Multi-physics/Multi-Scale Methods I: Methods for Multi-physics Coupled Solvers

GeoMath08 Workshop
Santa Fe, NM
15 September 2008

Daniel R. Reynolds – reynolds@smu.edu
SMU Department of Mathematics
DOE SciDAC – Towards Optimal Petascale Simulations
Introduction

My research is primarily concerned with the development of solvers for large-scale multi-physics simulations.

- I focus on implicitly-coupled methods, typically resulting in the solution of large-scale systems of nonlinear equations at each time step.
- Experience in extending legacy codes to include new physics/implicitness.
- Concerned with multi-scale problems in time, as opposed to problems with multiple spatial scales [Bangerth, Ghattas, ...].
- Application areas include
  - Fusion: hydrodynamics + electromagnetics
  - Astrophysics: hydrodynamics + radiation transport
  - Cosmology: hydrodynamics + self-gravity + radiation transport + chemical ionization kinetics
  - Materials Science: nonlinear elasticity + thermodynamics
Outline

I. Introduction

II. General Coupling Methods for Multi-physics Solvers
   - Explicit
   - Mixed Explicit & Implicit
   - Implicit

III. Case Studies
   - Magnetic Fusion
   - Cosmic Reionization
We consider time evolution of a coupled (nonlinear) modeling system,

\[ \frac{\partial}{\partial t} U = F(U) + S(U), \]

where \( U \) contains the physical variables and \( F \) and \( S \) represent processes that operate at different speeds, \( c_{\text{fast}} \) and \( c_{\text{slow}} \), respectively, e.g.

- fast chemical reactions coupled with slower global diffusion
- stiff surface gravity waves in climate models
- stiff diffusion processes coupled with non-stiff advection

While many models admit easily-separable components, allowing approximation through a variety of techniques (dropping terms, model reduction, steady-state assumptions), modern science is deriving higher-fidelity models where those simplifications become less clear (nonlinearities, complicated EOS, ...).
Explicitly-Coupled Multi-Physics Methods

Fully explicit, fixed step methods include the forward Euler method:

\[ U^{n+1} = U^n + \Delta t \left[ F(U^n) + S(U^n) \right]. \]

(+) The first approach in many multi-physics codes (simplest).

(+) High order accuracy possible through Runge-Kutta and explicit linear multistep methods.

(−) \( \Delta t \) is limited by the stability of the fastest physical process

\[ \Delta t_{CFL} \leq \frac{\Delta x^\alpha}{c_{fast}}. \]

(−) Intractable for stiff calculations, in which \( c_{fast} \gg c_{slow} \) but overall dynamics evolve at \( c_{slow} \).

(−) Cannot scale to very large problems, since \( \Delta x \to 0 \Rightarrow \Delta t \to 0 \).
Multiple Time Step Coupling Methods

Basic multiple time step methods iterate (explicitly) on the fast dynamics. Decomposing $U = (U_f, U_s)$ one may split up the physics to write:

$$U_f^{n+(j+1)/k} = U_f^{n+j/k} + \Delta t_f F\left( U_f^{n+j/k}, U_s^n \right), \quad j = 0, \ldots, k - 1$$

$$U_s^{n+1} = U_s^n + \Delta t_s S\left( U_f^{n+1}, U_s^n \right).$$

- Typically $\Delta t_s = c \Delta t_f$, where $c$ is some positive integer.

(+) Almost as simple as standard explicit methods.

(+) Not all terms need to be evaluated at each step.

(−) Splitting must be determined \textit{a priori}.

(−) Even if $\Delta t_f$ and $\Delta t_s$ adhere to stability restrictions, can result in numerical \textit{resonance instabilities} [Grubmüller 1991; Biesiadecki & Skeel 1993].
**Coarse Projective Integration**

Developed for problems with unknown microscale→macroscale constitutive laws. Laws are computed on the fly by integrating the microscale equations directly.

