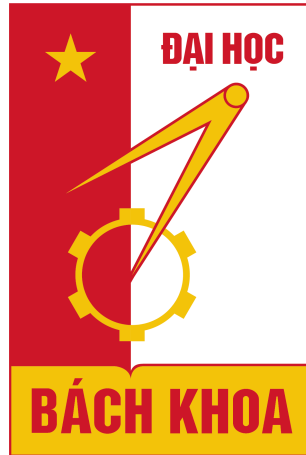


**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**KHOA TOÁN - TIN**  
—o0o—



**BÀI TẬP LỚN CUỐI KÌ**

**MÔN KỸ THUẬT LẬP TRÌNH**

***ĐỀ TÀI***

**GIẢI GẦN ĐÚNG PHƯƠNG TRÌNH ĐA THỨC  
BẰNG PHƯƠNG PHÁP DÂY CUNG**

Sinh viên thực hiện: Nguyễn Tất Cường  
MSSV: 20227090

Giảng viên hướng dẫn: TS. Nguyễn Thị Thanh Huyền

Hà Nội , 6/2024

## **Lời nói đầu**

Lời đầu tiên, em xin gửi lời cảm ơn đến Khoa Toán - Tin, Đại học Bách khoa Hà Nội đã tạo cơ hội cho em thực hiện báo cáo bài tập lớn trong một môi trường tốt nhất nhằm đáp ứng yêu cầu hoàn thành báo cáo của em.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến giảng viên TS.Nguyễn Thị Thanh Huyền, người đã giảng dạy, hướng dẫn và hỗ trợ em rất nhiều trong suốt thời gian qua.

Cuối cùng, em xin gửi lời cảm ơn tới các anh chị cựu sinh viên Khoa Toán - Tin, sự hỗ trợ, hợp tác của bạn bè đã giúp đỡ và truyền đạt những kinh nghiệm quý báu cho quá trình học tập của em.

Do kiến thức còn hạn chế nên báo cáo của em hoàn toàn không thể tránh khỏi những thiếu sót và sai sót. Vì vậy, em rất mong nhận được một số ý kiến đóng góp của cô giáo.

**Trân trọng cảm ơn  
Người báo cáo  
Nguyễn Tất Cường**

## Mở đầu

Trong lĩnh vực giải tích số, việc tìm nghiệm của phương trình phi tuyến  $f(x) = 0$  đóng vai trò vô cùng quan trọng và được ứng dụng rộng rãi trong nhiều bài toán khoa học và kỹ thuật. Phương pháp dây cung là một trong những kỹ thuật hiệu quả để tìm nghiệm gần đúng của các phương trình này.

Phương pháp dây cung dựa trên ý tưởng sử dụng đoạn thẳng nối hai điểm khác nhau trên đồ thị của hàm số để tiếp cận nghiệm. Cụ thể, nếu ta biết hai giá trị , ta có thể xây dựng đoạn thẳng cắt đồ thị và xác định điểm cắt với trục hoành làm điểm xấp xỉ tiếp theo. Quy trình này được lặp lại cho đến khi ta đạt được nghiệm gần đúng thỏa mãn yêu cầu chính xác cho trước.

Phương pháp dây cung không chỉ dễ hiểu và dễ triển khai mà còn mang lại hiệu suất tính toán cao, đặc biệt trong các trường hợp mà việc tính đạo hàm là phức tạp hoặc tốn kém. Chính vì vậy, nó là một công cụ hữu ích và mạnh mẽ trong bộ công cụ của nhà toán học và kỹ sư.

Trong bài toán này, hàm  $f(x)$  mà ta cần giải có dạng đa thức  $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ .

## Mục lục

1	Bài toán . . . . .	4
2	Nhắc lại một số công thức . . . . .	6
3	Thiết kế chương trình theo phương pháp tinh chỉnh từng bước . . . . .	8
4	Kết quả thực hiện chương trình . . . . .	30
5	Kết luận, đánh giá kết quả đạt được . . . . .	42
5.1	Đánh giá chung . . . . .	42
5.2	Đánh giá chi tiết . . . . .	42

## 1 Bài toán

### **Yêu cầu bài toán:**

Tìm nghiệm của phương trình  $f(x) = 0$  với số lần lặp xác định hoặc sai số cho trước .

### **Nhắc lại về khoảng phân ly nghiệm:**

Khoảng  $(a, b)$  được gọi là khoảng phân ly nghiệm của phương trình  $f(x) = 0$  nếu trong khoảng  $(a, b)$  có đúng một nghiệm của phương trình.

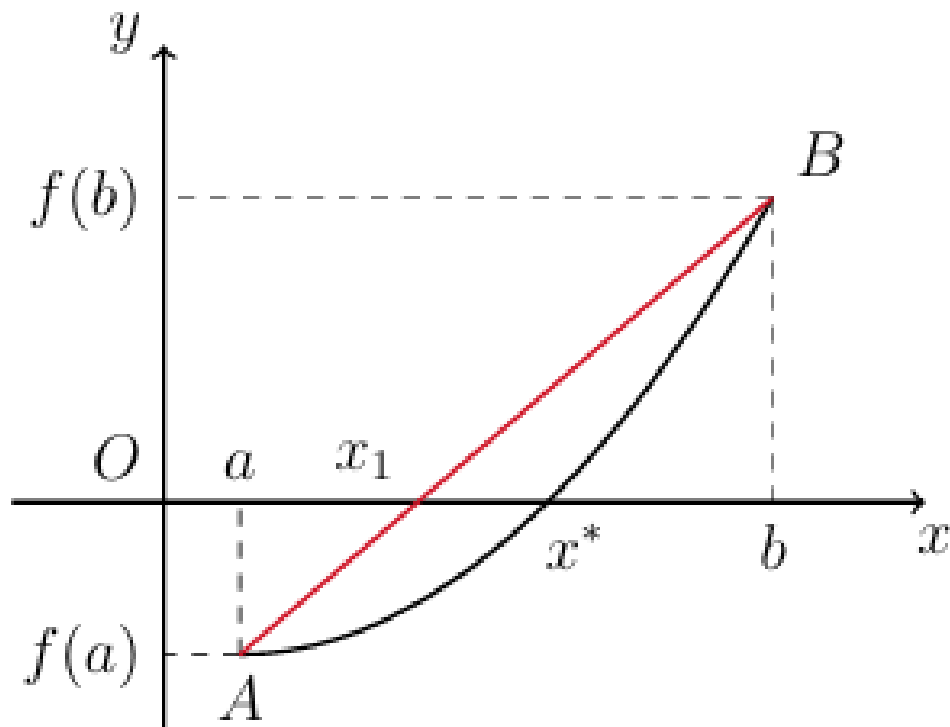
### **Phương pháp tìm khoảng phân ly nghiệm**

- Phương pháp khảo sát hàm số (lập bảng biến thiên)
- Phương pháp vẽ đồ thị hàm số

### **Ý tưởng cho phương pháp dây cung:**

Giả sử  $(a, b)$  là khoảng phân ly nghiệm  $x$  của phương trình thì  $f(x)$  là hàm số liên tục và có đạo hàm liên tục trên  $(a, b)$ . Sau đó ta sẽ khảo sát đồ thị hàm số  $y = f(x)$  trên đoạn  $(a, b)$ . Ta sẽ tiến hành khảo sát nhanh đường cong  $y = f(x)$  trên đoạn  $(a, b)$  và tiến hành các bước như sau:

- **Bước 1 :** Trên mặt phẳng tọa độ lấy 2 điểm  $A(a, f(a))$  và  $B(b, f(b))$
- **Bước 2 :** Vẽ phương trình đường thẳng đi qua 2 điểm, tìm giao điểm  $D(x_d, y_d)$  của đường thẳng với trục hoành, nếu  $f(x_d)$  cùng dấu  $f(a)$  thì cập nhật khoảng phân ly là  $(x_d, b)$ , ngược lại thì cập nhật khoảng phân ly là  $(a, x_d)$
- **Bước 3 :** Lặp lại quá trình trên đến khi thỏa mãn điều kiện về sai số hoặc điều kiện về số lần lặp



Hình 1: Mô tả thuật toán dây cung

## 2 Nhắc lại một số công thức

**Công thức lặp :**

$$x_n = x_{n-1} - \frac{f(x_{n-1})(x_{n-1} - x_d)}{f(x_{n-1}) - f(x_d)} \quad (1)$$

