

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



BÁO CÁO CUỐI KỲ KỸ THUẬT LẬP TRÌNH

Thiết kế, xây dựng chương trình giải gần đúng hệ phương trình đại số tuyến tính $Ax = b$ bằng phương pháp lặp đơn và lặp Seidel

GVHD : TS. NGUYỄN THỊ THANH HUYỀN
NHÓM SINH VIÊN THỰC HIỆN:

1. VŨ THỊ TÂM
2. ĐẶNG THỊ HỒNG NHUNG

MSSV: 20185403
MSSV: 20185391

Hà Nội, 07/2020

MỤC LỤC

I. Ý TƯỞNG CHUNG GIẢI QUYẾT BÀI TOÁN.....	2
II. QUÁ TRÌNH THIẾT KẾ CHƯƠNG TRÌNH	4
III. QUÁ TRÌNH XÂY DỰNG CHƯƠNG TRÌNH:	5
1. Input	5
2. Output.....	5
3. Ý tưởng.....	5
4. Function NhapMaTran.....	5
5. Function NhapVecto.....	5
6. Function XuatVecto.....	6
7. Function XuatMaTran	6
8. Function Cheotroi	6
9. Function norm_row	7
10. Function kt_hoi_tuDON	7
11. Function kt_hoi_tuSEIDEL	8
12. Function loop_kDON.....	9
13. Function loop_kSEIDEL.....	10
14. Function loop_eps1	11
15. Function loop_eps2.....	13
IV. KẾT QUẢ THỰC HIỆN CHƯƠNG TRÌNH	17

I. Ý TƯỞNG CHUNG GIẢI QUYẾT BÀI TOÁN

- Sử dụng phương pháp lặp đơn và lặp Seidel để giải hệ phương trình $AX=B$ (1), trong đó A là ma trận hệ số cỡ $n \times n$, B là vector hệ số tự do, X là vector nghiệm của hệ.

a) Phương pháp lặp đơn:

- Bằng cách biến đổi hệ về dạng $X = \alpha X + \beta$ với α là ma trận cỡ $n \times n$. Nếu chuẩn của ma trận $\alpha < 1$ thì công thức lặp sau hội tụ về nghiệm:

$$X^{(n)} = \alpha X^{(n-1)} + \beta$$

- Ma trận $\alpha = (I - A)$, vector $\beta = B$ trong đó I : ma trận đơn vị,

A : ma trận hệ số ẩn,

B : vector hệ số tự do

- Sử dụng vòng lặp while...do kết hợp với 2 vòng for...to...do lồng nhau để thực hiện công thức lặp tìm nghiệm ứng với 2 trường hợp:

- Lặp với số lần biết trước k
- Lặp với sai số epsilon cho trước

b) Phương pháp lặp Seidel:

- Kiểm tra tính chéo trội của ma trận A : các phần tử trên đường chéo chính A_{ii} có trị tuyệt đối lớn hơn tổng trị tuyệt đối các phần tử còn lại trên cùng 1 hàng
- Biến đổi hệ về dạng $X = \alpha X + \beta = (\alpha_1 + \alpha_2) X + \beta = \alpha_1 X + \alpha_2 X + \beta$ trong đó α_1 : ma trận tam giác trên, α_2 : ma trận tam giác dưới, vector $\beta = -B$;

Ma trận α được tạo thành từ ma trận A bằng cách chia các phần tử trên cùng một hàng cho phần tử chéo trội A_{ii} , phương pháp lặp hội tụ khi và chỉ khi chuẩn ma trận $\alpha = \lambda < 1$

- Xây dựng vòng lặp xấp xỉ nghiệm sử dụng while...do với vòng for lồng nhau trong 2 trường hợp:
 - Lặp với số lần biết trước k
 - Lặp với sai số epsilon cho trước

- Sử dụng con trỏ để cấp phát động mảng 2 chiều A , α và mảng 1 chiều B , β . Vì dùng con trỏ thì có thể dễ dàng tạo các hàm con với biến trả về là một mảng ta chỉ cần trả về con trỏ

