

Mục lục

1 Chuẩn bị	1
1.1 Kiến thức về giải tích	1
1.2 Sai số làm tròn và số học máy tính	3
1.3 Thuật toán và sự hội tụ	3
1.4 Python: ngôn ngữ tính toán và lập trình	3
1.5 Python + VS Code: giải tích và đại số	11
2 Giải phương trình một biến	22
2.1 Phương pháp chia đôi	22
2.2 Phương pháp Newton và mở rộng	24
2.3 Lập điểm bất động	30
2.4 Phân tích sai số của các phương pháp lặp	34
2.5 Tăng tốc độ hội tụ	34
2.6 Nghiệm của đa thức và phương pháp Müller	35
3 Nội suy và xấp xỉ bằng đa thức	36
3.1 Nội suy tổng quát	36
3.2 Đa thức nội suy	37
3.3 Xấp xỉ số liệu và phương pháp Neville	41
3.4 Sai phân chia	41
3.5 Nội suy Hermite	42
3.6 Nội suy Newton	42
3.7 Nội suy spline bậc ba	45
3.8 Đường cong tham số	45
4 Đạo hàm và tích phân bằng số	46
4.1 Đạo hàm bằng số	47
4.2 Ngoại suy Richardson	51
4.3 Tích phân bằng số	51
4.4 Tích phân Romberg	56

4.5	Phương pháp cầu phương thích ứng	56
4.6	Cầu phương Gauss	56
4.7	Tích phân bội	57
4.8	Tích phân suy rộng	57
5	Bài toán giá trị ban đầu của phương trình vi phân thường	58
5.1	Lý thuyết cơ bản về bài toán giá trị ban đầu	59
5.2	Phương pháp Picard	60
5.3	Phương pháp chuỗi Taylor	64
5.4	Phương pháp Euler	67
5.5	Phương pháp Taylor bậc cao	69
5.6	Phương pháp Runge–Kutta	70
5.7	Điều khiển sai số và phương pháp Runge–Kutta–Fehlberg	74
5.8	Phương pháp đa bước	74
5.9	Phương pháp đa bước với bước nhảy biến thiên	74
5.10	Phương pháp ngoại suy	74
5.11	Phương trình cấp cao và hệ phương trình vi phân	74
5.12	Sự ổn định	74
5.13	Phương trình vi phân cứng	74
6	Phương pháp trực tiếp giải hệ phương trình tuyến tính	68
6.1	Hệ phương trình tuyến tính	68
6.2	Chiến thuật chốt	69
6.3	Đại số tuyến tính và ma trận nghịch đảo	69
6.4	Định thức của ma trận	69
6.5	Phân tích ma trận	69
6.6	Các dạng ma trận đặc biệt	69
7	Kỹ thuật lặp trong đại số tuyến tính	70
7.1	Chuẩn của vectơ và ma trận	70
7.2	Giá trị riêng và vectơ riêng	72
7.3	Lặp điểm bất động	72
7.4	Kỹ thuật lặp Jacobi và Gauss–Seidel	76
7.5	Ma trận nghịch đảo	79
7.6	Kỹ thuật giảm dư giải hệ tuyến tính	80
7.7	Giới hạn sai số và tinh chỉnh phép lặp	80
7.8	Phương pháp gradient liên hợp	80

8 Lý thuyết xấp xỉ	81
8.1 Xấp xỉ bình phương nhỏ nhất	81
8.2 Đa thức trực giao và xấp xỉ bình phương nhỏ nhất	85
8.3 Đa thức Chebyshev và [Economization] chuỗi lũy thừa	86
8.4 Xấp xỉ hàm hữu tỷ	86
8.5 Xấp xỉ đa thức lượng giác	86
8.6 Biến đổi Fourier nhanh	86
9 Xấp xỉ giá trị riêng	84
9.1 Đại số tuyến tính và giá trị riêng	84
9.2 Ma trận trực giao và biến đổi đồng dạng	84
9.3 Phương pháp lũy thừa	84
9.4 Phương pháp Householder	84
9.5 Thuật toán QR	84
9.6 Phân tích giá trị kỳ dị	84
10 Nghiệm số của hệ phương trình phi tuyến	85
10.1 Điểm bất động của hàm nhiều biến	85
10.2 Phương pháp Newton	85
10.3 Phương pháp tựa Newton	85
10.4 Phương pháp độ dốc nhất	85
10.5 Đồng luân và các phương pháp mở rộng	85
11 Bài toán giá trị biên của phương trình vi phân thường	86
11.1 Phương pháp bắn tuyến tính	86
11.2 Phương pháp bắn cho bài toán phi tuyến	86
11.3 Phương pháp sai phân hữu hạn cho bài toán tuyến tính	86
11.4 Phương pháp sai phân hữu hạn cho bài toán phi tuyến	87
11.5 Phương pháp Rayleigh–Ritz	87
12 Nghiệm số của phương trình đạo hàm riêng	88
12.1 Phương trình đạo hàm riêng Elliptic	88
12.2 Phương trình đạo hàm riêng Parabolic	89
12.3 Phương trình đạo hàm riêng Hyperbolic	89
12.4 Giới thiệu về phương pháp phần tử hữu hạn	89

Chương 5

Bài toán giá trị ban đầu của phương trình vi phân thường

Xét phương trình vi phân cấp một, bài toán giá trị ban đầu

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

trong đó $y, f \in \mathbb{R}^m$, hay

$$\begin{cases} y'_1 = f_1(x, y_1, y_2, \dots, y_m) \\ y'_2 = f_2(x, y_1, y_2, \dots, y_m) \\ \dots \\ y'_m = f_m(x, y_1, y_2, \dots, y_m) \end{cases}$$

với $y_i(x_0) = y_i^{(0)}, i = \overline{1, m}$.

