Lecture 26

Explicit methods.
Local truncation error.
Consistency.

Then more on error and how we'll study global error!

HW5
↳ "Due" April 30 to be graded before final
↳ Otherwise May 3rd (full pts! but not graded before final)

ODEs, IVPs

$\frac{dy}{dt} = f(t, y)$

$y(0) = y_0$ given.

↓

Study Error!

Need to understand Method!

For the moment, we'll only consider *explicit methods*, those that do not depend on solving systems of equations such as the Backward or Implicit methods.

In general, a one-step method is[5]

$$y_{i+1} = y_i + h\Phi(ih, y_i; h).$$

Which we'll also write as:

$$y_+ = y + h\Phi(t, y; h)$$

to avoid the index i and make it slightly more general.

For explicit Euler, $\Phi = f(t, y)$.

The idea with Φ is that we should locally approximate the initial value problem

$\Phi(t, y; h)$ is the "method"

Cases: $\Phi(t, y; h) = f(t, y) \to$ forward Euler

$\Phi(t, y; h) = \frac{1}{2} f(t, y) + \frac{1}{2} f(t+h, y + h f(t,y))$

---

2 types of Error

- Local → is the method "Reasonable"
- Global → is the answer "right"?

↳ Local analysis is with Respect to

The idea with Φ is that we should locally approximate the initial value problem

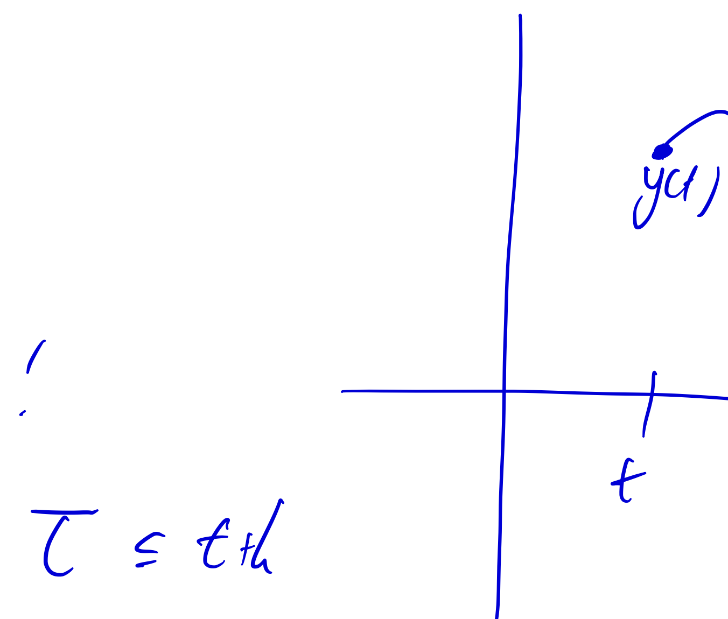
$$\frac{du}{d\tau} = f(\tau, u), \quad t \le \tau \le t+h, \quad u(t) = y$$

different function than y!

$\frac{du}{d\tau} = f(\tau, u) \quad t \le \tau \le t+h$

$u(t) = y \leftarrow$ e

$u(t) = y$
(Starts the same)
Exists for t+h time.



= true ODE Started @ y(t)

Now it's time to talk error! Here, we'll use $u(\tau)$ as the reference solution. We want to get as close as possible to this!

DEFINITION 3 *The truncation error of the method* Φ *with respect to* u *is:*

$$T(t, y; h) = \frac{1}{h}(y_+ - u(t+h)).$$

Q: Where is Φ ???

Nb: Remember $y_+ = y + h\Phi(t, y; h)$

⇓

$T(t, y; h) = \Phi(t, y; h) - \frac{1}{h}(u(t+h) - u(t))$

(Where did y go?? $y = u(t)$!)

To what Extent does Φ look like the approx. derivative $\frac{1}{h}(u(t+h) - u(t))$??

