu \{(1-\mu^2) \cdot \partial / \partial \mu\} + 1/(1-\mu^2) \cdot \partial^2 / \partial \lambda^2]$ is restricted to the spherical surface of radius a. Equation (2) is sufficiently general to illustrate the issues that lead to the fast multipole transformation.

Equation (2) is written in the coordinate representation with independent variables t, μ, and λ and fields f, U, V, s, and q that are functions of the independent variables. To express Eq. (2) in function space, we expand all fields in terms of associated Legendre functions and complex exponentials as follows (suppressing the functional dependence on t): For any function $y(\mu,\lambda,t)$ we have

$$\begin{aligned} y(\mu,\lambda) &= \sum_{m=-\infty}^{\infty} \sum_{n=|m|}^{\infty} y_n^m P_n^m(\mu) e^{im\lambda} \\ &= \sum_{m=-\infty}^{\infty} y^m(\mu) e^{im\lambda} \,, \end{aligned} \quad (3)$$

where $y^m(\mu) = \sum_{n=|m|}^{\infty} y_n^m P_n^m(\mu)$. When written in a function space representation using Eq. (3) and the orthogonality and projection properties of complex Fourier series and associated Legendre functions,[4] Eq. (2) takes the following forms:

$$\sum_{n=|m|}^{\infty} df_n^m / dt \cdot P_n^m(\mu) = -im / \{a(1-\mu^2)\} \cdot \sum_{n=|m|}^{\infty} (Uf)_n^m P_n^m(\mu) - 1/a \cdot \sum_{n=|m|}^{\infty} (Vf)_n^m dP_n^m(\mu)/d\mu$$
$$+\nu / a^2 \cdot \sum_{n=|m|}^{\infty} n(n+1) s_n^m P_n^m(\mu) + \sum_{n=|m|}^{\infty} q_n^m P_n^m(\mu) \tag{4}$$

or

$$\partial f^m(\mu)/\partial t = -im / \{a(1-\mu^2)\} \cdot (Uf)^m(\mu) - 1/a \cdot \partial (Vf)^m(\mu)/\partial\mu$$
$$-\nu / a^2 \cdot [\partial / \partial\mu \{(1-\mu^2)\partial s^m(\mu)/\partial\mu\} - m^2/(1-\mu^2) \cdot s^m(\mu)] + q^m(\mu), \tag{5}$$

where the nonlinear terms are evaluated in coordinate space and projected as combined quantities in function space. If, in the nonlinear terms, each factor, U and f for example, is independently projected in function space, then each multiplication becomes a convolution over a large set of contributing terms. The simplicity of the term involving s in Eq. (4) is a result of the fact that the associated Legendre functions satisfy the equation $\{\nabla^2 + n(n+1)\}(P_n^m(\mu)e^{im\lambda}) = 0$. The simplicity of Eqs. (4) and (5) is the primary factor motivating this form of the coordinate to function space transformation.

To solve the above equations numerically it is necessary to work with finite representations, both through discretization of the coordinate representation and truncation of the function space representation. Because of the necessity of retaining full information in transforming between the two representations, there is a relationship between the number of spatial gridpoints and function space terms retained. A discussion of this problem, known as aliasing, is given in Hamming.[8] Let us assume a discretization of the spherical coordinate grid having I longitude gridpoints and J latitude gridpoints, and let us also assume a truncation of the function space, with $-M \le m \le M$ constraining the λ coefficients and $|m| \le n \le N(m)$ limiting the associated Legendre function series. For interfacing with the FFT routines, the longitude gridpoints $\lambda_i, i = 0,\ldots,I$ are chosen to be equally spaced, and I is taken to be a power of 2, $I = 2^s$, where s is an integer. Periodicity in λ requires that point 0 be equivalent to point I. To avoid the loss of information associated with aliasing it is also necessary to to choose $I \ge 3M+1$, so the number of Fourier amplitudes that can be meaningfully retained is limited by the spatial resolution. The amount of computational work in carrying out the longitude transformations using FFTs is of the order $I \cdot \ln(I)$ at each latitude μ_j, compared with order I^2 using direct methods. Thus the total work for all latitudes is of order $J \cdot I \cdot \ln(I)$ or $J \cdot I^2$ for the FFT and direct transformations, respectively. Similar considerations apply to the latitude grid and the associated Legendre functions, as we now describe.

The transformation between the latitude grid and the associated Legendre function representation is conveniently carried out using Gaussian integration. This requires the latitude

coordinates $\mu_j, j = 1,\ldots,J$ to be Gaussian points, namely, the J roots of the Legendre polynomial $P_J(\mu)$. To avoid aliasing, J must be chosen to be greater than some value determined by the details of the function space truncation scheme. For example, a triangular truncation scheme requires $J > (3N+1)/2$. Starting with a function $y^m(\mu) = \sum_{n=|m|}^{\infty} y_n^m P_n^m(\mu)$, the coefficients y_n^m can be projected using the identity

$$\int_{-1}^{1} P_n^m(\mu) P_{n'}^m(\mu) d\mu = \delta_{nn'} (n+m)!/\{(n+1/2)(n-m)!\} \tag{6}$$

to get

$$y_n^m = (n+1/2)(n-m)!/(n+m)! \int_{-1}^{1} y^m(\mu) P_n^m(\mu) d\mu. \tag{7}$$

The integral in Eq. (7) is evaluated using Gaussian integration to give

$$y_n^m = (n+1/2)(n-m)!/(n+m)! \sum_{j=1}^{J} y^m(\mu_j) P_n^m(\mu_j) w_j \ , \tag{8}$$

where the quantities w_j are the Gaussian integration weights $w_j = 2(1-\mu_j^2)/\{J \cdot P_{J-1}(\mu_j)\}^2$. Equation (8) can be regarded as a projection operation for y_n^m when $y^m(\mu)$ is defined on the coordinate grid of Gaussian points μ_j. To obtain all coefficients y_n^m in this way requires work of order $N \cdot M \cdot J$, as does the inverse transformation from y_n^m to $y^m(\mu)$. For high-resolution calculations, the work associated with this transformation can become dominant, and it is precisely the replacement of this transformation by a fast technique that is the subject of this paper.

Assuming a Gaussian point latitude coordinate grid, Eq. (8) can be used in Eqs. (4) and (5) to obtain

$$\begin{aligned} df_n^m / dt = & -\sum_{j=1}^{J} \{im(Uf)^m(\mu_j) P_n^m(\mu_j) - (Vf)^m(\mu_j) H_n^m(\mu_j)\} \\ & \times w_j (n+1/2)(n-m)!/\{a(1-\mu_j^2)(n+m)!\} \\ & + \nu / a^2 \cdot n(n+1) s_n^m + q_n^m, \end{aligned} \tag{9}$$

where $H_n^m(\mu) = (1-\mu^2)\partial P_n^m(\mu)/\partial\mu$. Equation (9) contains simple linear terms in s_n^m and q_n^m, and nonlinear terms involving $(Uf)^m$ and $(Vf)^m$ and either associated Legendre functions or linear differential operators acting on associated Legendre functions. Thus the general nonlinear term takes the form

$$T_n^m = \sum_{j=1}^{J} (gh)^m(\mu_j) \cdot D(P_n^m(\mu_j)) \cdot w_j \,/\, \{a(1-\mu_j^2)\} \quad , \tag{10}$$

where g and h are functions such as U, V, or f, and D is a linear differential operator acting on $P_n^m(\mu)$. As stated above, the work associated with this transformation is of order $N \cdot M \cdot J$, and it is this quantity that we seek to reduce here. To do this, let us carry out the inverse projection on Eq. (10) to obtain

$$\begin{aligned} T^m(\mu_j) &= \sum_{n=|m|}^{N(m)} T_n^m \cdot P_n^m(\mu_j) \\ &= \sum_{k=1}^{J} (gh)^m(\mu_k) \cdot w_k \,/\, \{a(1-\mu_k^2)\} \cdot \{\sum_{n=|m|}^{N(m)} P_n^m(\mu_j) \cdot D(P_n^m(\mu_k))\}. \end{aligned} \tag{11}$$

