Lecture 13: Stiff IVPs

Today:

- Introduce ***stiff initial value problems***
  - What makes them challenging to solve numerically?
  - What finite difference methods are used to solve these nasty IVPs?

Stiff IVPs: A motivating problem

Today. We will continue to think about numerically solving IVPs, but we will consider a specific class of IVPs called ***stiff*** IVPs

To motivate what makes ***stiff IVPs*** tricky, consider this seemingly benign problem:

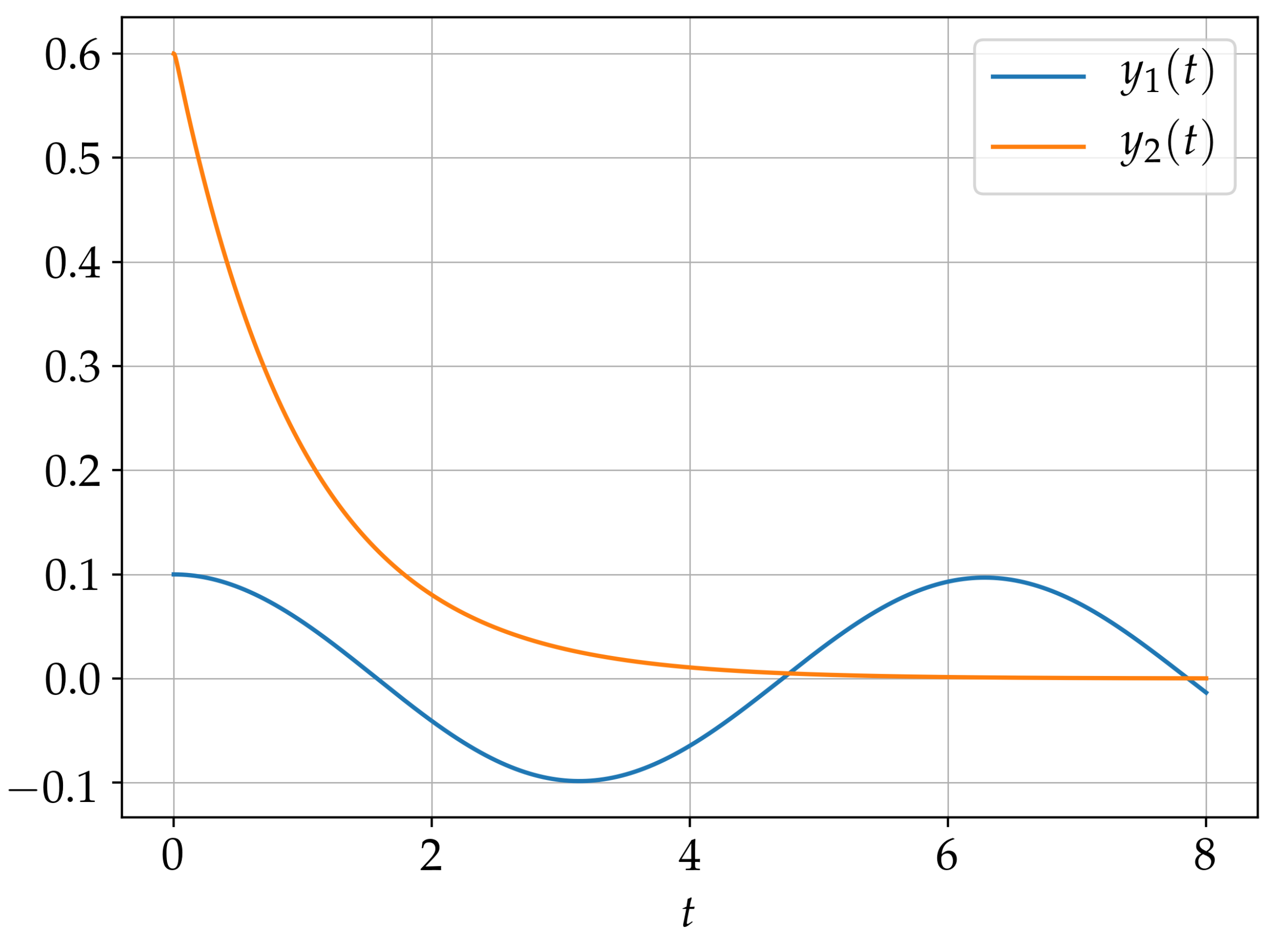
Initial conditions

