AMR Applications in Astrophysics

John Bell

Lawrence Berkeley National Laboratory

INT Exascale Workshop
Seattle, Washington
June 27 - July 1, 2011

Overview

- Block-structured AMR approach
- Hyperbolic conservation laws
- Elliptic and parabolic PDE
- Astrophyics codes
  - CASTRO – Compressible radiation hydrodynamics
  - NYX – Cosmology
  - MAESTRO – Low Mach number model
- Code framework and parallelization
- What's next in algorithms

Developers: A. Almgren, V. Beckner, M. Day, L. Howell, M. Lijewski, A. Nonaka, M. Singer, M. Zingale

Early users: S. Woosley, A. Burrows, A. Heger, P. Nugent, S. Dong, C. Joggerst, J. Nordhaus, D. Whalen, M. White

Block-structured AMR

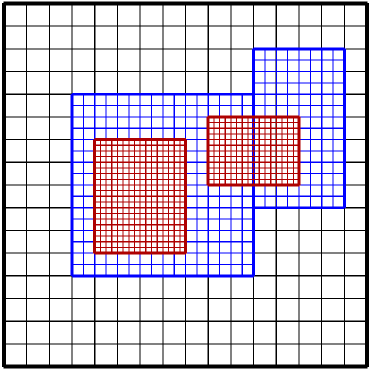
AMR – exploit varying resolution requirements in space and time

Block-structured hierarchical grids

- Amortize irregular work

Each grid patch (2D or 3D)

- Logically rectangular, structured
- Refined in space and (possibly) time by evenly dividing coarse grid cells
- Dynamically created/destroyed



2D adaptive grid hierarchy

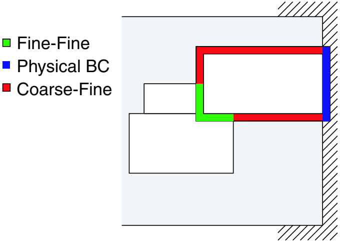
- How do we integrate PDE's on this type of grid structure
- How do we implement those algorithms
- How do we parallelize implementations

Consider a simple case – Hyperbolic Conservation Laws

AMR for conservation laws

Conservative explicit finite volume scheme

$$U^{n+1}_{i,j} = U^n_{i,j} - \frac{\Delta t}{\Delta x}(F_{i+1/2,j} - F_{i-1/2,j}) - \frac{\Delta t}{\Delta y}(G_{i,j+1/2} - G_{i,j-1/2})$$



Recursive integration with subcycling in time

- Integrate each grid patch separately
- Fill ghost cells for next finer level, interpolating in space and time from coarser grid where needed
- Integrate fine grid for r time steps

- Berger and Colella, JCP 1989
- Bell, Berger, Saltzman, Welcome, JCP 1994

Coarse and fine grids are at the same time but the overall process isn't conservative.

At c-f edges flux used on the coarse grid and average of fine grid fluxes don't agree

Reflux to make overall integration conservative – update coarse grid with difference in coarse and fine fluxes

$$\Delta x_c \Delta y_c U^c = \Delta x_c \Delta y_c U^c - \Delta t^c A^c F^c + \sum \Delta t^f A^f F^f$$

AMR Discretization Design

AMR discretization – solve on different levels separately

- Integrate on coarse grid
- Use coarse grid to supply Dirichlet data for fine grid at coarse / fine boundary
- Synchronize to correct errors that arise from advancing grids at different levels separately
  - Errors take the form of flux mismatches at the coarse/fine interface

Synchronization:

- Define what is meant by the solution on the grid hierarchy
- Identify the errors that result from solving the equations on each level of the hierarchy “independently” (motivated by subcycling in time)
- Solve correction equation(s) to “fix” the solution

Elliptic AMR

Look at 1d (degnerate) example

$$-\phi_{xx} = \rho$$

where ρ is a discrete approximation to the derivative of a δ function at the center of the domain

