

# KỸ THUẬT LẬP TRÌNH

Giảng viên: NGUYỄN THỊ THANH HUYỀN

Bùi Khương Duy 20195864

Vũ Văn Xứng 20195943

Nhóm sinh viên: Nhóm 25 - Lớp 125007 - học kỳ 20202



**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY



**VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC**  
School of Applied Mathematics and Informatics

# Mục lục

<b>I</b>	<b>Mở đầu</b>	<b>2</b>
<b>II</b>	<b>Cơ sở lý thuyết</b>	<b>3</b>
1	Nội dung phương pháp . . . . .	3
2	Sự hội tụ của phương pháp . . . . .	4
3	Sai số . . . . .	5
<b>III</b>	<b>Thiết kế chương trình</b>	<b>7</b>
1	Tiền xử lý chương trình . . . . .	7
2	Thiết kế chương trình bằng phương pháp tinh chỉnh từng bước . . . . .	8
<b>IV</b>	<b>Kiểm tra chương trình</b>	<b>34</b>
<b>V</b>	<b>Lời kết</b>	<b>43</b>



## Mở đầu

Bài toán Tìm nghiệm thực gần đúng của phương trình  $f(x) = 0$  (đại số hay siêu việt) là một trong những bài toán mà ta hay gặp phải trong quá trình tính toán kỹ thuật.

Sau khi tìm hiểu và nghiên cứu các phương pháp giải bài toán đã học trong môn Giải tích số, chúng em xin được trình bày nội dung Phương pháp dây cung. Bài báo cáo dưới đây ứng dụng những kiến thức đã học trong môn Kỹ thuật lập trình để trình bày phương pháp thiết kế thuật toán và chương trình tìm nghiệm xấp xỉ phương trình đa thức bằng phương pháp dây cung.

Chúng em xin cảm ơn cô Nguyễn Thị Thanh Huyền đã dạy lớp chúng em và những kiến thức cô đã truyền tải thông qua học phần này. Bài báo cáo được thực hiện trong khoảng thời gian hạn chế nên không thể tránh khỏi sai sót nên rất mong nhận được sự đóng góp của cô và các bạn để toàn nhóm có thể hoàn thiện bài báo cáo một cách hoàn chỉnh nhất.

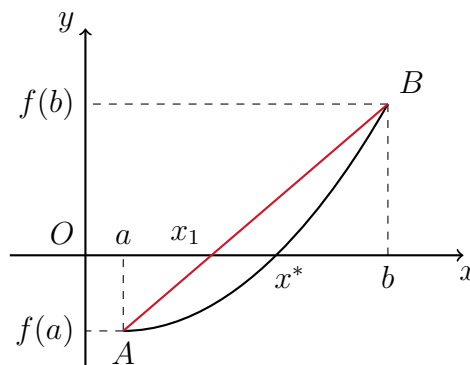
# Cơ sở lý thuyết

## 1 Nội dung phương pháp

Ý tưởng của phương pháp dây cung:

Giả sử  $(a, b)$  là khoảng phân li nghiệm của phương trình thì  $f(x)$  là hàm số liên tục và có đạo hàm liên tục trên  $[a, b]$ . Sau đó sẽ khảo sát đồ thị hàm số  $y = f(x)$  trên đoạn  $[a, b]$  theo các bước sau:

- B1:** Trên mặt phẳng tọa độ, ta có hai điểm  $A(a, f(a))$  và  $B(b, f(b))$ .
- B2:** Thực hiện thay thế đường cong  $y = f(x)$  trên  $[a, b]$  bằng dây cung nối hai đầu mút  $A$  và  $B$
- B3:** Tìm giao điểm của dây cung với trục hoành  
Tiếp tục các bước trên ta được dãy lặp nghiệm cho phương pháp như sau:



### Phương pháp dây cung

Giả thiết  $(a, b)$  là khoảng phân li nghiệm,  $f(x)$  liên tục và có đạo hàm liên tục đến cấp hai,  $f'(x)$  và  $f''(x)$  không đổi dấu. Ta có công thức lặp cho phương pháp dây cung:

$$x_n = x_{n-1} - f(x_{n-1}) \cdot \frac{d - x_{n-1}}{f(d) - f(x_{n-1})}$$

Với  $(d, f(d))$  là điểm **Fourier**

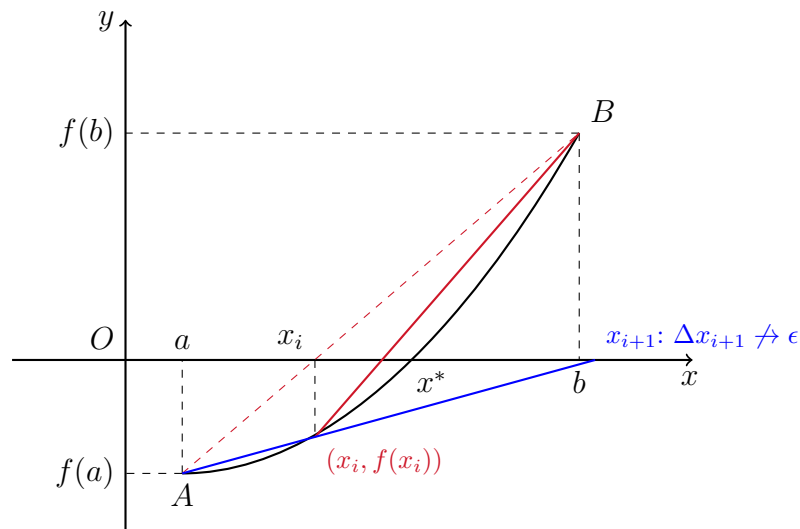
## 2 Sự hội tụ của phương pháp

### a) Điểm Fourier

#### Điểm Fourier

Gọi  $M(d, f(d))$  là điểm Fourier nếu  $f(d) \cdot f''(d) > 0$

Việc xác định điểm **Fourier** là vô cùng quan trọng. Mặc dù khi thay tùy ý  $a, b$  làm  $x_0, d$  ta vẫn có thu được công thức lặp trên nhưng khi tính toán, kết quả lặp sẽ không hội tụ về  $x^*$ , hình vẽ dưới đây cho ta cách nhìn trực quan về điều đó.



### b) Định lý về sự hội tụ

#### Định lý

Giả sử  $(a, b)$  là khoảng phân ly nghiệm  $x^*$  của phương trình  $f(x) = 0$ .

- Hàm số  $f(x)$  liên tục và có đạo hàm liên tục đến cấp 2 trên  $[a, b]$ .
- $f'(x), f''(x)$  giữ nguyên dấu trên đoạn đó.
- Đồng thời xác định được điểm  $M(d, f(d))$  - **Fourier** và  $M_0(x_0, f(x_0))$  thì: Dãy  $\{x_n\}$  xác định theo **Công thức lặp** sẽ hội tụ đơn điệu tới  $x^*$

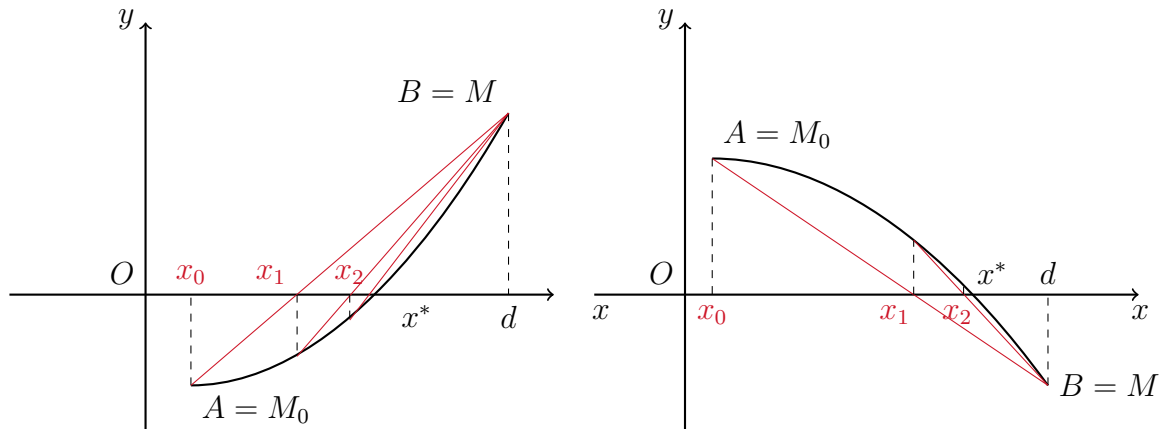
$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} \left[ x_{n-1} - \frac{d - x_{n-1}}{f(d) - f(x_{n-1})} \cdot f(x_{n-1}) \right] = x^*$$

Cụ thể: Khi  $n \rightarrow \infty$ , nếu

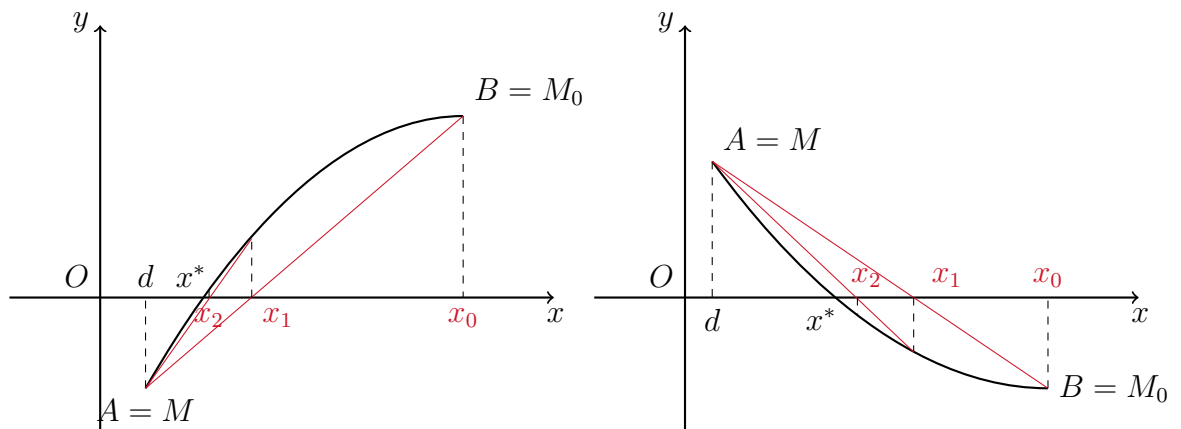
- $f'(x) \cdot f''(x) > 0$  thì dãy  $\{x_n\}$  đơn điệu tăng đến  $x^*$
- $f'(x) \cdot f''(x) < 0$  thì dãy  $\{x_n\}$  đơn điệu giảm về  $x^*$

### Minh họa định lý:

- Với  $f'(x).f''(x) > 0$  :  $\{x_n\}$  đơn điệu tăng về  $x^*$



- Với  $f'(x).f''(x) < 0$  :  $\{x_n\}$  đơn điệu giảm về  $x^*$



## 3 Sai số

**Ý nghĩa :** Sai số tốt sẽ cho bài toán kết quả đúng đắn mà vẫn đảm bảo cho chương trình thực thi hiệu quả.

a) Công thức 1 - Công thức sai số mục tiêu:

