Fast algorithms for high frequency wave propagation

Bjorn Engquist
University of Texas at Austin
and KTH, Stockholm

In collaboration with Lexing Ying

Numerical Analysis of Multiscale Problems,
Durham 7/5-15, 2010

Outline

1. Numerical challenges of high frequency wave propagation
2. Fast multi-pole methods (FMM)
3. Fast iterative solver for the full Helmholtz equation - preconditioners (variable $c(x)$)

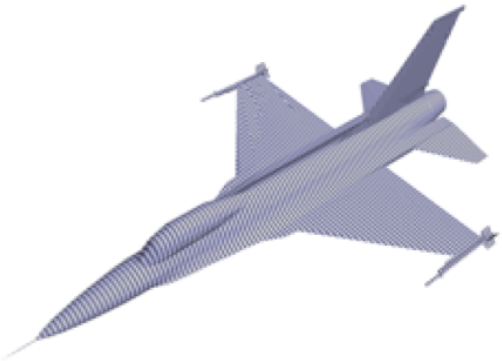
"New efficient solution techniques based on low rank matrix compressions"

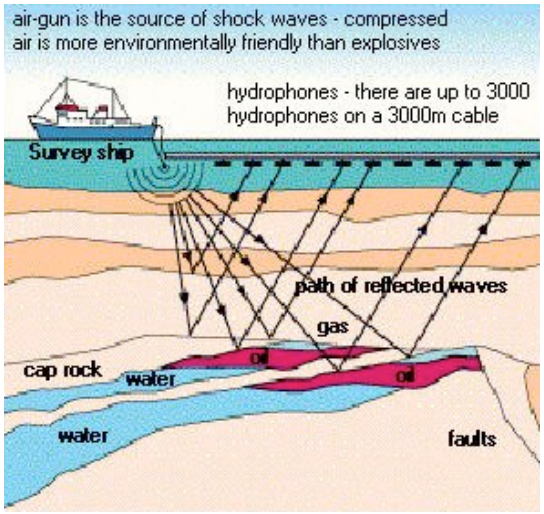
1. High frequency wave propagation and boundary integral techniques

- Applications: electro-magnetic, acoustic scattering and elastic wave propagation and scattering

$$\begin{cases} \mu \dfrac{\partial H}{\partial t} + \nabla \times E = 0, \\ \varepsilon \dfrac{\partial E}{\partial t} - \nabla \times H = 0 \end{cases}$$

$$\frac{\partial^2 p}{\partial t^2} = c(x)^2 \Delta p$$





High frequency wave propagation

- Computational challenge: Shannon sampling theorem - at least 2 unknowns/wavelenth $\rightarrow \geq O(\lambda^{-3})$ unknowns and $O(\lambda^{-4})$ flops
- Alternatives to FDM, FEM direct wave field approximation
  - Asymptotic techniques: geometrical optics, GTD, Gaussian beams
  - Dimensional reduction, frequency domain ($O(\lambda^{-4}) \rightarrow O(\lambda^{-3})$) and boundary integral formulation for constant coeff. ($\rightarrow O(\lambda^{-2})$)

Boundary integral techniques

- Applications: time harmonic wave propagation in piecewise homogeneous media

$$\begin{cases} \mu \dfrac{\partial H}{\partial t} + \nabla \times E = 0, \\ \varepsilon \dfrac{\partial E}{\partial t} - \nabla \times H = 0 \end{cases} \qquad n \times E = 0, x \in \partial\Omega$$

$$\downarrow$$

$$\lambda J(x) + \int_{\partial\Omega} G(x,y) J(y) dy = s(x)$$

$$\downarrow$$

$$(\lambda I + G) J = S$$

Discretisation (Galerkin, Nystrom) generates dense matrix problem

Boundary integral techniques, continued

- The choice of integral formulation and discretization technique are important but will not be discussed now
- Number of unknowns for given accuracy $O(\lambda^{-2})=O(\omega^2) = O(N)$
- Iterative method often enough - example GMRES or just

$$(\lambda I + G)J = S \rightarrow J^{n+1} = \lambda^{-1}(S - GJ^n)$$

- The challenge: large ($O(\omega^2)\times O(\omega^2)$), dense matrices
- For efficiency we need fast matrix vector multiplication algorithms

Fast matrix vector multiplications

- Exact decomposition

$$Ax = \left(\sum_j \prod_k A_{j,k} \right) x, \quad A_{j,k} \ sparse$$

  – Examples FFT (matrix-vector), Strassen (matrix-matrix)

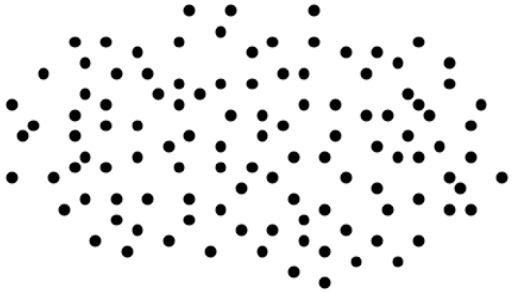
- Approximate decomposition based on approximate low rank interaction

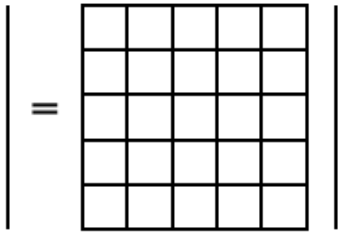
$$Ax \approx \left(\sum_j \prod_k A_{j,k} \right) x, \quad A_{j,k} \ sparse$$

  – Examples: FMM [Rokhlin, Greengard], H-matrix methods [Hackbusch], wavelet based compression [Beylkin]

2. Fast Multipole methods (FMM)

- Standard point to point charge interaction or equivalently matrix-vector multiply requires $O(N^2)$ operations
- For one level FMM: decompose domain or interaction matrix
- Example: compress interaction of domains that are far apart



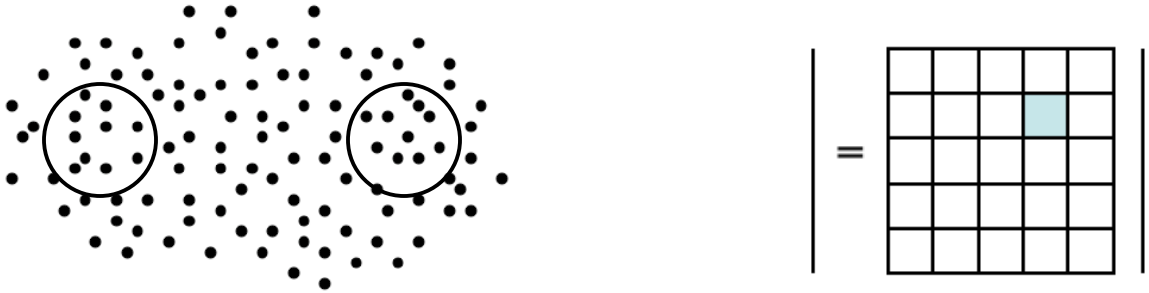


$$Example\ (Coulomb\ \ potential, Laplace\ eq.):$$

$$F_j = \sum_{k=1}^{N} a_{j,k} u_k = \sum_{k=1}^{N} \frac{u_k}{\left(x_k - x_j\right)}, \quad j = 1,..,N, \quad O(N^2)\ operations$$

Classical fast multipole methods (FMM)

- For one level FMM: decompose domain or interaction matrix
- Example, compress interaction of domains that are far apart



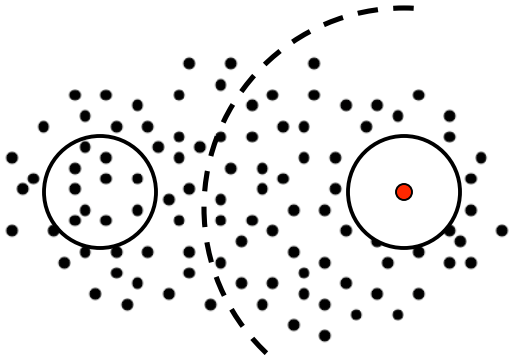
$$F_j = \sum_{k=1}^{N} a_{j,k} u_k = \sum_{k=1}^{N} \frac{u_k}{\left(x_k - x_j\right)}, \quad j = 1,..,N, \quad O(N^2)\ operations$$

