

THỰC HÀNH MÔN TOÁN KINH TẾ

Thời gian : 3 buổi

Công cụ tính : **MATLAB**, Mathematica, Python, Java, C++

Yêu cầu : sinh viên nắm bài trên lớp, có ý tưởng / công thức / sơ đồ khối của các thuật toán

Đánh giá : 10% ĐQT.

1 Giải gần đúng phương trình một biến

```
1 f = @(x) x^3 + 2*x - 1 % khai báo hàm số  $f(x) = x^3 + 2x - 1$ 
2 f(0) %  $\rightarrow -1$ 
3 f(2) %  $\rightarrow 11$ 

4 a = 0;
5 b = 2;
6 for n = 1:5
7     c = (a+b) / 2;
8     if f(c) == 0
9         c
10        break
11    elseif f(a) * f(c) < 0
12        b = c;
13    else
14        a = c;
15    end
16    ss = b - a
17    [n, a, b, ss] %  $n, a_n, b_n, \varepsilon_n$ 
18 end
```

Mã 1: Phương pháp chia đôi

```
1 f = @(x) x^3 - x^2 - 3
2 syms x
3 fplot(f(x), [1, 4])
4 diff(f(x))
5 fplot(diff(f(x)), [1, 4])
6 diff(f(x), 2)
7 fplot(diff(f(x), 2), [1, 4])
```

```

8 f(1)
9 f(4)

10 fplot(abs(diff(f(x), 2)), [1, 4])
11 M = 22

12 syms t
13 df = @(x) subs(diff(f(t)), x)

14 m = min(abs(df(1)), abs(df(4)))

15 x0 = 4 %  $x_{n-1}$ 
16 for n = 1:3
17     n
18     x = vpa(x0 - f(x0) / df(x0), 6) %  $x_n$ 
19     e = vpa(M / 2 / m * (x - x0)^2, 6) %  $\varepsilon_n$ 
20     x0 = x;
21 end

```

Mã 2: Phương pháp Newton

```

1 g = @(x) nthroot(x^2 + 3, 3)

2 syms x

3 fplot(g(x), [1, 4])

4 diff(g(x))
5 fplot(abs(diff(g(x))), [1, 4])

6 q = ...

7 x0 = 2.5 %  $x_{n-1}$ 
8 for n = 1:3
9     n
10    x = g(x0) %  $x_n$ 
11    e = q / (1-q) * abs(x - x0) %  $\varepsilon_n$ 
12    x0 = x;
13 end

```

Mã 3: Phương pháp lặp điểm bất động

2 Nội suy bằng đa thức

```

1 X = [-1, 0, 1, 2]
2 Y = [4, 3, 2, 7]
3 syms x

4 P = 0
5 for i = 1:4
6     L = 1;
7     for j = 1:4
8         if j ~= i
9             L = L * (x - X(j)) / (X(i) - X(j));
10        end
11    end
12    L
13    expand(L)
14    P = P + Y(i) * L;
15 end
16 expand(P)

```

Mã 4: Phương pháp Lagrange

```

1 X = [-1, 0, 1, 2]
2 Y = [4, 3, 2, 7]

3 d = zeros(4, 4);
4 d(1, :) = Y
5 for k = 2:4
6     for i = 1:5-k
7         d(k, i) = d(k-1, i+1) - d(k-1, i);
8     end
9 end
10 d

11 syms t x
12 P = 0;
13 for k = 0:3
14     N = d(k+1, 1) / factorial(k); % d(k+1, 4-k)
15     for i = 0:k-1
16         N = N * (t - i); % t + i
17     end
18     P = P + N;
19 end
20 P = subs(P, t, (x-X(1)) / 1) % X(4)

```

21 `expand(P)`

Mã 5: Phương pháp Newton tiến / lùi

3 Tính gần đúng đạo hàm, tích phân

```

1 X = [2.1, 2.2, 2.3, 2.4, 2.5, 2.6]
2 Y = [-1.71, -1.37, -1.12, -0.92, -0.75, -0.6]
3 h = 0.1

4 % Đạo hàm cấp 1: công thức sai phân tiến
5 for i = 1:5
6     (Y(i+1) - Y(i)) / h
7 end
8 (Y(6) - Y(5)) / h

9 % Đạo hàm cấp 1: công thức 3 điểm
10 for i = 2:5
11     (Y(i+1) - Y(i-1)) / 2 / h
12 end
13 (-Y(3) + 4*Y(2) - 3*Y(1)) / 2 / h
14 (3*Y(6) - 4*Y(5) + Y(4)) / 2 / h

15 % Đạo hàm cấp 2: công thức 3 điểm
16 for i = 2:5
17     (Y(i+1) - 2*Y(i) + Y(i-1)) / h^2
18 end
19 (Y(3) - 2*Y(2) + Y(1)) / h^2
20 (Y(6) - 2*Y(5) + Y(4)) / h^2