Phương trình vi phân cấp cao

$$\begin{cases} y^{(m)} = f(x, y, y', y'', \dots, y^{(m-1)}) \\ y^{(i)}(x_0) = y_i^{(0)}, i = \overline{0, m-1} \end{cases}$$

có thể đưa được về hệ phương trình vi phân cấp một. Đặt hàm phụ $y_1 = y, y_2 = y', y_3 = y'', \dots, y_m = y^{(m-1)}$, ta được

$$\begin{cases} y'_1 = y_2 \\ y'_2 = y_3 \\ \dots \\ y'_{m-1} = y_m \\ y'_m = f(x, y_1, y_2, \dots, y_m) \end{cases}$$

với $y_i(x_0) = y_{i-1}^{(0)}$, $i = \overline{1, m}$.

Xét các ví dụ:

Ví dụ 5.1. $y' = y - x$, $y(0) = 2$ (nghiệm $y = 1 + x + e^x$).

Ví dụ 5.2.
$$\begin{cases} y' = xy - z \\ z' = y + z - 1 \end{cases}, \begin{cases} y(1) = -1 \\ z(1) = 2. \end{cases}$$

Ví dụ 5.3. $y''' - xy'' + y = 0$; $y(-1) = 1$, $y'(-1) = 0$, $y''(-1) = -2$.

Đặt $y' = z$, $u = y'' (= z')$ thì $u' = y''' = xy'' - y = xu - y$. Ta có hệ

$$\begin{cases} y' = z \\ z' = u \\ u' = xu - y \end{cases}, \begin{cases} y(-1) = 1 \\ z(-1) = 0 \\ u(-1) = -2. \end{cases}$$

5.1 Lý thuyết cơ bản về bài toán giá trị ban đầu

Định nghĩa 5.1. Hàm $f(x, y)$ gọi là thỏa mãn điều kiện Lipschitz theo biến y trên miền $D \subset \mathbb{R}^2$ nếu $\exists L > 0$, $\forall (x, y_1), (x, y_2) \in D$,

$$|f(x, y_1) - f(x, y_2)| \leq L |y_1 - y_2|.$$

L gọi là hằng số Lipschitz của f .

Định nghĩa 5.2. Tập $D \subset \mathbb{R}^2$ khác rỗng gọi là lồi nếu

$$\forall (x_1, y_1), (x_2, y_2) \in D, \lambda \in [0, 1] \Rightarrow ((1 - \lambda)x_1 + \lambda x_2, (1 - \lambda)y_1 + \lambda y_2) \in D.$$

Định lý 5.1. Giả sử $f(x, y)$ xác định trên tập lồi $D \subset \mathbb{R}^2$. Nếu

$$\exists L > 0, |\partial_y f(x, y)| \leq L, \forall (x, y) \in D,$$

thì f thỏa mãn điều kiện Lipschitz trên D theo biến y với hằng số Lipschitz L .

Định lý 5.2. Giả sử $D = \{(x, y) \mid x_0 \leq x \leq \bar{x}, -\infty < y < \infty\}$, và $f(x, y)$ liên tục trên D theo biến y , thì bài toán giá trị ban đầu

$$y'(x) = f(x, y(x)), x_0 \leq x \leq \bar{x}; y(x_0) = y_0$$

có nghiệm duy nhất $y(x)$ với $x_0 \leq x \leq \bar{x}$.

Trong hai phần tiếp theo, ta xét các phương pháp giải tích để giải gần đúng bài toán giá trị ban đầu, gồm phương pháp Picard và phương pháp chuỗi Taylor. Trong các phương pháp này, ta tìm dãy hàm $y_n(x)$ hội tụ tới nghiệm $y(x)$.

5.2 Phương pháp Picard

$$\begin{aligned} y_0(x) &= y_0, \quad \forall x \\ y_n(x) &= y_0 + \int_{x_0}^x f[t, y_{n-1}(t)] dt, \quad n = 1, 2, \dots \end{aligned} \quad (5.1)$$

và với hệ phương trình, các vế của (5.1) đều là các vectơ, được thực hiện theo từng thành phần, trong đó phép tính tích phân của vectơ được hiểu là lấy tích phân của mọi thành phần.

$$\begin{aligned} y_i^{(0)}(x_0) &= y_i^{(0)}, \quad i = \overline{1, m} \\ y_i^{(n)}(x) &= y_i^{(0)} + \int_{x_0}^x f_i[t, y_1^{(n-1)}(t), y_2^{(n-1)}(t), \dots, y_m^{(n-1)}(t)] dt, \quad i = \overline{1, m}, \quad n = 1, 2, \dots \end{aligned}$$

Ví dụ 5.4. Trong Ví dụ 5.1, tính tới $y_3(x)$.

