A Pragmatic Overview of Fast Multipole Methods

James H. Strickland and Roy S. Baty

Abstract A number of physics problems can be modeled by a set of N elements which have pair-wise interactions with one another. A direct solution technique requires computational effort which is $O(N^2)$. Fast multipole methods (FMM) have been widely used in recent years to obtain solutions to these problems requiring a computational effort of only $O(N \ln N)$ or $O(N)$. In this paper we present an overview of several variations of the fast multipole method along with examples of its use in solving a variety of physical problems.

1. INTRODUCTION

Many problems in science and engineering can be cast in terms of integral equations of the form:

$$\psi(\underset{\sim}{x}) = \int W(\underset{\sim}{x}') K(\underset{\sim}{x}, \underset{\sim}{x}')\, dx' \tag{1}$$

for which a discrete form may be written in terms of the summation:

$$\psi(\underset{\sim}{x}) = \sum_{i=1}^{N} W(\underset{\sim}{x}'_i) K(\underset{\sim}{x}, \underset{\sim}{x}'_i)\ . \tag{2}$$

For example, Roach [1] shows that the solutions for a large class of linear differential equations given by $Lu = f$ where L is a linear differential operator may be cast in this form when L^{-1} exists. In this case, the kernel $K(\underset{\sim}{x}, \underset{\sim}{x}')$ is a Green's function for the operator L and $W(\underset{\sim}{x}') = f$. Examples of classical physics problems which may be cast in terms of Equation 2 as noted by Greengard [2] include N body gravitational problems, electrostatic fields, magneto-statics, heat conduction, and acoustic fields. Several example kernels $K(\underset{\sim}{x}, \underset{\sim}{x}')$ are given in Table 1. The primary interest of the authors in this subject area involves the use of fast algorithms to facilitate computations associated with gridless vortex methods in fluid mechanics. In these methods, it is necessary to obtain the velocity vector $\underset{\sim}{u}(\underset{\sim}{x})$ in terms of the vorticity vector $\underset{\sim}{\omega}(\underset{\sim}{x}')$ and the divergence of the velocity field $D(\underset{\sim}{x}')$. Use of the Helmholtz decomposition for a vector field [3] allows us to accomplish this:

$$\underset{\sim}{u}(\underset{\sim}{x}) = \nabla \times \int_{R_\infty} \underset{\sim}{\omega}(\underset{\sim}{x}') K(\underset{\sim}{x}, \underset{\sim}{x}')\, dR(\underset{\sim}{x}') - \nabla \int_{R_\infty} D(\underset{\sim}{x}') K(\underset{\sim}{x}, \underset{\sim}{x}')\, dR(\underset{\sim}{x}')\ . \tag{3}$$

1991 *Mathematics Subject Classification.* Primary 65D30, 65D32; Secondary 76C05, 85-08
This work performed at Sandia National Laboratories supported by the U. S. Dept. of Energy under contract No. DE-AC04-94AL85000.

No.	$K(\underset{\sim}{x}, \underset{\sim}{x}'_i)$	Physical interpretation of $\psi(\underset{\sim}{x})$
1	$\ln(\lvert\underset{\sim}{x} - \underset{\sim}{x}'_i\rvert)$	two-dimensional electrostatic potential and two-dimensional stream-function for vortex flows
2	$\dfrac{\underset{\sim}{x} - \underset{\sim}{x}'_i}{\lvert\underset{\sim}{x} - \underset{\sim}{x}'_i\rvert^3}$	three-dimensional gravitational force, force due to charged particles, velocity associated with a field of vorticity, Biot-Savart law
3	$e^{-\alpha\lvert\underset{\sim}{x} - \underset{\sim}{x}'_i\rvert^2}$	temperature due to heat conduction, diffusion
4	$\dfrac{e^{i\alpha\lvert\underset{\sim}{x} - \underset{\sim}{x}'_i\rvert}}{\lvert\underset{\sim}{x} - \underset{\sim}{x}'_i\rvert}$	three-dimensional acoustic pressure
5	$r'_i r \int_0^\infty e^{-k\lvert x - x'\rvert} J_1(kr'_i)\, J_1(kr)\, dk$	axisymmetric stream-function for vortex rings

Table 1: Example Kernel Functions

From Equation 2 one notes that in order to carry out a direct summation for N target points located at various values of $\underset{\sim}{x}$, a computational effort of order N^2 is required. This becomes very expensive for systems in which N exceeds 10^3 to 10^4. In many cases, one would like to make calculations with N on the order of 10^6 to 10^9 which is not practical using direct summation.

However, in the event that Equation 1 is a Hilbert-Schmidt integral equation [4], the kernel $K(\underset{\sim}{x}, \underset{\sim}{x}'_i)$ may be approximated by a degenerate kernel of the form:

$$K(\underset{\sim}{x}, \underset{\sim}{x}'_i) = \sum_{j=1}^{p} A_j(\underset{\sim}{x}'_i)\, B_j(\underset{\sim}{x}) \tag{4}$$

which produces a discrete problem requiring a computational effort of order Np. This can be achieved by first substituting Equation 4 into Equation 2 and rearranging the order of summation to yield:

$$\psi(\underset{\sim}{x}) = \sum_{j=1}^{p} B_j(\underset{\sim}{x}) \sum_{i=1}^{N} W(\underset{\sim}{x}'_i)\, A_j(\underset{\sim}{x}'_i)\,. \tag{5}$$

We now define the coefficients $C_j(\underset{\sim}{x}'_i)$ by:

$$C_j(\underset{\sim}{x}'_i) = \sum_{i=1}^{N} W(\underset{\sim}{x}'_i)\, A_j(\underset{\sim}{x}'_i) \tag{6}$$

so that:

$$\psi(\underset{\sim}{x}) = \sum_{j=1}^{p} B_j(\underset{\sim}{x})\, C_j(\underset{\sim}{x}'_i) \,. \tag{7}$$

The coefficients $C_j(\underset{\sim}{x}'_i)$ are computed using Equation 6 prior to carrying out the summation indicated in Equation 7. Computation of $C_j(\underset{\sim}{x}'_i)$ requires $O(Np)$ work. Since the summation in Equation 7 for all N target points at $\underset{\sim}{x}$ also requires $O(Np)$ work we see that this method requires $O(Np)$ work as opposed to $O(N^2)$ work associated with the direct solution of Equation 2. Obviously p must be significantly less than N for this scheme to provide a significant advantage.

It should be emphasized that the fast algorithm (Equation 7) is based on the fact that only information at $\underset{\sim}{x}'$ is required to obtain the precomputed coefficients $C_j(\underset{\sim}{x}'_i)$. This is, in turn, based on the ability to obtain the "separation of variables" indicated by the right hand side of Equation 4. In many problems of interest, Equation 4 is obtained from a truncated series which has a limited range of validity for a specified level of accuracy. For example the kernel $K(\underset{\sim}{x}, \underset{\sim}{x}'_i)$ might be represented by a truncated Laurent series or some other far-field representation which requires that the target point $\underset{\sim}{x}$ be far away or "well separated" from the source located at $\underset{\sim}{x}'_i$. On the other hand, the truncated series might be a Taylor series or some other near-field expansion written about some point near to a set of target points but well separated from the source points. The term "multipole expansion" has been used by Greengard and Rokhlin [6] to indicate the far-field series expansion and the term "local expansion" to indicate the near-field series expansion. This terminology will be adopted for the purposes of this paper.

As indicated, for most problems of interest, Equation 4 will be represented by multipole or local series expansions which are valid to some specified level of accuracy according to the relative separation of target and source points. Since the target and source points may, in general, be located anywhere within the domain of interest, somewhat elaborate schemes must be employed to insure that the series expansions are properly used. There are two major classes of methods which have been developed to accomplish this. The so called "hierarchical" or "tree code" schemes typified by the Barnes-Hut algorithm [5] use only multipole expansions while the Greengard Rokhlin algorithm [6] makes use of both multipole and local expansions.

2. SERIES EXPANSIONS

In this section, we will present examples of Equation 4 for several different physical problems which can be represented by multipole or local series expansions.

2.1 Multipole

In Figure 1, the geometry associated with the multipole expansion is represented. The source points are located inside the domain D_1 at points $\underset{\sim}{x}'_i$. The multipole expansion is constructed about the point $\underset{\sim}{x}_{sc}$ in D_1 and the target point is located outside of domain D_1 at the point $\underset{\sim}{x}$. It is assumed that a series expansion which is accurate to some precision can be obtained at the point $\underset{\sim}{x}$.