Thay vì $A[i][j]$ ta sẽ dùng con trỏ $a[i*n+j]$, vector B sẽ là con trỏ b, nghiệm của hệ là x[]

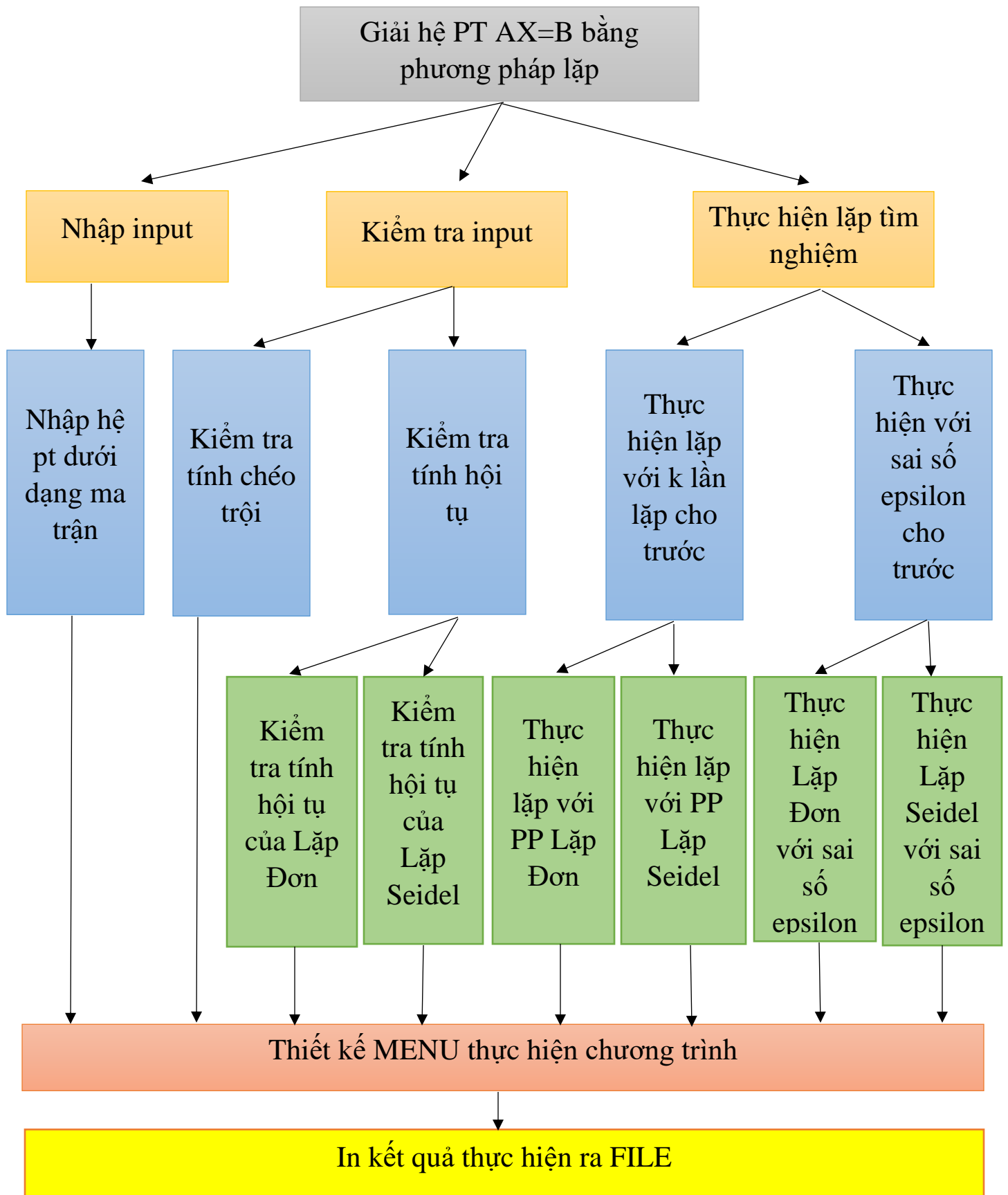
- Sử dụng con trỏ FILE để mở, đọc, viết vào file có địa chỉ là:

"D:\\C project\\final_exam.txt"

Bên trong mỗi hàm lồng ghép in kết quả ra màn hình đồng thời viết kết quả vào file đang mở.