Giải. Công thức lặp

$$\begin{aligned} y_0(x) &= 2 \\ y_n(x) &= 2 + \int_0^x [y_{n-1}(t) - t] dt. \end{aligned}$$

Quá trình tính cụ thể:

$$\begin{aligned} y_1(x) &= 2 + \int_0^x (2 - t) dt = 2 + 2x - \frac{x^2}{2} \\ y_2(x) &= 2 + \int_0^x \left[\left(2 + 2t - \frac{t^2}{2} \right) - t \right] dt = 2 + 2x + \frac{x^2}{2} - \frac{x^3}{6} \\ y_3(x) &= 2 + \int_0^x \left[\left(2 + 2t + \frac{t^2}{2} - \frac{t^3}{6} \right) - t \right] dt = 2 + 2x + \frac{x^2}{2} + \frac{x^3}{6} - \frac{x^4}{24}. \end{aligned} \quad \square$$

```
1 f = lambda x, y: y - x
2 from sympy import *
3 x, t = symbols('x t')
4 x0, y0 = 0, 2
```

Cách 1: đệ quy

```

1 def y(n, x):
2     if n == 0:
3         return y0
4     return y0 + f(t, y(n-1, t)).integrate((t, x0, x))
6 y(1, x)

```

Cách 2: Phương pháp lặp

```

1 y = y0 + 0*x # biểu thức phải xuất hiện biến mới dùng được cú
    pháp thay thế (xem dòng 4)
2 for _ in range(3):
3     y = y.subs(x, t)
4     y = y0 + f(t, y).integrate((t, x0, x))
5     display(y)

```

Bằng phương pháp quy nạp, ta có thể chứng minh

$$\begin{aligned}
 y_n(x) &= 2 + 2x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} - \frac{x^{n+1}}{(n+1)!} \\
 &= \left(1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}\right) + 1 + x - \frac{x^{n+1}}{(n+1)!} \\
 &\xrightarrow{n \rightarrow \infty} e^x + 1 + x = y(x).
 \end{aligned}$$

Ví dụ 5.5. Trong [Ví dụ 5.2](#), tính tới $y_2(x)$, $z_2(x)$.

Giải. Công thức lặp

$$y_0(x) = -1, z_0(x) = 2,$$

$$y_n(x) = -1 + \int_1^x [ty_{n-1}(t) - z_{n-1}(t)] dt, \quad z_n(x) = 2 + \int_1^x [y_{n-1}(t) + z_{n-1}(t) - 1] dt.$$

Ta có

$$* \quad y_1(x) = -1 + \int_1^x [t(-1) - 2] dt = \frac{3}{2} - 2x - \frac{x^2}{2}$$

$$z_1(x) = 2 + \int_1^x [-1 + 2 - 1] dt = 2$$

$$* \quad y_2(x) = -1 + \int_1^x \left[t \left(\frac{3}{2} - 2t - \frac{t^2}{2} \right) - 2 \right] dt = \frac{25}{24} - 2x + \frac{3x^2}{4} - \frac{2x^3}{3} - \frac{x^4}{8}$$

$$z_2(x) = 2 + \int_1^x \left[\frac{3}{2} - 2t - \frac{t^2}{2} + 2 - 1 \right] dt = \frac{2}{3} + \frac{5x}{2} - x^2 - \frac{x^3}{6}.$$

□

```

1 f = lambda x, y: [x * y[0] - y[1], y[0] + y[1] - 1]
2 f(0, [1, 2])

3 from sympy import *
4 x, t = symbols('x t')

5 import numpy as np

6 x0 = 1
7 y0 = np.array([-1, 2]) + 0*x

```

Cách 1: đệ quy

```

1 def y(n, x):
2     if n == 0:
3         return y0
4     return y0 + [fi.integrate((t, x0, x)) for fi in f(
5         t, y(n-1, t))]

```

y(1, x)

Cách 2: lặp

```

1 y = y0.copy() # không được gán y = y0, vì khi đó thay đổi y
                 sẽ làm thay đổi y0
2 for i in range(2):
3     y = [yi.subs(x, t) for yi in y]
4     y = y0 + [fi.integrate((t, x0, x)) for fi in f(t,
5         y)]
5     display(y)

```

Ví dụ 5.6. Trong [Ví dụ 5.3](#), tính tới $y_3(x)$.