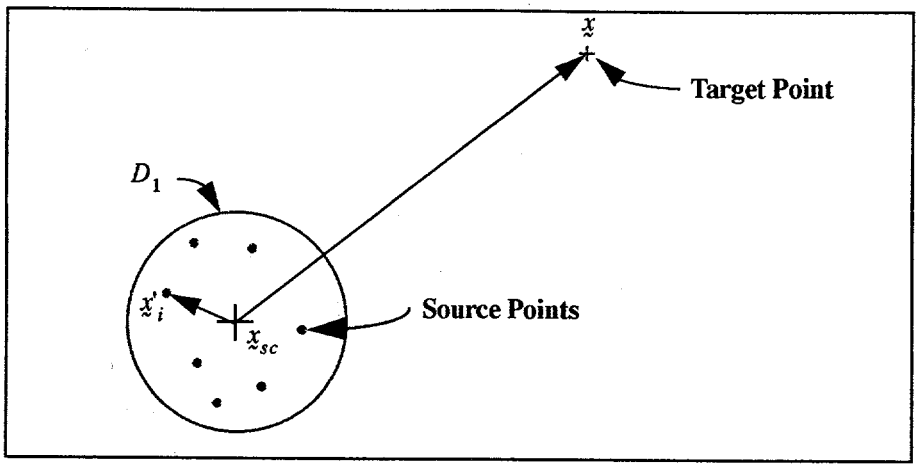


Figure 1. Multipole Expansion about Point in Source Domain

In order to illustrate the manner in which a multipole expansion may be generated, consider the simple one-dimensional example given by the equation:

$$\psi(x) = \sum_{i=1}^{N} W(x_i) K(x - x'_i) . \tag{8}$$

Our goal is to replace $K(x - x'_i)$ with a series expansion in which the variables x and x'_i are separated as in Equation 4. This may be easily achieved by expanding $K(x - x'_i)$ into a Taylor series about the point x_{sc} which is given by:

$$K(x - x'_i) = K(x - x_{sc}) - (x'_i - x_{sc}) K'(x - x_{sc}) + \frac{(x'_i - x_{sc})^2}{2!} K''(x - x_{sc}) - \ldots . \tag{9}$$

We note that if x and x_i' are replaced by positions in the complex plane z and z'_i that we have a planar two-dimensional multipole expansion. The problem now reduces to finding the derivatives of K and making sure that the series converges to the desired accuracy in a reasonable number of terms.

For a specific example, consider the case in which $K(x) = \ln(|x|)$. From Equation 9:

$$K(|x - x'_i|) = \ln(|x - x_{sc}|) - \sum_{j=1}^{p-1} \left\{ \frac{1}{j} \left(\left| \frac{x'_j - x_{sc}}{x - x_{sc}} \right| \right)^j \right\} - O\left[\frac{1}{p} \left(\left| \frac{x'_j - x_{sc}}{x - x_{sc}} \right| \right)^p \right] . \tag{10}$$

We see that the resulting multipole series in x converges for $|x'_j - x_{sc}| < |x - x_{sc}|$ to an accuracy ε :

$$\varepsilon = O\left[\frac{1}{p} \left(\left| \frac{x'_j - x_{sc}}{x - x_{sc}} \right| \right)^p \right] . \tag{11}$$

As a second example, consider the axisymmetric vortex flow problem studied by Strickland and Amos [7]. The contribution to the stream-function at (x, r) from a group of N axisymmetric vortex rings located at (x'_i, r'_i) is given by:

$$\psi(x, r) = \sum_{i=1}^{N} W_i K(x, x'_i, r, r'_i) , \tag{12}$$

where

$$K(x, x'_i, r, r'_i) = r'_i r \int_0^\infty e^{-k|x-x'|} J_1(kr'_i) J_1(kr)\, dk. \tag{13}$$

A Taylor series expansion can be written for ψ as:

$$\psi = \sum_{n=0}^{p} \left\{ \sum_{m=0}^{n} \left[A_{n-m,m} \frac{\partial^n}{\partial x^{n-m} \partial r_{sc}^m} (K(x, x_{sc}, r, r_{sc})) \right] \right\}, \tag{14}$$

where

$$A_{n-m,m} = \sum_{i=1}^{N} \left[W_i \frac{(-x'_i + x_{sc})^{n-m} (r'_i - r_{sc})^m}{(n-m)!m!} \right]. \tag{15}$$

Here again we see that there is a separation of variables with regard to information concerning the source points (x'_i, r'_i) and the target point x, r. The most difficult step is to obtain the mixed partial derivatives of K as indicated in Equation 14. In this particular case, these derivatives were obtained as a function of associated Legendre functions of the second kind. It is of interest to note that the partial differential equation for the stream-function given by:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial r^2} = \frac{1}{r} \frac{\partial \psi}{\partial r} \tag{16}$$

can be used to develop two equations for K. This can be accomplished by noting that each vortex ring must individually satisfy Equation 16. Since for a single ring $\psi = WK$ with K equal to a constant then:

$$\frac{\partial^2 K}{\partial x^2} + \frac{\partial^2 K}{\partial r^2} = \frac{1}{r} \frac{\partial K}{\partial r}. \tag{17}$$

Also, due to the symmetry of the formulation the following is true:

$$\frac{\partial^2 K}{\partial x^2} + \frac{\partial^2 K}{\partial r_{sc}^2} = \frac{1}{r_{sc}} \frac{\partial K}{\partial r_{sc}}. \tag{18}$$

These relationships can be differentiated to obtain relationships between higher order partial derivatives. This in effect reduces the number of partial derivatives which have to be calculated directly from associated Legendre functions. Thus, we see that utilization of the governing partial equation may be very beneficial.

As a third example, consider the three-dimensional vector potential given by:

$$\underset{\sim}{\psi}(r) = \sum_{i=1}^{N} \frac{\underset{\sim}{W}_i}{|r - r'_i|}. \tag{19}$$

Here r is the radial distance to the target point and r'_i is the radial distance to the source point. One could write Equation 19 in cartesian coordinates and obtain a Laurent series in x, y, and z for each component of ψ. This has in fact been accomplished by Zhao [8]. Since Equation 19 is a solution to the Laplace equation, there are a number of relationships between the various partial derivatives associated with the Laurent series expansion which allows a reduction in the number of terms. An equivalent expansion which involves the

use of spherical harmonics is given by Greengard [9] as:

$$\underset{\sim}{\Psi}(r) = \sum_{i=1}^{N} \sum_{n=0}^{p} \sum_{m=-n}^{n} \underset{\sim}{A}_n^m (r_i' - r_{sc}, \theta_i', \phi_i') \frac{Y_n^m(\theta, \phi)}{|r - r_{sc}|^{n+1}}. \tag{20}$$

Here, $Y_n^m(\theta, \phi)$ are spherical harmonics which are functions of the polar and azimuthal angles θ and ϕ associated with the target point. The radius to the center of expansion is given by r_{sc}. Since the coefficients $\underset{\sim}{A}_n^m$ are only a function of information available at the source points then we see that a separation of variables in the spirit of Equation 7 has been achieved.

As a last example, we will consider a case in which the derivatives associated with the Taylor series expansion can be conveniently represented by the use of a special function. The fast Gauss transform developed by Greengard and Strain [10] has a kernel given by:

$$K(x) = e^{-\alpha x^2}. \tag{21}$$

Direct application of Equation 9 to this problem (differentiation of $e^{-\alpha x^2}$) does not yield particularly satisfying results. However, if one notes that the derivatives of the kernel $K(x)$ can be written in terms of Hermite polynomials

$$K^n(x) = \frac{d^n}{dx}\left(e^{-\alpha x^2}\right) = (-1)^n e^{-\alpha x^2} H_n(\sqrt{\alpha} x) \tag{22}$$

then the kernel can be easily expanded to obtain:

$$K(x - x_i') = e^{-(x - x_{sc})^2} \sum_{n=0}^{p} \frac{1}{n!} \left[\sqrt{\alpha}(x_i' - x_{sc})\right]^n H_n\left(\sqrt{\alpha}(x - x_{sc})\right). \tag{23}$$

Greengard and Strain [10] further extend this one-dimensional fast Gauss transform to multi-dimensions.

In summary, the far-field or multipole expansions allow one to separate variables such that $O(N)$ computations may be made. These expansions may be obtained by writing a Taylor series in a chosen coordinate system. The resulting series may be further simplified by making use of the governing differential equation along with perhaps a set of special functions.

2.2 Translation of Source Domain Center

An essential requirement for virtually all of the fast solution techniques it is to be able to shift the center of the multipole expansion $\underset{\sim}{x}_{sc}$ in domain D_1 to a new center located at $\underset{\sim}{x}_{sc}'$ in domain D_2 as indicated by Figure 2. This allows one to efficiently obtain multipole expansions in large domains by summing the contributions from shifted multipole expansions associated with a number of sub-domains.