**Công thức đánh giá sai số :**

$$|x_n - x^*| \leq \frac{|f(x_n)|}{m1} \quad (2)$$

$$|x_n - x^*| \leq \frac{(M1 - m1)|x_n - x_{n-1}|}{m1} \quad (3)$$

Trong đó :

$$M1 = \max_{x \in (a,b)} |f'(x)| \quad (4)$$

$$m1 = \min_{x \in (a,b)} |f'(x)| \quad (5)$$

**Công thức tính bán kính nghiệm :**

$$R = 1 + \max_{i=0}^n \left( \frac{|a_i|}{|a_0|} \right) \quad (6)$$

Trong đó :  $a_i$  là các hệ số của đa thức,  $n$  là bậc của  $f(x)$

## Điều kiện thực hiện phương pháp dây cung

- Hàm số đơn điệu trên khoảng cách li, điều này tương đương :

$$\min_{x \in (a,b)} f'(x) \cdot \max_{x \in (a,b)} f'(x) > 0 \quad (7)$$

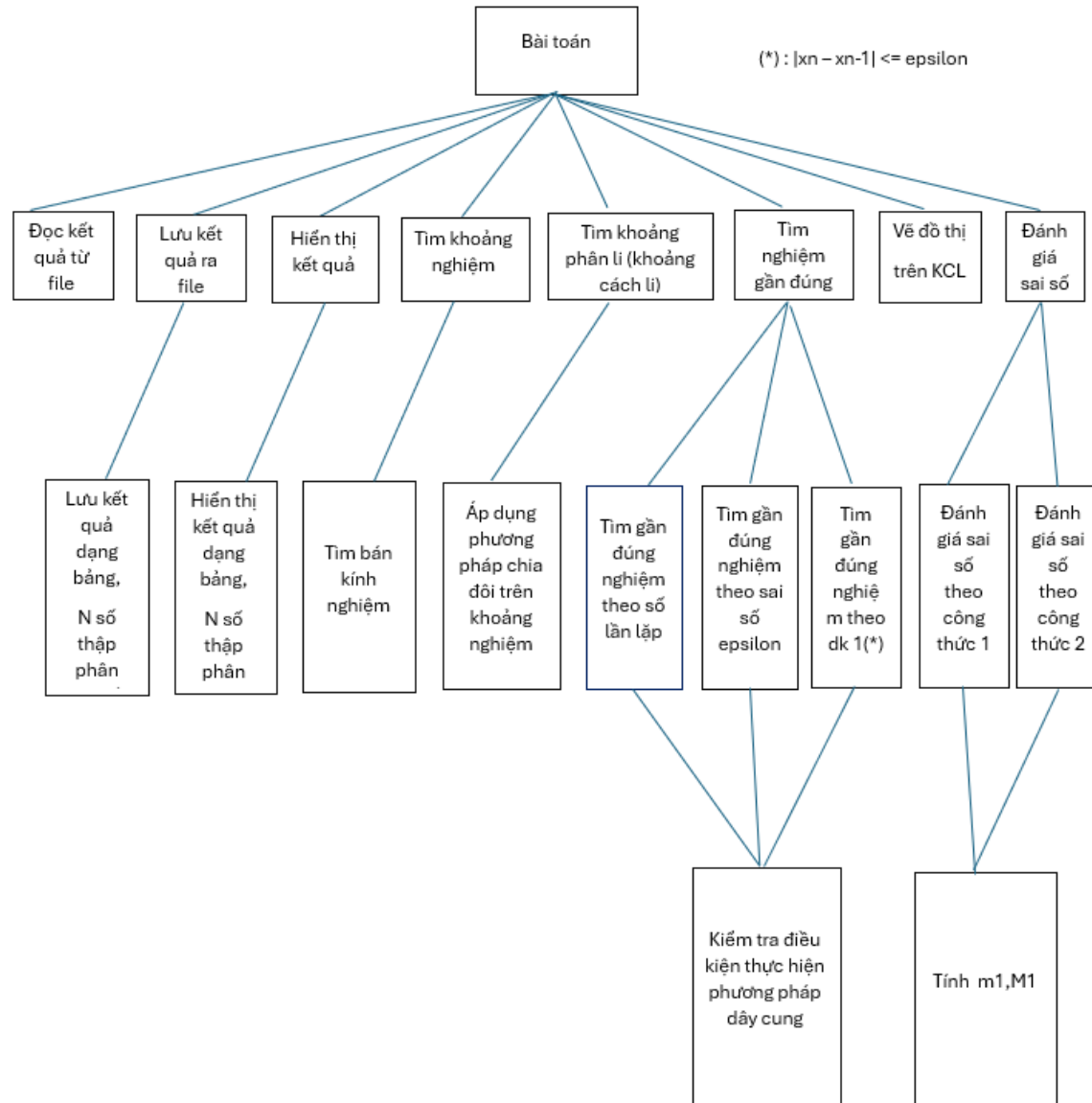
- Hàm số không thay đổi tính lồi/lõm trên khoảng cách li, điều này tương đương :

$$\min_{x \in (a,b)} f''(x) \cdot \max_{x \in (a,b)} f''(x) > 0 \quad (8)$$



### 3 Thiết kế chương trình theo phương pháp tinh chỉnh từng bước

#### Bước 0



Hình 2: Sơ đồ chương trình theo phương pháp tinh chỉnh dần

## Bước 1 : Chức năng nhập $f(x)$

-Input : Các hệ số của đa thức  $f(x)$

-Output : Mảng các hệ số  $f(x)$

-Ý tưởng : Lưu các hệ số của đa thức sau mỗi lần nhập vào 1 vector

-Mã giả :

```
1 Function fx(a, x):
2     n = size of array a
3     result = 0
4     for i from 0 to n-1 do:
5         result = result + (a[i] * (x ^ (n - 1 - i)))
6     return result
```

-Code C++ :

```
1 // Input for f(x)
2 void infuction(vector<double>& a,int n){
3     for i = 0 to n do {
4         cout<<"Nhập vào hệ số của x^"<< n-i;
5         double x;cin >> x;
6         a.push_back(x);
7         cout << endl;
8     }
9 }
```

## Bước 2 : Chức năng tính bán kính nghiệm

-Input : Các hệ số của đa thức

-Output : Giá trị bán kính nghiệm

-Ý tưởng : Thực hiện tính theo công thức

-Mã giả :

```
1 Function range(a):
2     n = size of array a
3     max = abs(a[0]) / abs(a[0])
4     for i from 0 to n-1 do:
5         if (abs(a[i]) / abs(a[0]) > max):
6             max = abs(a[i]) / abs(a[0])
7     return 1 + max
```

-Code C++

```
1 // Caculate for range (a,b)
2 double range(vector<double>& a) {
3     int n = a.size();
4     double max = abs(a[0])/abs(a[0]);
5     for i = 0 to n-1 do {
6         if (abs(a[i])/abs(a[0]) > max){
7             max = abs(a[i])/abs(a[0]);
8         }
9     }
10    return 1+max;
11 }
```

### Bước 3 : Chức năng tính $f(x)$ , $f'(x)$ , $f''(x)$ tại giá trị $x$

-Input : Các hệ số của đa thức, giá trị  $x$

-Output : Giá trị của  $f(x)$ ,  $f'(x)$ ,  $f''(x)$  tại  $x$

-Ý tưởng : Tính toán theo công thức

-Mã giả

```
1 Function fx(a, x):
2     n = size of array a
3     result = 0
4     for i from 0 to n-1 do:
5         result = result + (a[i] * (x ^ (n - 1 - i)))
6     return result
7
8 Function derivative1(a, x):
9     n = size of array a
10    result = 0
11    for i from 0 to n-1 do:
12        result = result + ((n - i - 1) * a[i] * (x ^ (n - i - 2)))
13    return result
14
15 Function derivative2(a, x):
16    n = size of array a
17    result = 0
18    for i from 0 to n-1 do:
19        result = result + ((n - i) * (n - i - 1) * a[i] * (x ^ (n - i - 2)))
20    return result
```

-Code C++ :

```
1 // Value of f(x) at a point
2 double fx(vector<double>& a, double x) {
3     int n = a.size();
4     double result = 0;
5     for i=0 to n-1 do {
6         result += a[i] * pow(x, n - 1 - i);
7     }
8     return result;
9 }
10 // Value of f'(x) at a point
11 double derivative1(vector<double>& a, double x) {
12     int n = a.size();
13     double result = 0;
14     for i=0 to n-1 do {
15         result += (n - i - 1) * a[i] * pow(x, n - i - 2);
16     }
17     return result;
18 }
19 // Value of f''(x) at a point
20 double derivative2(vector<double>& a, double x) {
21     int n=a.size();
22     double result=0;
23     for i=0 to n-1 do {
24         result += (n-i)*(n-i-1)*a[i]*pow(x,n-i-2);
25     }
26     return result;
27 }
```

## Bước 4 : Chức năng tìm khoảng cách li

-Input : Các hệ số của đa thức

-Output : Khoảng cách li của phương trình sau mỗi lần lặp của  $f(x)=0$

-Ý tưởng : Dùng phương pháp chia đôi để tìm khoảng cách li sau mỗi lần lặp và lưu kết quả cuối cùng vào vector

