

## Introduction to Simulation - Lecture 26

# **Numerical Techniques in Action: Biomolecule Electrostatics for Drug Design and Electrical Analysis of Semiconductor Substrates**

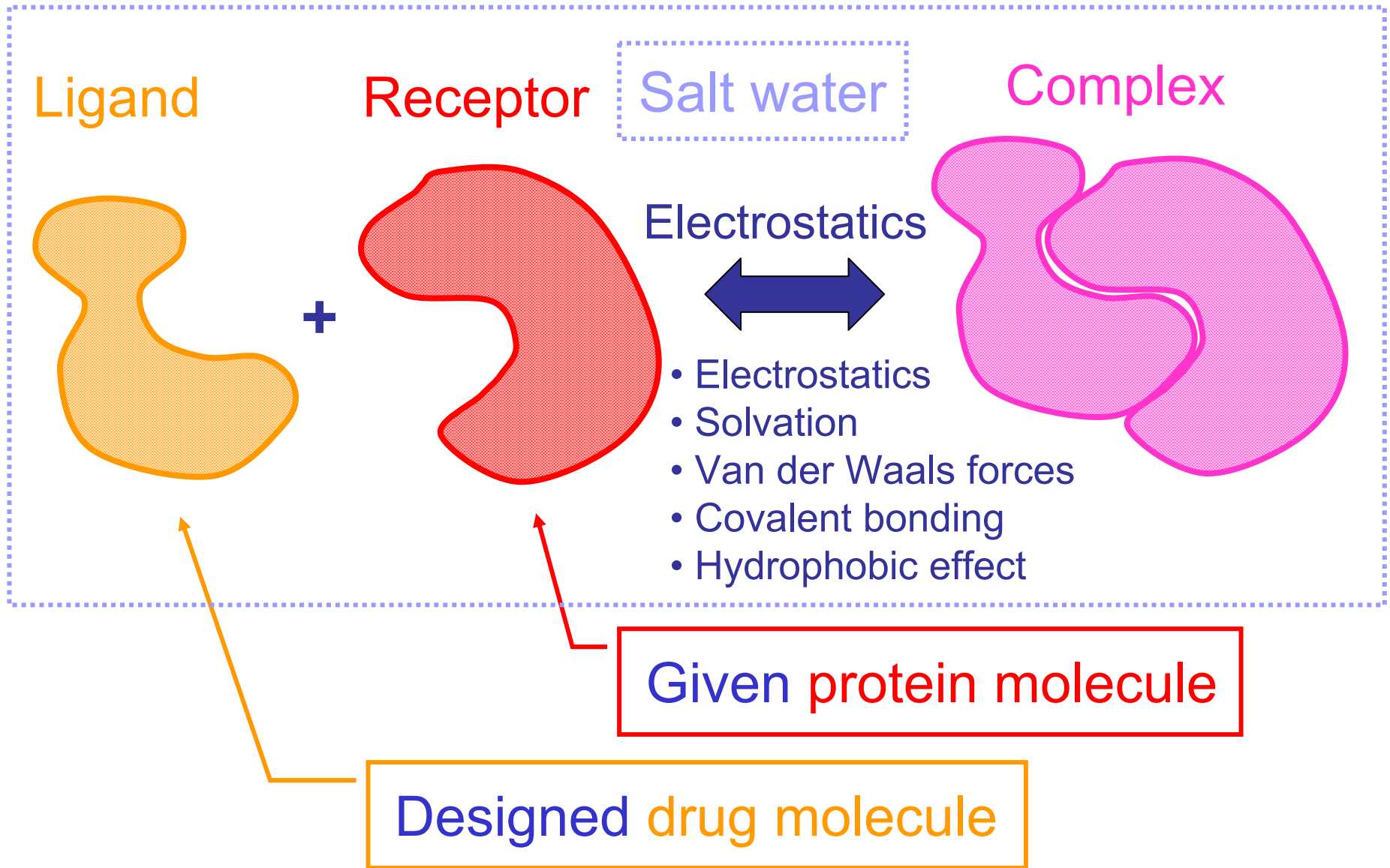
Jacob White

Thanks to Joe Kanapka, Joel Phillips, Shihhsien Kuo,  
Michael Altman, Jaydeep Bardhan, Bruce Tidor

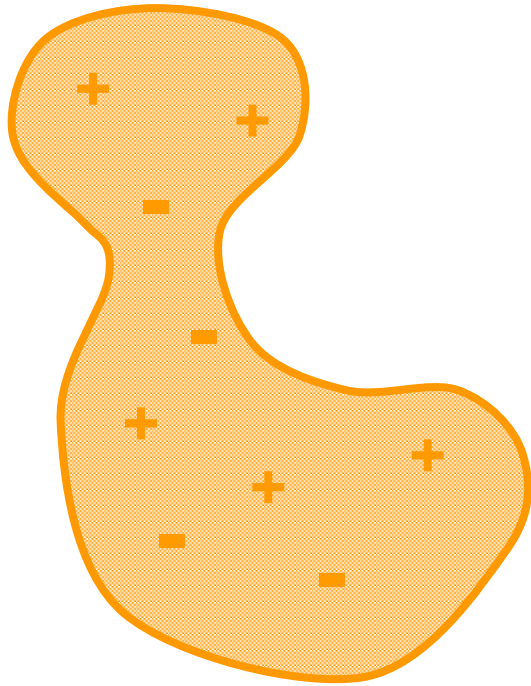
# Outline

- Problem statement
- Finite-difference approach and problems
- Integral equation method and advantages
- Fast solver implementation
- Computational results
- Conclusion and Future work

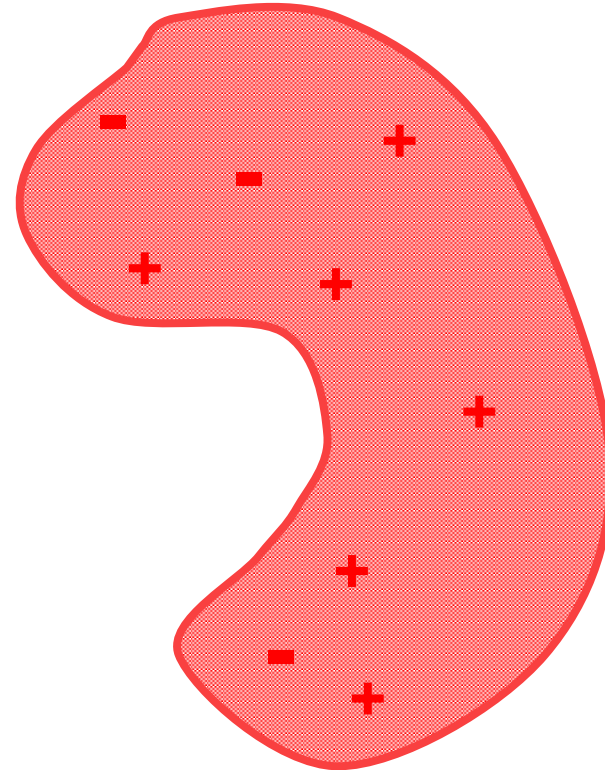
# The Problem of Drug Design



# Electrostatics View



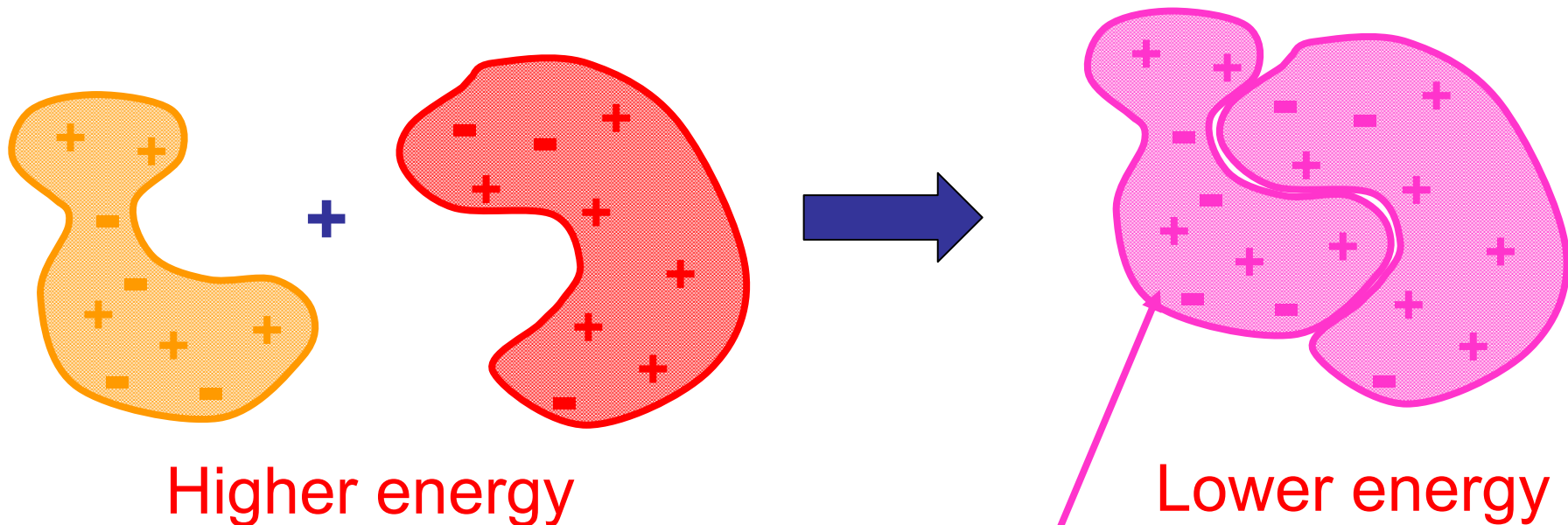
Ligand  
(drug molecule)



Receptor  
(protein molecule)

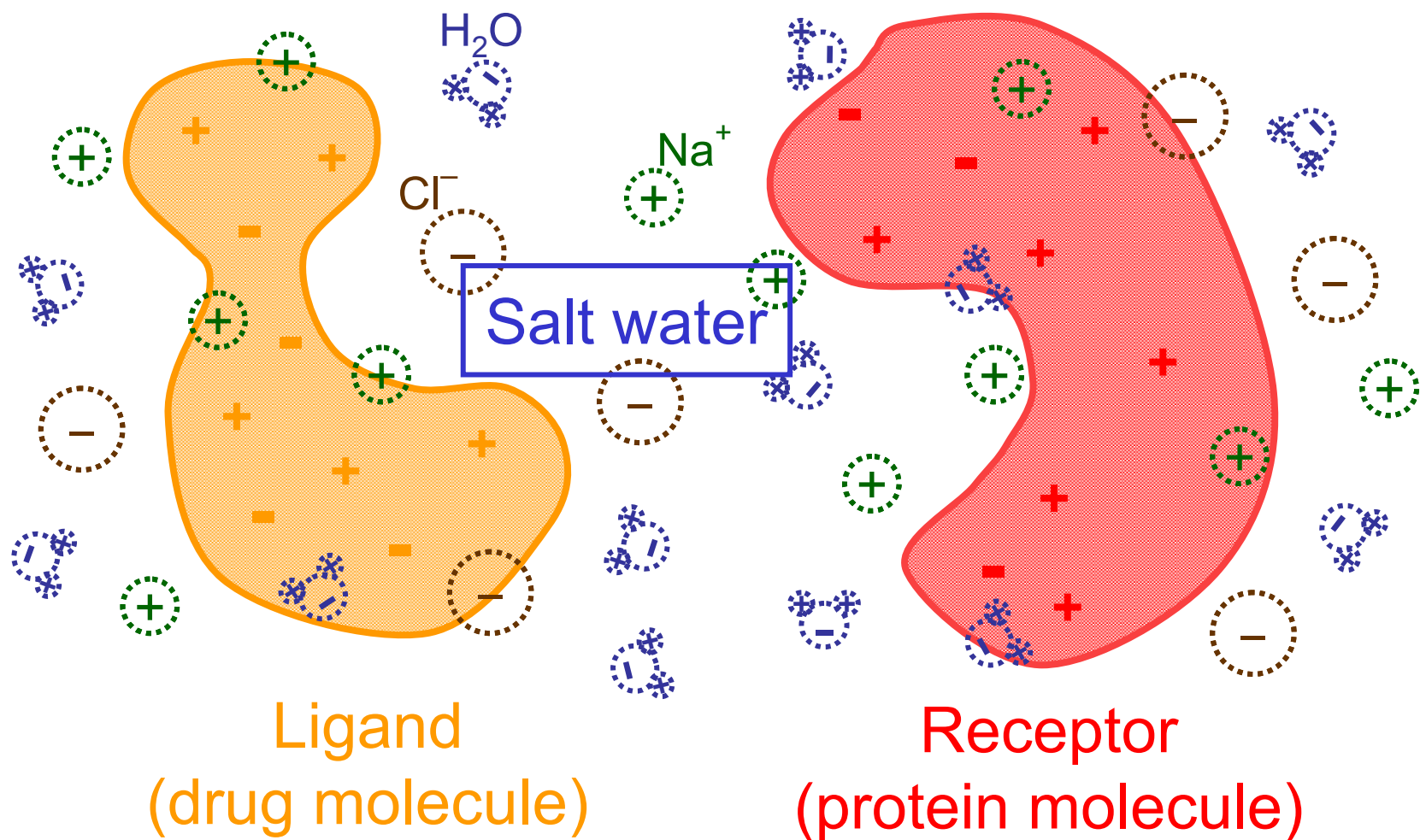
# Minimize Electrostatic Binding Energy

$$E_{binding} = |E_{desolvation}^{ligand}| + |E_{desolvation}^{receptor}| - |E_{interaction}|$$

