Analyzing the Error Bounds of Multipole-Based Treecodes

Vivek Sarin, *Ananth Grama*, *and* *Ahmed Sameh*
Department of Computer Sciences
Purdue University
W. Lafayette, IN 47906
Email: Sarin, Grama, Sameh
Web page: Sarin, Grama, Sameh

Abstract:

The problem of evaluating the potential due to a set of particles is an important and time-consuming one. The development of fast treecodes such as the Barnes-Hut and Fast Multipole Methods for *n*-body systems has enabled large scale simulations in astrophysics [9, 10, 13] and molecular dynamics [1]. Coupled with efficient parallel processing, these treecodes are capable of yielding several orders of magnitude improvement in performance [6, 14, 15]. In addition, treecodes have applications in the solution of dense linear systems arising from boundary element methods [3, 4, 5, 11, 12]. Using a *p*-term multipole expansion, the FMM reduces the complexity of a single timestep from $O(n^2)$ to $O(p^2 n)$ and Barnes-Hut method reduces it to $O(p^2 n \log n)$ for a uniform distribution. In this paper, we analyze the approximations introduced by these methods. We describe an algorithm that reduces the error significantly by selecting the multipole degree appropriately for different clusters. Furthermore, we show that for practical problem sizes, this increases the computational complexity marginally. We support our theoretical result with experiments in the context of particle simulations as well as boundary element methods. Our POSIX threads-based treecode yields excellent speedups on a 32 processor SGI Origin 2000, even for relatively small problems.

Keywords:

Barnes-Hut, fast multipole method, integral equations, boundary elements, parallel, iterative methods.

Acknowledgements

This research has been supported by NSF Grants EIA-9805743, EIA-9876014, and ESC-95-27123. Computational resources for the project were provided by the National Computational Science Alliance and the Minnesota Supercomputing Institute.

Global Error Estimate for Barnes-Hut Method

The potential due to a set of charges located within a sphere of radius r_s at an observation point at distance *r* from the origin can be expressed as a multipole series. The error in a truncated multipole series of degree *p* was first derived by Greengard and Rokhlin [7, 8]. The following theorem from [7] describes the multipole expansion and the associated error.

Theorem 1 (Multipole Error) *Suppose that k charges of strengths* $\{q_j,\ j=1,\ldots,k\}$ *are located at the points* $\{P_j=(\rho_j,\theta_j,\psi_j)\}$ *(in spherical coordinates), with* $|\rho_j| < r_a$. *Then for any point* $P=(r,\theta,\psi)\epsilon R^3$ *with* $r > r_a$, *the potential* $\phi(P)$ *is given by*

$$\phi(P)=\sum_{n=0}^{\infty}\sum_{m=-n}^{m=n}\frac{M_n^m}{r^{n+1}}\cdot Y_n^m(\theta,\psi),$$

where

$$M_n^m=\sum_{j=1}^{k}q_j\cdot\rho_j^n\cdot Y_n^{-m}(\theta_j,\psi_j),$$

and

$$Y_n^m(\theta,\psi)=\sqrt{\frac{(n-|m|)!}{(n+|m|)!}}\cdot P_n^{|m|}(\cos\theta)e^{im\phi},$$

in which $P_n^{|m|}(\cos\theta)$ *are the associated Legendre functions. Furthermore, for any* $p\geq 1$, *the error in the truncated multipole series of degree p is given by*

$$\epsilon=|\phi(P)-\sum_{n=0}^{p}\sum_{m=-n}^{m=n}\frac{M_n^m}{r^{n+1}}\cdot Y_n^m(\theta,\psi)|,$$

and is bounded by

$$\epsilon\leq\frac{A}{r-r_a}(\frac{r_a}{r})^{p+1}, \qquad (1)$$

where $A=\sum_{j=1}^{k}|q_j|$.

Proof

See [7, page 54,].

