A Deferred Correction Strategy for Advection-Diffusion-Reaction Systems with Complex Chemistry

Andy Nonaka
Lawrence Berkeley National Laboratory
October 18, 2011

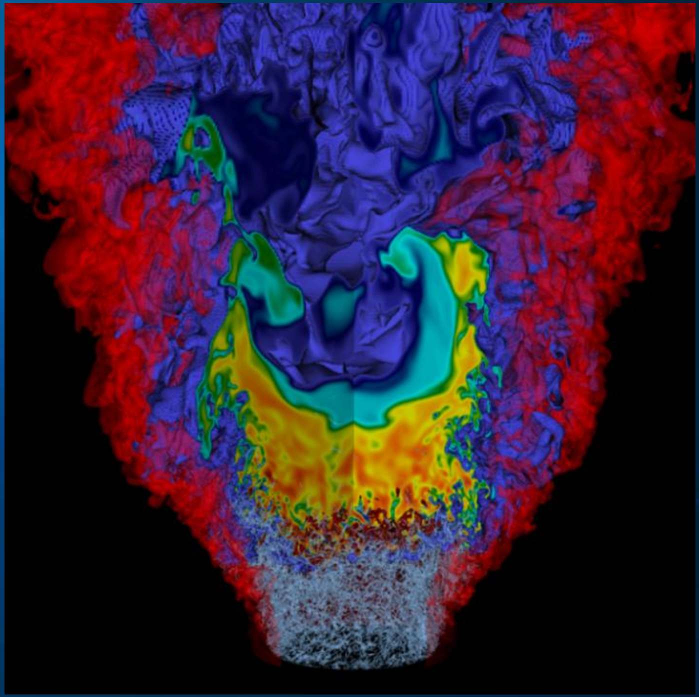
Collaborators

- Lawrence Berkeley National Laboratory
  - John Bell
  - Marc Day
  - Candace Gilet
  - Ann Almgren

- UNC Chapel Hill
  - Michael Minion

Overview

- Algorithms for multiphysics applications containing a broad range of time scales
- Example is high fidelity simulations of low Mach number reacting flow
  - Detailed chemistry and transport (10s – 100s of species)
  - Advection, diffusion, and reaction processes
  - Reactions very stiff compared to advection
  - Low Mach number formulation exploits separation of scales, allowing for advective CFL time step condition.
  - BoxLib software framework for parallel, block-structured AMR



Lean hydrogen flame on a laboratory-scale low-swirl burner

Model Equations

- Velocity update from conservation of momentum

stress tensor

$$(\rho U)_t = -\nabla \cdot (\rho U U) - \nabla \pi + \nabla \cdot \tau$$

density

velocity

perturbational pressure

- Divergence constraint found by differentiating equation of state along particle paths

$$\nabla \cdot U = S$$

incorporates local compressibility effects due to reactions and diffusion

- Thermodynamic update from conservation of mass and energy

species "m" mass fraction

species "m" production rate

$$\begin{aligned}(\rho Y_m)_t &= -\nabla \cdot (U \rho Y_m) + \nabla \cdot \rho \mathcal{D}_m \nabla Y_m + \dot{\omega}_m \\ (\rho h)_t &= -\nabla \cdot (U \rho h) + \nabla \cdot \frac{\lambda}{c_p} \nabla h + \sum_m \nabla \cdot h_m \left(\boxed{\rho \mathcal{D}_m} - \boxed{\frac{\lambda}{c_p}} \right) \nabla Y_m\end{aligned}$$

enthalpy

transport coefficients

Model Equations

- Transport coefficients highly nonlinear
- Diffusion and (particularly) reactions stiff compared to advection - computationally expensive
  - Diffusion requires implicit treatment on advective time scales
    - Crank-Nicolson – requires linear solves, multigrid
  - Reactions use VODE software package
    - Variable-coefficient, Adams-Moulton / Backward Differentiation Formula (BDF) stiff ODE solver
    - Adaptively subdivide time step to respect stiffest chemistry; requires many Jacobian, equation of state, and reaction rate evaluations