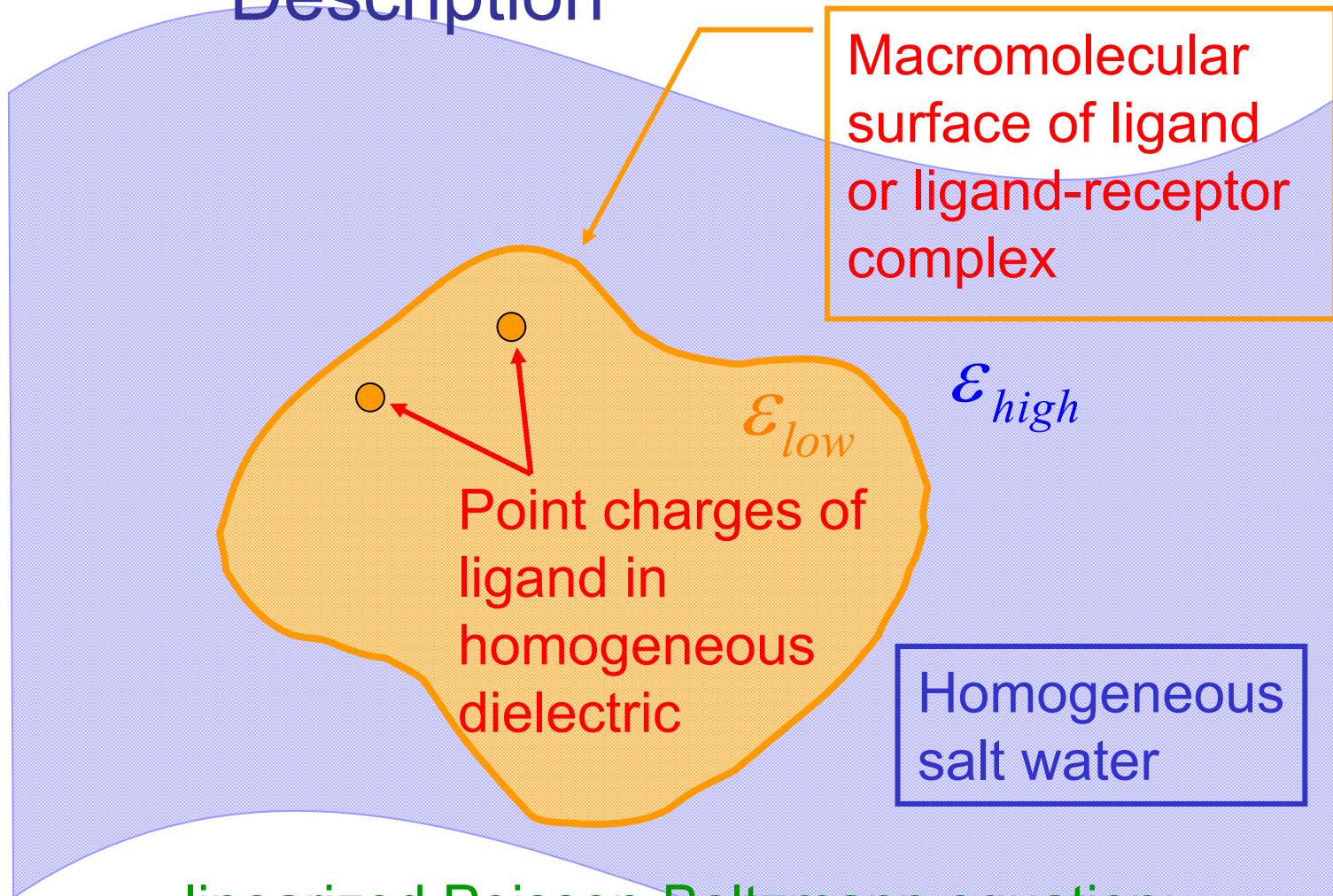


Determine the charge distribution in the ligand  
so that it is “Energetically Optimized” to bind

# An Electrostatic Analysis Problem



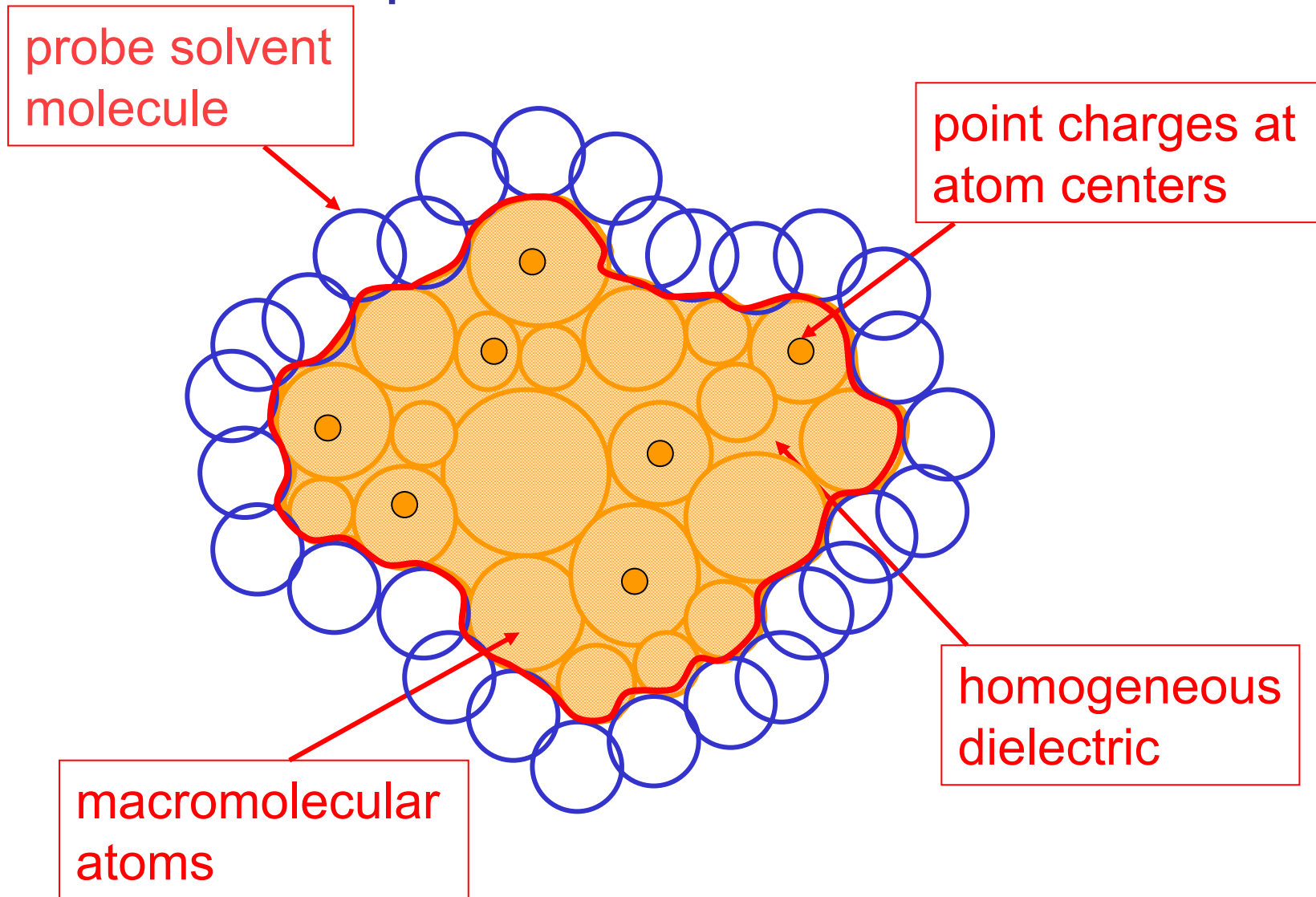
# A Simplified Physical Description



linearized Poisson-Boltzmann equation:

$$\nabla \cdot (\epsilon(\vec{r}) \nabla \phi(\vec{r})) - \epsilon(\vec{r}) \kappa^2(\vec{r}) \phi(\vec{r}) + \rho(\vec{r}) = 0$$

# Molecular Surface Representation





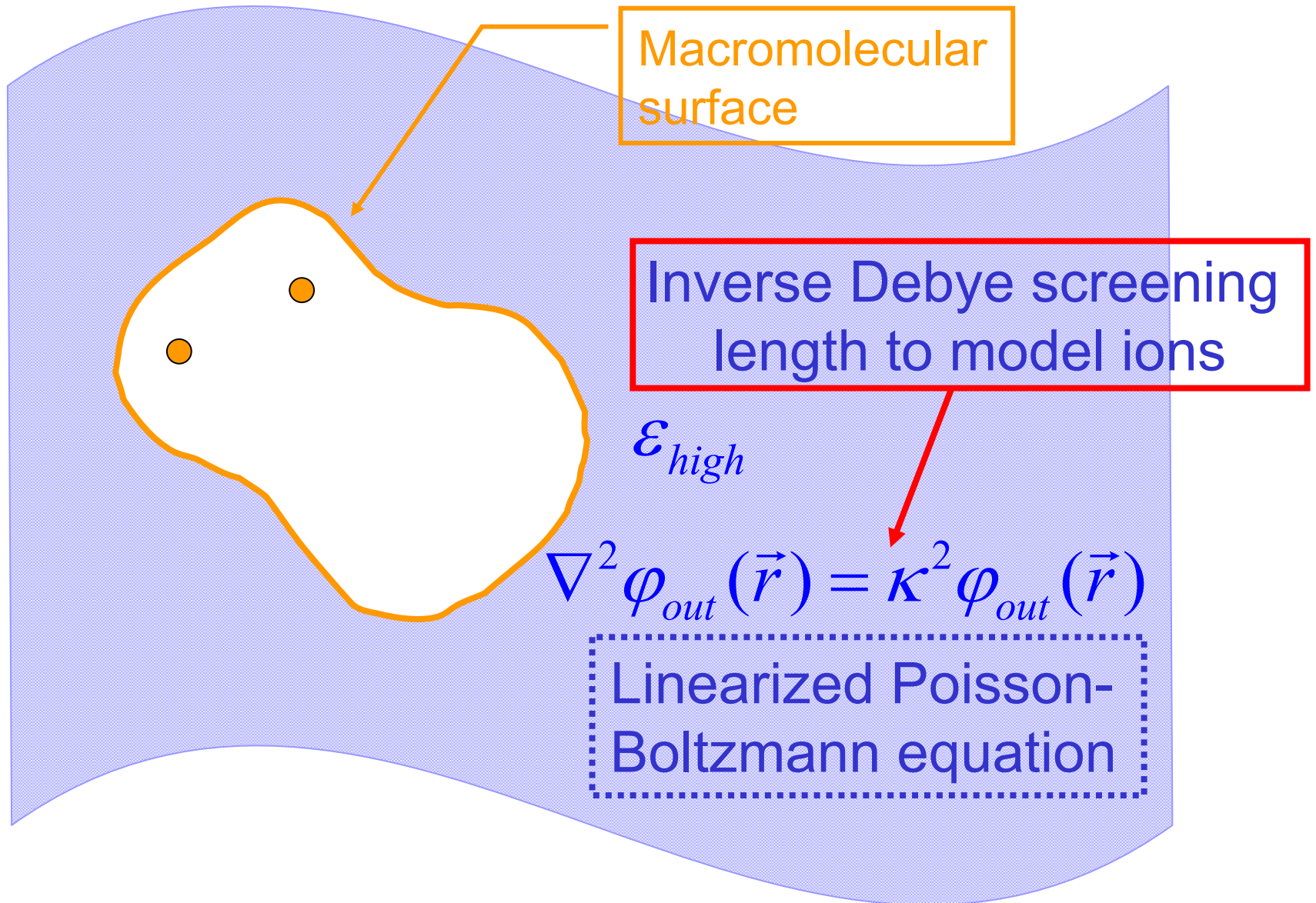
## Simplified Mathematical Model: Inside Macromolecule