1. **lifting**: create appropriate micro model ICs from macro model.
2. **evolution**: explicitly evolve micro model for some duration.
3. **restriction**: project the detailed solution to macro variables.
4. **projection**: combine multiple micro solutions to take large step.

May be combined with the *gap-tooth* method for the *patch dynamics* approach. Advantage lies in applicability to models for which no PDE is (yet) available, e.g. kinetic Monte Carlo and molecular dynamics.

**Projective integrators / Gap-Tooth Method**: [Gear, Kevrekedis, Theodoropoulos, Lee, Samaey]
Mixed Explicit & Implicit Coupling Methods

Mixed methods treat each component with a different solver, e.g.

\[ U^{n+1} = U^n + \Delta t \left[ F(U^{n+1}) + S(U^n) \right]. \]

- Prototypical “operator splitting”, since separated physics enables:
  - leveraging of optimized legacy codes to a common purpose,
  - optimal solvers (e.g. FFT) may be used on individual components,

- \( F \) is sometimes split into implicit and explicit parts, typically used so that implicit piece is linear, and remainder is explicit [“linearly-implicit”].

- Requires \textit{a priori} knowledge of easily-separable stiff & nonstiff parts.

- Typically low-order accurate, though higher-order in each component is feasible [\textit{IMEX} methods by Crouzeix 1980; Ascher et al. 1997].

- Often results in numerical instabilities purely due to the splitting that may be difficult to identify/rectify [see talks by John Shadid & Don Estep].
Implicitly-Coupled Methods

Implicitly-coupled approaches treat everything with an implicit method, e.g.

\[ U^{n+1} = U^n + \Delta t \left[ F(U^{n+1}) + S(U^{n+1}) \right]. \]

(−) May result in large-scale nonlinear problems, with possibly undesirable structure (e.g. dense or non-symmetric matrices) → difficult to solve.

(+) May guarantee stability for arbitrary \( \Delta t \).

(+) Allows high accuracy solutions within and between variables.

(+) May be constructed using modern applied math toolkits, e.g. PETSc, SUNDIALS, Trilinos, that

   − have come a long way since “Numerical Recipes”,
   − allow complicated data structures,
   − enable specialized solver capabilities and interfaces to some of the most scalable and efficient solver libraries in existence.

   − (and they’re free)
Implicit Solver Structure

Implicit systems are typically solved using a variant of Newton’s method:

- Let \( g(U) \) be the vector of all equations to be solved in a time step, e.g.
  \[
g(U) = U - U^n - \Delta t [F(U) + S(U)], \quad g(U) = 0 \Rightarrow U \approx U(t^{n+1}).
\]

- Newton solvers iterate to \( \|g(U)\| \leq \varepsilon \) via:
  set \( U_0 \approx U^n \),
  (i) solve \( J(U_k)\delta U_k = -g(U_k) \) where \( J(U) = \frac{\partial}{\partial U} g(U) \)
  (ii) update \( U_{k+1} = U_k + \lambda_k \delta U_k \) for \( 0 < \lambda_k \leq 1 \)
    - \( \| \cdot \| \) is a weighted norm, balancing multi-physics components

- Linear solvers for \( J\delta U = -g \) are typically iterative:
  - Do not require matrix \( J \), only its action, \( Jv \approx [g(U + \sigma v) - g(U)]/\sigma \).
  - Amenable to very large scale problems.
  - May be combined with other solvers through preconditioner/smoother.

\textbf{Newton}: [Dembo et al. 1982; Kelley 1995; . . .] \quad \textbf{Krylov}: [Saad & Schultz 1986; Greenbaum 1997; . . .]
\textbf{Multigrid}: [Brandt 1973; Tuminaro (talk); . . .] \quad \textbf{Schwarz}: [Keyes 1989; Widlund 1989; . . .]

September 15, 2008 (10)
Preconditioning

Preconditioner notes:

- Instead of $J \delta U = -g$, you solve $(JP^{-1})(P\delta U) = -g$ or $(P^{-1}J)\delta U = -P^{-1}g$.