-Mã giả :

```
1 Function interval(itv, a, eps, decimal, choose, filename):
2     r = range(a)
3     left = -r
4     right = r
5
6     if choose == 1:
7         i = 1
8         while left < right and abs(left - right) > eps:
9             mid = (left + right) / 2
10            output_interval(decimal, left, right, i)
11            if fx(a, mid) == 0:
12                if fx(a, mid - 0.00000001) * fx(a, mid + 0.00000001) < 0:
13                    print "Khoang cach li nghiem la (" + format(mid - 0.1, decimal) + " , "
14                        + format(mid + 0.1, decimal) + ")"
15                    itv.clear()
16                    itv.push_back(mid - 0.1)
17                    itv.push_back(mid + 0.1)
18                else:
19                    print "Khong ton tai khoang cach li tren mien chua nghiem"
20                    return
21            else if fx(a, mid) * fx(a, left) > 0 or fx(a, mid) * fx(a, right) < 0:
22                left = mid
23            else if fx(a, mid) * fx(a, right) > 0 or fx(a, mid) * fx(a, left) < 0:
24                right = mid
25            i += 1
26
27        if fx(a, left) * fx(a, right) < 0:
28            itv.clear()
29            itv.push_back(left)
30            itv.push_back(right)
31            print "Khoang cach li nghiem la (" + format(left, decimal) + " , " + format(
32                right, decimal) + ")"
33        else:
34            print "Khong ton tai khoang cach li tren mien chua nghiem"
35
36    else if choose == 2:
37        outfile = open(filename, 'w')
38        i = 1
39        if not outfile:
40            print "Khong the mo file"
41        while left < right and abs(left - right) > eps:
42            mid = (left + right) / 2
43            export_interval(decimal, left, right, outfile, i)
44            if fx(a, mid) == 0:
45                if fx(a, mid - 0.00000001) * fx(a, mid + 0.00000001) < 0:
46                    outfile.write("Khoang cach li nghiem la (" + format(mid - 0.1,
47                        decimal) + " , " + format(mid + 0.1, decimal) + ")\n")
48                    itv.clear()
49                    itv.push_back(mid - 0.1)
50                    itv.push_back(mid + 0.1)
```

```

48         else:
49             outfile.write("Khong ton tai khoang cach li tren mien chua nghiem\n")
50             return
51         else if fx(a, mid) * fx(a, left) > 0 or fx(a, mid) * fx(a, right) < 0:
52             left = mid
53         else if fx(a, mid) * fx(a, right) > 0 or fx(a, mid) * fx(a, left) < 0:
54             right = mid
55         i += 1
56
57     if fx(a, left) * fx(a, right) < 0:
58         itv.clear()
59         itv.push_back(left)
60         itv.push_back(right)
61         outfile.write("Khoang cach li nghiem la (" + format(left, decimal) + " , " +
62             format(right, decimal) + ")\n")
63     else:
64         outfile.write("Khong ton tai khoang cach li tren mien chua nghiem\n")
65         outfile.close()
66
67     else if choose == 3:
68         read_file(filename)
69
70     else:
71         while left < right and abs(left - right) > eps:
72             mid = (left + right) / 2
73             if fx(a, mid) == 0:
74                 if fx(a, mid - 0.00000001) * fx(a, mid + 0.00000001) < 0:
75                     itv.clear()
76                     itv.push_back(mid - 0.1)
77                     itv.push_back(mid + 0.1)
78                 else:
79                     print "Khong ton tai khoang cach li"
80                     return
81             else if fx(a, mid) * fx(a, left) > 0 or fx(a, mid) * fx(a, right) < 0:
82                 left = mid
83             else if fx(a, mid) * fx(a, right) > 0 or fx(a, mid) * fx(a, left) < 0:
84                 right = mid
85             itv.clear()
86             itv.push_back(left)
87             itv.push_back(right)
88             delete_file(filename)
89         return

```

## -Code C++ :

```
1 void interval(vector<double>& itv, vector<double>& a, double eps, int decimal, int choose
2 , string filename){
3     double r=range(a);
4     double left=-r; double right=r;
5     if (choose == 1){
6         int i=1;
7         while(left<right && abs(left-right)>eps){
8             double mid=(left+right)/2;
9             output_interval(decimal, left, right, i);
10            if (fx(a,mid) == 0){
11                if (fx(a,mid-0.00000001)*fx(a,mid+0.00000001)<0){
12                    cout<<"Khoang cach li nghiem la ("<<fixed<<setprecision(decimal)<<mid
13                    -0.1<<" , "<<fixed<<setprecision(decimal)<<mid+0.1<<" "<<endl;
14                    itv.clear();
15                    itv.push_back(mid-0.1);
16                    itv.push_back(mid+0.1);
17                }
18                else{
19                    cout<<"Khong ton tai khoang cach li tren mien chua nghiem"<<endl;
20                }
21                return;
22            }
23            else if (fx(a,mid)*fx(a,left)>0 || fx(a,mid)*fx(a,right)<0)
24            {
25                left=mid;
26            }
27            else if (fx(a,mid)*fx(a,right)>0 || fx(a,mid)*fx(a,left)<0){
28                right=mid;
29            }
30            i+=1;
31        }
32
33        if (fx(a,left)*fx(a,right) < 0){
34            itv.clear();
35            itv.push_back(left);
36            itv.push_back(right);
37            cout<<"Khoang cach li nghiem la ("<<fixed<<setprecision(decimal)<<left<<" , "
38            <<fixed<<setprecision(decimal)<<right<<" "<<endl;
39        }
40        else{
41            cout<<"Khong ton tai khoang cach li tren mien chua nghiem"<<endl;
42        }
43    }
44    else if (choose == 2){
45        ofstream outfile(filename);
46        int i=1;
47        if (!outfile){
48            cout<<"Khong the mo file"<<endl;
49        }
50        while(left<right && abs(left-right)>eps){
51            double mid=(left+right)/2;
52            export_interval(decimal, left, right, outfile, i);
53            if (fx(a,mid) == 0){
54                if (fx(a,mid-0.00000001)*fx(a,mid+0.00000001)<0){
55                    outfile<<"Khoang cach li nghiem la ("<<fixed<<setprecision(decimal)<<
56                    mid-0.1<<" , "<<fixed<<setprecision(decimal)<<mid+0.1<<" "<<endl;
57                    itv.clear();
58                    itv.push_back(mid-0.1);
59                    itv.push_back(mid+0.1);
60                }
61                else{
62                    outfile<<"Khong ton tai khoang cach li tren mien chua nghiem"<<endl;
63                }
64            }
65            return;
66        }
67    }
68 }
```

```

62         }
63         else if (fx(a,mid)*fx(a,left)>0 || fx(a,mid)*fx(a,right)<0)
64         {
65             left=mid;
66         }
67         else if ((fx(a,mid)*fx(a,right)>0) || (fx(a,mid)*fx(a,left)<0)){
68             right=mid;
69         }
70         i+=1;
71     }
72     if (fx(a,left)*fx(a,right) < 0){
73         itv.clear();
74         itv.push_back(left);
75         itv.push_back(right);
76         outfile<<"Khoang cach li nghiêm la ("<<fixed<<setprecision(decimal)<<left<<"
77             , "<<fixed<<setprecision(decimal)<<right<<")"<<endl;
78     }
79     else{
80         outfile<<"Khong ton tai khoang cach li tren mien chua nghiêm"<<endl;
81     }
82     outfile.close();
83 }
84 else if (choose == 3){
85     read_file(filename);
86 }
87 else{
88     while(left<right && abs(left-right)>eps){
89         double mid=(left+right)/2;
90         if (fx(a,mid) == 0){
91             if (fx(a,mid-0.00000001)*fx(a,mid+0.00000001)<0){
92                 itv.clear();
93                 itv.push_back(mid-0.1);
94                 itv.push_back(mid+0.1);
95             }
96             else{
97                 cout<<"Khong ton tai khoang cach li"<<endl;
98             }
99             return;
100         }
101     }
102     else if (fx(a,mid)*fx(a,left)>0 || fx(a,mid)*fx(a,right)<0)
103     {
104         left=mid;
105     }
106     else if (fx(a,mid)*fx(a,right)>0 || fx(a,mid)*fx(a,left)<0){
107         right=mid;
108     }
109     itv.clear();
110     itv.push_back(left);
111     itv.push_back(right);
112 }
113 delete_file(filename);
114 }
115 }
116 }

```

## Bước 5 : Chức năng tìm m1, M1

-Input : Các hệ số của đa thức, khoảng cách li

-Output : Giá trị m1, M1

-Ý tưởng : Lặp các điểm trên khoảng cách li với bước lặp rất nhỏ để tính toán  $f'(x)$ , sau đó chọn m1, M1 lần lượt là các giá trị lớn nhất và nhỏ nhất

-Mã giả :

```
1 Function m1_M1(a, itv, m1M1):  
2     d1 = empty list  
3     for i from itv[0] to itv[1] step 0.00000001:  
4         d1.push_back(abs(derivativel(a, i)))  
5     min1 = min_element(d1)  
6     max1 = max_element(d1)  
7     m1M1.push_back(min1)  
8     m1M1.push_back(max1)
```

-Code C++ :

```
1 void m1_M1(vector<double>& a, vector<double>& itv, vector<double>& m1M1){  
2     vector<double> d1;  
3     for(double i=itv[0]; i<itv[1]; i+=0.00000001){  
4         d1.push_back(abs(derivativel(a,i)));  
5     }  
6     auto min1=min_element(d1.begin(), d1.end());  
7     auto max1=max_element(d1.begin(), d1.end());  
8     m1M1.push_back(*min1);  
9     m1M1.push_back(*max1);  
10 }
```



