Training and Workshop on

Python Programming and Numerical Methods

An Introduction to Numerical Linear Algebra and Differential Equations

January 29-31, 2025

2F CCE Building, Visayas State University,
Visca, Baybay City, Leyte

Differential Equations

Arrianne Crystal Velasco

Institute of Mathematics
University of the Philippines Diliman

January 30, 2025

SESSION 1: INITIAL VALUE PROBLEMS

Differential Equations

A *differential equation* (DE) is an equation involving derivatives.

DEs are important mathematical tools in modeling understanding, and predicting systems that change with time.

Example:

- $y'(t) = -y$
- $y'(t) = t^2 + y^2$

When all the derivatives are with respect to only one independent variable, then we have an *ordinary differential equation* (ODE).

Some research work that we did that involved solving DEs.

- **Numerical solution of water quality model for Pasig river**

$$U\frac{\partial(AC)}{\partial t} + \alpha L(C) = Avr,$$

- **Numerical simulation of Taal volcanic dispersion**

$$\frac{\partial C}{\partial t} + \nabla \cdot J = S,$$

Initial-Value Problems

A first-order differential equation is of the form

$$y'(t) = f(t, y((t)).$$

There is usually an infinite family of functions that satisfy the above ODE. To single out a particular solution, an *initial condition*

$$y(t_0) = y_0$$

is given. An ODE together with an initial condition is called an *initial-value problem* (IVP).

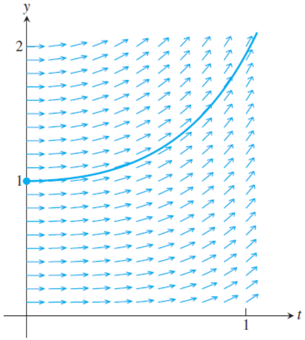
We aim to find the function $y(t), t \geq t_0$ satisfying both the ODE and the initial condition.

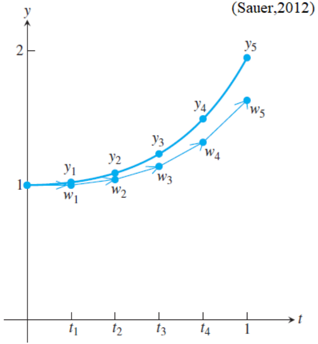
NUMERICAL METHODS

EULER'S METHOD

Euler's method is the most elementary (simple) approximation technique for solving initial-value problems.

Approximations to the solution $y(t)$ at various values, called *mesh points*, in the interval $[t_0, T]$ are obtained instead of a continuous approximation to y.





Euler's method

Start with $N+1$ mesh points in the interval $[t_0, T]$,

$$t_i = t_0 + ih, \quad i = 0, 1, \ldots, N$$

with **step size** $h = (T - t_0)/N = t_{i+1} - t_i$.

By Taylor's theorem, we have for each $i = 0, 1, \ldots, N-1$

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2} y''(\xi_i),$$

where $\xi_i \in (t_i, t_{i+1})$.

We can then approximate the values of y at the mesh points by

Euler's method

$$y(t_{i+1}) \approx y(t_i) + hf(t_i, y_i)$$

Example. Use Euler's method, with $h = 0.1$, to solve the IVP

$$\frac{dy}{dt} = -y, \quad 0 \leq t \leq 5, \quad y(0) = 1.$$

Runge-Kutta method of order 4

- The most popular and commonly used Runge-Kutta Method is the **Runge-Kutta method of order 4** (RK4) ($\mathcal{O}(h^4)$).

$$y_{i+1} = y_i + \frac{h}{6}(s_1 + 2s_2 + 2s_3 + s_4),$$

where

$$\begin{aligned}
s_1 &= f(t_i, y_i) \\
s_2 &= f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}s_1\right) \\
s_3 &= f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}s_2\right) \\
s_4 &= f(t_{i+1}, y_i + hs_3)
\end{aligned}.$$

Runge-Kutta method of order 4

Example. Use RK4, with $h = 0.1$, to solve the IVP

$$\frac{dy}{dt} = -y, \quad 0 \leq t \leq 5, \quad y(0) = 1.$$

Explicit vs Implicit methods

- Euler's and RK4 methods are examples of explicit methods since they uses only information at time t_i to advance the solution to time t_{i+1}.
- Explicit methods have a limited stability region as shown in the convergence of Euler's method.
- Implicit methods, on the other hand, have larger stability region because of the usage of information at time t_{i+1}.
- Backward Euler's method is an example of an implicit method.