(as $h \to 0$)

$T(t, y; h) \to 0 \Rightarrow$

$\Phi(t, y; h) - \underbrace{\frac{1}{h}(u(t+h) - u(t))}_{\xrightarrow{\lim} u'(t) = f(t,y)} \Rightarrow 0$

("by construction")

- A method is consistent if $T(t, y; h) \to 0$ as $h \to 0$ for all t, y in some domain, uniformly.
- Consistency is equivalent to $\Phi(t, y; 0) = f(t, y)$.

Sometimes f is called forcing function.

- A method has *order* p if (for some vector norm) $\|T(t, y; h)\| \le Ch^p$ for some constant.

Example:

forward Euler

$\Phi(t, y; h) = \underbrace{f(t, y)}$

So does not use h!

⇒ Consistent!

because $\Phi(t, y; 0) = f(t,y)$

How to Establish Order? ~~Def~~ order 1.

See textbook.

Taylor Extrapolation

Idea: Use more derivatives from Taylor's theorem!

$y(t+h) \approx y(t) + h y'(t) + \frac{h^2}{2} y''(t) + \frac{h^3}{3!} y'''(t) + O(h^4)$

→ Turn this into a method by approximating many derivatives @ (y, t)

$\Phi(t, y; h) = \underset{y'}{f(t,y)} + \frac{h}{2} \underset{y''}{f'(t,y)} + \frac{h^2}{3!} f''(t,y)$ ⇒ order 3

Why is Taylor Extrapolation barely used??

Usually only get access to actual forcing function f, not derivatives ☹

Multi-stage methods

(Multiple f Eval per Step)

- Heun method.

$\Phi = \frac{1}{2} f(t, y) + \frac{1}{2} f(-)$

- Mid-point Euler

$\Phi = f(t + \frac{h}{2}, y + \frac{h}{2} f(t,y))$

→ Φ general parametrize Sets

Let $k_1 = f(t, y)$ = Slope @ y, t

$k_2 = f(t + \mu h, y + \mu h k_1)$

= Slope @ $t + \mu h$

Use $\Phi = \alpha_1 k_1 + \alpha_2 k_2$

i.e. $\mu = \frac{1}{2}, \alpha_2 = 1 \quad (\alpha_1 = 0) \Rightarrow$ Mid-Euler

