TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TS VŨ ĐỖ HUY CƯỜNG

HƯỚNG DẪN THỰC THÀNH
PHƯƠNG PHÁP TÍNH

Lưu hành nội bộ

HỌC KỲ 1 NĂM HỌC 2024-2025

BÀI MỞ ĐẦU

LÀM QUEN VỚI PYTHON

1 GIỚI THIỆU PYTHON

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, van Rossum đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

1.1 Cài đặt Python

1.1.1 Cài đặt Python

Để cài đặt Python, vào trang chủ của Python ở link https://www.python.org/downloads/, nếu sử dụng Windows thì nhấp nút "Download Python 3.12.0" như hình dưới. Lưu ý, con số 3.12.0 là số phiên bản (version) của Python, nó sẽ thay đổi về sau (càng ngày càng lớn hơn).

Nếu máy dùng hệ điều hành khác (Linux, Mac OS, ...) hoặc muốn cài phiên bản Python khác thì có thể nhấp vào các liên kết phía dưới nút "Download Python 3.12.0".

Sau khi tải file cài đặt (file python-3.12.0.exe), nhấp đúp vào file đã tải sẽ thấy hộp thoại cài đặt như hình dưới. Bật lựa chọn "Add python.exe to PATH" và nhấp "Install Now". Sau đó nhấp "Yes" và đợi Python cài đặt. Khi cài đặt xong, Python sẽ thông báo là cài đặt thành công (setup was successful), nhấn nút Close và có thể bắt đầu dùng Python.

1.1.2 Cài đặt Visual Studio Code

Đầu tiên, hãy truy cập vào link https://code.visualstudio.com/download và tải bản cài đặt phù hợp cho máy mình.

Để tải bản cài đặt cho Windows, nhấp chọn vào ô Windows, sau đó chạy file được tải về, tiếp tục nhấp chọn vào ô tròn "I accept the agreement" để đồng ý với các điều kiện sử dụng và nhấn "Next" vài lần để tiếp tục cài đặt.

Tiếp theo, chỉ cần tích vào các ô như trong hình bên dưới và sau đó nhấn "Next" (cũng có thể chọn toàn bộ các tùy chọn, điều này không ảnh hưởng nhiều đến hoạt động của Visual Code).

Sau đó, chỉ cần nhấn "Install" để hoàn thành cài đặt.

Cài đặt các tiện ích mở rộng

Sau khi cài đặt VS Code, ứng dụng sẽ tự động được mở. Tiếp theo, vào phần "Extensions", chọn "Python" và nhấn "Install" để thêm vào VS Code một tiện ích mở rộng cho việc lập trình Python.

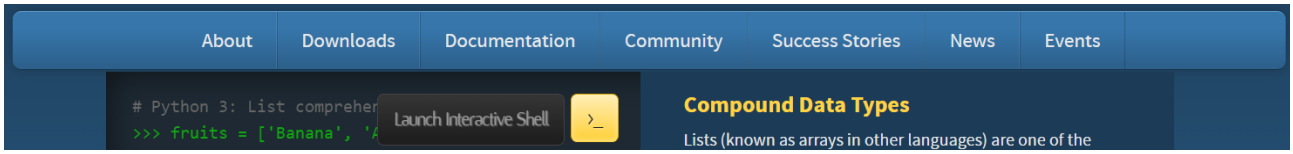
Tiếp theo, ta cài thêm thư viện Numpy. Chọn liên tiếp "···" → "Termainal" → "New Terminal"

Sau đó gõ "pip install numpy" và đợi hệ thống tự cài đặt.

Làm tương tự, gõ "pip install sympy" để cài thêm thư viện Sympy hoặc các thư viện khác.

1.1.3 Sử dụng online

Một lựa chọn khác, làm việc với Python mà không cần cài đặt, là dùng **Python trực tuyến** (online Python). Vào trang https://www.python.org/ và nhấp nút "Launch Interactive Shell".



1.2 Môi trường làm việc của Python

Python cung cấp hai môi trường làm việc, đó là môi trường Command Window và Script.

a) Môi trường Command Window: Đây là môi trường biên dịch chính của Python. Các kết quả của quá trình tính toán được biểu diễn ở đây. Đây là một điểm mạnh của Python so với các phần mềm khác, thể hiện tính "thân thiện" của Python khi người ta có thể vừa tính toán vừa thấy kết quả trên cùng một cửa sổ. Tuy nhiên, ta chỉ nên thực hiện các phép toán đơn giản và việc theo dõi cũng như sửa chữa câu lệnh khá khó khăn.

b) Môi trường Scprit: Đây là môi trường lập trình tính toán của Python. Các kết quả của các phép tính ở đây sẽ được biểu diễn ở môi trường Command Window. Vì vậy ta có thể thực hiện các phép toán phức tạp trong môi trường này. Thông thường, người ta sẽ viết các chương trình chính ở đây. Và có thể sửa chữa, thay đổi, thêm bớt các câu lệnh một cách dễ dàng mà vẫn theo dõi được toàn bộ chương trình.

Trong môi trường Scrpit, người ta thường xây dựng các function (chương trình con) để thực hiện những phép tinh được lặp lại nhiều hoặc để đóng gói cho cho các công đoạn phức tạp. Các function thường được lưu trong cùng thư mục với chương trình chính. Khi cần thực hiện function trong chương trình chính, người ta sẽ gọi tên của function cần sử dụng với đầy đủ các input.

1.3 Tính toán với Python

Một trong những công việc căn bản và quan trọng của con người là tính toán (computation). Thuật ngữ này có nghĩa rất rộng, ám chỉ những quá trình gồm các bước thao tác trên các đối tượng số hoặc không phải số. Có thể nói, Toán học, khoa học, kỹ thuật ra đời và phát triển từ nhu cầu tính toán. Máy tính (computer) được xem là một trong những phát minh lớn nhất giúp thực hiện tự động việc tính toán và thuật ngữ điện toán (computing) ám chỉ việc tính toán bằng máy tính. Ta sẽ thấy rằng Python là một công cụ mạnh mẽ và tiện dụng của điện toán. Bài này hướng dẫn dùng Python để thực hiện các tính toán của biểu thức đại số, các phép tính đạo đàm, vi tích phân, vẽ đồ thị hàm số,…

2. Thực hành trên máy tính

2.1 Chương trình đầu tiên

Bài tập 1. Viết chương trình trong Python để hiển thị ra:

a) Tổng, hiệu, tích, thương của 2468 và 1234.

b) Tính chu vi và diện tích hình chữ nhật có chiều dài và chiều rộng lần lượt là 7.8 và 6.4.

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
0	Mở PYTHON bằng VISUAL CODE hoặc SUBLIME TEXT	Mở phần mềm
1	print("2468 + 1234 =",2468 + 1234) print("2468 - 1234 =",2468 - 1234) print("2468 * 1234 =",2468 * 1234) print("2468 / 1234 =",2468 / 1234)	Hiển thị kết quả của phép tính.
2	print("Area =",6.4 * 7.8) print("Perimeter =", (6.4 + 7.8)*2)	Hiển thị kết quả của phép tính.

2.2 Tính toán trên trường số thực

2.2.1 Các hàm thông dụng trong thư viện math:

import math

Hàm toán học	Trên Python	**Hàm toán học**	Trên Python
G	math.pi	**sin(x)**	math.sin(x)
e	math.e	**cos(x)**	math.cos(x)
$+\infty$ **;** $-\infty$	math.inf ; -math.inf	**tan(x)**	math.tan(x)
$\sqrt{x}$	math.sqrt(x)	**cot(x)**	1/math.tan(x)
|x|	abs(x)	**arcsin(x)**	math.asin(x)
ln(x) ; lg(x) ; $\log_a x$	math.log(x) ; math.log10(x) ; math.log(x,a)	**arccos(a)**	math.acos(a)
$\mathbf{e^x}$ **;** $\mathbf{x^y}$	math.exp(x) ; math.pow(x, y)	**arctan(x)**	math.atan(x)
n!	math.factorial(n)	**arccot(x)**	math.atan(1/x)

2.2.2 Các hàm thông dụng trong thư viện sympy:

import sympy as sym

Hàm toán học	Trên Python	**Hàm toán học**	Trên Python
		sin(x)	sympy.sin(x)
		cos(x)	sympy.cos(x)
		tan(x)	sympy.tan(x)
$\sqrt{x}$	sympy.sqrt(x)	**cot(x)**	sympy.cot(x)
|x|	sympy.Abs(x)	**arcsin(x)**	sympy.asin(x)
ln(x) ; $\log_a x$	sympy.log(x) ; sympy.log(x,a)	**arccos(a)**	sympy.acos(a)
$\mathbf{e^x}$	sympy.exp(x)	**arctan(x)**	sympy.atan(x)

		arccot(x)	sympy.acot(x)

2.2.3 Các hàm thông dụng trong thư viện numpy:

import numpy as np

Hàm toán học	Trên Python	**Hàm toán học**	Trên Python
G	np.pi	**sin(x)**	np.sin(x)
e	np.e	**cos(x)**	np.cos(x)
$+\infty$ **;** $-\infty$	np.Inf; np.NINF	**tan(x)**	np.tan(x)
$\sqrt{x}$	np.sqrt(x)	**cot(x)**	1/np.tan(x)
|x|	np.absolute (x)	**arcsin(x)**	np.arcsin(x)
ln(x) ; $\log_a x$	np.log(x) ; np.log(x)/ np.log(a)	**arccos(a)**	np.arccos(a)
$\mathbf{e^x}$	np.exp(x)	**arctan(x)**	np.arctan(x)
		arccot(x)	np.arctan(1/x)

Bài tập 2. Tính các giá trị sau

a) $\sqrt{5.1}$

b) $\log_3 4$

c) $\arcsin^2\left(\frac{1}{2}\right)$

Hướng dẫn

STT	Cách 1: Dùng math	Cách 2: Dùng sympy	Cách 3: Dùng numpy
1	print(math.sqrt(5.1))	x = sympy.Symbol('**x**') y = sympy.lambdify(x,sympy.sqrt(x)) print(y(5.1))	print(np.sqrt(5.1))
2	print(math.log(4,3))	x = sympy.Symbol('**x**') y =sympy.lambdify(x,sympy.log(x,3)) print(y(4))	print(np.log(4)/np.log(3))
3	print((math.asin(1/2))**2)	x = sympy.Symbol('**x**') y = sympy.lambdify(x,(sympy.asin(x))**2) print(y(1/2))	print((np.arcsin(1/2))**2)

2.2.4 Viết function với python:

Bài tập 1. Tính giá trị các biểu thức sau

a) $A = 3^2 - 4.\dfrac{5+6}{2} + \sqrt{2} - \sqrt[3]{5}$

b) $B = \sin 3\alpha + 4\cos 2\alpha - \tan\alpha.\cot 5\alpha$ với $\alpha = \pi/6$

c) $C = e^{2+t} + C = e^{2+t} + \ln(4-t)$ với $t = 3$

d) $D = \left[B + C, A^2 - B.C, |A| + 1\right]$

A =	B =	C =	D =

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	A = 3**2 - 4 * (5 + 6) / 2 + np.sqrt(2) - np.cbrt(5) print("Giá trị của biểu thức A là:", A)	Tính giá trị của A. Xuất giá trị A.
3	alpha = np.pi / 6 B = np.sin(3 * alpha) + 4 * np.cos(2 * alpha) - np.tan(alpha) * np.cot(5 * alpha) print("Giá trị của biểu thức B là:", B)	Gán giá trị alpha. Tính giá trị của B. Xuất giá trị B.
4	t = 3 C = np.exp(2 + t) + np.exp(2 + t) + np.log(4 - t) print("Giá trị của biểu thức C là:", C)	Gán giá trị của t. Tính giá trị của C. Xuất giá trị C.
5	D = np.array([B + C, A**2 - B * C, np.abs(A) + 1]) print("Giá trị của biểu thức D là:", D)	Tính giá trị mảng D. Xuất giá trị D.

Bài tập 2. Tính giá trị các biểu thức sau

a) $E = (4-7)^2 + \sqrt[3]{3-1}\sqrt[4]{6+2}\dfrac{\sqrt{9-3}}{\sqrt{2}}$.

b) $F = a^2(\sin b + \cos c) - 2b\tan c + 4c(\cot a - \cot b)$ với $a = 2, b = 3, c = 1$

c) $G = \ln(x^2 - 2x + 1) + e^{4x+2}$ với $x = 2$.

d) $H = \left[G/2; \sqrt{EF}\right]$

E =	F =	G =	H =

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	E = (4 - 7)**2 + np.cbrt(3 - 1) * (6 + 2)**(1/4) * np.sqrt(9 - 3) / np.sqrt(2)) print("Giá trị của biểu thức E là:", E)	Tính giá trị E. Xuất giá trị E.
3	a = 2 b = 3 c = 1 F = a**2 * (np.sin(b) + np.cos(c)) - 2 * b * np.tan(c) + 4 * c * (np.cot(a) - np.cot(b)) print("Giá trị của biểu thức F là:", F)	Gán giá trị a. Gán giá trị b. Gán giá trị c. Tính giá trị của F. Xuất giá trị F.

4	x = 2 G = np.log(x**2 - 2*x + 1) + np.exp(4*x + 2) print("Giá trị của biểu thức G là:", G)	Gán giá trị x. Tính giá trị của G. Xuất giá trị G.
5	H = np.array([G / 2, np.sqrt(E * F)]) print("Giá trị của biểu thức H là:", H)	Tính giá trị mảng H. Xuất giá trị H.

Bài tập 3. Cho hàm số $y = f(x) = 4x^3 - 3x^2 - 5x + 2$

a) Tìm giá trị $y_1 = f(1), y_2 = f(2), y_3 = f(-4), y_4 = f(0)$

$y_1 =$	$y_2 =$	$y_3 =$	$y_4 =$

b) Vẽ đồ thị hàm số trên đoạn $[-4,1]$

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	def f(x): return 4*x**3 - 3*x**2 - 5*x + 2 y1 = f(1) print("y1 = f(1) =", y1) y2 = f(2) print("y2 = f(2) =", y2) y3 = f(-4) print("y2 = f(2) =", y2) y4 = f(0) print("y4 = f(0) =", y4)	Định nghĩa hàm số Tính giá trị y1. Xuất giá trị y1. Tính giá trị y2. Xuất giá trị y2. Tính giá trị y3. Xuất giá trị y3. Tính giá trị y4. Xuất giá trị y4.
3	import matplotlib.pyplot as plt x = np.linspace(-4, 1, 100) y = f(x) plt.figure(figsize=(8, 6)) plt.plot(x, y, label='y = 4x^3 - 3x^2 - 5x + 2', color='blue') plt.xlabel('x') plt.ylabel('y') plt.title('Đồ thị hàm số y = 4x^3 - 3x^2 - 5x + 2 trên đoạn [-4, 1]') plt.grid(True) plt.legend() plt.show()	Khai báo gói vẽ hình. Tạo mảng x gồm 100 giá trị từ -4 đến 1. Gán giá trị y. Vẽ biểu đồ. Vẽ đồ thị. Đặt tên trục x. Đặt tên trục y. Đặt tên đồ thị. Hiện lưới. Hiển thị chú thích. Hiện thị đồ thị ra màn hình.