- Các biến sẽ được dùng : x[max], x0[max], *a , *b, *f, *alpha1, *alpha2

II. QUÁ TRÌNH THIẾT KẾ CHƯƠNG TRÌNH



III. QUÁ TRÌNH XÂY DỰNG CHƯƠNG TRÌNH

1. Input

- Nhập các hệ số của phương trình vào mảng được cấp phát động bởi con trỏ cấp 1 gồm:
 - + ma trận A
 - + vector B

2. Output

- Toàn bộ kết quả thực hiện chương trình ra file “final_exam.txt”

3. Ý tưởng

- Dựa vào công thức lặp của từng phương pháp, ta xây dựng vòng lặp tìm nghiệm của hệ

4. Function NhapMaTran

- Vào: Con trỏ a; số dòng; số cột.
- Ra:
- Ý tưởng: Cho biến i chạy từ 0 -> dòng*cột, nhập vào từ bàn phím phần tử A[m][n] lưu vào con trỏ ở vị trí a+i

```
for( i = 0; i < dong * cot; i++)  
{  
    printf(" a[%d, %d] = ", i / cot, i % cot);  
    scanf("%f", a + i);  
}
```

5. Function NhapVecto

- Vào: Con trỏ b, số dòng
- Ra:
- Ý tưởng: Cho biến i chạy từ 0 -> dòng , nhập phần tử b+i

```
for(i = 0; i < dong ; i++)  
{  
    printf("b[%d] = ", i);  
    scanf("%f", b + i);  
}
```

6. Function XuatVecto

- Vào: Con trỏ b, số dòng
- Ra: Con trỏ b
- Ý tưởng: Cho biến i chạy từ 0 -> dòng in ra giá trị của con trỏ ở vị trí b+i

```
For (i = 0; i < dong ; i++)  
{  
    printf("b[%d] = ", i);  
    scanf("%f", b + i);  
}
```

7. Function XuatMaTran

- Vào: con trỏ a, số dòng, số cột
- Ra: Hiển thị ma trận ra màn hình
- Ý tưởng: Cho biến i chạy từ 0 -> dòng, j chạy từ 0-> cột, xuất ra màn hình giá trị của con trỏ ở vị trí a+i*cột+j;

```
For (i = 0; i < dong; i++)  
{  
    For (j = 0; j < cot; j++)  
        printf("%.3f ", *(a+i * cot + j));  
    printf("\n");  
}
```

8. Function Cheotroi

- Vào: Con trỏ file f, con trỏ a, dòng, cột
- Ra: Trả về 1 nếu là chéo trội.
- Ý tưởng:
 - Cho biến i chạy từ 0 -> dòng, với mỗi lần lặp i thì khởi tạo biến t=0 (biến t dùng để lưu tổng các phần tử cùng hàng trừ phần tử a_{ii} ; j chạy từ 0-> cột; nếu $i \neq j$ thì $t = t + |a_{ij}|$)
 - Nếu $t > |a_{ii}|$ thì lưu vào file và in ra thông báo “Ma trận A không chéo trội” ngược lại trả về 1;

```

for (i=0; i<dong; i++)
{
    t=0;
    for (j=0; j<cot; j++)
        if (i != j) t += fabs(a[i * cot + j]);
    if (t > fabs (a [i * cot + i]))
    {
        printf ("MA TRAN A KHONG CHEO TROI HANG\n");
        fprintf (f,"MA TRAN A KHONG CHEO TROI HANG\n");
        return 0;
    }
}
return 1;

