

Bài 4:

## TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH

Nhóm:

TT	Họ và tên	MSSV	Lớp	Ghi chú
1	Đỗ Minh Chương	21207126	21DTV_CLC3	

### Bài 1

Sử dụng hàm syms để tạo phương trình  $fx = x^2 - 2\sin(x) + 1/2$  dưới dạng symbol trong Matlab.

Dùng lệnh `fxi = matlabFunction(fx)` để chuyển phương trình sang dạng hàm. Tính  $fxi(2)$ .

Command Window

```
>> syms x;
>> fx = x^2 - 2*sin(x) + 1/2;
>> fxi = matlabFunction(fx);
>> fxi(2)

ans =

    2.6814
```

### Bài 2

Viết hàm (function) trong Matlab để thực hiện các phương pháp chia đôi, lặp, tiếp tuyến (Newton) và đây cung với định dạng tín hiệu vào và ra như sau:

**Phương pháp chia đôi:** `function [c,k] = ChiaDoi(fx, a, b, epsilon)`

```
function [c, k] = ChiaDoi(fx, a, b, epsilon)
    f = str2func(['@(x) ' fx]);
    if f(a) * f(b) > 0
        return;
    end
```

```
k = 0;
while abs(a - b) >= epsilon && k < 500
    c = (a + b) / 2;
    if f(a) * f(c) < 0
        b = c;
    else
        a = c;
    end
    k = k + 1;
end
end
```

**Phương pháp lặp:** function [x,k] = LapDon(fx, gx, a, b, epsilon)

```
function [x, k] = LapDon(fx, gx, a, b, epsilon)
    f = str2func(['@(x) ' fx]);
    g = str2func(['@(x) ' gx]);

    k = 0;
    c = (a + b) / 2;

    if f(a) * f(c) < 0
        x = a;
    else
        x = b;
    end

    while(1)
        y = g(x);

        if abs(y - x) < epsilon || k > 500
            break;
        end

        x = y;
        k = k + 1;
    end
end
```

**Phương pháp tiếp tuyến:** `function [x,k]=TiepTuyen(fx, a, b, epsilon)`

```
function [x, k] = TiepTuyen(fx, a, b, epsilon)
    f = str2func(['@(x) ' fx]);
    syms x_sym; % Define a symbolic variable

    k = 0;
    c = (a + b) / 2;

    if f(a) * f(c) < 0
        x = a;
    else
        x = b;
    end

    while(1)

        df = diff(f(x_sym));

        df_func = matlabFunction(df, 'Vars', x_sym);

        z = f(x) / df_func(x);
        y = x - z;

        if abs(y - x) < epsilon || k > 500
            break;
        end

        x = y;
        k = k + 1;
    end
end
```

**Phương pháp dây cung:** `function [x,k] = DayCung(fx, a, b, epsilon)`

```
function [x, k] = DayCung(fx, a, b, epsilon)
    f = str2func(['@(x) ' fx]);
    if f(a) * f(b) > 0
        return;
    end
    k = 0;
    while abs(a - b) >= epsilon && k < 500
        x = (a * f(b) - b * f(a)) / (f(b) - f(a));
        if f(x) * f(a) < 0
            b = x;
        else
            a = x;
        end
        k = k + 1;
    end
end
```

a. Tìm nghiệm gần đúng của phương trình  $x - \sin x = 0.25$  với khoảng phân li nghiệm  $[-1, 2]$  và sai số không lớn hơn 0.005. Cho biết phương pháp nào có số lần lặp nhiều nhất và giải thích:

```
Command Window
>> fx = 'x - sin(x) - 0.25';
gx = 'sin(x) + 0.25';

% ChiaDoi
[c, n] = ChiaDoi(fx, -1, 2, 0.005);
disp(['Nghiem (ChiaDoi): ' num2str(c)]);
disp(['So Lan Lap (ChiaDoi): ' num2str(n)]);

% LapDon
[c1, n1] = LapDon(fx, gx, -1, 2, 0.005);
disp(['Nghiem (LapDon): ' num2str(c1)]);
disp(['So Lan Lap (LapDon): ' num2str(n1)]);

% TiepTuyen
[c2, n2] = TiepTuyen(fx, -1, 2, 0.005);
disp(['Nghiem (TiepTuyen): ' num2str(c2)]);
disp(['So Lan Lap (TiepTuyen): ' num2str(n2)]);

% DayCung
[c3, n3] = DayCung(fx, -1, 2, 0.005);
disp(['Nghiem (DayCung): ' num2str(c3)]);
disp(['So Lan Lap (DayCung): ' num2str(n3)]);

Nghiem (ChiaDoi): 1.1709
So Lan Lap (ChiaDoi): 10
Nghiem (LapDon): 1.1665
So Lan Lap (LapDon): 2
Nghiem (TiepTuyen): 1.172
So Lan Lap (TiepTuyen): 3
Nghiem (DayCung): 1.1712
So Lan Lap (DayCung): 13
```

Phương pháp dây cung sẽ lặp nhiều nhất vì hàm  $f(x)$  không hội tụ trong khoảng  $[-1, 2]$

b. Lập lại câu a với khoảng phân li nghiệm  $[1, 2]$ . Cho biết phương pháp nào có số lần lặp ít nhất và giải thích.

