

RECEIVED  
NOV 17 1998  
OSTI

## Fast Multipole Solvers for Three-Dimensional Radiation and Fluid Flow Problems<sup>†</sup>

J. H. Strickland (jhstric@sandia.gov), L. A. Gritzo (lagritz@sandia.gov),  
R. S. Baty (rsbaty@sandia.gov), G. F. Homicz (gfhomic@sandia.gov),  
and S. P. Burns (spburns@sandia.gov)

*Engineering Sciences Center  
Sandia National Laboratories  
Albuquerque, NM 87185-0836, USA*

### Abstract

A number of physics problems can be modeled by a set of  $N$  elements, which have pair-wise interactions with one another. The use of such elements for the evolution of vorticity in fluid flows and the calculation of the velocity field from the evolving vorticity field is well known. Fast multipole methods for fluid flow problems have been developed in the past to reduce computational effort to something less than  $O(N^2)$ . In this paper we develop a fast multipole solver with application to both 3-D radiation problems (calculation of the heat flux from the evolving temperature field in an absorbing medium) and 3-D fluid flow. This is accomplished by using a more general kernel for the associated volume integrals. This kernel also encompasses other applications such as gravitational fields, electrostatics, scattering, etc. The present algorithm has been designed to have a very high "parallel efficiency" when used on massively parallel computers. This feature comes at the expense of computational effort, which is less than  $O(N^2)$  but greater than  $O(N)$  or  $O(N \ln N)$ .

## 1 INTRODUCTION

Motivation for the present work results from a desire to perform fast yet accurate calculation of the three-dimensional radiant heat flux field in an absorbing medium from the temperature field as well as calculation of the velocity field from the vorticity field. For many problems, the proper discretization of the source field (vorticity or temperature) and target field (velocity or heat flux) will require the use of more than  $N_s = 10^6$  source particles and  $N_t = 10^6$  target points. In order to perform efficient calculations in such cases, the combined use of fast multipole expansions and massively parallel computing is required. We therefore seek a solution methodology that will provide good "parallel efficiency," while at the same time reducing the CPU time from order  $O(N_s N_t)$  to something approaching  $O(N_s N_t)^{1/2}$ .

Overviews of various fast multipole schemes and their applicability to a large number of physics problems are given by Greengard [1] and by Strickland and Baty [2]. In these problems, the following types of sums over a set of source elements typically occur:

$$f(\mathbf{r}) = \sum_{i=1}^{N_s} \psi(\mathbf{r}, \mathbf{r}'_i) \approx \sum_{k=1}^p h_k(\mathbf{r}) \sum_{i=1}^{N_s} g_{ki}(\mathbf{r}'_i). \quad (1)$$

Such equations usually result from discretization of an integral over a volume. The computational complexity of computing  $N_t$  values of  $f(\mathbf{r})$  in a direct manner is  $O(N_s N_t)$ . For the  $p^{\text{th}}$  order approximation indicated in Equation (1), in which the source and target variables are separated, the complexity is reduced to  $O[p(N_s + N_t)]$  as a result of presumming over all of the source information.

<sup>†</sup> This work was performed at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U. S. Department of Energy under Contract DE-AC04-94AL85000

## 2 VORTEX AND RADIATION INTEGRAL EQUATIONS

### 2.1 Velocity/Vorticity Fields

According to the Helmholtz decomposition for a vector field [3], the velocity vector can be obtained from the vorticity vector and the divergence of the velocity field in an infinite domain according to:

$$\mathbf{u}(\mathbf{r}) = \nabla_{\mathbf{r}} \times \int_V \omega(\mathbf{r}') K(\mathbf{r} - \mathbf{r}') dV(\mathbf{r}') - \nabla_{\mathbf{r}} \int_V D(\mathbf{r}') K(\mathbf{r} - \mathbf{r}') dV(\mathbf{r}'), \quad (2)$$

where  $K(\mathbf{r} - \mathbf{r}') = 1/(4\pi |\mathbf{r} - \mathbf{r}'|)$  for 3-D fields, and  $D(\mathbf{r}') = \nabla_{\mathbf{r}'} \cdot \mathbf{u}(\mathbf{r}')$ . This may also be written as:

$$\mathbf{u}(\mathbf{r}) = \frac{1}{4\pi} \int_V \frac{\omega(\mathbf{r}') \times (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} dV(\mathbf{r}') + \frac{1}{4\pi} \int_V \frac{D(\mathbf{r}')(\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} dV(\mathbf{r}'). \quad (3)$$

Another useful quantity for 3-D flow field computations is the vector potential, which is in fact, the first integral quantity in Equation (2):

$$\Phi(\mathbf{r}) = \frac{1}{4\pi} \int_V \frac{\omega(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} dV(\mathbf{r}'). \quad (4)$$

### 2.2 Heat Flux/Temperature Fields

The heat flux field can be obtained from the infinite domain temperature field in an absorbing medium using the following integral relationship [4]:

$$\mathbf{q}(\mathbf{r}) = \frac{\sigma}{\pi} \int_V \alpha(\mathbf{r}') T^4(\mathbf{r}') \frac{e^{-\bar{\alpha}|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|^3} (\mathbf{r} - \mathbf{r}') dV(\mathbf{r}'). \quad (5)$$

Here  $\sigma$  is the Boltzmann constant,  $\alpha$  is the absorptivity coefficient at the source, and  $\bar{\alpha}$  is the average absorptivity along the line of sight between the source element and target point.