- Want $P \approx J$ for rapid convergence, but need $P^{-1}$ efficient.

- Inaccuracies in $P$ do not affect the accuracy of the nonlinear solution, only the convergence properties of the linear solver.

Allows domain-specific knowledge back into the solver:

- Any *a priori* knowledge can help → may only need to treat stiff parts.

- Operator-split approaches may be used to separately attack different components of $J(U)$, allowing optimal solvers on individual processes.

- Legacy codes can be repurposed as preconditioners.

- Physical approximations (lagging, model reduction, ...) may be incorporated into $P$, enabling physical intuition while retaining accuracy.

**Preconditioning Overview:** [Knoll & Keyes 2005]
Weak Scaling Limits

Assuming ideal-complexity algorithms ($O(N)$), quasi-uniform mesh size $h$, spatial dimension $d$, and textbook CFL stability criteria, we estimate the best case expected time $E$ for weak scaling of various algorithms:

- **Implicit**: $E \propto N^0 \propto P^0 \propto (1/h)^0$
- **Explicit Advection**: $E \propto N^{1/d} \propto P^{1/d} \propto (1/h)$
- **Explicit Diffusion**: $E \propto N^{2/d} \propto P^{2/d} \propto (1/h)^2$

Lower-Bound on Weak Scaling Times

Expected execution time vs unknowns (procs)

Expected execution time vs resolution

[Keyes, R. & Woodward, 2006]
Outline

I. Introduction

II. General Coupling Methods for Multi-physics Solvers
   - Explicit
   - Mixed Explicit & Implicit
   - Implicit

III. Case Studies
   - Magnetic Fusion
   - Cosmic Reionization
Case Study – Magnetic Fusion

Resistive MHD provides the simplest fluid description of fusion plasmas. The model couples the viscous, compressible Euler eqns (fluid flow – $\rho, \rho \mathbf{v}, e$) with the low-frequency Maxwell equations (electromagnetic fields – $\mathbf{B}$):

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0,$$

$$\partial_t (\rho \mathbf{v}) + \nabla \cdot \left( \rho \mathbf{v} \mathbf{v}^T - \mathbf{B} \mathbf{B}^T + \left( p + \frac{1}{2} \mathbf{B} \cdot \mathbf{B} \right) \mathbf{I} \right) = \nabla \cdot \mathbf{\tau},$$

$$\partial_t e + \nabla \cdot \left( (e + p + \frac{1}{2} \mathbf{B} \cdot \mathbf{B}) \mathbf{v} - \mathbf{B} (\mathbf{B} \cdot \mathbf{v}) \right) = \nabla \cdot \left( \mathbf{\bar{\tau}} \mathbf{v} + \kappa \nabla T \right) + \nabla \cdot \left( \eta \left( \frac{1}{2} \nabla (\mathbf{B} \cdot \mathbf{B}) - \mathbf{B} (\nabla \mathbf{B})^T \right) \right),$$

$$\partial_t \mathbf{B} + \nabla \cdot \left( \mathbf{v} \mathbf{B}^T - \mathbf{B} \mathbf{v}^T \right) = \nabla \cdot \left( \eta \nabla \mathbf{B} - \eta (\nabla \mathbf{B})^T \right).$$

Here $e = \frac{p}{\gamma - 1} + \rho \frac{\mathbf{v} \cdot \mathbf{v}}{2} + \frac{\mathbf{B} \cdot \mathbf{B}}{2}$, $T = \frac{p}{\rho \tau_{\text{gas}}}$, and $\mathbf{\bar{\tau}} = \mu \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) - \frac{2}{3} \mu (\nabla \cdot \mathbf{v}) \mathbf{I}.$

Condensing notation, we rewrite this in terms of $\mathbf{U} = (\rho, \rho \mathbf{v}, \mathbf{B}, e)^T$:

$$\partial_t \mathbf{U} = -\nabla \cdot \mathbf{F}_h(\mathbf{U}) + \nabla \cdot \mathbf{F}_d(\mathbf{U})$$
Stiffness in Resistive MHD

