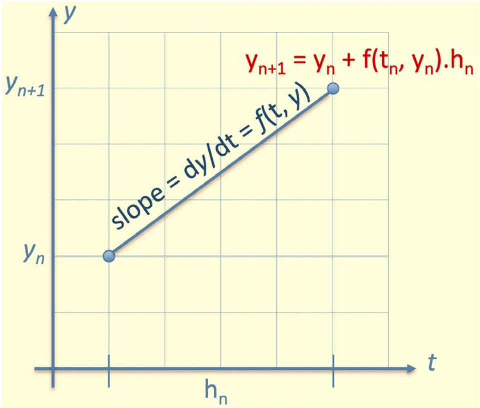


Lecture 19:
Euler's Method

CS 111: Intro to Computational Science

Spring 2023

Ziad Matni, Ph.D.

Dept. of Computer Science, UCSB

Administrative

- Final Exam Review (optional):

  Friday, June 9th

  PHELP 2510

  10:00 AM – 11:45 AM

Euler Method

A numerical method

to approximate an ODE solution

Comes from the identity: $\boldsymbol{\frac{dy}{dt} = f(t, y)}$

Which can be expressed as: $\boxed{\frac{y_{n+1} - y_n}{h_n} = f(t_n, y_n)}$

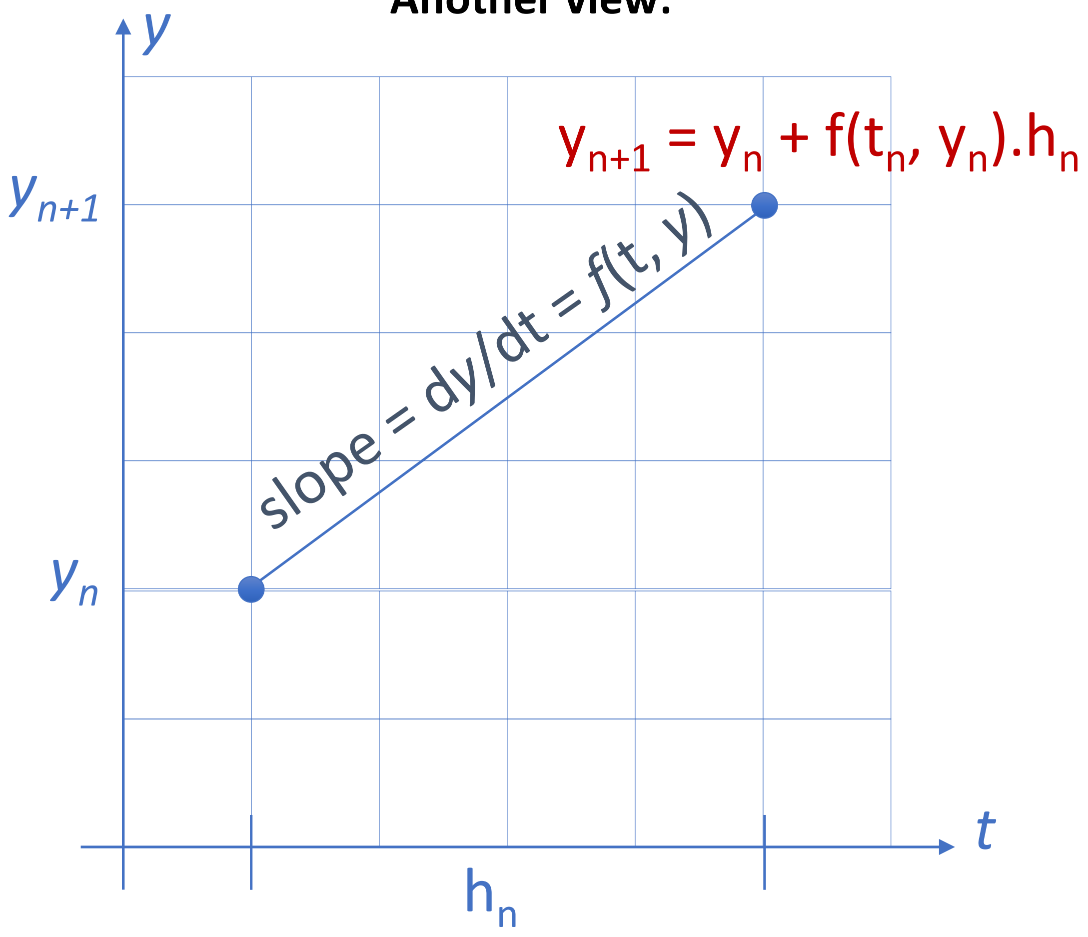
h is some small number

So: $$y_{n+1} = y_n + f(t_n, y_n).\, h_n$$

next value *current value* *change*

A basis for an iterative method!