The divergence of the radiant heat flux is the radiative source term in the energy equation. The contribution from radiation sources in the field to the divergence of the heat flux is given by the expression:

$$\nabla \cdot \mathbf{q}(\mathbf{r}) = \alpha(\mathbf{r}) [4\sigma T^4(\mathbf{r}) - G(\mathbf{r})], \quad (6)$$

where  $G(\mathbf{r})$  is the irradiance. The irradiance may, in turn, be expressed as:

$$G(\mathbf{r}) = \frac{\sigma}{\pi} \int_V \alpha(\mathbf{r}') T^4(\mathbf{r}') \frac{e^{-\bar{\alpha}|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|^2} dV(\mathbf{r}'). \quad (7)$$

### 2.3 General Integral Forms

By inspection it can be seen that the various volume integrals given in Equations (2) through (7) can be expressed in the following general form:

$$f(\mathbf{r}) = g(\mathbf{r}) \circ \int h(\mathbf{r}') \frac{e^{-\bar{\alpha}|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|^n} dV(\mathbf{r}'), \quad (8)$$

where  $g(\mathbf{r}) \circ h(\mathbf{r}')$  represents a scalar or vector multiplication. The discrete form of Equation (8) is:

$$f(\mathbf{r}) = g(\mathbf{r}) \circ \sum_{i=1}^{N_s} h(\mathbf{r}'_i) \delta V(\mathbf{r}'_i) \frac{e^{-\bar{\alpha}|\mathbf{r}-\mathbf{r}'_i|}}{|\mathbf{r} - \mathbf{r}'_i|^n} = g(\mathbf{r}) \circ \sum_{i=1}^{N_s} S_i(\mathbf{r}'_i) \frac{e^{-\bar{\alpha}|\mathbf{r}-\mathbf{r}'_i|}}{|\mathbf{r} - \mathbf{r}'_i|^n}, \quad (9)$$

where  $S_i(\mathbf{r}'_i)$  is a source strength and  $N_s$  is the number of sources in a particular source domain. It is also worth noting that a number of other 3-D physics problems may be cast into the same general form with various values of  $\alpha$  and  $n$  - i.e., gravitational problems [5], electrostatics, scattering problems [6], etc.

## **DISCLAIMER**

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, make 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.

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

### 3 METHOD OF SOLUTION

In this section we will present an overview of the domain decomposition scheme and the development of multipole expansions for the general kernel in Equation (9). In Section 4, additional details will be given concerning the relationship between the domain decomposition and the parallelization scheme.

#### 3.1 Domain Decomposition

Multipole expansions are used to calculate the influence of a group of sources in a domain on a target point located at a sufficient distance (relative to the source domain size) from the multipole expansion center. A review of several spatial partitioning methods is given in Reference [2]. These methods have been loosely categorized into Barnes-Hut [7] and Greengard-Rokhlin [8]-[9] domain decomposition schemes. The distinguishing difference between the two methods is that the Barnes-Hut scheme utilizes only multipole expansions about source domain centers, whereas the Greengard-Rokhlin scheme utilizes both multipole and local expansions. The local expansions are essentially Taylor series expansions about the center of each target domain. The Greengard-Rokhlin method of domain decomposition tends to be more difficult to implement but yields computational time scalings that range from  $O(N)$  to  $O(N)^{1.3}$ , depending upon the specific implementation and the arrangement of source and target points. Computational complexities for the Barnes-Hut schemes range from  $O(N \ln N)$  to  $O(N)^{1.5}$ .

We recently parallelized an axisymmetric fast solver for vortex rings that was developed by Strickland and Amos [10]. The domain decomposition was based on that of Reference [9]. Anecdotal evidence from that effort suggests that fast solvers based on the adaptive Greengard-Rokhlin scheme do not parallelize well. For the present work, we chose to use a Barnes-Hut domain decomposition scheme that was developed by Clarke and Tutty [11] for their two-dimensional vortex model; to our knowledge, this represents its first application to a three-dimensional problem. As will be shown subsequently, this domain decomposition scheme yields a very high parallel efficiency.