Bàu tập 4. Tính giá trị các hàm số sau tại $x = -2, x = 0, x = 1, x = 3$

a) $f(x) = x^5 - x^3 + 2x - 4$
b) $g(x) = \sin\frac{\pi x}{3} - \cos\frac{\pi}{4}$
c) $h(x) = e^x + \ln x^2 + 1$
d) $k(x) = \sqrt{x^2 + 3x + 9}$

$f(-2) =$	$f(0) =$	$f(1) =$	$f(3) =$
$g(-2) =$	$g(0) =$	$g(1) =$	$g(3) =$

$h(-2) =$	$h(0) =$	$h(1) =$	$h(3) =$
$k(-2) =$	$k(0) =$	$k(1) =$	$k(3) =$

Hướng dẫn:

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def f(x):
	  return x**5 - x**3 + 2*x - 4
	x_values = [-2, 0, 1, 3]
	for x in x_values:
	  print(f"f({x}) = {f(x)}")
		Định nghĩa hàm số.
		
		Khai báo mảng.
		Vòng lặp tính f.
		Hiện ra giá trị f.

3	def g(x):
	  return np.sin(np.pi * x / 3) - np.cos(np.pi / 4)
	for x in x_values:
	  print(f"g({x}) = {g(x)}")
		Định nghĩa hàm số.
		
		Vòng lặp tính g.
		Hiện ra giá trị g.

4	def h(x):
	  return np.exp(x) + np.log(x**2) + 1
	for x in x_values:
	  print(f"h({x}) = {h(x)}")
		Định nghĩa hàm số.
		
		Vòng lặp tính h.
		Hiện ra giá trị h.

5	def k(x):
	  return np.sqrt(x**2 + 3*x + 9)
	for x in x_values:
	  print(f"k({x}) = {k(x)}")
		Định nghĩa hàm số.
		
		Vòng lặp tính k.
		Hiện ra giá trị k.

Bài tập 5. Vẽ đồ thị của các hàm số sau:

a) $f(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$ trên $[-10,10]$.

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np import matplotlib.pyplot as plt	Khai báo thư viện
2	def f(x): return x**4 - 2*x**3 + 3*x**2 - 4*x + 5 x = np.linspace(-10, 10, 100) y = f(x) plt.figure(figsize=(8, 6)) plt.plot(x, y, label='f(x) = x^4 - 2x^3 + 3x^2 - 4x + 5', color='blue') plt.xlabel('x') plt.ylabel('f(x)') plt.title('Đồ thị của hàm số f(x) = x^4 - 2x^3 + 3x^2 - 4x + 5 trên đoạn [-10, 10]') plt.grid(True) plt.legend() plt.show()	Định nghĩa hàm số. Tạo mảng x gồm 100 giá trị từ -10 đến 10. Gán giá trị của y. Vẽ biểu đồ. Vẽ đồ thị. Đặt tên trục x. Đặt tên trục y. Đặt tên đồ thị. Hiện lưới. Hiển thị chú thích. Hiện thị đồ thị ra màn hình.

Ghi chú: *Cách làm các câu b, c, d hoàn toàn tương tự, chỉ thay đổi hàm số.*

b) $g(x) = \sin x - 2\cos x$ trên $\left[-\pi, \frac{\pi}{2}\right]$.

c) $h(x) = (x+1)e^{x-1}$ trên $[1,5]$.

d) $k(x) = \frac{x^2-1}{2x+1}$ trên $[-3,3]$.

Bài tập 6. Cho hàm số $y = f(x) = 4x^3 - 3x^2 - 5x + 2$.

a) Tìm giá trị $y_1 = f(1), y_2 = f(2), y_3 = f(-4), y_4 = f(0)$.

b) Vẽ đồ thị hàm số trên đoạn $[-4,1]$.

c) Tính đạo hàm của hàm số tại $x = 0$.

d) Tính tích phân của hàm số trên đoạn $[-2,3]$.

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	def f(x): return 4*x**3 - 3*x**2 - 5*x + 2	Khai báo hàm số.
3	x_1 = int(input('Nhập x_1 = ')) x_2 = int(input('Nhập x_2 = ')) x_3 = int(input('Nhập x_3 = ')) x_4 = int(input('Nhập x_4 = '))	Nhập giá trị x_1=1. Nhập giá trị x_2=2. Nhập giá trị x_3=-4. Nhập giá trị x_4=0.
4	y_1 = f(x_1) y_2 = f(x_2) y_3 = f(x_3) y_4 = f(x_4)	Gán giá trị cho y_1. Gán giá trị cho y_2. Gán giá trị cho y_3. Gán giá trị cho y_4.
5	print(f"f({x_1}) = {y_1}") print(f"f({x_2}) = {y_2}") print(f"f({x_3}) = {y_3}") print(f"f({x_4}) = {y_4}")	Xuất giá trị y_1 ra. Xuất giá trị y_2 ra. Xuất giá trị y_3 ra. Xuất giá trị y_4 ra.
6	Vẽ đồ thị (các bước làm tương tự như đã hướng dẫn ở Bài tập 5) import matplotlib.pyplot as plt x = np.arange(1,5,0.05) f_x = 4*x**3 - 3*x**2 - 5*x + 2 plt.plot(x, f_x, label='f(x) = 4x^3 - 3x^2 - 5x + 2') plt.xlabel('x') plt.ylabel('f(x)') plt.title('Đồ thị của f(x)') plt.grid(True) plt.legend() plt.show()	 Khai báo thư viện. Tạo mảng cho x. Khai báo hàm f_x. Vẽ đồ thị. Đặt tên trục x. Đặt tên trục y. Tên đồ thị. Hiện lưới. Hiện chú thích. Hiển thị đồ thị ra màn hình.

7	def f(x):
	  return 4*x**3 - 3*x**2 - 5*x + 2
	def f_prime(x):
	  return 12*x**2 - 6*x - 5
	x = 0
	derivative_at_x = f_prime(x)
	print("Đạo hàm của hàm số tại x=0 là:", derivative_at_x)
		Định nghĩa hàm số.
		
		Đạo hàm của hàm số f(x).
		Tính đạo hàm của hàm số tại x=0
		Hiển thị kết quả.

8	from scipy.integrate import quad integral_value, _ = quad(f, -2, 3) print("Giá trị của tích phân của hàm số trên đoạn [-2, 3] là:", integral_value)	Khai báo thư viện. Tính tích phân của hàm số trên đoạn [-2, 3]. Xuất ra kết quả.

Chú ý: Chúng ta sử dụng thư viện sympy để tính đạo hàm và tích phân, do thư viện numpy không hỗ trợ tính chính xác đạo hàm và tích phân của hàm số.

Bài tập 7. Tính giá trị của các hàm số sau tại $x=1$, tính đạo hàm bậc 1 và bậc 2 và tích phân trên đoạn $[1,2]$.

a) $f(x)=x^5-x^3+2x-4.$

b) $g(x)=\sin\frac{\pi x}{3}-\cos\frac{\pi}{4}.$

c) $h(x)=e^x+\ln x^2+1.$

$f(1)=$	$f'(x)=$	$f''(x)=$	$\int_1^2 f dx=$
$g(1)=$	$g'(x)=$	$g''(x)=$	$\int_1^2 g dx=$
$h(1)=$	$h'(x)=$	$h''(x)=$	$\int_1^2 h dx=$

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện.

2	def f(x):
	  return x**5 - x**3 + 2*x - 4
	x = 1
	f_value = f(x)
	print("f(1) =", f_value)
	def f_prime(x):
	  return 5*x**4 - 3*x**2 + 2
		Khai báo hàm f(x).
		
		Tính giá trị của hàm số tại x=1.
		Tính đạo hàm bậc 1 của hàm số.

	f_prime_value = f_prime(x) print("Đạo hàm bậc 1 tại x=1 là:", f_prime_value) def f_double_prime(x): return 20*x**3 - 6*x f_double_prime_value = f_double_prime(x) print("Đạo hàm bậc 2 tại x=1 là:", f_double_prime_value	Tính đạo hàm bậc 1 tại x=1 và in kết quả. Tính đạo hàm bậc 2 của hàm số. Tính đạo hàm bậc 2 tại x=1

3	def g(x):
	  return np.sin(np.pi * x / 3) - np.cos(np.pi / 4)
	g_value = g(x)
	print("g(1) =", g_value)
	def g_prime(x):
	  return (np.pi/3) * np.cos(np.pi * x / 3)
	g_prime_value = g_prime(x)
	print("Đạo hàm bậc 1 tại x=1 là:", g_prime_value)
		Định nghĩa hàm g(x).
		Tính giá trị của hàm số tại x=1.
		Tính đạo hàm bậc 1 của hàm số.
		Tính đạo hàm bậc 1 tại x=1.

Bài tập 8. Viết function tính tổng và tích hai số và áp dụng với $3,4$.

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	def tongtich(x,y): return(x+y, x*y)	Viết nội dung của function tính Tổng và Tích.
2	print('(S, P) =', tongtich(3,4))	Áp dụng fuction với 3 và 4.

Bài tập 9. Viết function tìm giá trị lớn nhất của ba số và áp dụng với $4,2,–6$.

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH

1	def Max(x,y,z):
	  a = x
	  if y > a: a = y
	  if z > a: a = z
	  return(a)
		Viết nội dung của function tìm Max của ba số.

2	print('Max =',Max(4,2,-6))	Áp dụng fuction cho 4, 2, -6.

Bài tập 10. Viết function tính giai thừa của một số tự nhiên và áp dụng với số 5.

Hướng dẫn:

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH

1	def tinhGiaithua(n):
	  GT = 1
	  for i in range(1,n+1):
	  GT *= i
	  return GT
		Định nghĩa hàm.

2	n = int(input("Nhập n = "))	Nhập số cần tính giai thừa từ bàn phím.
3	print(f"{n}! = {tinhGiaithua(n)}")	Xuất kết quả ra màn hình.

Bài tập 11. Viết function giải phương trình bậc nhất một ẩn $ax = b$ và áp dụng giải $3x = 5$.
Hướng dẫn:

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THICH

1	def giaiPTB1(a,b):
	  if a == 0 and b == 0:
	  print("Phương trình có vô số nghiệm")
	  elif a == 0 and b != 0:
	  print("Phương trình vô nghiệm")
	  else:
	  print("Phương trình có nghiệm duy nhất x =",b/a)
		Định nghĩa hàm.

2	a = float(input("Nhập a: ")) b = float(input("Nhập b: "))	Nhập các hệ số của phương trình.
3	giaiPTB1(a,b)	Xuất kết quả ra màn hình.

Bài tập 12. Viết function thực hiện các yêu cầu sau đối với một hàm số bất kì.
a) Tính đạo hàm và nguyên hàm, sau đó vẽ đồ thị của chúng trên cùng một hệ trục toạ độ.
b) Tìm các điểm mà đạo hàm bằng 0.
c) Tính tích phân trên đoạn $[-5; 5]$.
d) Áp dụng với $f(x) = x^3 - 2x^2 + x - 3$
Hướng dẫn:

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np import matplotlib.pyplot as plt	Khai báo thư viện.

2	def function(x):
	  return x**3 - 2*x**2 + x - 3
	def derivative(f, x):
	  h = 1e-5
	  return (f(x + h) - f(x - h)) / (2 * h)
	
	def find_zeros(f, a, b, step=0.1):
	  zeros = []
	  for x in np.arange(a, b, step):
	  if abs(derivative(f, x)) < 1e-5:
		Khai báo hàm số.
		
		Tính đạo hàm.

	zeros.append(x) return zeros def integral(f, a, b, num_points=1000): x = np.linspace(a, b, num_points) y = f(x) dx = (b - a) / num_points return np.sum(y) * dx	
3	x = np.linspace(-5, 5, 100) y = function(x) plt.plot(x, y, label='f(x)') plt.plot(x, [derivative(function, xi) for xi in x], label="f'(x)") plt.legend() plt.show() zeros = find_zeros(function, -5, 5) print("Các điểm mà đạo hàm bằng 0 là:", zeros) integral_value = integral(function, -5, 5) print("Giá trị tích phân trên đoạn [-5, 5] là:", integral_value) Áp dụng với f(x) = x^3 - 2x^2 + x - 3	Tính đạo hàm và vẽ đồ thị. Tìm các điểm mà đạo hàm bằng 0. Tính tích phân trên đoạn [-5, 5]. Làm tương tự như trên.

2. THỰC HÀNH TÍNH TOÁN

Bài tập 13. Tính giá trị các biểu thức sau

a) $A = 2^3 - \dfrac{(1+2)(2+3)}{3+4} + \dfrac{\sqrt{2}}{\sqrt[4]{3}}.$

b) $B = \sin\dfrac{\pi}{3} - 2\cos\dfrac{\pi}{4} + \dfrac{3\tan\dfrac{\pi}{6}}{2 - \cot\dfrac{5\pi}{6}}.$

c) $C = e^{-\sqrt{2}} - \ln\dfrac{3}{2} + \ln(e+2).$

d) $D = \dfrac{2A+3B}{C^2 - 2C}.$

Bài tập 14. Cho $a = 2, b = 3, c = 1$. Tính giá trị các biểu thức sau

a) $A = \dfrac{b + \sqrt{b^2 - 4ac}}{2a}.$

b) $B = \left[a\sin b\cos c, a\sin b\sin c, a\cos b\right].$

c) $C = \left[\frac{a+b}{a+b+c}, \frac{a-b+c}{a+b+c}, \frac{c^2-ab}{a+b+c}, \frac{1}{a+b+c} \right]$.

d) $D = AB - C$.

Bài tập 15. Cho hàm số $f(x) = x.\sin x$. Hãy tính giá trị của f tại $x = 1$ và $x = 3$ và vẽ đồ thị f trên $[-2, 4]$.

a) Sử dụng khai báo kiểu thay thế.

b) Sử dụng khai báo kiểu hàm số.

Bài tập 16. Cho hàm số $f(x, y) = |x| + 2|y|$. Hãy tính giá trị của f tại $(1, 2)$ và $\left(\sqrt{2}, e^{-1}\right)$ và vẽ đồ thị f trên $[-2, 4] \times [-3, 3]$.

a) Sử dụng khai báo kiểu thay thế.

b) Sử dụng khai báo kiểu hàm số.

Bài tập 17. Thực hiện các yêu cầu sau

a) Cho $f(x) = x^2 + 2x - 4$. Tính $f'(x), f'(2), \int f(x)dx, \int_0^1 f(x)dx$.

b) Cho $g(x) = \frac{x^2+1}{x+1}$. Tính $g''(x), g''(1), \int g(x)dx, \int_{-1}^1 g(x)dx$.

c) Cho $h(x) = \sin 2x$. Tính $h'(x), h'(0), \int h(x)dx, \int_0^\infty h(x)dx$.

Bài tập 18. Thực hiện các yêu cầu sau

a) Viết function giải phương trình bậc hai $ax^2 + bx + c = 0$.

b) Áp dụng để giải phương trình $2x^2 - 3x + 1 = 0$.

