Progress toward fast algorithms for protein design

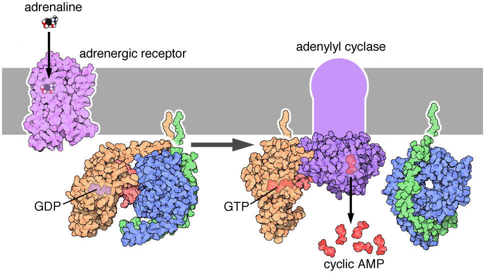
Kenneth L. Ho

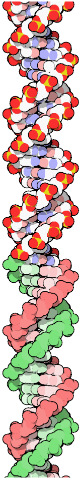
Stanford University

MBI Young Researchers Workshop 2014

Introduction

- Structure-function relationship is central to biochemistry
- "Theorem": structure $\implies$ function
- Examples: ligand-receptor binding, DNA replication
- Corollary: **function design** reduces to **structure design**





Images from RCSB PDB Molecule of the Month.

Protein design and structure prediction

- Protein defined by a sequence of amino acid residues
- **Protein design**: find a sequence folding to the desired stable structure
- **Protein structure prediction**: given a sequence, find the most stable fold
- Design is the inverse problem associated with the forward problem of prediction
- In principle, can do design if prediction is fast; focus on prediction
- Structural stability measured by energy via Boltzmann distribution

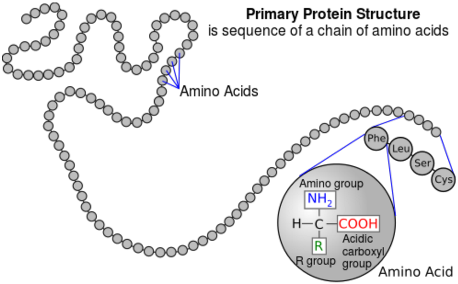
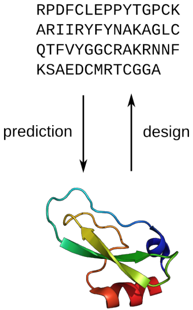


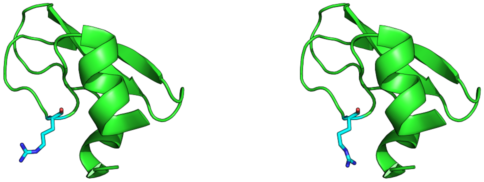
Image from Wikipedia.



RPDFCLEPPYTGPCK
ARIIRYFYNAKAGLC
QTFVYGGCRAKRNNF
KSAEDCMRTCGGA

Problem formulation

- Assume protein has a fixed backbone with flexible residue sidechains
- Each sidechain can be one of several rotamers $r_i \in R_i$
- Energy $E(\mathbf{r})$ depends on the joint rotamer configuration $\mathbf{r}$
- Goal: find $\mathbf{r}$ such that $E(\mathbf{r})$ is minimized



- NP-hard [Pierce/Winfree] but various strategies are available
- Essential to any scheme is an efficient way to compute $E(\mathbf{r})$
- One of many related formulations

Energy function

$$E = E_{\text{bonded}} + E_{\text{vdw}} + E_{\text{elec}}$$

- Bonded interactions are local/sparse
- Van der Waals interactions are short-ranged
- Electrostatic interactions are long-ranged
  - Very expensive to compute

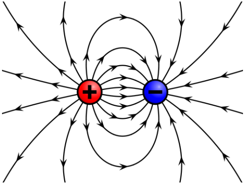
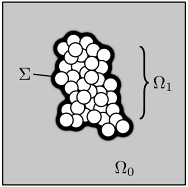


Image from Wikipedia.

In this talk, we focus on electrostatics.