```

9. Function norm_row

- Vào: Con trỏ a, số ẩn của hệ phương trình n
- Ra: Trả về chuẩn của ma trận hệ số hệ phương trình
- Ý tưởng: Tính tổng trị tuyệt đối các phần tử trên cùng 1 hàng => Chuẩn là tổng lớn nhất!

```

int i, j;
float norm = 0, sum[max];
for (i=0; i<n; i++)
{
    sum[i] = 0;
    for (j=0; j<n; j++)
        sum[i] += fabs (a [i*n + j]);
}
for (i=0; i<n; i++)
    if (sum[i] > norm) norm = sum[i];
return norm;

```

10. Function kt_hoi_tuDON

- Vào: Con trỏ file f, con trỏ a, con trỏ b, con trỏ alpha1, số dòng, số cột
- Ra: Trả về 1 nếu pp lặp hội tụ

- Ý tưởng: Tính chuẩn của ma trận alpha sử dụng hàm `norm_row`. Phương pháp lặp hội tụ khi chuẩn của ma trận alpha < 1

lamda = norm_row(alpha1, dong);

11. Function `kt_hoi_tuSEIDEL`

- Vào: Con trỏ file `f`, con trỏ `a`, con trỏ `b`, biến `h` là kết quả trả về của kiểm tra chéo trội, số dòng, số cột
- Ra: Trả về con trỏ ma trận alpha

*Function kt_hoi_tuSEIDEL (FILE *f, float *a, float *b, int h, int dong, int cot)*

{ float s[max], x[max], x0[max];

float lamda;

*float size = dong * cot * sizeof(int);*

*float*alpha = (float *) malloc (size);*

int i,j;

if (h==1)

{

for (i=0; i<dong; i++)

{ s [i] = 0;

for (j=0; j<cot; j++)

{

if (j != i)

*alpha [i*cot +j] = -(a+i * cot+j)/ *(a+i*cot+ i);*

}

*b[i] = b[i]/ *(a+i * cot + i);*

*alpha [i * cot +i] = 0;*

}

}

lamda= norm_row(alpha,dong);

if (lamda<1)

{ printf("\n >>>> PHUONG PHAP LAP HOI TU VI CHUAN MA TRAN

ALPHA = %.3f <1 <<<<<\n",lamda);

fprintf(f,"\n >>>> PHUONG PHAP LAP HOI TU VI CHUAN MA TRAN

ALPHA = %.3f <1 <<<<<\n",lamda);

}

printf("Ma tran alpha la:\n");

fprintf(f,"Ma tran alpha la:\n");

```

    for (i=0; i<dong; i++)
    { for (j=0; j<cot; j++)
      { printf("%f ",alpha[i * cot +j]);
        fprintf(f,"%f ",alpha[i * cot +j]);}
      printf("\n");
      fprintf(f,"\n");
    }

    printf("\nVector beta la: ");
    fprintf(f,"\nVector beta la: ");
    for (i=0; i<dong; i++)
    { printf("%f ",b[i]);
      fprintf(f,"%f ",b[i]);
    }
    printf("\n");  fprintf(f,"\n");
    return alpha;
}

```

12. Function loop_kDON

- Vào: Con trỏ file f, con trỏ a, con trỏ b, mảng lưu nghiệm x[], mảng lưu nghiệm ban đầu x0[], số ẩn của hệ phương trình, số lần lặp k.
 - Ra:
 - Ý tưởng:
- Khởi tạo mảng nghiệm ban đầu x₀ bằng vectơ b được lưu bằng con trỏ b, tạo mảng tạm tmp[] để lưu các giá trị nghiệm ở lần lặp cuối cùng
- Khởi tạo kt=0, sử dụng vòng lặp while... do....

```

    for (i=0; i<n; i++)
    {
        tmp[i] =0;
        x0[i] = b[i];
        printf(" %.3f ",x0[i]);
        fprintf(f," %.3f ",x0[i]);
    }
    while (kt == 0)

