The Chemical Kinetics Time Step

a detailed lecture

Andrew Conley

ACOM Division

Simulation Time Step

- Deep convection
- Shallow convection
- Stratiform tend (sedimentation, detrain, cloud fraction, microphysics)
- Aerosol wet chemistry
- Radiation
- Couple (land, ice, ocean)
- Transport
- Sink/source for chemicals
  - Deposition, kinetics, emissions
- Vertical diffusion
- Aerosol dry deposition
- Gravity wave drag

Trust the solver?

- Robust – It can’t break, ever!
  - 1 degree horizontal resolution, 100 yrs, dt = 15mins , 30 levels
  - => 3,079,296,000,000 solves of an ODE -- order(nspecies)
  - When it fails, it does something reasonable that can be handled.
- Stable
- Consistent, accurate enough for application
- Sufficient to capture the relevant behavior
- Scientists expect to trust results

Requirements

Consistency

- Resolution->0
- Number of reactions and species
- dt->0
- Conservation laws (mass, stoichiometry)
- Time splitting
- Positive Definite

Stability

Engineering Constraints

- Exothermic reactions – Flame fronts
- Computational time (number of matrix solves and f evals)
  - Computation of f
  - jacobian solves, sparse solves.
- Complexity of method (implication for adjoint)
- Robust
  - When things fail
  - Subdividing the time step
- Context
  - Software/hardware (threading / tasks / compilers)
  - Fundamental kinetics vs simulation of weather/climate
- Result is good enough – testing?

Simple Example (Terminator Chemistry Test)

$Cl_2 \rightarrow Cl + Cl$	j	Photolytic Decomposition
$Cl + Cl \rightarrow Cl_2$	1	Recombination

Lauritzen et. al. 2015

Assume
rate constants don't vary during time step
f(solar, T, P, Surface Area Density, …)

Rates

$R(1) = j * [Cl_2]$

$R(2) = 1 * [Cl]^2$

Differential Equation

$$\frac{d[Cl_2]}{dt} = -j\,[Cl_2] + 1\,[Cl]^2$$

$$\frac{d[Cl]}{dt} = 2j\,[Cl_2] - 2\,[Cl]^2$$

Simple Example

$Cl_2 \rightarrow Cl + Cl$	j	Photolytic Decomposition
$Cl + Cl \rightarrow Cl_2$	1	Recombination

Lauritzen et. al. 2015

Rates

R(1) = $j * [Cl_2]$

R(2) = 1 $* [Cl]^2$

Differential Equation

$$\frac{d[Cl_2]}{dt} = -j\,[Cl_2] + 1\,[Cl]^2$$

Non-linear

$$\frac{d[Cl]}{dt} = 2j\,[Cl_2] - 2\,[Cl]^2$$

$y' = y^2 \quad y(0) = y_0$

Solution: $y = 1 / (y_0 - t)$

Blows up at $t = y_0$

Implications?

- Mass increases without bound
- Believe that?
- But mass is bounded (or our list of reactions is wrong.)
- “Don’t worry, be happy”

y’ = f(y)   Theory

- $f'(y)$ exists -> unique solution exists
- $\vec{f}(\vec{y}, t)$ -> More complicated conditions for existence, uniqueness
- Hope?
  - Seems to work, except when we implement bad collections of reactions
  - Scientists ***trust*** the underlying numerical method
  - Learn from linear systems

Linearized (forward sensitivity)

$Cl_2 \rightarrow Cl + Cl$	j	Photolytic Decomposition
$Cl + Cl \rightarrow Cl_2$	1	Recombination

Lauritzen et. al. 2015

$$\frac{d[Cl_2]}{dt} = -j[Cl_2] + 1\,[Cl]^2$$

$$\frac{d[Cl]}{dt} = 2j[Cl_2] - 2\,[Cl]^2$$

$$\frac{dy}{dt} = f(y)$$

Linear Form

$$\frac{d}{dt}\begin{pmatrix} e_{Cl_2} \\ e_{Cl} \end{pmatrix} = \begin{pmatrix} -j & 2[Cl] \\ 2j & -4[Cl] \end{pmatrix}\begin{pmatrix} e_{Cl_2} \\ e_{Cl} \end{pmatrix}$$

$$\frac{dy}{dt} = \frac{df}{dy}y = Ay$$

Linearized (forward sensitivity)

$Cl_2 \rightarrow Cl + Cl$	j	Photolytic Decomposition
$Cl + Cl \rightarrow Cl_2$	1	Recombination

Lauritzen et. al. 2015

$$\frac{d[Cl_2]}{dt} = -j[Cl_2] + 1\,[Cl]^2$$

$$\frac{d[Cl]}{dt} = 2j[Cl_2] - 2\,[Cl]^2$$

$$\frac{dy}{dt} = f(y)$$

Forcing

Linear Form

$$\frac{d}{dt}\begin{pmatrix} e_{Cl_2} \\ e_{Cl} \end{pmatrix} = \begin{pmatrix} -j & 2[Cl] \\ 2j & -4[Cl] \end{pmatrix}\begin{pmatrix} e_{Cl_2} \\ e_{Cl} \end{pmatrix}$$

$$\frac{dy}{dt} = \frac{df}{dy}y = Ay$$

Jacobian

Stiffness

- $\frac{d\vec{y}}{dt} = A\,\vec{y}$ $\qquad A(y) = \frac{d\vec{f}}{d\vec{y}}$
- Jordan Normal Form
  - $E^{-1}\,A\,E = J$
  - Diagonal of eigenvalues $(\lambda_1, \lambda_2, ..)$
  - Rank of A is ~(number of species – # conservation laws)

Stiffness: $|\lambda_{max}|\;dt$

> 1 Be Careful. This is a stiff ODE.

