

CHƯƠNG 10 : GIẢI HỆ PHƯƠNG TRÌNH ĐẠI SỐ TUYẾN TÍNH

§1. PHƯƠNG PHÁP GAUSS

Có nhiều phương pháp để giải một hệ phương trình tuyến tính dạng $AX = B$. Phương pháp giải sẽ đơn giản hơn nếu ma trận A có dạng tam giác nghĩa là có dạng :

$$\begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \text{hay} \quad \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}$$

Trong trường hợp đầu tiên, ma trận được gọi là ma trận tam giác dưới và trường hợp thứ hai ma trận được gọi là ma trận tam giác trên. Phương trình tương ứng với ma trận tam giác dưới có dạng tường minh là :

$$\begin{cases} a_{11}x_1 + 0x_2 + 0x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + 0x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

Với phương trình dạng này chúng ta sẽ giải phương trình từ trên xuống. Chương trình giải phương trình ma trận tam giác dưới là :

Chương trình 10-1

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 10

void main()
{
    float a[max][max];
    float b[max], x[max];
    int i, j, k, n, t;
    float s, c;
    char tl;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
}
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("%15.5f",a[i][j]);
    printf("\n");
}
printf("\n");
printf("Cho cac phan tu cua ma tran b\n");
for (i=1;i<=n;i++)
{
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
printf("\n");
printf("Ma tran b ma ban da nhap");
printf("\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %10.5f\n",i,b[i]);
printf("\n");
flushall();
t=1;
while (t)
{

```

```

printf("Co sua ma tran b khong(c/k)?");
scanf("%c",&tl);
if (toupper(tl)=='C')
{
    printf("Cho chi so hang can sua : ");
    scanf("%d",&i);
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
if (toupper(tl)=='K')
    t=0;
}
printf("\n");
printf("Ma tran b ban dau");
printf("\n");
for (i=1;i<=n;i++)
    printf("%15.5f\n",b[i]);
{
    if (a[1][1]==0)
        if (b[1]!=0)
            printf("He da cho vo nghiem\n");
        else
        {
            printf("He da cho co vo so nghiem");
            x[n]=c;
        }
    else
        x[1]=b[1]/a[1][1];
    for (i=2;i<=n;i++)
    {
        s=0;
        for (k=1;k<=i-1;k++)
            s=s+a[i][k]*x[k];
        x[i]=(b[i]-s)/a[i][i];
    }
    printf("\n");
    printf("Nghiem cua he da cho la");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("x[%d] = %10.5f\n",i,x[i]);
    getch();
}
}

```

Phương trình tương ứng với ma trận tam giác trên có dạng tường minh là :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ 0x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ 0x_1 + 0x_2 + a_{33}x_3 = b_3 \end{cases}$$

Với phương trình này chúng ta giải từ dưới lên.

Chương trình giải phương trình ma trận tam giác trên là :

Chương trình 10-2

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 10

void main()
{
    float a[max][max];
    float b[max],x[max];
    int i,j,k,n,t;
    float s,c;
    char tl;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
    }
    printf("\n");
    t=1;
    fflush(stdin);
    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");
printf("\n");
for (i=1;i<=n;i++)
{
    for (j=1;j<=n;j++)
        printf("%15.5f",a[i][j]);
    printf("\n");
}
printf("\n");
printf("Cho cac phan tu cua ma tran b : \n");
for (i=1;i<=n;i++)
{
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
printf("\n");
printf("Ma tran b ma ban da nhap");
printf("\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %10.5f\n",i,b[i]);
printf("\n");
flushall();
t=1;
while (t)
{
    printf("Co sua ma tran b khong(c/k)?");
    scanf("%c",&tl);
    if (toupper(tl)=='C')
    {
        printf("Cho chi so hang can sua : ");
        scanf("%d",&i);
        printf("b[%d] = ",i);
        scanf("%f",&b[i]);
    }
    if (toupper(tl)=='K')
        t=0;
}
printf("\n");
printf("Ma tran b ban dau\n");
printf("\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %10.5f\n",i,b[i]);