$$\ddot{y}_1 + c_1\dot{y}_1 + k_1 y_1 = 0 \qquad y_1(0) = \gamma_1,\ \dot{y}_1(0) = 0,\ y_2(0) = \gamma_2,\ \dot{y}_2(0) = 0$$

$$\ddot{y}_2 + c_2\dot{y}_2 + k_2 y_2 = 0 \qquad k_1 = 1,\ c_1 = 1\times 10^{-2},\ k_2 = 100,\ c_2 = 100,\ \gamma_1 = 0.1\ \ \gamma_2 = 0.6$$

Parameter values

Here is the solution. Don't be lulled into a false sense of security…



So what could possibly be challenging about numerically solving this IVP?

Solve the pesky IVP with the trapezoid method & Heun's method

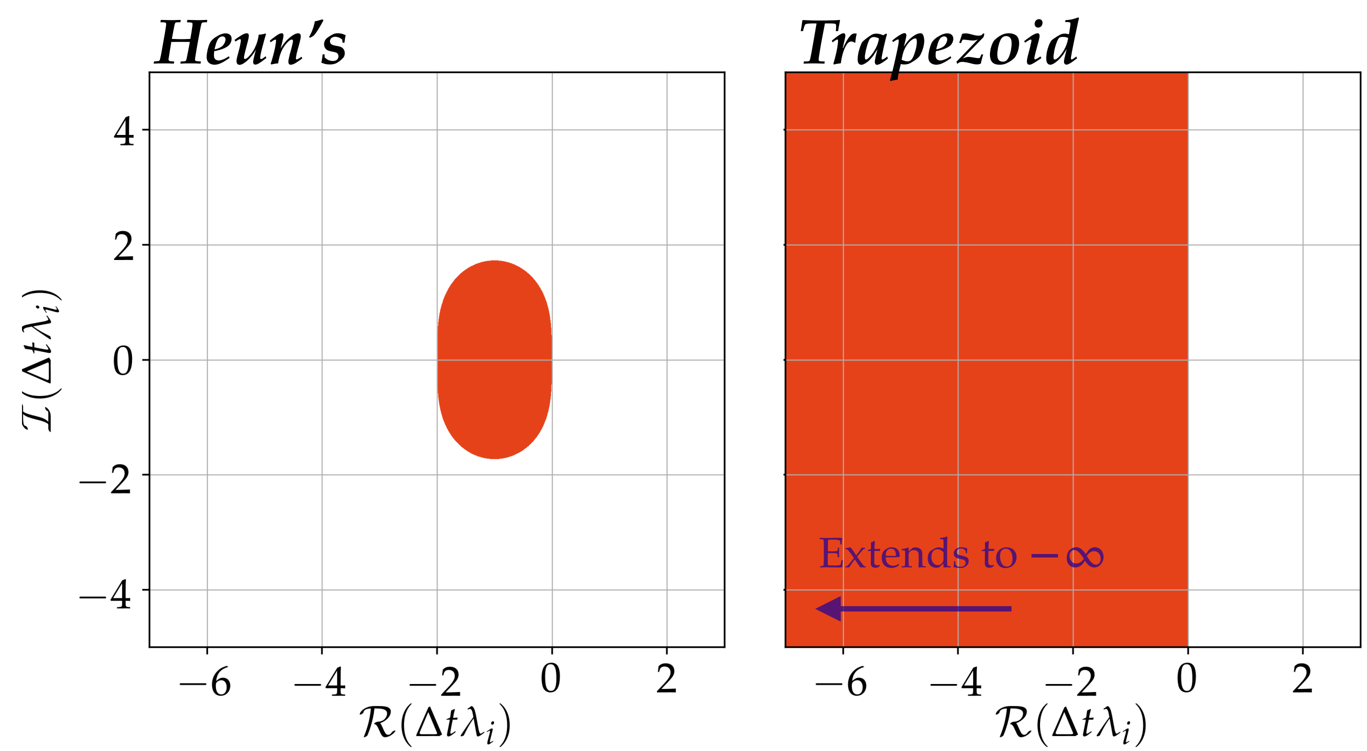
The ***error*** at $t = 8$ associated with solving the IVP via the two methods is:

