Multipole-Based Preconditioners for Sparse Linear Systems.

Ananth Grama
Purdue University.

Supported by the National Science Foundation.

Overview

- Summary of Contributions
- Generalized Stokes Problem
- Solenoidal Basis Methods and Preconditioning
- Multipole Methods as Preconditioners
- Performance of Multipole-Based Preconditioners
- Parallelization of Solver/Preconditioner
- Parallel Performance
- Concluding Remarks

Summary of Contributions

- Problem Formulation
- Excellent Convergence Properties of Multipole-Based Preconditioners
- Parallelism in Multipole-based Sparse Solvers
- Highly Scalable and Efficient Parallel Formulations

Generalized Stokes Problem

- Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \bullet \nabla \mathbf{u} = -\nabla p + \frac{1}{R} \Delta \mathbf{u} \quad \text{in}\, \Omega$$

$$\nabla \bullet \mathbf{u} = \mathbf{0} \quad \text{in}\, \Omega$$

$$\mathbf{u} = \mathbf{g} \quad \text{on}\ \partial\Omega$$

- Incompressibility Condition $\nabla \bullet \mathbf{u} = 0$

Generalized Stokes Problem

- Linear system

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathrm{u} \\ \mathrm{p} \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

- B^T is the discrete divergence operator and *A* collects all the velocity terms.

- Navier-Stokes:

$$A = \frac{1}{\Delta t} M + C + \frac{1}{R} L$$

- Generalized Stokes Problem:

$$A = \frac{1}{\Delta t} M + \frac{1}{R} L$$

Solenoidal Basis Methods

- Class of projection based methods
- Basis for divergence-free velocity
- Solenoidal basis matrix *P*: $B^T P = 0$
- Divergence free velocity: $\mathrm{u} = P\mathrm{x}$

$$B^T \mathrm{u} = B^T P\mathrm{x} = 0$$

- Modified system

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathrm{u} \\ \mathrm{p} \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad \Rightarrow \quad AP\mathrm{x} + B\mathrm{p} = f$$

Solenoidal Basis Method

- Reduced system:

$$B^T P = P^T B = 0$$

$$AP\mathrm{x} + B\mathrm{p} = f \quad ; \qquad P^T AP\mathrm{x} = P^T f$$

- Reduced system solved by suitable iterative methods such as CG or GMRES.

- Velocity obtained by : $\mathrm{u} = P\mathrm{x}$

- Pressure obtained by: $Bp = f - APx$

Preconditioning

- Observations

$$y = P^T u \quad \Rightarrow \quad \xi = \nabla \times \mathbf{u}$$

$$u = Pw \quad \Rightarrow \quad \mathbf{u} = \nabla \times \psi$$

$$y = P^T P w \quad \Rightarrow \quad \xi = \nabla \times \nabla \times \psi$$

- Vorticity-Velocity function formulation:

$$\frac{\partial \xi}{\partial t} = -\frac{1}{R} \nabla \times \nabla \times \xi, \qquad \xi = -\Delta \psi$$

Preconditioning

- Reduced system approximation:

$$P^T A P \approx \left[\frac{1}{\Delta t} M + \frac{1}{R} P^T P \right] P^T P$$

- Preconditioner:

$$G \approx \left[\frac{1}{\Delta t} M + \frac{1}{R} L_s \right] L_s$$

- Low accuracy preconditioning is sufficient

Preconditioning

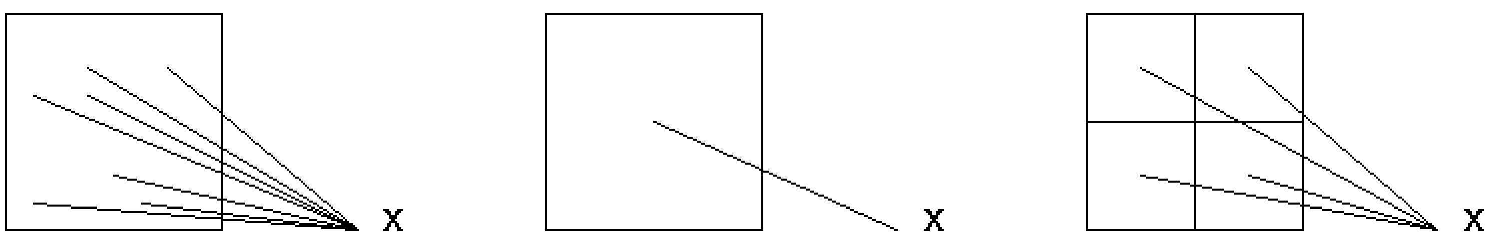
- Preconditioning can be affected by an approximate Laplace solve followed by a Helmholtz solve.
- The Laplace solve can be performed effectively using a Multipole method.

Preconditioning Laplace/Poisson Problems

- Given a domain with known internal charges and boundary potential (Dirichlet conditions), estimate potential across the domain.
  - Compute boundary potential due to internal charges (single multipole application)
  - Compute residual boundary potential due to charges on the boundary (vector operation).
  - Compute boundary charges corresponding to residual boundary potential (multipole-based GMRES solve).
  - Compute potential through entire domain due to boundary and internal charges (single multipole).

Overview of Multipole Methods

- Consider the problem of computing the trajectory of particles in space.
- Multipole methods use hierarchical approximations to reduce computational cost.



set of particles impacting x.

Can they be approximated by their center of mass?

If not, divide domain and recursively apply the same criteria to sub-domains.

Overview of Multipole Methods

- Accurate formulation requires $O(n^2)$ force computations (mat-vec with a coefficient matrix of Green's functions)
- Hierarchical methods reduce this complexity to O(n) or O(n log n)
- Since particle distributions can be very irregular, the tree can be highly unbalanced
- Commonly used hierarchical methods include FMM and the Barnes-Hut methods.

Overview of Multipole Methods

- Interactions (direct force computations) are replaced by Gaussian quadratures.
- Far-field interactions are replaced by multipole series representing remote Gauss points.
- Each matrix-vector product now becomes a single tree traversal taking O(n) or O(n log n) time.
- This mat-vec can be encapsulated in iterative solvers for dense systems.

Overview of Multipole Methods

- A number of results relating to the complexity of hierarchical methods [Rokhlin, Greengard], error analysis and control [Grama, Sarin, Sameh], and their use in dense solvers [Grama, Kumar, Sameh] have been shown. Their applications in applications such as inductance extraction [White, Sarin, Grama] have also been demonstrated.

Numerical Experiments

- 3D driven cavity problem in a cubic domain.
- Marker-and-cell scheme in which domain is discretized uniformly.
- Pressure unknowns are defined at node points and velocities at mid points of edges.
- x, y, and z components of velocities are defined on edges along respective directions.

Sample Problem Sizes

Mesh	Pressure	Velocity	Solenoidal Functions
8x8x8	512	1,344	1,176
16x16x16	4,096	11,520	10,800
32x32x32	32,768	95,232	92,256

Preconditioning Performance

Table 1: Effectiveness of the preconditioner for the generalized Stokes problem ($\tau = 10^{-3}$).

		Iterations	
Mesh Size	Unknowns	Unpreconditioned	Preconditioned
$8 \times 8 \times 8$	1856	66	8
$16 \times 16 \times 16$	15616	208	12
$32 \times 32 \times 32$	128000	772	17

Preconditioning Performance

Table 2: Effectiveness of the preconditioner for various instances of the generalized Stokes problem.

Mesh Size	Unknowns	$\tau = 10^{-3}$	$\tau = 10^{-1}$	$\tau = 10^{0}$	$\tau = 10^{1}$	$\tau = 10^{3}$
$8 \times 8 \times 8$	1856	8	8	6	5	5
$16 \times 16 \times 16$	15616	12	10	8	6	6
$32 \times 32 \times 32$	128000	17	13	10	7	7

Preconditioning Performance – Poisson Problem

Table 3: Impact of multipole expansion parameters on convergence (iterations) and acceleration (time, in seconds). A dashed entry in the table indicates that no convergence was observed in 20 iterations. All times were observed on a Pentium 4 workstation operating at 2.4GHz with 1 GB RAM running Solaris.

