The Process of Computational Science

Matthew Knepley

Computation Institute
University of Chicago

Department of Molecular Biology and Physiology
Rush University Medical Center

Applied Physics and Applied Mathematics
School of Engineering and Applied Science, Columbia University
New York, NY February 18, 2013

My approach to
Computational Science is

Holistic

My approach to
Computational Science is

Holistic

Computational Science

starting with the numerics of PDEs,

and mathematics of the computation,

through the distillation into

high quality numerical libraries,

to scientific discovery through computing.

Computational Science

starting with the numerics of PDEs,

and mathematics of the computation,

through the distillation into

high quality numerical libraries,

to scientific discovery through computing.

Computational Science

starting with the numerics of PDEs,

and mathematics of the computation,

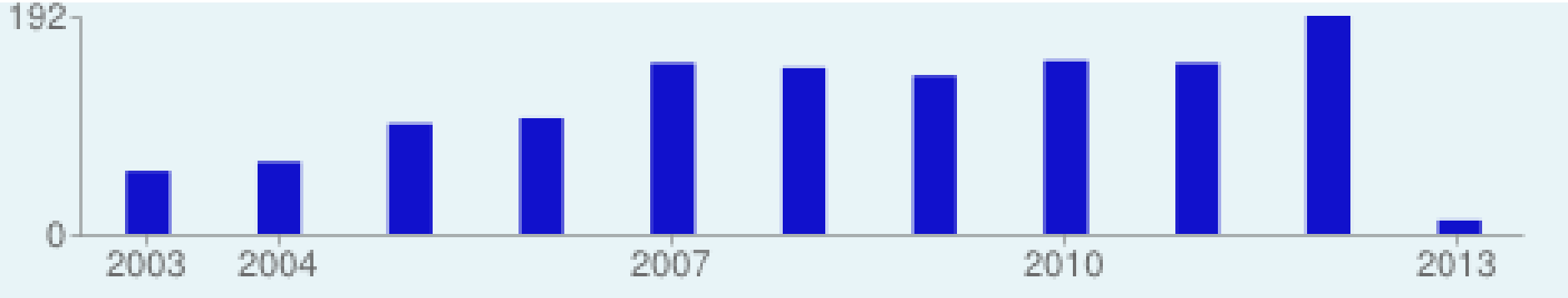
through the distillation into

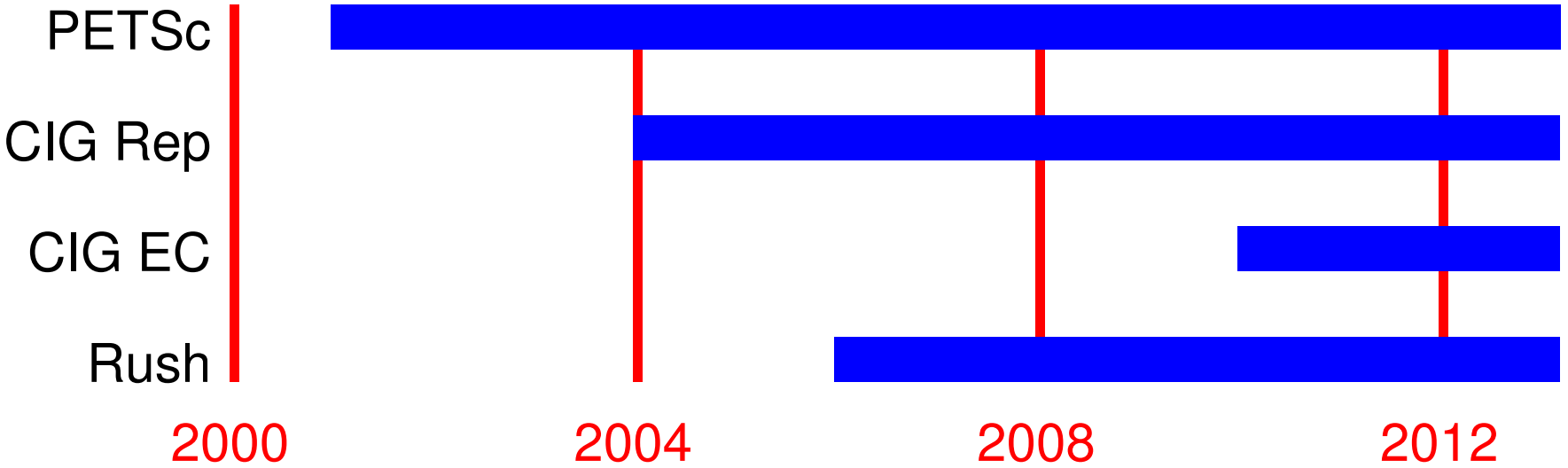
high quality numerical libraries,

to scientific discovery through computing.

