# Fitting techniques for estimating the trace of the inverse of a matrix

Andreas Stathopoulos, Lingfei Wu, Jesse Laeuchli
College of William and Mary

Vasilis Kalantzis
University of Minnesota

Stratis Gallopoulos
University of Patras, Greece

Acks: NSF, DOE SciDAC

## The problem

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

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

Common functions $f(A) =$

$$A^{-1}, \ \log(A), \ \exp(A), \ R_i^T A^{-1} R_j, \ \ldots$$

Applications: UQ, Data Mining, Quantum Monte Carlo, Lattice QCD

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

## Standard underlying method

Monte Carlo (Hutchinson 1989)

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

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

Monte Carlo Trace
for i=1:$n$
  $x$ = randZ2($N$,1)
  sum = sum + $x^T A^{-1} x$

trace = sum$/n$

## Standard underlying method

Monte Carlo (Hutchinson 1989)

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

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

Monte Carlo Trace

for i=1:$n$

  $x$ = randZ2($N$,1)

  sum = sum + $x^T A^{-1} x$

trace = sum$/n$

2 problems

Large number of samples

How to compute $x^T A^{-1} x$

## Standard underlying method

Monte Carlo (Hutchinson 1989)

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

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

Monte Carlo Trace
for i=1:$n$
  $x$ = randZ2($N$,1)
  sum = sum + $x^T A^{-1} x$

trace = sum$/n$

Solve $Ay = x$ vs quadrature $x^T A^{-1} x$
  Golub'69, Bai'95, Meurant'06,'09, Strakos'11
$O(100-1000s)$ statistically independent RHS
  Recycling (de Sturler), Deflation (Morgan, AS'07)

## Selecting the vectors in $x^T A^{-1} x$ to min variance or error

Random

| | |
|---|---|
| $x \in Z_2^N$ | best variance for real matrices (Hutchinson 1989) |
| $x = e_i$ | variance depends only on $\mathrm{diag}(A^{-1})$ |
| $x = F^T e_i$ | mixing $F =$ DFT, Hadamard (Avron et al. 2010) |

Deterministic

| | |
|---|---|
| $x = He_i, \ i = 1, \ldots, 2^k$ | 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) |
| $x = H(p_m, k_i)$ | Hierarchical Probing for lattices (A.S, J.L. 2013)<br>Maintains benefits of probing but cheap and incremental |

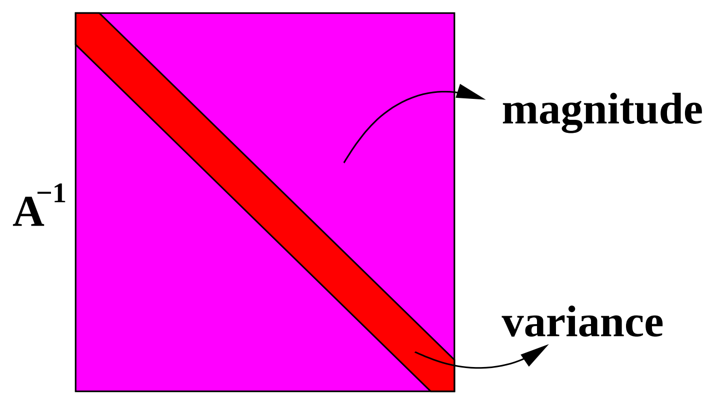
## Variance of the estimators

Rademacher vectors $x_i \in Z_2^N$

$\overline{Tr} = \frac{1}{s} \sum_{i=1}^{s} x_i^T A^{-1} x_i \qquad Var(\overline{Tr}) = \frac{2}{s} \|\tilde{A}^{-1}\|_F^2 = \frac{2}{s} \sum_{i \neq j} (A_{ij}^{-1})^2$

Diagonal $x = e_{j(i)}$

$\overline{Tr} = \frac{N}{s} \sum_{i=1}^{s} A_{j(i),j(i)}^{-1} \qquad Var(\overline{Tr}) = \frac{N^2}{s} Var(\mathrm{diag}(A^{-1}))$




Unclear which method is best a-priori

## Why focus on the diagonal method?

Trace = integral of a 1-D signal. Can we improve Monte Carlo?

Not without external information about the distribution of diagonal elements

Our goals:

- What if we have an approximation $M \approx \text{diag}(A^{-1})$?
- Is $\mathbf{Tr}(M) \approx \mathbf{Tr}(A^{-1})$ sufficient?
- If not, can we use fitting $p(M)$ (regression/interpolation/quadrature)?
- Can the fitting reduce $Var(p(M) - \text{diag}(A^{-1}))$?

## Approximations to diag($A^{-1}$)

- Inexpensive bounds on diagonal elements (Robinson and Wathen '92)
  e.g., for $A$ SPD, $1/A_{ii}$ often capture the pattern of $diag(A^{-1})$

- Let $[L,U] = \text{ILU}(A)$ (incomplete LU) and $M = \text{diag}(U^{-1}L^{-1})$
  Requires only $A^{-1}_{i,j}$ entries from sparsity of $L,U$ (Erisman, Tienny, '75)

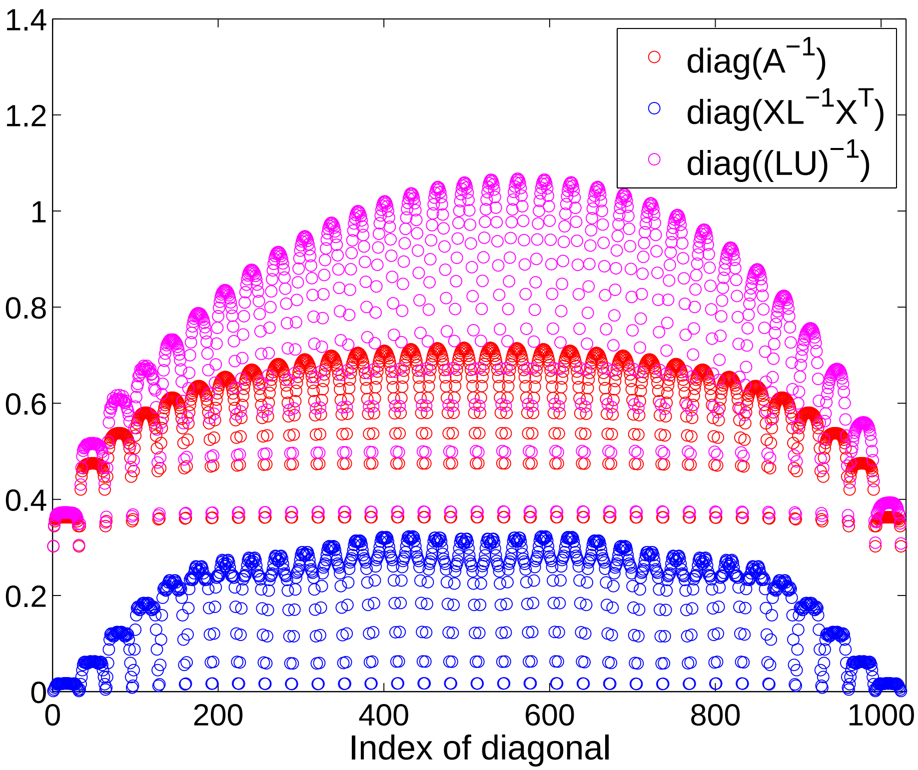
- Eigen/singular vectors

$$M = \text{diag}(X\Lambda^{-1}Y^T), \text{ for } nev \text{ smallest eigenvalues}$$

