

Fast multipole methods for three-dimensional N-body problems

By P. Koumoutsakos

1. Motivation and objectives

We are developing computational tools for the simulations of three-dimensional flows past bodies undergoing arbitrary motions. High resolution viscous vortex methods have been developed that allow for extended simulations of two-dimensional configurations such as vortex generators. Our objective is to extend this methodology to three dimensions and develop a robust computational scheme for the simulation of such flows.

A fundamental issue in the use of vortex methods is the ability of employing efficiently large numbers of computational elements to resolve the large range of scales that exist in complex flows. The traditional cost of the method scales as $\mathcal{O}(N^2)$ as the N computational elements/particles induce velocities at each other, making the method unacceptable for simulations involving more than a few tens of thousands of particles. In the last decade fast methods have been developed that have operation counts of $\mathcal{O}(N \log N)$ (Barnes and Hut, 1986) or $\mathcal{O}(N)$ (Greengard and Rokhlin, 1987) (referred to as BH and GR respectively) depending on the details of the algorithm. These methods are based on the observation that the effect of a cluster of particles at a certain distance may be approximated by a finite series expansion. In order to exploit this observation we need to decompose the element population spatially into clusters of particles and build a hierarchy of clusters (a tree data structure) - smaller neighboring clusters combine to form a cluster of the next size up in the hierarchy and so on. This hierarchy of clusters allows one to determine efficiently when the approximation is valid. This algorithm is an N-body solver that appears in many fields of engineering and science. Some examples of its diverse use are in astrophysics (Salmon and Warren 1992), molecular dynamics (Ding, Karasawa and Goddard 1992), micromagnetics (Yuan and Bertram 1992), boundary element simulations of electromagnetic problems (Kuster 1993), Nabors, Kim and White 1992), computer animation (Kuhn and Muller 1993), etc. More recently these N-body solvers have been implemented and applied in simulations involving vortex methods. Koumoutsakos and Leonard (1995) implemented the GR scheme in two dimensions for vector computer architectures allowing for simulations of bluff body flows using millions of particles. Winckelmans *et al.* (1995) presented three-dimensional, viscous simulations of interacting vortex rings, using vortons and an implementation of a BH scheme for parallel computer architectures. Bhatt *et al.* (1995) presented a vortex filament method to perform inviscid vortex ring interactions, with an alternative implementation of a BH scheme for a Connection Machine parallel computer architecture.

Historically the method of BH was first implemented for large scale astrophysical simulations. Several N-body algorithms are extensions of tree codes originally

developed for gravitational interactions. This has been motivated not only by the ease of implementation to a variety of applications, but also by the fact that in three dimensions the scheme of GR suffers from the large computational cost of $\mathcal{O}(p^4)$ associated with the translation of a p -th order multipole expansion. This cost has prohibited the use of large numbers of terms in the multipole expansions and lead to the general adoption of the BH method for N-body solvers. In a related effort to simplify the multipole expansion schemes, Anderson (1992) presented a computational scheme based on the Poisson Integral formula (hereafter referred to as PI). Greengard (1988) presented a strategy to reformulate the translation of the expansions as a convolution operators, thus enabling the use of FFT's and the reduction of the computational cost to $\mathcal{O}(p^2)$ operations. This strategy has been concisely summarized and extended in the work of Epton and Dembart (1995). Having overcome the $\mathcal{O}(p^4)$ difficulty, we believe that it is beneficial to follow the GR strategy as by using large number of expansions we avoid the costly pairwise interactions. The pairwise interactions determine the cost of the algorithm, and we try to minimize their number by using large numbers of expansions. Note that a large number of pairwise interactions may lead to algorithms of say $\mathcal{O}(N^{1.1})$ that would be inefficient for simulations involving hundreds of millions of particles (Bhatt *et al.*, 1995). The efficiency of the approach of using large numbers of terms in the expansions has been shown by our implementation of the method in two dimensions (Koumoutsakos 1996).

The objective of this report is to present a summary of the GR multipole expansion scheme with efficient $\mathcal{O}(p^2)$ multipole translations for general N-body problems. We discuss and compare the efficiency of computing the expansions based on the GR and the PI formulations. We document also the implementation of a suitable tree data structure for vector computer architectures by extending our previous work on the two-dimensional algorithm to strategies for a three-dimensional algorithm.

2. Accomplishments

We present a summary of the multipole expansions technique as presented by Greengard and Rohklin (1987) and Anderson (1992). We conduct some preliminary computations to assess the accuracy and efficiency of the two techniques and describe the fast multipole translation theory following Epton and Dembart (1995). Finally we describe our tree data structure and its implementation so as to take advantage of vector computer architectures.