Giải. Công thức lặp

$$y_0(x) = 1, z_0(x) = 0, u_0(x) = -2$$

$$y_n(x) = 1 + \int_{-1}^x z_{n-1}(t) dt, \quad z_n(x) = 0 + \int_{-1}^x u_{n-1}(t) dt,$$

$$u_n(x) = -2 + \int_{-1}^x [tu_{n-1}(t) - y_{n-1}(t)] dt.$$

Ta có

$$\begin{aligned}
 * \quad y_1(x) &= 1 + \int_{-1}^x 0 dt = 1 \\
 z_1(x) &= 0 + \int_{-1}^x -2 dt = -2 - 2x \\
 u_1(x) &= -2 + \int_{-1}^x [t(-2) - 1] dt = -2 - x - x^2. \\
 * \quad y_2(x) &= 1 + \int_{-1}^x (-2 - 2t) dt = -2x - x^2 \\
 z_2(x) &= 0 + \int_{-1}^x (-2 - t - t^2) dt = -\frac{11}{6} - 2x - \frac{x^2}{2} - \frac{x^3}{3} \\
 u_2(x) &= -2 + \int_{-1}^x [t(-2 - t - t^2) - 1] dt = -\frac{25}{12} - x - x^2 - \frac{x^3}{3} - \frac{x^4}{4} \\
 * \quad y_3(x) &= 1 + \int_{-1}^x \left(-\frac{11}{6} - 2t - \frac{t^2}{2} - \frac{t^3}{3} \right) dt = \frac{1}{12} - \frac{11x}{6} - x^2 - \frac{x^3}{6} - \frac{x^4}{12}.
 \end{aligned}$$

□

```

1 f = lambda x, y: [y[1], y[2], x * y[2] - y[0]]
2 f(0, [1, 2, 3])

3 from sympy import *
4 x, t = symbols('x t')

5 import numpy as np

6 x0 = -1
7 y0 = np.array([1, 0, -2]) + 0*x

```

Cách 1: đệ quy

```

1 def y(n, x):
2     if n == 0:
3         return y0
4     return y0 + [fi.integrate((t, x0, x)) for fi in f(
5         t, y(n-1, t))]

```

Cách 2: lặp

```

1 y = y0.copy()
2 for i in range(2):
3     y = [yi.subs(x, t) for yi in y]

```

```

4     y = y0 + [fi.integrate((t, x0, x)) for fi in f(t,
5         y)]
        display(y)

```

5.3 Phương pháp chuỗi Taylor

$$y_n(x) = \sum_{k=0}^n \frac{y^{(k)}(x_0)}{k!} (x - x_0)^k \quad (5.2)$$

và với hệ phương trình:

$$y_{in}(x) = \sum_{k=0}^n \frac{y_i^{(k)}(x_0)}{k!} (x - x_0)^k, \quad i = \overline{1, m}.$$

Ví dụ 5.7. Trong [Ví dụ 5.1](#), tính $y_3(x)$.

Giải. Ta có

$$\begin{aligned}
 y(0) &= 2 \\
 y' &= y - x \quad \Rightarrow y'(0) = y(0) - 0 = 2 \\
 y'' &= y' - 1 \quad \Rightarrow y''(0) = 2 - 1 = 1 \\
 y''' &= y'' \quad \Rightarrow y'''(0) = 1.
 \end{aligned}$$

Suy ra

$$y_3(x) = 2 + \frac{2}{1!}(x - 0) + \frac{1}{2!}(x - 0)^2 + \frac{1}{3!}(x - 0)^3 = 2 + 2x + \frac{x^2}{2} + \frac{x^3}{6}.$$

```

1  from sympy import *
2  x = symbols('x')
3  y = symbols('y', cls=Function)
4
5  P = 2
6  for k in range(1, 4):
7      d = y(x).diff(x, k)
8      for _ in range(k):
9          d = d.subs(y(x).diff(), y(x) - x).simplify() # mỗi
          bước lặp hạ được 1 cấp đạo hàm
10     P += d / factorial(k) * (x - 0)**k # y_k(x)
11 P

```

□

Để thấy $y^{(k)} = y'' \forall k \geq 2 \Rightarrow y^{(k)}(0) = y''(0) = 1$. Khi đó

$$\begin{aligned} y_n(x) &= 2 + 2x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} \\ &= \left(1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} \right) + 1 + x \\ &\xrightarrow{n \rightarrow \infty} e^x + 1 + x = y(x). \end{aligned}$$

Ví dụ 5.8. Trong [Ví dụ 5.2](#), tính $y_3(x)$, $z_3(x)$.

Giải. Ta có

$$\begin{aligned} y(1) &= -1, \quad z(1) = 2 \\ * \quad y' &= xy - z & \Rightarrow y'(1) &= 1y(1) - z(1) = 1(-1) - 2 = -3 \\ \quad z' &= y + z - 1 & \Rightarrow z'(1) &= y(1) + z(1) - 1 = -1 + 2 - 1 = 0 \\ * \quad y'' &= y + xy' - z' & \Rightarrow y''(1) &= -1 + 1(-3) - 0 = -4 \\ \quad z'' &= y' + z' & \Rightarrow z''(1) &= -3 + 0 = -3 \\ * \quad y''' &= 2y' + xy'' - z'' & \Rightarrow y'''(1) &= 2(-3) + 1(-4) - (-3) = -7 \\ \quad z''' &= y'' + z'' & \Rightarrow z'''(1) &= -4 + (-3) = -7. \end{aligned}$$

