Toán kinh tế: MATLAB buổi 5/15

6 Giải gần đúng phương trình vi phân

6.1 Phương pháp Picard

```
f = @(x, y) y - x   % VD2: f = @(x, y) [x*y(1) - y(2); y(1) + y(2) - 1]
                    % VD3: f = @(x, y) [y(2); y(3); x*y(3) - y(1)]

f(1, 2)             % VD2: f(1, [2; 3])
                    % VD3: f(0, [1; 2; 3])

x0 = 0              % VD2: x0 = 1
                    % VD3: x0 = -1

y0 = 2              % VD2: y0 = [-1; 2]
                    % VD3: y0 = [1; 0; -2]

syms x t

y = y0
for n = 1:3
    y = expand(y0 + int(f(t, subs(y, t)), x0, x))
end
```

Mã 28: Phương pháp Picard

6.2 Phương pháp Taylor

```
f = @(x, y) y - x
x0 = 0
y0 = 2

syms x y(x)
n = 3

d = y(x)
P = y0
for k = 1:n
    d = subs(diff(d), diff(y), f(x, y))
    d0 = subs(d, [x, y(x)], [x0, y0])
    P = P + d0 / factorial(k) * (x - x0)^k
end
```

Mã 29: Phương pháp Taylor: phương trình vi phân cấp một

```
f = @(x, y, z) [x*y - z, y + z - 1]
x0 = 1
y0 = -1
z0 = 2

syms y(x) z(x)
n = 3

```

```
d = y(x)   % z(x)
P = y0     % z0
for k = 1:n
    d = subs(diff(d), [diff(y), diff(z)], f(x, y, z))
    d = expand(d)
    d0 = subs(d, [x, y(x), z(x)], [x0, y0, z0])
    P = P + d0 / factorial(k) * (x - x0)^k
end
```

Mã 30: Phương pháp Taylor: hệ phương trình vi phân cấp một

```
x0 = -1
y0 = 1
y1 = 0
y2 = -2

syms y(x)
n = 5

d = y(x)
P = y0
for k = 1:n
    d = subs(diff(d), diff(y, 3), x * diff(y, 2) - y);
    d = expand(d)
    d0 = subs(d, [x, y(x), diff(y(x)), diff(y(x), 2)], [x0, y0, y1, y2])
    P = P + d0 / factorial(k) * (x - x0)^k
end
```

Mã 31: Phương pháp Taylor: phương trình vi phân cấp cao

6.3 Phương pháp Euler

```
f = @(x, y) y - x         % VD2: f = @(x, y) [x*y(1) - y(2), y(1) + y(2) - 1]
                          % VD3: f = @(x, y) [y(2), y(3), x*y(3) - y(1)]

X = [0, 0.2, 0.3, 0.5]    % VD2: X = [1, 1.1, 1.3, 1.5]
                          % VD3: X = [-1, -0.8, -0.6, -0.5]

y = 2                     % VD2: y = [-1, 2]
                          % VD3: y = [1, 0, -2]
for n = 1:3
    h = X(n+1) - X(n);
    y = y + h * f(X(n), y)
end
```

Mã 32: Phương pháp Euler

6.4 Phương pháp Runge-Kutta RK4

```
f = @(x, y) y - x         % VD2: f = @(x, y) [x*y(1) - y(2), y(1) + y(2) - 1]
                          % VD3: f = @(x, y) [y(2), y(3), x*y(3) - y(1)]

X = [0, 0.2, 0.3, 0.5]    % VD2: X = [1, 1.1, 1.3, 1.5]
```

```
                              % VD3: X = [-1, -0.8, -0.6, -0.5]

y = 2                         % VD2: y = [-1, 2]
                              % VD3: y = [1, 0, -2]
for n = 1:2
    h = X(n+1) - X(n);
    k1 = h * f(X(n), y);
    k2 = h * f(X(n) + h/2, y + k1/2);
    k3 = h * f(X(n) + h/2, y + k2/2);
    k4 = h * f(X(n) + h, y + k3);
    y = y + (k1 + 2*k2 + 2*k3 + k4) / 6
end
```

Mã 33: Phương pháp RK4