Δt	Heun's method	trapezoid method
0.5	8.28×10^{48}	0.169
0.25	1.65×10^{78}	5.26×10^{-3}
0.1	1.08×10^{128}	6.40×10^{-4}
0.05	6.81×10^{146}	1.60×10^{-4}
0.025	1.023×10^{64}	4.00×10^{-5}
0.01	1.28×10^{-5}	6.41×10^{-6}

Um, what?! Some notes:

(A) Remember the stability regions for Heun's and trapezoid:
Seems like stability is playing a role here

(B) The fact that we can only get either 10^{64} or 10^{-5} error with Heun's, without wiggle room in between, is ***not comforting***



How do we understand this phenomenon?

Understanding why stiff problems are hard: Rewriting our pesky IVP in 1st order form

Notice that we can rewrite the system in 1st order form (only involving 1st derivatives in time) by defining $\mathbf{z} = [y_1, \dot{y}_1, y_2, \dot{y}_2]^T$

Then we have

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}$$
$$\boldsymbol{z}(0) = \boldsymbol{z}_0$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -k_2 & -c_2 \end{bmatrix} \qquad \boldsymbol{z}_0 = [\gamma_1, 0, \gamma_2, 0]^T$$

This conversion to 1st order form is important, because our finite difference methods were derived assuming this form (i.e., without using second derivatives).

But we can do more! Notice that we can recast this further into the form of the model problem $\dot{\boldsymbol{u}} = \boldsymbol{\Lambda}\boldsymbol{u}$ that we used in defining absolute stability. The key is the eigendecomposition of $\mathbf{A}$, $\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$:

Now define $\mathbf{u} = \mathbf{V}^{-1}\mathbf{z}$

$$\dot{\boldsymbol{z}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}\boldsymbol{z}$$
$$\Longrightarrow \quad \boldsymbol{V}^{-1}\dot{\boldsymbol{z}} = \boldsymbol{\Lambda}\boldsymbol{V}^{-1}\boldsymbol{z}$$
$$\Longrightarrow \quad \dot{\boldsymbol{u}} = \boldsymbol{\Lambda}\boldsymbol{u}$$

And since $\mathbf{V}$ is not a function of time,
$\mathbf{V}^{-1}\dot{\mathbf{z}} = \dot{(\mathbf{V}^{-1}\mathbf{z})} = \dot{\mathbf{u}}$

Notice that we also have the initial condition $\mathbf{u}(t=0) = \mathbf{V}^{-1}\mathbf{z}_0$.
Let's just call this vector $[d_1, d_2, d_3, d_4]^T$