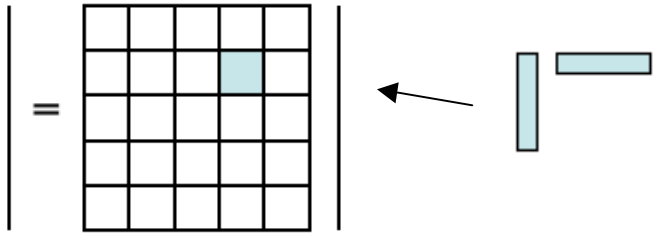
$$F_j = \sum_{k \in \Omega_2} \frac{u_k}{\left(x_k - x_j\right)} + \sum_{k\ else} \frac{u_k}{\left(x_k - x_j\right)}, \quad \sum_{k \in \Omega_2} \frac{u_k}{\left(x_k - x_j\right)} \approx \sum_{m=1}^{M} \frac{v_m}{\left(\bar{x} - x_j\right)^m}, \quad M << N, \quad j \in \Omega_j$$

From regularity at ∞, Puiseux series

Classical fast multipole methods (FMM)

- For one level FMM: decompose domain or interaction matrix
- Example, compress interaction of domains that are far apart





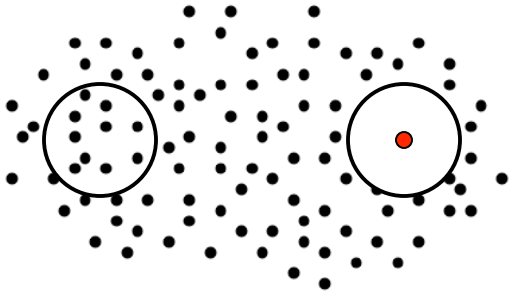
$$F_j = \sum_{k=1}^{N} a_{i,j} u_k = \sum_{k=1}^{N} \frac{u_k}{\left(x_k - x_j\right)}, \quad j = 1,..,N, \quad O(N^2)\ operations$$

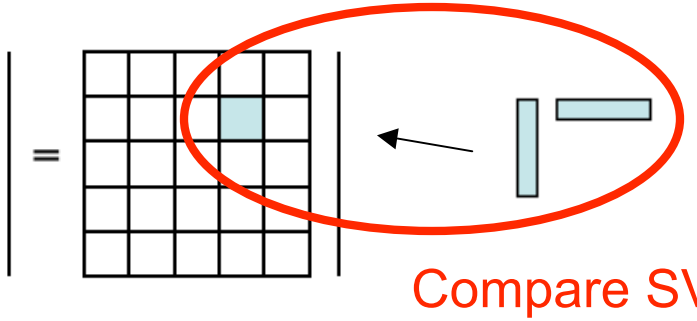
$$F_j = \sum_{(j\in\Omega_1),k\in\Omega_2} \frac{u_k}{\left(x_k - x_j\right)} + \sum_{j,k\ else} \frac{u_k}{\left(x_k - x_j\right)}, \quad \sum_{(j\in\Omega_1),k\in\Omega_2} \frac{u_k}{\left(x_k - x_j\right)} \approx \sum_{m=1}^{M} \frac{v_m}{\left(\bar{x} - x_j\right)^m}, \quad M << N$$

From regularity at ∞, Puiseux series

Classical fast multipole methods (FMM)

- For one level FMM: decompose domain or interaction matrix
- Example, compress interaction of domains that are far apart





Compare SVD

$$F_j = \sum_{k=1}^{N} a_{i,j} u_k = \sum_{k=1}^{N} \frac{u_k}{\left(x_k - x_j\right)}, \quad j = 1,..,N, \quad O(N^2)\ operations$$

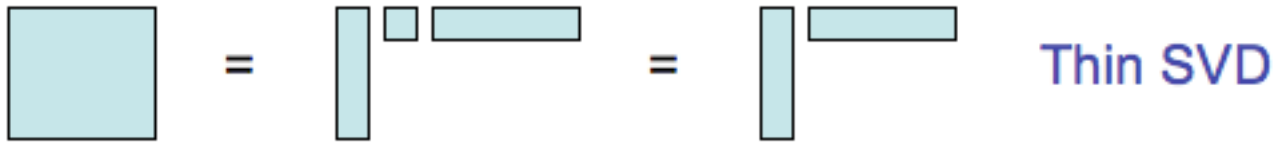
$$F_j = \sum_{j\in\Omega_1, k\in\Omega_2} \frac{u_k}{\left(x_k - x_j\right)} + \sum_{j,k\ else} \frac{u_k}{\left(x_k - x_j\right)}, \quad \sum_{j\in\Omega_1, k\in\Omega_2} \frac{u_k}{\left(x_k - x_j\right)} \approx \sum_{m=1}^{M} \frac{v_m}{\left(\bar{x} - x_j\right)^m}, \quad M << N$$

From regularity at ∞, Puiseux series

Singular Value Decomposition (SVD)

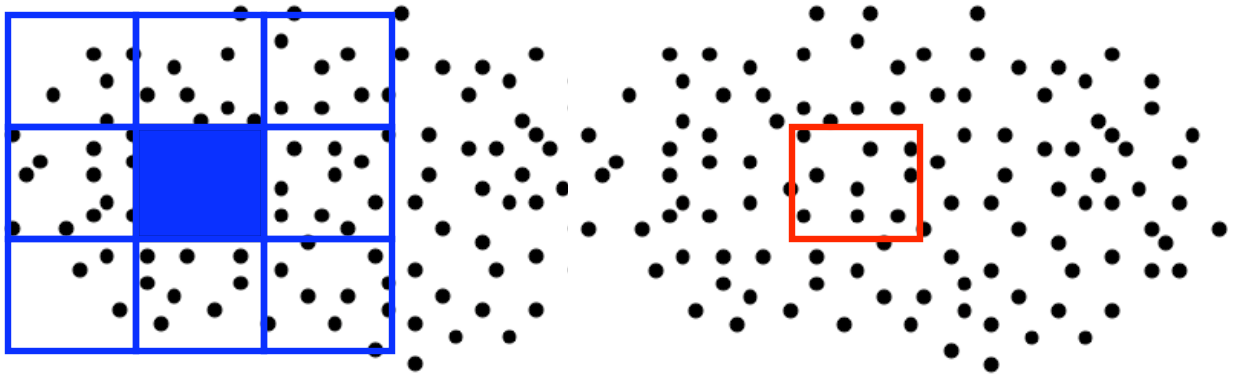
- Explores low rank interaction
- Reduces complexity in matrix-vector multiply

$$A = U\Sigma V^*, \quad U,V \text{ unitary}, \quad \Sigma \text{ diagonal non negative}$$



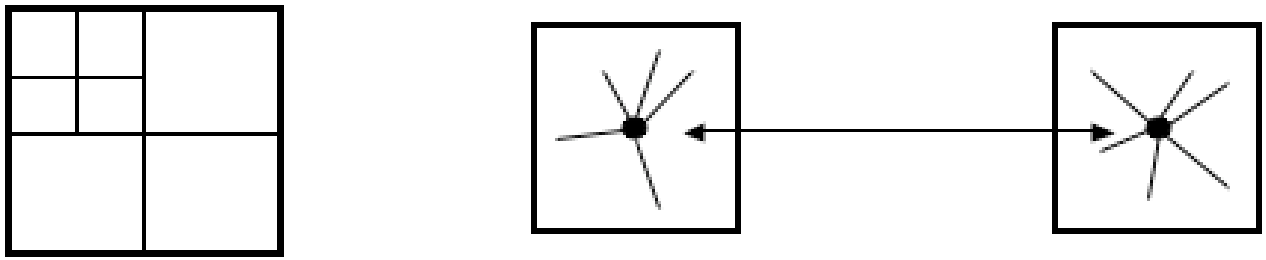
FMM continued: $O(N^{3/2})$ flops

- Computation of the multipole expansion requires $O(N^{1/2})$ flops in each of the $O(N^{1/2})$ clusters $\rightarrow$ $O(N)$ flops
- Every point (N) interacts with a finite number of nearby clusters of $N^{1/2}$ points $\rightarrow$ $O(N^{3/2})$ flops
- Every point also interacts with $O(N^{1/2})$ far field clusters and the interaction with each cluster requires O(1) flops $\rightarrow$ $O(N^{3/2})$ flops



FMM continued: O(N) flops

- $O(N^{1.5}) \rightarrow O(N)$ by, near field approximation, hierarchical sub-clustering and simplified translation operators
  - Far field representation built hierarchically
  - Uses near field Taylor expansion



