Numerical Solutions to Differential Equations

Lecture Notes #10 — Stiff ODEs – Multiscale Phenomena

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2015

Outline

Consider the simple ODE:

$$y'(t) = \lambda y(t) + \sin(t), \quad y(0) = y_0,$$

which has the solution

$$y(t) = \left[y_0 + \frac{1}{1+\lambda^2} \right] e^{\lambda t} - \frac{1}{1+\lambda^2} \cos(t) - \frac{\lambda}{1+\lambda^2} \sin(t).$$

If $Re(\lambda)$ is negative, then after some finite time (say T_c) the solution is pretty much independent of the initial conditions:

$$y(t) = -\frac{1}{1+\lambda^2} \cos(t) - \frac{\lambda}{1+\lambda^2} \sin(t), \quad t > T_c,$$

(the dependence on the initial condition is exponentially small).

Stiff ODEs: Introduction, I.ii

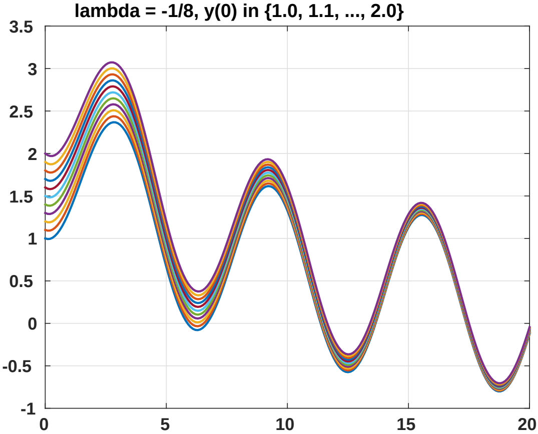


Figure: Illustration of how rapidly different initial conditions converge to the "forced oscillation."

Our solution

$$y(t) = -\frac{1}{1+\lambda^2}\cos(t) - \frac{\lambda}{1+\lambda^2}\sin(t), \quad t > T_c,$$

is a very well-behaved 2π-periodic function.

- The larger (in magnitude) the negative real part of λ is, the faster we settle into this solution.
- If for instance $\lambda = -1000$, then after 0.01s the size of $e^{\lambda t}$ is 0.000045. After 0.1s it is 10^{-44}...
- We may think (but oh how wrong we would be) that a numerical method for this would require a step-size which resolves the periodic part of the solution, say $h = \frac{2\pi}{63} \approx 0.1$ should do the trick!?!

We have forgotten about stability!

Recall that $\mathbf{h}\lambda$ must be inside the stability region!!!

Depending on what method we are using, this may impose a very restrictive step-size — assuming λ is real and negative we get the following:

Method	**Stability Interval**	**Step-size**
Explicit Euler	$-2 \leq h\lambda \leq 0$	$h < 2/\|\lambda\|$
Implicit Euler	$-\infty \leq h\lambda \leq 0$	no restriction
RK (2nd order, explicit)	$-2 \leq h\lambda \leq 0$	$h < 2/\|\lambda\|$
RK (3rd order, explicit)	$-2.5 \leq h\lambda \leq 0$	$h < 2.5/\|\lambda\|$
RK (4th order, explicit)	$-2.785 \leq h\lambda \leq 0$	$h < 2.785/\|\lambda\|$

Hence, if we used an explicit 4th-order RK-method we would need a step-size smaller than $h < 0.0027$ — which means more than 2250 points per 2π-period.

A-ha!!! Now we see why we need to care about the size of the stability regions!!!

Pseudo-Definition #1: **Stiffness**

Stiffness occurs when some component(s) of the solution decay much more rapidly than others.

Real Sources of Stiffness — "Stiffness Happens!"

- Chemically reacting systems — some reactions are very fast, others are slower.
- Computational Fluid Dynamics
  - Book: "Fundamentals of CFD"
- Interacting Particle Systems
  - Article: Implicit-Explicit Schemes

For an $(n \times n)$-system

$$\tilde{\mathbf{y}}'(t) = A\tilde{\mathbf{y}}(t) + \tilde{\phi}(t),$$

the solution is of the form

$$\tilde{\mathbf{y}}(t) = \underbrace{\sum_{k=1}^{n} \kappa_k e^{\lambda_k t} \tilde{\mathbf{v}}_k}_{\text{Transient Solution}} + \underbrace{\Psi(t)}_{\text{Steady-state}},$$

where κ_k are constants used to satisfy the initial conditions, $\tilde{\mathbf{v}}$ the eigenvectors of A and λ_k the eigenvalues of A.

If $Re(\lambda_k) < 0 \ \forall k$ then the system settles into the **steady-state solution** $\Psi(t)$ after the exponential decay of the **transient solution**.

If we order the eigenvalues so that

$$|Re(\lambda_1)| \geq |Re(\lambda_2) \geq \cdots \geq |Re(\lambda_n)|,$$

then λ_1 corresponds to the fastest, and λ_n to the slowest transient.