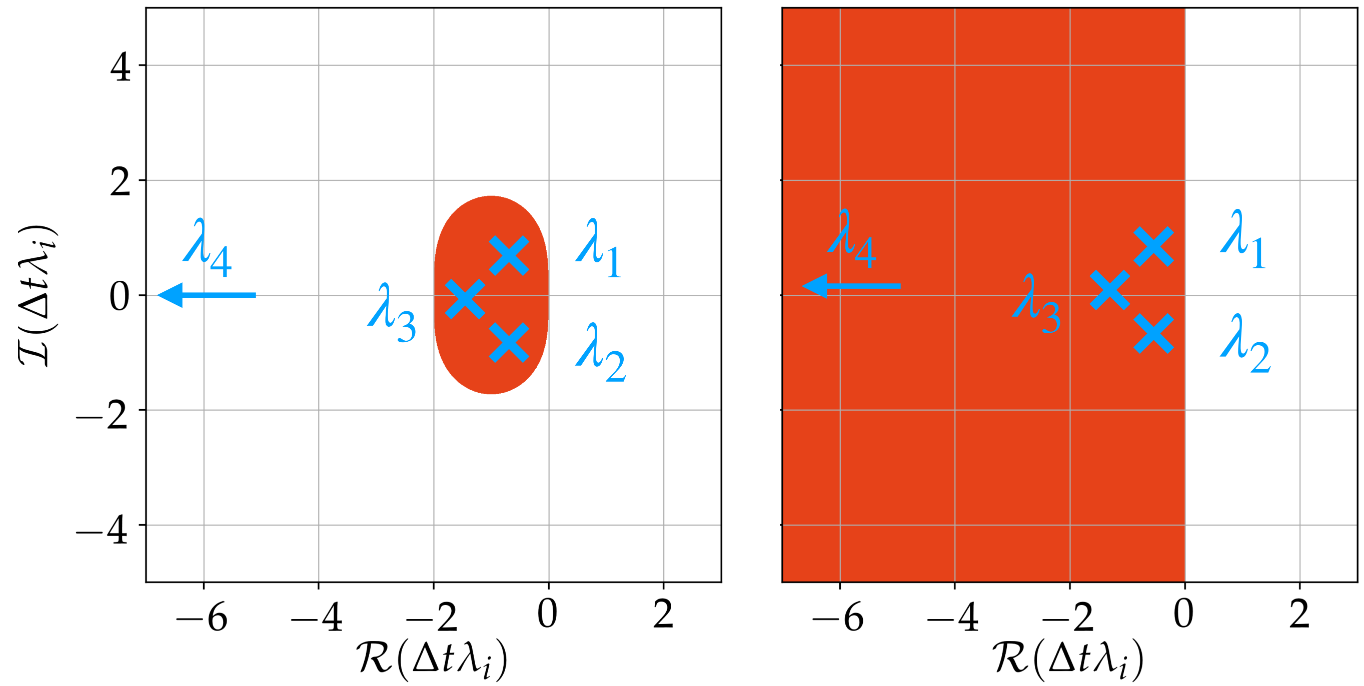
Understanding why stiff problems are hard: probing the solution to the model problem

We can compute $\mathbf{\Lambda}$ and $\mathbf{V}$ using eig(A)

From this we have that

$$\mathbf{\Lambda} = \begin{bmatrix} -0.005+i & 0 & 0 & 0 \\ 0 & -0.005-i & 0 & 0 \\ 0 & 0 & -1.01 & 0 \\ 0 & 0 & 0 & -98.99 \end{bmatrix}$$

λ_1 λ_2 λ_3 λ_4



And therefore the solution to the problem $\dot{\boldsymbol{u}} = \mathbf{\Lambda}\boldsymbol{u}$ is

Remember that $d_1, d_2, d_3,$ and d_4 are just initial condition values. So in general they can all have some nonzero value

$$\boldsymbol{u} = \begin{bmatrix} e^{(-0.005+1i)t} d_1 \\ e^{(-0.005-1i)t} d_2 \\ e^{-1.01t} d_3 \\ e^{-98.99t} d_4 \end{bmatrix}$$

This term doesn't matter for the dynamics (it decays way before the other terms), but it is imposing the stability constraint for Heun's method.

Need to pick a tiny Δt so that $\Delta t \lambda_4$ lies in the stability region.

Not true for trapezoid method. Everything is in the stability region for ***any*** Δt

Synthesize: what are stiff IVPs?

