Def: A *discrete function* is a vector of values

$$\mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \\ f_{n+1} \end{bmatrix}$$ and a "step size" h

---

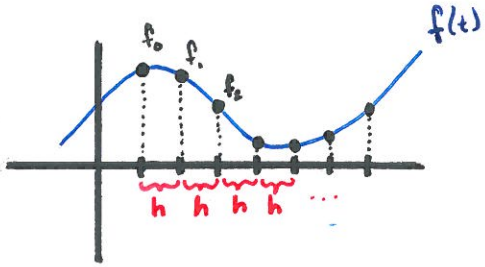
Imagine observing some process in the world and recording measurements at regular time intervals.

→ vector of measurements $= \mathbf{f}$

→ amount of time between measurements $= h$

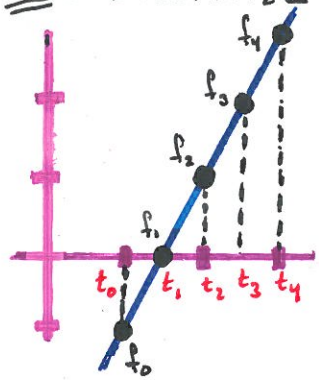
Given a continuous function $f(t)$ and a desired step size h we can "*discretize*"

⟹ Convert $f(t)$ into a vector $\mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \end{bmatrix}$

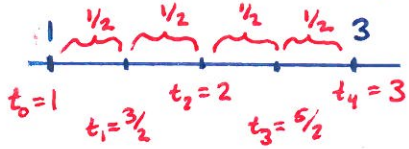


Note: If $f(t)$ has a large domain, you may need to restrict to small $[a, b]$

EX: Discretize $f(t) = 2t - 3$ on $[1, 3]$ with $h = 1/2$



We will sample $f(t)$ at points t between 1 & 3 spaced $1/2$



"sample points" t ⇝ values $f = 2t - 3$

$$\mathbf{t} = \begin{bmatrix} 1 \\ 3/2 \\ 2 \\ 5/2 \\ 3 \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} 2\cdot 1 - 3 \\ 2\cdot 3/2 - 3 \\ 2\cdot 2 - 3 \\ 2\cdot 5/2 - 3 \\ 2\cdot 3 - 3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

Note: Discretizing $f(t)$ on $[a, b]$ with step size h involves "sampling" f at points

t	⇝	f
$t_0 = a$		$f_0 = f(a)$
$t_1 = a + h$		$f_1 = f(a+h)$
$t_2 = a + 2\cdot h$		$f_2 = f(a+2h)$
⋮		⋮
$t_k = a + k\cdot h$		$f_k = f(a+k\cdot h)$
⋮		⋮

This is the reason that we start indexing $\mathbf{f}$ with "f_0" instead of "f_1"

EX: Discretize $f(t) = t^2$ on $[-1, 1]$ with $h = 1/2$

$$t = \begin{matrix} t \\ \begin{bmatrix} -1 \\ -1/2 \\ 0 \\ 1/2 \\ 1 \end{bmatrix} \end{matrix} \rightsquigarrow f = \begin{matrix} f(t)=t^2 \\ \begin{bmatrix} (-1)^2 \\ (-1/2)^2 \\ 0^2 \\ (1/2)^2 \\ 1^2 \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 \\ 1/4 \\ 0 \\ 1/4 \\ 1 \end{bmatrix}$$

Ans.

$f_0 = 1$, $f_1 = 1/4$, i.e. $f_2 = 0$, $f_3 = 1/4$, $f_4 = 1$

EX: Discretize $f(t) = \cos t$ on $[0, 2\pi]$ with $h = \pi/3$

$$t = \begin{matrix} t \\ \begin{bmatrix} 0 \\ \pi/3 \\ 2\pi/3 \\ \pi \\ 4\pi/3 \\ 5\pi/3 \\ 2\pi \end{bmatrix} \end{matrix} \rightsquigarrow f = \begin{matrix} f(t)=\cos t \\ \begin{bmatrix} \cos(0) \\ \cos(\pi/3) \\ \cos(2\pi/3) \\ \cos(\pi) \\ \cos(4\pi/3) \\ \cos(5\pi/3) \\ \cos(2\pi) \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 \\ 1/2 \\ -1/2 \\ -1 \\ -1/2 \\ 1/2 \\ 1 \end{bmatrix}$$

This is called "discrete cosine" (for $h = \pi/3$)... If you have forgotten values of sin & cos, then now is a good time to recall them - they will be important!

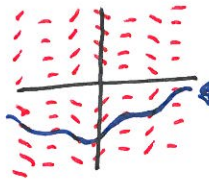
Discretization converts functions to vectors: moving objects into the world of linear algebra. Our goal is to discretize all of calculus (derivatives, integrals) as well as differential equations.