Molecular electrostatics



Molecule: discrete collection of charged atoms

Ω_0: solvent

Ω_1: (solvent-excluded) molecular volume

Σ: molecular surface

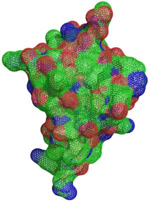
- Poisson/linearized Poisson-Boltzmann system for the electrostatic potential φ:

$$
\begin{aligned}
-(\Delta - \kappa^2)\varphi &= 0 && \text{in } \Omega_0 \\
-\Delta\varphi &= \frac{1}{\varepsilon_1}\sum_i q_i\delta(\mathbf{x}-\mathbf{x}_i) && \text{in } \Omega_1 \\
[\varphi] = \left[\varepsilon\frac{\partial\varphi}{\partial\nu}\right] &= 0 && \text{on } \Sigma
\end{aligned}
$$

- Uniform dielectric ε_i in Ω_i, inverse Debye length $\kappa(\varepsilon_0)$, charge strength q_i at $\mathbf{x}_i$
- Electrostatic energy: $E_{\text{elec}} = \frac{1}{2}\sum_i q_i\varphi(\mathbf{x}_i)$

Features of an ideal electrostatics solver for protein design

- **Accurate**: well-conditioned, controlled numerical error
- **Adaptive**: complex geometries
- **Fast**: linear or quasilinear computational complexity
- **Updatable**: reuse for local geometric perturbations



Other applications with similar requirements:

- Docking, pK_a calculations, structure refinement, charge optimization, etc.

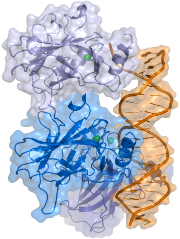
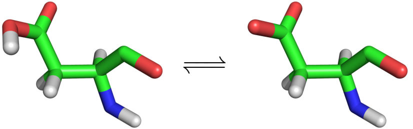


Image from Wikipedia.



Approach

Boundary integral equations

- Well-conditioned, exact interface conditions, dimensional reduction
- Contrast with finite differences or finite elements: ill-conditioning
- Formulation for LPBE [Juffer/Botta/van Keulen/van der Ploeg/Berendsen]

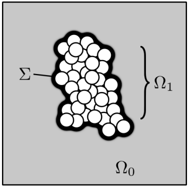
Fast direct solvers

- Directly compute compressed inverse or factorization
- Very fast solves, rapid updates
- Contrast with iterative methods: information reuse can be difficult
- Accelerate with fast-multipole–type ideas
- Main thrust of my work; many other contributors [Ambikasaran, Bebendorf, Börm, Bremer, Chandrasekaran, Chen, Corona, Darve, Gillman, Greengard, Gu, Hackbusch, Li, Martinsson, Rokhlin, Xia, Ying, Zorin]

Potential theory

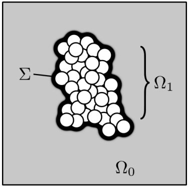
- Green's function: $G_k(\mathbf{x},\mathbf{y}) = \dfrac{e^{-k|\mathbf{x}-\mathbf{y}|}}{4\pi|\mathbf{x}-\mathbf{y}|}$
- Single-layer potential: $S_k[\sigma](\mathbf{x}) = \displaystyle\int_\Sigma G_k(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\,d\Sigma_{\mathbf{y}}$ in Ω_i
- Double-layer potential: $D_k[\mu](\mathbf{x}) = \displaystyle\int_\Sigma \frac{\partial G_k}{\partial \nu_{\mathbf{y}}}(\mathbf{x},\mathbf{y})\mu(\mathbf{y})\,d\Sigma_{\mathbf{y}}$ in Ω_i
- Jump relations as $\mathbf{x} \to \mathbf{y} \in \Sigma$:

$$
\begin{aligned}
S_k'[\sigma](\mathbf{x}) &\to \mp\frac{1}{2}\sigma(\mathbf{y}) + S_k'[\sigma](\mathbf{y}) \\
D_k[\mu](\mathbf{x}) &\to \pm\frac{1}{2}\mu(\mathbf{y}) + D_k[\mu](\mathbf{y})
\end{aligned}
$$



Boundary integral Poisson-Boltzmann system

$$
\begin{aligned}
-(\Delta - \kappa^2)\varphi &= 0 && \text{in } \Omega_0\\
-\Delta\varphi &= \frac{1}{\varepsilon_1}\sum_i q_i\delta(\mathbf{x}-\mathbf{x}_i) && \text{in } \Omega_1\\
[\varphi] &= \left[\varepsilon\frac{\partial\varphi}{\partial\nu}\right] = 0 && \text{on } \Sigma
\end{aligned}
$$