< 1 Ok

$< \frac{1}{8}$ Smaller is better, but…

Stiffness

- $\frac{d\vec{y}}{dt} = A\,\vec{y}$ $\qquad A(y) = \frac{d\vec{f}}{d\vec{y}}$

Stiffness: $|\lambda_{max}|\ dt$

> 1	Be Careful. This is a stiff ODE.
< 1	Ok
$< \frac{1}{8}$	Smaller is better

Eigenvalues vary with mechanism and local environment

Time step, dt, imposed by external model

A-priori estimates are rare

Computation can be expensive (Krylov methods)

If it is atmospheric chemistry

- it is probably stiff (or make dt larger)
- or uninteresting from a solver viewpoint

$$\frac{d}{dt}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -5 & 1 \\ 1 & -\frac{1}{5} \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \qquad \lambda = \{0, -5.2\}$$

Initial Condition: $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix}$

Solution: $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}(t) = \begin{pmatrix} 1 \\ 5 \end{pmatrix} e^{0*t}$

Initial Condition: $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 5.1 \end{pmatrix}$

Solution: $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}(t) = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + 0.1\begin{pmatrix} -5 \\ 1 \end{pmatrix} e^{-(5.2)t}$

Forward Euler (explicit)

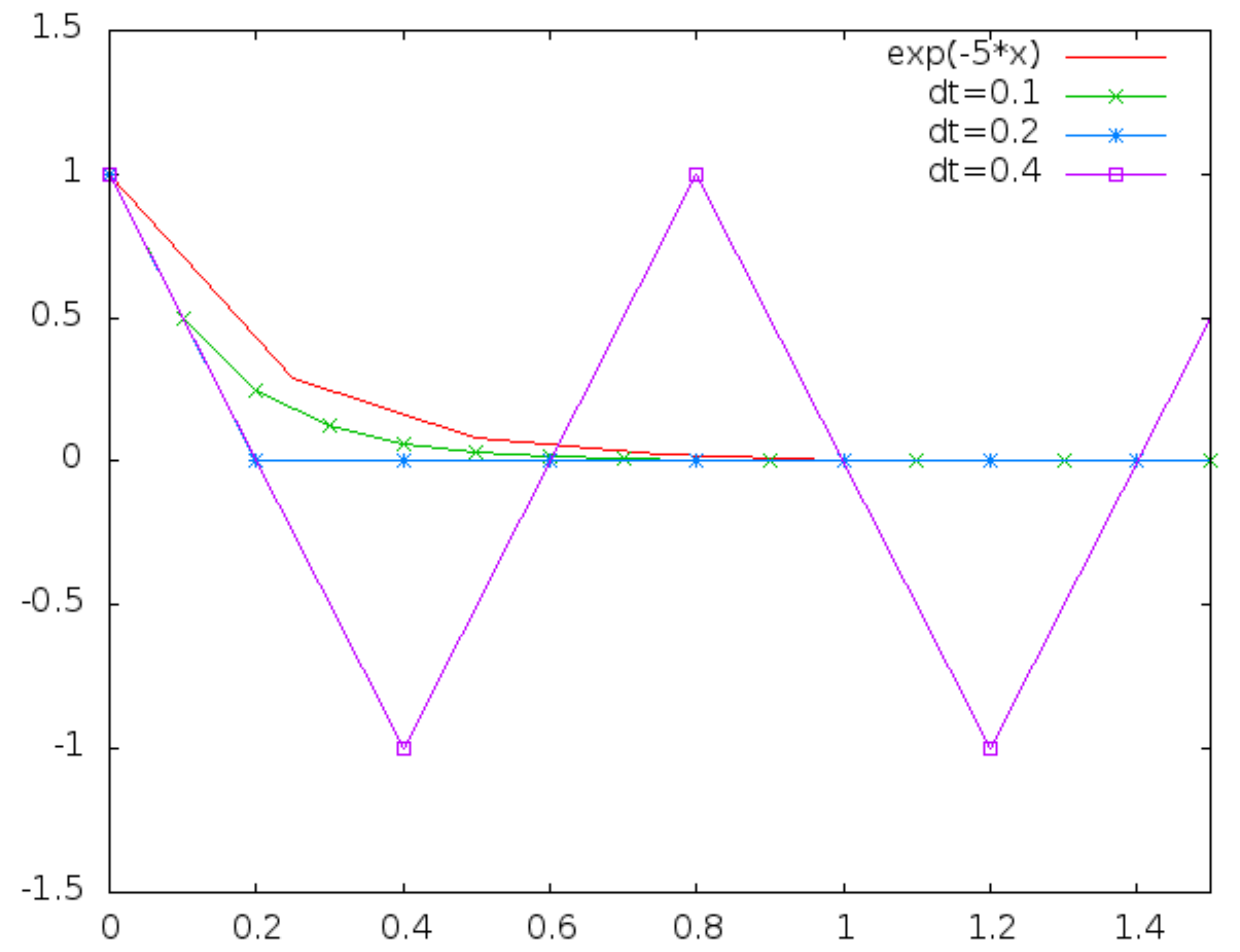
$$\frac{y(dt) - y(0)}{dt} = f(y(0))$$

$$y' = \lambda y \quad -> \qquad \frac{y(dt) - y(0)}{dt} = \lambda\, y(0)$$

$$y(dt) = (1 + \lambda\, dt)\ y(0)$$

$$y(n\, dt) = (1 + \lambda\, dt)^n\, y(0)$$

Forward Euler $y' = -5\,y$



Forward Euler

$$\frac{y(dt) - y(0)}{dt} = f(0) = \lambda\, y(0)$$

$$[\lambda dt = z]$$

$$y\left(\frac{n}{\lambda}\, z\right) = (1+z)^n\ \ y(0) = \phi(z)^n$$

Need $|\phi(z)| = \ |1+z| < 1$

Forward Euler $y' = f(y) = \lambda\, y$ $y = y_0 e^{\lambda t}$

$$\frac{y(dt) - y(0)}{dt} = f(0) = \lambda\, y(0)$$

$$y(dt) = (1 + \lambda\, dt)\; y(0)$$

$$y(n\, dt) = (1 + \lambda\, dt)^n\, y(0)$$

$$[\lambda dt = z]$$

Numerical: $y\left(\frac{n}{\lambda}\, z\right) = (1 + z)^n\; y(0)$

Exact: $y\left(\frac{n}{\lambda}\, z\right) = e^{n\, z}\, y(0)$

Backward Euler (implicit)