```

```

printf("\n");
{
    if (a[n][n]==0)
        if (b[n]!=0)
            printf("He da cho vo nghiem");
        else
            {
                printf("He da cho co vo so nghiem");
                x[n]=c;
            }
    else
        x[n]=b[n]/a[n][n];
        for (i=n-1;i>=1;i--)
            {
                s=0;
                for (k=i+1;k<=n;k++)
                    s=s+a[i][k]*x[k];
                x[i]=(b[i]-s)/a[i][i];
            }
        printf("\n");
        printf("Nghiem cua he da cho la\n");
        printf("\n");
        for (i=1;i<=n;i++)
            printf("x[%d] = %10.5f\n",i,x[i]);
        getch();
    }
}

```

Tuy nhiên, các hệ phương trình đơn giản hiếm khi gặp trong thực tế. Các hệ phương trình tuyến tính có thể biểu diễn dưới dạng tam giác nếu định thức của nó khác không, nghĩa là phương trình có nghiệm. Chúng ta biết rằng các nghiệm của hệ không đổi nếu ta thay một hàng bằng tổ hợp tuyến tính của các hàng khác. Như vậy bằng một loạt các biến đổi ta có thể đưa hệ ban đầu về dạng tam giác. Đó chính là nội dung của phương pháp loại trừ Gauss. Chúng ta hãy xét hệ phương trình :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

Nhân hàng thứ nhất với a_{21}/a_{11} ta có :

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \frac{a_{21}}{a_{11}}a_{13}x_3 = \frac{a_{21}}{a_{11}}b_1$$

Số hạng đầu của phương trình bằng số hạng đầu của hàng thứ hai trong hệ phương trình ban đầu. Khi trừ hàng một đã được biến đổi cho hàng 2 ta nhận được hàng 2 mới

$$0x_1 + (a_{22} - \frac{a_{21}}{a_{11}}a_{12})x_2 + (a_{23} - \frac{a_{21}}{a_{11}}a_{13})x_3 = b_2 - \frac{a_{21}}{a_{11}}b_1$$

Ta tiếp tục cách này để loại trừ x_1 ra khỏi hàng thứ 3. Phương trình trở thành :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b'_2 \\ b'_3 \end{pmatrix}$$

$$\text{với } a'_{11} = a_{11}; a'_{12} = a_{12}; a'_{13} = a_{13}; a'_{13} = a_{13}; b'_1 = b_1$$

$$a'_{22} = a_{22} - \frac{a_{21}}{a_{11}} a_{12} \quad a'_{23} = a_{23} - \frac{a_{21}}{a_{11}} a_{13} \quad a'_{32} = a_{32} - \frac{a_{31}}{a_{11}} a_{12} \quad a'_{33} = a_{33} - \frac{a_{31}}{a_{11}} a_{13} \quad b'_2 = b_2 - \frac{a_{21}}{a_{11}} b_1$$

$$b'_3 = b_3 - \frac{a_{31}}{a_{11}} b_1$$

Ta loại trừ số hạng chứa x_3 trong dòng thứ 3 bằng cách tương tự. Ta nhân hàng thứ 2 trong hệ $A'X = B'$ với a'_{32}/a'_{22} và đem trừ đi hàng thứ 3 trong hệ mới. Như vậy số hạng chứa x_3 biến mất và ta nhận được ma trận tam giác trên.

$$\begin{pmatrix} a''_{11} & a''_{12} & a''_{13} \\ 0 & a''_{22} & a''_{23} \\ 0 & 0 & a''_{33} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b''_1 \\ b''_2 \\ b''_3 \end{pmatrix}$$

$$\text{với } a''_{11} = a'_{11} \quad a''_{12} = a'_{12} \quad a''_{13} = a'_{13} \quad b''_1 = b'_1 \quad a''_{22} = a'_{22} \quad a''_{23} = a'_{23} \quad b''_2 = b'_2$$

$$a''_{33} = a'_{33} - \frac{a'_{32}}{a'_{22}} a'_{23} \quad b''_3 = b'_3 - \frac{a'_{32}}{a'_{22}} b'_2$$

Các phép tính này chỉ thực hiện được khi $a_{11} \neq 0$ và $a'_{11} \neq 0$.

