Boundary-Integral Methods in Molecular Science and Engineering

Lecture 3: Models on Models.

Jaydeep P. Bardhan

Rush University Medical Center

jaydeep_bardhan@rush.edu

Outline for Lectures

- ✓ Biology is awesome. If you can solve Poisson, you can join in the fun!
- ✓ There's more than one way to skin a cat. Sometimes PDEs can be advantageously reframed as *integral equations*.
- ✓ There's no such thing as a free lunch (or, what it takes to solve really big problems)
- A diversity of unusual computational challenges will continue to drive biological simulation.

Today:

- Interfaces between models and numerics

- Examples:
  - Electrostatic optimization
  - Approximate local electrostatics

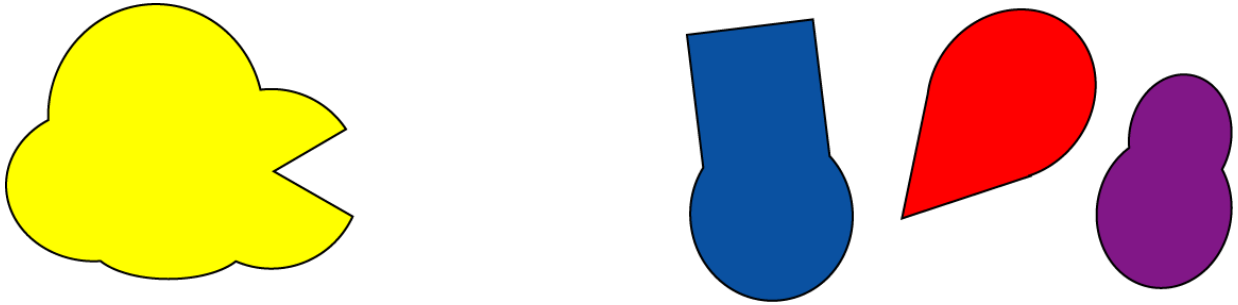
The Crucial Role of Interfaces

- Prof. Spiegelman talked yesterday about the idea of exploring *model space*, meaning PDE models as hypotheses about geophysics and geodynamics
  - It is rarely worth betting on the universal applicability of implementation details

- Today: a PDE model employed as a means to explore the origins of molecular binding affinity and specificity
  - The PDE model is *not* the hypothesis.
  - Here, it is sensible to re-engineer the interface between the PDE model and the formalism built *on top of it*
  - Exposing more details about the PDE led to a new form of approximate model more rigorous than competing approximations

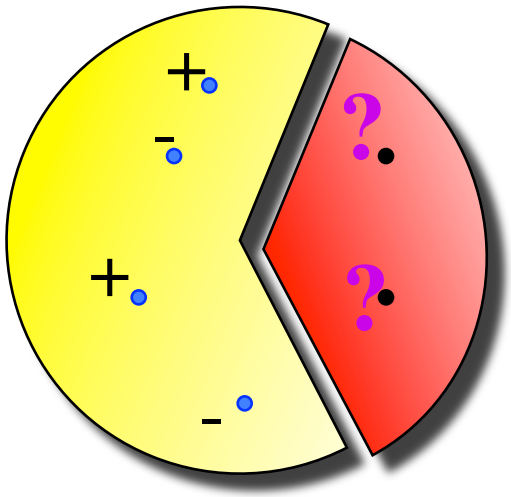
Biomolecule Electrostatic Optimization

- A molecular design problem: optimize a molecule (*ligand*) for tight binding to a target (*receptor*)



- Estimating binding free energies:

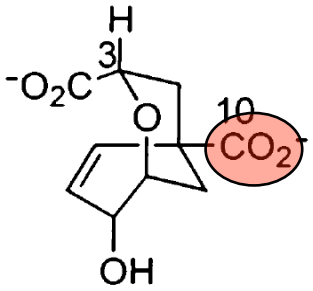
$$\Delta G^0 = \Delta G^0_{\text{non es}} + \Delta G^0_{\text{es}}$$

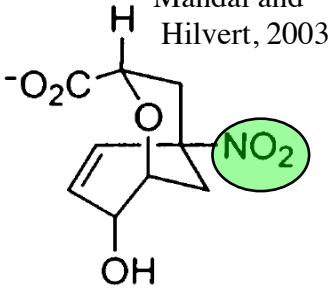


- The electrostatic problem:
  - Take ligand shape as given
  - What **charge distribution** gives the best binding free energy?

Electrostatic Optimization of Biomolecules:
Applications in Analysis and Design

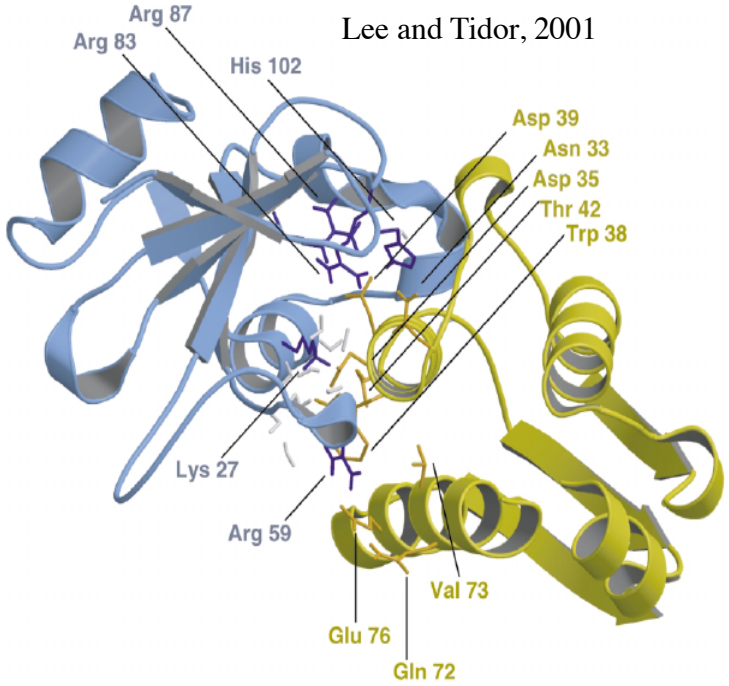
- *E. coli* chorismate mutase inhibitors:
  - Analyzed by Kangas and Tidor
  - Suggested substitution experimentally verified: result is the tightest-binding inhibitor yet known





Mandal and Hilvert, 2003

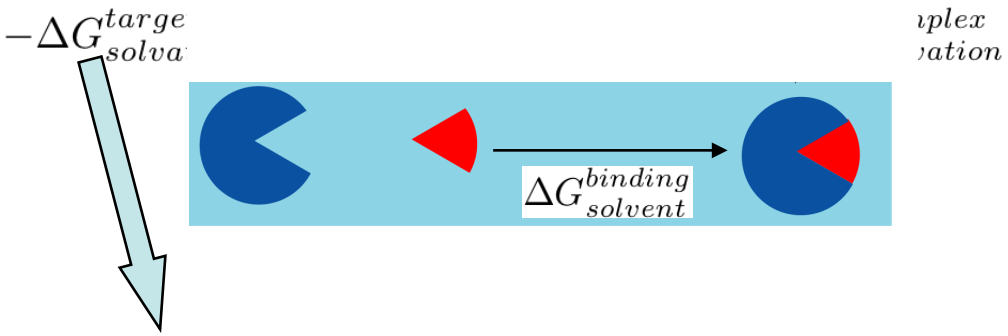
- Barnase/barstar protein complex:
  - Tight-binding complex
  - Optimal charge distribution closely matches “wild-type” charge distribution



Lee and Tidor, 2001

Reminder: Binding Is A Trade-off

- Molecular binding involves sacrificing solute--solvent interactions for solute--solute interactions:



$$\Delta G^0_{\text{solv}} = \Delta G^0_{\text{non es}} + \Delta G^0_{\text{es}}$$

$$\Delta G^{bind} = \gamma \Delta SA + \Delta G^{es}_{L-R} - \Delta G^{es}_{L} - \Delta G^{es}_{R}$$

δ = 0 δ = 1

This is only a VERY SIMPLE MODEL for molecular binding!

