Combustion Simulation on Next Generation Architectures

J. Bell

Lawrence Berkeley National Laboratory

Los Alamos National Laboratory
Los Alamos, NM
May 17, 2011

Introduction

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

Exascale architectures – likely characteristics

- Much 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
- Reduction in relative I/O system performance
- High failure rates for components

Proposed Combustion Exascale Co-Design Center

- PI: Jackie Chen
- Redesign all aspects of combustion simulation to enable effective use of exascale architectures
- Large interdisciplinary team: math, cs, data science
- Centered around three vertically integrated co-design efforts in PDE, UQ and analysis with cross-cutting CS efforts

CCSE Berkeley Lab

Preliminary observations

Need to integrate analysis with simulation

- Combustion simulations are data rich
- Writing data to disk for subsequent analysis is currently near infeasibility
- Several potential strategies for how to do this
- Makes simulation look much more like physical experiments in terms of methodology

Current programming models are inadequate for the task

- MPI provides reasonable approach 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

Need to rethink how we approach PDE discretization methods for multiphysics applications

- More concurrency
- More locality with reduced synchronization
- Less memory / FLOP
- Analysis of algorithms has typically been based on a performance $\equiv$ FLOPS paradigm – can we analyze algorithms in terms of a more realistic performance model

Review current state of the art and discuss some possible future directions

CCSE Berkeley Lab

Lean Premixed Turbulent Combustion

Rod-stabilized V-flame

4-jet Low-swirl burner (LSB)

Slot burner

- Potential for efficient, low-emission power systems
- Design issues because of flame instabilities
- Want to simulate complex fuels at realistic pressure and turbulence conditions

CCSE Berkeley Lab

Compressible Navier Stokes

Gas phase combustion – mixture model for diffusion

$$\begin{aligned}
\textbf{Mass} \quad & \rho_t + \nabla \cdot \rho U = 0 \\
\textbf{Momentum} \quad & (\rho U)_t + \nabla \cdot (\rho UU + p) = \rho \vec{g} + \nabla \cdot \tau \\
\textbf{Energy} \quad & (\rho E)_t + \nabla \cdot (\rho UE + pU) = \nabla \cdot \kappa \nabla T + \nabla \cdot \tau U \\
& \qquad + \sum_m \nabla \cdot (\rho h_m D_m \nabla Y_m) \\
\textbf{Species} \quad & (\rho Y_m)_t + \nabla \cdot (\rho U Y_m) = \nabla \cdot (\rho D_m \nabla Y_m) + \dot{\omega}_m
\end{aligned}$$

Augmented with

- Thermodynamics
- Reaction kinetics
- Transport coefficients

Need to preserve chemical and transport fidelity

CCSE Berkeley Lab

Relevant Scales

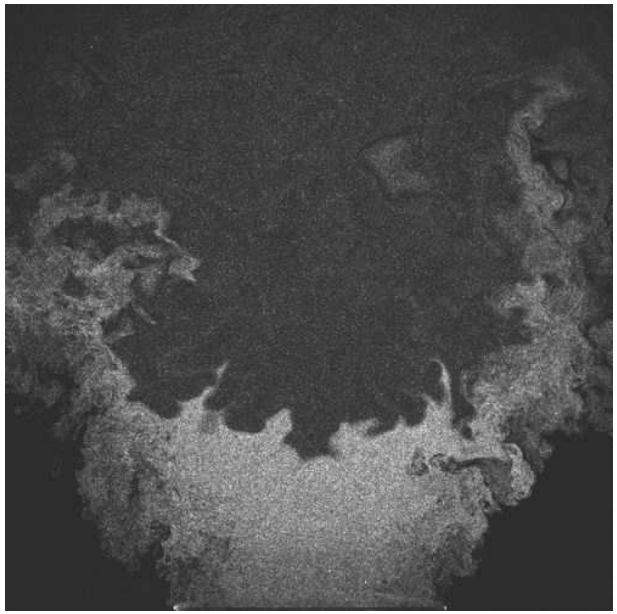
Spatial Scales

- Domain: $\approx$ 10 cm
- Flame thickness: $\delta_T \approx 1$ mm
- Integral scale: $\ell_t \approx 2 - 6$ mm

Temporal Scales

- Flame speed $O(10^2)$ cm/s
- Mean Flow: $O(10^3)$ cm/s
- Acoustic Speed: $O(10^5)$ cm/s

Fast chemical time scales but energy release coupling chemistry to fluid is on slower time scales



Mie Scattering Image

CCSE Berkeley Lab

Time step strategies

Scaling is paramount: Low communication, explicit discretizations, balanced work load – let the machine do the work