  Already available from deflating multiple right hand sides!
  Number of eigenvectors can be increased while solving $Ax = e_i$ (eigCG)

## For some problems $M$ captures pattern of diag$(A^{-1})$ well

Laplacian `delsq(numgrid('S',34))`




Deflation: 15 smallest eigenpairs

`ILU('crout','row',0.01)`

Traces not close but

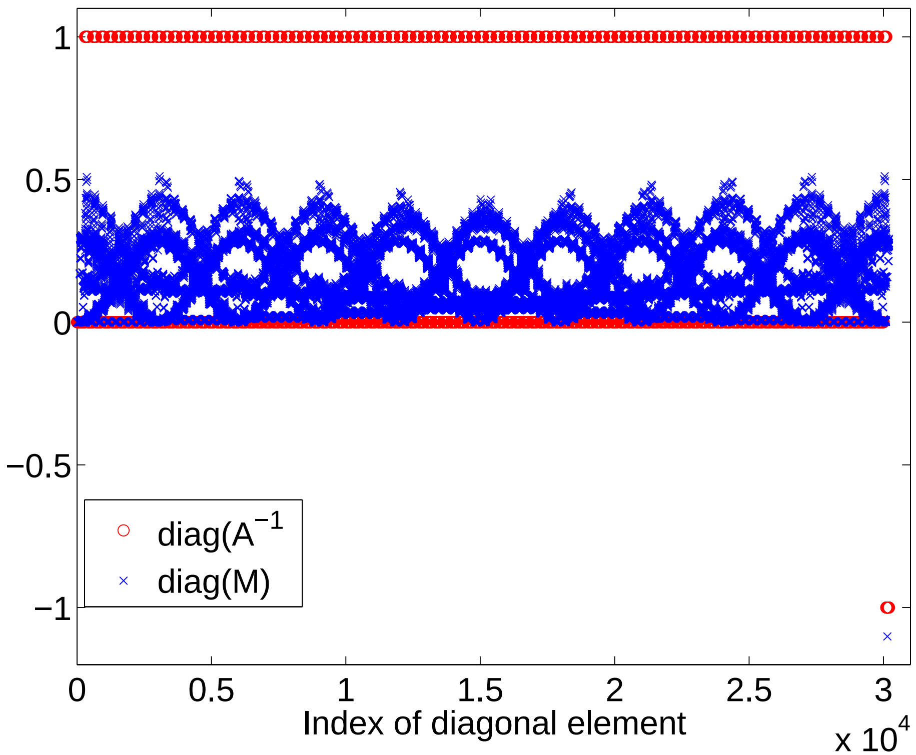
$Var(\mathrm{diag}(X\Lambda^{-1}X^T - A^{-1})) = $ 4e-4

$Var(\mathrm{diag}((LU)^{-1} - A^{-1})) = $ 1e-2

MC on diag$(A^{-1} - M)$ can be competitive to Hutchinson's method

# In some cases approximation is pointless

Rajat10 circuit simulation matrix (size 30202)




$M$ from 100 smallest singular triplets

## Capture pattern better by fitting $M$ to $D = \mathbf{diag}(A^{-1})$

MC resolves shift $D = c + M$, but not scale $D = bM$ (variance may increase!)

Approach 1. Least squares fit with $bM + c$

1. Solve $D_i = e_i^T A^{-1} e_i$, for $i \in S$ a set of $k$ indices
2. Find $[b, c] = \text{argmin}\{\|D(S) - (bM(S) + c)\|_2,\ b, c \in \Re\}$

Not many points (linear systems) are needed. Typically 10-20.

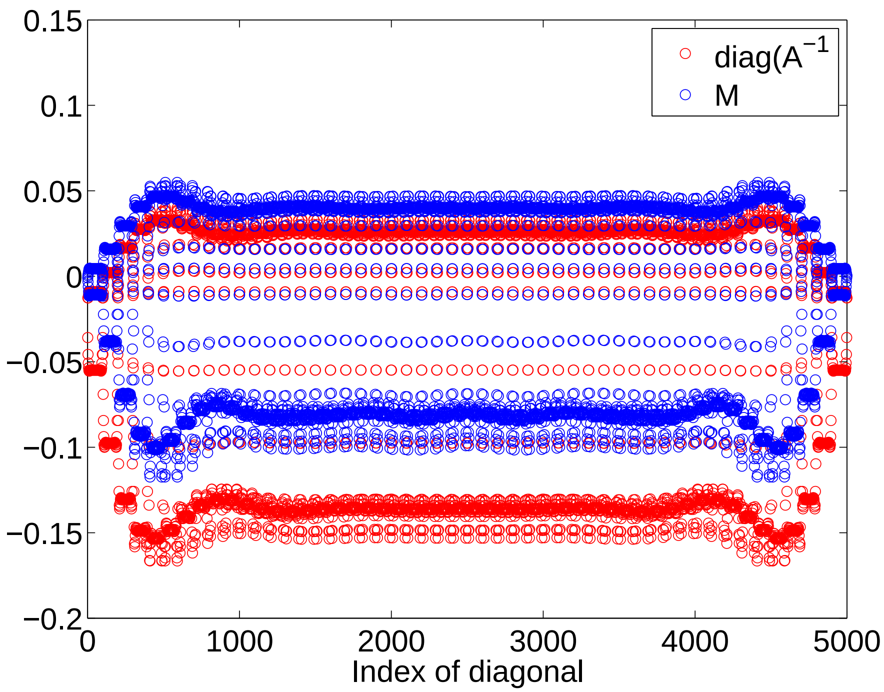
Significant improvement in the estimation of trace

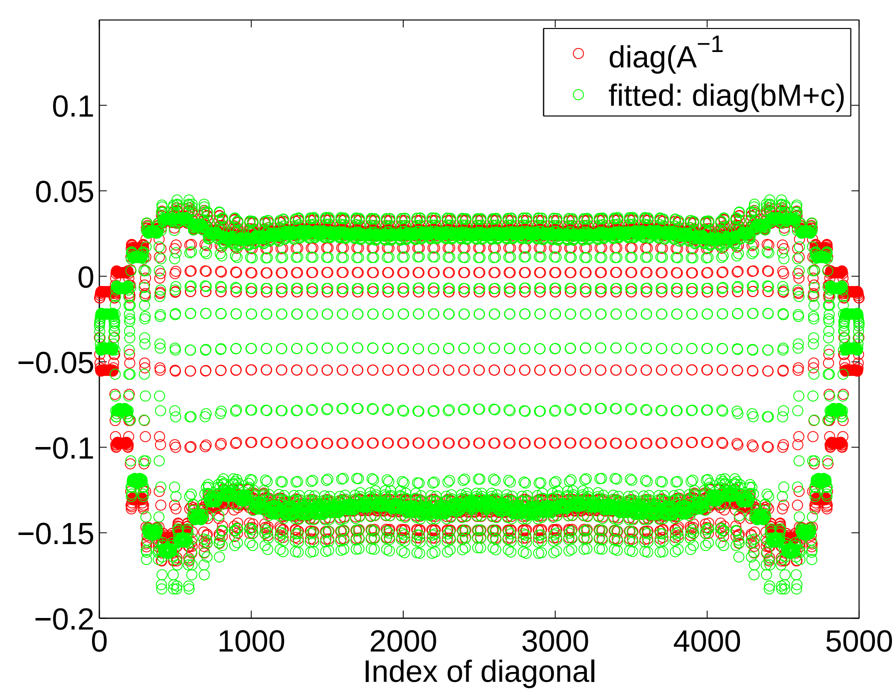
Reduces variance for potentially continuing with MC