The next step in obtaining a fast method arises from the possibility of simplifying the summation involving the associated Legendre functions. For many simple linear differential operators, D, including those that appear in the dynamical equations of global climate models, simplifications are possible. As an example, consider the simplest of all operators, namely $D = I$, where I is the identity operator. In this case, the summation can be reduced, using the Christoffel-Darboux formula,[10,11]

$$\sum_{n=|m|}^{N} P_n^m(\mu_j) P_n^m(\mu_k) = \begin{cases} \varepsilon_{N+1}^m \{P_{N+1}^m(\mu_j) P_N^m(\mu_k) - P_N^m(\mu_j) P_{N+1}^m(\mu_k)\} \,/\, (\mu_j - \mu_k) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ,\mu_j \neq \mu_k \\ \varepsilon_{N+1}^m \{P'^m_{N+1}(\mu_j) P_N^m(\mu_k) - P'^m_N(\mu_j) P_{N+1}^m(\mu_k)\}, \mu_j = \mu_k, \end{cases} \tag{12}$$

where $\varepsilon_n^m = \sqrt{(n^2 - m^2) / (4n^2 - 1)}$. Substituting Eq. (12) into Eq. (11), with $D = I$, gives

$$\begin{aligned} T^m(\mu_j) = & \sum_{k=1, k\neq j}^{J} (gh)^m(\mu_k) \cdot w_k \,/\, \{a(1-\mu_k^2)\} \varepsilon_{N+1}^m \\ & \times \{P_{N+1}^m(\mu_j) P_N^m(\mu_k) - P_N^m(\mu_j) P_{N+1}^m(\mu_k)\} \,/\, (\mu_j - \mu_k) \\ & + (gh)^m(\mu_j) \cdot w_j \,/\, \{a(1-\mu_j^2)\} \varepsilon_{N+1}^m \{P'^m_{N+1}(\mu_j) P_N^m(\mu_j) - P'^m_N(\mu_j) P_{N+1}^m(\mu_j)\}, \end{aligned} \tag{13}$$

which can be written compactly in the form

$$\begin{aligned} T^m(\mu_j) = & A_N^m(\mu_j) \\ & + [B_N^m(\mu_j) \sum_{k=1, k\neq j}^{J} b_N^m(\mu_k) + C_N^m(\mu_j) \sum_{k=1, k\neq j}^{J} c_N^m(\mu_k)] \,/\, (\mu_j - \mu_k). \end{aligned} \tag{14}$$

In Eq. (14) the coefficients $A_N^m(\mu_j)$, $B_N^m(\mu_j)$, $b_N^m(\mu_j)$, $C_N^m(\mu_j)$, and $c_N^m(\mu_j)$ actually depend on $N+1$ as well as on N, but since $N+1$ can be viewed as a function of N, we can correctly write the coefficients as shown.

Equation (14) was derived from Eq. (10) under the assumption that the operator D is the identity. This derivation was originally given by Jakob.[11] For other commonly occurring differential operators, such as $D=(1-\mu^2)\partial/\partial\mu$ and $D=\nabla^2=1/a^2\{\partial/\partial\mu[(1-\mu^2)\partial/\partial\mu]-m^2/(1-\mu^2)\}$, expressions similar to Eq. (14) are easily derived. The primary difference between the equations for these latter operators and Eq. (14) involves the power of $\mu_j-\mu_k$ appearing in the denominator of the summation. For the operator $D=(1-\mu^2)\partial/\partial\mu$, the denominator involves $(\mu_j-\mu_k)^2$; for $D=\nabla^2$ we obtain $(\mu_j-\mu_k)^3$. These results will be detailed in an extension of the present work. Hence a dynamical equation such as Eq. (9) can be rewritten in the form

$$\begin{aligned}\partial f^m(\mu_j)\partial t = {} & A_0^m(\mu_j)+A_1^m(\mu_j)\sum_{k=1,k\neq j}^{J} a_1^m(\mu_k)/(\mu_j-\mu_k) \\ & +A_2^m(\mu_j)\sum_{k=1,k\neq j}^{J} a_2^m(\mu_k)/(\mu_j-\mu_k)^2+\ldots \\ & +A_p^m(\mu_j)\sum_{k=1,k\neq j}^{J} a_p^m(\mu_k)/(\mu_j-\mu_k)^p+\ldots,\end{aligned} \tag{15}$$

where the subscript p denotes the power of $\mu_j-\mu_k$ appearing in the denominator, the dependence on N is suppressed, and the series in p usually terminates after one to three terms. The quantity of work associated with solving the dynamical equations in the form of Eq. (15) is seen to be proportional $O(M\cdot J^2)$, which shows no advantage compared with the $O(N\cdot M\cdot J)$ required for the solution of Eq. (9). The purpose of casting the dynamical equations in the form of Eq. (15) arises from the possibility of reducing the computational work through the application of fast multipole techniques to the summation terms. The discussion in the next section will show that the use of these techniques reduces the scaling of work with the size of the problem to $O(M\cdot J)$, which does provide significant gains for large problems.

3. FAST MULTIPOLE TRANSFORMATION IN ONE DIMENSION

Fast multipole techniques were originally developed to provide fast methods to compute sums of pairwise force terms that appear in computer particle pushing.[9] Such sums are typically of the form

$$F_j = Q_j \sum_{k=1,k\neq j}^{J} q_k \cdot f(x_j - x_k) \quad , \tag{16}$$

where the total number of particles is J, the indices j and k denote particles, F_j is the total force on particle j due to all the other particles, Q_j is a charge or strength associated with the force on particle j, q_k is a charge or strength associated with the generation of force by particle k, and $f(x_j - x_k)$ is a function of the distance between particles. The naive evaluation of the summation in Eq. (16) for all the particles j requires computational work proportional to the number of particles squared. For common force or potential distance functions $f(x_j - x_k)$ it is possible to partition the coordinate domain into bins and to utilize Taylor series expansions about selected points within these bins to derive arbitrarily accurate evaluations of the forces that, in the limit of many particles, scale as the number of particles to the first power. The approach is derived in detail and an algorithm given for a two-dimensional (2-D) coordinate space by Greengard and Rokhlin.[9]

This applicability of Taylor expansion is the real motivation for writing the dynamical climate equations in the form of Eq. (15). We see that the terms in Eq. (15) are of the same form as the force term in Eq. (16), with the coefficients $A_p^m(\mu_j)$ and $a_p^m(\mu_k)$ playing the roles of the charges Q_j and q_k, and the coordinates μ_j and μ_k serving as x_j and x_k. The force functions $f(x_j - x_k)$ take the form $1/(\mu_j - \mu_k)^p$ and are conducive to the fast multipole approach. The coordinate domain associated with Eq. (15) is $-1 \leq \mu_j, \mu_k \leq 1$. It is not the purpose of this section to repeat the entire development of the fast multipole algorithm given by Greengard and Rokhlin.[9] Rather, we will adapt their procedure to the case at hand: namely, the evaluation of summations having the form of Eq. (16) in a one-dimensional (1-D) coordinate domain with $-1 \leq x_j, x_k \leq 1$ and force distance functions $f(x_j - x_k) = 1/(x_j - x_k)^p$. The purpose of this section is to describe a fast method to evaluate summations of the form

$$F_j = Q_j \sum_{k=1,k\neq j}^{J} q_k / (x_j - x_k)^p \quad , \tag{17}$$

where $1 \leq j,k \leq J$, $-1 \leq x_j, x_k \leq 1$, and p is a positive integer. By a fast method, we mean one in which the operation count in the evaluation of Eq. (17) for all J terms scales with large values of J as J, rather than J^2, as in the naive method. Although our application here is the evaluation of terms in global climate equations, we will use the particle, charge, and force terminology in the discussion of this section because of its intuitive clarity.