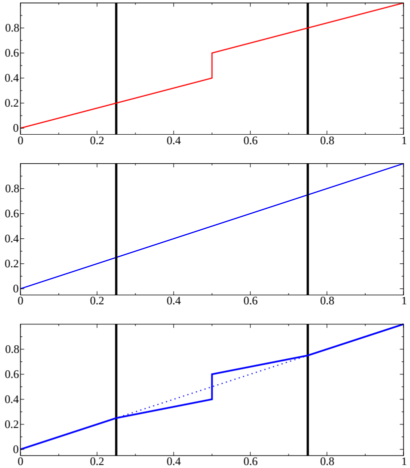
$$\rho^f_J = -\alpha \qquad \rho^f_{J+1} = \alpha$$

but $\rho^c \equiv 0$

Define a composite discretization

$$L^{c-f}\phi^{c-f} = \rho^{c-f}$$

and solve



Apply design principles above

- Solve $L^c\bar{\phi}^c = \rho^c$
- Solve $L^f\bar{\phi}^f = \rho^f$ using Dirichlet boundary conditions at $c - f$ interface

Elliptic AMR – cont'd

How do we correct the solution

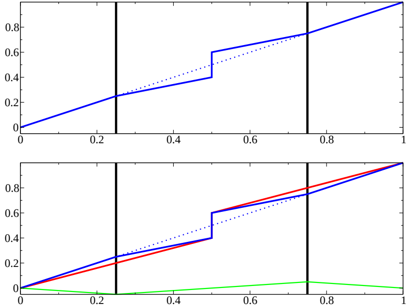
If we define $e = \phi - \bar{\phi}$ then

$$L^{c-f} e = R$$

where $R = 0$ except at $c - f$ boundary where the it is proportional to the jump in ϕ_x.

Solve for e and form $\phi = \bar{\phi} + e$

- e exactly corrects the mismatch
- Residual is localized to the $c - f$ boundary but correction is global
- The error equation is a discrete layer potential problem
- e is a discrete harmonic function on the fine grid $\rightarrow$ solve only on coarse grid and interpolate



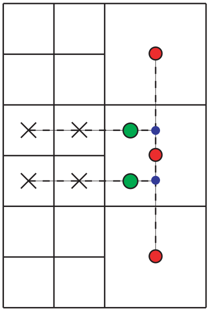
Spatial accuracy – cell-centered

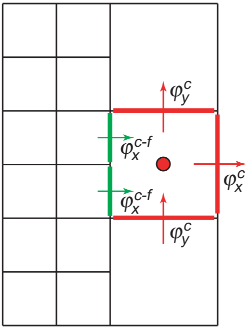
Modified equation gives

$$\psi^{comp} = \psi^{exact} + \Delta^{-1}\tau^{comp}$$

where τ is a *local* function of the solution derivatives.

Simple interpolation formulae are not sufficiently accurate for second-order operators





Parabolic discretization

Consider $u_t + \nabla \cdot F = \varepsilon \Delta u$ and the semi-implicit time-advance algorithm:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + \frac{f_{i+1/2,j}^{n+\frac{1}{2}} - f_{i-1/2,j}^{n+\frac{1}{2}}}{\Delta x} + ... = \frac{\varepsilon}{2}\left((\Delta^h u^{n+1})_i + (\Delta^h u^n)_i\right)$$

The difference e^{n+1} between the exact composite solution u^{n+1} and the solution $\overline{u}^{n+1}$ found by advancing each level separately satisfies

$$(I - \frac{\varepsilon \Delta t^c}{2}\Delta^h)\ e^{n+1} = \frac{\Delta t^c}{\Delta x_c}(\delta F + \delta D)$$

where, for example, on a constant x edge

$$\begin{aligned}
\Delta t^c\ \delta f &= -\Delta t^c\ \overline{f}_{J-1/2} + \sum \Delta t^f f_{j+1/2} \\
\Delta t^c\ \delta D &= \frac{\varepsilon \Delta t^c}{2}(\overline{u}_{x,J-1/2}^{c,n} + \overline{u}_{x,J-1/2}^{c,n+1}) \\
&- \sum \frac{\varepsilon \Delta t^f}{2}(\overline{u}_x^{c-f,n} + \overline{u}_x^{c-f,n+1})
\end{aligned}$$

AMR algorithms for astrophysics

Basic integration paradigm works for hyperbolic, elliptic and parabolic PDEs (also works for DO radiation)

Synchronization equations match the structure of the process being corrected.