Macromolecular  
surface

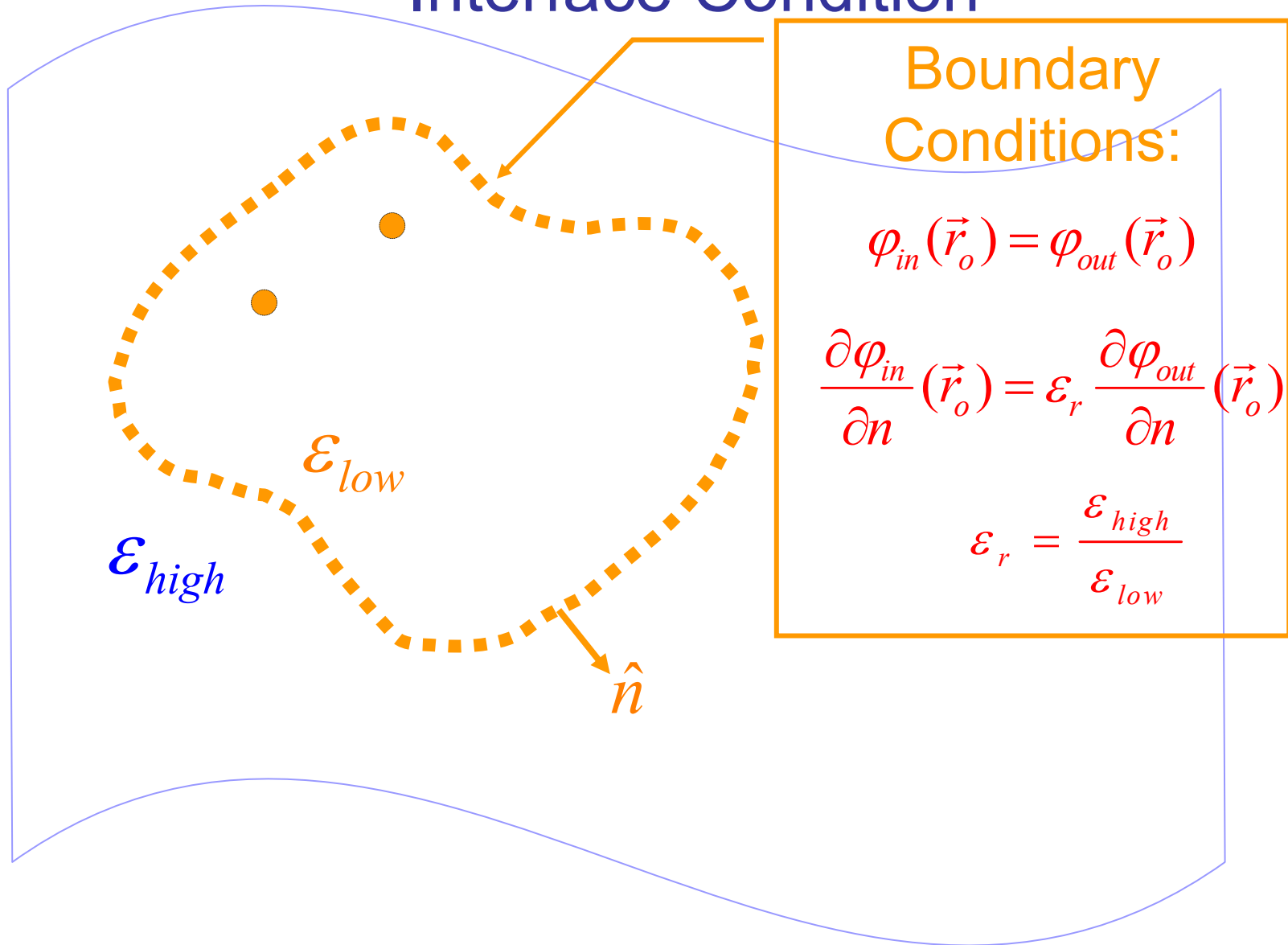
$$\nabla^2 \varphi_{in}(\vec{r}) = - \sum_{i=1}^N \frac{q_i}{\epsilon_{low}} \delta(\vec{r} - \vec{r}_i)$$

Poisson equation

# Simplified Mathematical Model: Salt Water Outside



## Interface Condition

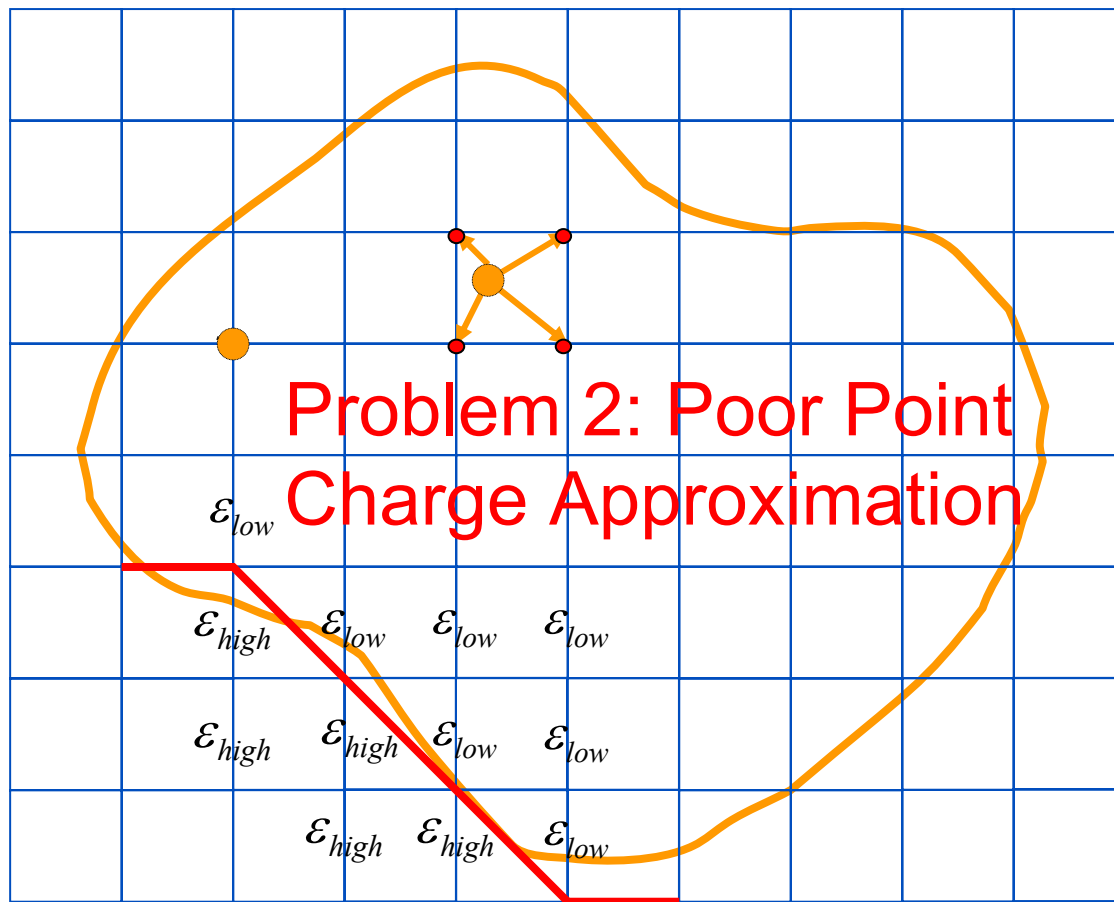


# Why Use this Simplified Model?

- Atomistic Level Simulation is too expensive
  - Salt ions and water molecules treated individually
- Continuum Model Matches Well with Experimental Data

# Standard Finite-Difference Method

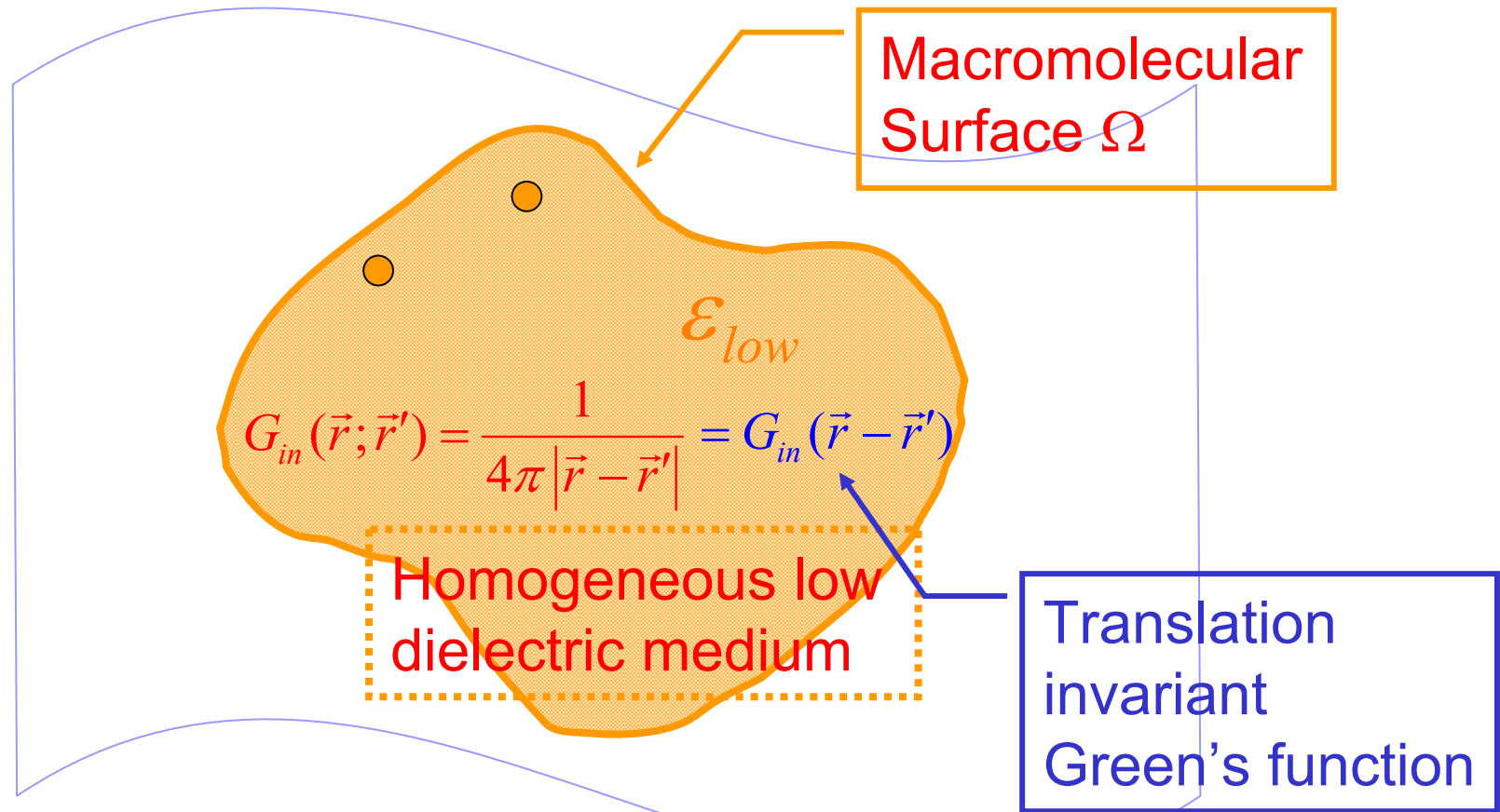
set up boundary conditions and solve for grid potentials



Problem 3:  
Inexact  
Boundary  
Conditions

Problem 1: Inaccurate Molecular Surface

# Integral equation: Interior Problem



$$\int_{\Omega} \left[ \varphi_{in}(\vec{r}') \frac{\partial G_{in}(\vec{r}; \vec{r}')}{\partial n} - G_{in}(\vec{r}; \vec{r}') \frac{\partial \varphi_{in}(\vec{r}')}{\partial n} \right] d\vec{r}' = \sum_{k=1}^N \frac{q_k}{\epsilon_{low}} G_{in}(\vec{r}; \vec{r}_k)$$