The convergence of the fast multipole method as applied to Eq. (17) will depend on the convergence properties of the Taylor series expansion of the function

$$1/(1-x)^p = \sum_{i=0}^{\infty} \binom{p-1+i}{i} x^i \quad , \tag{18}$$

which has a radius of convergence of $|x| < 1$. If the series in Eq. (18) is truncated to retain terms through x^n, then the remainder is given by $R_n^p = 1/(1-x)^p - S_n^p$, where

$$\begin{aligned} \left|R_n^p\right| &= \left|1/(1-x)^p - \sum_{i=0}^{n} \binom{p-1+i}{i} x^i\right| \\ &= \left|\sum_{i=n+1}^{\infty} \binom{p-1+i}{i} x^i\right|. \end{aligned} \tag{19}$$

A straightforward analysis of Eq. (19) shows that the remainder term can be rewritten as

$$\left|R_n^p\right| = \left|1/(1-x)^p \sum_{j=0}^{p-1} \binom{p+n}{j} (1-x)^j x^{p+n-j}\right| \quad . \tag{20}$$

Equation (20) provides an expression for the absolute error made in using the Taylor series expansion (Eq. (18)), truncated to $n+1$ terms, to evaluate $1/(1-x)^p$. Dividing both sides of Eq. (20) by $1/(1-x)^p$ shows that the relative error is just the summation. In the dynamical equations of global climate calculations, we typically have terms for which $1 \le p \le 3$. If we restrict $|x| \le r < 1$, then straightforward manipulation of Eq. (2) yields the upper error bounds shown in Table 1. Hence the Taylor series expansion shown in Eq. (18) converges to

Table 1. Error bounds on $n+1$ term Taylor series approximation of $1/(1-x)^p$

	Relative error	Absolute error
$p=1$	r^{n+1}	$r^{n+1}/(1-r)$
$p=2$	$(n+2)r^{n+1}$	$(n+2)r^{n+1}/(1-r)^2$
$p=3$	$(n+3)(n+2)r^{n+1}/2$	$(n+3)(n+2)r^{n+1}/[2(1-r)^2]$

$1/(1-x)^p$ for $|x| \le r < 1$, and Table 1 gives a practical upper bound for the truncation error shown in Eq. (20).

The fast multipole algorithm consists of forward and backward sweeps. The forward sweep begins by partitioning the coordinate domain into equal-sized bins into which the particles are gathered. Then the totality of the individual forces due to the particles in each bin is replaced by an equivalent "multipole force" expressed as a function of distance from the center of the bin. This situation is achieved through the use of a Taylor expansion for $1/(x-x_k)^p$ about the bin center, as will be discussed later. The convergence properties of the Taylor expansion then guarantee rapid convergence of the multipole series for points x well removed from the local bin

containing x_k. The forward sweep proceeds by collecting the multipole forces from each bin into successively larger bins, each time re-expressing the force as a multipole series about the larger bin center. This force is strongly convergent for points well outside the bin. The reverse sweep distributes these collected multipole forces back to successively smaller bins, this time by Taylor expansions of the multipole contributions of "remote" bins about the centers of the smaller bins. The inclusion of multipole forces only from remote (nonadjacent) bins guarantees rapid convergence of the series. Upon completion of the reverse sweep, the remaining forces due to individual particles in the same and adjacent bins are summed explicitly. To reduce computational work using the fast multipole algorithm, the number of nearby particles with force terms to be evaluated explicitly must be small compared with the total number of particles. Hence there must be many bins at the most local level. We will now fill in some of the details in this description of the fast multipole algorithm.

In the forward and backward sweeps described above, each step involves either the combination of two bins into one or the separation of one bin into two. For that reason we begin with 2^s bins, where s is an integer. Grouping the forces generated by the particles inside each bin to create a bin multipole force is carried out as follows: consider bin b with radius r_b and center x_b, so that particles having coordinates $x_b - r_b \le x_k < x_b + r_b$ lie inside bin b. We then obtain

$$\begin{aligned}\sum_{k\in b} q_k / (x - x_k)^p &= \sum_{k\in b} [q_k / (x - x_b)^p] \cdot \{1/[1-(x_k - x_b)/(x - x_b)]^p\} \\ &\cong \sum_{i=p}^{n+p} m_{i,b} / (x - x_b)^i ,\end{aligned} \tag{21}$$

where

$$m_{i,b} = \binom{i-1}{i-p} \sum_{k\in b} q_k (x_k - x_b)^{i-p} \quad . \tag{22}$$

This result comes from making a Taylor expansion of $1/[1-(x_k - x_b)/(x - x_b)]^p$, and truncating at $n+1$ terms. For particles with coordinates x_k inside the bin, $|x_k - x_b| < r_b$, and for evaluation points x outside the bin, $|x - x_b| > r_b$, the Taylor expansion converges. In the fast multipole algorithm presented here, the field point x is always taken to lie outside bin b and even outside the bins adjacent to bin b. Hence $|x - x_b| > 3 \cdot r_b$ is satisfied, $(x_k - x_b)/(x - x_b) < 1/3$. Thus in the error analysis in Eq. (20) and Table 1, the value $r = 1/3$ is approriate for this algorithm. The coefficients $m_{i,b}$ in the expansion of Eqs. (21) and (22) are the multipole moments of the particles in bin b; hence the force field generated by the multipole moments located at the bin center r_b is equivalent, up to the error in the truncated Taylor expansion, to the force field of the actual particles in the bin. Therefore, this algorithm is known as a fast multipole method.

The second part of the fast multipole procedure involves the forward sweep, in which the multipole fields associated with each bin are expressed as expansions about the centers of successively larger bins. This process is carried out in a pairwise fashion, with the fields from pairs of adjacent bins expressed relative to the center of the combined bin consisting of the two

bins together, and summed. Thus, each iteration in the forward sweep halves the number and doubles the size of the resulting bins. Assume that bin b and its adjacent bin b' combine to make bin B with radius $R_B = 2 \cdot r_b$ and center X_B. Then the center of bin B lies at the edge between bin b and its adjacent partner b', and the radius of bin B extends to cover bin b on one side bin b' on the other. The contribution of the particles in bin b to the Taylor series for the multipole expression about X_B proceeds as follows:

$$\begin{aligned}\sum_{k \in b} q_k / (x - x_k)^p &\cong \sum_{i=p}^{n+p} m_{i,b} / (x - x_b)^i \\ &= \sum_{i=p}^{n+p} [m_{i,b} / (x - X_B)^i] \cdot [1 - (x_b - X_B) / (x - x_B)]^i \\ &\cong \sum_{j=p}^{n+p} M_{j,B} / (x - X_B)^j ,\end{aligned} \tag{23}$$

where

$$M_{j,B} = \sum_{i=p}^{j} \binom{j-1}{j-i} m_{i,b} (x_b - X_B)^{j-i} \quad . \tag{24}$$

The quantities $M_{j,B}$ are the multipole moments for the field of the particles in bin b referenced to the center of bin B. The addition to Eq. (24) of a similar expression for the adjacent bin b then gives the multipole expansion about X_B of the force fields of all the particles in bin B. Again, the field point x is assumed to lie outside bin B and its adjacent bins. Therefore, in Eq. (23), $|x - X_B| > 3 \cdot R_B$ and $|x_b - X_B| = r_b = R_B / 2$, so that taking the ratio of these terms gives $r = 1/6$ in the convergence analysis of Eq. (20) and Table 1. The forward sweep is repeated a total of $s - 2$ iterations, so that finally only four bins cover the entire domain and the centers are at the 1/8, 3/8, 5/8, and 7/8 locations in x. About the center of each bin is the multipole expansion for all the particles in that quarter of the domain.

The reverse sweep is carried out to accumulate the contributions of all multipoles at sufficiently remote field coordinates that the radius of convergence requirements are satisfied. Consider a bin B and a nonadjacent bin B', both with radius R_B and with bin centers at X_B and $X_{B'}$, respectively. Let the field coordinate x be in bin B', so that $|x - X_{B'}| < R_B$ and $|X_B - X_{B'}| \geq 4 \cdot R_B$. Then it is possible to expand the multipole field due to the particles in bin B about the point $X_{B'}$ in bin B' as follows:

$$\sum_{k\in B} q_k / (x - x_k)^p \cong \sum_{j=p}^{n+p} M_{j,B} / (x - X_B)^j$$

$$= \sum_{j=p}^{n+p} [M_{j,B} / (X_{B'} - X_B)^j] \cdot 1 / [1 - (x - X_{B'}) / (X_B - X_{B'})]^j \qquad (25)$$

$$\cong \sum_{i=0}^{n} C_{i,B,B'} \cdot (x - X_{B'})^i \quad ,$$

where

$$C_{i,B,B'} = \sum_{j=p}^{n+p-i} (-1)^i \binom{i+j-1}{j-1} M_{j,B} / (X_{B'} - X_B)^{i+j} \ . \qquad (26)$$