The Reaction-Potential Matrix

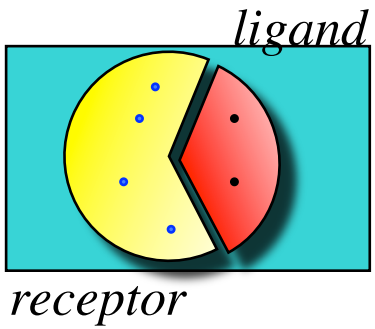
- A weighted combination of charge distributions in the solute molecule produces a weighted combination of the individual responses:

$$\varphi^{\mathrm{REAC}} = Lq \Longrightarrow E = \frac{1}{2}\varphi^{\mathrm{REAC},T} q = \frac{1}{2} q^T L q$$

- The “canonical” basis is the natural, atom-based point of view
- We can also use the eigenvector basis for analysis!
- In comparing models we don’t just have to use the total electrostatic solvation free energy
  - This, too, is a sort of “interface”
  - We will revisit this point shortly

The Electrostatic Optimization Problem:

Assume ligand rigidity, and no charge transfer:

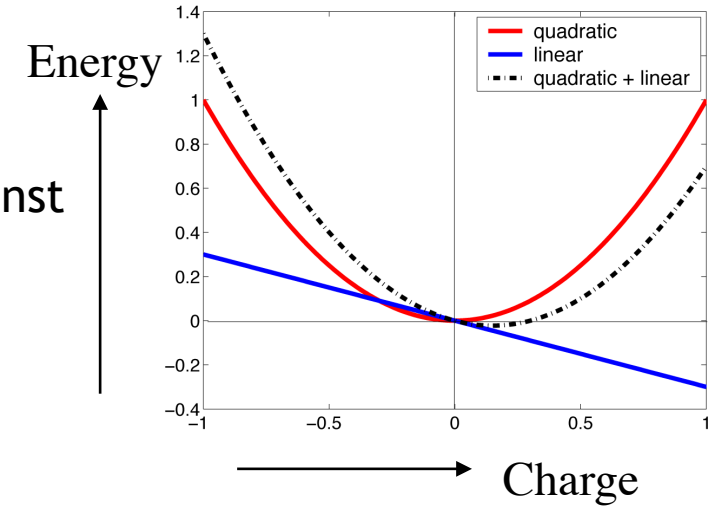


$$\Delta G^{0}_{\mathrm{var,es}} = \mathrm{E}\left(\right) - \mathrm{E}\left(\right)$$

- Under our assumptions, this energy function is convex
- The idea: It always *costs energy* to remove the water from the receptor volume

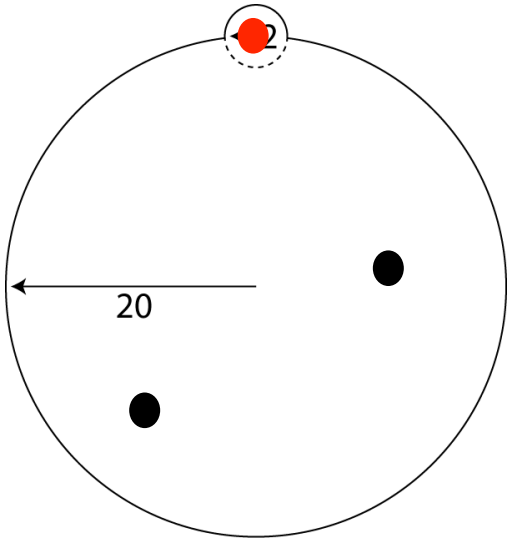
- May also want to enforce *constraints*

- The optimal charge distribution…
  - … balances the "desolvation penalty" against ligand-receptor interactions
  - … is a *guide* for design
  - … serves as a template and benchmark



Simple Case of Optimization: A Single Ion

$$A_u x = B_u q$$
$$\varphi = C_u \left(A_u^{-1} B_u \right) q$$

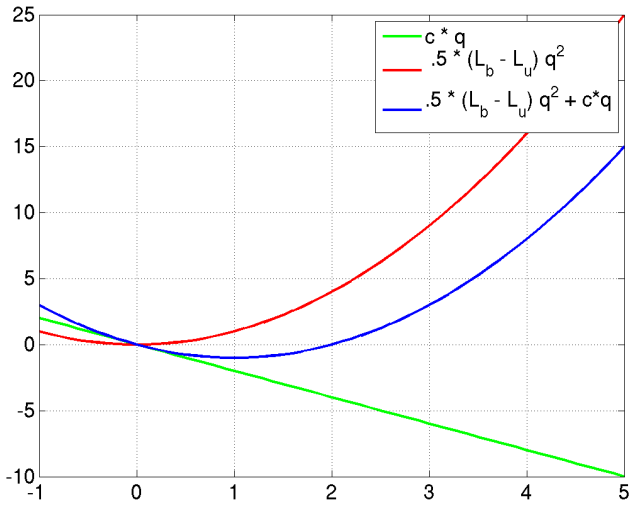


$$\Delta G^{es} = \frac{1}{2} q^T (C_u A_u^{-1} B_u) q \qquad \Delta G^{es} = \frac{1}{2} q^T (C_b A_b^{-1} B_b) q + (c_{Coul} + c_{REAC})^T q$$

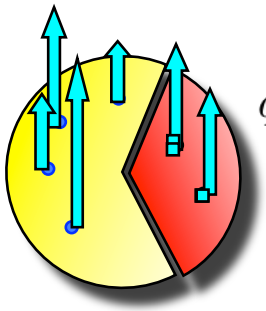
- The electrostatic contribution to binding is

$$\Delta G^{es,var} = \frac{1}{2} (L_b - L_u) q^2 + cq$$

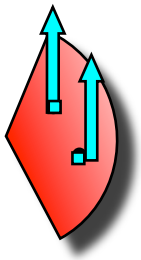
- A total of three simulations is needed.



Optimization in Multiple Dimensions



$$q_i = 1$$
$$q_j = 0 \quad \text{if} j \neq i$$



$$A_b \sigma_b = B_b q_L$$
$$\varphi_b^{reac} = C_b \varphi_b^{surf}$$

$$A_u \sigma_u = B_u q_L$$
$$\varphi_u^{reac} = C_u \varphi_u^{surf}$$

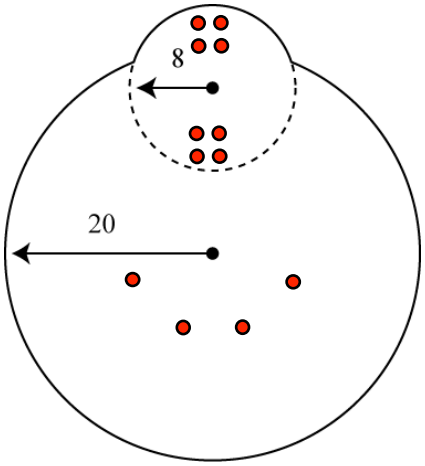
$$L_b = \left[\begin{array}{ccc} | & & | \\ \varphi_{b,1}^{reac} & \cdots & \varphi_{b,n}^{reac} \\ | & & | \end{array} \right] \qquad L_u = \left[\begin{array}{ccc} | & & | \\ \varphi_{u,1}^{reac} & \cdots & \varphi_{u,n}^{reac} \\ | & & | \end{array} \right]$$

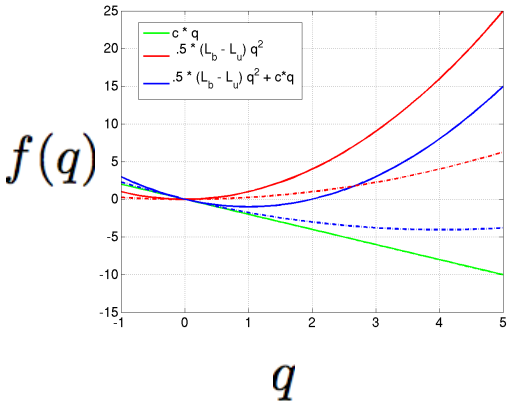
$$L = L_b - L_u$$

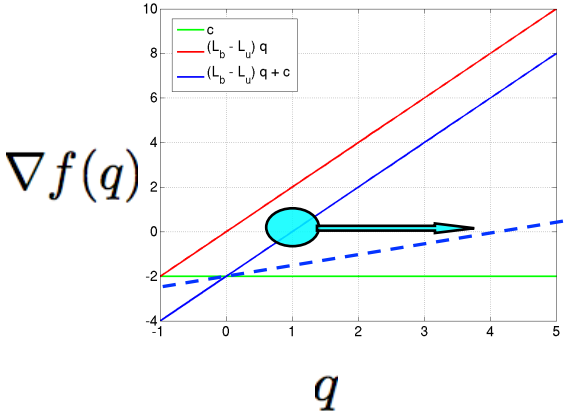
Regularizing Electrostatic Optimization Problems

- The Hessian matrix can have many (or even most) of its eigenvalues close to zero