Clarke and Tutty's [11] domain decomposition method is based on a binary tree structure of source boxes. As shown in Figure 1, the boundaries of the source domain are first defined at the coarsest level by the range of source positions in the x, y, and z directions. This box is then divided into two sub-domains along its longest dimension such that an equal number of source points lie within each sub-domain. Each sub-domain is then shrunk so as to exactly contain its designated sources. This process continues until the number of sources at the finest level is below some specified value. Multipole coefficients obtained at the finest level are shifted to their parent's center and added together. This continues until all boxes at all levels possess a set of multipole coefficients. Advantages of this scheme include the facts that each box at a given level contains essentially the same number of source points (the number of source points in boxes at a given level may differ by one) and the shrunk domains allow one to use the multipole expansions more often.

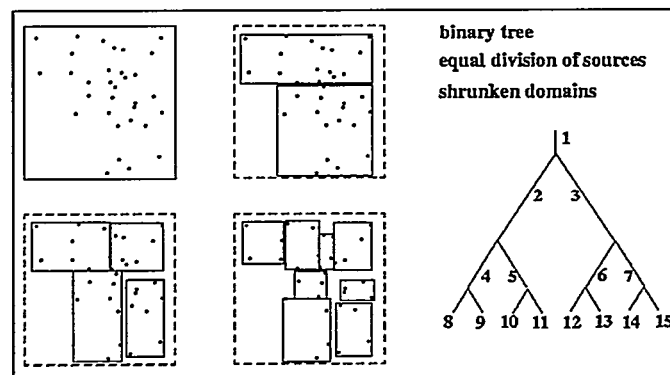


Figure 1. Spatial Partitioning of Vortex or Radiation Sources

### 3.2 Multipole Expansions

Greengard and Rokhlin [8] have developed three-dimensional expansions for the vorticity kernel but expansions for the three-dimensional radiation kernel were not available prior to the present work. This work follows the recent development by Gritzo and Strickland [12] of one-dimensional expansions of the radiation kernel along with an associated fast solver that is also based on the Clarke-Tutty binary domain decomposition.

#### 3.2.1 Multipole expansions about the source domain center

In order to develop a multipole expansion for Equation (9), we only need to be concerned with expansion of the kernel:

$$\psi(|\mathbf{R}|) = \frac{e^{-\alpha|\mathbf{R}|}}{|\mathbf{R}|^n}, \quad (10)$$

where  $|\mathbf{R}| = R = |\mathbf{r} - \mathbf{r}'|$ , as indicated in Figure 2.

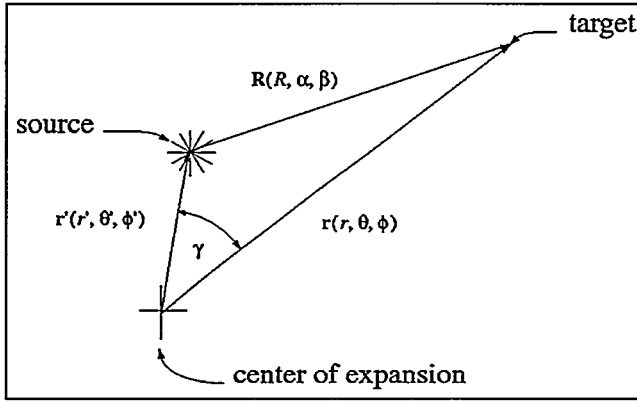


Figure 2. Multipole Expansion Geometry

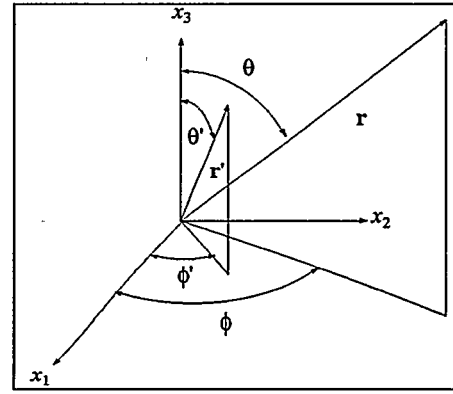


Figure 3. Spherical Coordinate System

We begin the task of constructing this expansion by noting that:

$$|\mathbf{r} - \mathbf{r}'| = \sqrt{r^2 - 2rr' \cos \gamma + r'^2}, \quad (11)$$

which allows the kernel to be represented in terms of the scalar values  $r$  and  $r'$ . Limiting the value of  $r'/r$  to always be less than unity, the functions of Equation (10) can be expanded in terms of a Taylor series in  $r'/r$  of the form:

$$\psi(|\mathbf{R}|) = \frac{e^{-\alpha r}}{r^n} \sum_{m=0}^p a_m(\alpha r, n, z) \left( \frac{r'}{r} \right)^m + \varepsilon, \quad (12)$$

where  $z = \cos \gamma$ . The coefficients  $a_m$  consist of polynomials in  $z$ . The coefficients of these polynomials are, in turn, functions of  $n$  and  $\alpha r$ . The maximum error  $\varepsilon$  may be found by use of Lagrange's form of the remainder which is simply the  $p+1$  term in the series with  $r$  being replaced with a value that is someplace between  $r'$  and  $r$ .