$$\begin{aligned}
(\rho Y_m)_t &= \color{green}\boxed{-\nabla\cdot(U\rho Y_m)} \color{white}+ \color{yellow}\boxed{\nabla\cdot\rho\mathcal{D}_m\nabla Y_m} \color{white}+ \color{red}\boxed{\dot{\omega}_m} \\
(\rho h)_t &= \color{green}\boxed{-\nabla\cdot(U\rho h)} \color{white}+ \color{yellow}\boxed{\nabla\cdot\frac{\lambda}{c_p}\nabla h+\sum_m\nabla\cdot h_m\left(\rho\mathcal{D}_m-\frac{\lambda}{c_p}\right)\nabla Y_m}
\end{aligned}$$

$$\phi_t = {\color{green}A} + {\color{yellow}D} + {\color{red}R}$$

Current Approach

- Fractional step scheme
  - Advance solution and project velocity back onto constraint – variable coefficient elliptic PDE, multigrid
- Strang splitting for thermodynamic variable advance
  - Reactions: Δt/2
  - Advection-Diffusion: Δt (Godunov integration for advection)
  - Reactions: Δt/2
- Strang splitting is robust…
  - … but inherently second-order
  - … and can suffer from large splitting error

Alternative Strategies

- Implicit method of lines (MOL) and/or implicit-explicit (IMEX) Runge-Kutta strategies
  - Computationally expensive – very large linear systems coupling all thermodynamic variables in every cell
- Spectral deferred corrections (SDC)
  - MOL strategy introduced by Dutt, Greengard, and Rokhlin for ODEs
  - Semi-implicit (SISDC) and multi-implicit (MISDC) extensions for PDEs with multiple time scales (Minion, Layton, Bourlioux)
  - Advantages of SDC methods
    - Each process discretized individually using well-suited numerical method
    - Able to dramatically reduce splitting error
    - Extendable to higher-order temporal integration

Spectral Deferred Corrections (SDC)

- The basic idea is take an approximate (or lower order) solution and iteratively solve correction equations to improve the solution
  - The correction equations use low-order numerical discretizations and are relatively easy to solve
- To derive correction equations, first consider an ODE: $\phi_t = F\left(t, \phi(t)\right), \quad \phi(t^n) \equiv \phi^n$
- The solution can be represented as an integral: $\phi(t) = \phi^n + \int_{t^n}^{t} F(\phi)\; d\tau$
- Given an approximate, or guess at the solution, $\phi^{(k)}(t)$, we define the residual:

$$E(t, \phi^{(k)}) = \phi^n + \int_{t^n}^{t} F(\phi^{(k)})\; d\tau - \phi^{(k)}(t)$$

- Now define the error, $\delta^{(k)}(t) = \phi(t) - \phi^{(k)}(t)$

Spectral Deferred Corrections

- Combining the residual and error equations gives

$$\delta^{(k)}(t) = \int_{t^n}^{t} \left[F(\phi^{(k)} + \delta^{(k)}) - F(\phi^{(k)}) \right] d\tau + E(t, \phi^{(k)})$$

  - The idea behind SDC algorithms is to evaluate the integral in the residual, E, using high-order numerical quadrature, and a simple discretization (e.g., forward or backward Euler) for the remaining integral

- If we define an iterative update, $\phi^{(k+1)} = \phi^{(k)} + \delta^{(k)}$ it is straightforward to derive the correction equation

$$\phi^{(k+1)}(t) = \phi^n + \int_{t^n}^{t} \left[F(\phi^{(k+1)}) - F(\phi^{(k)}) \right] d\tau + \int_{t^n}^{t} F(\phi^{(k)})\ d\tau$$

  - The first integral uses a simple discretization and the second is evaluated using high-order quadrature
  - Each iteration of the correction equation will increase the order of the method by 1, up to order of the numerical quadrature

MISDC Correction Equations

- Compute advection using iteratively lagged reaction source term

$$A^{n+1/2,(k+1)} = A(\phi^n, D^n, R^{(k)})$$

- Update solution by correcting diffusion with a backward Euler discretization

$$\begin{aligned}\phi_{\mathrm{AD}}^{n+1,(k+1)} \quad &= \quad \phi^n + \Delta t \left(A^{n+1/2,(k+1)} - A^{n+1/2,(k)} \right) \\ &+ \Delta t \left[D(\phi_{\mathrm{AD}}^{n+1,(k+1)}) - D(\phi^{n+1,(k)}) \right] + \int_{t^n}^{t^{n+1}} \left[A^{(k)} + D^{(k)} + R^{(k)} \right] d\tau\end{aligned}$$