→ Converting calculus & diff. eq. questions into much simpler matrix questions...

In order to motivate our work, we begin with a simple, but powerful, idea from 219

Euler's Method

→ Used by computers to "integrate" a direction field, drawing integral curves



Integral curve follows slopes in direction field.

Euler's Method is a simple recursive algorithm which computes approximate solution to the IVP

$$y' = f(t, y) \qquad y(a) = y_0 \qquad \text{on } [a, b]$$

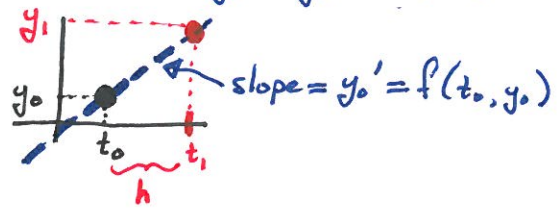
Euler's Method can be computed by hand by iterating a basic step.

{ Approximate solution to $y' = f(t,y)$ beginning at $y(a) = y_0$ on $[a,b]$ with step-size h. }

- **Step 1.** $t_0 = a$ and $y_0 = y(a)$ (from initial values)

  compute: $y_0' = f_0 = f(t_0, y_0)$

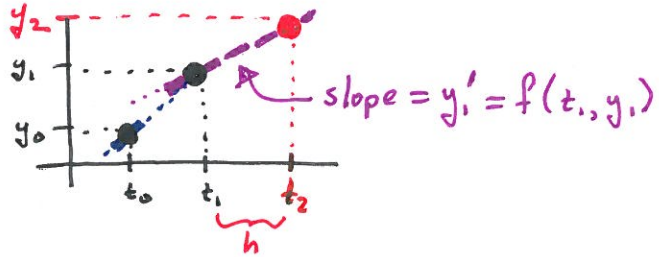
  next val: $t_1 = a + h$ $\quad y_1 = y_0 + f_0 \cdot h$



- **Step 2.** t_1 & y_1 are from previous step

  compute: $y_1' = f_1 = f(t_1, y_1)$

  next val: $t_2 = t_1 + h$ $\quad y_2 = y_1 + f_1 \cdot h$



- Repeat until $t_n = b$.

EX: Use Euler's Method to get approx. solution to $y' = t + 2y$ on $[1,3]$ with $y(1) = -1$ using $h = 1/2$.

Step 1. $t_0 = 1$ $\quad y_0 = -1$

$f_0 = 1 + 2(-1) = -1$

Step 2.

$t_1 = 1 + 1/2 = 3/2$ $\quad y_1 = -1 + (-1)\cdot 1/2 = -3/2$

$f_1 = 3/2 + 2(-3/2) = -3/2$

$t_2 = 3/2 + 1/2 = 2$ $\quad y_2 = -3/2 + (-3/2)\cdot 1/2$

$= -9/4$

Step 3.

$f_2 = 2 + 2(-9/4) = -5/2$

$t_3 = 2 + 1/2 = 5/2$ $\quad y_3 = -9/4 + (-5/2)\cdot 1/2$

$= -7/2$

Step 4.

$f_3 = 5/2 + 2(-7/2) = -9/2$

$t_4 = 5/2 + 1/2 = 3$ $\quad y_4 = -7/2 + (-9/2)\cdot 1/2$

$= -23/4$

Stop here!

at $t = \begin{bmatrix} 1 \\ 3/2 \\ 2 \\ 5/2 \\ 3 \end{bmatrix}$ solution $y \approx \begin{bmatrix} -1 \\ -3/2 \\ -9/4 \\ -7/2 \\ -23/4 \end{bmatrix}$ $\begin{matrix} \leftarrow y_0 = y(1) \\ \leftarrow y_1 \approx y(3/2) \\ \leftarrow y_2 \approx y(2) \\ \leftarrow y_3 \approx y(5/2) \\ \leftarrow y_4 \approx y(3) \end{matrix}$

EX: Use Euler's Method to get approx solution to $y' = y/t$ on $[2, 3]$ with $y(2) = 10$ using $h = 1/5$