- Integral representation of solution:

$$
\varphi \equiv \begin{cases} S_\kappa\sigma + D_\kappa\mu & \text{in } \Omega_0\\ S_0\sigma + \alpha D_0\mu + \varphi_s & \text{in } \Omega_1\end{cases} \qquad \alpha = \frac{\varepsilon_0}{\varepsilon_1}, \quad \varphi_s(\mathbf{x}) = \frac{1}{\varepsilon_1}\sum_i q_i G_0(\mathbf{x},\mathbf{x}_i)
$$

- Interface conditions give equation for (σ,μ) on Σ (second-kind Fredholm):

$$
\begin{aligned}
\frac{1}{2}(1+\alpha)\mu + (S_\kappa - S_0)\sigma + (D_\kappa - \alpha D_0)\mu &= \varphi_s,\\
-\frac{1}{2}(1+\alpha)\sigma + (\alpha S'_\kappa - S'_0)\sigma + \alpha(D'_\kappa - D'_0)\mu &= \frac{\partial\varphi_s}{\partial\nu}
\end{aligned}
$$

[Juffer/Botta/van Keulen/van der Ploeg/Berendsen]

Properties of integral equation matrices

Dense integral equation matrix $A \in \mathbb{C}^{N\times N}$

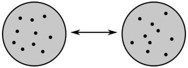
- Cost of applying A: $O(N^2)$
- Cost of inverting A: $O(N^3)$

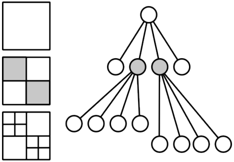
Basic idea for acceleration:

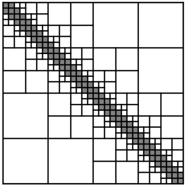
- Low-rank off-diagonal blocks, exploit rank structure hierarchically
- FMM: matrix-vector multiplication in $O(N)$ work [Greengard/Rokhlin]

Fast direct solvers

- $\mathcal{H}$-matrices, HSS matrices, recursive skeletonization, etc.







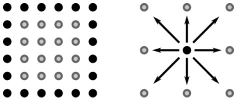
Interpolative decomposition

If $A_{:,q}$ is numerically low-rank, then there exist

- skeleton ($\hat{q}$) and redundant ($\check{q}$) columns partitioning $q = \hat{q} \cup \check{q}$
- an interpolation matrix T_q

such that

$$A_{:,\check{q}} \approx A_{:,\hat{q}} T_q.$$



- Essentially a pivoted QR written slightly differently:

$$A_{:,(\hat{q},\check{q})} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} \approx Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix}$$

$$\implies A_{:,\check{q}} \approx Q_1 R_{12} = \underbrace{Q_1 R_{11}}_{A_{:,\hat{q}}} \underbrace{\left(R_{11}^{-1} R_{12} \right)}_{T_q}$$

- Rank-revealing to any specified precison $\epsilon > 0$

[Cheng/Gimbutas/Martinsson/Rokhlin, Gu/Eisenstat]

Skeletonization

- Let $A = \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} \end{bmatrix}$ with A_{pq} and A_{qp} low-rank

- Apply ID to $\begin{bmatrix} A_{qp} \\ A^*_{pq} \end{bmatrix}$: $\quad \begin{bmatrix} A_{q\check{p}} \\ A^*_{\check{p}q} \end{bmatrix} \approx \begin{bmatrix} A_{q\hat{p}} \\ A^*_{\hat{p}q} \end{bmatrix} T_p \implies \begin{array}{c} A_{q\check{p}} \approx A_{q\hat{p}} T_p \\ A_{\check{p}q} \approx T_p^* A_{\hat{p}q} \end{array}$

- Reorder $A = \begin{bmatrix} A_{\check{p}\check{p}} & A_{\check{p}\hat{p}} & A_{\check{p}q} \\ A_{\hat{p}\check{p}} & A_{\hat{p}\hat{p}} & A_{\hat{p}q} \\ A_{q\check{p}} & A_{q\hat{p}} & A_{qq} \end{bmatrix}$, define $Q_p = \begin{bmatrix} I & & \\ -T_p & I & \\ & & I \end{bmatrix}$