Because contributions to bin B' come from nonadjacent bins B, the Taylor series expansion of $1/[1-(x-X_{B'})/(X_B - X_{B'})]^j$ in Eqs. (25) and (26) satisfies $r = 1/4$ in the error analysis of Eq. (20) and Table 1. In the reverse sweep, bins are evenly subdivided at each step. This doubles the number and halves the size of the bins in the partition. Then at each stage Eqs. (25) and (26) are used to gather the contributions to each bin from nonadjacent bins that have not previously contributed. For the contributions of larger bins already gathered in previous iterations, it is necessary to recenter the expansion of Eqs. (25) and (26) to the new bins b. Assume that bin b with center x_b and radius r_b is obtained in the subdivision of bin B. Then $|x_b - X_B| = r_b = R_B / 2$, and

$$\sum_{i=0}^{n} C_{i,B} \cdot (x - X_B)^i = \sum_{i=0}^{n} C_{i,B} \cdot [(x - x_b) + (x_b - X_B)]^i$$

$$= \sum_{j=0}^{n} c_{j,b} \cdot (x - x_b)^j \quad , \qquad (27)$$

where

$$c_{j,b} = \sum_{i=j}^{n} C_{i,B} \binom{i}{j} (x_b - X_B)^{i-j} \ . \qquad (28)$$

The backward sweep is applied for a total of $s-2$ iterations, so that at the end we have once again arrived at 2^s bins, and in each of these bins an expression of the form Eq. (27) gives the contribution of the fields of all particles, except those in the given and immediately adjacent bins, to $\sum_{k=1}^{J} q_k / (x - x_k)^p$. The final part of the fast multipole algorithm involves the completion of the summation by specific evaluation of Eq. (27) multiplied by Q_j at point x_j for each particle j, together with the direct evaluation of the remaining terms due to particles in the same or adjacent bins.

Before describing our findings using the fast multipole algorithm in actual calculations, let us review some properties of the algorithm. First, let us consider the number $n+1$ of terms to retain in the three truncated Taylor expansions used in the procedure. All three take the form of expansions of $1/(1-x)^i$ and are thus described by Eq. (20) and Table 1. The values of the convergence parameter r, in the notation of Table 1, were shown to be $r = 1/3$, 1/6, and 1/4 for the three expansions, respectively. Because the error estimates in Table 1 all vary as r^n, an approximate estimate of the number of terms required to obtain a relative error less than ε can be made by requiring $r^n < \varepsilon$, or equivalently $n > \log(\varepsilon)/\log(r)$. For $r = 1/3$, the largest value of r occurring in the algorithm, and $\varepsilon = 10^{-15}$ this inequality requires $n > 32$; while for $\varepsilon = 10^{-8}$, the required value is $n > 17$. Because of the exponential dependence of the error on n, these estimates provide approximate guidance to the number of terms required in practice. We will see in the following that the actual convergence of the fast multipole algorithm to direct summation in the evaluation of Eq. (16) is somewhat slower than this simple estimate. This is due to the fact that the present error estimate applies to the individual terms in the summation, and the error is compounded when these terms are summed.

Let us now consider the scaling of computational work involved in evaluating Eq. (16) for all j and k by the fast multipole algorithm. Clearly, in the direct evaluation of Eq. (16) the work involved scales as the number of pairs, namely $J(J+1)/2$ (or simply as J^2). In the fast multipole method, the relevent parameters are the number of particles, J, the number of bins at the finest level, 2^s (or simply the bin parameter, s), and the number of terms retained in the Taylor expansion, n. The first step involves the multipole expansion of all the particle forces about the centers of their local bins [Eqs. (21) and (22)] and requires a summation of approximately n terms for each of J particles, so the work scales as $J \cdot n$ in this step. The second step is the forward sweep, which involves the combination of bins into fewer, larger bins to gather the multipole expansions. Equations (23) and (24) show that this step takes about $n^2/2$ operations for each bin combined. The number of bins so combined is 2^s on the first iteration, 2^{s-1} on the second iteration, and so on until $2^3 = 8$ bins are merged into 4 bins on the last pass. Summing all of these bins gives approximately 2^{s+1} for the total number of bins operated upon. Hence the computational work in the forward sweep scales as $2^{s+1} \cdot n^2/2 = 2^s \cdot n^2$. In the third phase, or backward sweep, of the algorithm Eqs. (25) and (26) show that each contribution of one bin to another involves about $n^2/2$ operations. The number of contributions of one bin to another in level l, where $l = 2,\ldots,s$ during the backward sweep, is $3(2^l - 2)$. Summing this over all the levels and multiplying by $n^2/2$ yields computational work scaling as $3(2^{s+1} - 2s) \cdot n^2/2 = 3(2^s - s) \cdot n^2$ during the backward sweep. The final part of the fast multipole algorithm involves an evaluation of the expansion at the field points and the summation of near-neighbor terms not involved in the multipole expansions. The work associated with the evaluation of the multipole series scales as $J \cdot n$. Assuming the number of particles per bin is the average number of $J/2^s$, then the work in evaluating the near-neighbor terms explicitly scales as $3J \cdot J/2^s = 3J^2/2^s$, where the factor 3 arises from the inclusion of the particles in adjacent bins. Collecting the scalings from all the parts together into a single expression gives the overall scaling

$$Work \propto (3/2^s)J^2 + (2n)J + (2^{s+2} - 3s) \cdot n^2 \quad . \tag{29}$$

Upon first examination, Eq. (29) appears to give an operation count that scales as J^2, rather than J, in which case the complications of the fast multipole approach would yield no gain. However, Eq. (29) depends on the three parameters J, n, and s. Clearly, J is determined by the number of particles, and n is fixed by the desired accuracy of the multipole expansion. However, the number of bins, 2^s, is still free to choose. In order to minimize the amount of work in Eq. (29), we differentiate with respect to s and equate the result to zero. The result is $n^2[4-3/(2^s \cdot \ln(2)] - 3(J/2^s)^2 = 0$ and, assuming 2^s is significantly greater than 1, we obtain the result

$$2^s \cong \sqrt{3}J/(2n) \quad . \tag{30}$$

Substituting Eq. (30) back into Eq. (29) and ignoring $3s$ in comparison with 2^{s+2}, we find that the overall work in the fast multipole method scales as

$$Work \propto J(2n)(1+2\sqrt{3}) \quad . \tag{31}$$

Hence for any desired precision, determined by n, the work scales linearly with the number of particles in the fast multipole method (Eq. (31)), provided that the number of bins is determined by Eq. (30).

This section completes our description of the 1-D adaptation of the fast multipole method and its properties. The next section presents numerical studies that both demonstrate the properties of the fast multipole method and evaluate the tradeoff between its computational overhead and linear scaling in comparison with direct summation of Eq. (16).

4. RESULTS

In order to study and compare the evaluation of Eq. (16) for all particles $j = 1,\ldots,J$ by the fast multipole algorithm described in Sect. 3 and by direct summation, both of these methods were coded in Fortran on an IBM 580 RS-6000 work station. Using the notation of Sect. 3, a set of problems was defined by various choices of particle numbers J, coordinates x_j, strengths Q_j and q_k, and spatial dependence powers p. Specifically, values of $J = 30$, 100, 300, 1000, 3000, and 10000 were studied. The coordinates x_j were chosen to be random numbers lying between -1 and +1 in most of the calculations, but cases were also carried out using the Gaussian points μ_j as the coordinates. In all cases the strengths Q_j and q_k were assigned as random numbers, and powers of $p = 1$, 2, and 3 were used. Unless stated otherwise, the results shown will use random number generation of the coordinates x_j and $p = 1$ for the spatial dependence of the forces. Convergence and timing studies of the summation algorithms are nearly independent of these choices so long as the coordinates aren't extremely bunched in one region of the domain and p isn't extremely large.

The first question we consider is the convergence of the fast multipole algorithm with increasing number of terms n retained in the Taylor expansion. The results are plotted for $J = 100$, 1000, and 10000 particles in Fig. 1, which shows that the error decreases exponentially with n, consistent with Table 1, until leveling off at large values of n. The relative error here is defined to be that of the standard deviation of the results of Eq. (16) between fast mutltipole and direct summations, namely

$$(Error)^2 = \sum_{j=1}^{J} [(F_j^{fast} - F_j^{direct}) / F_j^{direct}]^2 / (J-1) \quad . \tag{32}$$

It should be noted that not all the error in Eq. (32) lies with the fast multipole method; the direct summation is subject to roundoff error, and contributes to the result. For all three values of J the curves are parallel in the exponential decreasing phase, and the slope of the curves is consistent with a value of $r = 0.45$ in Table 1. The figure shows that for 1000 particles the relative error reaches 10^{-8} at about $n = 22$ and 10^{-13} at $n = 35$. It is interesting that, for the results obtained here, the relative errors become independent of n for $n > 35$, and also that the relative errors become larger for increasing numbers of particles. The former effect is a function of the machine precision. In the calculations shown, double precision was used throughout. A similar survey using single precision resulted in the error being independent of n for $n > 22$, at which value the relative error was about 10^{-8} for 1000 particles. The increase in relative error with increasing number of particles is fairly uniform as a function of n. Studies with varying numbers of bins (parameter s) and with different selections of coordinates x_j and particle charges Q_j and q_k show little or no dependence upon these parameters. The relative error estimates in Table 1 pertain to individual terms in the summation of Eq. (16). As the number of particles increases, so does the number of terms and the accumulated errors, both for the direct and fast multipole summations. Although the same J behavior of convergence was observed in single precision, we note that the improvement in convergence in going from single to double precision (about a six order-of-magnitude reduction in relative error) was independent of J.

