November 21, 2016

Terms, Implicit Methods, Stiff Problems & BVPs

In this class you should learn:

- *Understand that consistent + stability implies convergence of an ODE method.*
- *See the backward Euler method for solving an equation, and what this has to do with Hooke's law and stiff problems.*
- *Then we'll have a group exercise on 2-point BVPs*

Next class

Optimization
Chapter 4

Next next class

Review & Misc. topics

Convergent ODEs

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\texttt{step}[\mathbf{y}_k, t, h]$$

- The ***global error*** of an approximation is:

$$\max_{k=1,\ldots,N} \|\mathbf{y}_k - \mathbf{y}^*(t_k)\|.$$

Worst approx at any time point

- A scheme `step` is ***convergent*** if global error $\to 0$ as $h \to 0$.

Convergent ODEs

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\mathtt{step}[\mathbf{y}_k, t, h]$$

- The ***global error*** of an approximation is:

$$\max_{k=1,\ldots,N} \|\mathbf{y}_k - \mathbf{y}^*(t_k)\|.$$

Worst approx at any time point

- A scheme `step` is ***convergent*** if global error $\to 0$ as $h \to 0$.

We want this!

- All schemes `step` we look at in this class are ***stable***

Just a technical notion of “super-continuous”

- The local truncation error of `step` is

$$\frac{1}{h}(\mathbf{y}^*(t+h) - \mathbf{y}^*(t)) - \mathtt{step}[\mathbf{y}^*(t), t, h]$$

Convergent ODEs

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\mathtt{step}[\mathbf{y}_k, t, h]$$

- A method is ***consistent*** if

$$\lim_{h \to 0} \mathtt{step}[\mathbf{y}_k, t, h] = \mathbf{f}(\mathbf{y}_k, t).$$

Theorem 11.2.2 If a method is consistent and stable with local truncation error $O(h^p)$, then the global error is $O(h^p)$ and the method is convergent.

Corollary If a method is consistent and stable, then it is convergent.

Forward Euler is Convergent

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\texttt{step}[\mathbf{y}_k, t, h]$$

$$\texttt{step}[\mathbf{y}_k, t, h] = \mathbf{f}(\mathbf{y}_k, t)$$ Step for FE

Stability by Prof. assertion & guarantee.

Hence, this is convergent! By THEOREM

$$\mathbf{y}^*(t+h) = \mathbf{y}^*(t) + h\tfrac{d}{dt}\mathbf{y}^*(t) + O(h^2)$$

$$\tfrac{1}{h}(\mathbf{y}^*(t+h) - \mathbf{y}^*(t)) = \tfrac{d}{dt}\mathbf{y}^*(t) + O(h) = \mathbf{f}(\mathbf{y}^*(t), t) + O(h) = \texttt{step}[\mathbf{y}^*(t), t, h] + O(h)$$

So local truncation error is O(h) and so is Global Error!

Convergent

- Fixed time window!
- EXTREMELY large constants.
- Just an asymptotic statement
  Global Error -> 0 as
  h -> 0
  in some window [0,T]

Absolute Stability

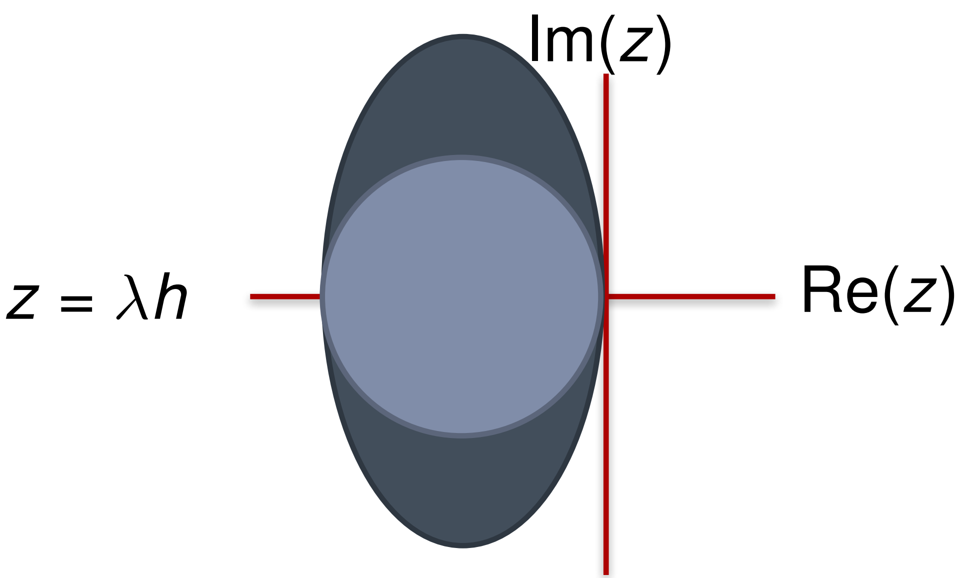
- Infinite time window
- One specific equation

$$\mathbf{y}_k \to 0 \text{ for } \frac{dy}{dt} = \lambda y \text{ when } \mathrm{Re}(\lambda) < 0$$