## Bước 6 : Chức năng kiểm tra điều kiện thực hiện phương pháp dây cung

-Input : Các hệ số của đa thức, khoảng cách li

-Output : Kết quả đúng/sai của khi kiểm tra điều kiện của phương pháp dây cung

-Ý tưởng : Tìm min, max của các đạo hàm bậc 1, bậc 2 trên khoảng cách li

-Mã giả :

```
1 Function check_condition(a, itv):
2     d1 = empty list
3     d2 = empty list
4     for i from itv[0] to itv[1] step 0.000001:
5         d1.push_back(derivative1(a, i))
6         d2.push_back(derivative2(a, i))
7     max_d1 = max_element(d1)
8     min_d1 = min_element(d1)
9     max_d2 = max_element(d2)
10    min_d2 = min_element(d2)
11
12    if ((max_d1 * min_d1 > 0) and (max_d2 * min_d2 > 0)):
13        return True
14    return False
```

-Code C++ :

```
1 bool check_condition(vector<double>& a, vector<double>& itv){
2     vector<double> d1; vector<double> d2;
3     for(double i=itv[0]; i<itv[1]; i+=0.000001){
4         d1.push_back(derivative1(a,i));
5         d2.push_back(derivative2(a,i));
6     }
7     auto max_d1=max_element(d1.begin(), d1.end());
8     auto min_d1=min_element(d1.begin(), d1.end());
9     auto max_d2=max_element(d2.begin(), d2.end());
10    auto min_d2=min_element(d2.begin(), d2.end());
11
12    if (((*max_d1)*(*min_d1)>0) && ((*max_d2)*(*min_d2)>0)) return 1;
13    return 0;
14 }
```

## Bước 7 : Chức năng giải gần đúng (Theo số lần lặp)

-Input : Các hệ số của đa thức, khoảng cách li

-Output : Kết quả  $x$  sau mỗi lần lặp

-Ý tưởng : Dùng phương pháp dây cung để tìm nghiệm sau mỗi lần lặp

-Mã giả :

```
1 Function solution1(a, itv, iter, decimal, choose, filename):
2     if choose == 1:
3         if size of a == 2:
4             print "Phương trình bậc 1 có nghiệm duy nhất", -a[1] / a[0] with decimal
               places
5             return
6         if size of a == 1:
7             print "Ham hang"
8             return
9         if check_condition(a, itv):
10            print "Thực hiện phương pháp dây cung..."
11            x, d, tmp = 0, 0, 0
12            m1M1 = vector of size 2
13            call m1_M1(a, itv, m1M1)
14            if derivative2(a, (itv[0] + itv[1]) / 2) * fx(a, x) > 0:
15                cnt = 1
16                d = itv[0]
17                x = itv[1]
18                while cnt <= iter:
19                    tmp = x
20                    x = x + dx(a, x, d)
21                    call output_solution(decimal, x, cnt)
22                    cnt += 1
23            else:
24                cnt = 1
25                d = itv[1]
26                x = itv[0]
27                while cnt <= iter:
28                    tmp = x
29                    x = x + dx(a, x, d)
30                    call output_solution(decimal, x, cnt)
31                    cnt += 1
32            print "Giá trị m1 = min(|f(x)|) trên đoạn", itv[0], ",", itv[1], "là:", m1M1
               [0] with decimal places
33            print "Giá trị M1 = max(|f(x)|) trên đoạn", itv[0], ",", itv[1], "là:", m1M1
               [1] with decimal places
34            print "Tính theo công thức 1, sai số |xn-x*| <=", abs(fx(a, x)) / m1M1[0]
               with decimal places
35            print "Tính theo công thức 2, sai số |xn-x*| <=", (m1M1[1] - m1M1[0]) / m1M1
               [0] * abs(x - tmp) with decimal places
36        else:
37            print "Không thỏa mãn điều kiện phương pháp dây cung"
38
39    else if choose == 2:
40        open file with name filename for writing as outfile
41        if size of a == 2:
42            write to outfile "Phương trình bậc 1 có nghiệm duy nhất", -a[1] / a[0] with
               decimal places
43            return
44        if size of a == 1:
45            write to outfile "Ham hang"
46            return
47        if check_condition(a, itv):
```

```

48     print "Thuc hien phuong phap day cung..."
49     x, d, tmp = 0, 0, 0
50     m1M1 = vector of size 2
51     call ml_M1(a, itv, m1M1)
52     if derivative2(a, (itv[0] + itv[1]) / 2) * fx(a, x) > 0:
53         cnt = 1
54         d = itv[0]
55         x = itv[1]
56         while cnt <= iter:
57             tmp = x
58             x = x + dx(a, x, d)
59             call export_solution(decimal, x, cnt, outfile)
60             cnt += 1
61     else:
62         cnt = 1
63         d = itv[1]
64         x = itv[0]
65         while cnt <= iter:
66             tmp = x
67             x = x + dx(a, x, d)
68             call export_solution(decimal, x, cnt, outfile)
69             cnt += 1
70     write to outfile "Gia tri m1 = min(|f(x)|) tren doan", itv[0], ",", itv[1], "
71     la:", m1M1[0] with decimal places
72     write to outfile "Gia tri M1 = max(|f(x)|) tren doan", itv[0], ",", itv[1], "
73     la:", m1M1[1] with decimal places
74     write to outfile "Tinh theo cong thuc 1, sai so |xn-x*| <=", abs(fx(a, x)) /
75     m1M1[0] with decimal places
76     write to outfile "Tinh theo cong thuc 2, sai so |xn-x*| <=", (m1M1[1] - m1M1
77     [0]) / m1M1[0] * abs(x - tmp) with decimal places
78
79     else:
80         write to outfile "Khong thoa man dieu kien phuong phap day cung"
81         close outfile
82
83     else if choose == 3:
84         call read_file(filename)
85
86     else:
87         call delete_file(filename)

```

## -Code C++ :

```

1  //Solution with n iterations
2  void solution1(vector<double>& a, vector<double>& itv, int iter, int decimal, int choose,
3  string filename){
4      if (choose == 1){
5          if (a.size() == 2){
6              cout<<"Phuong trinh bac 1 co nghiem duy nhat "<<fixed<<setprecision(decimal)
7              <<-a[1]/a[0]<<endl;
8              return;
9          }
10         if (a.size () == 1){
11             cout<<"Ham hang"<<endl;
12             return;
13         }
14         if (check_condition(a,itv)){
15             cout<<"Thuc hien phuong phap day cung..."<<endl;
16             double x,d,tmp;
17             vector<double> m1M1;
18             ml_M1(a, itv, m1M1);
19             if(derivative2(a, (itv[0]+itv[1])/2)*fx(a,x)>0){
20                 int cnt=1;
21                 d=itv[0];
22                 x=itv[1];
23                 while(cnt <= iter){
24                     tmp=x;

```

```

23         x=x+dx(a,x,d);
24         output_solution(decimal, x, cnt);
25         cnt+=1;
26     }
27
28 }
29 else{
30     int cnt=1;
31     d=itv[1];
32     x=itv[0];
33     while(cnt <= iter){
34         tmp=x;
35         x=x+dx(a,x,d);
36         output_solution(decimal, x, cnt);
37         cnt+=1;
38     }
39 }
40 cout<<"Gia tri m1 = min(|f(x)| tren doan ("<<fixed<<setprecision(decimal)<<
    itv[0]<<" , "<<fixed<<setprecision(decimal)<<itv[1]<<") la : "
41 <<fixed<<setprecision(decimal)<<m1M1[0]<<endl;
42 cout<<"Gia tri M1 = max(|f(x)| tren doan ("<<fixed<<setprecision(decimal)<<
    itv[0]<<" , "<<fixed<<setprecision(decimal)<<itv[1]<<") la : "
43 <<fixed<<setprecision(decimal)<<m1M1[1]<<endl;
44 cout<<"Tinh theo cong thuc 1, sai so |xn-x*| <= " <<fixed<<setprecision(
    decimal)<<abs(fx(a, x))/m1M1[0]<<endl;
45 cout<<"Tinh theo cong thuc 2, sai so |xn-x*| <= " <<fixed<<setprecision(
    decimal)<<(m1M1[1]-m1M1[0])/m1M1[0]*abs(x-tmp)<<endl;
46 }
47 else{
48     cout<<"Khong thoa man dieu kien phuong phap day cung"<<endl;
49 }
50 }
51 else if (choose == 2){
52     ofstream outfile(filename);
53     if (a.size() == 2){
54         outfile<<"Phuong trinh bac 1 co nghiem duy nhat " <<fixed<<setprecision(
            decimal)<<-a[1]/a[0]<<endl;
55         return;
56     }
57     if (a.size() == 1){
58         cout<<"Ham hang"<<endl;
59         return;
60     }
61     if (check_condition(a,itv)){
62         cout<<"Thuc hien phuong phap day cung..."<<endl;
63         double x,d,tmp;
64         vector<double> m1M1;
65         m1_M1(a, itv, m1M1);
66         if(derivative2(a, (itv[0]+itv[1])/2)*fx(a,x)>0){
67             int cnt=1;
68             d=itv[0];
69             x=itv[1];
70             while(cnt <= iter){
71                 tmp=x;
72                 x=x+dx(a,x,d);
73                 export_solution(decimal, x, cnt,outfile);
74                 cnt+=1;
75             }
76         }
77     }
78     else{
79         int cnt=1;
80         d=itv[1];
81         x=itv[0];
82         while(cnt <= iter){
83             tmp=x;
84             x=x+dx(a,x,d);