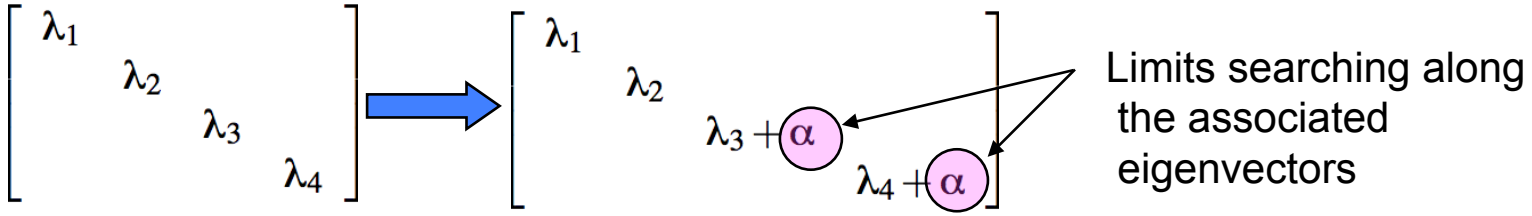
$$\nabla f(q) = 0 \Longrightarrow (L_b - L_u)q = -c$$







- Adding a penalty function is easy enough when one has an explicit Hessian--use eigendecomposition:



Limits searching along the associated eigenvectors

One Approach to Accelerated Optimization

- The unconstrained problem can be solved by nesting Krylov methods:

$$\hat{q}^i \in \text{span}\{-c, -(L_b - L_u)c, \ldots, -(L_b - L_u)^{i-1}c\}$$

  - Two Krylov solves are required for each application of $L_b - L_u$
  - Effectively, treat the PDE solver and the optimization method as "black boxes"
  - This approach is known in some communities as a *nested analysis and design* method

- Pros:
  - Easy to implement

- Cons:
  - Performance will depend on finding a good preconditioner
  - Unclear how to regularize
  - Seems wasteful! Two full electrostatic solves at each outer Krylov step?

Another Natural Approach:
Simultaneous Analysis and Design

- Include the state variables (associated with the simulation) as decision variables

minimize
$$\frac{1}{2}\begin{bmatrix} q \\ x_b \\ x_u \end{bmatrix}^T \begin{bmatrix} 0 & \frac{1}{2}C_b & -C_u \\ \frac{1}{2}C_b^T & & \\ -\frac{1}{2}C_u^T & & \end{bmatrix} \begin{bmatrix} q \\ x_b \\ x_u \end{bmatrix} + \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} q \\ x_b \\ x_u \end{bmatrix}$$

subject to
$$\begin{bmatrix} -B_b & A_b & \\ -B_u & & A_u \end{bmatrix} \begin{bmatrix} q \\ x_b \\ x_u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Pros:
  - These methods are well-known (see, *e.g.*, Biros *et al.*)
- Cons:
  - Requires an adjoint solve in addition to standard solve
  - Seems like "overkill" for the simple relation between the objective and the decision variables (charges)
  - Regularization still problematic

A Novel Method: The Reverse-Schur Approach

- For these PDE constraints, we really only need to solve multiple systems simultaneously:

$$M_3 M_2^{-1} M_1 y = b \Longrightarrow \begin{bmatrix} & M_3 \\ M_1 & -M_2 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

- The unconstrained problem is therefore

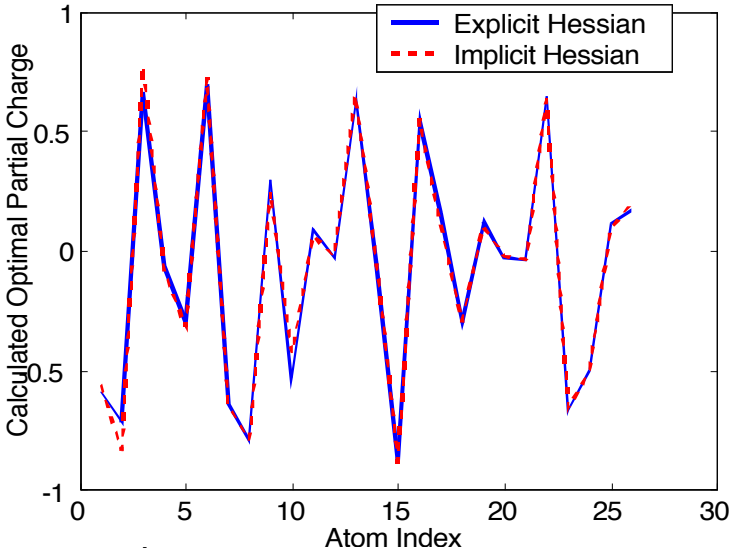
$$(L_b - L_u)q = -c \Longrightarrow \begin{bmatrix} & C_b & -C_u \\ -B_b & A_b & \\ -B_u & & A_u \end{bmatrix} \begin{bmatrix} q \\ x_b \\ x_u \end{bmatrix} = \begin{bmatrix} -c \\ 0 \\ 0 \end{bmatrix}$$

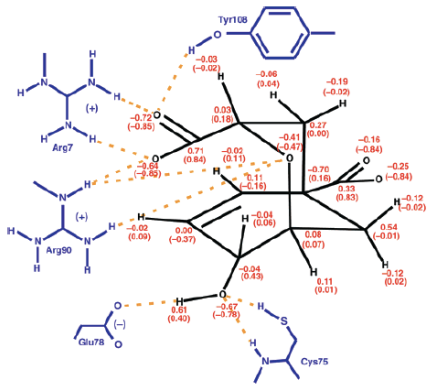
- Pros:
  - Easily solved using preconditioned Krylov methods
  - No adjoint solve needed
- Cons:
  - Regularization is **still** an issue

Proof-Of-Concept Implementation

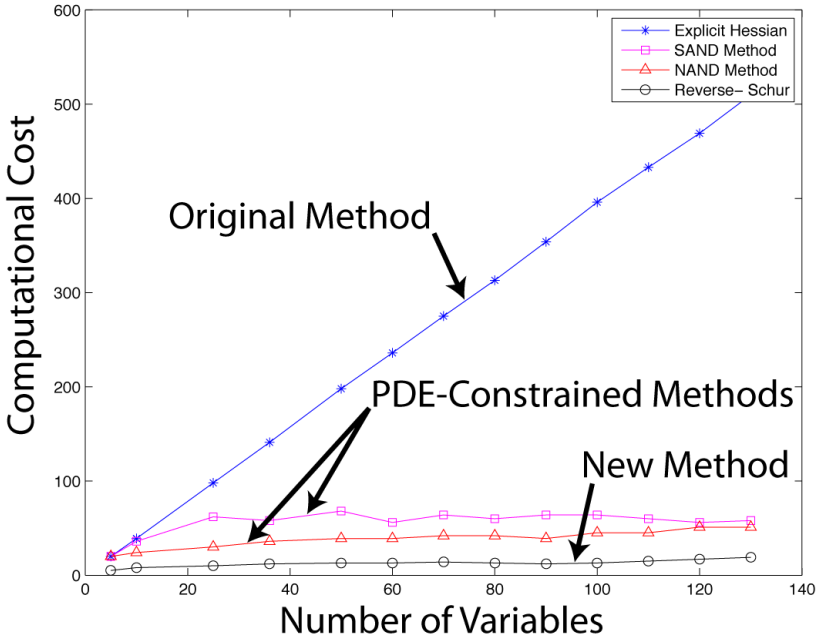
- A full-scale solver was implemented using PETSc and precorrected-FFT

Computed charges agree closely





Method scales comparably with normal PDE-constrained approaches

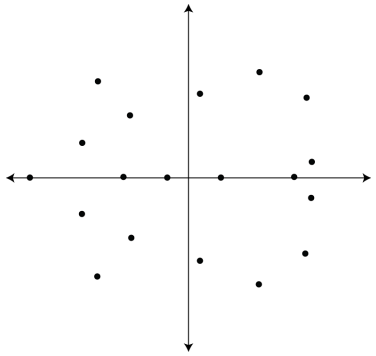


This test problem does not need any regularization!

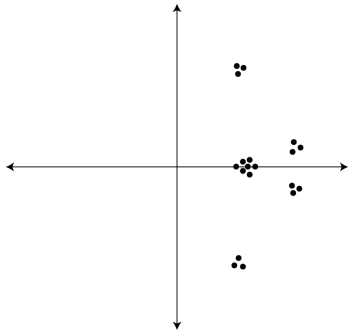
Bardhan, Altman, Lee, Tidor, White, 2004

A Quick Reminder About Preconditioners

- Krylov convergence rate depends on the matrix eigenvalues having some "nice" properties such as eigenvalue clustering:



Bad: eigenvalues not clustered. Many iterations will be required!



Good: eigenvalues tightly clustered. Few iterations will be required!

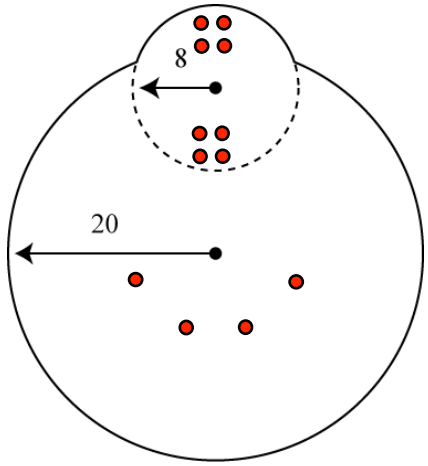
- The goal is to find a "preconditioner" matrix *P* that clusters the eigenvalues of *A* so it will take fewer applications of *A* to solve

$$PAx = Pb$$

- The *ideal* preconditioner is A^{-1}: all eigenvalues are mapped to unity. For a *diagonally dominant* (or nearly so) matrix A, the diagonal entries often work well enough.