In order to make the expansion useful, the angles associated with  $\mathbf{r}'$  and  $\mathbf{r}$  must be uncoupled. In other words, the  $z$  quantities in Equation (12) must be expressed in terms of the angles associated with  $\mathbf{r}'$  and  $\mathbf{r}$ , as shown in Figure 3. To accomplish this, Equation (12) is first expressed in terms of Legendre Polynomials:

$$\psi(|\mathbf{R}|) = \sum_{m=0}^p b_m P_m(z). \quad (13)$$

This expansion is achieved by using the orthogonality condition for Legendre Polynomials:

$$\int_{-1}^1 P_n(z) P_m(z) dz = \frac{2}{2m+1} \delta_{nm}. \quad (14)$$

Application of Equation (14) to Equation (13) leads to:

$$b_m = \frac{2m+1}{2} \int_{-1}^1 P_m(z) \psi(|\mathbf{R}|) dz. \quad (15)$$

Next, the Legendre Polynomials are replaced by using the addition theorem for spherical harmonics [13]:

$$P_m(z) = \frac{4\pi}{2m+1} \sum_{n=-m}^m Y_{m,n}^*(\theta', \phi') Y_{m,n}(\theta, \phi). \quad (16)$$

After performing the indicated manipulations, the details of which will not be presented here, the multipole expansion for this kernel may be written as:

$$\frac{e^{-\alpha|\mathbf{R}|}}{|\mathbf{R}|^n} = \frac{e^{-\alpha r}}{r^n} \sum_{i=0}^p \sum_{m=0}^i \sum_{l=0}^i \sum_{k=-m}^m \frac{4\pi}{2m+1} \alpha^{i-l} g_{(m,i,l,n)} Y_{m,k}^*(\theta', \phi') Y_{m,k}(\theta, \phi) \left( \frac{r^i}{r^l} \right) + \varepsilon. \quad (17)$$

The values for the coefficients  $g_{(i,j,k,n)}$  for  $p = 0$  to 3 are given in Table 1. It can be noted from the table that the  $g_{(i,j,k,n)}$  coefficients are equal to zero for  $i + j$  equal to an odd integer or for  $j < i$ .

$i$	$j$	$k$	$g_{(i,j,k,n)}$	$i$	$j$	$k$	$g_{(i,j,k,n)}$
0	0	0	1	1	3	3	$n(n-1)(n+2)/10$
0	2	0	1/6	2	2	0	1/3
0	2	1	$(n-1)/3$	2	2	1	$(2n+1)/3$
0	2	2	$n(n-1)/6$	2	2	2	$n(n+2)/3$
1	1	0	1	3	3	0	1/15
1	1	1	$n$	3	3	1	$(n+1)/5$
1	3	0	1/10	3	3	2	$(n^2+3n+1)/5$
1	3	1	$(3n-2)/10$	3	3	3	$n(n+2)(n+4)/15$
1	3	2	$(n-1)(3n+2)/10$				

Table 1.  $g_{(i,j,k,n)}$  Coefficients for  $p = 0$  to 3

We are now in a position to write the multipole expansion for sources near the expansion center. Using Equation (17), the multipole expansion can be written in terms of coefficients that are a function of the strengths  $S_j(\mathbf{r}'_j)$  and locations of the sources.

$$\sum_{j=1}^{N_s} S_j(\mathbf{r}'_j) \psi(\mathbf{R}) = \frac{e^{-\alpha r}}{r^n} \sum_{i=0}^p \sum_{m=0}^i \sum_{l=0}^i \sum_{k=-m}^m A_{(i,m,k)} \alpha^{i-l} g_{(m,i,l,n)} Y_{m,k}(\theta, \phi) r^{-l} + \varepsilon, \quad (18)$$

$$A_{(i,m,k)} = \sum_{j=1}^{N_s} \left[ \frac{S_j 4\pi r_j^i}{2m+1} Y_{m,k}^*(\theta'_j, \phi'_j) \right].$$

By rearranging the sums, Equation (18) may be rewritten more compactly as:

$$\sum_{j=1}^{N_s} S_j(\mathbf{r}'_j) \psi(\mathbf{R}) = \frac{e^{-\alpha r}}{r^n} \sum_{l=0}^p \sum_{m=0}^p \sum_{k=-m}^m Q_{(m,k,l,n)} Y_{m,k}(\theta, \phi) r^{-l} + \varepsilon, \quad (19)$$

$$Q_{(m,k,l,n)} = \sum_{i=\max(l,m)}^p A_{(i,m,k)} \alpha^{i-l} g_{(m,i,l,n)}.$$

We note that in the special case where  $\alpha = 0$ , one of the sums in Equation (18) can be removed since  $i = l$  yields the only nonzero values of  $\alpha^{i-l}$ . For the additional special case where  $n = 1$ , we find that  $g_{(m,i,l,n)}$  is equal to 1 for  $m = i$  and zero otherwise. This allows removal of yet another sum to yield the familiar multipole expansion for the vector potential associated with the vortex problem. The number of  $A_{(i,j,k)}$  coefficients for this special case is  $N_{Aco} = (p+1)^2$ . It can be shown that for the general case, the number of  $A_{(i,j,k)}$  coefficients is equal to  $N_{Aco} = (p+1)(p+2)(p+3)/6$  which is exactly the number of coefficients that would be required if one were to use a Cartesian coordinate system. As long as  $p$  is less than 8,  $N_{Aco}$  is never more than twice  $(p+1)^2$  which is the number of coefficients required for the multipole expansion of  $1/|\mathbf{R}|$ .