```

```

{
    dem += 1;
    printf("\nLan lap thu %d : ",dem);
    fprintf(f,"\nLan lap thu %d : ",dem);

    for (i=0; i<n ; i++)
    {
        tmp[i] = 0;
        for (j=0; j<n; j++)
            tmp[i] += a[i * n +j]*x0[j];
        tmp[i] = tmp[i]+b[i];
    }
    for (i=0; i<n; i++)
    {
        x[i]= tmp[i];
        printf(" %.20f ",x[i]);
    }
}

```

➤ Thiết lập và kiểm tra sai số:

```

    for (j=0; j<n; j++)
        if (fabs(x[j]-x0[j]) > ss)
            ss = fabs(x[j] - x0[j]) ;
    ss = ss * (lamda/(1-lamda));

    printf("\tSai so = %.15f",ss);
    fprintf(f,"\tSai so = %.15f",ss);

    if (dem==1)
    {
        for (j=0; j<n; j++)
            if (fabs(x[j]-x0[j]) > ss1)
                ss1 = fabs(x[j] - x0[j]) ;
        ss1 = ss1 * (pow(lamda,k)/(1-lamda));
    }

    if (dem >= k)
        kt=1;
    else
        for(int count=0; count<n; count++)
            x0[count] = x[count];
    end while;

```

13. Function loop_kSEIDEL

- Vào: Con trỏ file f, con trỏ a, con trỏ b, mảng lưu nghiệm x[], mảng lưu nghiệm ban đầu x₀[], số ẩn của hệ phương trình n, số lần lặp tìm nghiệm k.
- Ra:
- Ý tưởng:

- Tính chuẩn của ma trận alpha qua function norm_row
- Khởi tạo mảng nghiệm ban đầu x₀ và x bằng vectơ b được lưu bằng con trỏ b
- Khởi tạo kt=0, sử dụng vòng lặp while... do....

While (kt == 0) do :

Dem = dem+1;

For i = 1 to n do:

For j = 1 to n do:

$$x_i = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^n a_{ij} x_{0j} + b_i \right]$$

- Thiết lập và kiểm tra sai số:

for i = 1 to n do :

*ss ← || x(i) - x⁽⁰⁾(i) ||_∞ * λ / (λ-1)*

end for;

if dem ≥ k then :

kt = 1;

else x⁰(i) ← x(i)

if dem == 1 then:

for j=0 to n-1 do:

if (|x[j] - x₀[j]|) then:

*ss1 ← || x⁽¹⁾(j) - x⁽⁰⁾(j) ||_∞ * λ^k / (λ-1)*

14. Function loop_eps1

- Vào: con trỏ FILE, con trỏ a, con trỏ b, sai số epsilon , cỡ ma trận n
- Ra:

- Ý tưởng: Thực hiện lặp theo công thức lặp đơn đến khi sai số $< \epsilon$ thì dừng

```
void loop_eps1(FILE *f, float *a, float *b, float x[max], float x0[max], float epsi, int n)
```

```
{
    int i,j, kt = 0, dem = 0;
    float tmp[max];
    float ss=0,ss1=0, lamda;
    lamda = norm_row(a,n);
    printf("\nChuan ma tran alpha lamda = %.3f",lamda);
    //thuc hien lap den khi sai so < epsilon
    printf("\nVecto xap xi ban dau X0 la: ");
    fprintf(f,"\nVecto xap xi ban dau X0 la: ");
    for (i=0; i<n; i++)
    {
        x0[i] = b[i];
        printf(" %.8f ",x0[i]);
        fprintf(f," %.8f ",x0[i]);
    }
    while (kt == 0)
    {
        dem += 1;
        printf("\nLan lap thu %d : ",dem);
        fprintf(f,"\nLan lap thu %d : ",dem);
        for (i=0; i<n; i++)
            { tmp[i] = 0;
        for (j=0; j<n; j++)
            tmp[i] += a[i * n + j]*x0[j];
        tmp[i] = tmp[i] + b[i];
        }
        for (i=0; i<n; i++)
        {
            x[i] = tmp[i];
            printf(" %.20f ",x[i]);
            fprintf(f," %.20f ",x[i]);
        }
        for (j=0; j<n ; j++)
            if ( fabs(x[j]-x0[j]) > ss )
```

```

        ss = fabs(x[j] - x0[j]) ;
ss = ss * (lamda/(1-lamda));
printf("\tSai so = %.20f",ss);
fprintf(f,"\tSai so = %.20f",ss);

if (dem==1)
    { for (j=0; j<n; j++)
        if (fabs(x[j]-x0[j]) > ss1)
            ss1 = fabs(x[j] - x0[j]) ;
    }
if (ss < epsi)
    kt = 1;
else
    for(int count=0; count<n; count++)
        x0[count] = x[count];
}
ss1 = ss1 * (pow(lamda,dem)/(1-lamda));
}