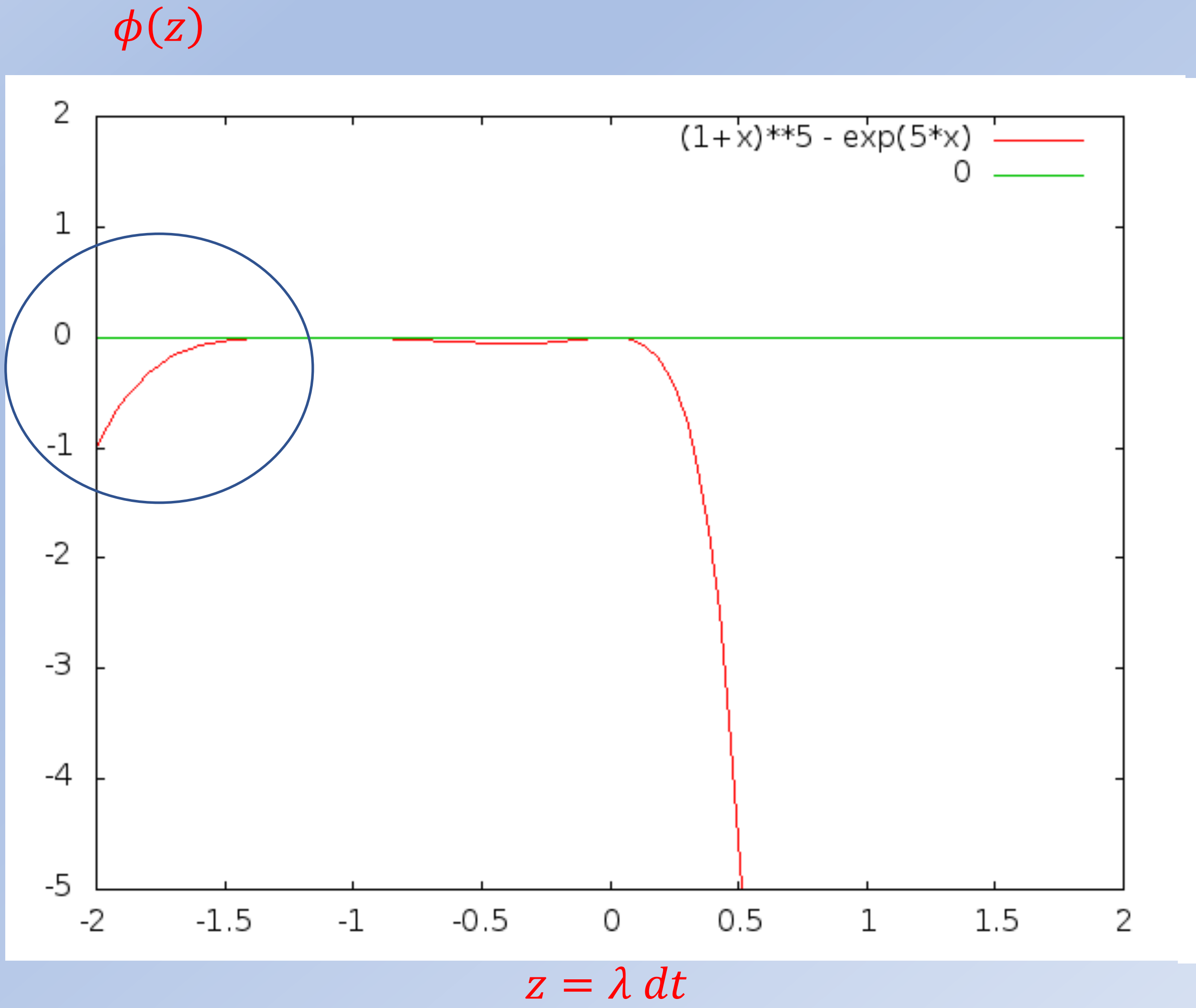
$$\frac{y(dt) - y(0)}{dt} = f(dt) = \lambda\, y(dt)$$