## Approach 1 example $D \approx bM + c$

Matrix RDB5000, 50 smallest singular triplets, k=20 points used to fit

Accuracy of systems and singular vectors is 1e-6.




| | | | |
|---|---|---|---|
| $\mathbf{Tr}(A^{-1})$ | $-267.880$ | $Var(D)$ | $6.1e-3$ |
| $\mathbf{Tr}(b*M+c)$ | $-267.544$ | $Var(D-M)$ | $4.3e-4$ |
| $Rel.Err.$ | $1E-3$ | $Var(D-bM-c)$ | $4.3e-5$ |

## Better fitting

Linear model preserves shape of $M$, thus relies too much on the quality of $M$

Interpolating with a higher degree polynomial could be noisy.

Approach 2 basic. Piecewise Cubic Hermitian Spline Interpolation (PCHIP)

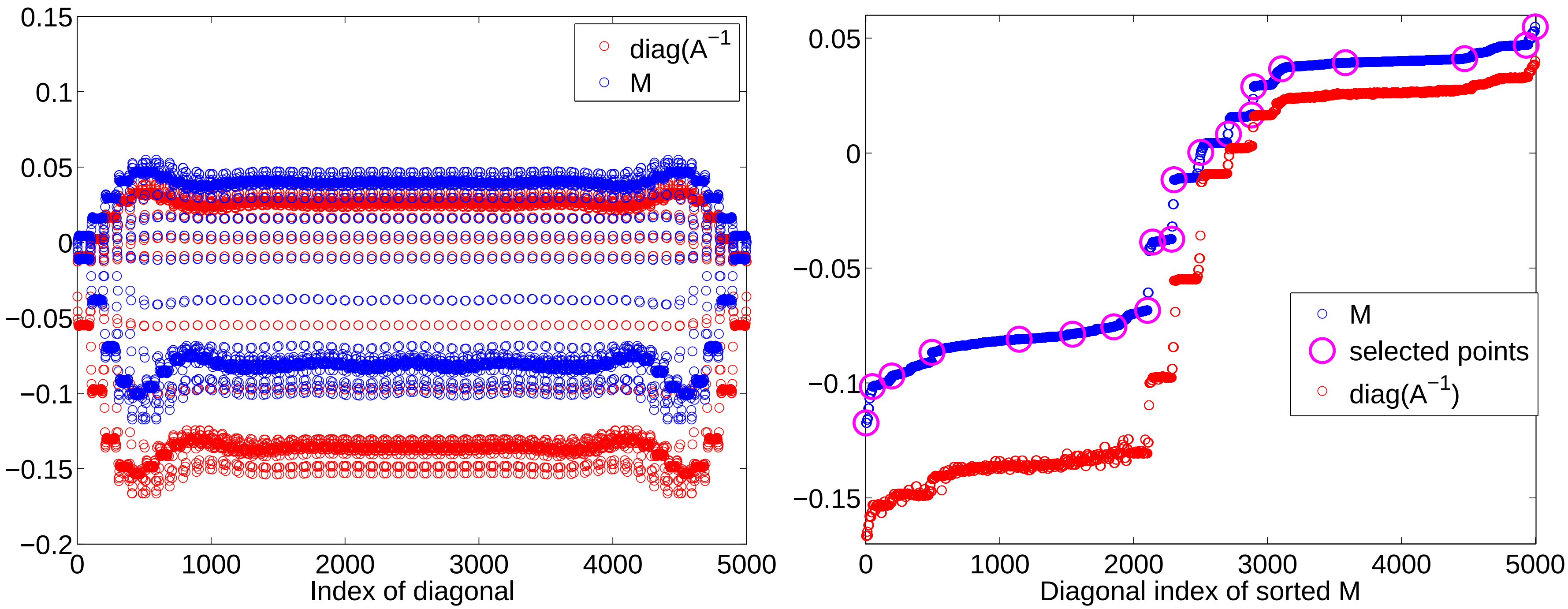
1. Solve $D_i = e_i^T A^{-1} e_i$, for $i \in S$ a set of $k$ indices
2. Fit $p(M(S)) = D(S)$

For PCHIP to effectively capture the pattern (global and local) of $D$ it needs:

- smoothness of the approximant
- elements of $M(S)$ to appear in increasing order
- to capture the whole range of values of $D$
- to capture where most of the action in $D$ is happening

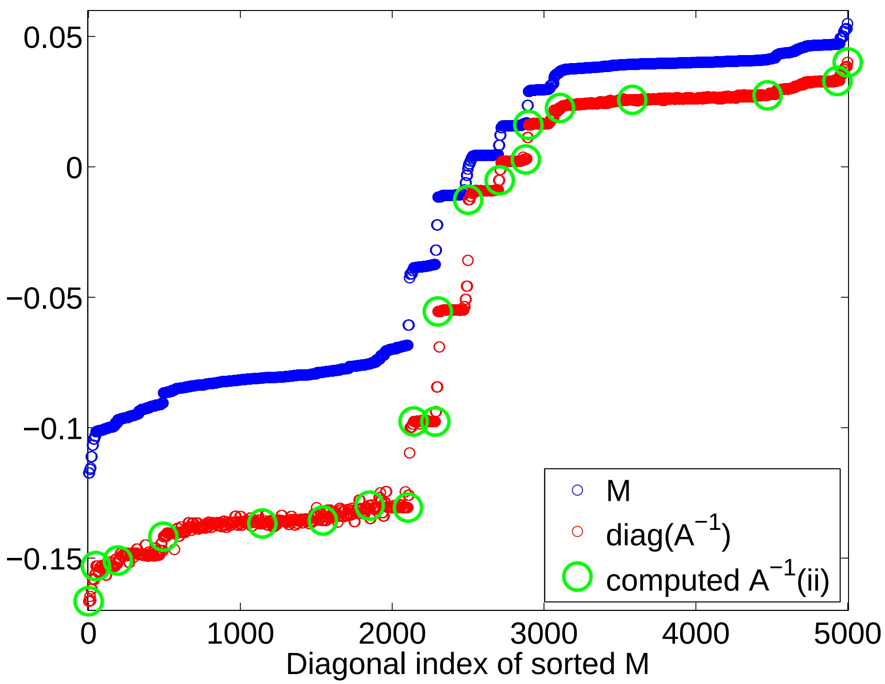
## Approach 2. Piecewise Cubic Hermitian Spline Interpolation (PCHIP)

1. $[\tilde{M}, J] = \text{sort}(M)$ to obtain a CDF-like, smooth graph
2. Choose $Q$ a set of $k$ indices: $\{1,2\} \in Q$ and the $k-2$ are chosen such that they minimize the integration error with trapezoidal rule of $\tilde{M}$. Do not consider indices that produce non-unique $\tilde{M}_i$ values.