2.1 The Greengard-Rohklin formulation

In order to introduce the subject of multipole expansions, we consider a unit source at a point $Q(\vec{x}')$ (Fig. 1). This unit source induces a potential at point $P(\vec{x})$ given by:

$$\Psi(P; Q) = \Psi(\vec{x}; \vec{x}') = \frac{1}{|\vec{x} - \vec{x}'|} \quad (1)$$

where the spherical coordinates of \vec{x} and \vec{x}' are given by (r, θ, ϕ) and (ρ, α, β) respectively. The distance between the two points is denoted by R and the angle

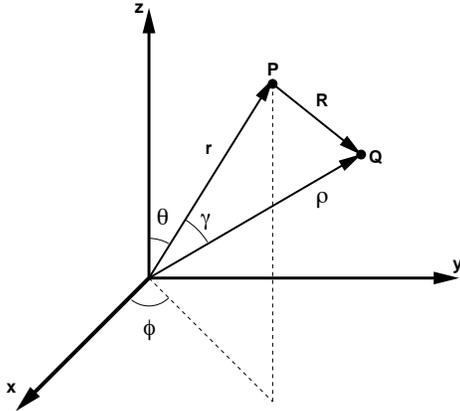


FIGURE 1. Coordinate definition for multipole expansion.

between vectors \vec{x} and \vec{x}' is denoted γ . If we define $\mu = \rho/r$ and $u = \cos\gamma$ then the potential at point Q may be expressed as:

$$\Psi(P; Q) = \frac{1}{R} = \frac{1}{r\sqrt{1 - 2u\mu + \mu^2}} \quad (2)$$

For $\mu = \rho/r < 1$, we use the generating formula for Legendre polynomials P_n so that the potential is expressed as:

$$\Psi(P; Q) = \sum_{n=0}^{\infty} \frac{\rho^n}{r^{n+1}} P_n(u) \quad (3)$$

This equation describes the *far field* potential at a point P, due to a charge of unit strength centered at Q. To obtain a computationally tractable formulation we proceed to express the Legendre functions in terms of *spherical harmonics*:

$$P_n(\cos \gamma) = \sum_{m=-n}^n Y_n^{-m}(\alpha, \beta) Y_n^m(\theta, \varphi) \quad (4)$$

and the *spherical harmonics* in terms of the *associated Legendre polynomials*:

$$Y_n^m(\theta, \varphi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos(\theta)) e^{im\varphi} \quad (5)$$

The following numerically stable formulas are used for calculations:

$$(n-m)P_n^m(u) = (2n-1)uP_{n-1}^m(u) - (n+m-1)P_{n-2}^m(u) \quad (6)$$

and

$$P_m^m(u) = (-1)^m (2m-1)! (1-u^2)^{\frac{m}{2}} \quad (7)$$

Summarizing then we see that the *far field representation* of the potential induced by a collection of N_v sources centered around Q with coordinates $(\rho_i, \alpha_i, \beta_i)$ is expressed as:

$$\Psi(P; \{q_i\}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{M_n^m}{r^{n+1}} Y_n^m(\theta, \varphi) \quad (8)$$

where:

$$M_n^m = \sum_{i=1}^{N_v} q_i \rho_i^n Y_n^{-m}(\alpha_i, \beta_i) \quad (9)$$

Note that if we wish to form a *local expansion* of the field around the origin then we express $1/R$ as:

$$\begin{aligned} \frac{1}{R} &= \frac{1}{\rho \sqrt{1 - 2\frac{r}{\rho}u + \left(\frac{r}{\rho}\right)^2}} \\ &= \sum_{n=0}^{\infty} \frac{r^n}{\rho^{n+1}} P_n(u) \end{aligned} \quad (10)$$

2.2 Translation of multipole expansions

Once the multipole expansions due to a collection of sources have been computed, one is usually interested in computing the far-field coefficients of the same collection expanded about some other point, say S, so that the potential would be represented as:

$$\Psi(S; P) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{L_n^m}{\sigma^{n+1}} Y_n^m(\Theta, \Phi) \quad (11)$$

where (σ, Θ, Φ) are the spherical coordinates of the distance between points P and S.