### Định lý - Công thức mục tiêu

Giả sử  $(a, b)$  là khoảng phân ly nghiệm  $x^*$  của phương trình  $f(x)$ ,  $x_i \in (a, b)$  là nghiệm gần đúng tìm được

Nếu  $|f'(x)| \geq m > 0 \quad \forall x \in [a, b]$ ,  $m = \text{const}$  (hay  $m = \min_{x \in [a, b]} |f'(x)|$ )

thì

$$|x_i - x^*| \leq \frac{|f(x_i)|}{m} \quad (1)$$

### Chứng minh Công thức (1)

- Xét  $f(x_i) = f(x_i) - f(x^*)$   
Áp dụng **Định lý Lagrange** ta có

$$f(x_i) = f(x_i) - f(x^*) = f'(c)(x_i - x^*) \text{ với } c \text{ nằm giữa } x^* \text{ và } x_i$$

- Như vậy,  $c \in (a, b) \Rightarrow |f'(c)| \geq m > 0$   
Và

$$|f(x_i)| = |f'(c)||x_i - x^*| \geq m|x_i - x^*|$$

Hay

$$|x_i - x^*| \leq \frac{|f(x_i)|}{m}$$

b) Công thức 2 - Sai số theo hai bước lặp liên tiếp:

#### Công thức 2 - Đánh giá sai số theo hai bước lặp liên tiếp

Với  $(a, b)$  là khoảng phân ly nghiệm  $x^*$  của  $f(x)$

Đặt  $M = \max_{x \in [a, b]} |f'(x)|$ ,  $m = \min_{x \in [a, b]} |f'(x)|$ , ta có công thức:

$$|x_{i+1} - x^*| \leq \frac{M - m}{m} |x_{i+1} - x_i| \quad (2)$$

### Chứng minh công thức (2)

- Đặt  $M = \max_{x \in [a, b]} |f'(x)|$ ,  $m = \min_{x \in [a, b]} |f'(x)|$
- Từ Công thức lặp:

$$x_i = x_{i-1} - \frac{d - x_{i-1}}{f(d) - f(x_{i-1})} \cdot f(x_{i-1})$$

và  $f(x^*) = 0$  nên

$$[f(d) - f(x_{i-1})](x_i - x_{i-1}) = [f(x^*) - f(x_{i-1})](d - x_{i-1})$$

- Áp dụng **Định lý Lagrange** vào hai ngoặc vuông:

$$f'(c_1)(d - x_{i-1})(x_i - x_{i-1}) = f'(c_2)(x^* - x_{i-1})(d - x_{i-1})$$

với  $c_1, c_2$  là các trị trung gian đều thuộc  $(a, b)$

# Thiết kế chương trình

## 1 Tiền xử lý chương trình

- Với yêu cầu điều khiển chương trình bằng menu, chúng em sử dụng dạng menu popup với dòng sáng đỏ chạy lên xuống, mỗi chức năng biểu thị tại một hàng và người dùng sẽ dùng nút mũi lên xuống để điều khiển chương trình.
- Về vấn đề nhập và tính giá trị hàm số, lưu các giá trị đặc trưng (bậc của hàm số, đầu mút các khoảng phân ly nghiệm)
- Để lưu trữ một đa thức hay đạo hàm của nó, ta cần lưu hệ số và số mũ. Như vậy cần một kiểu dữ liệu dạng mảng hoặc danh sách, trong đó khả dụng là:
  - i) **Array**: Mảng là một cấu trúc dữ liệu khá hiệu quả, cho phép truy cập phần tử nhanh, dễ dàng tính toán. Tuy nhiên kích thước có hạn nên sẽ không phù hợp với đa thức hợp bởi rất nhiều đơn thức.
  - ii) **Linked list**: Danh sách móc nối là một cấu trúc dữ liệu ổn định trong việc thêm bớt. Tuy nhiên, cấu trúc này không phù hợp với bài toán vì để lưu một tập hợp dữ liệu  $n$  phần tử cần  $2n$  ô nhớ (một để dữ liệu, một để chứa con trỏ tới ô khác), thực tế bài toán sẽ cần đến  $3n$  ô nhớ. Hơn nữa, trong bài toán cần tính giá trị hàm số tại một điểm rất nhiều, do đó cần duyệt danh sách nhiều lần. Khi đó tốc độ của danh sách móc nối là chậm.
  - iii) **Tree**: Cấu trúc cây sẽ không phù hợp với bài toán vì mục tiêu bài toán không phải là thêm bớt phần tử hay duyệt tìm đơn thức.

Như vậy chúng em sẽ dùng cấu trúc mảng, và bài toán yêu cầu độ chính xác cao nên ta sẽ dùng kiểu dữ liệu double. Để thuận tiện, chúng em tạo 2 kiểu dữ liệu riêng là kiểu dữ liệu **struct** để thuận tiện cho việc lưu hàm số và khoảng phân ly.

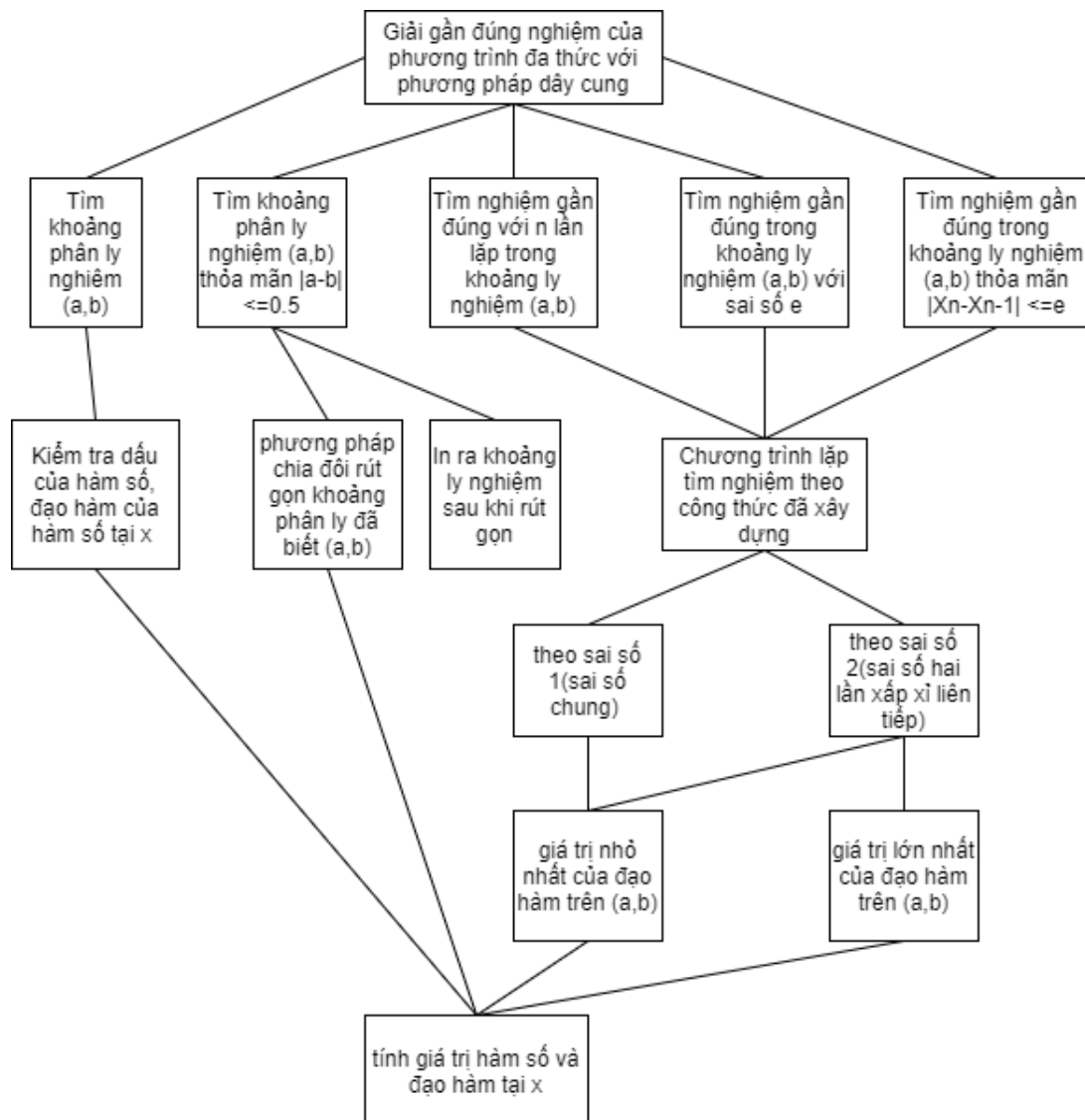
- Với yêu cầu thao tác với dữ liệu từ file văn bản chúng em sử dụng format dữ liệu theo cột hàng, cột: trên mỗi hàng phần tử đầu tiên là bậc ( $i$ ) của hạng tử  $x^i$ , phần tử thứ hai là số hạng của  $x^i$  tương ứng.
- Với yêu cầu in quá trình thực hiện chương trình ra file văn bản chúng em sẽ dùng lệnh **printf()** và **fprintf()** để thực hiện thao tác đó với format dạng:

```
+ - + - + - - - - - - - - - - - - - - - + - + - +
|
+ - + - + - - - - - - - - - - - - - - - + - + - +
|
+ - + - + - - - - - - - - - - - - - - - + - + - +
|
+ - + - + - - - - - - - - - - - - - - - + - + - +
```



## 2 Thiết kế chương trình bằng phương pháp tinh chỉnh từng bước

Bước 0:



Bước 1:

Cấu trúc dữ liệu:

Mỗi hạng tử của đa thức gồm được biểu diễn bởi cấu trúc dữ liệu struct gồm hai thành phần:

- Bậc: degree
- Hệ số: coefficient.

Mỗi khoảng phân ly nghiệm của đa thức được biểu diễn bởi cấu trúc dữ liệu struct gồm 2 thành phần:

- Cận trái: a
- Cận phải b

Mảng tối đa 15 phần tử chứa lưu trữ các hạng tử đa thức.

Mảng tối đa 15 phần tử lưu trữ các khoảng phân ly nghiệm của đa thức.