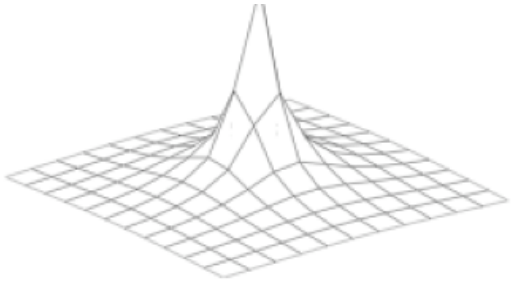
FMM continued

- Successfully applied to a variety of N-particle problems and potential formulations of elliptic equations.
- Requires low rank far field interaction (regularity at ∞), example discrete Calderon-Zygmund operators

$$A: \text{"}a_{j,k} \approx G(x_j, x_k)\text{"}$$

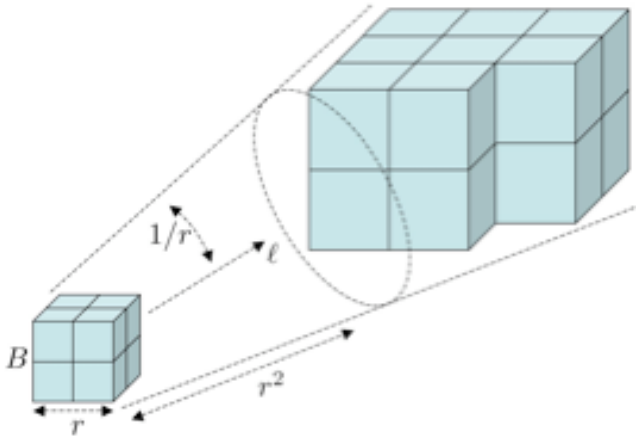
Oscillatory kernel $G_\omega(x,y)$

- Standard FMM works well for smooth kernels, ex. Laplace equation
- No far field low rank interaction for oscillatory kernels, ex Helmholtz equation
- Compare gravity and light from moon



FMM for oscillatory kernels

- Standard fast multi-pole methods does not apply to oscillatory kernels that are reasonably discretized.
- There are special purpose O(NlogN) methods [Rohklin], based on far field approximation of G(x,y) and also FFT based methods
- Now also oscillatory low rank O(NlogN) approximations [E.,Ying,] for based on decomposition in distance and angle



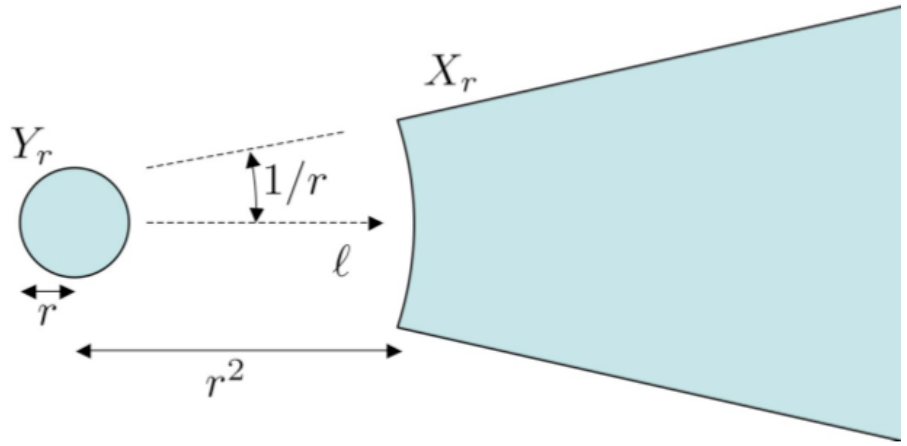
- Original interaction matrix replace by a sum of simpler matrix products

$$G(x_j, x_k) \rightarrow \quad = \sum \quad \leftarrow \begin{matrix} columns: f_\alpha(x_j) \\ rows: g_\beta(x_k) \end{matrix}$$

- Analytical foundation (Theorem)

$$\left| G(x,y) - \sum_{j=1}^{M(\varepsilon)} f_j(x) g_j(y) \right| \le \varepsilon, \quad \forall \omega, x \in X, y \in Y$$

$$G(x,y) = \exp(\omega i |x-y|)/|x-y|$$



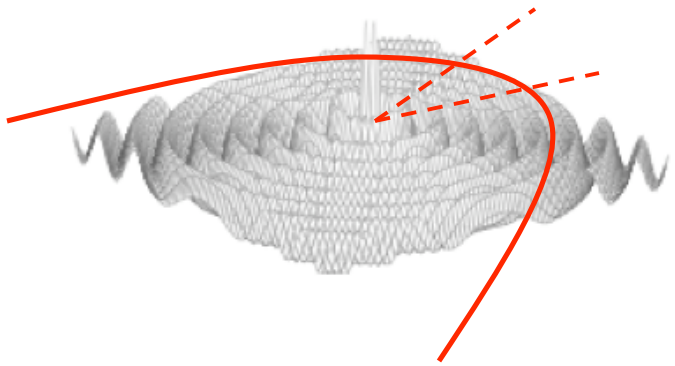
- Original interaction matrix replace by a sum of simpler matrix products

$$G(x_j, x_k) \rightarrow \quad = \sum \quad \leftarrow \begin{matrix} columns: f_\alpha(x_j) \\ rows: g_\beta(x_k) \end{matrix}$$

- Analytical foundation (Theorem)

$$\left| G(x,y) - \sum_{j=1}^{M(\varepsilon)} f_j(x) g_j(y) \right| \le \varepsilon, \quad \forall \omega, x \in X, y \in Y$$

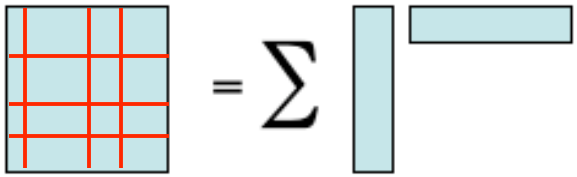
$$G(x,y) = \exp(\omega i |x-y|)/|x-y|$$



Remarks

- Proof based on uniform approximation lemmas
- Constructive proof but not practical as algorithm
- Random selection of rows and columns coupled to sequence of QR steps gives better practical results
- Instead of multipole: equivalent densities
- The O(NlogN) for nodes on co-dimension one manifold

Randomized construction of separated representation



1. A_s from A, randomly selected rows and columns (10 times over-sampling)

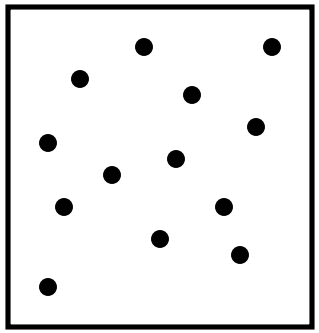
2. $\rightarrow$ $"\left|A_s - \tilde{Q}_c M \tilde{Q}_r^*\right| = O(\varepsilon)", \quad M = (\tilde{Q}_c)^+ A (\tilde{Q}_r^*)^+$

3. $\rightarrow$ Equivalent densities from $D = (R_c)^+ M, \quad Q_c R_c = A_c$

4. $$"\left| \frac{e^{2\pi i \left| x_i - x_j \right|}}{\left| x_i - x_j \right|} - \sum_{p,q} \frac{e^{2\pi i \left| x_i - x_p^c \right|}}{\left| x_i - x_p^c \right|} d_{p,q} \frac{e^{2\pi i \left| x_q^c - x_j \right|}}{\left| x_q^c - x_j \right|} \right| = O(\varepsilon)"$$

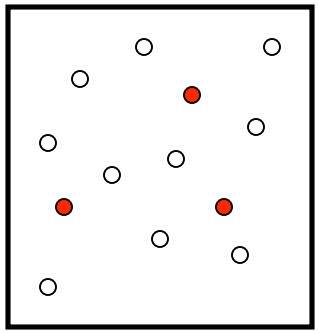
Equivalent densities

$$" \left| \frac{e^{2\pi i |x_i - x_j|}}{|x_i - x_j|} - \sum_{p,q} \frac{e^{2\pi i |x_i - x_p^c|}}{|x_i - x_p^c|} d_{p,q} \frac{e^{2\pi i |x_q^c - x_j|}}{|x_q^c - x_j|} \right| = O(\varepsilon) "$$