- It is somewhat counter-intuitive that the part of the solution which decays the fastest will impose the most stringent step-size restriction due to stability concerns.

The **stiffness ratio** (*c.f.* condition number)

$$\frac{|Re(\lambda_1)|}{|Re(\lambda_n)|}$$

is an intrinsic measure of how “resistant” the problem is to numerical solution (from our point of view). Or, rather, a measure of multi-scale behavior.

The stiffness ratio really tells us how hard we have to work to solve the system numerically:

λ_1 Imposes the step-size restriction ($h < \mathcal{C}/|Re(\lambda_1)|$).

λ_n Tells us for how long a time we have to compute the solution in order to reach steady-state (this is usually what we are interested in — long-time behavior.)
Since the slowest transient decays as $e^{-|Re(\lambda_n)|t}$ we must compute until $t > T_c$ where

$$T_c \sim \left| \frac{\ln(TOL)}{Re(\lambda_n)} \right|,$$

and TOL is the requirement on the decay of the transient solution (this will depend on your application, maybe 10^{-8}???)

Since the number of steps is inversely proportional to the step-size h, we get the total work as:

$$\frac{1}{h} \cdot \left| \frac{\ln(TOL)}{Re(\lambda_n)} \right| = \frac{|Re(\lambda_1)|}{\mathcal{C}} \cdot \left| \frac{\ln(TOL)}{Re(\lambda_n)} \right| = \mathcal{C}^* \frac{|Re(\lambda_1)|}{|Re(\lambda_n)|}.$$

The bottom line: **The larger the stiffness ratio, the more work we (our computer) have to do!**

Pseudo-Definition #2:

A linear coefficient system is stiff if all of its eigenvalues have negative real part and the stiffness ratio is large.

Pseudo-Definition #3:

Stiffness occurs when stability requirements, rather than those of accuracy constrain the step-length.

Pseudo-Definition #4:

A system is said to be stiff in a given interval of t if in that interval the neighboring solution curves approach the solution curve at a rate which is very large in comparison with the rate at which the solution varies in that interval.

Pseudo-Definition #5:

If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use a — in a certain interval of integration — step-length which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be stiff in that interval.

Each pseudo-definition represents a different point of view, and is useful/meaningful in different scenarios.

At this point we have a good idea what stiffness means, and some of its sources. The question that begs to be asked is: **what are we going to do about stiffness???**

We have some options:

1. Give up and go home.
2. Pick a small h, start the computer, and come back in 3 weeks.
3. Think some more.

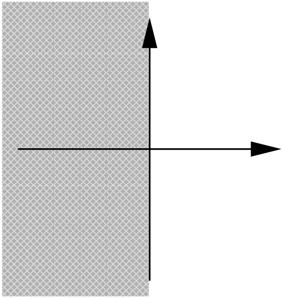
Take a wild guess at what route we are going?!?

Dealing with Stiffness: Stability Regions

- Clearly we are going to have to pay even closer attention to the size of the stability regions.
- Since the stability regions for **explicit** methods tend to be very limited, it is very likely we are going to have to take a closer look at some implicit methods.
- First, we introduce some additional stability definitions that are needed in the context of stiffness.

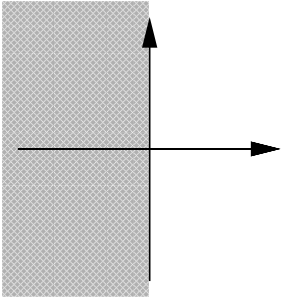
Definition (A-stability)

A method is said to be **A-stable** if its the region of absolute stability contains the left-half-plane: $\mathcal{R}_A \supseteq \{\widehat{h} : Re(\widehat{h}) < 0\}$.



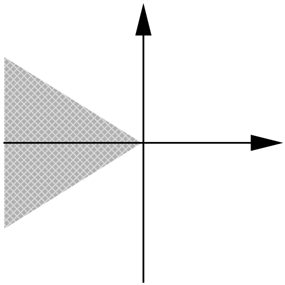
Definition (L-stability)

A **one-step** method is said to be **L-stable**, if it is **A-stable** — $\mathcal{R}_A \supseteq \{\widehat{h} : Re(\widehat{h}) < 0\}$, and **in addition**, when applied to the test equation $y'(t) = \lambda y(t)$, $Re(\lambda) < 0$, it yields $y_{n+1} = R(\widehat{h})y_n$, where $|R(\widehat{h})| \to 0$ as $Re(\widehat{h}) \to -\infty$.



Definition (A(α)-stability)

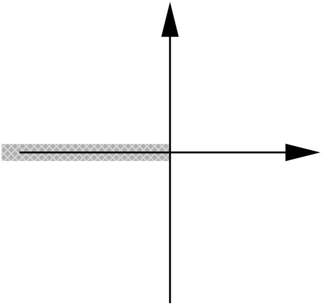
A method is said to be **A(α)-stable**, $\alpha \in (0, \pi/2)$ if $\mathcal{R}_A \supseteq \{\widehat{h} : -\alpha < \pi - \arg(\widehat{h}) < \alpha\}$; it is said to be **A(0)-stable** if it is A(α)-stable for some $\alpha \in (0, \pi/2)$.