Bài tập 19. Thực hiện các yêu cầu sau

a) Viết function để tìm cực trị của một hàm số bất kì.

b) Áp dụng để tìm cực trị của hàm số $f(x) = x^3 - 6x$.

Bài tập 20. Thực hiện các yêu cầu sau

a) Viết function tính các đạo hàm riêng cấp hai của hàm số 2 biến $f(x, y)$ và kiểm tra biểu thức $f_{xy} = f_{yx}$.

b) Áp dụng để tính đạo hàm riêng cấp hai của hàm số $f(x, y) = \frac{x}{y} \sin \frac{y}{x}$.

BÀI THỰC HÀNH SỐ 1
SAI SỐ TRONG TÍNH TOÁN

1 Tóm tắt lý thuyết

1.1 Khái niệm cơ bản

Xét một đại lượng p có giá trị chính xác là p^* và giá trị gần đúng (xấp xỉ) là $\overline{p}$, ta có các định nghĩa sau:

a) Sai số tuyệt đối: Δp

$$\Delta p = \left| p^* - \overline{p} \right|$$

b) Sai số tương đối δp

$$\delta p = \left| \frac{p^* - \overline{p}}{p^*} \right| = \frac{\Delta p}{\left| p^* \right|}$$

trong một số trường hợp, mẫu thức được thay bởi $\overline{p}$

1.2 Sai số do làm tròn

Giả sử đại lượng p có giá trị chính xác là một số thập phân vô hạn như sau

$$p^* = s_m s_{m-1} \ldots s_1 s_0 . s_{-1} s_{-2} \ldots s_n s_{n-1} \ldots$$

Khi đó, trong quá trình tính toán, ta phải tìm biểu diễn gần đúng của nó dưới dạng

$$p^* = s_m s_{m-1} \ldots s_1 s_0 . s_{-1} s_{-2} \ldots \overline{s_n}$$

trong đó $\overline{s_n}$ được tính bằng các phương pháp sau

a) Phương pháp làm tròn

$$\overline{s_n} = \begin{cases} s_n + 1, & s_{n-1} \geq 5 \\ s_n, & s_{n-1} < 5 \end{cases}$$

b) Phương pháp chặt cụt

$$\overline{s_n} = s_n$$

1.3 Sai số giới hạn

Xét một đại lượng p, người ta thường đề cập đến giá trị tiêu chuẩn $\overline{p}$ và giá trị thực tế p^* thỏa quan hệ :

$$\overline{p} - \overline{\Delta} p \leq p^* \leq \overline{p} + \overline{\Delta} p \text{ hay } p^* \in \left[\overline{p} - \overline{\Delta} p, \overline{p} + \overline{\Delta} p \right]$$

Khi đó $\overline{\Delta} p$ là là sai số tuyệt đối giới hạn trong phép tính p.

1.4 Sai số trong tính toán

Cho hàm số y phụ thuộc vào các biến $x_1, x_2, ..., x_n$ với x_i có giá trị gần đúng $\overline{x_i}$ và sai số Δx_i. Khi đó sai số của $y(x_1, x_2, ..., x_n)$ được tính như sau

$$\Delta y = \sum_{i=1}^{n} \left| \frac{\partial y}{\partial x_i}(\bar{x}_1, \bar{x}_2, ..., \bar{x}_n) \right| \Delta x_i$$

2 Thực hành trên máy tính

2.1 Xác định sai số tuyệt đối, sai số tương đối

Bài tập 1. Tính sai số tuyệt đối Δp và sai số tương đối δp của các đại lượng có giá trị chính xác p^* và giá trị gần đúng $\bar{p}$

p^*	$\bar{p}$	Δp	δp
0.9857	0.9768		
421	397		
1102	1113		
2.5743	2.6381		

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện.
2	p_true = np.array([0.9857, 421, 1102, 2.5743]) p_approx = np.array([0.9768, 397, 1113, 2.6381]) delta_p = np.abs(p_true - p_approx) delta_p_ratio = (delta_p / np.abs(p_true)) * 100 print("Sai số tuyệt đối Delta p:") print(delta_p) print("\nSai số tương đối delta p (%):") print(delta_p_ratio)	Dữ liệu đầu vào. Tính sai số tuyệt đối và sai số tương đối. Hiển thị kết quả.

Bài tập 2. Tính sai số tuyệt đối Δp và sai số tương đối δp của các đại lượng có giá trị chính xác p^* và giá trị gần đúng $\bar{p}$

p^*	$\bar{p}$	Δp	δp
0.9857564312	0.9768463123		
42189376	39773891		
1102.34598	1113.24691		
2.574314893	2.638100358		

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	p_true = np.array([0.9857564312, 42189376, 1102.34598, 2.574314893]) p_approx = np.array([0.9768463123, 39773891, 1113.24691, 2.638100358]) delta_p = np.abs(p_true - p_approx) delta_p_ratio = (delta_p / np.abs(p_true)) * 100 for i in range(len(p_true)): print(f"Đối với p* = {p_true[i]} và p_bar = {p_approx[i]}:")	Dữ liệu đầu vào. Tính sai số tuyệt đối và sai số tương đối.

	print(f"Sai số tuyệt đối Delta p: {delta_p[i]}") print(f"Sai số tương đối delta p (%): {delta_p_ratio[i]}\n")	Tạo vòng lặp và hiển thị kết quả.

Bài tập 3. Tính sai số tuyệt đối và sai số tương đối của các đại lượng sau khi sử dụng giá trị xấp xỉ (3 số thập phân) bằng phương pháp làm tròn và phương pháp chặt cụt

p^*	$\overline{p}_1$ (làm tròn)	Δp_1	δp_1	$\overline{p}_2$ (chặt cụt)	Δp_2	δp_2
π						
e						
$\ln 2$						
$\sqrt{2}$						
$\sin 1$						

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	p_true = np.array([np.pi, np.e, np.log(2), np.sqrt(2), np.sin(1)]) p_rounded = np.round(p_true, 3) delta_p_rounded = np.abs(p_true - p_rounded) delta_p_ratio_rounded = (delta_p_rounded / np.abs(p_true)) * 100 p_truncated = np.trunc(p_true * 1000) / 1000 delta_p_truncated = np.abs(p_true - p_truncated) delta_p_ratio_truncated = (delta_p_truncated / np.abs(p_true)) * 100 for i in range(len(p_true)): print(f"Đối với p* = {p_true[i]}:") print(f"Giá trị xấp xỉ sau làm tròn: {p_rounded[i]}") print(f"Sai số tuyệt đối Delta p (làm tròn): {delta_p_rounded[i]}") print(f"Sai số tương đối delta p (%) (làm tròn): {delta_p_ratio_rounded[i]}") print(f"Giá trị xấp xỉ sau chặt cụt: {p_truncated[i]}") print(f"Sai số tuyệt đối Delta p (chặt cụt): {delta_p_truncated[i]}") print(f"Sai số tương đối delta p (%) (chặt cụt): {delta_p_ratio_truncated[i]}\n")	Định nghĩa các giá trị đúng. Làm tròn giá trị xấp xỉ. Chặt cụt giá trị xấp xỉ. Tạo vòng lặp và hiển thị kết quả.

Bài tập 4. Thực hiện lại bài tập với số thập phân được cho trước.

p^*	$\overline{p}_1$ (làm tròn)	Δp_1	δp_1	$\overline{p}_2$ (chặt cụt)	Δp_2	δp_2
π						
e						
$\ln 2$						
$\sqrt{2}$						
$\sin 1$						

2.2 Kiểm tra quan hệ giữa giá trị chính xác và giá trị xấp xỉ

Bài tập 5. Kiểm tra các đại lượng có phù hợp với đánh giá (dựa trên giá trị tiêu chuẩn $\overline{p}$ và sai số giới hạn $\overline{\Delta p}$)?

p^*	$\overline{p}$	Δp	p_L	p_R	Kết quả
17.351	15.932	1.247			
11205	11115	120			
38.735	36.215	1.327			
319	297	15			

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	p_star = np.array([17.351, 11205, 38.735, 319])
	p_bar = np.array([15.932, 11115, 36.215, 297])
	delta_p_bar = np.array([1.247, 120, 1.327, 15])
	
	p_L = p_bar - delta_p_bar
	p_R = p_bar + delta_p_bar
	
	print("Kết quả:")
	for i in range(len(p_star)):
	  print(f"Đối với p* = {p_star[i]}:")
	  print(f"p_L = {p_L[i]}, p_R = {p_R[i]}\n")
		Dữ liệu đầu vào
		
		Tính p_L và p_R
		
		Hiển thị kết quả

Bài tập 6. Kiểm tra các đại lượng có phù hợp với đánh giá (dựa trên giá trị tiêu chuẩn p^* và sai số giới hạn $\overline{\delta p}$)?

p^*	$\overline{p}$	δp	p_L	p_R	Kết quả
218	200	0.05			
6.035	5.897	0.02			
2545	2300	0.1			
37.54	35.89	0.03			

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	p_star = np.array([218, 6.035, 2545, 37.54]) p_bar = np.array([200, 5.897, 2300, 35.89]) delta_p_bar = np.array([0.05, 0.02, 0.1, 0.03]) p_L = p_bar - delta_p_bar p_R = p_bar + delta_p_bar result = np.where((p_star >= p_L) & (p_star <= p_R), "Phù hợp", "Không phù hợp") print("Kết quả:") for i in range(len(p_star)): print(f"Đối với p* = {p_star[i]}:")	Dữ liệu đầu vào Tính p_L và p_R Kiểm tra các đại lượng có phù hợp với đánh giá Hiển thị kết quả

	print(f"p_L = {p_L[i]}, p_R = {p_R[i]}, Kết quả: {result[i]}\n")	

2.3 Tính toán sai số trong biểu thức toán học

Bài tập 7. Tìm sai số tuyệt đối của hàm số sau

$y = f(x_1, x_2, x_3)$	$\bar{x}_1$	Δx_1	$\bar{x}_2$	Δx_2	$\bar{x}_3$	Δx_3	Δy	δy
$y = x_1 + x_2.x_3$	5	0.03	3	0.06	7	0.04		
$y = x_1^2 + x_2.x_3^3$	2	0.05	4	0.02	6	0.03		
$y = x_3 + \sqrt{x_1 + x_2}$	3	0.05	7	0.07	3	0.02		
$y = x_1.x_2 / x_3$	3	0.08	7	0.03	10	0.1		
$y = x_1.(x_2 + x_3) - x_2x_3$	8	0.09	4	0.02	3	0.04		
$y = \ln(x_1.x_2 - x_3)$	7	0.05	5	0.02	2	0.03		
$y = x_1 \sin x_2 - \cos x_3$	3	0.06	0	0.02	1	0.04		

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	x1_bar, delta_x1 = 5, 0.03 x2_bar, delta_x2 = 3, 0.06 x3_bar, delta_x3 = 7, 0.04 y_bar = x1_bar + x2_bar * x3_bar partial_y_x1 = 1 partial_y_x2 = x3_bar partial_y_x3 = x2_bar delta_y = abs(partial_y_x1) * delta_x1 + abs(partial_y_x2) * delta_x2 + abs(partial_y_x3) * delta_x3 delta_y_relative = delta_y / abs(y_bar) print(f"Sai số tuyệt đối Δy: {delta_y}") print(f"Sai số tương đối δy: {delta_y_relative}")	Giá trị trung bình và sai số của các biến Tính giá trị của y Tính các đạo hàm riêng Tính sai số tuyệt đối Tính sai số tương đối In kết quả

3 Lập trình tính toán

Bài tập 8. Function sau được viết để tính sai số tuyệt đối và sai số tương đối giữa số chính xác và số gần đúng.

```
function[aEp,rEp]=saiso(p_e,p_a)
aEp = abs(p_e - p_a);
rEp = aEp/abs(p_e);
```

a) Cho biết input và output là gì?

b) Function này được viết ở đâu? đặt tên là gì?

c) Khi muốn function này được thực hiện, phải làm gì? như thế nào?

d) Hãy sử dụng function này để thực hiện lại bài tập 1.

Bài tập 9. Hãy viết function để tính số gần đúng do của đại lượng có vô hạn số thập phân
a) Cho biết input và output là gì?
b) Sử dụng phương pháp làm tròn.
c) Sử dụng phương pháp chặt cụt.
d) Hãy sử dụng function này để thực hiện lại bài tập 4.

Bài tập 10. Hãy viết function để kiểm tra số chính xác có phù hợp với đánh giá ($\overline{p}$ và $\overline{\Delta}p$) hay không?
a) Cho biết input và output là gì?
b) Hãy sử dụng function này để thực hiện lại bài tập 5.
c) Hãy sử dụng function này để kiểm tra sự phù hợp của một hộp bánh có khối lượng 438 g với tiêu chuẩn 425 ± 15 g.
d) Hãy sử dụng function này để kiểm tra sự phù hợp của một toa xe lửa có chiều dài 15 659 cm với tiêu chuẩn 15 586 ± 123 cm.

Bài tập 11. Hãy viết function để kiểm tra số chính xác có phù hợp với đánh giá ($\overline{p}$ và $\overline{\delta}p$) hay không?
a) Cho biết input và output là gì?
b) Hãy sử dụng function này để thực hiện lại bài tập 6.
c) Hãy sử dụng function này để kiểm tra sự phù hợp của một tuýp kem đánh răng có khối lượng 138 g với tiêu chuẩn 135 ± 3% cm.
c) Hãy sử dụng function này để kiểm tra sự phù hợp của một chai nước mắm có độ đạm $43,789^o$ với tiêu chuẩn 42,5 ± 5% cm.

Bài tập 12. Hãy viết function tính sai số tuyệt đối và tương đối của biểu thức toán học
a) Biểu thức chứa hai biến.
b) Biểu thức chứa ba biến.
c) Biểu thức chứa n biến.
d) Hãy sử dụng function này để thực hiện lại bài tập 7.

Bài tập 13. Tìm giá trị hàm số u (lấy 3 chữ số thập phân) và tính sai số tuyệt đối giới hạn, sai số tương đối giới hạn do việc làm tròn số tại các điểm cho trước
a) $u = ln(2y + x^2)$ tại x = 1,976; xy = 0,532.
b) $u = ye^x - x^2$ tại x = 1,675; y = 1,073.
c) $u = x\,tan\,y + (x + y)^2$ tại x = −1,395; y = 1,643.
d) Viết function tổng quát cho các bài toán trên.

BÀI THỰC HÀNH SỐ 2

GIẢI PHƯƠNG TRÌNH SIÊU VIỆT (PHẦN 1)

1. Tóm tắt lý thuyết

1.1 Phương pháp chia đôi

Xét hàm số f(x) liên tục trên khoảng [a, b] sao cho f(a) · f(b) < 0. Thuật toán phương pháp chia đôi để giải phương trình f(x) = 0 được trình bày như sau

Thuật toán phương pháp chia đôi
Bước 1: Khai báo hàm số f(x).
Bước 2: Nhập a và b đồng thời kiểm tra f(a) · f(b) < 0.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 3: Gán $c = \frac{a+b}{2}$.
Nếu |f(c)| < Δf phá vòng lặp.
Bước 4: Nếu f(a) · f(c) > 0 thì gán a = c. Ngược lại, gán b = c.
Gán k = k + 1
(Đóng vòng lặp)
Kết luận $\bar{x}$ = c.