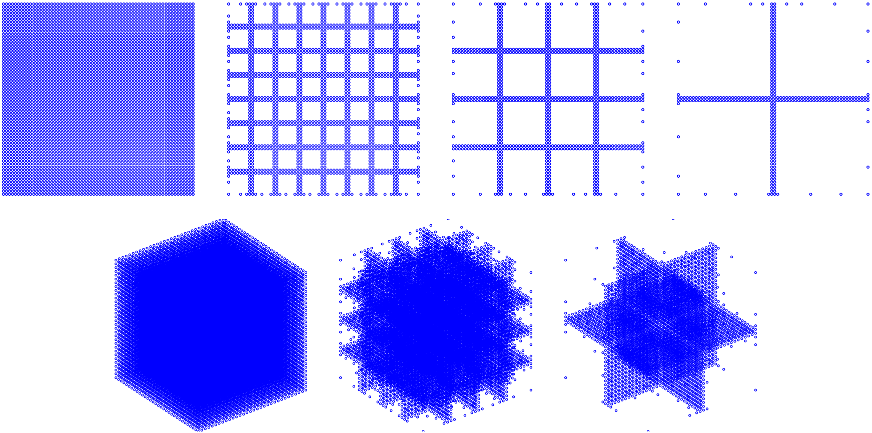
- Sparsify via ID: $\quad Q_p^* A Q_p \approx \begin{bmatrix} * & * & \\ * & A_{\hat{p}\hat{p}} & A_{\hat{p}q} \\ & A_{q\hat{p}} & A_{qq} \end{bmatrix} \xrightarrow{\text{elim}} \begin{bmatrix} * & & \\ & * & A_{\hat{p}q} \\ & A_{q\hat{p}} & A_{qq} \end{bmatrix}$

- Reduces to a subsystem involving skeletons only

[Ho/Ying, Xia/Xi/Gu]

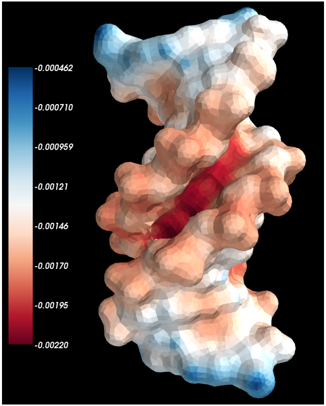
Recursive skeletonization factorization

- Skeletonize cells hierarchically up a tree
- Analogous to nested dissection multifrontal method [Duff/Reid, George]



[Gillman/Young/Martinsson, Ho/Greengard, Ho/Ying, Martinsson/Rokhlin]

RSF for molecular electrostatics

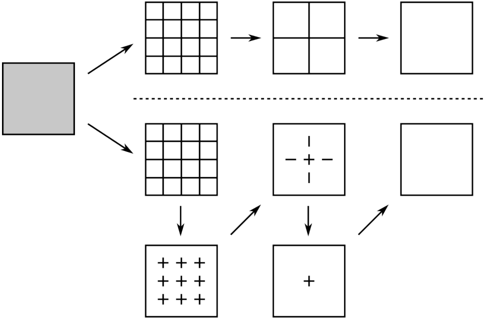


- Computational complexities
  - Factorization: $O(N^{3/2})$
  - Solve: $O(N \log N)$
- Suboptimal but hopefully fast like MF
- DNA system with $N = 19752$, $\epsilon = 10^{-3}$
  - FMM/GMRES: 30 s
  - RSF factorization: 10 min
  - RSF solve: 0.1 s
- Break-even point: 20 solves
- Effective for small molecules
- Does not scale well to macromolecules ($N \gtrsim 10^6$)

[Ho/Greengard]