### 3.2.2 Shifted multipole expansion

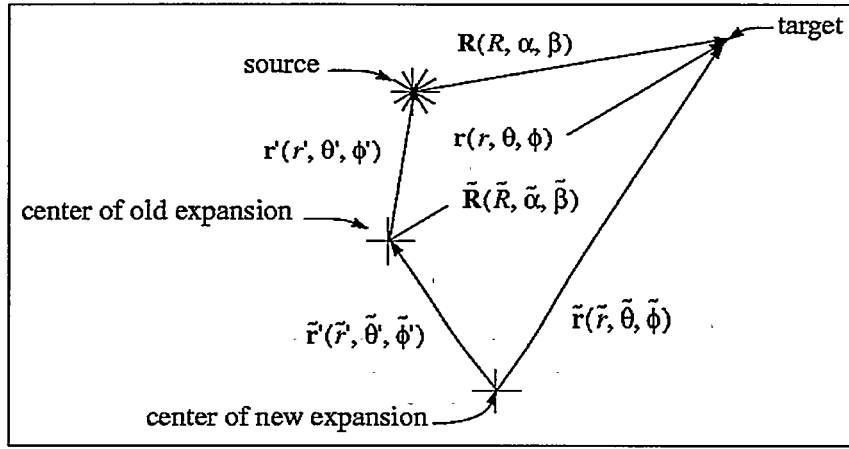


Figure 4. Geometry for Shifted Multipole

Shifting the multipole expansion of Equation (19) so that the  $Q_{(m,k,l,n)}$  coefficients might be reused is an involved process too lengthy to describe in detail here. In order to shift the expansion center according to Figure 4, an addition theorem for the quantity:

$$\frac{e^{-\alpha r}}{r^{n+l}} Y_{m,k}(\theta, \phi) \quad (20)$$

must first be developed. This quantity appears in the original unshifted multipole expansion of Equation (17). This new addition theorem is obtained by multiplying Greengard's First Addition Theorem [14], which contains the Spherical Harmonic function, times the multipole expansion of Equation (17). Multiplication of series is performed using relationships from Gradshteyn and Ryzhik [15]. This addition theorem is then placed into Equation (19) and after considerable manipulation yields the following shifted multipole expansion:

$$\sum_{j=1}^{N_s} S_j(\mathbf{r}'_j) \psi(\mathbf{R}) = \frac{e^{-\alpha \tilde{r}}}{\tilde{r}^n} \sum_{t=0}^p \sum_{q=0}^p \sum_{s=-q}^q Q'_{(q,s,t,n)} Y_{q,s}(\tilde{\theta}, \tilde{\phi}) \tilde{r}^{-t} + \varepsilon, \quad (21)$$

$$Q'_{(q,s,t,n)} = \sum_{m=0}^p \sum_{l=0}^t \sum_{k=-m}^m Q_{(m,k,l,n)} D_{(m,k,q,s,n+l,t-l)}.$$



The  $D_{(m,k,q,s,n+l,t-l)}$  operator of Equation (21) is given by the following:

$$D_{(I,J,q,s,N,M)} = \sum_{k=M}^p \sum_{K=0}^M \sum_{m=0}^{k-K} \sum_{j=-m}^m \sum_{n=-K}^K \alpha^{k-M} \mathcal{G}_{(m,k-K,M-K,N-I-1)} \frac{A_I^J A_K^n B_I^K}{A_{K+I}^{n+J}} \Lambda_{(k,m,j,K,n)} \xi_{[\{m,j\},\{K+I,n+J\},\{q,s\}]}, \quad (22)$$

where

$$A_i^j = \frac{(-1)^i}{\sqrt{(i-j)!(i+j)!}}, \quad B_i^m = \sqrt{\frac{4\pi(2i+1)}{(2m+1)[2(m+i)+1]}}, \quad (23)$$

$$\Lambda_{(k,m,j,i,n)} = \frac{4\pi \tilde{r}'}{2m+1} Y_{m,j}^* (\tilde{\theta}', \tilde{\phi}) Y_{i,n}^* (\tilde{\theta}', \tilde{\phi}), \quad (24)$$

and

$$\xi_{[\{i,j\},\{k,l\},\{q,s\}]} = \sqrt{\frac{(2i+1)(2k+1)}{4\pi(2q+1)}} \begin{pmatrix} i & k & q \\ j & l & s \end{pmatrix} \begin{pmatrix} i & k & q \\ 0 & 0 & 0 \end{pmatrix}. \quad (25)$$

The symbols  $\begin{pmatrix} * & * & * \\ * & * & * \end{pmatrix}$  in Equation (25) are Clebsch-Gordon coefficients [16].