Community Involvement

PETSc Citations





Outline

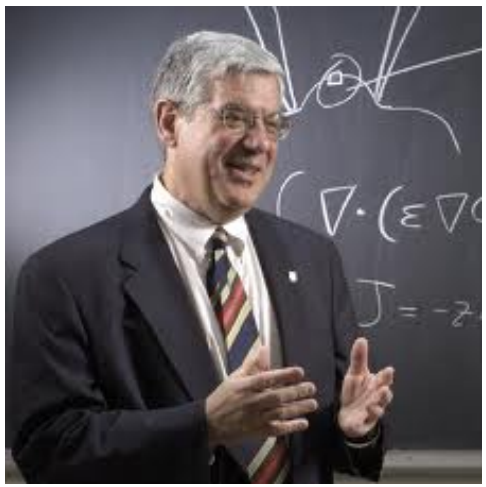
Collaborators

BIBEE Researchers

Jaydeep Bardhan

Classical DFT Researchers

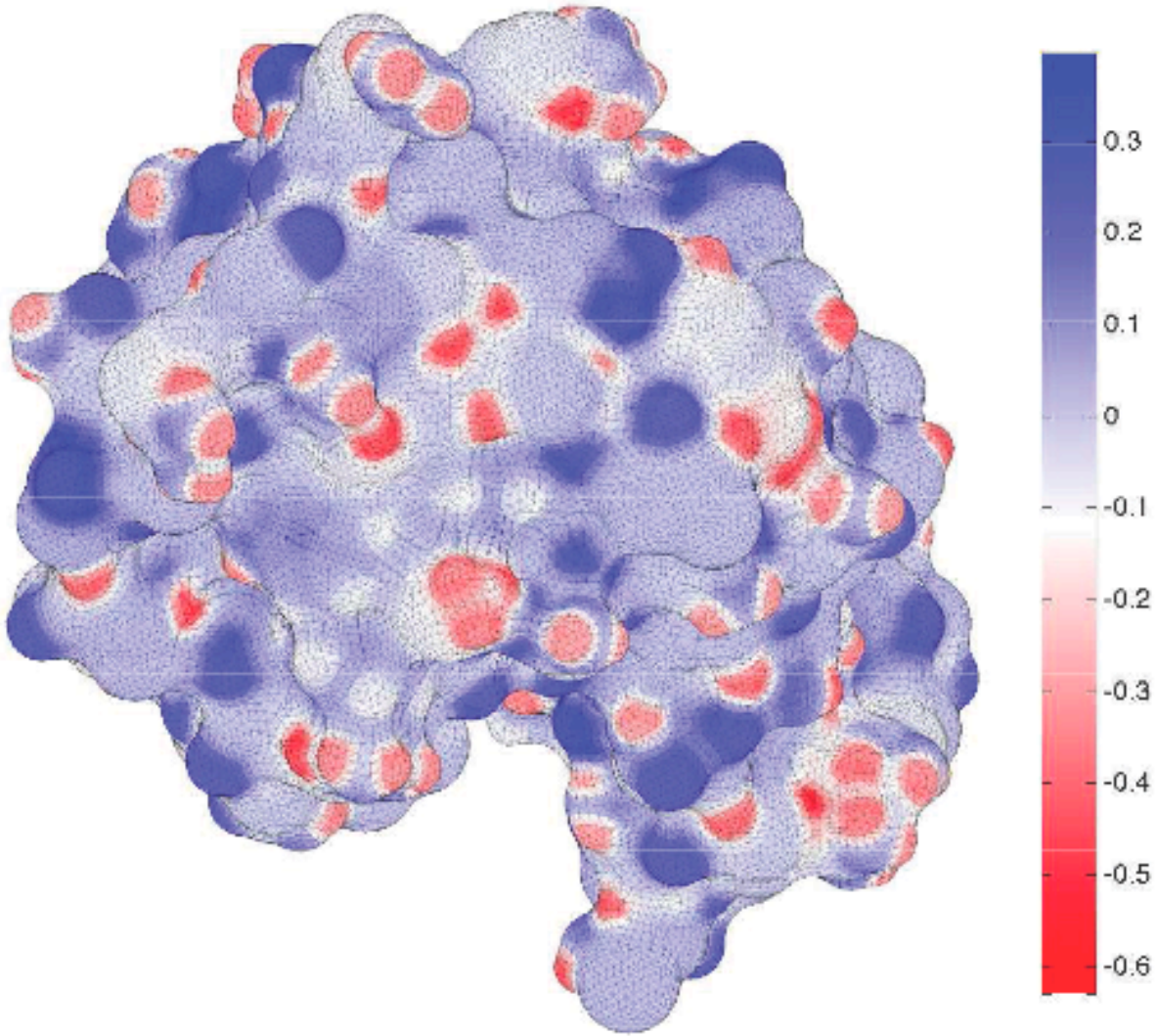
Dirk Gillespie



Bob Eisenberg

Bioelectrostatics

The Natural World

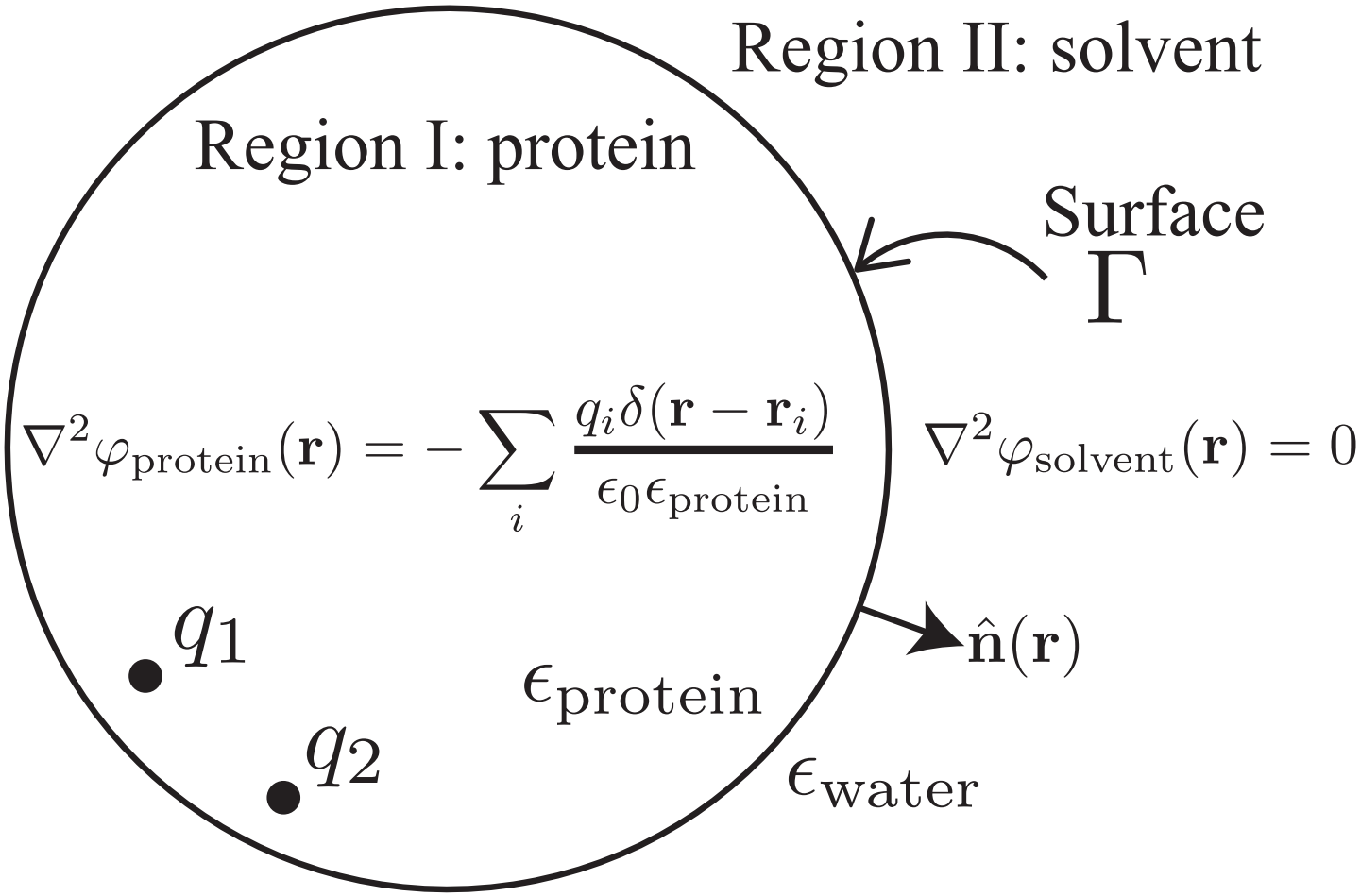


Induced Surface Charge on Lysozyme

Bioelectrostatics

Physical Model

Electrostatic Potential ϕ



Bioelectrostatics

Mathematical Model

We can write a Boundary Integral Equation (BIE) for the induced surface charge σ,

$$\sigma(\vec{r}) + \hat{\epsilon} \int_\Gamma \frac{\partial}{\partial n(\vec{r})} \frac{\sigma(\vec{r}')d^2\vec{r}'}{4\pi||\vec{r} - \vec{r}'||} = -\hat{\epsilon} \sum_{k=1}^{Q} \frac{\partial}{\partial n(\vec{r})} \frac{q_k}{4\pi||\vec{r} - \vec{r}_k||}$$

$$(\mathcal{I} + \hat{\epsilon}\mathcal{D}^*)\,\sigma(\vec{r}) =$$

where we define

$$\hat{\epsilon} = 2\frac{\epsilon_I - \epsilon_{II}}{\epsilon_I + \epsilon_{II}} < 0$$

Bioelectrostatics

Mathematical Model

The *reaction* potential is given by

$$\phi^R(\vec{r}) = \int_\Gamma \frac{\sigma(\vec{r}')d^2\vec{r}'}{4\pi\epsilon_1||\vec{r}-\vec{r}'||} = C\sigma$$

which defines G_{es}, the electrostatic part of the solvation free energy

$$\begin{aligned}\Delta G_{es} &= \frac{1}{2}\left\langle q, \phi^R \right\rangle \\ &= \frac{1}{2}\langle q, Lq\rangle \\ &= \frac{1}{2}\left\langle q, CA^{-1}Bq\right\rangle\end{aligned}$$

where

$$\begin{aligned}Bq &= -\hat{\epsilon}\int_\Omega \frac{\partial}{\partial n(\vec{r})}\frac{q(\vec{r}')d^3\vec{r}'}{4\pi||\vec{r}-\vec{r}'||} \\ A\sigma &= \mathcal{I} + \hat{\epsilon}\mathcal{D}^*\end{aligned}$$

Problem

Boundary element discretizations of the solvation problem (Eq. **??**):

- can be expensive to solve
- are more accurate than required by intermediate design iterations