Nhận xét: tốc độ hội tụ của thuật toán khá chậm vì khoảng nghiệm sau mỗi bước lặp chỉ giảm một nửa. Điểm mạnh của phương pháp này là luôn chỉ ra được nghiệm của bài toán

1.2 Phương pháp lặp

Xét hàm số f(x) liên tục trên khoảng [a, b] sao cho f(a) · f(b) < 0. Thuật toán phương pháp lặp để giải phương trình f(x) = 0 được trình bày như sau

Thuật toán phương pháp lặp
Bước 1: Khai báo hàm f(x), ϕ(x).
Bước 2: Nhập x_0.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 3: Gán $x_k = \phi(x_{k-1})$.
Nếu $|f(x_k)| < \Delta f$ phá vòng lặp.
Bước 4: Gán k = k + 1.
(Đóng vòng lặp)
Kết luận $\bar{x} = x_k$.

Nhận xét: tốc độ hội tụ của thuật toán phụ thuộc vào hàm ϕ(x). Nếu giá trị của đạo hàm của ϕ(x) càng tiến về 0 thì thuật toán hội tụ càng nhanh. Nếu giá trị của đạo hàm lớn hơn 1, thuật toán không cho ra nghiệm.

2 Thực hành trên máy tính.

2.1 Giải phương trình bằng phương pháp chia đôi.

Bài tập 1: Giải phương trình $x + \sin x - 2 = 0$ bằng phương pháp chia đôi với a =1, b =1.4, $\Delta f = 10^{-3}$.

STT	a	b	c	f(c)	$\lvert f(c)\rvert \le \Delta f$	δc

1	1.0	1.4				
2						
3						
4						
5						
6						

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def bisection_method(f, a, b, tol=1e-3, max_iter=100):
	  if f(a) * f(b) >= 0:
	  print("Phương pháp chia đôi không áp dụng được vì f(a) và f(b) cùng dấu.")
	  return None
	
	  iter_count = 0
	  while (b - a) / 2.0 > tol and iter_count < max_iter:
	  iter_count += 1
	  midpoint = (a + b) / 2.0
	  if f(midpoint) == 0:
	
	  return midpoint
	  elif f(a) * f(midpoint) < 0:
	  b = midpoint
	  else:
	  a = midpoint
	
	  return (a + b) / 2.0
	
	def f(x):
	  return x + np.sin(x) - 2
	
	a = 1
	b = 1.4
	tolerance = 1e-3
	
	root = bisection_method(f, a, b, tol=tolerance)
	
	if root is not None:
	  print(f"Nghiệm của phương trình là: {root}")
	else:
	  print("Không tìm thấy nghiệm trong khoảng đã cho.")
		
		Nếu midpoint là nghiệm chính xác
		
		Định nghĩa hàm số f(x) = x + sin(x) – 2
		
		Khoảng [a, b] và sai số cho phép
		
		Gọi hàm bisection_method

2.2 Giải bằng phương pháp lặp

Bài tập 2. Giải phương trình $x + \sin x - 2 = 0$ bằng phương pháp lặp với $\varphi(x) = 2 - \sin x$ và $x = 1.05,\ \Delta f = 10^{-3}$.

STT	x	f(x)	$\lvert f(x)\rvert \le \Delta f$	δx_n
1	1.05			
2				
3				
4				
5				
6				

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def fixed_point_iteration(phi, x0, tol=1e-3, max_iter=100):
	  x = x0
	  for i in range(max_iter):
	  x_new = phi(x)
	  if abs(x_new - x) < tol:
	  return x_new
	  x = x_new
	  return x
	
	def phi(x):
	  return 2 - np.sin(x)
	
	x0 = 1.05
	tolerance = 1e-3
	
	root = fixed_point_iteration(phi, x0, tol=tolerance)
	
	print(f"Nghiệm của phương trình là: {root}")
		
		Định nghĩa hàm lặp phi(x) = 2 - sin(x)
		
		Giá trị ban đầu và sai số cho phép
		
		Gọi hàm fixed_point_iteration

Bài tập 3. Giải phương trình $x^2 + x = 5$ bằng phương pháp lặp với $\varphi(x) = 5 - x^2$ và $x = 1.5,\ \Delta f = 10^{-3}$

STT	x	f(x)	$f(x) \le \Delta f$	δx
1	1.5			
2				
3				

Giải thích tại sao giá trị f(x) càng lúc càng tăng. Phải thay đổi điều gì để giải được phương trình

3. **Lập trình tính toán**

Bài tập 4. Function giải phương trình đại số và siêu việt bằng phương pháp chia đôi được trình bày như sau

```
import matplotlib.pyplot as plt
def chiadoi(f, a, b, Df):
    k = 1
    c_values = []
    fc_values = []
    while True:
        c = (a + b) / 2
        fc = f(c)
        c_values.append(c)
        fc_values.append(fc)
        print(f"Iteration {k}: c = {c}, f(c) = {fc}")
        if abs(fc) < Df:
            break
        if f(a) * f(c) > 0:
            a = c
        else:
            b = c
        k += 1
    plt.plot(range(1, k + 1), fc_values, 'ro-')
    plt.xlabel('Iteration (k)')
    plt.ylabel('f(c)')
    plt.title('Bisection Method')
    plt.grid(True)
    plt.show()
    return c, fc
```

a) Khi chạy chương trình trên (với đầy đủ input) thì trên COMMAND WINDOW sẽ xuất hiện gì?

b) Hình vẽ xuất hiện có trục x thể hiện điều gì và trục Oy thể hiện điều gì?

c) Thực hiện lại bài tập 1.

Bài tập 5. Sử dụng function được viết ở Bài tập 4 để giải bài toán $e^x - x = 3$ với các input sau

a) a = 0, b = 3, $\Delta f = 10^{-3}$

b) a = 0, b = 2, $\Delta f = 5.10^{-3}$

c) a = −3, b = 0, $\Delta f = 10^{-4}$

d) a = −3, b = −1, $\Delta f = 10^{-4}$

.

Bài tập 6. Xét bài toán tìm nghiệm của phương trình đại số và siêu việt bằng phương pháp lặp.

a) Viết function cho bài toán trên với output là nghiệm bài toán và input là hàm số f, hàm lặp φ, tọa độ hai cận a, b, tọa độ nghiệm thử ban đầu x_0, sai số giới hạn Δf.

b) Thêm các dòng lệnh để function tự động in ra bảng giá trị và vẽ đồ thị của nghiệm của phương trình.

c) Thực hiện lại Bài tập 2.

d) Thực hiện lại Bài tập 3.

Bài tập 7. Sử dụng function được viết ở Bài tập 6 để giải bài toán x − x/2 = 1/x với $\varphi(x) = x/2 + 1/x$ và các input sau

a) $x_0 = 1, \Delta f = 10^{-3}$.

b) $x_0 = 2, \Delta f = 3 \cdot 10^{-3}$.

c) $x_0 = -2, \Delta f = 10^{-2}$.

d) $x_0 = -5, \Delta f = 10^{-4}$.

Bài tập 8. Sử dụng phương pháp chia đôi và phương pháp lặp để giải quyết các bài tập sau với cùng $\Delta f = 10^{-3}$

a) $e^x + 2^{-x} + 2 \cos x = 6$.

b) $\ln(x - 1) + \cos(x - 1) = 0$.

c) $(x - 2)^2 - \ln x = 0$.

d) $\sin x = e^{-x}$.

BÀI THỰC HÀNH SỐ 3
GIẢI PHƯƠNG TRÌNH SIÊU VIỆT (PHẦN 2)

1. Tóm tắt lý thuyết
1.1 Phương pháp tiếp tuyến

Xét hàm số f(x) liên tục trên khoảng [a, b] sao cho f(a) · f(b) < 0. Thuật toán phương pháp tiếp tuyến để giải phương trình f(x) = 0 được trình bày như sau

Thuật toán phương pháp tiếp tuyến
Bước 1: Khai báo hàm số f(x) và đạo hàm f'(x).
Bước 2: Nhập x_0.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 3: Gán $x_k = x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$
Nếu $|f(x_k)| < \Delta f$ phá vòng lặp.
Bước 4: Gán k = k + 1
(Đóng vòng lặp)
Kết luận $\bar{x} = x_k$

Nhận xét: tốc độ hội tụ của thuật toán nhanh. Tuy nhiên đạo hàm cấp hai của hàm số phải không đổi dấu trên [a; b] thì thuật toán mới cho nghiệm hội tụ

1.2 Phương pháp dây cung

Xét hàm số f(x) liên tục trên khoảng [a; b] sao cho f(a) · f(b) < 0. Thuật toán phương pháp dây cung để giải phương trình f(x) = 0 được trình bày như sau

Thuật toán phương pháp dây cung
Bước 1: Khai báo hàm số f(x).
Bước 2: Nhập a và b đồng thời kiểm tra f(a) · f(b) < 0.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 3: Gán $c = a - \dfrac{b-a}{f(b)-f(a)} f(a).$
Nếu $|f(c)| < \Delta f$ phá vòng lặp.
Bước 4: Nếu f(a) · f(c) > 0 thì chọn a = c. Ngược lại, chọn b = c.
Gán k = k + 1
(Đóng vòng lặp)
Kết luận x = c.

Nhận xét: Phương pháp dây cung có tốc độ hội tụ trung bình

4. Thực hành trên máy tính
4.1 Giải phương trình bằng phương pháp tiếp tuyến

Bài tập 1. Giải phương trình $x^2 - \sin x = 50$ bằng phương pháp tiếp tuyến với $x_0 = 2, \Delta f = 10^{-3}$.

STT	x_0	x_n	$f(x_n)$	$f(x_n) \le \Delta f$	δx_n
1	2				
2					

3					
4					

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def f(x):
	  return x**2 - np.sin(x) - 50
	
	def f_prime(x):
	  return 2*x - np.cos(x)
	
	x0 = 2
	delta_f = 1e-3
	
	print("STT\t$x_{0}$\t$x_{n}$\t$f(x_{n})$\t$\|f(x_{n})\| \leq \Delta f$\t$\delta x_{n}$")
	
	n = 1
	while True:
	  fn = f(x0)
	  f_prime_n = f_prime(x0)
	  xn = x0 - fn / f_prime_n
	  delta_xn = np.abs(xn - x0)
	
	  print(f"{n}\t{x0:.6f}\t{xn:.6f}\t{fn:.6f}\t{np.abs(fn) <= delta_f}\t{delta_xn:.6f}")
	
	  if np.abs(fn) <= delta_f:
	  break
	
	  x0 = xn
	  n += 1
		Định nghĩa hàm f(x) và đạo hàm f'(x)
		
		Điểm bắt đầu và độ chính xác mong muốn
		In ra tiêu đề của bảng
		
		Bắt đầu lặp

Bài tập 2. Giải phương trình $x^3 - 6x^2 + 2x + 25 = 0$ bằng phương pháp tiếp tuyến với $x_0 = 4, \Delta f = 10^{-3}$.

STT	x_0	x_n	$f(x_n)$	$f(x_n) \le \Delta f$	δx_n
1	4				
2					
3					
4					

4.2 **Giải phương trình bằng phương pháp dây cung**

Bài tập 3. Giải phương trình $x^2 - \sin x = 50$ bằng phương pháp dây cung với a = 0, b = 8, $\Delta f = 3.10^{-3}$

STT	a	b	c	$f(c)$	$f(c) \le \Delta f$	δc
1						

2						
3						
4						
5						

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def f(x):
	  return x**2 - np.sin(x) - 50
	
	a = 0
	b = 8
	delta_f = 3e-3
	
	print("STT\t$a$\t$b$\t$c$\t$f(c)$\t$\|f(c)\| \leq \Delta f$\t$\delta c$")
	
	n = 1
	while True:
	  c = a - f(a) * (b - a) / (f(b) - f(a))
	  fc = f(c)
	  delta_c = np.abs(c - a)
	
	  print(f"{n}\t{a:.6f}\t{b:.6f}\t{c:.6f}\t{fc:.6f}\t{np.abs(fc) <= delta_f}\t{delta_c:.6f}")
	
	  if np.abs(fc) <= delta_f:
	  break
	
	  if f(a) * fc < 0:
	  b = c
	  else:
	  a = c
	
	  n += 1
		Định nghĩa hàm f(x)
		
		Khoảng ban đầu [a, b] và độ chính xác mong muốn
		
		In ra tiêu đề của bảng
		
		Bắt đầu lặp

5. Lập trình tính toán

Bài tập 4. Xét bài toán tìm nghiệm của phương trình đại số và siêu việt bằng phương pháp tiếp tuyến

a) Viết function cho bài toán trên với output là nghiệm bài toán và input là hàm số f, đạo hàm df, tọa độ nghiệm thử ban đầu x_0 và sai số giới hạn Δf.

b) Thêm các dòng lệnh để function tự động in ra bảng giá trị và vẽ đồ thị của nghiệm của phương trình.

c) Sử dụng function trên để giải Bài toán 1.

d) Sử dụng function trên để giải Bài toán 2.

Bài tập 5. Sử dụng function được viết ở Bài toán 4 để giải phương trình x + ln(x + 2) − 10 với các input sau

a) $x_0 = 7; \Delta f = 10^{-3}$.
b) $x_0 = 9; \Delta f = 2 \cdot 10^{-3}$.
c) $x_0 = 5; \Delta f = 5 \cdot 10^{-3}$.
d) $x_0 = 3; \Delta f = 5 \cdot 10^{-3}$.

Bài tập 6. Xét bài toán tìm nghiệm của phương trình đại số và siêu việt bằng phương pháp dây cung.

a) Viết function cho bài toán trên với output là nghiệm bài toán và input là hàm số f, tọa độ hai cận a; b và sai số giới hạn Δf.

b) Thêm các dòng lệnh để function tự động in ra bảng giá trị và vẽ đồ thị của nghiệm của phương trình.

Bài tập 7. Sử dụng function được viết ở Bài toán 6 để giải phương trình $2^x + 3^x - 10x = 30$ với các input sau

a) $a = -5; b = -3; \Delta f = 10^{-3}$.
b) $a = -4; b = -2; \Delta f = 2 \cdot 10^{-3}$.
c) $a = 2; b = 4; \Delta f = 3 \cdot 10^{-3}$.
d) $a = 2; b = 4; \Delta f = 3 \cdot 10^{-3}$.

Bài tập 8. Sử dụng phương pháp tiếp tuyến và phương pháp dây cung để giải quyết các bài tập sau với cùng $\Delta f = 10^{-3}$

a) $e^x + 2^{-x} + 2 \cos x = 6$.

b) $\ln(x - 1) + \cos(x - 1) = 0$.

c) $(x - 2)^2 - \ln x = 0$.

d) $\sin x = e^{-x}$.