$\mu = 1, \alpha_1 = \frac{1}{2} \quad \alpha_2 = \frac{1}{2} \to$ Heun

---

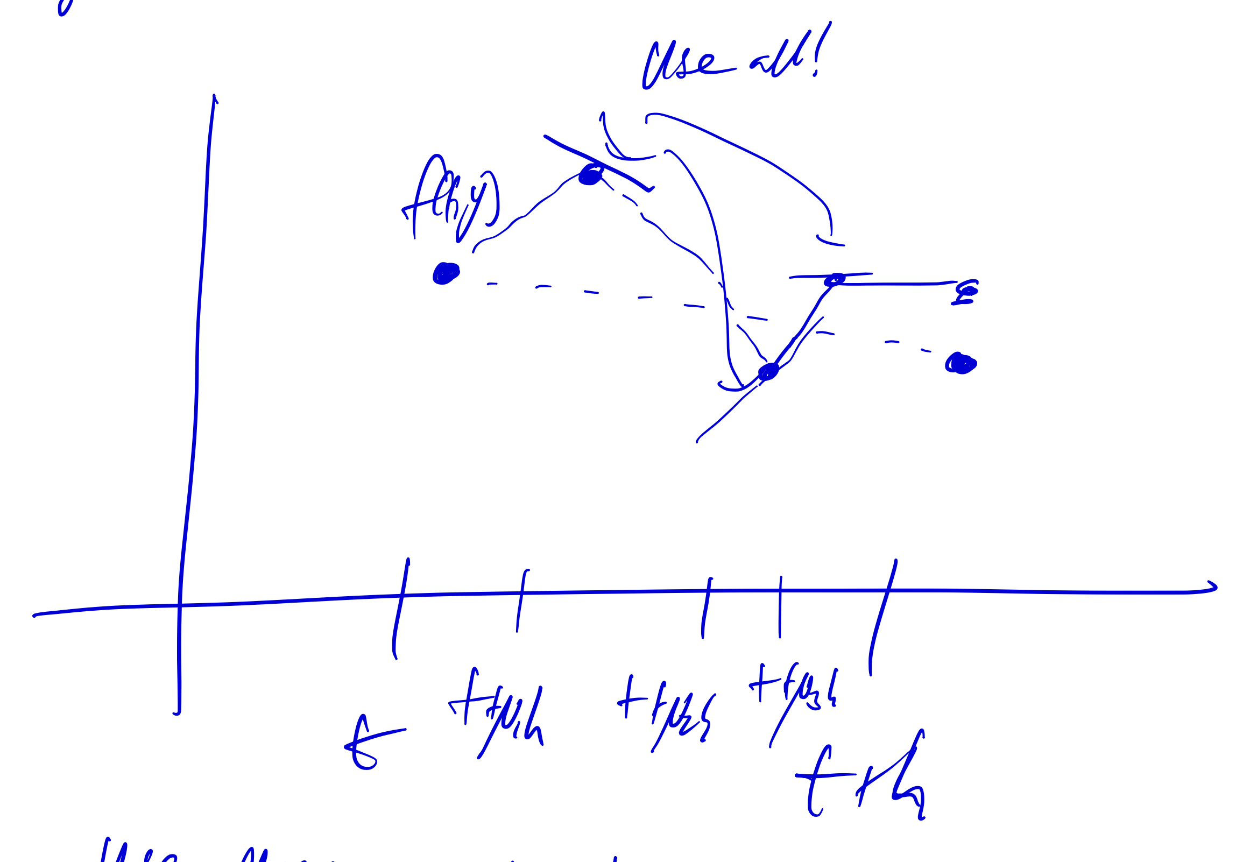
Now the game is... What needs to be true for $O(h^2)$ / order 2 method?

⋮ Lots of Algebra

$\alpha_1 + \alpha_2 = 1$

$\alpha_2 \mu = \frac{1}{2}$

Runge Kutta methods



Use many points to Eval Φ.

RUNGE-KUTTA SCHEMES

The most general scheme are the RK (Runge-Kutta) integrators. These take the previous idea to a general setting.

$$\Phi(t, \mathbf{y}; h) = \sum_{s=1}^{r} \alpha_s \mathbf{k}_s$$

$\sum \alpha_s k_s$

$$\mathbf{k}_1 = \mathbf{f}(t, \mathbf{y})$$

$$\mathbf{k}_s = \mathbf{f}(t + \mu_s, \mathbf{y} + h \sum_{j=1}^{s-1} \lambda_{sj} \mathbf{k}_j).$$

Here, we have $\boxed{\mu_s = \sum_{j=1}^{s-1} \lambda_{sj}}$ and $\boxed{\sum \alpha_s = 1.}$

$\lambda_{s,j}$ combo of prev. points use in point j.

We can get order r for $1 \le r \le 4$.

Using $s=8$ or 9, we get order 6 or 7.

For instance, the *classic* formula is order 4 with:

$$\Phi(t, \mathbf{y}; h) = \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