Suy ra

$$\begin{aligned} y_3(x) &= -1 + (-3)(x-1) + \frac{-4}{2!}(x-1)^2 + \frac{-7}{3!}(x-1)^3 \\ &= 1 - 3(x-1) - 2(x-1)^2 - \frac{7}{6}(x-1)^3 \\ z_3(x) &= 2 + 0(x-1) + \frac{-3}{2!}(x-1)^2 + \frac{-7}{3!}(x-1)^3 = 2 - \frac{3}{2}(x-1)^2 - \frac{7}{6}(x-1)^3. \end{aligned}$$

Mã Python gần giống [Ví dụ 5.7](#), để tính $y_3(x)$, chỉ cần sửa dòng 3 thành

```
y, z = symbols('y z', cls=Function)
```

và dòng 8 thành

```
d = d.subs({y(x).diff(): x * y(x) - z(x), z(x).diff():
            y(x) + z(x) - 1}).simplify()
```

Còn để tính $z_3(x)$, thay dòng 6 bởi

```
d = z(x).diff(x, k)
```

□

Ví dụ 5.9. Trong [Ví dụ 5.3](#), tính $y_5(x)$.

Giải. Ta có

$$y(-1) = 1, y'(-1) = 0, y''(-1) = -2$$

$$y''' = xy'' - y \Rightarrow y'''(-1) = (-1)(-2) - 1 = 1$$

$$y^{(4)} = y'' + xy''' - y' \Rightarrow y^{(4)}(-1) = -2 + (-1)1 - 0 = -3$$

$$y^{(5)} = 2y''' + xy^{(4)} - y'' \Rightarrow y^{(5)}(-1) = 2 \cdot 1 + (-1)(-3) - (-2) = 7$$

Suy ra

$$\begin{aligned} y_5(x) &= 1 + 0(x+1) + \frac{-2}{2!}(x+1)^2 + \frac{1}{3!}(x+1)^3 + \frac{-3}{4!}(x+1)^4 + \frac{7}{5!}(x+1)^5 \\ &= 1 - (x+1)^2 + \frac{1}{6}(x+1)^3 - \frac{1}{8}(x+1)^4 + \frac{7}{120}(x+1)^5. \end{aligned}$$

Mã Python tương tự [Ví dụ 5.7](#).

```

1 from sympy import *
2 x = symbols('x')
3 y = symbols('y', cls=Function)

4 P = 1 + 0*(x+1) + (-2) / factorial(2) * (x+1)**2
5 for k in range(3, 6):
6     d = y(x).diff(x, k)
7     for _ in range(k - 2): # 2 = cấp của phương trình vi phân -
1      1
8         d = d.subs({y(x).diff(x, 3): x * y(x).diff(x, 2) -
9                     y(x)}).simplify()
9     d = d.subs({x: -1, y(x): 1, y(x).diff(): 0, y(x).diff(
10      x, 2): -2})
11     P += d / factorial(k) * (x + 1)**k
12 P

```

□

Các phương pháp giải bài toán giá trị ban đầu trong chương này chủ yếu là các phương pháp số, ở đây ta xấp xỉ $y(x_n)$ bởi y_n tại các điểm $x_0 < x_1 < \dots < x_N \leq \bar{x}$.

Ký hiệu $h_n = x_{n+1} - x_n$, $n = \overline{0, N-1}$. Trong hầu hết trường hợp, các bước nhảy bằng nhau, tức là $x_{n+1} - x_n = h$, $\forall n$.

5.4 Phương pháp Euler

$$y_{n+1} = y_n + h_n f(x_n, y_n), \quad n = 0, 1, \dots \quad (5.3)$$

và với hệ phương trình:

$$y_i^{(n+1)} = y_i^{(n)} + h_n f_i(x_n, y_1^{(n)}, y_2^{(n)}, \dots, y_m^{(n)}), \quad i = \overline{1, m}.$$

Các phép tính được ghi trong bảng

n	x_n	y_n	$h_n f(x_n, y_n)$
...			
n	x_n	y_n	$d = h_n f(x_n, y_n)$
$n+1$	$x_{n+1} = x_n + h_n$	$y_{n+1} = y_n + d$	
...			

và trong trường hợp hệ phương trình vi phân, các cột ứng với các vectơ y_n và $h_f(x_n, y_n)$ được tách thành nhiều cột, tương ứng với $y_i^{(n)}$ và $h_n f_i(x_n, y_1^{(n)}, y_2^{(n)}, \dots, y_m^{(n)}), i = \overline{1, m}$.

Ví dụ 5.10. Trong **Ví dụ 5.1**, tính nghiệm gần đúng tại các điểm 0.2, 0.3, 0.5. So sánh với nghiệm đúng $y(x_n) = e^{x_n} + 1 + x_n$.