**Input:** đa thức, khoảng tìm nghiệm  $(-1000.5; 999.5)$

**Output:** khoảng phân ly, nghiệm gần đúng, các bước trung gian tìm nghiệm

*Ý tưởng thuật toán:*

- Phần 1) Tìm khoảng phân ly nghiệm  $(a, b)$  của đa thức trong miền giá trị  
*Input:* khoảng tìm nghiệm  $(left, right)$ , đa thức  
*Output:* mảng *phanly*[ ] có kiểu dữ liệu ICS lưu khoảng phân ly và biến nguyên count lưu số lượng khoảng tìm được  
*Ý tưởng thuật toán:*
  - + ) xét dấu của tích các giá trị hàm số tại các điểm liên tiếp các nhau 1 khoảng  $< 1$ .
  - + ) Nếu tích này  $< 0$  thì khoảng 2 điểm đang xét có ít nhất 1 khoảng ly nghiệm.
- Phần 2) Tìm khoảng phân ly nghiệm được rút gọn từ khoảng ly nghiệm  $(a, b)$  bằng phương pháp chia đôi  
*Input:* khoảng phân ly nghiệm  $(a, b)$  bất kì, độ lệch  $m$  mong muốn của hiệu  $|a - b|$   
*Output:* Mảng chứa khoảng phân ly  $(a, b)$  được rút gọn thỏa mãn  $|a - b| \leq m$   
*Ý tưởng thuật toán:*
  - + ) xác định dấu của  $f(a)$
  - + ) while  $|b - a| > m$ {  
kiểm tra dấu của  $f(c) : c = \frac{(a + b)}{2}$   
kiểm tra tính cùng phía, khác phía của  $f(a)$  và  $f(c)$  để thực hiện gán giá trị các biến  
}
- Phần 3) Tìm nghiệm gần đúng  $x_n$  bằng phương pháp dây cung trong khoảng phân ly nghiệm  $(a, b)$  đúng với  $n$  lần lặp.  
*Input:* số lần lặp  $n$ , khoảng ly nghiệm  $(a, b)$   
*Output:* nghiệm gần đúng sau  $n$  lần lặp trong khoảng  $(a, b)$   
*Ý tưởng thuật toán:*
  - + ) tìm giá trị nhỏ nhất và giá trị lớn nhất của đạo hàm trên khoảng  $(a, b)$ .
  - + ) xác định  $f(a), f(b)$
  - + ) xác định sai số đầu  $x$  với  $x = a - f(a) * (a - b) / (f(a) - f(b))$
  - + ) nếu  $f(x)$  và  $f(a)$  khác phía: gán  $b = a, f(b) = f(a)$
  - + ) while  $i < 1$ : Lặp với công thức

$$x_n = x_{n-1} - f(x_{n-1}) \cdot \frac{b - x_{n-1}}{f(b) - f(x_{n-1})}$$

Tìm các sai số và giá trị của hàm số theo công thức sai số (1)

Tìm các sai số và giá trị của hàm số theo công thức sai số (2)

- Phần 4) Tìm nghiệm gần đúng  $x_n$  bằng phương pháp dây cung trong khoảng phân ly nghiệm  $(a, b)$  với sai số  $eps$ .  
*Input:* sai số  $eps$  mong muốn, khoảng ly nghiệm  $(a, b)$   
*Output:* nghiệm gần đúng trong khoảng  $(a, b)$   
*Ý tưởng thuật toán:*
  - + ) tìm giá trị nhỏ nhất và giá trị lớn nhất của đạo hàm trên khoảng  $(a, b)$ .

- +) xác định  $f(a), f(b)$
- +) xác định sai số đầu  $x$  với  $x = a - f(a) * (a - b) / (f(a) - f(b))$
- +) nếu  $f(x)$  và  $f(a)$  khác phía, gán  $b = a, f(b) = f(a)$
- +) vòng lặp while với điều kiện:  $sai\_so > eps$  và số vòng lặp đã thực hiện  $<$  số vòng lặp cho phép.

{  
Lặp với công thức:

$$x_n = x_{n-1} - f(x_{n-1}) \cdot \frac{d - x_{n-1}}{f(d) - f(x_{n-1})}$$

tìm các sai số và giá trị của hàm số theo công thức sai số (1)

tìm các sai số và giá trị của hàm số theo công thức sai số (2)

}

- Phần 5) Tìm nghiệm gần đúng xn bằng phương pháp dây cung trong khoảng ly nghiệm  $(a, b)$  thỏa mãn  $|x_n - x_{n-1}| \leq e$ .

*Input:* sai số e mong muốn, khoảng ly nghiệm  $(a, b)$

*Output:* nghiệm gần đúng trong khoảng  $(a, b)$

*Ý tưởng thuật toán:*

- +) tìm giá trị nhỏ nhất và giá trị lớn nhất của đạo hàm trên khoảng  $(a, b)$
- +) xác định  $f(a), f(b)$
- +) xác định sai số đầu  $x$  với  $x = a - f(a) * (a - b) / (f(a) - f(b))$
- +) nếu  $f(x)$  và  $f(a)$  khác phía: gán  $b = a, f(b) = f(a)$
- +) vòng lặp while với điều kiện:  $sai\_so > eps$  và số vòng lặp đã thực hiện  $<$  số vòng lặp cho phép

{  
Lặp với công thức

$$x_n = x_{n-1} - f(x_{n-1}) \cdot \frac{b - x_{n-1}}{f(b) - f(x_{n-1})}$$

$sai\_so = |x_n - x_{n-1}|$

}

**Bước 2:** Hàm tìm khoảng phân ly

*Input:* Khoảng tìm nghiệm  $(-1000.5; 999.5)$

*Output:* Mảng các khoảng phân ly

Function **find\_ICS**

- +) count = 0
  - +) While( $left < right$ )
- {

*check* trùng dấu của tích giá trị hàm số tại 2 điểm  $a$  và  $b$  cách nhau 1 khoảng delta trong miền giá trị

if  $check < 0$  thì  $(a, b)$ : có thể là khoảng ly nghiệm

$count = count + 1$

else  $(a, b)$  có thể có nhiều hơn 1 nghiệm.

$left = left + delta$

**Bước 3:** Phương pháp chia đôi*Input:* Mảng các khoảng phân ly đã tìm được*Output:* Mảng các khoảng phân ly đã thu hẹp

```
function bisection_method
    +) kiểm tra dấu của  $f(a)$ 
    +) while  $|b - a| > m$ 
{
     $c = (a + b)/2.0$ 
    kiểm tra dấu của  $f(c)$ 
    if  $(f(a) * f(c) < 0)$  : gán  $b = c$ 
    else: gán  $a = c$ 
}
```

**Bước 4:** Phương pháp dây cung tìm nghiệm gần đúng với  $n$  lần lặp*Input:* Khoảng phân ly  $(a, b)$ *Output:* Nghiệm gần đúng  $x_n$ 

```
function secant_Method_n
    +) tìm giá trị nhỏ nhất và lớn nhất của đạo hàm cấp 1 trên khoảng  $(a, b)$ 
    +) tính giá trị của  $f(a), f(b)$ 
    +) tính giá trị hàm số  $f(x)$  tại  $x = a - f(a) * (a - b) / (f(a) - f(b))$ 
    +) kiểm tra dấu của tích  $f(x) * f(a)$  nếu trái dấu thì gán  $b = a, f(b) = f(a)$ 
    +) sai số tính toán
    +) while( biến lặp  $i < n$ )
{
    xác định giá trị  $x = a - f(a) * (a - b) / (f(a) - f(b))$ 
    xác định giá trị  $f(x)$  tại  $x$ 
     $i++$ 
}
```

**Bước 5:** Phương pháp dây cung tìm nghiệm gần đúng  $x_n$  với sai số  $e$ *Input:* Khoảng phân ly  $(a, b)$ *Output:* Nghiệm gần đúng  $x_n$ 

```
function secant_Method_epsilon
    +) tìm giá trị nhỏ nhất và lớn nhất của đạo hàm cấp 1 trên khoảng  $(a, b)$ 
    +) tính giá trị của  $f(a), f(b)$ 
    +) tính giá trị hàm số  $f(x)$  tại  $x = a - f(a) * (a - b) / (f(a) - f(b))$ 
    +) kiểm tra dấu của tích  $f(x) * f(a)$  :
    if  $(f(x) * f(a) < 0)$ : gán  $b = a, f(b) = f(a)$ 
    +) sai số tính toán
    +) while( $sai\_so > eps$  và  $i < \text{số vòng lặp giới hạn}$ )
{
    Xác định  $x = a - f(a) * (a - b) / (f(a) - f(b))$ 
    Xác định giá trị hàm số tại  $x$ 
     $i++$ 
     $sai\_so$  theo công thức sai số (1)
}
```

*sai\_so* theo công thức sai số (2)  
}

**Bước 6:** Phương pháp dây cung tìm nghiệm gần đúng  $x_n$  thỏa mãn  $|X_n - X_{n-1}| \leq e$

*Input:* khoảng phân ly  $(a, b)$

*Output:* Nghiệm gần đúng  $x_n$

function **secant\_Method\_e**

- +) tìm giá trị nhỏ nhất và lớn nhất của đạo hàm cấp 1 trên khoảng  $(a, b)$
- +) tính giá trị của  $f(a), f(b)$
- +) tính giá trị hàm số  $f(x)$  tại  $x = a - f(a) * (a - b) / (f(a) - f(b))$
- +) kiểm tra dấu của tích  $f(x) * f(a)$ : nếu trái dấu thì gán  $b = a, f(b) = f(a)$
- +) sai số tính toán
- +) while(*sai\_so* > *eps* và  $i < \text{số vòng lặp giới hạn}$ )

{

$x = x - f(x) * (x - b) / (f(x) - f(b))$

Xác định giá trị hàm số tại  $x$

$i++$

*sai\_so* theo công thức sai số (2)

}

**Bước 7:** Hàm kiểm tra dấu

*Input:* điểm  $x$

*Output:*  $-1$  nếu  $f(x) < 0$  hoặc  $f'(x) < 0$ .  $1$  nếu  $f(x) > 0$  hoặc  $f'(x) > 0$ .

Function **check\_sign**

- +)  $sign = 1$
- +) if (choose == 0)
- if  $f(x) < 0 : sign = -1$
- else if (choose == 1)
- . if  $f'(x) < 0 : sign = -1;$
- return**  $sign$

**Bước 8:** Công thức sai số theo sai số (1)

*Input:* điểm  $x$

*Output:* Sai số tại điểm  $x$  theo công thức (1).

Function **sai\_so\_1**

**return**  $\frac{|f(x)|}{m}$

Với  $f(x)$  là giá trị hàm số tại  $x$  và  $m$  là giá trị nhỏ nhất của đạo hàm của hàm số  $f(x)$  trên khoảng ly nghiệm  $(a, b)$

**Bước 9:** Công thức sai số theo sai số hậu nghiệm

*Input:* điểm  $x$

*Output:* Sai số tại điểm  $x$  theo công thức (2).

Function **sai\_so\_2**

**return**  $(M - m) / m * |x_n - x_{n-1}|$

Với  $M$  là giá trị lớn nhất của đạo hàm của hàm số  $f(x)$  trên khoảng ly nghiệm  $(a, b)$   
 $m$  là giá trị nhỏ nhất của đạo hàm của hàm số  $f(x)$  trên khoảng ly nghiệm  $(a, b)$   
 $|x_n - x_{n-1}|$  là độ lệch của giá trị  $x$  tại 2 lần lặp liên tiếp.