```

15. Function loop_eps2

- Vào: Con trỏ file f, con trỏ a, con trỏ b, mảng lưu nghiệm lần thứ k x[], mảng lưu nghiệm lần k-1 x0[], số ẩn của hệ phương trình n, sai số epsilon.
- Ra:
- Ý tưởng:
- Tính chuẩn của ma trận alpha qua function norm_row
- Khởi tạo mảng nghiệm ban đầu x0 và x bằng vecto b được lưu bằng con trỏ b
- Khởi tạo kt=0, sử dụng vòng lặp while... do....

While (kt == 0) do :

dem = dem+1;

For i = 0 to n-1 do:

For j = 0 to n-1 do:

$$x_i = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^n a_{ij} x_{0j} + b_i \right]$$

- Thiết lập và kiểm tra sai số:

```

ss1 = 0; ss = 0;
for i = 0 to n-1 do :
    ss ← || x(i) - x(0)(i) ||∞ * λ / (λ-1)
end for;
if (ss < epsi) then :
    kt = 1;
else for i=0 to n-1 do:
    x0(i) ← x(i)
end for;
if dem==1 then:
    for j=0 to n-1 do:
        if (|x[j] - x0[j]| > ss1) then:
            ss1 ← || x(1)(j) - x(0)(j) ||∞
    end while;
ss1 = ss1 * λdem / (λ-1);

```

Code function hoàn chỉnh sau:

```

void loop_eps2(FILE *f,float *a, float *b, float x[max], float x0[max],float epsi, int n)
{
    int i, j, kt = 0, dem = 0;
    float sum1, sum2, ss=0,ss1=0, lamda;
    lamda = norm_row(a,n);
    printf("\nChuan ma tran alpha lamda = %.3f",lamda);
    //thuc hien lap den khi sai so < epsilon
    printf("\nVecto xap xi ban dau X0 la: ");
    fprintf(f,"\nVecto xap xi ban dau X0 la: ");
    for (i=0; i<n; i++)

```

```

    {
        x[i] = b[i];
        x0[i] = b[i];
        printf(" %.8f ",x0[i]);
        fprintf(f," %.8f ",x0[i]);
    }
while (kt == 0)
{
    dem += 1;
    printf("\nLan lap thu %d : ",dem);
    fprintf(f,"\nLan lap thu %d : ",dem);
    for (i=0; i<n; i++)
    {
        sum1 = 0;
        sum2 = 0;
        for (j=0; j<n; j++)
        {
            if (j<i)
                sum1 += a[i * n +j] * x[j];
            else if (j>i)
                sum2 += a[i * n +j] * x0[j];
        }
        x[i] = b[i] + sum1 + sum2;
        printf(" %.20f ",x[i]);
        fprintf(f," %.20f ",x[i]);

        for (j=0; j<n ; j++)
            if (fabs(x[j]-x0[j]) > ss)
                ss = fabs(x[j] - x0[j]) ;
    }
}