Regularization in Implicit-Hessian Approaches

- As we have seen, breaking the interface between optimization and simulation complicates regularization



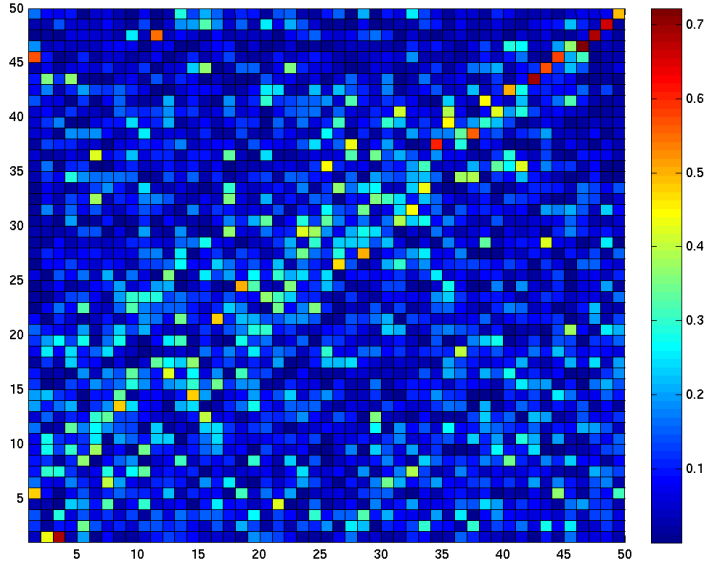
- One needs an approximation that gets the eigenvectors of the desolvation matrix right, and the eigenvectors at least ranked correctly

- Use the Krylov preconditioner on the *Green's theorem* formulation to com an approximate Hessian:

$$\begin{bmatrix} \frac{1}{2}I + D^a_{I,a} & -S^a_{I,a} \\ \frac{1}{2}I - D^a_{II,a} & \epsilon_{I,II} S^a_{II,a} \end{bmatrix} \begin{bmatrix} \phi_a \\ \frac{\partial \phi_a}{\partial n} \end{bmatrix} = \begin{bmatrix} \sum_i \frac{q_i}{\epsilon_I} G^a_{I,i} \\ 0 \end{bmatrix}$$

$$H = C_b A_b^{-1} B_b - C_u A_u^{-1} B_u$$

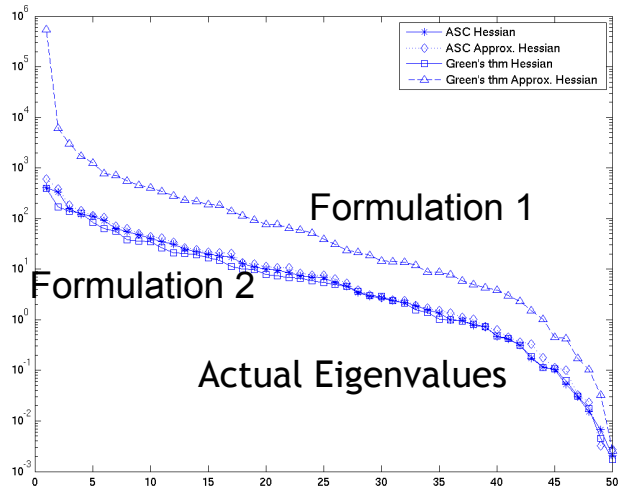
$$\hat{H} = C_b P_b B_b - C_u P_u B_u$$



Implementation Issue:
Impact of the Integral Formulation

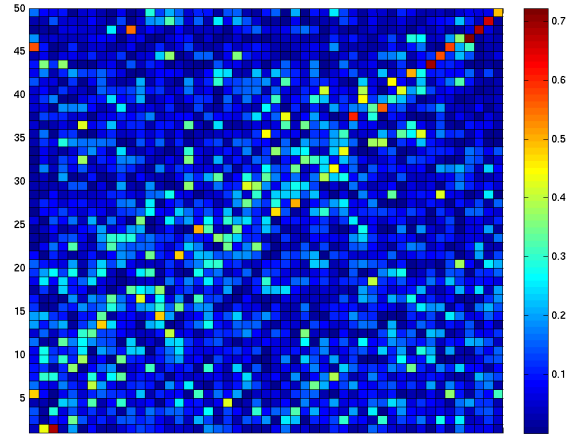
- Surface-charge formulation generates superior Hessian approximations

Eigenvalues

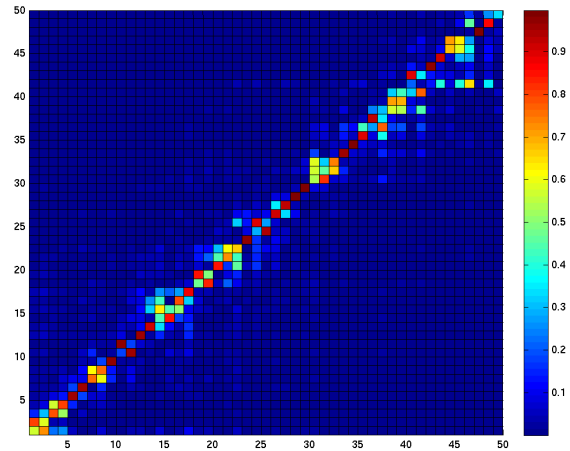


Approximate Eigenvectors Projected onto Calculated Eigenvectors

Green's thm



Surface Charge



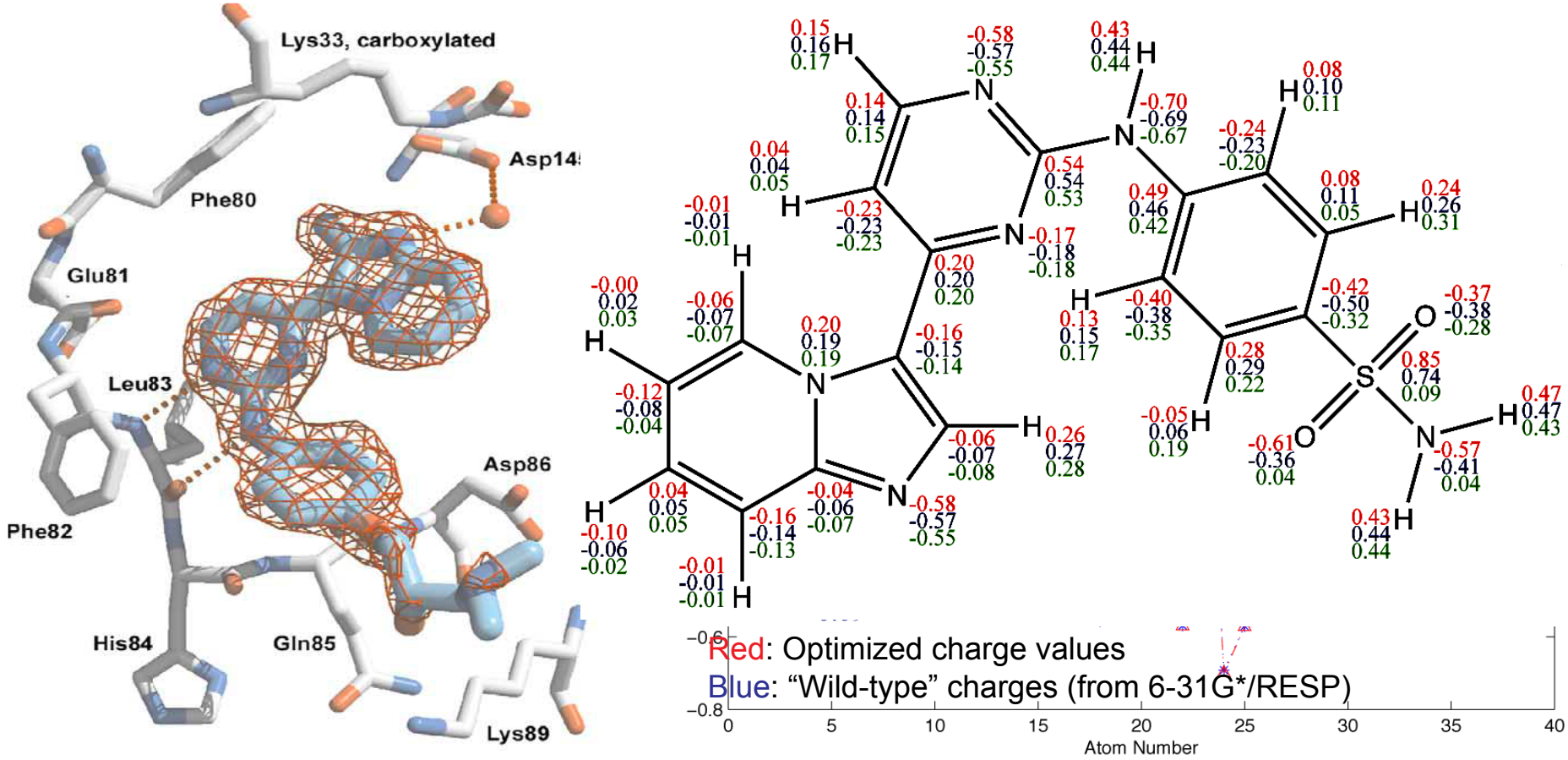
(similar plots obtained for both flat and curved panels)