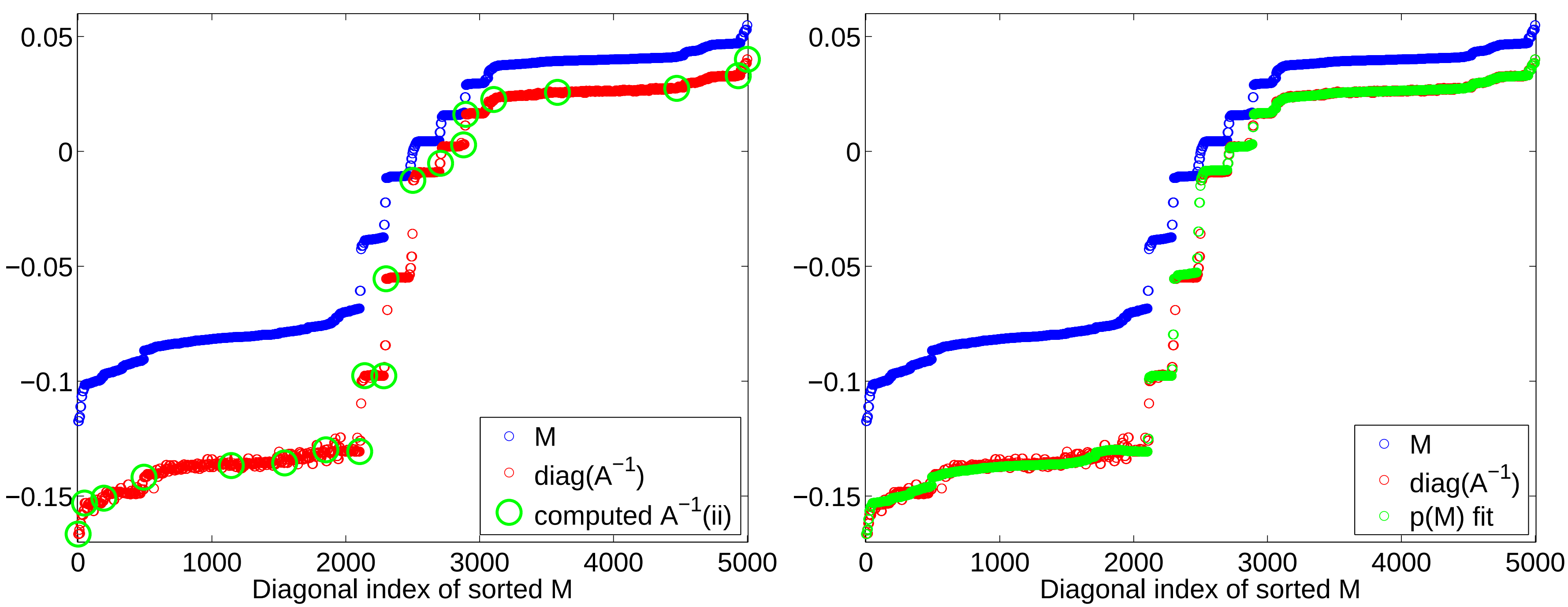
## Approach 2. Piecewise Cubic Hermitian Spline Interpolation (PCHIP)

1. $[\tilde{M}, J] = \text{sort}(M)$
2. Choose $Q$ a set of $k$ indices.
3. $S = J(Q)$ the corresponding indices in original ordering
4. Solve $D_i = e_i^T A^{-1} e_i$, for $i \in S$



# Approach 2. Piecewise Cubic Hermitian Spline Interpolation (PCHIP)

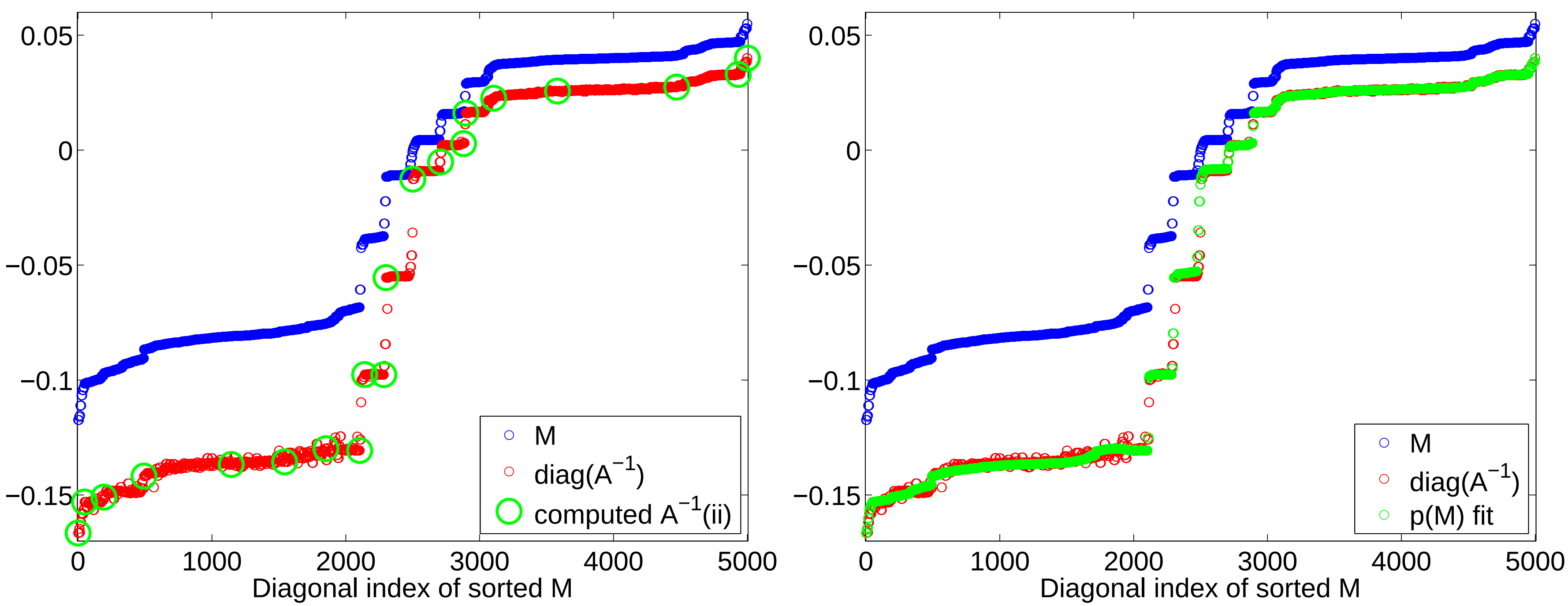
1. $[\tilde{M}, J] = \text{sort}(M)$
2. Choose $Q$ a set of $k$ indices.
3. $S = J(Q)$ original ordering
4. Solve $D_i = e_i^T A^{-1} e_i$, for $i \in S$
5. PCHIP fit $p(M(S)) = D(S)$. Use $p(M) \approx D$



