Ordinary Differential Equations

Stiffness, Stability and Implicit Methods

Sachin Shanbhag

Department of Scientific Computing
Florida State University,
Tallahassee, FL 32306.

References

- S. Chapra and R. Canale, Numerical Methods for Engineers
- A. Greenbaum and T. Chartier, Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms
- Carnahan and Wilkes, Applied Numerical Methods, University of Michigan, Class Notes, 1996.
- Burden and Faires, Numerical Analysis, 1993
- Pal, Numerical Analysis for Scientists and Engineers, 2007.
- M. Heath, Scientific Computing: An Introductory Survey
- wikipedia.org

Contents

Stability

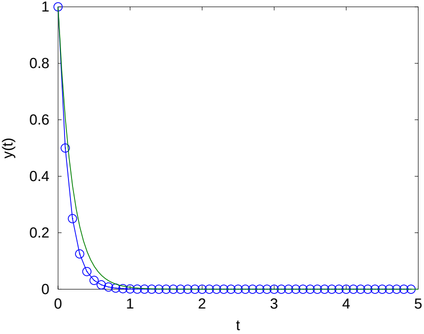
- So far, our focus has primarily been on the accuracy of methods (truncation error)
- In addition to accuracy, stability of numerical methods is an important practical concept
- A numerical method is stable if small perturbations do not cause solutions to diverge away without bound
- Test equation method is often used to assess stability
- Often provides the same insight into the stability of a method, as other more complex analyses.

Motivating Example

- Consider a simple IVP

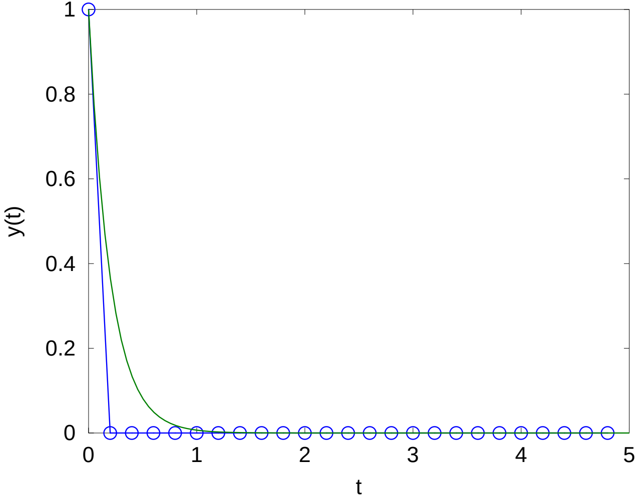
$$y' = -5y, \quad y(0) = 1$$

- The analytical solution to this ODE is $y(t) = \exp(-5t)$.
- Let us use Euler's method to solve this ODE with a step size of $h = 0.1$



