Sparse Days -CERFACS 6/1/2016

Hierarchical Probing for General Graphs, a method for computing diag(f(A))

Andreas Stathopoulos and Jesse Laeuchli

Computer Science Department
College of William and Mary

Acknowledgment: NSF, DOE SciDAC

The problem

Given a large, $N \times N$ matrix A and a function f

$\boxed{\text{find } \mathrm{diag}(f(A))}$

$\boxed{\text{Related problem: find trace of } f(A)\text{: } \mathbf{Tr}(f(A))}$

Recall: $\mathbf{Tr}(A) = \sum_{i=1}^{N} A_{ii} = \sum_{i=1}^{N} \lambda_i(A)$

$\boxed{\text{Constraint: factorization methods prohibitive}}$

The problem

Common functions $f(A) : A^{-1},\ \log(A),\ \exp(A),\ R_i^T A^{-1} R_j,\ A^k,\ \ldots$

Applications:

- Graph/Data Mining/Uncertainty Quantification
  - Counting loops in network analysis
  - Node centralities are diagonals of $\exp(A)$
  - Diagonal of inverse of covariance matrices measures confidence in data
- Quantum Monte Carlo
  - Stopping criterion based on ratio of determinants, $\det(A) = \exp(\mathbf{Tr}(\log(A)))$
- Lattice QCD
  - $\mathbf{Tr}(A^{-1})$ needed for sequences of matrices

Our focus: $f(A) = A^{-1}$ but techniques general

Basic method

Standard approach: (Hutchinson, a Monte Carlo method)

If x is a vector of random Z_2 variables

$$x_i = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$$

then

$$\boxed{E(x \odot A^{-1}x) = \mathrm{diag}(A^{-1})}$$

```
for i=1:$n$
    $x$ = randi($N$,1);
    Sum = Sum + $x \odot A^{-1} x$
    diag = Sum / i;
end
```

Basic method

Standard approach: (Hutchinson, a Monte Carlo method)

If x is a vector of random Z_2 variables

$$x_i = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$$

then

$$\boxed{E(x^T A^{-1} x) = \mathbf{Tr}(A^{-1})}$$

```
for i=1:$n$
    $x$ = randi($N$,1);
    sum = sum + $x^T A^{-1} x$
    trace = sum / i;
end
```

$$t(A^{-1}) = \tfrac{1}{n} \textstyle\sum_{j=1}^{n} x_j^H A^{-1} x_j$$

Basic method

Standard approach: (Hutchinson, a Monte Carlo method)

If x is a vector of random Z_2 variables

$$x_i = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$$

then

$$\boxed{E(x^T A^{-1} x) = \mathbf{Tr}(A^{-1})}$$

```
for i=1:$n$
    $x$ = randi($N$,1);
    sum = sum + $x^T A^{-1} x$
    trace = sum / i;
end
```

$$t(A^{-1}) = \frac{1}{n}\sum_{j=1}^{n} x_j^H A^{-1} x_j$$

Our goal: Reduce Variance

Variance of the Hutchinson estimator

The "squared error" of the statistical estimator $t(A^{-1})$ is its variance

$$\boxed{Var(t(A^{-1})) = \frac{2}{n}\|\tilde{A}^{-1}\|_F^2 = \frac{2}{n}\left(\|A^{-1}\|_F^2 - \sum_{i=1}^{L}|A_{i,i}^{-1}|^2\right)}$$

where $\tilde{A}^{-1} = A^{-1} - diag(diag(A^{-1}))$

Variance of the Hutchinson estimator

The “squared error” of the statistical estimator $t(A^{-1})$ is its variance

$$\boxed{Var(t(A^{-1})) = \frac{2}{n}\|\tilde{A}^{-1}\|_F^2 = \frac{2}{n}\left(\|A^{-1}\|_F^2 - \sum_{i=1}^{L}|A_{i,i}^{-1}|^2\right)}$$

where $\tilde{A}^{-1} = A^{-1} - diag(diag(A^{-1}))$

Thus, the goal of variance reduction:

$$\boxed{\text{remove weight from the off-diagonals elements of } A^{-1}}$$

Variance of the Hutchinson estimator

The “squared error” of the statistical estimator $t(A^{-1})$ is its variance

$$\boxed{Var(t(A^{-1})) = \frac{2}{n}\|\tilde{A}^{-1}\|_F^2 = \frac{2}{n}\left(\|A^{-1}\|_F^2 - \sum_{i=1}^{L} |A_{i,i}^{-1}|^2\right)}$$

where $\tilde{A}^{-1} = A^{-1} - diag(diag(A^{-1}))$

Thus, the goal of variance reduction:

$$\boxed{\text{remove weight from the off-diagonals elements of } A^{-1}}$$

- Approximate $M \approx A^{-1}$, $\mathbf{Tr}(A^{-1}) = \mathbf{Tr}(M) + \mathbf{Tr}(A^{-1} - M)$
  hope that $t(A^{-1} - M)$ has smaller variance
- Choose vectors that remove particular patterns of A^{-1}

Selecting the vectors in $x^T A^{-1} x$

Random

$x \in Z_2^N$	best variance for real matrices (Hutchinson 1989)
$x = \text{randn}(N,1)$	worse variance than Z_2
$x = e_i$	variance depends only on $\text{diag}(A^{-1})$
	single large element?
$x = F^T e_i$	mixing of diagonal elements: (Toledo et al. 2010)
	F DFT — Best mixing but complex
	H Hadamard — Best mixing and real

Deterministic

$x = H^T e_i, \; {\scriptstyle i=1,\ldots,2^k}$	Hadamard in natural order (Bekas et al. 2007)
$x_i^m = \begin{cases} 1 & i \in C_m \\ 0 & \text{else} \end{cases}$	Probing. Assumes multicolored graph (Tang et al. 2011)

Deterministic more effective if they could capture matrix structure

Hadamard vectors in natural order

$$H_1 = 1, \quad H_{2^{k+1}} = \begin{bmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{bmatrix}$$

$\mathbf{Tr}(A^{-1}) \approx \frac{1}{m}\mathbf{Tr}(h^T A^{-1} h)$, where $h =$`H(:,1:`m`)` the first $m = 2^k$ vectors

Eliminates error contribution of all diagonals of A except $1 + i \cdot m, \; i = 0, 1, 2, \ldots$

e.g., for $m = 4$,

$$\frac{1}{4}hh^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \\ & & & & \vdots & & & & \end{bmatrix}$$

no contribution from A^{-1}_{ij} elements corresponding to a 0

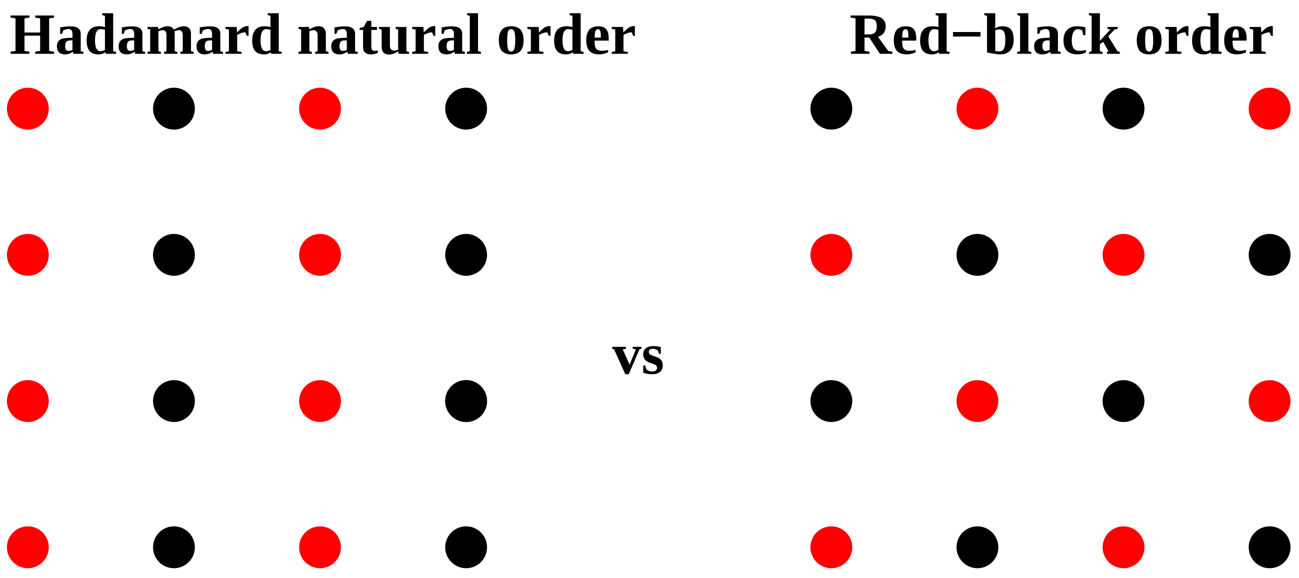
Hadamard vectors in natural order

1-D grid: $m = 2$ Hadamard vectors sufficient for $\mathbf{Tr}(A)$
2-D grid: $m = 2$ natural order does not align with diagonals of A

But two different Hadamard vectors align.

Given `ip`, the inverse red-black permutation, `h = H(ip,[1,` $\frac{N}{2}$`])`

```
1    1
1   -1
1    1
1   -1
1    1
1   -1
1    1
1   -1
1    1
1   -1
1    1
1   -1
1    1
1   -1
1    1
1   -1
```



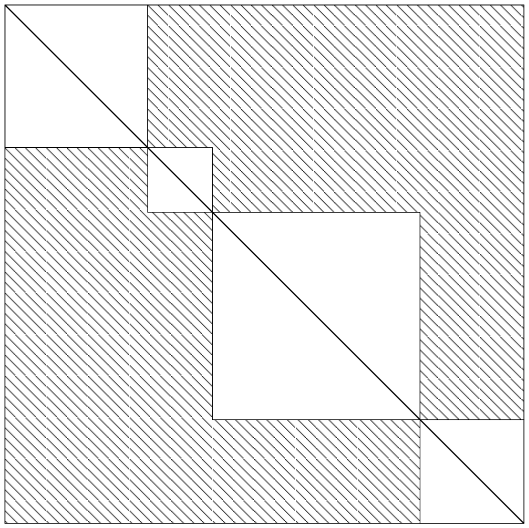
```
1    1
1   -1
1    1
1   -1
1   -1
1    1
1   -1
1    1
1    1
1   -1
1    1
1   -1
1   -1
1    1
1   -1
1    1
```

Probing, Green's function, and $\mathbf{A^{-1}}$

Probing:

The trace of an m-colorable sparse matrix is recovered exactly by m vectors

$$\mathbf{Tr}(A) = \mathbf{Tr}(h^T A h)$$



$$\begin{matrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Green's function

Elements of A^{-1} decay in magnitude away from the non-zero structure of A

Probing for $\mathbf{Tr}(A^{-1})$

Color $A^k \equiv$ distance-k coloring of A. Captures largest elements of A^{-1}

Problems:

(1) If $\mathbf{Tr}(A^{-1})$ not accurate enough, discard work and repeat for larger k

(2) Coloring expensive for large k

A hierarchical probing method

To avoid discarding work, previous quadrature vectors must be contained within the subspace of the vectors of the new colors:

$$\begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix} \not\subset \begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix} \quad \textbf{but} \quad \begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix} \subset \begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix}$$

Colorings for successive distances must be nested

A hierarchical probing method

To avoid discarding work, previous quadrature vectors must be contained within the subspace of the vectors of the new colors:

$$\begin{matrix}1&0&0\\1&0&0\\0&1&0\\0&1&0\\0&0&1\\0&0&1\end{matrix} \not\subset \begin{matrix}1&0&0&0\\0&1&0&0\\0&1&0&0\\0&0&1&0\\0&0&1&0\\0&0&0&1\end{matrix} \quad \textbf{but} \quad \begin{matrix}1&0&0\\1&0&0\\0&1&0\\0&1&0\\0&0&1\\0&0&1\end{matrix} \subset \begin{matrix}1&0&0&0\\1&0&0&0\\0&1&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix}$$

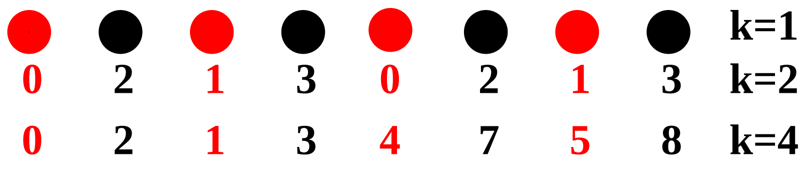
Colorings for successive distances must be nested

Since, $\begin{bmatrix} e & 0 \\ 0 & e \end{bmatrix} \in < \begin{bmatrix} e & e \\ e & -e \end{bmatrix} >$, with $e = [1, \ldots, 1]^T$, no probing vectors needed.