## Approach 2. Piecewise Cubic Hermitian Spline Interpolation (PCHIP)

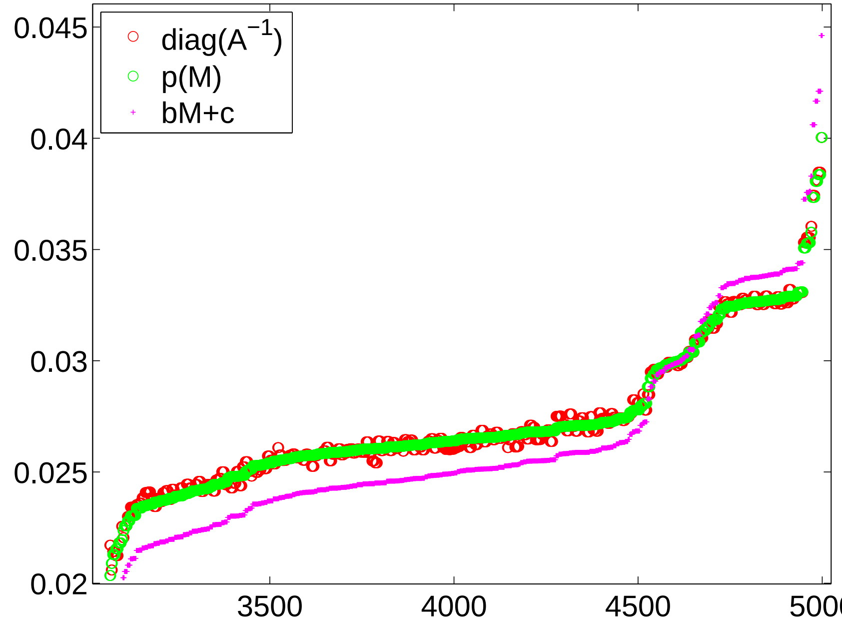
1. $[\tilde{M}, J] = \mathrm{sort}(M)$
2. Choose $Q$ a set of $k$ indices.
3. $S = J(Q)$ original ordering
4. Solve $D_i = e_i^T A^{-1} e_i$, for $i \in S$
5. PCHIP fit $p(M(S)) = D(S)$. Use $p(M) \approx D$

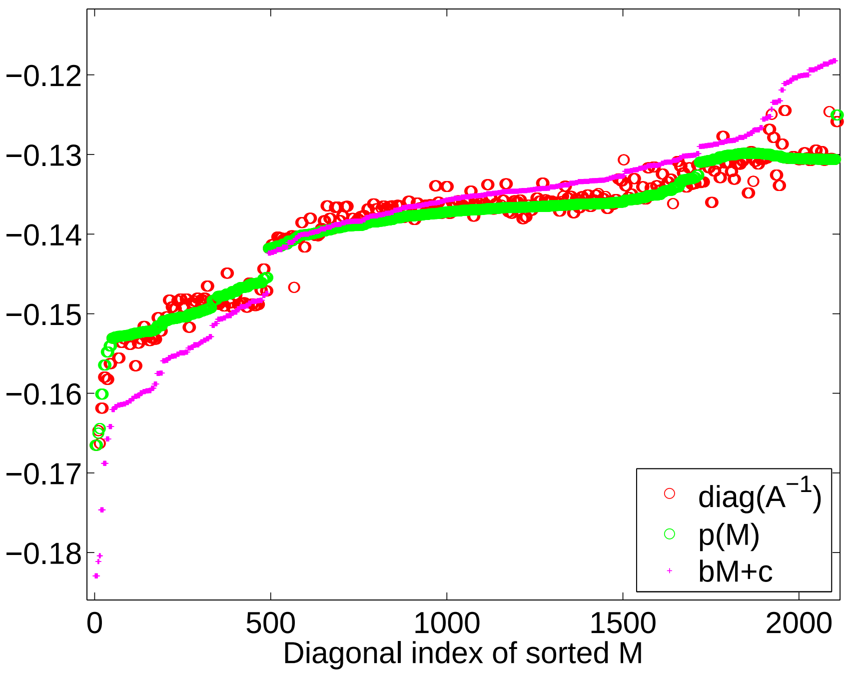
If (4) computes also evecs, update points incrementally