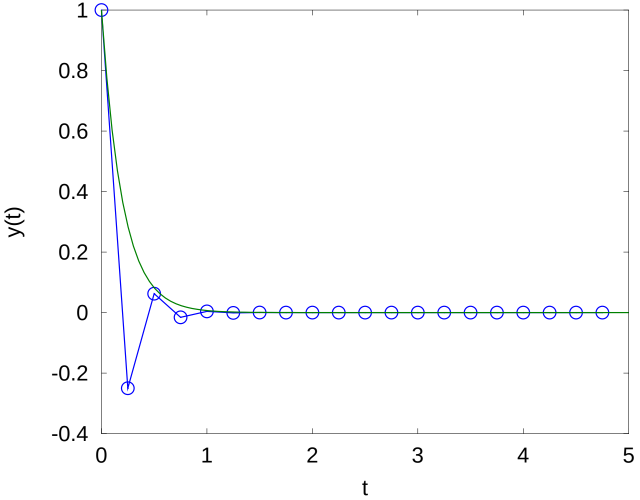
Motivating Example

Repeat with $h = 0.2$



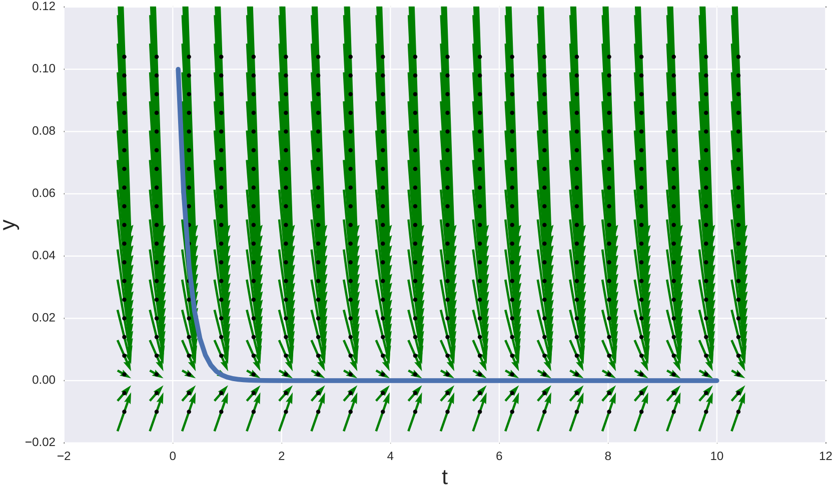
Motivating Example

Repeat with $h = 0.25$



Can you explain what you observe?

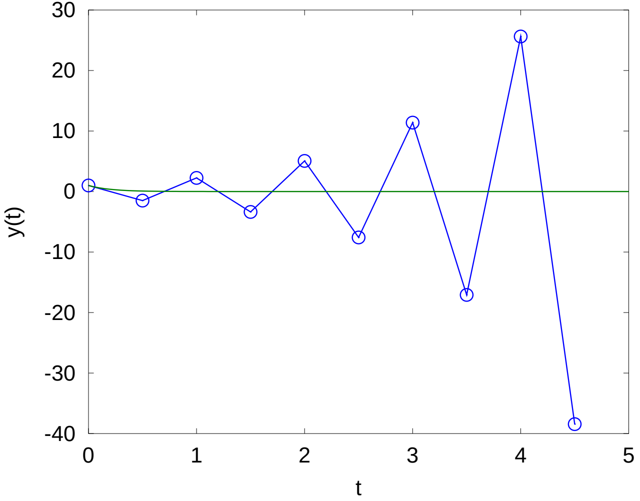
Motivating Example



blue line $=$ true solution, arrows $= f(t, y)$

Motivating Example

Repeat with $h = 0.5$ (very different y-scale)



What is happening?

- The method overshoots the analytical solution at each time step since it uses the slope at the beginning of the interval
- If the computed $y_n > 0$, then $f_n = f(t_n, y_n) = -5y_n$ implies a negative slope (for $h = 0.1$), and the method does relatively alright
- When a computed $y_n < 0$, then $f_n = f(t_n, y_n) = -5y_n$ implies a positive slope (for $h = 0.25$), and the method may oscillate
- Beyond a certain h, the oscillations may increase without bound (for $h = 0.50$).
- We would like to understand this behavior better, by using more general tools

Stability: Test Equation Method

- For complex λ and real y_0

$$y' = \lambda y, \quad y(0) = y_0$$

- Exact Solution

$$y(t) = y_0 e^{\lambda t}$$

- Behavior depends on value of λ
- Test how the numerical method responds to this equation.

Test Equation Method

- Characteristics of the exact solution

$$\text{damped} \quad Re(\lambda) < 0 \quad \Rightarrow \quad \lim_{t\to\infty} |y(t)| \to 0$$

$$\text{oscillatory} \quad Re(\lambda) = 0 \quad \Rightarrow \quad \lim_{t\to\infty} |y(t)| \to y_0$$

$$\text{unbounded} \quad Re(\lambda) > 0 \quad \Rightarrow \quad \lim_{t\to\infty} |y(t)| \to \infty$$

- Want numerical scheme to mimic this behavior especially for $\lambda <= 0$

Test Equation Method

- The region of absolute stability of a numerical scheme is defined as the region on the complex plane defined by $h\lambda$ for which the solution to the numerical scheme remains bounded
- If bounded in the left-hand plane $Re(h\lambda) < 0$, then, the method is called *A-Stable*
- Right-hand plane behavior depends on the application: stable, unstable, doesn't matter