Bài tập 9. Một cái xà nhà chịu lực có độ võng w phụ thuộc vào vị trí $x \in [0; L]$ được cho bởi phương trình

$$w(x) = \frac{w_0}{120EIL}(-x^5 + 2L^2x^3 - L^4x)$$

trong đó E = 50.000 là độ cứng, I = 30.000 là tiết diện, L = 600 là chiều dài và $w_0 = 2,5$ là hệ số vật liệu của xà nha. Để tìm độ võng lớn nhất của xà nhà, người ta thực hiện như sau

Bước 1: Tính đạo hàm w'(x) và vẽ đồ thị của nó để tìm khoảng phân ly nghiệm.

Bước 2: Giải phương trình w'(x) = 0 để tìm vị trí mà độ võng đạt cực trị.

Bước 3: So sánh độ võng tại cực trị và biên (x = 0 và x = L) để tìm độ võng lớn nhất.
Hãy thực hiện các bước trên và tìm độ võng cực đại.x

BÀI THỰC HÀNH SỐ 4

GIẢI HỆ PHƯƠNG TRÌNH ĐẠI SỐ TUYẾN TÍNH

1 Tóm tắt lý thuyết

Xét hệ phương trình đại số tuyến tính có dạng AX = C trong đó A là ma trận vuông cấp n chứa các hệ số đứng trước các biến, C là vector n dòng chứa giá trị của các biểu thức chứa biến và X là vector n dòng chứa các biến cần tìm.

1.1 Phương pháp lặp

Thuật toán phương pháp lặp để giải hệ phương trình trên được trình bày như sau:

Thuật toán phương pháp lặp
Bước 1: Khai báo ma trận vuông A và vector dòng C.
Bước 2: Tính ma trận B = −A./diag(A) + In và vector dòng G = C./diag(A).
Chọn nghiệm thử ban đầu X0 = G.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 3: Gán Xk = BXk−1 + G.
Nếu |AXk − C| < ΔF thì phá vòng lặp.
Bước 4: Gán k = k + 1
(Đóng vòng lặp)
Kết luận $X = X_k$

Nhận xét: tốc độ hội tụ của thuật toán chậm.

1.2 Phương pháp lặp Seidel

Thuật toán phương pháp lặp Seidel để giải hệ phương trình trên được trình bày như sau

Thuật toán phương pháp lặp Seidel
Bước 1: Khai báo ma trận vuông A và vector dòng C.
Bước 2: Tính ma trận B = −A./diag(A) + In và vector dòng G = C./diag(A).
Chọn nghiệm thử ban đầu X0 = G.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 3: Gán $(X_k)_1 = \sum_{j=1}^{n} B_{1j}(X_{k-1})_j + G_1$

Gán $(X_k)_i = \sum_{j=1}^{i-1} B_{ij}(X_k)_j + \sum_{j=i}^{n} B_{ij}(X_{k-1})_j + G_i$ với $2 \le i \le n$

Nếu |A X_k − C| < ΔF thì phá vòng lặp.
Bước 4: Gán k = k + 1
(Đóng vòng lặp)
Kết luận $\overline{X} = X_k$

Nhận xét: Phương pháp Seidel là sự cải tiến của phương pháp lặp nên thuật toán phức tạp hơn nhưng có tốc độ hội tụ nhanh.

2 Thực hành trên máy tính

2.1 Giải hệ phương trình bằng phương pháp lặp

Bài tập 1. Giải hệ phương trình sau bằng phương pháp lặp với $\Delta F = 10^{-3}$

$$\begin{cases} 5x_1 + x_2 + x_3 = 7 \\ x_1 + 10x_2 + x_3 = 12 \\ x_1 + x_2 + 20x_3 = 22 \end{cases}$$

STT	X_1	X_2	X_3	F(X)	$\lvert F(X)\rvert \le \Delta F$	δX_n
0						
1						
2						
3						
4						
5						
6						

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	A = np.array([[5, 1, 1],
	[1, 10, 1],
	[1, 1, 20]], dtype=float)
	
	B = np.array([7, 12, 22], dtype=float)
	
	tolerance = 1e-3
	max_iterations = 100
	
	X = np.zeros_like(B)
	for _ in range(max_iterations):
	X_new = np.zeros_like(X)
	
	for i in range(len(A)):
	sum1 = sum(A[i][j] * X[j] for j in range(len(A)) if j != i)
	X_new[i] = (B[i] - sum1) / A[i][i]
	
	if np.linalg.norm(X_new - X, ord=np.inf) < tolerance:
	break
	
	X = X_new
	
	print("Nghiệm của hệ phương trình:", X)
		Khởi tạo ma trận hệ số A và vector kết quả B
		
		Điều kiện dừng
		
		Khởi tạo nghiệm ban đầu
		
		Phương pháp lặp
		
		Kiểm tra điều kiện dừng
		
		In kết quả

2.2 Giải hệ phương trình bằng phương pháp lặp Seidel

Bài tập 2. Giải hệ phương trình sau bằng phương pháp lặp với $\Delta F = 10^{-3}$

$$\begin{cases} 5x_1 + x_2 + x_3 = 7 \\ x_1 + 10x_2 + x_3 = 12 \\ x_1 + x_2 + 20x_3 = 22 \end{cases}$$

STT	X_1	X_2	X_3	F(X)	$\lvert F(X)\rvert \le \Delta F$	δX_n
0						
1						
2						
3						
4						
5						
6						

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	A = np.array([[5, 1, 1],
	[1, 10, 1],
	[1, 1, 20]], dtype=float)
	
	B = np.array([7, 12, 22], dtype=float)
	
	tolerance = 1e-3
	max_iterations = 100
	
	X = np.zeros_like(B)
	
	for _ in range(max_iterations):
	X_new = np.copy(X)
	
	for i in range(len(A)):
	sum1 = sum(A[i][j] * X_new[j] for j in range(len(A)) if j != i)
	X_new[i] = (B[i] - sum1) / A[i][i]
	
	if np.linalg.norm(X_new - X, ord=np.inf) < tolerance:
	break
	
	X = X_new
	
	print("Nghiệm của hệ phương trình:", X)
		Khởi tạo ma trận hệ số A và vector kết quả B
		
		Điều kiện dừng
		
		Khởi tạo nghiệm ban đầu
		
		Phương pháp lặp Gauss-Seidel
		
		Kiểm tra điều kiện dừng
		
		In kết quả

3 Lập trình tính toán

Bài tập 3. Viết function cho bài toán giải hệ 3 phương trình 3 ẩn

a) Bằng phương pháp lặp.

b) Bằng phương pháp Seidel.

c) Dùng các function mới viết để giải lại bài tập 1.

c) Dùng các function mới viết để giải lại bài tập 2 bằng.

Bài tập 4. Viết function cho bài toán giải hệ 5 phương trình 5 ẩn

a) Bằng phương pháp lặp.

b) Bằng phương pháp Seidel.

c) Dùng các function mới viết để giải bài toán sau

$$\begin{cases} 6x_1 + x_2 + x_3 + x_4 + x_5 = 9 \\ 2x_1 + 9x_2 + 3x_3 + x_4 + 2x_5 = 1 \\ 2x_1 + x_2 + 10x_3 + 4x_4 + 2x_5 = -12 \\ x_1 + 2x_2 + x_3 + 8x_4 + 3x_5 = -12 \\ 2x_1 + x_2 + 2x_3 + 3x_4 + 9x_5 = 5 \end{cases}$$

Bài tập 5. Viết function cho bài toán giải hệ n phương trình n

a) Bằng phương pháp lặp.

b) Bằng phương pháp Seidel.

BÀI THỰC HÀNH SỐ 5

XẤP XỈ HÀM SỐ BẰNG ĐA THỨC NỘI SUY

1. Tóm tắt lý thuyết

Khi khảo sát đối tượng y phụ thuộc biến x, người ta không tìm được công thức biểu diễn của hàm y(x) mà chỉ tìm được vài giá trị quan hệ giữa y và x thường được cho bởi bảng sau

x_0	x_1	x_2	...	x_i	...	x_n
y_0	y_1	y_2	⋯	y_i	⋯	y_n

Tiếp đến, người ta cần tìm giá trị $\overline{y_c}$ tương ứng với tọa độ x_c không có trong bảng trên. Để thực hiện điều này, ta xây dựng một đa thức $P_n(x)$ đi qua tất cả các điểm trong bảng dữ liệu.

$$P_n(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$$

Sau đó giá trị giá trị $\overline{y_0}$ được tính bởi $P_n(x_0)$.

Dưới đây trình bày ba phương pháp dùng đa thức để xấp xỉ bảng giá trị (x_i, y_i) với $1 \le i \le n + 1$. Tuy thực hiện theo các thuật toán khác nhau nhưng chúng cùng cho ra một kết quả. Vì chỉ có duy nhất một đa thức cấp n có đồ thị đi qua n = 1 điểm cho trước.

1.1 Xấp xỉ bằng đa thức tổng quát

Ta lần lượt thay n + 1 điểm vào đa thức $P_n(x)$ và dẫn đến hệ phương trình n + 1 phương trình

$$\begin{cases} a_0 + a_1x_1 + a_2x_1^2 + ... + a_nx_1^n = y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + ... + a_nx_1^n = y_1 \\ \quad\quad ... \\ a_0 + a_1x_1 + a_2x_1^2 + ... + a_nx_1^n = y_n \end{cases}$$

Phương trình trên được viết lại dưới dạng ma trận X · A = Y trong đó X, Y và A được biểu diễn như sau

$$X = \begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ ... & ... & ... & ... & ... \\ 1 & x_n & x_n^2 & ... & x_n^n \end{pmatrix} \quad Y = \begin{pmatrix} y_0 \\ y_1 \\ ... \\ y_n \end{pmatrix} \quad A = \begin{pmatrix} a_0 \\ a_1 \\ ... \\ a_n \end{pmatrix}$$

Ta cần giải hệ trên để tìm n + 1 ẩn $a_0, a_1, ..., a_n$. Khi đó đa thức $P_n(x)$ hoàn toàn xác định
Thuật toán xấp xỉ bằng đa thức tổng quát được trình bày như sau:

Thuật toán phương pháp đa thức tổng quát

Bước 1: Khai báo Bảng dữ liệu (xx; yy) và tọa độ x_c.

Bước 2: Xây dựng ma trận X và Y có dạng trên.

Bước 3: Giải phương trình X · A = Y tìm A.

Bước 4: Xây dựng đa thức nội suy có dạng trên với $A = [a_0, a_1, a_2, ..., a_n]$ tìm được ở bước 2.

Bước 5: Xấp xỉ giá trị y_c bởi $\overline{y_c} = P_n(x_c)$

Nhận xét: Bài toán giải hệ phương trình ở bước 3 rất mất thời gian và công sức nếu n lớn.

1.2 Xấp xỉ bằng đa thức Lagrange

Ta xây dựng các đa thức $L_{i,n}(x)$ có tính chất $L_{i,n}(x_i) = 1$ và $L_{i,n}(x_j) = 0$. Đa thức này có dạng

$$L_{i,n}(x) = \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}, \qquad 0 \le i \le n$$

Khi đó đa thức $L_{i,n}(x) = \sum_{i=0}^{n} L_{i,n}(x) y_i$ là đa thức bậc n đi qua tất cả $(x_i; y_i)$ với 0 ≤ i ≤ n.

Thuật toán phương pháp đa thức Lagrange

Bước 1: Khai báo Bảng dữ liệu (xx; yy) và tọa độ x_0.

Bước 2: Xây dựng các đa thức Lagrange $L_{i,n}(x)$.

Bước 3: Xây dựng đa thức nội suy $L_{i,n}(x) = \sum_{i=1}^{n} L_{i,n}(x) y_i$

Bước 4: Xấp xỉ giá trị yc bởi $\overline{y_c} = L_n(x_c)$

Nhận xét: Bài toán xây dựng đa thức Lagrange ở bước 2 khá cồng kềnh trong việc biểu diễn về đa thức. Nhưng nhìn chung thì dễ thực hiện hơn phương pháp đa thức tổng quát.

1.3 Xấp xỉ bằng đa thức Newton

Ta xây dựng các tỷ sai phân $n[x_i x_{i+1}, ..., x_{i+j}]$ có dạng như sau

$$n[x_i] = y_i$$

$$n[x_i, x_{i+1}] = \frac{n[x_{i+1}] - n[x_i]}{x_{i+1} - x_i}$$

$$n[x_i, x_{i+1}, ..., x_{i+j}] = \frac{n[x_{i+1}, ..., x_{i+j}] - n[x_i, ..., x_{i+j-1}]}{x_{i+j} - x_i}$$

Khi đó đa thức xấp xỉ theo phương pháp Newton tiến có dạng

$$N_n(x) = \sum_{i=0}^{n} n[x_0, x_2, ..., x_i](x - x_0)(x - x_1)(x - x_2)...(x - x_{i-1})$$

và đa thức xấp xỉ theo phương pháp Newton tiến có dạng

$$N_n(x) = \sum_{j=n}^{0} n[x_j, x_{j+1}, ..., x_n](x - x_j)(x - x_{j+1})...(x - x_n)$$

Thuật toán phương pháp đa thức Lagrange

Bước 1: Khai báo Bảng dữ liệu (xx; yy) và tọa độ x_c.

Bước 2: Xây dựng các tỉ sai phân $n\left[x_i, x_{i+1}, ..., x_{i+j}\right]$.

Bước 3: Xây dựng đa thức nội suy $N_n(x)$.

Bước 4: Xấp xỉ giá trị y_0 bởi $y_0 = N_n(x_0)$

Nhận xét: Thuật toán xấp xỉ bằng phương pháp Newton có độ phức tạp ngang bằng với phương pháp Lagrange và giải quyết bài toán tốt hơn phương pháp đa thức tổng quát.