- Use VODE to integrate reactions using updated advection-diffusion source terms

$$\begin{aligned}\phi_t \quad &= \quad R^{(k+1)} + \left(A^{n+1/2,(k+1)} - A^{n+1/2,(k)} \right) \\ &+ \left[D(\phi_{\mathrm{AD}}^{n+1,(k+1)}) - D(\phi^{n+1,(k)}) \right] + A^{(k)} + D^{(k)}\end{aligned}$$

MISDC Predictor

- Compute advection using time-lagged reaction source term

$$A^{n+1/2,(0)} = A(\phi^n, D^n, R^{n-1,(k_{\max})})$$

- Advance solution with Crank-Nicolson diffusion and time-lagged reaction source term

$$\begin{aligned}\phi_{\mathrm{AD}}^{n+1,(0)} = \phi^n \quad &+ \quad \Delta t A^{n+1/2,(0)} \\ &+ \quad \frac{\Delta t}{2}\left[D(\phi_{\mathrm{AD}}^{n+1,(0)}) + D(\phi^n)\right] + \int_{t^{n-1}}^{t^n} R^{(k_{\max})}\ d\tau\end{aligned}$$

- Use VODE to integrate reactions using updated advection-diffusion source terms

$$\phi_t = R^{(0)} + A^{n+1/2,(0)} + \frac{\Delta t}{2}\left[D(\phi_{\mathrm{AD}}^{n+1,(0)}) + D(\phi^n)\right]$$

MISDC Algorithm

- Underlying projection methodology is second-order, so we only need 1 correction iteration for overall second-order accuracy
  - Second-order numerical quadrature – time-centered piecewise constant representations of advection and diffusion
    - Advection – we already have this since Godunov integrator returns time-centered fluxes
    - Diffusion – midpoint rule; average diffusion contribution from beginning and end of time step
- Computational requirements - both Strang splitting and MISDC require:
  - 2 calls to VODE to advance (1 + nspecies) reaction equations
  - 2 * (1 + nspecies) advection-diffusion linear solves
  - Efficiency of Strang splitting vs. MISDC all comes down to how much work VODE requires since everything else is comparable
    - We will explore this in the results discussion

Test Problem – Hydrogen Flame

- Premixed hydrogen flame (1D)
  - GRI-Mech reaction network without carbon chemistry
    - 9 species, 27 reactions
  - 1.2 cm domain
  - Initial profile obtained from CHEMKIN-III library (PREMIX code)
  - “Fuel state” = $H_2 + O_2 + N_2$
  - Most abundant product is H_20
  - Flame propagates at approximately 15 cm/s
- We divide the domain into 256, 512, and 1024 cells and run to 2 ms using Δt = 25, 12, and 6.5 μs (fixed CFL)
- We compare each simulation to a very high resolution simulation (2048 cells) computed with the same CFL

Error and Convergence – Hydrogen Flame

MISDC

Variable	Error-256	Rate	Error-512	Rate	Error-1024
U	2.87E-02	1.81	8.19E-03	2.05	1.98E-03
ρ	4.82E-08	2.01	1.20E-08	2.20	2.60E-09
ρh	6.75E+01	1.89	1.82E+01	2.12	4.18E+00
$Y(H_2)$	3.31E-07	1.96	8.50E-08	2.15	1.91E-08
$Y(O_2)$	1.08E-05	1.97	2.78E-06	2.17	6.18E-07
$Y(H_2O)$	1.07E-05	1.96	2.74E-06	2.17	6.11E-07
$Y(N_2)$	1.38E-06	1.97	3.52E-07	2.05	7.99E-08