```



```

        ss = ss * (lamda/(1-lamda));
    }
    printf("\tSai so = %.20f",ss);
    fprintf(f,"\tSai so = %.20f",ss);

    if (dem==1)
    {
        for (j=0; j<n; j++)
            if (fabs(x[j]-x0[j]) > ss1)
                ss1 = fabs(x[j] - x0[j]) ;
    }

    if (ss < epsi)
        kt = 1;
    else
        for(int count=0; count<n; count++)
            x0[count] = x[count];
}

ss1 = ss1 * (pow(lamda,dem)/(1-lamda));
printf("\nSai so tien nghiem = %.20f",ss1);
fprintf(f,"\nSai so tien nghiem = %.20f",ss1);
printf("\nSai so hau nghiem sau %d lan lap la : %.20f",dem,ss);
fprintf(f,"\nSai so hau nghiem sau %d lan lap la : %.20f",dem,ss);
}

```

IV. KẾT QUẢ THỰC HIỆN CHƯƠNG TRÌNH

Sau đây là kết quả được đọc ra từ file “final_exam.txt” khi chạy toàn bộ chương trình cho ví dụ giải hệ phương trình sau :

$$\begin{cases} 1,02x_1 - 0,05x_2 - 0,1x_3 = 0,795 \\ -0,11x_1 + 1,03x_2 - 0,05x_3 = 0,849 \\ -0,11x_1 - 0,12x_2 + 1,04x_3 = 1,398 \end{cases}$$

-----CAC KET QUA THU DUOC LUU O DAY!!!-----

Ma tran A la:

1.020000 -0.050000 -0.100000
-0.110000 1.030000 -0.050000
-0.110000 -0.120000 1.040000

Vecto B la: 0.795000 0.849000 1.398000 2.Tinh chuan va kiem tra tinh hoi tu cua
phuong phap lap don

Ma tran alpha la:

-0.020000 0.050000 0.100000
0.110000 -0.030000 0.050000
0.110000 0.120000 -0.040000

Vector beta la: 0.795000 0.849000 1.398000

>>>>> PHUONG PHAP LAP DON HOI TU VI CHUAN MA TRAN ALPHA =
0.270 < 1 <<<<<

3.Tinh nghiem gan dung voi k cho truooc va kiem tra sai so hau nghiem(tien nghiem)
cho pp Lap don

Vecto xap xi ban dau X0 la: 0.795 0.849 1.398

Lan lap thu 1 : 0.96135002374649048000 0.98088002204895020000
1.53140997886657710000 Sai so = 0.061526704579592

Lan lap thu 2 : 0.97795802354812622000 1.00189256668090820000
1.56019783020019530000 Sai so = 0.022756448015571

Lan lap thu 3 : 0.98155528306961060000 1.00452852249145510000
1.56339466571807860000 Sai so = 0.008416767232120

Lan lap thu 4 : 0.98193478584289551000 1.00500500202178960000
 1.56397879123687740000 Sai so = 0.003113050246611
 Lan lap thu 5 : 0.98200947046279907000 1.00506162643432620000
 1.56405436992645260000 Sai so = 0.001151401898824
 Lan lap thu 6 : 0.98201835155487061000 1.00507187843322750000
 1.56406629085540770000 Sai so = 0.000425860896939
 Lan lap thu 7 : 0.98201990127563477000 1.00507318973541260000
 1.56406807899475100000 Sai so = 0.000157510163262
 Lan lap thu 8 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.000058257173805
 Lan lap thu 9 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.000021547170036
 Sai so tien nghiem = 0.000001737693310
 Sai so hau nghiem sau 9 lan lap = 0.000021547170036
 4.Lap don voi sai so cho truoc epsilon = 0.0000010000