Equivalent densities

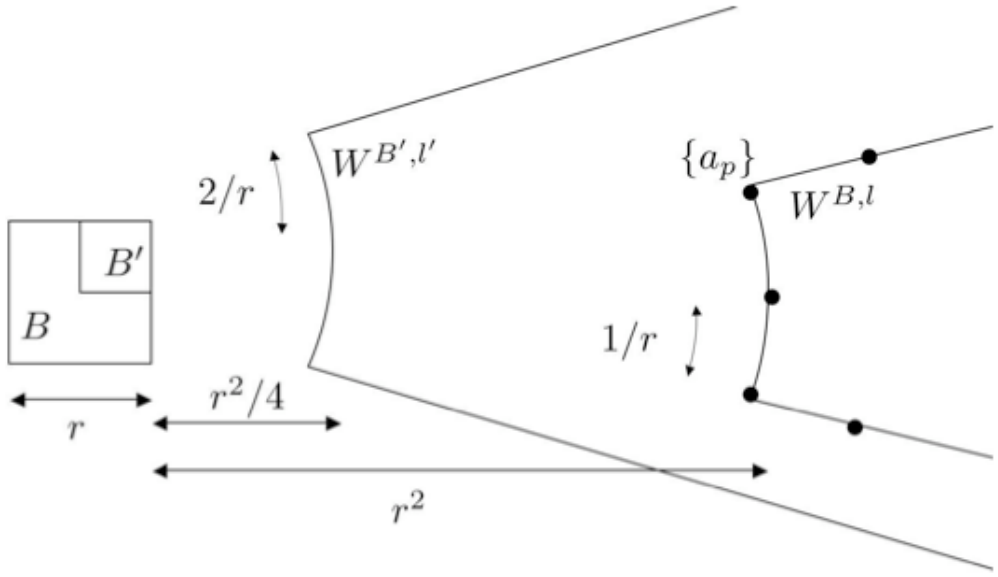
$$" \left| \frac{e^{2\pi i \left| x_i - x_j \right|}}{\left| x_i - x_j \right|} - \sum_{p,q} \frac{e^{2\pi i \left| x_i - x_p^c \right|}}{\left| x_i - x_p^c \right|} d_{p,q} \frac{e^{2\pi i \left| x_q^c - x_j \right|}}{\left| x_q^c - x_j \right|} \right| = O(\varepsilon) "$$

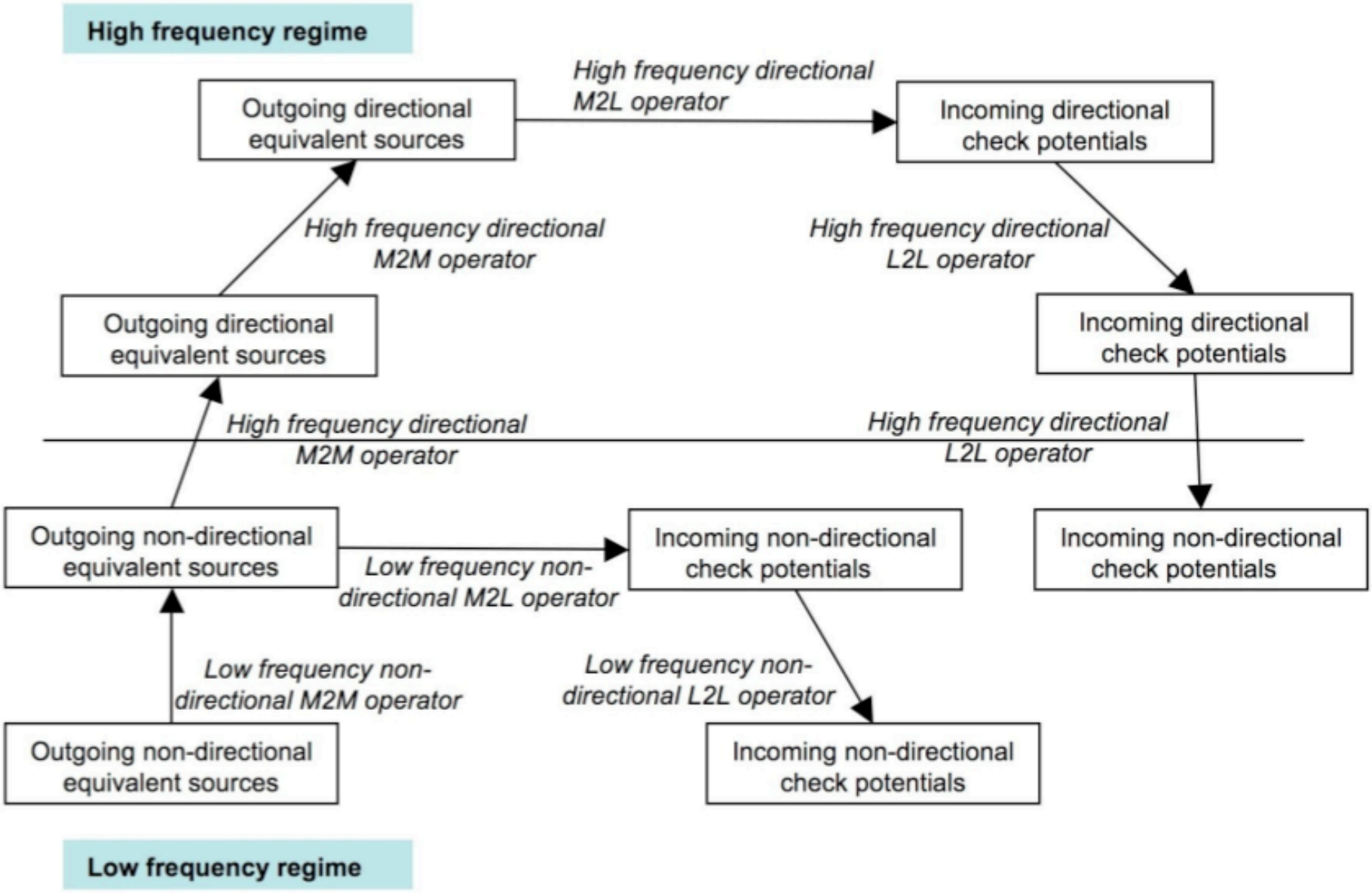


Alternative to Multipole [Anderson], [Ying]

Algorithm

1. Construct octree (hierarchy of boxes)
2. Ascend in octree. Compute near field equivalent densities
3. Ascend in octree. Compute far field equivalent densities
4. Descend in octree. Compute far field effect on potentials
5. Descend in octree. Compute near field effect on potentials




High frequency regime
Outgoing directional equivalent sources
High frequency directional M2L operator
Incoming directional check potentials
High frequency directional M2M operator
High frequency directional L2L operator
Outgoing directional equivalent sources
Incoming directional check potentials
High frequency directional M2M operator
High frequency directional L2L operator
Outgoing non-directional equivalent sources
Low frequency non-directional M2L operator
Incoming non-directional check potentials
Incoming non-directional check potentials
Low frequency non-directional M2M operator
Low frequency non-directional L2L operator
Outgoing non-directional equivalent sources
Incoming non-directional check potentials
Low frequency regime

Computational complexity

1. Construct octree (hierarchy of boxes)
   $\rightarrow$ O(N) (=O(ω^2)) operations

   Standard octree algorithm

Computational complexity

2. Ascend in octree. Compute near field equivalent densities
   $\rightarrow$ O(N)

   At most O(N) low frequency boxes. Each requires finite number of operations. From finite number of points per wave length for given accuracy

Computational complexity

3. Ascend in octree. Compute far field equivalent densities
   → O(NlogN) high frequency, compare FMM

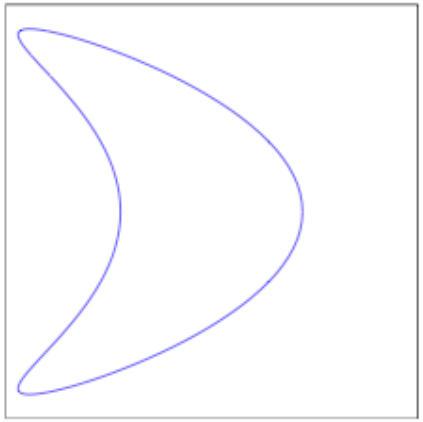
- $O(\log\omega)\ levels\ of\ boxes : w = \omega^{-1}, 2\omega^{-1}, 4\omega^{-1}, .., \omega^{-1/2},$
  $(w\ box\ side\ width)$
- $O(w^{-2})\ boxes\ at\ level\ w,\ from\ 2D - mainifold$
- $O((\omega w)^2)\ wedges/box\ at\ level\ w$
- $finite\ number\ of\ operations\ per\ wedge$