## Command Window

```
>> fx = 'x - sin(x) - 0.25';  
gx = 'sin(x) + 0.25';  
  
% ChiaDoi  
[a, b] = deal(1, 2);  
[c, n] = ChiaDoi(fx, a, b, 0.005);  
disp(['Nghiem (ChiaDoi): ' num2str(c)]);  
disp(['So Lan Lap (ChiaDoi): ' num2str(n)]);  
  
% LapDon  
[c1, n1] = LapDon(fx, gx, a, b, 0.005);  
disp(['Nghiem (LapDon): ' num2str(c1)]);  
disp(['So Lan Lap (LapDon): ' num2str(n1)]);  
  
% TiepTuyen  
[c2, n2] = TiepTuyen(fx, a, b, 0.005);  
disp(['Nghiem (TiepTuyen): ' num2str(c2)]);  
disp(['So Lan Lap (TiepTuyen): ' num2str(n2)]);  
  
% DayCung  
[c3, n3] = DayCung(fx, a, b, 0.005);  
disp(['Nghiem (DayCung): ' num2str(c3)]);  
disp(['So Lan Lap (DayCung): ' num2str(n3)]);  
Nghiem (ChiaDoi): 1.168  
So Lan Lap (ChiaDoi): 8  
Nghiem (LapDon): 1.1658  
So Lan Lap (LapDon): 4  
Nghiem (TiepTuyen): 1.1718  
So Lan Lap (TiepTuyen): 2  
Nghiem (DayCung): 1.1712  
So Lan Lap (DayCung): 10
```

Phương pháp tiếp tuyến sẽ có số lần lặp ít nhất và hội tụ nhanh nhất vì sử dụng thông tin đạo hàm giúp xác định hướng và tốc độ hội tụ tối ưu.

c. Dùng phương pháp chia đôi, dây cung, và tiếp tuyến để tính gần đúng nghiệm của phương trình  $x^2 - \sin \pi x = 0$  với sai số nhỏ hơn  $5 \cdot 10^{-5}$  và khoảng phân li nghiệm lần lượt là  $[-0.5, 0.5]$  và  $[-0.05, 0.5]$ . Giải thích sự khác biệt giữa 2 trường hợp

Phương pháp chia đôi:

```
Command Window
>> fx = 'x^2 - sin(pi*x)';
>> % Khoảng phân li nghiệm [-0.5, 0.5]
[a, b] = deal(-0.5, 0.5);
[c1, n1] = ChiaDoi(fx, a, b, 5e-5);
disp(['ChiaDoi - Nghiem: ' num2str(c1)]);
disp(['ChiaDoi - So Lan Lap: ' num2str(n1)]);
ChiaDoi - Nghiem: 0.49997
ChiaDoi - So Lan Lap: 15
>>
>> % Khoảng phân li nghiệm [-0.05, 0.5]
[a, b] = deal(-0.05, 0.5);
[c2, n2] = ChiaDoi(fx, a, b, 5e-5);
disp(['ChiaDoi - Nghiem: ' num2str(c2)]);
disp(['ChiaDoi - So Lan Lap: ' num2str(n2)]);
ChiaDoi - Nghiem: -1.5259e-05
ChiaDoi - So Lan Lap: 14
fx >> |
```

Phương pháp dây cung:

```
Command Window
>> fx = 'x^2 - sin(pi*x)';
>>
>> [a, b] = deal(-0.5, 0.5);
[c3, n3] = DayCung(fx, a, b, 5e-5);
disp(['DayCung - Nghiem: ' num2str(c3)]);
disp(['DayCung - So Lan Lap: ' num2str(n3)]);
DayCung - Nghiem: 7.3171e-20
DayCung - So Lan Lap: 10
>>
>>
>> [a, b] = deal(-0.05, 0.5);
[c4, n4] = DayCung(fx, a, b, 5e-5);
disp(['DayCung - Nghiem: ' num2str(c4)]);
disp(['DayCung - So Lan Lap: ' num2str(n4)]);
DayCung - Nghiem: 2.4983e-19
DayCung - So Lan Lap: 10
```

Phương pháp tiếp tuyến:

```

Command Window

>> fx = 'x^2 - sin(pi*x)';
>>
>> [a, b] = deal(-0.5, 0.5);
[c5, n5] = TiepTuyen(fx, a, b, 5e-5);
disp(['TiepTuyen - Nghiem: ' num2str(c5)]);
disp(['TiepTuyen - So Lan Lap: ' num2str(n5)]);
TiepTuyen - Nghiem: 0.78724
TiepTuyen - So Lan Lap: 4
>>
>> [a, b] = deal(-0.05, 0.5);
[c6, n6] = TiepTuyen(fx, a, b, 5e-5);
disp(['TiepTuyen - Nghiem: ' num2str(c6)]);
disp(['TiepTuyen - So Lan Lap: ' num2str(n6)]);
TiepTuyen - Nghiem: -4.5334e-08
TiepTuyen - So Lan Lap: 2
fx >> |

```

### ***Giải thích sự khác biệt:***

Sự khác biệt giữa hai trường hợp phát sinh do đặc điểm của hàm số trong khoảng phân li. Khoảng phân li  $[-0.05, 0.5]$  có kích thước nhỏ hơn và chứa nghiệm, do đó các phương pháp sẽ hội tụ nhanh hơn. Ngược lại, trong khoảng phân li lớn hơn  $[-0.5, 0.5]$ , các phương pháp có thể phải thực hiện nhiều lần lặp hơn để đạt được sai số mong muốn.