2. Thực hành trên máy tính

2.1 Nội suy hàm số bằng đa thức tổng quát

Bài tập 1. Tìm giá trị tại x = 2.5 của hàm y = f(x) được cho bởi bảng

x	1	2.2	3.1	4	2.5
y	1.678	3.267	2.198	3.787	

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	x = np.array([1, 2.2, 3.1, 4]) y = np.array([1.678, 3.267, 2.198, 3.787]) coefficients = np.polyfit(x, y, 4) poly = np.poly1d(coefficients) x_interp = 2.5 y_interp = poly(x_interp) print(f"Giá trị tại x = {x_interp} là: {y_interp}")	Các điểm đã cho Tìm đa thức nội suy bậc 4 Tính giá trị tại x = 2.5 bằng đa thức nội suy

Bài tập 2. Tìm giá trị tại x = 5 và x = 10 của hàm y = f(x) được cho bởi bảng

x	2	4.5	5.7	7.2	9.3	5	10
y	3.218	1.642	2.398	2.145	3.135		

Bài tập 3. Tìm giá trị tại x = −2 và x = 0 của hàm y = f(x) được cho bởi bảng

x	-3.2	-2.5	-1.7	-0.8	0.3	1.5	-2	0
y	-8.982	-5.831	-4.261	-1.837	-3.298	-0.249		

2.2 Nội suy hàm số bằng đa thức Lagrange

Bài tập 4. Tìm giá trị tại x = 2.5 của hàm y = f(x) được cho bởi bảng

x	1	2.2	3.1	4	2.5
y	1.678	3.267	2.198	3.787	

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	x = np.array([1, 2.2, 3.1, 4]) y = np.array([1.678, 3.267, 2.198, 3.787]) poly = np.polynomial.polynomial.Polynomial.fit(x, y, deg=len(x)-1) x_interp = 2.5 y_interp = poly(x_interp) print(f"Giá trị tại x = {x_interp} là: {y_interp}")	Các điểm đã cho Tạo đa thức Lagrange từ các điểm đã cho Tính giá trị tại x = 2.5 bằng đa thức Lagrange

Bài tập 5. Tìm giá trị tại x = 5 và x = 10 của hàm y = f(x) được cho bởi bảng

x	2	4.5	5.7	7.2	9.3	5	10
y	3.218	1.642	2.398	2.145	3.135		

Bài tập 6. Tìm giá trị tại x = −2 và x = 0 của hàm y = f(x) được cho bởi bảng

x	-3.2	-2.5	-1.7	-0.8	0.3	1.5	-2	0
y	-8.982	-5.831	-4.261	-1.837	-3.298	-0.249		

2.3 Nội suy hàm số bằng đa thức Newton

Bài tập 7. Tìm giá trị tại x = 2.5 của hàm y = f(x) được cho bởi bảng

x	1	2.2	3.1	4	2.5
y	1.678	3.267	2.198	3.787	

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	x = np.array([1, 2.2, 3.1, 4]) y = np.array([1.678, 3.267, 2.198, 3.787])	Các điểm đã cho

	coefficients = np.polynomial.polynomial.polyfit(x, y, len(x)-1) poly = np.polynomial.Polynomial(coefficients) x_interp = 2.5 y_interp = poly(x_interp) print(f"Giá trị tại x = {x_interp} là: {y_interp}")	Tạo đa thức Newton từ các điểm đã cho Tính giá trị tại x = 2.5 bằng đa thức Newton

Bài tập 8. Tìm giá trị tại x = 5 và x = 10 của hàm y = f(x) được cho bởi bảng

x	2	4.5	5.7	7.2	9.3	5	10
y	3.218	1.642	2.398	2.145	3.135		

Bài tập 9. Tìm giá trị tại x = −2 và x = 0 của hàm y = f(x) được cho bởi bảng

x	-3.2	-2.5	-1.7	-0.8	0.3	1.5	-2	0
y	-8.982	-5.831	-4.261	-1.837	-3.298	-0.249		

3. Lập trình tính toán

Bài tập 10. Viết function xây dựng hàm đa thức xấp xỉ và nội suy dùng đa thức tổng quát cho bảng 4 dữ liệu

a) Thực hiện với các yêu cầu sau:

(i) Input là giá trị xx, yy và tọa độ x_c

(ii) Vẽ đồ thị hàm xấp xỉ và bảng dữ liệu

b) Sử dụng function trên để giải bài tập 1.

Bài tập 11. Viết function xây dựng hàm đa thức xấp xỉ và nội suy dùng đa thức Lagrange cho bảng 4 dữ liệu

a) Thực hiện với các yêu cầu sau:

(i) Input là giá trị xx, yy và tọa độ x_c

(ii) Vẽ đồ thị hàm xấp xỉ và bảng dữ liệu

b) Sử dụng function trên để giải bài tập 4

Bài tập 12. Viết function xây dựng hàm đa thức xấp xỉ và nội suy dùng đa thức Newton cho bảng 4 dữ liệu

a) Thực hiện với các yêu cầu sau:

(i) Input là giá trị xx, yy và tọa độ x_c

(ii) Vẽ đồ thị hàm xấp xỉ và bảng dữ liệu

b) Sử dụng function trên để giải bài tập 7

Bài tập 13. Viết function xây dựng hàm đa thức xấp xỉ và nội suy dùng đa thức tổng quát

a) Cho bảng 6 dữ liệu và giải lại bài tập 3.

b) Cho bảng n dữ liệu.

Bài tập 14. Viết function xây dựng hàm đa thức xấp xỉ và nội suy dùng đa thức Lagrange

a) Cho bảng 6 dữ liệu và giải lại bài tập 6.

b) Cho bảng n dữ liệu.

Bài tập 15. Viết function xây dựng hàm đa thức xấp xỉ và nội suy dùng đa thức Newton

a) Cho bảng 6 dữ liệu và giải lại bài tập 9.

b) Cho bảng n dữ liệu.

BÀI THỰC HÀNH SỐ 6

XẤP XỈ HÀM SPLINE và BÌNH PHƯƠNG NHỎ NHẤT

1 Tóm tắt lý thuyết

1.1 Xấp xỉ bằng phương pháp đường cong Spline

Với một bộ dữ liệu (x_i, y_i), ta xây dựng một đường cong trơn từng khúc sao cho nó đi qua hết tất cả các điểm. Một trong những đường cong phù hợp và được sử dụng nhất là đường cong spline tự nhiên bậc ba có dạng

$$S(x) = \begin{cases} s_1(x) & x \in [x_0, x_1] \\ s_2(x) & x \in [x_1, x_2] \\ & \dots \\ s_n(x) & x \in [x_{n-1}, x_n] \end{cases}$$

Cơ sở để xây dựng đường cong spline S(x) là sự liên tục và khả vi tại các điểm nối tiếp (cũng là những điểm trong bộ dữ liệu). Từ đó ta có

$$s_i(x_i) = s_{i+i}(x_i) = S(x_i)$$
$$s'_i(x_i) = s'_{i+i}(x_i) = S'(x_i)$$
$$s''_i(x_i) = s''_{i+i}(x_i) = S''(x_i)$$

Đặt $m_i = S''(x_i)$, tích phân hai lần phương trình cuối và áp dụng hai phương trình trên , ta thu được hệ phương trính sau

$$\begin{cases} m_i \frac{h_i}{6} + m_{i+1} \frac{h_i + h_{i+1}}{3} + m_{i+2} \frac{h_{i+1}}{6} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \\ m_0 = m_n = 0 \end{cases}$$

với $h_i = x_{i+1} - x_i$.

Khi đó ta thu được dạng tường minh của $s_i(x)$

$$s_i(x) = m_{i+1} \frac{(x - x_i)^3}{6h_i} + m_i \frac{(x_{i+1} - x)^3}{6h_i} + M_i \frac{(x_{i+1} - x)^3}{h_i} + N_i \frac{x - x_i}{h_i}$$

Và

$$\begin{cases} M_i = y_i - m_i \frac{h_i^2}{6} \\ N_i = y_{i+1} - m_{i+1} \frac{h_i^2}{6} \end{cases}$$

Thuật toán phương pháp đường cong Spline tự nhiên bậc 3

Bước 1: Khai báo Bảng dữ liệu (xx, yy) và tọa độ x_c.

Bước 2: Tìm các giá trị m_i

Bước 3: Tìm các giá trị M_i, N_i.

Bước 4: Xây dựng đa thức nội suy S(x) với các $s_i(x)$

Bước 5: Xấp xỉ giá trị y_c bởi $\overline{y_c} = S(x_c)$

Nhận xét: Đường cong spline gồm nhiều đoạn cong nhỏ với bậc đa thức của chúng là bậc 3. Tại các điểm nối, hàm số, đạo hàm cấp 1 và cấp 2 đều tồn tại. Trong sự so sánh với hàm đa thức cấp n, việc xây dựng hàm spline tuy phức tạp về thuận toán nhưng sử dụng ít tính toán hơn. Ngoài ra, đường cong spline có vẻ gần với các dữ liệu hơn. Ngược lại hàm đa thức có thuật toán đơn giản nhưng sử dụng nhiều tính toán. Tuy nhiên hàm đa thức có mặt mạnh là nó khả vi đến cấp n trong khi spline chỉ khả vi đến cấp 3.

1.2 Xấp xỉ bằng phương pháp bình phương nhỏ nhất

Với một bộ dữ liệu (x_i, y_i), ta xây dựng một đường cong sao cho khoảng cách của nó đến các dữ liệu là nhỏ nhất. Những đường cong này có dạng được dự đoán trước dựa vào tính chất của bộ dữ liệu.

Theo phương pháp bình phương nhỏ nhất, người ta đặt hàm số xấp xỉ là R(x) và định nghĩa khoảng cách từ bộ dữ liệu đến đường cong bởi

$$T = \sum_{i=1}^{n} (\overline{R}(x_i) - y_i)^2$$

Để tổng khoảng cách này nhỏ nhất, ta đưa về giải hệ phương trình

$$\frac{\partial T}{\partial a_i} = 0$$

Hai dạng thường gặp nhất của R(x) là $R_1(x) = ax + b$ và $R_2(x) = ae^{bx}$.

a) Hàm xấp xỉ $\overline{R}_1(x) = ax + b$ (x) = ax + b là hàm đơn giản nhất và thường được sử dụng nhất.

Thuật toán phương pháp bình phương nhỏ nhất dùng đường thẳng

Bước 1: Khai báo Bảng dữ liệu (xx, yy) và tọa độ x_c.

Bước 2: Tính các giá trị $X = \sum_{i=1}^{n} x_i, Y = \sum_{i=1}^{n} y_i, XX = \sum_{i=1}^{n} x_i^2, XY = \sum_{i=1}^{n} x_i y_i$

Bước 3: Giải hệ phương trình sau để tìm a và b

$$\begin{cases} XXa + Xb = XY \\ Xa + nb = Y \end{cases}$$

Bước 4: Xây dựng hàm xấp xỉ $R_1(x) = ax + b$.

Bước 5: Xấp xỉ giá trị y_c bởi $\overline{y}_c = R_1(x_c)$

b) Hàm xấp xỉ $R_2(x) = ae^{bx}$ là một hàm khá phổ biến, lí do là nó mô phỏng tốt các bộ dữ liệu có biến thiên lớn. Lưu ý rằng trong trường hợp tổng quát các giá trị y_i phải dương.

Thuật toán phương pháp bình phương nhỏ nhất dùng đường cong

Bước 1: Khai báo Bảng dữ liệu (xx, yy) và tọa độ x_c.

Bước 2: Tính các giá trị $X = \sum_{i=1}^{n} x_i, lY = \sum_{i=1}^{n} \ln y_i, XX = \sum_{i=1}^{n} x_i^2, XlY = \sum_{i=1}^{n} x_i \ln y_i$

.

Bước 3: Giải hệ phương trình sau để tìm A và B

$$\begin{cases} XXA + XB = XlY \\ XA + nB = lY \end{cases}$$

Gán $a = e^B, b = A$

Bước 4: Xây dựng hàm xấp xỉ $R_2(x) = ae^{bx}$

Bước 5: Xấp xỉ giá trị y_c bởi $\overline{y_c} = R_2(x_c)$

Ngoài ra còn có một số xấp xỉ bình phương nhỏ nhất phi tuyến như dạng lượng giác, dạng parabol,... tùy vào mô hình bài toán

2 Thực hành trên máy tính

2.1 Nội suy hàm số bằng hàm spline

Bài tập 1. Tìm giá trị tại x = 3, x = 7.5, của hàm y = f(x) được bởi bảng sau bằng spline bậc ba tự nhiên

x	2	4	7	8	3	7.5
y	2.2	1.8	2.7	3.1		

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	x = np.array([2, 4, 7, 8])
	y = np.array([2.2, 1.8, 2.7, 3.1])
	
	n = len(x) - 1
	
	h = np.diff(x)
	
	A = np.zeros((n - 1, n - 1))
	b = np.zeros(n - 1)
	
	for i in range(1, n):
	  A[i - 1, i - 1] = 2 * (h[i - 1] + h[i]) # Phần tử đường chéo
	  if i - 1 > 0:
	  A[i - 1, i - 2] = h[i - 1] # Phần tử dưới đường chéo
	  if i < n - 1:
	  A[i - 1, i] = h[i] # Phần tử trên đường chéo
	
	  b[i - 1] = 3 * ((y[i + 1] - y[i]) / h[i] - (y[i] - y[i - 1]) / h[i - 1])
		Dữ liệu từ bảng
		
		Số đoạn spline
		
		Tính khoảng h giữa các điểm
		Tính hệ số của hệ phương trình
		
		Vế phải của phương trình

	c = np.zeros(n + 1)
	c[1:n] = np.linalg.solve(A, b)
	
	b = np.zeros(n)
	d = np.zeros(n)
	
	for i in range(n):
	  b[i] = (y[i + 1] - y[i]) / h[i] - h[i] * (2 * c[i] + c[i + 1]) / 3
	  d[i] = (c[i + 1] - c[i]) / (3 * h[i])
	
	def spline_interpolate(x_query):
	  for i in range(n):
	  if x[i] <= x_query <= x[i + 1]:
	  dx = x_query - x[i]
	  return y[i] + b[i] * dx + c[i] * dx**2 + d[i] * dx**3
	  return None
	
	x_values = [3, 7.5]
	y_values = [spline_interpolate(xq) for xq in x_values]
	
	for xi, yi in zip(x_values, y_values):
	  print(f"f({xi}) = {yi:.4f}")
		Giải hệ phương trình tuyến tính để tìm c
		Tính hệ số b và d
		
		Hàm nội suy spline bậc ba
		
		Trả về None nếu x nằm ngoài khoảng nội suy
		Tính giá trị tại x = 3 và x = 7.5
		In kết quả

Bài tập 2. Tìm giá trị tại x = 4, x = 5.5 của hàm y = f(x) cho bởi bảng sau bằng spline bậc ba tự nhiên

x	2.2	3.6	4.9	5.2	5.7	6.4	4	5.5
y	1.4	3.2	5.1	4.4	3.9	3.2		

2.2 Nội suy hàm số bằng bình phương nhỏ nhất

Bài tập 3. Tìm giá trị tại x = 6 của hàm y = f(x) được cho bởi bảng sau bằng hàm xấp xỉ R(x) = ax + b

x	2	4	7	8.5	9.5	11	6
y	2.2	4.2	6.8	8.1	9.7	10.5	

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	x = np.array([2, 4, 7, 8.5, 9.5, 11]) y = np.array([2.2, 4.2, 6.8, 8.1, 9.7, 10.5]) n = len(x) X = np.vstack((x, np.ones(n))).T # Ma trận thiết kế [[x, 1]] a, b = np.linalg.lstsq(X, y, rcond=None)[0] # Giải phương trình X * [a, b] = y x_query = 6	Dữ liệu từ bảng Tính các hệ số a và b bằng công thức bình phương nhỏ nhất Dự đoán giá trị tại x = 6

	y_query = a * x_query + b print(f"Hàm hồi quy: R(x) = {a:.4f}x + {b:.4f}") print(f"Giá trị nội suy tại x = {x_query}: f({x_query}) = {y_query:.4f}")	In kết quả

Bài tập 4. Tìm giá trị tại x = 4.5 của hàm y = f(x) được cho bởi bảng sau bằng hàm xấp xỉ R(x) = ax + b

x	1	2.1	2.9	3.8	5	6.2	4.5
y	3.021	4.219	5.018	5.986	7.139	8.138	

Bài tập 5. Tìm giá trị tại x = 3.5 của hàm y = f(x) được cho bởi bảng sau bằng hàm xấp xỉ R(x) = ax + b

x	1	1.6	2.1	3	3.3	3.7	4.1	4.9	6.2	3.5
y	1.1	2.2	3.5	4.9	7.2	7.8	7.9	8.5	10	

Bài tập 6. Tìm giá trị tại x = 8.5 của hàm y = f(x) được cho bởi bảng sau bằng hàm xấp xỉ $R(x) = ae^{kx}$

x	1.1	3.2	5.1	7.7	9.6	12.2	8.5
y	3.1	29.9	65.7	100.4	195.7	300.4	

Bài tập 7. Tìm giá trị tại x = 7.7 của hàm y = f(x) được cho bởi bảng sau bằng hàm xấp xỉ $R(x) = ae^{kx}$

x	2	4	7	8.5	9.5	11	7.7
y	2.2	2.5	2.7	3.1	3.2	3.5	

Bài tập 8. Tìm giá trị tại x = 5 của hàm y = f(x) được cho bởi bảng sau bằng hàm xấp xỉ $R(x) = ae^{kx}$

x	3	3.4	4.1	4.3	4.7	5.3	6	6.4	5
y	1.23	1.4	1.69	1.79	2.13	2.52	2.45	2.97	3.44

3 Lập trình tính toán

Bài tập 9. Viết function xấp xỉ và nội suy hàm số bằng phương pháp bình phương nhỏ nhất dạng R(x) = ax + b.

a) Dùng function trên để giải lại bài toán 3.

b) Dùng function trên để giải lại bài toán 4.

c) Dùng function trên để giải lại bài toán 5.