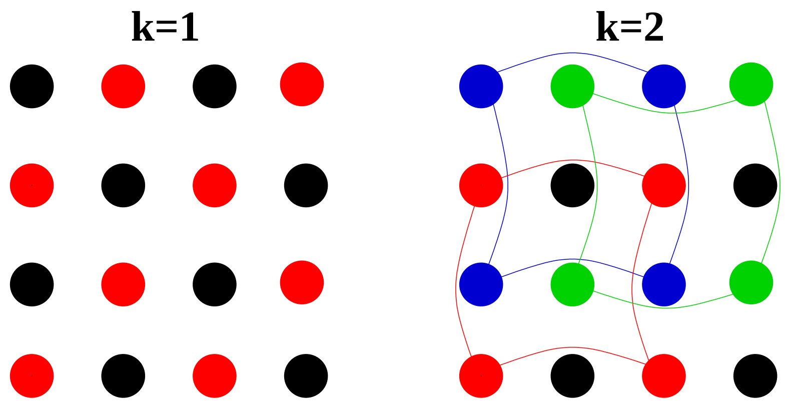
More colors means adding selected Hadamard vectors

Hierarchical coloring on the lattice (JL/AS 2014)

Hierarchy $\Rightarrow$ distance $k = 2^m$



1D:
Doubling k splits grid to 2 1D subgrids
Color R-B each with two new colors



2D:
Doubling k splits grid to 4 2D subgrids
(e.g., Reds split to 4 reds and 4 greens)
Color R-B each with two new colors

Algorithm in d-dimensions:
1. Recursively split a lattice $\prod_{i=1}^{d} 2^{c_i}$ to 2^d sublattices of size $\prod_{i=1}^{d} 2^{c_i-1}$
2. When no further splits possible, Red-Black each sublattice with unique colors.
3. Choose unique colors appropriately to guarantee nesting

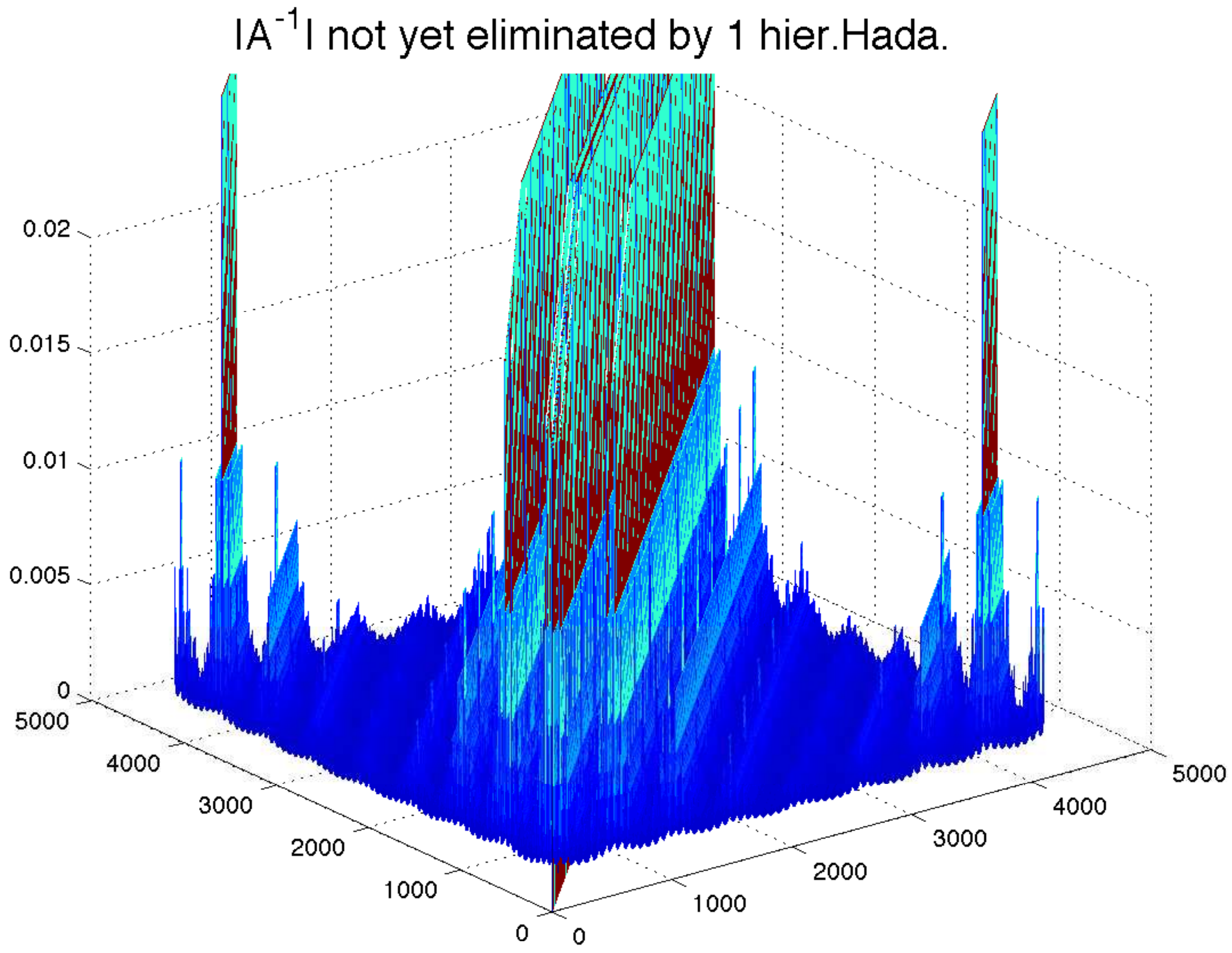
Hierarchical coloring on the lattice (JL/AS 2014)

Properties:

- $2k^d$ colors at distance k, but taking k to max dist produces 1 permutation that contains all previous colorings
- Efficient bit-arithmetic algorithm $O(N\log(N)/d)$
- No additional memory
- Used incrementally with powers of 2 of Hadamard vectors
- Permutation and generation of Hadamard naturally parallel on each site

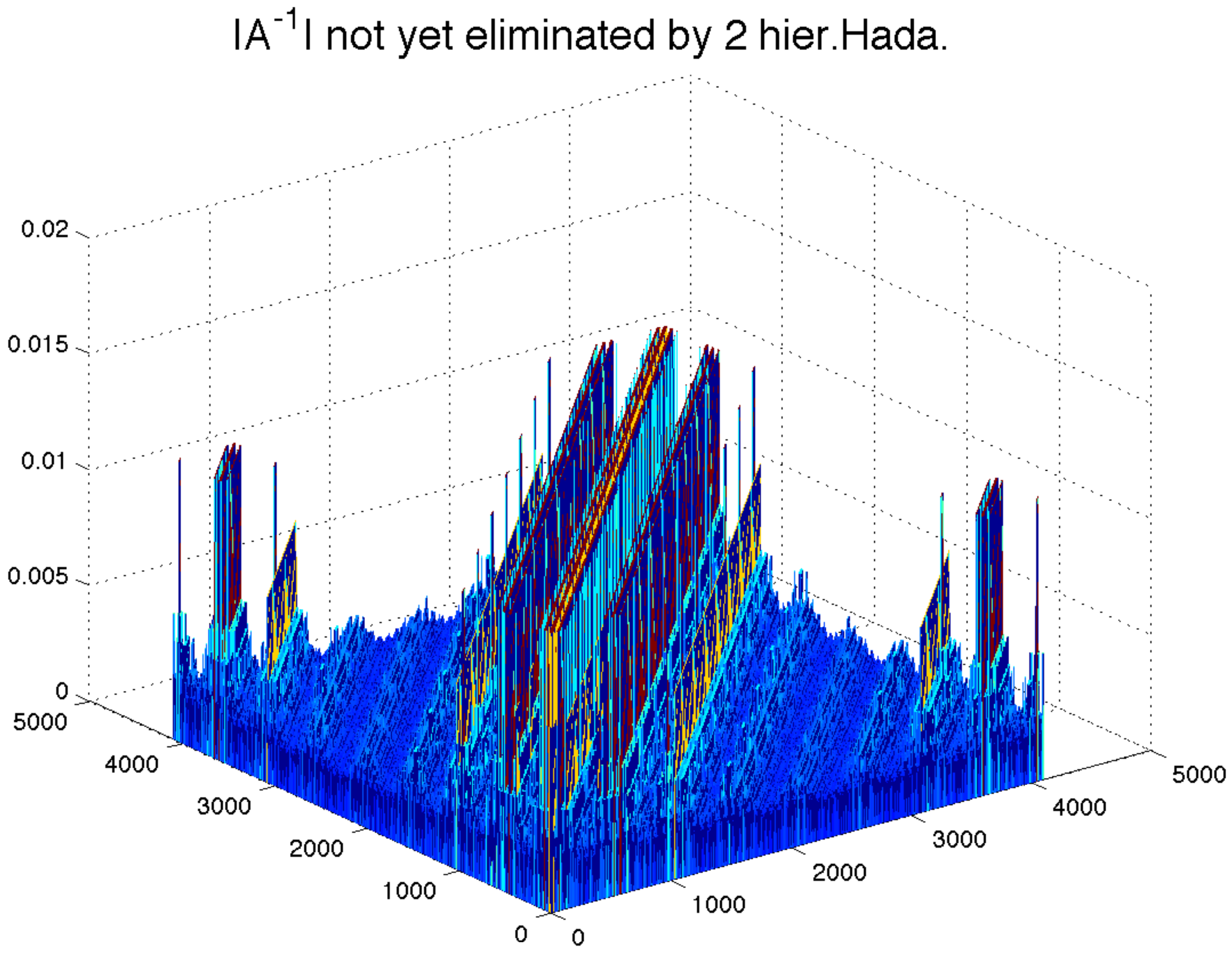
$\boxed{\text{Resolved the two problems of probing}}$