In Barnes-Hut method, the potential at a point is computed as a sum of the contributing potentials from clusters of particles. An interaction with a cluster is computed only if the point is *well-separated* from the cluster. This is enforced using a multipole acceptance criterion such as the α-criterion which requires that the ratio of the distance between the point and the center of mass of the cluster and the dimension of the box enclosing the cluster be greater than $1/\alpha$ for a constant α less than unity. Theorem 1 can be used to estimate the error in Barnes-Hut method.

Theorem 2 (Barnes-Hut Multipole Error) *Suppose that k charges of strengths* $\{q_j, j=1,...,k\}$ *are located within a sphere of radius* r_a. *Then, for the Barnes-Hut method with* α*-criterion, the error in the truncated multipole series approximation of the potential at a distance r is bounded by*

$$\epsilon\leq\frac{A}{r_a}\cdot\frac{\alpha^{p+2}}{1-\alpha},$$

where $p>1$ *and* $A=\sum_{j=1}^{k}|q_j|$.

Proof

The α -criterion of the Barnes-Hut method ensures that $r/r_a \geq 1/\alpha > 1$. Subtracting unity and inverting, we obtain the following relation

$$\frac{r_a}{r - r_a} \leq \frac{\alpha}{1 - \alpha}.$$

From Thm. 1,

$$\epsilon \leq \frac{A}{r_a} \cdot \frac{r_a}{r - r_a} \cdot \left(\frac{r_a}{r}\right)^{p+1} \leq \frac{A}{r_a} \cdot \frac{\alpha}{1 - \alpha} \cdot \alpha^{p+1},$$

which proves the theorem.

This theorem illustrates the main problem with aggregate error in Barnes-Hut method. The error grows linearly with the net charge of the particle clusters. Moreover, the size of the largest cluster with which an interaction is computed can be shown to be within constant factor of the total simulation domain. Thus, the aggregate error can be large, and even unbounded for unstructured distributions. For instance, in applications such as protein simulations, the charge density is largely uniform across the domain of simulation; therefore, the overall error in the Barnes-Hut method grows linearly with the magnitude of charge in the system. In general, for large simulation domains, the aggregate error may become unacceptable.

Fortunately, this theorem also suggests an alternative strategy to control the error. By increasing the polynomial degree *p* for clusters with increased net charge *A*, the error in each interaction can be restricted to a constant value. The error in potential at each point would then be proportional to the total number of interactions. The multipole acceptance criterion along with the hierarchical decomposition of the domain can be used to establish the following:

- the number of interactions with clusters of a particular size are bounded by constant, and
- the number of distinct sizes of clusters is equal to the height of the decomposition tree.

For structured distributions with uniform charge density, this translates to $O(\log n)$ aggregate error.

We now prove these assertions and outline an improved algorithm for selecting the polynomial degree *p* to restrict the error.

In order to bound the number of interactions for fixed size clusters, we first establish limits on the ratio r_a/r .

Lemma 1 *In the Barnes-Hut method, the ratio r_a/r for particle-cluster interactions is bounded as follows:*

$$\alpha' < \frac{r_a}{r} < \alpha,$$

where α' and α are constants.

Proof

The interaction of particle *s* with box *b* indicates that *s* could not interact with its parent box *B*

based on the α -criterion (see Fig. 1). Therefore,

$$r \geq r_0, \text{ and } R < R_0,$$

where $r_0 = r_s/\alpha$ and $R_0 = 2r_s/\alpha$.

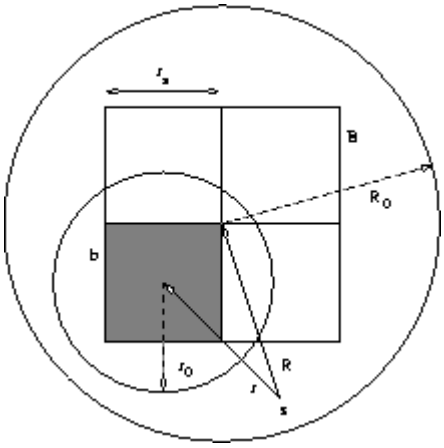

Figure 1: Establishing bounds on r_s/r in Barnes-Hut method.