BIBEE

Approximate $\mathcal{D}^*$ by a diagonal operator

Boundary Integral-Based Electrostatics Estimation

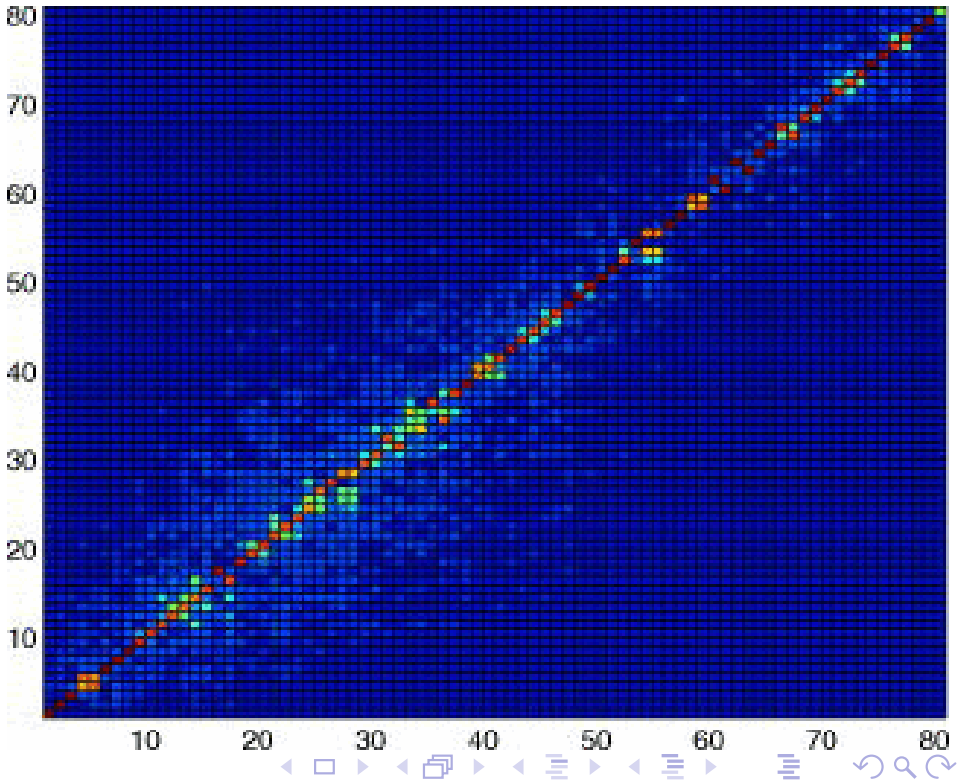
Coulomb Field Approximation:
uniform normal field

$$\left(1 - \frac{\hat{\epsilon}}{2}\right) \sigma_{CFA} = Bq$$

Lower Bound:
no good physical motion

$$\left(1 + \frac{\hat{\epsilon}}{2}\right) \sigma_{LB} = Bq$$

Eigenvectors: BEM $e_i \cdot e_j$ BIBEE/P



BIBEE

Approximate $\mathcal{D}^*$ by a diagonal operator

Boundary Integral-Based Electrostatics Estimation

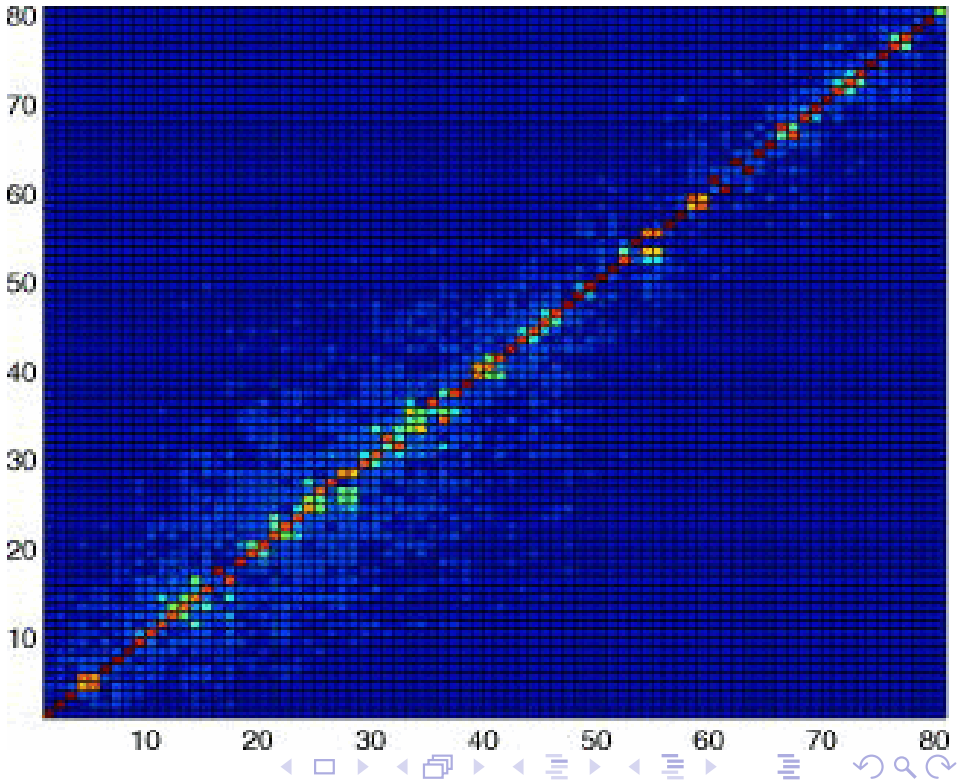
Coulomb Field Approximation:
uniform normal field

$$\left(1 - \frac{\hat{\epsilon}}{2}\right) \sigma_{CFA} = Bq$$

Preconditioning:
consider only local effects

$$\sigma_P = Bq$$

Eigenvectors: BEM $e_i \cdot e_j$ BIBEE/P



BIBEE Bounds on Solvation Energy

Theorem: The electrostatic solvation energy ΔG_{es} has upper and lower bounds given by

$$\frac{1}{2}\left(1+\frac{\hat{\epsilon}}{2}\right)^{-1}\langle q, CBq\rangle \leq \frac{1}{2}\left\langle q, CA^{-1}Bq\right\rangle \leq \frac{1}{2}\left(1-\frac{\hat{\epsilon}}{2}\right)^{-1}\langle q, CBq\rangle,$$

and for spheres and prolate spheroids, we have the improved lower bound,

$$\frac{1}{2}\langle q, CBq\rangle \leq \frac{1}{2}\left\langle q, CA^{-1}Bq\right\rangle,$$

and we note that

$$|\hat{\epsilon}| < \frac{1}{2}.$$

Energy Bounds:

Proof: Bardhan, Knepley, Anitescu, JCP, **130**(10), 2008

I will break the proof into three steps,

- Replace C with B
- Symmetrization
- Eigendecomposition

shown in the following slides.

We will need the single layer operator $\mathcal{S}$ for step 1,

$$\mathcal{S}\tau(\vec{r}) = \int \frac{\tau(\vec{r}')d^2\vec{r}'}{4\pi||\vec{r} - \vec{r}'||}$$

Energy Bounds: First Step

Replace C with B

The potential at the boundary Γ given by

$$\phi^{Coulomb}(\vec{r}) = C^T q$$

can also be obtained by solving an exterior Neumann problem for τ,

$$\begin{aligned}
\phi^{Coulomb}(\vec{r}) &= \mathcal{S}\tau \\
&= \mathcal{S}(\mathcal{I} - 2\mathcal{D}^*)^{-1}(\frac{2}{\hat{\epsilon}}Bq) \\
&= \frac{2}{\hat{\epsilon}}\mathcal{S}(\mathcal{I} - 2\mathcal{D}^*)^{-1}Bq
\end{aligned}$$

so that the solvation energy is given by

$$\frac{1}{2}\left\langle q, CA^{-1}Bq\right\rangle = \frac{1}{\hat{\epsilon}}\left\langle \mathcal{S}(\mathcal{I} - 2\mathcal{D}^*)^{-1}Bq, (\mathcal{I} + \hat{\epsilon}\mathcal{D}^*)^{-1}Bq\right\rangle$$