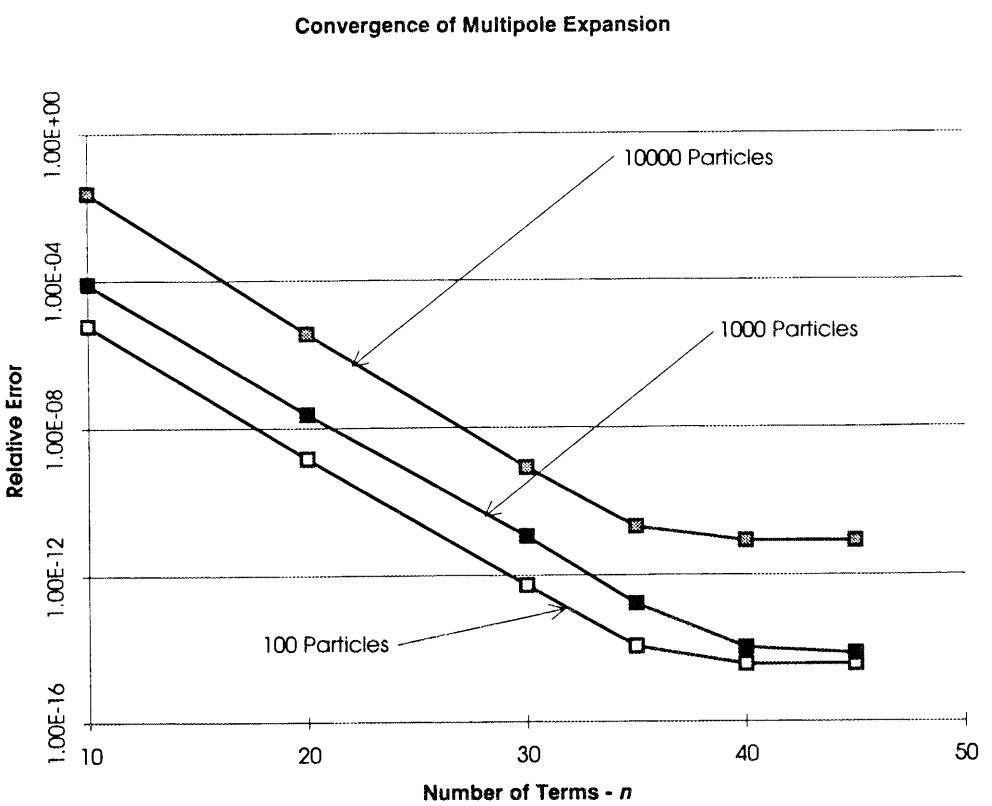

Fig. 1. Relative error in fast multipole expansion vs n for various numbers of particles.

The execution times of the fast multipole algorithm are shown as functions of the number of terms retained in the Taylor series expansion in Fig. 2. The several curves correspond to various choices of the bin parameter s. We note that each curve for the execution time corresponds to Eq. (29), with J and s fixed as n is varied. Consistent with Eq. (29), the curves show a parabolic behavior with increasing n. For any given values of J and n, Eq. (30) can be used to obtain the optimal value of s with respect to minimization of work. For $J = 10000$, Eq. (30) gives the following values of optimal s as a function of n: $(n,s) =$ (10, 9.8), (20, 8.8), (30, 8.2), (35, 8.0), (40, 7.8), (45, 7.6), and (50, 7.4). Since the fast multipole algorithm constrains s to be an integer, the results of Eq. (30) predict the observed set of optimal s: $(n,s) =$ (10, 10), (20, 9), (30, 8), (40, 8), (45, 8), and (50, 7). These results are completely consistent with the timings presented in Fig. 2. We note that the combination of the convergence requirements discussed above and the optimal timing indicate "best case" parameters for $J =$ 10000 when $(n,s) =$ (35-40, 8). For such cases the fast multipole algorithm completes the summation of Eq. (16) for all j in about 5 to 6 s of cpu time. Although the results presented here are for cases with $p = 1$ and randomly selected coordinates and charges, the errors and timings are similar for other selections of these parameters (including gaussian integration coordinates).

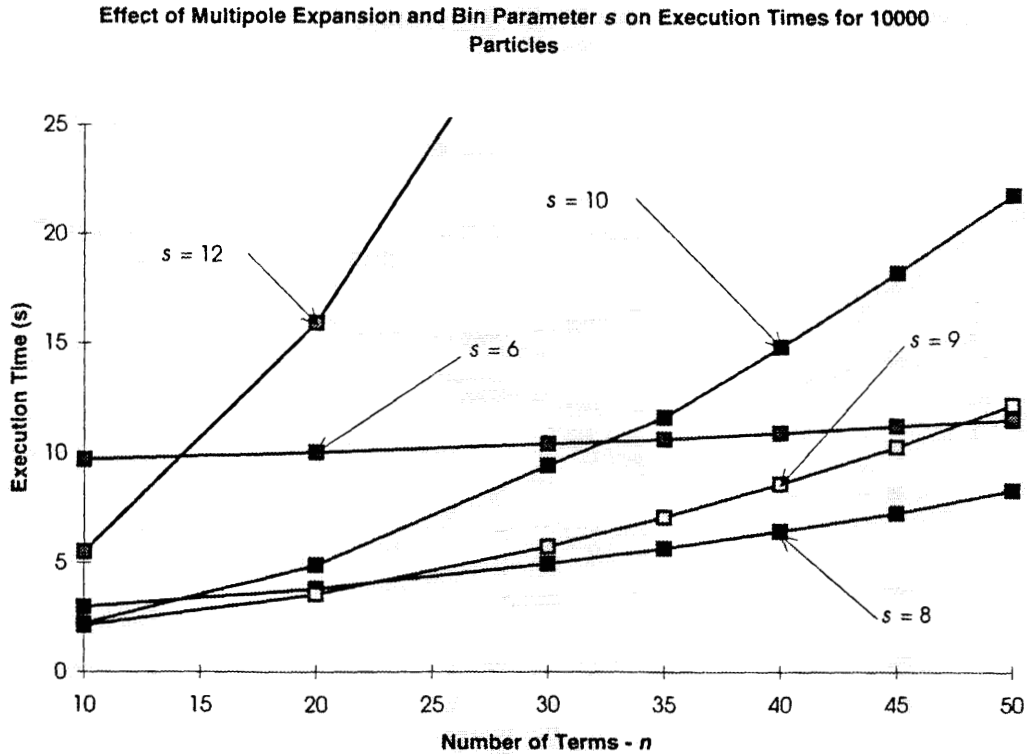

Fig. 2. Execution times of fast multipole algorithm for 10000 particles plotted vs n for various choices of bin parameter s.

Figure 3 plots the execution time for the fast multipole algorithm as a function of the bin parameter s for $n = 40$ and various numbers of particles J. According to Eq. (29), for large values of s the computational work becomes approximately proportional to $n^2 \cdot 2^s$ and independent of J, so that for fixed n the work doubles each time s is increased by 1. This scaling with s is obeyed quite accurately by the points having minimum execution time and $s \geq 5$ in Fig. 3. For these points, the increase in execution time with s is nearly linear on the logarithmic scale of Fig. 3, and the resulting scaling is $t \propto 2.08^s$. The other noteworthy feature that can be observed in Fig. 3 is the location of the minimum execution time as a function of s on each curve of fixed J. According to Eq. (30) with $n = 40$, the optimum values of s for given J are as follows: $(J,s) = (50, 0.1)$, (100, 1.1), (300, 2.7), (1000, 4.4), (3000, 6.0), and (10000, 7.8). Again restricting s to integer values, these results are: $(J,s) = (50, 0)$, (100, 1), (300, 3), (1000, 4), (3000, 6), and (10000, 8). These results are completely consistent with those shown in Fig. 3. For $J = 50$ and $J = 100$, the algorithmic constraint of $s \geq 2$ precludes observing the minima. In cases having such small numbers of particles the fast multipole algorithm is not competitive with direct summation, and the results of Eq. (30) that place s outside the sensible range serve to indicate this fact.