Mesh Size	Multipole deg.	alpha	Iterations	time
Unpreconditioned				
$20 \times 20 \times 20$			799	0.82
Preconditioned				
$20 \times 20 \times 20$	0	0.91	-	-
$20 \times 20 \times 20$	0	0.67	-	-
$20 \times 20 \times 20$	0	dense	3	12.54
$20 \times 20 \times 20$	1	0.91	-	-
$20 \times 20 \times 20$	1	0.77	-	-
$20 \times 20 \times 20$	1	0.67	-	-
$20 \times 20 \times 20$	2	0.91	3	3.23
$20 \times 20 \times 20$	2	0.77	3	4.03
$20 \times 20 \times 20$	2	0.67	3	5.19

Preconditioning Performance– Poisson Problem

Unpreconditioned				
$30 \times 30 \times 30$			1895	13.35
Preconditioned				
$30 \times 30 \times 30$	0	0.91	-	-
$30 \times 30 \times 30$	0	0.67	-	-
$30 \times 30 \times 30$	0	dense	2	93.36
$30 \times 30 \times 30$	1	0.91	3	8.42
$30 \times 30 \times 30$	1	0.77	3	11.28
$30 \times 30 \times 30$	1	0.67	3	15.30
$30 \times 30 \times 30$	2	0.91	3	17.63
$30 \times 30 \times 30$	2	0.77	3	22.71
$30 \times 30 \times 30$	2	0.67	3	30.03

Preconditioning Performance–Poisson Problem

Unpreconditioned				
$40 \times 40 \times 40$			3451	59.87
Preconditioned				
$40 \times 40 \times 40$	0	0.91	-	-
$40 \times 40 \times 40$	0	0.67	-	-
$40 \times 40 \times 40$	0	dense	2	454.28
$40 \times 40 \times 40$	1	0.91	3	20.82
$40 \times 40 \times 40$	1	0.77	3	28.83
$40 \times 40 \times 40$	1	0.67	3	40.00
$40 \times 40 \times 40$	2	0.91	3	34.50
$40 \times 40 \times 40$	2	0.77	2	23.61
$40 \times 40 \times 40$	2	0.67	2	32.51