Energy Bounds: Second Step

Quasi-Hermiticity

Plemelj's symmetrization principle holds that

$$\mathcal{S}\mathcal{D}^* = \mathcal{D}\mathcal{S}$$

and we have

$$\mathcal{S} = \mathcal{S}^{1/2}\mathcal{S}^{1/2}$$

which means that we can define a Hermitian operator H similar to $\mathcal{D}^*$

$$H = \mathcal{S}^{1/2}\mathcal{D}^*\mathcal{S}^{-1/2}$$

leading to an energy

$$\frac{1}{2}\left\langle q, CA^{-1}Bq\right\rangle = \frac{1}{\hat{\epsilon}}\left\langle Bq, \mathcal{S}^{1/2}(\mathcal{I} - 2H)^{-1}(\mathcal{I} + \hat{\epsilon}H)^{-1}\mathcal{S}^{1/2}Bq\right\rangle$$

Energy Bounds: Third Step

Eigendecomposition

The spectrum of $\mathcal{D}^*$ is in $[-\frac{1}{2}, \frac{1}{2})$, and the energy is

$$\frac{1}{2}\left\langle q, CA^{-1}Bq\right\rangle = \sum_i \frac{1}{\hat{\epsilon}}\left(1-2\lambda_i\right)^{-1}\left(1+\hat{\epsilon}\lambda_i\right)^{-1}x_i^2$$

where

$$H = V\Lambda V^T$$

and

$$\vec{x} = V^T\mathcal{S}^{1/2}Bq$$

Energy Bounds: Diagonal Approximations

The BIBEE approximations yield the following bounds

$$\frac{1}{2}\left\langle q, CA_{CFA}^{-1}Bq\right\rangle = \sum_i \frac{1}{\hat{\epsilon}}\left(1-2\lambda_i\right)^{-1}\left(1-\frac{\hat{\epsilon}}{2}\right)^{-1}x_i^2$$

$$\frac{1}{2}\left\langle q, CA_{P}^{-1}Bq\right\rangle = \sum_i \frac{1}{\hat{\epsilon}}\left(1-2\lambda_i\right)^{-1}x_i^2$$

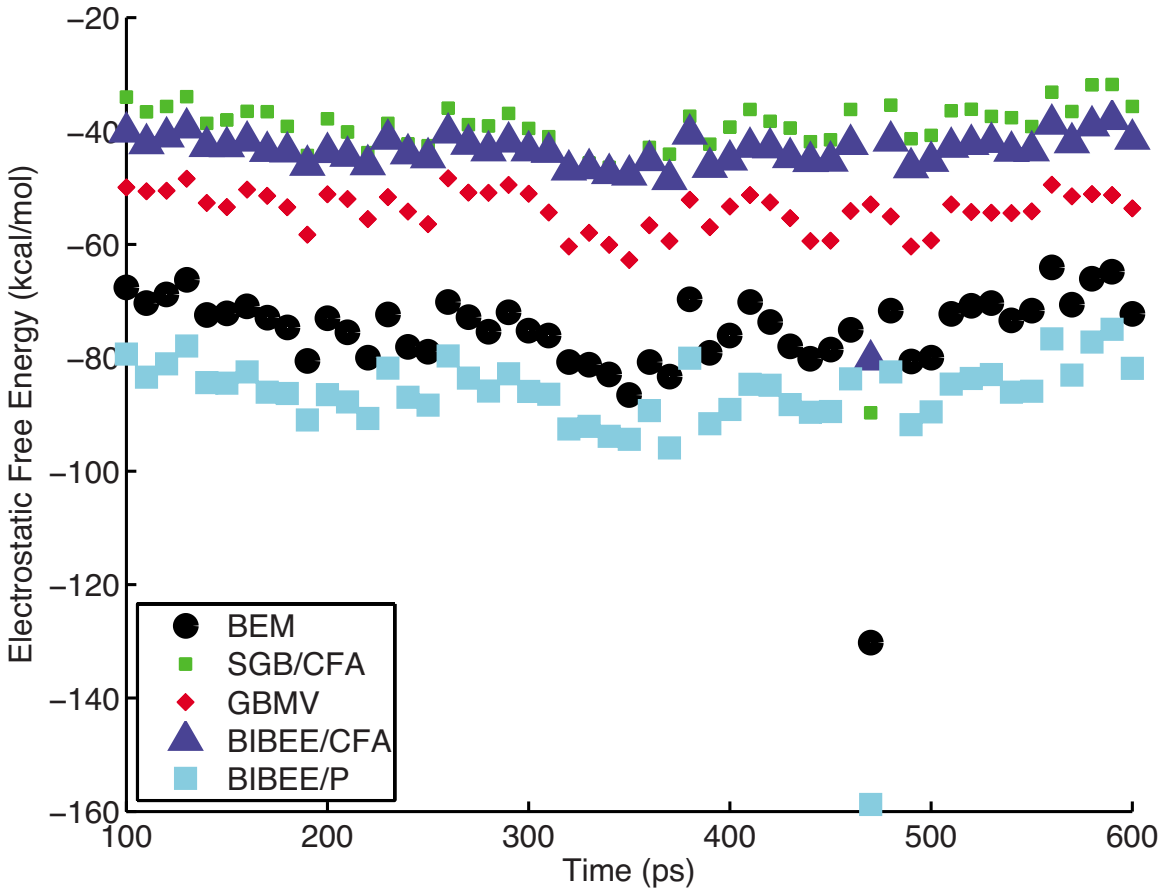
$$\frac{1}{2}\left\langle q, CA_{LB}^{-1}Bq\right\rangle = \sum_i \frac{1}{\hat{\epsilon}}\left(1-2\lambda_i\right)^{-1}\left(1+\frac{\hat{\epsilon}}{2}\right)^{-1}x_i^2$$