- Regularization can be performed using "approximate" penalty functions
- Varying the penalty function can be done approximately:

$$\delta q^* \approx -\left(\hat{H}+W\right)^{-1} U \left(I + V^T \left(\hat{H}+W\right)^{-1} U\right)^{-1} V^T q^*,$$

Application: Cyclin-Dependent Kinase 2 and Inhibitor

PDE-constrained optimization is **almost 200 times faster** for this **small** molecule



Anderson, et al. 2003 (not exactly the optimized ligand)

Bardhan *et al.*, (submitted)

Boundary-Element Preconditioners Give a New Electrostatic Model

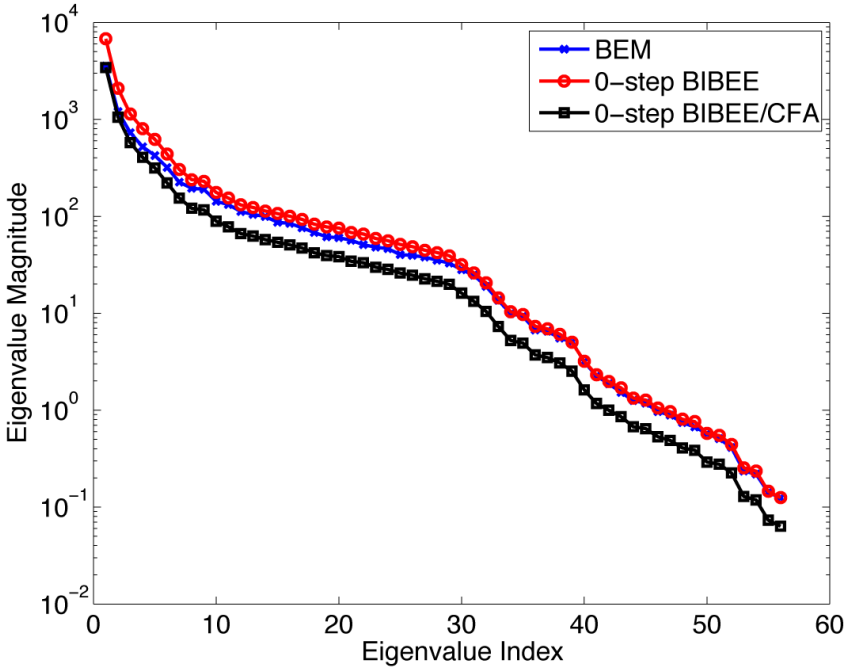
- We have used a boundary-element preconditioner *P* that takes the diagonal matrix elements:

$$PAx = PBq$$
$$\hat{x} \in PBq, PAPBq, \ldots$$

- This is tantamount to assuming that there is *no* contribution from the remainder of the operator

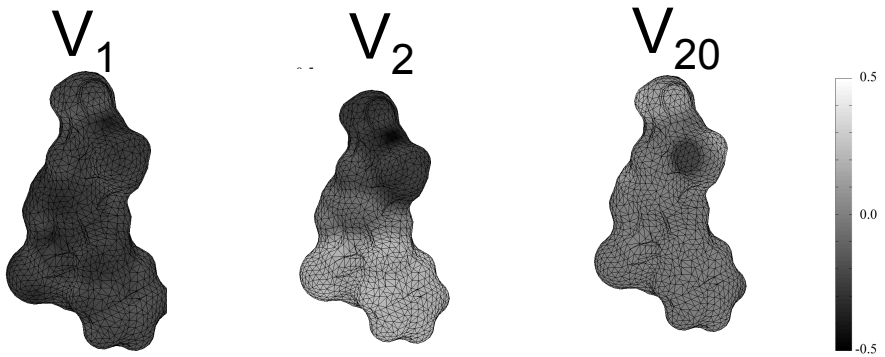
$$(I - \hat{\epsilon}D - \hat{\epsilon}E)\sigma = \hat{\epsilon}f$$

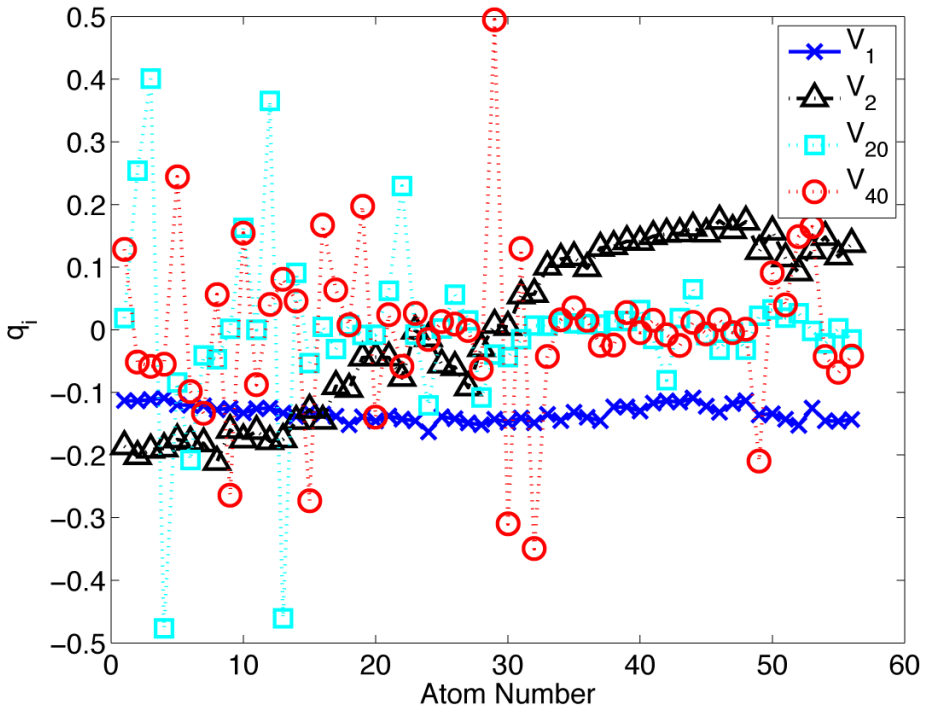
The BIBEE/P approximation estimates the *smallest* eigenvalues accurately and overestimates the large eigenvalues.



Accuracy Dependence on Charge Distribution

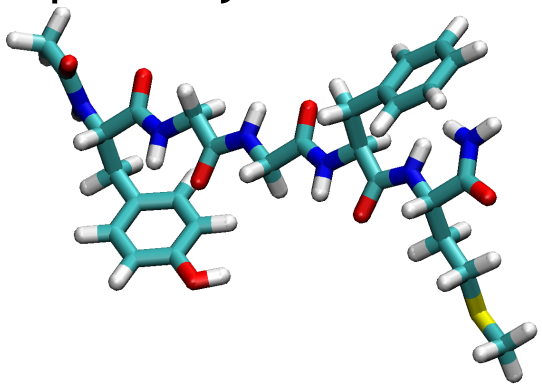
- The largest eigenvalues are most accurately predicted by BIBEE/CFA
- Look at V_1: the induced displacement fields are “like” low-order multipoles
- Small eigenvalues --> rapidly varying displacement fields, and these are approximated poorly