This defines the translation problem for multipole expansions for fast multipole methods. Following Greengard (1988) and Epton and Dembart (1994) we present a concise summary of the theory underlying the translation of multipole expansions. We make use of the following definitions of harmonic outer functions O_n^m and inner functions I_n^m :

$$O_n^m(\vec{x}) = O_n^m(r, \theta, \varphi) = \frac{(-1)^n i^{|m|}}{A_n^m} \frac{Y_n^m(\theta, \varphi)}{r^{n+1}} \quad (12)$$

$$I_n^m(\vec{x}) = I_n^m(r, \theta, \varphi) = i^{-|m|} A_n^m r^n Y_n^m(\theta, \varphi) \quad (13)$$

More specifically for $|\vec{x}| > |\vec{x}'|$ we obtain:

$$O_n^m(\vec{x} - \vec{x}') = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} (-1)^{n'} I_{n'}^{-m'}(\vec{x}') O_{n+n'}^{m+m'}(\vec{x}) \quad (14)$$

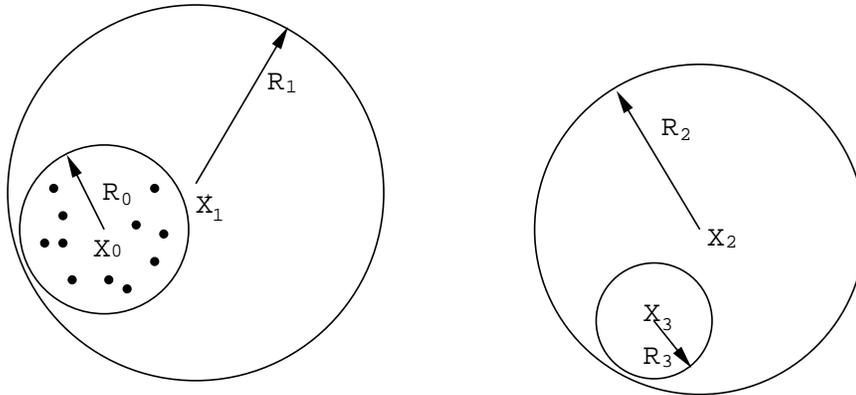


FIGURE 2. Sketch for the translation of the multipole expansions.

and for the inner expansions we get that:

$$I_n^m(\vec{x} - \vec{x}') = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} (-1)^{n'} I_{n'}^{m'}(\vec{x}') I_{n-n'}^{m-m'}(\vec{x}) \quad (15)$$

so now we may express the equation for the potential induced at point \vec{x} from a unit source at point \vec{x}' as:

$$\frac{1}{|\vec{x} - \vec{x}'|} = O_0^0(\vec{x} - \vec{x}')$$

In order to further exhibit the formulation of these translation operators we consider the configuration shown in Fig. 2. We wish to determine the potential induced by a collection of sources within a sphere centered at \vec{x}_0 and having radius R_0 (denoted as $s(\vec{x}_0, R_0)$) to a collection of points/sources within a sphere $s(\vec{x}_3, R_3)$. This is achieved in the following steps:

(i) We compute a set of multipole expansion coefficients C_n^m (using Eq. 8) for the *far-field representation* of a set of sources distributed within $s(\vec{x}_0, R_0)$ Then the *far field* representation of the field induced by this cluster of particles at a location \vec{x} is given by:

$$\Psi(\vec{x}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n C_n^m O_n^{-m}(\vec{x} - \vec{x}_0) \quad (16)$$

(ii) We translate the *Outer* expansion about \vec{x}_0 to an *Outer* expansion about \vec{x}_1 (*child to parent*):

$$\psi(\vec{x}) = \sum_{l=0}^{\infty} \sum_{j=-l}^l D_l^j O_l^{-j}(\vec{x} - \vec{x}_1) \quad (17)$$

where

$$D_l^j = \sum_{n=0}^l \sum_{m=-n}^n C_n^m I_{l-n}^{j-m}(\vec{x}_1 - \vec{x}_0) \quad (18)$$

(iii) We translate the *Outer* expansion about \vec{x}_1 to an *Inner* expansion about \vec{x}_2 (*box - box* interaction):

$$\psi(\vec{x}) = \sum_{l=0}^{\infty} \sum_{j=-l}^l E_l^j I_l^j(\vec{x} - \vec{x}_2) \quad (19)$$

where

$$E_l^j = \sum_{n=0}^l \sum_{m=-n}^n D_n^m O_{l+n}^{-j-m}(\vec{x}_2 - \vec{x}_1) \quad (20)$$

(iv) We translate the *Inner* expansion about \vec{x}_2 to an *Inner* expansion about \vec{x}_3 (*parent to child*):

$$\psi(\vec{x}) = \sum_{l=0}^{\infty} \sum_{j=-l}^l F_l^j I_l^j(\vec{x} - \vec{x}_3) \quad (21)$$

where

$$F_l^j = \sum_{n=0}^l \sum_{m=-n}^n E_n^m I_{n-1}^{m-j}(\vec{x}_3 - \vec{x}_2) \quad (22)$$

(v) Once the coefficients of the multipole expansions have been computed in the sphere $s(\vec{x}_3, R_3)$ we perform a *local expansion* using Eq. 10 to compute the potential at the individual points.