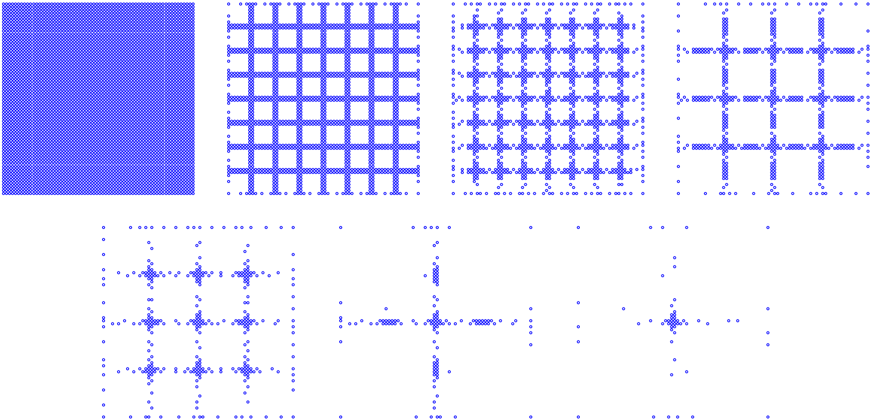
Accelerating RSF

- RSF: $O(N)$ in 1D, $O(N^{3/2})$ in 2D, $O(N^2)$ in 3D
- Superlinear cost in 2D/3D due to skeleton growth
- Skeletons cluster near cell interfaces by Green's theorem
- Exploit skeleton geometry by further skeletonizing along interfaces
- Recursive dimensional reduction [Corona/Martinsson/Zorin, Xia/Chandrasekaran/Gu/Li]



Hierarchical interpolative factorization in 2D

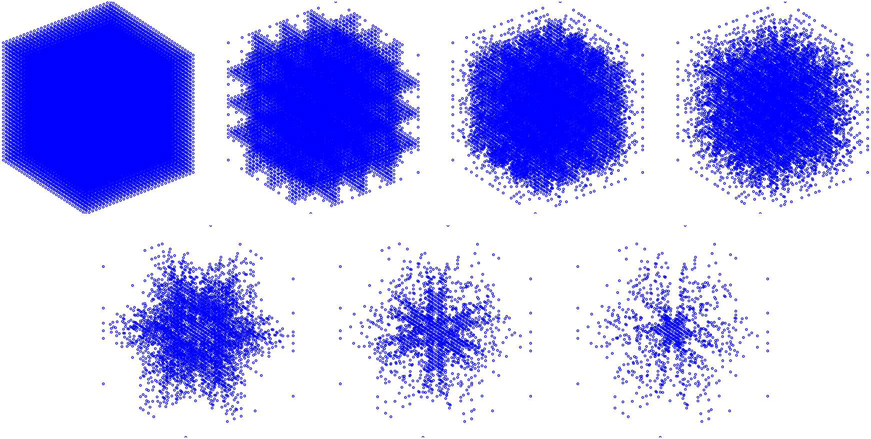
- Skeletonize cells (2D), then edges (1D) hierarchically up a tree



[Ho/Ying]

Hierarchical interpolative factorization in 3D

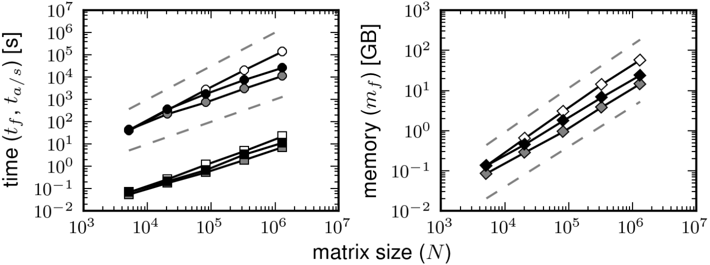
- Skeletonize cells (3D), then faces (2D), then edges (1D) hierarchically up a tree



[Ho/Ying]

Numerical results for HIF

Second-kind equation for interior Dirichlet Laplace on the sphere at $\epsilon = 10^{-3}$:



- **rskelf3** (white), **hifie3** (gray), **hifie3x** (black)
- Factorization time (○), solve time (□), memory (◇)
- Reference scalings (gray dashes):
  - Left: $O(N)$ and $O(N^{3/2})$
  - Right: $O(N)$ and $O(N \log N)$

[Ho/Ying]

Remarks on HIF

- Efficient factorization of structured operators in 2D/3D
- Empirical **linear complexity** but no proof yet
- Constructs approximate generalized LU decomposition
  - Fast matrix-vector multiplication (generalized FMM)
  - Fast direct solver at high accuracy, preconditioner otherwise
- Extensions: $A^{1/2}$, $\log \det A$, $\operatorname{diag} A^{-1}$
- Modification for sparse PDEs based on MF
- Highly parallelizable [with A. Benson, Y. Li, J. Poulson, L. Ying]
- MATLAB codes freely available at `https://github.com/klho/FLAM/`