**Bước 10:** Giá trị nhỏ nhất hàm số trên đoạn.

*Input:* khoảng ly nghiệm  $(a, b)$ , giá trị của hàm số tại  $x$  với mọi  $x$  thuộc  $(a, b)$

*Output:* giá trị nhỏ nhất của hàm số trên khoảng ly nghiệm  $(a, b)$

Function **minf**

```

+) Gán  $x_0 = a, x_{min} = a$ 
+) While ( $x_0 < b$ )
{
  if ( $|f(x_0)| < |f(x_{min})|$  và  $x_0 < b$ ):  $x_{min} = x_0$ 
   $x_0 = x_0 + delta$ 
}
+) if ( $|f(x_{min})| > |f(b)|$ ): return  $|f(b)|$ 
else: return  $|f(x_{min})|$ 
```

**Bước 11:** Giá trị lớn nhất của hàm số trên đoạn

*Input:* khoảng ly nghiệm  $(a, b)$ ,  $delta$

*Output:* giá trị lớn nhất của hàm số trên khoảng ly nghiệm  $(a, b)$

Function **maxf**

```

+) Gán  $x_0 = a, x_{max} = a$ 
+ While ( $x_0 < b$ )
{
  if ( $|f(x_0)| > |f(x_{max})|$  và  $x_0 < b$ ):  $x_{max} = x_0$ 
   $x_0 = x_0 + delta$ 
}
+) if ( $|f(x_{max})| < |f(b)|$ ): return  $|f(b)|$ 
else: return  $|f(x_{max})|$ 
```

**Bước 12:** giá trị hàm số  $f(x)$

*Input:* giá trị  $x$ , mảng *pholy*[ ] chứa hạng tử của  $f(x)$

*Output:* giá trị hàm số  $f(x)$  tại  $x$

Function **f(x)**

```

+) biến SUM lưu giá trị của hàm số, khởi tạo SUM = 0;
+) for( $i = 0; i < num; i++$ )
{
   $SUM = SUM + polynum[i].coefficient * pow(x, polynom[i].degree)$ 
}
+) return SUM
```

**Bước 13:** tính giá trị đạo hàm

*Input:* giá trị  $x$

*Output:* giá trị đạo hàm của hàm số  $f(x)$  tại  $x$

*Thuật toán:* Theo phương pháp xấp xỉ hai phía với  $h$  là 1 số rất nhỏ

Function **f1(x)**

return  $\frac{f(x+h) - f(x-h)}{2 * h}$

Cài đặt chương trình:

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <math.h>
4 #include <windows.h>
5
6 int NUM, count_ ;
7 const double h = 1e-9;
8 const int MAX_LOOP = 10000;
9 double left = -1000.5, right = 999.5;
10 double a, b, x, x_old, fa, fb, fx, eps, m, M;
11 FILE *fin;
12 FILE *fout;
13
14 typedef struct element_Of_The_Polynomial
15 {
16     int degree;
17     double coefficient;
18 } NODE;
19
20 typedef struct intervals_Containing_The_Solution
21 {
22     double a;
23     double b;
24 } ICS;
25
26 //-----MANG LUU HAM SO-----
27 NODE polynom[15];
28
29 //-----KHOANG PHAN LY-----
30 ICS phanly[15];
31
32 //-----KHAI BAO HAM-----
33 void color(int color);
34 void gotoxy(int x, int y);
35
36 //Cac ham tinh gia tri f(x), f'(x), f''(x)
37 double f(double x);
38 double f1(double x);
39 double f2(double x);
40
41 //Ham kiem tra dau cua ham so
42 int check_sign(double x, int choose);
43
44 // ham ve khung menu
45 void drawframe();
46 //ham xu ly input
47 void menu_input();
48 //ham in bieu thuc
49 void display_Function();
50 //ham in cac khoang phan ly
51 int printICS();
52

```

```

53 //Cac ham tinh sai so
54 double sai_so_1();
55 double sai_so_2();
56
57 //Nhom ham tim MIN MAX ham so
58 double minf(double f(double x), ICS local);
59 double maxf(double f(double x), ICS local);
60
61 //Tim khoang phan ly
62 void find_ICS(double temp_left, double temp_right);
63 //Chia doi khoang phan ly
64 int bisection_Method();
65 //Ham tim nghiem voi so lan lap cho truoc
66 void secant_Method_n(ICS local);
67 //Ham tim nghiem voi sai so cho truoc
68 void secant_Method_epsilon(ICS local);
69 //Ham tim nghiem sai so  $|x_n - x_{n-1}| < \epsilon$ 
70 void secant_Method_e(ICS local);
71
72
73
74 int main(){
75
76     int hold, cnt;
77     ICS local;
78     //bien sd trong layer menu 1
79     int Set[] = {7, 7, 7, 7, 7, 7};
80     int counter = 1;
81     char key;
82     //bien sd trong layer menu 2
83     int Set1[] = {7, 7, 7};
84     int counter1 = 1;
85     char key1;
86
87     if ( ( fout = fopen( "output.txt", "w" ) ) == NULL ){
88         printf("Open file output error");
89     }
90     else
91     {
92         menu_input();
93         while(1){
94             drawframe();
95             if(counter == 1){
96                 Set[0] = 74;
97             }
98             gotoxy(40,6);
99             color(Set[0]);
100             printf(" 1. TIM KHOANG PHAN LY NGHIEM ");
101             gotoxy(40,8);
102             color(Set[1]);
103             printf(" 2. RUT GON KHOANG PHAN LY NGHIEM ");
104             gotoxy(40,10);
105             color(Set[2]);
106             printf(" 3. PHUONG PHAP DAY CUNG VOI \"N\" LAN LAP");
107             gotoxy(40,12);
108             color(Set[3]);
109             printf(" 4. PHUONG PHAP DAY CUNG VOI SAI SO \"e\"");
110             gotoxy(40,14);
111             color(Set[4]);
112             printf(" 5. PHUONG PHAP DAY CUNG T/M  $|X_n - X_{n-1}| \leq e$ ");

```



```

113         gotoxy(40,16);
114         color(Set[5]);
115         printf(" 6. THOAT CHUONG TRINH");
116
117         key = getch();
118         if(key == 72 && (counter >= 2 && counter <=6)){
119             counter--;
120
121         }
122         else if(key == 80 && (counter >= 1 && counter <=5)){
123             counter++;
124
125         }
126         else if(key == 72 && (counter <=1)){
127             counter = 6;
128         }
129         else if(key == 80 && (counter >= 6)){
130             counter = 1;
131         }
132         if(key == '\r'){
133             if(counter == 6){
134                 color(7);
135                 system("cls");
136                 break;
137             }
138             system("cls");
139             gotoxy(0,5);
140             if(counter == 1){
141                 find_ICS(left, right);
142                 printf("\nKhoang phan ly: ");
143                 fprintf(fout, "\nKhoang phan ly: ");
144                 printf("\n\t\t\t\t+-----+-----+-----+");
145                 printf("\n\t\t\t\t| STT | a | b |");
146                 printf("\n\t\t\t\t+-----+-----+-----+");
147                 fprintf(fout, "\n\t\t\t\t+-----+-----+-----+");
148                 fprintf(fout, "\n\t\t\t\t| STT | a | b |");
149                 fprintf(fout, "\n\t\t\t\t+-----+-----+-----+");
150                 for(cnt = 0; cnt < count_; cnt++)
151                 {
152                     if(phanly[cnt].a < 0 && phanly[cnt].b < 0){
153                         printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
154                         fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
155                     }
156                     else if(phanly[cnt].a > 0 && phanly[cnt].b < 0){
157                         printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
158                         fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
159                     }
160                     else if(phanly[cnt].a < 0 && phanly[cnt].b > 0){
161                         printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
162                         fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
163                     }
164                     else
165                     {
166                         printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,

```

```

phanly[cnt].a, phanly[cnt].b);
167         fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
168     }
169     printf("\n\t\t\t\t\t+-----|-----|-----+");
170     fprintf(fout, "\n\t\t\t\t\t+-----|-----|-----+");
171 }
172 }
173 else if(counter == 2)
174 {
175     bisection_Method();
176     printf("\nKhoang phan ly sau khi thu hep: ");
177     fprintf(fout, "\nKhoang phan ly sau khi thu hep: ");
178     printf("\n\t\t\t\t\t+-----|-----|-----+");
179     printf("\n\t\t\t\t\t STT | a | b |");
180     printf("\n\t\t\t\t\t+-----|-----|-----+");
181     fprintf(fout, "\n\t\t\t\t\t+-----|-----|-----+");
182     fprintf(fout, "\n\t\t\t\t\t STT | a | b |");
183     fprintf(fout, "\n\t\t\t\t\t+-----|-----|-----+");
184     for(cnt = 0; cnt < count_; cnt++)
185     {
186         if(phanly[cnt].a < 0 && phanly[cnt].b < 0){
187             printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
188             fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
189         }
190         else if(phanly[cnt].a > 0 && phanly[cnt].b < 0){
191             printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
192             fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
193         }
194         else if(phanly[cnt].a < 0 && phanly[cnt].b > 0){
195             printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
196             fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
197         }
198         else
199         {
200             printf("\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
201             fprintf(fout, "\n\t\t\t\t\t %d | %.3lf | %.3lf |", cnt,
phanly[cnt].a, phanly[cnt].b);
202         }
203         printf("\n\t\t\t\t\t+-----|-----|-----+");
204         fprintf(fout, "\n\t\t\t\t\t+-----|-----|-----+");
205     }
206 }
207 else if(counter == 3)
208 {
209     Set1[0] = 7;
210     Set1[1] = 7;
211     Set1[2] = 7;
212     counter1 = 1;
213     while(1){
214         drawframe();
215         if(counter1 == 1){
216             Set1[0] = 74;

```

```

217     }
218     gotoxy(40,6);
219     color(Set1[0]);
220     printf(" 1. Khoảng phân ly nhập vào tu bàn phím ");
221     gotoxy(40,8);
222     color(Set1[1]);
223     printf(" 2. Khoảng phân ly đã tìm được ở trước");
224     gotoxy(40,10);
225     color(Set1[2]);
226     printf(" 3. THOAT");
227
228     key1 = getch();
229     if(key1 == 72 && (counter1 >= 2 && counter1 <=3)){
230         counter1--;
231
232     }
233     else if(key1 == 80 && (counter1 >= 1 && counter1 <=2)){
234         counter1++;
235
236     }
237     else if(key1 == 72 && (counter1 <=1)){
238         counter1 = 3;
239     }
240     else if(key1 == 80 && (counter1 >= 3)){
241         counter1 = 1;
242     }
243     if(key1 == '\r'){
244         if(counter1 == 3){
245             color(7);
246             system("cls");
247             break;
248         }
249         system("cls");
250         gotoxy(0,5);
251         if(counter1 == 1){
252             printf("Nhập khoảng phân ly (a, b): ");
253             printf("\na = "); scanf("%lf", &local.a);
254             printf("b = "); scanf("%lf", &local.b);
255             secant_Method_n(local);
256             break;
257         }
258         if(counter1 == 2){
259             hold = printICS();
260             local.a = phanly[hold - 1].a;
261             local.b = phanly[hold - 1].b;
262             secant_Method_n(local);
263             break;
264         }
265         key1 = getch();
266         if(key1 == '\r')
267             system("cls");
268     }
269     Set1[0] = 7;
270     Set1[1] = 7;
271     Set1[2] = 7;
272     if(counter1 == 1){
273         Set1[0] = 74;
274     }
275     if(counter1 == 2){
276         Set1[1] = 74;

```

