Ordinary Differential Equations 3

CS 111: Introduction to Computational Science

Spring 2019 Lecture #16

Ziad Matni, Ph.D.

Administrative

- Homework #7
  - Newly issued. Due **WEDNESDAY (6/5) @ 6:00 pm**

- **Prof. Matni has office hours NEXT WEEK**
  - **Monday, June 10th 2:30 pm – 4:00 pm**

Final Exam

- In this classroom on **Wednesday, June 12th at 12:00 PM**
- Arrive 10-15 minutes early
- Pre-arranged seating

- Material to study: Everything!
- Allowed in the exam:
  1 sheet of paper (8.5"x11") both sides ok

In-Class Exercise

The error function $\mathrm{erf}(x)$ is usually defined by an integral,

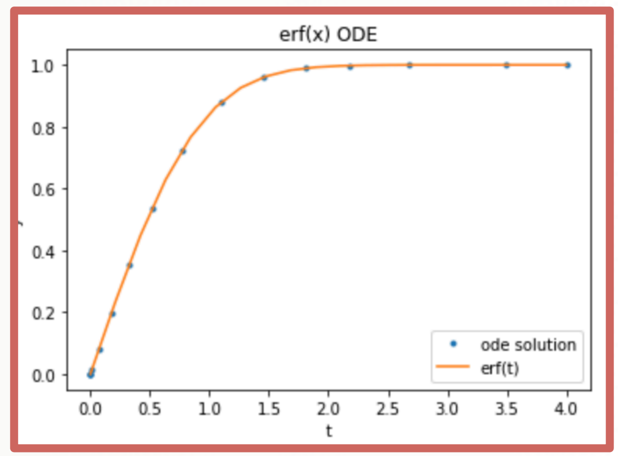
$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2} dx,$$

but it can also be defined as the solution to the differential equation

$$y'(x) = \frac{2}{\sqrt{\pi}} e^{-x^2},$$
$$y(0) = 0.$$

- Use **solve_ivp()** to solve this differential equation on the interval $0 \le x \le 2$

- Compare the results with the built-in **numpy** function **np.erf(x)** by overlapping their plots



When ODE Solvers Become Confused

- In certain cases, the solution of an ODE starts to vary slowly with "nearby" solutions that vary rapidly

- Looks like "instability" in the solution…
- This problem is described as "**stiff**"

Solution to "Stiffness"

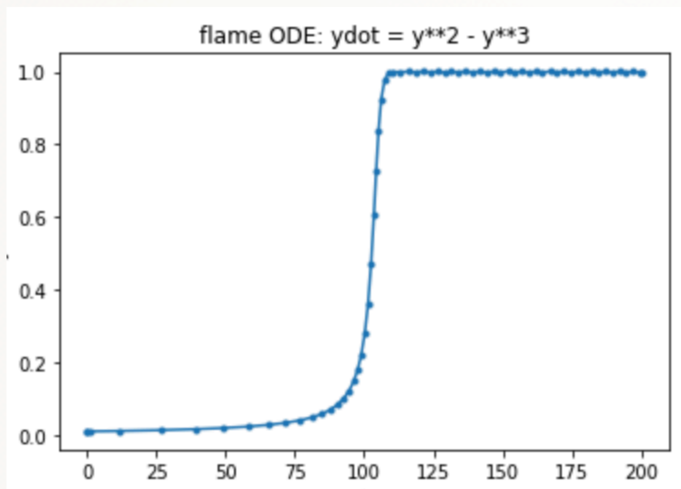
- The numerical method must thus take small steps to get good results
- What's the problem with making steps smaller??
  - They take up more resources and make the solution take a longer time
- What's a better alternative??
  - Use a better algorithm/method

The Flame Example

- When you light a match, the ball of flame grows rapidly. Can be described as an ODE:

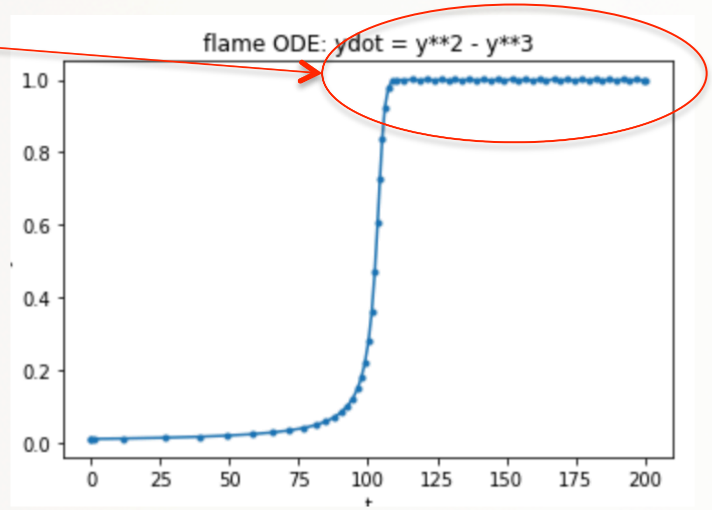
$$y' = y^2 - y^3 \qquad \text{with} \qquad y(0) = \delta\text{, a small number}$$

- Can be described graphically as $0 \le t \le 2/\delta$:



Problem?