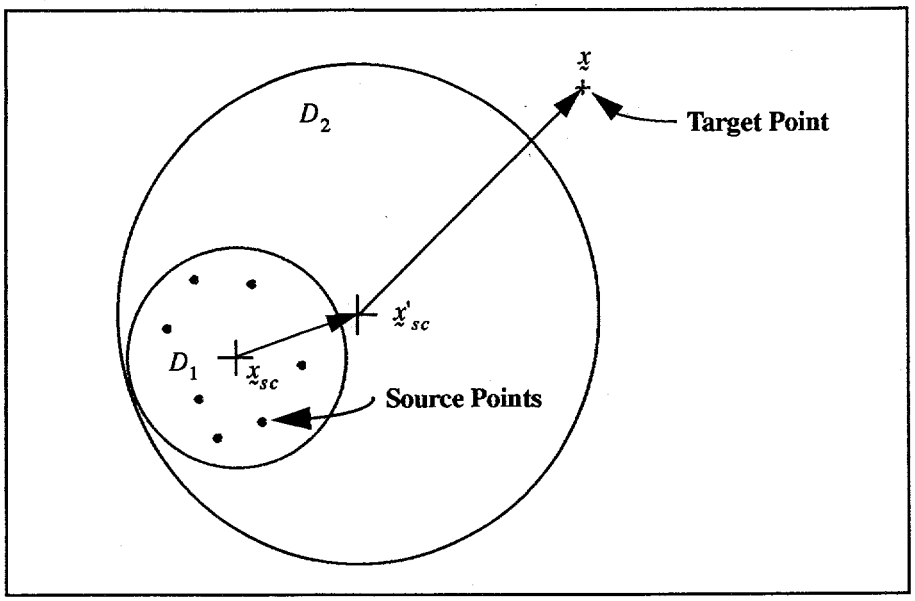


Figure 2. Translation of Source Domain Center

For the sake of brevity, we will not present shifted multipoles for all of the examples in the previous section but only consider the simplest case given by Equations 8 and 9 in order to illustrate the basic concept. From Equations 8 and 9 the function $\psi(x)$ may be written as:

$$\psi(x) = \sum_{n=0}^{p} \left[C_n(x'_i) \frac{d^n}{dx^n} K(x - x_{sc}) \right] + \varepsilon, \tag{24}$$

where

$$C_n(x'_i) = \sum_{i=1}^{N} \left[\frac{(-1)^n}{n!} W(x'_i) \, (x'_i - x_{sc})^n \right]. \tag{25}$$

In order to shift the expansion to the new center $\underset{\sim}{x}'_{sc}$ we rewrite Equation 24 as:

$$\psi(x) = \sum_{n=0}^{p} \left\{ C_n(x'_i) \frac{d^n}{dx^n} K[(x - x'_{sc}) - (x_{sc} - x'_{sc})] \right\} + \varepsilon. \tag{26}$$

This can be written in terms of a new Taylor series given by:

$$\psi(x) = \sum_{n=0}^{p} \left[C'_n(x'_i) \frac{d^n}{dx^n} K(x - x'_{sc}) \right] + \varepsilon, \tag{27}$$

where

$$C'_n(x'_i) = \sum_{q=0}^{n} \left[\frac{(-1)^q}{q!} C_{n-q}(x'_i) \, (x_{sc} - x'_{sc})^q \right]. \tag{28}$$

Thus if we have expansions for several groups of sources about several centers in domain D_2, then we can simply add their shifted coefficients $C'_n(x'_i)$ together to form a new far field expansion. The basic concept may be extended to multi-dimensional problems by using multi-dimensional Taylor series expansions. Various kernel functions may be used but they must always produce a convergent series and of course, one must be able to differentiate them if the series are formed using a Taylor series approach.

2.3 Local Expansion About Point in Target Domain

For many of the fast solution techniques, the far field or multipole expansion along with the ability to shift the expansion to a new center is all that is required. Greengard and Rokhlin [6] also take advantage of a local expansion about a point $\underset{\sim}{x}_{tc}$ in a target domain D_3 which is well separated from the source domain D_2. This local expansion allows one to obtain the influence of a group of sources on a target point as a function of the relative position between the target point at $\underset{\sim}{x}$ and the center of the local expansion at $\underset{\sim}{x}_{tc}$. This is sometimes referred to as a group-to-group interaction whereas the multipole expansion is a group-to-point interaction.

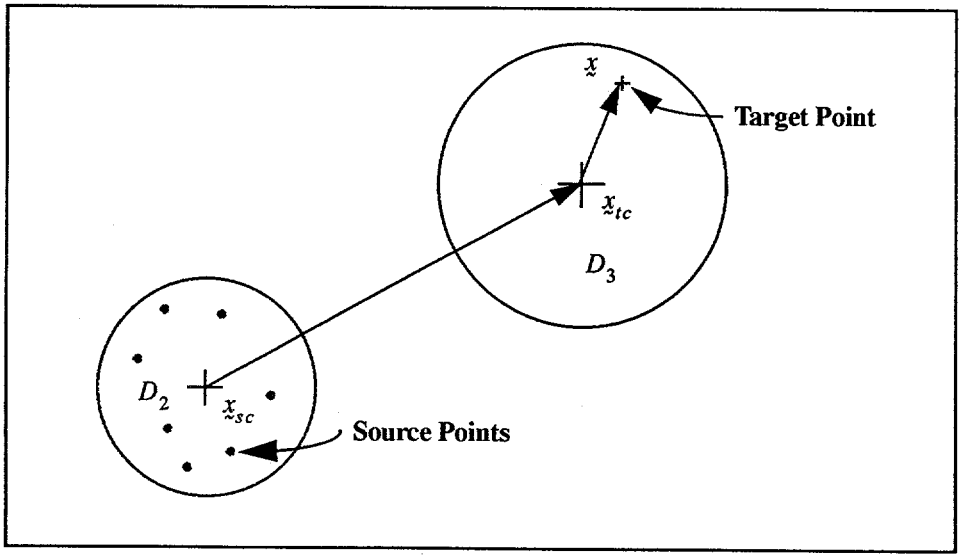


Figure 3. Local Expansion About Point in Target Domain

We can rewrite Equation 24 in the following form.

$$\psi(x) = \sum_{n=0}^{p} \left[C_n(x'_i) \frac{d^n}{dx^n} K\left[(x - x_{tc}) - (x_{sc} - x_{tc}) \right] \right] + \varepsilon . \tag{29}$$

It should be noted that Equation 27 can also be written in this form if we simply let the shifted center of expansion for the multipole be relabeled as x_{sc}. Equation 29 can be expanded and rewritten in terms of a local series given by:

$$\psi(x) = \sum_{n=0}^{p} \left[D_n(x'_i) \, (x - x_{tc})^n \right] + \varepsilon \tag{30}$$

where

$$D_n(x'_i) = \sum_{q=0}^{p-n} \left[\frac{C_q(x'_i)}{n!} \frac{\partial^{n+q}}{\partial x^{n+q}} K(x_{sc} - x_{tc}) \right]. \tag{31}$$

We note that Equation 30 is a power series in $(x - x_{tc})$. Typically, for higher dimensional problems a power series containing products of the independent variables will result.

2.4 Translation of Target Domain Center

Finally, for the Greengard Rokhlin method we must be able to shift the local expansion as indicated in Figure 4 from a center at $\underset{\sim}{x}_{tc}$ to a center at $\underset{\sim}{x}'_{tc}$.

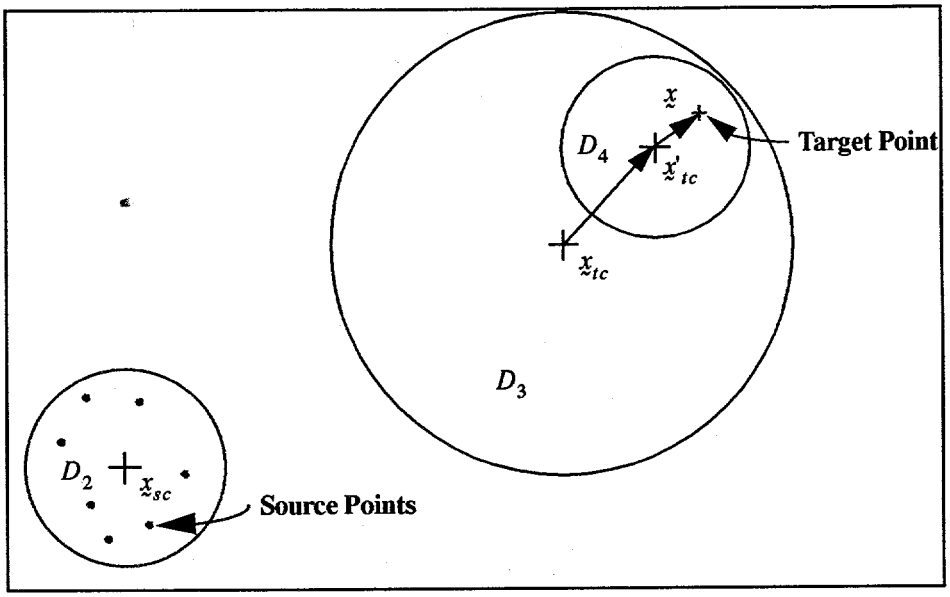


Figure 4. Translation of Target Domain Center

We first rewrite Equation 30 by inserting the center of the new target domain x'_{tc}.

$$\psi(x) = \sum_{n=0}^{p} \left[D_n(x'_i) \left[(x - x'_{tc}) - (x_{tc} - x'_{tc}) \right]^n \right] + \varepsilon. \tag{32}$$

This can be rearranged to obtain a local power series in $x - x'_{tc}$ given by:

$$\psi(x) = \sum_{n=0}^{p} \left[D'_n(x'_i) (x - x'_{tc})^n \right] + \varepsilon, \tag{33}$$

where

$$D'_n(x'_i) = \sum_{q=0}^{p-n} \left[D_{n+q}(x'_i) \frac{(n+q)!}{n!q!} (x'_{tc} - x_{tc})^q \right]. \tag{34}$$

3. SPATIAL PARTITIONING

As discussed previously, the fast solution technique is predicated on the ability to separate variables into those containing information associated with the sources and those containing information associated with the target points. In order to accomplish this, the kernel is written in terms of truncated series expansions which are valid only under certain conditions. These conditions are generally associated with the separation between sources and target points. In order to insure that these conditions are met, one is required to develop some sort of scheme to quantify the relative separation between groups of sources and target points. Virtually all of the methods which have been developed thus far possess a tree structure or hierarchy produced by the division of the space containing source and target points into progressively smaller regions. In the following two sections we will present an overview of such schemes. The methods will be divided into two broad classes which we will loosely categorize as Barnes-Hut schemes and Greengard-Rokhlin schemes. The distinguishing difference between the two methods is that the Barnes-Hut scheme utilizes only far-field expansions while the Greengard-Rokhlin scheme utilizes both near-field and far-field expansions.