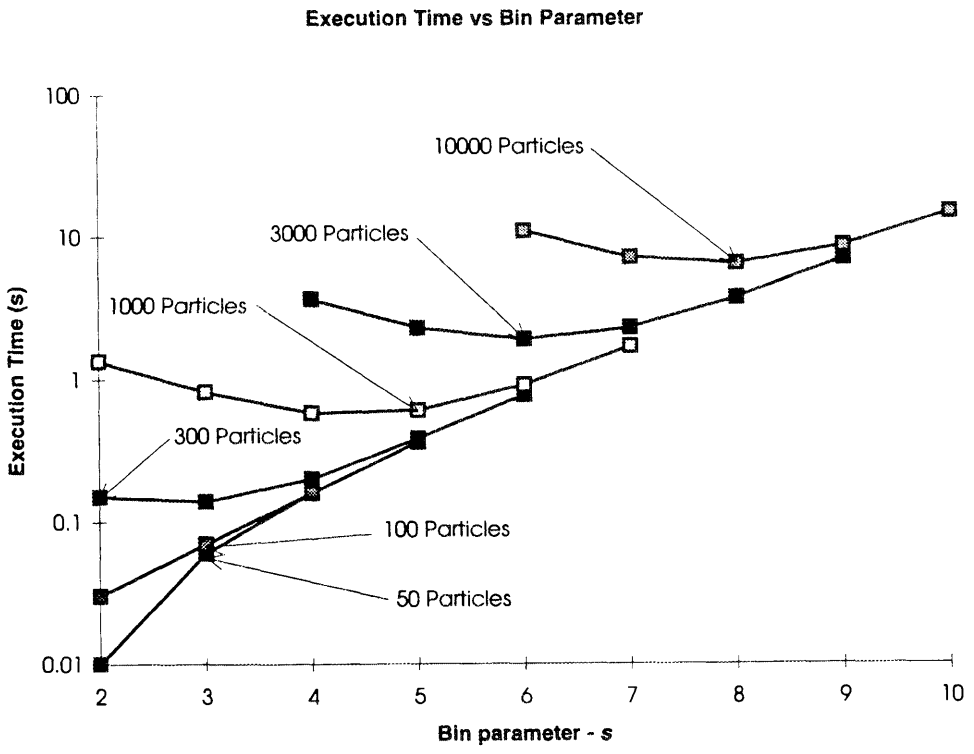

Fig. 3. Fast multipole algorithm execution time vs bin parameter s for various numbers of particles J.

The comparison between the prediction of Eq. (30) for the optimal bin parameter s, given J particles in the summation and n terms in the truncated Taylor expansion, and the observed optimal value is plotted in Fig. 4 for $J = 30$ - 10000 and $n = 20$ and 40. Given the constraints of the fast multipole algorithm that s is an integer and $s \geq 2$, the agreement between the observed and theoretical curves is exact: the observed values are as close as possible to the theoretical results.

The results to this point have been in the nature of tuning and studying the behavior of the fast multipole algorithm. The convergence of the truncated multipole expansion to the results of direct summation has been studied as a function of the number n of terms retained. Also, the systematics of execution time with the number of particles J, number of expansion terms n, and bin parameter s have been examined. In both these areas, the results have been demonstrated to agree with the predictions of Sect. 3. We conclude this study by comparing the timings of the fast multipole algorithm with those obtained by direct summation. The scaling and magnitude of these quantities with the number of particles J constitutes the central question

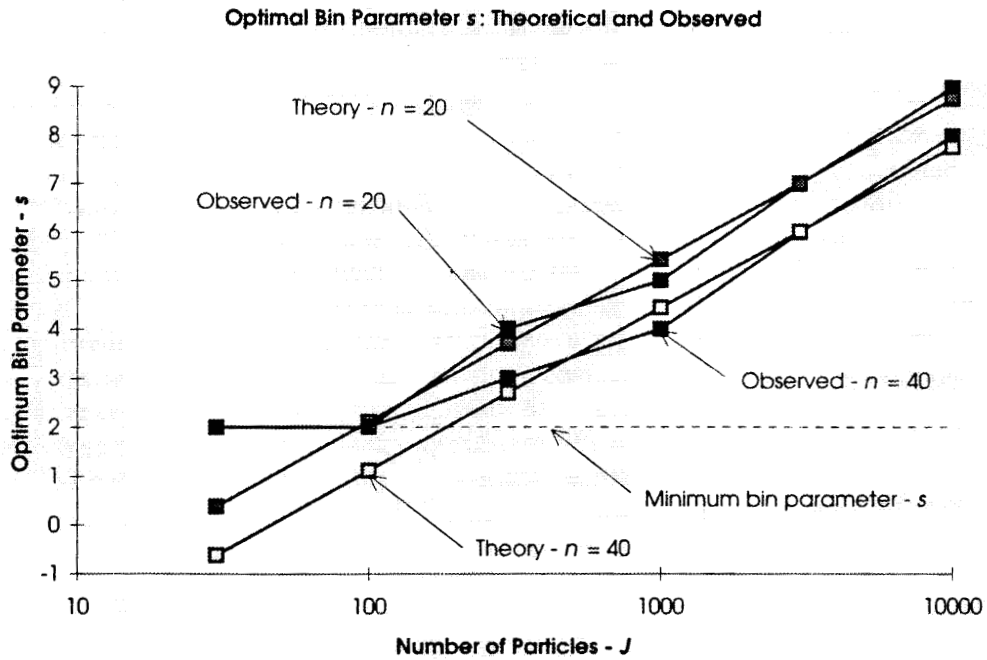

Fig. 4. Theoretical and observed optimal bin parameter s vs number of particles J for $n = 20$ and $n = 40$.

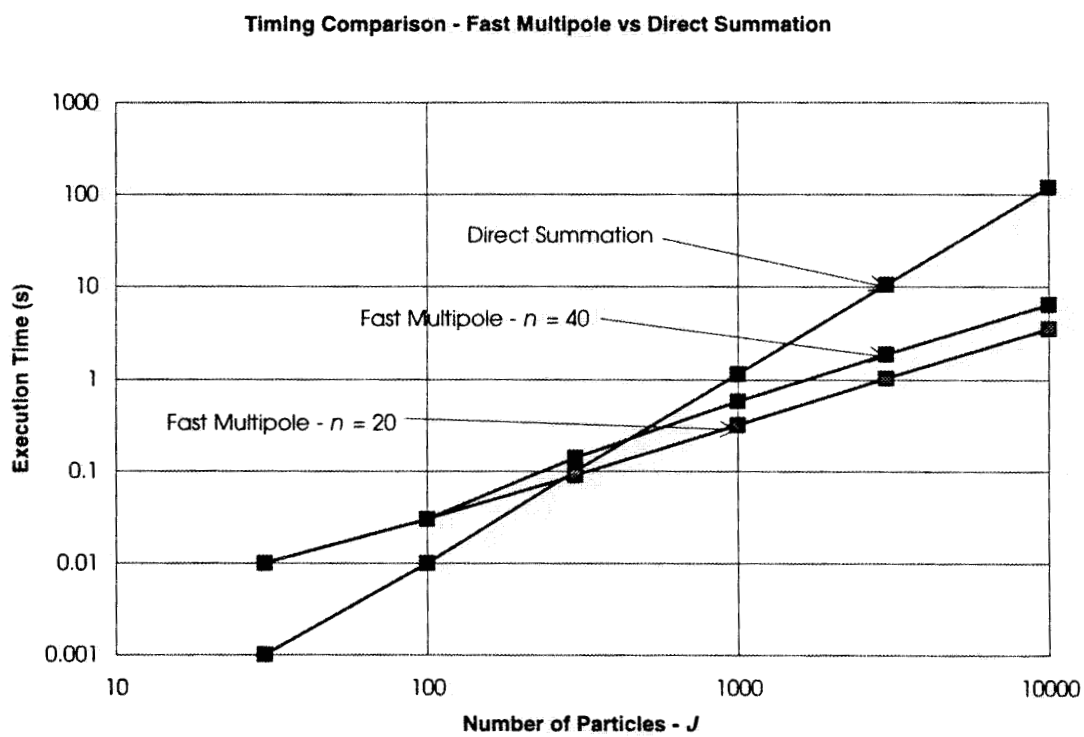

Fig. 5. Execution times for direct summation and fast multipole algorithms (with Taylor series truncation $n = 20$ and $n = 40$) vs number of particles J.