Với một hệ có n phương trình, thuật tính hoàn toàn tương tự. Sau đây là chương trình giải hệ phương trình n ẩn số bằng phương pháp loại trừ Gauss.

Chương trình 10-3

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 10

void main()
{
    float b[max], x[max];
    float a[max][max];
    int i, j, k, n, t;
    float c, s, d;
    char tl;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
}
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("%15.5f",a[i][j]);
    printf("\n");
}
printf("\n");
printf("Cho cac phan tu cua ma tran b : \n");
for (i=1;i<=n;i++)
{
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
printf("\n");
printf("Ma tran b ma ban da nhap\n");
printf("\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %15.5f\n",i,b[i]);
printf("\n");
flushall();
t=1;
while (t)
{

```



```

printf("Co sua ma tran b khong(c/k)?");
scanf("%c",&tl);
if (toupper(tl)=='C')
{
    printf("Cho chi so hang can sua : ");
    scanf("%f",&i);
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
if (toupper(tl)=='K')
    t=0;
}
printf("\n");
printf("Ma tran b\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %15.5f\n",i,b[i]);
printf("\n");
for (k=1;k<=n-1;k++)
{
    for (i=k+1;i<=n;i++)
    {
        b[i]=b[i]-b[k]*a[i][k]/a[k][k];
        for (j=k+1;j<=n;j++)
            a[i][j]=a[i][j]-a[k][j]*a[i][k]/a[k][k];
    }
}

{
    if (a[n][n]==0)
        if (b[n]==0)
            printf("He da cho vo nghiem");
else
    {
        printf("He da cho co vo so nghiem");
        x[n]=c;
    }
else
    x[n]=b[n]/a[n][n];
    for (i=n-1;i>=1;i--)
    {
        s=0;
        for (k=i+1;k<=n;k++)
            s=s+a[i][k]*x[k];
        x[i]=(b[i]-s)/a[i][i];
    }
    printf("\n");
    printf("Nghiem cua he da cho la\n");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("x[%d] = %15.5f\n",i,x[i]);
}

```