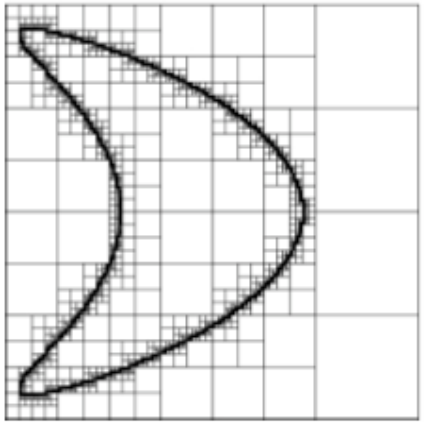
$\rightarrow complexity : O(\log\omega) \cdot O(w^{-2}) \cdot O((\omega w)^2) = O(N \log N)$

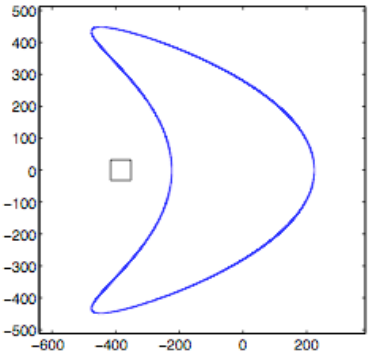
Computational complexity

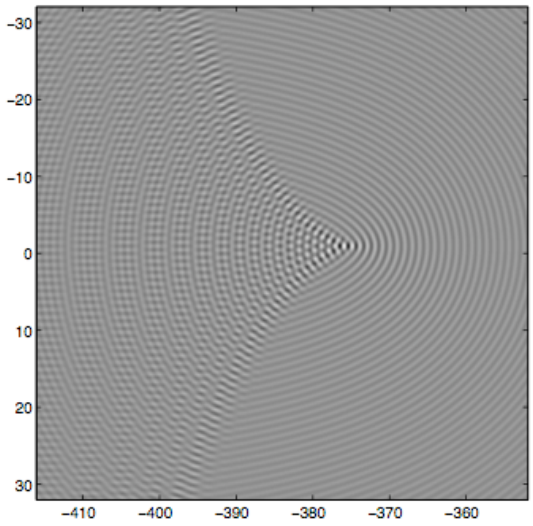
Analogous complexities when descending in octree

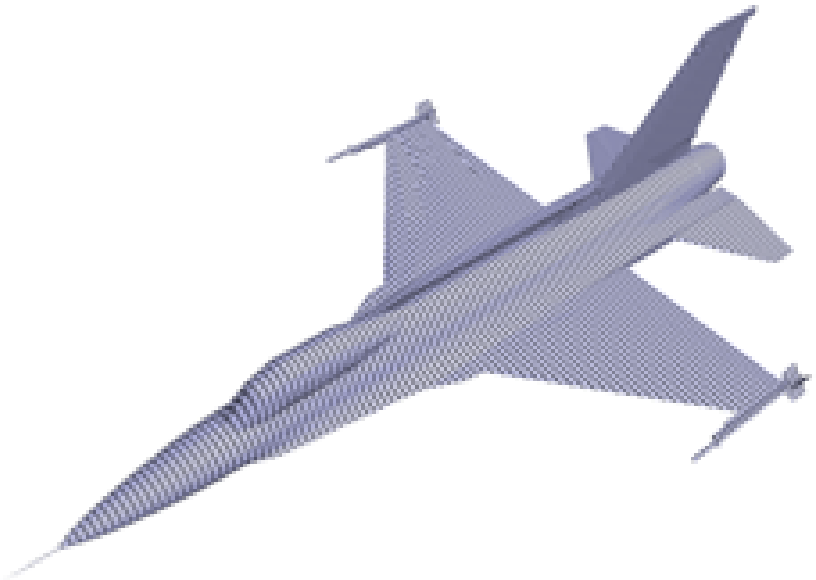
Numerical results and comments







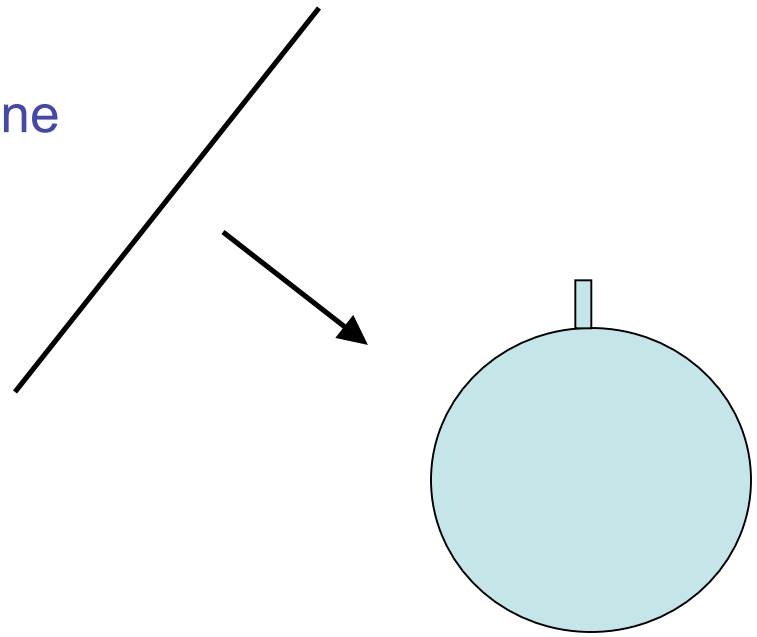




(K, ε)	N	T_a	T_d	T_d/T_a	ε_a
(32, 1e-4)	1.87e+5	5.00e+1	4.17e+3	8.34e+1	6.13e-4
(64, 1e-4)	7.46e+5	2.27e+2	6.58e+4	2.90e+2	6.69e-4
(128,1e-4)	2.98e+6	1.04e+3	1.03e+6	9.87e+2	6.89e-4
(256,1e-4)	1.19e+7	5.04e+3	1.64e+7	3.25e+3	7.63e-4
(32, 1e-6)	1.87e+5	1.18e+2	4.06e+3	3.44e+1	2.72e-6
(64, 1e-6)	7.46e+5	6.12e+2	6.56e+4	1.07e+2	3.30e-6
(128,1e-6)	2.98e+6	3.07e+3	1.06e+6	3.45e+2	4.16e-6
(32, 1e-8)	1.87e+5	2.38e+2	4.07e+3	1.71e+1	6.34e-8
(64, 1e-8)	7.46e+5	1.29e+3	6.64e+4	5.14e+1	8.10e-8
(128,1e-8)	2.98e+6	6.42e+3	1.06e+6	1.64e+2	6.55e-8