Using the triangle inequality $R + r_s/\sqrt{2} \geq r$, it can be shown that

$$\left(\frac{2}{\alpha} + \frac{1}{\sqrt{2}}\right)^{-1} \leq \frac{r_s}{r} \leq \alpha,$$

which completes the proof.

As α is reduced, this bound tends to $\alpha/2 \leq r_s/r \leq \alpha$, indicating a tight bound. It is now easy to show that the number of interactions with a box of size r_s is bounded by a constant.

Lemma 2 *In Barnes-Hut method, a particle interacts with a bounded number of boxes of a given size.*

Proof

Lemma 1 shows that $\alpha' \leq r_s/r \leq \alpha$; therefore, the centers of all boxes of size r_s lie within an annular region defined by the following relation:

$$\frac{r_s}{\alpha} \leq r \leq \frac{r_s}{\alpha'},$$

and the boxes themselves lie completely within the annular region:

$$\frac{r_s}{\alpha} - \frac{r_s}{\sqrt{2}} \leq r \leq \frac{r_s}{\alpha'} + \frac{r_s}{\sqrt{2}}.$$

The ratio of the volume of this annular region and the volume of a single box gives an upper bound on the number of boxes of size r_s . For a three dimensional problem,

$$n_{max} \leq \frac{4\pi}{3}\left[\left(\frac{1}{\alpha}+\frac{1}{\sqrt{2}}\right)^3-\left(\frac{1}{\alpha}-\frac{1}{\sqrt{2}}\right)^3\right]$$
$$= \frac{4\pi}{3}\left[8\left(\frac{1}{\alpha}+\frac{1}{\sqrt{2}}\right)^3-\left(\frac{1}{\alpha}-\frac{1}{\sqrt{2}}\right)^3\right],$$

where n_{max} is the maximum number of boxes of a fixed size interacting with any particle.

The polynomial degree *p* needs to be selected for each particle-cluster interaction in order to restrict the error. The next theorem shows how to determine the multipole degree to keep interaction error constant.

Theorem 3 (Piecewise approximations) *The polynomial degree p_k required for a particle-cluster interaction for constant error is given by*

$$p_k = p_0 + k\log_\alpha 2 + \log_\alpha \frac{A_0}{A_k}$$

where A_k is the net charge on the cluster at level k, and A_0 is the smallest net charge cluster at lowest level in the tree.

Proof

Let b_j be a cluster of particles in a box of size r_{sj} at the *j*th level of the tree. To fix the error for clusters at different levels in the tree, we force the bound on error (Thm. 2) to be equal:

$$\frac{A_k}{r_{sk}}\cdot\frac{\alpha^{p_k+2}}{1-\alpha} = \frac{A_j}{r_{sj}}\cdot\frac{\alpha^{p_j+2}}{1-\alpha},$$

for the pair of clusters b_j and b_k. This simplifies to

$$\frac{A_j}{A_k}\cdot\frac{r_{sk}}{r_{sj}} = \alpha^{p_k-p_j},$$

where $r_{sk}/r_{sj} = 2^{k-j}$. The theorem follows from the choice of b_j as the smallest net charge cluster at lowest level.

In the original multipole method, for all other domains with higher aggregate charge, this error criteria will be violated. In general, we select a minimum degree of interaction p_0 associated with a threshold value A_0 and increase multipole degree for larger cluster sizes. For structured domains, it is easy to control the polynomial degree in this manner since increase in the polynomial degree is not large. The multipole series are computed *a-priori* to the maximum required degree (this is possible since all parameters for the degree of an interaction are available at the time of tree construction). However, this technique can result in very large degree multipoles for unstructured domains. This difficulty is overcome by:

2. computing and storing the increased degree multipoles, or
3. using alternate height-balanced tree constructions.

In this paper, we concentrate primarily on uniform distributions; but will demonstrate empirically that the paradigm works for unstructured domains as well.