$$y\left(\frac{n}{\lambda}\, z\right) = (1 - z)^{-n}\ y(0)$$

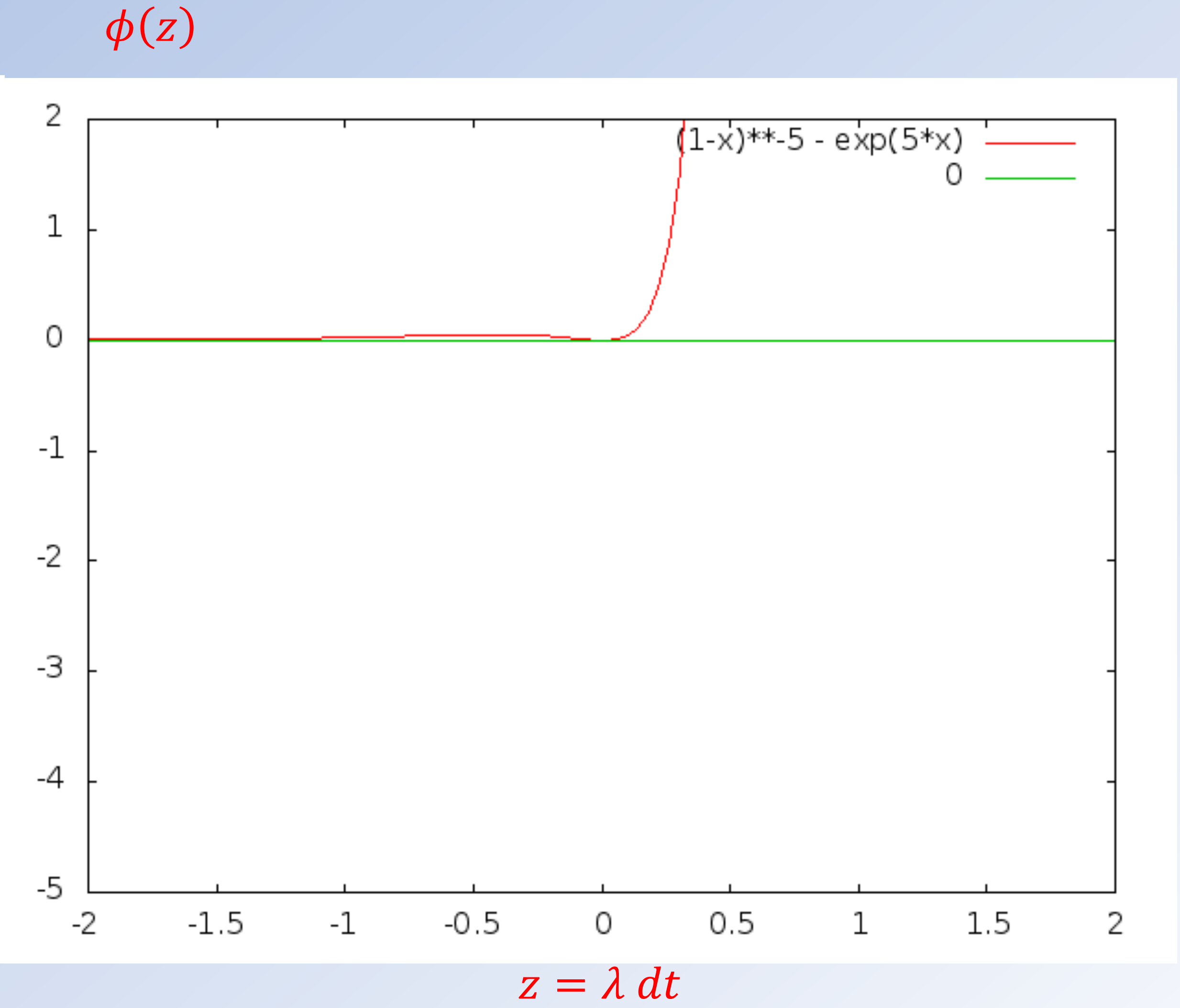
Exact: $y\left(\frac{n}{\lambda}\, z\right) = e^{n\, z}\, y(0)$