Parallel Formulation - Outer Solve

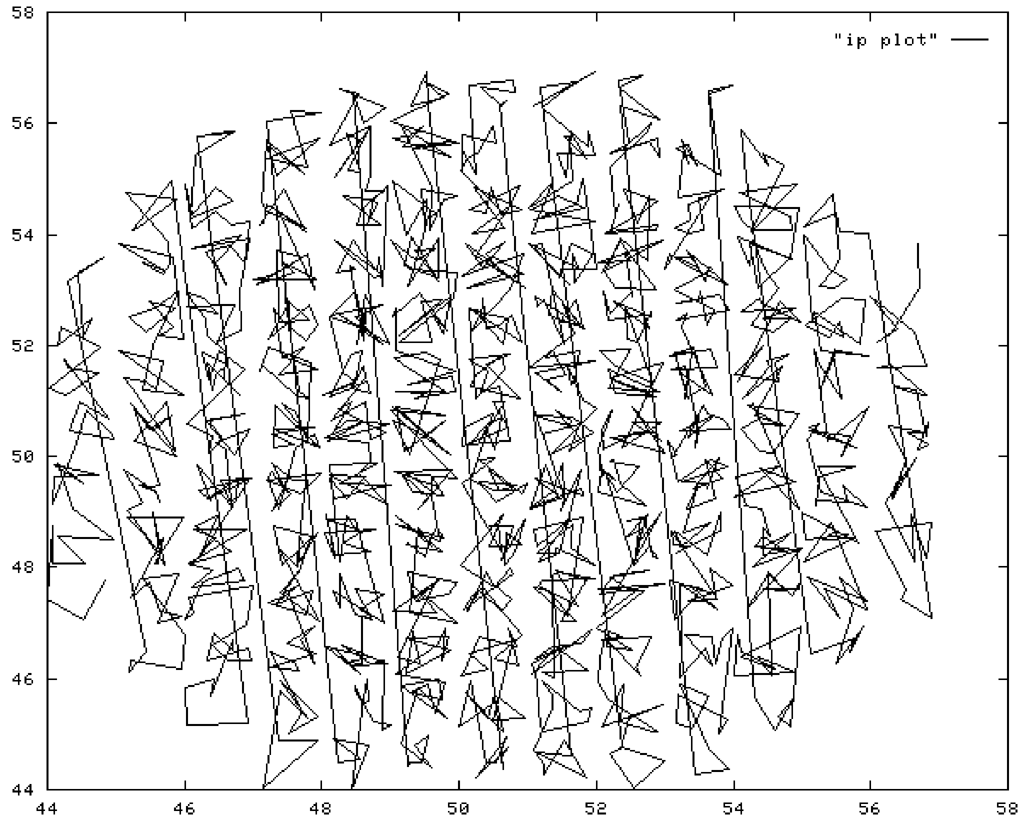
- Partition domain across processors
- Computation and storage of solenoidal basis matrix P
- Matrix-vector products: $y = P^T A P x$
  - Local operations on the grid
- Vector operations
- Preconditioning
- Scalable parallel implementations have been developed for all these operations
- Matrix free approach

Parallel Formulation (Multipole Solve)

- Each node evaluation can potentially be executed as a thread.
- Since there may be a large number of nodes, this may lead to too many threads. In general, we build some granularity into the thread by gathering k nodes into a single thread.
- The Barnes-Hut tree is a read-only data structure. Therefore, good performance can be achieved if we can build spatial and temporal locality (and the working set size does not exceed local memory).

Parallel Formulation(Multipole Solve)

- Spatially proximate particles are likely to interact with nearly identical parts of the tree. Therefore particles must be traversed in a spatial-proximity preserving order.



Parallel Performance

Table 4: Parallel performance on the IBM p690 and SGI Origin shared-memory multiprocessors.

Mesh Size	IBM p690 Time (s)			SGI Origin Time (s)		
	$p = 1$	$p = 8$	Efficiency	$p = 1$	$p = 32$	Efficiency
$25 \times 25 \times 25$	7.72	1.21	0.80	12.06	1.17	0.32
$30 \times 30 \times 30$	19.96	3.25	0.77	30.78	2.44	0.39
$40 \times 40 \times 40$	31.18	4.80	0.81	48.29	3.40	0.44
$50 \times 50 \times 50$	68.88	9.69	0.81	96.28	5.73	0.52
$60 \times 60 \times 60$	108.17	15.47	0.87	163.63	7.41	0.69

Parallel Performance

Table 5: Runtime (in seconds) and efficiency of parallel algorithm on x86 Solaris shared-memory multiprocessors.

	4-proc SMP (550 MHz P3)			8-proc SMP (750 MHz Xeon)		
Mesh Size	$p = 1$	$p = 4$	Efficiency	$p = 1$	$p = 8$	Efficiency
$25 \times 25 \times 25$	16.51	5.73	0.72	11.44	2.73	0.52
$30 \times 30 \times 30$	41.39	13.19	0.78	28.78	5.97	0.60
$40 \times 40 \times 40$	67.37	20.76	0.81	47.23	7.19	0.82
$50 \times 50 \times 50$	129.87	39.49	0.82	115.29	16.70	0.86

Concluding Remarks

- Multipole methods provide highly effective preconditioning techniques that yield excellent parallel speedups.
- The accuracy parameters of the multipole solve (degree, multipole acceptance criteria) significantly impact time and convergence rate.
- Rely on closed form Green’s function, but can be adapted to other scenarios.