Strang Splitting

Variable	Error-256	Rate	Error-512	Rate	Error-1024
U	1.03E-01	4.10	6.05E-03	0.00	6.04E-03
ρ	1.82E-07	2.01	4.51E-08	2.00	1.13E-08
ρh	6.19E+02	1.38	2.38E+02	1.71	7.27E+01
$Y(H_2)$	1.36E-06	1.40	5.17E-07	1.20	2.25E-07
$Y(O_2)$	3.42E-05	1.79	9.91E-06	1.46	3.60E-06
$Y(H_2O)$	3.22E-05	1.74	9.66E-06	1.56	3.27E-06
$Y(N_2)$	4.50E-06	1.25	1.90E-06	1.49	6.73E-07

Test Problem – Methane Flame

- Premixed methane flame (1D)
  - GRI-Mech reaction network – even stiffer, and more detailed chemistry
    - 53 species, 325 reactions
  - 1.2 cm domain
  - Initial conditions obtained from CHEMKIN-III library (PREMIX code)
  - “Fuel state” = $N_2 + O_2 + CH_4$
  - Most abundant products are H_20 and CO_2
  - Flame propagates at approximately 19 cm/s
- We divide the domain into 256, 512, and 1024 cells. Run to 1 ms using $\Delta t = 12.5, 6.25$, and 3.125 μs (fixed CFL).
- We compare each simulation to a very high resolution simulation (2048 cells) computed with the same CFL.

Error and Convergence – Methane Flame

MISDC

Variable	Error-256	Rate	Error-512	Rate	Error-1024
U	1.33E-01	1.85	3.68E-02	2.03	8.99E-03
ρ	4.74E-08	2.08	1.12E-08	2.23	2.39E-09
ρh	1.17E+01	2.08	2.77E+01	2.22	5.96E+00
$Y(O_2)$	1.29E-05	2.03	3.17E-06	2.21	6.86E-07
$Y(H_20)$	7.35E-06	2.01	1.82E-06	2.17	4.05E-07
$Y(CH_4)$	3.49E-06	2.04	8.47E-07	2.22	1.81E-07
$Y(CO_2)$	9.32E-06	2.01	2.31E-06	2.21	4.99E-07
$Y(N_2)$	1.06E-06	2.01	2.63E-07	2.22	5.65E-08

Strang Splitting

Variable	Error-256	Rate	Error-512	Rate	Error-1024
U	8.74E-01	2.50	1.55E-01	3.10	1.81E-02
ρ	2.48E-07	2.45	4.55E-08	2.82	6.47E-09
ρh	6.45E+02	2.50	1.59E+02	1.97	4.08E+01
$Y(O_2)$	7.41E-05	2.41	1.39E-05	2.37	2.69E-06
$Y(H_20)$	4.54E-05	2.45	8.32E-06	2.75	1.24E-06
$Y(CH_4)$	2.02E-05	2.34	3.99E-06	2.50	7.07E-07
$Y(CO_2)$	4.57E-05	2.47	8.25E-06	1.79	2.38E-06
$Y(N_2)$	5.11E-06	1.83	1.44E-06	1.97	3.66E-07

