Notes

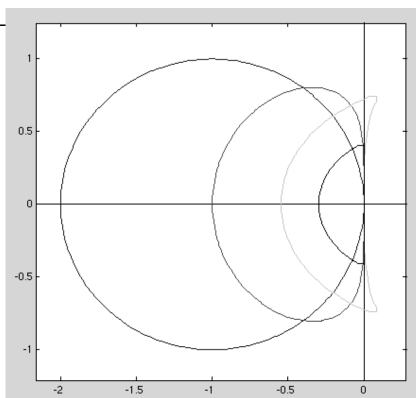
- Notes for last part of Oct 11 and all of Oct 12 lecture online now
- Another extra class this Friday 1-2pm

Adams-Bashforth

- Adams-Bashforth family are examples of **linear multistep methods**
  - Linear: the new y is a linear combination of y's and f's
  - Multistep: the new y depends on several old values
- Efficient
  - Can get high accuracy with just one evaluation of f per time step
  - Can even switch order/accuracy as you go
- Reasonably stable
  - AB3 and higher include some of the imaginary axis
- Rephrased as a "multivalue method", can easily accommodate variable time steps…

Adams-Bashforth Stability

- AB1-4
- Note: gets smaller with increasing order…



Starting Up

- Problem: how do you get a multistep method started?
  - Without sacrificing global accuracy
- Need an alternate approach to high order, **single-step methods**
- Classic example: Runge-Kutta (RK) methods
- Extra information comes from additional evaluations of f, not old values
  - Avoiding old (and thus distant) data helps for stability and magnitude of truncation error too…
  - RK is thus very popular on its own merits

Example Runge-Kutta Methods

- Forward Euler
- Heun's method (predictor/corrector) RK2
  - Based on trapezoidal rule for integration…

$$y^{(1)} = y_n + \Delta t\, f(y_n, t_n)$$
$$y_{n+1} = y_n + \Delta t \tfrac{1}{2}\left(f(y_n, t_n) + f\left(y^{(1)}, t_{n+1}\right)\right)$$

- Midpoint RK2
  - Based on midpoint rule for integration…

$$y_{n+1/2} = y_n + \frac{\Delta t}{2} f(y_n, t_n)$$
$$y_{n+1} = y_n + \Delta t\, f\left(y_{n+1/2}, t_{n+1/2}\right)$$

Finding RK methods

- Often described by how many evaluations ("stages") and order of accuracy
  - Usually not uniquely determined though – many, many RK methods out there
- Generally finding "optimal" methods (minimum # stages for given accuracy) is an unsolved problem
- Several standard schemes exist out there

Classic RK4

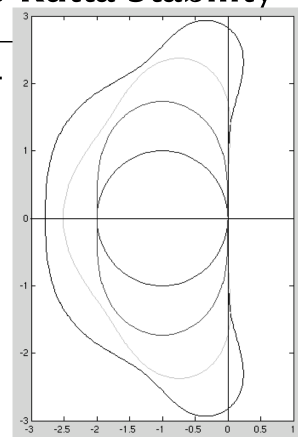
- Probably the most widely used higher order time integration scheme

$$k_1 = \Delta t\, f\left(y_n, t_n\right)$$
$$k_2 = \Delta t\, f\left(y_n + \tfrac{1}{2}k_1, t_{n+\frac{1}{2}}\right)$$
$$k_3 = \Delta t\, f\left(y_n + \tfrac{1}{2}k_2, t_{n+\frac{1}{2}}\right)$$
$$k_4 = \Delta t\, f\left(y_n + k_3, t_{n+1}\right)$$
$$y_{n+1} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

Runge-Kutta Stability

- Forward Euler
- 2-stage RK2
- 3-stage RK3
- 4-stage RK4

- Can trade accuracy for stability…



Adaptive time steps

- General idea: take large time steps where solution is smooth
  - Truncation error is $O\left(\Delta t^p \frac{\partial^p y}{\partial t^p}\right)$
- Example approach:
  - Use p'th and p+1'st order integrators
  - Difference estimates error of p'th order scheme
  - Modify Δt for next time step to attempt to keep error per unit time constant
  - N.B.: use p+1'st order answer to go forward…
- Runge-Kutta-Fehlberg (RKF) pairs: can sometimes reuse much of computation of p'th method to get p+1'st method

Looking at error

- Heuristic error control isn't guaranteed!
- Usual validation approaches:
  - Test your method on a known exact solution
  - Test your method against real experimental data (modeling error also included)
  - Run solver multiple times, with smaller and smaller time steps
    - Plot error against Δt
    - Look at ratio of error when Δt halved

Stiffness

- Things may go wrong however!
- Simple example: $\frac{dy}{dt} = 1 - 1000(y - t), \quad y(0) = 1$
- Forward Euler stability restriction: always need Δt < 0.002
- First order accuracy: for t>0.05, can use gigantic Δt

- Problem is **stiffness**: stability of method requires much smaller time step than accuracy demands
- So far we can't efficiently solve stiff problems

Stiffness analyzed

- Usually results from hugely different time-scales in the problem
- Linear example: $\frac{dy}{dt} = \begin{bmatrix} -100 & \\ & -0.01 \end{bmatrix} y$

- The "fast" mode may be transient–quickly decays to zero–so the "slow" mode determines truncation error
- But the "fast" mode determines stability time step restriction

Reversing time

- Consider λ with positive real part
- Unstable when going forwards in time (and FE etc. are similarly unstable, particularly for big time steps)
- Now, **reverse time**
  - Exponential growth, in reverse, is stable exponential decay
  - Reversed methods are stable!
- Equivalent to regular time, λ with negative real part

Backwards Euler

- Backwards Euler: reverse version of FE

$$y_{n+1} = y_n + \Delta t\, f\left(y_{n+1}, t_{n+1}\right)$$

- This is an **implicit** method: new y defined implicitly (appears on both sides)
- Methods from previous slides are all **explicit**: new y explicitly computed from known values
- Going implicit is the key to handling stiffness

Other implicit methods

- Backwards Euler is over-stable:

$$\left|1 - \lambda \Delta t\right| > 1$$

- **A-stable**: region of stability includes left half-plane (stable when exact solution is)
- Implicit mid-point

$$y_{n+1} = y_n + \Delta t\, f\left(\tfrac{1}{2} y_n + \tfrac{1}{2} y_{n+1}, t_{n+1/2}\right)$$

- Trapezoidal rule

$$y_{n+1} = y_n + \Delta t\left[\tfrac{1}{2} f\left(y_n, t_n\right) + \tfrac{1}{2} f\left(y_{n+1}, t_{n+1}\right)\right]$$

Even more

- Implicit multistep methods:

  Adams(-Bashforth)-Moulton

  Backwards Differentiation Formula (BDF)

- Implicit Runge-Kutta
  - Might need to solve for multiple intermediate values simultaneously…