```

    getch();
}
}

```

§2. PHƯƠNG PHÁP GAUSS-JORDAN

Xét hệ phương trình $AX=B$. Khi giải hệ bằng phương pháp Gauss ta đưa nó về dạng ma trận tam giác sau một loạt biến đổi. Phương pháp khử Gauss-Jordan cải tiến khử Gauss bằng cách đưa hệ về dạng :

$$EX = B^*$$

và khi đó nghiệm của hệ chính là B^* . Trong phương pháp Gauss-Jordan mỗi bước tính phải tính nhiều hơn phương pháp Gauss nhưng lại không phải tính nghiệm. Để đưa ma trận A về dạng ma trận E tại bước thứ i ta phải có $a_{ii} = 1$ và $a_{ij} = 0$. Như vậy tại lần khử thứ i ta biến đổi :

$$1. a_{ij} = a_{ij}/a_{ii} \quad (j=i+1, i+2, \dots, n)$$

$$2. k=1, 2, \dots, n$$

$$a_{kj} = a_{kj} - a_{ij}a_{ki} \quad (j=i+1, i+2, \dots, n)$$

$$b_k = b_k - b_i a_{ki}$$

Ví dụ : Cho hệ

$$\begin{pmatrix} 8 & 4 & 2 & 0 \\ 4 & 10 & 5 & 4 \\ 2 & 5 & 6.5 & 4 \\ 0 & 4 & 4 & 9 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 24 \\ 32 \\ 26 \\ 21 \end{pmatrix}$$

Biến đổi lần 1 : ta chia hàng 1 cho $a_{11} = 8$; nhân hàng 1 vừa nhận được với 4 và lấy hàng 2 trừ đi; nhân hàng 1 vừa nhận được với 2 và lấy hàng 3 trừ đi; giữ nguyên hàng 4 vì phân tử đầu tiên đã bằng 0 ta có

$$\begin{pmatrix} 1 & 0.5 & 0.25 & 0 \\ 0 & 8 & 4 & 4 \\ 0 & 4 & 6 & 4 \\ 0 & 4 & 4 & 9 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ 20 \\ 20 \\ 21 \end{pmatrix}$$

Biến đổi lần 2 : ta chia hàng 2 cho $a_{22} = 8$; nhân hàng 2 vừa nhận được với 0.5 và lấy hàng 1 trừ đi; nhân hàng 2 vừa nhận được với 4 và lấy hàng 3 trừ đi; nhân hàng 2 vừa nhận được với 4 và lấy hàng 4 trừ đi ta có :

$$\begin{pmatrix} 1 & 0 & 0 & -0.25 \\ 0 & 1 & 0.5 & 0.5 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 2 & 7 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1.75 \\ 2.5 \\ 10 \\ 11 \end{pmatrix}$$

Biến đổi lần 3 : ta chia hàng 3 cho $a_{33} = 4$; giữ nguyên hàng 1; nhân hàng 3 vừa nhận được với 0.5 và lấy hàng 2 trừ đi; nhân hàng 3 vừa nhận được với 2 và lấy hàng 4 trừ đi ta có :

$$\begin{pmatrix} 1 & 0 & 0 & -0.25 \\ 0 & 1 & 0 & 0.25 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 6 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1.75 \\ 1.25 \\ 2.5 \\ 6 \end{pmatrix}$$

Biến đổi lần 4 : ta chia hàng 4 cho $a_{44} = 6$; nhân hàng 4 vừa nhận được với -0.25 và lấy hàng 1 trừ đi; nhân hàng 4 vừa nhận được với 0.25 và lấy hàng 2 trừ đi; nhân hàng 4 vừa nhận được với 0.5 và lấy hàng 3 trừ đi ta có :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 1 \end{pmatrix}$$

và ta có ngay vec tơ nghiệm.

Chương trình 10-4

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define spt 10

void main()
{
    float a[spt][2*spt];
    float b[spt];
    int i,j,k,n,m,t;
    float max,c;
    char tl;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
    }
    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\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");
printf("Cho cac phan tu cua ma tran b : \n");
for (i=1;i<=n;i++)
{
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
printf("\n");
printf("Ma tran b ma ban da nhap\n");
printf("\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %15.5f\n",i,b[i]);
printf("\n");
t=1;
flushall();
while (t)
{
    printf("Co sua ma tran b khong(c/k)?");
    scanf("%c",&tl);
    if (toupper(tl)=='C')
    {
        printf("Cho chi so hang can sua : ");
        scanf("%d",&i);
        printf("b[%d] = ",i);
        scanf("%f",&b[i]);
    }
}

```

```

        if (toupper(tl)=='K')
            t=0;
    }
    printf("\n");
    printf("Ma tran b\n");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("%15.5f\n",b[i]);
    printf("\n");
    t=1;
    fflush();
    i=1;
    while (t)
    {
        if (a[i][i]==0)
        {
            max=0;
            m=i;
            for (k=i+1;k<=n;k++)
                if (max<fabs(a[k][i]))
                {
                    m=k;
                    max=fabs(a[i][i]);
                }
            if (m!=i)
            {
                for (j=i;j<=n;j++)
                {
                    c=a[i][j];
                    a[i][j]=a[m][j];
                    a[m][j]=c;
                }
                c=b[i];
                b[i]=b[m];
                b[m]=c;
            }
            if (m==i)
            {
                t=0;
                printf("MA TRAN SUY BIEN");
            }
        }
        if (a[i][i]!=0)
        {
            c=1/a[i][i];
            for (j=i;j<=n;j++)
                a[i][j]=a[i][j]*c;
            b[i]=b[i]*c;
            for (k=1;k<=n;k++)
                if (k!=i)