Combine these elements to make a number of astrophysics applications codes

For multiphysics problems, key issue is keeping tracking of different aspects of synchronization and performing them in the right order

Same set of tools can be used for a variety of applications

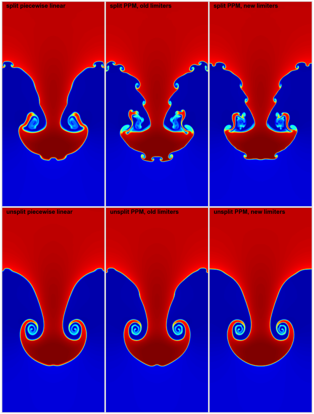
- Self-gravitational rad / hydro
- Cosmology
- Low Mach number model

SN Ia

CASTRO

Compressible hydrodynamics

- Unsplit, general EOS integrators
- Self-gravity
  - Monopole approximation
  - General Poisson solver
- Level integration and synchronization designed for second-order accuracy of overall algorithm



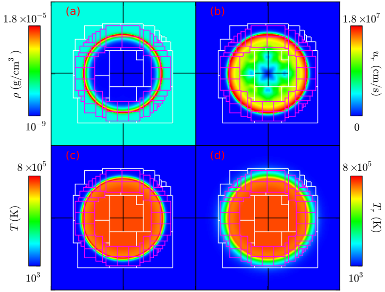
Applications

- SN Ia deflagration studies (flame model) – Woosley, Dong, Ma
- Core collapse – Burrows, Nordhaus, Rantsiou
- Pair instability supernovae – Heger, Chen
- Nucleosynthesis – Joggerst, Heger, Woosley, Whalen

Radiation

Castro Radiation

- Flux-limited diffusion
- Mixed-frame formulation
- Coupled gas pressure and radiation pressure in hydro step
- Gray or multigroup photons
- Neutrino model in progress (coupling to Y_e)

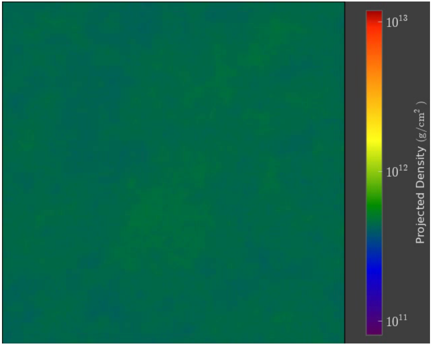


Three dimensional simulation of radiative blast wave with AMR

NYX

Cosmology

- Dark matter
  - Collisionless particles
  - Couple to baryonic matter through gravity
- Hydrodynamics equation in comoving coordinates
- Self gravity for baryonic and dark matter



Simulation of Santa Barbara cluster test problem

Animation courtesy of Casey Stark

Low Mach number models

Compressible models work well for modeling supernovae explosion, cosmology, etc., but . . .

Another class of phenomena are characterized by low Mach number flows where $U << c$

- Convection leading up to igntion in an SNIa
- Type I XRB
- Convection in main sequence stars.

Construct specialized low Mach number models that exploit the separation of scales between fluid motion and acoustic wave propagation

Model needs to include a number of effects

- Background stratification
- Nonideal equation of state
- Reactions and heat release
- Overall expansion of the star

