Numerical Lab #2

- **Eigenvalue Problems (ARPACK)**
- **N-Body Simulators (FMM)**
- **Multigrid**

Overview : Lab #2

- Eigenvalue Problems
  - Algorithm Overview
  - ARPACK Introduction
- FMM in 3D
- "Hands-On" Experiments
  - Explore ARPACK
  - Multigrid Software

ARPACK: Overview

- **ARPACK** = **AR**noldi **PACK**age
- ARPACK is freely available:
  - http://www.caam.rice.edu/software/ARPACK/
  - Fortran 77 code
- Solves large scale Hermitian, non-Hermitian, standard or generalized eigenvalue problems.

ARPACK: Overview

- Computes a few (k) eigenvalues with specified features:
  - eigenvalues of largest magnitude
  - eigenvalues of smallest magnitude
  - simultaneous computation of eigenvalues at both ends of the spectrum
- Eigenvectors also available upon request.

ARPACK: Overview

- Uses these driver routines:
  - dsaupd : Real double precision symmetric problems
  - dnaupd : Real double precision non-symmetric problems
  - znaupd : Complex double precision problems (Hermitian or non-Hermitian)
- For each of these, there are associated routines to compute eigenvectors.

ARPACK: Overview

- Uses a reverse communication interface:

```
10 continue
   call dnaupd(ido, n, which, wrk, …)
   if (ido .eq. newprod) then
     call matvec('A',n,wrk(iptr(1)),wrk(iptr(2)))
   else
     return
   endif
   goto 10
```

- ARPACK doesn't care how A is stored!
- Ideal when w⟵Av has O(n) cost.

ARPACK: Overview

- Requires a fixed, predetermined amount of storage.
- $n \cdot O(k) + O(k^2)$, where
  - n = order of the matrix
  - k = number of requested eigenvalues

ARPACK : Lab Exercises

- Use the real non-symmetric driver to compute eigenvalues for matrices generated by MATPDE.

FMM : 3D

- See Leslie Greengard's Ph.D. thesis: *The Rapid Evaluation of Potential Fields in Particle Systems*.

FMM : 3D

- Obvious way of doing N-body simulation:
  - Compute forces on all N particles using Coulomb's law.
  - Advance position of particles for some timestep Δt using Newton's laws.
  - Repeat.
- Cost Per Timestep: $O(N^2)$

FMM : 3D

- FMM Goal : Do N-body simulation using only O(N) steps.
- Some N-body solvers compute forces directly - FMM computes the potential.
- Solution to Laplace's Equation:

$$\varphi = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left(L_n^m r^n + \frac{M_n^m}{r^{n+1}} \right) Y_n^m(\theta, \phi)$$

FMM : 3D

- Suppose that k charges of strengths $\{q_i, i = 1\ldots k\}$ are located at the points $\{S_i = (\rho_i, \alpha_i, \beta_i), i = 1\ldots k\}$, with $|\rho_i| \leq a$, for some a.
- The potential at $\vec{r} = (\rho, \theta, \phi) \in \Re^3$ is

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

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

FMM : 3D

- For some precision ϵ, is we keep only p terms in the expansion, we have

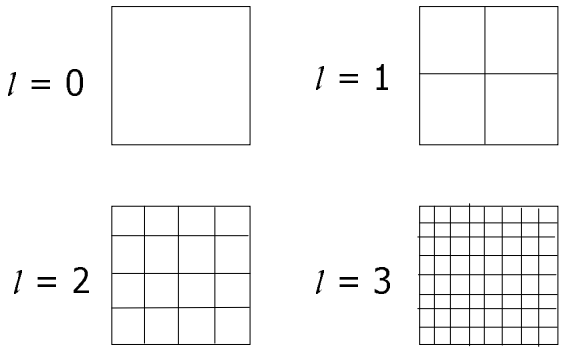
$$\left|\varphi(\vec{r}) - \sum_{n=0}^{p}\sum_{m=-n}^{n}\frac{M_n^m}{r^{n+1}}Y_n^m(\theta,\phi)\right| \le \frac{Q}{r-a}\left(\frac{a}{r}\right)^{p+1}$$

where $Q = \sum_{i=1}^{k}|q_i|$

Basic Data Structure

- In the 3D case, an octtree is used.
- Octtree:
  - Root of tree is 3D cube containing entire system
  - Each child of a node is a subdivision of that node's region into eight equal parts
  - For a tree of depth l, there are 8^l leaves.

Basic Data Structure (2D)



Basic Definitions

- *nearest neighbor*: For box *i* at level *l* in the octtree, a nearest neighbor is a box at the same level of refinement which shares a boundary point with box *i*.
- *second nearest neighbor*: For box *i* at level *l* in the octtree, a second nearest neighbor is a box at the same level of refinement which shares a boundary point with a nearest neighbor of box *i*.