Bài tập 10. Viết function xấp xỉ và nội suy hàm số bằng phương pháp bình phương nhỏ nhất dạng R(x) = ae^{bx}

.

a) Dùng function trên để giải lại bài toán 6.

b) Dùng function trên để giải lại bài toán 7.

c) Dùng function trên để giải lại bài toán 8.

Bài tập 11. Viết function xấp xỉ và nội suy hàm số bằng phương pháp bình phương nhỏ nhất.

a) hàm xấp xỉ có dạng dạng R(x) = $ax^2 + b$ và thử lại với bộ dữ liệu thỏa $y_i = 0.5x_i^2 + 1.5$

b) hàm xấp xỉ có dạng dạng R(x) = ax^b và thử lại với bộ dữ liệu thỏa $y_i = 2x_i^{1.3}$

Bài tập 12. Viết function xấp xỉ và nội suy hàm số bằng phương pháp bình phương nhỏ nhất.

a) hàm xấp xỉ có dạng dạng $R(x) = ax^2 + bx + c$ và thử lại với bộ dữ liệu thỏa
$y_i = 0.3x_i^2 + 0.7x_i - 2.5$

b) hàm xấp xỉ có dạng dạng $R(x) = ax + b\ sin\ x + c\ sin\ x$ và thử lại với bộ dữ liệu thỏa
$y_i = 3x_i + 1.5\sin x_i - 3.5\cos x$

Bài tập 13. Viết function xấp xỉ và nội suy hàm số bằng hàm spline bậc ba tự nhiên.

a) với bộ dữ liệu 4 số và giải lại bài toán 1.

b) với bộ dữ liệu 6 số và giải lại bài toán 2.

c) với bộ dữ liệu n số và thử lại với bộ dữ liệu thỏa $y_i = 3^{x_i}$

BÀI THỰC HÀNH SỐ 7

ĐẠO HÀM SỐ và TÍCH PHÂN SỐ

1. Tóm tắt lý thuyết

1.1 Tính đạo hàm số

Đạo hàm của hàm số được tính bằng giới hạn của tỉ số giữa số gia của hàm số và số gia của biến số. Công thức đạo hàm số được xây dựng từ ý tưởng bỏ qua giới hạn và chỉ giữ lại phần tỉ số giữa các số gia. Để tăng thêm độ chính xác cho phép tính, người ta thường thay đổi phần số gia của hàm số. Kết quả là có nhiều công thức tính đạo hàm số, nhưng những công thức có sự cân bằng (số gia của hàm số xoay quanh giá trị đang xét) là cho kết quả tốt hơn (sai số ít hơn).

• Công thức sai phân là công thức đơn giản nhất nhưng cho sai số khá lớn:

a) Công thức sai phân tiến

$$\overline{f'_{SPT}}(x_0) \simeq \frac{f(x_0+h)-f(x_0)}{h}$$

b) Công thức sai phân lùi

$$\overline{f'_{SPL}}(x_0) \simeq \frac{f(x_0)-f(x_0-h)}{h}$$

• Công thức ba điểm có biểu thức phức tạp hơn và cho kết quả khá tốt. Đặc biệt công thức ba điểm giữa là công thức thường được sử dụng nhất.

a) Công thức ba điểm cuối

$$\overline{f'_{3DC}}(x_0) = \frac{1}{2h}\left(-3f(x_0) + 4f(x_0+h) - f(x_0+2h)\right)$$

b) Công thức ba điểm giữa:

$$\overline{f'_{3DC}}(x_0) = \frac{1}{2h}\left(-f(x_0-h) + f(x_0+h)\right)$$

• Công thức năm điểm cho kết quả rất tốt nhưng rất cồng kềnh. Nếu không vì một lý do đặc biệt, rất ít khi người ta sử dụng nó.

a) Công thức năm điểm cuối:

$$\overline{f'_{5DC}}(x_0) = \frac{1}{12h}\left(-25f(x_0) + 48f(x_0+h) - 36f(x_0+2h) + 16f(x_0+3h) - 3f(x_0+4h)\right)$$

b) Công thức năm điểm giữa:

$$\overline{f'_{5DG}}(x_0) = \frac{1}{12h}\left(f(x_0-2h) - 8f(x_0-h) + 8f(x_0-h) - f(x_0+2h)\right)$$

Việc tính đạo hàm số được thực hiện như sau

Thuật toán tính đạo hàm số
Bước 1: Khai báo Bảng dữ liệu (xx; yy).
Bước 2: Tính đạo hàm số theo công thức.

1.2 Tích phân số

Với các hàm số được cho dưới dạng bảng dữ liệu, ta không thể áp dụng cách tính tích phân theo dạng giải tích toán học được. Vì vậy, người ta sử dụng tích phân số để tìm giá trị xấp xỉ của đối tượng cần khảo sát.

• Công thức Newton - Cotes: Xuất phát từ việc sử dụng đa thức nội suy Lagrange và thực hiện việc đổi biến $x = x0 + th$. Với t là biến mới và h là độ dài khoảng chia trong phân hoạch $[x_0, x_n]$, công thức tính tích phân được viết như sau

$$\int_{x_d}^{x_c} f(x)dx \simeq (x_c - x_d)\sum_{k=0}^{n} H_{k,n} y_k$$

trong đó

$$H_{k,n} = \frac{(-1)^{n-k} C_n^k}{n.n!}\int_0^n \frac{t(t-1)...(t-n)}{t-k}dt$$

Từ đây ta có các công thức tính tích phân đơn giản hơn như sau

a) Công thức tích phân hình thang:

$$\int_{x_0}^{x_n} f(x)dx = \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} f(x)dx \simeq \sum_{i=1}^{n}\frac{x_i - x_{i-1}}{2}(y_{i-1} + y_i)$$

trong đó $[x_{i-1},\ x_i]$ là một ô tích phân được áp dụng với n = 1.

b) Công thức tích phân Simpson $1/3$:

$$\int_{x_0}^{x_n} f(x)dx = \sum_{j=1}^{n/2}\int_{x_{j-1}}^{x_j} f(x)dx \simeq \sum_{j=1}^{n/2}\frac{x_j - x_{j-1}}{6}(y_{j-1} + 4y_{j-1/2} + y_j)$$

trong đó $[x_{j-1}, x_j]$ là một ô tích phân gồm hai khoảng chia bằng nhau.

c) Công thức tích phân Simpson $3/8$:

$$\int_{x_0}^{x_n} f(x)dx = \sum_{j=1}^{n/3}\int_{x_{k-1}}^{x_k} f(x)dx \simeq \sum_{k=1}^{n/3}\frac{x_k - x_{k-1}}{8}(y_{k-1} + 3y_{k-2/3} + 3y_{k-1/3} + y_k)$$

trong đó $[x_{k-1}, x_k]$ là một ô tích phân gồm ba khoảng chia bằng nhau.

Việc tính tích phân theo phương pháp Newton - Cotes đươc tổng quát hóa như sau

Thuật toán tính tích phân Newton - Cotes
Bước 1: Khai báo Bảng dữ liệu (xx; yy).
Bước 2: Tính tích phân theo công thức .

• Tích phân Gauss: Cũng như tích phân Newton - Cotes, tích phân Gauss được tính dưới tổ hợp đại số của giá trị hàm tại một số điểm và trọng số tương ứng

$$\int_{-1}^{1} f(x)dx \simeq \sum_{k=1}^{n} w_k . f(x_k)$$

trong đó wk là các trọng số tương ứng với các vị trí x_k. Việc tìm các trọng số w_k dựa vào đa thức Legendre giúp độ chính xác của phương pháp Gauss rất cao. Tuy nhiên khi sử dụng đa thức Legendre cũng có điểm yếu là chỉ lấy tích phân trên khoảng [−1, 1] và các điểm x_k phải được xác định cụ thể. Sau này, người ta cũng tìm cách cải tiến phương pháp này nhưng cũng không khắc phục được hoàn toàn những điểm yếu này.

Dựa vào đa thức Legendre, người ta tính được tọa độ và trọng số tương ứng của tích phân Gauss như sau

n	Tọa độ	Trọng số
1	0	2
2	-0.577350269189626 0.577350269189626	
3	-0.774596669241483 0 0 0.774596669241483	0.555555555555556 0.888888888888889 0.555555555555556
4	-0.861136311594053 -0.339981043584856 0.339981043584856 0.861136311594053	0.347854845137454 0.652145154862546 0.652145154862546 0.347854845137454
5	-0.906179845938664 -0.538469310105683 0 0.538469310105683 0.906179845938664	0.236926885056189 0.478628670499367 0.568888888888888 0.478628670499367 0.236926885056189

Trong trường hợp cận tích phân là khoảng [a, b], ta có thể sử dụng công thức đổi biến $t = \frac{b-a}{2}x + \frac{b+a}{2}$. Khi đó tích phân Gauss được viết lại như sau

$$\int_{a}^{b} f(t)dt = \int_{-1}^{1} f(t(x))\frac{b-a}{2}dx \simeq \frac{b-a}{2}\sum_{k=1}^{n} w_k . f(t(x_k))$$

Công thức cho phép tính tích phân với cận bất kì, tuy nhiên các tọa độ lấy tích phân đều phải xác định trước. Vì vậy phương pháp Gauss thường được dùng để tính nhanh tích phân của hàm sô (dạng công thức) chứ không phải dùng để tính tích phân hàm dạng bảng. Hoặc chỉ bảng đặc biệt có tọa độ Gauss mới được sử dụng.

Việc tính tích phân theo phương pháp Gauss đươc tổng quát hóa như sau

2. Thực hành trên máy tính

2.1 Tính toán đạo hàm số

Bài tập 1. Cho hàm số cho bởi bảng sau. Tìm đạo hàm của nó tại x = 1 bằng các công thức sai phân và ba điểm. Sau đó tính sai số tương đối của từng kết quả biết giá trị đạo hàm chính xác là $f'(1) = 0.5403$.

x	0.8	0.9	1	1.1	1.2	1.3	1.4
y	0.7174	0.7833	0.8415	0.8912	0.9320	0.9636	0.9854

f'_{spt}	$\delta f'_{spt}$	f'_{spl}	$\delta f'_{spl}$	f'_{3dc}	$\delta f'_{3dc}$	f'_{3dg}	$\delta f'_{3dg}$

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	xx = np.array([0.9, 1, 1.1, 1.2]) yy = np.array([0.7833, 0.8415, 0.8912, 0.9320]) df = 0.5403 df_SPT = (yy[2] - yy[1]) / (xx[2] - xx[1]) rEdf_SPT = np.abs((df - df_SPT) / df) df_SPL = (yy[1] - yy[0]) / (xx[1] - xx[0]) rEdf_SPL = np.abs((df - df_SPL) / df) df_3DC = (-3 * yy[1] + 4 * yy[2] - yy[3]) / (xx[3] - xx[0]) rEdf_3DC = np.abs((df - df_3DC) / df) df_3DG = (-yy[0] + yy[2]) / (xx[2] - xx[0]) rEdf_3DG = np.abs((df - df_3DG) / df)	Nhập vector x và y

Bài tập 2. Thực hiện lại bài toán 1 bằng công thức năm điểm.

f'_{5dc}	$\delta f'_{5dc}$	f'_{5dg}	$\delta f'_{5dg}$

2.2 Tính toán tích phân số

Bài tập 3. Tìm tính phân của hàm số cho bởi bảng sau bằng 3 công thức Newton - Cotes. Sau đó tính sai số tương đối của từng kết quả biết tích phân chính xác là I = 42

x	1	2	3	4	5	6	7
y	4	-6	-14	-14	0	34	94

I_1	δI_1	I_2	δI_2	I_3	δI_3

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THICH

1	import numpy as np	Khai báo thư viện
2	x = np.array([1, 2, 3, 4, 5, 6, 7]) y = np.array([4, -6, -14, -14, 0, 34, 94]) I_exact = 42 I1 = (x[2] - x[0]) * (y[0] + 4*y[1] + y[2]) / 6 I2 = (x[3] - x[0]) * (y[0] + 3*y[1] + 3*y[2] + y[3]) / 8 I3 = (x[4] - x[0]) * (7*y[0] + 32*y[1] + 12*y[2] + 32*y[3] + 7*y[4]) / 90 delta_I1 = np.abs((I_exact - I1) / I_exact) delta_I2 = np.abs((I_exact - I2) / I_exact) delta_I3 = np.abs((I_exact - I3) / I_exact) print("I1:", I1) print("Relative error for I1:", delta_I1) print("I2:", I2) print("Relative error for I2:", delta_I2) print("I3:", I3) print("Relative error for I3:", delta_I3)	Thêm dữ liệu Giá trị tích phân chính xác Tính tích phân bằng công thức Newton-Cotes 3 điểm Tính toán sai số tương đối Hiển thị ra kết quả

Bài tập 4. Tìm tính phân của hàm số cho bởi bảng sau bằng 3 công thức Newton - Cotes. Sau đó tính sai số tương đối của từng kết quả biết tích phân chính xác là I = 7.2

x	-2	-1	0	1	2	3	4
y	24	1	4	3	-8	-11	36

I_1	δI_1	I_2	δI_2	I_3	δI_3

Bài tập 5. Tìm tính phân của hàm số $f(x) = e^x$ trên [2, 4] bằng phương pháp Gauss. Sau đó tính sai số tương đối của từng kết quả so với tích phân chính xác.