The above representations for the translation operations of spherical harmonics reveal that they require the evaluation of double summations that are essentially **convolution** operations over the coefficients of the expansions and can essentially be computed using 2-D FFT's.

2.3 The Poisson integral method

In order to approximate the potential due to a collection of particles Anderson (1992) proposed an alternative simplified technique. This technique is based on the observation that a harmonic function (Ψ) external to a sphere of radius R may be described using its boundary values $g(R\vec{s})$ on the surface of the sphere. So given a point \vec{x} and \vec{x}_p a point on the *unit* sphere that points in the direction of \vec{x} then:

$$\Psi(\vec{x}) = \frac{1}{4\pi} \int_{S^2} \left[\sum_{n=0}^{\infty} (2n+1) \left(\frac{R}{r}\right)^{n+1} P_n(\vec{s} \cdot \vec{x}_p) \right] g(R\vec{s}) \, ds \quad (23)$$

where S^2 denotes the surface of the unit sphere and P_n is the n -th order Legendre polynomial. We use a quadrature formula to integrate the function on the surface of the sphere with K integration points \vec{s}_i and weights w_i to obtain an approximation of the form:

$$\Psi(\vec{x}) \approx \frac{1}{4\pi} \sum_{i=1}^K \left[\sum_{n=0}^M (2n+1) \left(\frac{R}{r}\right)^{n+1} P_n(\vec{s}_i \cdot \vec{x}_p) \right] g(R\vec{s}_i) w_i \quad (24)$$

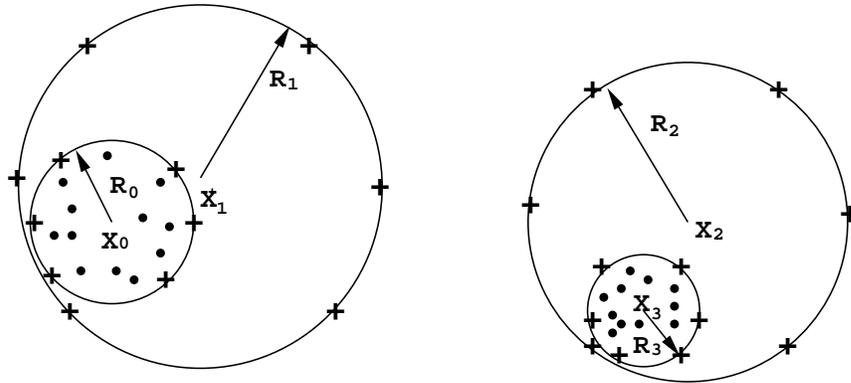


FIGURE 3. Sketch for the translation of the multipole expansions using the PI technique.

In order to compute the far field multipole expansion of a cluster of particles based on this formulation, the function $g(R\vec{s})$ is determined on certain quadrature points on the surface of a sphere using the direct interactions of the potential induced by the sources onto the evaluation points. Subsequently these coefficients are used in Eq. 24 to compute the potential induced at distances sufficiently large compared to the radius of the cluster. In order to translate the expansions the above formula may be used again by considering the coefficients of $g(R\vec{s}_i)$ on the inner sphere to be sources themselves. The method is completed by observing that a local expansion approximation may be constructed using the following formula. Note that the expansions are in terms of r/R in this formula.

$$\Psi(\vec{x}) \approx \frac{1}{4\pi} \sum_{i=1}^K \left[\sum_{n=0}^M (2n+1) \left(\frac{r}{R}\right)^n P_n(\vec{s}_i \cdot \vec{x}_p) \right] g(R\vec{s}_i) w_i \quad (25)$$

Anderson (1992) showed that approximations with $M = 2p + 1$ terms may be compared with multipole schemes that have p terms retained in the expansions. The strength of this method relies on its simplicity and its easy extension from two to three dimensions. This is exhibited by considering the implementation of this technique in the context of an $\mathcal{O}(N)$ algorithm for the computation of the potential field induced by a set of particles within a sphere $s(\vec{x}_0, R_0)$ to a cluster of points in $s(\vec{x}_0, R_0)$ (Fig. 4). This interaction is performed in the following steps:

(i) The potential induced by the particles (denoted by dots) is computed on quadrature points properly selected on the surface of the enclosing sphere (denoted by +), thus constructing the function $g(R_0\vec{s}_i)$

(ii) The potential induced by the quadrature points on $s(\vec{x}_0, R_0)$ is computed on the quadrature points of $s(\vec{x}_1, R_1)$ using Eq. 24.

(iii) The coefficients $g(R_1\vec{s}_i)$ are considered to be sources themselves so that the coefficients $g(R_2\vec{s}_i)$ are computed using Eq. 24.