Basic Definitions

- *Interaction list*: for box *i* at level *l*, it is the set of boxes which are children of the nearest and second nearest neighbors of *i*'s *parent* box and the *parent* box's nearest and second nearest neighbors.
- $\Phi_{l,i}$: the p[th]-order multipole expansion (about the box center) of the potential field created by the particles contained inside box *i* at level *l*.

Basic Definitions

- $\Psi_{l,i}$: the p[th]-order multipole expansion about the center of box *i* at level *l*, describing the potential field due to all the particles outside the box and its nearest and second nearest neighbors.
- $\tilde{\Psi}_{l,i}$: the p[th]-order multipole expansion about the center of box *i* at level *l*, describing the potential field due to all the particles outside of *l*'s *parent* box and the *parent* box's nearest and second nearest neighbors.

Basic Assumptions

- All the particles are stored in an octtree.
- The leaf nodes of the octtree contain the particles.
- For a fixed precision ϵ, we choose the number of terms in the multipole expansion to be p, where $p = \lceil -\log_2(\varepsilon) \rceil$

FMM Step #1 (Upward Pass)

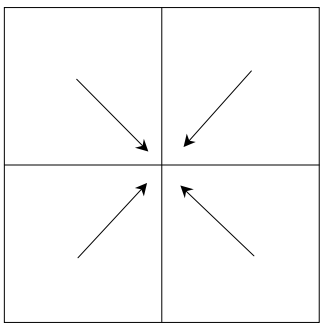
- For all leaf nodes *i* in the octtree, form a p[th]-degree multipole expansion $\Phi_{n,i}$ where *n* is the depth of the octtree.
- This expansion is the potential that all well-separated boxes at level *n* will feel due to the particles in box *i*.
- Really, we just need to form the moments M_n^m. This is the only data the other boxes would need.

FMM Step #2 (Upward Pass)

- For $l = n$-1 to 0 do
  - For all interior nodes i in the octtree at level l, form a p^{th}-degree multipole expansion $\Phi_{l,i}$ by using a translation operator to shift the center of each child's expansion to the current box center and add them together.
- For each box, the expansion represents the potential field due to all particles contained in one box

FMM Step #2 (Upward Pass)

- 2D Pictorial Representation:



FMM Step #3a (Downward Pass)

- Set $\tilde{\Psi}_{1,1} = \tilde{\Psi}_{1,2} = \tilde{\Psi}_{1,3} = \tilde{\Psi}_{1,4} = \tilde{\Psi}_{1,5} =$
- $\tilde{\Psi}_{1,6} = \tilde{\Psi}_{1,7} = \tilde{\Psi}_{1,8} = 0$

- There is no potential being felt from anything *outside* the octtree!

FMM Step #3b (Downward Pass)

- For $l = 1$ to $n-1$ do
  - For all interior nodes i in the octtree at level l, form $\Psi_{l,i}$ by translating each multipole expansion $\Phi_{l,j}$ for each box j in the interaction list of box i to a local expansion about the center of box i, adding these expansions together, and adding the result to $\tilde{\Psi}_{l,i}$
- This accounts for all interactions that were not accounted for at the parent's level.

FMM Step #3c (Downward Pass)

- For $l = 1$ to $n-1$ do
  - For all interior nodes i in the octtree at level l, form $\tilde{\Psi}_{l+1,i}$ for i's children by translating $\Psi_{l,i}$ to each of the children's centers.

FMM Step #4 (Downward Pass)

- For all leaf nodes i in the octtree, form $\Psi_{n,i}$ by translating each multipole expansion $\Phi_{l,j}$ for each box j in the interaction list of box i to a local expansion about the center of box i, adding these expansions together, and adding the result to $\tilde{\Psi}_{n,i}$
- This accounts for all interactions that were not accounted for at the parent's level, and which can be safely computed using a multipole expansion.

FMM Step #5 (Downward Pass)

- For all leaf nodes i in the octtree, and for each particle p_j at point P_j in the i^{th} box, evaluate $\Psi_{n,k}(P_j)$.
- We have now computed the potential experienced by every particle due to those particles that are *not* in its nearest or second nearest neighbors.
- This is the potential which can be safely computed using a multipole expansion!

FMM Step #6 (Downward Pass)

- For all leaf nodes i in the octtree, and for each particle p_j in the i^{th} box, compute interactions with all other particles within the box and its nearest and second nearest neighbors.
- These cannot be computed using a multipole expansion, and must be computed directly using the $O(N^2)$ algorithm.

FMM Step #7 (Downward Pass)

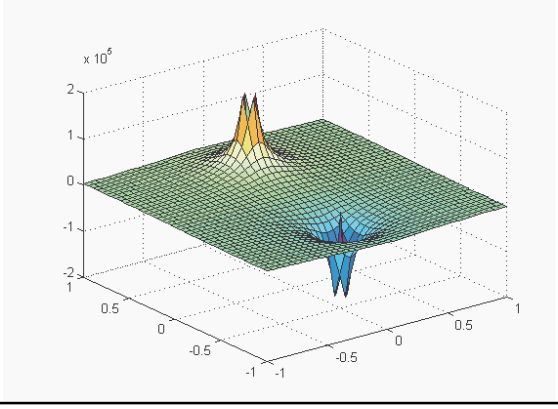
- For all leaf nodes i in the octtree, and for each particle p_j in the i^{th} box, add the direct terms (computed in step #6) and the far-field terms (computed in step #5) together.