#### 4 PARALLELIZATION

The Clarke-Tutty scheme, in which each parent box is divided into two child boxes with equal populations, offers several advantages from the perspective of parallelization. First, the number of levels and boxes in the tree are predetermined once the total number of sources is known, and the maximum number of sources that a box on the last level may contain is specified. Secondly, all boxes on a given level will contain the same number of sources (within plus or minus one). Thirdly, all branches of the tree extend to the same level. Finally, a given box's descendants (ancestors) can easily be identified by recursively multiplying (dividing) by two using integer arithmetic, without having to search the entire tree. Because the number of boxes at each level will always be an integer power of two, we found it advantageous to require that the number of processors applied to the problem also be an integer power of two. The number of boxes at each level is then always evenly divisible by the number of processors, or *vice versa*, depending on the level and number of processors chosen. Thus, by equipartitioning the boxes at each level among the processors, equipartitioning of the sources, and hence computational effort, is also guaranteed. Finally, simple algebraic relationships can be used to determine, for a given box number, the number (rank) of the processor(s) holding its source data and, conversely, on a given rank processor, the box(es) which hold source data.

Initially, it is assumed that the source data are distributed in some unknown fashion across all the processors. As each level in the tree is generated using recursive coordinate bisection, sources are moved between processors as needed to maintain the box/processor associations noted above. Once the last level is generated, each processor independently computes multipole coefficients for each box for which it holds source data. A global communication is then performed to transmit the results to all the other processors. Each processor now holds multipole coefficients for all the boxes on the last level and can independently compute, by adding and shifting coefficients from child boxes to their parent as noted earlier, multipole coefficients at all the other (coarser) levels in the tree. This last step is admittedly redundant but, fortunately, consumes a negligible amount of time. The end result is that all processors will hold a complete set of coefficients for all boxes in the tree.

The target points are treated as a completely separate population from that of the sources, even though in many applications the majority of target points may coincide with source points. Target points are distributed evenly across all processors at the outset and since they do not participate in the source tree generation, remain on the same processor. To evaluate the field quantities for a given target, each processor starts at the coarsest level in the tree and determines which, if any, boxes on that level are sufficiently removed from the target that their influence can be computed using their multipole expansion. If a box's expansion is used, any further contributions from any of its descendant boxes are ignored, since they have already been included. This process is repeated for progressively finer levels in the tree through the last, ignoring those boxes whose contribution has already been accounted for. The tree traversal is repeated anew for each target; the result is that each processor will then hold the contribution to the field at all of its targets

from those sources lying in boxes for which multipole expansions can be used. Note that this step requires no inter-processor communication.

It remains to compute the contributions from sources in the childless boxes on the last tree level whose influence has not already been accounted for by the multipole expansions. These are computed as direct pair-wise interactions between the target and the individual sources, and inter-processor communication is necessary for this step. Each processor first computes interactions between its target points and those childless boxes whose source points it is also holding. Direct interactions with a box's sources are performed *only if the childless box and all its ancestors fail the test for using the multipole expansion*. Then a ring-type communication loop is entered; on the first pass through the loop, processor  $q$  receives all the source data from processor  $q-1$  and stores it in a temporary buffer, while simultaneously sending a copy of its own source data to processor  $q+1$ ; then all processors compute direct interactions between their targets and the just-received source data. On the second pass, source data is received from processor  $q-2$ , overwriting the temporary buffer and  $q$ 's source data is sent to processor  $q+2$ , etc., until, after  $N_{proc} - 1$  passes through the loop, every processor has had the opportunity of computing direct interactions between its targets and the sources held by every other processor. After this loop is completed, each processor will hold the final field values for all of its targets. The load balancing achieved by this algorithm has been excellent; for the three largest populations considered in Section 5, the maximum difference in CPU times between processors was always less than 1%.

## 5 BENCHMARK TESTS

Two sets of computations were made in order to obtain some assessment of the performance of the fast solver. In the first set of computations, the irradiance field as given by Equation (7) was calculated for a uniform temperature field on the inside of a cube. The sides of the cube were one optical path in length. In the second set of computations, the velocity vector field given by Equation (3), as well as the vector potential given by Equation (4), were obtained inside a cube with a uniform vorticity field specified to be  $\omega = (3\hat{i} + 4\hat{j} + 5\hat{k})/\sqrt{50}$ .