Implicit Euler's method

Recall the backward-difference formula for the first derivative:

$$f'(x_0+h) \approx \frac{f(x_0+h)-f(x_0)}{h}.$$

Implicit Euler's method:

$$y(t_{i+1}) \approx y(t_i) + hf(t_{i+1}, y_{i+1})$$

Implicit Euler's method

Example. Use Implicit Euler's method, with $h = 0.1$, to solve the IVP

$$\frac{dy}{dt} = -y, \quad 0 \leq t \leq 5, \quad y(0) = 1.$$

Higher-order ODEs

Consider an n-order differential equation

$$y^{(n)} = f(t, y, y', \ldots, y^{(n-1)})$$

with initial conditions

$$y(t_0) = \alpha_1,\ y'(t_0) = \alpha_2,\ \ldots,\ y^{(n-1)}(t_0) = \alpha_n$$

If we let $y_1 = y$, $y_2 = y'$, ... , $y_n = y^{(n-1)}$, then we have

$$\begin{cases} \begin{array}{rlr} y_1 = & y, & y_1 = \alpha_1 \\ y_2 = & y', & y_2 = \alpha_2 \\ \vdots & & \vdots \\ y_{n-1} = & y^{(n-2)}, & y_{n-1} = \alpha_{n-1} \\ y_n = & y^{(n-1)}, & y_n = \alpha_n \end{array} \end{cases}$$

Higher-order ODEs

Example: Convert $y''' = 2(y'')^2 - y' + yy'' + \sin t$ to an equivalent system of IVP.

System of Differential Equations

An m**th-order system of first-order initial-value problems has the form**

$$\begin{cases} y_1' &= f_1(t, y_1, y_2, \ldots, y_m), \\ y_2' &= f_2(t, y_1, y_2, \ldots, y_m), \\ &\vdots \\ y_m' &= f_m(t, y_1, y_2, \ldots, y_m), \end{cases} \tag{1}$$

for $t_0 \le t \le T$ with the initial conditions

$$y_1(t_0) = \alpha_1,\; y_2(t_0) = \alpha_2, \ldots, y_m(t_0) = \alpha_m.$$

Our goal is to find m functions $y_1(t), y_2(t), \ldots, y_m(t)$ satisfying each of the differential equations together with all the initial conditions.

Euler's method:

$$\begin{cases} y_1^{i+1} & = & y_1^i + hf_1(t, y_1^i, y_2^i, \ldots, y_m^i), \\ y_2^{i+1} & = & y_2^i + hf_2(t, y_1^i, y_2^i, \ldots, y_m^i), \\ & \vdots & \\ y_m^{i+1} & = & y_m^i + hf_m(t, y_1^i, y_2^i, \ldots, y_m^i), \end{cases}$$

Let $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$ and $F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$. Then Euler's method in matrix form is

$$Y^{i+1} = Y^i + h * F(t, Y)$$

Example: Convert $y'' - ty = 0$ to an equivalent system of IVP and solve it on the interval $[0,1]$ using Euler's method with $h = 0.25$ and initial conditions $y(0) = y'(0) = 1$.

RUNGE-KUTTA METHOD OF ORDER 4

Runge-Kutta method of order 4 for system of IVPs. The $i+1$-th update for the j-th unknown function in the system is given by

$$y_j^{i+1} = y_j^i + \frac{h}{6}(s_j^1 + 2s_j^2 + 2s_j^3 + s_{4,j}),\ j = 1, 2, \ldots, m$$

where

$$\begin{aligned}
s_{1,j} &= f_j(t, y_1^i, y_2^i, \ldots, y_m^i) \\
s_{2,j} &= f_j\left(t_i + \frac{h}{2}, y_1^i + \frac{h}{2}s_{1,1}, y_2^i + \frac{h}{2}s_{1,2}, \ldots, y_m^i + \frac{h}{2}s_{1,m}\right) \\
s_{3,j} &= f_j\left(t_i + \frac{h}{2}, y_1^i + \frac{h}{2}s_{2,1}, y_2^i + \frac{h}{2}s_{2,2}, \ldots, y_m^i + \frac{h}{2}s_{2,m}\right) \\
s_{4,j} &= f_j(t_{i+1}, y_1^i + hs_{3,1}, y_2^i + hs_{3,2}, \ldots, y_m^i + hs_{3,m})
\end{aligned}.$$

Hands-on Activity

1. Solve the IVP

$$\frac{dy}{dt} = ty + t^3, \quad 0 \leq t \leq 1, \quad y(0) = 1.$$

   using the following methods:
   - Euler's method
   - RK4 method
   - Implicit method
2. Write the code for for solving systems of IVPs using RK4 by modifying the code using Euler's method. Use the same example in Euler's method to test your code.

Session 2: Boundary Value Problems

Boundary Value Problem

Boundary-value problems (BVP) are differential equations with conditions imposed at different points. For first-order differential equations, only one condition is specified, so there is no distinction between initial-value and boundary-value problems.

Physical problems that are position-dependent rather than time-dependent are often described in terms of differential equations with conditions imposed at more than one point.

