10.34, Numerical Methods Applied to Chemical Engineering
Professor William H. Green
Lecture #13: Stiffness. MATLAB® Ordinary Differential Equation (ODE) Solvers.

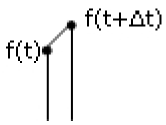
From Last Lecture: Numerical Integration

$d\underline{Y}/dt = \underline{F}(\underline{Y})$ $\underline{Y}(t_0) = \underline{Y}_0$ $\underline{G}$: estimated time average slope from $t \rightarrow t+\Delta t$

General Algorithm: $\underline{Y}(t+\Delta t) = \underline{Y}(t) + \Delta t * \underline{G}(\underline{Y})$ $\underline{G}$ = (time avg. slope) + δ (error)

Rectangle Rule: Explicit Euler $\underline{G} = \underline{F}(\underline{Y}(t))$ EXPLICIT

Trapezoid Rule: $\underline{G} = ½(\underline{F}(\underline{Y}(t)) + \underline{F}(\underline{Y}(t+\Delta t)))$ IMPLICIT
($\underline{Y}(t+\Delta t)$: unknown)



$\delta \sim O((\Delta t)^m)$
want $\Delta t \downarrow$
Requirement for accuracy sets ceiling on Δt

Figure 1. Linear approximation to a function.

MATLAB

ode45

Runge-Kutta: G formula where error scales $(\Delta t)^5$

If Δt is small, error is small, but takes many steps (tradeoff)

* new $t \leftarrow t+\Delta t$

Adding big numbers and small numbers $\rightarrow$ lose $\log_{10}(N_{timesteps})$ sig figs

as Δt decreases. This can be a significant problem.

If computer has 14 sig figs

If you want 6 sig figs in $\underline{Y}(t_f)$: $N_{timesteps} < 10^8$

$(t_f - t_0)/\langle\Delta t\rangle < 10^8$ {FLOOR}

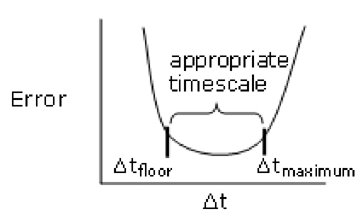


Figure 2. If Δt_{floor} is larger than $\Delta t_{maximum}$, then a solution cannot be found.

Adaptive Timestepping

Use small Δt when necessary (to keep δ small)

Use big Δt everywhere else to save CPU time and minimize roundoff error.

$\delta \sim O((\Delta t)^m)$

Richardson Extrapolation

Solve ODE using $\Delta t = 0.1$s $\quad$ $\underline{Y}(t_f;\ \Delta t=0.1)$

Solve same ODE using $\Delta t = 0.05$s $\quad$ $\underline{Y}(t_f;\ \Delta t=0.05)$

$\underline{Y}(t_f;\ \Delta t) = \underline{Y}_{time}(t_f) + c(\Delta t)^m + \ldots$ (unknown higher order of error)

$\underline{Y}(t_f;\ {}^{\Delta t}/_2) = \underline{Y}_{time}(t_f) + c({}^{\Delta t}/_2)^m + \ldots$

if m = 2 $\underline{Y}_{true} = {}^4/_3\underline{Y}(t_f;\ 0.05) - {}^1/_3\underline{Y}(t_f;\ 0.1)$

c is approximately the same is both equations:

For example: $\frac{1}{6}\frac{\partial^3 f}{\partial t^3}\Big|_{Y_0}(\Delta t)^3$

Romberg Extrapolation is Richardson Extrapolation Applied to Integrals

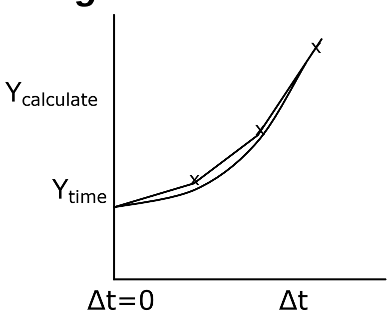


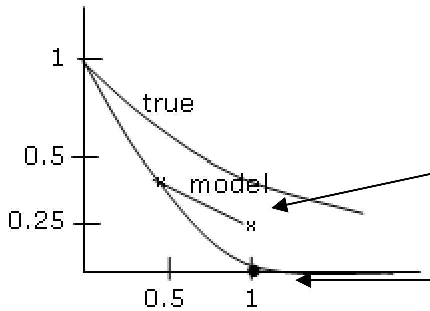
Figure 3. Diagram of Romberg Extrapolation on an increasing function.