- Generic mathematical model
- Define spatial discretization – structured, unstructured, adaptive
- Identify time step based on stability requirements
- Integrate with explicit ODE algorithm
- Range of time scales determines performance
- Basis for S3D – explicit DNS

Coupling is paramount: Fully implicit, method of lines, iterative algorithms – preconditioners do the work

- Generic mathematical model
- Define spatial discretization – structured, unstructured, adaptive
- Identify time step based on accuracy requirements
- Integrate with implicit ODE algorithm (1 explicit scale)
- Efficiency of solver/preconditioner determines performance

Custom approach based on exploiting mathematical structure of the system

CCSE Berkeley Lab

Mathematical formulation

Exploit natural separation of scales between fluid motion and acoustic wave propagation

Low Mach number model, $M = U/c \ll 1$ (Rehm & Baum 1978, Majda & Sethian 1985)

Start with the compressible Navier-Stokes equations for multicomponent reacting flow, and expand in the Mach number, $M = U/c$.

Asymptotic analysis shows that:

$$p(\vec{x}, t) = p_0(t) + \pi(\vec{x}, t) \quad \text{where} \quad \pi/p_0 \sim \mathcal{O}(M^2)$$

- p_0 does not affect local dynamics, π does not affect thermodynamics
- For open containers p_0 is constant
- Pressure field is instanteously equilibrated – removed acoustic wave propagation

Low Mach number equations

Momentum $$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) = -\nabla \pi + \nabla \cdot \tau$$

Species $$\frac{\partial(\rho Y_m)}{\partial t} + \nabla \cdot (\rho U Y_m) = \nabla \cdot (\rho D_m \nabla Y_m) + \dot{\omega}_m$$

Mass $$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0$$

Energy $$\frac{\partial \rho h}{\partial t} + \nabla \cdot \left(\rho h \vec{U}\right) = \nabla \cdot (\lambda \nabla T) + \sum_m \nabla \cdot (\rho h_m D_m \nabla Y_m)$$

Equation of state $p_0 = \rho \mathcal{R} T \sum_m \frac{Y_m}{W_m}$

System contains four evolution equations for U, Y_m, ρ, h, with a constraint given by the EOS.

Low Mach number system can be advanced at fluid time scale instead of acoustic time scale but . . .

We need effective integration techniques for this more complex formulation

CCSE Berkeley Lab

Constraint for reacting flows

Low Mach number system is a system of PDE's evolving subject to a constraint; differential algebraic equation (DAE) with index 3

Differentiate constraint to reduce index

$$\nabla \cdot U = \frac{1}{\rho c_p T}\left(\nabla \cdot (\lambda \nabla T) + \sum_m \rho D_m \nabla Y_m \cdot \nabla h_m\right) + \frac{1}{\rho}\sum_m \frac{W}{W_m}\nabla(D_m \rho \nabla Y_m) + \frac{1}{\rho}\sum_m \left(\frac{W}{W_m} - \frac{h_m(T)}{c_p T}\right)\dot{\omega}_m$$

Generalized projection method framework

- Finite amplitude density variation
- Inhomogeneous constraint
- Requires solution of variable coefficient, self-adjoint elliptic PDE

Low Mach number numerics

Fractional step scheme

- Advance velocity and thermodynamic variables
  - Advection
  - Diffusion
  - Stiff reactions
- Project solution back onto constraint – variable coefficient elliptic PDE, multigrid

Stiff kinetics relative to fluid dynamical time scales

$$\frac{\partial(\rho Y_m)}{\partial t} + \nabla \cdot (\rho U Y_m) = \nabla \cdot (\rho D_m \nabla Y_m) + \dot{\omega}_m$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho U h) = \nabla \cdot (\lambda \nabla T) + \sum_m \nabla \cdot (\rho h_m D_m \nabla Y_m)$$

Operator split approach

- Chemistry $\Rightarrow \Delta t/2$
- Advection – Diffusion $\Rightarrow \Delta t$
- Chemistry $\Rightarrow \Delta t/2$

