CHƯƠNG 9 : CÁC VẤN ĐỀ VỀ MA TRẬN

§1.ĐỊNH THỨC CỦA MA TRẬN

Cho một ma trận vuông cấp n.Ta cần tìm định thức của nó.Trước hết chúng ta nhắc lại một số tính chất quan trọng của định thức:

- nếu nhân tất cả các phần tử của một hàng (hay cột) với k thì định thức được nhân với k
- định thức không đổi nếu ta cộng thêm vào một hàng tổ hợp tuyến tính của các hàng còn lại.

Ta sẽ áp dụng các tính chất này để tính định thức của một ma trận cấp 4 như sau(phương pháp này có thể mở rộng cho một ma trận cấp n) bằng phương pháp trụ:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Lấy giá trị trụ là $p_1 = a_{11}$.Ta chia các phần tử của hàng thứ nhất cho $p_1 = a_{11}$ thì định thức sẽ là D/p_1 (theo tính chất 1) và ma trận còn lại là:

$$\begin{pmatrix} 1 & a'_{12} & a'_{13} & a'_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Lấy hàng 2 trừ đi hàng 1 đã nhân với a_{21},lấy hàng 3 trừ đi hàng 1 đã nhân với a_{31} và lấy hàng 4 trừ đi hàng 1 đã nhân với a_{41} (thay hàng bằng tổ hợp tuyến tính của các hàng còn lại) thì định thức vẫn là D/p_1 và ma trận là:

$$\begin{pmatrix} 1 & a'_{12} & a'_{13} & a'_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{pmatrix}$$

Lấy giá trị trụ là $p_2 = a'_{22}$.Ta chia các phần tử của hàng thứ hai cho p_2 thì định thức sẽ là $D/(p_1p_2)$ và ma trận còn lại là:

$$\begin{pmatrix} 1 & a'_{12} & a'_{13} & a'_{14} \\ 0 & 1 & a''_{23} & a''_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{pmatrix}$$

Lấy hàng 1 trừ đi hàng 2 đã nhân với a'_{12},lấy hàng 3 trừ đi hàng 2 đã nhân với a'_{32} và lấy hàng 4 trừ đi hàng 2 đã nhân với a'_{42} thì định thức vẫn là D/p_1 và ma trận là:
thì định thức vẫn là $D/(p_1p_2)$ và ma trận là:

$$\begin{pmatrix} 1 & 0 & a''_{13} & a''_{14} \\ 0 & 1 & a''_{23} & a''_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{pmatrix}$$

Tiếp tục lấy hàng 3 rồi hàng 4 làm trụ thì ma trận sẽ là:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Định thức của ma trận này là $D/(p_1p_2p_3p_4)$= $D/(a_{11}a'_{22}a''_{33}a'''_{44})$ =1 nên định thức của ma trận A là D = $p_1p_2p_3p_4$.

Sau đây là chương trình tìm định thức của một ma trận:

Chương trình 9-1

```
//tinh dinh thuc
#include <conio.h>#include <stdio.h>#include <ctype.h>
#include <stdlib.h>

void main()
  {
        int i,j,k,n,ok1,ok2,t;
        float d,c,e,f,g,h;
        float a[50][50];
        char tl;

        clrscr();
        printf("** TINH DINH THUC CAP n **");
        printf("\n");
        printf("\n");
        printf("Cho cap cua dinh thuc n = ");
        scanf("%d",&n);
        printf("Nhap ma tran a\n");
        for (i=1;i<=n;i++)
        {
          printf("Dong %d:\n",i);
                for (j=1;j<=n;j++)
                  {
                        printf("a[%d][%d] = ",i,j);
                        scanf("%f",&a[i][j]);
                  }
                printf("\n");
          }
        printf("\n");
        printf("Ma tran a ma ban da nhap\n");
        printf("\n");
        for (i=1;i<=n;i++)
          {
                for (j=1;j<=n;j++)
```

```
          printf("%.5f\t",a[i][j]);
        printf("\n");
    }
    printf("\n");
    t=1;
 flushall();
    while (t)
    {
        printf("Co sua ma tran a khong(c/k)?");
        scanf("%c",&tl);
        if (toupper(tl)=='C')
          {
            printf("Cho chi so hang can sua : ");
            scanf("%d",&i);
            printf("Cho chi so cot can sua : ");
            scanf("%d",&j);
            printf("a[%d][%d] = ",i,j);
            scanf("%f",&a[i,j]);
          }
        if (toupper(tl)=='K')
          t=0;
    }
    printf("Ma tran a ban dau\n");
    printf("\n");
    for (i=1;i<=n;i++)
    {
        for (j=1;j<=n;j++)
          printf("%.5f\t",a[i][j]);
        printf("\n");
    }
    printf("\n");
    d=1;
    i=1;
    ok2=1;
    while ((ok2)&&(i<=n))
    {
        if (a[i][i]==0)
          {
            ok1=1;
            k=k+1;
            while ((ok1)&&(k<=n))
              if (a[k,i]!=0)
                {
                 for (j=i;j<=n;j++)
                   {
                     c=a[i][j];
                     a[i][j]=a[k][j];
                     a[k][j]=c;
                   }
                 d=-d;
```

```
                ok1=0;
               }
         else
               k=k+1;
               if (k>n)
                {
                      printf("\n");
                      printf("** MA TRAN SUY BIEN **");
                      ok2=0;
                      d=0;
                }
      }
    if (a[i][i]!=0)
      {
          c=a[i][i];
          for (j=i+1;j<=n;j++)
            a[i][j]=a[i][j]/c;
            for (k=i+1;k<=n;k++)
               {
                c=a[k][i];
                for (j=i+1;j<=n;j++)
                      a[k][j]=a[k][j]-a[i][j]*c;
               }
      }
    i=i+1;
  }
  if (ok2)
    {
      for (i=1;i<=n;i++)
        d=d*a[i][i];
      printf("\n");
      printf("** GIA TRI DINH THUC D **");
      printf("\n");
      printf("%.3f",d);
    }
  getch();
}
```

§2.NGHỊCH ĐẢO MA TRẬN

Gọi $\mathbf{A}^{-1}$ là ma trận nghịch đảo của một ma trận **A** bậc n ta có $\mathbf{A}\mathbf{A}^{-1} = \mathbf{E}$.(trong biểu thức này **E** là một ma trận vuông có các phần tử trên đường chéo chính bằng 1). Dạng của ma trận E,ví dụ cấp 4,là:

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Phương pháp loại trừ để nhận được ma trận nghịch đảo $\mathbf{A}^{-1}$ được thực hiện qua nhiều giai đoạn (n),mỗi một giai đoạn gồm hai bước.Đối với giai đoạn thứ k:

- chuẩn hoá phần tử a_{kk} bằng cách nhân hàng với nghịch đảo của nó
- làm cho bằng không các phần tử phía trên và phía dưới đường chéo cho đến cột thứ k.Khi k = n thì $\mathbf{A}^{(k)}$ sẽ trở thành ma trận đơn vị và **E** trở thành $\mathbf{A}^{-1}$

Ví dụ: Tính ma trận nghịch đảo của ma trận

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

Ta viết lại ma trận A và ma trận đơn vị tương ứng với nó

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \quad E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Giai đoạn 1: ***Bước a***: Nhân hàng 1 với $1/a_{11}$,nghĩa là $a'_{1j} = a_{1j}/a_{11}$ đối với dòng thứ nhất,$a'_{ij} = a_{ij}$ đối với các dòng khác

$$A = \begin{pmatrix} 1 & 1/2 & 1/2 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \quad E = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Bước b: Trừ hàng 3 và hàng 2 cho hàng 1,nghĩa là $a^{(1)}_{1j} = a_{ij} - a_{i1}a_{ij}$ đối với $i \neq 1$.

$$A = \begin{pmatrix} 1 & 1/2 & 1/2 \\ 0 & 3/2 & 1/2 \\ 0 & 1/2 & 3/2 \end{pmatrix} \quad E = \begin{pmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/2 & 0 & 1 \end{pmatrix}$$

Giai đoạn 2: ***Bước a***: Lấy hàng 2 làm chuẩn,nhân hàng 2 với 2/3,để nguyên các hàng khác

$$A = \begin{pmatrix} 1 & 1/2 & 1/2 \\ 0 & 1 & 1/3 \\ 0 & 1/2 & 3/2 \end{pmatrix} \quad E = \begin{pmatrix} 1/2 & 0 & 0 \\ -1/3 & 2/3 & 0 \\ -1/2 & 0 & 1 \end{pmatrix}$$

Bước b: Lấy hàng 1 trừ đi hàng 2 nhân 1/2 và lấy hàng 3 trừ đi hàng 2 nhân 1/2

$$A = \begin{pmatrix} 1 & 0 & 1/3 \\ 0 & 1 & 1/3 \\ 0 & 0 & 4/3 \end{pmatrix} \quad E = \begin{pmatrix} 2/3 & -1/3 & 0 \\ -1/3 & 2/3 & 0 \\ -1/3 & -1/3 & 1 \end{pmatrix}$$