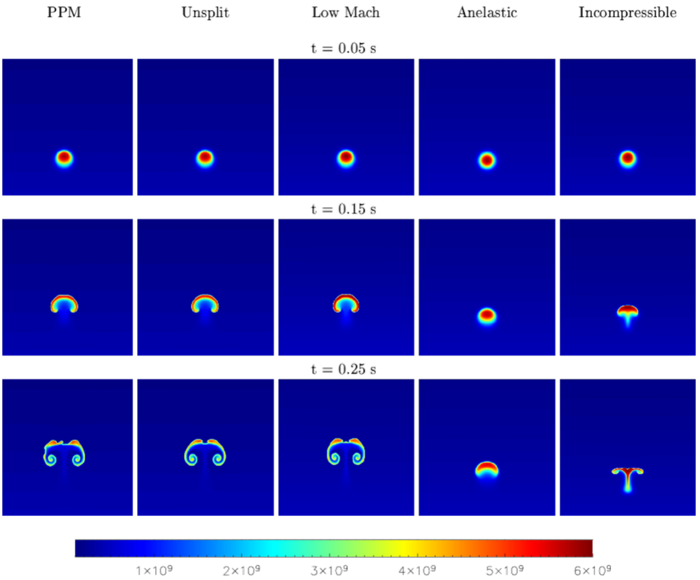
A hierarchy of possible models

We want to eliminate acoustic waves (so we don't have to track them) but make as few assumptions as possible about the magnitude of density and temperature variations.

Possible models for convective motion:

- Boussinesq: simplest model - allows heating-induced buoyancy in a constant density background (constant p_0, ρ_0, T_0)
- Variable-ρ incompressible: finite amplitude density variation but incompressible
- Anelastic: allows small variations in temperature and density from a stratified background state $(p_0(r), \rho_0(r), T_0(r))$
- Low Mach number : large variations in temperature and density in a time-varying stratified background state $(p_0(r), \rho_0(r), T_0(r))$

Buoyant bubble rise



Low Mach Number Approach

Asymptotic expansion in the Mach number, $M = |U|/c$, leads to a decomposition of the pressure into thermodynamic and dynamic components:

$$p(\mathbf{x}, t) = p_0(r, t) + \pi(\mathbf{x}, t)$$

where $\pi/p_0 = O(M^2)$.

- p_0 affects only the thermodynamics; π affects only the local dynamics,
- Physically: acoustic equilibration is instantaneous; sound waves are “filtered” out
- Mathematically: resulting equation set is no longer strictly hyperbolic; a constraint equation is added to the evolution equations
- Computationally: time step is dictated by fluid velocity, not sound speed.

Low Mach Number Model

$$
\begin{aligned}
\frac{\partial(\rho X_k)}{\partial t} &= -\nabla \cdot (U\rho X_k) + \rho\dot{\omega}_k \ , \\
\frac{\partial(\rho h)}{\partial t} &= -\nabla \cdot (U\rho h) + \frac{Dp_0}{Dt} - \sum_k \rho q_k \dot{\omega}_k + \rho H_{ext} \ , \\
\frac{\partial U}{\partial t} &= -U \cdot \nabla U - \frac{1}{\rho}\nabla\pi - \frac{(\rho - \rho_0)}{\rho} g\mathbf{e}_r \ , \\
\nabla \cdot (\beta_0 U) &= \beta_0 \left(S - \frac{1}{\bar{\Gamma} p_0} \frac{\partial p_0}{\partial t} \right)
\end{aligned}
$$

where

$$
S = -\sigma \sum_k \xi_k \dot{\omega}_k + \frac{1}{\rho p_\rho} \sum_k p_{X_k} \dot{\omega}_k + \sigma H
$$

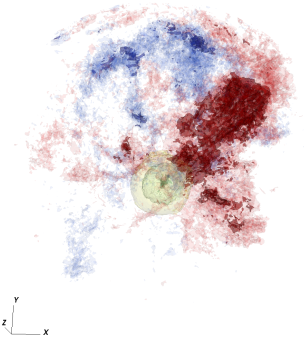
Cannot assume fixed background for net large-scale heating. We need evolution equations for p_0, ρ_0, etc.