```

```

        {
            c=a[k][i];
            for (j=i;j<=n;j++)
                a[k][j]=a[k][j]-a[i][j]*c;
            b[k]=b[k]-b[i]*c;
        }
        i=i+1;
        if (i==(n+1))
            t=0;
    }
    if (i==(n+1))
    {
        printf("NGHIEM CUA HE");
        printf("\n");
        for (i=1;i<=n;i++)
            printf("x[%d] = %15.5f\n",i,b[i]);
    }
    getch();
}

```

§3. PHƯƠNG PHÁP CHOLESKY

Trong phương pháp Cholesky một ma trận đối xứng A được phân tích thành dạng $A = R^T R$ trong đó R là một ma trận tam giác trên. Hệ phương trình lúc đó chuyển thành $AX = R^T R X = B$. Như vậy trước hết ta phân tích ma trận A thành tích hai ma trận. Sau đó giải hệ phương trình $R^T Y = B$ và cuối cùng là hệ $R X = Y$. Chương trình mô tả thuật toán này được cho dưới đây :

Chương trình 10-5

```

#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 6

void main()
{
    float a[max][max], r[max][max];
    float b[max], x[max], y[max];
    int i, j, k, l, n, t;
    float s;
    char tl;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
    }
printf("\n");
flushall();
t=1;
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[",i,"","j,"] = ");
                scanf("%f",&a[i][j]);
            }
        if (toupper(tl)=='K')
            t=0;
    }
printf("Ma tran a\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");
printf("Cho cac phan tu cua ma tran b : \n");
for (i=1;i<=n;i++)
    {
        printf("b[%d] = ",i);
        scanf("%f",&b[i]);
    }

```

```

    }
    printf("\n");
    printf("Ma tran b ma ban da nhap\n");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("b[%d] = %15.5f\n",i,b[i]);
    printf("\n");
    fflush();
    t=1;
    while (t)
    {
        printf("Co sua ma tran b khong(c/k)?");
        scanf("%c",&tl);
        if (toupper(tl)=='C')
        {
            printf("Cho chi so hang can sua : ");
            scanf("%d",&i);
            printf("b[%d] = ",i);
            scanf("%f",&b[i]);
        }
        if (toupper(tl)=='K')
            t=0;
    }
    printf("\n");
    printf("Ma tran b\n");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("b[%d] = %15.5f\n",i,b[i]);
    printf("\n");
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            r[i][j]=0.0;
    for (i=1;i<=n;i++)
    {
        if (a[i][i]>=0)
        {
            r[i][i]=sqrt(a[i][i]);
            for (j=1+i;j<=n;j++)
                r[i][j]=a[i][j]/r[i][i];
            for (k=i+1;k<=n;k++)
                for (l=k;l<=n;l++)
                    a[k][l]=a[k][l]-r[i][k]*r[i][l];
        }
    }
    for (k=1;k<=n;k++)
    {
        s=b[k];
        if (k!=1)
            for (i=1;i<=k-1;i++)
                s=s+r[i][k]*y[i];
    }

```



```

        y[k]=-s/r[k][k];
    }
    for (i=n;i>=1;i--)
    {
        s=-y[i];
        if (i!=n)
            for (k=i+1;k<=n;k++)
                s=s-r[i][k]*x[k];
        x[i]=s/r[i][i];
    }
    printf("Nghiem cua he phuong trinh la\n ");
    for (i=1;i<=n;i++)
        printf("x[%d] = %10.5f\n",i,x[i]);
    getch();
}

```

§4.PHƯƠNG PHÁP CROUT

Phương pháp Crout là một dạng của phương pháp Gauss. Với phương pháp Gauss, chúng ta biến đổi ma trận A thành một ma trận tam giác thì ở phương pháp Crout chúng ta phân tích ma trận này thành tích của ma trận tam giác trên R và ma trận tam giác dưới L. Trong ma trận L, các hệ số trên đường chéo chính bằng 1. Như vậy phương trình $AX = B$ được viết thành :

$$A.X = L.R.X = B$$

Chúng ta đặt $RX = Y$

nên : $LY = B$

Như vậy trước hết chúng ta phân tích ma trận thành tích của L.R. Tiếp theo ta giải phương trình $LY = B$ và sau đó giải phương trình $RX = A$ để tìm nghiệm X.