Defining property of stiff IVPs: there is a term that is ***unimportant to the dynamics*** but that imposes stability restrictions.

This can be quantified by inspecting the eigenvalues of $\mathbf{A}$:

Characterization of stiffness for IVPs

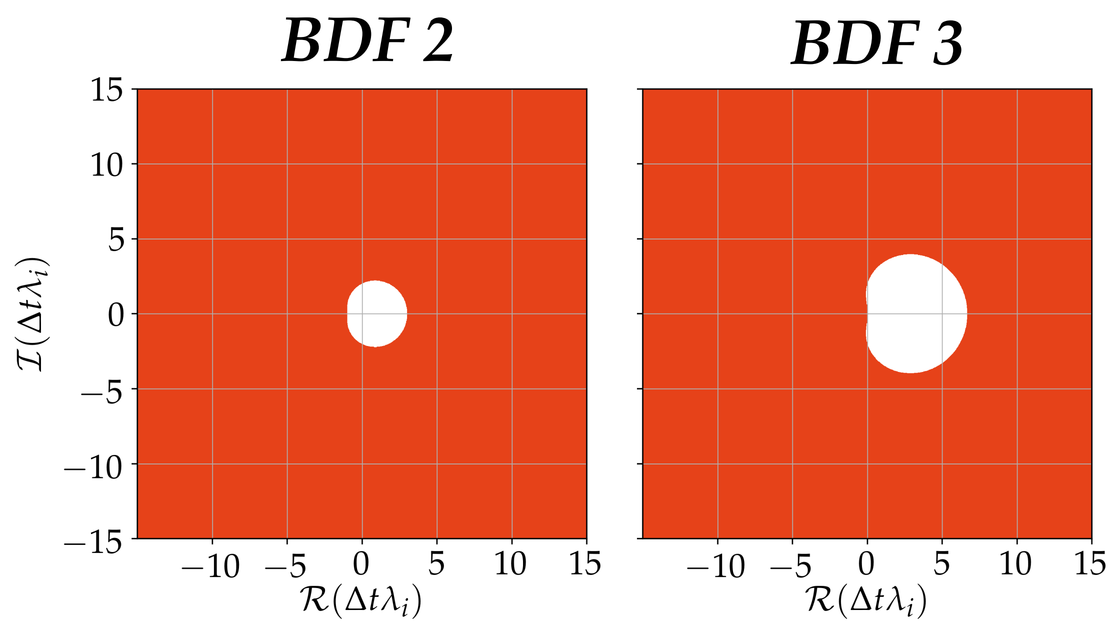
The stiffness of an IVP $\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}$ is characterized by its "stiffness ratio",

$$\mathcal{R}_s = \frac{\max_{j=1,\dots,n}(|\lambda_j|)}{\min_{j=1,\dots,n}(|\lambda_j)|} \tag{11}$$

where n is the dimension of the IVP. The IVP is called stiff if $\mathcal{R}_s \gg 1$.

Synthesize:
What implications does "stiffness" have in picking a numerical method?

How to solve stiff IVPs numerically: pick a FD method with large stability regions. The trapezoid method is a good option, but it is only 2nd order accurate. Another class of multi-step methods for stiff IVPs is the Backwards Differentiation Formulae (BDF). Here are some of these methods for different numbers of steps:



1-step BDF method (Backwards Euler):

$$u_{k+1} - u_k = \Delta t f(u_{k+1}, t_{k+1})$$

2-step BDF method:

$$3u_{k+1} - 4u_k + u_{k-1} = 2\Delta t f(u_{k+1}, t_{k+1})$$

3-step BDF method:

$$11u_{k+1} - 18u_k + 9u_{k-1} - 2u_{k-2} = 6\Delta t f(u_{k+1}, t_{k+1})$$

4-step BDF method:

$$25u_{k+1} - 48u_k + 36u_{k-1} - 16u_{k-2} + 3u_{k-2} = 12\Delta t f(u_{k+1}, t_{k+1})$$