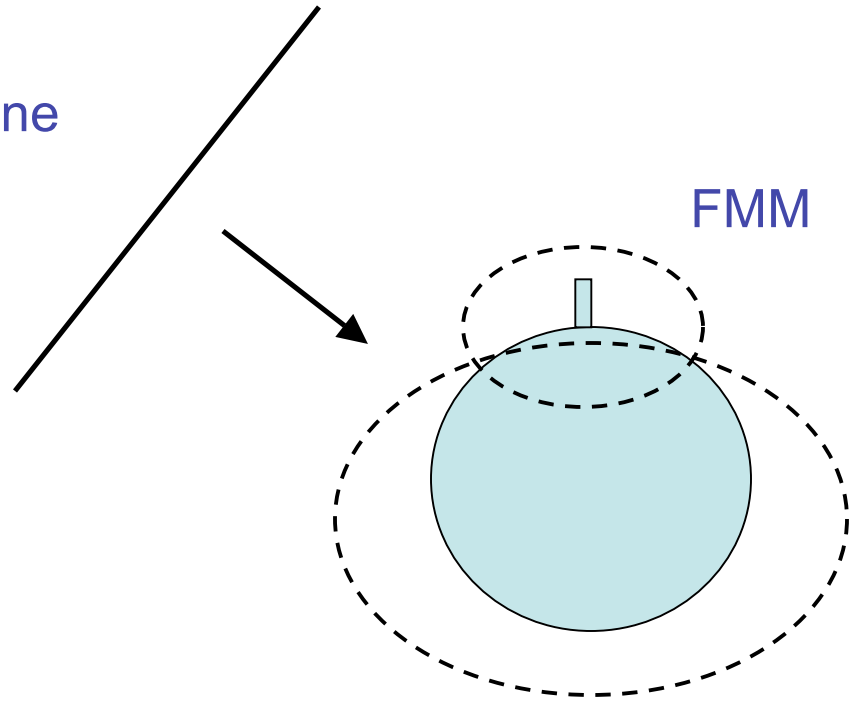
Current research: coupling

Incoming plane wave



Current research: coupling

Incoming plane wave



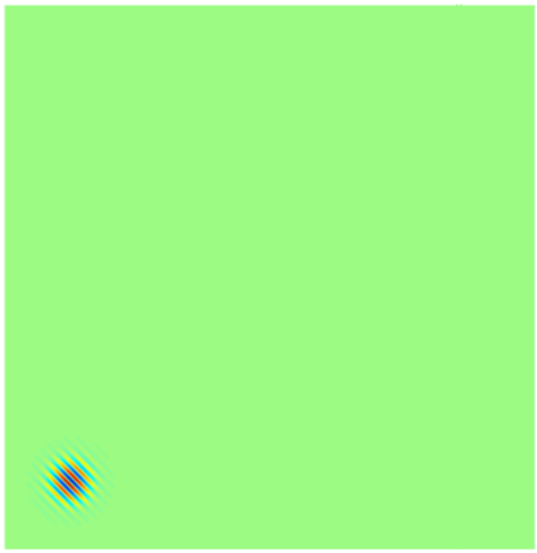
$$J = \sum_{j=1}^{J} \mu_{slow}^{j}(x) e^{i\omega\alpha(x)}$$

Asymptotic domain
[Taylor,..], [Bruno,..], [Chandler-Wilde],..

3. Fast iterative solver for the full Helmholtz equation, preconditioners (variable $c(x)$)

$$\Delta u + k(x)^2 u = f(x), \quad k(x) = \frac{\omega}{\mathrm{c(x)}}$$

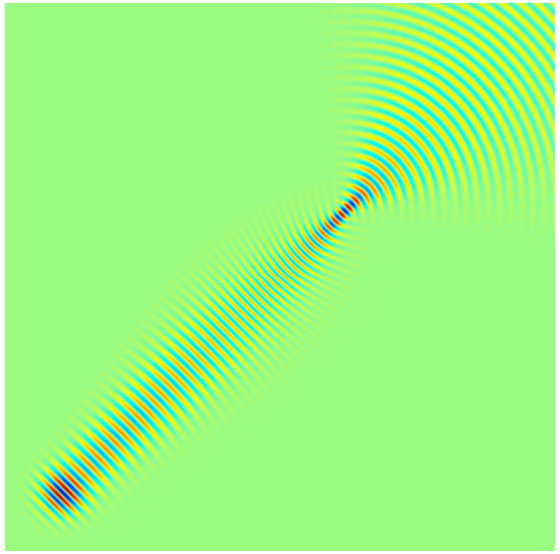
- Well-known difficulties with iterative solvers for the non-positive definite discrete Helmholtz equation
- High frequencies allows for highly oscillatory solutions with sharp signals over large domains
- Elliptic type techniques are inefficient (i.e. multigrid)



3. Fast iterative solver for the full Helmholtz equation, preconditioners (variable $c(x)$)

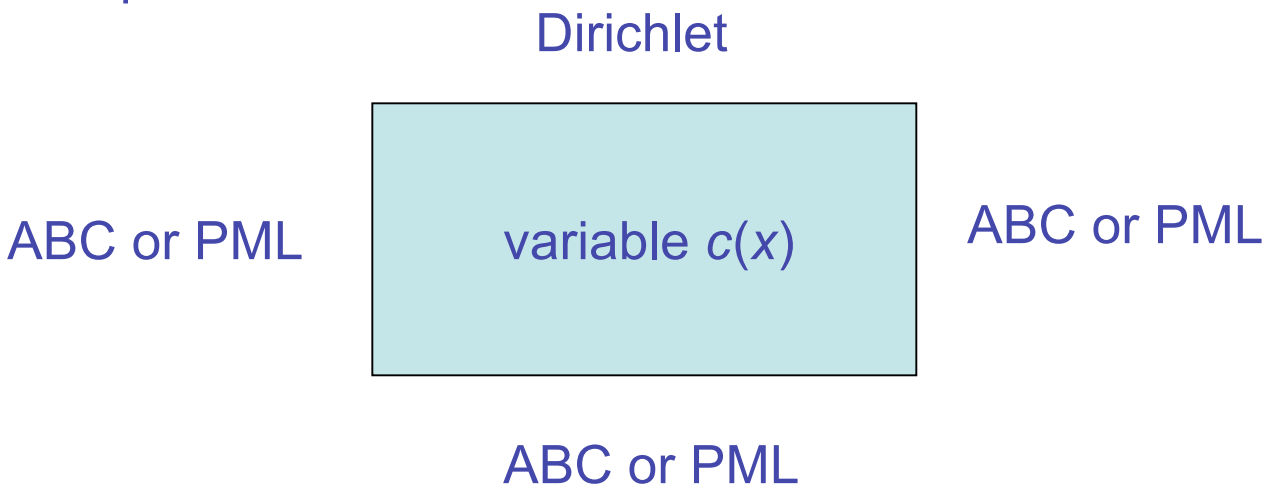
$$\Delta u + k(x)^2 u = f(x), \quad k(x) = \frac{\mathrm{w}}{\mathrm{c(x)}}$$

- Well-known difficulties with iterative solvers for the non-positive definite discrete Helmholtz equation
- High frequencies allows for highly oscillatory solutions with sharp signals over large domains
- Elliptic type techniques are inefficient (i.e. multigrid)



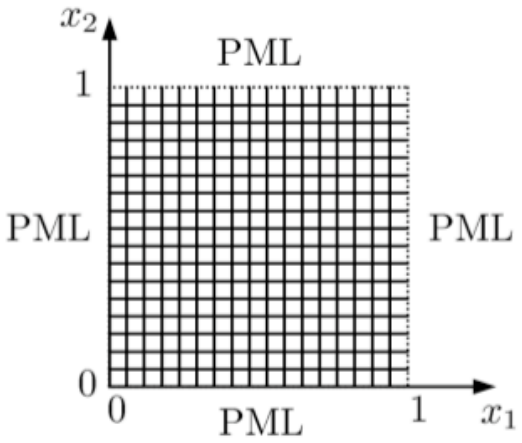
Disctretization

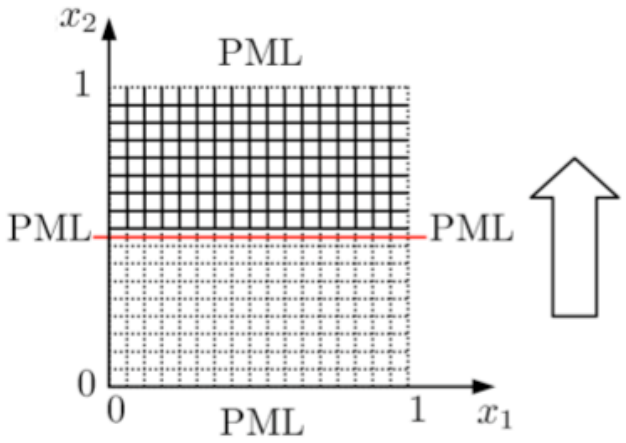
- For simplicity assume a regular grid with standard centered differences (5-point stencil in 2D and 7-point in 3D)
- Outer loop: GMRES
- The domain should have part of boundary open with with far field boundary conditions (ABC or PML)
- Example



Sweeping preconditioner

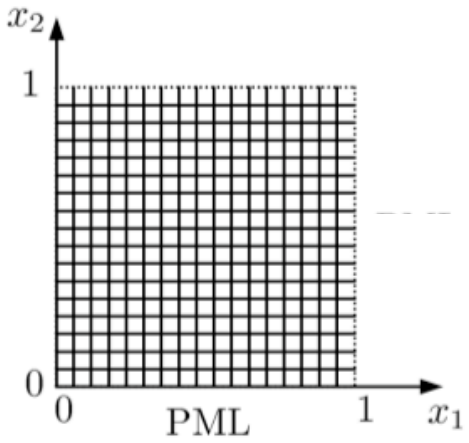
- Compare multi-frontal solver (Gaussian elimination)
- Efficient low rank representation of interaction in eliminating edges (discrete half space Green's function with PML or ABC)
- N by N unknowns (N = O(k))