(iv) The coefficients $g(R_3\vec{s}_i)$ are computed subsequently by performing a *local expansion* from the quadrature points on sphere $s(\vec{x}_2, R_2)$ to the quadrature points on the sphere $s(\vec{x}_2, R_2)$ using Eq. 25.

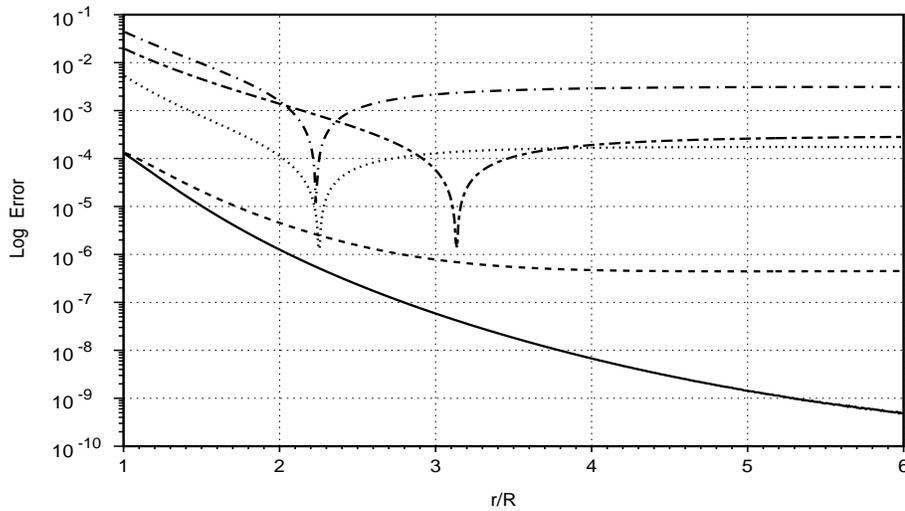


FIGURE 4. Relative error of the GR and PI multipole expansion schemes. Symbols: (GR):—— : $P = 8$; (PI): - - - - : $m = 14$, : $m = 9$, - - - - : $m = 7$, - · - · : $m = 5$.

(v) By using Eq. 25 the potential on the particles inside the sphere $s(\vec{x}_3, R_3)$ is computed using the coefficients $g(R_3 \vec{s}_i)$.

The simplicity of the formulas implemented in this technique make it an attractive alternative to the multipole expansion method of GR. We consider below a comparison of the two methods in terms of their accuracy and computational cost.

2.4 Computational cost

The computational cost, associated with the multipole expansions of the GR scheme, scales as $\mathcal{O}(p^2 N)$ for the multipole-particle operations and as $\mathcal{O}(\log N p^2)$ for the multipole translation operations using the convolution formulation discussed above. For the Poisson Integral formulation, assuming K integration points and M terms in the Poisson kernel, the cost scales as $\mathcal{O}(K \times M \times N)$ for the particle-multipole interactions and as $\mathcal{O}(K \times K \times M)$ for the multipole translation.

Both algorithms have error terms that scale as $(R/r)^{L+1}$ where L corresponds to p terms in the multipole method and to $M + 1$ for the Poisson Integral scheme. The number of the required integration points (K) corresponding to 5th, 7th, 9th, and 14th order quadrature formulas require 12, 24, 32, and 72 points respectively or approximately $K \approx 2m^2/5$ points for an m -th order integration formula. So in order to achieve the same accuracy with the PI method as with the multipole scheme we require that $m \approx M \approx P$. This then implies a computational cost of $\mathcal{O}(p^3 N)$ for the particle-multipole interactions while it implies a $\mathcal{O}(\log N p^5)$ cost associated with the multipole translations.

Although such estimates depend on the particular implementation of the method, it is evident that for the same order of accuracy the multipole method with fast translation operations would lead to much faster computations, especially for large

expansion orders. Moreover, in the particular implementation of the methods it was easier to unroll the computational loops involving the particle multipole interactions for the GR scheme than the respective operations in the PI scheme. In Fig. 4 we present also the relative error as computed by the two methods using $P = 8$ and different values of m in the Poisson Integral method.

In the following table we present some representative timing results for the construction of the multipole expansions as well as for the particle-multipole interactions. A number of N particles was distributed randomly inside a cube of side 2 and the potential was evaluated on L points uniformly distributed along a line extending from the center of the cube and along one of its sides. The first line corresponds to the CPU time required for the construction of the multipole expansions while the second line corresponds to the CPU time required to perform the particle-multipole expansions. It is observed then that using the GR multipole expansion scheme an order of magnitude faster calculations are achieved for the same order of accuracy. This dictates the use of the GR technique for computations using multipole expansions.