Resistive MHD stiffness results from fast hyperbolic waves and diffusive effects. The hyperbolic wave speeds are given by

\[\begin{align*}
\lambda_e &= \mathbf{v} \quad \text{(entropy wave)} \\
\lambda_d &= \mathbf{v} \quad \text{(magnetic-flux wave)} \\
\lambda_s &= \mathbf{v} \pm c_s \quad \text{(slow magnetosonic)} \\
\lambda_a &= \mathbf{v} \pm A_n \quad \text{(Alfvén waves)} \\
\lambda_f &= \mathbf{v} \pm c_f \quad \text{(fast magnetosonic)}
\end{align*}\]

- For MHD, \(c_s \ll A_n < c_f\), and typically \(c_f \approx 10^6\).
- Moreover, diffusive (resistive, viscous) effects induce stiffness due to the quadratic CFL condition \(\Delta t \propto \Delta x^2\).

Coupling implicitly and using a standard N-K solver [SUNDIALS], we have

\[J(U) = I + \Delta t \frac{\partial}{\partial U} [\nabla \cdot \mathbf{F}_h(U)] - \Delta t \frac{\partial}{\partial U} [\nabla \cdot \mathbf{F}_d(U)] = I + \Delta t J_h - \Delta t J_d.\]

SUNDIALS: http://www.llnl.gov/casc/sundials/
MHD Preconditioning

We employ an operator-splitting strategy for preconditioning, and set

\[ P = P_h P_d = [I + \Delta t J_h] [I - \Delta t J_d] = J + O(\Delta t^2). \]

- \( P_h \) solves for the stiff wave effects within the hyperbolic subsystem.
  - 8 coupled linear advection equations,

  \[ v + \Delta t [A \partial_x v + B \partial_y v + C \partial_z v] = b. \]

  - Split by direction and then decompose along characteristics.
  - Solved using parallel tridiagonal solvers on structured spatial grids.

- \( P_d \) solves the remaining diffusive effects.
  - 3 decoupled “heat-like” equations (1 scalar, 2 vector),

  \[ (I - \Delta t \nabla^2)w = c. \]

  - Solved using optimal geometric multigrid methods [HYPRE].

\( P_h \): [R., Samtaney & Woodward, 2008]
Combined \( P \): [R., Samtaney & Woodward, in prep.]
MHD Preconditioner Results

<table>
<thead>
<tr>
<th>Δx</th>
<th>Method</th>
<th>CPU Time</th>
<th>Nt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>Exp</td>
<td>75 s.</td>
<td>1636</td>
</tr>
<tr>
<td>0.2</td>
<td>Exp</td>
<td>768 s.</td>
<td>3247</td>
</tr>
<tr>
<td>0.1</td>
<td>Exp</td>
<td>8214 s.</td>
<td>6493</td>
</tr>
<tr>
<td>0.05</td>
<td>Exp</td>
<td>80348 s.</td>
<td>12985</td>
</tr>
<tr>
<td>0.4</td>
<td>Imp</td>
<td>47 s.</td>
<td>1144</td>
</tr>
<tr>
<td>0.2</td>
<td>Imp</td>
<td>285 s.</td>
<td>1158</td>
</tr>
<tr>
<td>0.1</td>
<td>Imp</td>
<td>1817 s.</td>
<td>1075</td>
</tr>
<tr>
<td>0.05</td>
<td>Imp</td>
<td>14203 s.</td>
<td>1473</td>
</tr>
</tbody>
</table>

Explicit/Implicit time ratios: 1.6, 2.7, 4.5, 5.7

Explicit vs implicit (no $P$) timings on Magnetic Reconnection problem (above),

Weak CPU scaling using $P_h$ preconditioner:
Kelvin Helmholtz test (corner),

Linear iterations using $P_d$ preconditioner:
diffusion-dominated test (right).