Coupled to block structured AMR

CCSE Berkeley Lab

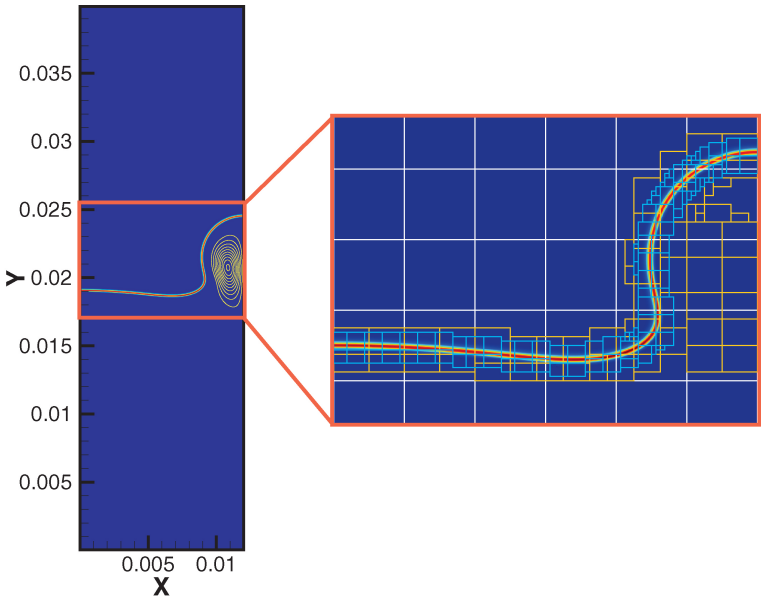
AMR

AMR – exploit varying resolution requirements in space and time
Block-structured hierarchical grids

- Amortize irregular work

Each grid patch (2D or 3D)

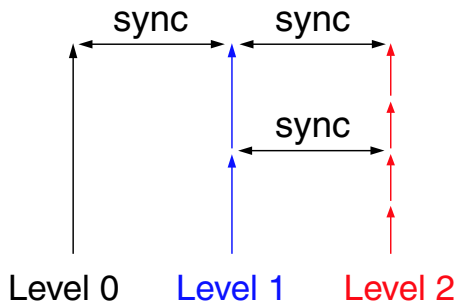
- Logically structured, rectangular
- Refined in space and time by evenly dividing coarse grid cells
- Dynamically created/destroyed



2D adaptive grid hierarchy

Subcycling:

- Advance level ℓ, then
  - Advance level $\ell + 1$
    level ℓ supplies boundary data
  - Synchronize levels ℓ and $\ell + 1$



CCSE Berkeley Lab

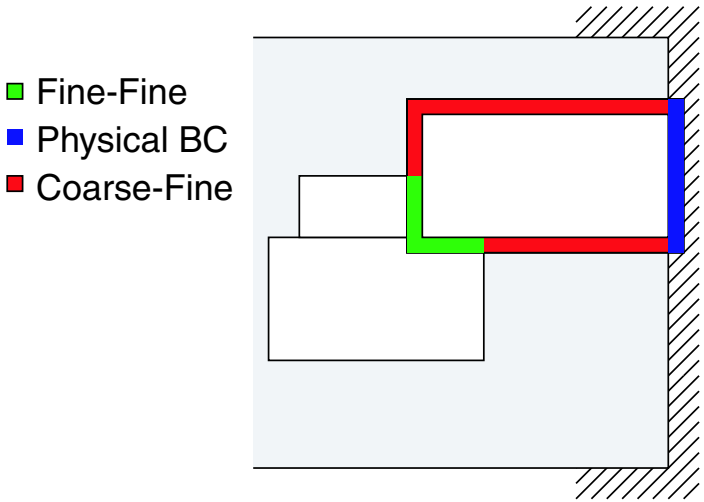
AMR Synchronization

Coarse grid supplies Dirichlet data as boundary conditions for the fine grids.

Errors take the form of flux mismatches at the coarse/fine interface.

Design Principles:

- 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”.
- Solve correction equation(s) to “fix” the solution.
- Correction equations match the structure of the process they are correcting.



Preserves properties of single-grid algorithm

Parallelization framework

Algorithms implemented in BoxLib software framework

- Reusable software framework for grid-based methods
- Support for block-structured AMR
- Lagrangian particles – for diagnostic or other purposes
- Multigrid for elliptic / parabolic solves
  - Native
  - Links to other software
- Modular EOS and reaction networks – “plug ’n play”

Parallelization strategy

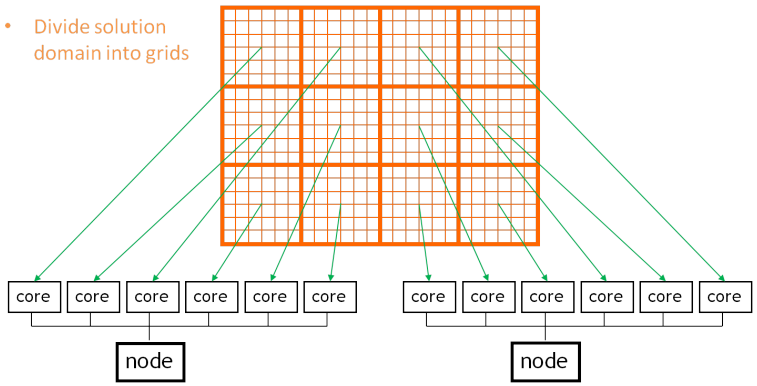
- Coarse-grained parallelization based on domain decomposition
- Distribute grid patches in AMR to processors using MPI
- Dynamic load balancing for heterogeneous physics