3.1 Barnes-Hut Schemes

Barnes and Hut [5] developed a fast algorithm to study the interaction of two spherical galaxies which are moving relative to each other and which are made up of several thousand particles. Their work was based in part on that of Appel [11] who also developed a fast algorithm to study the interaction of particles in a gravitational force field. Barnes and Hut improved the tree structure of Appel by regularizing the spatial partitioning and were thereby able to predict and improve the accuracy of the method. In both cases, the force on a target point was calculated using the gravitational force produced by the total mass of a source cluster assumed to be located at the cluster centroid. In essence, they used a far-field multipole expansion in which only the first two terms were retained. Translation of the source domain centers was achieved by calculating the centroid of the sources in the domain.

The tree-structure of Barnes and Hut was obtained by first surrounding the source and target points by a square (cube in three-dimensions) as indicated in Figure 5. This box is further subdivided into four squares (eight cubes in three-dimensions). Subdivision of parent boxes is continued until there is only one particle per box. Empty boxes are discarded. Each box is tagged with the total mass contained in the box and the center of gravity. This includes all of the parent boxes as well. The force on any particle is obtained by sequentially examining the boxes starting with the largest. The ratio of the source box size to the distance between the box center of gravity and the target point is checked to see if it is below a certain value or if there is only one particle in the box. This, of course, insures that a certain minimum accuracy will be maintained. If the error criteria is met, the contribution from that box is computed. If not, the children of that box are examined to see if they in turn meet the error criteria. This process is carried out in such a manner so as to include the effect of every source in the field.

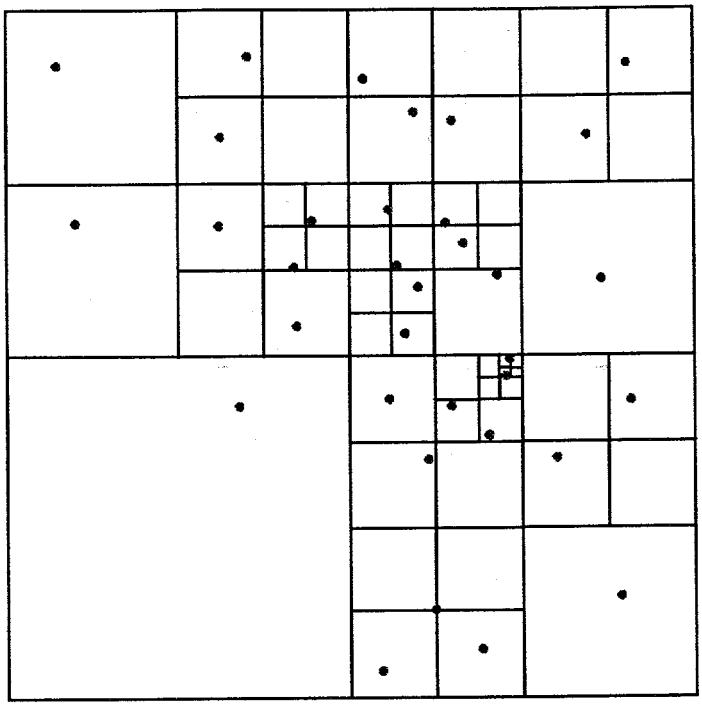


Figure 5. Barnes-Hut Hierarchical Boxing [5]

It should be pointed out, that the adaptive domain meshes which are generated here are not to be confused with the adaptive meshes which are generated for conventional CFD formulations. In the present case, the meshes represent a domain decomposition which is very simple when compared to meshes generated for conventional CFD formulations. Mesh generation for conventional CFD formulations is for the purpose of writing difference or element equations which satisfy the governing differential flow equations whereas the domain meshes generated here are for the purpose of grouping sources or target points. The domain meshes in the present case do not have to conform to any flow boundaries, they extend only to regions where sources or target points reside, and they are generated using a very simple algorithm. Generation of adaptive domain meshes is achieved using a very small amount of CPU time whereas adaptive mesh generation for conventional CFD problems can be quite CPU time intensive.

The Barnes-Hut algorithm, in its original form, is very simple. The multipole algorithm which they used and the associated translation operation were simple but only of first order. The hierarchical box structure had to have a large number of levels in order to insure that each particle would eventually be isolated. One obvious improvement to the original Barnes-Hut algorithm is to use higher order multipoles which improves the accuracy and decreases the number of levels in the box hierarchy. For example, McMillan and Aarseth [12] applied the Barnes-Hut algorithm to a stellar dynamics problem in which they used up to 8 terms in the multipole expansion.

Van Dommelen and Rundensteiner [13] developed an algorithm in which they used high order multipoles with 13 to 23 terms to model vortex interactions. They found that in order to optimize the procedure, each childless box should contain about 100 particles (vortices). They also used a binary box numbering scheme which had the position and size of the box encrypted into it.

Clarke and Tutty [14] found that with the number of multipole terms equal to 25, approximately 30 vortices in a box was optimum. However, the spatial decomposition used by Clarke and Tutty was different than that used in the original Barnes-Hut algorithm. As shown in Figure 6, the original box is divided into only two smaller boxes with half of the particles in each box. The boxes are rectangular and extend only far enough to capture half of the particles. At each step, the rectangular parent boxes are subdivided along their major axis in order to form relatively square child boxes. This method is very efficient in that the number of particles in each box at each level is essentially constant. This also allows one to start out with a rectangular box of any aspect ratio instead of a square one.

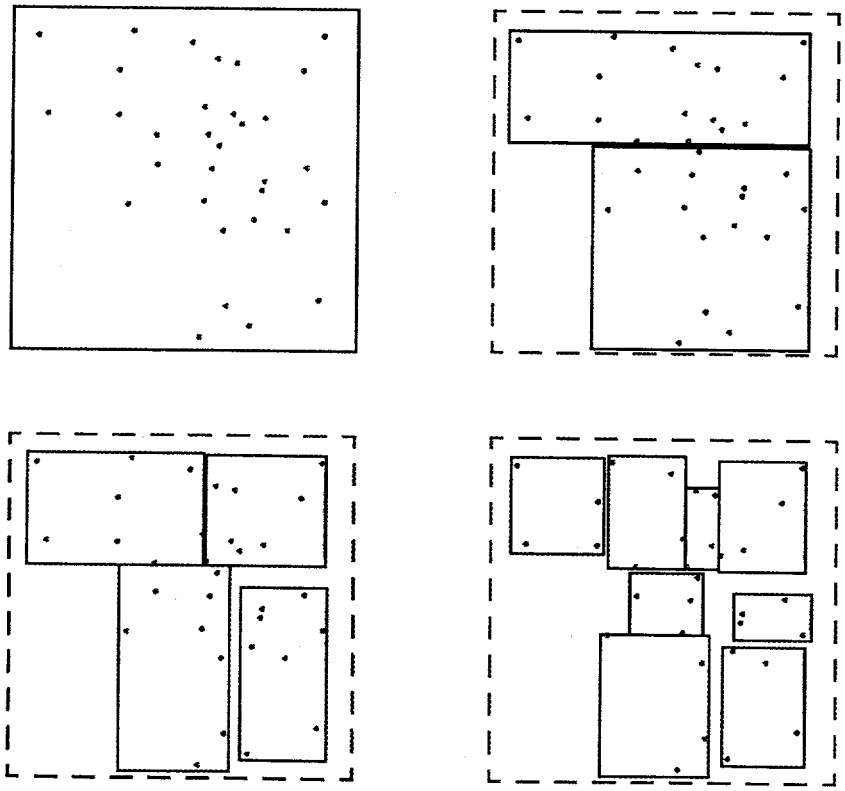


Figure 6. Clarke and Tutty Spatial Decomposition [14]

Another variation of the binary tree spatial decomposition is presented in the work of Draghicescu [15], [16]. In this method, the original box may be square or rectangular with an aspect ratio equal to two. As each box is divided into two halves, a pair of rectangles

with aspect ratio equal to two or a pair of squares will result. While this approach is not as adaptive as the Clarke and Tutty method with regard to maintaining a uniform number of particles in each box at a given level, it is somewhat less complex. It is certainly more adaptive than the quad or oct trees associated with the original Barnes-Hut method.

We close this section by noting that there are a large number of possible variations with regard to spatial decomposition. For example, in an interesting paper by Niedermeier and Tavan [17] on the dynamics of proteins produced by electrostatic interactions, the spatial decomposition is based on the structural features of the proteins themselves. The structural features of a certain protein allows them to develop an electrostatic model for individual protein molecules which may then be used in a dynamic analysis of a system of molecules.