$$\phi(z) = \ (1 - z)^{-1}$$

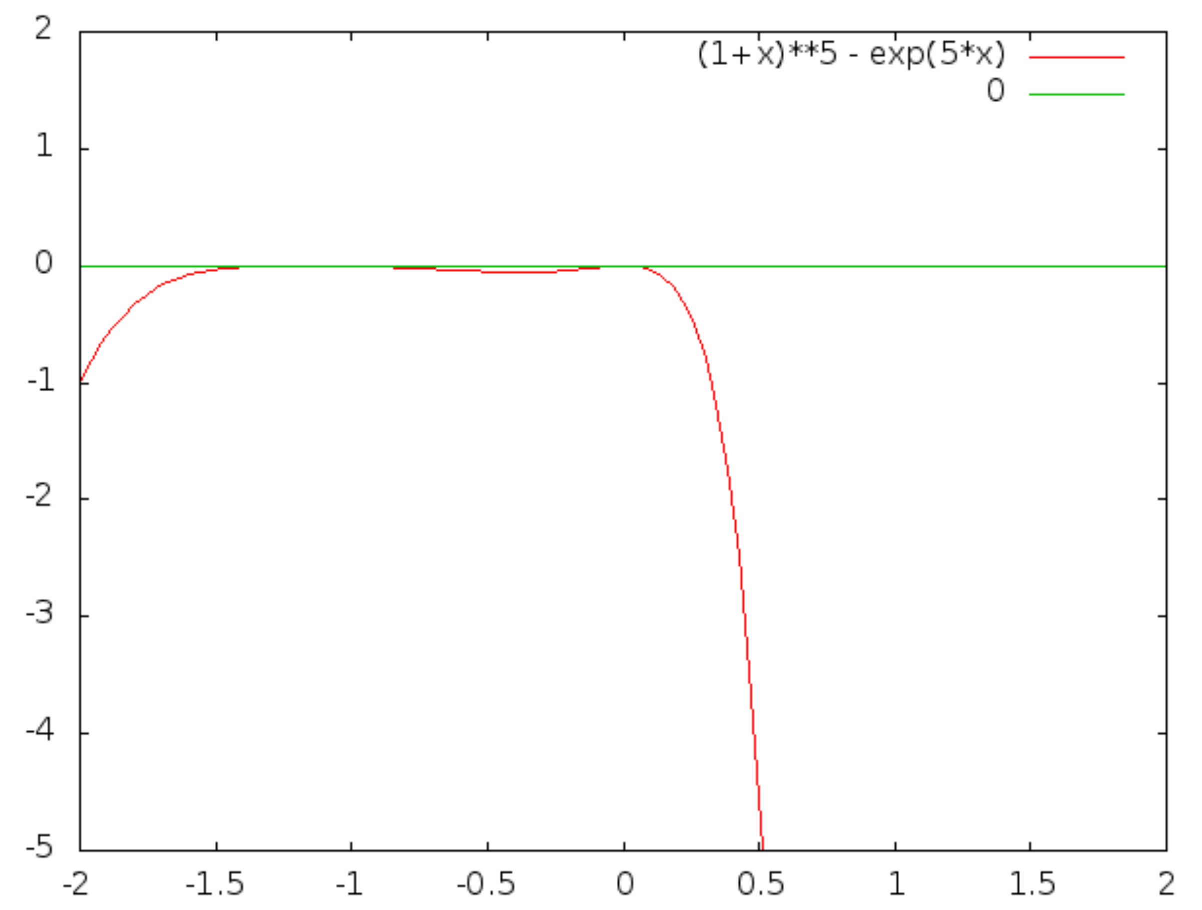
Forward Euler
Not A-stable

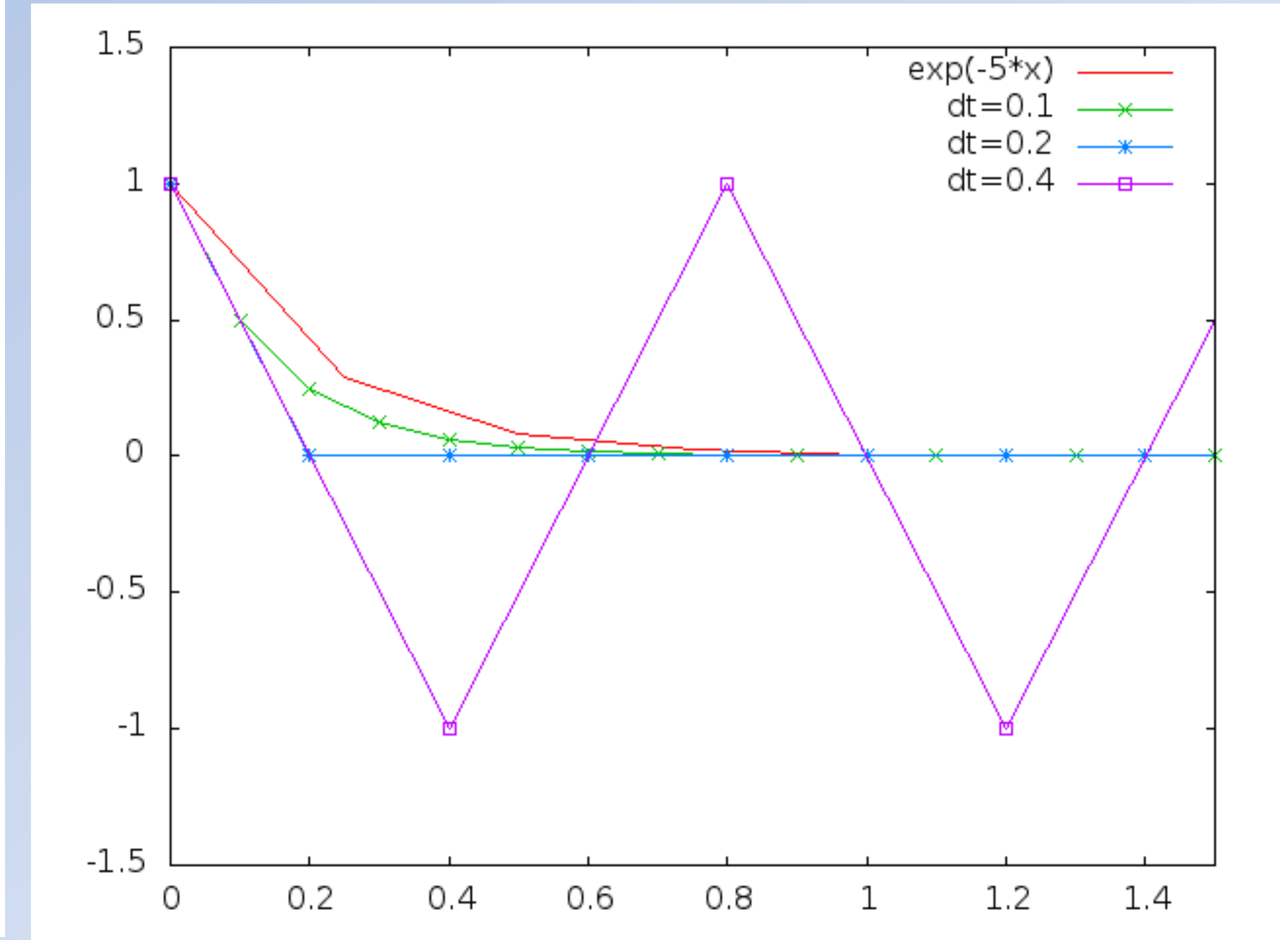


Backward Euler
A-stable



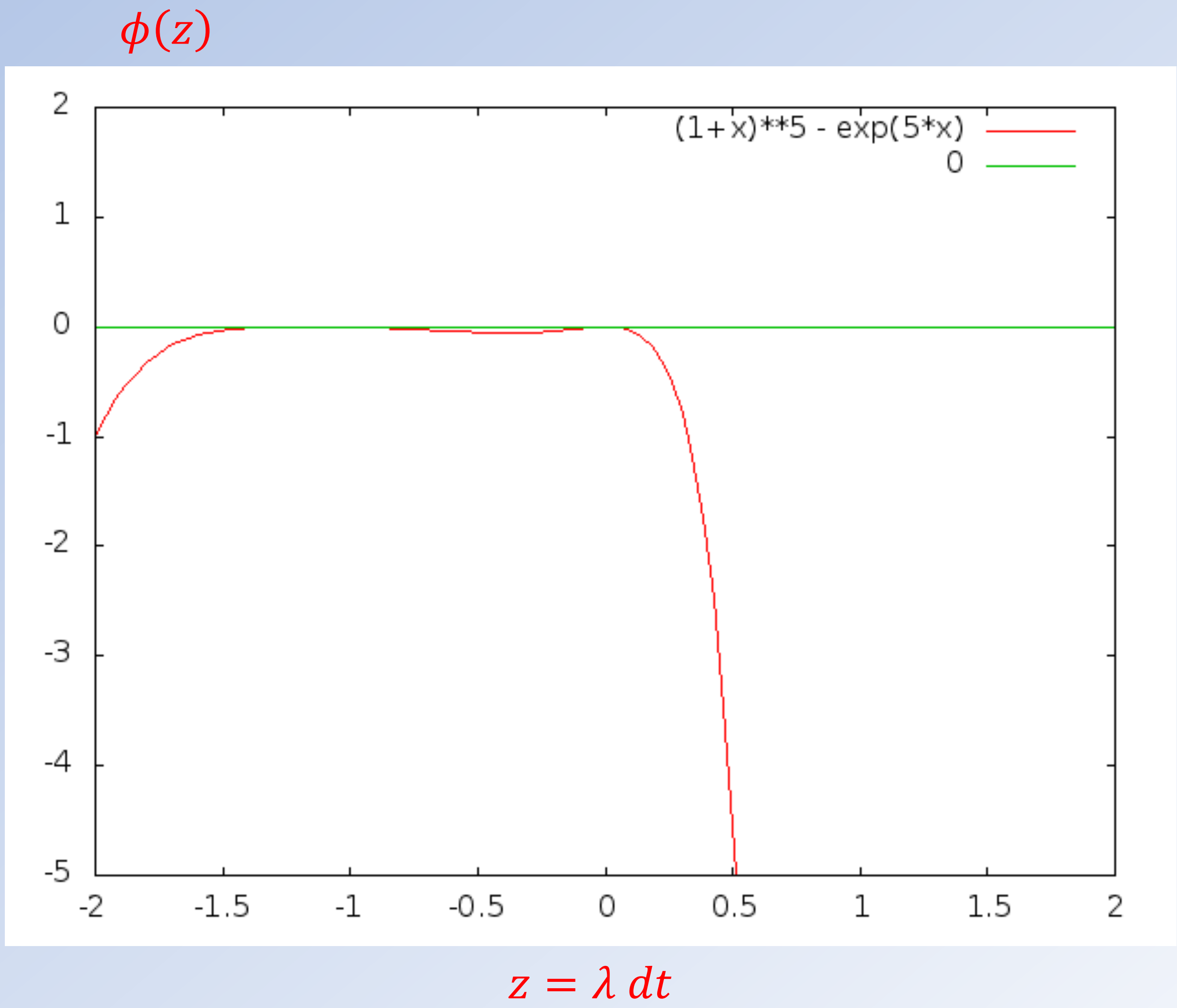
Forward Euler $y' = -5\,y$





Concerned: $\lambda > 0$?

- Mass conservation
- But away from equilibrium, could $\lambda > 0$?
- Also, not everything is linear near equilibrium



Popular Methods

- Trapezoidal (Implicit)
  - $\frac{y_{n+1} - y_n}{dt} = \frac{1}{2}\,[\,f(y_{n+1}) + f(y_n)\,]$

$$\phi(z) = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$$

- Adams-Bashforth (multistep, explicit)
  - $\frac{y_{n+1} - y_n}{dt} = \frac{3}{2}\,f(y_n) - \frac{1}{2} f(y_{n-1})$
- Runge-Kutta (commonly RK4, explicit)
  - Multistep, but y_{stage} depends on $f\left(y_{stage-1}\right)$
  - Reduces to Simpson's rule if $f(y,t)$ not a function of y

Linear Stability

- A-stable
  - If $|\phi(z)| < 1$ includes all negative real numbers
- L-stable
  - If, additionally $\phi(z) \rightarrow 0 \ as \ z \rightarrow -\infty$
- B-stability
  - Stability for Runge-Kutta methods
- No explicit method is linearly A-stable

Choose

- Trapezoidal (Implicit)
  - $\frac{y_{n+1} - y_n}{dt} = \frac{1}{2}\,[\,f(y_{n+1}) + f(y_n)\,]$ $\phi(z) = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$
- Adams-Bashforth (multistep, explicit)
  - $\frac{y_{n+1} - y_n}{dt} = \frac{3}{2}\,f(y_n) - \frac{1}{2} f(y_{n-1})$
- Runge-Kutta (commonly RK4, explicit)
  - Multistep, but y_{stage} depends on $f\left(y_{stage-1}\right)$