Forward Euler

- Stability via test equation

$$y_n = y_{n-1} + h\lambda y_{n-1}$$

$$\frac{y_n}{y_{n-1}} = 1 + h\lambda$$

- Stable if:

$$|1 + h\lambda| \leq 1$$

That is:

$$(h\lambda - (-1))^2 \leq 1^2$$

- circle of unit radius centered at (-1,0) ($(x-a)^2 \leq r^2$)
- Not A-Stable

Backward Euler

- Stability via test equation

$$y_n = y_{n-1} + h\lambda y_n$$

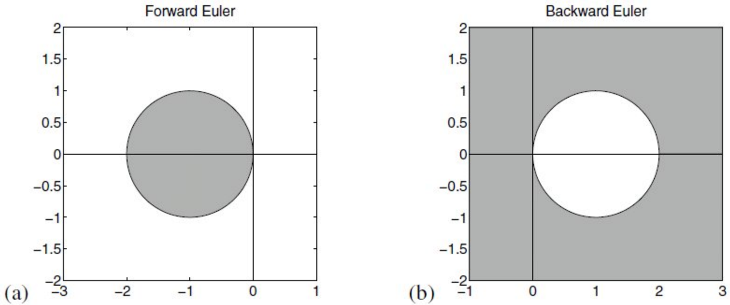
$$\frac{y_n}{y_{n-1}} = \frac{1}{1 - h\lambda}$$

- Stable if:

$$|1 - h\lambda| \geq 1$$

- outside a circle of unit radius centered at (1,0)
- A-Stable

Region of Stability



Motivating Example

- Recall the example considered earlier ($\lambda = -5$)

$$y' = -5y, \quad y(0) = 1$$

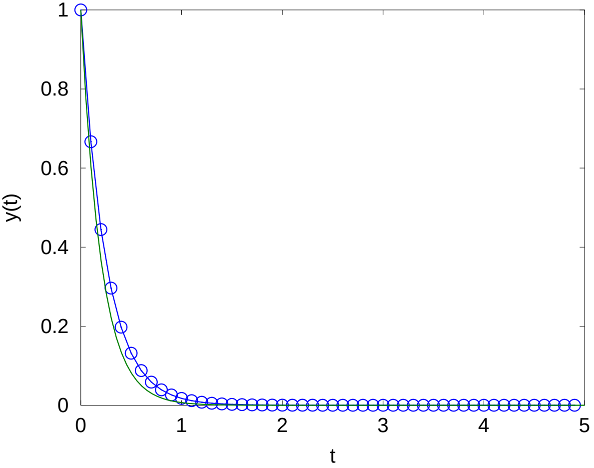
- For stability: forward Euler $|1 + \lambda h| = |1 - 5h| \leq 1$

$\mathbf{h}$	$\mathbf{|1 - 5h|}$	**behavior**
0.10	0.5	stable
0.20	0.0	stable
0.25	0.25	oscillatory, stable
0.50	1.5	oscillatory, unstable

- You can easily confirm that the maximum h for which the method is stable is $h_{\text{stability}} = 0.4$.
- This is larger than the step size required for accuracy h_{accuracy}, since even with $h = 0.1$ we do not get a very accurate solution

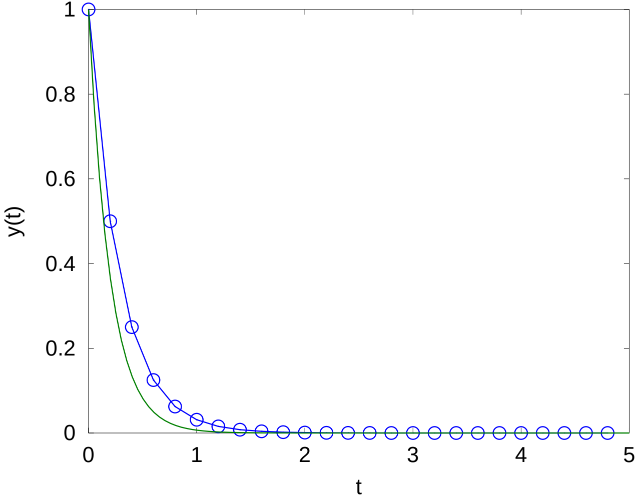
Backward Euler

- The stability of backward Euler can be easily demonstrated by working out the same problem
- With $h = 0.1$