```

277         }
278         if(counter1 == 3){
279             Set1[2] = 74;
280         }
281     }
282 }
283
284 else if(counter == 4)
285 {
286     Set1[0] = 7;
287     Set1[1] = 7;
288     Set1[2] = 7;
289     counter1 = 1;
290     while(1){
291         drawframe();
292         if(counter1 == 1){
293             Set1[0] = 74;
294         }
295         gotoxy(40,6);
296         color(Set1[0]);
297         printf(" 1. Khoảng phân ly nhập vào tu ban phim ");
298         gotoxy(40,8);
299         color(Set1[1]);
300         printf(" 2. Khoảng phân ly đã tìm được ở trước");
301         gotoxy(40,10);
302         color(Set1[2]);
303         printf(" 3. THOAT");
304
305         key1 = getch();
306         if(key1 == 72 && (counter1 >= 2 && counter1 <=3)){
307             counter1--;
308
309         }
310         else if(key1 == 80 && (counter1 >= 1 && counter1 <=2)){
311             counter1++;
312
313         }
314         else if(key1 == 72 && (counter1 <=1)){
315             counter1 = 3;
316         }
317         else if(key1 == 80 && (counter1 >= 3)){
318             counter1 = 1;
319         }
320         if(key1 == '\r'){
321             if(counter1 == 3){
322                 color(7);
323                 system("cls");
324                 break;
325             }
326             system("cls");
327             gotoxy(0,5);
328             if(counter1 == 1){
329                 printf("Nhập khoảng phân ly (a, b): ");
330                 printf("\na = "); scanf("%lf", &local.a);
331                 printf("b = "); scanf("%lf", &local.b);
332                 secant_Method_epsilon(local);
333                 break;
334             }
335             if(counter1 == 2){
336                 hold = printICS();

```

```

337         local.a = phanly[hold - 1].a;
338         local.b = phanly[hold - 1].b;
339         secant_Method_epsilon(local);
340         break;
341     }
342     key1 = getch();
343     if(key1 == '\r')
344         system("cls");
345 }
346 Set1[0] = 7;
347 Set1[1] = 7;
348 Set1[2] = 7;
349 if(counter1 == 1){
350     Set1[0] = 74;
351 }
352 if(counter1 == 2){
353     Set1[1] = 74;
354 }
355 if(counter1 == 3){
356     Set1[2] = 74;
357 }
358 }
359 }
360
361 else if(counter == 5)
362 {
363     Set1[0] = 7;
364     Set1[1] = 7;
365     Set1[2] = 7;
366     counter1 = 1;
367     while(1){
368         drawframe();
369         if(counter1 == 1){
370             Set1[0] = 74;
371         }
372         gotoxy(40,6);
373         color(Set1[0]);
374         printf(" 1. Khoảng phân ly nhập vào tu bàn phím ");
375         gotoxy(40,8);
376         color(Set1[1]);
377         printf(" 2. Khoảng phân ly đã tìm được ở trước");
378         gotoxy(40,10);
379         color(Set1[2]);
380         printf(" 3. THOÁT");
381
382         key1 = getch();
383         if(key1 == 72 && (counter1 >= 2 && counter1 <=3)){
384             counter1--;
385
386         }
387         else if(key1 == 80 && (counter1 >= 1 && counter1 <=2)){
388             counter1++;
389
390         }
391         else if(key1 == 72 && (counter1 <=1)){
392             counter1 = 3;
393         }
394         else if(key1 == 80 && (counter1 >= 3)){
395             counter1 = 1;
396         }

```

```

397         if(key1 == '\r'){
398             if(counter1 == 3){
399                 color(7);
400                 system("cls");
401                 break;
402             }
403             system("cls");
404             gotoxy(0,5);
405             if(counter1 == 1){
406                 printf("Nhap khoang phan ly (a, b): ");
407                 printf("\na = "); scanf("%lf", &local.a);
408                 printf("b = "); scanf("%lf", &local.b);
409                 secant_Method_e(local);
410                 break;
411             }
412             if(counter1 == 2){
413                 hold = printICS();
414                 local.a = phanly[hold - 1].a;
415                 local.b = phanly[hold - 1].b;
416                 secant_Method_e(local);
417                 break;
418             }
419             key1 = getch();
420             if(key1 == '\r')
421                 system("cls");
422         }
423         Set1[0] = 7;
424         Set1[1] = 7;
425         Set1[2] = 7;
426         if(counter1 == 1){
427             Set1[0] = 74;
428         }
429         if(counter1 == 2){
430             Set1[1] = 74;
431         }
432         if(counter1 == 3){
433             Set1[2] = 74;
434         }
435     }
436 }
437
438 key = getch();
439 if(key == '\r')
440     system("cls");
441 }
442 Set[0] = 7;
443 Set[1] = 7;
444 Set[2] = 7;
445 Set[3] = 7;
446 Set[4] = 7;
447 Set[5] = 7;
448 if(counter == 1){
449     Set[0] = 74;
450 }
451 if(counter == 2){
452     Set[1] = 74;
453 }
454 if(counter == 3){
455     Set[2] = 74;
456 }

```

```

457         if(counter == 4){
458             Set[3] = 74;
459         }
460         if(counter == 5){
461             Set[4] = 74;
462         }
463         if(counter == 6){
464             Set[5] = 74;
465         }
466     }
467 }
468
469 fclose(fout);
470 return 1;
471
472 }
473
474
475 //Nhom ham setup mau sac
476 void color(int color){
477
478     SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),color);
479 }
480
481 void gotoxy(int x, int y){
482     COORD c;
483     c.X=x;
484     c.Y=y;
485     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),c);
486 }
487
488 void drawframe(){
489     int i;
490     gotoxy(60,3);
491     color(12);
492     printf("MENU");
493     color(4);
494     for(i=0; i<63; i++){
495         gotoxy(31+i,2);
496         printf("_");
497         gotoxy(31+i,4);
498         printf("_");
499         gotoxy(31+i,19);
500         printf("-");
501     }
502     for(i=0; i<=15; i++){
503         gotoxy(30,2);
504         gotoxy(29,3+i);
505         printf("||");
506         gotoxy(94,3+i);
507         printf("||");
508     }
509     color(7);
510 }
511
512 //Nhom ham nhan INPUT du lieu
513 void menu_input(){
514     int Set[] = {7,7,7,7,7};
515     int counter = 1, i = 0;
516     char key;

```

```

517
518 while(1){
519     drawframe();
520     if(counter == 1){
521         Set[0] = 74;
522     }
523     gotoxy(40,6);
524     color(Set[0]);
525     printf(" 1. NHAP HAM SO TU FILE ");
526     gotoxy(40,8);
527     color(Set[1]);
528     printf(" 2. NHAP HAM SO TU TERMINAL ");
529     gotoxy(40,10);
530     color(Set[2]);
531     printf(" 3. THOAT NHAP LIEU ");
532
533     key = getch();
534     if(key == 72 && (counter >= 2 && counter <=3)){
535         counter--;
536     }
537
538     else if(key == 80 && (counter >= 1 && counter <=2)){
539         counter++;
540     }
541
542     else if(key == 72 && (counter <=1)){
543         counter = 3;
544     }
545     else if(key == 80 && (counter >= 3)){
546         counter = 1;
547     }
548     if(key == '\r'){
549         if(counter == 3){
550             color(7);
551             system("cls");
552             break;
553         }
554         system("cls");
555         gotoxy(0,5);
556         if(counter == 1){
557             if((fin = fopen("input.txt", "r")) ==NULL)
558                 printf("open file error");
559             else{
560                 while(!feof(fin)){
561
562                     fscanf(fin, "%d", &polynom[i].degree);
563                     fscanf(fin, "%lf", &polynom[i].coefficient);
564
565                     if(feof(fin))
566                         break;
567                     else
568                         i++;
569                 }
570                 NUM = i;
571                 fclose(fin);
572             }
573             display_Function();
574         }
575         if(counter == 2){
576             do{

```



```

577         printf("Cho biet so cac so hang cua da thuc: ");
578         scanf("%d", &NUM);
579         if (NUM <= 1)
580             printf("\nBan da nhap sai. Vui long nhap lai voi n >1!!!");
581     }while (NUM < 0);
582
583     for (i = 0; i < NUM; i++)
584     {
585         printf("So hang thu %d la: \n", i+1);
586         printf("Bac: \n");
587         scanf("%d", &polynom[i].degree);
588         printf("He so: \n");
589         scanf("%lf",&polynom[i].coefficient);
590     }
591     display_Function();
592 }
593 key = getch();
594 if(key == '\r')
595     system("cls");
596 }
597 Set[0] = 7;
598 Set[1] = 7;
599 Set[2] = 7;
600 if(counter == 1){
601     Set[0] = 74;
602 }
603 if(counter == 2){
604     Set[1] = 74;
605 }
606 if(counter == 3){
607     Set[2] = 74;
608 }
609 }
610 }
611
612 void display_Function(){
613     int i;
614     printf("Bieu thuc: \n");
615     fprintf(fout, "Bieu thuc: \n");
616     for(i = 0; i < NUM; i++)
617     {
618         if(polynom[i].degree == 0)
619         {
620             printf(" %f ", polynom[i].coefficient);
621             fprintf(fout, " %f ", polynom[i].coefficient);
622         }
623         else
624         {
625             printf(" %fx^%d ", polynom[i].coefficient, polynom[i].degree);
626             fprintf(fout, " %fx^%d ", polynom[i].coefficient, polynom[i].degree);
627         }
628         if(i < NUM-1)
629         {
630             printf("+");
631             fprintf(fout, "+");
632         }
633     }
634 }
635
636 //Nhom ham tinh gia tri f(x), f'(x)

```

```

637 double f(double x){
638     int i;
639     double SUM = 0;
640     for (i = 0; i < NUM; i++)
641     {
642         SUM += polynom[i].coefficient * pow(x,polynom[i].degree);
643     }
644     return SUM;
645 }
646
647 double f1(double x){
648     return (f(x + h) - f(x - h)) / (2 * h);
649 }
650
651
652 //Ham check dau cua ham so
653 int check_sign(double x, int choose){
654     int sign = 1;
655     if(choose == 0)
656         if (f(x) < 0)
657         {
658             sign = -1;
659         }
660     else if(choose == 1)
661         if (f1(x) < 0)
662         {
663             sign = -1;
664         }
665     return sign;
666 }
667
668 //Nhom ham tim MIN MAX ham so tren doan
669 double minf(double f(double x), ICS local){
670     double x0, x_min = local.a;
671     double delta = (local.b - local.a) / 100000;
672
673     x0 = local.a;
674     while(x0< local.b)
675     {
676         if(fabs(f(x0)) < fabs(f(x_min)) && x0 < local.b)
677         {
678             x_min = x0;
679         }
680         x0 += delta;
681     }
682     if(fabs(f(x_min)) > fabs(f(local.b)))
683         return fabs(f(local.b));
684     else
685         return fabs(f(x_min));
686 }
687
688 double maxf(double f(double x), ICS local){
689     double x0, x_max = local.a;
690     double delta = (local.b - local.a) / 100000;
691     x0 = local.a;
692     while(x0 < local.b)
693     {
694         if((fabs(f(x0)) > fabs(f(x_max))) && (x0 < local.b))
695         {
696             x_max = x0;

```