Giai đoạn 3: ***Bước a***: Lấy hàng 3 làm chuẩn,nhân hàng 3 với 3/4,để nguyên các hàng khác

$$A = \begin{pmatrix} 1 & 0 & 1/3 \\ 0 & 1 & 1/3 \\ 0 & 0 & 1 \end{pmatrix} \quad E = \begin{pmatrix} 2/3 & -1/3 & 0 \\ -1/3 & 2/3 & 0 \\ -1/4 & -1/4 & 3/4 \end{pmatrix}$$

Bước b: Lấy hàng 1 trừ đi hàng 3 nhân 1/3 và lấy hàng 2 trừ đi hàng 3 nhân 1/3

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad E = \begin{pmatrix} 3/4 & -1/4 & -1/4 \\ -1/4 & 3/4 & -1/4 \\ -1/4 & -1/4 & 3/4 \end{pmatrix}$$

Như vậy A^{-1} là:

$$A^{-1} = \begin{pmatrix} 3/4 & -1/4 & -1/4 \\ -1/4 & 3/4 & -1/4 \\ -1/4 & -1/4 & 3/4 \end{pmatrix}$$

Áp dụng phương pháp này chúng ta có chương trình sau:

Chương trình 9-2

```
#include <conio.h>#include <stdio.h>#include <math.h>#include <stdlib.h>#include
<ctype.h>void main() { int i,j,k,n,t,t1;float c,a[50][50],b[50][50]; char tl; clrscr();
	printf("	**MA TRAN NGHICH DAO** \n");
	printf("Cho bac cua ma tran n = ");
	scanf("%d",&n);
	printf("Vao ma tran ban dau a\n");
	for (i=1;i<=n;i++)
	  {
		printf("Vao hang thu %d :\n",i);
		for (j=1;j<=n;j++)
		  {
			printf("a[%d][%d] = ",i,j);
			scanf("%f",&a[i][j]);
		  }
		printf("\n");
	  }
	printf("\n");
	printf("Ma tran ban da nhap\n");
    printf("\n");
	for (i=1;i<=n;i++)
	  {
		for (j=1;j<=n;j++)
		  printf("%.5f\t",a[i][j]);
		printf("\n");
	  }
	t=1;
	flushall();
	while (t)
	  {
		printf("\nCo sua ma tran khong(c/k)?");
		scanf("%c",&tl);
		if(toupper(tl)=='C')
		  {
			printf("Cho chi so hang can sua : ");
			scanf("%d",&i);
			printf("Cho chi so cot can sua : ");
			scanf("%d",&j);
			printf("a[%d][%d] = ",i,j);
			scanf("%f",&a[i][j]);
		  }
		if (toupper(tl)=='K')
		  t=0;
	  }
	printf("\nMa tran ban dau\n");
```

```
printf("\n");
for (i=1;i<=n;i++)
  {
          for (j=1;j<=n;j++)
            printf("%.5f\t",a[i][j]);
          printf("\n");
  }
printf("\n");
for (i=1;i<=n;i++)
  for (j=n+1;j<=2*n;j++)
          {
            if (j==i+n)
                    a[i][j]=1;
            else
                    a[i][j]=0;
          }
  i=1;
  t1=1;
  while (t1&&(i<=n))
          {
            if (a[i][i]==0)
                    {
                     t=1;
                     k=i+1;
                     while (t&&(k<=n))
                            if (a[k][i]!=0)
                              {
                                    for (j=1;j<=2*n;j++)
                                      {
                                            c=a[i][j];
                                            a[i][j]=a[k][j];
                                      a[k][j]=c;
                                      }
                                    t=0;
                              }
                            else
                              k=k+1;
                            if (k==n+1)
                              {
                                    if (a[i][k-1]==0)
                                      {
                                            printf("MA TRAN SUY BIEN\n ");
                                            t1=0;
                                      }
                              }
                     }
                    if (a[i][i]!=0)
                     {
                            c=a[i][i];
                            for (j=i;j<=2*n;j++)
```

```
                        a[i][j]=a[i][j]/c;
                  }
                for (k=1;k<=n;k++)
                  {
                        if (k!=i)
                          {
                                c=a[k][i];
                                 for (j=i;j<=2*n;j++)
                                   a[k][j]=a[k][j]-a[i][j]*c;
                          }
                  }
                i=i+1;
          }
        if (t1)
          {
                printf("\n");
                printf("\nMA TRAN KET QUA\n");
                printf("\n");
                for (i=1;i<=n;i++)
                  {
                        for (j=n+1;j<=2*n;j++)
                          printf("%.4f\t\t",a[i][j]);
                        printf("\n");
                  }
                printf("\n");
          }
    getch();
}
```

Dùng chương trình tính nghịch đảo của ma trận:

$$\begin{pmatrix} 9 & 9 & 8 \\ 9 & 8 & 7 \\ 8 & 7 & 6 \end{pmatrix} \text{ cho ta kết quả } \begin{pmatrix} -1 & 2 & -1 \\ 2 & -10 & 9 \\ -1 & 9 & -9 \end{pmatrix}$$

§3.TÍCH HAI MA TRẬN

Giả sử ta có ma trận $\mathbf{A}_{mn}$ và ma trận $\mathbf{B}_{np}$.Tích của $\mathbf{A}_{mn}$ và $\mathbf{B}_{np}$ là ma trận $\mathbf{C}_{mp}$ trong đó mỗi phần tử của $\mathbf{C}_{mp}$ là: $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

Chương trình dưới đây thực hiện nhân hai ma trận với nhau.

Chương trình 9-3

```
#include <conio.h>#include <stdio.h>#include <math.h>#include <stdlib.h>#include <ctype.h>

#define max  50

void main()
```

```
{
        int n,l,m,i,j,k,t;
        float a[max][max],b[max][max],c[max][max];
        char tl;

        clrscr();
        printf("Cho so hang cua ma tran a : ");
        scanf("%d",&n);
        printf("Cho so cot cua ma tran a : ");
        scanf("%d",&l);
        printf("Cho so cot cua ma tran b : ");
        scanf("%d",&m);
        printf("\nNHAP MA TRAN A\n");
        for (i=1;i<=n;i++)
          for (j=1;j<=l;j++)
                {
                  printf("a[%d][%d] = ",i,j);
                  scanf("%f",&a[i][j]);
                }
        printf("\n");
        printf("Ma tran a ma ban da nhap\n");
        for (i=1;i<=n;i++)
          {
                for (j=1;j<=l;j++)
                  printf("%10.5f",a[i][j]);
                printf("\n");
          }
        flushall();
        t=1;
        while (t)
          {
                printf("Co sua ma tran khong(c/k)?");
                scanf("%c",&tl);
                if (toupper(tl)=='C')
                  {
                        printf("Cho chi so hang can sua : ");
                        scanf("%d",&i);
                        printf("Cho chi so cot can sua : ");
                        scanf("%d",&j);
                        printf("a[%d][%d] = ",i,j);
                        scanf("%f",&a[i][j]);
                  }
                if (toupper(tl)=='K')
                  t=0;
          }
        printf("Ma tran a ban dau");
        printf("\n");
        for (i=1;i<=n;i++)
          {
                for (j=1;j<=l;j++)
```

```
            printf("%10.5f",a[i][j]);
          printf("\n");
  }
printf("\n");

printf("NHAP MA TRAN B\n");
for (i=1;i<=l;i++)
  for (j=1;j<=m;j++)
          {
            printf("b[%d][%d] = ",i,j);
            scanf("%f",&b[i][j]);
          }
printf("\n");
printf("Ma tran b ban da nhap\n");
for (i=1;i<=l;i++)
  {
          for (j=1;j<=m;j++)
            printf("%10.5f",b[i][j]);
          printf("\n");
  }
flushall();
t=1;
while (t)
  {
          printf("Co sua ma tran khong(c/k)?");
          scanf("%c",&tl);
          if (toupper(tl)=='C')
            {
                    printf("Cho chi so hang can sua : ");
                    scanf("%d",&i);
                    printf("Cho chi so cot can sua : ");
                    scanf("%d",&j);
                    printf("b[%d][%d] = ",i,j);
                    scanf("%f",&b[i][j]);
            }
          if (toupper(tl)=='K')
            t=0;
  }
printf("Ma tran b ban dau");
printf("\n");
for (i=1;i<=l;i++)
  {
          for (j=1;j<=m;j++)
            printf("%10.5f",b[i][j]);
          printf("\n");
  }
printf("\n");
for (i=1;i<=n;i++)
  for (j=1;j<=m;j++)
          {
```

```
            c[i][j]=0;
            for (k=1;k<=l;k++)
                c[i][j]=c[i][j]+a[i][k]*b[k][j];
          }
      printf("Ma tran tich c :\n");
      for (i=1;i<=n;i++)
        {
          for (j=1;j<=m;j++)
            printf("%10.5f",c[i][j]);
          printf("\n");
        }
      getch();
  }
```

Dùng chương trình tính tính hai ma trận ta nhận được kết quả

$$\begin{pmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 0 \\ 5 & 3 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & -2 \\ 3 & -4 & 3 \end{pmatrix} = \begin{pmatrix} 5 & 0 & -1 \\ 8 & -14 & 11 \\ 1 & 2 & -2 \\ 14 & -2 & -1 \end{pmatrix}$$