```

Mã 6: Tính gần đúng đạo hàm

```

1 X = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
2 Y = [0.69, 0.87, 1.07, 1.29, 1.52, 1.77, 2.03, 2.31, 2.6]

3 % Hình thang
4 I = 0
5 for i = 2:9
6     I = I + (X(i) - X(i-1)) * (Y(i) + Y(i-1)) / 2;
7 end
8 I

9 % Simpson

```

```

10 I = 0
11 for i = 1:4
12     I = I + (X(2*i+1) - X(2*i-1)) * (Y(2*i+1) + 4*Y(2*i) + Y(2*i-1)
13     ) / 6;
14 end
15 I

```

Mã 7: Tính gần đúng tích phân: hàm rời rạc, hình thang, Simpson

```

1 f = @(x) exp(2*x) .* sin(3*x)

2 X = linspace(0, 2, 9)
3 Y = f(X)

4 syms x

5 % Hình thang
6 I = 0
7 for i = 2:9
8     I = I + (X(i) - X(i-1)) * (Y(i) + Y(i-1)) / 2;
9 end
10 I

11 diff(f(x), 2)
12 simplify(diff(f(x), 2))
13 fplot(abs(diff(f(x), 2)), [0, 2])
14 M2 = 700
15 M2 * (2 - 0)^3 / 12 / 8^2

16 % Simpson
17 I = 0
18 for i = 1:4
19     I = I + (X(2*i+1) - X(2*i-1)) * (Y(2*i+1) + 4*Y(2*i) + Y(2*i-1)
20     ) / 6;
21 end
22 I

23 diff(f(x), 4)
24 simplify(diff(f(x), 4))
25 fplot(abs(diff(f(x), 4)), [0, 2])
26 M4 = 4500

```

26 M4 * (2 - 0)^5 / 180 / 8^4

Mã 8: Tính gần đúng tích phân: hàm liên tục, hình thang, Simpson

4 PTVP: bài toán giá trị ban đầu

```

1 f = @(x, y) y - x          % VD2: f = @(x, y) [x*y(1) - y(2), y(1) + y(2) - 1]
2                             % VD3: f = @(x, y) [y(2), y(3), x*y(3) - y(1)]
3 X = [0, 0.2, 0.3, 0.5]    % VD2: X = [1, 1.1, 1.3, 1.5]
4                             % VD3: X = [-1, -0.8, -0.6, -0.5]
5 y = 2                     % VD2: y = [-1, 2]
6                             % VD3: y = [1, 0, -2]
7 for n = 1:3
8     h = X(n+1) - X(n);
9     y = y + h * f(X(n), y)
10 end

```

Mã 9: Phương pháp Euler

```

1 f = @(x, y) y - x          % VD2: f = @(x, y) [x*y(1) - y(2), y(1) + y(2) - 1]
2                             % VD3: f = @(x, y) [y(2), y(3), x*y(3) - y(1)]
3 X = [0, 0.2, 0.3, 0.5]    % VD2: X = [1, 1.1, 1.3, 1.5]
4                             % VD3: X = [-1, -0.8, -0.6, -0.5]
5 y = 2                     % VD2: y = [-1, 2]
6                             % VD3: y = [1, 0, -2]
7 for n = 1:2
8     h = X(n+1) - X(n);
9     k1 = h * f(X(n), y);
10    k2 = h * f(X(n) + h/2, y + k1/2);
11    k3 = h * f(X(n) + h/2, y + k2/2);
12    k4 = h * f(X(n) + h, y + k3);
13    y = y + (k1 + 2*k2 + 2*k3 + k4) / 6
14 end

```

Mã 10: Phương pháp Runge–Kutta RK4

5 Kỹ thuật lập trong đại số tuyến tính

```

1 B = [-0.21, -0.28, 0.05;
2       0.19, 0.01, -0.26;
3       0.39, -0.12, -0.06]

```

```
4 g = [-0.9; 3.8; -2.9]
5 q = norm(B, inf)

6 % Lặp điểm bất động
7 x0 = [0; 2; -1]
8 for k = 1:5
9     x = B*x0 + g
10    ss = q / (1-q) * norm(x-x0, inf)
11    x0 = x;
12 end

13 % Lặp Gauss-Seidel
14 x0 = [0; 2; -1]
15 for k = 1:5
16     x = x0;
17     for i = 1:3
18         x(i) = B(i, :) * x + g(i);
19     end
20     x
21     ss = norm(x-x0, inf)
22     x0 = x;
23 end
```