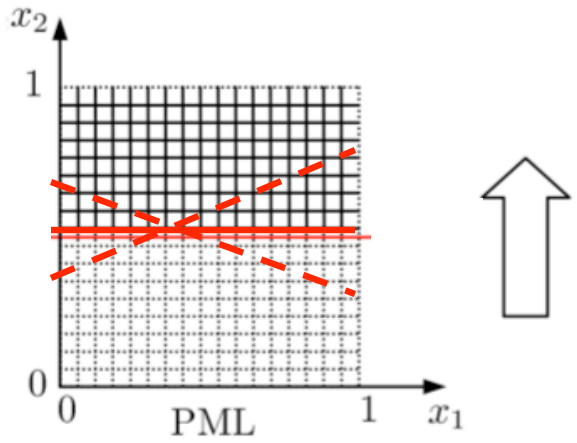




Sweeping preconditioner

- Compare multi-frontal solver (Gaussian elimination)
- Efficient low rank representation of interaction in eliminating edge (discrete half space Green's function with PML or ABC)
- N by N unknowns (N = O(k))





LDL^T factorization

- Factorization of original block tridiagonal matrix
- Submatrices stored in compressed approximate form

$$AU = F \rightarrow \begin{pmatrix} A_1 & B_1^T & 0 & \dots & \\ B_1 & A_2 & \dots & & \\ 0 & \dots & \dots & & \\ \dots & & & \dots & B_{N-1}^T \\ & & & B_{N-1} & A_N \end{pmatrix} \begin{pmatrix} u_1 \\ \\ \\ \\ u_N \end{pmatrix} = \begin{pmatrix} f_1 \\ \\ \\ \\ f_N \end{pmatrix}, \quad \left(A \; is \; N^2 \times N^2\right)$$

$$A = L_1 \begin{pmatrix} S_1 & 0 & \dots & \\ 0 & S_2 & B_2^T & \\ & B_2 & \dots & \dots \\ & & \dots & \\ & & & \end{pmatrix} L_1^T \rightarrow U = L_1^{-T} \dots L_{N-1}^{-T} \begin{pmatrix} S_N^{-1} & & & \\ & S_N^{-1} & & \\ & & \dots & \\ & & & S_N^{-1} \end{pmatrix} L_{N-1}^{-1} \dots L_1^{-1} F$$

$$S_1 = A_1, \quad S_2 = A_1 - B_1 S_1^{-1} B_1^T \dots$$

LDL^T factorization

- Without compression and with exact arithmetic we of course get the "ideal" preconditioner but at the cost O(N^6)
- Low rank compression gives the near optimal O($N^4 log(N)$)
- Low rank possible from following theorem:

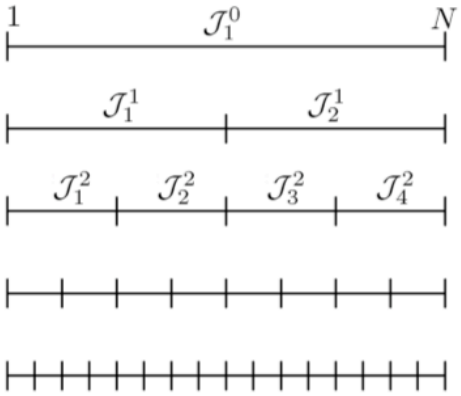
For any $\varepsilon > 0$ there exists $R = O(\varepsilon)$ s,t

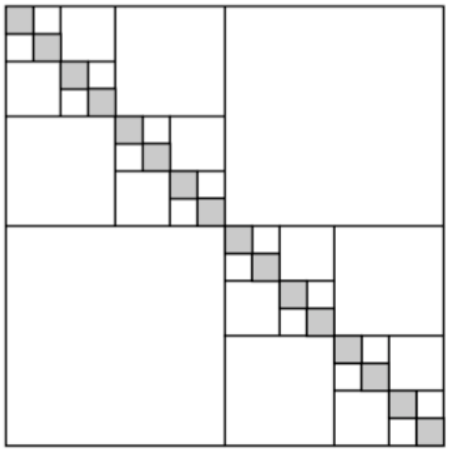
$$\left| G(x,y) - \sum_{j=1}^{R} f_j(x) g_j(y) \right| \leq \varepsilon,$$

For some f_j, g_j, and any $k > 0$, G is the half space Green's function and $a < x < s < y < b$.

Matrix representation and algebra

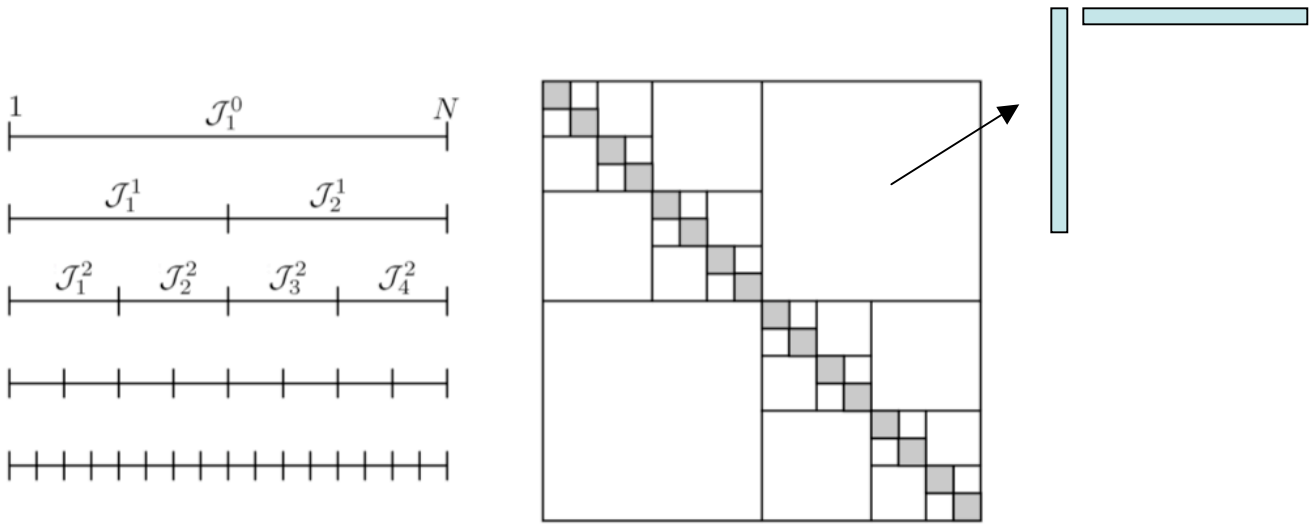
- *H* - matrix style representation with weak admissibility condition for the sub blocks (*S* and *B*), compare [Hackbusch]





Matrix representation and algebra

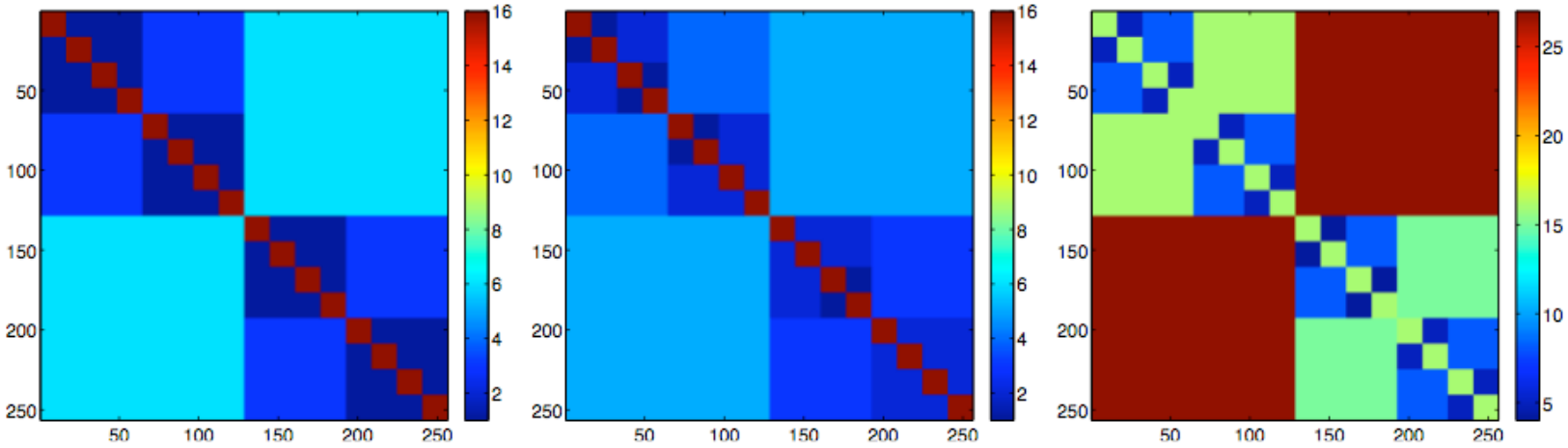
- *H* - matrix style representation with weak admissibility condition for the sub blocks (*S* and *B*)