Hierarchical Hadamard eliminates largest elements first



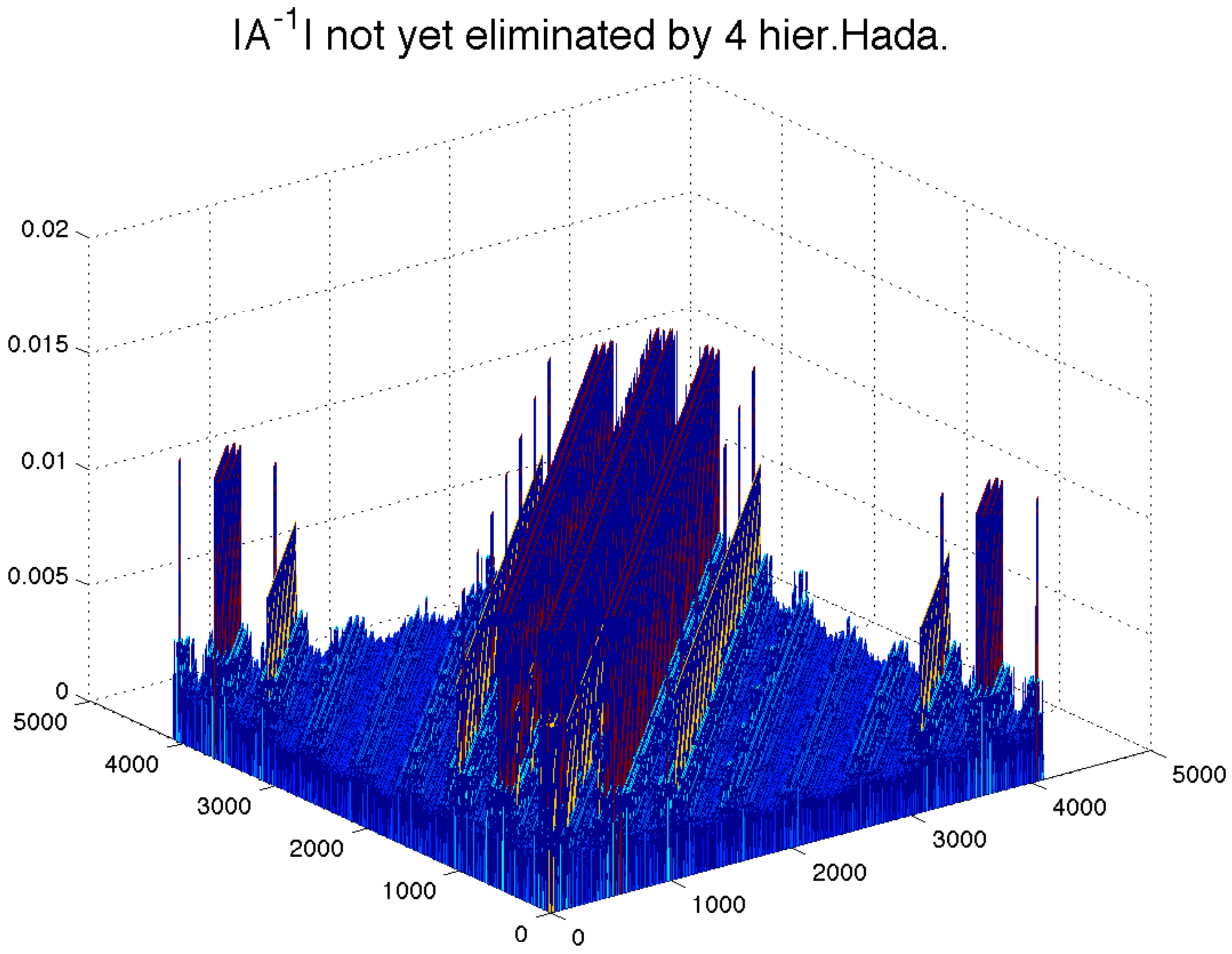
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times8\times8\times8$

Hierarchical Hadamard eliminates largest elements first



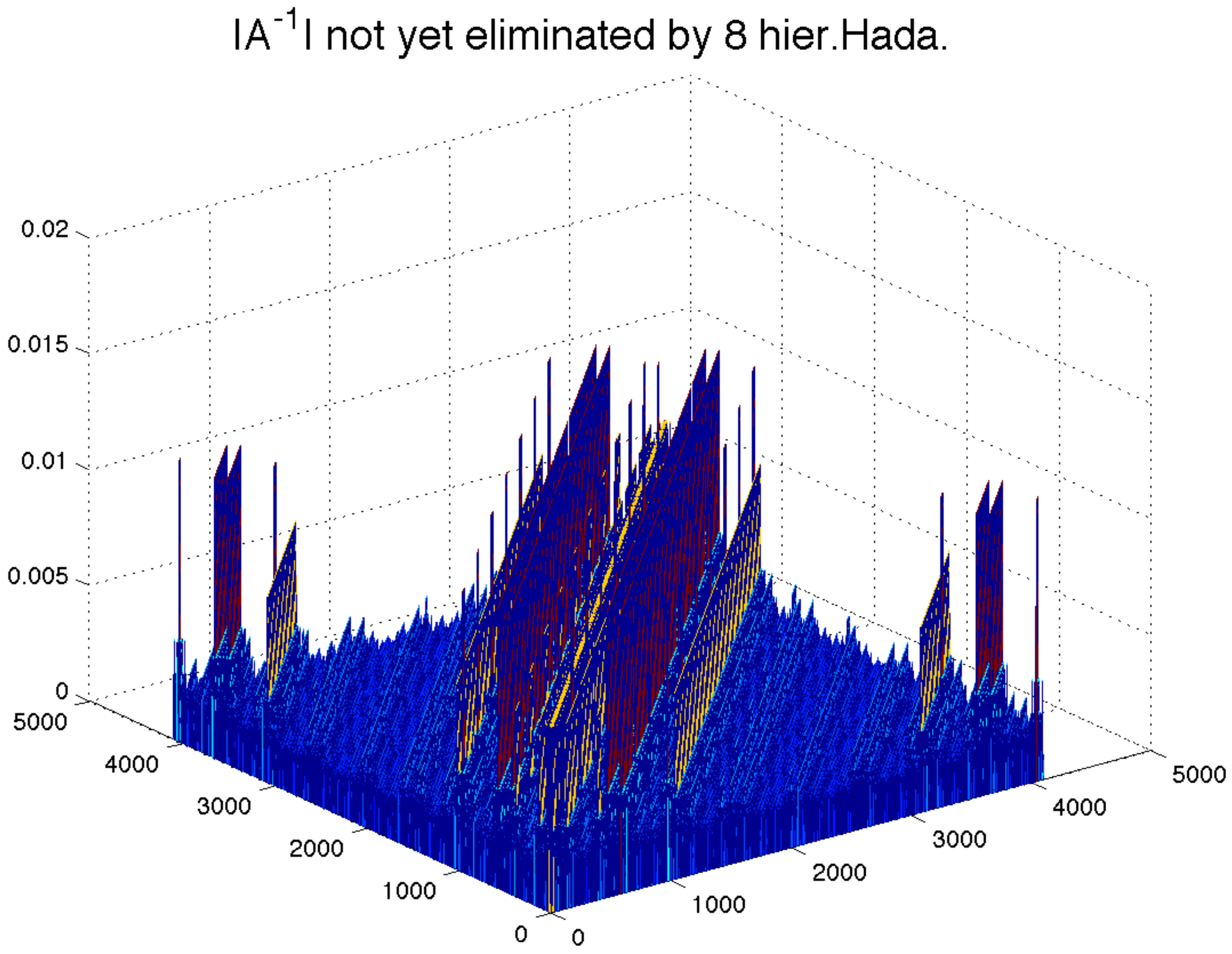
$(A + 0.1I)^{-1}$ Laplacian with periodic BC, $8 \times 8 \times 8 \times 8$

Hierarchical Hadamard eliminates largest elements first



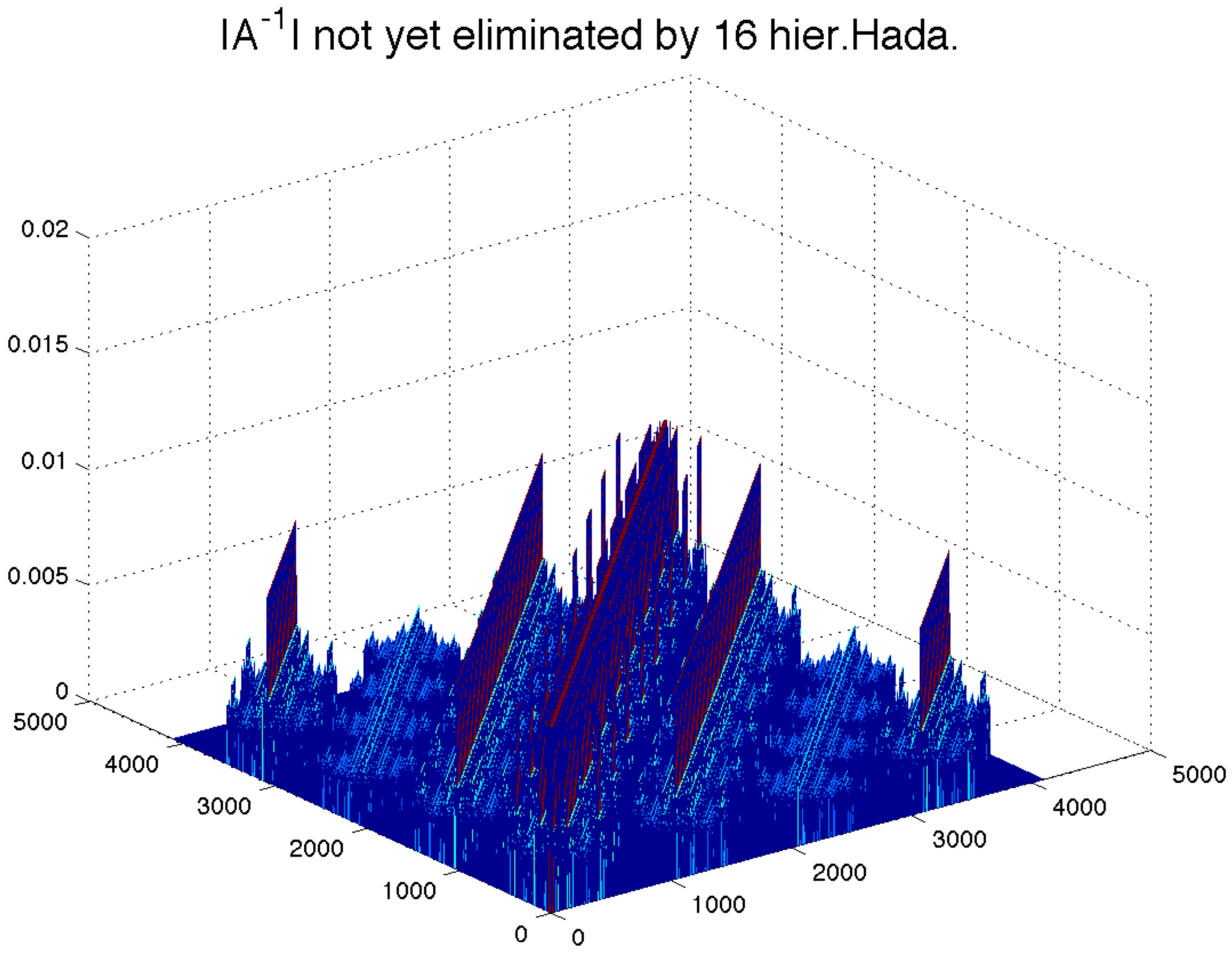
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8 \times 8 \times 8 \times 8$