I_1	δI_1	I_2	δI_2	I_3	δI_3

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	from scipy.integrate import quad def f(x): return np.exp(x) I_exact, _ = quad(f, 2, 4) I1, _ = quad(f, 2, 4, points=[3]) I2, _ = quad(f, 2, 4, points=[2, 4/3*2**(1/2)-2/3, 4/3*2**(1/2)+2/3])	Tính tích phân chính xác để so sánh

	I3, _ = quad(f, 2, 4, points=[2, 4/3*2**(1/2)-2/3, 4/3*2**(1/2)+2/3, 4]) delta_I1 = np.abs((I_exact - I1) / I_exact) delta_I2 = np.abs((I_exact - I2) / I_exact) delta_I3 = np.abs((I_exact - I3) / I_exact) print("| I1\t\t| δI1\t\t| I2\t\t| δI2\t\t| I3\t\t| δI3\t\t|") print("| {:.6f}\t| {:.6f}\t| {:.6f}\t| {:.6f}\t| {:.6f}\t| {:.6f}\t|".format(I1, delta_I1, I2, delta_I2, I3, delta_I3))	Sử dụng phương pháp Gauss để tính tích phân Tính sai số tương đối Hiển thị kết quả trên bảng

Bài tập 6. Tìm tính phân của hàm số f(x) = sin x trên [0; π] bằng phương pháp Gauss. Sau đó tính sai số tương đối của từng kết quả so với tích phân chính xác.

I_1	δI_1	I_2	δI_2	I_3	δI_3

3. Lập trình tính toán

Bài tập 7. Thực hiện các yêu cầu sau

a) Viết function tính đạo hàm bằng phương pháp sai phân tiến và sai phân lùi.

b) Áp dụng function vùa viết để giải lại Bài tập 1.

Bài tập 8. Thực hiện các yêu cầu sau

a) Viết function tính đạo hàm bằng phương pháp ba điểm.

b) Áp dụng function vùa viết để giải lại Bài tập 1.

Bài tập 9. Thực hiện các yêu cầu sau

a) Viết function tính tích phân bằng phương pháp hình thang.

b) Áp dụng function vùa viết để giải lại Bài tập 3 và Bài tập 4.

Bài tập 10. Thực hiện các yêu cầu sau

a) Viết function tính tích phân bằng phương pháp simpson 1=3.

b) Áp dụng function vùa viết để giải lại Bài tập 3 và Bài tập 4.

Bài tập 11. Thực hiện các yêu cầu sau

a) Viết function tính tích phân bằng phương pháp simpson 3=8.

b) Áp dụng function vùa viết để giải lại Bài tập 3 và Bài tập 4.

Bài tập 12. Thực hiện các yêu cầu sau

a) Viết function tính các hệ số Hk;n trong (7.8) với n.

b) Viết function tính tích phân bằng phương pháp Newton-Cotes tổn quát.

c) Áp dụng function vừa viết để giải lại bài 3 với n = 4.

Bài tập 13. Thực hiện các yêu cầu sau

a) Viết function tính tích phân bằng phương pháp Gauss 2 điểm nút.

b) Áp dụng function vừa viết để giải lại Bài tập 5 và Bài tập 6.

Bài tập 14. Thực hiện các yêu cầu sau

a) Viết function tính tích phân bằng phương pháp Gauss 3 điểm nút.

b) Áp dụng function vừa viết để giải lại Bài tập 5 và Bài tập 6.

Bài tập 15. Thực hiện các yêu cầu sau

a) Viết function tính tích phân bằng phương pháp Gauss 4 điểm nút.

b) Áp dụng function vừa viết để giải lại Bài tập 5 và Bài tập 6.

BÀI THỰC HÀNH SỐ 8

GIẢI PHƯƠNG TRÌNH VI PHÂN

1 Tóm tắt lý thuyết

Trong khoa học, kĩ thuật chúng ta gặp rất nhiều bài toán liên quan đến phương trình vi phân thường (chẳng hạn như bài toán tính vận tốc của một vật thể khi biết độ dài quãng đường trong những khoảng thời gian khác nhau bài toán tính toán cường độ dòng điện theo điện lượng). Có nhiều trường hợp nghiệm đúng của phương trình vi phân không thể tìm ra được.

Các phương pháp gần đúng có thể chia làm hai nhóm: Nhóm thứ nhất được gọi là phương pháp giải tích, nhóm thứ hai được gọi là phương pháp số. Các phương pháp giải tích cho phép tìm nghiệm gần đúng dưới dạng một biểu thức giải tích. Các phương pháp số cho phép tìm nghiệm dưới dạng bảng. Dưới đây, ta chỉ giới thiệu một phương pháp giải tích thường dùng gọi là phương pháp lặp đơn, và một số phương pháp số (bao gồm phương pháp Euler, Euler cải tiến, Rung - Kutta).

Bài toán phương trình vi phân có dạng

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

với $x \in D_x = [x_0 - a,\ x_0 + a], y \in D_y = [y_0 - \mathrm{b},\ y_0 + \mathrm{b}]$ Trong nội dung chương trình, ta sẽ giải bài toán này với nghiệm hàm bằng phương pháp lặp, và nghiệm số bằng phương pháp Euler hoặc Runge - Kutta.

1.1 Tìm nghiệm hàm bằng phương pháp lặp

Ta tìm cách xấp xỉ nghiệm của bài toán bằng việc lặp nghiệm với sai số giảm dần. Trước hết, ta tích phân phương trình thứ nhất kết hợp với điều kiện thứ hai và thu được

$$y(x) = y_0 + \int_{x_0}^{x} f(x, y(s))ds$$

Từ đây ta thiết lập dãy

$$\begin{cases} y_0(x) = y_0 \\ y_{k+1}(x) = y_0 + \int_{x_0}^{x} f(s, y_k(s))ds \end{cases}$$

Dãy $y_k(x)$ được chứng minh là hội tụ về y(x) là nghiệm của bài toán.

Tìm nghiệm hàm bằng phương pháp lặp
Bước 1: Khai báo hàm số f(x, y) và điều kiệu đầu y_0.
(Mở vòng lặp - bắt đầu với k = 1)
Bước 2: Gán $y_{k+1} = y_0 + \int_{x_0}^{x} f(s, y_k(s))ds$
Nếu $\delta y_k < \delta y$ phá vòng lặp.
Bước 3: Gán k = k + 1
(Đóng vòng lặp)

Kết luận $\overline{y} = y_k$

1.2 Tìm nghiệm số bằng phương pháp Euler

Phương pháp Euler và Euler cải tiến cho ta các nghiệm số của phương trình tại một số điểm xác định trước. Ý tưởng của phương pháp Euler là sử dụng khai triển Taylor cho hàm y(x) tại điểm x_0

$$y(x) \simeq \sum \frac{y^k}{k!}(x - x_0)^k$$

Trường hợp đơn giản nhất ứng với m = 1 với lưu ý $y' = f(x, y)$, ta có phương pháp Euler

$$y_{k+1} \simeq y_k + (x_{k+1} - x_k) f(x_k, y_k)$$

Để tìm nghiệm chính xác hơn, người ta thay đổi phương trình trên và thu được phương pháp Euler cải tiến

$$\begin{cases} y_{k+1} \simeq y_k + (x_{k+1} - x_k) f(x_k, y_k) \\ y_{k+1} \simeq y_k + \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, y_{k+1})] \end{cases}$$

Tìm nghiệm số bằng phương pháp Euler
Bước 1: Khai báo hàm số f(x, y) và điều kiệu đầu y_0.
Bước 2: Tìm các giá trị xấp xỉ y_i

1.3 Tìm nghiệm số bằng phương pháp Runge-Kutta

Nội dung cơ bản của phương pháp Runge-Kutta là tăng độ chính xác của y_{i+1} bằng cách thêm các điểm trung gian giữa x_i và $x_i + 1$. Khi đó phần số gia $\Delta y_i = y_{i+1} - y_i$ được tính bởi.

$$\Delta y_i = c_1 k_1(h_i) + c_2 k_2(h_i) + ... + c_r k_r(h_i)$$

với c_j là các hệ số và $k_j\ (h_i)$ là các hàm số được xác định đựa trên $x_i y_j$ và $f(x_i, x_j)$

a) Công thức Runge-Kutta bậc hai

$$\begin{cases} k_1 = h_i f(x_i, y_i) \\ k_2 = h_i f(x_{i+1}, y_i + k_1) \\ y_{i+1} \simeq y_i + \frac{1}{2}(k_1 + k_2) \end{cases}$$

b) Công thức Runge-Kutta bậc ba

$$\begin{cases} k_1 = h_i f(x_i, y_i) \\ k_2 = h_i f(x_{i+1} + \frac{1}{2}h_i, y_i + \frac{1}{2}k_1) \\ k_3 = h_i f(x_{i+1} + h_i, y_i - k_1 + 2k_2) \\ y_{i+1} \simeq y_i + \frac{1}{6}(k_1 + 4k_2 + k_3) \end{cases}$$

Tìm nghiệm số bằng phương pháp Runge-Kutta
Bước 1: Khai báo hàm số f(x, y) và điều kiệu đầu y_0.
Bước 2: Tìm các giá trị xấp xỉ y_i

2 Thực hành trên máy tính

2.1 Tìm nghiệm hàm bằng phương pháp lặp

Bài tập 1. Tìm nghiệm của phương trình sau với sai số tương đối cho phép δy = 0.1%

$$\begin{cases} y' = x + y \\ y(0) = 1 \end{cases} \quad x \in [0, 0.4]$$

STT	y(x)	δy_n	$\delta y_n < \delta y$
1			
2			
3			
4			

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def f(x, y):
	  return x + y
	
	x0, y0 = 0, 1
	x_end = 0.4
	delta_y = 0.001
	
	h = 0.1
	x_values = [x0]
	y_values = [y0]
	
	while x_values[-1] < x_end:
	  x_n = x_values[-1]
	  y_n = y_values[-1]
	
	  y_star = y_n + h * f(x_n, y_n)
	
	  y_next = y_n + (h / 2) * (f(x_n, y_n) + f(x_n + h, y_star))
		Định nghĩa phương trình vi phân y' = x + y
		
		Điều kiện ban đầu
		
		Khởi tạo giá trị bước h ban đầu
		
		Kiểm tra sai số

	error = abs((y_next - y_n) / y_next)
	if error > delta_y: # Nếu sai số lớn hơn mức cho phép, giảm bước h
	h /= 2
	continue
	
	x_values.append(x_n + h)
	y_values.append(y_next)
	
	if error < delta_y / 10:
	h *= 2
	
	for x_i, y_i in zip(x_values, y_values):
	print(f"x = {x_i:.4f}, y = {y_i:.6f}")
		Cập nhật giá trị mới
		Nếu sai số quá nhỏ, có thể tăng bước h để tối ưu
		
		In kết quả

Bài tập 2. Tìm nghiệm của phương trình sau với sai số tương đối cho phép δy = 0.1%

$$\begin{cases} y' = 2x^2 + y \\ y(0) = 1 \end{cases} \quad x \in [0, 0.5]$$

STT	y(x)	δy_n	$\delta y_n < \delta y$
1			
2			
3			
4			

2.2 Tìm nghiệm số bằng phương pháp Euler

Bài tập 3. Tìm nghiệm của phương trình sau với các khoảng chia h = 0.1

$$\begin{cases} y' = x + y \\ y(0) = 1 \end{cases} \quad x \in [0, 0.4]$$

x	0	0.1	0.2	0.3	0.4
y_1	1				
y_2	1				

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện
2	def f(x, y): return x + y x0, y0 = 0, 1 x_end = 0.4 h = 0.1	Định nghĩa phương trình vi phân y' = x + y Điều kiện ban đầu

	x_values = np.arange(x0, x_end + h, h)
	y_values = np.zeros_like(x_values)
	y_values[0] = y0 # Giá trị ban đầu
	
	for i in range(len(x_values) - 1):
	  y_values[i + 1] = y_values[i] + h * f(x_values[i], y_values[i])
	
	for x_i, y_i in zip(x_values, y_values):
	  print(f"x = {x_i:.2f}, y = {y_i:.6f}")
		Tạo mảng các giá trị x
		
		Phương pháp Euler
		
		In kết quả

Bài tập 4. Tìm nghiệm của phương trình sau với các khoảng chia h = 0.1

$$\begin{cases} y' = 2x^2 + y \\ y(0) = 1 \end{cases} \quad x \in [0, 0.5]$$

x	0	0.1	0.2	0.3	0.4	0.5
y_1	1					
y_2	1					

2.3 Tìm nghiệm số bằng phương pháp Runge - Kutta

Bài tập 5. Tìm nghiệm của phương trình sau với các khoảng chia h = 0.1

$$\begin{cases} y' = x + y \\ y(0) = 1 \end{cases} \quad x \in [0, 0.4]$$

x	0	0.1	0.2	0.3	0.4
y	1				

Hướng dẫn

STT	THỰC HIỆN TRÊN MÁY TÍNH	GIẢI THÍCH
1	import numpy as np	Khai báo thư viện

2	def f(x, y):
	  return x + y
	
	x0, y0 = 0, 1
	x_end = 0.4
	h = 0.1
	
	x_values = np.arange(x0, x_end + h, h)
	y_values = np.zeros_like(x_values)
	y_values[0] = y0
	
	for i in range(len(x_values) - 1):
	  x_i, y_i = x_values[i], y_values[i]
	
	  k1 = h * f(x_i, y_i)
	  k2 = h * f(x_i + h/2, y_i + k1/2)
		Định nghĩa phương trình vi phân y' = x + y
		
		Điều kiện ban đầu
		
		Tạo mảng các giá trị x
		
		Phương pháp Runge-Kutta bậc 4

	k3 = h * f(x_i + h/2, y_i + k2/2) k4 = h * f(x_i + h, y_i + k3) y_values[i + 1] = y_i + (k1 + 2*k2 + 2*k3 + k4) / 6 print("Bảng giá trị x và y tính theo phương pháp Runge-Kutta bậc 4:") print("x:", x_values) print("y:", y_values)	In kết quả

Bài tập 6. Tìm nghiệm của phương trình sau với các khoảng chia h = 0.1

$$\begin{cases} y' = 2x^2 + y \\ y(0) = 1 \end{cases} \qquad x \in [0, 0.5]$$

x	0	0.1	0.2	0.3	0.4	0.5
y	1					

3 Lập trình tính toán

Bài tập 7. Thực hiện các yêu cầu sau

a) Viết function giải phương trình vi phân bằng phương pháp lặp.

b) Sử dụng function mới viết giải lại Bài tập 1 và Bài tập 2.

Bài tập 8. Thực hiện các yêu cầu sau

a) Viết function giải phương trình vi phân bằng phương pháp Euler.

b) Sử dụng function mới viết giải lại Bài tập 3 và Bài tập 4.

Bài tập 9. Thực hiện các yêu cầu sau

a) Viết function giải phương trình vi phân bằng phương pháp Euler cải tiến.

b) Sử dụng function mới viết giải lại Bài tập 3 và Bài tập 4.

Bài tập 10. Thực hiện các yêu cầu sau

a) Viết function giải phương trình vi phân bằng phương pháp Runge - Kutta bậc 2.

b) Sử dụng function mới viết giải lại Bài tập 5 và Bài tập 6.

Bài tập 11. Thực hiện các yêu cầu sau

a) Viết function giải phương trình vi phân bằng phương pháp Runge - Kutta bậc 3.

b) Sử dụng function mới viết giải lại Bài tập 5 và Bài tập 6.