Step 1. $t_0 = 2$ $\quad y_0 = 10$

$f_0 = 10/2 = 5$

Step 2 $t_1 = 2 + 1/5 = 11/5$ $\quad y_1 = 10 + 5 \cdot 1/5 = 11$

$f_1 = 11/11/5 = 5$

Step 3 $t_2 = 11/5 + 1/5 = 12/5$ $\quad y_2 = 11 + 5 \cdot 1/5 = 12$

$f_2 = 12/12/5 = 5$

Step 4 $t_3 = 12/5 + 1/5 = 13/5$ $\quad y_3 = 12 + 5 \cdot 1/5 = 13$

$f_3 = 13/13/5 = 5$

Step 5 $t_4 = 13/5 + 1/5 = 14/5$ $\quad y_4 = 13 + 5 \cdot 1/5 = 14$

$f_4 = 14/14/5 = 5$

$t_5 = 14/5 + 1/5 = 3$ $\quad y_5 = 14 + 5 \cdot 1/5 = 15$

Stop!

at $t = \begin{bmatrix} 2 \\ 11/5 \\ 12/5 \\ 13/5 \\ 14/5 \\ 3 \end{bmatrix}$ solution $y = \begin{bmatrix} 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{bmatrix}$ $\begin{matrix} y_0 = y(2) \\ y_1 \approx y(11/5) \\ y_2 \approx y(12/5) \\ y_3 \approx y(13/5) \\ \vdots \end{matrix}$

Euler's Method is nice, but it is not the best... to improve on it, we will look at how it works more closely.

→ If $f(t,y)$ is linear in y then the iterative algorithm in Euler's method can be written as a matrix equation computing solution vector $\mathbf{y}$ using the vector $\mathbf{f}$.

Basic example. $y' = f(t)$ $\quad y(a) = 0$.

(Note: Answer is anti-derivative $y = F(t) - F(a)$)

$t_0 = a$ $\quad y_0 = 0$

$t_1 = a + 1h$ $\quad y_1 = 0 + f(t_0) \cdot h = f_0 \cdot h$ $\quad f(t_0) = f_0$

$t_2 = a + 2h$ $\quad y_2 = y_1 + f(t_1) \cdot h = f_0 \cdot h + f_1 \cdot h$ $\quad f(t_1) = f_1$

$t_3 = a + 3h$ $\quad y_3 = y_2 + f(t_2) \cdot h = (f_0 \cdot h + f_1 \cdot h) + f_2 \cdot h$ $\quad f(t_2) = f_2$

etc.

Euler's Method for approx. solution to

$y' = f(t)$ on $[a, b]$ with $y(a) = 0$

and step-size h

$\Updownarrow$

$y_0 = y(a) = 0$

Note: Matrix has n rows, $(n+1)$ columns

Matrix Multiplication

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = h \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 1 & 1 & 0 & 0 & \cdots \\ 1 & 1 & 1 & 0 & \cdots \\ \vdots & & & \ddots & \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix}$$

More generally: if $y(a) = y_0$ ← Maybe not 0 then

$y_0 = y(a)$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = h \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 1 & 1 & 0 & 0 & \cdots \\ 1 & 1 & 1 & 0 & \cdots \\ \vdots & & & \ddots & \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix} + y_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

solves $y' = f(t)$ where $\mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \end{bmatrix}$

is the discretization of $f(t)$ on $[a, b]$ with step-size h.

EX: Give discrete solution for

$y' = t^2$ on $[-1, 1]$ with $y(-1) = 0$

using step size $h = 1/2$.

t	f
-1	1
$-1/2$	$1/4$
0	0
$1/2$	$1/4$
1	1

$$\mathbf{y} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1/4 \\ 0 \\ 1/4 \\ 1 \end{bmatrix}$$

EX: Give discrete solution for

$y' = 2t + 1$ on $[1, 2]$ with $y(1) = 2$

using step size $h = 1/3$.

t	f
1	3
$4/3$	$11/3$
$5/3$	$13/3$
2	5

$$\mathbf{y} = \frac{1}{3} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 11/3 \\ 13/3 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

Name this matrix "S minus"

$$S^- = \begin{bmatrix} 1 & 0 & 0 & \cdots & \\ 1 & 1 & 0 & \cdots & \\ \vdots & \vdots & \ddots & \ddots & \\ 1 & 1 & 1 & \cdots & 1 \; 0 \end{bmatrix}$$

S^- = matrix with 1 on "lower triangle"

Note: S^- computes **discrete anti-derivative**!

$$y' = f(t)$$

$$\downarrow$$

$$y = \int f(t)\,dt$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} = S^- \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \end{bmatrix} \cdot h$$

$\int$ → S^- ; $f(t)$ → $\begin{bmatrix} f_0 \\ f_1 \\ \vdots \end{bmatrix}$; dt → h