  - Reduces to Simpson's rule if $f(y,t)$ not a function of y

Requirements

Consistency

- Resolution->0
- Number of reactions and species
- Order of Approximation
- Conservation laws (mass, stoichiometry)

Stability $|\phi(z)| < 1$ $\phi(z \to \infty) \to 0$

Engineering Constraints

- dt > 0
- Positive Definite
- Time splitting
- Exothermic reactions – Flame fronts
- Computational time (number of matrix solves and f evals)
  - Computation of f
  - jacobian solves, sparse solves.
- Complexity of method (implication for adjoint)
- Robust
  - When things fail
  - Subdividing the time step
- Context
  - Software/hardware (threading / tasks / compilers)
  - Fundamental kinetics vs simulation of weather/climate
- Result is good enough – testing?

Accuracy (consistency)

- Adams-Bashforth
  - $\frac{y_{n+1} - y_n}{dt} = \frac{3}{2} f(y_n) - \frac{1}{2} f(y_{n-1})$
  - Taylor expand and get errors relative to $y' = f(y)$
  - Error $< \frac{5(dt)^3}{12} \max(|f'''|)$
  - Local order 3, global order 2
  - Consistent, not A-stable
- Order has to have local order larger than 1 to be consistent
- Error terms can drive subdivision of time step (Rosenbrock method)
- Convergence = Stabile + Consistent So what? (dt > 0)

Conservation

- Stoichiometry implies atoms are conserved.

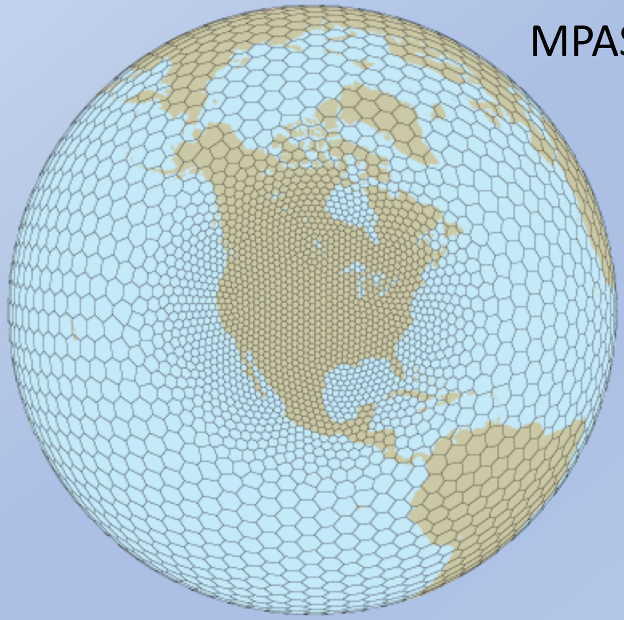
$Cl_2 \rightarrow Cl + Cl$	j	Photolytic Decomposition
$Cl + Cl \rightarrow Cl_2$	1	Recombination

- $2\ Cl_2 + \ Cl =$ Constant
- As many 0 eigenvalues as atoms – perhaps.
- Numerical conservation?