```

697     }
698     x0 += delta;
699 }
700 if(fabs(f(x_max)) < fabs(f(local.b)))
701     return fabs(f(local.b));
702 else
703     return fabs(f(x_max));
704 }
705
706 //Nhóm hàm tính sai số
707 double sai_so_1(){
708     return fabs(fx) / m;
709 }
710 double sai_so_2(){
711     return (M-m) / m * fabs(x-x_old);
712 }
713
714 //Hàm in khoảng phân ly
715 int printICS(){
716     int Set[count_];
717     int counter = 1, i = 0;
718     char key;
719
720     for(i = 0; i < count_; i++)
721     {
722         Set[i] = 7;
723     }
724     fprintf(fout, "\n\n\t\t\t\t\t+--+-----+--+");
725     for(i = 0; i < count_; i++)
726     {
727         fprintf(fout, "\n\t\t\t\t\t %d. Khoảng phân li thu %d: (%.3lf, %.3lf) |",
728             i+1, i+1, phanly[i].a, phanly[i].b);
729     }
730     fprintf(fout, "\n\t\t\t\t\t+--+-----+--+");
731
732     while(1){
733         drawframe();
734         if(counter == 1){
735             Set[0] = 74;
736         }
737         for(i = 0; i < count_; i++)
738         {
739             gotoxy(40, 6 + i);
740             color(Set[i]);
741             printf(" %d. Khoảng phân li thu %d: (%.3lf, %.3lf)", i+1, i+1, phanly[i].a,
742                 phanly[i].b);
743         }
744
745         key = getch();
746         if(key == 72 && (counter >= 2 && counter <= count_)){
747             counter--;
748         }
749         else if(key == 80 && (counter >= 1 && counter <= count_ - 1)){
750             counter++;
751         }
752         else if(key == 72 && (counter <= 1)){
753             counter = count_;
754         }

```

```

755     else if(key == 80 && (counter >= count_)){
756         counter = 1;
757     }
758     if(key == '\r'){
759         color(7);
760         system("cls");
761         fprintf(fout, "\nLua chon: %d\n", counter);
762         return counter;
763     }
764 }
765
766     for(i = 0; i < count_; i++)
767     {
768         Set[i] = 7;
769     }
770     Set[counter - 1] = 74;
771 }
772 }
773
774 //Nhom ham chinh cua de bai
775
776 void find_ICS(double temp_left, double temp_right){           //Ham tim khoang phan ly
777     double check = 0, temp;
778     int i, dem;
779     float delta = 0.701;
780     count_ = 0;
781     while(temp_left <= temp_right)
782     {
783         x_old = check_sign(temp_left, 0);
784         x = check_sign(temp_left + delta, 0);
785         check = x*x_old;
786         if (check < 0)
787         {
788             temp = temp_left;
789             dem = 0;
790             for(i = 0; i < 1001; i++)
791             {
792                 if(check_sign(temp, 1) != check_sign(temp + delta/1001, 1))
793                 {
794                     dem ++;
795                 }
796                 temp = temp + delta/1001;
797             }
798             if(dem == 0)
799             {
800                 phanly[count_].a = temp_left;
801                 phanly[count_].b = temp_left + delta;
802                 count_++;
803             }
804             else{
805                 printf("Trong khoang (%.2lf, %.2lf) co it nhat %d nghiem. Moi khao sat
ham so.", temp_left, temp_left + delta, dem);
806                 fprintf(fout, " \n\nTrong khoang (%.2lf, %.2lf) co it nhat %d nghiem.
Moi khao sat ham so.", temp_left, temp_left + delta, dem);
807             }
808         }
809         temp_left = temp_left + delta;
810     }
811 }
812

```

```

813 int bisection_Method(){ //Ham thu hep khoang phan ly bang phuong phap chia
    doi
814     float m;
815     double p, q, c, f_p, f_c;
816     int i = 0, check_sign_fp, check_sign_fc;
817     printf("\n\nVoi khoang phan ly nghiem (p,q) voi |p-q| <= m. Nhap vao so m: ");
818     fprintf(fout, "\n\nVoi khoang phan ly nghiem (p,q) voi |p-q| <= m. Nhap vao so m: ");
819     scanf("%f", &m);
820     fprintf(fout, "\nm = %f", m);
821
822     while (i < count_)
823     {
824         p = phanly[i].a;
825         q = phanly[i].b;
826
827         f_p = f(p);
828         check_sign_fp = check_sign(p, 0);
829
830         while(fabs(q-p) > m)
831         {
832             c = (p + q) / 2.0;
833             f_c = f(c);
834             check_sign_fc = check_sign(c, 0);
835             if(check_sign_fp != check_sign_fc)
836                 q = c;
837             else
838                 p = c;
839         }
840         phanly[i].a = p;
841         phanly[i].b = q;
842         i++;
843     }
844 }
845
846 void secant_Method_n(ICS local){ //Ham tim nghiem voi so lan lap cho truoc
847     int i, n;
848     double sai_so;
849     printf("Nhap so lan lap: ");
850     scanf("%d", &n);
851     fprintf(fout, "\n\nNhap so lan lap: ");
852     fprintf(fout, "\t%d", n);
853     m = minf(f1, local);
854     M = maxf(f1, local);
855     a = local.a;
856     b = local.b;
857     fa = f(a);
858     fb = f(b);
859     x = a - fa * (a - b) / (fa - fb);
860     fx = f(x);
861     if (fx * fa < 0)
862     {
863         b = a;
864         fb = fa;
865     }
866
867     printf("\n\t\t\t\t\t+--+ Lap nghiem voi %d lam lap +--+ ", n);
868     fprintf(fout, "\n\n\t\t\t\t\t+--+ Lap nghiem voi %d lan lap +--+ \n", n);
869
870     printf("\n\t\t\t\t\t+-----|-----|-----+");
871     printf("\n\t\t\t\t\t| Lan lap |          x_n          |          f(x_n)          |");

```

```

872 printf("\n\t\t\t\t+-----|-----|-----+");
873 fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
874 fprintf(fout, "\n\t\t\t\t| Lan lap |          x_n          |          f(x_n)          |");
875 fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
876
877 if (fx < 0)
878 {
879     printf("\n\t\t\t\t| 1\t | x = %.8lf | f(x) = %.8lf |", x, fx);
880     fprintf(fout, "\n\t\t\t\t| 1\t | x = %.8lf | f(x) = %.8lf |", x, f(x));
881 }
882 else
883 {
884     printf("\n\t\t\t\t| 1\t | x = %.8lf | f(x) = %.8lf |", x, fx);
885     fprintf(fout, "\n\t\t\t\t| 1\t | x = %.8lf | f(x) = %.8lf |", x, f(x));
886 }
887 i = 1;
888 while (i < n)
889 {
890     x_old = x;
891     x = x - fx * (x - b) / (fx - fb);
892     fx = f(x);
893     i++;
894     printf("\n\t\t\t\t+-----|-----|-----+");
895     fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
896     if (fx < 0)
897     {
898         printf("\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x, fx);
899         fprintf(fout, "\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x,
900             fx);
901     }
902     else
903     {
904         printf("\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x, fx);
905         fprintf(fout, "\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x,
906             fx);
907     }
908 }
909 printf("\n\t\t\t\t+-----|-----|-----+");
910 fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
911 printf("\n\nVay nghiem x* = %.12lf", x);
912 printf("\nSo lan lap: %d", i);
913 printf("\n\nSai so 1: %.12lf", sai_so_1());
914 printf("\nSai so 2: %.12lf", sai_so_2());
915 fprintf(fout, "\n\nVay nghiem x* = %.10lf", x);
916 fprintf(fout, "\nSo lan lap: %d", i);
917 fprintf(fout, "\n\nSai so 1: %.12lf", sai_so_1());
918 fprintf(fout, "\nSai so 2: %.12lf", sai_so_2());
919 }
920 void secant_Method_epsilon(ICS local){ //Ham tim nghiem voi sai so cho truoc
921     int i;
922     double sai_so;
923     printf("Nhap sai so \"e\" mong muon: ");
924     scanf("%lf", &eps);
925     fprintf(fout, "\n\nNhap sai so \"e\" mong muon: ");
926     fprintf(fout, "\t%lf", eps);
927     m = minf(f1, local);
928     M = maxf(f1, local);
929     a = local.a;

```







```

1046     fprintf(fout, "\n\nVay nghiem x* = %10lf", x);
1047     fprintf(fout, "\nSo lan lap: %d", i);
1048 }
1049 }
1050
1051 void secant_Method_e(ICS local){    //Ham tim nghiem voi |x_n - x_{n-1}| < eps
1052     int i;
1053     double sai_so;
1054
1055     printf("Nhap sai so \"e\" mong muon: ");
1056     scanf("%lf", &eps);
1057     fprintf(fout, "\n\nNhap sai so \"e\" mong muon: ");
1058     fprintf(fout, "\t%lf", eps);
1059     m = minf(f1, local);
1060     M = maxf(f1, local);
1061     a = local.a;
1062     b = local.b;
1063     fa = f(a);
1064     fb = f(b);
1065     x = a - fa * (a - b) / (fa - fb);
1066     fx = f(x);
1067     if (f(x) * fa < 0)
1068     {
1069         b = a;
1070         fb = fa;
1071     }
1072     sai_so = fabs(x - a);
1073     printf("\n\t\t\t\t Lap nghiem thoa man |X{n} - X{n-1}| < %.9lf", eps);
1074     fprintf(fout, "\n\t\t\t\t+---+ Lap nghiem thoa man |X{n} - X{n-1}| < %lf +---+\n",
        eps);
1075
1076     printf("\n\t\t\t\t+-----|-----|-----+");
1077     printf("\n\t\t\t\t| Lan lap |          x_n          |          f(x_n)          |");
1078     printf("\n\t\t\t\t+-----|-----|-----+");
1079     fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
1080     fprintf(fout, "\n\t\t\t\t| Lan lap |          x_n          |          f(x_n)          |");
1081     fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
1082
1083     if(fx < 0)
1084     {
1085         printf("\n\t\t\t\t\t 1\t | x = %.8lf | f(x) = %.8lf |", x, fx);
1086         fprintf(fout, "\n\t\t\t\t\t 1\t | x = %.8lf | f(x) = %.8lf |", x, f(x));
1087     }
1088     else
1089     {
1090         printf("\n\t\t\t\t\t 1\t | x = %.8lf | f(x) = %.8lf |", x, fx);
1091         fprintf(fout, "\n\t\t\t\t\t 1\t | x = %.8lf | f(x) = %.8lf |", x, f(x));
1092     }
1093     i = 1;
1094     while (sai_so > eps && i < MAX_LOOP)
1095     {
1096         x_old = x;
1097         x = x - fx * (x - b) / (fx - fb);
1098         fx = f(x);
1099         i++;
1100         sai_so = fabs(x - x_old);
1101         printf("\n\t\t\t\t+-----|-----|-----+");
1102         fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
1103         if(fx < 0)
1104         {

```