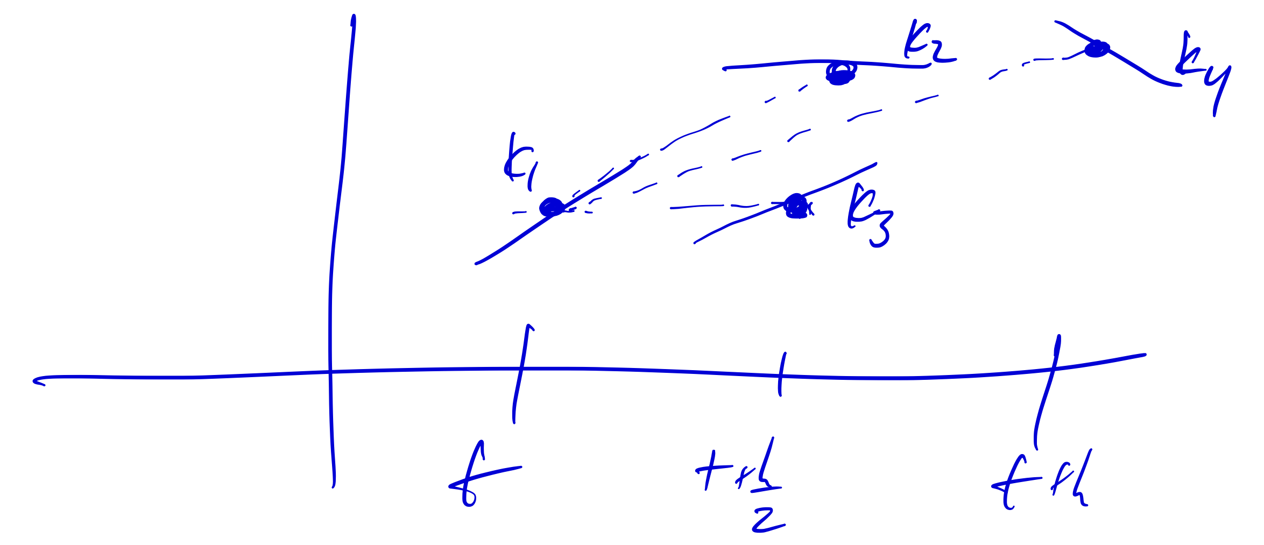
$$\boxed{\mathbf{k}_1 = \mathbf{f}(t, \mathbf{y})}$$

same as

$$\mathbf{k}_2 = \mathbf{f}(t + h/2, \mathbf{y} + h/2\mathbf{k}_1)$$

$$\mathbf{k}_3 = \mathbf{f}(t + h/2, \mathbf{y} + h/2\mathbf{k}_2)$$

$$\mathbf{k}_4 = \mathbf{f}(t + h, \mathbf{y} + h\mathbf{k}_3)$$



SOFTWARE

Matlab's ODE suite implements:[6]

1. `ode45`: 4th-order method (best general choice)

2. `ode23`: 2nd-order method
3. `ode23s`: 2nd-order method tuned for stiff problems using a Jacobian.
4. `ode15s`: 1st-order method tuned for stiff problem

we will see what stiff probs are soon.

This was designed by Shampine and Bogacki.

Julia has a suite of ODE solvers that's based on the same ideas as Matlab in the `ODE` package.[7] The methods are the same as Matlab.

SciPy has many of the same methods implemented.[8]

Global Error.

THE PROBLEM

We are considering numerical methods for the initial value problem:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}) \qquad \mathbf{y}(0) = \mathbf{y}_0, t \in [0, T].$$

where $\mathbf{f}$ is continuous and outputs an $\mathbb{R}^d$ vector.

Let $\mathbf{y}(t)$ be the true solution. Let $\mathbb{H}$ be the grid defined by step-sizes: $h_1, \ldots, h_N$ such that

$\mathbb{H}$

$0 = t_0$

$$0 < t_1 < \ldots < t_N$$

fix a grid values $\mathbb{H}$

satisfies $t_i = \sum_{\ell=1}^{i} h_i$.[1] Let $|\mathbb{H}| = \max_i h_i$.

Let $\mathbf{y}_{\mathbb{H}}$ be the value of $\mathbf{y}(t)$ at each of the points in the grid $\mathbb{H}$. We think of this as a set of vectors $\mathbf{y}_k = \mathbf{y}(t_k)$, where each $\mathbf{y}_k \in \mathbb{R}^d$.