Another view:



Euler Method

- Let's see how we can use *discretization approximation* methods to solve an ODE

- We'll do the case of $dy/dt = y$ with $y(0) = 1$
  - We know that this solution should be: $y(t) = e^t$

- *To the blackboard!!...*

Consider $\frac{dy}{dt} = y(t)$ & $y(0) = 1$

i.e. $y(t) = e^t$

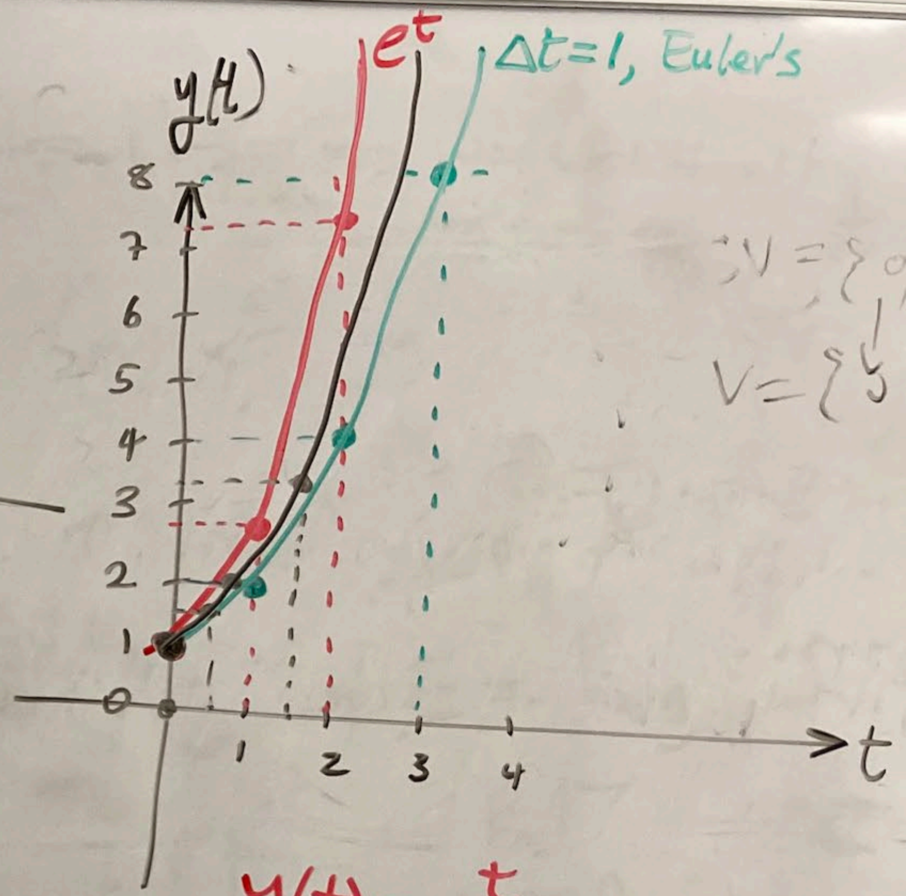
EULER's: Assume a step size $\Delta t = 1$

$\Delta t = 1$

t	$y(t)$	$\frac{dy}{dt}$
0	1 (+1)	1
1	2 (+2)	2
2	4	4
3	8	8

$\Delta t = 0.5$

t	$y(t)$	dy/dt
0	1 +0.5	1
0.5	1.5 0.75	1.5
1.0	2.25	2.25
1.5	3.375	3.375
⋮		



$y(t) = e^t$

$y(0) = e^0 = 1$

$y(1) = e^1 \cong 2.71$

$y(2) = e^2 \cong 7.39$

Euler's Method

- It's prone to giving larger errors *if the step size isn't small enough*
  - i.e. the **error** is *proportional* to the **step size**

- It's called an "*explicit method*" because
  it only uses information at time $\mathbf{t_n}$
  to advance the solution to time $\mathbf{t_{n+1}}$
  - This has implications for **stability** of this method

Developing Algorithms for Euler's Method

- We'll develop function `ode1(fun, t_span, y0, h)` as our "**Forward Euler Method**" and use it to solve ODEs
  - We'll analyze its effectiveness

- We'll then develop `ode2(fun, t_span, y0, h)` that will utilize 2 slopes instead of 1 to give a better approximation
  - This is akin to a higher-order polynomial approach which is what the Runge-Kutta Methods use!

We Will Demo…

- $y' = 0.5\,y$; $y(0) = 1$
  - Note: this has a solution of **$y = e^{0.5t}$**

- $y'' = 1 - y$
  - This has a general solution of $y = A.\sin(\omega_0.t) + B.\cos(\omega_0.t) + C$
    - When you graph this, it just looks like a generic sinusoidal wave
    - Also known as an undamped harmonic oscillator

Quick! To the Python-mobile!

Your To-Dos

- Study for the Final Exam! ☺

Good luck with all your Exams!

Have a Great Summer Break!

</LECTURE>