Mã 11: Hệ điểm bất động: lặp điểm bất động, Gauss–Seidel

```
1 A = [-15.4, 1, 6.3;
2       -4.2, 10.8, 3.3;
3       -2.4, 5.3, 15.9]
4 b = [30; 25; -10]

5 B = zeros(3, 3)
6 g = zeros(3, 1)
7 for i = 1:3
8     for j = 1:3
9         if i ~= j
10            B(i, j) = - A(i, j) / A(i, i);
11        end
12    end
13    g(i) = b(i) / A(i, i);
14 end

15 B
16 g
```

```

17 % Lặp điểm bất động
18 x = [0; 0; 0]
19 for k = 1:5
20     k
21     x = B*x + g
22 end
23 % Lặp Gauss-Seidel
24 x = [0; 0; 0]
25 for k = 1:5
26     k
27     for i = 1:3
28         x(i) = B(i, :) * x + g(i);
29     end
30     x
31 end

```

Mã 12: Hệ chéo trội: lặp điểm bất động, Gauss–Seidel

6 Lý thuyết xấp xỉ

```

1 X = [1, 1.3, 1.7, 2] % VD2: X = [-0.7, 1.7, -4.9, 3.1, -1.3]
2 % VD2: Y = [-2.9, -1.1, -2.9, 1.5, 0.8]
3 Y = [1.8, 1.4, 1.1, 1.5] % VD2: Z = [7.1, 5.8, -3.1, -1, -8.7]
4 syms x % VD2: syms x y
5 cs = [1, x, log(x)] % VD2: [1, x, y]
6 V = zeros(4, 3) % VD2: zeros(5, 3)
7 for i = 1:3
8     V(:, i) = subs(cs(i), X); % VD2: subs(cs(i), {x, y}, {X, Y})
9 end
10 V
11 A = V' * V
12 b = V' * Y' % VD2: V' * Z'
13 c = linsolve(A, b) % các hệ số của P(x), P(x,y)
14 P = vpa(dot(c, cs), 6)
15 P0 = vpa(subs(P, X), 6) % VD2: subs(P, {x, y}, {X, Y})
16 d = vpa(Y - P0, 6) % VD2: Z - P0

```



```
17 e = vpa(norm(d), 6)
```

Mã 13: Phương pháp bình phương nhỏ nhất

7 Giải gần đúng hệ phương trình phi tuyến

```
1 g = @(x) [(x(1)^2 + x(2)^2 + 3) / 8;
2          (-x(1)^2 + 2*x(1)*x(2) + 17) / 10]
3 x0 = [-1; 2.4] % x0 ← x^(k-1)
4 for k = 1:5
5     k
6     x = g(x0) % x ← x^(k)
7     e = norm(x - x0, inf) % e ← ε_k
8     x0 = x;
9 end
```

Mã 14: Phương pháp lặp điểm bất động

```
1 f = @(x) [x(1)^2 + x(2)^2 - 8*x(1) + 3;
2          x(1)^2 - 2*x(1)*x(2) + 10*x(2) - 17]
3 x = sym('x', [2, 1])
4 J = [diff(f(x), x(1)), diff(f(x), x(2))]
5 X0 = [-1; 2.4]
6 for k = 1:5
7     k
8     X = vpa(X0 - subs(J, x, X0)^-1 * f(X0), 6)
9     e = vpa(norm(X - X0, inf), 6)
10    X0 = X;
11 end
```