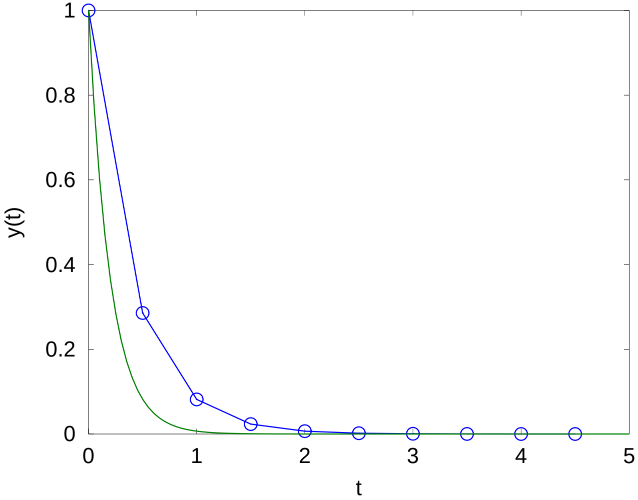
Motivating Example

Repeat with $h = 0.2$ using Backward Euler



Motivating Example

Repeat with $h = 0.5$ using Backward Euler



Stiffness

- Both accuracy and stability prefer small step-size h
- Normally h required for accuracy is smaller than the step-size required for stability, and is a bigger concern

$$h_{\text{accuracy}} \ll h_{\text{stability}}$$

  - accuracy: discrete solution approaches "true" continuous solution
  - stability: discrete solution doesn't blow up!
- For a stiff problem, the opposite is true

$$h_{\text{accuracy}} \gg h_{\text{stability}}$$

stability is a bigger concern

Stiffness

- It is a subtle, important and somewhat loose concept and depends on:
  - the actual initial value problem
  - interval of integration - size and location
  - accuracy requirements
  - the region of absolute stability of the method
- Physically: a process that has multiple disparate time-scales (weather modeling), or a timescale that is very short compared to the interval of integration
- Mathematically, $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ is stiff if the Jacobian matrix $\mathbf{J_f}$ has eigenvalues which are disparate in magnitude
- Usually, *we are after a smooth and slowly varying solution, when perturbations to the data have rapidly varying solutions*.

Example

- Scalar example

$$y' = -\sin(t); \quad y(0) = 1$$

- Solution

$$y(t) = \cos(t)$$

- Add the solution back into the problem

$$y' = \lambda[y - \cos(t)] - \sin(t); \quad y(t_0) = y_0$$

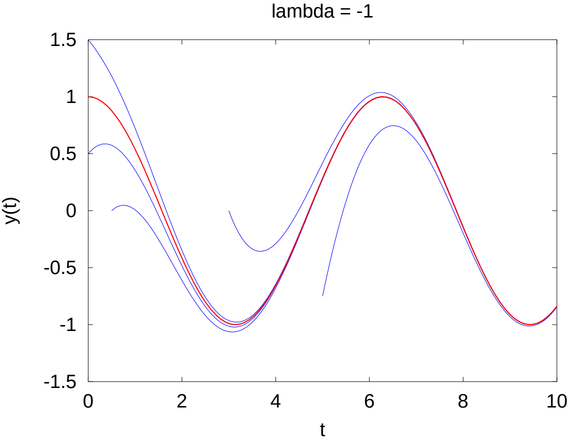
- Has the solution

$$y(t) = e^{\lambda(t-t_0)}[y_0 - \cos(t_0)] + \cos(t)$$

- Transient decays for $Re(\lambda) < 0$

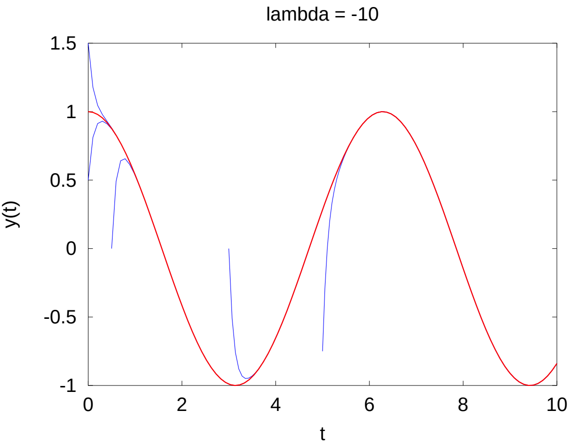
Example

- Different initial conditions (t_0, y_0)
- Set $\lambda = -1$.