Use average heating to evolve base state. Remaining dynamics evolves perturbations

$$
\frac{\partial p_0}{\partial t} = -w_0 \frac{\partial p_0}{\partial r} \quad \text{where} \quad \mathrm{w}_0(\mathrm{r},\mathrm{t}) = \int_{\mathrm{r}_0}^{\mathrm{r}} \overline{\mathrm{S}}(\mathrm{r}',\mathrm{t})\, \mathrm{dr}'
$$

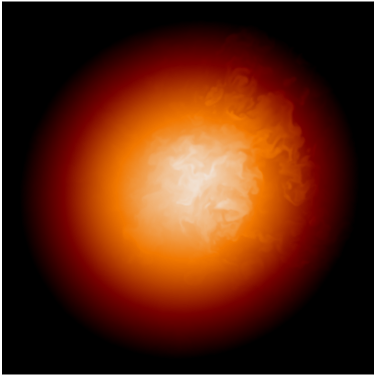
Self gravity introduces additional complexity

White dwarf convection



Convective flow pattern on inner 1000 km of star

- Red / blue is outward / inward radial velocity
- Yellow / green shows burning rate



Two dimensional slices of temperature a few minutes before ignition

Distribution of ignition

What we would like to know the the distribution of the ignition site and the structure of the turbulence

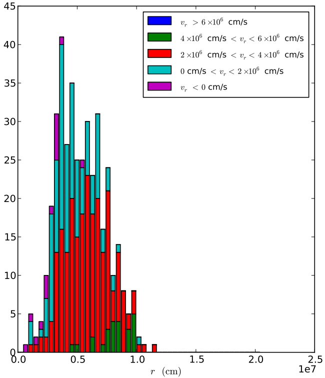
Monitor peak temperature and radius during simulation

Filter data

Bin data to form histogram

Assume that hot spot locations are “almost” ignitions

Map data into compressible code to model explosion



Implementation

One wants to implement these types of algorithms within a software framework that supports the development of block-structured AMR algorithms

- Represent dynamically changing hierarchical solution
- Manage error estimation and regridding operations
- Orchestrate multistep algorithms and synchronization
- Support for iterative methods for implicit algorithms

There are a number of frameworks that support implementation of these types of algorithms

We use BoxLib

- Data structures
- Operations on those data structures
- Model for parallelization

Data Structures

Index space

- Box : a rectangular region in index space
- BoxArray : a union of Boxes at a level

Real data at a level

- FAB: FORTRAN-compatible data on a single box
  - Data on a patch
  - These patches are quite large – thousands of points
- MultiFAB: FORTRAN-compatible data on a union of rectangles
  - Data at a level
- FluxRegister: FORTRAN-compatible data on the border of a union of rectangles
  - Data for synchronization

Parallel Data Distribution

AMR hierarchy represented by a BoxArray and MultiFAB at each level

- Each processor contains the full BoxArray.
  - Simplifies data-communications: send-and-forget
- Data itself is distributed among processors; different resolutions are distributed independently, separately load-balanced.
- Owner computes rule on FAB data.
- Issues for efficient implementation
  - Dynamic load balancing
  - Efficient manipulation of metadata
  - Optimizing communication patterns
  - Fast linear solvers

Metadata, communications and solvers

Index space operations are naively $O(n^2)$

- Each box needs to know its neighbors
- Bin BoxArray spatially
- Limit searches to boxes in neighboring bins

Communication

- Every MultiFAB with the same BoxArray has the same distribution
- Each processor caches list of its grids' nearest neighbors and their processors
- Each processor caches list of coarse grids and their processors used to supply boundary conditions
- Messages are ganged: no more than one message is ever exchanged between processors in an operation

Solvers

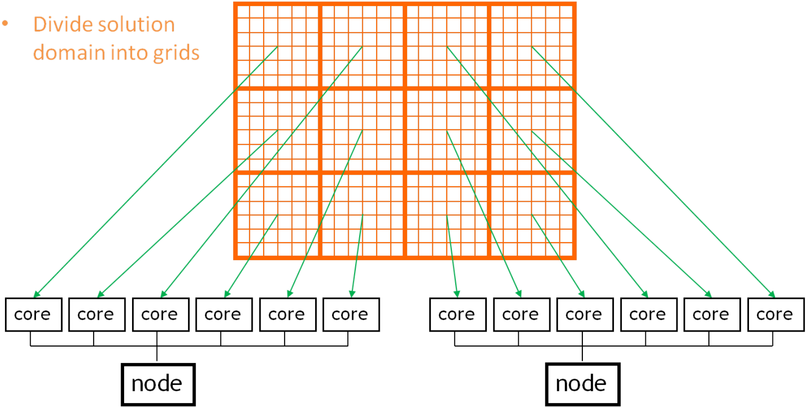
- Semi-structured solvers
- Current approaches based on multigrid
  - As problem is coarsened, floating point to communication gets small
  - Communication avoiding algorithms
  - Consolidate data at coarse levels of multigrid