We now examine the error associated with the Barnes-Hut method with this improved multipole degree selection criteria.

Theorem 4 (Global error in Barnes-Hut) *The error in the piecewise approximate Barnes-Hut method for structured distributions with uniform charge density is* $O(\alpha^{p+1} \log A)$.

Proof

Observe that the number of particle-cluster interactions with fixed size clusters is bounded (Lemma 2), the number of distinct sizes of clusters equals the height of the decomposition tree ($O(\log n)$ for structured distributions), and the error associated with each interaction is constant (Thms. 2 and 3). From this it can be concluded that the error for uniform charge density is $O(\alpha^{p+1} \log n)$. The proof follows directly from the observation that for uniform charge density, $\log n$ is equivalent to $\log A$.

The reader will note that this error is considerably less than the error bound on the original fixed-degree multipole based Barnes-Hut method. The only issue that remains to be resolved is the increased computation introduced by the additional multipole evaluations. The next theorem shows that this additional computation is minimal.

Theorem 5 (Complexity) *For a structured particle distribution, the computational complexity of the piecewise approximate Barnes-Hut method is given by* $O(n(p+l)^3)$, *where* l *is the number of levels of the hierarchical decomposition.*

Proof

For each particle, we need to compute at most n_{max} interactions with p_k degree multipole for levels $k = 0, \ldots, l$. The total computation is proportional to $n \cdot n_{max} \sum_{k=0}^{l} p_k^2$. For uniform charge density, Thm. 3 prescribes $p_k = p_0 + ck$ where c is a constant that depends on α only. Therefore, the overall computation is proportional to $n(p_0 + l)^3$, where $p_0 = p$.

This result can be extended to unstructured distributions as well using the box-collapsing and flexible splitting techniques of Callahan and Kosaraju [2]. It is useful to note that the complexity of the original Barnes-Hut method grows as $O(p^2 n \log n)$. The number of levels in a uniform distribution l grows as $\log_8 n$ assuming a single particle per leaf cell. For typical values of p (6-7 degree approximations), this corresponds to between 256K-2M particles. In order to optimize cache performance and for lower algorithmic constants, leaf nodes of the tree often represent clusters of up to 32 or 64 particles. This increases the number of particles to between 8M and 64M.

Thus, even for very large scale simulations, the improved method is within a small constant off the fixed-degree method. In general, for $l \leq p$, the complexity of the improved method is within 7/3 of the original method. Clearly, the new method yields significant improvements in error while incurring minimum additional overhead.

Experimental Results

Experimental Setup

The improved and original Barnes-Hut methods are coded for an Origin 2000 and tested with up to 32 processors. The code is based on POSIX threads and optimized for single-processor cache performance, data-locality across processors, and false sharing. The parallel formulation exploits the concurrency available in independent tree traversal of each particle. The particles are sorted in a proximity-preserving order (a Peano-Hilbert ordering) and force computation for sets of *w* particles are aggregated into a single thread. We refer the reader to [5, 6, 15, 14] for a more detailed explanation of these schemes.

The treecode was tested in the context of particle simulations as well as boundary element solvers. Problem instances for particle simulations range from uniform to highly irregular distributions in three dimensions. Uniform distributions correspond to a random distribution of points distributed equally across the domain. Irregular distributions are generated using a Gaussian density function or overlapped Gaussian distributions (multiple Gaussians superimposed).

The notion of error in a simulation is formally defined as follows: let *a* be the vector corresponding to the accurate potentials at *n* particles; if the potentials computed from the treecode are represented by the vector *a*' then the error ϵ in the simulation is defined as:

$$\epsilon = \max |a_i - a'_i| \; i = 1 \,..\, n$$

Serial Complexity and Error Bounds

To compare the serial complexities of the new and original methods, we use the number of multipole terms evaluated. The number of terms is an excellent indication of the serial computation time. Using this instead of wall clock time allows us to eliminate impact of potentially varying parallel efficiencies of the two methods and processor loads. Issues of parallel efficiencies are addressed subsequently in this section.

