High Performance Computing Workshop

DOE CSGF Fellow and Alumni HPC Stories:

Richard Tran Mills

Outline

- Brief computing bio
- Super-condensed intro to PFLOTRAN:
  What does it solve? Why do we care?
- Leveraging PETSc to develop PFLOTRAN
  (and stay sane in the process!)

Key waypoints in my computing career

- Received a Commodore-64 for Christmas of 1984!
- Studied geology and physics at University of Tennessee, 1995-1999
  - Particular interest in computational geophysics; senior thesis on computational geomorphology
- Decided to go to grad school in computer science, with specialization in computational science
  - Felt that I was better at computing than physics
  - Better employment prospects
  - Still get to work on cool physics problems, but no one expects me to be good at the physics
- CSGF practicum in 2003 with Peter Lichtner at LANL's EES Division

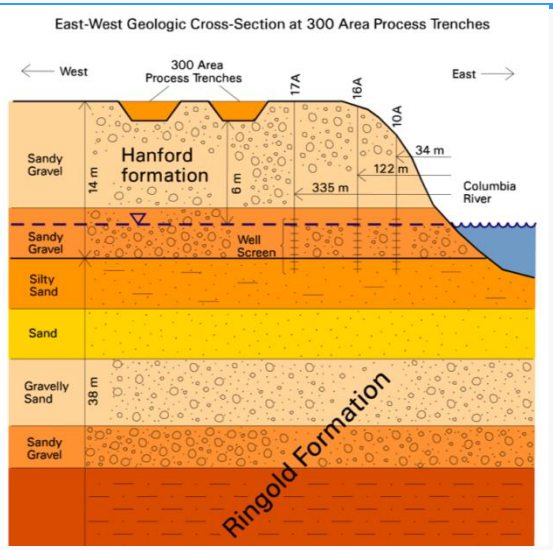
CSGF Practicum at LANL

- Ended up at LANL because I saw Glenn Hammond's CSGF talk.
- Glenn had worked on the reactive transport solver in PFLOTRAN (http://software.lanl.gov/pflotran), a code for continuum-scale simulation of multiscale, multiphase, multicomponent flow and reactive transport in porous media.
- I worked on initial implementation of the flow solver.
- I had worked on HPC solvers/kernels from a computer scientist's perspective, but this was my first experience working on a full application targeting HPC.
- ASCI QSC ("Little Q"), an AlphaServer machine was our big HPC resource
- Today, PFLOTRAN development is SciDAC-funded and we use machines like Jaguar, the Cray XT5 at ORNL
- We have used PETSc since the code's inception to manage complexity, aid portability, and enable scalable performance

Outline

- Brief computing bio
- Super-condensed intro to PFLOTRAN:
  What does it solve? Why do we care?
- Leveraging PETSc to develop PFLOTRAN
  (and stay sane in the process!)