Chương trình 10-6

```

#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 6

void main()
{
    float b[max],x[max],y[max];
    float a[max][max],r[max][max],l[max][max];
    int i,j,k,n,t;
    float c,tr,tl,s;
    char tloi;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
    }
printf("\n");
t=1;
flushall();
while (t)
    {
        printf("Co sua ma tran a khong(c/k)?");
        scanf("%c",&tloi);
        if (toupper(tloi)=='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(tloi)=='K')
            t=0;
    }
printf("Ma tran a\n");
printf("\n");
for (i=1;i<=n;i++)
    {
        for (j=1;j<=n;j++)
            printf("%10.5f",a[i][j]);
        printf("\n");
    }
printf("\n");
printf("Cho cac phan tu cua ma tran b : \n");
for (i=1;i<=n;i++)
    {
        printf("b[%d] = ",i);
    }

```

```

        scanf("%f",&b[i]);
    }
    printf("\n");
    printf("Ma tran b ma ban da nhap");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("b[%d] = %10.5f\n",i,b[i]);
    printf("\n");
    t=1;
    fflush();
    while (t)
    {
        printf("Co sua ma tran b khong(c/k)?");
        scanf("%c",&tloi);
        if (toupper(tloi)=='C')
        {
            printf("Cho chi so hang can sua : ");
            scanf("%d",&i);
            printf("b[%d] = ",i);
            scanf("%f",&b[i]);
            fflush();
        }
        if (toupper(tloi)=='K')
            t=0;
    }
    printf("\n");
    printf("Ma tran b\n");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("b[%d] = %10.5f\n",i,b[i]);
    printf("\n");
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            r[i][j]=0.0;
            l[i][j]=0.0;
        }
    for (i=1;i<=n;i++)
    {
        r[1][i]=a[1][i];
        l[i][i]=1.0;
        l[i][1]=a[i][1]/a[1][1];
    }
    for (k=2;k<=n;k++)
    {
        for (j=k;j<=n;j++)
        {
            tr=0.0;
            for (i=1;i<=k;i++)
                tr=tr+l[k][i]*r[i][j];

```

```

        r[k][j]=a[k][j]-tr;
    }
    if (k!=n)
    {
        for (i=1;i<=n;i++)
        {
            tl=0.0;
            for (j=1;j<=k-1;j++)
                tl=tl+l[i][j]*r[j][k];
            l[i][k]=(a[i][k]-tl)/r[k][k];
        }
    }
    else
        printf("\n");
}
if (l[1][1]==0.0)
    if (b[1]==0.0)
        printf("He da cho vo nghiem\n");
    else
    {
        printf("He da cho co vo so nghiem\n");
        y[n]=c;
    }
else
    y[1]=b[1]/l[1][1];
for (i=2;i<=n;i++)
{
    s=0.0;
    for (k=1;k<=i-1;k++)
        s=s+l[i][k]*y[k];
    y[i]=(b[i]-s)/l[i][i];
}
if (r[n][n]==0.0)
    if (y[n]==0.0)
        printf("He da cho vo nghiem\n");
    else
    {
        printf("He da cho co vo so nghiem\n");
        x[n]=c;
    }
else
    x[n]=y[n]/r[n][n];
for (i=n-1;i>=1;i--)
{
    s=0.0;
    for (k=i+1;k<=n;k++)
        s+=r[i][k]*x[k];
    x[i]=(y[i]-s)/r[i][i];
}
printf("\n");