Structured Distributions

n	ϵ_{orig}	ϵ_{new}	Terms(orig)	Terms(new)
10000	0.012027	0.012027	12 million	12 million
15000	0.017326	0.010399	25 million	25 million
20000	0.025982	0.016820	60 million	60 million
30000	0.036880	0.017386	95 million 254 million	96 million 297 million

80000	0.098395	0.019327		

Unstructured Distributions

n	ϵ_{orig}	ϵ_{new}	Terms(orig)	Terms(new)
45000	2.479027	0.334691	70 million	102 million
82000	2.307508	0.268452	179 million	224 million

Table 1: Comparison of the new method with the original method.

Table 1 illustrates the errors and the number of term expansions. From these tables, it is easy to see that the growth in error is much faster in the original method than in the improved method. Furthermore, the term expansions of the two methods are similar. This is also illustrated graphically in Fig. 2 and is in good agreement with our theoretical results.

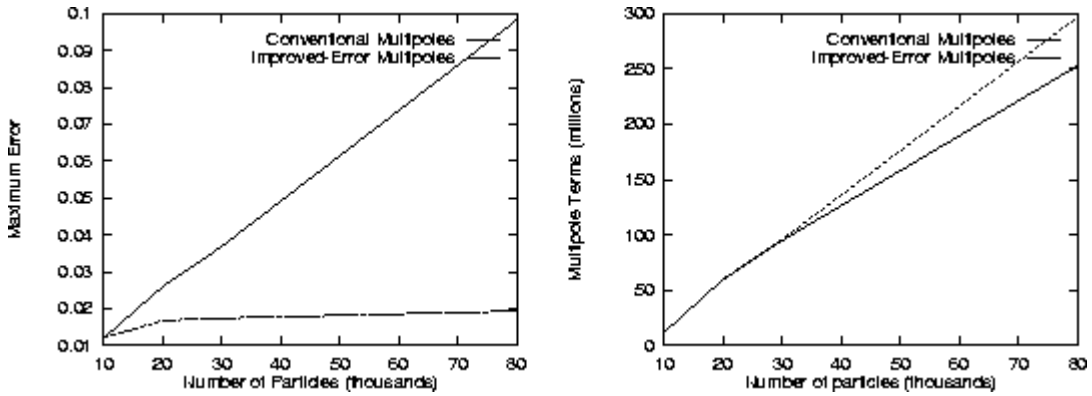

Figure 2: A comparison of the error and computational cost of the original and new methods illustrates the close agreement with theoretical results and advantages of the new scheme.

Parallel Performance

Results on parallel performance are presented in Table 2 for a 32 processor SGI Origin 2000. The speedup is computed as the ratio of the runtime of the threaded version with multiple kernel threads to that of the single thread version. It is evident from the table that the performance of the treecode is extremely good, with parallel efficiencies in the range of 80 - 90%. This must be tempered by the observation that the dataset for the two simulations presented is only slightly larger than the total L2 cache. Nevertheless, the treecode yields excellent speedups on the Origin 2000.

Problem	Serial		Parallel	
	Original	New	Original	New
uniform40K	195.46	212.41	6.68 (29.26)	7.37 (28.82)
non-uniform46K	360.93	390.68	11.67 (30.92)	12.97 (27.83)

Table 2: Runtimes (in seconds) and speedups (in parenthesis) for single-thread and multi-threaded versions of a single iteration of the treecode on a 32 processor SGI Origin 2000.

From Table 2, we can conclude that our threaded parallel formulations yield excellent speedups.

Furthermore, the new algorithm yields slightly poorer speedups than the original algorithm. This is because the new algorithm fetches longer multipole series. However, the effect of this increased communication is not very significant because a large fraction of the data is local to the processor. The increased communication volume can also be estimated in a manner similar to the computation and shown to be bounded.