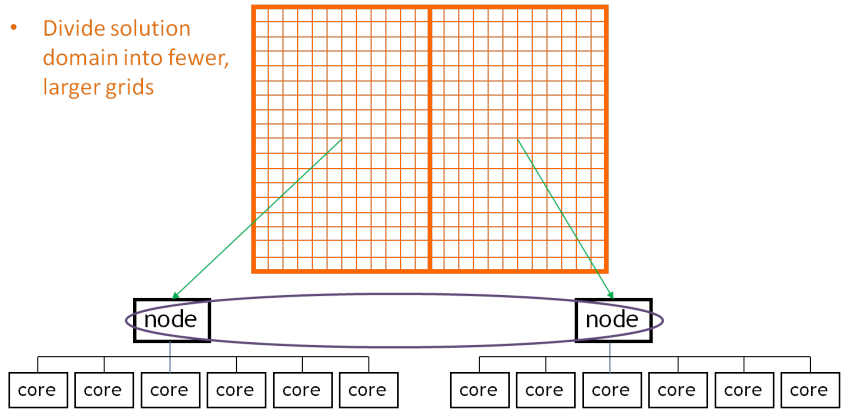
CCSE Berkeley Lab

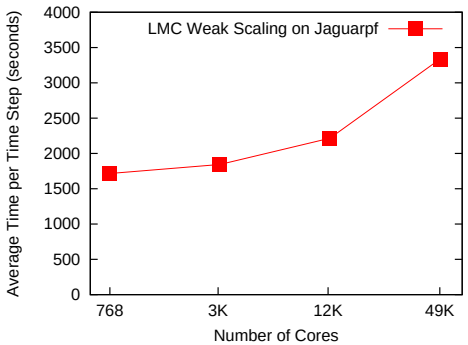
Parallelization

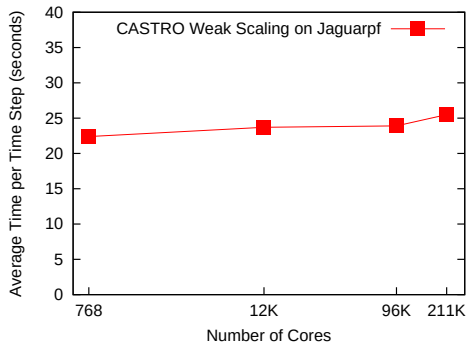
Hybrid approach combining MPI with OpenMP

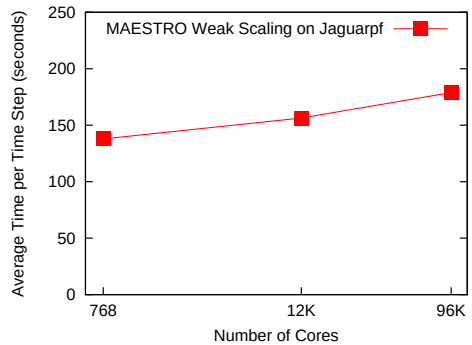
- MPI to communicate between nodes
- Fine-grained parallelism using OpenMP threads within nodes