where we note that

$$|\hat{\epsilon}| < \frac{1}{2}$$

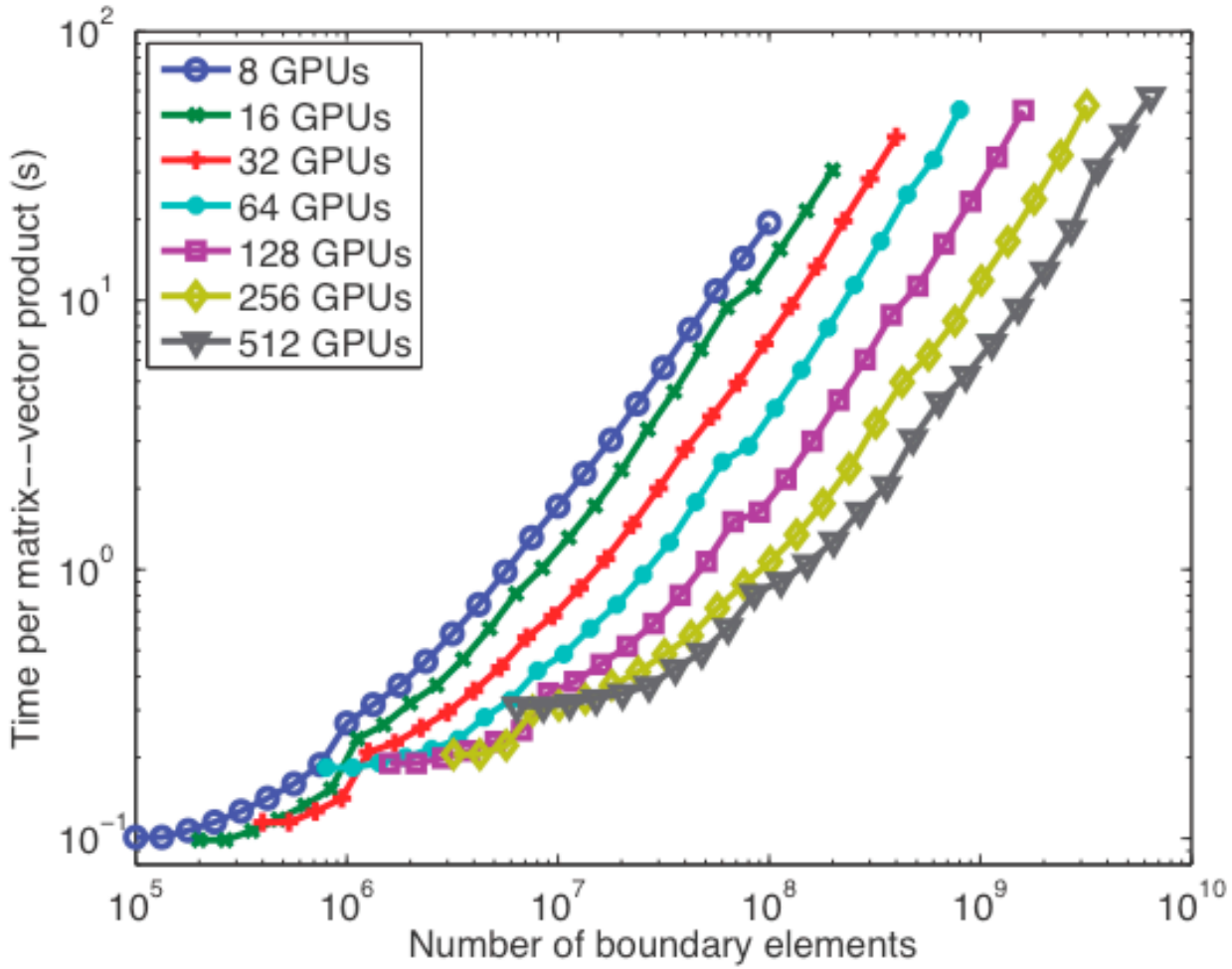
BIBEE Accuracy

Electrostatic solvation free energies of met-enkephalin structures



Snapshots taken from a 500-ps MD simulation at 10-ps intervals.
Bardhan, Knepley, Anitescu, JCP, 2009.

BIBEE Scalability



Yokota, Bardhan, Knepley, Barba, Hamada, CPC, 2011.

Resolution

Boundary element discretizations of the solvation problem:

- can be expensive to solve
  - **Bounding the electrostatic free energies associated with linear continuum models of molecular solvation**, JCP, 2009
- are more accurate than required by intermediate design iterations
  - Accuracy is not tunable

Evolution of BIBEE

- Sharp bounds for solvation energy
- Exploration of behavior in simplified geometries
  - **Mathematical Analysis of the BIBEE Approximation for Molecular Solvation: Exact Results for Spherical Inclusions**, JCP, 2011
  - Represent BIBEE as a deformed boundary condition
  - Fully developed series solution
  - Improve accuracy by combining CFA and P approximations
- Application to protein-ligand binding
  - **Analysis of fast boundary-integral approximations for modeling electrostatic contributions of molecular binding**, Molecular-Based Mathematical Biology, 2013

Future of BIBEE

- Framework for systematic exploration
  - Both analytical and computational foundation
- Reduced-basis Method with analytic solutions
  - Tested in protein binding paper above
  - The spatial high frequency part is handled by BIBEE/P topology is not important
  - The spatial low frequency part is handled by analytic solutions insensitive to bumpiness
  - **Computational science and re-discovery: open-source implementations of ellipsoidal harmonics for problems in potential theory**, CSD, 2012.
- Extend to other kernels, e.g. Yukawa
- Extend to full multilevel method

Outline

Collaborators

PETSc Developers

Barry Smith

Jed Brown

Former UC Students

Andy Terrel

Peter Brune

Problem

Traditional PDE codes cannot:

- Compare different discretizations
  - Different orders, finite elements
  - finite volume vs. finite element
- Compare different mesh types
  - Simplicial, hexahedral, polyhedral, octree
- Run 1D, 2D, and 3D problems
- Enable an optimal solver
  - Fields, auxiliary operators

Problem

Traditional Mesh/Solver Interface is Too General:

- Solver not told about discretization data, e.g. fields
- Cannot take advantage of problem structure
  - blocking
  - saddle point structure
- Cannot use auxiliary data
  - Eigen-estimates
  - null spaces

Problem

Traditional Mesh/Solver Interface is Too Specific:

- Assembly code specialized to each discretization
  - dimension
  - cell shape
  - approximation space
- Explicit references to element type
  - getVertices(faceID), getAdjacency(edgeID, VERTEX), getAdjacency(edgeID, dim = 0)
- No interface for transitive closure
  - Awkward nested loops to handle different dimensions

Mesh Representation

We represent each mesh as a Hasse Diagram:

- Can represent any CW complex
- Can be implemented as a Directed Acyclic Graph
- Reduces mesh information to a single *covering* relation
- Can discover dimension, since meshes are ranked posets

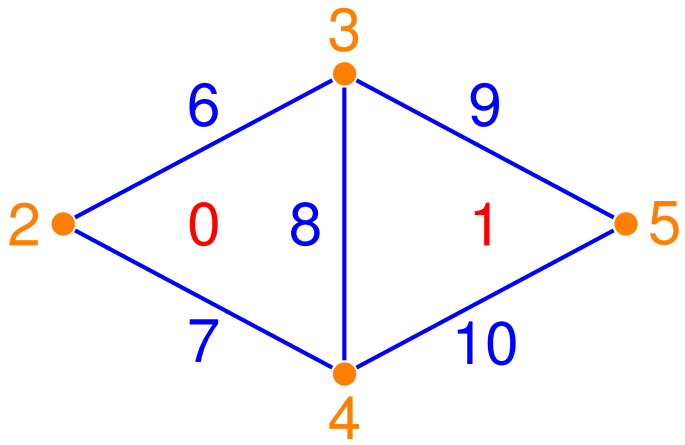
We use an abstract topological interface to organize traversals for:

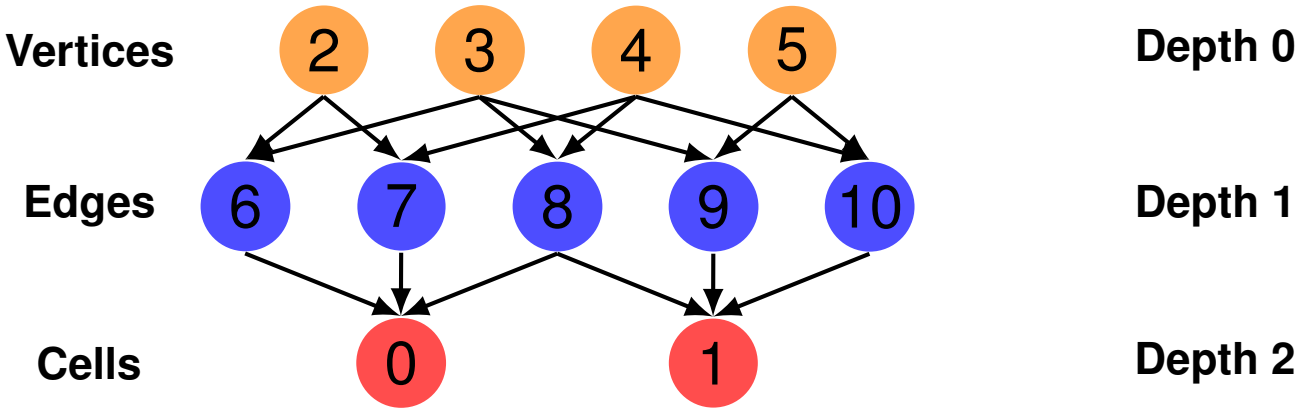
- discretization integrals
- solver size determination
- computing communication patterns

Mesh geometry is treated as just another mesh function.