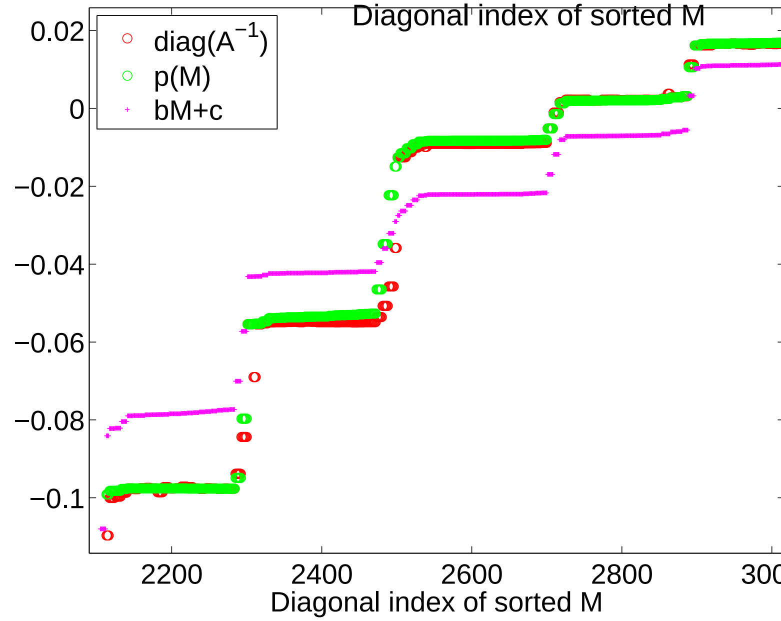


# Approach 2. Piecewise Cubic Hermitian Spline Interpolation (PCHIP)

Improvement over bM+c

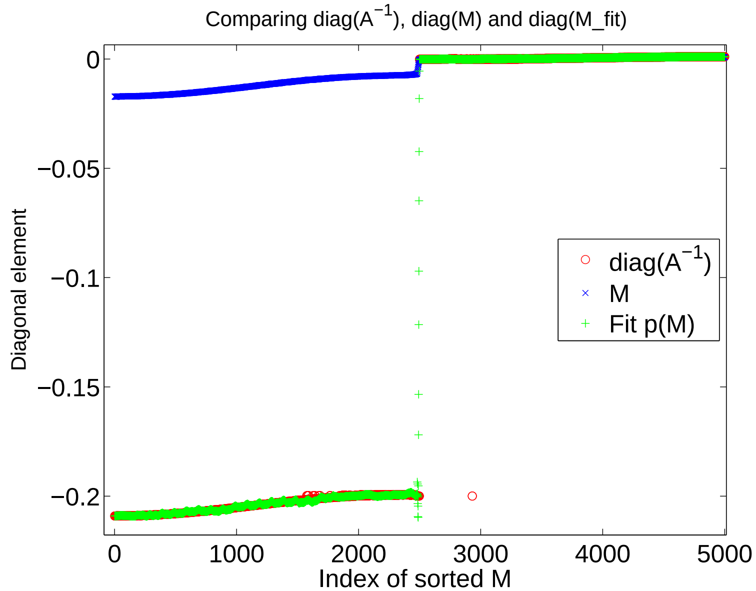


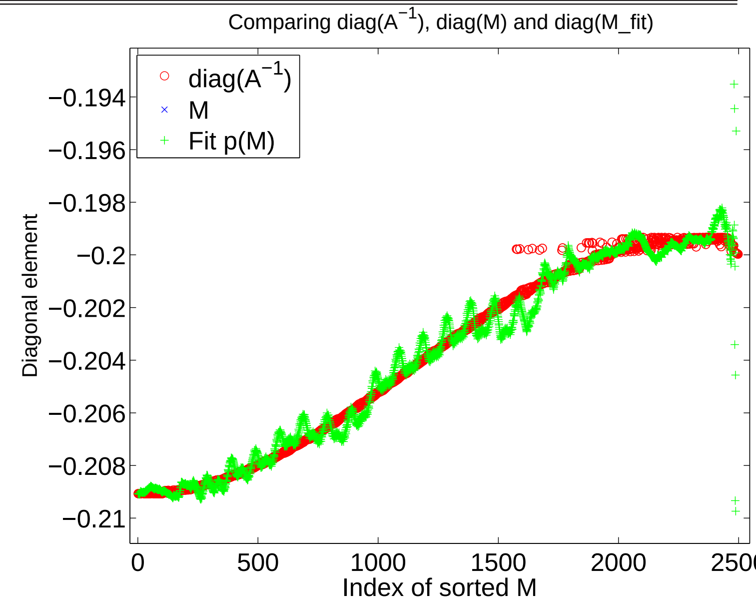



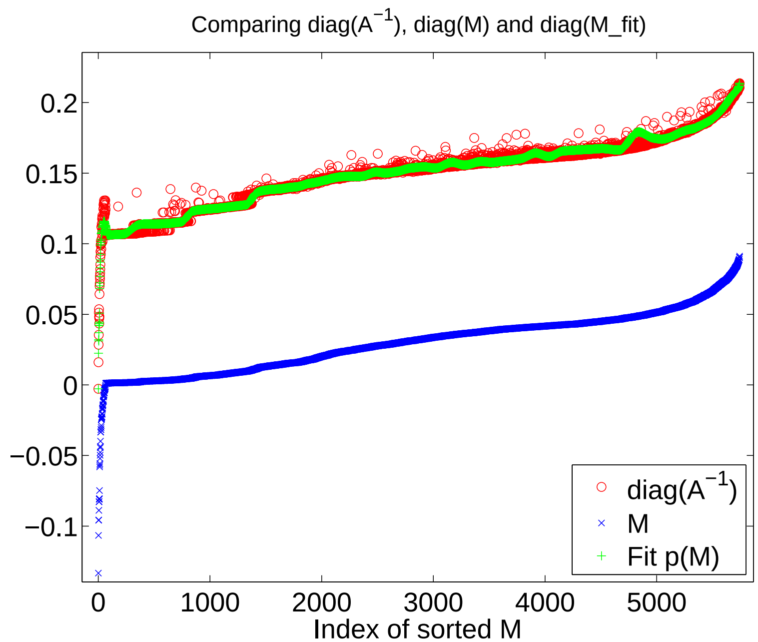
# Fitting examples nev=k=100: OLM5000, SiNa, KUU