# Integral equation: Exterior Problem

Match boundary conditions

NOTE:  $\kappa$  is real, electrostatic not fullwave problem

Homogeneous high dielectric medium

$\epsilon_{high}$

$\hat{n}$

$G_{out}(\vec{r}; \vec{r}') = \frac{e^{-\kappa|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} = G_{out}(\vec{r} - \vec{r}')$

$$\int_{\Omega} \left[ \underbrace{-\varphi_{out}(\vec{r}')}_{\varphi_{in}(\vec{r}')} \frac{\partial G_{out}(\vec{r}; \vec{r}')}{\partial n} + G_{out}(\vec{r}; \vec{r}') \underbrace{\frac{\partial \varphi_{out}(\vec{r}')}{\partial n}}_{\frac{1}{\epsilon_r} \frac{\partial \varphi_{in}(\vec{r}')}{\partial n}} \right] d\vec{r}' = 0$$

# Advantages For Integral Equation Formulation

- Directly discretize surfaces
- Point charges treated exactly
- Handles infinite exterior

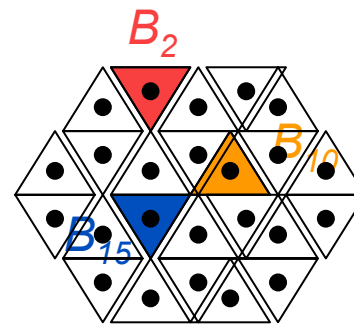
$$\int_{\Omega} \left[ \varphi_{in}(\vec{r}') \frac{\partial G_{in}}{\partial n}(\vec{r}; \vec{r}') - G_{in}(\vec{r}; \vec{r}') \frac{\partial \varphi_{in}}{\partial n}(\vec{r}') \right] d\vec{r}' = \sum_{k=1}^N \frac{q_k}{\epsilon_{low}} G_{in}(\vec{r}; \vec{r}_k)$$

$$\int_{\Omega} \left[ -\varphi_{in}(\vec{r}') \frac{\partial G_{out}}{\partial n}(\vec{r}; \vec{r}') + G_{out}(\vec{r}; \vec{r}') \frac{1}{\epsilon_r} \frac{\partial \varphi_{in}}{\partial n}(\vec{r}') \right] d\vec{r}' = 0$$



# Standard piecewise constant collocation discretization method

$$\varphi_{in}(\vec{r}) \approx \sum_j a_j B_j(\vec{r})$$
$$\frac{\partial \varphi_{in}}{\partial n}(\vec{r}) \approx \sum_j b_j B_j(\vec{r})$$
$$\vec{r} \in \Omega$$



- Piecewise constant basis functions
- Collocation points at panel centroids

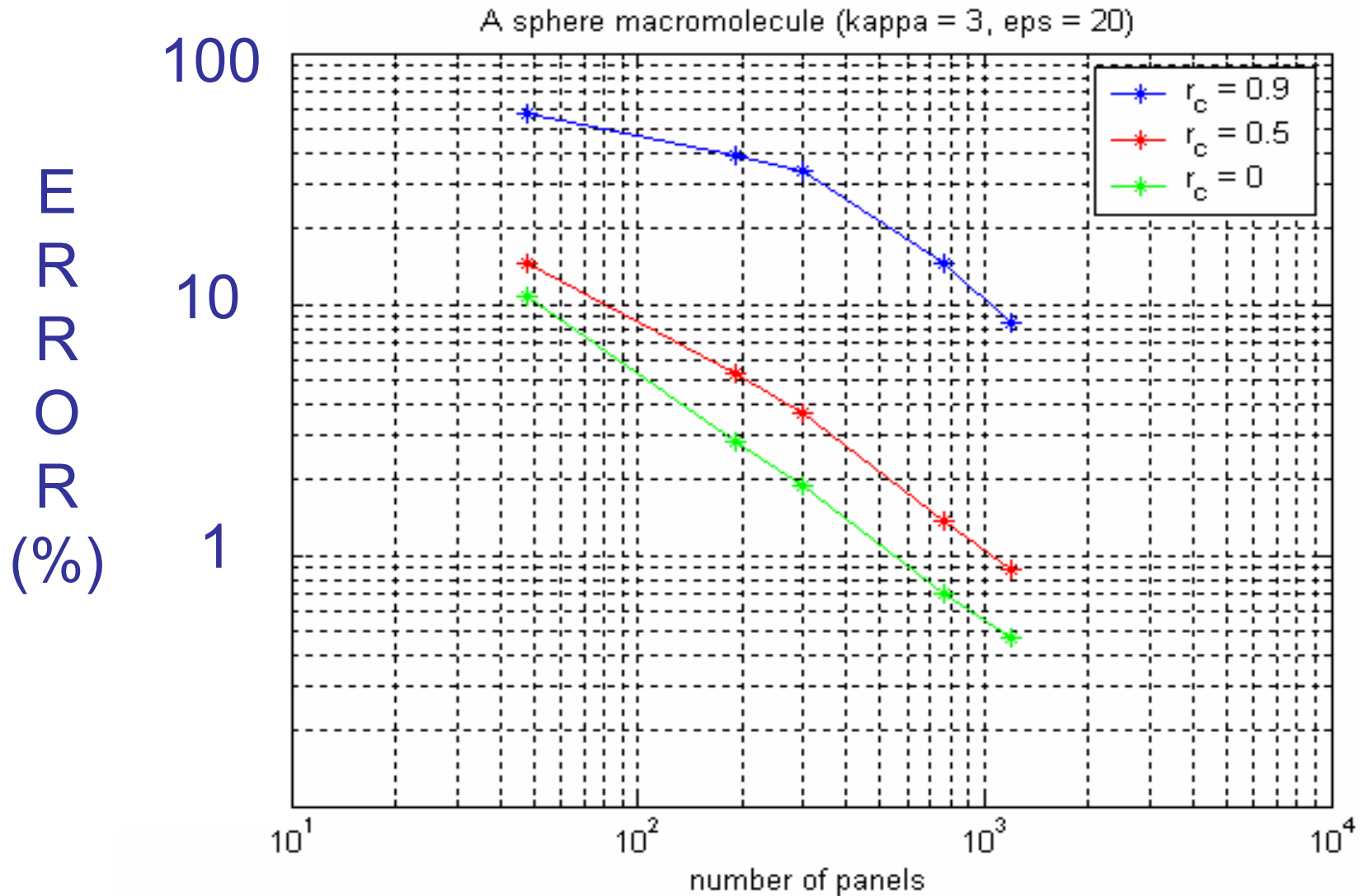
# Matrix Equation

$$\begin{bmatrix} D^{in} & S^{in} \\ D^{out} & S^{out} \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^N \frac{q_k}{4\pi |\vec{r}_i - \vec{r}_k|} \\ 0 \end{bmatrix}$$

$$D_{ij}^{in} = \int_{panel_j} \frac{\partial}{\partial n'} \left( \frac{1}{4\pi |\vec{r}_i - \vec{r}'|} \right) d\vec{r}' \quad S_{ij}^{in} = - \int_{panel_j} \frac{1}{4\pi |\vec{r}_i - \vec{r}'|} d\vec{r}'$$

$$D_{ij}^{out} = - \int_{panel_j} \frac{\partial}{\partial n'} \left( \frac{e^{-\kappa |\vec{r}_i - \vec{r}'|}}{4\pi |\vec{r}_i - \vec{r}'|} \right) d\vec{r}' \quad S_{ij}^{out} = \frac{1}{\epsilon_r} \int_{panel_j} \frac{e^{-\kappa |\vec{r}_i - \vec{r}'|}}{4\pi |\vec{r}_i - \vec{r}'|} d\vec{r}'$$

# A sphere molecule: comparison with analytical result



# Iterative solver

$$\varphi(\vec{r}) \equiv \int K(\vec{r}; \vec{r}') \sigma(\vec{r}') d\vec{r}'$$

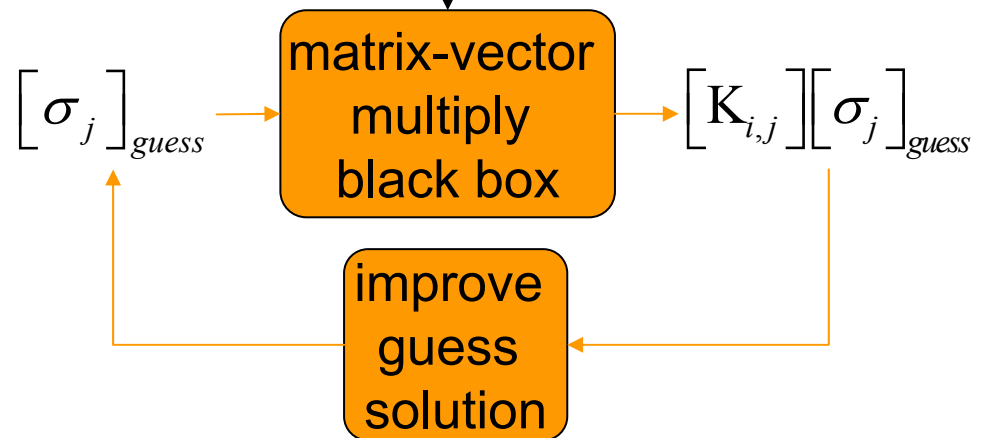
Discretization

Gaussian Elimination  $\left[ \mathbf{K}_{i,j} \right] \left[ \sigma_j \right] = \left[ \varphi_i \right]$  Iterative Solver

$$\left[ \sigma_j \right] = \left[ \mathbf{K}_{i,j} \right]^{-1} \left[ \varphi_i \right]$$

```
for i = 1:n-1
  for j = i+1:n
     $K_{j,i} = K_{j,i} / K_{i,i}$ 
    for k = i+1:n
       $K_{j,k} = K_{j,k} - K_{j,i} / K_{i,k}$ 
    end
  end
end
end
```

$O(N^3)$



$O(N^2)$  per iteration

# Use Fast Integral Equation Solver

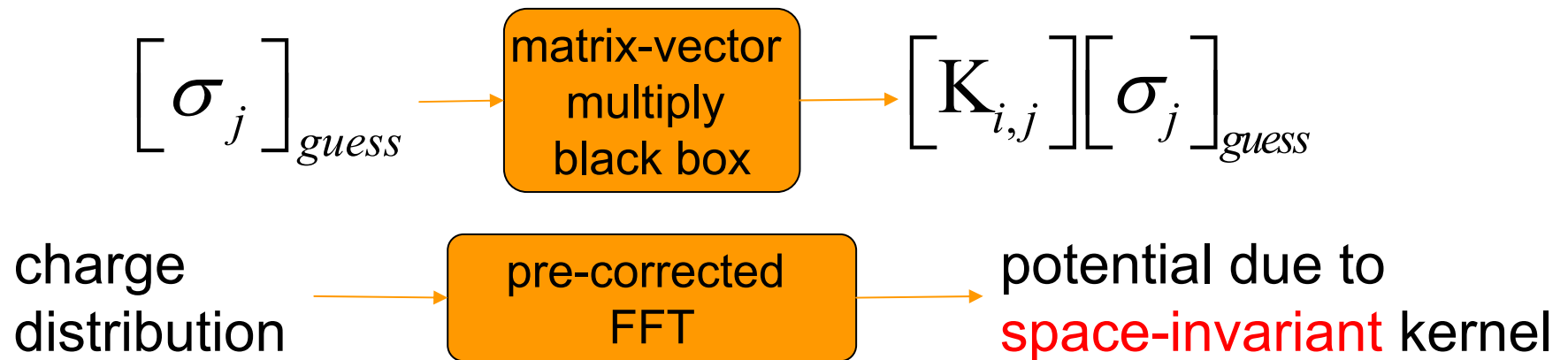


$O(N \log N)$  Matrix-vector multiply

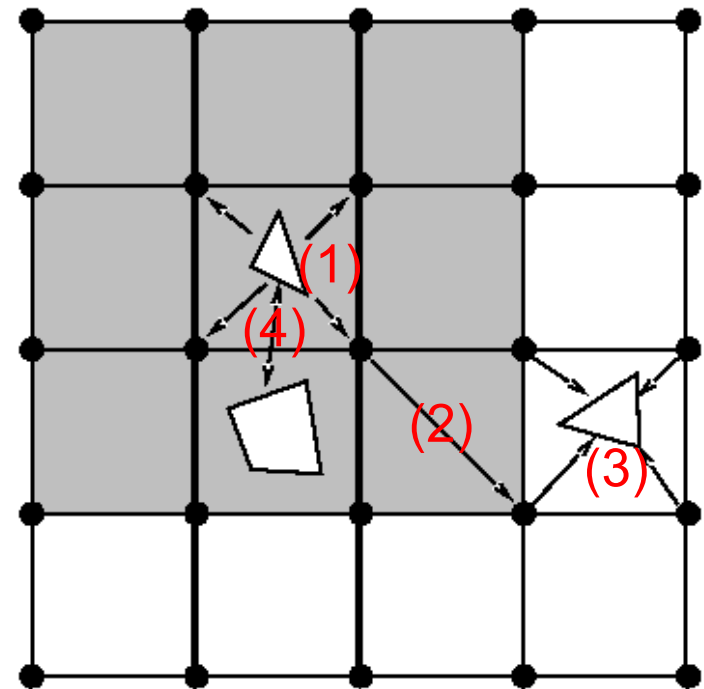
- Multiple Green's functions
- Translation Invariant kernel

$$G_{in}(\vec{r}; \vec{r}') = \frac{1}{4\pi |\vec{r} - \vec{r}'|}$$
$$G_{out}(\vec{r}; \vec{r}') = \frac{e^{-\kappa |\vec{r} - \vec{r}'|}}{4\pi |\vec{r} - \vec{r}'|}$$