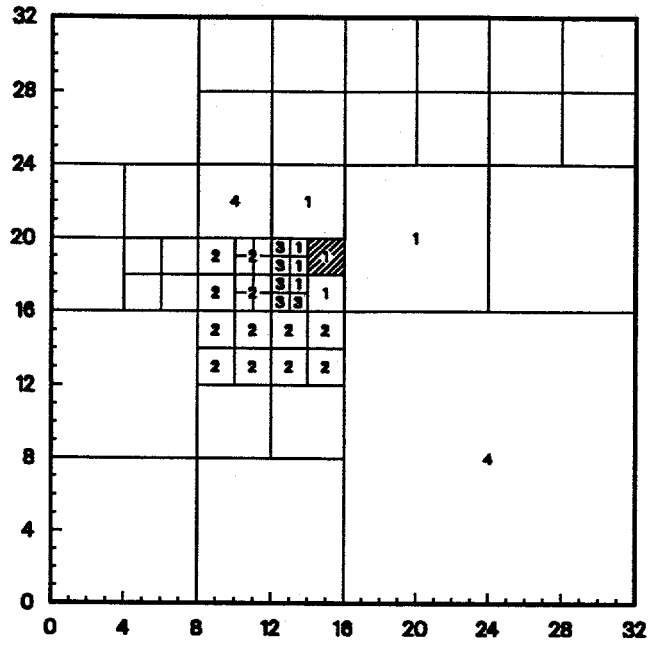
3.2 Greengard-Rokhlin Schemes

As mentioned previously, the distinguishing difference between the two methods which have come to be known as the Barnes-Hut and Greengard-Rokhlin schemes is that in addition to utilizing far-field expansions the Greengard-Rokhlin scheme also takes advantage of near-field or local expansions. This idea is clearly presented in a paper by Rokhlin [18] which predates the work by Barnes and Hut and no doubt formed the basis for the ensuing work by Greengard and Rokhlin [6], [19] as well as that of Carrier, Greengard, and Rokhlin [20]. The Greengard and Rokhlin method allows one to compute the influence of a cluster of source points on a cluster of target points in a very efficient manner. The Barnes-Hut method, on the other hand, allows one to compute the influence of a cluster of sources on a single target point in an efficient manner.

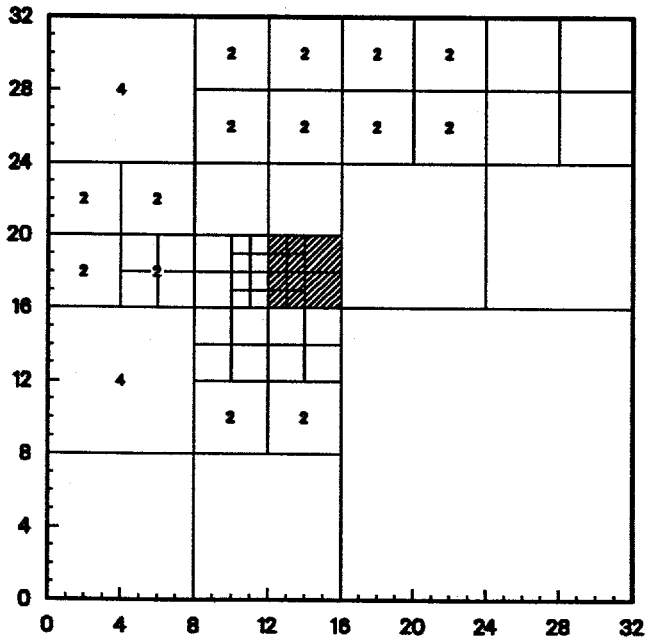
In the original Greengard-Rokhlin schemes [6], [19], the domain decomposition is non-adaptive, meaning that the domain is broken up into a set of uniform squares or cubes. The two-dimensional adaptive domain decomposition used in the work of Carrier, Greengard, and Rokhlin [20] is identical to that shown in Figure 5 except that there are typically more than one source point per box. For the adaptive scheme, a somewhat involved procedure is used to define the separation condition between a particular target box and each of the source boxes. This procedure determines the way in which sources in a particular source box influence the target points in a particular target box. In general, each target box at each level has five possible relationships with each source box in the mesh. A formal description of these "box lists" will not be given here, but in general, the five lists produce the following types of restrictions on the use of the series expansions:

1. Direct calculations must be made. Multipole and local series expansions cannot be used.
2. Both multipole and local series expansions can be used.
3. Multipole series expansions can be used, local series expansions cannot.
4. Local series expansions can be used, multipole expansions cannot.
5. Contributions from distant source boxes reside in the parent of the target box.

These lists are illustrated in Figures 7 and 8 for the cross-hatched target boxes. It is assumed that there are sources in all boxes. In Figure 7a, the list 1, 2, 3, and 4 source boxes are shown for the 2x2 target box. Figures 7b, 8a, and 8b illustrate how information from other parts of the domain is brought in through ancestors of the target box in Figure 7a.

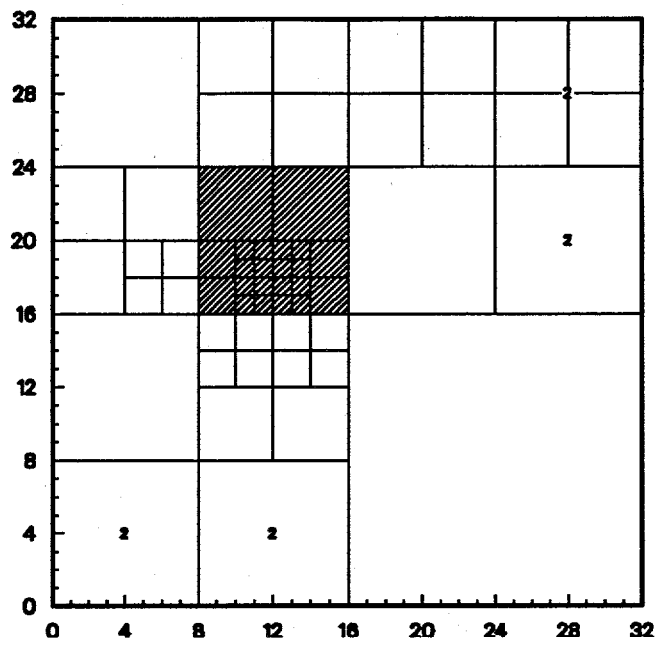


a) Contribution from childless target box's own box list.

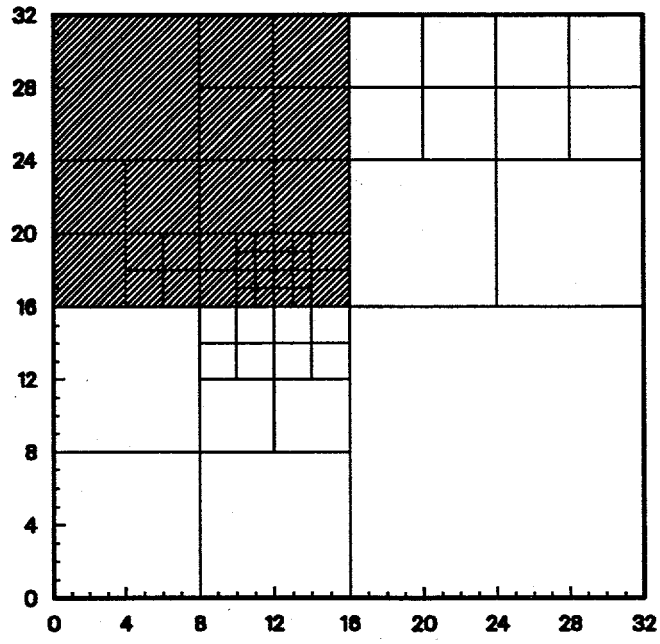


b) Contribution from childless target box's parent's box list.

Figure 7. Box List Example



c) Contribution from childless target box's grandparent's box list.



d) Contribution from childless target box's great grandparent's box list.

Figure 8. Box List Example (continued)

The Greengard-Rokhlin algorithm is significantly more complex than the Barnes-Hut algorithm especially when applied in an adaptive manner. It is perhaps for this reason that there have not been very many extensions to the basic concept. Schmidt and Lee [21] developed an algorithm for including periodic boundary conditions for the purpose of evaluating Coulomb potentials in a three-dimensional non-adaptive scheme. Strickland and Amos [7] used the adaptive technique of Carrier, Greengard, and Rokhlin [20] with a new kernel to solve for the stream function associated with axisymmetric vortical flow fields. Strickland and Baty [22] have written an adaptive three-dimensional code for the vector potential. In order to allow the source and target points to be different sets, hierarchical meshes for both the source and target points are constructed by this algorithm. The authors report that this code is under further development to include the FFT fast shift algorithms of Greengard and Rokhlin [23]. Aluru [24] has conceptualized a method which should make this and other hierarchical schemes more efficient by effectively removing redundant nodes in the tree which contain information about the same set of particles.

3.3 Comparison of Methods

In this section we provide a brief comparison of the methods. As noted previously, the Greengard-Rokhlin algorithm is significantly more complex than the Barnes-Hut algorithm and is more memory intensive. The apparent advantage of the Greengard-Rokhlin algorithm is its execution time of $O(N)$ whereas the Barnes-Hut method executes in $O(N \ln N)$.

There is some dispute in the literature about the $O(N)$ dependance for the Greengard-Rokhlin algorithm. For instance, Aluru [24] has hypothesized that both methods will execute in $O(N \ln N)$ but only after redundant nodes in the hierarchical tree structures are removed. His argument is based in part on the fact that creation of the tree structure requires $O(N \ln N)$ work. However, experience has shown that the amount of time required to create the tree structure for the Greengard-Rokhlin algorithm is trivial compared to the rest of the calculation. The major portion of the calculation, on the other hand, tends to scale like $O(N)$ depending on the distribution of the particles. Simulations in which the particle distributions tend to be non-uniform (especially concentrated along lines or surfaces) generally execute in $O(N)$ time. While creation of the tree may indeed require $O(N \ln N)$ operations, the constant of proportionality apparently is relatively small.