Performance results from the serial irradiance calculations are shown in Figure 5. In making these calculations, we found that the results are optimal when the number of sources per box at the finest level of the tree is on the order of 30. It appears that the CPU time for the fast solver scales as approximately  $N^{3/2}$  over the range of source and target populations  $N_S = N_T$  between 1,000 and 200,000. For  $N_S = 200,000$ , the fast solver computes the irradiance field in less than 1/10 of the time required by the direct solution technique. For this particular set of calculations, only the zeroth and first order terms ( $p=1$ ) were used in the multipole expansion. Errors in local values of the irradiance were, in general, less than 0.1% of the maximum irradiance in the cube.

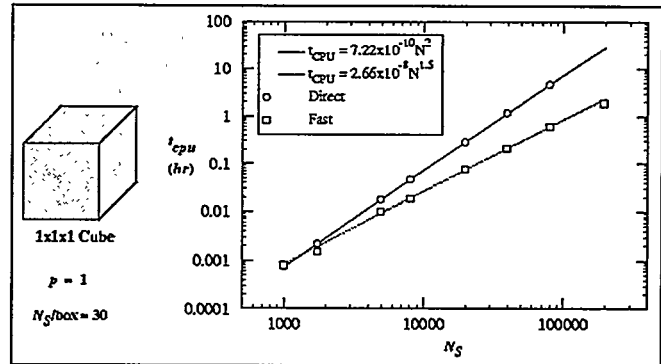


Figure 5. Direct and Fast Solver CPU Time Comparisons (Sun Sparc 20, 200 MHz Processor Speed)

Both serial and parallel computations were made for the vortex model problem. For these calculations, a second order multipole expansion ( $p=2$ ) was used. This produces a relative RMS error between the serial fast solver and the direct solver that is on the order of  $10^{-5}$  for the vector potential and on the order of  $10^{-4}$  for the velocity field. For  $N_S = N_T = 10^6$ , the optimum number of sources per box at the finest level of the tree was found to be on the order of 250. The optimum range is actually fairly broad, ranging from about

125 to 500. All parallel computations were performed on Sandia's TeraFlop machine, using between 1 and 2,048 processors; each processor is an Intel Pentium Pro rated at 200 MHz with access to 128 MB of RAM.

A measure of the parallel efficiency can be obtained by plotting the number of processors,  $N_{proc}$ , divided by the number of vortices,  $N_v$ , times the CPU time versus the number of vortices divided by the number of processors. Such plots for the present set of runs are shown in Figure 6. For perfect efficiency, curves of constant  $N_v$  would be horizontal lines. Based on Figure 6, it appears that for cases where there are at least 1,000 vortices per processor, the parallel efficiency is quite good.

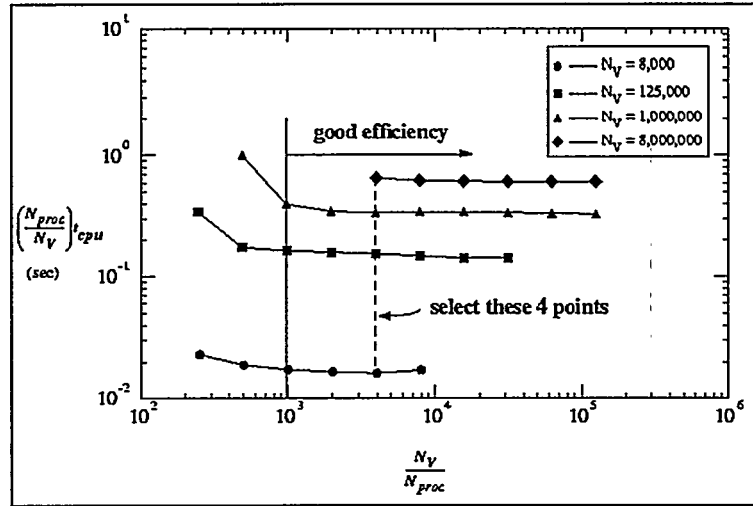


Figure 6. Parallel Efficiency Trends

In order to compare the CPU time of the parallelized vortex code with the direct solver, the four points indicated in Figure 6 are plotted in Figure 7. As can be seen from Figure 7, the present fast solver is about 30 times faster than the direct solution for  $10^7$  vortices and target points. In Figure 7 we also plot the CPU times of Warren and Salmon [17] for the case where vortex elements are initially placed on the surface of a sphere and allowed to convect in the flow. We present this data only to show the similarity in the CPU time scaling trend versus the number of sources and targets. There are several differences in the conditions under which the two computations were made and this comparison should be viewed as being only qualitative.