# Pre-corrected FFT algorithm



- 1) projection of panel charges onto grid charges
- 2) grid potentials due to grid charges are computed by FFT
- 3) potentials on panel centroids are interpolated from grid potentials
- 4) direct interaction and correction among near neighbors



Picture courtesy of J. Phillips

# Preconditioner on Two Examples

Need to find a  
good preconditioner

$$[P] \approx [K_{i,j}]$$

And solve

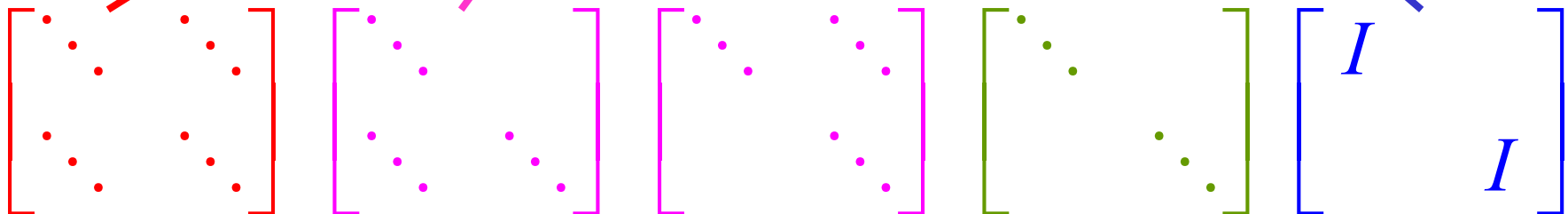
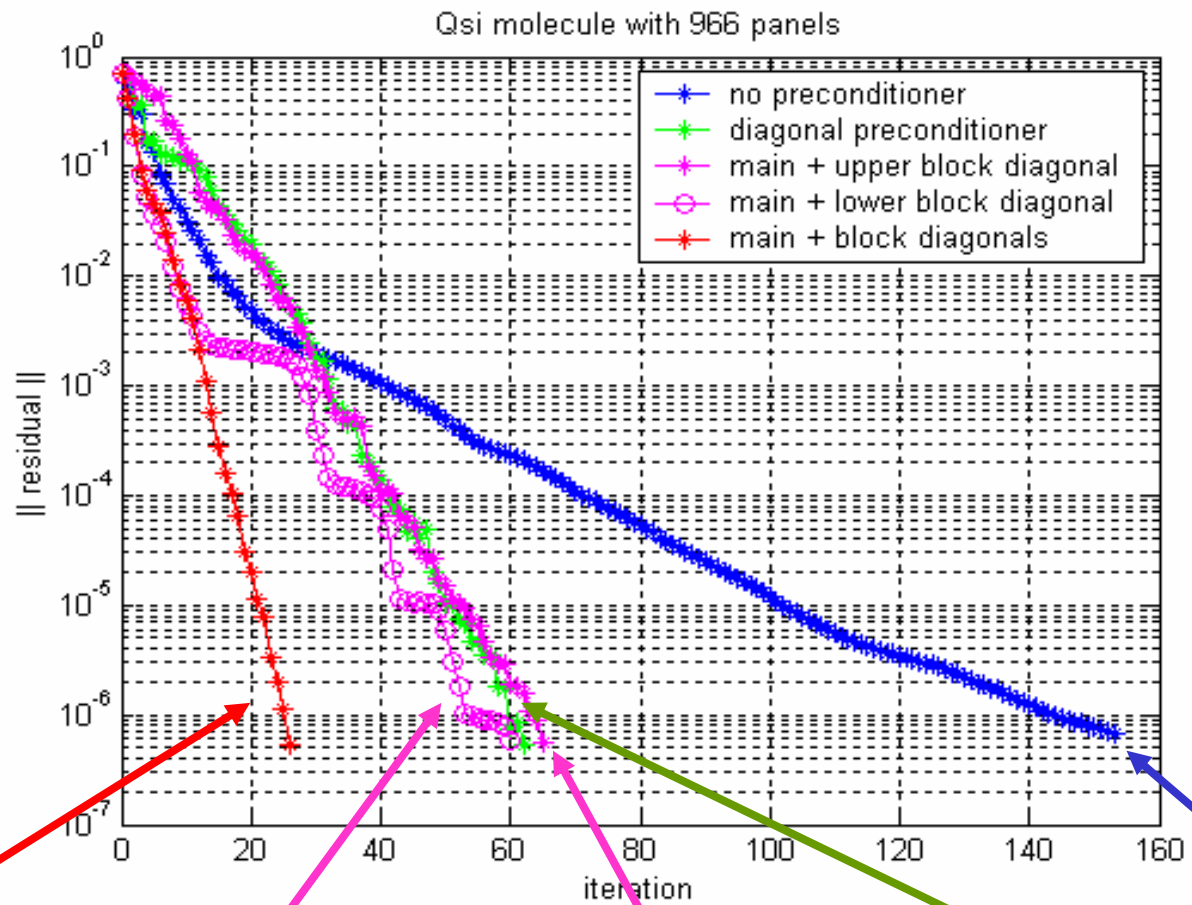
$$[P]^{-1} [K_{i,j}] [\sigma_j] = [P]^{-1} [\varphi_i]$$

hopefully better conditioned than  $[K_{i,j}]$

Qsi molecule

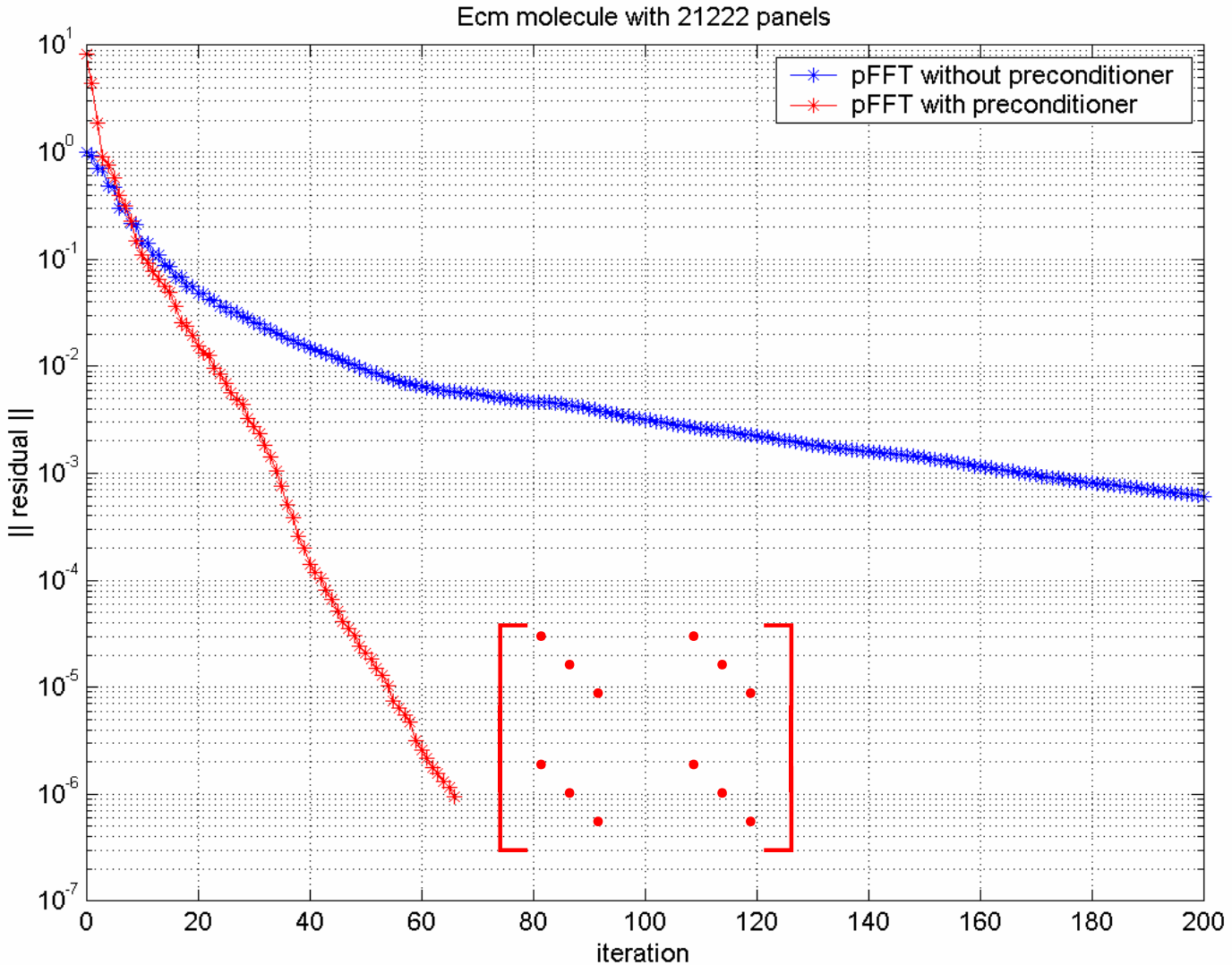
Ecm protein

# Preconditioner result: Qsi molecule





# Preconditioner result: Ecm protein



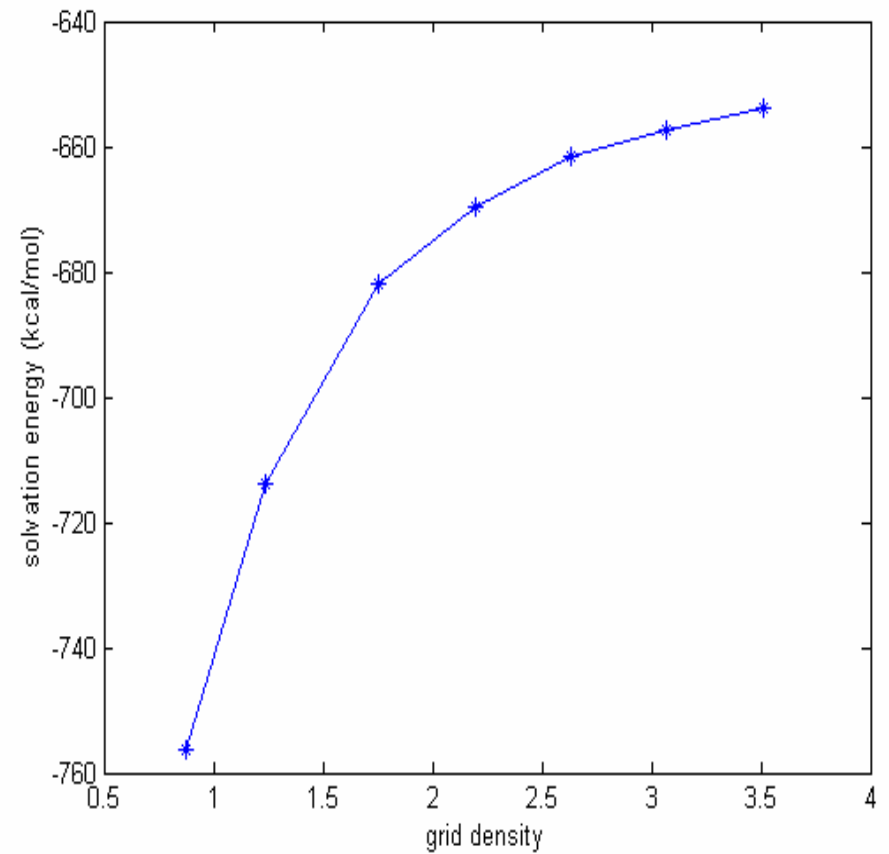
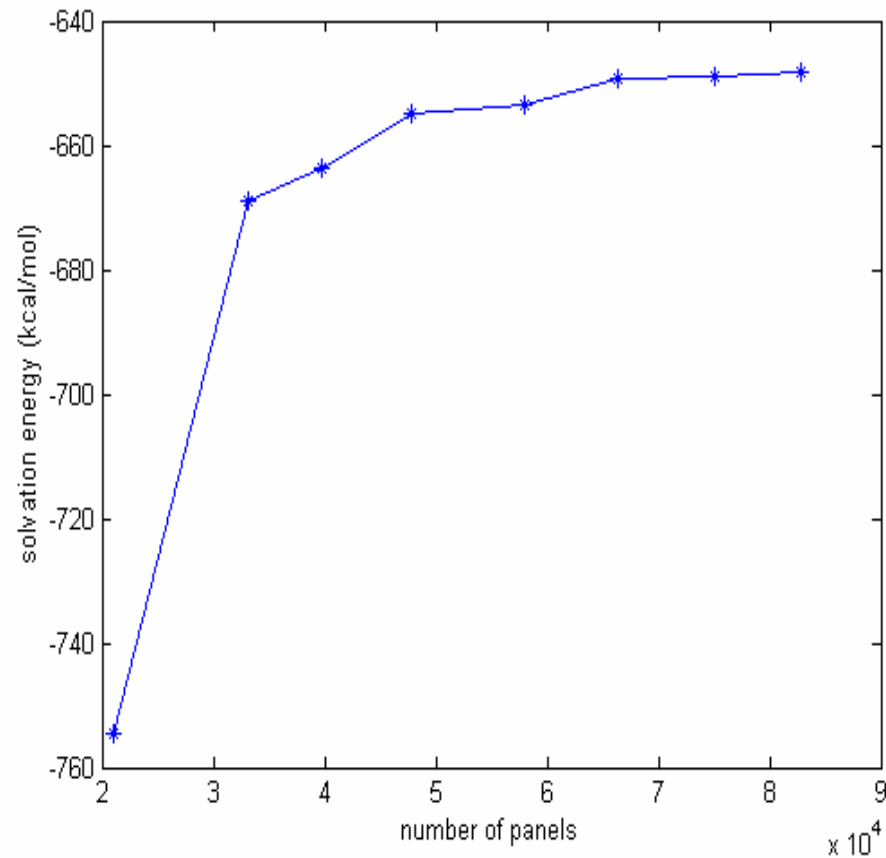
# Accuracy comparison with DelPhi