- The problem becomes "stiff" when the solution approaches a "steady state"

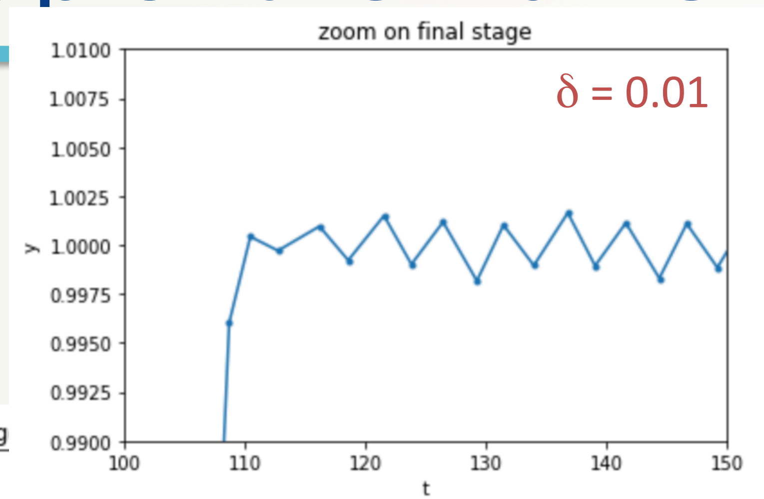


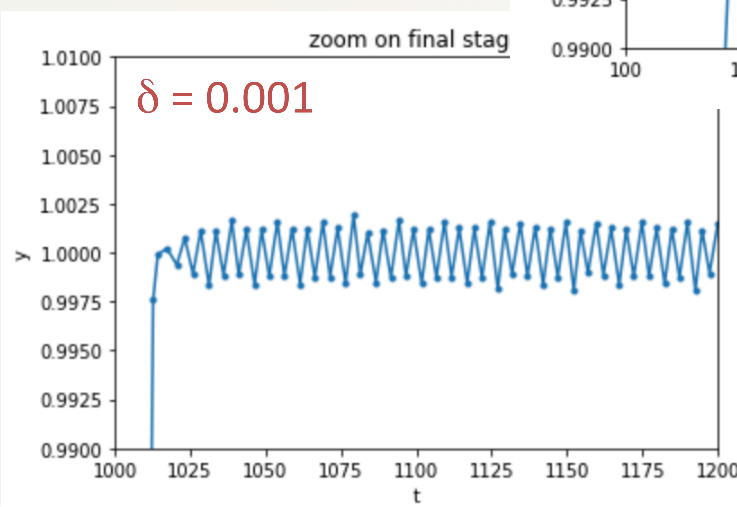
- Any solution around y = 1 (in this example) jumps around y = 1 rapidly

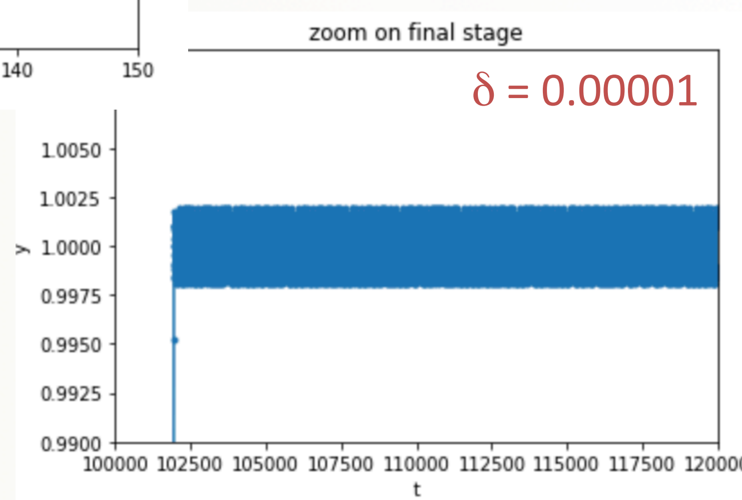
- A lot of "explicit" methods (like the Euler Method) don't work well here, but some "implicit" methods do…

Close-up of the Flame Problem

Using the RK23 method every time







The Radau Method

- The Radau solvers are **efficient *implicit*** Runge-Kutta methods that increase solutions in order
  - “order” is a measure of how well it approximates the solution

- It is especially good for fast solving at low tolerances (<1e-9) and **is a favorite of stiff problems**.

def GetPython():

Python!

return(OMG_ItsASnake)

In-Class Exercise

Consider the ODE:

$y' = -50y$ **with** $y(0) = -2$

for $0 \leq t \leq 2$

- Use **solve_ivp()** to solve this ODE using the RK23 method and plot the graph.
- Does it seem like a "stiffness problem"? Why or why not?
- Re-do with RK45 and Radau methods – describe what you see

Your To-Dos

- **Homework 7 due WEDNESDAY**

</LECTURE>