Blelloch and Narlikar [25] made a direct comparison for the potential problem between the three-dimensional Barnes-Hut and Greengard-Rokhlin methods. They found that at a high level of accuracy (RMS-error $< 10^{-5}$), the Greengard-Rokhlin method was faster than the Barnes-Hut method for $N > 10^4$, assuming a random distribution of points. At a lower level of accuracy (RMS-error $< 10^{-3}$), the Greengard-Rokhlin method did not outperform Barnes-Hut until $N > 10^8$. Both algorithms were implemented on parallel machines. Thus, we see that the choice of methodology is a function of the number of particles in the field and the error requirements.

We also note that memory requirements may also play a role. The Greengard-Rokhlin method requires more memory than the Barnes-Hut algorithm since both the far-field and local expansions for each box at each level must be retained.

4. PARALLELIZATION

This section discusses briefly two approaches which have been developed to parallelize the numerical evaluation of the discrete N-body problem given by Equation 2. The parallelization of the N-body problem is currently a very active area of computational research; therefore, a general discussion is beyond the scope of the present paper.

Over the last several years, a significant amount of research has been directed toward developing and implementing algorithms to evaluate efficiently the discrete N-body problem on parallel computers. While some of this work has focused on general parallelization methods, a great deal of it has emphasized specific computer architectures and codes. Two techniques applied commonly are hierarchical methods and direct methods. Most parallelizations of the N-body problem utilize one or both of these methods. In a general sense, the hierarchical methods include both the tree method of Barnes and Hut [5] and the fast multipole method of Greengard and Rokhlin [6] discussed in this paper.

Greengard and Gropp [26] have developed a parallel two-dimensional non-adaptive fast multipole method for the Encore Multimax 320 computer. In this work, Greengard and Gropp parallelized the computation of the initial moments and the evaluation of the local series expansions and found that the main limitation was the coordination of the processors required in passing information between the different mesh levels. For the details of the performance of this scheme see [26]. Work on a parallel version of an adaptive fast multipole method has presented in Singh et al. [27], while Zhao and Johnsson [28] have developed a parallel three-dimensional multipole method for the Connection CM-2 computer. Work on a parallel version of the tree method of Barnes and Hut has also been presented, for example see Salmon [29] and Grama et al. [30].

Parallel versions of the direct method have also been developed. This approach makes no attempt to simplify or reduce Equation 2. An example of this approach is given in the recent technical note of Stiller et al. [31]. In this work, some simple coding procedures are given which have been shown to speed up the direct method an order of magnitude on the massively parallel computer CM-200. A further example of a parallelized direct method and its application to vortex methods is given by Sethian et al. [32].

5. EXAMPLE CALCULATIONS

In the following examples, we present a mixture of model problems and physical simulations. In the model problems, the distributions of source and target points are prescribed. These problems allow one to carefully benchmark the performance of the various methods in terms of CPU time, accuracy, and the specified geometry. Most of the physical simulations represent an evolutionary process in which the sources and targets may be moving at each time step.

5.1 A Two-Dimensional Fast Adaptive Multipole Code

Carrier, Greengard, and Rokhlin [20] developed a two-dimensional adaptive code which is an extension of the two-dimensional non-adaptive code of Greengard and Rokhlin [6]. This code was used to obtain the potential and electrostatic field due to a distribution of charged particles. The precision was set so as to yield an RMS error roughly equal to single precision on a VAX-8600 which they were using (10^{-5} to 10^{-6}).

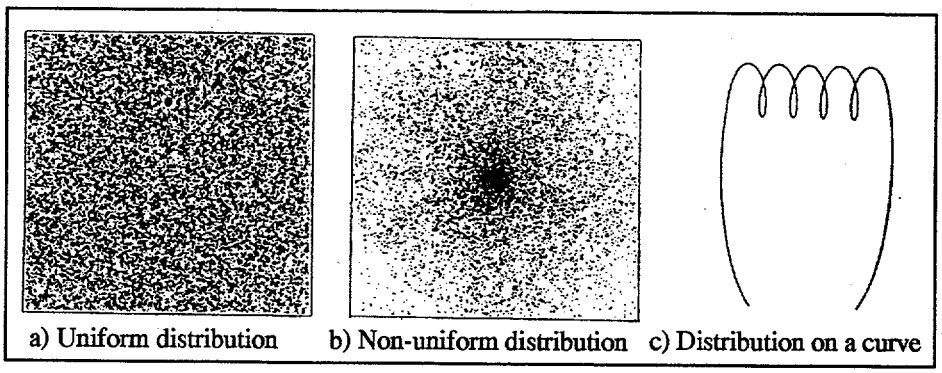


Figure 9. Benchmark Distributions

Three benchmark source and target distributions which they studied are indicated in Figure 9. For a benchmark problem in which 25,600 sources and targets were uniformly distributed in a square, the direct solution required 9694 seconds while the non-adaptive algorithm required 138 seconds, and the adaptive algorithm required 97 seconds. For a highly non-uniform distribution in a square, the direct and adaptive methods performed essentially as before while the non-adaptive algorithm required 2318 CPU seconds. For all of the points distributed along a curve, the direct solution performed as before, the non-adaptive algorithm required 152 seconds, while the adaptive algorithm required only 48 seconds. It should be noted that the adaptive algorithm performed as an $O(N)$ algorithm for this case but not for the other two distributions.

5.2 A Three-Dimensional Fast Adaptive Multipole Code

Strickland and Baty [22] developed an extended version of the three-dimensional Greengard algorithm [9]. This algorithm is adaptive and features a box hierarchy for the sources which may be different from that of the targets. In this code, each component of the vector potential is first obtained. For the case where the vector potential is used to represent a blob of vorticity, the curl of the vector potential yields the velocity vector. Thus we see that this code is required to do more than three times as much work as one which calculates a simple scalar potential. Three different sets of source and target point distributions used to benchmark the method are shown in Figure 10. In each case, the number of sources and targets were both equal to N.

The code was run at a precision such that the RMS error in the magnitude of the velocity vector would be on the order of 0.1%. The RMS error in the magnitude of the vector potential was one to two orders of magnitude less than that associated with the velocity vector. All runs were made on a SUN Sparc 10 workstation. For the cube domain the fast solver was faster than direct calculations for $N > 5000$ and was an order of magnitude faster for $N = 100,000$. Computation time for the fast solver was about 8,000 seconds. For the parallelepiped configuration, the break-even point occurred slightly earlier at $N = 3000$ but the CPU times for $N = 100,000$ were essentially the same as for the cube configuration. For the separated cube domain configuration, the break-even occurred at $N = 1000$ with the fast solver requiring only 800 seconds of CPU time for $N = 100,000$. In this later case, the fast solution technique truly performed as an $O(N)$ algorithm.

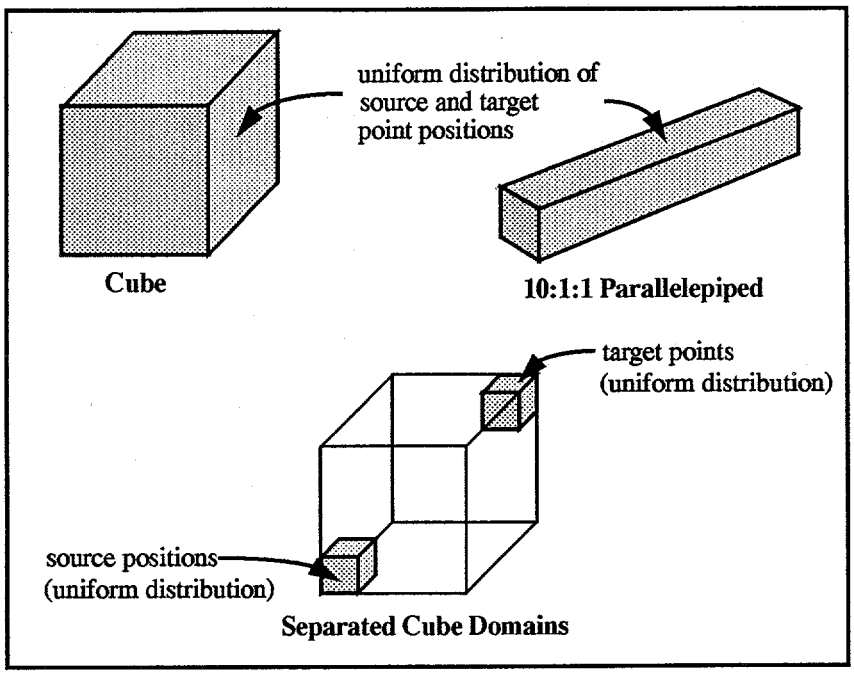


Figure 10. Benchmark Configurations

5.3 Fast Gauss Transform

Greengard and Strain [10] used their fast Gauss transform (case number 3 in table 1) to make calculations for N sources at N target points for the two geometries shown in Figure 11. The source and target point locations are congruent. In Figure 11a the points are distributed uniformly in a unit square while in Figure 11b the points are distributed uniformly on a unit circle. Computations using several different levels of precision and numbers of source and target points were made. For $N = 102,400$ the direct solution required more than 8 days of CPU time on a Sun-4 workstation. Using the fast method, the solution for the square was obtained in only 9 minutes and the solution for the circle was obtained in only 5 minutes. Errors normalized by the total source strengths were approximately 5×10^{-10} and 4×10^{-11} for the square and circle respectively.