§4.GIÁ TRỊ RIÊNG VÀ VEC TƠ RIÊNG CỦA MA TRẬN

1.Khái niệm chung: Trong nghiên lí thuyết và ứng dụng,ta gặp bài toán về ma trận cấp n.Cho một ma trận A cấp n,giá trị λ được gọi là giá trị riêng và vectơ X được gọi là vectơ riêng của ma trận A nếu:

$$AX = \lambda X \qquad (1)$$

Vectơ riêng phải là vectơ khác không.Tương ứng với một giá trị riêng có vô số vectơ riêng.Nếu X là một véc tơ riêng tương ứng với giá trị riêng λ thì cX cũng là vec tư riênh ứng với λ.Có nhiều thuật toán tìm giá trị riêng và vectơ riêng của một ma trận.Giả sử ta có ma trận A,gọi E là ma trận đơn vị thì theo (1) ta có:

$$(A-\lambda E)X = 0 \qquad (2)$$

và (A - λE) là ma trận có dạng:

$$\begin{pmatrix} a_{11}-\lambda \; a_{12} \cdots \; a_{1n} \\ a_{21} \; a_{22}-\lambda \; \cdots \; a_{2n} \\ \ldots\ldots. \\ a_{n1} \; a_{n2} \cdots \; a_{nn}-\lambda \end{pmatrix} \qquad (3)$$

Như vậy do (2) là hệ phương trình tuyến tính thuần nhất nên điều kiện cần và đủ để λ là giá trị riêng của ma trận trên là định thức của nó bằng không:

$$\det(A - \lambda E) = 0 \qquad (4)$$

Phương trình (4) được gọi là phương trình đặc trưng của ma trận A.Định thức det(A - λE) được gọi là định thức đặc trưng của ma trận A.Định thức $P_A(\lambda)$ của ma trận trên được gọi là đa thức đặc trưng của ma trận vuông A.

Ví dụ tìm vec tơ riêng và trị riêng của ma trận:

$$\begin{pmatrix} 3 & 1 & -3 \\ 3 & 1 & -1 \\ 2 & -2 & 0 \end{pmatrix}$$

Trước hết ta tính đa thức đặc trưng của ma trận A:

$$P_A(\lambda)=\begin{pmatrix}3-\lambda & 1 & -3\\ 3 & 1-\lambda & -1\\ 2 & -2 & -\lambda\end{pmatrix}=(4-\lambda)\ (\lambda^2+4)$$

Nghiệm của $P_A(\lambda) = 0$ là $\lambda_1 = 4, \lambda_2 = 2j$ và $\lambda_3 = -2j$.Vì trường cơ sở là số thực nên ta chỉ lấy $\lambda = 4$.Để tìm vec tơ riêng tương ứng với $\lambda = 4$ ta giải hệ

$$\begin{pmatrix}3-\lambda & 1 & -3\\ 3 & 1-\lambda & -1\\ 2 & -2 & -\lambda\end{pmatrix}\times\begin{pmatrix}\xi_1\\ \xi_2\\ \xi_3\end{pmatrix}=0$$

ta nhận được các giá trị của ξ,chúng tạo thành vec tơ riêng ứng với λ.
Như vậy khi khai triển định thức ta có một đa thức bậc n có dạng:

$$P_n(\lambda) = \lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \ldots - p_n = 0$$

Muốn xác định các hệ số của đa thức đặc tính này ta dùng phương pháp Fadeev-Leverrier.Ta xét ma trận A:

$$A=\begin{pmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ a_{n1} & a_{n2} & \cdots & a_{nn}\end{pmatrix}$$

Ta gọi vết của ma trận A là số:

$$\mathrm{vet}(A)= a_{11} + a_{22} +...+ a_{nn}$$

Khi đó tham số p_i của $P_n(\lambda)$ được các định như sau:

$p_1 = \mathrm{vet}(B_1)$ với $B_1 = A$
$p_2 = (1/2)\mathrm{vet}(B_2)$ với $B_2 = A(B_1-p_1E)$
$p_3 = (1/3)\mathrm{vet}(B_3)$ với $B_3 = A(B_2-p_2E)$
......

Chương trình tính các hệ số p_i như sau:

Chương trình 9-4

```
// Faddeev_Leverrier;
#include <stdio.h>
#include <conio.h>
#include <ctype.h>

#define max 50

void main()
  {

        int i,j,k,m,n,k1,t;
        float vet,c1,d;
        char tl;
        float p[max];
        float a[max][max],b[max][max],c[max][max],b1[max][max];

        clrscr();
        printf("Cho bac cua ma tran n = ");
        scanf("%d",&n);
        printf("Cho cac phan tu cua ma tran a : \n");
```

```
for (i=1;i<=n;i++)
  for (j=1;j<=n;j++)
          {
            printf("a[%d][%d] = ",i,j );
            scanf("%f",&a[i][j]);
          }
printf("\n");
clrscr();
printf("Ma tran ban da nhap");
printf("\n");
for (i=1;i<=n;i++)
  {
          for (j=1;j<=n;j++)
            printf("%10.5f",a[i][j]);
          printf("\n");
  }
t=1;
flushall();
while (t)
  {
          printf("\n");
          printf("Co sua ma tran khong(c/k)?");
          scanf("%c",&tl);
          if (toupper(tl)=='C')
            {
                  printf("Cho chi so hang can sua : ");
                  scanf("%d",&i);
                  printf("Cho chi so cot can sua : ");
                  scanf("%d",&j);
                  printf("a[%d][%d] = ",i,j);
                  scanf("%f",&a[i][j]);
                  flushall();
            }
          if (toupper(tl)=='K')
            t=0;
  }
printf("Ma tran ban dau");
printf("\n");
for (i=1;i<=n;i++)
  {
          for (j=1;j<=n;j++)
            printf("%10.5f",a[i][j]);
          printf("\n");
  }
for (i=1;i<=n;i++)
  for (j=1;j<=n;j++)
          b[i][j]=a[i][j];
for (k=1;k<=n-1;k++)
  {
          vet=0.0;
```

```
            for (i=1;i<=n;i++)
              vet+=b[i][i];
            p[k]=vet/k;
            for (i=1;i<=n;i++)
              for (j=1;j<=n;j++)
                  {
                    if (j!=i)
                        c[i][j]=b[i][j];
                    if (j==i)
                        c[i][j]=b[i][j]-p[k];
                  }
            for (i=1;i<=n;i++)
              for (j=1;j<=n;j++)
                  {
                    b[i][j]=0.0;
                    for (k1=1;k1<=n;k1++)
                        b[i][j]+=a[i][k1]*c[k1][j];
                  }
        }
      vet=0.0;
      for (i=1;i<=n;i++)
        vet+=b[i][i];
      p[n]=vet/n;
      printf("\n");
      printf("Cac he so cua da thuc dac trung\n");
      printf("\n");
      d=1.0;
      printf("%6.2f",d);
      for (i=1;i<=n;i++)
        {
            c1=-p[i];
            printf("%5c%6.2f",' ',c1);
        }
      getch();
}
```

2.Phương pháp Mises: Thuật toán Mises tìm giá trị riêng lớn nhất của một ma trận A. Nếu ma trận A là thực và và mỗi trị riêng bội k có đủ k vec tơ riêng độc lập tuyến tính thì việc tính toán sẽ cho ta giá trị riêng lớn nhất.

Một vectơ V bất kì có thể được viết dưới dạng:

$$V = v_1X_1 + v_2X_2 + \cdots + v_nX_n = \sum_{i=1}^{n} v_iX_i \qquad (5)$$

Trong đó $X_1,X_2,..,X_n$ là các vec tơ riêng tương ứng với các giá trị riêng $\lambda_1,\lambda_2,\lambda_3,..,\lambda_n$ và $v_1,v_2,v_3,...,v_n$ là các hằng số.

Khi nhân A với V ta có:

$AV = Av_1X_1 + Av_2X_2 +....+ Av_nX_n$

do: $Av_1X_1 = v_1AX_1 = v_1\lambda_1X_1$; $Av_2X_2 = v_2AX_2 = v_2\lambda_2X_2$ v.v.