			<b><math>E_{\text{solvation}}</math> (kcal/mol)</b>	
	# of dielectric panels	# of salt panels	pFFT	DelPhi
<b>Water</b>	17204	9330	<b>-3.14</b>	<b>-3.17</b>
<b>TSA</b>	34114	5842	<b>-34.62</b>	<b>-34.75</b>
<b>ECM</b>	82868	18596	<b>-646.42</b>	<b>-653.88</b>

# Convergence Results of Ecm Protein

pFFT

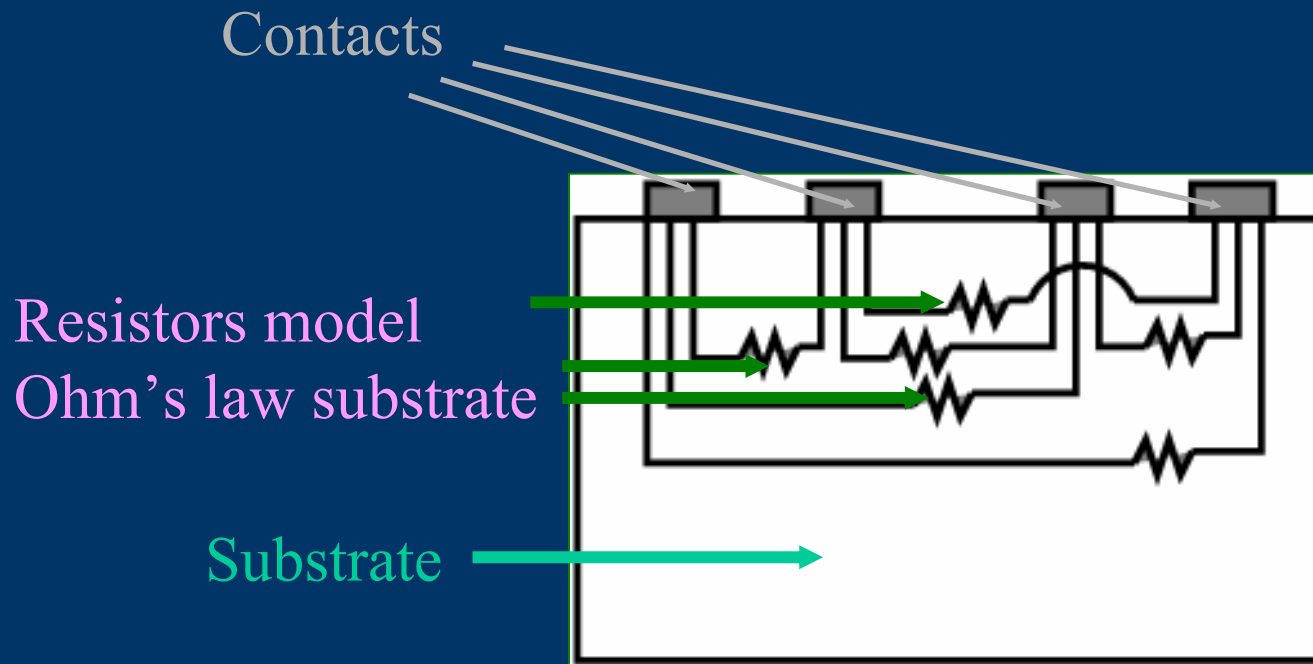
DelPhi



## Binding energy calculation of a protein-peptide complex

	Energy calculated (kcal/mol)			
	$R_{\text{desolvation}}$	$L_{\text{desolvation}}$	$(R \rightarrow L)_{\text{interaction}}$	$(L \rightarrow R)_{\text{interaction}}$
<b>pFFT</b>	14.52	24.47	130.80	130.91
<b>DelPhi</b>	14.51	24.47	131.03	131.03

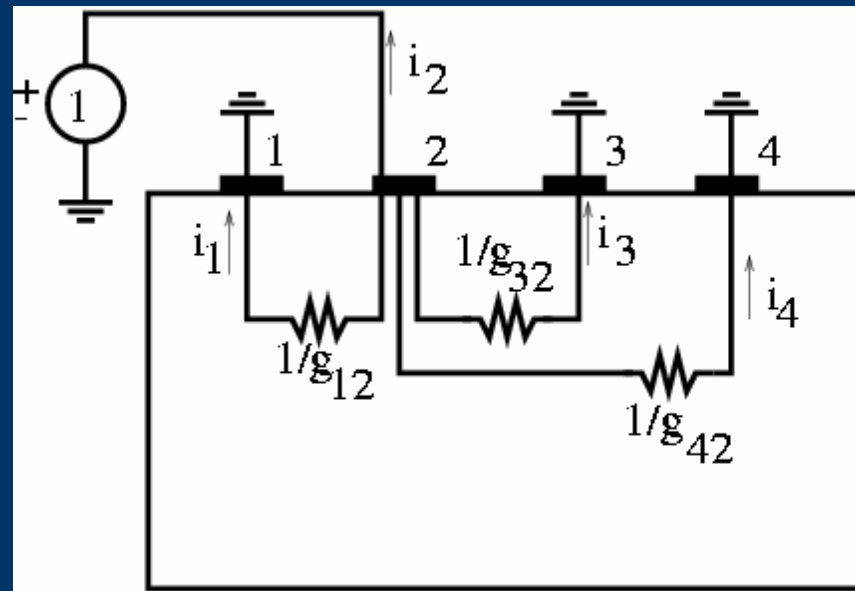
# Substrate coupling problem



# The computational problem

- Real problem for designers:
  - Block isolation difficult in analog designs
  - Accurate simulation needed: calculate the conductance matrix numerically
- Key issues
  - Large number of contacts
  - Voltage at one contact drives current in *all* the contacts
- Want conductance matrix  $G$  so that  $Gv = i$  (voltage vector  $v$ , current vector  $i$ )
  - Hard to obtain: *unlike  $1/r$  or other known-kernel potential calculations, entries of  $G$  unknown a priori*
  - Hard to use for circuit simulation

# Circuit View



- Circuit view of conductances  $1 = i_j / g_{j2}$
- Conductances=currents  $g_{j2} = i_j$

# Matrix View

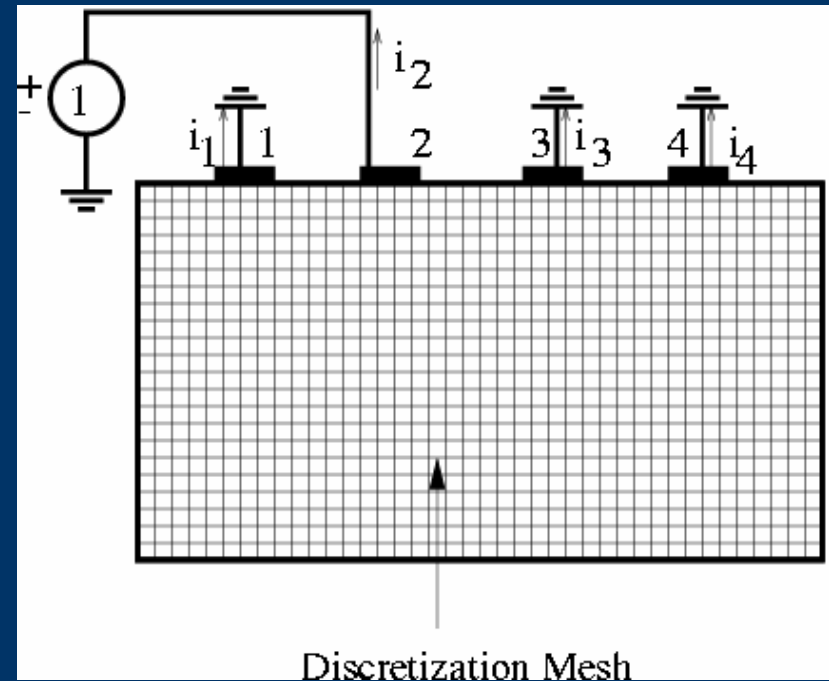
$$\begin{pmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} g_{12} \\ g_{22} \\ g_{32} \\ g_{42} \end{pmatrix} = \begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{pmatrix}$$

- Matrix  $G$  in *standard basis*
  - $i$ th voltage (input) vector component = voltage on contact  $i$
  - $i$ th current (output) vector component = current out of contact  $i$



# Multiple solves get $G$

- 1 column of  $G = 1$  solve for currents given voltages
- $n$  solves for  $n$  contacts
- Our solver
  - Finite-difference formulation (not essential)
  - Iterative solver (preconditioned conjugate gradient)



# Sparsification

- $G$  is dense: 10000 contacts =
  - 100 million resistor model: hard for circuit simulator
  - 10000 solves each with millions of unknowns
- Does  $G$  have a sparse representation? Two benefits if it does:
  - Better circuit simulator performance
  - Faster extraction of  $G$  by reducing number of solves
- When is the coupling dense for practical purposes?
  - Always dense in terms of nonzeros
  - But can be *numerically* sparse: entries drop off very quickly
  - Goal: find a new representation where  $G$  is numerically sparse