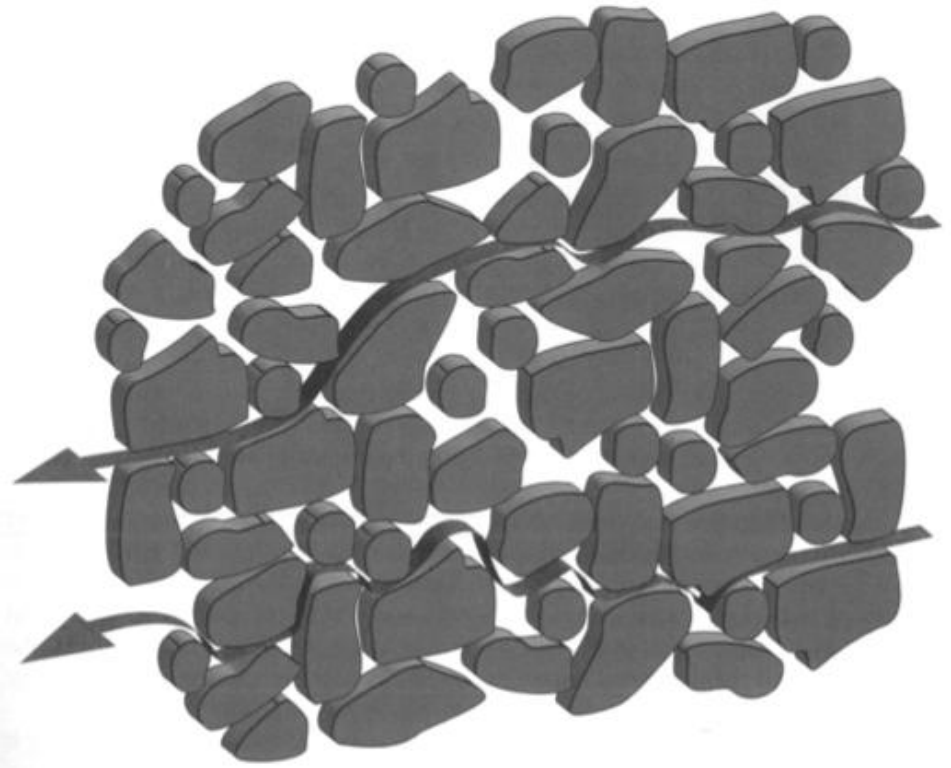
Motivating example



- At the 300 area, U(VI) plumes continue to exceed drinking standards.
- Calculations predicted cleanup by natural attenuation years ago!
- Due to long in-ground residence times, U(VI) is present in complex, microscopic inter-grain fractures, secondary grain coatings, and micro-porous aggregates. (Zachara et al., 2005).
- Models assuming constant K_d (ratio of sorbed mass to mass in solution) do not account for slow release of U(VI) from sediment grain interiors through mineral dissolution and diffusion along tortuous pathways.
- In fact, the constant K_d approach implies behavior contrary to observations!
- We must accurately incorporate millimeter scale effects over a domain measuring approximately 2000 x 1200 x 50 meters!
- PFLOTRAN addresses this challenge with: 1) Massively parallel computing, 2) Multi-continuum ("sub-grid") models, 3) Adaptive mesh refinement

Porous media flow

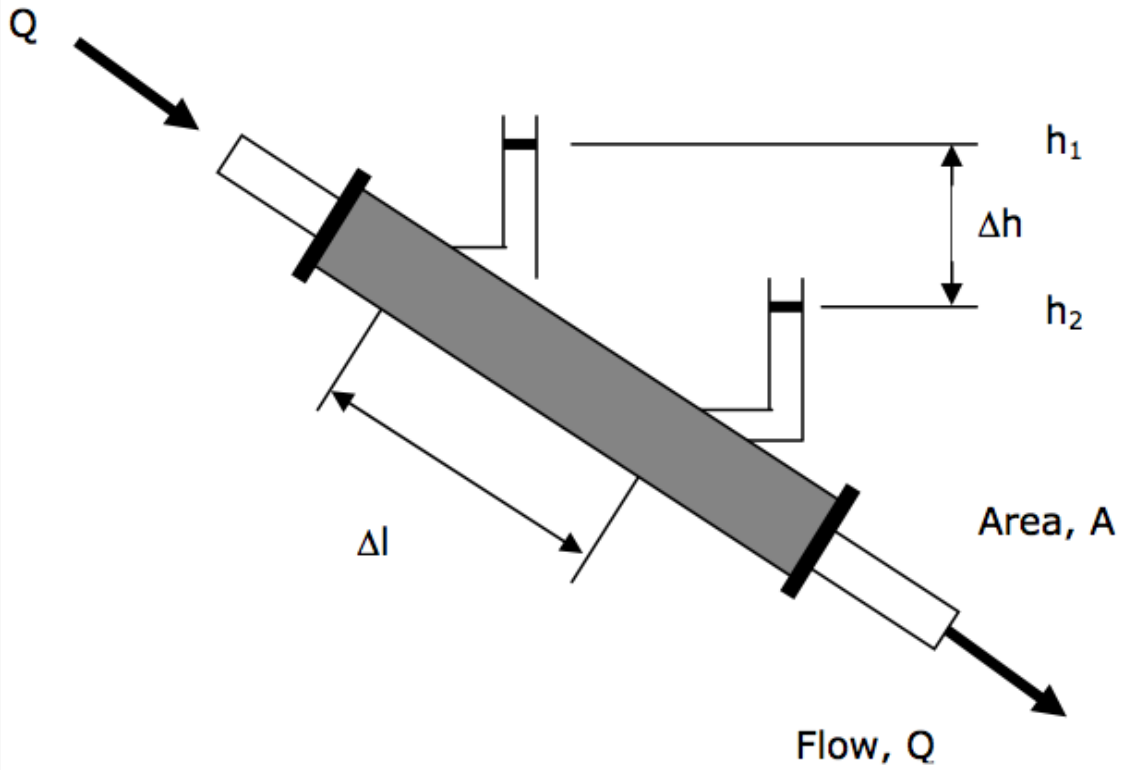
- Continuity equation (mass conservation)
- Darcy's law in place of momentum eqn.