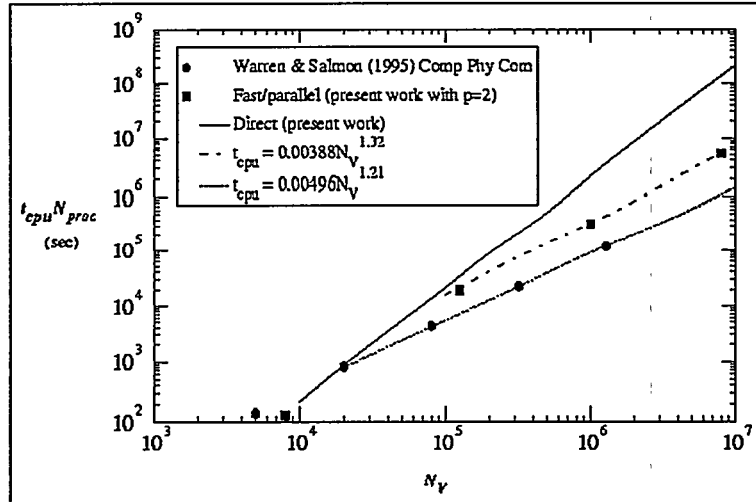


Figure 7. Comparison of CPU Times for Parallelized Fast Solvers

## 6 SUMMARY

In this paper, we have presented a synopsis of the derivation of the multipole and shifted multipole expansions for the three-dimensional radiation fast solver. A fast solver, that yields good parallel efficiencies on massively parallel distributed memory machines, has been developed. This fast solver has been benchmarked for a model problem in which the irradiance field was calculated from the temperature field. For 200,000 sources and targets, the serial version of the fast solver was more than an order of magnitude faster than the direct solver. The parallel version of the code was used on a model problem to extract the velocity field from the vorticity field. Results for up to 125,000,000 sources and targets have been computed on Sandia's TeraFlop computer. High parallel efficiencies were obtained in all cases in which at least 1,000 sources and targets were placed on each processor.

## 7 REFERENCES

- [1] Greengard, L., "Fast Algorithms for Classical Physics," *Science*, Vol. 265, pp. 909-914, August 1994.
- [2] Strickland, J. H. and Baty, R. S., "A Pragmatic Overview of Fast Multipole Methods," *Lectures in Applied Mathematics*, Vol. 32, pp. 807-830, 1996.
- [3] Batchelor, G. K., "An Introduction to Fluid Dynamics," Cambridge University Press, pp. 84-87, 1992.
- [4] Gritzo, L. A. and Strickland, J. H., "A Vortex and Transport Element Method Compatible Technique for Solving 1D and 3D Radiative Heat Transfer," Third International Workshop on Vortex Flows and Related Numerical Methods, Toulouse, France, August 24-27, 1998.
- [5] Salmon, J. K., Warren, M. S., and Winkelmans, G. S., "Fast Parallel tree Codes for gravitational and Fluid Dynamical N-Body Problems," *Int. J. Supercomputer App. & High Perf. Comp.*, Vol. 8, NO. 2, pp. 129-142, Summer 1994.
- [6] Coifman, R., Rokhlin, V., and Wandzura, S., "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Antennas and Propagation Magazine*, Vol. 35, No. 3, June 1993.
- [7] Barnes, J. E. and Hut, P., "A Hierarchical Force Calculation Algorithm," *Nature*, Vol. 324, No. 4, pp. 446-449, December 1986.
- [8] Greengard, L. and Rokhlin, V., "On the Efficient Implementation of the Fast Multipole Algorithm," Yale University Report YALEU/DCS/RR-602, February 1988.
- [9] Carrier, J., Greengard, L., and Rokhlin, V., "A Fast Adaptive Multipole Algorithm for Particle Simulations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, No. 4, pp. 669-696, July, 1988.
- [10] Strickland, J. H., Amos, D. E., "A Fast Solver for Systems of Axisymmetric Ring Vortices," *AIAA Journal*, Vol. 30, No. 3, pp. 737-746, March 1992.
- [11] Clarke, N. R. and Tutty, O. R. "Construction and Validation of a Discrete Vortex Method For the Two-Dimensional Incompressible Navier-Stokes Equations," *Computers & Fluids*, Vol. 23, No. 6, pp.751-783, 1994.
- [12] Gritzo, L. A. and Strickland, J. H., "A Gridless Solution of the Radiative Transfer Equation for Fire and Combustion Calculations," to appear in *Combustion Theory and Modeling*, 1998.
- [13] Jackson, J. D., *Classical Electrodynamics*, Second Edition, pp. 99-102, John Wiley and Sons, Inc., New York, 1975.
- [14] Greengard, L., "The Rapid Evaluation of Potential Fields in Particle Systems," Yale University Report YALEU/DCS/RR-533, April 1987.
- [15] Gradshteyn, I. S., and Ryzhik. I. M. "Table of Integrals, Series, and Products," Academic Press, New York, 1980.
- [16] CRC Standard Mathematical Tables and Formulae, 30th Edition, pp. 492-493, D. Zwillinger, Ed., CRC Press, 1996.
- [17] Warren, M. S., Salmon, J. K., "A Portable Parallel Particle Program," *Computer Physics Communications*, Vol. 87, pp. 266-290, 1995.