Hierarchical Hadamard eliminates largest elements first



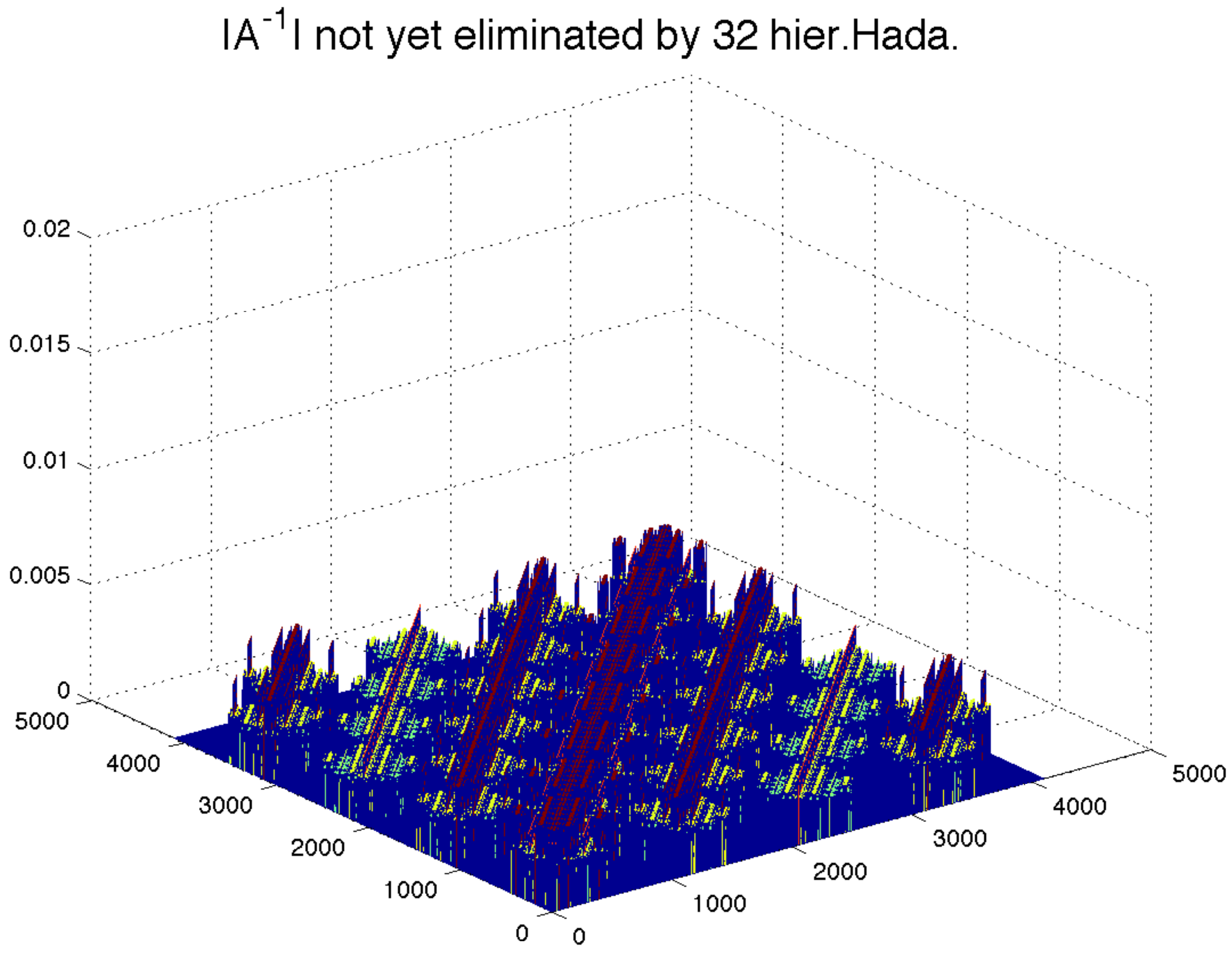
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8 \times 8 \times 8 \times 8$

Hierarchical Hadamard eliminates largest elements first



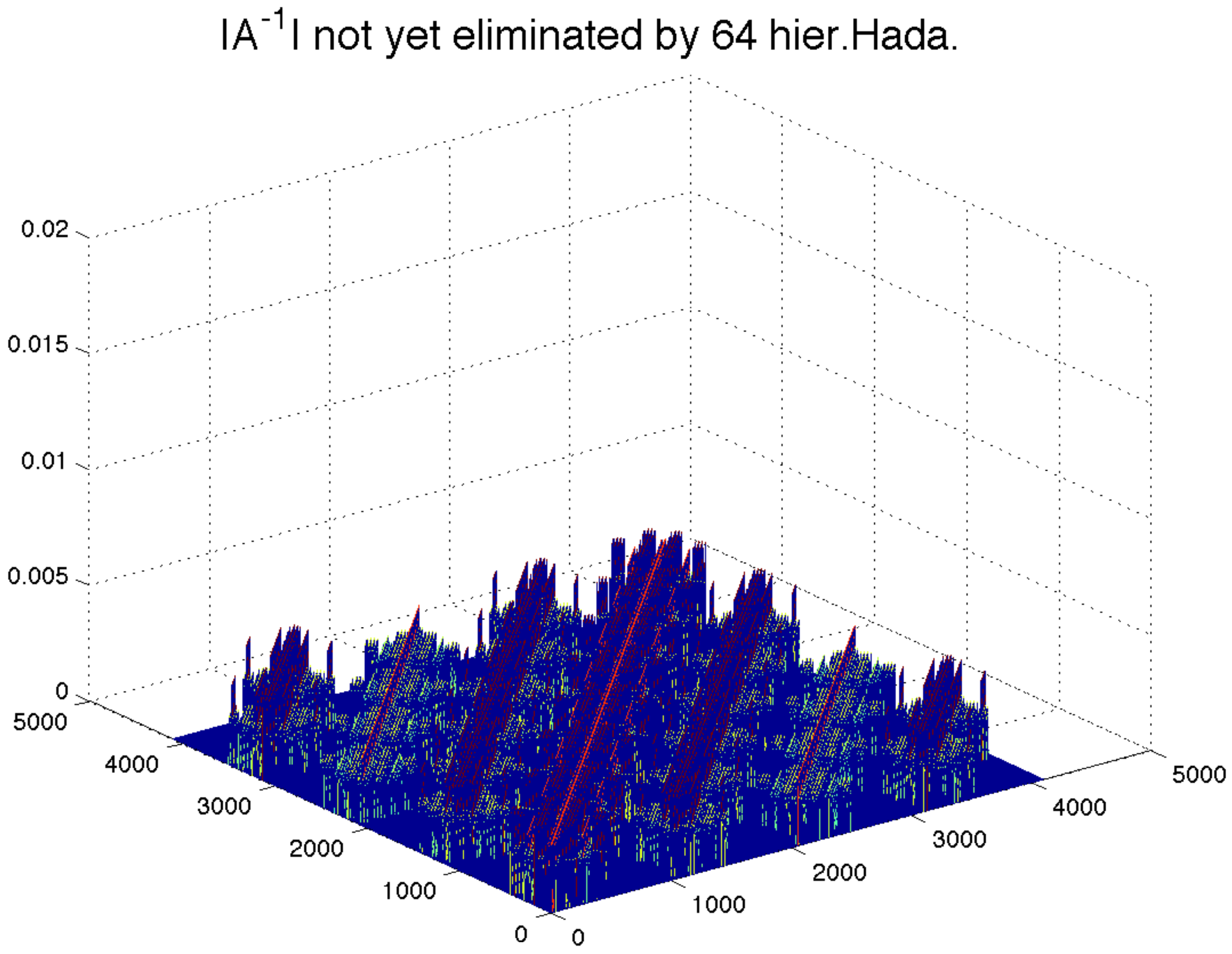
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times8\times8\times8$

Hierarchical Hadamard eliminates largest elements first



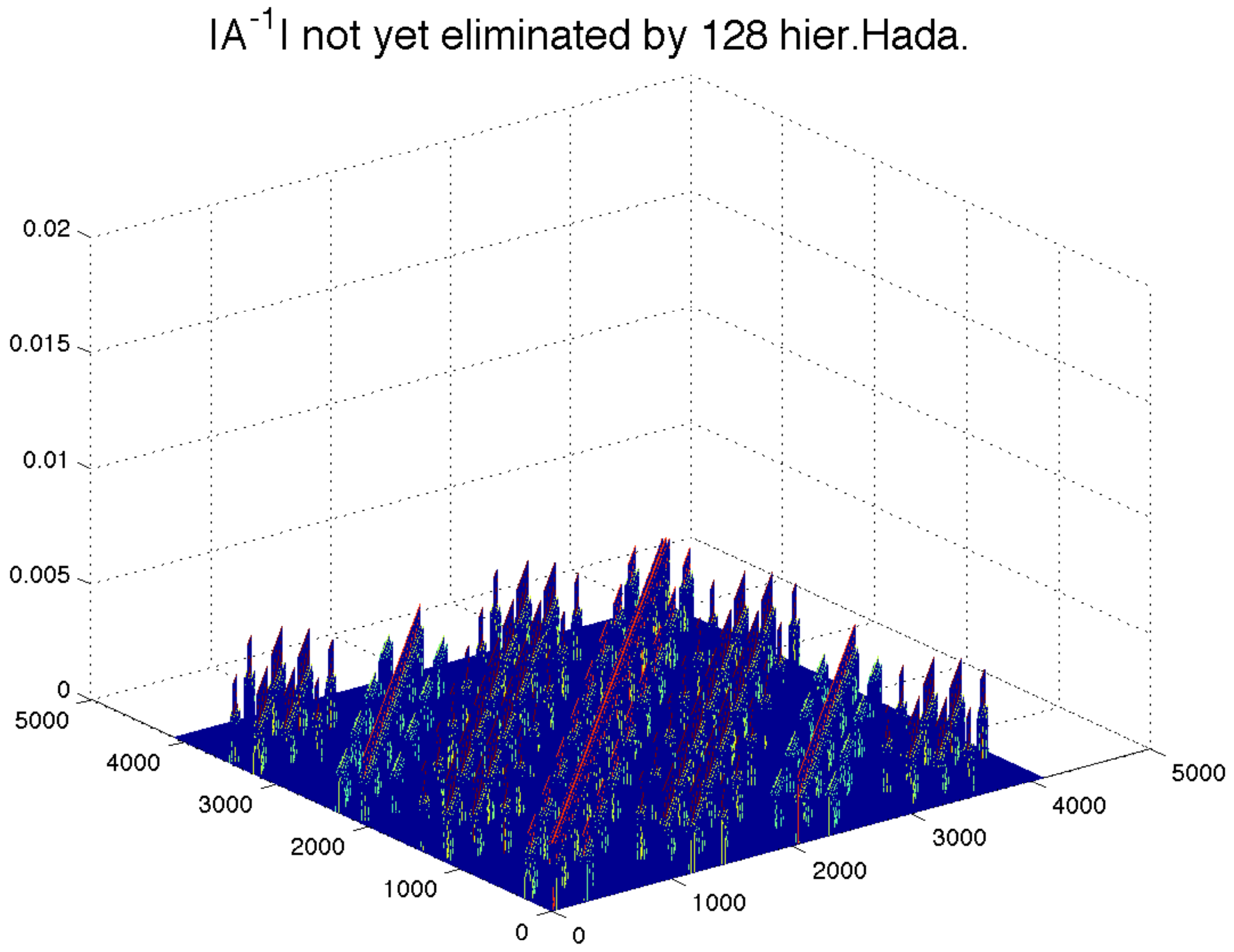
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times8\times8\times8$