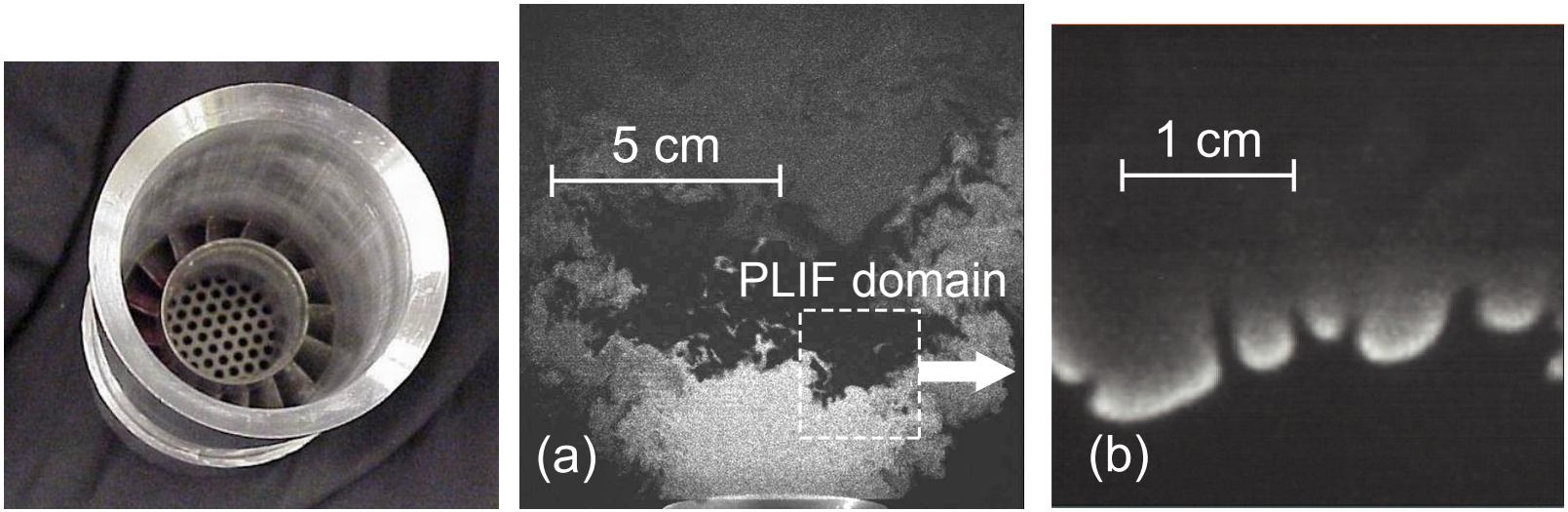




Almgren *et al.*, *SciDAC*, 2010

CCSE BERKELEY LAB

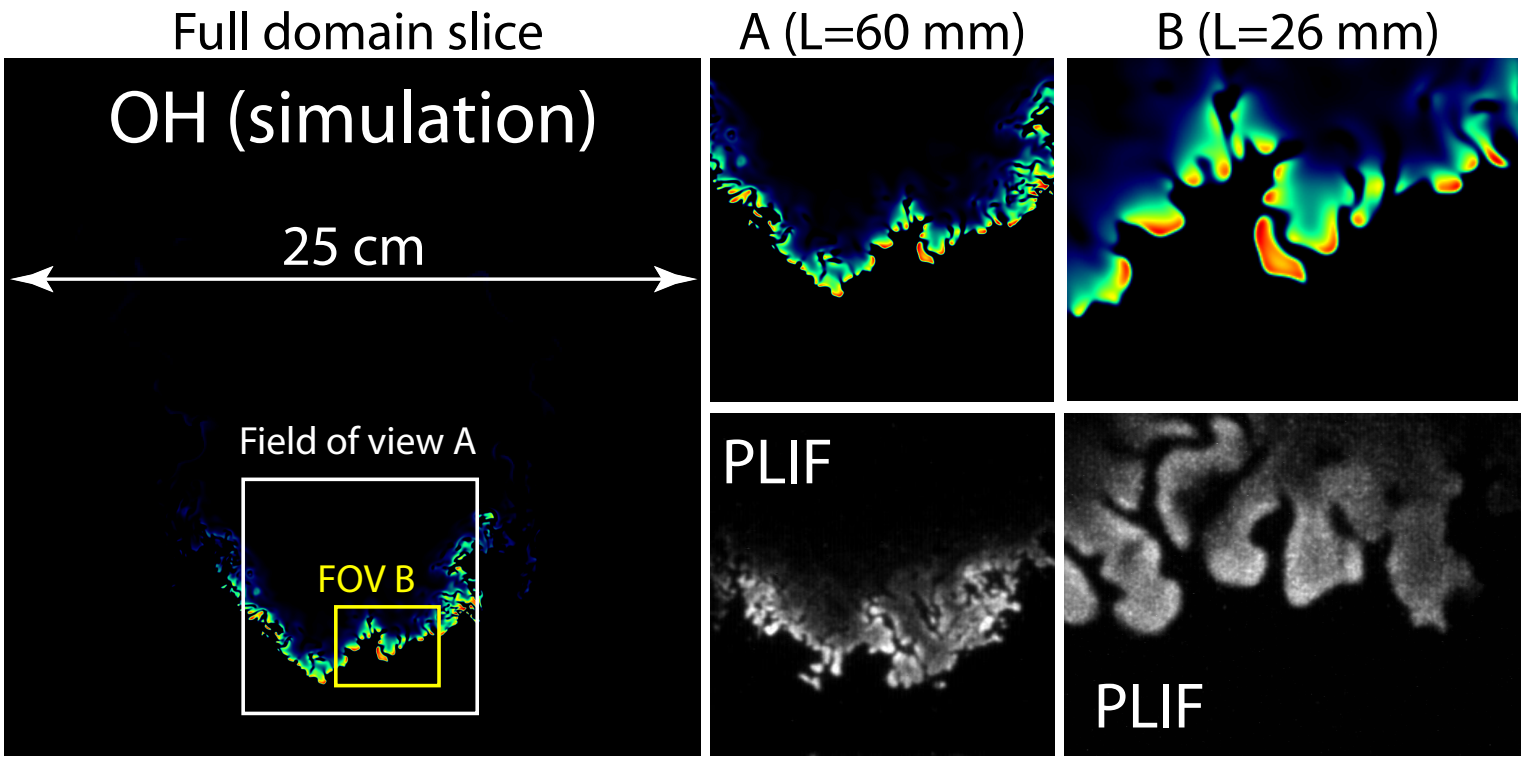
Hydrogen combustion



- OH PLIF shows gaps in the flame
- How do these flames burn?
- Are existing engineering models applicable?
- Can standard flame analysis techniques be used to analyze structure?

CCSE Berkeley Lab

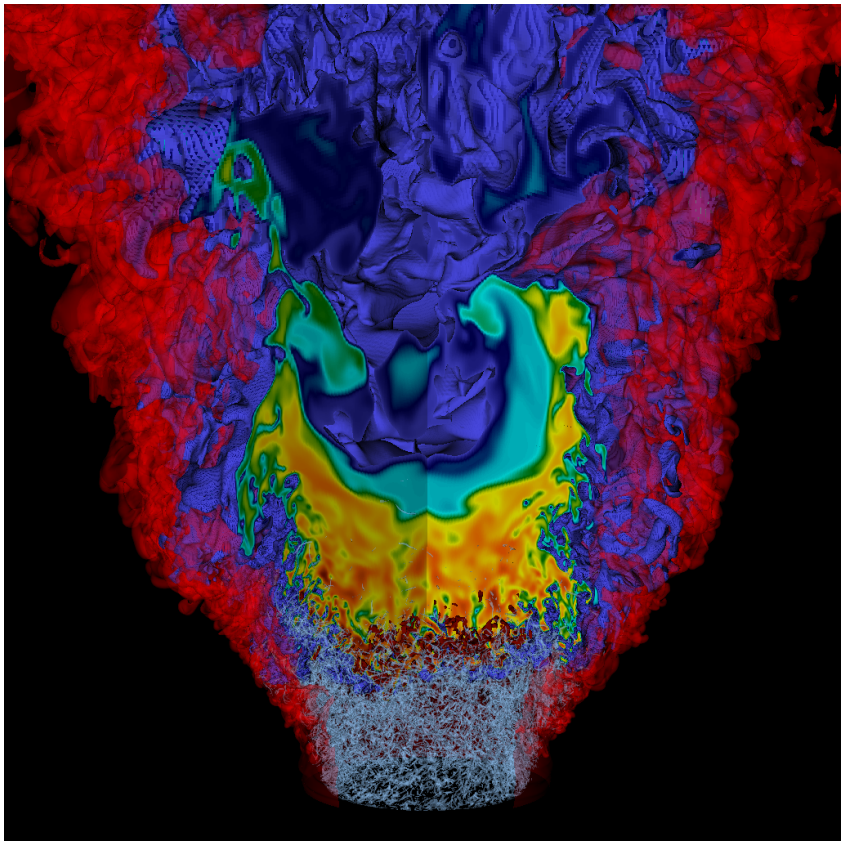
Comparisons with OH-PLIF from LSB experiment



- Comparison of OH slice with typical OH-PLIF measurements, global and fine scales
- Instantaneous large- and fine-scale flame shape/extremely similar, in terms of shape and variability

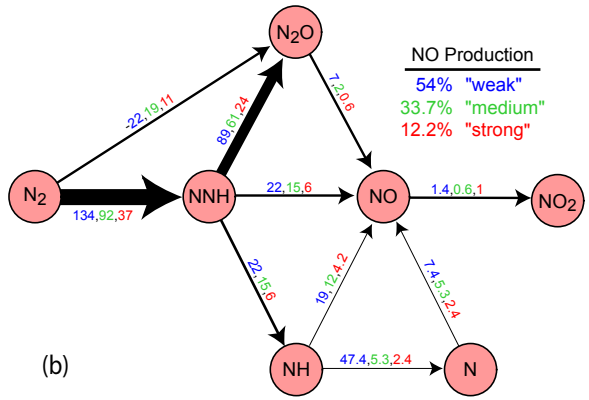
CCSE Berkeley Lab

Hydrogen low swirl burner simulation



Flame surface from simulation of low swirl burner with lean hydrogen colored by local fuel consumption rate