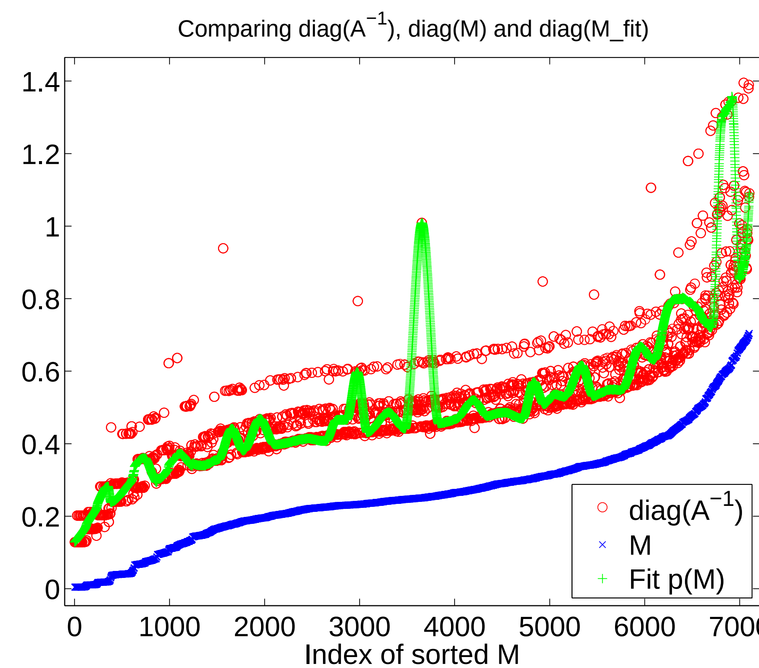
OLM5000




SiNa



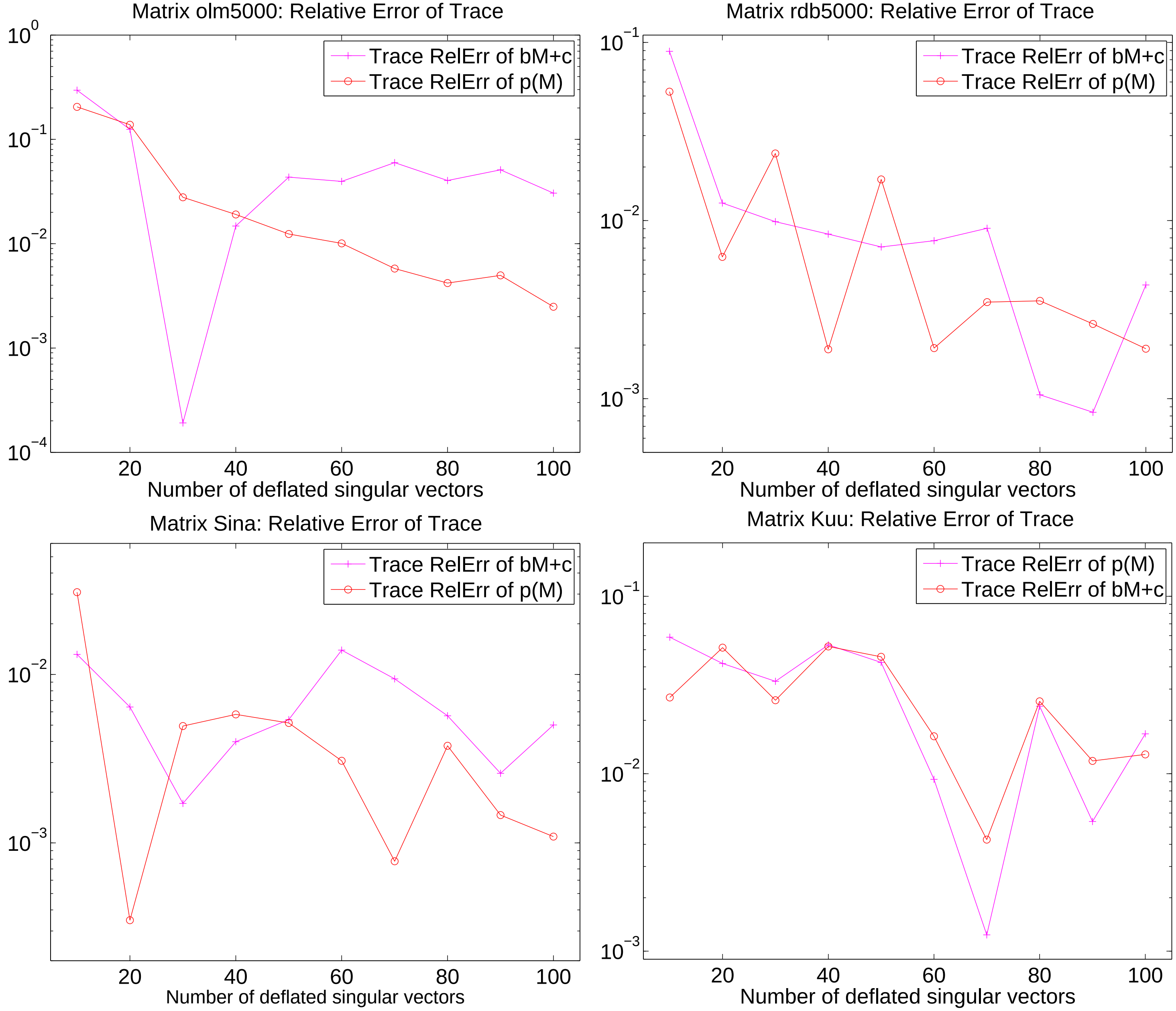




KUU

# Very good eigenvalue approximation

OLM5000

RDB5000



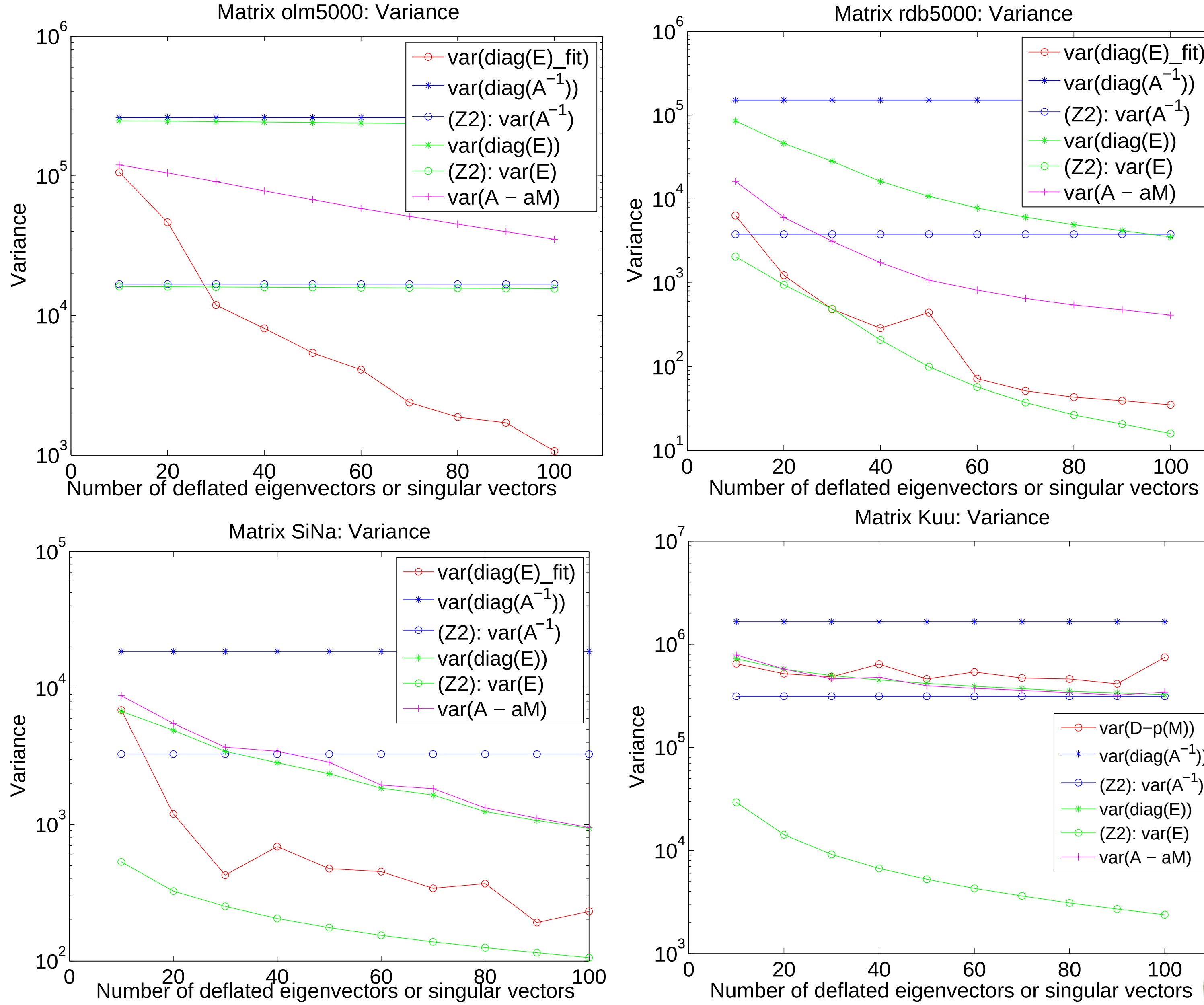

SiNa

KUU

# Variance: Z2 on $E = A^{-1} - Y\Sigma^{-1}X^T$ vs MC on diag $D - p(M)$

OLM5000

RDB5000




SiNa

KUU

## When to use it? Estimate dynamically:

1. Relative trace error

   Cross validation:

   (a) Use $m$ subsets $S_i \subset S$

   (b) Fit $p(\tilde{M}(S_i))$ and compute the mean error $\varepsilon_i$ of the $S - S_i$ points

   (c) Confidence interval for error: $\pm 2\sqrt{Var(\varepsilon_i)}$

2. Variance of $(D - p(M))$ vs Z2 on $E$

   (a) Compute $a_j = A^{-1}e_j, \; j \in S$.