Hierarchical Hadamard eliminates largest elements first



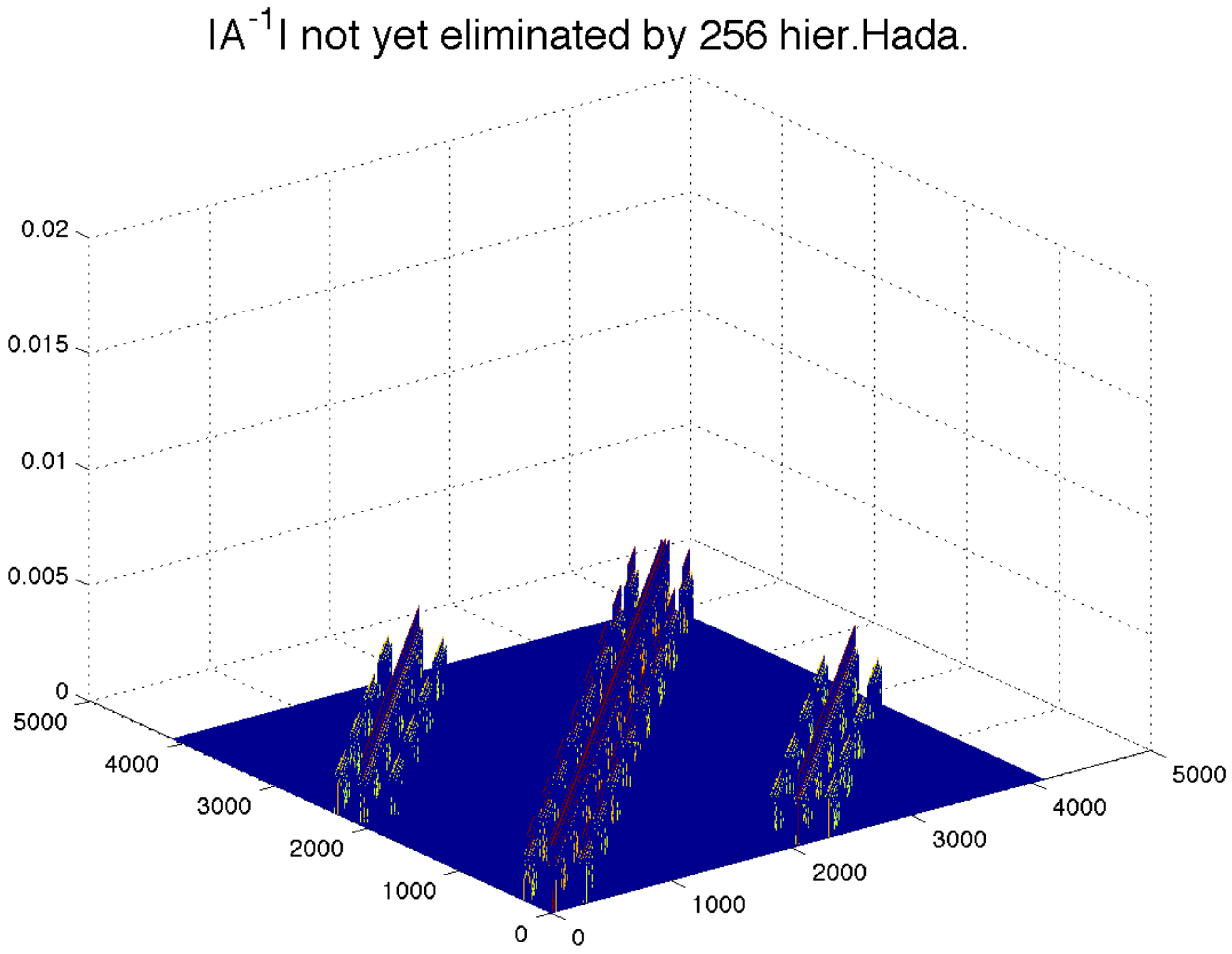
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8 \times 8 \times 8 \times 8$

Hierarchical Hadamard eliminates largest elements first



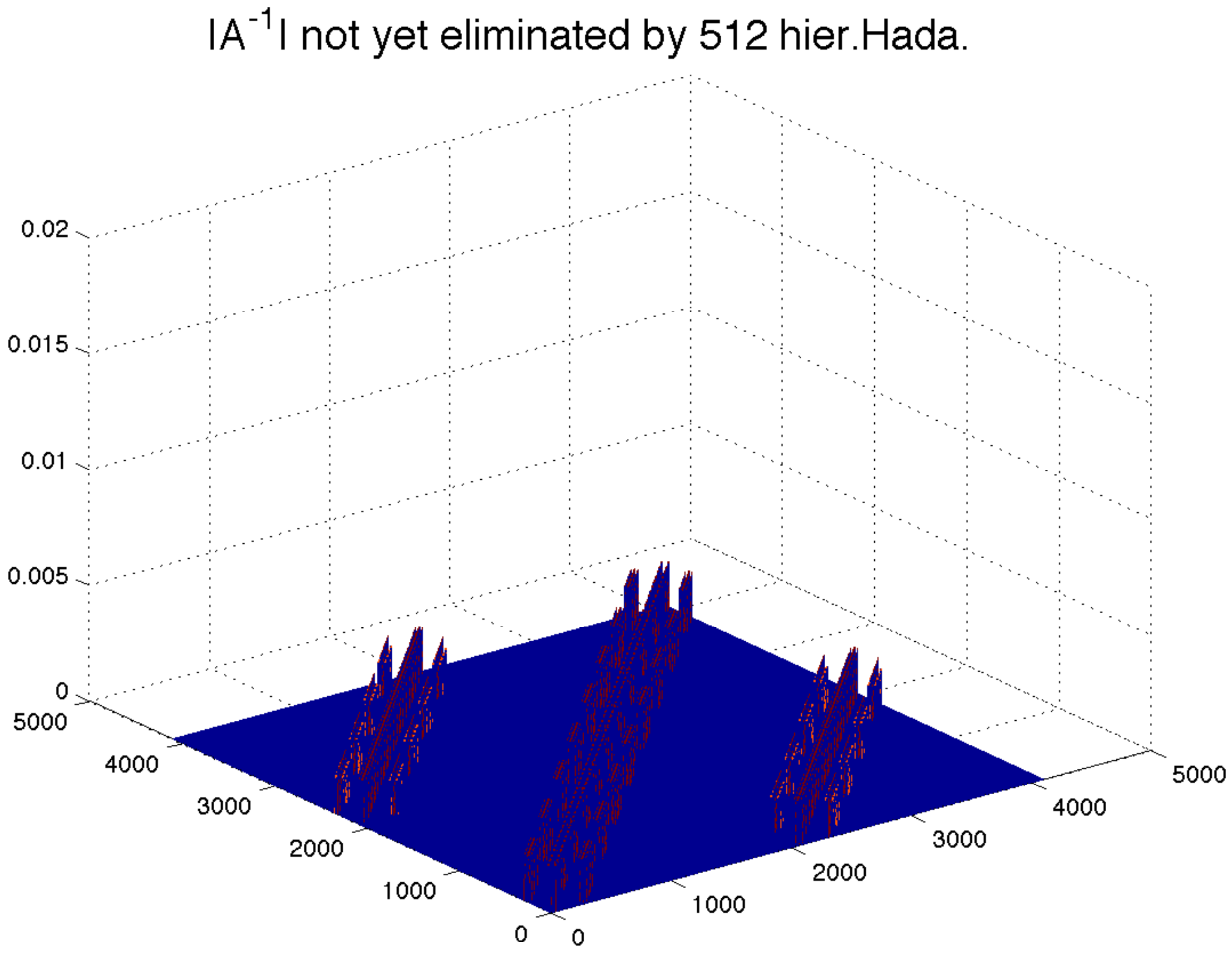
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times 8\times 8\times 8$

Hierarchical Hadamard eliminates largest elements first



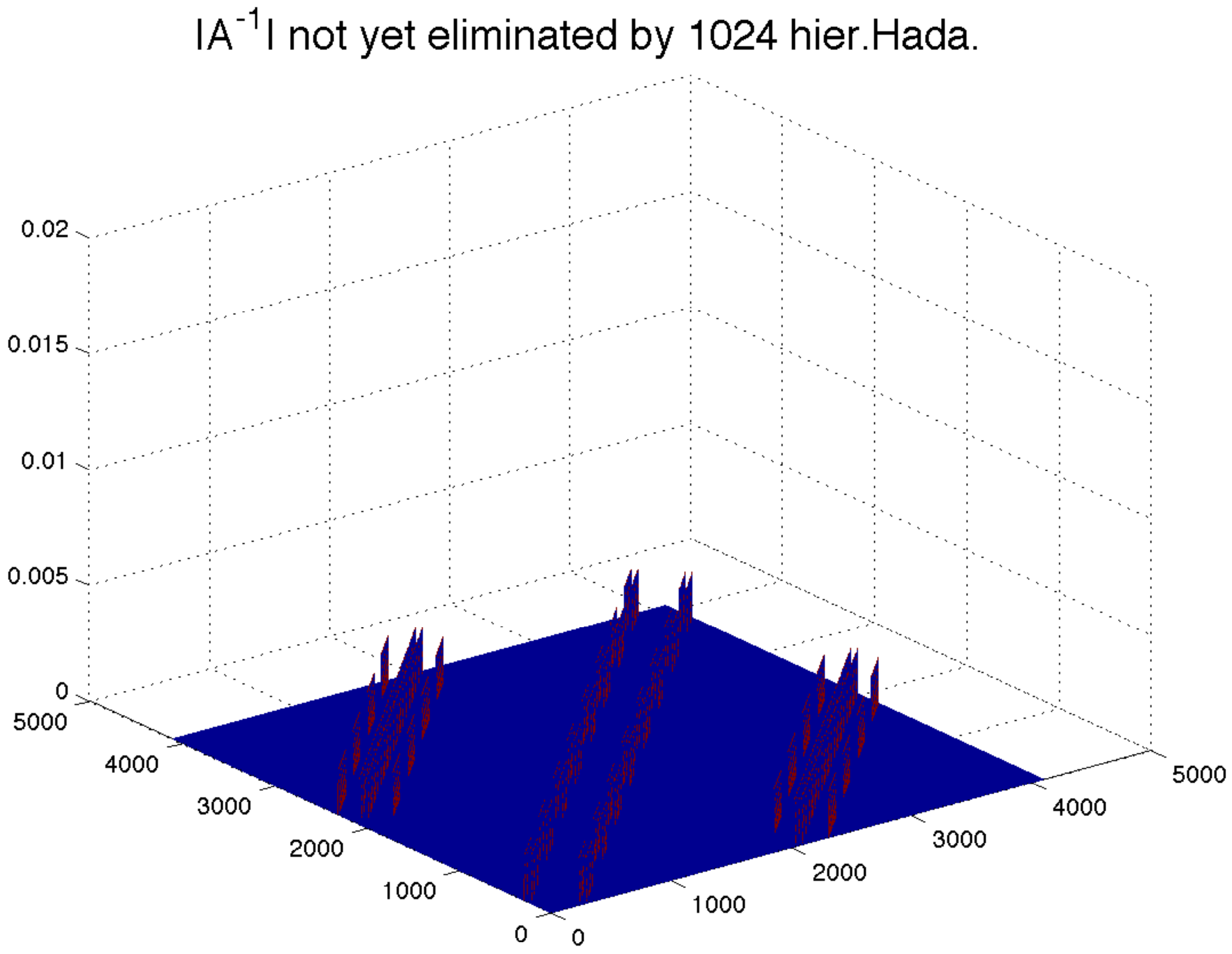
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times8\times8\times8$