Current simulations model formation of NO_x emissions



Roughly 50B unknowns

What's wrong with the above

Characteristics of the algorithm

- Second-order in space and time
- Strang split coupling between processes
- Drift off of the constraint
- 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

Programming model issues

CCSE Berkeley Lab

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

CCSE Berkeley Lab

Spectral Deferred Corrections

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

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

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

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

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

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

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

CCSE Berkeley Lab

SDC for ADR

Generalized SDC framework

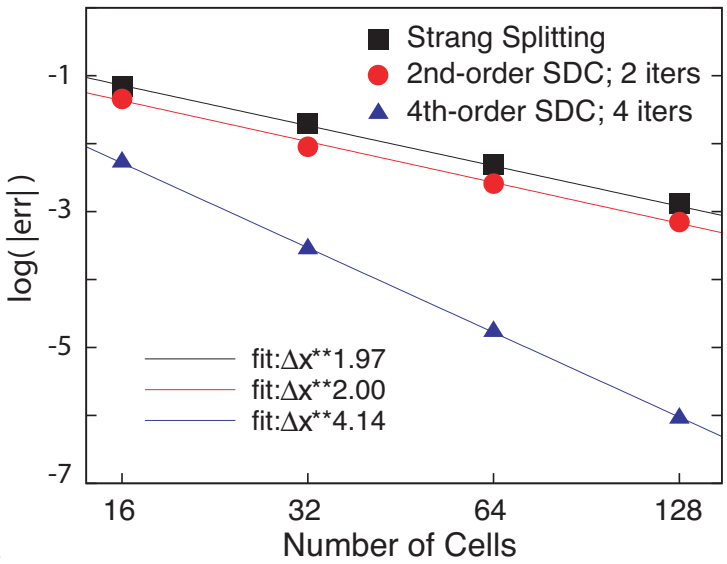
- Use different representations for each physical process
- Reuse existing components of the methodology
- Integrate reactions using VODE – Think of VODE as ”exact”

Consider a simple model problem

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

where

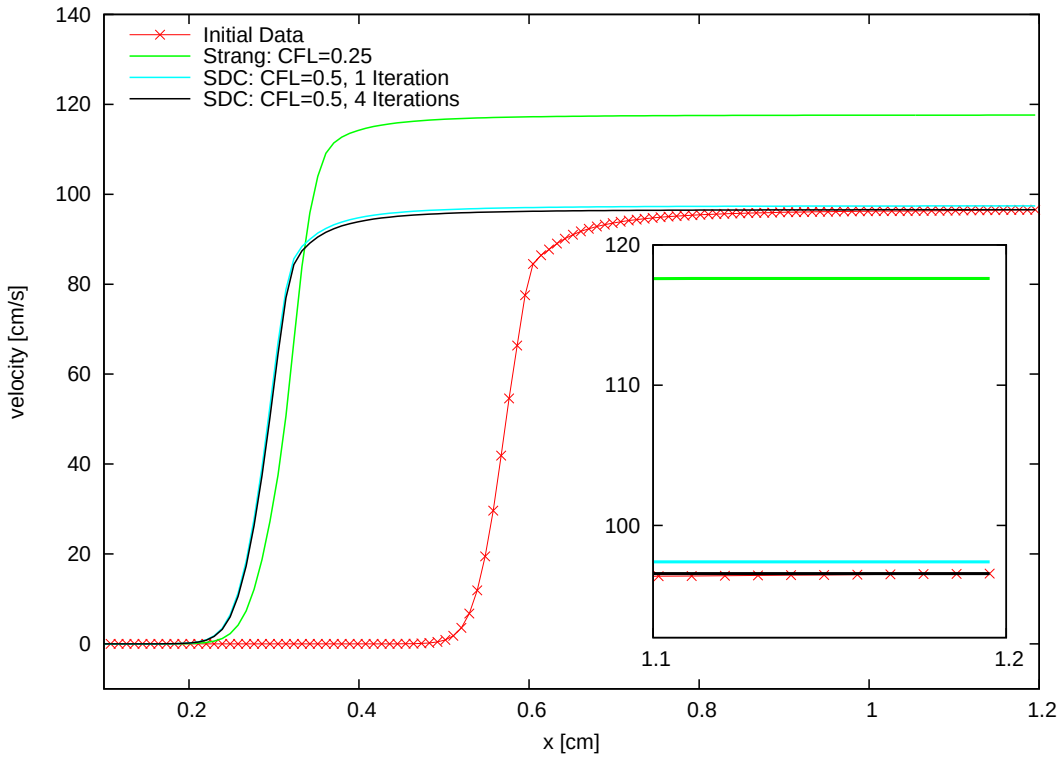
$$\begin{aligned} R_1 &= -k_1 u_1 u_4 - k_2 u_1, \quad R_2 = k_1 u_1 u_4 \\ R_3 &= k_2 u_1, \quad R_4 = -k_1 u_1 u_4. \end{aligned}$$