considered in this work. The answer is shown in Fig. 5, which plots the execution times for the fast multipole algorithm and for direct summation as a function of J from $J = 30$ - 10000. In the fast multipole calculations, cases were carried out with the Taylor series truncated to $n = 20$ and $n = 40$ terms, and optimal values of bin parameter s were used throughout. As predicted in Sect. 3, the execution time is proportional to J^2 for the direct summation method and to J for the fast multipole algorithm. For small values of J the direct summation of Eq. (16) is faster than its evaluation by fast multipoles because, for few enough particles, the overhead involved in connstructing and executing the steps of the fast multipole algorithm exceeds the work involved in simply carrying out the summation. However, for large numbers of particles the fast multipole algorithm is much faster than direct summation, and the speed advantage of the fast multipole approach increases with J. For example, when $J = 10000$ the fast multipole algorithm requires more than an order of magnitude less computer time than direct summation. Figure 5 indicates that the crossover point at which the fast multipole approach becomes faster than direct summation is in the vicinity of $J = 300$ to 500 particles. Hence for more than about 500 particles, or latitudes in the global climate models, it is advantageous to use the fast multipole algorithm to carry out summations of the form of Eq. (16).

5. CONCLUSIONS

This research demonstrates the applicability of a 1-D fast multipole transformation to the evaluation of summations occurring in dynamical global climate equations. These equations are written and evaluated in spherical coordinates, for which a natural set of basis functions is the spherical harmonics, or equivalently, the complex exponential functions with integral coefficients in the longitudinal angle multiplied by the associated Legendre functions in the angle of latitude. We discussed the desirability of evaluating linearized terms in function space and the nonlinear terms in coordinate space, and the transformation that this necessitates. This transformation can be separated into two parts: namely, the evaluation of fast Fourier transforms between the longitude angles and the complex exponentials, and the transformation between angles of latitude and the associated Legendre functions. The computational work associated with the latter transformation was shown to scale as $O(N \cdot M \cdot J)$, where M and N are the numbers of functions in longitude and latitude, respectively, and J is the number of latitudes in the coordinates space. Through the use of Christoffel-Darboux[10,11] identities and Gaussian integration techniques, the latitude to associated Legendre function transformation was recast in terms of summations in the form of Eq. (16). Direct evaluation of the resulting sums yields computational work scaling as $O(M \cdot J^2)$, which represents no gain over the original form.

We then considered the evaluation of summations in the form of Eq. (16) using an adaptation of a fast multipole algorithm derived by Greengard and Rokhlin[9] for particle simulations. The adaptation involves the replacement of the individual particles within preselected bins by equivalent multipole forces located at the bin centers. These forces are then accumulated into larger and larger bins, and then distributed out to successively smaller bins in forward and reverse sweeps that result in the summation of all terms involving nonlocal particles. The force terms due to local particles are then evaluated explicitly. We studied the numerical convergence of the fast multipole algorithm to the direct summation method as a function of the number of terms n retained in the multipole expansion; we also examined the scaling of computational work in terms of n, the bin parameter s, and number of particles J. From these studies we predicted, for given J, values for parameters n and s required for convergence and optimal timing. Numerical studies confirmed the predicted parametric behaviors for convergence and scaling of work, as well as the predicted optimal values.

The most important result of this research involved determining and comparing the scalings of computational work with number of particles J for both the direct and the converged optimized fast multipole summation methods. For each summation over all j in Eq. (16), the timing of the fast multipole transformation scales linearly with the number of latitude gridpoints, while timing for direct evaluation scales quadratically. In terms of the overall scaling of the latitude to associated Legendre function transformation, the use of the fast multipole algorithm yields computational work scaling as $O(M \cdot J)$, which for large values of J should represent a significant reduction of work. In spite of a larger computational overhead, the numerical calculations show that this scaling advantage renders the fast multipole method faster than direct evaluation for transformations involving greater than approximately 300 to 500 gridpoints. The advantage of the fast multipole algorithm is greater than a factor of 10 for $J = 10000$. Convergence of the fast multipole transformation is accurate to machine precision. As the resolution in global climate calculations continues to increase, an increasingly large fraction of the computational work will involve the transformation between spatial and spherical harmonic representations. The fast multipole transformation will offer significant reduction in computational time for these high-resolution cases.

6. REFERENCES

1. W. M. Washington and C. L. Parkinson, *An Introduction to Three-Dimensional Climate Modeling*, Oxford University Press, Oxford, 1986.

2. J. P. Peixoto and A. H. Oort, *Physics of Climate*, American Institute of Physics, New York, 1992.

3. J. J. Hack, B. A. Boville, B. T. Briegleb, J. T. Kiehl, P. J. Rasch, and D. L. Williamson, *Description of the NCAR Community Climate Model (CCM2)*, National Center for Atmospheric Research, NCAR/TN-382+STR, Boulder, Colo., 1993.

4. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1965.

5. P. M. Morse and H. Feshbach, *Mathematical Methods of Physics*, McGraw-Hill Book Company, New York, 1953.

6. H. R. Hicks, B. Carreras, J. A. Holmes, D. K. Lee, and B. V. Waddell, "3D Nonlinear Calculations of Resistive Tearing Modes," *J. Comput. Phys.* **44**, 46 (1981).

7. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. of Comp.* **19**, 297 (1965).

8. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, McGraw-Hill Book Company, New York, 1973.

9. L. Greengard and V. Rokhlin, "A Fast Algorithm for Particle Simulations," *J. Comput. Phys.* **73**, 325 (1987).

10. D. Jackson, *Fourier Series and Orthogonal Polynomials*, Mathematical Association of America, 1941.

11. R. Jakob, *Fast and Parallel Spectral Transform Algorithms for Global Shallow Water Models*, University of Colorado and National Center for Atmospheric Research, NCAR/CT-144, Boulder, Colo., 1993.

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. T. S. Darland
3. J. J. Dongarra
4. J. B. Drake
5. T. H. Dunigan
6. W. R. Emanuel
7. R. E. Flanery
8. R. H. Fowler
9. J. A. Holmes
10. M. A. Kuliasha
11. B.F. Lyon
12. S. A. Raby
13. R. F. Sincovec
14. D. W. Walker
15. P. H. Worley
16. Central Research Library
17. ORNL Patent Office
18. K-25 Applied Technology Library
19. Y-12 Technology Library
20. Laboratory Records - RC
21. Laboratory Records Department

EXTERNAL DISTRIBUTION

22. Christopher R. Anderson, Department of Mathematics, University of California Los Angeles, CA 90024

23. David C. Bader, Atmospheric and Climate Research Division, Office of Health and Environmental Research, Office of Energy Research, ER-76, U.S. Department of Energy, Washington, DC 20585

24. David H. Bailey, NASA Ames, Mail Stop 258-5, NASA Ames Research Center Moffet Field, CA 94035

25. Edward H. Barsis, Computer Science and Mathematics, P.O. Box 5800 Sandia National Laboratories, Albuquerque, NM 87185

26. Colin Bennett, Department of Mathematics, University of South Carolina, Columbia, SC 29208

27. Dominique Bennett, CERFACS, 42 Avenue Gustave Coriolis, 31057 Toulouse Cedex, FRANCE

28. Marsha J. Berger, Courant Institute of Mathematical Sciences, 251 Mercer Street New York, NY 10012

29. Mike Berry, Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301

30. Ake Bjorck, Department of Mathematics, Linkoping University, S-581 83, Linkoping, SWEDEN

31. John H. Bolstad, Lawrence Livermore National Laboratory, L-16, P.O. Box 808, Livermore, CA 94550

32. George Bourianoff, Superconducting Super Collider Laboratory, 2550 Beckleymeade Avenue, Suite 260, Dallas, TX 75237-3946

33. Roger W. Brockett, Wang Professor of EE and CS, Division of Applied Sciences, Harvard University, Cambridge, MA 02138