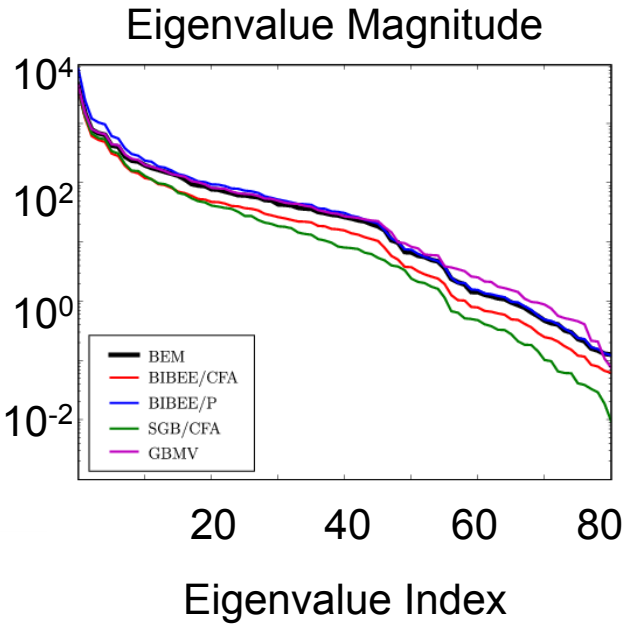




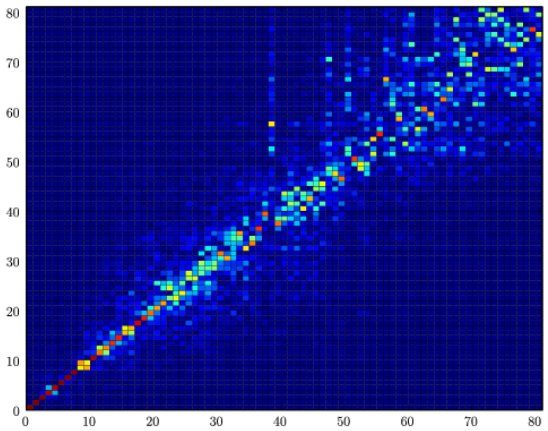
Comparison to Previous Approaches

- Met-enkephalin has 5 residues and 81 atoms
- Widely used in computational studies of peptide dynamics

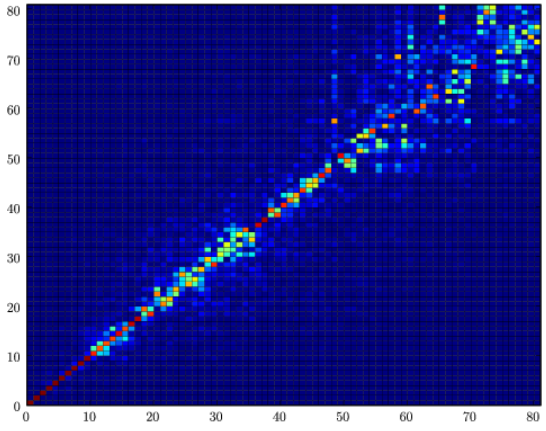




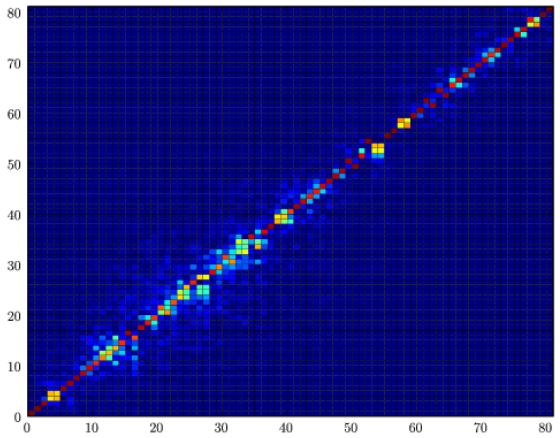
SGB/CFA



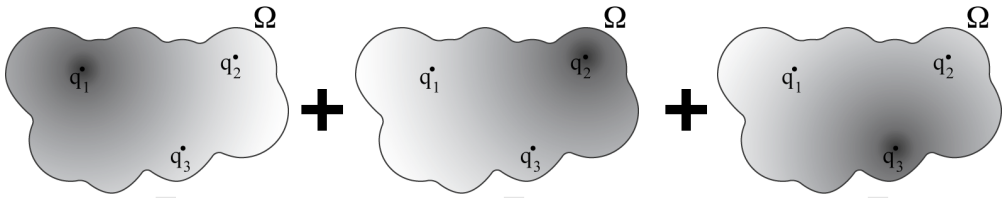
GBMV



BIBEE



Coulomb-Field Approximation: GB and BIBEE



$$(I - \hat{\epsilon}\mathcal{K})\sigma = \hat{\epsilon}\sum_i f_i$$

BIBEE approx. charge includes **all** contributions

$$\tilde{\sigma} = \hat{\epsilon}\sum f_i$$

Coulomb-field approximation: corresponds exactly to ignoring the integral operator.

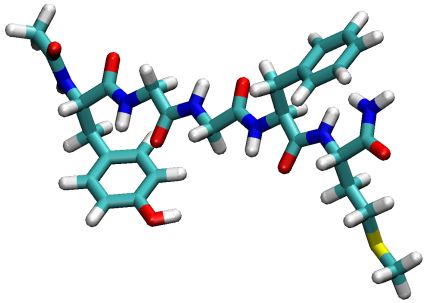
BIBEE/CFA is the extension of GB/CFA to multiple charges!

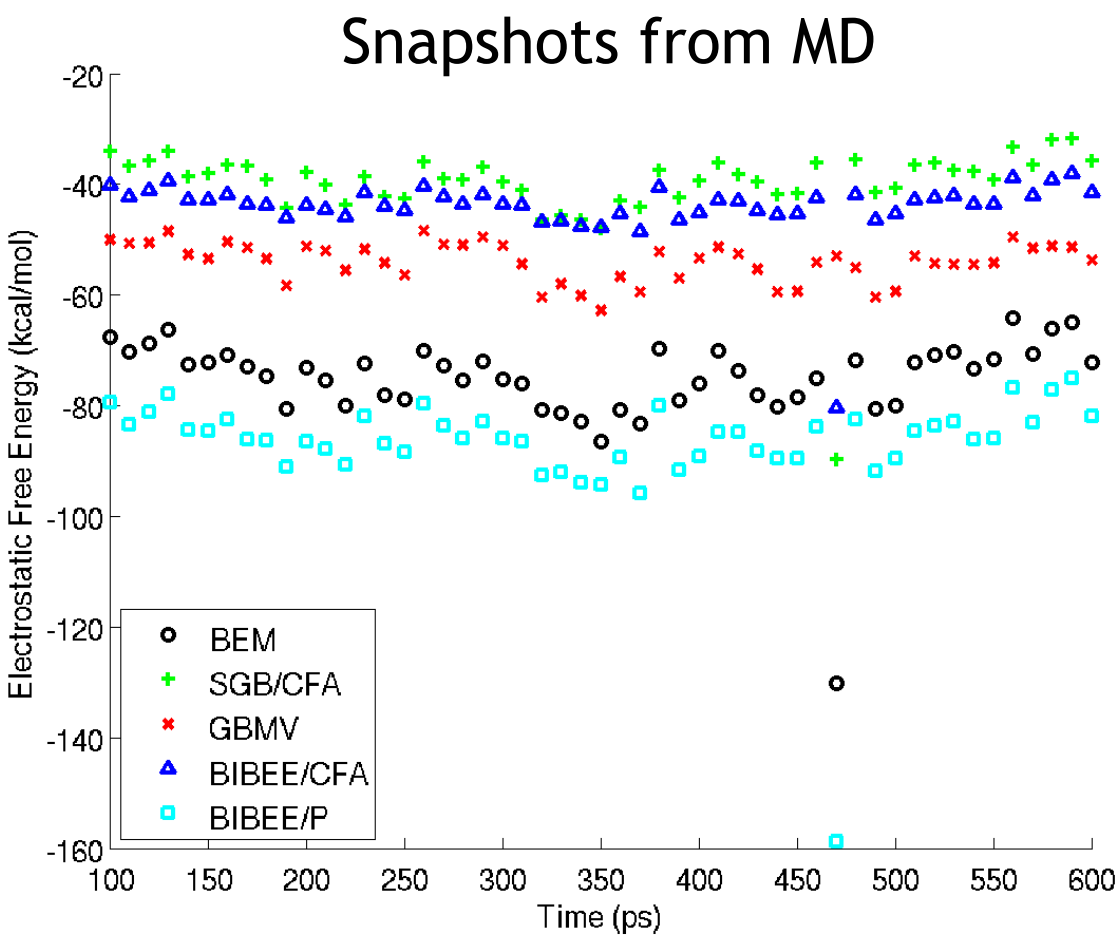
No *ad hoc* parameters, no heuristic interpolation