Giải. Công thức lặp

$$y_{n+1} = y_n + h_n (y_n - x_n).$$

Ta có bảng tính

n	x_n	y_n	$h_n f(x_n, y_n)$	$y(x_n)$	
0	0	2	0.4		
1	0.2	2.4	0.22	2.4214	$= 0.1(2.4 - 0.2)$
2	0.3	2.62	0.464	2.64986	$= 2.4 + 0.22$
3	0.5	3.084		3.14872	

```

1 f = lambda x, y: y - x
2 X = [0, 0.2, 0.3, 0.5]

3 y = 2
4 for n in range(3):
5     h = X[n+1] - X[n]
6     y = y + h * f(X[n], y)
7     print(y)

```

□

Dòng 4–7 là đoạn mã chính, được giữ nguyên cho hai ví dụ sau về giải hệ phương trình vi phân cấp một, trong đó phương trình vi phân cấp ba được đưa về hệ ba phương trình vi phân cấp một. Ngoài ra, vì trong dòng 6, biến y và hàm f là vectơ, nên chỉ cần khai báo ban đầu f là vectơ.

Ví dụ 5.11. Trong [Ví dụ 5.2](#), tìm nghiệm gần đúng tại các điểm 1.1, 1.3, 1.5.

Giải. Công thức lặp

$$\begin{aligned}y_{n+1} &= y_n + h_n (x_n y_n - z_n) \\ z_{n+1} &= z_n + h_n (y_n + z_n - 1).\end{aligned}$$

Bảng tính

n	x_n	y_n	z_n	$h_n f_y(x_n, y_n, z_n)$	$h_n f_z(x_n, y_n, z_n)$
0	1	-1	2	-0.3	0
1	1.1	-1.3	2	-0.686	-0.06
2	1.3	-1.986	1.94	-0.90436	-0.2092
3	1.5	-2.89036	1.7308		

```
1 import numpy as np
2 f = lambda x, y: np.array([x * y[0] - y[1], y[0] + y[1] -
3     1])
4 f(0, [1, 2]) # tính thử → array([-2, 2])
5
6 X = [1, 1.1, 1.3, 1.5]
7 y = [-1, 2]
8 for n in range(3):
9     h = X[n+1] - X[n]
10    y = y + h * f(X[n], y)
11    print(y)
```

□

Ví dụ 5.12. Trong [Ví dụ 5.3](#), tìm nghiệm gần đúng tại $-0.8, -0.6, -0.5$.

Giải. Công thức lặp

$$\begin{aligned}y_{n+1} &= y_n + h_n z_n \\ z_{n+1} &= z_n + h_n u_n \\ u_{n+1} &= u_n + h_n (x_n u_n - y_n).\end{aligned}$$

Bảng tính

n	x_n	y_n	z_n	u_n	$h_n f_y$	$h_n f_z$	$h_n f_u$
0	-1	1	0	-2	0	-0.4	0.2
1	-0.8	1	-0.4	-1.8	-0.08	-0.36	0.088
2	-0.6	0.92	-0.76	-1.712	-0.076	-0.1712	0.01072
3	-0.5	0.844					

```

1 import numpy as np
2 f = lambda x, y: np.array([y[1], y[2], x * y[2] - y[0]])
3 f(0, [1, 2, 3]) # tính thử → array([ 2, 3, -1])

4 X = [-1, -0.8, -0.6, -0.5]
5 y = [1, 0, -2]
6 for n in range(3):
7     h = X[n+1] - X[n]
8     y = y + h * f(X[n], y)
9     print(y)

```

□

Định lý 5.3. Giả sử f liên tục và thỏa mãn điều kiện Lipschitz với hằng số L trên

$$D = \{(x, y) \mid x_0 \leq x \leq \bar{x}, -\infty < y < \infty\},$$

và

$$\exists M, |y''(x)| \leq M, \forall x \in [x_0, \bar{x}],$$

trong đó $y''(x)$ là nghiệm duy nhất của bài toán giá trị ban đầu

$$y' = f(x, y), \quad x_0 \leq x \leq \bar{x}; \quad y(x_0) = y_0.$$

Đặt y_0, y_1, \dots, y_N là các xấp xỉ sinh bởi phương pháp Euler. Khi đó $\forall n = \overline{0, N}$,

$$|y(x_n) - y_n| \leq \frac{hM}{2L} [e^{L(x_n - a)} - 1].$$

5.5 Phương pháp Taylor bậc cao

$$y_{n+1} = y_n + h_n T^{(r)}(x_n, y_n), \quad i = \overline{0, N-1}, \quad (5.4)$$

trong đó

$$T^{(r)}(t_n, y_n) = f(x_n, y_n) + \frac{h}{2} f'(x_n, y_n) + \cdots + \frac{h^{r-1}}{r!} f^{(r-1)}(x_n, y_n).$$

Phương pháp Euler là phương pháp Taylor bậc một.

Định lý 5.4. Nếu dùng phương pháp Taylor bậc k để xấp xỉ nghiệm của bài toán

$$y'(x) = f(x, y(x)), \quad x_0 \leq x \leq \bar{x}; \quad y(x_0) = y_0,$$

với bước chia h , và nếu $y \in C^{r+1}[a, b]$, thì sai số cụt địa phương là $O(h^r)$.