   (b) Based on $a_{jj}$ update estimates for $var(D), \; var(D-M), \; var(D-p(M))$

   (c) Based on $a_{ij}$ and $\mu_i = Y\Sigma^{-1}X^T e_i$ update Hutchinson variance estimates

   $\text{var}(A) = 2\|\overline{A}\|_F^2$

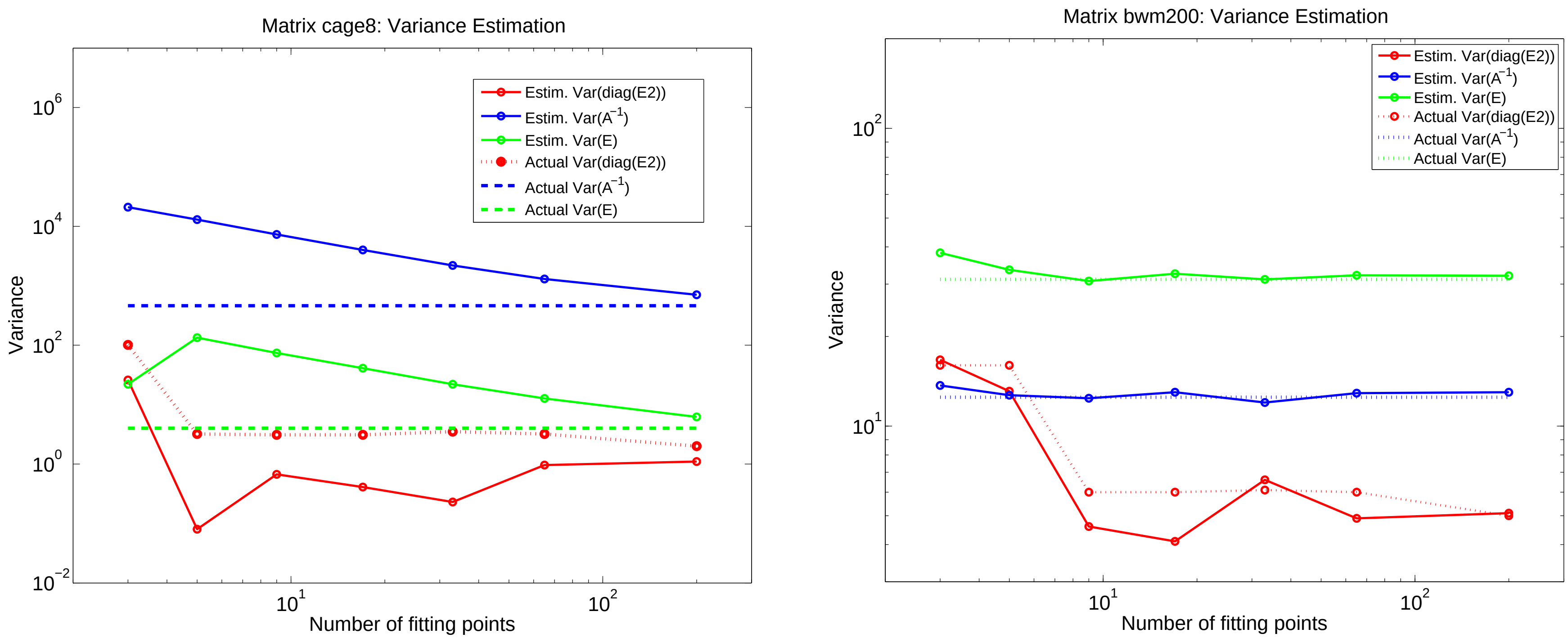
   $\text{var}(A - Y\Sigma^{-1}X^T)$

Large differences in various methods would show after a few points

# Dynamically identifying smallest variance

Estimated variance converges to actual variance

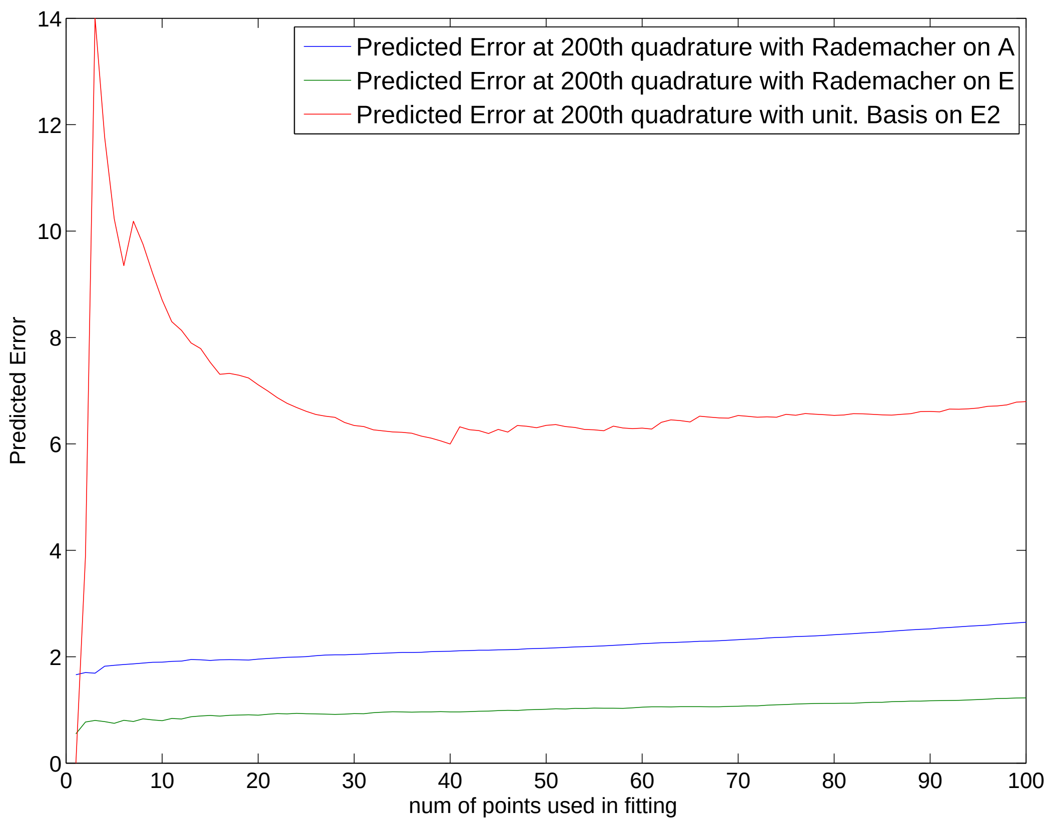
Relative differences apparent almost immediately




# Dynamically identifying smallest variance

If a total of $s$ steps allowed, what method will give the smallest error at $s$?

Eg., the matb5 QCD matrix:




After 10 steps, excellent match between estimated and observed variances

## Conclusions

If $M$ approximates qualitatively well $D$, our technique combines deterministic regression and stochastic estimation to achieve good accuracy on $\sum D_i$ with as few samples as possible.

- Most eigenvectors are a by product of solving right hand sides (samples).
- Fitting achieves good eigenvalue accuracy, soon (less expensive than MC)
- Fitting may or may not improve variance
- Dynamic monitoring possible. Some improvements are needed.