BIBEE Is An Accurate, Parameter-Free Model

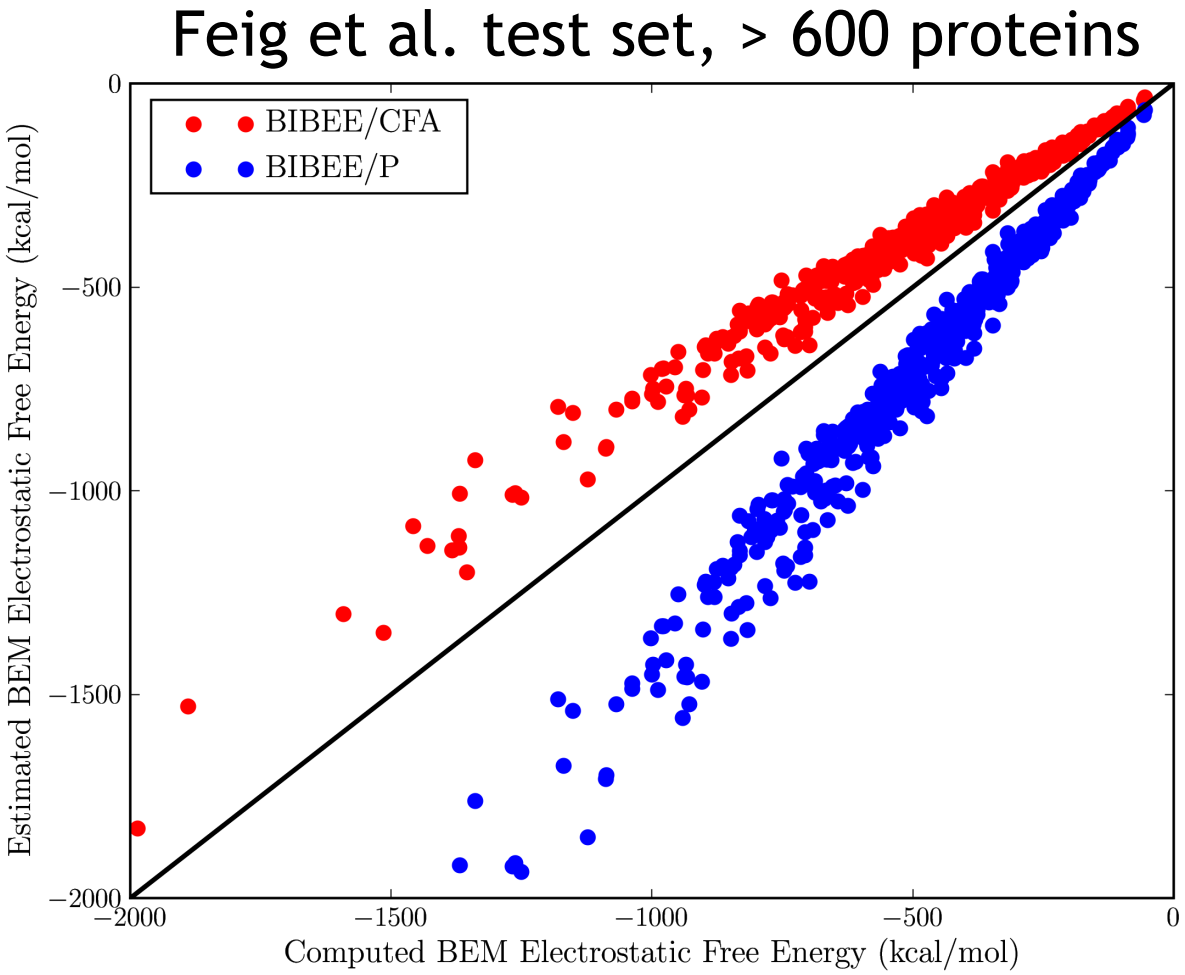
- Peptide example

  Met-enkephalin





BIBEE/CFA Energy Is a Provable Upper Bound

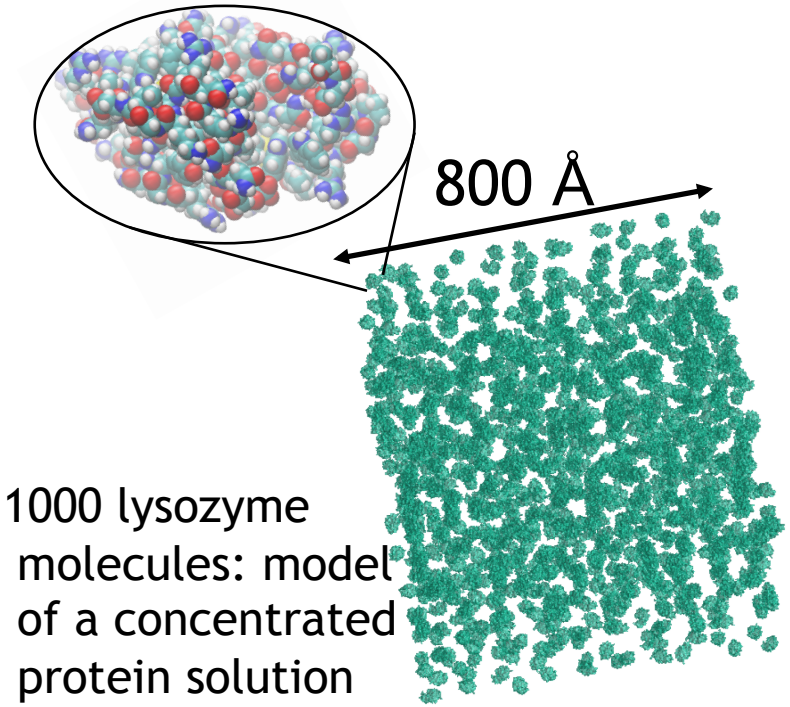


- BIBEE/P is an effective lower bound, provable in some but not all geometries

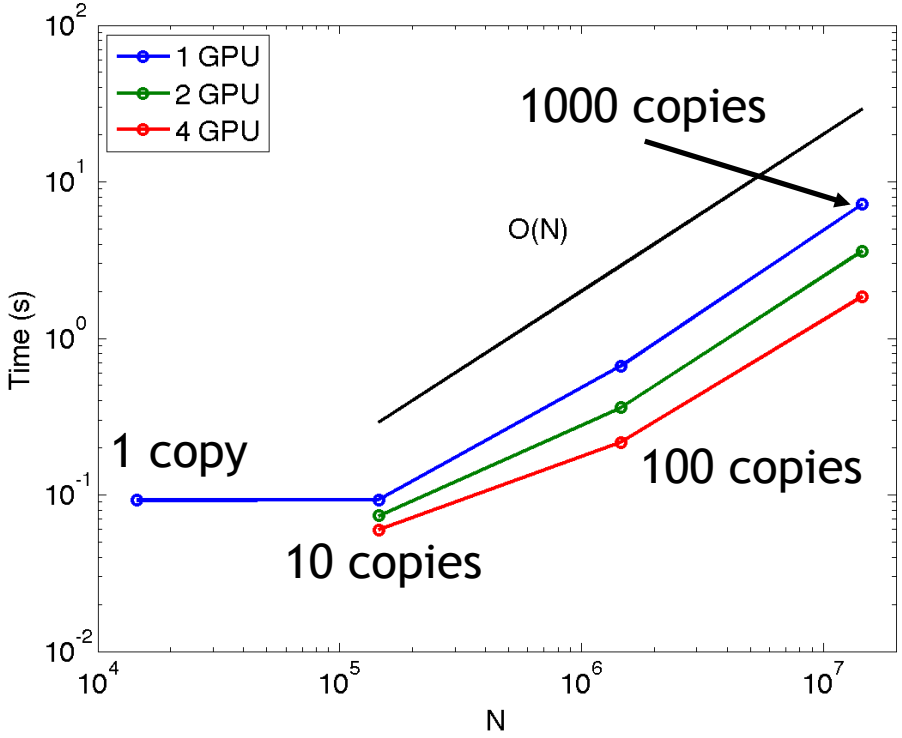
Bardhan, Knepley, Anitescu (2009)