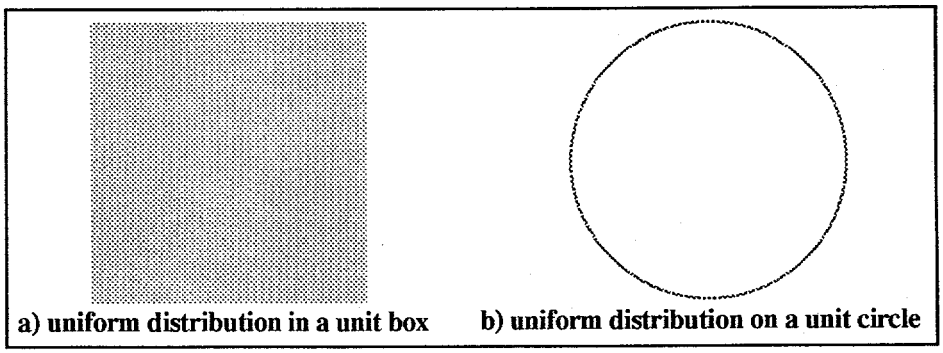


Figure 11. Example Fast Gauss Transform Source and Target Distributions

5.4 An Axisymmetric Bluff-Body Flow

The stream function and velocity components produced by axisymmetric ring vortices are much more expensive to compute than for those produced by vortices associated with the two-dimensional planar case. As mentioned previously, Strickland and Amos [7] developed a fast multipole algorithm based upon the adaptive Carrier, Greengard, and Rokhlin method [20] but utilizing the kernel associated with the stream function for axisymmetric ring vortices. This algorithm breaks even for $N = 100$ and is about 30 to 100 times faster than the direct solver for $N = 10,000$. Precisions for the velocity calculations were set to produce RMS errors on the order of 0.1%. Errors associated with the stream function were one to two orders of magnitude lower. The original benchmark studies were run on a VAX-8600. The CPU time required to compute the field for $N = 10,000$ ranged from about 400 seconds to about 1350 seconds. Present runs on a SUN Sparc 10 are roughly an order of magnitude faster.

Strickland [33] developed an axisymmetric vortex code to study the flow over parachutes and other axisymmetric shells which utilizes this fast solver. In the work of Higuchi, Balligand, and Strickland [34] unsteady flow over a disk was simulated and compared with experimental results. The disk was first accelerated to a constant velocity, held at that velocity for about three diameters of travel, and then decelerated to rest. In Figure 12, experimental flow visualization of the wake is shown just after the disk begins to decelerate and just prior to its coming to rest. Also, the experimental drag coefficient C_d versus non-dimensional tome T is shown along with simulation data obtained from the fast vortex method (FVM).

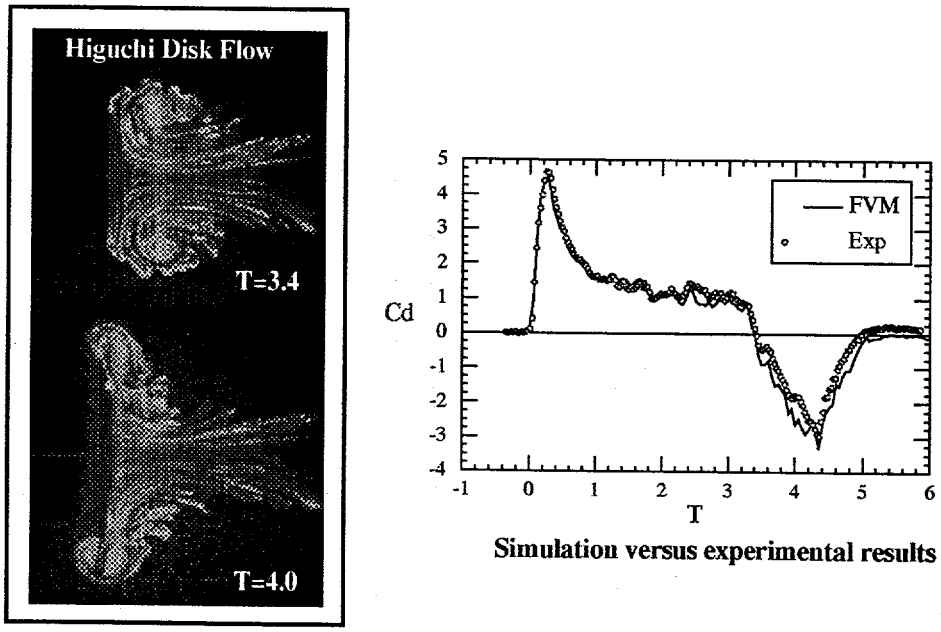


Figure 12. Simulation of Unsteady Flow over a Disk

5.5 An Astrophysics Simulation

Salmon, Warren, and Winckelmans [35] present a gravitational simulation of the formation of large scale structures in the universe. The particles in their simulation "represent the so-called "Dark Matter" which is believed to dominate the mass of the universe." The region which they simulated was "a sphere containing 1,099,135 bodies." Each of these bodies had a mass of 3.3×10^7 solar masses. A snapshot of the simulation is shown in Figure 13 at a late period in the evolution after significant clumping has occurred.

They used a Touchstone Delta system to perform the computations. The original Barnes-Hut scheme was used in which the three-dimensional space was divided into an adaptive oct-tree and the mass and center of mass provided a two-term multipole. They describe in some detail the methodology which they used in parallelizing the code. For 512 processors, approximately 50 seconds was required to compute a single time step with 1×10^6 bodies. With 32 processors the time increased to approximately 400 seconds per time step.

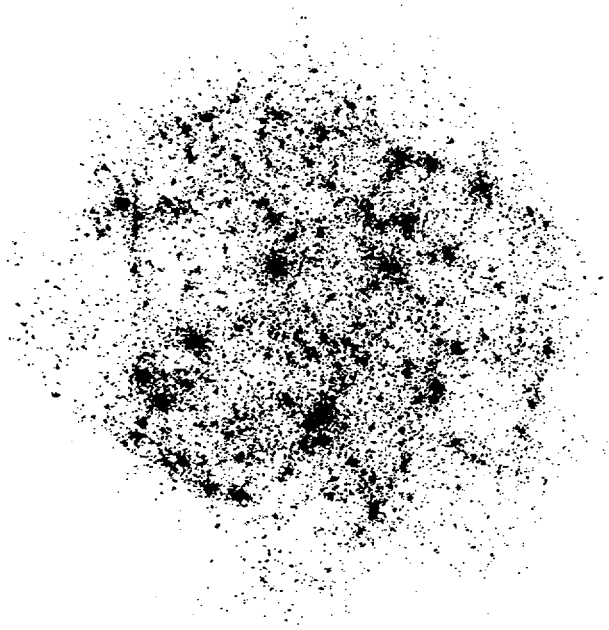


Figure 13. Clumping of Dark Matter in the Universe

5.6 Flow Past a Pitching Airfoil

The Barnes-Hut method of Van Dommelen and Rundensteiner [13] was used to reduce the CPU time for a vortex simulation produced by Shih, Lourenco, Van Dommelen, and Krothapalli [36] of flow over a pitching airfoil. In Figure 14, the vorticity which has been generated at the airfoil surface is shown as discrete points along with the instantaneous streamlines. In Figure 15 the calculated streamlines are compared with experimentally obtained iso-vorticity lines. We note that the streamlines and iso-vorticity lines display the same flow morphology.