5.6 Phương pháp Runge–Kutta

5.6.1 Phương pháp Runge–Kutta bậc hai

5.6.2 Phương pháp trung điểm

5.6.3 Phương pháp Euler cải biên

5.6.4 Phương pháp Runge–Kutta bậc cao

5.6.5 Runge–Kutta bậc bốn

$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}, \quad n = 1, 2, \dots \quad (5.5)$$

trong đó k_1, k_2, k_3, k_4 là các giá trị tạm thời tính tại mỗi bước n :

$$\begin{cases} k_1 = h_n f(x_n, y_n) \\ k_2 = h_n f\left(x_n + \frac{h_n}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 = h_n f\left(x_n + \frac{h_n}{2}, y_n + \frac{k_2}{2}\right) \\ k_4 = h_n f(x_n + h_n, y_n + k_3) \end{cases}$$

và với hệ phương trình

$$y_i^{(n+1)} = y_i^{(n)} + \frac{k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i}}{6}, \quad i = \overline{1, m}.$$

trong đó, với $i = \overline{1, m}$:

$$\begin{aligned} k_{1i} &= h_n f_i(x_n, y_1^{(n)}, \dots, y_m^{(n)}) \\ k_{2i} &= h_n f_i\left(x_n + \frac{h_n}{2}, y_1^{(n)} + \frac{k_{11}}{2}, \dots, y_m^{(n)} + \frac{k_{1m}}{2}\right) \end{aligned}$$

$$k_{3i} = h_n f_i \left(x_n + \frac{h_n}{2}, y_1^{(n)} + \frac{k_{21}}{2}, \dots, y_m^{(n)} + \frac{k_{2m}}{2} \right)$$

$$k_{4i} = h_n f_i \left(x_n, y_1^{(n)} + k_{31}, \dots, y_m^{(n)} + k_{3m} \right).$$

Trong bảng tính của phương pháp, cột x và y được tính toán để làm đối số “phù hợp” cho cột $k = hf(x, y)$. Đối với hệ phương trình, tương tự bảng tính của phương pháp Euler, ba cột này cũng tách thành nhiều cột, ứng với các thành phần của vectơ tương ứng.

n	x	y	$k = h_n f(x, y)$
...			
n	x_n	y_n	$k_1 = h_n f(x_n, y_n)$
	$x_n + \frac{h_n}{2}$	$y_n + \frac{k_1}{2}$	$k_2 = h_n f \left(x_n + \frac{h_n}{2}, y_n + \frac{k_1}{2} \right)$
	$x_n + \frac{h_n}{2}$	$y_n + \frac{k_2}{2}$	$k_3 = h_n f \left(x_n + \frac{h_n}{2}, y_n + \frac{k_2}{2} \right)$
	$x_n + h_n$	$y_n + k_3$	$k_4 = h_n f(x_n + h_n, y_n + k_3)$
	$d = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$		
$n + 1$	$x_{n+1} = x_n + h_n$	$y_{n+1} = y_n + d$...

Ví dụ 5.13. Trong [Ví dụ 5.1](#), tính nghiệm gần đúng tại 0.2, 0.3. So sánh với nghiệm đúng.

Giải. Công thức lặp

$$k_1 = h_n (y_n - x_n)$$

$$k_2 = h_n \left[\left(y_n + \frac{k_1}{2} \right) - \left(x_n + \frac{h_n}{2} \right) \right]$$

$$k_3 = h_n \left[\left(y_n + \frac{k_2}{2} \right) - \left(x_n + \frac{h_n}{2} \right) \right]$$

$$k_4 = h_n [(y_n + k_3) - (x_n + h_n)]$$

$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}.$$

Bảng tính

n	x	y	$h_n f(x, y)$	$y(x_n)$
0	0	2	0.4	
	0.1	2.2	0.42	
	0.1	2.21	0.422	
	0.2	2.422	0.4444	
	0.4214			
1	0.2	2.4214	0.22214	2.4214
	0.25	2.53247	0.228247	
	0.25	2.53552	0.228552	
	0.3	2.64995	0.234995	
	0.228456			
2	0.3	2.64986		2.64986

```

1 f = lambda x, y: y - x
2 X = [0, 0.2, 0.3]

3 y = 2
4 for n in range(len(X) - 1):
5     h = X[n+1] - X[n]
6     k1 = h * f(X[n], y)
7     k2 = h * f(X[n] + h/2, y + k1/2)
8     k3 = h * f(X[n] + h/2, y + k2/2)
9     k4 = h * f(X[n] + h, y + k3)
10    y = y + (k1 + 2*k2 + 2*k3 + k4)/6
11    print(y)

```

□

Dòng 1–3 giống dòng 1–3 ở Ví dụ 5.10, còn dòng 4–11 là đoạn chương trình chính, cũng được giữ nguyên cho hai ví dụ dưới đây.

Ví dụ 5.14. Trong Ví dụ 5.2, tìm nghiệm gần đúng tại 1.1, 1.3.