[scaling tests run on LLNL Thunder cluster]
Case Study – Cosmic Reionization

We consider the cosmological radiation-hydrodynamics-chemistry system,

\[ \partial_t \rho_b + \frac{1}{a} \mathbf{v}_b \cdot \nabla \rho_b = -\frac{1}{a} \rho_b \nabla \cdot \mathbf{v}_b, \]

\[ \partial_t \mathbf{v}_b + \frac{1}{a} (\mathbf{v}_b \cdot \nabla) \mathbf{v}_b = -\frac{\dot{a}}{a} \mathbf{v}_b - \frac{1}{a \rho_b} \nabla p - \frac{1}{a} \nabla \phi, \]

\[ \partial_t e + \frac{1}{a} \mathbf{v}_b \cdot \nabla e = -\frac{2 \dot{a}}{a} e - \frac{1}{a \rho_b} \nabla \cdot (p \mathbf{v}_b) - \frac{1}{a} \mathbf{v}_b \cdot \nabla \phi + G - \Lambda, \]

\[ \partial_t n_i + \nabla \cdot (n_i \mathbf{v}_b) = -3 \frac{\dot{a}}{a} n_i - n_i \Gamma^p_{ih} + \alpha_{i,j}^{rec} n_e n_j, \]

\[ \partial_t E + \frac{1}{a} \nabla \cdot (E \mathbf{v}_b) - \frac{1}{a^2} \nabla \cdot (D \nabla E) = -4 \frac{\dot{a}}{a} E + 4 \pi \eta - c k E, \]

\[ \nabla^2 \phi = -\frac{4 \pi G}{a} \rho_b, \]

coupling hydrodynamics \((\rho, \mathbf{v}_b, e)\), species number densities \((n_i)\), grey radiation energy density \((E)\), and gravitational potential \((\phi)\). \(G\) and \(\Lambda\) correspond to energy sources and sinks due to radiation and chemical couplings.

\[ a(t) = \frac{1}{1+z} \] provides cosmological expansion, \( x = \frac{r}{a(t)} \) is the comoving distance.
We wish to add radiation-chemistry physics to ENZO, an optimized code for AMR hydrodynamics and self-gravity. Decomposing $e = e_h + e_c$, we have

$$\partial_t (e_h + e_c) + \frac{1}{a} \mathbf{v}_b \cdot \nabla e = -\frac{2\dot{a}}{a} (e_h + e_c) - \frac{1}{a\rho_b} \nabla \cdot (p\mathbf{v}_b) - \frac{1}{a} \mathbf{v}_b \cdot \nabla \phi + G - \Lambda.$$  

In operator-split fashion, Enzo explicitly evolves one time step of the system:

$$\partial_t \rho_b + \frac{1}{a} \mathbf{v}_b \cdot \nabla \rho_b = -\frac{1}{a} \rho_b \nabla \cdot \mathbf{v}_b,$$

$$\partial_t \mathbf{v}_b + \frac{1}{a} (\mathbf{v}_b \cdot \nabla) \mathbf{v}_b = -\frac{\dot{a}}{a} \mathbf{v}_b - \frac{1}{a\rho_b} \nabla p - \frac{1}{a} \nabla \phi,$$

$$\partial_t e_h + \frac{1}{a} \mathbf{v}_b \cdot \nabla e_h = -\frac{2\dot{a}}{a} e_h - \frac{1}{a\rho_b} \nabla \cdot (p\mathbf{v}_b) - \frac{1}{a} \mathbf{v}_b \cdot \nabla \phi,$$

$$\partial_t \mathbf{n}_i + \nabla \cdot (\mathbf{n}_i \mathbf{v}_b) = 0,$$

$$\partial_t E + \frac{1}{a} \nabla \cdot (E\mathbf{v}_b) = 0,$$

$$\nabla^2 \phi = -\frac{4\pi G}{a} \rho_b.$$
We then tackle the remainder of the original system with an implicit approach,

\[
\partial_t e_c = -2\frac{\dot{a}}{a} e_c + G - \Lambda,
\]

\[
\partial_t n_i = -3\frac{\dot{a}}{a} n_i - n_i \Gamma_i^{ph} + \alpha_i^{rec} n_i n_j, \quad (i, j = 1, \ldots, n_s)
\]

\[
\partial_t E = \frac{1}{a^2} \nabla \cdot (D \nabla E) - 4\frac{\dot{a}}{a} E + 4\pi \eta - c k E.
\]