Synthesis: GPU, PetFMM, and BIBEE

Lysozyme: ~2K atom charges, ~15K surface charges



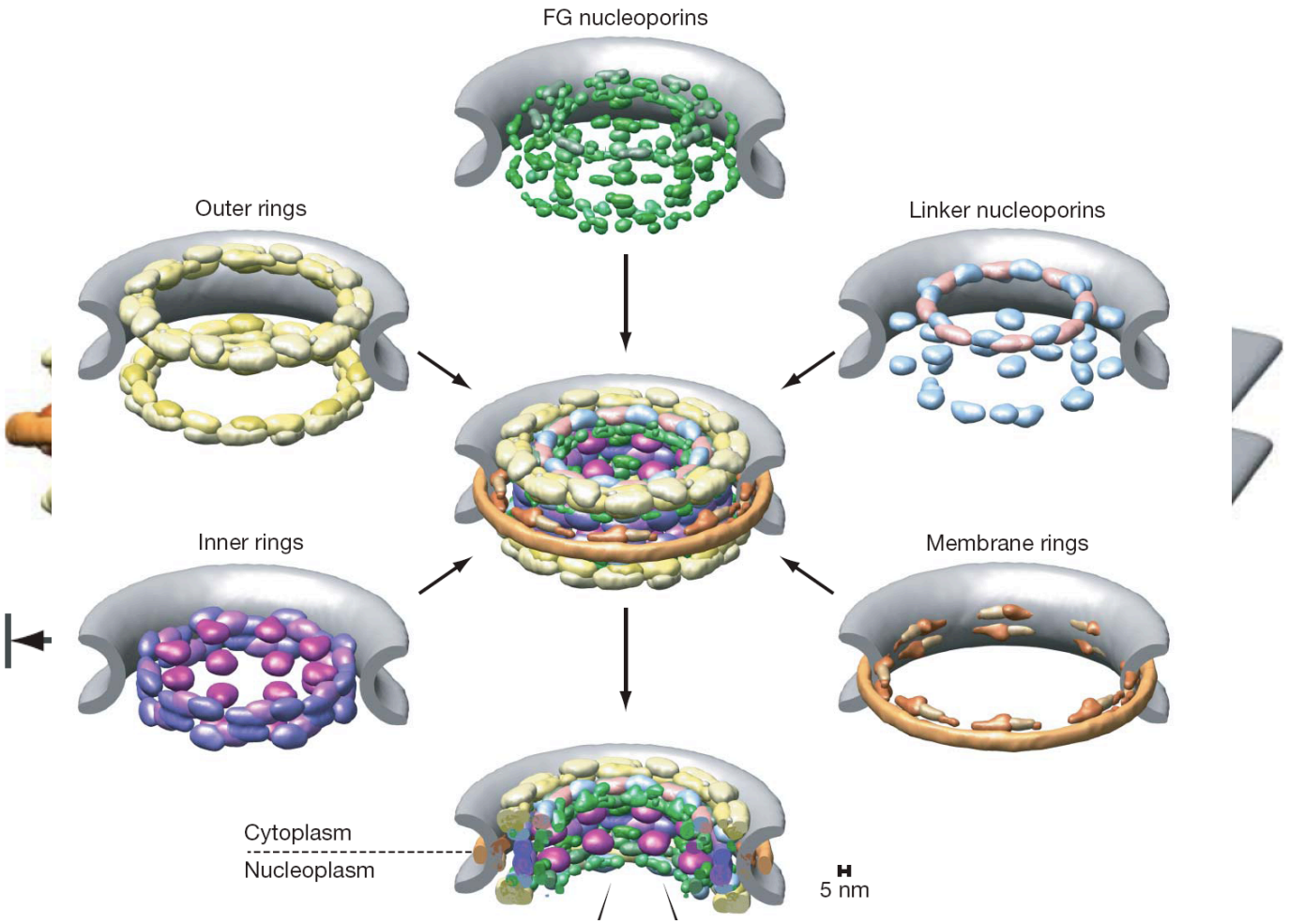
1000 lysozyme molecules: model of a concentrated protein solution



- 10X-20X faster than full BEM simulation
- Real continuum theory at competitive speed

Vision Statement

- Some day, we *will* design and build molecular systems this sophisticated.

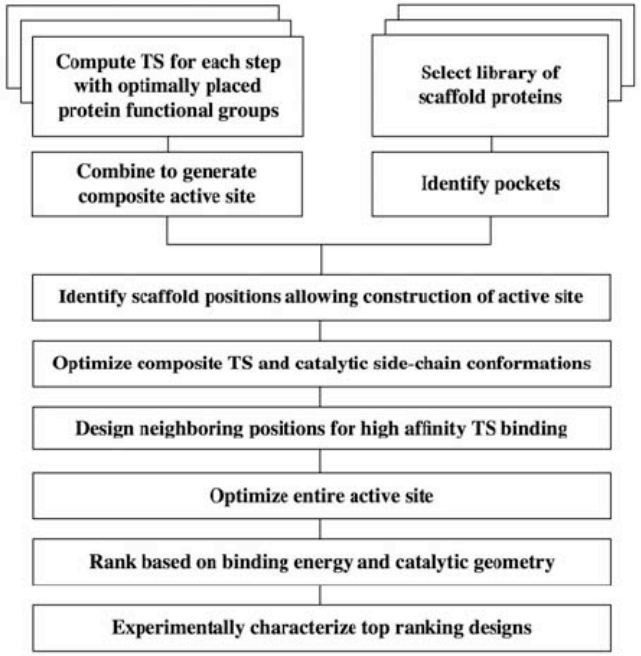


Alberts *et al*. Mol. Biol. of the Cell; Alber *et al.*, 2007

Enabling Nanotechnology CAD through Computational Biophysics

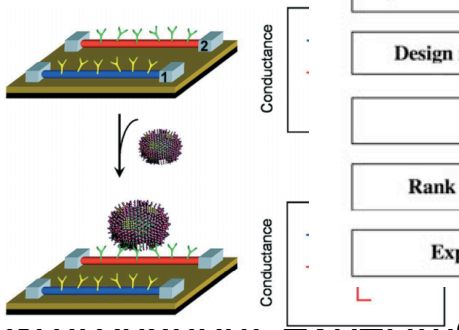
- Biologically-focused CAD has immediate applications
  1. Helping refine our understanding of biological systems
  2. Protein design and engineering in biotechnology
  3. Computational drug design efforts

Jiang, Baker et al. ('08)



- Many molecular tecl
  biological systems a
  1. Medical nanotechr
  2. Nanotoxicology (b

Allen e

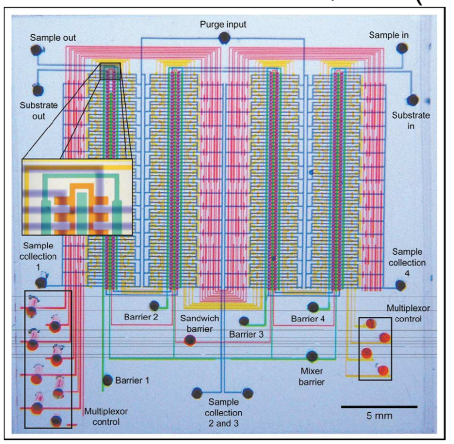


- Biology
  model
  1. Mut
  2. High-throughput experimental methods

al

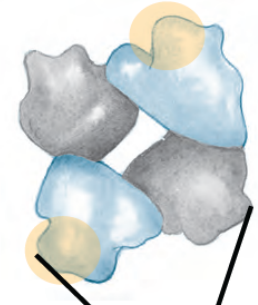
on

Hansen and Quake ('03)

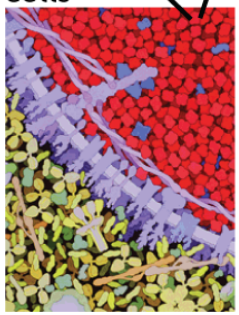


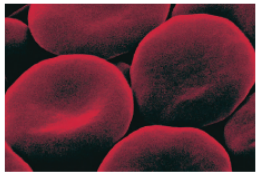
Complex Global Challenges

"Normal" Hemoglobin



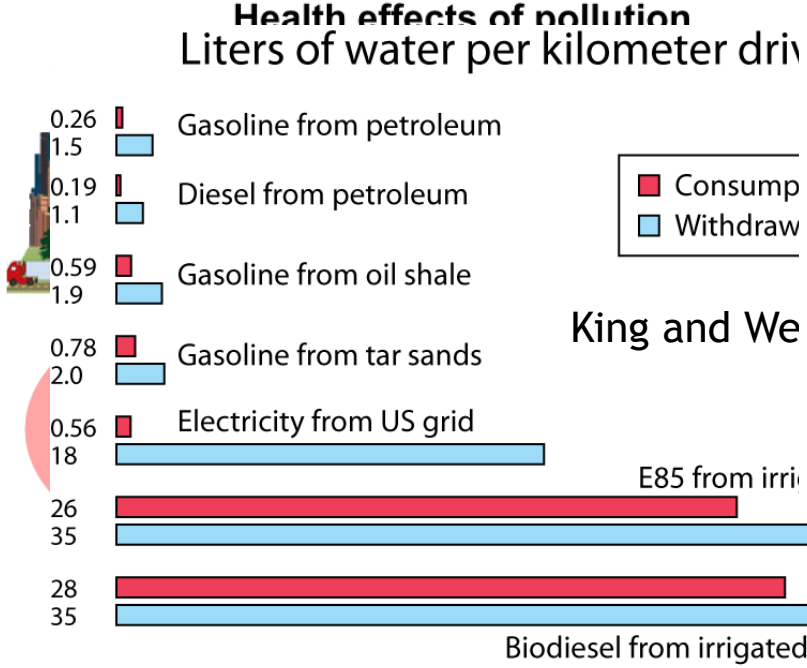
Highly concentrated in red blood cells





"Normal" red blood cells

Lehninger



Closing:

- Biology and biophysics are really cool, and the modeling problems are extremely demanding

- Boundary-integral equation approaches are sometimes very useful alternatives to PDEs

- One of the most important responsibilities that you have as future leaders in scientific computing: thinking at a high level about *why* you apply your talents to a given problem.