Vậy nên: $AV = v_1\lambda_1X_1 + v_2\lambda_2X_2 +...+ v_n\lambda_nX_n$

$$AV = \sum_{i=1}^{n} v_i A_i X_i = \sum_{i=1}^{n} v_i \lambda_i X_i$$

Lại nhân biểu thức trên với A ta có:

$$A^2 V = v_1\lambda_1 AX_1 + v_2\lambda_2 AX_2 + ... + v_n\lambda_n AX_n$$
$$= v_1\lambda^2{}_1 X_1 + v_2\lambda_2{}^2 X_2 + ... + v_n\lambda_n{}^2 X_n$$

và tiếp đến lần thứ p ta có:

$$A^p V = \sum_{i=1}^{n} v_i \lambda_i^p X_i = v_1\lambda_1^p X_1 + v_1\lambda_2^p X_2 + \cdots + v_n\lambda_n^p X_n$$

Lấy $\lambda^p{}_1$ làm thừa số chung ta có:

$$A^p V = \lambda_1^p \left[v_1 X_1 + v_2 \left(\frac{\lambda_2}{\lambda_1}\right)^p X_2 + v_3 \left(\frac{\lambda_3}{\lambda_1}\right)^p X_3 + \cdots + v_n \left(\frac{\lambda_n}{\lambda_1}\right)^p X_n \right]$$

Tương tự ta có:

$$A^{p+1} V = \lambda_1^{p+1} \left[v_1 X_1 + v_2 \left(\frac{\lambda_2}{\lambda_1}\right)^{p+1} X_2 + v_3 \left(\frac{\lambda_3}{\lambda_1}\right)^{p+1} X_3 + \cdots + v_n \left(\frac{\lambda_n}{\lambda_1}\right)^{p+1} X_n \right]$$

Khi p rất lớn,vì $\lambda_1 > \lambda_2 > \lambda_3 >,...,\lambda_n$ nên:

$$\left(\frac{\lambda_i}{\lambda_1}\right) \to 0 \quad \text{khi} \quad p \to \infty$$

Do đó: $$\lim_{p\to\infty} A^p V = \lambda_1^p v_1 X_1 \qquad (6)$$

$$\lim_{p\to\infty} A^{p+1} V = \lambda_1^{p+1} v_1 X_1$$

nghĩa là khi p đủ lớn thì:

$$A^p V = \lambda_1^p v_1 X_1$$
$$A^{p+1} V = \lambda_1^{p+1} v_1 X_1$$

do đó: $$A^{p+1} V = \lambda_1 A^p V$$

hay: $$A\left(A^p V\right) = \lambda_1 A^p V$$

Như vậy $A^p V$ là véc tơ riêng của A ứng với λ_1 còn giá trị riêng λ_1 sẽ là:

$$\lim_{p\to\infty} \frac{A^{p+1} V}{A^p V} = \lambda_1$$

Trong thực tế để tránh vượt quá dung lượng bộ nhớ khi λ_1 khá lớn,các vectơ V_k được chuẩn hoá sau mỗi bước bằng cách chia các phần tử của nó cho phần tử lớn nhất m_k và nhận được vectơ V'_k

Như vậy các bước tính sẽ là:

- cho một vec tơ V bất kì (có thể là $V = \{ 1,1,1,...,1\}^T$)
- tính $V_1 = AV$ và nhận được phần tử lớn nhất là m_{1j} từ đó tính tiếp $V'_1 = V_1/m_{1j}$

Một cách tổng quát,tại lần lặp thứ p ta nhận được vectơ V_p và phần tử lớn nhất m_{pj} thì $V'_p = V_p/ m_{pj}$.

- tính $V_{p+1} = AV'_p$ với $v_{p+1,j}$ là phần tử thứ j của V_{p+1}.Ta có:

$$\begin{cases} \lim_{p\to\infty} V'_p = X_1 \\ \lim_{p\to\infty} v_{p+1,j} = \lambda_1 \end{cases}$$

Ví dụ: Tìm giá trị riêng lớn nhất và vec tơ riêng tương ứng của ma trận:

$$A = \begin{pmatrix} 17 & 24 & 30 & 17 \\ 8 & 13 & 20 & 7 \\ 2 & 10 & 8 & 6 \\ -23 & -43 & -54 & -26 \end{pmatrix}$$

Chọn V= $\{1,1,1,1\}^T$ ta tính được

	V	$V_1 = AV$	V'_1	$V_2 = AV'_1$	V'_2
	1 1 1 1	88 48 26 -146	-0.6027 -0.3288 -0.1781 1	-6.4801 -5.6580 0.0818 11.6179	-0.5578 -0.4870 0.0070 1
λ				11.6179	
	$V_3 = AV'_2$	V'_3	$V_4 = AV'_3$	V'_4	$V_5 = AV'_4$
	-3.9594 -3.6526 0.0707 7.3902	-0.5358 -0.4942 0.0096 1	-3.6823 -3.5196 0.0630 7.0573	-0.5218 -0.4987 0.0089 1	-3.5718 -3.4791 0.0408 6.9638
λ	7.3902		7.0573		6.9638

	V'_5	$V_6 = AV'_5$	V'_6	$V_7 = AV'_6$	V'_7
	-0.5129 -0.4996 0.0059 1	-3.5341 -3.4809 0.0250 6.9634	-0.5075 -0.4999 0.0036 1	-3.5173 -3.4868 0.0147 6.9742	-0.5043 -0.5000 0.0021 1
λ		6.9634		6.9742	

Dùng thuật toán trên ta có chương trình sau:

Chương trình 9-5

```
#include <conio.h>#include <stdio.h>#include <math.h>#include <stdlib.h>#include
<ctype.h>#define max 50void main() { int i,j,k,n,t; char tl; float t0,t1,epsi,s;
     float a[max][max];
     float x0[max],x1[max];

     clrscr();
     printf("Phuong phap lap luy thua tim tri rieng lon nhat\n");
     printf("Cho so hang va cot cua ma tran n = ");
     scanf("%d",&n);
     printf("Cho cac phan tu cua ma tran a : \n");
     for (i=1;i<=n;i++)
       for (j=1;j<=n;j++)
```

```
            {
             printf("a[%d][%d] = ",i,j);
             scanf("%f",&a[i][j]);
            }
     printf("\n");
     printf("Ma tran ban da nhap\n");
printf("\n");
     for (i=1;i<=n;i++)
       {
            for (j=1;j<=n;j++)
             printf("%15.5f",a[i][j]);
        printf("\n");
       }
     flushall();
     t=1;
     while (t)
       {
            printf("\nCo sua ma tran khong(c/k)?");
            scanf("%c",&tl);
            if (toupper(tl)=='C')
              {
                    printf("Cho chi so hang can sua : ");
                    scanf("%d",&i);
                    printf("Cho chi so cot can sua : ");
                    scanf("%d",&j);
                    printf("a[%d][%d] = ",i,j);
                    scanf("%f",&a[i][j]);
              }
            if (toupper(tl)=='K')
             t=0;
       }
     epsi=1e-5;
     printf("\nMa tran ban dau\n");
     printf("\n");
     for (i=1;i<=n;i++)
       {
            for (j=1;j<=n;j++)
             printf("%15.5f",a[i][j]);
            printf("\n");
       }
     printf("\n");
     for (i=1;i<=n;i++)
       x0[i]=1;
     k=1;
     t=0;
     t1=0;
     do
        {
            t0=t1;
            for (i=1;i<=n;i++)
```

```
                {
                        x1[i]=0;
                        for (j=1;j<=n;j++)
                          x1[i]=x1[i]+a[i][j]*x0[j];
                }
              s=0;
              j=0;
              for (i=1;i<=n;i++)
                if (s<fabs(x1[i]))
                        {
                          j=i;
                          s=fabs(x1[i]);
                        }
              t1=x1[j];
              for (i=1;i<=n;i++)
                x1[i]=x1[i]/t1;
              if (fabs(t1-t0)<epsi)
                {
                        printf("Da thuc hien %d buoc lap\n",k);
                        printf("Gia tri rieng lon nhat Vmax = %15.5f\n",t1);
                        printf("Vec to rieng tuong ung\n");
                        for (i=1;i<=n;i++)
                          printf("%.5f\n",x1[i]);
                        t=1;
                }
              if (fabs(t1-t0)>epsi)
                {
                        for (i=1;i<=n;i++)
                          x0[i]=x1[i];
                        k=k+1;
                }
              if (k>max)
                t=1;
          }
        while(t==0);
        getch();
  }
```

Dùng chương trình này tính gía trị riêng và vec tơ riêng của ma trận:

$$\begin{pmatrix} 2 & -1 & 0 \\ 9 & 4 & 6 \\ -8 & 0 & -3 \end{pmatrix}$$

ta nhận được giá trị riêng là 3.0000 và vec tơ riêng là x = { -0.75 ; 0.75 ; 1 }^{T}

Như chúng ta đã nói trước đây,phương pháp Mises (hay còn gọi là phương pháp lặp lũy thừa) chỉ cho phép tìm giá trị riêng lớn nhất và vec tơ riêng tương ứng của ma trận.Để xác định các giá trị riêng khác,ma trận A được biến đổi thành một ma trận khác A_1 mà các giá trị riêng là $\lambda_2 > \lambda_3 > ... > \lambda_n$.Phương pháp này gọi là phương pháp xuống thang.Sau đây là phương pháp biến đổi ma trận:

Giả sử X_1 là vec tơ riêng của ma trận A tương ứng với giá trị riêng λ_1 và W_1 là vec tơ riêng của ma trận A^T tương ứng với giá trị riêng λ_1.Từ định nghĩa $AX_1 = \lambda_1 X_1$ ta viết:

$$(A - \lambda E)X_1 = 0$$

Ta tạo ma trận A_1 dạng:

$$A_1 = A - \frac{\lambda_1}{W_1^T X_1} X_1 W_1^T \qquad (7)$$

Ta chú ý là $X_1 W_1^T$ là một ma trận còn $W_1^T X_1$ là một con số.Khi nhân hai vế của biểu thức (7) với X_1 và chý ý đến tính kết hợp của tích các ma trận ta có:

$$A_1 X_1 = AX_1 - \frac{\lambda_1}{W_1^T X_1} X_1 W_1^T X_1 \qquad (8)$$
$$= AX_1 - \lambda_1 X_1 \frac{W_1^T X_1}{W_1^T X_1}$$
$$= AX_1 - \lambda_1 X_1$$
$$= 0$$

A_1 chấp nhận giá trị riêng bằng không.