```

1105     printf("\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x, fx);
1106     fprintf(fout, "\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x,
fx);
1107 }
1108 else
1109 {
1110     printf("\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x, fx);
1111     fprintf(fout, "\n\t\t\t\t| %2d\t | x = %.8lf | f(x) = %.8lf |", i, x,
fx);
1112 }
1113 }
1114 printf("\n\t\t\t\t+-----|-----|-----+");
1115 fprintf(fout, "\n\t\t\t\t+-----|-----|-----+");
1116 printf("\n\nVay nghiem x* = %.10lf", x);
1117 printf("\nSo lan lap: %d", i);
1118 fprintf(fout, "\n\nVay nghiem x* = %.10lf", x);
1119 fprintf(fout, "\nSo lan lap: %d", i);
1120
1121 }

```

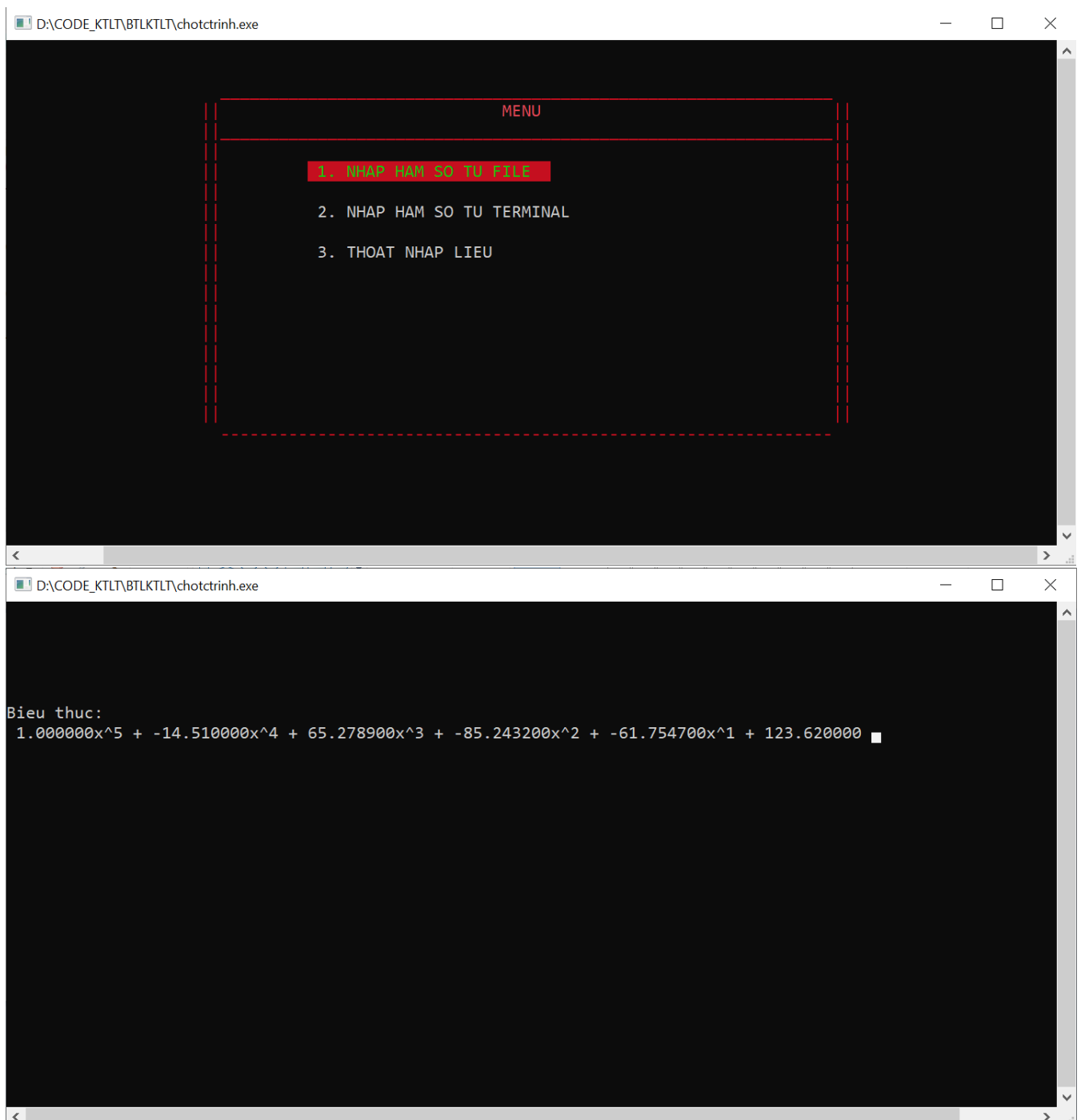
# IV

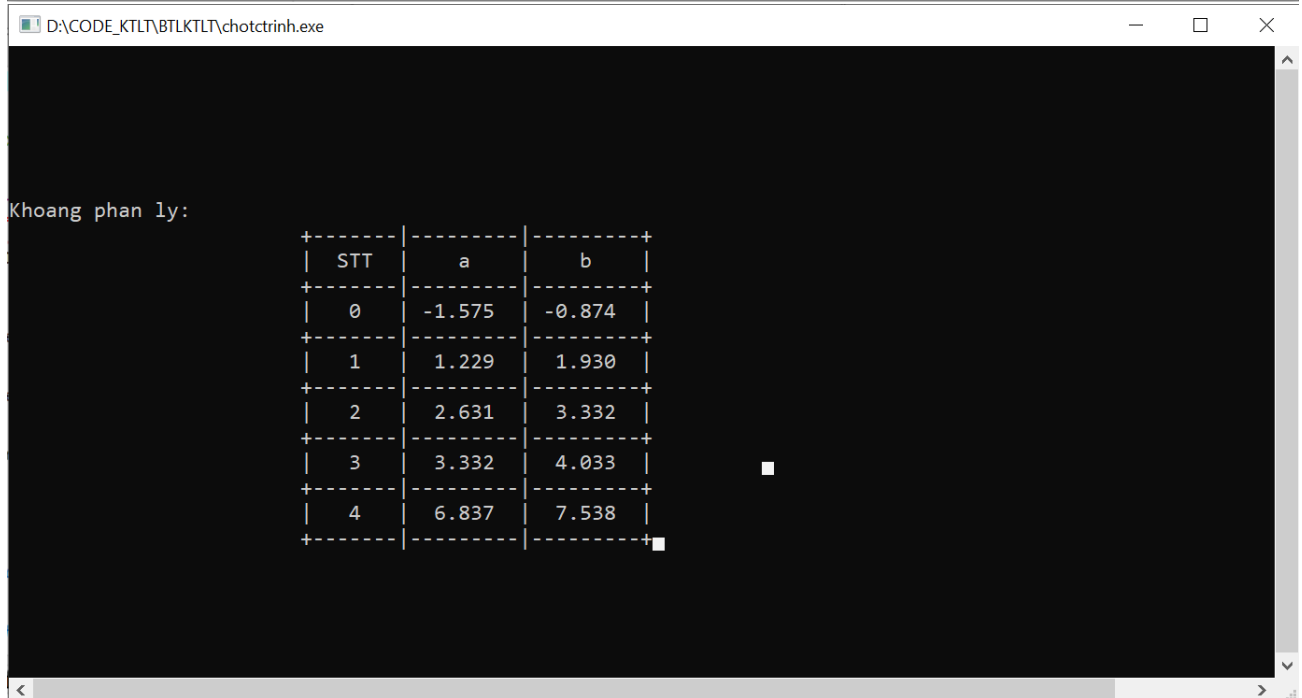
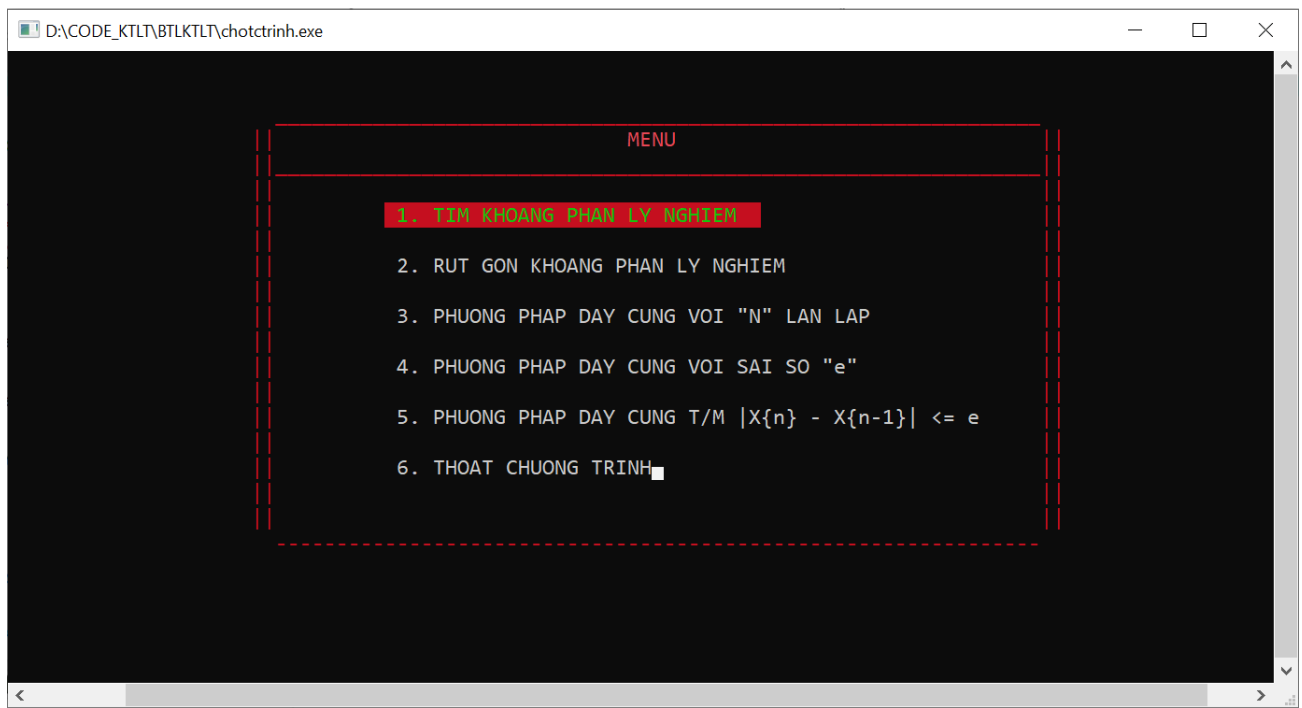
## Kiểm tra chương trình

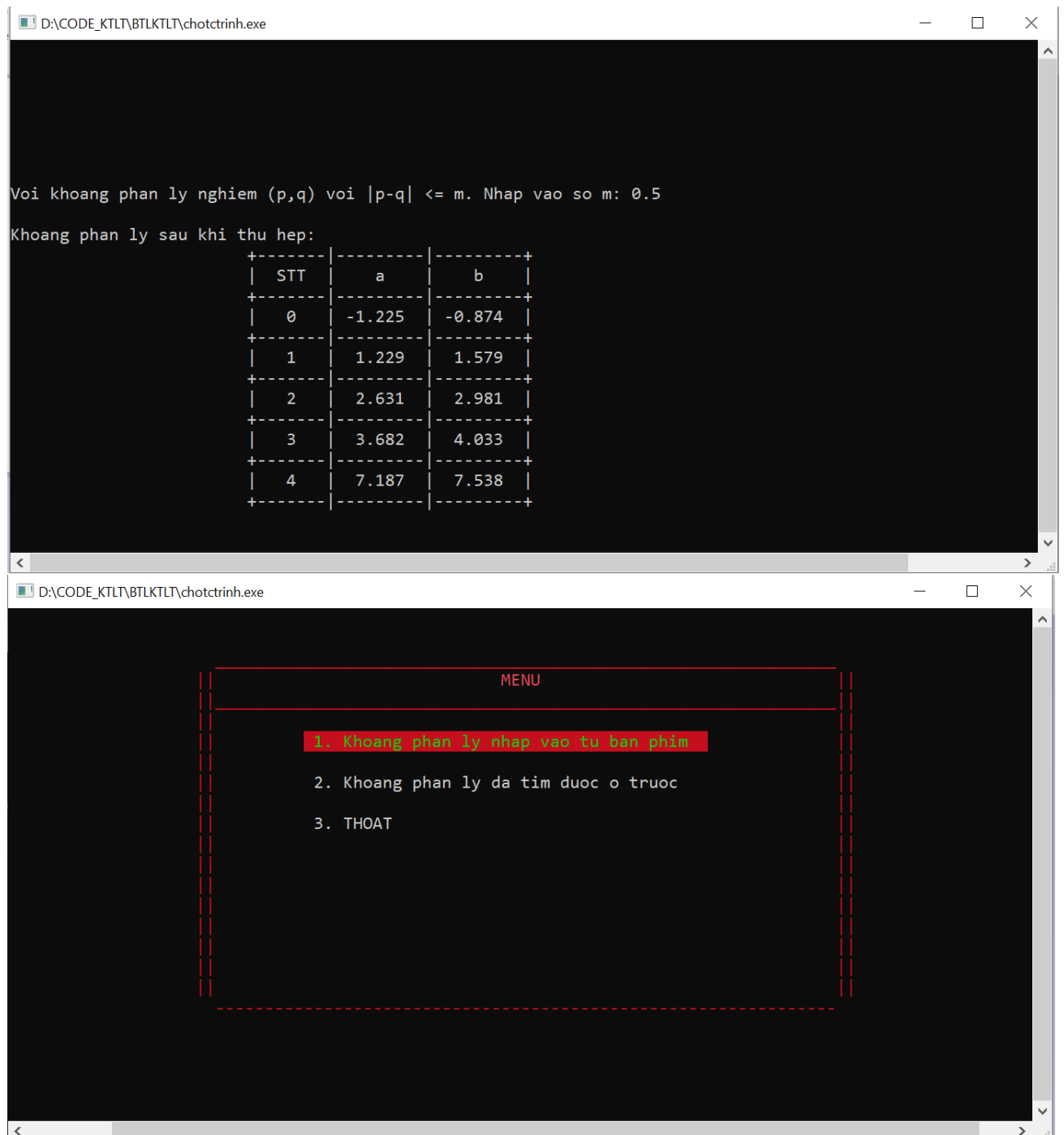
Sau khi thiết kế chương trình xong chúng em đã kiểm tra lại với nhiều hàm số, trong báo cáo này chúng em sẽ chạy thử chương trình với ví dụ như sau:

$$f(x) = x^5 - 14.51x^4 + 65.2789x^3 - 85.2432x^2 - 61.7547x + 123.62$$

Quá trình chạy chương trình:







```
Select D:\CODE_KTLT\BTLKTLT\chotctrinh.exe
```

MENU

- 1. Khoảng phân li thu 1: (-1.225, -0.874)
- 2. Khoảng phân li thu 2: (1.229, 1.579)
- 3. Khoảng phân li thu 3: (2.631, 2.981)
- 4. Khoảng phân li thu 4: (3.682, 4.033)
- 5. Khoảng phân li thu 5: (7.187, 7.538)

```
D:\CODE_KTLT\BTLKTLT\chotctrinh.exe
```

Nhap so lan lap: 5

t+--+ Lap nghiem voi 5 lam lap +--+

Lan lap	x_n	f(x_n)
1	x = -1.02012947	f(x) = 11.78852497
2	x = -1.04533213	f(x) = 1.88854806
3	x = -1.04928069	f(x) = 0.29218842
4	x = -1.04988947	f(x) = 0.04496081
5	x = -1.04998310	f(x) = 0.00691259

Vay nghiem x\* = -1.049983097866

So lan lap: 5

Sai so 1: 0.000024817654

Sai so 2: 0.000094100789

```
D:\CODE_KTLT\BTLKTLT\chotctrinh.exe
MENU
1. Khoảng phân li thu 1: (-1.225, -0.874)
2. Khoảng phân li thu 2: (1.229, 1.579)
3. Khoảng phân li thu 3: (2.631, 2.981)
4. Khoảng phân li thu 4: (3.682, 4.033)
5. Khoảng phân li thu 5: (7.187, 7.538)

D:\CODE_KTLT\BTLKTLT\chotctrinh.exe
Nhap sai so "e" mong muon: 0.01

      +-+-+ Lap nghiem theo sai so 1 +-+-+
+-----+-----+-----+
| Lan lap | x_n      | f(x_n) |
+-----+-----+-----+
| 1        | x = 1.40438471 | f(x) = -1.39883857 |
+-----+-----+-----+
| 2        | x = 1.38257207 | f(x) = -0.15017309 |
+-----+-----+-----+

Vay nghiem x* = 1.3825720700
So lan lap: 2

      +-+-+ Lap nghiem theo sai so 2 +-+-+
+-----+-----+-----+
| Lan lap | x_n      | f(x_n) |
+-----+-----+-----+
| 1        | x = 1.40438471 | f(x) = -1.39883857 |
+-----+-----+-----+

Vay nghiem x* = 1.4043847118
So lan lap: 1
```

```
D:\CODE_KTLT\BTLKTLT\chotctrinh.exe
MENU
1. Khoảng phân li thu 1: (-1.225, -0.874)
2. Khoảng phân li thu 2: (1.229, 1.579)
3. Khoảng phân li thu 3: (2.631, 2.981)
4. Khoảng phân li thu 4: (3.682, 4.033)
5. Khoảng phân li thu 5: (7.187, 7.538)

D:\CODE_KTLT\BTLKTLT\chotctrinh.exe
Nhap sai so "e" mong muon: 0.001

Lap nghiem thoa man  $|X\{n\} - X\{n-1\}| < 0.001000000$ 
+-----+-----+-----+
| Lan lap | x_n | f(x_n) |
+-----+-----+-----+
| 1 | x = 2.95105371 | f(x) = 0.02948374 |
+-----+-----+-----+
| 2 | x = 2.95003668 | f(x) = 0.00079024 |
+-----+-----+-----+
| 3 | x = 2.95000942 | f(x) = 0.00002101 |
+-----+-----+-----+

Vay nghiem x* = 2.9500094206
So lan lap: 3
```



Kết quả lưu trong file output:

Bieu thuc:

$$1.000000x^5 + -14.510000x^4 + 65.278900x^3 \\ +-85.243200x^2 +-61.754700x^1 + 123.620000$$

Khoang phan ly:

STT	a	b
1	-1.575	-0.874
2	1.229	1.930
3	2.631	3.332
4	3.332	4.033
5	6.837	7.538

Voi khoang phan ly nghiem (p,q) voi  $|p-q| \leq m$ . Nhap vao so m:

m = 0.5

Khoang phan ly sau khi thu hep:

STT	a	b
1	-1.225	-0.874
2	1.229	1.579
3	2.631	2.981
4	3.682	4.033
5	7.187	7.538

1. Khoang phan li thu 1: (-1.225, -0.874)
2. Khoang phan li thu 2: (1.229, 1.579)
3. Khoang phan li thu 3: (2.631, 2.981)
4. Khoang phan li thu 4: (3.682, 4.033)
5. Khoang phan li thu 5: (7.187, 7.538)

Lua chon: 1

Phan 3:

Nhap so lan lap: 5

+--+ Lap nghiem voi 5 lan lap +--+

+-----+	+-----+	+-----+
Lan lap	x_n	f(x_n)
+-----+	+-----+	+-----+
1	x = -1.02012947	f(x) = 11.78852497
+-----+	+-----+	+-----+
2	x = -1.04533213	f(x) = 1.88854806
+-----+	+-----+	+-----+
3	x = -1.04928069	f(x) = 0.29218842
+-----+	+-----+	+-----+
4	x = -1.04988947	f(x) = 0.04496081
+-----+	+-----+	+-----+
5	x = -1.04998310	f(x) = 0.00691259
+-----+	+-----+	+-----+

Vay nghiem x\* = -1.0499830979

So lan lap: 5

Sai so 1: 0.000024817654

Sai so 2: 0.000094100789

+--+ +-----+ +--+
1. Khoang phan li thu 1: (-1.225, -0.874)
2. Khoang phan li thu 2: (1.229, 1.579)
3. Khoang phan li thu 3: (2.631, 2.981)
4. Khoang phan li thu 4: (3.682, 4.033)
5. Khoang phan li thu 5: (7.187, 7.538)
+--+ +-----+ +--+

Lua chon: 2

Phan 4:

Nhap sai so "e" mong muon: 0.01

+--+ Lap nghiem theo sai so 1 +--+

+-----+	+-----+	+-----+
Lan lap	x_n	f(x_n)
+-----+	+-----+	+-----+
1	x = 1.40438471	f(x) = -1.39883857
+-----+	+-----+	+-----+
2	x = 1.38257207	f(x) = -0.15017309
+-----+	+-----+	+-----+

Vay nghiem x\* = 1

So lan lap: 2

+---+ Lap nghiem theo sai so 2 +---+

+-----+   -----+   -----+ +
Lan lap   x_n   f(x_n)
+-----+   -----+   -----+ +
1   x = 1.40438471   f(x) = -1.39883857
+-----+   -----+   -----+ +

Vay nghiem x\* = 1.404385

So lan lap: 1

+---+   -----+ +
1. Khoang phan li thu 1: (-1.225, -0.874)
2. Khoang phan li thu 2: (1.229, 1.579)
3. Khoang phan li thu 3: (2.631, 2.981)
4. Khoang phan li thu 4: (3.682, 4.033)
5. Khoang phan li thu 5: (7.187, 7.538)
+---+   -----+ +

Lua chon: 3

Phan 5:

Nhap sai so "e" mong muon: 0.001

Lap nghiem thoa man  $|X\{n\} - X\{n-1\}| < 0.001$

+-----+   -----+   -----+ +
Lan lap   x_n   f(x_n)
+-----+   -----+   -----+ +
1   x = 2.95105371   f(x) = 0.02948374
+-----+   -----+   -----+ +
2   x = 2.95003668   f(x) = 0.00079024
+-----+   -----+   -----+ +
3   x = 2.95000942   f(x) = 0.00002101
+-----+   -----+   -----+ +

Vay nghiem x\* = 2.9500094206

So lan lap: 3

## Lời kết

- Bằng cách sử dụng những kiến thức đã được học tại học phần này cũng như các học phần bổ trợ như: Cấu trúc dữ liệu và giải thuật, giải tích số, chúng em đã phân tích đề bài và cho ra được chương trình xử lý như trên.
- Mặc dù chương trình đã xử lý được phần nào các bộ dữ liệu chúng em nhập vào nhưng do một vài yếu tố như thuật toán chưa chặt chẽ ( thuật toán vét cạn của tìm khoảng phân ly nghiệm ... ) nên có thể trong lúc xử lý các bộ dữ liệu đặc biệt sẽ xảy ra sai sót.
- Học kì vừa qua do tình hình dịch bệnh phức tạp, chúng em xin cảm ơn cô đã tận tình truyền đạt kiến thức, chỉ bảo để chúng em có thể hoàn thành được bài tập lớn cũng như học phần này.
- Chúng em xin chúc cô và gia đình mạnh khỏe, công tác tốt và có nhiều niềm vui trong cuộc sống !

Trên đây là bài báo cáo cuối kì môn học Kỹ thuật lập trình của nhóm chúng em.