Solving Boundary Integral Equations

The treecode can be used to solve dense linear systems arising from boundary element methods for solving integral equations. In particular, the treecode was used to compute matrix-vector products with the approximation of the dense matrices in each iteration of the GMRES iterative solver. The surface of the domain is discretized into triangular elements. Gaussian quadrature is used for integration over the surface. Typically, a fixed number of Gauss-points are located inside each element and inserted into the hierarchical domain representation. Using this hierarchical domain, the potential is computed at the vertices of the elements and matched to the boundary values. This process forms a single matrix-vector product that is required at each step of GMRES.

This technique was used to solve dense linear systems arising from three complex domains: propeller (140,800 elements, 70,439 nodes), gripper1 (142,296 elements, 71,152 nodes), and gripper2 (185,856 elements, 92,918 nodes). The first instance is a propeller from an airplane and the next two are surface discretizations of an industrial gripper. These correspond to highly unstructured problem instances, since a bulk of the volume is empty and the nodes are concentrated on the surface.

In Table 3, we present single iteration errors and execution times for the improved and original methods. The errors are computed with respect to a 9 degree polynomial since the exact method took an inordinately large amount of time. From the table, once again it is evident that the improved method yields significantly better error properties while adding minimal computational overhead. The matrix-vector product was used in a GMRES solver with a restart of 10 and was observed to converge very well. Using this method, we were able to solve dense systems with over 100,000 unknowns within a few minutes.

Propeller

140,800 elements, 70,439 nodes, 6 Gauss points per element

Algorithm	Degree	Time	ϵ
Original	4	31.83	0.000406
Improved	4*	33.60	0.000026
Reference	9	100.81	_____

Gripper

185,856 elements, 92,918 nodes, 6 Gauss points per element

Algorithm	Degree	Time	ϵ
Original	4	46.40	0.000516
Improved	4*	50.13	0.000028
Reference	9	151.21	_____

Table 3: Single iteration errors and execution times (seconds) on a 32 processor SGI Origin 2000 for the improved and original methods. Accuracy is compared with a reference using 9

degree multipole expansion (the exact computation takes over 900 seconds).

Ongoing Work and Conclusions

Hierarchical treecodes have proven to be a critical component of large scale n-body computations. In this paper, we have presented an improved treecode that yields considerably better error bounds while incurring minimal computational overhead. We prove these bounds theoretically and demonstrate them experimentally for uniform as well as non-uniform distribution. Parallel formulations of these techniques are shown to yield excellent speedups on a 32 processor SGI Origin 2000. The treecode is also applied to solving large scale boundary element problems. The performance of the new matrix-vector product is shown to be superior to the original method. The results presented in this paper can easily be extended to the the Fast Multipole Method as well. We are currently exploring this and extending our theoretical results to unstructured distributions.

References

1 J. A. Board, J. W. Causey, J. F. Leathrum, A. Windemuth, and K. Schulten. **Accelerated molecular dynamics with the fast multipole algorithm**. *Chem. Phys. Let.*, 198:89, 1992.

2 P. B. Callahan and S. R. Kosaraju. **A decomposition of multi-dimensional point-sets with applications to k-nearest-neighbors and n-body potential fields**. In *Proceedings of 24th Annual ACM Symp. on Theory of Computing*, pages 546-556, May 1992.

3 R. Coifman, V. Rokhlin, and S. Wandzura. **The fast multipole algorithm for the wave equation: A pedestrian prescription**. *IEEE Antennas and Propagation Magazine*, 35(3), June, 1993.

4 N. Engheta, W. Murphy, V. Rokhlin, and M. Vassiliou. **The fast multipole method for electromagnetic scattering problems**. *IEEE Transactions on Antennas and Propagation*, 40 (6), June, 1992.

5 Ananth Grama, Vipin Kumar, and Ahmed Sameh. **Parallel hierarchical solvers and preconditioners for boundary element methods**. Technical report, Purdue University, W. Lafayette, IN 47907, 1997. Accepted for publication in SIAM Journal of Scientific Computing.