➡ **Fundamental Theorem of Calculus**

"$\int_a^t f(\tau)\,d\tau = F(t)$ anti-deriv. of $f(t)$"

$$y_0 = 0 \longleftrightarrow F(a) = \int_a^a f(\tau)\,d\tau$$

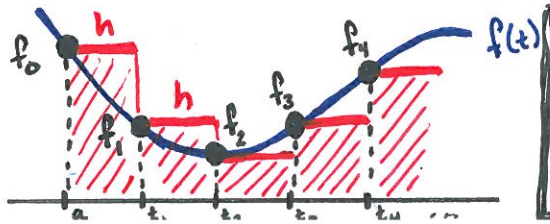
$$y_1 = f_0 \cdot h \longleftrightarrow F(t_1) = \int_a^{t_1} f(\tau)\,d\tau$$

$$y_2 = (f_0 + f_1)\cdot h \longleftrightarrow F(t_2) = \int_a^{t_2} f(\tau)\,d\tau$$

$$y_3 = (f_0 + f_1 + f_2)\cdot h \longleftrightarrow F(t_3) = \int_a^{t_3} f(t)\,d\tau$$

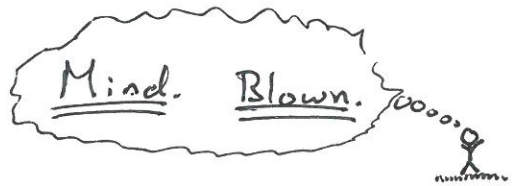
$$\vdots$$

$$y_k = (f_0 + \cdots + f_{k-1})\cdot h \longleftrightarrow F(t_k) = \int_a^{t_k} f(\tau)\,d\tau$$

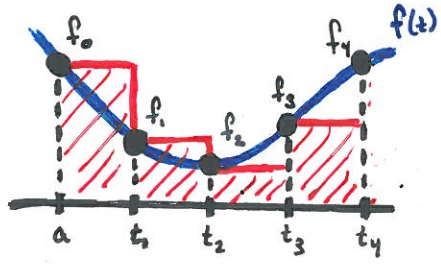


Rows of S^- compute left-endpoint Riemann sums

Conclusion: Using Euler's Method to solve $y' = f(t)$ is equivalent (via S^-) to computing approximate integral using left-endpoint Riemann sums.

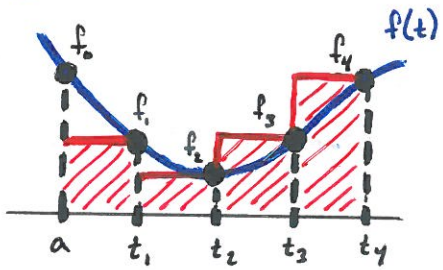


Basic Calculus: left-endpoint Riemann sums are not the only way to approximate... (they aren't even that great)



"left-endpoint" area

$$\text{area} = (f_0 + f_1 + \cdots + f_n)\cdot h$$



"right-endpoint" area

$$\text{area} = (f_1 + f_2 + \cdots + f_{n+1})\cdot h$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}}_{S^-} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \cdot h$$

left-endpoint anti-deriv.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}}_{S^+} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \cdot h$$

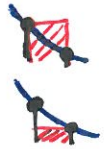
right-endpoint anti-deriv.

Observation from basic calculus:

When f is decreasing $(f' < 0)$

→ left-endpoint area is too big

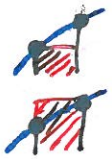
→ right-endpoint area is too small



When f is increasing $(f' > 0)$

→ left-endpoint area is too small

→ right-endpoint area is too big



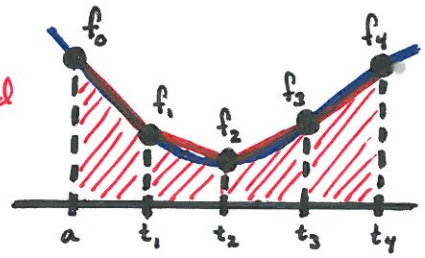
- Use average of left & right areas

"trapezoidal area"

area = average of left-endpoint and right-endpoint

$= \frac{1}{2}(f_0 + \cdots + f_n) \cdot h$

$+ \frac{1}{2}(f_1 + \cdots + f_{n+1}) \cdot h$



↳ area $= \left(\frac{f_0}{2} + f_1 + \cdots + f_n + \frac{f_{n+1}}{2}\right) \cdot h$

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & 1 & 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \cdot h$$

(left-endpoint anti-deriv.) $S^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

(right-endpoint anti-deriv.) $S^+ = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

(trapezoidal anti-deriv.) $S^\circ = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & 1 & \frac{1}{2} \end{bmatrix}$

→ Using trapezoidal anti-derivative gives much more accurate answer than Euler's Method for solving

$$y' = f(t)$$

In the next few lectures we will see how to extend this to a method for solving even higher order differential equations

(Euler's method only works for 1st order.)