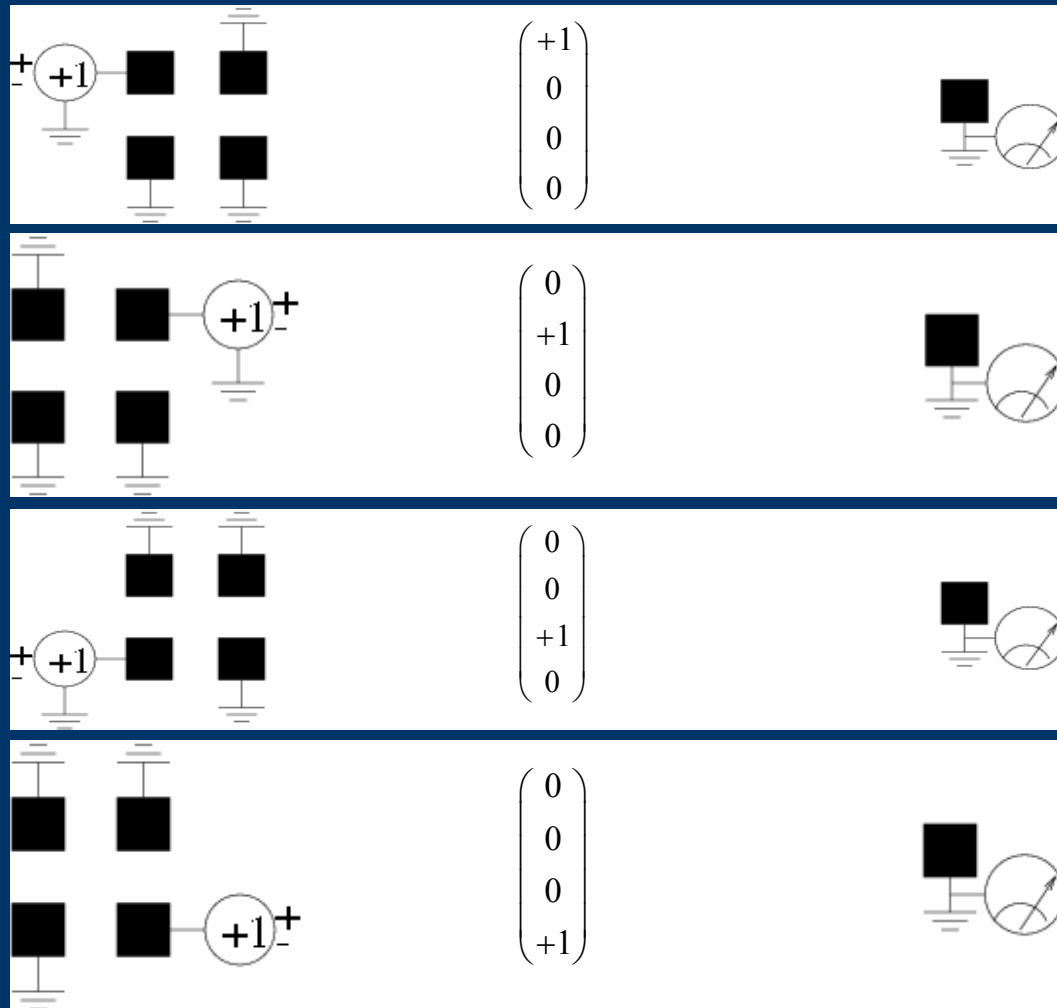
# How to sparsify?

- Two choices:
  - Threshold  $G$ 
    - Zero out entries  $<$  threshold  $t$
    - Fine for fast current dropoff
    - Serious accuracy loss for slow current dropoff
  - Change of basis
    - Get conductance matrix *in new basis*
    - *Fast current decay* in new basis: thresholding works well

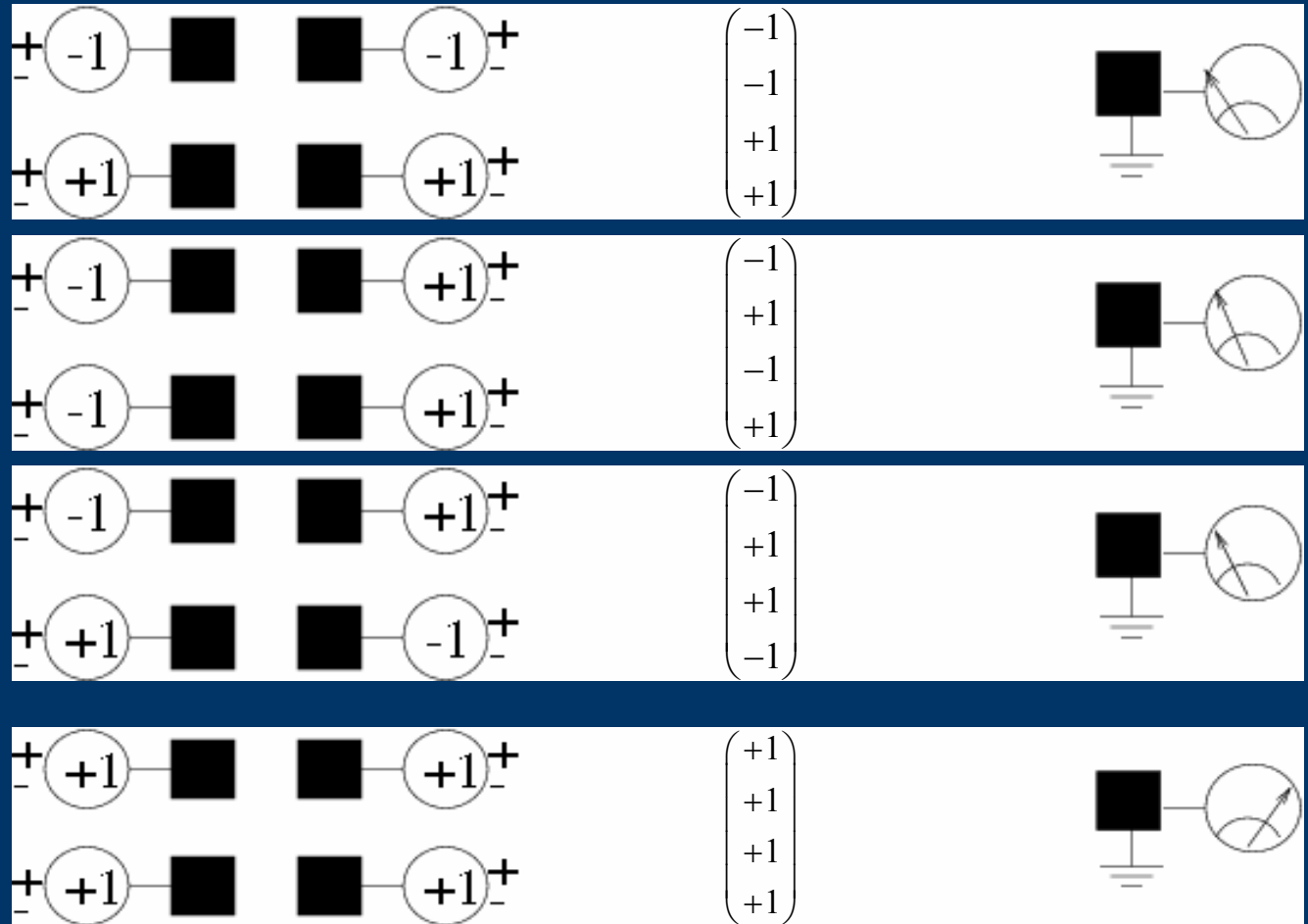
# The algorithm: motivation

- Currents due to standard basis functions (1 volt on one contact, 0 on all others) may decay slowly
- But current responses for two nearby contacts look similar
- Try *balanced* voltages for nearby contacts
  - make average voltage 0 for new basis functions

# Standard basis: faraway currents



# Transformed basis: faraway currents

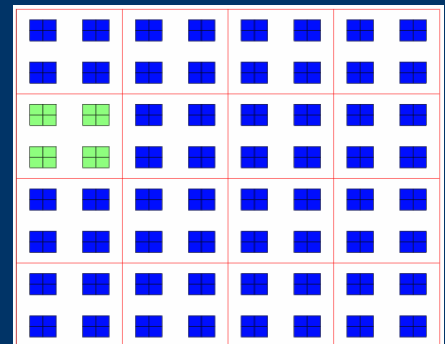
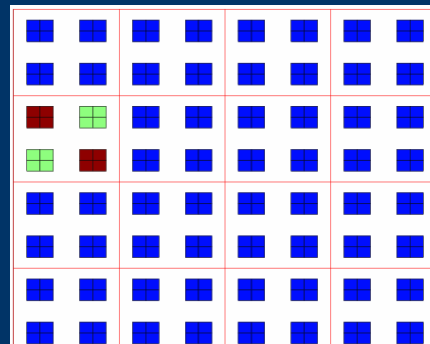
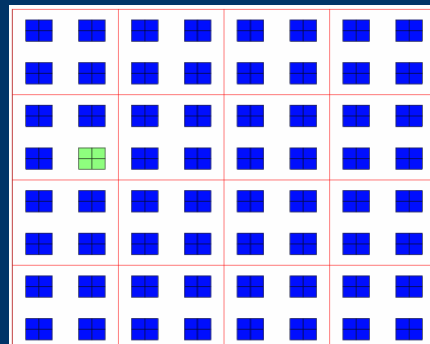
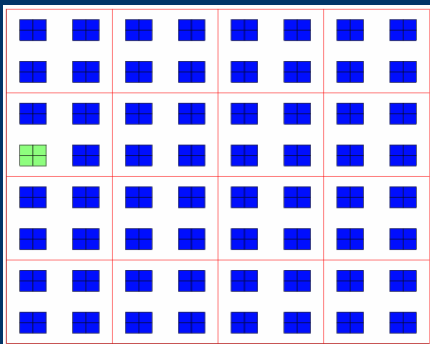
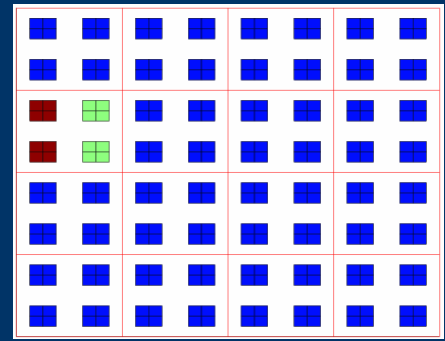
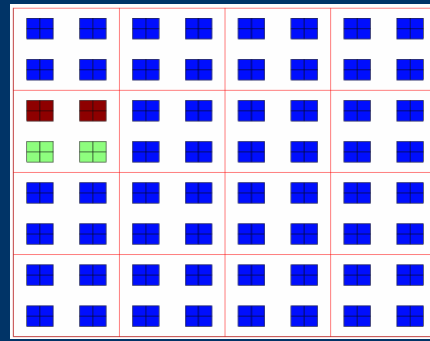
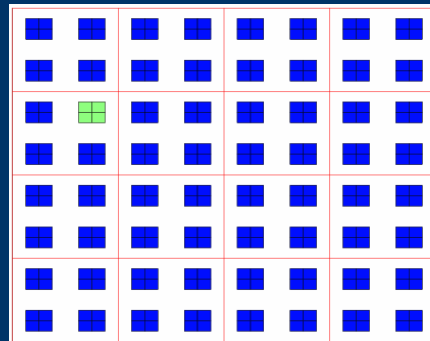
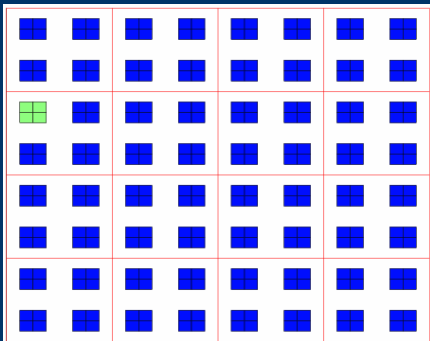


# Multilevel method: bottom level

Standard basis functions

Transformed basis functions

Voltage: +1 -1 0

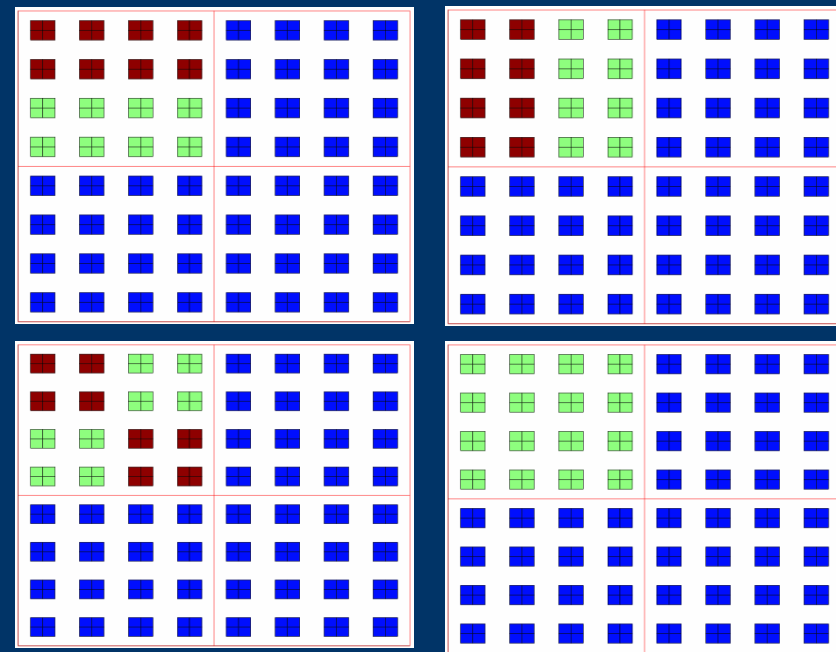
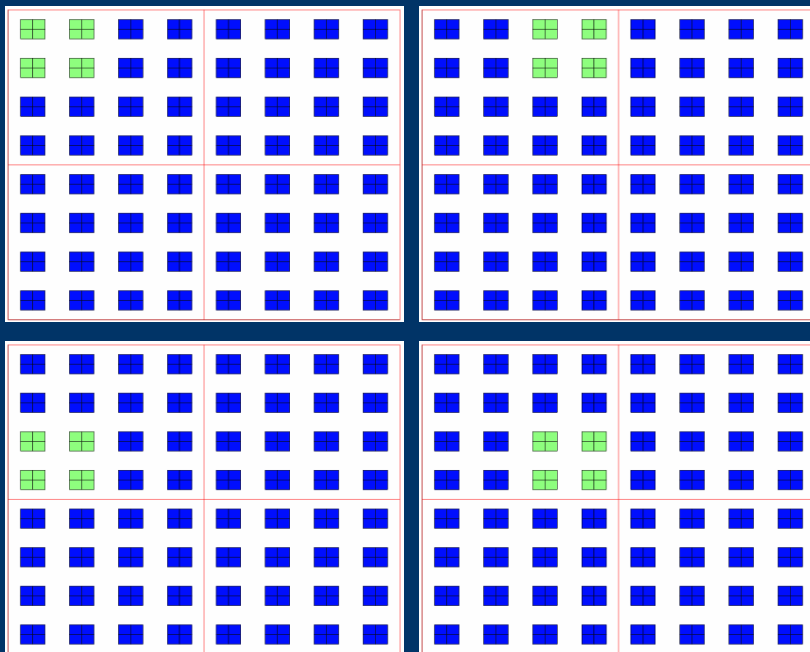