CCSE Berkeley Lab

Multi-core architectures

Pure MPI approach

- Divide solution domain into grids

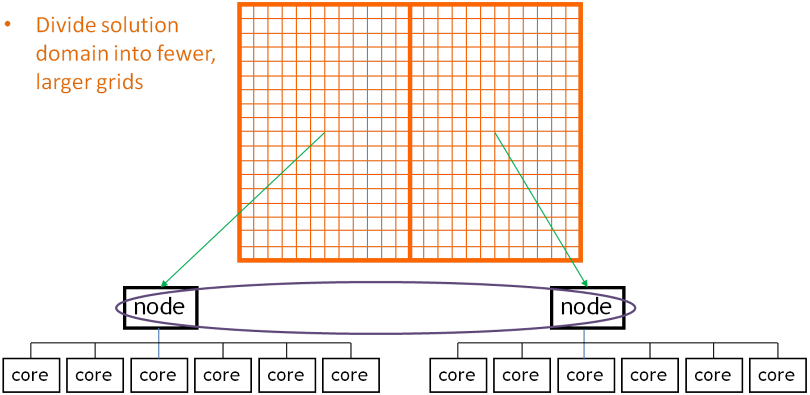


Each grid is assigned to a core

Cores communicate using MPI

Hybrid model

- Divide solution domain into fewer, larger grids



Each grid is assigned to a node

OpenMP used to spawn threads so that cores within a node work on the grids simultaneously

Nodes communicate using MPI

MPI versus Hybrid model

Advantages of hybrid model

- Fewer MPI processes lead to reduced communication time
- Less memory for storing ghost cell information
- Reduced work from larger grids – surface to volume effect

Disadvantages of hybrid model

- Spawning threads is expensive – makes performance worse for small core counts
- Can't hide parallelization from physics modules

With hybrid model, we have been able to scale multiphysics applications to O(100K) processors

CCSE BERKELEY LAB

Exascale architectures

Power and cost constraints $\rightarrow$ a significant shift in architectural design for next generation systems

- Higher concurrency in low-power many-core, possibly heterogeneous nodes
- Much lower memory per core
- Performance based on memory access patterns and data movement, not FLOPS
- High synchronization costs
- High failure rates for components
- Reduction in relative I/O system performance
  - Need to integrate analysis with simulation
  - Makes simulation look much more like physical experiments
- Current programming models are inadequate for the task
  - MPI reasonable for coarse-grained parallelism but at fine-grained level we write basically serial and add bandaids (OpenMP) to express parallelism
  - We express codes in terms of FLOPS and let the compiler figure out the data movement
  - Non-uniform memory access is already an issue but programmers can't easily control data layout
- Rethink discretization methods for multiphysics applications
  - More concurrency
  - More locality with reduced synchronization
  - Less memory / FLOP
  - Analysis of algorithms currently based on a performance $\equiv$ FLOPS paradigm – can we analyze algorithms in terms of a more realistic performance model

What's next

Where to we need to go with algorithms?

Characteristics of current generation algorithms

- Second-order in space and time
- Strang split coupling between processes
- Lots of synchronization
- AMR metadata bottlenecks
  - Communication-rich multigrid
  - AMR synchronization points and bottlenecks

For exascale we would like things that are

- Higher-order in space and time
  - Implicitly requires more sophisticated coupling
  - Better way to deal with constrained systems
- Distributed AMR metadata
- Communication avoiding iterative methods
- Refactor AMR to reduce synchronization and bottlenecks