Nếu X_2 là vec tơ riêng tương ứng với giá trị riêng λ_2,thì khi nhân A_1 với X_2 ta có:

$$A_1 X_2 = AX_2 - \frac{\lambda_1}{W_1^T X_1} X_1 W_1^T X_2 \qquad (9)$$
$$= AX_2 - \lambda_1 X_1 \frac{W_1^T X_2}{W_1^T X_1}$$

Theo định nghĩa vì W_1 là vectơ riêng của A^T nên:

$$\lambda_1 W_1 = A^T W_1 \qquad (10)$$

Mặt khác do:

$$(AX)^T = X^T A^T \text{ và } (A^T)^T = A$$

Nên khi chuyển vị (10) ta nhận được:

$$(A^T W_1)^T = \lambda_1 W^T_1$$

Hay:

$$W_1^T A = \lambda_1 W_1^T \qquad (11)$$

Khi nhân (11) với X_2 ta có:

$$\lambda_1 W_1^T X_2 = W_1^T A X_2$$

và do định nghĩa:

$$AX_2 = \lambda_2 X_2$$

nên:

$$\lambda_1 W_1^T X_2 = W_1^T \lambda_2 X_2$$

vậy thì:

$$(\lambda_1 - \lambda_2)\, W_1^T X_2 = 0$$

khi $\lambda_1 \neq \lambda_2$ thì:

$$W_1^T X_2 = 0 \qquad (12)$$

Cuối cùng thay (12) vào (9) ta có:

$$A_1 X_2 = AX_2 = \lambda_2 X_2$$

Như vậy λ_2 là giá trị riêng lớn nhất của ma trận A_1 và như vậy có thể áp dụng thuật toán này để tìm các giá trị riêng còn lại của ma trận.Các bước tính toán như sau

- khi đã có λ_1 và X_1 ta tìm W_1 là vec tơ riêng của A^T ứng với giá trị riêng λ_1 (ví dụ tìm W_1 bằng cách giải phương trình $(A^T - \lambda_1 E)W_1 = 0$).Từ đó tính ma trận A_{12} theo (7).

- tìm giá trị riêng và vec tơ riêng của A_1 bằng cách lặp công suất và cứ thế tiếp tục và xuống thang (n-1) lần ta tìm đủ n giá trị riêng của ma trận A.

Ví dụ: Tìm giá trị riêng và vectơ riêng của ma trận sau:

$$A = \begin{pmatrix} 17 & 24 & 30 & 17 \\ 8 & 13 & 20 & 7 \\ 2 & 10 & 8 & 6 \\ -23 & -43 & -54 & -26 \end{pmatrix}$$

Ta đã tìm được giá trị riêng lớn nhất $\lambda_1 = 7$ và một vectơ riêng tương ứng:

$X_1 = \{ 1,1,0,-2\}^T$.

Ma trận A^T có dạng:

$$A^T = \begin{pmatrix} 17 & 8 & 2 & -23 \\ 24 & 13 & 10 & -43 \\ 30 & 20 & 8 & -54 \\ 17 & 7 & 6 & -26 \end{pmatrix}$$

và theo phương trình A^T-λ_1E)$W_1 = 0$ ta tìm được vectơ $W_1 = \{293,695,746,434\}^T$

Ta lập ma trận mới A_1 theo (7):

$$\lambda_1 \frac{X_1 W_1^T}{W_1^T X_1} = \frac{7}{120}\begin{pmatrix} 293 & 695 & 746 & 434 \\ 293 & 695 & 746 & 434 \\ 0 & 0 & 0 & 0 \\ -586 & -1390 & -1492 & -868 \end{pmatrix}$$

và:

$$A_1 = \begin{pmatrix} -0.0917 & -16.5417 & -13.5167 & -8.3167 \\ -9.0917 & -27.5417 & -23.5167 & -18.3167 \\ 2 & 10 & 8 & 6 \\ 11.1833 & 38.0833 & 33.0333 & 24.6333 \end{pmatrix}$$

Từ ma trận A_1 ta tìm tiếp được λ_2 theo phép lặp luỹ thừa và sau đó lại tìm ma trận A_3 và tìm giá trị riêng tương ứng.

Chương trình lặp tìm các giá trị riêng và vec tơ riêng của ma trận như sau:

Chương trình 9-6

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max  50

void main()
  {
        float a[max][max],vv[max][max],at[max][max];
        float x[max],y[max],vd[max];
        int i,j,k,n,l,t;
        float vp,v1,z,epsi,va,ps;
        char tl;

        clrscr();
    epsi=0.000001;
        printf("Cho bac cua ma tran n = ");
        scanf("%d",&n);
        printf("Cho cac phan tu cua ma tran a : \n");
        for (i=1;i<=n;i++)
          for (j=1;j<=n;j++)
```

```
          {
            printf("a[%d][%d] = ",i,j);
            scanf("%f",&a[i][j]);
          }
  printf("\n");
  clrscr();
  printf("Ma tran ban da nhap");
  printf("\n");
  for (i=1;i<=n;i++)
    {
          for (j=1;j<=n;j++)
            printf("%15.5f",a[i][j]);
          printf("\n");
    }
  t=1;
  flushall();
  while (t)
    {
          printf("\n");
          printf("Co sua ma tran khong(c/k)?");
          scanf("%c",&tl);
          if (toupper(tl)=='C')
              {
                    printf("Cho chi so hang can sua : ");
                    scanf("%d",&i);
                    printf("Cho chi so cot can sua : ");
                    scanf("%d",&j);
                    printf("a[%d][%d] = ",i,j);
                    scanf("%f",&a[i][j]);
            }
          if (toupper(tl)=='K')
            t=0;
    }
  for (l=1;l<=n;l++)
    {
          for (i=1;i<=n;i++)
            x[i]=1;
          vp=1.23456789;
          k=0;
          for (k=1;k<=40;k++)
            {
                    for (i=1;i<=n;i++)
                      {
                              y[i]=0;
                              for (j=1;j<=n;j++)
                                y[i]=y[i]+a[i][j]*x[j];
                      }
                    v1=y[1]/x[1];
                    z=0;
                    for (i=1;i<=n;i++)
```

```
            if (fabs(y[i])>z)
                    z=y[i];
          for (i=1;i<=n;i++)
            x[i]=y[i]/z;
          if (fabs(vp-v1)<epsi)
            break;
          vp=v1;
  }
  {
          printf("Gia tri rieng : %9.6f\n",v1);
          printf("Vec to rieng : \n");
            for (i=1;i<=n;i++)
                    printf("%.5f\n",x[i]);
          printf("\n");
          getch();
  }
vd[l]=v1;
va=v1;
for (i=1;i<=n;i++)
  vv[l][i]=x[i];
for (i=1;i<=n;i++)
  for (j=1;j<=n;j++)
          at[i][j]=a[j][i];
for (i=1;i<=n;i++)
  x[i]=1;
vp=1.23456;
k=0;
for (k=1;k<=40;k++)
  {
          for (i=1;i<=n;i++)
            {
                    y[i]=0;
                    for (j=1;j<=n;j++)
                      y[i]=y[i]+at[i][j]*x[j];
            }
          v1=y[1]/x[1];
          z=0;
          for (i=1;i<=n;i++)
            if (fabs(y[i])>z)
                    z=y[i];
          for (i=1;i<=n;i++)
            x[i]=y[i]/z;
          if (fabs(vp-v1)<epsi)
            break;
          vp=v1;
  }
if (fabs(vp-v1)>epsi)
  {
          printf("Khong hoi tu sau 40 lan lap\n");
          getch();
```

```
                exit(1);
          }
        if (fabs(va-v1)>3*epsi)
            {
                printf("Co loi\n");
                getch();
                exit(1);
          }
        ps=0;
        for (i=1;i<=n;i++)
          ps=ps+x[i]*vv[l][i];
        ps=v1/ps;
        for (i=1;i<=n;i++)
          for (j=1;j<=n;j++)
                a[i][j]=a[i][j]-ps*vv[l][i]*x[j];
    }
}
```

Do (6) ®óng víi mäi n nªn cho n = 1 , 2 , 3 , . . . ta cã :

$$| x_2 - x_1 | \le q | x_1 - x_o |$$
$$| x_3 - x_2 | \le q | x_2 - x_1 |$$
$$. . . . .$$
$$| x_{n+1} - x_n | \le q | x_n - x_{n-1} |$$

§iÒu nµy cã nghÜa lµ d·y $x_{i+1} - x_i$, mét c¸ch gÇn ®óng,lµ mét cÊp sè nh©n . Ta còng coi r»ng d·y x_n - y víi y lµ nghiÖm ®óng cña (1) , gÇn ®óng nh mét cÊp sè nh©n cã c«ng sai q . Nh vËy :

$$\frac{x_{n+1} - y}{x_n - y} = q < 1 \qquad (7)$$

hay : $$x_{n+1} - y = q(x_n - y) \qquad (8)$$

T¬ng tù ta cã : $$x_{n+2} - y = q(x_{n+1} - y) \qquad (9)$$

Tõ (8) vµ (9) ta cã :

$$q = \frac{x_{n+2} - x_{n+1}}{x_{n+1} - x_n} \qquad (10)$$

Thay gi¸ trÞ cña q võa tÝnh ë (10) vµo biÓu thøc cña q ë trªn ta cã :

$$y = x_n - \frac{(x_n - x_{n+1})^2}{x_n - 2x_{n+1} + x_{n+1}} \qquad (11)$$

C«ng thøc (11) ®îc gäi lµ c«ng thøc ngo¹i suy Adam.Nh vËy theo (11) tríc hÕt ta dïng ph¬ng ph¸p lÆp ®Ó tÝnh gi¸ trÞ gÇn ®óng x_{n+2},x_{n+1},x_n cña nghiÖm vµ sau ®ã theo (11) t×m ®îc nghiÖm víi sai sè nhá h¬n.

§Ó lµm vÝ dô chóng ta xÐt ph¬ng tr×nh :

$$\ln x - x^2 + 3 = 0$$

Ta ®a vÒ d¹ng lÆp :

$$x = \sqrt{\ln(x) + 3}$$

$$f'(x) = \frac{1}{2x\sqrt{\ln x + 3}}$$

PhÐp lÆp héi tô trong ®o¹n [0.3,∝].Ta cho x_1 = 1 th× tÝnh ®îc :

x_2 = 1,7320508076
x_3 = 1.883960229
x_4 = 1.90614167
y = 1.909934347

§Ó gi¶m sai sè ta cã thÓ lÆp nhiÒu lÇn

Chương trình 8-9