(Figure from Hornberger et al., 1998, *Elements of Physical Hydrology*)

Porous media flow

- Continuity equation (mass conservation)
- Darcy's law in place of momentum eqn.



$$q = \frac{Q}{A} = -K\nabla h$$

(Figure from *Groundwater Hydrology* lecture notes, Prof. Charles Harvey, MIT)

PFLOTRAN governing equations

Mass Conservation: Flow Equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha \right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla (p_\alpha - W_\alpha \rho_\alpha g z) \qquad p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Energy Conservation Equation

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T \right] + \nabla \cdot \left[\sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T \right] = Q_e$$

Multicomponent Reactive Transport Equations

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \Psi_j^\alpha \right] + \nabla \cdot \left[\sum_\alpha \Omega_\alpha \right] = -\sum_m v_{jm} I_m + Q_j$$

Total Concentration

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total Solute Flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha) \Psi_j^\alpha$$

Mineral Mass Transfer Equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m$$

$$\phi + \sum_m \phi_m = 1$$

PFLOTRAN governing equations

Mass Conservation: Flow Equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha \right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla (p_\alpha - W_\alpha \rho_\alpha g z) \qquad p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Darcy's law
(homogenized momentum eq.)

Energy Conservation Equation

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T \right] + \nabla \cdot \left[\sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T \right] = Q_e$$

Multicomponent Reactive Transport Equations

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \Psi_j^\alpha \right] + \nabla \cdot \left[\sum_\alpha \Omega_\alpha \right] = -\sum_m v_{jm} I_m + Q_j$$

Total Concentration

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total Solute Flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha) \Psi_j^\alpha$$

Mineral Mass Transfer Equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m$$

$$\phi + \sum_m \phi_m = 1$$

PFLOTRAN governing equations

Mass Conservation: Flow Equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha \right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla (p_\alpha - W_\alpha \rho_\alpha g z) \qquad p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Relative permeability depends on saturation -- introduces nonlinearity

Energy Conservation Equation

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T\right] + \nabla \cdot \left[\sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T\right] = Q_e$$

Multicomponent Reactive Transport Equations

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \Psi_j^\alpha\right] + \nabla \cdot \left[\sum_\alpha \Omega_\alpha\right] = -\sum_m v_{jm} I_m + Q_j$$

Total Concentration

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total Solute Flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha)\Psi_j^\alpha$$

Nonlinear function of the concentrations of primary chemical components

Mineral Mass Transfer Equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m$$

$$\phi + \sum_m \phi_m = 1$$

Implicit time-step

- At each time step:
  - Calculate residual

$$R_n = \left(A_n^{k+1} - A_n^k\right)\frac{V_n}{\Delta t} + \sum_{n'} F_{nn'} A_{nn'} - S_n V_n$$

    —For each node, calculate accumulation term

    —For each connection, calculate flux term

$$F_{nn'} = (q\rho)_{nn'} X_{nn'} - (\varphi D\rho)_{nn'} \frac{X_n - X_{n'}}{d_n + d_{n'}}$$