Vecto xap xi ban dau X0 la: 0.79500002 0.84899998 1.39800000
 Lan lap thu 1 : 0.96135002374649048000 0.98088002204895020000
 1.53140997886657710000 Sai so = 0.06152670457959175100
 Lan lap thu 2 : 0.97795802354812622000 1.00189256668090820000
 1.56019783020019530000 Sai so = 0.02275644801557064100
 Lan lap thu 3 : 0.98155528306961060000 1.00452852249145510000
 1.56339466571807860000 Sai so = 0.00841676723212003710
 Lan lap thu 4 : 0.98193478584289551000 1.00500500202178960000
 1.56397879123687740000 Sai so = 0.00311305024661123750
 Lan lap thu 5 : 0.98200947046279907000 1.00506162643432620000
 1.56405436992645260000 Sai so = 0.00115140189882367850
 Lan lap thu 6 : 0.98201835155487061000 1.00507187843322750000
 1.56406629085540770000 Sai so = 0.00042586089693941176
 Lan lap thu 7 : 0.98201990127563477000 1.00507318973541260000
 1.56406807899475100000 Sai so = 0.00015751016326248646
 Lan lap thu 8 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00005825717380503193
 Lan lap thu 9 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00002154717003577389
 Lan lap thu 10 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000796950007497799
 Lan lap thu 11 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000294762276098481

Lan lap thu 12 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000109021641492291
 Lan lap thu 13 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000040323064354197
 Sai so tien nghiêm = 0.00000000923480758530
 Sai so hau nghiêm sau 13 lan lap la : 0.00000040323064354197
 5.Kiem tra tinh cheo troi cua ma tran A (pp lap Seidel)
 >>>>> MA TRAN A LA MA TRAN CHEO TROI HANG! <<<<<

6.Tinh chuan va kiem tra tinh hoi tu cua phuong phap lap Seidel

>>>>> PHUONG PHAP LAP SEIDEL HOI TU VI CHUAN MA TRAN ALPHA =
 0.221 <1 <<<<<
 Ma tran alpha la:
 0.000000 0.049020 0.098039
 0.106796 0.000000 0.048544
 0.105769 0.115385 0.000000

Vector beta la: 0.779412 0.824272 1.344231

7.Tinh nghiêm gan dung voi k cho truoc va kiem tra sai so hau nghiêm(tien nghiêm)
 cho pp Lap Seidel

Vecto xap xi ban dau X0 la: 0.779 0.824 1.344
 Lan lap thu 1 : 0.95160460472106934000 0.99115341901779175000
 1.55924510955810550000 Sai so = 0.06105345487594604500
 Lan lap thu 2 : 0.98086488246917725000 1.00471591949462890000
 1.56390488147735600000 Sai so = 0.00830847397446632390
 Lan lap thu 3 : 0.98198658227920532000 1.00506186485290530000
 1.56406342983245850000 Sai so = 0.00031850737286731601
 Lan lap thu 4 : 0.98201906681060791000 1.00507307052612300000
 1.56406807899475100000 Sai so = 0.00000922400340641616
 Lan lap thu 5 : 0.98202008008956909000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000028772120685971
 Lan lap thu 6 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000001692477624715
 Lan lap thu 7 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000000038748187992

Lan lap thu 8 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.000000000000887114947
 Lan lap thu 9 : 0.98202013969421387000 1.00507342815399170000
 1.56406831741333010000 Sai so = 0.00000000000020309930
 Sai so tien nghiêm = 0.00000034935453641083
 Sai so hau nghiêm sau 9 lan lap = 0.00000000000020309930
 8.Lap seidel voi sai so cho truoc epsilon = 0.0000100000

Vecto xap xi ban dau X0 la: 0.77941179 0.82427186 1.34423077
 Lan lap thu 1 : 0.95160460472106934000 0.99115341901779175000
 1.55924510955810550000 Sai so = 0.06105345487594604500
 Lan lap thu 2 : 0.98086488246917725000 1.00471591949462890000
 1.56390488147735600000 Sai so = 0.00830847397446632390
 Lan lap thu 3 : 0.98198658227920532000 1.00506186485290530000
 1.56406342983245850000 Sai so = 0.00031850737286731601
 Lan lap thu 4 : 0.98201906681060791000 1.00507307052612300000
 1.56406807899475100000 Sai so = 0.00000922400340641616
 Sai so tien nghiêm = 0.00066037982469424605
 Sai so hau nghiêm sau 4 lan lap la : 0.00000922400340641616