2.5 Tree data structures

A key issue in the implementation of fast multipole methods is the associated data structure and the computer platform. The present methodology has been successfully implemented for vector computer architectures and several of its features could be carried over to parallel platforms involving large numbers of vector processors. The scheme of GR relies in a predetermined tree data structure and a large number of terms to be kept in the expansions while the scheme of BH determines the interaction list while traversing the tree data structure, and a tradeoff is made between the number of terms kept in the expansions (usually two to four) and the distance at which the expansions are favored over direct interactions.

In order to exploit the observation that the effect of a cluster of particles at a certain distance may be approximated by a finite sum of series expansions using the equations described above, we need to organize the particles in a hierarchy of clusters. This hierarchy of clusters allows one to efficiently determine when the approximation is valid. In order to establish the particle clusters one may resort to a tree building algorithm.

The straightforward method of computing the pairwise interaction of all particles requires $\mathcal{O}(N^2)$ operations for N vortex elements. In the last decade fast methods have been developed that have operation counts of $\mathcal{O}(N \log N)$ (Barnes and Hut, 1986) or $\mathcal{O}(N)$ (Greengard and Rokhlin, 1987) depending on the details of the algorithm. The basic idea of these methods is to decompose the element population spatially into clusters of particles and build a hierarchy of clusters or a tree - smaller neighboring clusters combine to form a cluster of the next size up in the hierarchy and so on.

The contribution of a cluster of particles to the potential of a given vortex can then be computed to desired accuracy if the particle is sufficiently far from the cluster in proportion to the size of the cluster and a sufficiently large number of terms in the multipole expansion is taken. This is the essence of the ‘particle-box’

N	L	m = 5	m = 7	m = 9	m = 14	Multipoles p = 8
10^5	10^5	0.0708	0.1397	0.1863	0.4191	0.1023
		0.5268	0.5844	0.6395	1.0870	0.0959
10^6	10^6	0.7111	1.4219	1.8968	4.2680	1.0251
		5.2724	5.8516	6.4145	10.8925	0.9585

FIGURE 5. Computational cost of the FMM and Poisson integral method.

method, requiring $\mathcal{O}(N \log N)$ operations. One then tries to minimize the work required by maximizing the size of the cluster used while keeping the number of terms in the expansion within a reasonable limit and maintaining a certain degree of accuracy.

The ‘box-box’ scheme goes one step further as it accounts for box-box interactions as well. These interactions are in the form of shifting the expansions of a certain cluster with the desired accuracy to the center of another cluster. Then those expansions are used to determine the velocities of the particles in the second cluster. This has as an effect the minimization of the tree traversals for the individual particles requiring only $\mathcal{O}(N)$ operations.

In order to construct the tree data structure, the three-dimensional space is considered to be a cube enclosing all computational elements. We apply the operation of continuously subdividing a cube into eight identical cubes until each cube has only a certain maximum number of particles in it or the maximum allowable level of subdivisions has been reached.

The hierarchy of boxes defines a tree data structure which is common for both algorithms. The tree construction proceeds level by level starting at the finest level of the particles and proceeding upwards to coarser box levels. Due to the simplicity of the geometry of the computational domain, the addressing of the elements of the data structure is facilitated significantly. As the construction proceeds pointers are assigned to the boxes so that there is direct addressing of the first and last particle index in them as well as direct access to their children and parents. This facilitates the computation of the expansion coefficients of the children from the expansions of the parents for the *BB* algorithm and the expansions of the parents from those of the children for the *PB*

The data structure is used to determine when the expansions are to be used and when pairwise interactions have to be calculated. It helps in communicating to the computer the geometric distribution of the particles in the computational domain. The particles reside at the finest level of the structure. Clusters of particles form the interior nodes of the tree and hierarchical relations are established. The data

structure adds to the otherwise minimal memory requirements of the vortex method.

The tree has to be reconstructed at every step as the particles change positions in the domain. There are several ways that nearby particles could be clustered together and some of the decisions to be made are:

The center of expansions. In the present study the geometric center of the cells is used as it facilitates the addressing of the data structure.

The cluster size. In the present algorithms we follow a hybrid strategy as we keep at least L_{\min} particles per box until we reach a predetermined finest level of boxes. The number L_{\min} may be chosen by the user depending on the particle population and configuration so as to achieve an optimal computational cost.

Addressing the clusters. As particles are usually associated with a certain box, it is efficient to sort the particle locations in the memory so that *particles that belong to the same box occupy adjacent locations in the memory devoted to the particle arrays*. Such memory allocation enhances the vectorization significantly as very often we loop over particles of the same box (e.g., to construct the expansions at the finest level, or to compute interactions) and the loops have an optimum stride of one.