Hierarchical Hadamard eliminates largest elements first



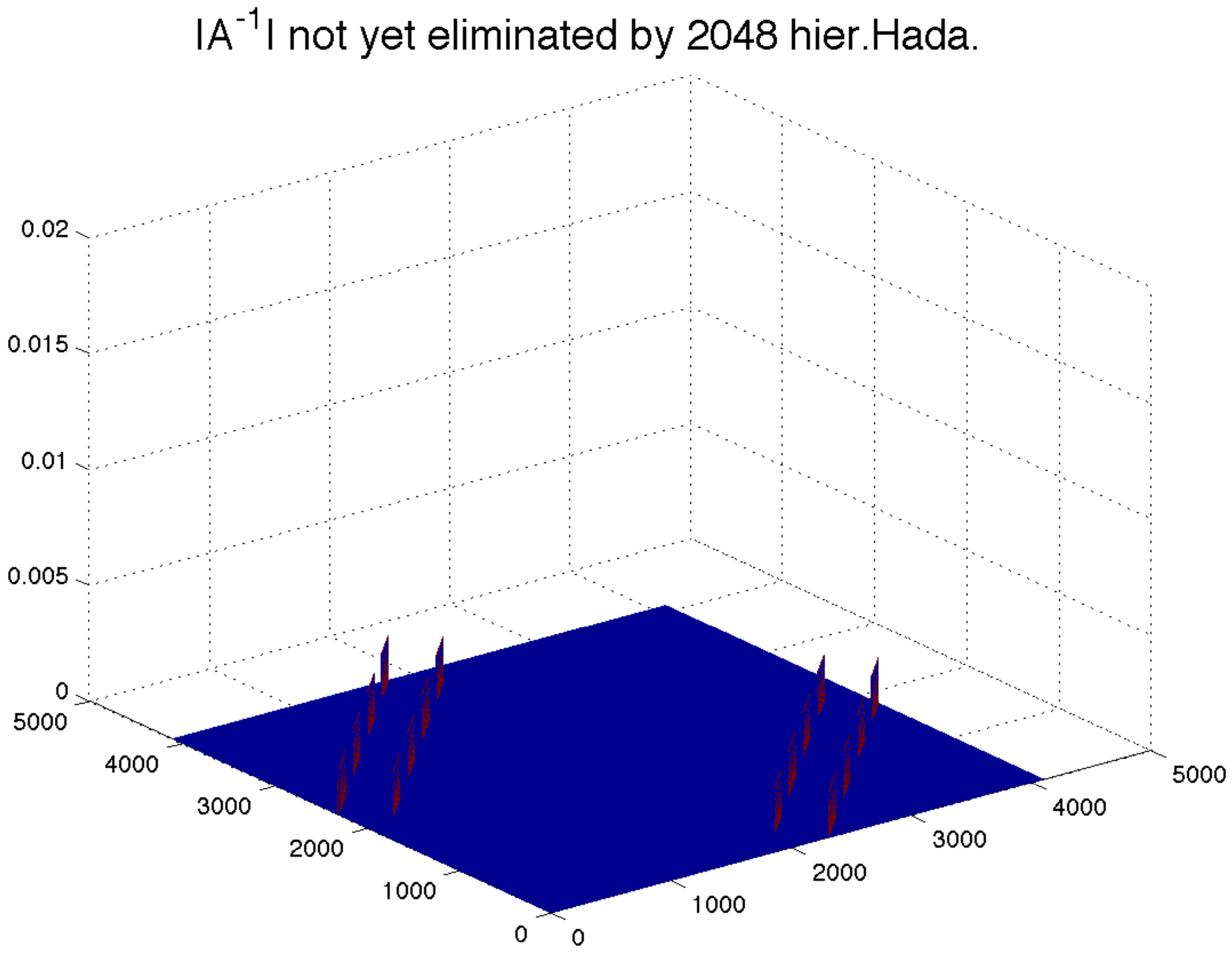
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8 \times 8 \times 8 \times 8$

Hierarchical Hadamard eliminates largest elements first



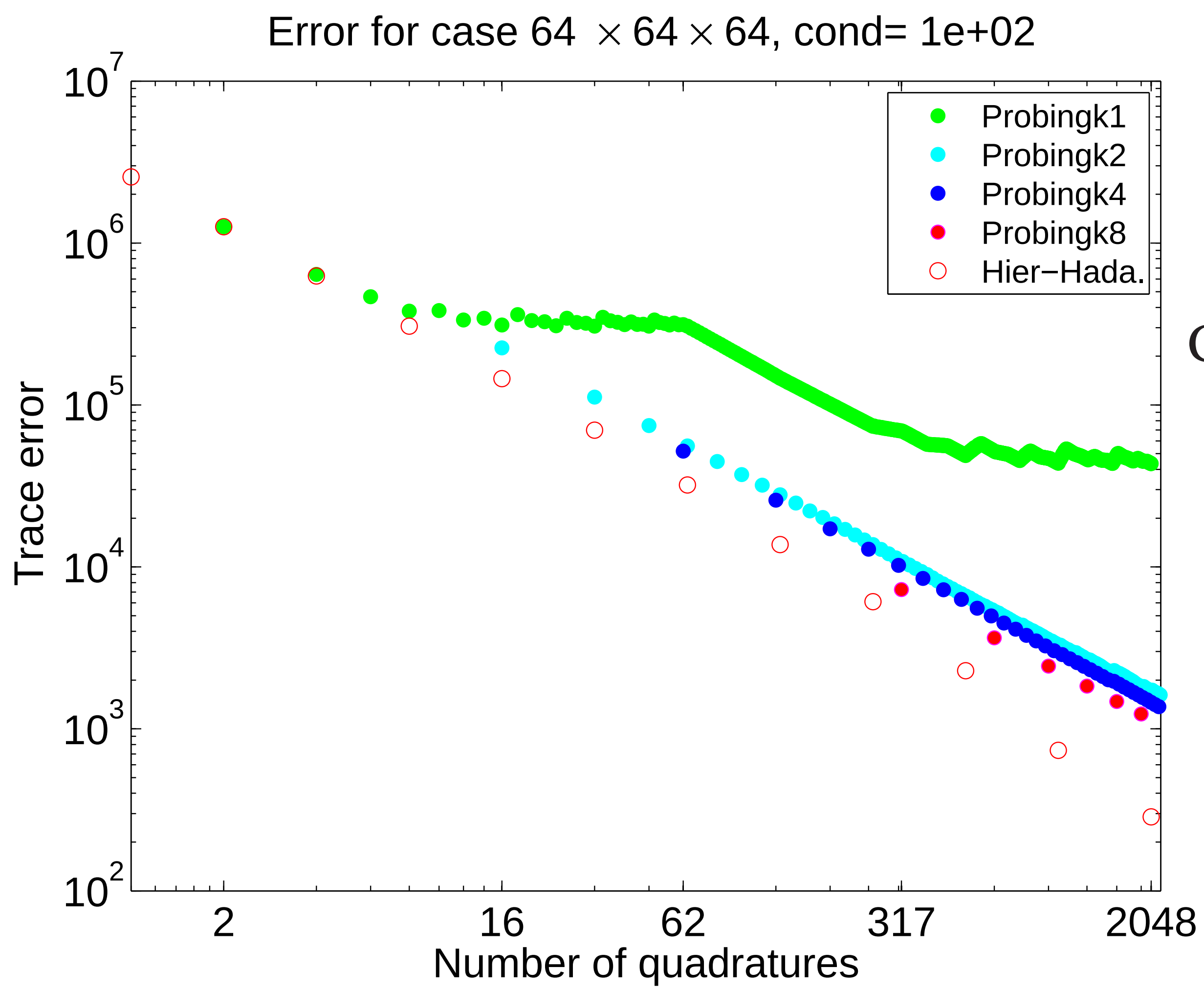
$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times 8\times 8\times 8$

Hierarchical Hadamard eliminates largest elements first



$(A+0.1I)^{-1}$ Laplacian with periodic BC, $8\times8\times8\times8$

Classic probing extended vs Hierarchical Probing



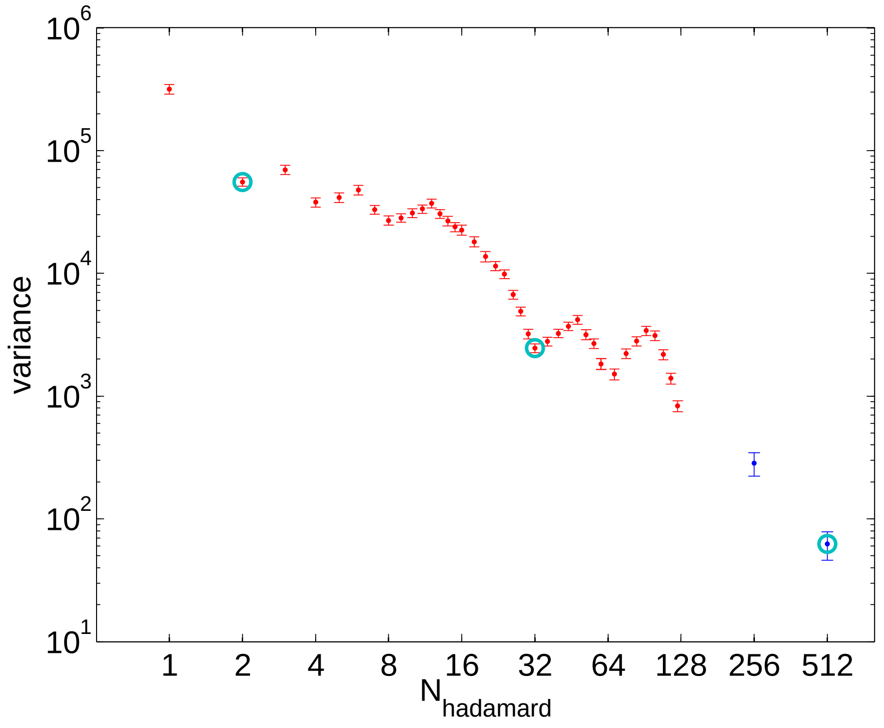
Classic Probing:

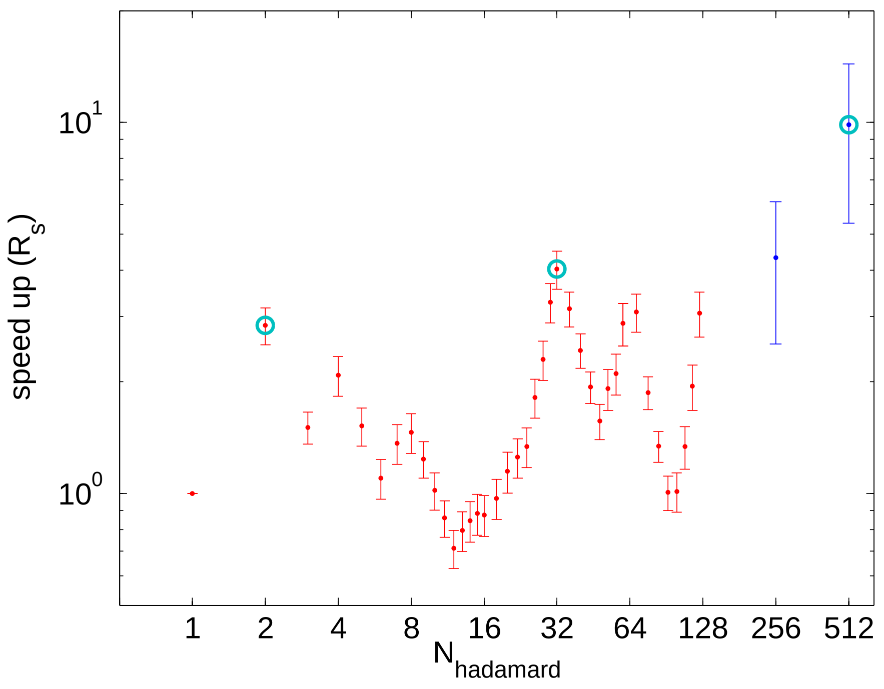
- Extend each color by a Hadamard basis
- More colors beneficial

Hierarchical Hadamard scheme better w/o discarding info

An LQCD lattice of size $32^3 \times 64$ — MPI on 8 GPUs

Explicit dilution of the 12×12 links on each space-time site





Variance reduction (left) and speedup over random noise method (right)

$O(1/n)$ variance reduction

Max speedup when powers of two/colors complete

Hierarchical probing on general graphs

Problems:

(1) expensive to compute $A^k, k = 1, 2, \ldots$

(2) non-hierarchical coloring for A^k graphs, $k = 1, 2, \ldots$

Our solutions:

(1) Multilevel: perform the A^k in coarser graphs

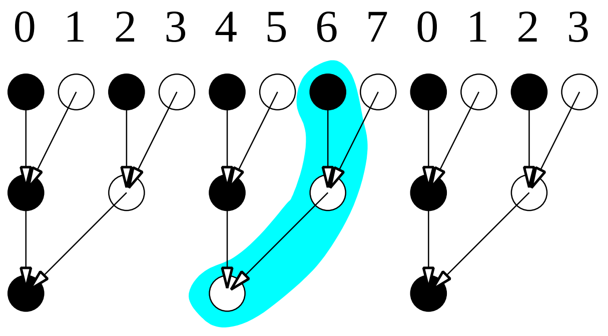
(2) Color hierarchy through mixed radix base: (color$_0$, color$_1$, ... , color$_{level}$)

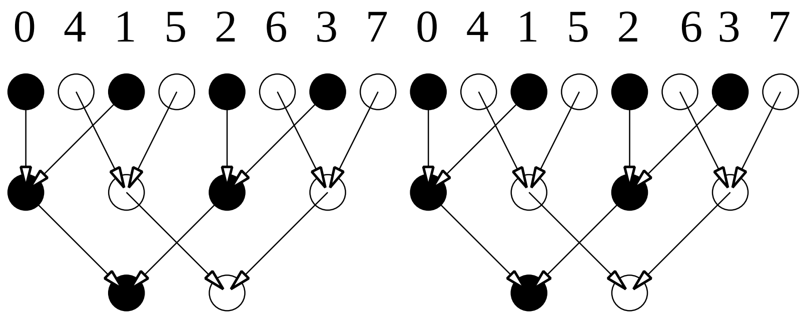
Key of multilevel method: Coarsification

For 1-dimensional grids any coarsification of nearby nodes works

Example: Three levels, merging distance-1 or distance-2 neighbors

Each level is red-black colored. The highlighted fine grid node has colors:
0 at level 0
2 = (10) at level 1
6 = (110) at level 2



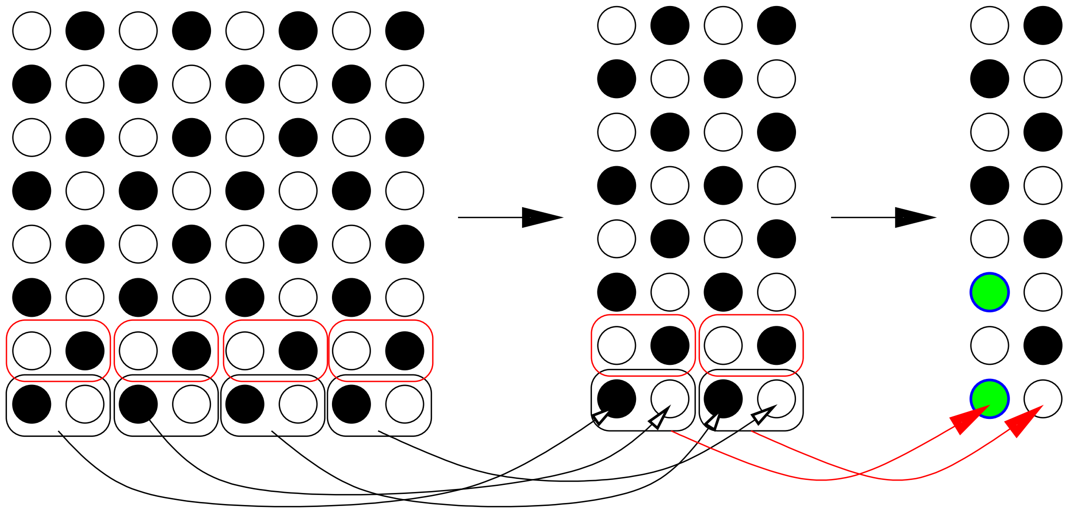


Same color neighbor at distance−8

Key of multilevel method: Coarsification

Problems with higher connectivity

Example: 2-D grid



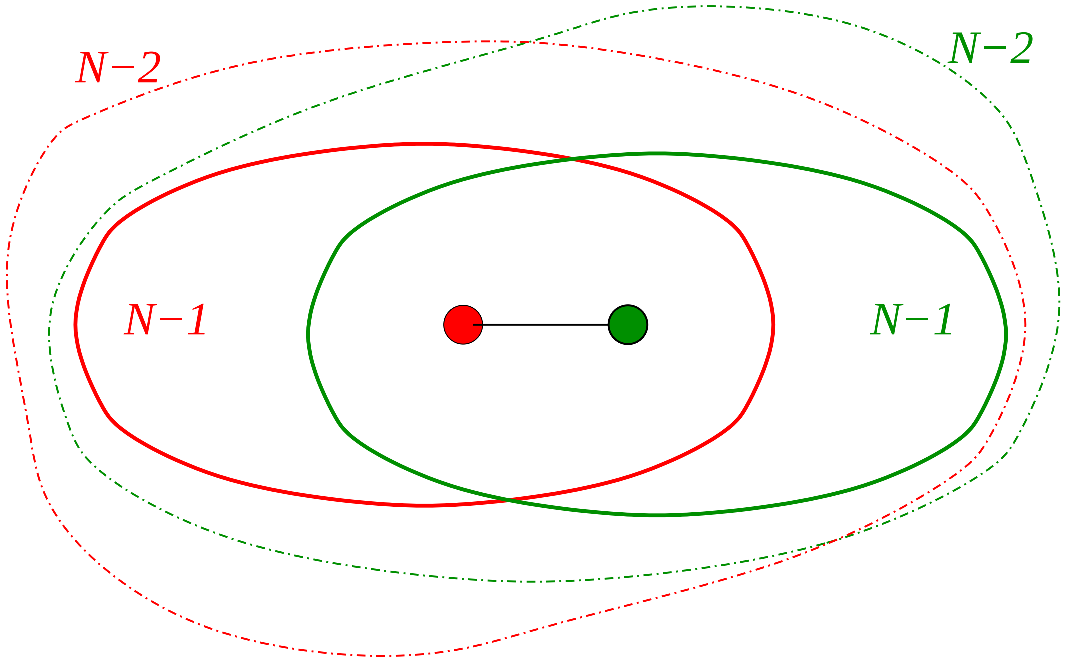
Two green nodes still at distance−2 after 3 levels

What neighbors should we merge at each level to ensure distance 2^{level}?

Coarsification solution

Focus on neighborhoods, not on nodes

If we merge the distance-2 neighborhoods of the coalesced nodes we guarantee a distance 2^{level} coloring



However, coarse graphs become much denser (coarse form of A^{level})

The algorithm

```
MultiLevelColor($A$)
 1. Color graph of $A$
 2. Compute graph of $A^2$
 3. Initialize sparse coarse graph $A_c$
 4. For each node $v$ in $A$
        find $w$ an unmerged neighbor of $v$
        Coalesce $z = (v, w)$ as a node of $A_c$
        $N(A_c, z) = N(A^2, v) \cup N(A^2, w)$
 5. MultiLevelColor($A_c$)
```

The algorithm

```
MultiLevelColor($A$)
 1. Color graph of $A$                              Could generate too many colors
 2. Compute graph of $A^2$                          Expensive to compute
 3. Initialize sparse coarse graph $A_c$
 4. For each node $v$ in $A$
        find $w$ an unmerged neighbor of $v$
        Coalesce $z = (v, w)$ as a node of $A_c$
        $N(A_c, z) = N(A^2, v) \cup N(A^2, w)$
 5. MultiLevelColor($A_c$)
```

Two algorithmic variations

Instead of greedy coloring of A,

- Compute 2 largest eigenectors of the graph Laplacian
- Use spectral coloring with 2 colors (similar to graph partitioning)