An implicit discretization of this system results in a Jacobian of the form

\[
J = I + \Delta t \begin{bmatrix}
J_{e,e} & J_{e,n} & J_{e,E} \\
J_{n,e} & J_{n,n} & J_{n,E} \\
J_{E,e} & J_{E,n} & J_{E,E}
\end{bmatrix},
\]

in which all blocks are spatially-local except for \( J_{E,E} \), which contains the term

\[- \frac{\partial}{\partial E} [\nabla \cdot (D \nabla E)].\]
Schur-Krylov-MG Linear Solver

Combining variables $x_M = (x_e, x_n)$ we write $Jx = b$ as

\[
\begin{bmatrix}
M & U \\
L & D
\end{bmatrix}
\begin{pmatrix}
x_M \\
x_E
\end{pmatrix}
= 
\begin{pmatrix}
b_M \\
b_E
\end{pmatrix}.
\]

Since $M^{-1}$ is simple to compute (block-diagonal), we use a Schur complement formulation to solve for $x$,

\[
Mx_M + Ux_E = b_M \quad \Rightarrow \quad x_M = M^{-1}(b_M - Ux_E),
\]

\[
(D - LM^{-1}U)x_E = b_E - LM^{-1}b_M.
\]

Implementation notes:

- The “heat-like” system $(D - LM^{-1}U)x_E = b_E - LM^{-1}b_M$ is solved with a Conjugate Gradient iteration.
- The CG solver is preconditioned with geometric multigrid [HYPRE].
- $x_M$ is then easily computed from $x_E$. 

September 15, 2008 (21)
Cosmology Results

Solution convergence on streaming radiation test (top),

Solution convergence on radiating shock test due to Lowrie & Edwards (corner),

Weak CPU scaling on a cosmological ionization problem due to Shapiro & Giroux (right).

[scaling tests run on NSF/NICS Kraken machine]
Conclusions

• Multi-physics couplings may be achieved in a variety of ways:
  – Decision depends on stiffness & separability of system.
  – Coupling method will dictate stability, accuracy, and scalability limitations of the overall simulation.

• Implicitly-coupled formulations promise scalability, stability and accuracy, but at a cost:
  – Large-scale nonlinear solvers become a necessity.
  – There is a well-documented approach (Newton-Krylov), available through a number of highly configurable and effective solver libraries.
  – True robustness and scalability for real-world problems may require application-specific knowledge in the preconditioner.
Acknowledgements

We gratefully acknowledge support by

- DOE SciDAC program
- NSF AAG program

and endeavor to point out contributions by collaborators

- Carol Woodward, Lawrence Livermore National Lab (CASC)
- Ravi Samtaney, Princeton Plasma Physics Lab (Theory)
- Michael Norman, U. California at San Diego (Astrophysics)
- John Hayes, Lawrence Livermore National Lab (B-Division)
- David Keyes, Columbia University (Appl. Physics & Math)
- Doug Swesty, Stony Brook University (Astrophysics)
Newton Robustness Improvements

Newton robustness w.r.t. initial guess is often accomplished with globalization methods: attempt to generate iterates within Newton convergence radius:

- **Line search**: adjusts Newton update to increase Newton radius of convergence.

- **Begin with another method**: use a slower, more globally-convergent method (e.g. Picard iteration) at first to obtain initial Newton iterate.

- **Improved initial guess**: use another method to generate a better initial Newton iterate (e.g. explicit predictor for time-dependent problems).

- **Trust region**: combines Newton direction with steepest-descent direction to ensure initial convergence.

- **Continuation**: iterative solver using Newton method for problems involving bifurcations, phase transitions or steady-state calculations.