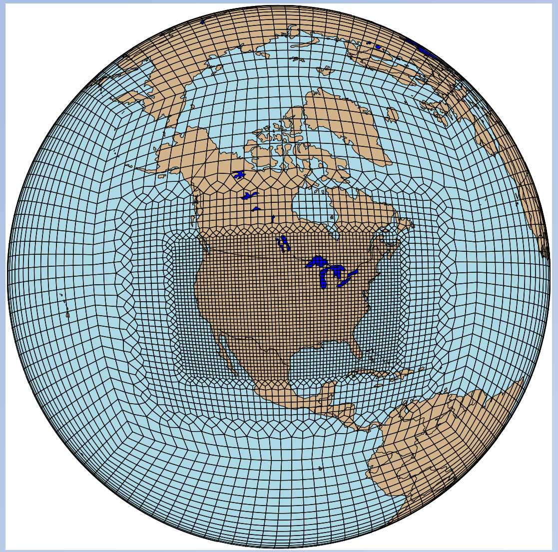
Other consistency-like considerations

- Do we have all the relevant reactions to support study?
  - “chemical resolution/consistency”
- Horizontal resolution:
  - Nonlinear
  - $<v_1> <v_2> ?=? <v_1 v_2>$

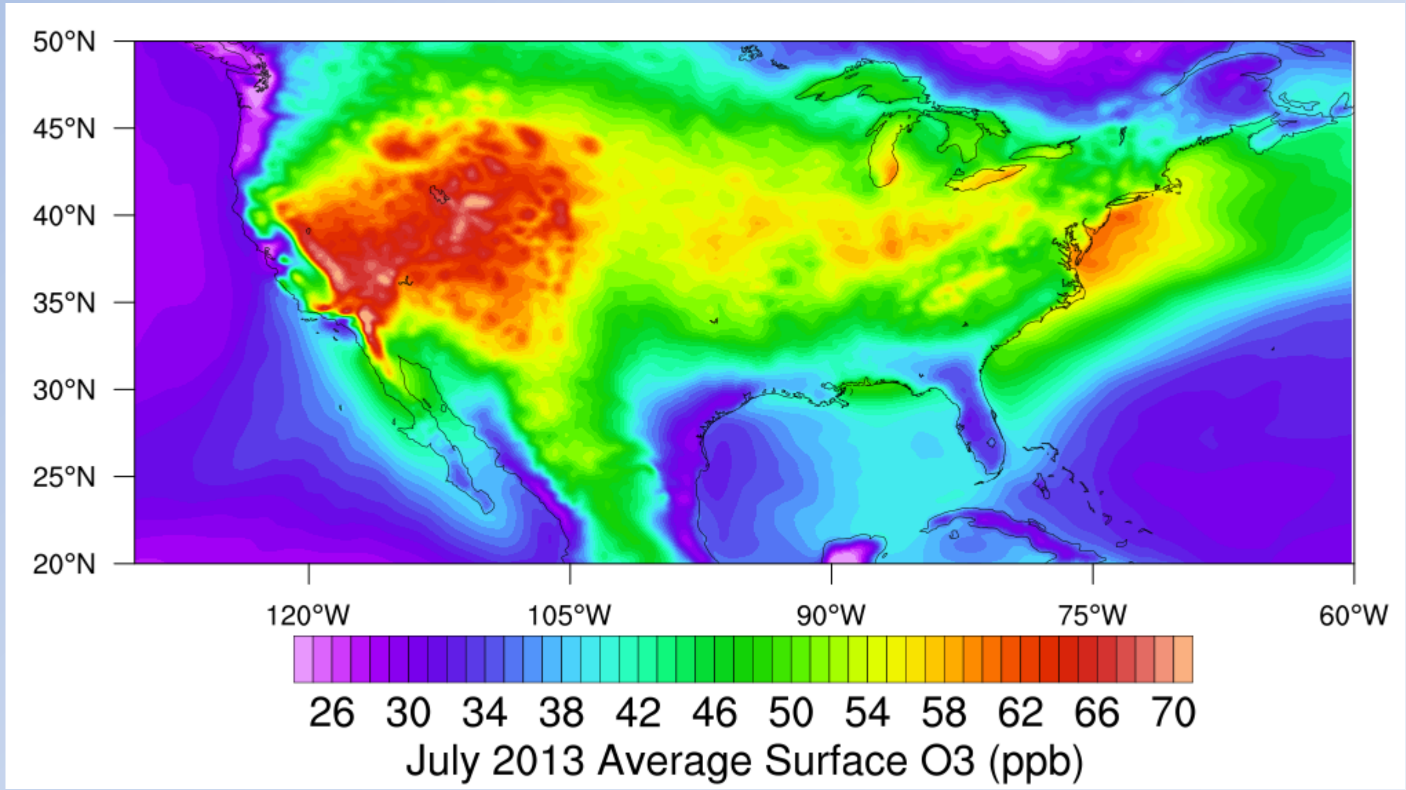
MPAS



CESM-SE



Biogenic Emissions, chemistry, CONUS refinement



Rebecca Schwantes, Forrest Lacey, In Development

Requirements

Consistency

- Resolution->0
- Number of reactions and species
- Order of Approximation
- Conservation laws (mass, stoichiometry)

Stability $|\phi(z)| < 1$

Engineering Constraints

- Positive Definite
- Time splitting
- Exothermic reactions – Flame fronts
- Computational time (number of matrix solves and f evals)
  - Computation of f
  - jacobian solves, sparse solves.
- Complexity of method (implication for adjoint)
- Robust
  - When things fail
  - Subdividing the time step
- Context
  - Software/hardware (threading / tasks / compilers)
  - Fundamental kinetics vs simulation of weather/climate
- Result is good enough – testing?

Trust the solver?