Let $\Gamma_{\mathbb{H}}$ be the space of grid-functions.[2] Let $\mathbf{z} \in \Gamma_{\mathbb{H}}$, then[3]

$$\boxed{\|\mathbf{z}\|_\infty = \max_i \|\mathbf{z}_i\|_2}$$

although can be arbitrary

We can look at the difference between two grid functions via this norm:

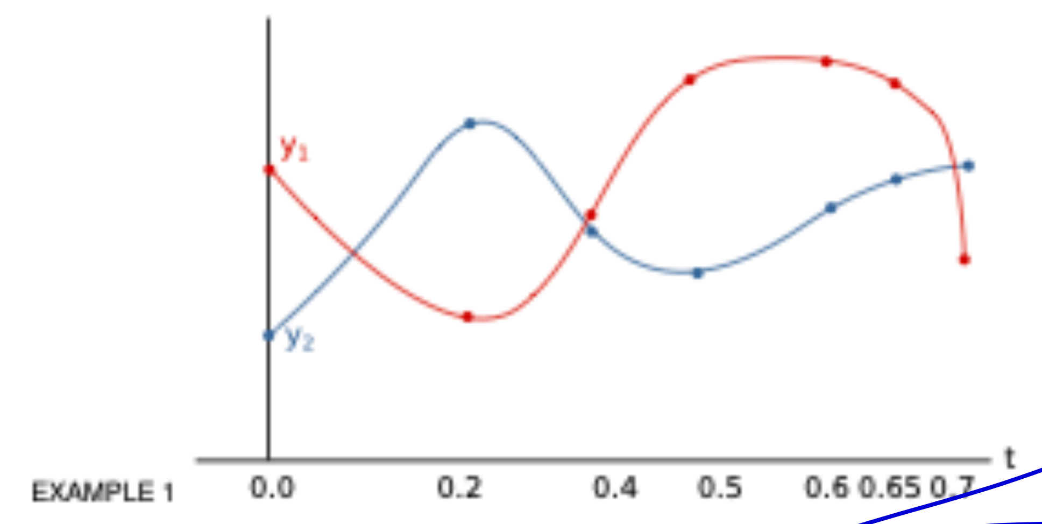
$$\|\mathbf{z} - \mathbf{w}\|_\infty \quad \mathbf{z}, \mathbf{w} \in \Gamma_{\mathbb{H}}$$

Grid functions.

Let $y_{\mathbb{H}}$ be the value of a function y on the grid $\mathbb{H}$.

$$y_{\mathbb{H}} = [y(t_0), y(t_1), y(t_2) \ldots y(t_N)]$$

An Example



EXAMPLE 1

$$\mathbb{H} = (0, 0.2, 0.4, 0.5, 0.6, 0.65, 0.7)$$

write n 2.

$$\mathbf{y}_{\mathbb{H}} = \left(\begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.6 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 1.1 \\ 0.9 \end{bmatrix}, \begin{bmatrix} 1.85 \\ 0.75 \end{bmatrix}, \begin{bmatrix} 1.9 \\ 1.1 \end{bmatrix}, \begin{bmatrix} 1.8 \\ 1.25 \end{bmatrix}, \begin{bmatrix} 0.85 \\ 1.35 \end{bmatrix} \right) \in \Gamma_{\mathbb{H}}$$

the space of grid funct.

If $\|\cdot\| = \|\cdot\|_\infty$ too.

$$\|y_{\mathbb{H}}\|_\infty = \max_i \|y_i\| = 1.9.$$

If $\|\cdot\| = \|\cdot\|_1$, then $\|y_{\mathbb{H}}\|_\infty = 3.05$

Notes on notation The choice of grid is "given" by the space $\Gamma_{\mathbb{H}}$, so we can't compare functions drawn from different grids.

DEFINITION 2 (5.7.2) *Let* $\mathbf{u}$ *be the result of approximating* $\mathbf{y}(t)$ *via a method* Φ *on a grid* $\mathbb{H}$*. Then we say that* Φ *converges if*

$$\|\mathbf{u} - \mathbf{y}_{\mathbb{H}}\|_\infty \to 0 \qquad \text{as } |\mathbb{H}| \to 0.$$

$|\mathbb{H}| = \max_i t_{i+1} - t_i$ = max differ.

$\Rightarrow$ as max differen $\to 0$ then $\|u - y_{\mathbb{H}}\|_\infty \to 0$

i.e. the values converge!

---

[2] You can think of ~~this~~ as the space of $d \times N$ matrices where each column is a time-step.