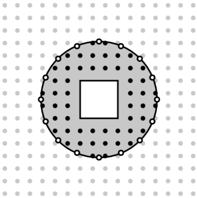
Updating

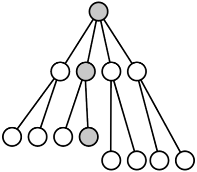
Matrix augmentation [Greengard/Gueyffier/Martinsson/Rokhlin]:

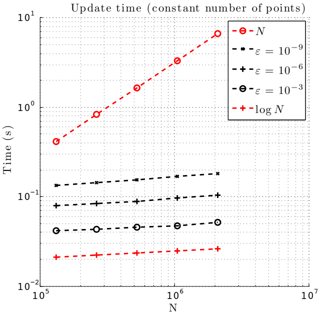
- Local geometric perturbations as low-rank updates of an augmented base matrix
- Sherman-Morrison-Woodbury: rank $k \implies O(Nk)$ cost

Full updating [with A. Damle, V. Minden, L. Ying]:

- Use Green's theorem to localize effect of perturbation
- Redo computation up only one branch of the tree: $O(\log N)$ cost
- Can accumulate updates







Summary

- **Problem**: electrostatics in protein design
- **Goal**: accurate, adaptive, fast, updatable methods
- Achieved using boundary integral equations and fast direct solvers
- To do: test HIF on real macromolecular geometries
- Remaining issue of how to locally remesh after perturbation
- Pieces slowly coming together, future looks promising
- Aim to incorporate into structural biology software

References

- H. Cheng, Z. Gimbutas, P.G. Martinsson, V. Rokhlin. On the compression of low rank matrices. SIAM J. Sci. Comput. 26 (4): 1389–1404, 2005.
- E. Corona, P.-G. Martinsson, D. Zorin. An $O(N)$ direct solver for integral equations on the plane. Preprint, arXiv:1303.5466 [math.NA], 2013. To appear in Appl. Comput. Harmon. Anal.
- A. Gillman, P.M. Young, P.-G. Martinsson. A direct solver with $O(N)$ complexity for integral equations on one-dimensional domains. Front. Math. China 7 (2): 217–247, 2012.
- L. Greengard, D. Gueyffier, P.-G. Martinsson, V. Rokhlin. Fast direct solvers for integral equations in complex three-dimensional domains. Acta Numer. 18: 243–275, 2009.
- L. Greengard, V. Rokhlin. A fast algorithm for particle simulations. J. Comput. Phys. 73: 325–348, 1987.
- M. Gu, S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. SIAM J. Sci. Comput. 17 (4): 848–869, 1996.
- K.L. Ho, L. Greengard. A fast direct solver for structured linear systems by recursive skeletonization. SIAM J. Sci. Comput. 34 (5): A2507–A2532, 2012.
- K.L. Ho, L. Ying. Hierarchical interpolative factorization for elliptic operators: differential equations. Preprint, arXiv:1307.2895 [math.NA], 2013.
- K.L. Ho, L. Ying. Hierarchical interpolative factorization for elliptic operators: integral equations. Preprint, arXiv:1307.2666 [math.NA], 2013.
- A.H. Juffer, E.F.F. Botta, B.A.M. van Keulen, A. van der Ploeg, H.J.C. Berendsen. The electric potential of a macromolecule in a solvent: A fundamental approach. J. Comput. Phys. 97: 144–171, 1991.
- P.G. Martinsson, V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. J. Comput. Phys. 205: 1–23, 2005.
- N.A. Pierce, E. Winfree. Protein design is *NP*-hard. Protein Eng. 15 (10): 779–782, 2002.
- J. Xia, S. Chandrasekaran, M. Gu, X.S. Li. Superfast multifrontal method for large structured linear systems of equations. SIAM J. Matrix Anal. Appl. 31 (3): 1382–1411, 2009.
- J. Xia, Y. Xi, M. Gu. A superfast structured solver for Toeplitz linear systems via randomized sampling. SIAM J. Matrix Anal. Appl. 33 (3): 837–858, 2012.