```

```

85         export_solution(decimal, x, cnt,outfile);
86         cnt+=1;
87     }
88 }
89 outfile<<"Gia tri m1 = min(|f(x)| tren doan ("<<fixed<<setprecision(decimal)
90     <<itv[0]<<" , "<<fixed<<setprecision(decimal)<<itv[1]<<") la : "
91     <<fixed<<setprecision(decimal)<<m1M1[0]<<endl;
92 outfile<<"Gia tri M1 = max(|f(x)| tren doan ("<<fixed<<setprecision(decimal)
93     <<itv[0]<<" , "<<fixed<<setprecision(decimal)<<itv[1]<<") la : "
94     <<fixed<<setprecision(decimal)<<m1M1[1]<<endl;
95 outfile<<"Tinh theo cong thuc 1, sai so |xn-x*| <= " <<fixed<<setprecision(
96     decimal)<<abs(fx(a, x))/m1M1[0]<<endl;
97 outfile<<"Tinh theo cong thuc 2, sai so |xn-x*| <= " <<fixed<<setprecision(
98     decimal)<<(m1M1[1]-m1M1[0])/m1M1[0]*abs(x-tmp)<<endl;
99 }
100 else{
101     outfile<<"Khong thoa man dieu kien phuong phap day cung"<<endl;
102 }
103 outfile.close();
104 }
105 else if (choose == 3){
106     read_file(filename);
107 }
108 else{
109     delete_file(filename);
110 }
111 }

```

## Bước 8 : Chức năng giải gần đúng(Theo sai số yêu cầu)

-Input : Các hệ số của đa thức, khoảng cách li

-Output : Kết quả  $x$  sau mỗi lần lặp

-Ý tưởng : Dùng phương pháp dây cung để tìm nghiệm đến khi sai số thỏa mãn điều kiện cho trước

-Mã giả :

```
1 Function solution2(a, itv, ep, decimal, choose, filename):
2     if size of a == 2:
3         print "Phương trình bậc 1 có nghiệm duy nhất", -a[1] / a[0] with decimal places
4         return
5     if size of a == 1:
6         print "Ham hang"
7         return
8
9     if choose == 1:
10        if check_condition(a, itv):
11            print "Thực hiện phương pháp dây cung..."
12            x, d, tmp = 0, 0, 0
13            m1M1 = vector of size 2
14            call m1_M1(a, itv, m1M1)
15
16            if derivative2(a, (itv[0] + itv[1]) / 2) * fx(a, x) > 0:
17                cnt = 1
18                d = itv[0]
19                x = itv[1]
20                while fx(a, x) >= ep * m1M1[0]:
21                    tmp = x
22                    x = x + dx(a, x, d)
23                    call output_solution(decimal, x, cnt)
24                    cnt += 1
25            else:
26                cnt = 1
27                d = itv[1]
28                x = itv[0]
29                while fx(a, x) >= ep * m1M1[0]:
30                    tmp = x
31                    x = x + dx(a, x, d)
32                    call output_solution(decimal, x, cnt)
33                    cnt += 1
34            else:
35                print "Không thỏa mãn điều kiện phương pháp dây cung"
36
37        else if choose == 2:
38            open file with name filename for writing as outfile
39            if check_condition(a, itv):
40                print "Thực hiện phương pháp dây cung..."
41                x, d, tmp = 0, 0, 0
42                m1M1 = vector of size 2
43                call m1_M1(a, itv, m1M1)
44
45            if derivative2(a, (itv[0] + itv[1]) / 2) * fx(a, x) > 0:
46                cnt = 1
47                d = itv[0]
48                x = itv[1]
49                while abs(fx(a, x)) >= ep * m1M1[0]:
50                    tmp = x
51                    x = x + dx(a, x, d)
```

```

52         call_export_solution(decimal, x, cnt, outfile)
53         cnt += 1
54     else:
55         cnt = 1
56         d = itv[1]
57         x = itv[0]
58         while abs(fx(a, x)) >= ep * m1M1[0]:
59             tmp = x
60             x = x + dx(a, x, d)
61             call_export_solution(decimal, x, cnt, outfile)
62             cnt += 1
63     else:
64         write to outfile "Khong thoa man dieu kien phuong phap day cung"
65     close outfile
66
67     else if choose == 3:
68         call_read_file(filename)
69
70     else:
71         call_delete_file(filename)

```

## -Code C++ :

```

1  // Solution with epsilon
2  void solution2(vector<double>& a, vector<double>& itv, double ep, int decimal, int choose
, string filename){
3      if (a.size() == 2){
4          cout<<"Phuong trinh bac 1 co nghiem duy nhat "<<fixed<<setprecision(decimal)<<-a
[1]/a[0]<<endl;
5          return;
6      }
7      if (a.size () == 1){
8          cout<<"Ham hang"<<endl;
9          return;
10     }
11     if (choose == 1){
12         if (check_condition(a,itv)){
13             cout<<"Thuc hien phuong phap day cung..."<<endl;
14             double x,d,tmp;
15             vector<double> m1M1;
16             m1_M1(a, itv, m1M1);
17             if(derivative2(a, (itv[0]+itv[1])/2)*fx(a,x)>0){
18                 int cnt=1;
19                 d=itv[0];
20                 x=itv[1];
21                 while(fx(a,x) >= ep*m1M1[0]){
22                     tmp=x;
23                     x=x+dx(a,x,d);
24                     output_solution(decimal, x, cnt);
25                     cnt+=1;
26                 }
27             }
28         }
29         else{
30             int cnt=1;
31             d=itv[1];
32             x=itv[0];
33             while(fx(a,x) >= ep*m1M1[0]){
34                 tmp=x;
35                 x=x+dx(a,x,d);
36                 output_solution(decimal, x, cnt);
37                 cnt+=1;
38             }
39         }
40     }
41 }

```

```

42         else{
43             cout<<"Khong thoa man dieu kien phuong phap day cung"<<endl;
44         }
45     }
46     else if (choose == 2){
47         ofstream outfile(filename);
48         if (check_condition(a,itv)){
49             cout<<"Thuc hien phuong phap day cung..."<<endl;
50             double x,d,tmp;
51             vector<double> m1M1;
52             m1_M1(a, itv, m1M1);
53             if (derivative2(a, (itv[0]+itv[1])/2)*fx(a,x)>0){
54                 int cnt=1;
55                 d=itv[0];
56                 x=itv[1];
57                 while (abs (fx(a,x)) >= ep*m1M1[0]){
58                     tmp=x;
59                     x=x+dx(a,x,d);
60                     export_solution(decimal, x, cnt,outfile);
61                     cnt+=1;
62                 }
63             }
64             else{
65                 int cnt=1;
66                 d=itv[1];
67                 x=itv[0];
68                 while (abs (fx(a,x)) >= ep*m1M1[0]){
69                     tmp=x;
70                     x=x+dx(a,x,d);
71                     export_solution(decimal, x, cnt,outfile);
72                     cnt+=1;
73                 }
74             }
75         }
76     }
77     else{
78         outfile<<"Khong thoa man dieu kien phuong phap day cung"<<endl;
79     }
80     outfile.close();
81 }
82 else if (choose == 3){
83     read_file(filename);
84 }
85 else{
86     delete_file(filename);
87 }
88
89 }

```



## Bước 9 : Chức năng giải gần đúng nghiệm thỏa mãn $|x_n - x_{n-1}| \leq \epsilon$

-Input : Các hệ số của đa thức, khoảng cách li

-Output : Kết quả  $x$  sau mỗi lần lặp

-Ý tưởng : Dùng phương pháp dây cung để tìm nghiệm đến khi thỏa mãn  $|x_n - x_{n-1}| \leq \epsilon$

-Mã giả : Tương tự bước 7,8(thay điều kiện ở vòng lặp while)

-Code C++

```
1 void solution3(vector<double>& a, vector<double>& itv, double ep, int decimal, int choose
  , string filename){
2   if (a.size() == 2){
3     cout<<"Phương trình bậc 1 có nghiệm duy nhất "<<fixed<<setprecision(decimal)<<-a
      [1]/a[0]<<endl;
4     return;
5   }
6   if (a.size () == 1){
7     cout<<"Ham hang"<<endl;
8     return;
9   }
10  if (choose == 1){
11    if (check_condition(a,itv)){
12      cout<<"Thực hiện phương pháp dây cung..."<<endl;
13      double x,d,tmp;
14      if(derivative2(a, (itv[0]+itv[1])/2)*fx(a,x)>0){
15        int cnt=1;
16        d=itv[0];
17        x=itv[1];
18        double tmp=-inf;
19        while (abs(x-tmp)>ep){
20          tmp=x;
21          x=x+dx(a,x,d);
22          output_solution(decimal, x, cnt);
23          cnt+=1;
24        }
25      }
26    }
27    else{
28      int cnt=1;
29      d=itv[1];
30      x=itv[0];
31      double tmp=-inf;
32      while (abs(x-tmp) > ep){
33        tmp=x;
34        x=x+dx(a,x,d);
35        output_solution(decimal, x, cnt);
36        cnt+=1;
37      }
38    }
39  }
40  }
41  }
42  else{
43    cout<<"Không thỏa mãn điều kiện phương pháp dây cung"<<endl;
44  }
45 }
```

```

46     else if (choose == 2){
47         ofstream outfile(filename);
48         if (check_condition(a,itv)){
49             cout<<"Thuc hien phuong phap day cung..."<<endl;
50             double x,d,tmp;
51             if(derivative2(a,(itv[0]+itv[1])/2)*fx(a,x)>0){
52                 int cnt=1;
53                 d=itv[0];
54                 x=itv[1];
55                 double tmp=-inf;
56                 while(abs(x-tmp)){
57                     tmp=x;
58                     x=x+dx(a,x,d);
59                     export_solution(decimal, x, cnt,outfile);
60                     cnt+=1;
61                 }
62             }
63             else{
64                 int cnt=1;
65                 d=itv[1];
66                 x=itv[0];
67                 tmp=-inf;
68                 while(abs(x-tmp) > ep){
69                     tmp=x;
70                     x=x+dx(a,x,d);
71                     export_solution(decimal, x, cnt,outfile);
72                     cnt+=1;
73                 }
74             }
75         }
76     }
77     else{
78         outfile<<"Khong thoa man dieu kien phuong phap day cung"<<endl;
79     }
80     outfile.close();
81 }
82 else if (choose == 3){
83     read_file(filename);
84 }
85 else{
86     delete_file(filename);
87 }
88 }