- The bending of an elastic beam under a distributed transverse load
- The distribution of electrical potential between two flat electrodes

We consider second-order equations with two boundary values.

BVP vs IVP

Consider a general second-order DE

$$y'' = f(t, y, y'),\ a \leq t \leq b$$

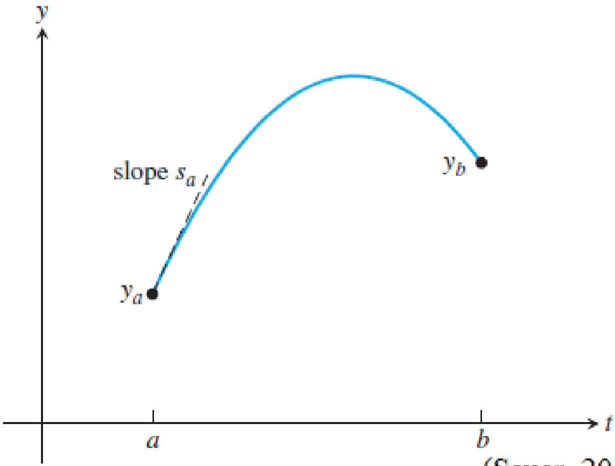
Inital Value Problem:

$$y(a) = y_a, \qquad y'(a) = s_a$$

Boundary Value Problem:

$$y(a) = y_a, \qquad y(b) = y_b$$

BVP vs IVP



(Sauer, 2012)

FINITE DIFFERENCE METHOD

The **finite difference method** converts a BVP directly into a system of algebraic equations.

- A set of mesh points is introduced within the interval.
- Any derivatives appearing in the ODE or boundary conditions are replaced by finite difference approximations at the mesh points.

Finite difference method

We consider a BVP of the form

$$y'' = f(x, y, y'),\ a \leq t \leq b$$

$$y(a) = \alpha, \qquad y(b) = \beta$$

The mesh points are of the form

$$t_i = a + ih, i = 0, 1, \ldots, N+1,$$

where $h = \dfrac{b-a}{N+1}$.

We seek approximate solution values $y_i \approx y(t_i), i = 1, \ldots, N$. We already have $y_0 = y(a) = y_a$ and $y_{N+1} = y(b) = y_b$.

Finite difference method

The derivatives appearing in the ODE are then replaced by finite difference approximations (central-difference formulas).

$$y'(t_i) \approx \frac{y_{i+1} - y_{i-1}}{2h} \quad \text{and} \quad y''(t_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

The ODE now is equivalent to

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f\left(t_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\right).$$

The resulting system of equations determine the approximate solution at all mesh points *simultaneously*.

The system may be linear or nonlinear, depending on whether f is linear or nonlinear in y and y'.

Example: We illustrate the finite difference method with $h = 0.1$ on the two-point BVP for the second-order scalar ODE

$$y'' = 6t, \ 0 \leq t \leq 1,$$

with boundary conditions

$$y(0) = 0, \qquad y(1) = 1.$$

Hands-on Activity

Solve the two-point BVP using the finite difference method with $h = 0.1$

$$y'' = 4y,\ 0 \leq t \leq 1,$$

with boundary conditions

$$y(0) = 1, \qquad y(1) = 3.$$

Remark:

- In practice, achieving acceptable accuracy with a finite difference method requires *many more mesh points*. We would expect convergence to the true solutions as the number of mesh points goes to infinity.

Finite difference method

Consider the Laplace equation on the unit square

$$\begin{cases} u_{xx} + u_{yy} = 0, & 0 \le x, y \le 1 \\ u(0,y) = u(1,y) = 0, & 0 \le y \le 1 \\ u(x,0) = 0, & 0 \le x \le 1 \\ u(x,1) = 1 - 4(x-0.5)^2, & 0 \le x \le 1, \end{cases}$$

where u is some time-independent function. Divide the x-interval [0,1] and y-interval [0,1] both into N equal parts.

Hands-on Activity

Consider the BVP

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = -2\pi^2 \sin(\pi x)\sin(\pi y), \quad 0 < x < 1, \quad 0 < y < 1 \\ \quad u(x,y) = 0 \quad \text{if } x = 0,1 \text{ or } y = 0,1 \end{cases}$$

where u is some time-independent function. Divide the x-interval [0,1] and y-interval [0,1] both into n equal parts.

References:

- R.L. Burden, D.J. Faires. *Numerical Analysis.* Brooks Cole, 2011.
- T. Sauer. *Numerical Analysis.* Pearson, 2012.
- Michael T. Heath, *Scientific Computing: An Introductory Survey.* Society for Industrial and Applied Mathematics, 2002.
- Atkinson, K.E. *An Introduction to Elementary Numerical Analysis (2e).* Wiley, 1989.