Description of the Algorithms:

In both algorithms, described herein, we may distinguish three stages:

- Building the data structure (tree).
- Establishing the interaction lists (by non-recursively descending the tree).
- Pairwise interactions for all particles in the domain.

The building of the data structure is common for both algorithms, but they differ in the tree descent and the pairwise interactions. Care has been exercised at all stages to maximize vectorization. In our respective two-dimensional implementation, the building of the data structure consumes about 5 – 7% of the time, the descent consumes another 5 – 10% so that the largest amount is spent in computing the pairwise interactions.

2.6 The particle-box algorithm

Step 1. Building the data structure (tree)

Step 1a : For each of the cubes at each level that are not further subdivided, we compute the p -terms of the multipole expansions. These expansions are used to describe the influence of the particles at locations that are well separated from their cluster.

Step 1b : The expansions of all parent boxes are constructed by shifting the expansion coefficients of their children. The tree is traversed upwards in this stage. Rather than constructing the expansions of all the members of a family (that is traverse each branch until the root is reached) we *construct the expansions of all parent boxes at each level simultaneously*. This enables long loops over the parent boxes at each level. Care is taken so that the procedure is fully vectorized by taking advantage of the regularity of the data structure and the addressing of the boxes in the memory. Moreover, the regularity of the data structure allows us to precompute many coefficients that are necessary for the expansions. Straightforward implementation of these translations leads to computational cost of $\mathcal{O}(p^4)$. This has been

the major reason that most implementations of the algorithm have employed only up to $p=3$ terms in the multipole expansions. However, such an approach results in large numbers of particle-particle interactions and hence a large computational cost. We implement the technique proposed by Greengard (1988) (see above) to reduce the computational cost of this translation to $\mathcal{O}(p^2)$ operations, by observing that this translation amounts to a convolution, and employ FFT's.

Step 2. Establishing of interaction lists

In the present algorithm a **breadth-first** search is performed at each level to establish the interaction lists of each particle (cell). This search is facilitated by the regularity of the data structure and the identification arrays of the cells in the tree. At each level interaction lists are established for the particles (cells) by looping across the cells of a certain level.

Note that this depth-first search for interaction lists is further facilitated by the fact that every particle belongs to a childless box. It is easy then to observe that *all particles in the same box share the same interaction list, which is comprised of members of the tree that belong to coarser levels*. In this way the tree is traversed upwards for all particles in a childless box together and downwards separately for each particle. It is evident that this procedure is more efficient for uniformly clustered configurations of particles because there would be more particles that belong to childless boxes at the finest level.

Step 3. Computation of the interactions.

Once the interaction lists have been established, the velocities of the particles are computed by looping over the elements of the lists. For particles that have the same boxes in their interaction list, this is performed simultaneously so that memory referencing is minimized. Moreover by systematically traversing the tree, the particle-particle interactions are made symmetric so that the cost of this computation is halved. The *cost* of this step is $\mathcal{O}(Np^2)$.

2.7 The box-box algorithm.

This scheme is similar to the *PB* scheme except that here every node of the tree assumes the role of a particle. In other words, interactions are not limited to particle-particle and particle-box but interactions between boxes are considered as well. Those interactions are in the form of shifting the expansion coefficients of one box into another and the interaction lists are established with respect to the locations of every node of the tree.

The scheme distinguishes five categories of interacting elements of the tree with respect to a cell denoted by \mathbf{c} .

- **List 1:** All childless boxes neighboring \mathbf{c} .
- **List 2:** Children of colleagues (boxes of the same size) of \mathbf{c} 's parents that are well separated from \mathbf{c} . All such boxes belong to the same level with \mathbf{c} .
- **List 3:** Descendants (not only children) of \mathbf{c} 's colleagues, whose parents are adjacent to \mathbf{c} but are not adjacent to \mathbf{c} themselves. All such boxes belong to finer levels.
- **List 4:** All boxes such that box \mathbf{c} belongs to *their* List 3. All such boxes are childless and belong to coarser levels.

•**List 5:** All boxes well separated from \mathbf{c} 's parents. Boxes in this category do not interact directly with the cell \mathbf{c} .

If the cell \mathbf{c} is childless it may have interacting pairs that belong to all four lists. However if it is a parent it is associated with boxes that belong to lists 2 and 4 as described above. These observations are directly applied in our algorithm and we may distinguish again the following 4 steps.

Step 1: Building the data structure.

This procedure is the same as for the *PB* scheme. This fact enables us to compare directly the two algorithms and assess their efficiency.

Step 2: Construction of interaction lists.

To establish the interaction lists we proceed again level by level, starting at the coarsest level. For each level we distinguish childless and parent boxes. In establishing lists 1 and 3 we need only loop over childless boxes whereas to establish lists 2 and 4 we loop over all cells that are active in a certain level.