```

## Bước 10 : Chức năng vẽ đồ thị hàm số trên khoảng cách li

-Input : Các hệ số của đa thức, khoảng cách li

-Output : Đồ thị hàm số trên khoảng cách li

-Ý tưởng : Vẽ từng điểm của đồ thị hàm số với bước nhảy nhỏ

-Mã giả :

```
1 Function drawPolynomialGraph(a, x_center, y_center, scale, x_start, x_end):
2     Set drawing color to YELLOW
3     Define step size as 0.00001
4
5     For each x from x_start to x_end, incrementing by step:
6         Compute the y value using fx(a, x)
7         Convert x to pixel coordinate: x_pixel = x_center + (x * scale)
8         Convert y to pixel coordinate: y_pixel = y_center - (y * scale)
9         Draw a pixel at (x_pixel, y_pixel) in YELLOW
10
11 Function drawAxes(x_center, y_center):
12     Set drawing color to WHITE
13
14     // Draw horizontal x-axis:
15     Draw a line from (0, y_center) to the right edge of the screen
16
17     // Draw vertical y-axis:
18     Draw a line from (x_center, 0) to the bottom edge of the screen
```

## -Code C++ :

```
1 // Draw graph
2 void drawPolynomialGraph(vector<double>& a, int x_center, int y_center, int scale, double
   x_start, double x_end) {
3     setcolor(YELLOW);
4     double step = 0.00001;
5     for (double x = x_start; x < x_end ; x+=step) {
6         double y=fx(a,x);
7         int x_pixel = x_center + (x * scale);
8         int y_pixel = y_center - (y * scale);
9         putpixel(x_pixel, y_pixel, YELLOW);
10    }
11 }
12 void drawAxes(int x_center, int y_center) {
13     setcolor(WHITE);
14     // Draw x axe
15     line(0, y_center, getmaxx(), y_center);
16     // Draw y axe
17     line(x_center, 0, x_center, getmaxy());
18 }
```

## Bước 11 : Chức năng lưu kết quả vào file .txt

-Input : Kết quả khoảng cách li/nghiệm sau từng lần lặp của thuật toán

-Output : File .txt có kết quả của thuật toán (kết quả trung gian và kết quả cuối cùng)

-Ý tưởng : Đọc kết quả vào file sau mỗi lần lặp của thuật toán

-Mã giả :

```
1 Function export_range(decimal, radius, filename):
2     Open file with filename for writing
3
4     If file cannot be opened:
5         Print "Cannot open file"
6         Exit function
7
8     Create a string stream for formatting numbers
9     Set the precision of the string stream to the given decimal places
10    Format the range as a string: "(-radius, radius)"
11
12    Write a header line to the file
13    Write a formatted line for the range:
14        Include "Radius" and the formatted range string
15
16    Write a footer line to the file
17
18    Print "File export successful!"
19    Close the file
20
21 Function export_interval(decimal, start, end, file, count):
22     Create a string stream for formatting numbers
23     Set the precision of the string stream to the given decimal places
24     Format the interval as a string: "(start, end)"
25
26     If count is 1:
27         Write a header line to the file
28         Write a formatted line for the first iteration:
29             Include "Iteration" and the formatted interval string
30         Write a footer line to the file
31
32     Else:
33         Write a formatted line for subsequent iterations:
34             Include "Iteration" and the formatted interval string
35         Write a footer line to the file
36
37
38 Function export_solution(decimal, solution, count, file):
39     If count is 1:
40         Write a header line to the file
41         Write a formatted line for the first iteration:
42             Include "Iteration" and the formatted solution
43         Write a footer line to the file
44
45     Else:
46         Write a formatted line for subsequent iterations:
47             Include "Iteration" and the formatted solution
48         Write a footer line to the file
```

## -Code C++ :

```
1 void export_range(int decimal, double r, string filename){
2     ofstream outfile(filename);
3     if (!outfile){
4         cout<<"Khong the mo file"<<endl;
5     }
6     ostringstream ss;
7     ss << fixed << setprecision(decimal);
8     ss << "(" << -r << ", " << r << ")";
9     outfile << "
10         +-----+
11         << endl;
12     outfile << "| " << setw(colWidth1) << left << "Ban kinh nghiem"
13         << " | " << setw(colWidth2) << left << "Khoang nghiem" << " |" <<
14         endl;
15     outfile << "
16         +-----+
17         << endl;
18     outfile << "| " << setw(colWidth1) << left << fixed << setprecision(decimal) << r
19         << " | " << setw(colWidth2) << left << ss.str() << " |" << endl;
20     outfile << "
21         +-----+
22         << endl;
23     cout <<"Xuat file thanh cong !"<<endl;
24     outfile.close();
25 }
26 void export_interval(int decimal, double a, double b, ofstream& outfile, int count){
27     ostringstream ss;
28     ss << fixed << setprecision(decimal);
29     ss << "(" << a << ", " << b << ")";
30     if (count == 1){
31         outfile << "
32             +-----+
33             << endl;
34         outfile << "| " << setw(colWidth1) << left << "Lan lap"
35             << " | " << setw(colWidth2) << left << "Khoang nghiem" << " |"
36             << endl;
37         outfile << "
38             +-----+
39             << endl;
40
41         outfile << "| " << setw(colWidth1) << left << count
42             << " | " << setw(colWidth2) << left << ss.str() << " |" << endl
43             ;
44         outfile << "
45             +-----+
46             << endl;
47     }
48     else{
49         outfile << "| " << setw(colWidth1) << left << count
50             << " | " << setw(colWidth2) << left << ss.str() << " |" << endl
51             ;
52         outfile << "
53             +-----+
54             << endl;
55     }
56 }
57 void export_solution(int decimal, double x, int count, ofstream& outfile ){
58     if (count == 1){
59         outfile << "
60             +-----+
61             << endl;
62         outfile << "| " << setw(colWidth1) << left << "Lan lap"
63             << " | " << setw(colWidth2) << left << "Nghiem" << " |" << endl
64             ;
65     }
```

```

45     outfile << "
        +-----+
        << endl;
46
47     outfile << "| " << setw(colWidth1) << left << count
48         << " | " << setw(colWidth2) << left << fixed << setprecision(decimal) <<
        x << " | " << endl;
49     outfile << "
        +-----+
        << endl;
50 }
51 else{
52     outfile << "| " << setw(colWidth1) << left << count
53         << " | " << setw(colWidth2) << left << fixed << setprecision(decimal) << x
        << " | " << endl;
54     outfile << "
        +-----+
        << endl;
55 }
56 }

```

## Bước 12 : Chức năng đọc file .txt

-Input : Tên file

-Output : Nội dung file

-Mã giả :

```

1  Function read_file(filename):
2      # Check if the file is empty
3      If the file is empty (isEmpty(filename)):
4          Print "The file is empty, no content to read"
5      Else:
6          # Open the file for reading
7          Open the file with filename for reading (inFile)
8
9          # Read each line from the file
10         For each line in the file (line):
11             Print the line
12
13         # Close the file after reading
14         Close the file (inFile)

```

## -Code C++

```

1  void read_file(string filename){
2      if (isEmpty(filename)){
3          cout << "File rỗng, không có nội dung để đọc"<<endl;
4      }
5      else{
6          ifstream inFile(filename);
7          string line;
8          while (getline(inFile, line)) {
9              cout << line <<endl;
10             }
11         inFile.close();
12     }
13 }

```

## **4 Kết quả thực hiện chương trình**

```
PS D:\NGUYEN_TAT_CUONG_20227090> .\main.exe

Chon chuc nang

1. Nhap ham so
2. Tim mien chua nghiem
3. Tim khoang cach li nghiem (a, b) bang phuong phap chia doi thoa man  $|a-b| \leq 0.5$ 
4. Ve do thi ham so tren khoang cach li nghiem (a, b)
5. Tim nghiem gan dung tren khoang cach li nghiem theo n va danh gia sai so theo 2 cach
6. Tim nghiem gan dung tren khoang cach li nghiem theo ep va danh gia sai so theo 2 cach
7. Tim nghiem gan dung tren khoang cach li nghiem thoa man  $|x(n)-x(n-1)| \leq e$ 
8. Thoat chuong trinh

Cach dung menu: Nhap lua chon bang ban phim