Sample Meshes

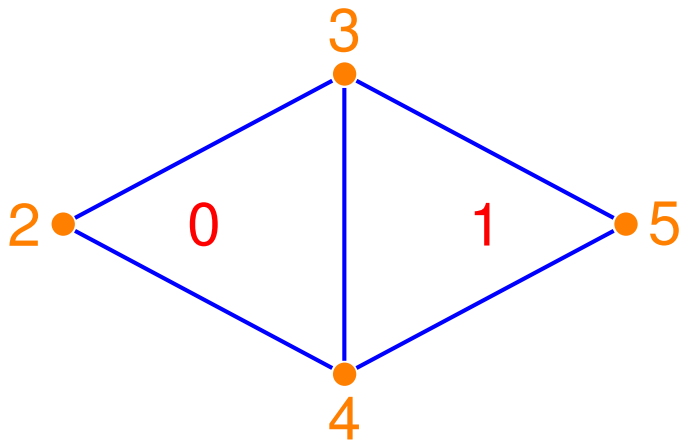
Interpolated triangular mesh

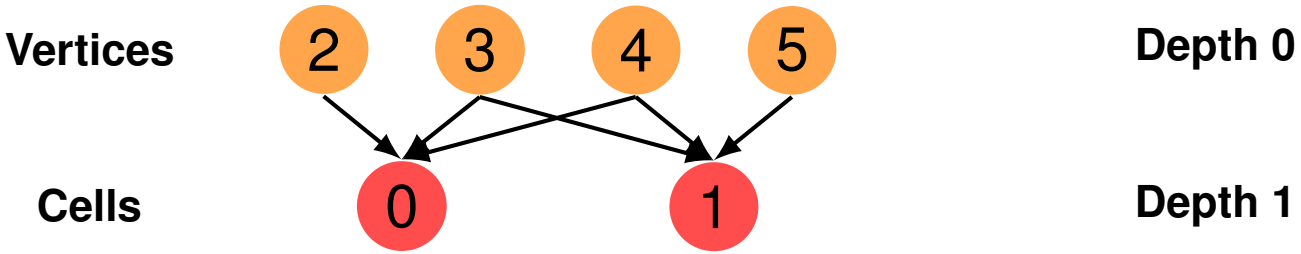




Sample Meshes

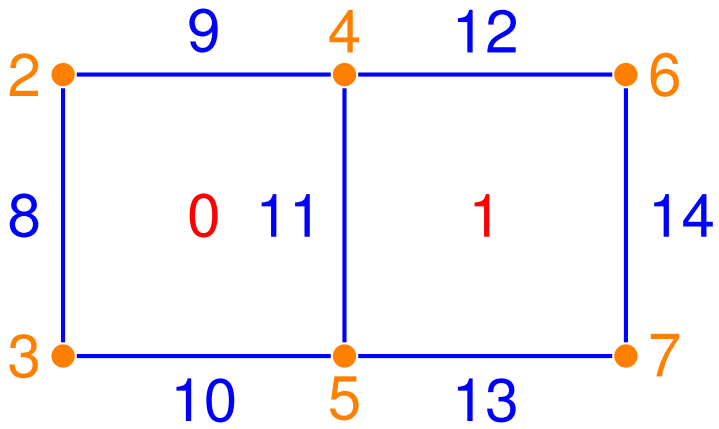
Optimized triangular mesh

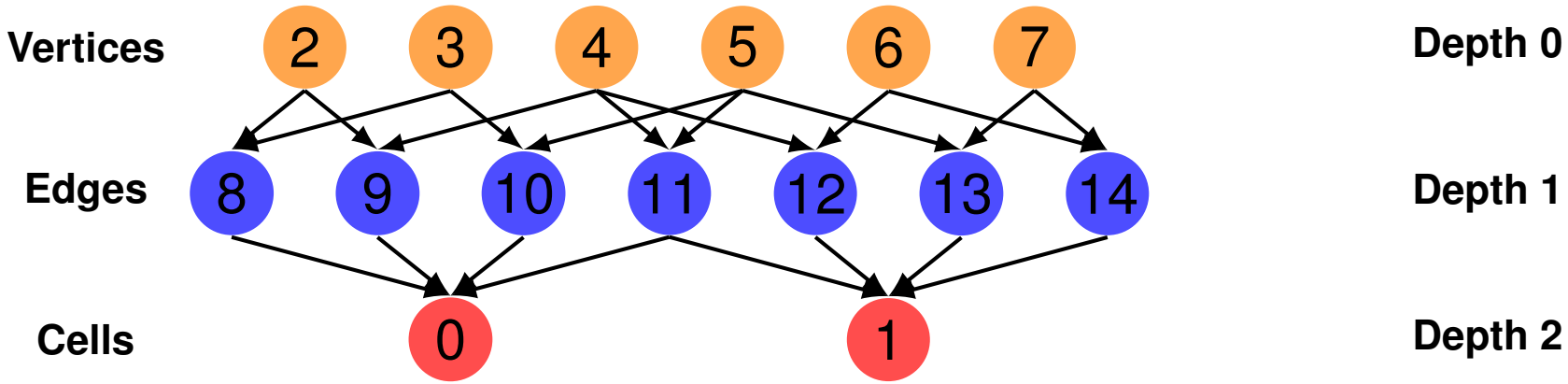




Sample Meshes

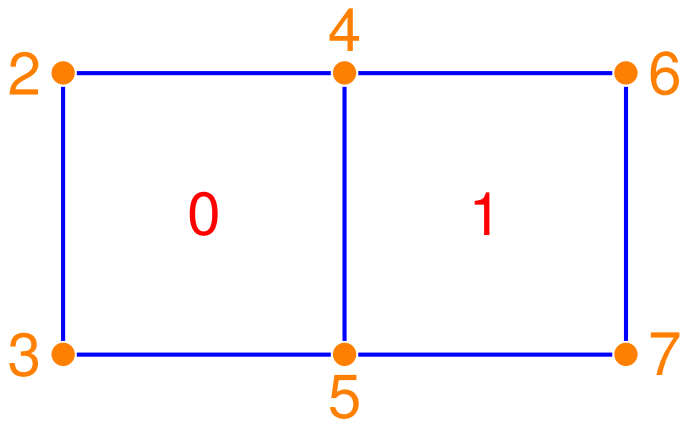
Interpolated quadrilateral mesh

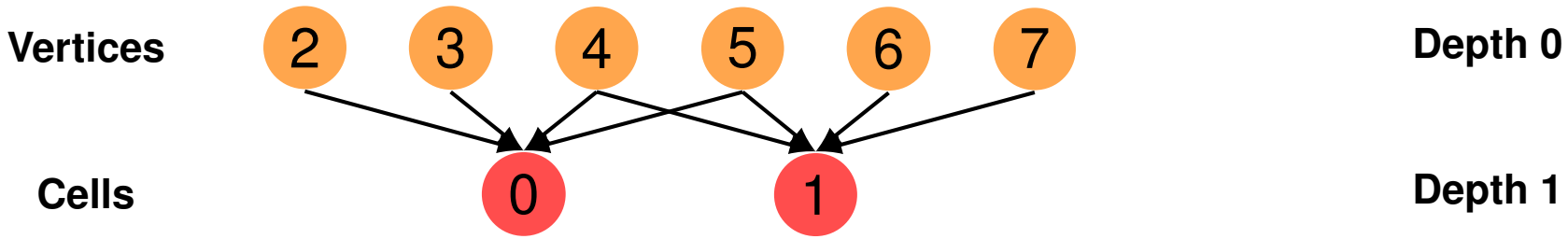




Sample Meshes

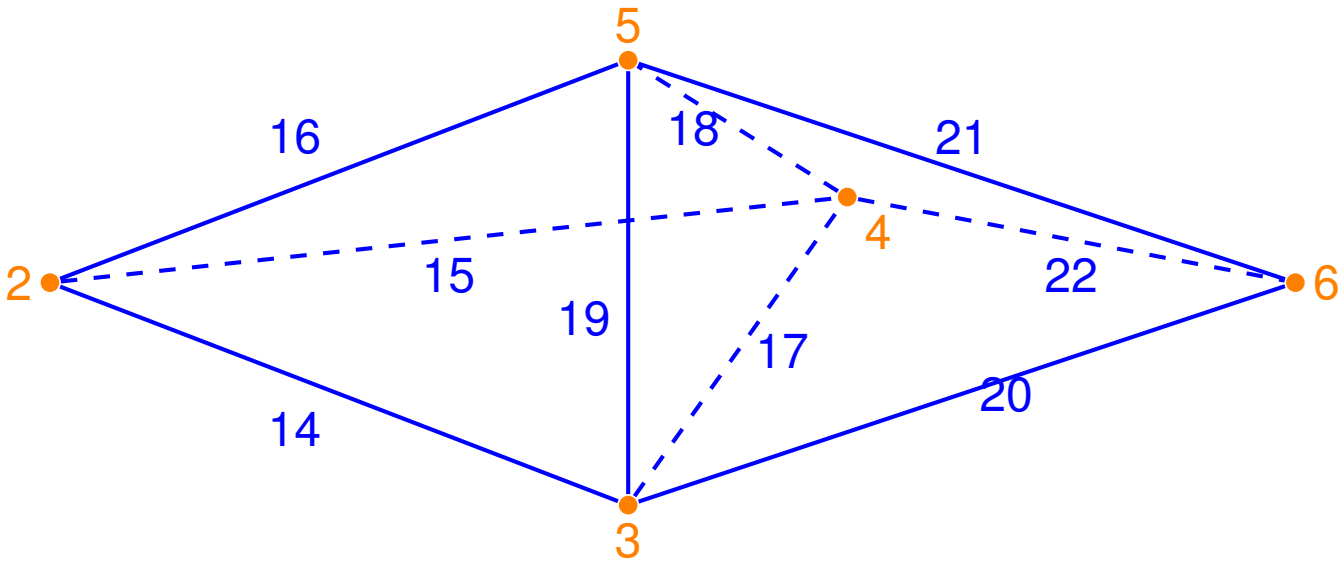
Optimized quadrilateral mesh

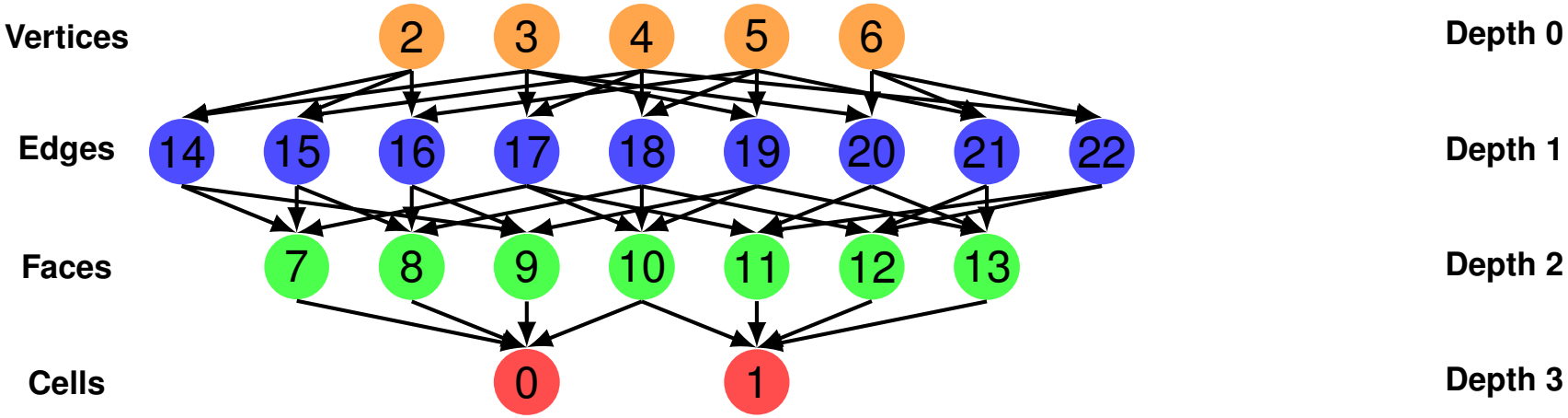




Sample Meshes

Interpolated tetrahedral mesh





Mesh Abstraction

Interface Design

By abstracting on the key topological relations, the interface can be both concise and quite general

- Single relation
- Enables dimension-independent programming
- Dual is obtained by reversing arrows
- Can associate function(al)s with DAG points
- Dual operation gives the support of the function

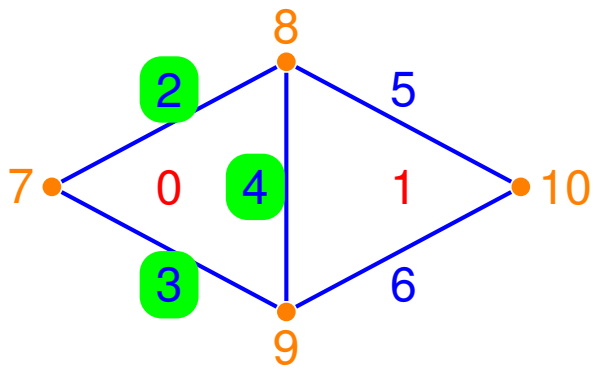
Mesh Algorithms for PDE with Sieve I: Mesh Distribution, Knepley, Karpeev, Sci. Prog., 2009.