Step 2a: Here we establish lists 1 and 3. We start at the level of the parents of box \mathbf{c} and we proceed level by level examining again breadth first, until we reach the finest level of the structure (the particles). The elements of lists 1 are basically the particles and account for the particle-particle interactions. Care is exercised so that this computation is symmetric, and we need to traverse the tree *downwards only* thus minimizing the search cost. The elements of List 3 are the boxes and are accountable for the particle-box interactions in this scheme.

Step 2b: Here we establish interaction lists 2 and 4 for all boxes in the hierarchy. We start at the coarsest possible level and proceed downward until reaching the level of box \mathbf{c} to establish the interaction lists. To do so for a certain box we start by examining boxes that are not well separated from their parents (otherwise they would have been dealt with at the coarser level). Subsequently the children of those boxes are examined to establish interaction lists.

Step 3: Computations of the interactions

In this scheme we consider three kinds of interactions: the box-box, particle-box, and particle-particle interactions. The latter two categories were discussed in the previous section. For the box-box interactions once the respective interaction lists have been established (with members of lists 2 and 4), we need to transfer those expansions down to the ones of the children and add them to the existing ones. This procedure is vectorized by looping over the number of boxes at each level. The use of pointers to access the children of each box enhances this vectorization. Note that an arbitrarily high number of expansions can be calculated efficiently by unrolling the loop over the number of expansions into the previously mentioned loop.

3. Conclusions and future work

We are in the process of developing a three-dimensional N -body problem solver with the objective of applying it to the solution of engineering problems involving boundary element methods. This solver would be mainly used for the implementation of vortex methods for three-dimensional simulations involving unsteady flows past complex moving configurations. Furthermore, of particular interest is the

application of the code to simulations of rarefied flows using molecular dynamics methods. The code is envisioned as a computational tool that would easily enable the transition between continuum flows and flows using molecular dynamics.

REFERENCES

- ANDERSON, C. R. 1992 An Implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comp.* **13**, 923-947.
- BARNES, J. & HUT, P. 1986 A Hierarchical $O(N \log N)$ force-calculation algorithm *Nature*. *324*. **446-449**.
- BHATT, S., LIU, P., FERNANDEZ, V. & ZABUSKY, N. 1995 Tree codes for vortex dynamics: Application of a programming framework. *Workshop on Solving Irregular Problems on Parallel Machines*. Santa Barbara, April 25-28.
- DING, H-Q., KARASAWA, N. & GODDARD, W. A. III 1992 Atomic level simulations on a million particles: The cell multipole method for Coulomb and London nonbond interactions. *J. Chem. Phys.* **97**, 4309-4315.
- EPTON, M. A. & DEMBART, B. 1995 Multipole Translation Theory for the Three-Dimensional Laplace and Helmholtz Equations. *SIAM J. Sci. Stat. Comp.* **16**, 865-897.
- GREENGARD, L. 1988 On the efficient implementation of fast multipole algorithms. *Dept. of Computer Science Report 602*. Yale University.
- GREENGARD, L. & ROHKLIN, V. 1987 Rapid Evaluation of Potential Fields in Three Dimensions. *Dept. of Computer Science Report 515*. Yale University.
- KOUMOUTSAKOS, P. & LEONARD, A. 1995 High Resolution simulations of the flow past an impulsively started cylinder. *J. Fluid Mech.* **96**, 1-32.
- KUHN, V., & MULLER, W. 1993 Advanced Object-oriented Methods and Concepts for Simulation of Multi-body Systems. *J. Visual. & Comp. Anim.* **4**, 95-111.
- KUSTER, N. 1993 Multiple Multipole Method for Simulating EM Problems Involving Biological Bodies. *IEEE Trans. Biomed. Engr.* **40**, 611-620.
- NABORS, K., KIM, S. & WHITE, J. 1992 Fast Capacitance Extraction of General Three-Dimensional Structures. *IEEE Trans. Microwave Theory and Tech.* **40**, 1496-1506.
- SALMON, J. K. & WARREN, M. 1994 Skeletons from the treecode closet. *J. Comp. Phys.* **111**, 136-155.
- YUAN, S. & BERTRAM, N. 1992 Fast Algorithms for Micromagnetics. *IEEE Trans. Magnetism.* **28**, 2031-2036.
- WINCKELMANS, G. S., SALMON, J. K., WARREN, M. S., LEONARD, A. & JODOIN, B. 1995 Application of fast parallel and sequential tree codes to computing three-dimensional flows with the vortex element and boundary elements method. *Proc. of 2nd International Workshop on Vortex Flows and Related Numerical Methods*. Montreal, August 20-24.