```
//phuong phap Aitken
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define  m  5

void main()
  {
```

```
        float x[m];
        float epsi,n,y;
        int i,z;
        float f(float);

        clrscr();
        printf("Cho tri so ban dau x[1] = ");
        scanf("%f",&x[1]);
        printf("Cho tri so sai so epsilon = ");
        scanf("%f",&epsi);
        printf("\n");
        printf( "Ngoai suy Aitken cua ham\n");
        z=0;
        while (z<=20)
            {
                for (i=2;i<=4;i++)
                  x[i]=f(x[i-1]);
                n=x[4]-2*x[3]+x[2];
                if ((fabs(n)<1e-09)||(fabs(x[1]-x[2])<epsi*fabs(x[1])))
                  z=20;
                else
                  {
                        y=x[2]-(x[3]-x[2])*(x[3]-x[2])/n;
                        if (z>20)
                          printf("Khong hoi tu sau hai muoi lan lap\n");
                        x[1]=y;
                  }
                z=z+1;
          }
        printf("Nghiem cua phuong trinh y = %.6f",y);
        getch();
  }

float f(float x)
  {
        float s=sqrt(log(x)+3);
        return(s);
  }
```

Víi gi¸ trÞ ban ®Çu lµ 1 vµ sai sè lµ 1e-8,ch¬ng tr×nh cho kÕt qu¶ y = 1.9096975944

§10.PHƯƠNG PHÁP BAIRSTOW

Nguyªn t¾c cña ph¬ng ph¸p Bairstow lµ trÝch tõ ®a thøc $P_n(x)$ mét tam thøc $Q_2(x) = x^2 - sx + p$ mµ ta cã thÓ tÝnh nghiÖm thùc hay nghiÖm phøc cña nã mét c¸ch ®¬n gi¶n b»ng c¸c ph¬ng ph¸p ®· biÕt.

ViÖc chia ®a thøc $P_n(x)$ cho tam thøc $Q_2(x)$ ®a tíi kÕt qu¶ :

$$P_n(x) = Q_2(x).P_{n-2}(x) + R_1(x)$$

víi $$P_n(x) = a_ox^n + a_1x^{n-1} + a_2x^{n-2} +...+ a_n$$

$Q_2(x) = x^2 - sx + p$
$P_{n-2}(x) = b_o x^{n-2} + b_1 x^{n-3} + b_2 x^{n-4} +...+ b_{n-2}$
$R_1(x) = \alpha x + \beta$

§Ó cã ®-îc mét th-¬ng ®óng,cÇn t×m c¸c gi¸ trÞ cña s vµ p sao cho $R_1(x) = 0$ (nghÜa lµ α vµ β triÖt tiªu).Víi s vµ p ®· cho,c¸c hÖ sè b cña ®a thøc $P_{n-2}(x)$ vµ c¸c hÖ sè α vµ β ®-îc tÝnh b»ng ph-¬ng ph¸p truy håi.C¸c c«ng thøc nhËn ®-îc khi khai triÓn biÓu thøc $P_n(x) = Q_2(x).P_{n-2}(x) + R_1(x)$ vµ s¾p xÕp l¹i c¸c sè h¹ng cïng bËc :

$$a_o x^n + a_1 x^{n-1} + a_2 x^{n-2} +...+ a_n = (x^2 - sx + p)(b_o x^{n-2} + b_1 x^{n-3} + b_2 x^{n-4} +...+ b_{n-2})$$

Sè h¹ng bËc	HÖ sè cña $P_n(x)$	HÖ sè cña $Q_2(x).P_{n-2}(x)$
x^n	a_o	b_o
x^{n-1}	a_1	$b_1 - sb_o$
x^{n-2}	a_2	$b_2 - sb_1 + pb_o$
......	......	.....
x^{n-k}	a_k	$b_k - sb_{k-1} + pb_{k-2}$
x	a_{n-1}	$\alpha - sb_{n-2} + pb_{n-3}$
x^o	a_n	$\beta + pb_{n-2}$

Nh- vËy : $b_o = a_o$ (1)
$b_1 = a_1 + sb_o$
$b_2 = a_2 + sb_1 - pb_o$
....
$b_k = a_k + sb_{k-1} - pb_{k-2}$
$\alpha = a_{n-1} + sb_{n-2} - pb_{n-3}$
$\beta = a_n - pb_{n-2}$

Chóng ta nhËn thÊy r»ng α ®-îc tÝnh to¸n xuÊt ph¸t tõ cïng mét c«ng thøc truy håi nh- c¸c hÖ sè b_k vµ t-¬ng øng víi hÖ sè b_{n-1}

$$b_{n-1} = a_{n-1} + sb_{n-2} - pb_{n-3} = \alpha$$

HÖ sè b_n lµ :

$$b_n = a_n + sb_{n-1} - pb_{n-2} = sb_{n-1} + \beta$$

vµ cuèi cïng :

$$R_1(x) = \alpha x + \beta = b_{n-1}(x - s) + b_n$$

Ngoµi ra c¸c hÖ sè b_i phô thuéc vµo s vµ p vµ b©y giê chóng ta cÇn ph¶i t×m c¸c gi¸ trÞ ®Æc biÖt s^* vµ p^* ®Ó cho b_{n-1} vµ b_n triÖt tiªu.Khi ®ã $r_1(x) = 0$ vµ nghiÖm cña tam thøc $x^2 - s^*x + p^*x$ sÏ lµ nghiÖm cña ®a thøc $P_n(x)$.Ta biÕt r»ng b_{n-1} vµ b_n lµ hµm cña s vµ p :

$$b_{n-1} = f(s,p)$$
$$b_n = g(s,p)$$

ViÖc t×m s^* vµ p^* ®-a ®Õn viÖc gi¶i hÖ ph-¬ng tr×nh phi tuyÕn:

$$\begin{cases} f(s,p)=0 \\ g(s,p)=0 \end{cases}$$

Ph-¬ng tr×nh nµy cã thÓ gi¶i dÔ dµng nhê ph-¬ng ph¸p Newton.ThËt vËy víi mét ph-¬ng tr×nh phi tuyÕn ta cã c«ng thøc lÆp :

$$x_{i+1} = x_i - f(x_i)/f'(x_i)$$

hay $f'(x_i)(x_{i+1} - x_i) = -f(x_i)$

Víi mét hÖ cã hai ph-¬ng tr×nh,c«ng thøc lÆp trë thµnh:

$$J(X_i)(X_{i+1} - X_i) = -F(X_i)$$

víi $X_i = \{ s_i,p_i\}^T$ vµ $X_{i+1} = \{ s_{i+1},p_{i+1}\}^T$

$$F(X_i)=\begin{vmatrix} f(s_i,p_i) \\ g(s_i,p_i) \end{vmatrix}$$

$$J(X_i)=\begin{vmatrix} \dfrac{\partial f}{\partial s} & \dfrac{\partial f}{\partial p} \\ \dfrac{\partial g}{\partial s} & \dfrac{\partial g}{\partial p} \end{vmatrix}$$

Quan hÖ : $J(X_i)\Delta X = -F(X_i)$ víi $\Delta X = \{s_{i+1} - s_i,p_{i+1} - p_i\}^T$ t-¬ng øng víi mét hÖ ph-¬ng tr×nh tuyÕn tÝnh hai Èn sè $\Delta s = s_{i+1} - s_i$ vµ $\Delta p = p_{i+1} - p_i$:

$$\begin{cases} \dfrac{\partial f}{\partial s}\Delta s + \dfrac{\partial f}{\partial p}\Delta p = -f(s_i,p_i) \\ \dfrac{\partial g}{\partial s}\Delta s + \dfrac{\partial g}{\partial p}\Delta p = -g(s_i,p_i) \end{cases}$$

Theo c«ng thøc Cramer ta cã :

$$\Delta s = \frac{-f\dfrac{\partial g}{\partial p} + g\dfrac{\partial f}{\partial p}}{\delta}$$

$$\Delta p = \frac{-g\dfrac{\partial f}{\partial s} + f\dfrac{\partial g}{\partial s}}{\delta}$$

$$\delta = \frac{\partial f}{\partial s}\frac{\partial g}{\partial p} - \frac{\partial f}{\partial p}\frac{\partial g}{\partial s}$$

§Ó dïng ®-îc c«ng thøc nµy ta cÇn tÝnh ®-îc c¸c ®¹o hµm $\dfrac{\partial f}{\partial s},\dfrac{\partial f}{\partial p},\dfrac{\partial g}{\partial s},\dfrac{\partial g}{\partial p}$.C¸c ®¹o hµm nµy ®-îc tÝnh theo c«ng thøc truy håi.

Do $b_o = a_o$ nªn

$$\frac{\partial b_o}{\partial s} = 0 \qquad \frac{\partial b_o}{\partial p} = 0$$