Improving the coupling

Potential options to couple advection, diffusion and reaction

- Weak (lagged) coupling of operators
  - Boris and Oran
  - Iterated operator splitting methods
  - Approximate factorization
  - Difficult to make higher-order
- Fully implicit MOL approaches
  - BDF or IRK integration methodology
  - Fully coupled nonlinear solve
- IMEX methods
  - Treat one scale explicit, rest implicit; two-scale model
  - Fully coupled nonlinear solve
- Spectral deferred corrections
  - Introduced by Dutt, Greengard and Rokhlin for ODE
  - Minion – SISDC
  - Bourlioux, Layton, Minion – MISDC
  - Layton, Minion – Conservative MISDC

Spectral Deferred Corrections

Basic idea (Dutt, Greengard, Rokhlin): write solution of ODE, $u_t = f(t, u)$ on $[a, b]$ as

$$u(t) = u_a + \int_a^t f(\tau, u)\, d\tau \quad .$$

If we have an approximate solution $\hat{u}(t)$, we can define the residual

$$E(t, \hat{u}) = u_a + \int_a^t f(\tau, \hat{u})\, d\tau - \hat{u}(t) \quad .$$

Then, the error $\delta(t) \equiv u(t) - \hat{u}$ satisfies

$$\delta(t) = u(t) - u\hat{(t)} = (u_a + \int_a^t f(\tau, u)\, d\tau) - (u_a + \int_a^t f(\tau, \hat{u})\, d\tau - E(t, \hat{u}))$$

MISDC for advection/diffusion/reaction (Minion *et al.*):

- Treat each term separately using a simple approach
- Explicit advection, implicit diffusion, implicit reactions
- Use different time steps for each process
- Iterate SDC correction equation
    - Interpolating polynomial couples the processes

SDC for ADR

Generalized SDC framework

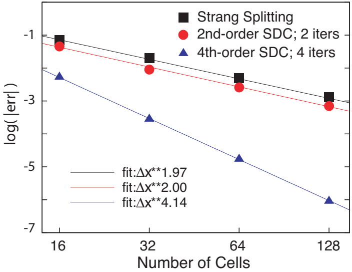
- Use different representations for each physical process
- Reuse existing components of the methodology
- Integrate reactions using VODE – Think of VODE as ”exact”

Consider a simple model problem

$$(u_i)_t + a(u_i)_x = D(u_i)_{xx} + R_i \quad i = 1, 4$$

where

$$R_1 = -k_1 u_1 u_4 - k_2 u_1, \quad R_2 = k_1 u_1 u_4$$
$$R_3 = k_2 u_1, \quad R_4 = -k_1 u_1 u_4.$$



Spectral Deferred Corrections

Develop a general framework for coupling processes in multiphysics applications

- Treat individual processes uses representation appropriate for that process
- Solver simpler subproblems but iterate to couple processes
- Higher-order in time is a change in quadrature rule
- Potential to evolve processes simultaneously

Need detailed understanding of SDC properties

- Accuracy and robustness of the overall discretization
- Convergence properties of the SDC iterations

and how these properties are related to

- Properties of processes
- Choice of quadrature rules
- Initialization and correction algorithms
- Potential acceleration strategies

Some work in this area by Minion and collaborators for ODE / DAE

This lays the ground work for higher-order temporal discretization

SDC, AMR and Parallel

SDC and Parallel

- Simultaneous evaluation of different processes with best available approximation to other processes
- Initial iterations at lower resolution / lower fidelity
- SDC - Parallel in time (Minion, CAMCOS 2011)

Standard block-structured AMR integration advances levels sequentially from coarsest to finest

Use SDC ideas to restructure core AMR time-step strategy

- No need to complete iteration at a given level before starting the next level
- Use initial iterations on coarse grid to compute initial fine grid solutions
- This enables integration of different levels in the AMR hierarchy simultaneously
- Requires substantive changes to the underlying infrastructure to support efficient implementation

All of these ideas will reduce serial performance but they expose more concurrency and have potential for improving parallel performance

Summary

Shift in architectures as we move to the exascale

- FLOPS don't matter (much)
- Memory and data movement are the key
- Focus is on changes to the node-level architecture – issues are likely broader than just exascale
- Slow I/O

For astrophysics

- Need higher-order in space and time with AMR but avoid difficult nonlinear systems
- Ideally, use formulation that respects the scales in the problem
- SDC framework for coupling processes
- New programming model