```

Hình 3: Menu của chương trình

```
PS D:\NGUYEN_TAT_CUONG_20227090> .\main.exe

Chon chuc nang

1. Nhap ham so
2. Tim mien chua nghiem
3. Tim khoang cach li nghiem (a, b) bang phuong phap chia doi thoa man  $|a-b| \leq 0.5$ 
4. Ve do thi ham so tren khoang cach li nghiem (a, b)
5. Tim nghiem gan dung tren khoang cach li nghiem theo n va danh gia sai so theo 2 cach
6. Tim nghiem gan dung tren khoang cach li nghiem theo ep va danh gia sai so theo 2 cach
7. Tim nghiem gan dung tren khoang cach li nghiem thoa man  $|x(n)-x(n-1)| \leq e$ 
8. Thoat chuong trinh

Cach dung menu: Nhap lua chon bang ban phim
1
Nhap vao bac cua da thuc : 2

Nhap vao he so cua  $x^2$  : -1

Nhap vao he so cua  $x^1$  : 4

Nhap vao he so cua  $x^0$  : 5

```

Hình 4: Nhập hàm số



```

                                Nhập lựa chọn
1. In ra kết quả trên màn hình
2. Xuất kết quả ra file văn bản
3. Hiện thị kết quả từ file văn bản
4. Menu

Nhập lựa chọn từ bàn phím
1
+-----+
| Ban kinh nghiệm          | Khoảng nghiệm          |
+-----+
| 6                        | (6, -6)                |
+-----+

```

Hình 5: Tìm khoảng nghiệm và in ra màn hình

```

≡ output.txt
1 +-----+
2 | Ban kinh nghiệm          | Khoảng nghiệm          |
3 +-----+
4 | 6                        | (-6, 6)                |
5 +-----+
6

OUTPUT  TERMINAL  PROBLEMS  PORTS

+-----+
| 6                        | (6, -6)                |
+-----+

                                Nhập lựa chọn
1. In ra kết quả trên màn hình
2. Xuất kết quả ra file văn bản
3. Hiện thị kết quả từ file văn bản
4. Menu

Nhập lựa chọn từ bàn phím
2
Xuất file thành công !

```

Hình 6: Tìm khoảng nghiệm và lưu vào file

```

                                Nhap lua chon
1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu

Nhap lua chon tu ban phim
1
+-----+
| Ban kinh nghiem          | Khoang nghiem          |
+-----+
| 6                        | (6, -6)                |
+-----+

```

Hình 7: Hiển thị khoảng nghiệm từ file văn bản

```

                                Nhap lua chon
1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu

Nhap lua chon tu ban phim
4
                                Chon chuc nang
1. Nhap ham so
2. Tim mien chua nghiem
3. Tim khoang cach li nghiem (a, b) bang phuong phap chia doi thoa man  $|a-b| \leq 0.5$ 
4. Ve do thi ham so tren khoang cach li nghiem (a, b)
5. Tim nghiem gan dung tren khoang cach li nghiem theo n va danh gia sai so theo 2 cach
6. Tim nghiem gan dung tren khoang cach li nghiem theo ep va danh gia sai so theo 2 cach
7. Tim nghiem gan dung tren khoang cach li nghiem thoa man  $|x(n)-x(n-1)| \leq e$ 
8. Thoat chuong trinh

Cach dung menu: Nhap lua chon bang ban phim

```

Hình 8: Quay trở lại menu

```

3
Nhap vao so chu so thap phan hien thi sau dau phay :
7

      Nhap lua chon
-----
1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu
-----

Nhap lua chon tu ban phim
1
+-----+
| Lan lap          | Khoang nghiem          |
+-----+
| 1                 | (-6.000000, 6.000000) |
+-----+
| 2                 | (0.000000, 6.000000)  |
+-----+
| 3                 | (3.000000, 6.000000)  |
+-----+
| 4                 | (4.500000, 6.000000)  |
+-----+
| 5                 | (4.500000, 5.250000)  |
+-----+
Khoang cach li nghiem la (4.875000 ,5.250000)

```

Hình 9: Tìm khoảng cách theo phương pháp chia đôi và in ra màn hình

The screenshot shows a code editor with four tabs: `function.cpp`, `main.cpp`, `interval.txt` (active), and `function.h`. The `interval.txt` file contains the same output as Figure 9, but with line numbers 1 through 15. Below the editor, the `TERMINAL` tab is active, showing the program's execution. It displays the menu option '3. Hien thi ket qua tu file van ban' and '4. Menu', followed by the prompt 'Nhap lua chon tu ban phim' with a cursor.

```

Go Run ... < -> NGUYEN_TAT_CUONG_20227090
function.cpp M main.cpp M interval.txt U x function.h M
interval.txt
1 +-----+
2 | Lan lap          | Khoang nghiem          |
3 +-----+
4 | 1                 | (-6.000000, 6.000000) |
5 +-----+
6 | 2                 | (0.000000, 6.000000)  |
7 +-----+
8 | 3                 | (3.000000, 6.000000)  |
9 +-----+
10 | 4                 | (4.500000, 6.000000)  |
11 +-----+
12 | 5                 | (4.500000, 5.250000)  |
13 +-----+
14 Khoang cach li nghiem la (4.875000 ,5.250000)
15
OUTPUT TERMINAL PROBLEMS PORTS
3. Hien thi ket qua tu file van ban
4. Menu
Nhap lua chon tu ban phim

```

Hình 10: Tìm khoảng cách li theo phương pháp chia đôi và lưu vào file

```

Nhap lua chon tu ban phim
3
+-----+
| Lan lap          | Khoang nghiem      |
+-----+
| 1                | (-6.000000, 6.000000) |
+-----+
| 2                | (0.000000, 6.000000)  |
+-----+
| 3                | (3.000000, 6.000000)  |
+-----+
| 4                | (4.500000, 6.000000)  |
+-----+
| 5                | (4.500000, 5.250000)  |
+-----+
Khoang cach li nghiem la (4.875000 ,5.250000)

      Nhap lua chon
1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu

Nhap lua chon tu ban phim

```

Hình 11: Hiển thị khoảng cách li từ file văn bản

```

2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu
Nhap lua chon tu
4
      Nhap lua chon
1. Nhap ham s
2. Tim mien c
3. Tim khoang
4. Ve do thi
5. Tim nghiem
6. Tim nghiem
7. Tim nghiem
8. Thoat chuo

      Windows BGI
      [A window titled 'Windows BGI' showing a coordinate grid with a yellow line segment drawn on it.]

Cach dung menu: Nhap lua chon bang ban phim
4
2
Khoang cach li (4.87500 ,5.25000)
Nhan phim T de quay lai MENU

```

Hình 12: Vẽ đồ thị hàm số trên khoảng cách li

```

Nhap vao so chu so thap phan hien thi sau dau phay :
19
Nhap vao so lan lap
2

      Nhap lua chon
      1. In ra ket qua tren man hinh
      2. Xuat ket qua ra file van ban
      3. Hien thi ket qua tu file van ban
      4. Menu

Nhap vao lua chon
1
Thuc hien phuong phap day cung...
+-----+
| Lan lap          | Nghiem          |
+-----+
| 1                | 4.9948979591836737413 |
+-----+
| 2                | 5.0001086484137333699 |
+-----+
Gia tri m1 = min(|f(x)| tren doan (4.87500000000000000000 ,5.25000000000000000000) la : 5.75000000000000000000
Gia tri M1 = max(|f(x)| tren doan (4.87500000000000000000 ,5.25000000000000000000) la : 6.4999999954418967718
Tinh theo cong thuc 1, sai so |xn-x*| <= 0.0001133743107616332
Tinh theo cong thuc 2, sai so |xn-x*| <= 0.0006796551128337152

```

Hình 13: In kết quả giải gần đúng bằng phương pháp dây cung theo số lần lặp và đánh giá sai số

```

function.cpp M ×  main.cpp M  solution1.txt U ×  function.h M
D:\NGUYEN_TAI_CUONG_20227090\function.cpp • Modified
1  +-----+
2  | Lan lap          | Nghiem          |
3  +-----+
4  | 1                | 4.9948979591836737413 |
5  +-----+
6  | 2                | 4.9997957516339868178 |
7  +-----+
8  Gia tri m1 = min(|f(x)| tren doan (4.87500000000000000000 ,5.25000000000000000000) la : 5.75000000000000000000
9  Gia tri M1 = max(|f(x)| tren doan (4.87500000000000000000 ,5.25000000000000000000) la : 6.4999999954418967718
10 Tinh theo cong thuc 1, sai so |xn-x*| <= 0.0002131214745539321
11 Tinh theo cong thuc 2, sai so |xn-x*| <= 0.0006388424896365503
12

OUTPUT  TERMINAL  PROBLEMS  PORTS
main +

      2. Xuat ket qua ra file van ban
      3. Hien thi ket qua tu file van ban
      4. Menu

Nhap vao lua chon

2
Thuc hien phuong phap day cung...

      Nhap lua chon
      1. In ra ket qua tren man hinh
      2. Xuat ket qua ra file van ban