Mã 15: Phương pháp Newton

8 Phương trình đạo hàm riêng

```
1 f = @(x, y) x * exp(y)
2 % Chia lưới
3 a = 0; b = 1; c = 1; d = 1.4;
4 n = 5; m = 4;
5 h = (b-a)/n, k = (d-c)/m
```

```

6 % Các phép tính lặp lại nhiều lần
7 aL = "A(r, r-1) = 1;" % hệ số của  $u_{i-1,j}$ 
8 aR = "A(r, r+1) = 1;" % . . . . .  $u_{i+1,j}$ 
9 aD = "A(r, r-n+1) = (h/k)^2;" % . . . . .  $u_{i,j-1}$ 
10 aU = "A(r, r+n-1) = (h/k)^2;" % . . . . .  $u_{i,j+1}$ 
11 lh = "B(r) = h^2 * f(a+i*h, b+j*k)" %  $h^2 f(x_i, y_j)$ , chưa có dấu ; để còn
    nối phép tính
12 sL = "- g(a+(i-1)*h, b+j*k)" %  $-g(x_{i-1}, y_j)$ 
13 sR = "- g(a+(i+1)*h, b+j*k)" %  $-g(x_{i+1}, y_j)$ 
14 sD = "- (h/k)^2 * g(a+i*h, b+(j-1)*k)" %  $-(\frac{h}{k})^2 g(x_i, y_{j-1})$ 
15 sU = "- (h/k)^2 * g(a+i*h, b+(j+1)*k)" %  $-(\frac{h}{k})^2 g(x_i, y_{j+1})$ 

16 % Lập hệ phương trình
17 A = zeros((n-1)*(m-1));
18 B = zeros((n-1)*(m-1), 1);
19 for i = 1:n-1 % duyệt từng cột trên hình
20     for j = 1:m-1
21         r = (j-1)*(n-1) + i; % phương trình thứ r
22         A(r, r) = -2 * ((h/k)^2 + 1); % hệ số của  $u_{ij}$ 
23         if i == 1 && j == 1 % góc dưới trái
24             eval(aR + aU + lh + sL + sD + ";")
25         end
26         if i == 1 && j == m-1 % góc trên trái
27             eval(aR + aD + lh + sL + sU + ";")
28         end
29         if i == n-1 && j == 1 % góc dưới phải
30             eval(aL + aU + lh + sR + sD + ";")
31         end
32         if i == n-1 && j == m-1 % góc trên phải
33             eval(aL + aD + lh + sR + sU + ";")
34         end
35         if i == 1 && 1 < j && j < m-1 % cạnh trái
36             eval(aR + aD + aU + lh + sL + ";")
37         end
38         if i == n-1 && 1 < j && j < m-1 % cạnh phải
39             eval(aL + aD + aU + lh + sR + ";")
40         end
41         if j == 1 && 1 < i && i < n-1 % cạnh dưới
42             eval(aL + aR + aU + lh + sD + ";")
43         end
44         if j == m-1 && 1 < i && i < n-1 % cạnh trên

```

```

45         eval(aL + aR + aD + lh + sU + ";")
46     end
47     if 1 < i && i < n-1 && 1 < j && j < m-1 % giữa
48         eval(aL + aR + aD + aU + lh + ";")
49     end
50 end
51 end
52 A
53 B
54 % Giải hệ và hiển thị kết quả
55 u = linsolve(A, B)
56 sol = flipud(reshape(u, n-1, m-1)')
57 % So sánh với nghiệm đúng
58 u = @(x, y) x * exp(y)
59 U = zeros(n-1, m-1)
60 for i = 1:n-1
61     for j = 1:m-1
62         x = a + i*h; y = b + j*k;
63         U(i, j) = u(x, y);
64     end
65 end
66 U
67 flip(U')
68 function val = g(x, y) % khai báo trước ở ô cuối cùng của sổ tay
69 if x == 0
70     val = 0;
71 end
72 if x == 1
73     val = exp(y);
74 end
75 if y == 1
76     val = exp(1) * x;
77 end
78 if y == 1.4
79     val = exp(1.4) * x;
80 end

```

81 `end`

Mã 16: Phương trình elliptic

9 Tối ưu

```

1 f = @(x) x.^2/10 - 2*sin(x)
2 syms t
3 df = @(x) subs(diff(f(t)), x)
4 d2f = @(x) subs(diff(f(t), 2), x)
5 x = 2.5
6 f(x)
7 for n = 1:3
8     x = vpa(x - df(x) / d2f(x), 6)
9     vpa(f(x), 6)
10 end

```

Mã 17: Tối ưu không ràng buộc một biến: phương pháp Newton

```

1 f = @(x) 100*(x(2) - x(1)^2)^2 + (1 - x(1))^2
2 x = sym('x', [2, 1])
3 g = [diff(f(x), x(1)); diff(f(x), x(2))]
4 H = [diff(g, x(1)), diff(g, x(2))]
5 X = [-1.2; 1]
6 f(X)
7 for k = 1:5
8     X = vpa(X - (subs(H, x, X))^-1 * subs(g, x, X), 6)
9     vpa(f(X), 6)
10 end

```

Mã 18: Tối ưu không ràng buộc nhiều biến: phương pháp Newton

```

1 c = [-1; -1/3]
2 A = [1, 1; 1, 1/4; 1, -1; -1/4, -1; -1, -1; -1, 1]
3 b = [2; 1; 2; 1; -1; 2]
4 Ae = [1, 1/4]
5 be = [1/2]
6 l = [-1; -0.5]

```

```
7 u = [1.5; 1.25]
8 [x, f] = linprog(c, A, b, Ae, be, l, u)
```

Mã 19: Quy hoạch tuyến tính