- Not just matrix-vector multiply as in FMM

Matrix representation and algebra

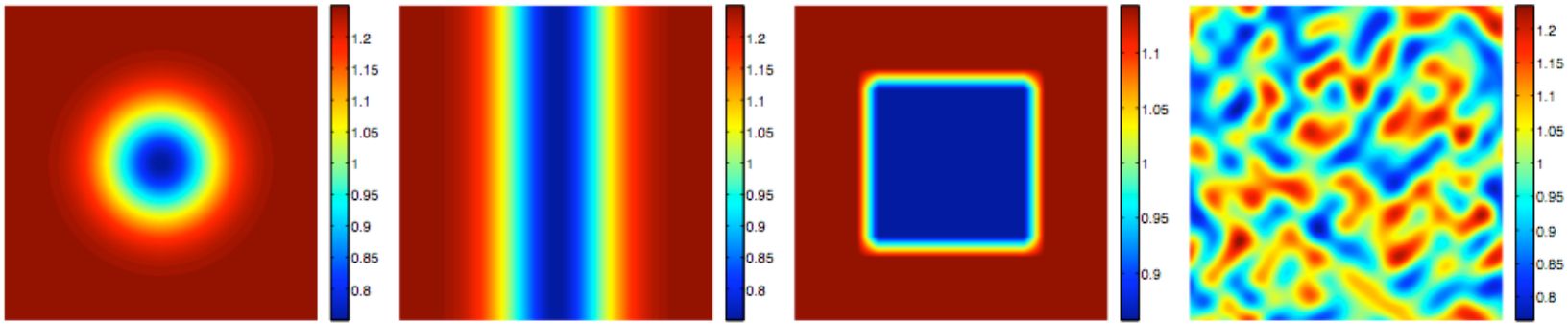
- *H* - matrix style representation with weak admissibility condition for the sub blocks (*S* and *B*). Computed rank
- SVD in 2D, randomized SVD in 3D



Constant c Variable c Dirichlet bc

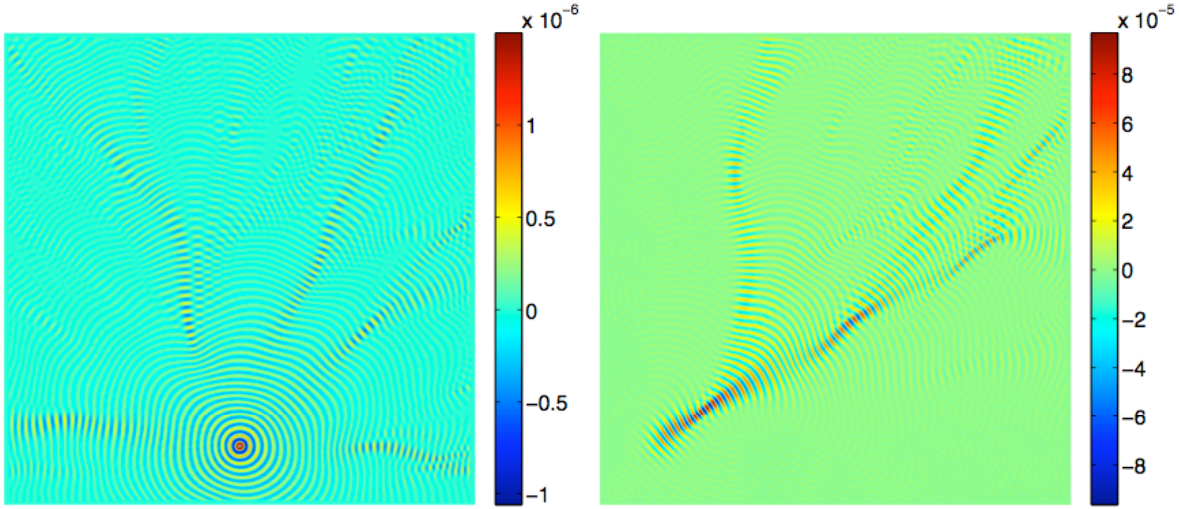
Computational results

- Variable coefficients - examples



Computational results

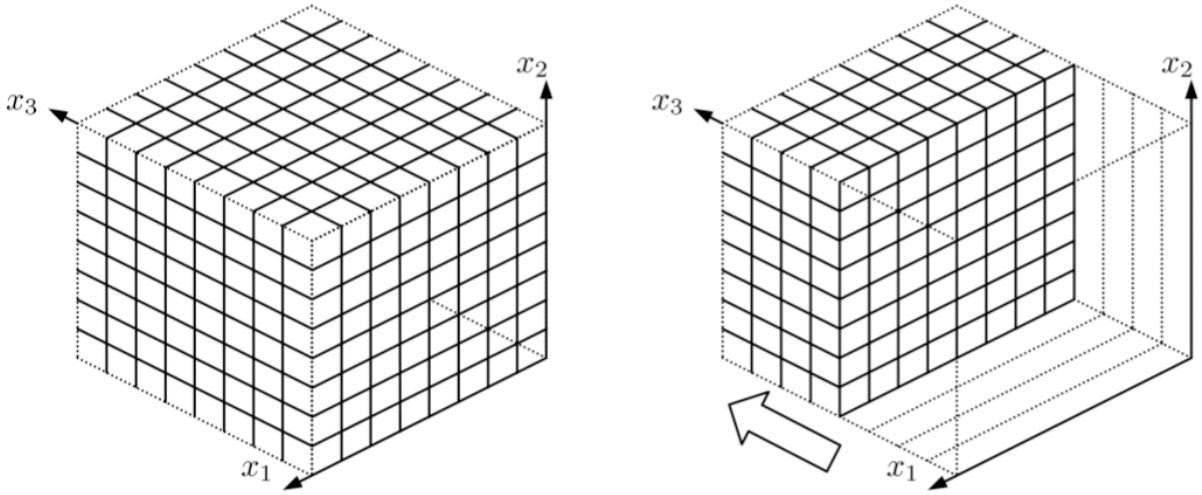
- Test 1 and 2: different source functions (random c)



				Test 1		Test 2	
$\omega/(2\pi)$	N	R	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{iter}
16	128	2	2.80e-01	3	4.00e-02	3	5.00e-02
32	256	2	1.47e+00	4	2.60e-01	4	2.60e-01
64	512	3	1.03e+01	4	1.32e+00	4	1.32e+00
128	1024	3	5.77e+01	5	6.96e+00	5	6.86e+00
256	2048	4	4.67e+02	5	3.36e+01	6	3.83e+01

3D: incomplete but promising

- Similar sweeping structure, face by face



3D: incomplete but promising

- Hierarchical matrix representation more complex
- Lack of compatible low rank theorem. The FMM low rank theorem applies but does not give near optimal complexity
- Near optimal result with 8 points/wave, (4,000,000 unknowns)

				Test 1		Test 2	
$\omega/(2\pi)$	N	R	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{iter}
5	40	2	4.07e+01	3	5.40e-01	4	6.60e-01
10	80	3	9.70e+02	9	1.59e+01	7	1.27e+01
20	160	4	1.83e+04	6	1.14e+02	6	1.13e+02

Conclusions

- High frequency FMM fully based on approximate low rank interaction
  - Directional decomposition
  - Rigorous foundation for approximate decomposition
  - Efficient randomized algorithm

- Efficient sweeping preconditioning for iterative solutions of the high frequency Helmholtz equation