6 Ananth Grama, Vipin Kumar, and Ahmed Sameh. **Scalable parallel formulations of the Barnes-Hut method for n-body simulations**. Technical report, Purdue University, W. Lafayette, IN 47907, 1997. Accepted for publication in the Parallel Computing Journal.

7 L. Greengard. ***The Rapid Evaluation of Potential Fields in Particle Systems***. MIT Press, 1987.

8 L. Greengard and V. Rokhlin. **A fast algorithm for particle simulations**. *J. Comp. Physics*, 73:325-348, 1987.

9 Neal Katz, Thomas Quinn, Edmund Bertschinger, and James M. Gelb. **Formation of**

quasars at high redshift. *Mon. Not. R. Astron. Soc.*, 270:L71-L74, 1994.

10 Neal Katz and Simon White. **Hierarchical galaxy formation: overmerging and the formation of an x-ray cluster**. *Astrophysical Journal*, 412(2):412 - 455, 1993.

11 K . Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. **Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory**. *SIAM Journal on Scientific Computing*, 15(3):713 - 735, May 1994.

12 V. Rokhlin. **Rapid solution of integral equations of classical potential theory**. *Journal of Computational Physics*, 60:187 - 207, Sept 1985.

13 John Salmon and Michael S. Warren. **Parallel out-of-core methods for n-body simulation**. In *8th SIAM Conf. on Parallel Processing for Scientific Computing*, 1997.

14 J. Singh, C. Holt, T. Totsuka, A. Gupta, and J. Hennessy. **Load balancing and data locality in hierarchical n-body methods**. *Journal of Parallel and Distributed Computing*, 1994.

15 M. Warren and J. Salmon. **A parallel hashed oct tree n-body algorithm**. In *Proceedings of Supercomputing Conference*, 1993.

Author Biography

Vivek Sarin: Dr. Sarin's current research interests include scientific computing, numerical methods for engineering applications, and parallel algorithms and architectures. His recent work has focused on iterative methods, preconditioning of large sparse systems, and their application to problems arising in computational fluid dynamics and electromagnetics. At present, he is working on NSF Grand Challenge problems in large scale simulation of particles in fluids.

Dr. Sarin received his M.S. in Computer Science from University of Minnesota in 1993 and Ph.D. in Computer Science from University of Illinois at Urbana-Champaign in 1997. He is a member of Phi Kappa Phi.

Ananth Grama: Prof. Grama's research interests span the areas of parallel and distributed computing architectures, algorithms, and applications. His work on distributed infrastructure deals with development of software support for dynamic clustered and multiclustered environments. Models for platform abstractions and performance modeling are also being developed. His research on applications has focused on particle dynamics methods and their applications to dense linear system solvers. More recently, he has also been working on fast algorithms for data mining applications.

Prof. Grama has authored several papers and a text book "Introduction to Parallel Computing: Design and Analysis of Algorithms" (with Vipin Kumar, Anshul Gupta, and George Karypis) with another book forthcoming on " Principles of Parallel Programming" (with Vipin Kumar and George Karypis).

Ahmed Sameh: Ahmed Sameh is the Head, and Samuel D. Conte Professor of Computer Science

at Purdue University, West Lafayette. He joined Purdue in January, 1997, after being the Head of Computer Science at the University of Minnesota, Minneapolis, and the holder of the William Norris Chair in Large-Scale Computing, and prior to that the Director of the Center for Supercomputing Research and Development at the University of Illinois at Urbana-Champaign. His current research interests include numerical linear algebra, and the design and performance analysis of parallel numerical algorithms.

Sameh received his B.Sc. from the University of Alexandria, Egypt in 1961, M.S. from Georgia Institute of Technology in 1964, and Ph.D. from the University of Illinois (U-C) in 1968, all in Civil Engineering (Structural Mechanics). He was a faculty member of the Department of Computer Science at the University of Illinois (U-C) from 1968 to 1991. He is a member of ACM, and SIAM, a senior member of IEEE, and a fellow of the American Association for the Advancement of Science (AAAS).