$b_1 = a_1 + sb_o$ nªn

$$\frac{\partial b_1}{\partial s} = b_o \qquad \frac{\partial b_1}{\partial p} = 0$$

$b_2 = a_2 + sb_1- pb_o$ nªn

$$\frac{\partial b_2}{\partial s} = \frac{\partial a_2}{\partial s} + \frac{\partial (sb_1)}{\partial s} - \frac{\partial (pb_o)}{\partial s}$$

MÆt kh¸c :

$$\frac{\partial a_2}{\partial s} = 0 \qquad \frac{\partial (sb_1)}{\partial s} = s\frac{\partial b_1}{\partial s} + b_1 \qquad \frac{\partial (pb_o)}{\partial s} = 0$$

nªn :

$$\frac{\partial b_2}{\partial s} = b_1 + sb_o$$

$b_3 = a_3 + sb_2- pb_1$ nªn

$$\frac{\partial b_3}{\partial s} = b_2 + s\frac{\partial b_2}{\partial s} - p\frac{\partial b_1}{\partial s}$$

NÕu chóng ta ®Æt :

$$\frac{\partial b_k}{\partial s} = c_{k-1}$$

th× : $c_o = b_o$ (2)

$c_1 = b_1 + sb_o = b_1 + sc_o$

$c_2 = b_2 + sc_1 - pc_o$

....

$c_k = b_k + sc_{k-1} - pc_{k-2}$

$c_{n-1} = b_{n-1} + sc_{n-2} - pc_{n-3}$

Nh- vËy c¸c hÖ sè còng ®-îc tÝnh theo c¸ch nh- c¸c hÖ sè b_k.Cuèi cïng víi $f = b_{n-1}$ vµ $g = b_n$ ta ®-îc:

$$\frac{\partial f}{\partial s} = c_{n-2} \quad \frac{\partial f}{\partial s} = c_{n-3} \quad \frac{\partial f}{\partial s} = c_{n-1} \quad \frac{\partial f}{\partial s} = c_{n-2}$$

$$\Delta s = \frac{b_{n-1}c_{n-2} - b_n c_{n-3}}{c_{n-1}c_{n-3} - c_{n-2}^2} \qquad (3)$$

$$\Delta p = \frac{b_{n-1}c_{n-1} - b_n c_{n-2}}{c_{n-1}c_{n-3} - c_{n-2}^2} \qquad (4)$$

Sau khi ph©n tÝch xong $P_n(x)$ ta tiÕp tôc ph©n tÝch $P_{n-2}(x)$ theo ph-¬ng ph¸p trªn C¸c b-íc tÝnh to¸n gåm :

- Chän c¸c gi¸ trÞ ban ®Çu bÊt k× s_0 vµ p_0
- TÝnh c¸c gi¸ trÞ b_o,..,b_n theo (1)
- TÝnh c¸c gi¸ trÞ c_o,...,c_n theo (2)
- TÝnh Δs_o vµ Δp_o theo (3) vµ (4)
- TÝnh $s_1 = s_0 + \Delta s_o$ vµ $p_1 = p_o + \Delta p_o$
- LÆp l¹i b-íc 1 cho ®Õn khi $p_{i+1} = p_i = p$ vµ $s_{i+1} = s_i = s$
- Gi¶i ph-¬ng tr×nh $x_2 - sx + p$ ®Ó t×m 2 nghiÖm cña ®a thøc
- B¾t ®Çu qu¸ tr×nh trªn cho ®a thøc $P_{n-2}(x)$

VÝ dô : T×m nghiÖm cña ®a thøc $P_4(x) = x^4 - 1.1x^3 + 2.3x^2 + 0.5x^2 + 3.3$.

Víi lÇn lÆp ban ®Çu ta chän s = -1 vµ p =1,nghÜa lµ tam thøc cã d¹ng $x^2 + x + 1$

	a_0	a_1	a_2	a_3	a_4
	1	-1.1	2.3	0.5	3.3
sb_i		-1	2.1	-3.4	0.8
$-pb_{i-1}$			-1	2.1	-3.4
b_i	1	-2.1	3.4	$-0.8 = b_{n-1}$	$0.7=b_n$
sb_i		-1.0	3.1	-5.5	
$-pb_{i-1}$			-1.0	3.1	
c_i	1	-3.1	5.5	-3.2	

$$\Delta s = \frac{\begin{vmatrix} 0.8 & -3.1 \\ -0.7 & 5.5 \end{vmatrix}}{\begin{vmatrix} 5.5 & -3.1 \\ -3.2 & 5.5 \end{vmatrix}} = 0.11$$

$$\Delta p = \frac{\begin{vmatrix} 5.5 & 0.8 \\ -3.2 & 0.7 \end{vmatrix}}{\begin{vmatrix} 5.5 & -3.1 \\ -3.2 & 5.5 \end{vmatrix}} = 0.06$$

$s^* = -1 + 0.11 = -0.89$

$p^* = 1 + 0.06 = 1.06$

TiÕp tôc lÆp lÇn 2 víi $s_1 = s^*$ vµ $p_1 = p^*$ ta cã :

	a_0	a_1	a_2	a_3	a_4
	1	-1.1	2.3	0.5	3.3
sb_i		-0.89	1.77	-2.68	0.06
$-pb_{i-1}$			-1.06	2.11	-3.17
b_i	1	-1.99	3.01	-0.07 = b_{n-1}	0.17=b_n
sb_i		-0.89	2.56	-4.01	
$-pb_{i-1}$			-1.0	3.1	
c_i	1	-2.88	4.51	-1.03	

$$\Delta s = \frac{\begin{vmatrix} 0.07 & -2.88 \\ -0.7 & 5.5 \end{vmatrix}}{\begin{vmatrix} 4.51 & -2.88 \\ -1.03 & 4.51 \end{vmatrix}} = -0.01$$

$$\Delta p = \frac{\begin{vmatrix} 4.51 & 0.07 \\ -1.03 & -0.17 \end{vmatrix}}{\begin{vmatrix} 4.51 & -2.88 \\ -1.03 & 4.51 \end{vmatrix}} = 0.04$$

$s^* = -0.89 - 0.01 = -0.9$

$p^* = 1.06 + 0.04 = 1.1$

Nh- vËy $P_4(x) = (x^2+0.9x+1.1)(x^2 + 2x+3)$

Ch-¬ng tr×nh sau ¸p dông lÝ thuyÕt võa nªu ®Ó t×m nghiÖm cña ®a thøc.

Chương trình 8-10

```
//phuong phap Bairstow
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define  m   10

void main()
  {
        float a[m],b[m],c[m];
        int i,n,v;
        float s,e1,t,p,q,r,p1,q1;

        clrscr();
        printf("Cho bac cua da thuc n = ");
        scanf("%d",&n);
        printf("Cho cac he so cua da thuc can tim nghiem\n");
        for (i=n;i>=0;i--)
          {
```

```
          printf("a[%d] = ",n-i);
          scanf("%f",&a[i]);
  }
printf("\n");
e1=0.0001;
if (n<=2)
    if (n==1)
          {
            printf("Nghiem cua he\n");
            printf("%.8f",(a[0]/(-a[1])));
            getch();
            exit(1);
          }
do
  {
          v=0;
          p=1;
          q=-1;
          b[n]=a[n];
          c[n]=a[n];
          do
            {
                    b[n-1]=b[n]*p+a[n-1];
                    c[n-1]=b[n-1]+b[n]*p;
                    for (i=n-2;i>=0;i--)
                      {
                              b[i]=b[i+2]*q+b[i+1]*p+a[i];
                              c[i]=c[i+2]*q+c[i+1]*p+b[i];
                        }
                    r=c[2]*c[2]-c[1]*c[3];
                    p1=p-(b[1]*c[2]-b[0]*c[3])/r;
                    q1=q-(b[0]*c[2]-b[1]*c[1])/r;
                    if ((fabs(b[0])<e1)&&(fabs(b[1])<e1))
                      goto tt;
                    v=v+1;
                    p=p1;
                    q=q1;
            }
          while (v<=40);
          if(v>40)
            {
                    printf("Khong hoi tu sau 40 lan lap");
                    getch();
                    exit(1);
            }
          tt:s=p1/2;
          t=p1*p1+4*q1;
          if(t<0)
            {
                    printf("Nghiem phuc\n");
```