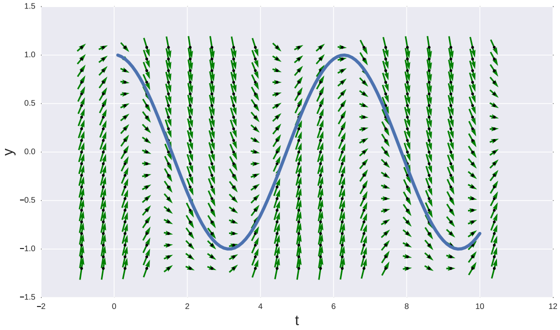


Example

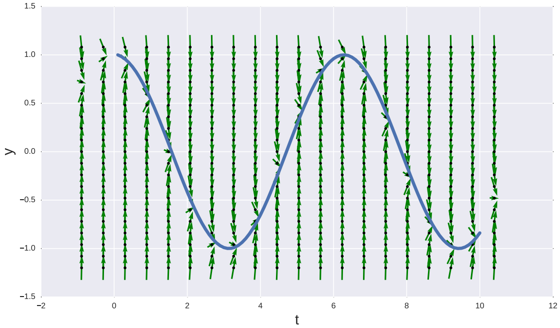
- Different initial conditions (t_0, y_0)
- Set $\lambda = -10$.



- very fast convergence can cause stiffness



$$\lambda = -1$$



$$\lambda = -10$$

Numerical Solution

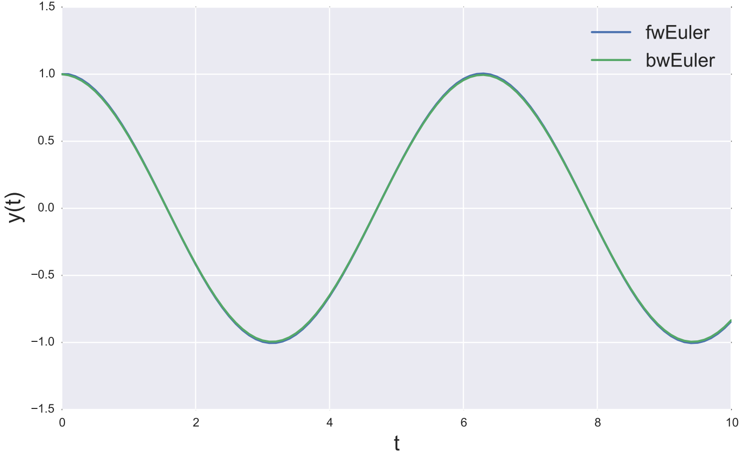
- In numerical solutions the perturbations occur naturally due to truncation errors
- Consider the previous problem:

$$y' = \lambda[y - \cos(t)] - \sin(t); \quad y(0) = 1$$

- Let me solve it using forward and backward Euler, with $\lambda = -10$, and changing step-size h
- Numerical solution is controlled by the fast-transient (stability depends on λ), while accuracy depends on the slow-varying part.
- Note that both forward and backward Euler are $\mathcal{O}(h)$

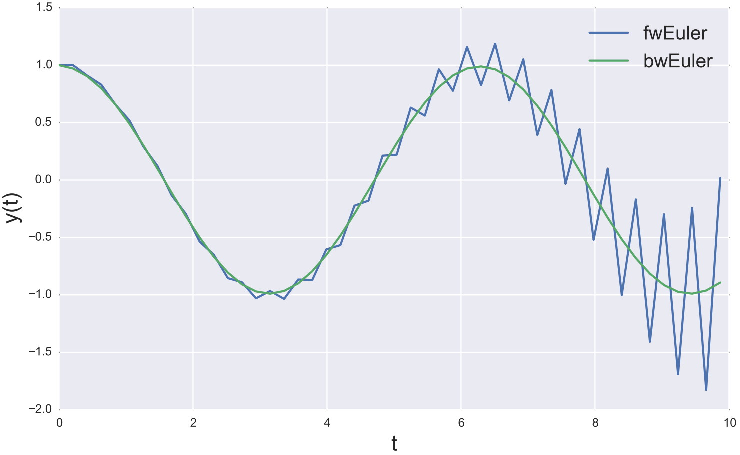
$$h\lambda = -1$$

$$h = 0.1$$



$h\lambda = -2.1$

$h = 0.21$



Manifestation of usual instability $|1 + h\lambda| > 1$.

Stiffness

- Generally implicit methods work well.
- There are several higher order implicit methods available, including Gear's backward difference formulas, implicit Runge Kutta methods, etc.
- Stiff equations need not be "hard" to solve once the appropriate method is chosen
- Much more common in systems of equations with disparate time scales. In the scalar example, the two timescales came from λ and $\sin(t)$
- As Moler puts it "Stiffness is an efficiency issue. If we weren't concerned with how much time a computation takes, we wouldn't be concerned about stiffness. Nonstiff methods can solve stiff problems; they just take a long time to do it."