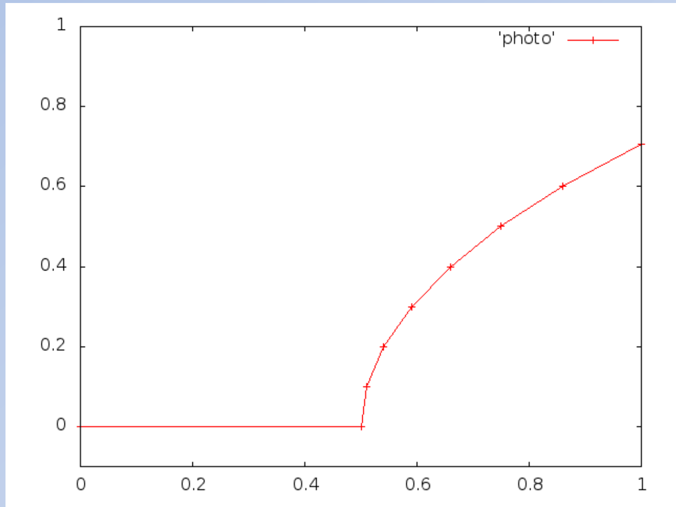
- Global simulation:
  - 1 degree horizontal resolution, 100 yrs, dt = 15mins , 30 levels
  - => 3,079,296,000,000 solves of an ODE of order(nspecies)
  - When it fails, it does something reasonable that can be handled.
- Stable, accurate
- Sufficient to capture the relevant behavior

Implicit Method Example (Trapezoidal)

$$\frac{y_{n+1} - y_n}{dt} = \frac{1}{2}\left[\, f(y_{n+1}) + f(y_n) \,\right]$$

- Algebraic solve for y_{n+1}
  - $\mathrm{g}(y_{n+1}) = 0$
- Often use Newton method (or peudo-Newton Iterations)
  - $\mathrm{g}(z) = 0$
  - $z_{n+1} = z_n - \frac{g(z_n)}{g'(z_n)}$
  - $g'(z_n)$ is the Jacobian.
- expensive, Sparse solve, yet more methods
- Computing $g'(z_n)$ is ~2.5 times as expensive as forcing
- Factoring/solving Jacobian is very expensive – many optimizations
- Some methods compute these iterates approximately
- Sometimes Newton iterates diverge

Considerations



- Solar Terminator
  - Very rapid transients
  - Time-varying rate constants
  - Rosenbrock fails if solar terminus crosses cell during time step
    - Sun can rise inside the chem-solve time step (during dt)
    - Simply use j-values at one end or other of time step
- Conservation. (CL + 2*CL_2)
- Positive definite   vmr > 0
- Convergence constraints   scale = max(C, vmr)   Y_scale = Y / scale

Contexts

- Size of Mechanism (2 reactants -> 1e6 reactants)
- Size of domain (box, 3d global, LES; number of time steps)
- Efficiency, robustness
- Need for adjoint
- Types of dependency on environmental conditions (T, P, SAD, aqueous phase storage)
- Cost (wallclock vs charges for solution vs energy consumed)
- Do rate constants vary inside chemistry time step?
- Exothermic? Might need Jacobian w.r.t. temperature
- Can you construct a Jacobian? if not, difficult to make stable. Numerical Jacobian or projection controls.
- Accuracy vs speed of computation
- Operator splitting (photolysis and kinetics, vs constant photolysis)
- Software Engineering (thread-safe, thread-aware, multi-tasking, data size and typing, languages/compilers/frameworks)
- Complexity of representation vs generality of solution method (e.g., Linear Solver)
- Hardware architecture (vectorization, GPU’s, Nspecies/cache size)
- Optimizations (Hard-coded LU factorization and solves)
- Restarts due to accuracy constraints (accuracy estimates on solution; solar changes during time step)
- Absolute and relative tolerances on errors in solution of ODE for each species

NCAR model usage

CESM

- Implicit, except for slowly varying chemicals
- If Newton iteration fails, subdivide the time step
- Sparse linear algebra is sophisticated

WRF

- Rosenbrock – higher order, a bit more fussy
- Implicit, similar to Runge-Kutta, with a few other corrections
- Error estimates lead to subdivisions of the time step
- Sparse linear algebra is sophisticated

GECKO

- So many chemicals that explicit representation of forcing uses indirect addressing
- No Jacobian, estimate of unstable directions

Choose among these

- Fast Computation
- Fast Development Cycle
- Robust
- Extensible code
- Easy-to-read code
- Inexpensive to develop

Research

- Continuous between Implicit and Explicit – exposes reasons behind instability
  - $y_p(dt) - y_p(0) = dt\left[(1-p)\left(f\left(y_p(0)\right)\right) + (p)\left(f\left(y_p(dt)\right)\right)\right]$
- GPU's
- "Shove everything into the Chemistry solver"
  - Emissions
  - Deposition
  - Microphysics
- Solve for both concentration and variability of concentration
- Chemistry solvers in models
  - Unique requirements speed and robustness
  - Use cases (number of solves, consistency of Jacobian form during solve)
  - Non-symmetric Sparse Jacobian
- Computational cost vs Energy cost
- Cost relative to other processes (e.g., transport)
- Test Suites?

In Application

- Community
- Familiarity with tools (KPP, mozartPP, fortran, run scripts, software)
- Tested in many experiments (Trust)
- Speed/cost
- Embedding model probably has something implemented
- Documentation (user, traceability)
- Access to baseline chemistry mechanisms

Requirements

Consistency (Order > 2)

- Resolution->0
- Number of reactions and species
- Order of Approximation
- Conservation laws (mass, stoichiometry)

Stability $|\phi(z)| < 1$

Engineering Constraints

- Time splitting
- Exothermic reactions – Flame fronts
- Computational time
  - Computation of f
  - jacobian solves, sparse solves.
- Complexity of method (implication for adjoint)
- Robust
  - When things fail
  - Subdividing the time step
- Context
  - Software/hardware (threading / tasks / compilers)
  - Fundamental kinetics vs simulation of weather/climate
- Result is good enough – testing?

Thank you

Resources

- Stoer, Bulirsch; Introduction to Numerical Analysis
- Press et. al.; Numerical Recipies
- Wilkinson; The Algebraic Eigenvalue Problem
- Golub, VanLoan; Matrix Computations
- Packages
  - FATODE
  - VSODE
  - KPP