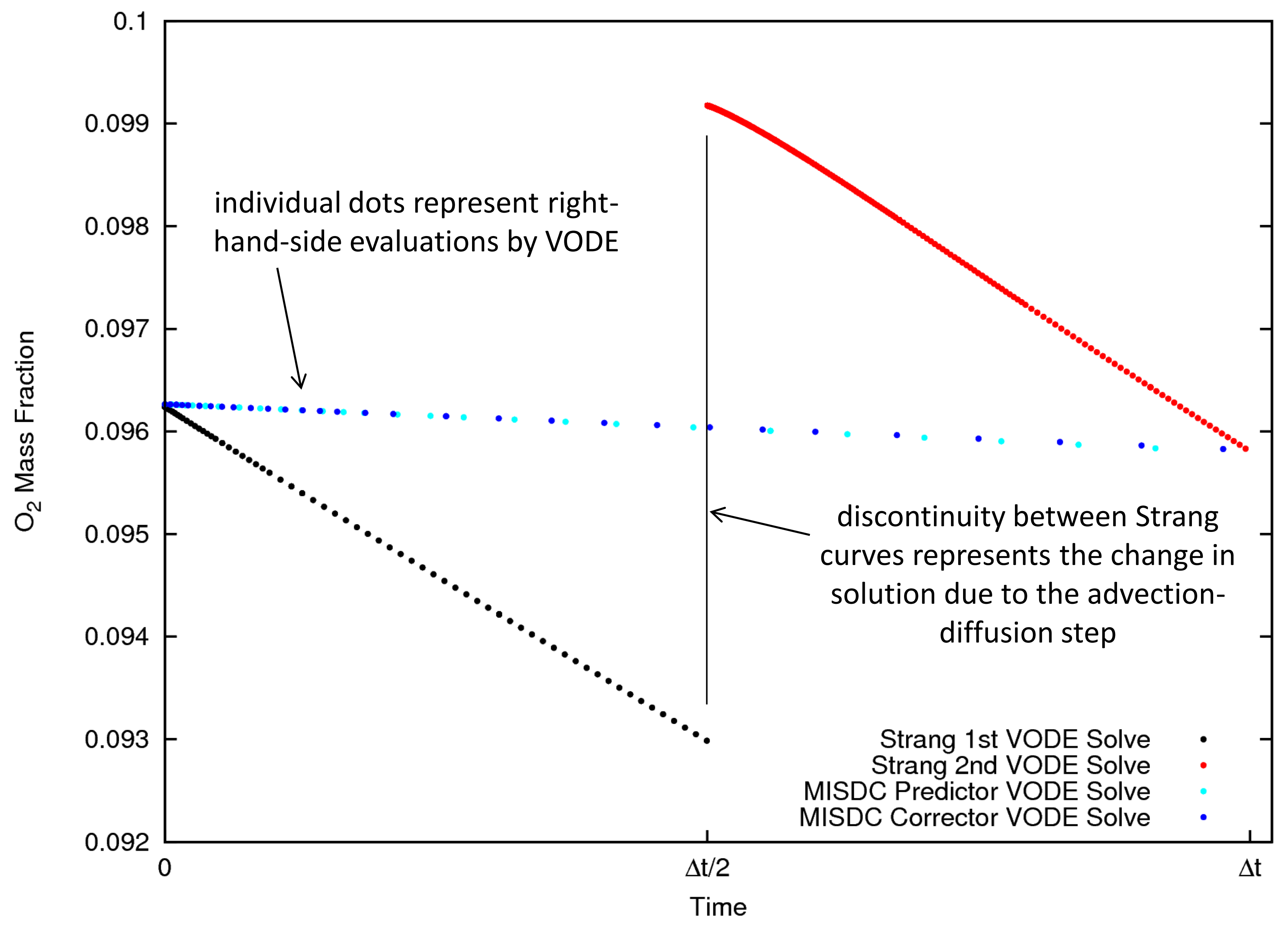
Computational Efficiency

- Consider the methane flame example with 256 cells
  - We count total right-hand-side evaluations by VODE in all chemistry solves over all cells in one time step, including those used for Jacobian evaluations
    - Strang - 48,536 evaluations
    - MISDC - 9,803 evaluations
- Thus, MISDC is actually more efficient due to the reduced time in VODE chemistry solves
  - Why is MISDC chemistry "easier"?

$$\begin{aligned}(\rho Y_m)_t &= \boxed{-\nabla\cdot(U\rho Y_m)} + \boxed{\nabla\cdot\rho\mathcal{D}_m\nabla Y_m} + \boxed{\dot{\omega}_m}\\ (\rho h)_t &= \boxed{-\nabla\cdot(U\rho h)} + \boxed{\nabla\cdot\frac{\lambda}{c_p}\nabla h + \sum_m \nabla\cdot h_m\left(\rho\mathcal{D}_m - \frac{\lambda}{c_p}\right)\nabla Y_m}\end{aligned}$$