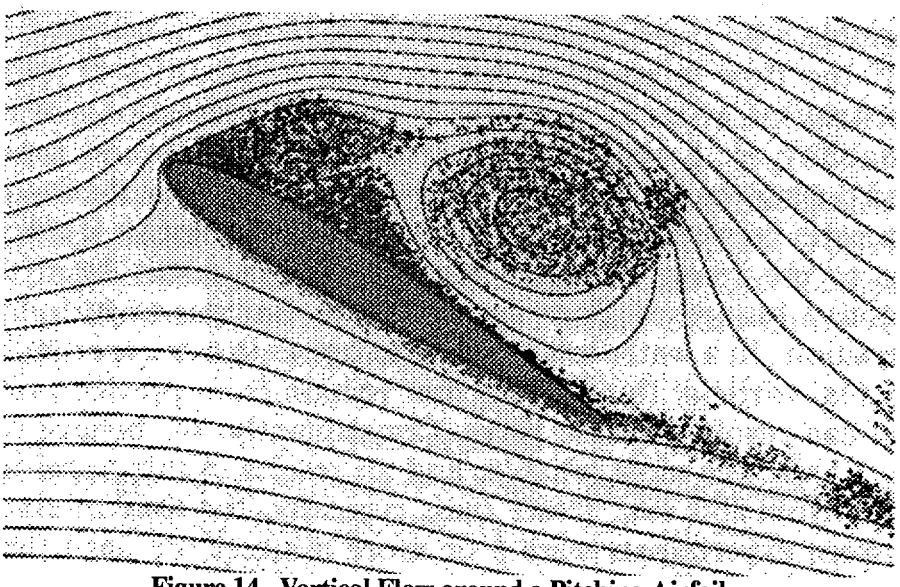


Figure 14. Vortical Flow around a Pitching Airfoil

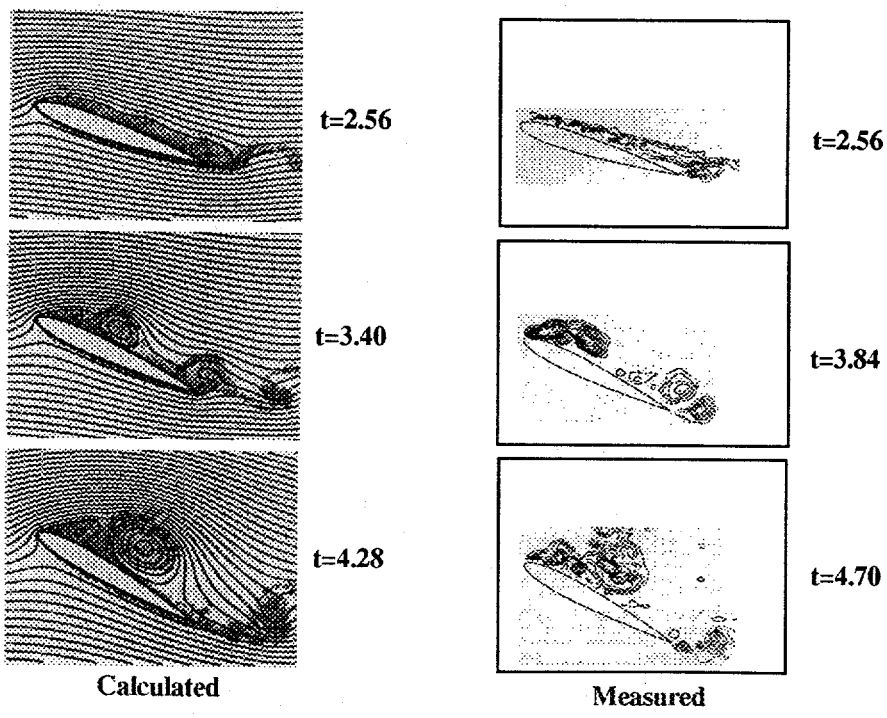


Figure 15. Calculated Streamlines and Measured Iso-Vorticity Lines

Reportedly, this simulation required up to 32,000 vortices at the later time steps. The CPU time for these later steps would have been on the order of several minutes per time step when running on a CYBER 205 had the fast solver not been used. According to [13], the computation using the fast solver should be between 13 and 28 times faster.

6. CONCLUSIONS

We have presented an overview of several fast multipole methods and several simulations which rely upon their use. We believe that these fast algorithms and their extensions represent a very important numerical tool for the following reasons:

1. There are a very large number of physical and numerical problems which may be cast in terms of Equations 2 and 4.
2. Many of these problems are intractable without the efficiencies afforded by these fast methods.
3. In the future, the solution of much larger "N-body" problems will be required in order to advance numerical simulation capabilities. The difference between an $O(N^2)$, $O(N\ln N)$, and $O(N)$ simulation will become even more important.

7. REFERENCES

1. Roach, G. F., *Green's Functions*, second edition, Cambridge University Press, p. 141, 1982.
2. Greengard, L., "Fast Algorithms for Classical Physics," *Science*, Vol. 265, pp. 909-914, August 1994.
3. Batchelor, G. K., *An Introduction to Fluid Dynamics*, Cambridge University Press, pp. 84-87, 1992.
4. Hassani, S., *Foundations of Mathematical Physics*, Allyn & Bacon, pp. 841-846, 1991.
5. Barnes, J. E. and Hut, P., "A Hierarchical $O(N\log N)$ Force Calculation Algorithm," *Nature*, Vol. 324, No. 4, pp. 446-449, December 1986.
6. Greengard, L. and Rokhlin, V. "A Fast Algorithm for Particle Simulations," *J. Compt. Phys.*, Vol. 73, pp. 325-348, 1987.
7. Strickland, J. H., Amos, D. E., "A Fast Solver for Systems of Axisymmetric Ring Vortices," Sandia National Laboratory Report SAND90-1925, 52 pages, September (1990). Also *AIAA Journal*, Vol. 30, No. 3, pp. 737-746, March 1992.
8. Zhao, F., An $O(N)$ Algorithm for Three-Dimensional N-Body Simulations, M. S. Thesis, Massachusetts Institute of Technology, Boston MA, 1987.
9. Greengard, L., "The Rapid Evaluation of Potential Fields in Particle Systems," Yale University Report YALEU/DCS/RR-533, April 1987.
10. Greengard, L. and Strain, J., "The Fast Gauss Transform," *SIAM J. Sci. Stat. Comput.*, Vol. 12, No. 1, pp79-94, January 1991.
11. Appel, A. W., "An efficient Program for Many-Body Simulation," *SIAM J. Sci. Stat. Comput.*, Vol. 6, No. 1, pp. 85-103, January 1985.
12. McMillan, S. L. W. and Aarseth, S. J., "An $O(N\ln N)$ Integration Scheme for Collisional Stellar Systems," *The Astrophysics Journal*, Vol. 414, pp. 200-212, September 1993.
13. Van Dommelen, L. and Rundensteiner, E., "Fast, Adaptive Summation of Point Forces in the Two-Dimensional Poisson Equation," *J. Compt. Phys.*, Vol. 83, pp. 126-147, 1989.
14. 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.
15. Draghicescu, C. I., "An Efficient Implementation of Particle Methods For The Incompressible Euler Equations," *SIAM J. Numer. Anal.*, Vol. 31, No. 4, pp. 10909-1108, August 1994.
16. Draghicescu, C. I and Draghicescu, M., "A Fast Algorithm for Vortex Blob Interactions," *J. Compt. Phys.*, Vol. 116, pp. 69-78, 1995.
17. Niedermeier, C. and Tavan, P., "A Structure Adapted Multipole Method for Electrostatic Interactions in Protein Dynamics," *J. Chem. Phys.*, Vol. 101, No. 1, pp. 734-740, July 1994.

18. Rokhlin, V., "Rapid Solution of Integral Equations of Classical Potential Theory," *J. Compt. Phys.*, Vol. 60, pp. 187-207, 1983.

19. Greengard, L. and Rokhlin, V., "On the Efficient Implementation of the Fast Multipole Algorithm," Yale University Report YALEU/DCS/RR-602, February 1988.

20. 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.

21. Schmidt, K. E. and Lee, M. A., "Implementing the Fast Multipole Method in Three Dimensions," *J. Stat. Phy.*, Vol. 63, Nos. 5/6, 1991.

22. Strickland, J. H., and Baty, R. S., "A Three-Dimensional Fast Solver for Arbitrary Vorton Distributions," Sandia National Laboratories Report SAND93-1641, May 1994.

23. Greengard, L. and Rokhlin, V., "On the Efficient Implementation of the Fast Multipole Algorithm," Yale University Report YALEU/DCS/RR-602, February 1988.

24. Aluru, S., "Distribution-Independent Hierarchical N-body Methods," Ph.D. Dissertation, Iowa State University, Ames, Iowa, 1994.

25. Belloch, G. and Narlikar, G., "A Comparison of two N-Body Algorithms," School of Computer Science, Carnegie Mellon University, October 1994.

26. Greengard, L. and Gropp, W. D., "A Parallel Version of The Fast Multipole Method," *Computers Math. Applic.*, Vol. 20, No. 7, pp. 63-71, 1990.

27. Singh, J. P., Holt, C., Hennessy, J. L., and Gupta, A., "A Parallel Adaptive Fast Multipole Method," IEEE Conference SUPERCOMPUTER'93.

28. Zhao, F., Johnsson, S. L., "The Parallel Multipole Method on the Connection Machine," *SIAM J. Sci. & Stat. Compt.*, Vol. 12, No. 6, 1991.

29. Salmon J., "Parallel $N\log N$ N-Body Algorithms and Applications to Astrophysics," IEEE Conference COMPCON Spring 91.

30. Grama, A. Y., Kumar, V., and Sameh, A., "Scalable Parallel Formulations for the Barnes-Hut Method for N-Body Simulations," Proceedings of Supercomputing'94.

31. Stiller, L., Daemen, L. L., and Gubernatis, J. E., "N-Body Simulations on Massively Parallel Architectures," *J. Compt. Phys.*, Vol. 115, 1994.

32. Sethian, J. A., Brunet, J. P., Greenberg, A., and Mesirov, J. P., "Vortex Methods and Massively Parallel Processors," *Lectures in Applied Mathematics*, Vol. 28, 1991.

33. Strickland, J. H., "A Prediction Method For Unsteady Axisymmetric Flow Over Parachutes," *AIAA Journal of Aircraft*, Vol. 31, No. 3, pp. 637-643, May-June 1994.

34. Higuchi, H., Balligand, H., and Strickland, J. H., "Numerical and Experimental Investigations of the Unsteady Flow Over a Disk," Forum on Vortex Methods for Engineering Applications, Sandia National Laboratories, Albuquerque NM, pp. 277-297, February 22-24, 1995.

35. 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.

36. Shih, C., Lourenco, Van Dommelen, L., and Krothapalli, "Unsteady Flow Past an Airfoil Pitching at a Constant Rate," *AIAA Journal*, Vol. 30, No. 5, pp. 1153-1161, May 1992.

Engineering Sciences Center, Sandia National Laboratories, Albuquerque, NM 87185-0836
E-mail address: jhstric@sandia.gov
E-mail address: rsbaty@sandia.gov

DISCLAIMER