Linear Stability for Stiff Problems: A_0-stability

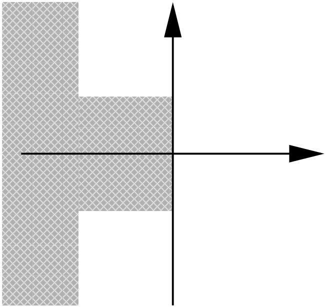
Definition (A_0-stability)

A method is said to be $\mathbf{A_0}$**-stable**, if $\mathcal{R}_A \supseteq \{\widehat{h} : Re(\widehat{h}) < 0, Im(\widehat{h}) = 0\}$.



Definition (Stiff stability)

A method is said to be **stiffly stable**, if $\mathcal{R}_A \supseteq R_1 \cup R_2$ where $R_1 = \{\widehat{h} : Re(\widehat{h}) < -a\}$, and $R_2 = \{\widehat{h} : -a \leq Re(\widehat{h}) < 0, -c \leq Im(\widehat{h}) \leq c\}$ where a and c are positive real numbers.



Linear Stability for Stiff Problems: Stability Hierarchy

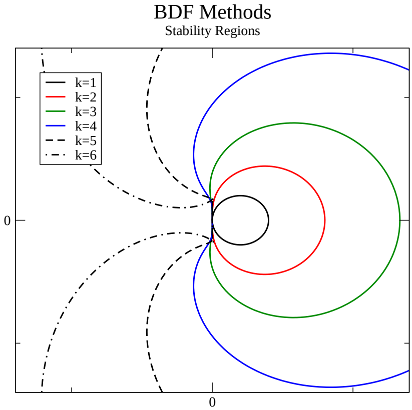
We have the following relations between the different types of stability:

L-stability $\Rightarrow$
A-stability $\Rightarrow$
Stiff stability $\Rightarrow$
A(α)-stability $\Rightarrow$
A(0)-stability $\Rightarrow$
A_0-stability

Do we really need all these classifications???

Clearly, L-stability and A-stability are very restrictive — particularly for Linear Multistep Methods. Hence we need more fine-tuned tools to classify out methods.

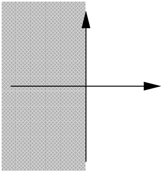
Recall the Backward Differentiation Formula methods:



Stability, Stability, Stability... II

- A(α)-stability is clearly a relaxation which fits the BDF methods.
- A(0)-stability just says that there is an α for which the method is A(α)-stable.
- A_0-stability is just concerned with real eigenvalues (λ).
- Stiff stability divides the eigenvalues into two classes — ones far away from the origin (fast transients) and ones clustered near the origin (slower transients, long-time behavior).

It can be argued that A-stability is not restrictive enough!



Consider the trapezoidal rule, which is A-stable (the region of absolute stability is exactly the left half plane):

$$y_{n+1} - y_n = \frac{h}{2}\left[f_{n+1} + f_n\right]$$

We apply trapezoidal rule to the test equation $y'(t) = Ay(t)$ where A is an $(n \times n)$-matrix with distinct eigenvalues λ_k, satisfying $Re(\lambda_k) < 0$.

After a bit of massaging (linear algebra[Math 524]), we get the following system of difference equations

$$y_{n+1} = By_n, \quad B = (I - hA/2)^{-1}(I + hA/2)$$

Let $\overline{\lambda}$ be the eigenvalue which has the largest (in absolute value) real part. It can be shown (Linear Algebra) that B must have an eigenvalue

$$\overline{\mu} = \frac{1 + h\overline{\lambda}/2}{1 - h\overline{\lambda}/2}$$

Now if $|h\overline{\lambda}|$ is large (remember, we want to take semi-long steps h), then

$$\overline{\mu} \sim -1 + \left[\frac{2}{h\overline{\lambda}}\right]^2.$$

With

$$\overline{\mu} \sim -1 + \left[\frac{2}{h\overline{\lambda}}\right]^2.$$

there will be (at least) one mode of the numerical solution which oscillates $(+, -, +, -, +, \dots)$ and **is slowly damped**.

The exact solution with respect to that mode is a quickly decaying exponential solution.

Thus A-stability is **not restrictive enough** (in some circumstances).

This is why we need the concept of **L-stability** — it deals with the behavior of the numerical method when we have $h\lambda$ far left in the complex plane.

Trapezoidal Rule is still useful!

From the previous discussion, we may think that trapezoidal rule is an unsafe method.

It is very useful indeed, but care must be taken — in order to avoid oscillations we must implement an adaptive version, where the step-size is changed to keep the error at a reasonable level.

Initially (while the transients are still "alive") the step-size will be small, but as the transients decay away, the step-size can safely be increased.

Moral of the story: Never compute with a fixed step-size, especially not for stiff problems!

Next...

The implications of stiffness on the use of Linear Multistep Methods and Runge-Kutta Methods.