Computing the Force:

- How can we compute $-\nabla\varphi$ without using finite differences?
- Recall that we have computed a *complex-valued* potential at each point.
- There exist formulas that allow us to compute the force directly from the *complex* potential.

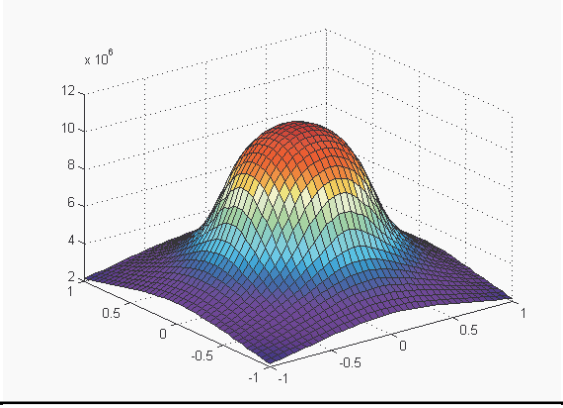
Sample Images

- Two Point Charges in X-Y Plane:



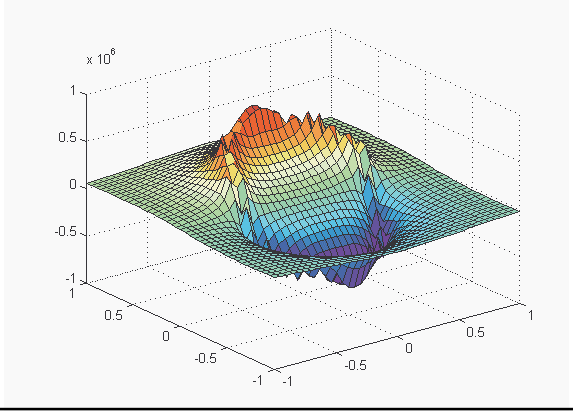
Sample Images

- (Approximate) Ring of Charge:



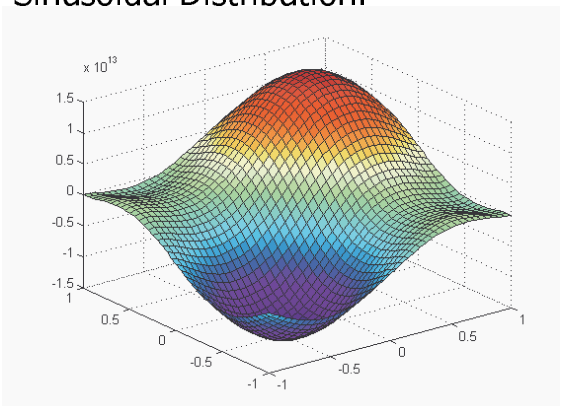
Sample Images

- Two Half-Rings of Charge of Opposite Sign:



Sample Images

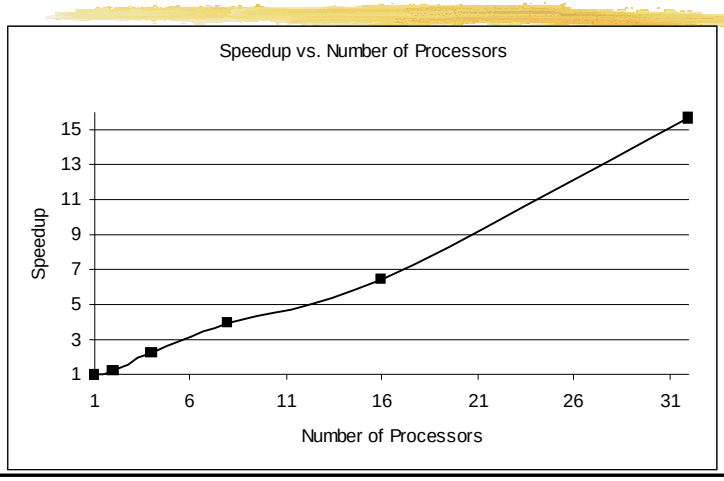
- Sinusoidal Distribution:



FMM: Parallel Implementation

- Assume distributed memory message passing environment (MPI)
- Assign each octtree node to an individual processor.
- Each processor processes the nodes it owns, blocking if information unavailable, and sending information it computes to other processors if necessary.

FMM: Parallel Implementation



ARPACK: Lab Exercises

- 1) Copy /scratch/tra1/ARPACK_work/* to your own workspace.
- 2) Type 'make all'.
- 3) Read (skim) dnsimp.f
- 4) Run dnsimp
- 5) Change parameters (nev, etc.) within dnsimp.f and explore.