```

Hình 14: Lưu kết quả giải nghiệm ra theo số lần lặp ra file văn bản

```

      Nhap lua chon
    1. In ra ket qua tren man hinh
    2. Xuat ket qua ra file van ban
    3. Hien thi ket qua tu file van ban
    4. Menu
Nhap vao lua chon
3
+-----+
| Lan lap          | Nghiem          |
+-----+
| 1                | 4.9948979591836737413 |
+-----+
| 2                | 4.9997957516339868178 |
+-----+
Gia tri m1 = min(|f(x)| tren doan (4.875000000000000000 ,5.250000000000000000) la : 5.750000000000000000
Gia tri M1 = max(|f(x)| tren doan (4.875000000000000000 ,5.250000000000000000) la : 6.4999999954418967718
Tinh theo cong thuc 1, sai so |xn-x*| <= 0.0002131214745539321
Tinh theo cong thuc 2, sai so |xn-x*| <= 0.0006388424896365503

```

Hình 15: Hiển kết quả giải gần đúng nghiệm theo số lần lặp từ file

```

Nhap vao lua chon
1
Thuc hien phuong phap day cung...
+-----+
| Lan lap          | Nghiem          |
+-----+
| 1                | 4.995           |
+-----+
| 2                | 5.000           |
+-----+
| 3                | 5.000           |
+-----+
Gia tri m1 = min(|f(x)| tren doan (4.875 ,5.250) la : 5.750
Gia tri M1 = max(|f(x)| tren doan (4.875 ,5.250) la : 6.500
Tinh theo cong thuc 1, sai so |xn-x*| <= 0.000
Tinh theo cong thuc 2, sai so |xn-x*| <= 0.000

```

Hình 16: In kết quả giải gần đúng nghiệm theo sai số và đánh giá sai số

```

1  +-----+
2  | Lan lap                                | Nghiem |
3  +-----+
4  | 1                                    | 4.995  |
5  +-----+
6  | 2                                    | 5.000  |
7  +-----+
8  | 3                                    | 5.000  |
9  +-----+
10 Gia tri m1 = min(|f(x)| tren doan (4.875 ,5.250) la : 5.750
11 Gia tri M1 = max(|f(x)| tren doan (4.875 ,5.250) la : 6.500
12 Tinh theo cong thuc 1, sai so |xn-x*| <= 0.000
13 Tinh theo cong thuc 2, sai so |xn-x*| <= 0.000
14

OUTPUT  TERMINAL  PROBLEMS  PORTS

Tinh theo cong thuc 2, sai so |xn-x*| <= 0.000

      Nhap lua chon
-----
1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu
-----
Nhap vao lua chon
2
Thuc hien phuong phap day cung...

```

Hình 17: Lưu kết quả giải gần đúng nghiệm theo sai số ra file văn bản

```

solution2.txt
1  +-----+
2  | Lan lap                      | Nghiem                      |
3  +-----+
4  | 1                          | 4.995                      |
5  +-----+
6  | 2                          | 5.000                      |
7  +-----+
8  | 3                          | 5.000                      |
9  +-----+

OUTPUT  TERMINAL  PROBLEMS  PORTS

+-----+
| 2. Xuat ket qua ra file van ban |
+-----+
| 3. Hien thi ket qua tu file van ban |
+-----+
| 4. Menu |
+-----+

Nhap vao lua chon
3
+-----+
| Lan lap                      | Nghiem                      |
+-----+
| 1                          | 4.995                      |
+-----+
| 2                          | 5.000                      |
+-----+
| 3                          | 5.000                      |
+-----+

Gia tri m1 = min(|f(x)| tren doan (4.875 ,5.250) la : 5.750
Gia tri M1 = max(|f(x)| tren doan (4.875 ,5.250) la : 6.500
Tinh theo cong thuc 1, sai so |xn-x*| <= 0.000

```

Hình 18: Hiển thị kết quả giải gần đúng nghiệm theo sai số ra màn hình



```

Cach dung menu: Nhap lua chon bang ban phim
7
Nhap vao so chu so thap phan hien thi sau dau phay :
5
Nhap vao sai so
0.001

      Nhap lua chon
1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu

Nhap vao lua chon
1
Thuc hien phuong phap day cung...
+-----+
| Lan lap          | Nghiem          |
+-----+
| 1                | 4.99490        |
+-----+
| 2                | 4.99980        |
+-----+
| 3                | 4.99999        |
+-----+

```

Hình 19: In kết quả giải gần đúng nghiệm thỏa mãn  $|x_n - x_{n-1}| \leq \epsilon$

```

≡ solution3.txt
1  +-----+
2  | Lan lap          | Nghiem          |
3  +-----+
4  | 1                | 4.99490        |
5  +-----+
6  | 2                | 4.99980        |
7  +-----+
8  | 3                | 4.99999        |
9  +-----+
10

```

---

```

OUTPUT  TERMINAL  PROBLEMS  PORTS

      2. Xuat ket qua ra file van ban
      3. Hien thi ket qua tu file van ban
      4. Menu

Nhap vao lua chon
2
Thuc hien phuong phap day cung...

```

Hình 20: Lưu ra file văn bản kết quả giải gần đúng nghiệm thỏa mãn  $|x_n - x_{n-1}| \leq \epsilon$

```
Cach dung menu: Nhap lua chon bang ban phim
7
Nhap vao so chu so thap phan hien thi sau dau phay :
5
Nhap vao sai so
0.001

      Nhap lua chon

1. In ra ket qua tren man hinh
2. Xuat ket qua ra file van ban
3. Hien thi ket qua tu file van ban
4. Menu

Nhap vao lua chon
1
Thuc hien phuong phap day cung...
+-----+
| Lan lap          | Nghiem          |
+-----+
| 1                 | 4.99490         |
+-----+
| 2                 | 4.99980         |
+-----+
| 3                 | 4.99999         |
+-----+
```

Hình 21: Hiển thị kết quả giải gần đúng nghiệm thỏa mãn  $|x_n - x_{n-1}| \leq \epsilon$

## **5 Kết luận, đánh giá kết quả đạt được**

### **5.1 Đánh giá chung**

Qua bài tập lớn lần này, em đã học được một số điều như sau:

- Cách xây dựng chương trình theo phương pháp tính chỉnh dần.
- Cách tổ chức file trong 1 dự án (các hàm viết vào 1 file function.cpp, gọi tên hàm ở 1 file function.h, include file function.h vào file main.cpp).
- Được tìm hiểu về 1 thư viện đồ họa và áp dụng vào bài tập của bản thân.
- Ngoài ra, còn được học, ôn tập lại kiến thức trong môn Giải tích số và rèn luyện tính kiên trì, cẩn thận trong quá trình viết code .

### **5.2 Đánh giá chi tiết**

TT	Nội dung yêu cầu(theo đề)	Đã viết code?(Y/N)	Đã thực hiện đúng?(Y/N)	Tự đánh giá ưu/nhược điểm, sáng tạo
1	Tìm miền chứa nghiệm của phương trình	Y	Y	<b>Ưu điểm:</b> Tốc độ tính toán nhanh với độ phức tạp $O(n)$ . <b>Nhược điểm:</b> Không áp dụng cho hàm hằng
2	Tìm khoảng cách li bằng phương pháp chia đôi để thu hẹp khoảng nghiệm sao cho độ dài khoảng cách li nhỏ hơn hoặc bằng 0.5	Y	Y	<b>Ưu điểm:</b> Hội tụ nhanh về khoảng cách li <b>Nhược điểm:</b> Xử lý các trường hợp ngoại lệ như: không có khoảng cách li/khi chia đôi trùng vào nghiệm của phương trình
3	Vẽ đồ thị trên khoảng cách li	Y	Y	<b>Ưu điểm:</b> Chương trình đơn giản, dễ thực hiện <b>Nhược điểm:</b> Vẽ các chấm nhỏ để tạo thành đồ thị nên nhìn chưa thẩm mỹ
4	Tìm nghiệm gần đúng với số lần lặp cho trước trong khoảng phân li nghiệm đã tìm trước đó, sau đó đánh giá sai số theo 2 công thức	Y	Y	<b>Ưu điểm:</b> Phương pháp hội tụ nhanh về nghiệm nếu số lần lặp đủ lớn <b>Nhược điểm:</b> Nếu số lần lặp ít, phương pháp khó hội tụ về nghiệm
5	Tìm nghiệm gần đúng trên khoảng cách li với sai số nhập vào trong khoảng phân li đã tìm trước đó, sau đó đánh giá sai số theo 2 công thức	Y	Y	<b>Ưu điểm:</b> Sai số đủ nhỏ thì phương pháp sẽ hội tụ về nghiệm gần đúng <b>Nhược điểm:</b> Sai số lớn thì phương pháp có thể giải ra kết quả sai khác lớn với kết quả thực tế
6	Tìm nghiệm gần đúng trên khoảng cách li thỏa mãn $ x_n - x_{n-1}  \leq \epsilon$	Y	Y	<b>Ưu điểm:</b> Nếu $ x_n - x_{n-1} $ đủ nhỏ, phương pháp sẽ hội tụ về nghiệm gần đúng <b>Nhược điểm:</b> Nếu $ x_n - x_{n-1} $ lớn, phương pháp có thể giải ra kết quả sai khác lớn với kết quả thực tế

Hình 22: Bảng đánh giá chi tiết