    —Calculate source-sink term for appropriate connections
  - Calculate Jacobian

$$J^i_{nn'} = \frac{\partial R^i_n}{\partial x^i_{n'}}$$

    —Via finite differences

    —…or analytically (analogous to residual calculation)
  - Solve linear system

$$\sum_{n'} J^i_{nn'} \delta x^{i+1}_{n'} = -R^i_n$$

Outline

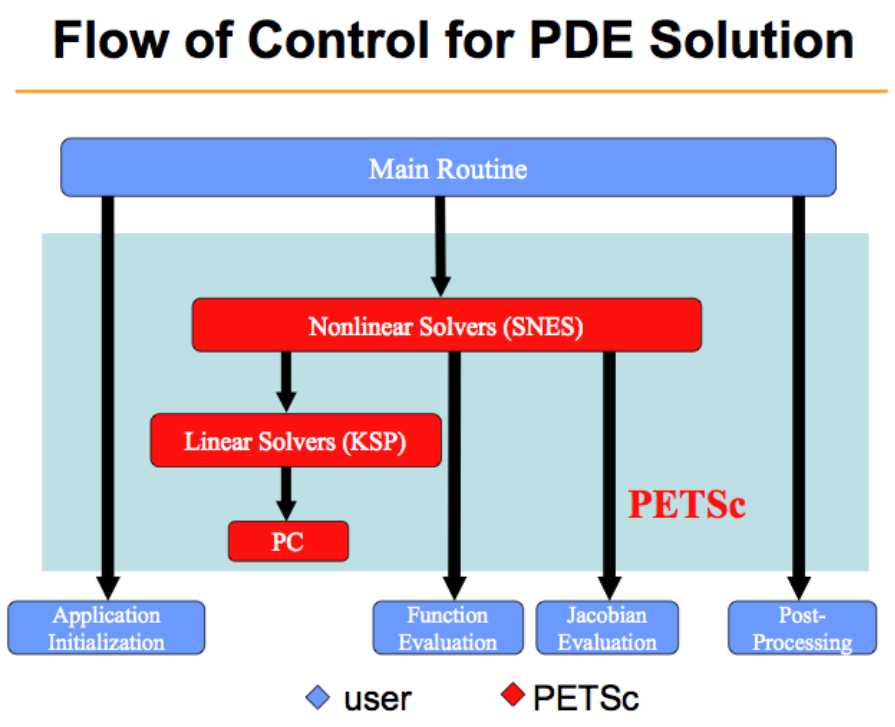
- Brief computing bio
- Super-condensed intro to PFLOTRAN:
  What does it solve? Why do we care?
- Leveraging PETSc to develop PFLOTRAN
  (and stay sane in the process!)

Building PFLOTRAN with PETSc

- PETSc has allowed us to develop a complex, scalable code in very little time:
  - Initial TRAN by Glenn Hammond for DOE CSGF practicum
  - Initial FLOW by Richard Mills for DOE CSGF practicum
  - Subsequent development of multiphase modules by Peter Lichtner and Chuan Lu during Lu's postdoc
  - Rapid improvements during SciDAC project
- PETSc is more than just a "solvers package"
- Provides a comprehensive framework:
  - Parallel mesh and associated linear algebra object management
  - Nonlinear solvers
  - Linear (iterative) solvers
  - Performance logging and debugging
  - Interfaces to many other packages

PFLOTRAN architecture

- Built on top of PETSc, which provides
  - Object-oriented management of parallel data structures,
    - Create parallel objects (e.g., matrices and vectors) over a set of processors.
    - Methods (e.g., `MatMult`())called on those objects handle parallel coordination.
  - Parallel solvers and preconditioners,
- Efficient parallel construction of Jacobians and residuals
- We provide
  - Initialization, time-stepping, equations of state, post-processing
  - Functions to form residuals (and, optionally, Jacobians) on a local patch (PETSc routines help us with patch formation by setting up scatter/gather contexts)