Spectral Deferred Corrections

For realistic flames the behavior of the algorithm is sensitive to the details

- Choice of representation of the processes
- Initial approximation
- Algorithm for the correction equation



Methane flame with detailed chemistry

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

CCSE Berkeley Lab

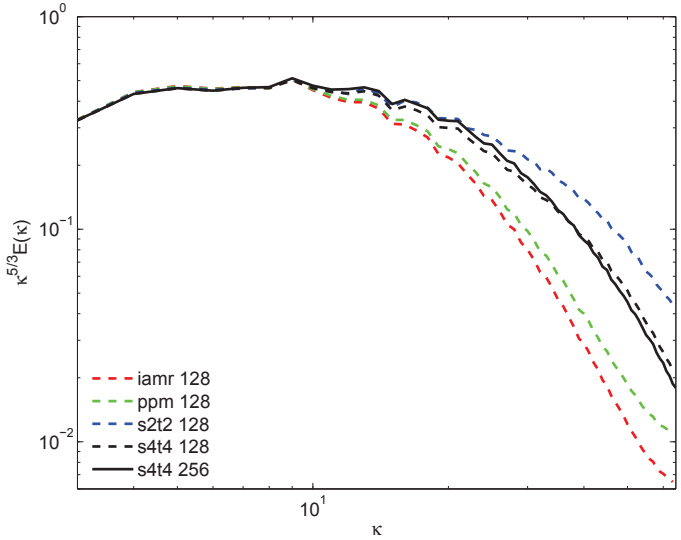
SDC for low Mach number flows

Fourth order projection algorithm for viscous compressible flow

- Impulse variable formulation
- Fourth-order finite volume spatial discretization
- SDC temporal discretization
- Algorithm converges at design rates for smooth flow

Key issue is whether higher-order algorithm reduces resolution requirements for turbulent flows.

Consider forced isotropic turbulence – examine minimum resolution needed to resolve flow at a given Reynolds number



Compensated energy spectra for forced turbulence

Results obey expected Reynolds number scaling provided *Re* is sufficiently large

Almgren *et al.*, submitted

CCSE Berkeley Lab

SDC – Low Mach number

Auxiliary variable formulation for more general low Mach number flow

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) &= 0 \\ \frac{\partial (\rho U)}{\partial t} + \nabla \cdot (\rho U U) &= -\nabla \pi + \nabla \cdot \tau \\ \nabla \cdot U &= S \end{aligned}$$

Define $m = \rho U_d + \rho \nabla \xi$ where $\nabla \cdot U_d = 0$ and $\Delta \xi = S$

Then we can write an auxiliary variable form of the momentum equation as

$$m_t^* + \nabla(\rho(U_d + \nabla\xi)(U_d + \nabla\xi)) + \nabla q = \nabla \cdot \tau(U_d + \nabla\xi)$$

where $m^* = m + \nabla\phi$ so that ϕ and q are related to p by

$$\nabla\phi_t + \nabla q - \nabla \cdot \tau(\frac{1}{\rho}\nabla\phi) = \nabla p$$

This provides the framework needed to extend the higher-order incompressible flow solver to more general low Mach number flows . . . almost

- A number of issues related to boundary conditions
- Need higher-accuracy representation of constraint (linearized EOS to obtain divergence constraint)

CCSE Berkeley Lab

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

CCSE BERKELEY LAB

Programming model

Current programming model is inadequate

- Can't express data layout and data motion
- Much potential fine-grained parallelism is lost

Elements of new programming model

- Abstract machine model
  - Able to predict actual performance (with autotuning)
  - Humanly comprehensible
- Language to express algorithms that maps onto abstract machine model – DSL
  - Express necessary control over algorithm
  - Reasonable implementation of algorithms
  - Use of autotuning to understand compiler optimizations
  - Obtain performance near prediction with autotuning
- PDE algorithm analysis in terms of abstract machine model

CCSE Berkeley Lab

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 combustion

- Uniform grid, explicit seems problematic (4/3 - law)
- 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

CCSE Berkeley Lab