```
                    printf("%.8f+%.8fj\n",s,(sqrt(-t)/2));
                    printf("%.8f-%.8fj\n",s,(sqrt(-t)/2));
                    printf("\n");
                 }
               else
                 {
                    printf("Nghiem thuc\n");
                    printf("%.8f\n",(s+sqrt(t)/2));
                    printf("%.8f\n",(s-sqrt(t)/2));
                    printf("\n");
                 }
               for (i=2;i<=n;i++)
                 a[i-2]=b[i];
                 n=n-2;
          }
        while ((n>2)&(r!=0.0));
        s=-a[1]/(2*a[2]);
        t=a[1]*a[1]-4*a[2]*a[0];
        if (t<0)
          {
               printf("Nghiem phuc\n");
               printf("%.8f+%.8fj\n",s,(sqrt(-t)/(2*a[2])));
               printf("%.8f-%.8fj\n",s,(sqrt(-t)/(2*a[2])));
               printf("\n");
          }
        else
          {
               printf("Nghiem thuc\n");
               printf("%.8f\n",(s-sqrt(t)/(2*a[2])));
               printf("%.8f\n",(s-sqrt(t)/(2*a[2])));
               printf("\n");
          }
        getch();
  }
```

Dïng ch-¬ng tr×nh trªn ®Ó x¸c ®Þnh nghiÖm cña ®a thøc :

$$x^6 - 2x^5 - 4x^4 + 13x^3 - 24x^2 + 18x - 4 = 0$$

ta nhËn ®-îc c¸c nghiÖm :

$x_1 = 2.61903399$

$x_2 = -2.73205081$

$x_3 = 0.732050755$

$x_4 = 0.381966055$

$x_5 = 0.500011056 + i*1.3228881$

$x_6 = 0.500011056 - i*1.3228881$

§11.HỆ PHƯƠNG TRÌNH PHI TUYẾN

Ph-¬ng ph¸p Newton cã thÓ ®-îc tæng qu¸t ho¸ ®Ó gi¶i hÖ ph-¬ng tr×nh phi tuyÕn d¹ng :

$$\begin{cases} f_1(x_1,x_2,x_3,...,x_2)=0 \\ f_2(x_1,x_2,x_3,...,x_2)=0 \\ f_3(x_1,x_2,x_3,...,x_2)=0 \\ .... \\ f_n(x_1,x_2,x_3,...,x_2)=0 \end{cases}$$

hay viÕt gän h¬n díi d¹ng :

$$F(X) = 0$$

Trong ®ã : $X = (x_1,x_2,x_3,.....,x_n)$

Víi mét ph¬ng tr×nh mét biÕn,c«ng thøc Newton lµ :

$$x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}$$

hay : $f'(x_i).\Delta x = -f(x_i)$

víi $\Delta x = x_{i+1} - x_i$

§èi víi hÖ,c«ng thøc lÆp lµ :

$$J(X_i)\Delta x = -F(X_i)$$

Trong ®ã $J(X_i)$ lµ to¸n tö Jacobi.Nã lµ mét ma trËn bËc n (n - t¬ng øng víi sè thµnh phÇn trong vect¬ X) cã d¹ng :

$$J(x_i)=\begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3}............ & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3}............ & \frac{\partial f_2}{\partial x_n} \\ . & & & \\ . & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \frac{\partial f_n}{\partial x_3}............ & \frac{\partial f_n}{\partial x_n} \end{vmatrix}$$

vµ $\Delta X = X_{i+1} - X_i$

Ph¬ng ph¸p Newton tuyÕn tÝnh ho¸ hÖ vµ nh vËy víi mçi bíc lÆp cÇn gi¶i mét hÖ ph¬ng tr×nh tuyÕn tÝnh (mµ biÕn lµ Δx_i) x¸c ®Þnh bëi c«ng thøc lÆp cho tíi khi vect¬ $X(x_1,x_2,x_3,.....,x_n)$ gÇn víi nghiÖm.

Díi ®©y lµ ch¬ng tr×nh gi¶i hÖ ph¬ng tr×nh phi tuyÕn

$$\begin{cases} x_1^3-x_2^3-3x_1x_2x_4-8=0 \\ x_1+x_2+x_3+x_4-5=0 \\ \sqrt{25-x_1^2}+8x_3+4=0 \\ 2x_1x_2x_3-x_4+8=0 \end{cases}$$

Ma trËn ®¹o hµm riªng $J(x_i)$lµ :

$$\begin{pmatrix} 3x_1^2-3x_2x_4 & -3x_2^2-3x_1x_4 & 0 & -3x_1x_2 \\ 1 & 1 & 1 & 1 \\ \frac{-x_1}{\sqrt{25-x_1^2}} & 0 & 8 & 0 \\ 2x_2x_3 & 2x_2x_3 & 2x_2x_3 & -1 \end{pmatrix}$$

Ma trËn nµy ®îc ch¬ng tr×nh ®äc vµo nhê thñ tôc doc.Trong thñ tôc nµy,c¸c hÖ sè a[i,5] lµ c¸c hµm $f_i(x)$.Vect¬ nghiÖm ban ®Çu ®îc chän lµ $\{ 0,-1,-1,1\}^T$.KÕt qu¶ tÝnh cho ta : x = $\{0.01328676,-1.94647929,-1.12499779,8.05819031 \}^T$ víi ®é chÝnh x¸c 0.000001.Vect¬ sè d r = $\{ 0.00000536,-0.00000011,-0.00000001,-0.00000006\}^T$.

Chương trình 8-11

```
//giai he pt phi tuyen
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define n  4

float a[n+1][n+2];
float x[n+1],y[n+1];
int i,j,k,l,z,r;
float e,s,t;

void main()
  {
        void doc();
    clrscr();
        printf("Cho cac gia tri nghiem ban dau\n");
        for (i=1;i<=n;i++)
          {
                printf("x[%d] = ",i);
                scanf("%f",&x[i]);
          }
        e=1e-6;
        z=30;
        for (r=1;r<=z;r++)
          {
                doc();
                for (k=1;k<=n-1;k++)
                  {
                        s=0 ;
                        for (i=k;i<=n;i++)
                          {
                                t=fabs(a[i][k]);
                                if (s<=t)
                                  {
                                        s=t;
                                        l=i;
                                  }
                          }
                        for (j=k;j<=n+1;j++)
                          {
                                s=a[k][j];
                                a[k][j]=a[l][j];
                                a[l][j]=s;
                          }
                        if (a[1][1]==0)
                          {
```

```
                printf("Cac phan tu duong cheo cua ma tran bang khong");
                getch();
                exit(1);
              }
            else
              {
                if (fabs(a[k][k]/a[1][1])<(1e-08))
                  {
                    printf("Ma tran suy bien");
                    goto mot;
                  }
              }
            for (i=k+1;i<=n;i++)
              {
                if (a[k][k]==0)
                  {
                    printf("Cac phan tu duong cheo cua ma tran bang
khong\n");
                    goto mot;
                  }
                s=a[i][k]/a[k][k];
                a[i][k]=0;
                for (j=k+1;j<=n+1;j++)
                 a[i][j]=a[i][j]-s*a[k][j];
              }
            y[n]=a[n][n+1]/a[n][n];
            for (i=n-1;i>=1;i--)
              {
                s=a[i][n+1];
                for (j=i+1;j<=n;j++)
                  s=s-a[i][j]*y[j];
                if (a[i][i]==0)
                  {
                    printf("Cac phan tu duong cheo cua ma tran bang
khong\n");
                    goto mot;
                  }
                y[i]=s/a[i][i];
              }
          }
        if (r!=1)
          for (i=1;i<=n;i++)
            {
             if (fabs(y[i])<e*fabs(x[i]))
                goto ba;
            }
        for (i=1;i<=n;i++)
          x[i]=x[i]-y[i];
        printf("\n");
    }
```

```
        printf("Khong hoi tu sau %d lan lap\n",z);
        goto mot;
        clrscr();
        ba:printf("Vec to nghiem\n");
        for (i=1;i<=n;i++)
          printf("%.5f\n",(x[i]-y[i]));
        printf("\n");
        printf("Do chinh xac cua nghiem la %.5f: \n", e);
        printf("\n");
        printf("Vec to tri so du :\n");
        for (i=1;i<=n;i++)
          printf("%.5f\n",(a[i][n+1]));
        mot:printf("\n");
        getch();
  }

void doc()
  {
        a[1][1]=3*x[1]*x[1]-3*x[2]*x[4];
        a[1][2]=-3*x[2]*x[2]-3*x[1]*x[4];
        a[1][3]=0;
        a[1][4]=-3*x[1]*x[2];
        a[1][5]=x[1]*x[1]*x[1]-x[2]*x[2]*x[2]-3*x[1]*x[2]*x[4]-8;

        a[2][1]=1;
        a[2][2]=1;
        a[2][3]=1;
        a[2][4]=1;
        a[2][5]=x[1]+x[2]+x[3]+x[4]-5;

        a[3][1]=-x[1]/sqrt(25-x[1]*x[1]);
        a[3][2]=0;
        a[3][3]=8;
        a[3][4]=0;
        a[3][5]=sqrt(25-x[1]*x[1])+8*x[3]+4;

        a[4][1]=2*x[2]*x[3];
        a[4][2]=2*x[1]*x[3];
        a[4][3]=2*x[1]*x[2];
        a[4][4]=-1;
        a[4][5]=2*x[1]*x[2]*x[3]-x[4]+8;

  }
```