# Multilevel method: next level

Basis functions pushed  
up to next level

Transformed basis functions  
on next level

- Voltage: +1 -1 0





# More Precisely: Insure vanishing moments

- Just “balanced voltages”: somewhat faster dropoff
- If several vanishing moments: faster dropoff
- Moments defined:  $\mu_{\alpha,\beta,s}(\sigma) = \int_s x'^{\alpha} y'^{\beta} \sigma(x, y) dy dx$   
with  $(x', y') = (x, y) - \text{centroid}(s)$
- Want basis functions w/vanishing moments to order  $p$ : for our examples  $p = 2$
- Balanced voltages: just 0-order moments

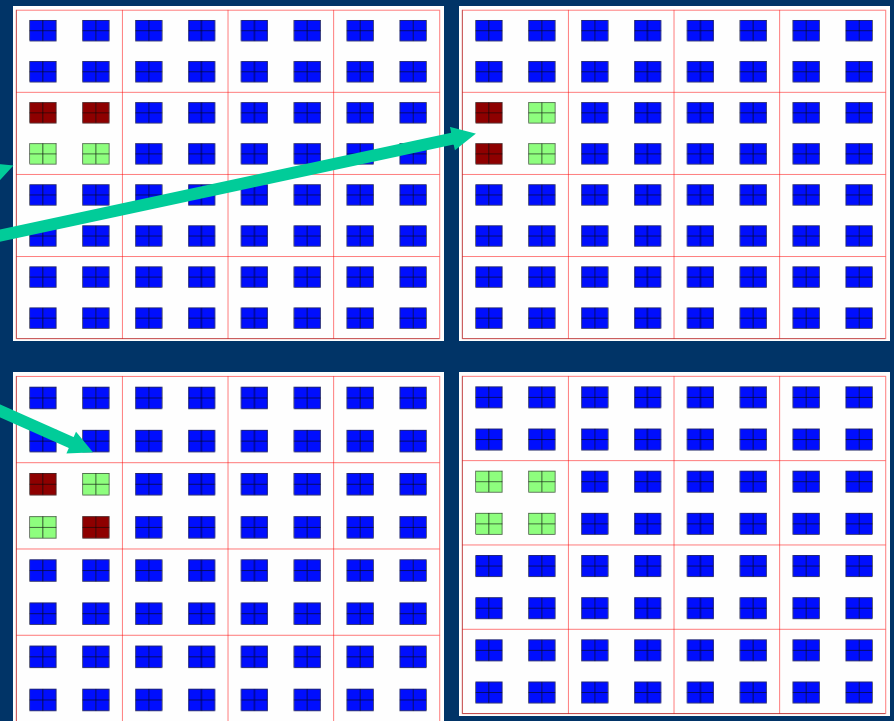
$$\mu_{0,0,s}(\sigma) = \int_s \sigma(x, y) dy dx$$

# Multilevel method: moment view

- Voltage  $\sigma(x, y) : +1 \ -1 \ 0$

$$\int \sigma(x, y) dx dy = 0$$

Transformed basis functions



# Sparsified representation of $G$

- Get current responses to *transformed* voltage basis vectors
- Put current responses in the *transformed* basis
- Get wavelet-basis matrix  $G_w$ 
  - Numerically sparse matrix
  - $G = QG_wQ'$ 
    - $Q$  is change of basis matrix
    - Defined by multilevel transformation
- $G_w$  is numerically sparse
  - Threshold out small entries to obtain
  - $G_{wt}$  : cheap to apply ( $O(n \log n)$  for  $n$  contacts)

$$QG_{wt}Q' \approx G$$

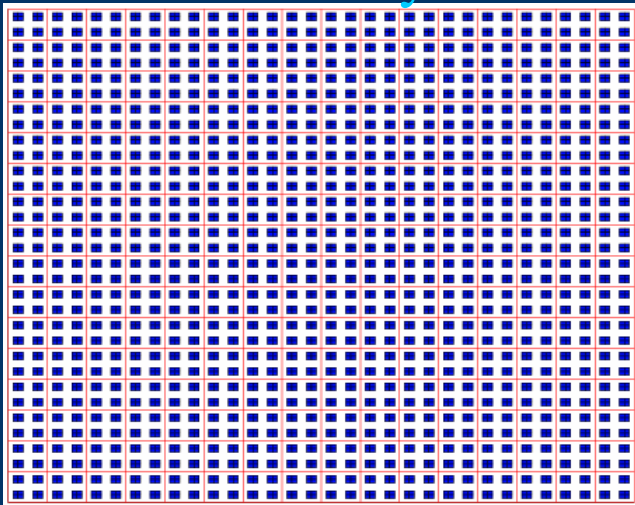
# Measuring results

- Sparsity of  $G_{wt}$  obtained by thresholding
  - Error of approximation depends on threshold
  - Arbitrary sparsity possible with high enough threshold
- Key is estimation of error  $\text{norm}(QG_{wt}Q' - G)$ 
  - maximum error vector/input vector length ratio:  
 $\text{norm}(QG_{wt}Q' - G)$
  - How to get error estimate without calculating  $G$ ?
    - Use iterative method for norm error estimation
    - Only need apply
    - Can apply  $G$  by using the solver
  - For comparison: find  $\text{norm}(G_t - G)$ 
    - $G_t$  is thresholded  $G$

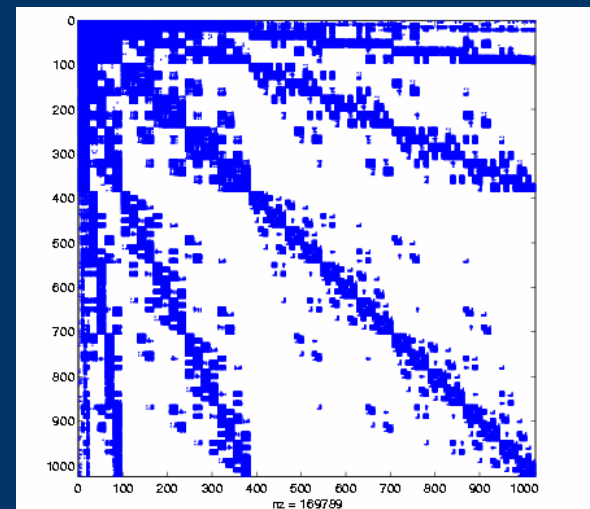
# Results: regular grid

- 1024 contacts on  $\sigma / 100\sigma$  conductivity profile
- $G_{wt}$  : 16% nonzero, .001 scaled L2 error
- $G_t$  : 37% nonzero, .3 scaled L2 error

Contact layout



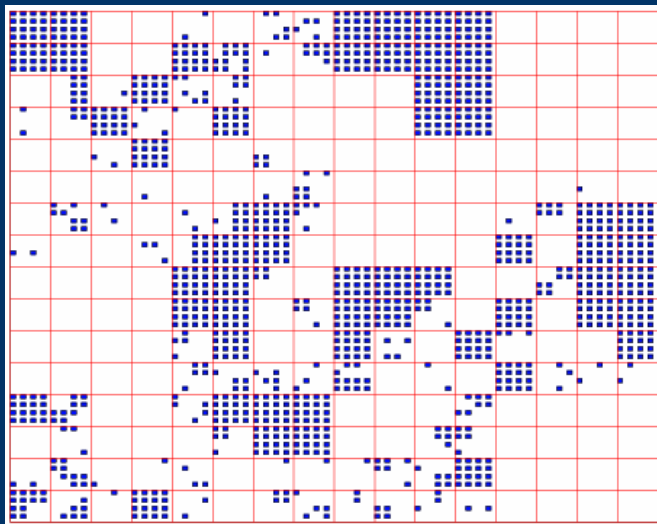
$G_{wt}$  sparsity structure



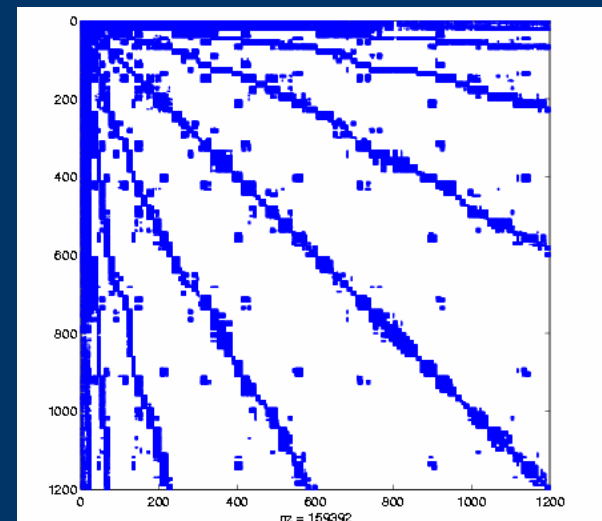
# Results: irregular grid

- 1199 contacts on  $\sigma/100\sigma$  conductivity profile
- $G_{wt}$  : 11% nonzero entries, .002 scaled L2 error
- $G_t$  : 21% nonzero entries, .2 scaled L2 error

Contact layout



$G_{wt}$  sparsity structure



# Wavelet Sparsification Summary

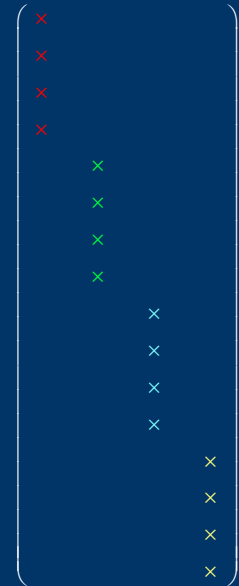
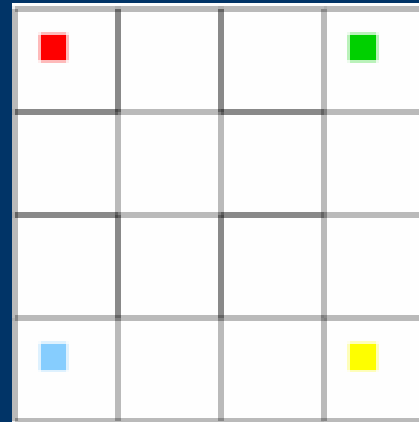
Example	Contacts	Sparsification	L2 Error
Regular	1024	6X	1e-3
Irregular	1199	9X	2e-3
Regular	4096	16X	2e-3
Irregular	10000	30X	1e-3





# Our sparsity structure: same principle

- Similar to tridiagonal example:
  - Add several voltage vectors
    - Feed the sum to the solver
  - OK to do this when?
    - Current responses have no overlapping non-zero entries
    - Reasonable if there are no overlapping large entries



# Solve reduction results

Example	Contacts	Solve Reduction	L2 Error
Regular	1024	3x	1e-3
Irregular	1199	3x	3e-3
Regular	4096	8x	2e-3
Irregular	10000	20X	1e-3

Implies nearly  $O(N^2) \rightarrow O(N)$  complexity reduction

# Summary

- Biomolecule Electrostatics
  - Carefully Chosen Integral Formulation
  - Sparsification technique for multiple kernels
  - Problem specific preconditioning
- Sparsification of IC Substrate Coupling
  - Carefully chosen basis for representation (Wavelet like)
  - Overlapping solves exploits sparse representation

**For everything you see, for everything you do,  
computational techniques are right for you**