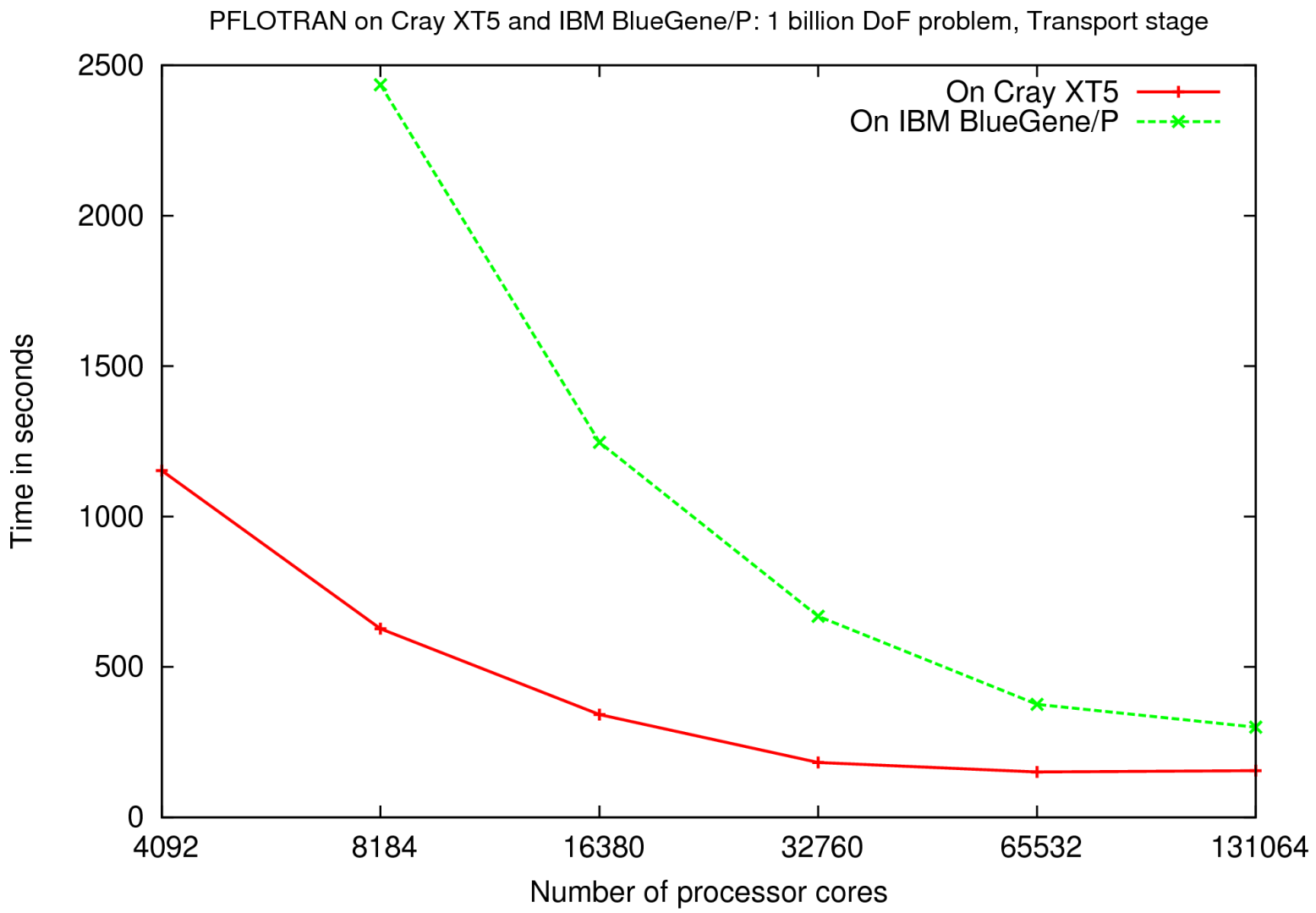
PETSc mesh management

- PETSc has greatly facilitated mesh management
- DM for general grids (new/experimental)
- DA for structured grids
- Provides support for
  - Scatter/gather of ghost points according to a stencil
  - Mapping between natural (application), global (PETSc), and local (ghosted) orderings
  - Hierarchies of grids via DMMG objects
    - —Multilevel solve via DMMGSolve().

PETSc nonliner solver framework

- Inexact Newton with various globalization strategies (line search, trust region)
- User provides SNES with
  - Residual: `PetscErrorCode (*func) (SNES snes, Vec x, Vec r, void *ctx)`
  - Jacobian: `PetscErrorCode (*func) (SNES snes, Vec x, Mat *J, Mat *M, MatStructure *flag, void *ctx)`
- Our functions expect a patch
  (local, contiguous region at single refinement level).
- Assembly of patch facilitated by PETSc
  (or SAMRAI in case of AMR)