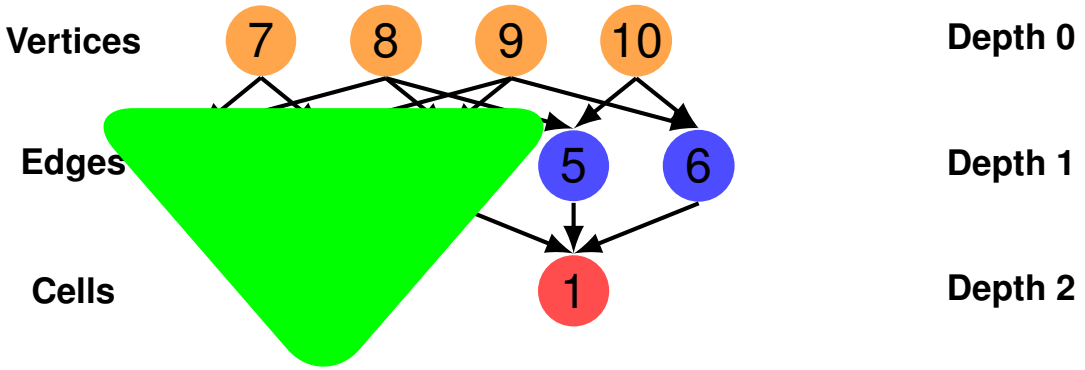
Basic Operations

Cone

We begin with the basic covering relation,

cone(0) = {2, 3, 4}



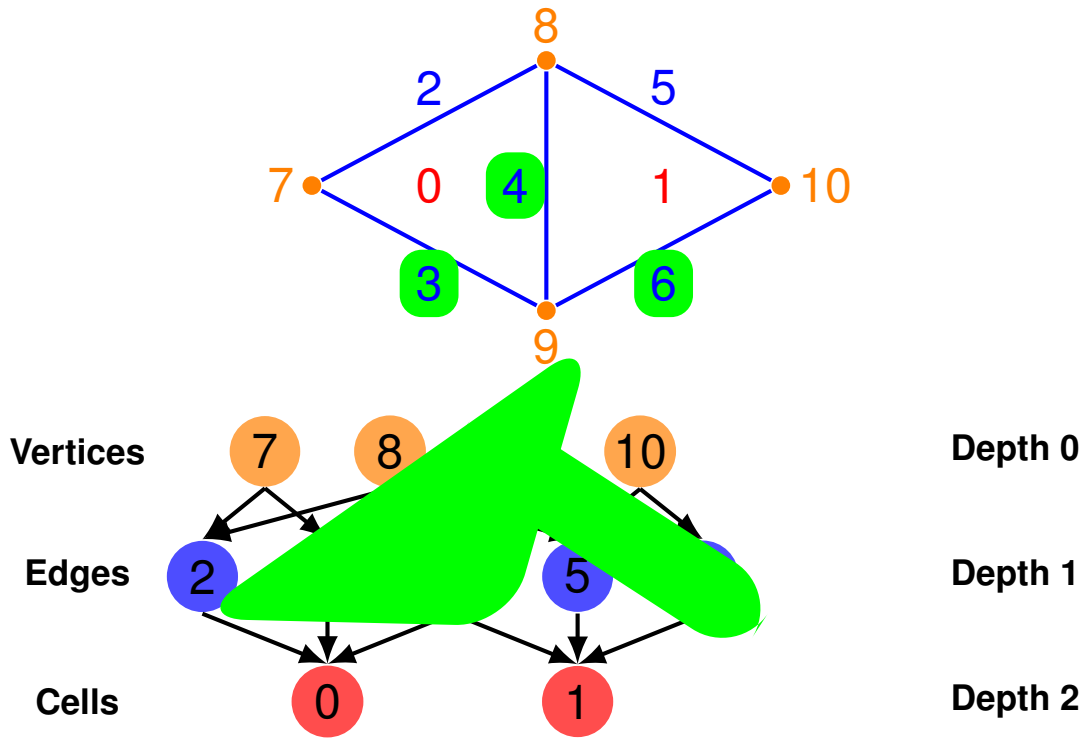


Basic Operations

Support

reverse arrows to get the dual operation,

support(9) = {3, 4, 6}

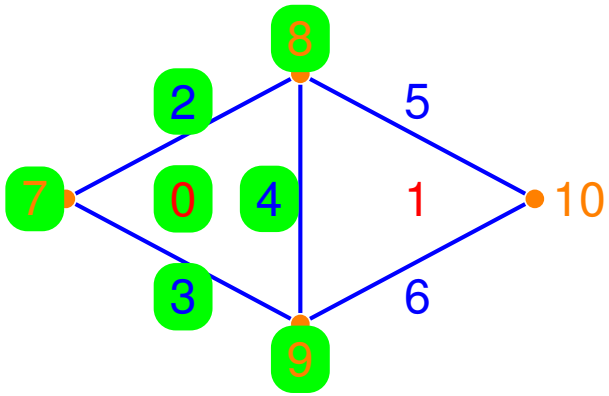


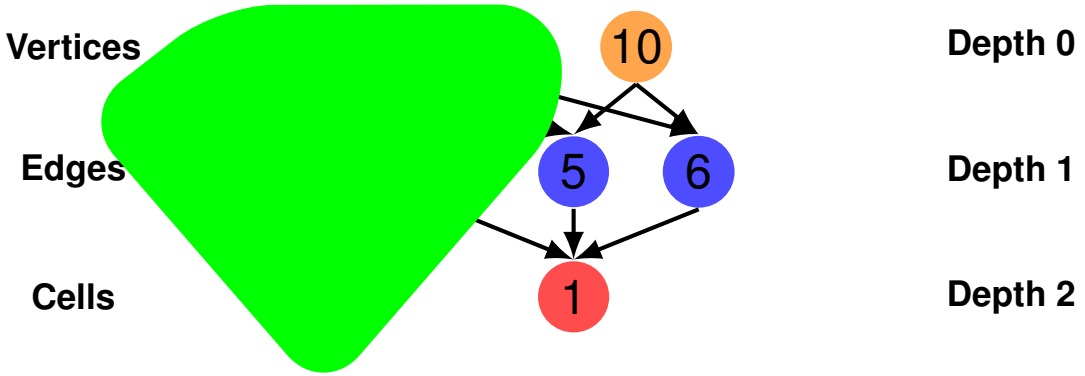
Basic Operations

Closure

add the transitive closure of the relation,

closure(0) = {0, 2, 3, 4, 7, 8, 9}



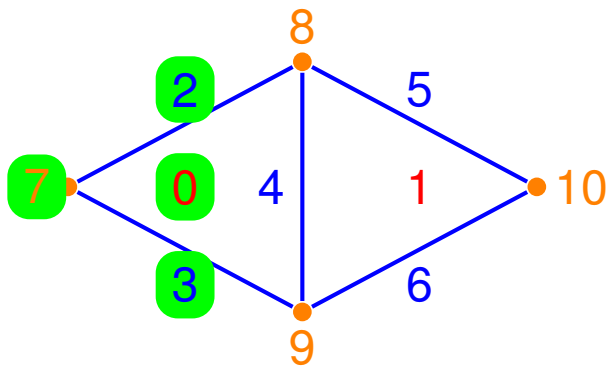


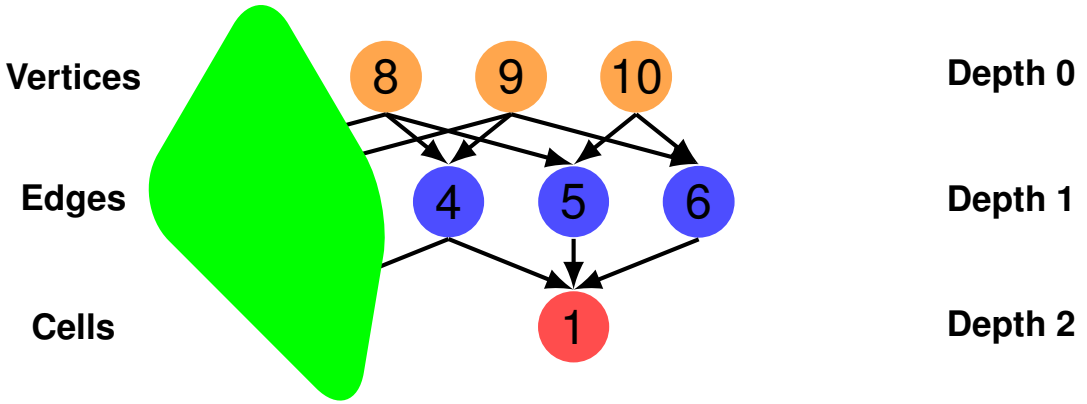
Basic Operations

Star

and the transitive closure of the dual,

star(7) = {7, 2, 3, 0}



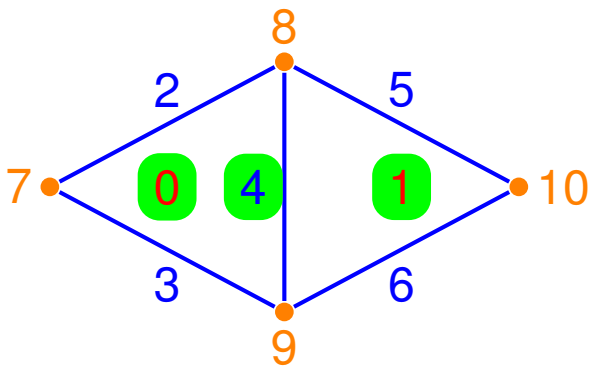


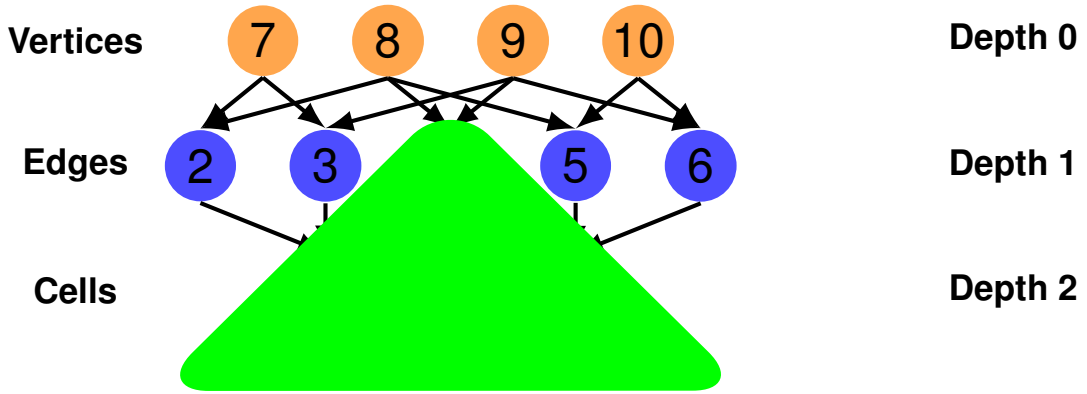
Basic Operations

Meet

and augment with lattice operations.

meet(0, 1) = {4}





Basic Operations

Join

and augment with lattice operations.

join(8, 9) = {4}