$$\phi_t = {\color{green}A} + {\color{yellow}D} + {\color{red}R}$$

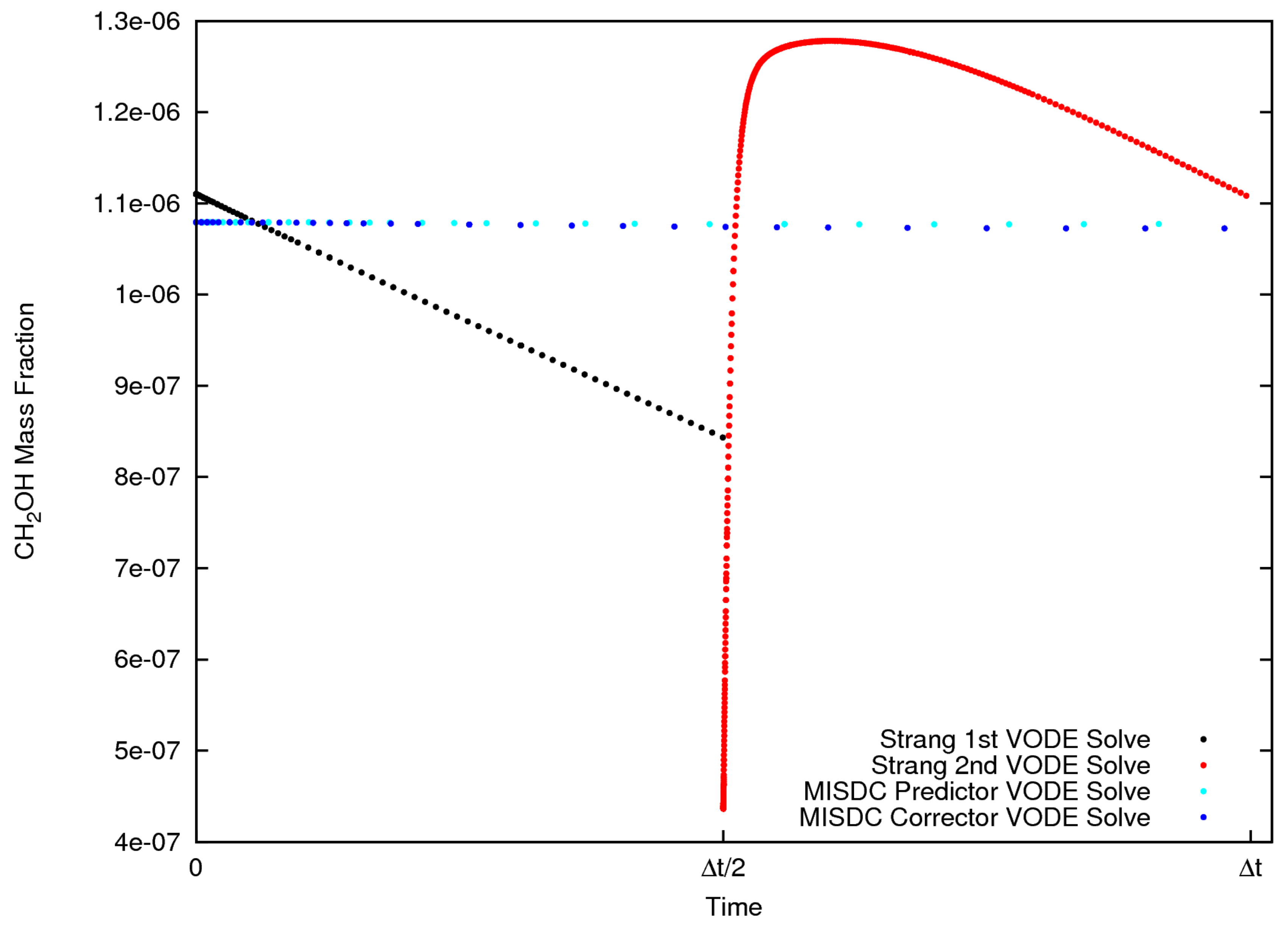
- O_2 evolution over 1 time step in one particular cell



individual dots represent right-hand-side evaluations by VODE

discontinuity between Strang curves represents the change in solution due to the advection-diffusion step

- CH_2OH evolution over 1 time step in the same cell



Summary

- We have developed a new thermodynamic advancement strategy for our low Mach number combustion simulations
- Use ideas from spectral deferred corrections to provide better coupling between advection, diffusion, and reactions
- More efficient and accurate than Strang splitting for nonlinear diffusion and detailed chemical kinetics characteristic of realistic combustion applications
- Future directions
  - Improved coupling for velocity/projection
  - Higher-order discretization
    - Suitable for many-core architectures due to reduced communication and memory per flop
  - General framework for coupling different physical processes in a multiphysics simulation