Parallel scalability



BiCGStab improvements

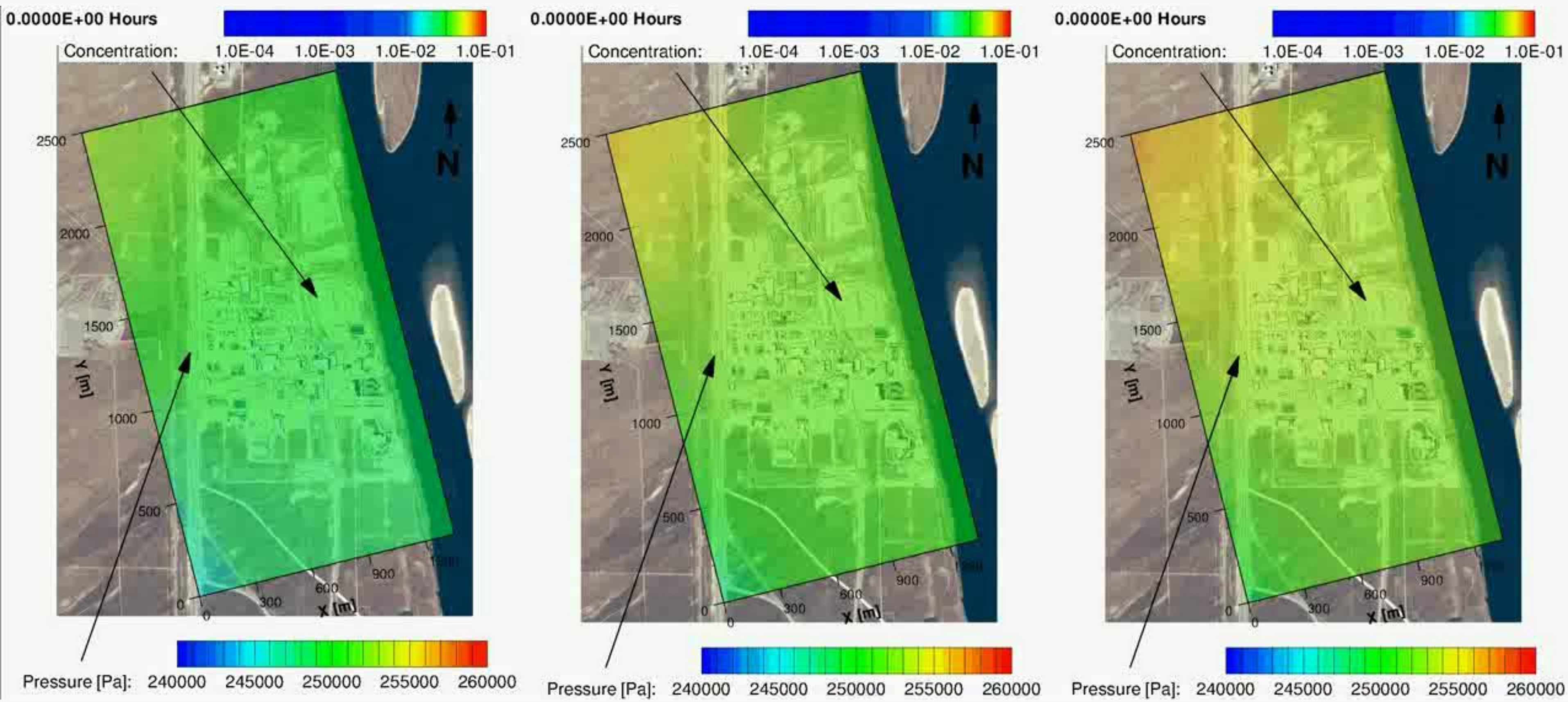
- PFLOTRAN requirements lead to improvements in PETSc that benefit all PETSc users. Recent example: BiCGStab improvements
- Cost of MPI_Allreduce() calls inside Krylov solver are big scalability barrier
- Original PETSc BiCGstab had 4 allreduces/iteration (including convergence check)
- Reworked PETSc BiCGstab has 3
- Also added "Improved" BiCGStab (IBCGS)
  - Considerably more complicated: requires transpose matrix vector product, extra vector operations
  - Only 2 MPI_Allreduce()'s per iteration required
  - By lagging residual norm calc., can wrap everything into one MPI_Allreduce(), at cost of doing one additional IBCGS iteration

16384 core,
30 time step run:

Group	BCGS	IBCGS
MPI_SYNC	196	120
MPI	150	79
User	177	200

Hypothetical U(IV) plume migration

- High performance is important, since resolution does matter!



Getting started with a code like this?

- The complexity of a multi-physics code like PFLOTRAN can seem overwhelming!
- Don't need to understand entire application: Understand components.
- Work on single component can sometimes significantly enhance overall application performance.
  - E.g., Improved matrix-vector product, new preconditioner
- Careful experimentation to choose where improvements are needed. (Never optimize code without empirical proof that it is needed!)
- Pick up other aspects of application as you go along.

Credits/Acknowledgements

- DOE CSGF for funding my and Glenn's practica
- SciDAC-II project, "Modeling Multiscale-Multiphase-Multicomponent Subsurface Reactive Flows using Advanced Computing"
- Developed by a team of subsurface scientists, applied mathematicians, and computer scientists at several institutions:
  - LANL: Peter Lichtner (PI), David Moulton
  - PNNL: Glenn Hammond
  - ORNL: Richard Mills, Bobby Philip
  - U. Utah: Chuan Lu
  - ANL: Barry Smith
  - U. Illinois: Al Valocchi
  - Additional collaborators
    - N.C. State: G. Mahinthakumar, Vamsi Sripathi (SciDAC PERI)
- Computing time provided by DOE INCITE

- For more information about PFLOTRAN and PETSc, see http://software.lanl.gov/pflotran, and http://www.mcs.anl.gov/petsc