Hooke's Law

$$\frac{d\mathbf{y}}{dt} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix}}_{=\boldsymbol{A}} \mathbf{y}$$

$$\lambda(\boldsymbol{A}) = \pm\sqrt{k}i$$



Implicit Methods & Backward Euler

Consider our derivation of Forward Euler

$$\frac{1}{h}(\mathbf{y}(h) - \mathbf{y}(0)) \approx \dot{\mathbf{y}}/dt = \mathbf{f}(\mathbf{y}(0), 0)$$

We get the step from this idea, then iterate!

The following is just as valid!

$$\frac{1}{h}(\mathbf{y}(h) - \mathbf{y}(0)) \approx \dot{\mathbf{y}}/dt = \mathbf{f}(\mathbf{y}(h), h)$$

Use backwards in time instead of foreword approx.

i.e. the derivative holds *at the unknown future*

Using this idea requires us to *implicitly* assume that we known $\mathbf{y}(h)$ and *solve* for its value.

Implicit Methods & Backward Euler

Backward Euler
Given $\mathbf{y}_k$, solve

$$\mathbf{y}_k + h\mathbf{f}(\mathbf{x}, t_{k+1}) - \mathbf{x} = 0$$

This is, generally speaking, very hard to do!

for $\mathbf{x}$ and set $\mathbf{y}_{k+1} = \mathbf{x}$. (This is a nonlinear equation that we'll see how to solve in the next class)

Backward Euler for $\frac{d\mathbf{y}}{dt} = \boldsymbol{A}\mathbf{y}$
Given $\mathbf{y}_k$, solve

$$\mathbf{y}_k + h\boldsymbol{A}\mathbf{x} - \mathbf{x} = 0 \Leftrightarrow (\boldsymbol{I} - h\boldsymbol{A})\mathbf{x} = \mathbf{y}_k$$

This is often much easier to do!

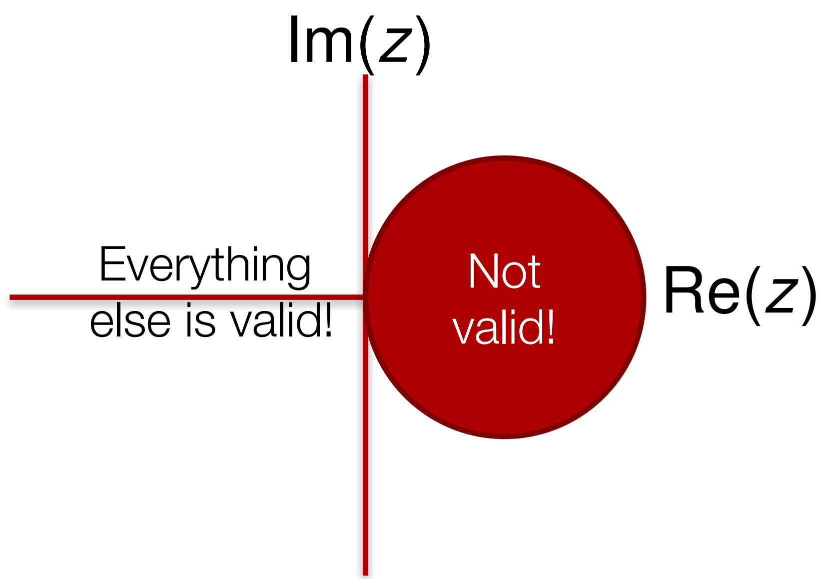
for $\mathbf{x}$ and set $\mathbf{y}_{k+1} = \mathbf{x}$. (This is a linear equation!)

Why use implicit methods?

- Much better stability properties for long time integration! e.g. The region of absolute stability for backwards Euler is

$$\left|\frac{1}{1 - h\lambda}\right| < 1$$

$z = \lambda h$



Example of Backwards Euler

Juliabox!

Why use implicit methods?

- They work better for Stiff Problems! These are problems where Foreward Euler would need a very small time-step.
  (Last 3 mins of class!)

Now!

Team exercise on BVPs!

- Organize into small reading groups
- Work through as much of the BVP notes as you can. Ask questions! There may be typos! You really do know this material at this point!