34. Bill L. Buzbee, Scientific Computing Division, National Center for Atmospheric, Research, P.O. Box 3000, Boulder, CA 80307

35. Captain Edward A. Carmona, Parallel Computing Research Group, U.S. Air Force Weapons Laboratory, Kirtland AFB, NM 87117

36. Peter Campbell, Environmental Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

37. John Cavallini, Acting Director, Scientific Computing Staff, Applied Mathematical Sciences, Office of Energy Research, U.S. Department of Energy, Washington, DC 20585

38. Professor I-Liang Chern, Department of Mathematics, National Taiwan University Taipei, Taiwan, R.O.C,

39. Ray Cline, Sandia National Laboratories, Livermore, CA 94550

40. Alexandra Chorin, Mathematics Department, Lawrence Berkeley Laboratory, Berkeley, CA 94720

41. James Corones, Ames Laboratory, Iowa State University, Ames, IA 50011

42. Jean Cot'e, RPN, 2121 Transcanada Highway, Suite 508, Dorval, Quebec H9P 1J3 CANADA

43. William Dannevik, Lawrence Livermore National Laboratory, P.O. Box 808, L-16, Livermore, CA 94550

44. John J. Dorning, Department of Nuclear Engineering Physics, Thornton Hall, McCormick Road, University of Virginia, Charlottesville, VA 22901

45. Iain S. Duff, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 0QX, England

46. John Dukowicz, Los Alamos National Laboratory, Group T-3, Los Alamos, NM 87545

47. Richard E. Ewing, Department of Mathematics, University of Wyoming, Laramie, WY 82071

48. Ian Foster, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

49. Geoffrey C. Fox, NPAC, 111 College Place, Syracuse University, Syracuse, NY 13244-4100

50. Dr. Rhys Francis, Division of Information Technology, CSIRO, 723 Swanston Street, Carlton, Vic. 3053, AUSTRALIA

51. Paul O. Frederickson, ACL, MS B287, Los Alamos National Laboratory, Los Alamos, NM 87545

52. J. Alan George, Vice President, Academic and Provost, Needles Hall, University of Waterloo, Waterloo, Ontario, CANADA N2L 3G1

53. James Glimm, Department of Mathematics, State University of New York, Stony Brook, NY 11794

54. Gene Golub, Computer Science Department, Stanford University, Stanford, CA 94305

55. John Gustafson, 236 Wilhelm, Ames Laboratory, Iowa State University, Ames, IA 50011

56. Phil Gresho, Lawrence Livermore National Laboratory, L-262, P.O. Box 808 Livermore, CA 94550

57. William D. Gropp, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

58. Eric Grosse, AT&T Bell Labs 2T-504, Murray Hill, NJ 07974

59. James J. Hack, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307

60. Michael T. Heath, Center for Supercomputing Research and Development, 305 Talbot Laboratory, University of Illinois, 104 South Wright Street, Urbana, IL 61801-2932

61. Michael Henderson, Los Alamos National Laboratory, Group T-3, Los Alamos, NM 87545

62. Lennart Johnsson, Thinking Machines Inc., 245 First Street, Cambridge, MA 02142-1214

63. J. R. Jump, ECE Dept., Rice University, P. O. Box 1892, Houston, TX 77251

64. Malvyn Kalos, Cornell Theory Center, Engineering and Theory Center Bldg., Cornell University, Ithaca, NY 14853-3901

65. Hans Kaper, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

66. Alan H. Karp, IBM Scientific Center, 1530 Page Mill Road, Palo Alto, CA 94304

67. Kenneth Kennedy, Department of Computer Science, Rice University, P.O. Box 1892, Houston, TX 77001

68. Tom Kitchens, ER-7, Applied Mathematical Sciences, Scientific Computing Staff Office of Energy Research, Office G-437, Germantown, Washington, DC 20585

69. Peter D. Lax, Courant Institute of Mathematical Sciences, New York University 251 Mercer Street, New York, NY 10012

70. James E. Leiss, Rt. 2, Box 142C, Broadway, VA 22815

71. Rich Loft, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307

72. Michael C. MacCracken, Lawrence Livermore National Laboratory, L-262, P.O. Box 808, Livermore, CA 94550

73. Norman D. Malmuth, Science Center, Rockwell International Corporation, 1049 Camino Dos Rios, P.O. Box 1085, Thousand Oaks, CA 91358

74. Robert Malone, C-DO/ACL, MS B287, Los Alamos National Laboratory, Los Alamos, NM 87545

75. Len Margolin, Los Alamos National Laboratory, Los Alamos, NM 87545

76. Hal Marshall, Laboratory for Scientific Computation, Rm 271, Cooley Bldg., University of Michigan, Ann Arbor, MI 48109-2104

77. Frank McCabe, Department of Computing, Imperial College of Science and Technology, 180 Queens Gate, London SW7 2BZ, ENGLAND

78. James McGraw, Lawrence Livermore National Laboratory, L-306, P.O. Box 808, Livermore, CA 94550

79. L. David Meeker, Mathematics Department, University of New Hampshire, Kingsbury Hall, Durham, NH 03824

80. Paul C. Messina, Mail Code 158-79, California Institute of Technology, 1201 E. California Blvd., Pasadena, CA 91125

81. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, IL 61801

82. V. E. Oberacker, Department of Physics, Vanderbilt University, Box 1807, Station B, Nashville, TN 37235

83. Joseph Oliger, Computer Science Department, Stanford University, Stanford, CA 94305

84. Robert O'Malley, Department of Mathematical Sciences, Rensselaer Polytechnic, Institute, Troy, NY 12180-3590

85. James M. Ortega, Department of Applied Mathematics, Thornton Hall, University of Virginia, Charlottesville, VA 22901

86. Ron Peierls, Applied Mathematical Department, Brookhaven National Laboratory, Upton, NY 11973

87. Richard Pelz, Dept. Of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08855-0909

88. Paul Pierce, Intel Scientific Computers, 15201 N.W. Greenbrier Parkway, Beaverton, OR 97006

89. Robert J. Plemmons, Dept. Of Mathematics and Computer Science, North Carolina State University, Raleigh, NC 27650

90. Jesse Poore, Computer Science Department, University of Tennessee, Knoxville, TN 37996-1300

91. Andrew Priestley, Institute for Computational Fluid Dynamics, Reading University Reading RG6 2AX, ENGLAND

92. Lee Riedinger, Director, The Science Alliance Program, University of Tennessee Knoxville, TN 37996

93. Garry Rodrigue, Numerical Mathematics Group, Lawrence Livermore National Laboratory, Livermore, CA 94550

94. Ahmed Sameh, Department of Computer Science, 200 Union Street, S.E., University of Minnesota, Minneapolis, MN 55455

95. Dave Schneider, University of Illinois at Urbana-Champaign, Center for Supercomputing Research and Development, 319 E. Talbot - 104 S. Wright Street Urbana, IL 61801

96. David S.Scott, Intel Scientific Computers, 15201 N.W.Greenbrier Parkway, Beaverton, OR 97006

97. William C. Skamarock, 3973 Escuela Court, Boulder, CO 80301

98. Richard Smith, Los Alamos National Laboratory, Group T-3, Mail Stop B2316 Los Alamos, NM 87545

99. Peter Smolarkiewicz, National Center for Atmospheric Research, MMM Group P.O. Box 3000, Boulder, CO 80307

100. Jurgen Steppeler, DWD, Frankfurterstr 135, 6050 Offenbach, WEST GERMANY

101. Rick Stevens, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

102. Paul N. Swarztrauber, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307

103. Wei Pai Tang, Department of Computer Science, University of Waterloo, Waterloo Ontario, CANADA N2L 3G1

104. Harold Trease, Los Alamos National Laboratory, Mail Stop B257, Los Alamos, NM 87545

105. Robert G.Voigt, ICASE, MS 132-C, NASA Langley Research Center, Hampton, VA 23665

106. Mary F. Wheeler, Rice University, Department of Mathematical Sciences, P.O. Box 1892, Houston, TX 77251

107. Andrew B. White, Los Alamos National Laboratory, P.O. Box 1663, MS-265, Los Alamos, NM 87545

108. David L. Williamson, National Center for Atmospheric Research, P.O. Box 3000 Boulder, CO 80307

109. Samuel Yee, Air Force Geophysics Lab, Department LYP, Hancom AFB, Bedford, MA 01731

110. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge Operations Office, P.O. Box 2001, Oak Ridge, TN 37831-8600

111-112. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831