Computational Frameworks enabling multi-scale multi-physics models

Elizabeth & Frederick White Conference
Mastering the Data Explosion in the Earth and Environmental sciences

Bill Appelbe
CEO, Victorian Partnership for Advanced Computing

Steve Quenette
Head - Computational Software Development, VPAC
Outline

• Why computational frameworks?
  - And what does that have to do with information explosions?

• Our computational framework
  - A cooks tour

• Where to from here
Why Computational Frameworks

- **Increasing Quality and Quantity of Data**
  - Allows us to develop and test ever more accurate models of the evolution of the earth
    - For vastly more model runs
  - Data has no value unless it is turned into information

- **Traditional computer modeling**
  - 2D models, fixed format input
  - Hand-written hero codes

- **We need a way to construct far more flexible computer models - frameworks**
Why Computational Frameworks (cont.)

- **Frameworks are everywhere in commercial software:** .NET and J2EEE

- **What do they BUY you in geophysics? Reuse!**
  - **Coupling**
    - Models share coupling components & techniques
  - **Data logging (model archives) components**
  - **Visualization**
    - Glucifer in our framework
  - **Grid computing**
A complex problem
Long term geodynamics

Image courtesy of Professor Louis Moresi
Codes that simulate these processes
Rephrase: …“almost simulate”…

Each of these *long term geophysics* processes fail to reliably and truthfully simulate their respective parts of the Earth…

- The Earth is complex, with (what we assume as) these processes interacting with each other

- These codes have constrained scope of signals
  - We make assumptions of behaviour at smaller/larger scales
  - We make assumptions of behaviour at process boundaries

- It is difficult to validate models
  - Can’t go build an Earth in a lab
  - It’s an inverse problem
    - We potentially have data for today,
    - The codes start from the past
Assumptions

• We make these (scale/boundary) assumptions to deal with:
  - Complexity of the model
  - Time to run the model

• As a consequence, codes have numerics optimised for its scale and structures adequate for its physics
The emerging need…

**Multi-process simulations**
- Across scales
- Across physical models

An example:
Mantle influences on basin models

- Different numerics
- Different physics
Evolving to multi-process problems

In practice, changing a code’s numerics and physics is difficult because they significantly influence low-level coding structures.

Either...

- Take existing solutions/codes, and couple them together

- Create a framework that enables interchangeability of and encourages development of the necessary numerics and physics

Can we be in a position to solve it either way?
Enabling scale crossing codes
• **With Multigrid, you effectively solve the same problem at numerous coarseness levels**
  - *Separate linear systems*
  - *We couple by residuals*
Multigrid scaling

- **The problem with (most) Finite Element codes is that they rely on building large systems of linear equations that must be solved either:**
  - Exactly / Directly \( \sim O(C*U^3) \)
  - Approximately / Iterative \( O(K*U^{<3}), K>C \)

- **Result:** finer meshes (and use more processors) are useless because it just takes too long!

- **Multigrid - scales**
  - ~linearly to unknowns
  - Very problem specific
  - Needs more memory
Adaptive Mesh Refinement

• **We don’t do it... yet.**

• **Allows one to start with a coarse, large area. The domain refines in areas where more accuracy is needed**
  - *Can turn on/off appropriate physics in different parts of the domain*

• **Benefits:**
  - Continuous domain
  - Accurate

• **Problems:**
  - *Not easy load balance in parallel - no freely available tools to help*
PICellerator

- **Framework for Lagrangian Integration schemes**

- **Presently has PIC**
  - *Material “are” integration points - coincidence*
  - *Brilliant where material histories are important*

- **Under development**
  - *Material points and integration points are not the same*
    - *E.g. Integration moves to reduce error, material advects with the material flow*
  - *Hybrids*
    - *Gaussian (standard FEM) is the cheapest (memory & time)*
Representative volumes

- *Homogenisation by representative volumes*

- *Have different physics models at two distinct scales*

- *The macro (larger) scale’s constitutive behaviour for a certain sub-domain is evaluated by a representative volume (micro)*

- *Potential huge memory and load, but conceptually clear multi-scale*
Where to from here?

- Our frameworks are public domain and open source
  - www.csd.vpac.org
  - Dozens of active users/developers
  - But they will succeed only if there is a community behind them (users and developers)

- Increasing international collaboration/users
  - Especially with the USA, the Centre for Computational Infrastructure (CIG)
  - www.geodynamics.org