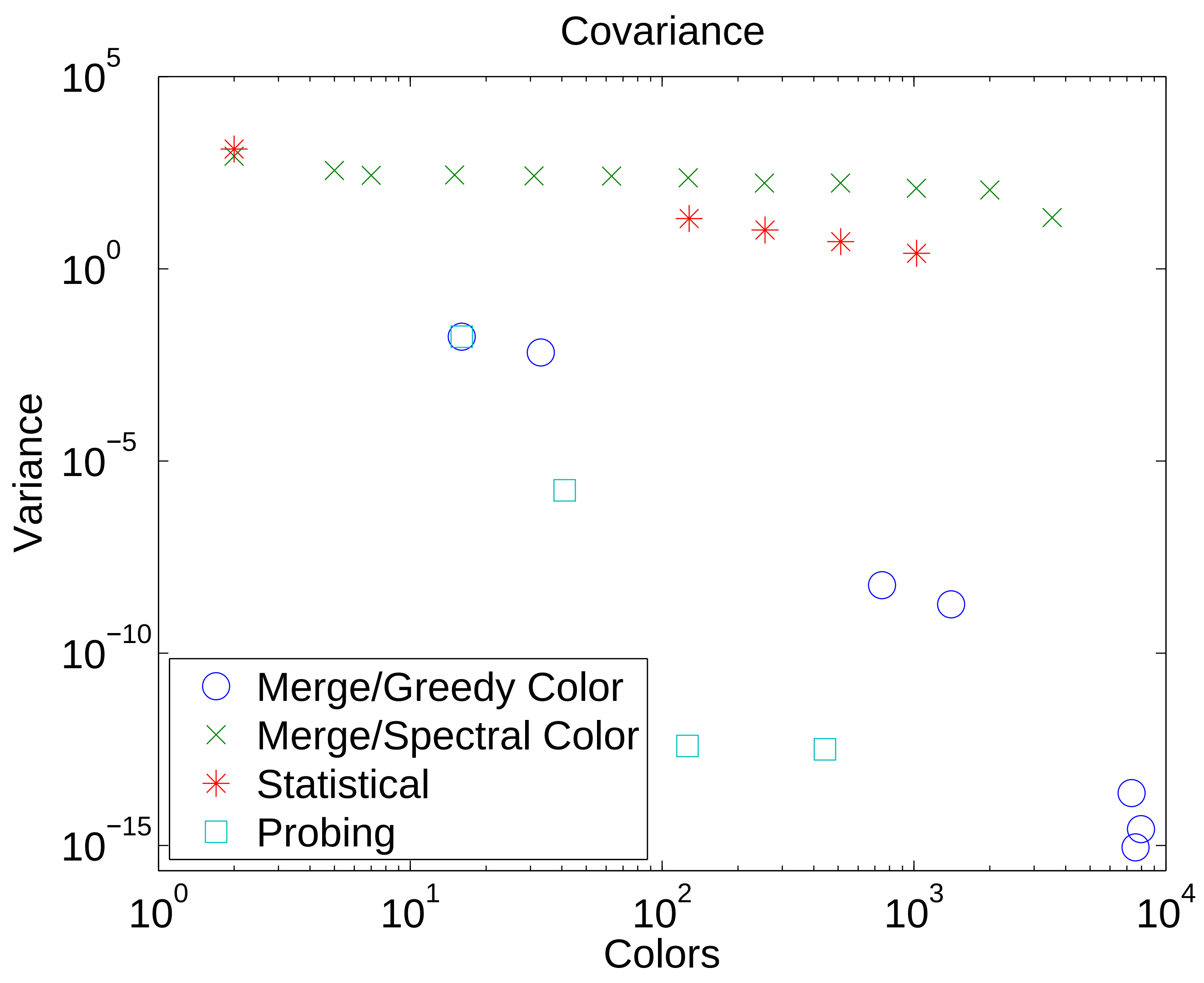
Approximately bipartite, identifies most important links in the first few levels. When the coarse matrix size is small enough, continue with greedy coloring

Instead of computing A^2

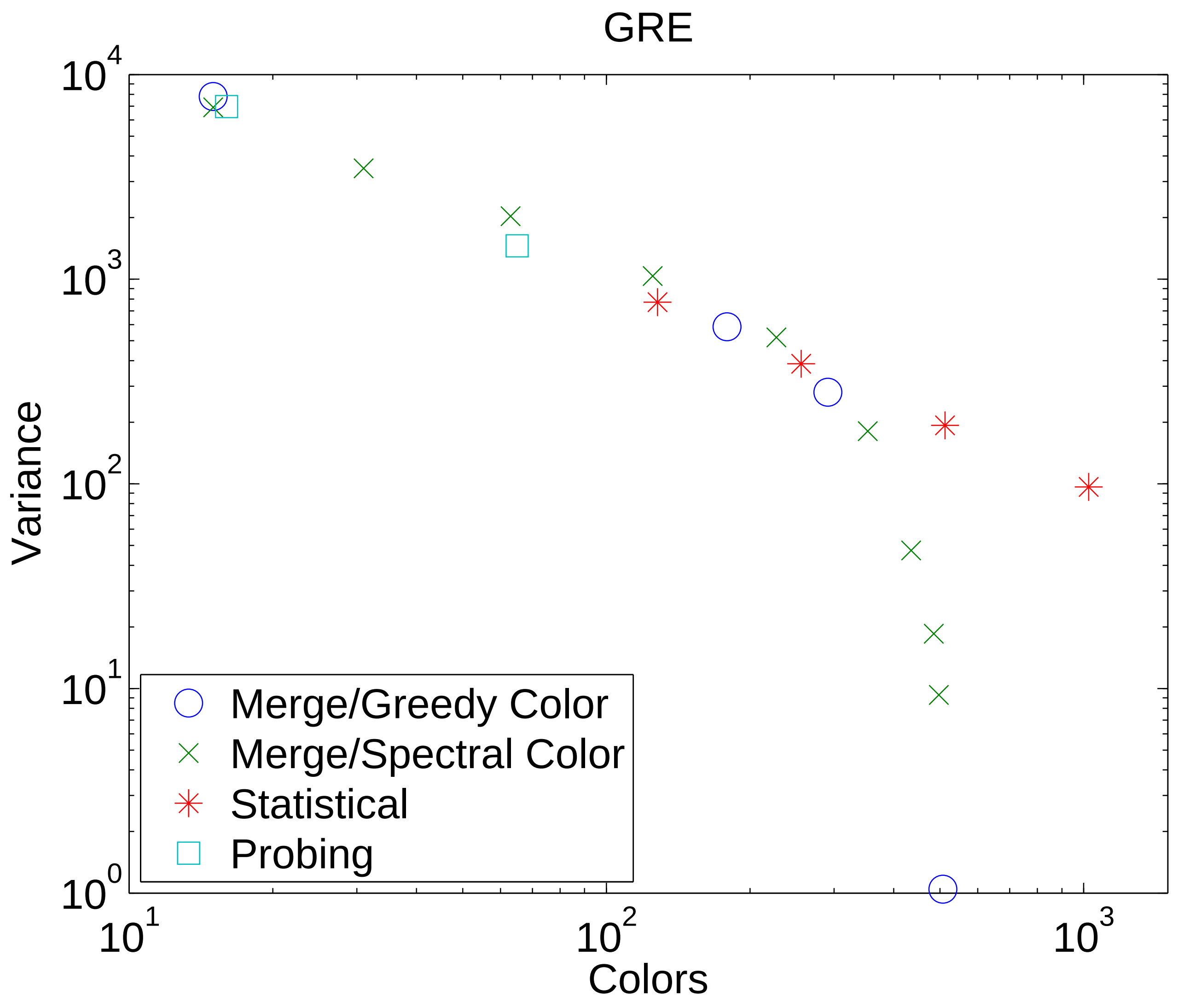
- Implicitly color A^2 and produce the A_c

Avoids the first expensive step, and reduces cost of subsequent steps

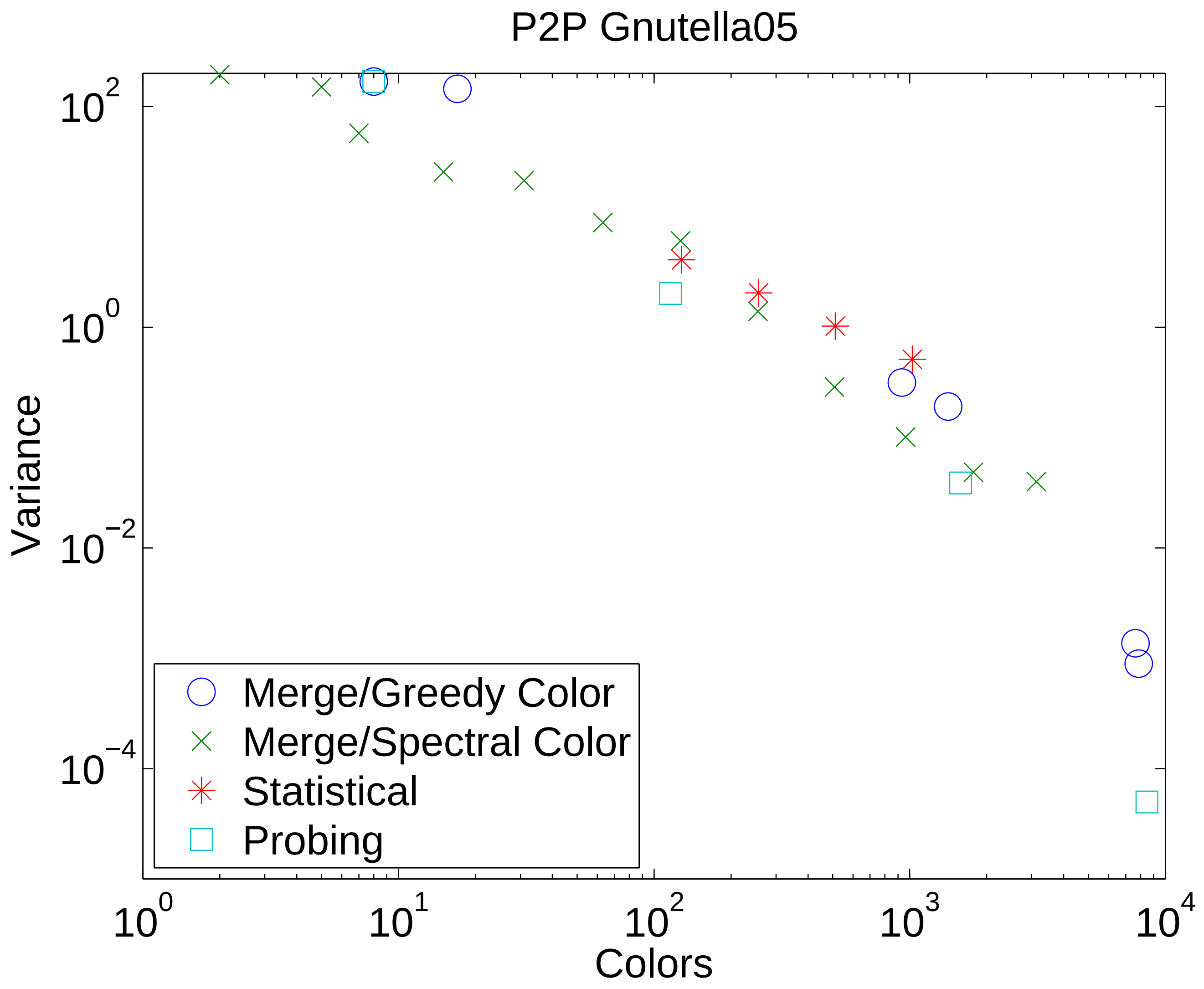
A few preliminary experiments



A few preliminary experiments



A few preliminary experiments



Hierarchical Probing conclusions

For regular grids significant speedups at no additional cost

For general matrices that are amenable to probing, our multilevel algorithm

- is close to the benefits of probing
- is far less expensive than probing
- requires an optimized C implementation