Numerical Instability

Example uses Explicit Euler

$\frac{dy}{dt} = -y \qquad y(t_0) = 1 \qquad y_{true} = e^{-t}$

$Y_1 = Y_0 + \Delta t * F(Y_0)$



$Y_2 = Y_1 + \Delta t * F(Y_1)$

$\Delta t = 0.5 \quad (.)$

$Y_1 = Y_0 + 0.5 * F(Y_0)$

$= 1 + 0.5 * -1 = 0.5$

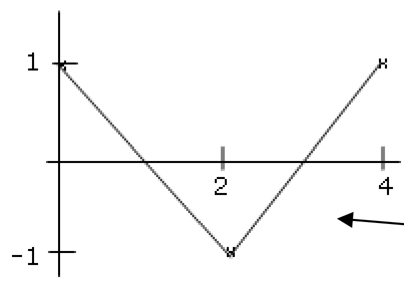
$Y_2 = 0.5 + 0.5 * -0.5 = 0.25$

$\Delta t = 1.0 \quad (0)$

$Y_1 = 0$

$Y_2 = 0$

Figure 4. Graphs of function's true and model values.



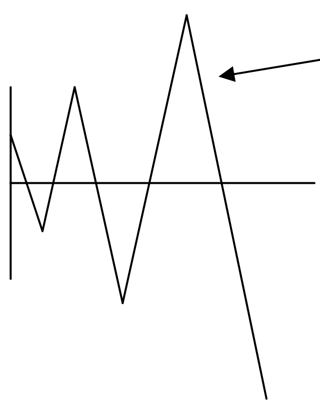
$\Delta t = 2.0 \quad (x)$

$Y_1 = -1 \qquad Y_2 = 1$

Figure 5. Difference between true and model values.

$\Delta t = 3.0$

$Y_1 = -2 \qquad Y_2 = 4$



$\left\| \underline{\underline{I}} + \Delta t \underline{\underline{J}} \right\|$ needs to be between -1 and 1 for stability

$Y_0 = \ldots\ldots\ldots\ldots$

$Y_1 = \ldots\ldots\ldots\ldots$

$Y_2 = \ldots\ldots\ldots\ldots + \left(\underline{\underline{I}} + \Delta t \underline{\underline{J}}\right)^N \underline{G}(Y_0)$ stability

$\overline{J} = \frac{\partial \underline{G}}{\partial \underline{Y}}$ True slope +δ

Figure 6. Growth of instability.

Stability

$-1 \le \lambda_i$ of $\left(\underline{\underline{I}} + \Delta t \underline{\underline{J}}\right) \le +1$

Numerical Stability (For Explicit Methods):

$||\underline{\underline{I}} + \Delta t\, \underline{\underline{J}}\,|| \leq 1$

* In Beers' textbook… implicit/explicit averaging

Stiffness if $(t_f - t_0) > \text{Max } N_{timesteps}$ ← adding big & small

max Δt
we can ← (accuracy δ) — not a huge deal
tolerate

($||\underline{\underline{I}} + \Delta t\, \underline{\underline{J}}\,|| \leq 1$) → very important

$$\underline{Y} = \begin{pmatrix} \text{major} \\ \text{minor} \end{pmatrix}$$

{λ} of $\underline{\underline{J}}$ slow λ → major time scales
fast λ → reactive intermediates

$$\frac{(t_f - t_0) \sim O(1/\lambda_{slow})}{(\max \Delta t) \sim O(1/\lambda_{fast})} \Rightarrow \frac{\lambda_{fast}}{\lambda_{slow}}$$

If too stiff, you cannot use explicit methods and must turn to implicit methods such as *Trapezoid*. To keep stable, keep Δt small. But cannot go too small in Δt: major stays the same if Δt< eps.