Giải. Đặt $Y = \begin{bmatrix} y \\ z \end{bmatrix}$, $Y_n = \begin{bmatrix} y_n \\ z_n \end{bmatrix}$, $f(x, Y) = \begin{bmatrix} xy - z \\ y + z - 1 \end{bmatrix}$, ta có $Y' = f(x, Y)$, với $Y(1) = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$, và công thức lặp

$$k_1 = h_n f(x_n, Y_n)$$

$$k_2 = h_n f\left(x_n + \frac{h_n}{2}, Y_n + \frac{k_1}{2}\right)$$

$$\begin{aligned}
 k_3 &= h_n f \left(x_n + \frac{h_n}{2}, Y_n + \frac{k_2}{2} \right) \\
 k_4 &= h_n f (x_n + h_n, Y_n + k_3), \\
 Y_{n+1} &= Y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}.
 \end{aligned}$$

Bảng tính

n	x	y	z	$h_n f_y$	$h_n f_z$
0	1	−1	2	−0.3	0
	1.05	−1.15	2	−0.32075	−0.015
	1.05	−1.16038	1.9925	−0.321089	−0.0167875
	1.1	−1.32109	1.98321	−0.343641	−0.0337877
				−0.32122	−0.0162271
1	1.1	−1.32122	1.98377	−0.687423	−0.0674894
	1.2	−1.66493	1.95003	−0.789589	−0.142981
	1.2	−1.71601	1.91228	−0.7943	−0.160746
	1.3	−2.11552	1.82303	−0.91464	−0.258499
				−0.794974	−0.155574
2	1.3	−2.11619	1.8282		

Mã Python gồm (1) khai báo: dòng 1–5 của Ví dụ 5.11, và (2) chương trình chính: dòng 5–11 của Ví dụ 5.13. □

Ví dụ 5.15. Trong Ví dụ 5.3, tìm nghiệm gần đúng tại $-0.8, -0.6$.

Giải. Đặt $Y = \begin{bmatrix} y \\ z \\ u \end{bmatrix}$, $Y_n = \begin{bmatrix} y_n \\ z_n \\ u_n \end{bmatrix}$, $f(x, Y) = \begin{bmatrix} z \\ u \\ xu - y \end{bmatrix}$, ta có $Y' = f(x, Y)$, với $Y(-1) = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$. Công thức lặp được trình bày như trong Ví dụ 5.14.

Bảng tính

n	x	y	z	u	$h_n f_y$	$h_n f_z$	$h_n f_u$
0	−1	1	0	−2	0	−0.4	0.2
	−0.9	1	−0.2	−1.9	−0.04	−0.38	0.142
	−0.9	0.98	−0.19	−1.929	−0.038	−0.3858	0.15122
	−0.8	0.962	−0.3858	−1.84878	−0.07716	−0.369756	0.103405
	−0.03886					−0.383559	0.148307
1	−0.8	0.96114	−0.383559	−1.85169	−0.0767119	−0.370339	0.104043
	−0.7	0.922784	−0.568729	−1.79967	−0.113746	−0.359934	0.0673971
	−0.7	0.904267	−0.563526	−1.81799	−0.112705	−0.363599	0.0736657
	−0.6	0.848435	−0.747158	−1.77803	−0.149432	−0.355605	0.0436763
	−0.113174					−0.362168	0.0716408
2	−0.6	0.847966					

Mã Python gồm (1) khai báo: dòng 1–5 của Ví dụ 5.12 và (2) chương trình chính: dòng 5–11 của Ví dụ 5.13. □

5.7 Điều khiển sai số và phương pháp Runge–Kutta–Fehlberg

5.8 Phương pháp đa bước

5.9 Phương pháp đa bước với bước nhảy biến thiên

5.10 Phương pháp ngoại suy

5.11 Phương trình cấp cao và hệ phương trình vi phân

5.12 Sự ổn định

5.13 Phương trình vi phân cứng

Tài liệu tham khảo

- [1] Phạm Kỳ Anh. *Giải tích số*. Đại học Quốc gia Hà Nội, 2002. 284 trang.
- [2] Richard L. Burden, Douglas J. Faires **and** Annette M. Burden. *Numerical Analysis*. phiên bản 10. Cengage Learning, 2016. 918 trang.
- [3] NumPy community. *NumPy User Guide*. phiên bản 1.22.0. 531 trang. URL: <https://numpy.org/doc/stable>.
- [4] SciPy community. *SciPy Reference Guide*. phiên bản 1.8.1. 3584 trang. URL: <https://docs.scipy.org/doc>.
- [5] Phan Văn Hạp **and** Lê Đình Thịnh. *Phương pháp tính và các thuật toán*. Nhà xuất bản Giáo dục, 2000. 400 trang.
- [6] Doãn Tam Hòe. *Toán học tính toán*. Đại học Quốc gia Hà Nội, 2009. 240 trang.
- [7] Matplotlib development team. *Matplotlib documentation*. phiên bản 3.5.1. URL: <https://matplotlib.org/3.5.1/tutorials/index.html>.
- [8] SymPy Development Team. *SymPy Documentation*. phiên bản 1.8. 2750 trang. URL: <https://github.com/sympy/sympy/releases>.