```

```

printf("Nghiem cua he da cho la\n");
printf("\n");
for (i=1;i<=n;i++)
    printf("x[%d] = %15.5f\n",i,x[i]);
getch();
}

```

§5. PHƯƠNG PHÁP LẬP ĐƠN

Xét hệ phương trình $AX = F$. Bằng cách nào đó ta đưa hệ phương trình về dạng

$$X = BX + G$$

trong đó

$$B = (b_{ij})_{n,n}$$

$$G = (g_1, g_2, \dots, g_n)^T$$

Chọn vectơ

$$X = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T$$

làm xấp xỉ thứ 0 của nghiệm đúng và xây dựng xấp xỉ

$$X^{(m+1)} = BX^{(m)} + G \quad (m = 0, 1, \dots)$$

Người ta chứng minh rằng nếu phương trình ban đầu có nghiệm duy nhất và một trong ba chuẩn của ma trận B nhỏ hơn 1 thì dãy xấp xỉ hội tụ về nghiệm duy nhất đó. (Cho một ma trận B , chuẩn của ma trận B , kí hiệu là $\|B\|$ là một trong 3 số :

$$\|B\|_1 = \max_i \sum_{j=1}^n b_{ij}$$

$$\|B\|_2 = \max_{j=1}^n \sum_{i=1}^n b_{ij}$$

$$\|B\|_3 = \left(\sum_{i=1}^n \sum_{j=1}^n b_{ij}^2 \right)^{1/2}$$

(Chuẩn của ma trận quan hệ tới sự hội tụ của phương pháp lặp)

Ví dụ chúng ta có phương trình

$$\begin{cases} 10x_1 + 2x_2 + x_3 = 10 \\ x_1 + 10x_2 + 2x_3 = 12 \\ x_1 + x_2 + 10x_3 = 8 \end{cases}$$

Chúng ta đưa phương trình về dạng :

$$\begin{cases} x_1 = -\frac{1}{5}x_2 - \frac{1}{10}x_3 + 1 \\ x_2 = -\frac{1}{10}x_1 - \frac{1}{5}x_3 + \frac{6}{5} \\ x_3 = -\frac{1}{10}x_1 - \frac{1}{10}x_2 + \frac{4}{5} \end{cases}$$

Như vậy :

$$B = \begin{bmatrix} 0 & -\frac{1}{5} & -\frac{1}{10} \\ -\frac{1}{10} & 0 & -\frac{1}{5} \\ -\frac{1}{10} & -\frac{1}{10} & 0 \end{bmatrix} \quad \text{và} \quad G = \begin{bmatrix} 1 \\ \frac{5}{6} \\ \frac{4}{5} \end{bmatrix}$$

Dễ thấy $\|B\|_1 = 3/10$; $\|B\|_2 = 3/10$ và $\|B\|_3 = 12/100$ nên phép lặp hội tụ .Chương trình lập đơn là :

Chương trình 10-7

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 10

void main()
{
    float a[max][max];
    float f[max],x0[max],x1[max];
    int i,j,k,n,l,t;
    float s,c,epsi;
    char tl;

    clrscr();
    printf("Cho so phuong trinh 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 a ma 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");
    }
    printf("\n");
    t=1;
    fflush();
    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]);
        fflush();
    }
    if (toupper(tl)=='K')
        t=0;
}
printf("Ma tran a\n");
printf("\n");
for (i=1;i<=n;i++)
{
    for (j=1;j<=n;j++)
        printf("%10.5f",a[i][j]);
    printf("\n");
}
printf("\n");
printf("Cho cac phan tu cua ma tran f : \n");
for (i=1;i<=n;i++)
{
    printf("f[%d] = ",i);
    scanf("%f",&f[i]);
}
printf("\n");
printf("Ma tran f ma ban da nhap");
printf("\n");
for (i=1;i<=n;i++)
    printf("f[%d] = %10.5f\n",i,f[i]);
printf("\n");
t=1;
fflush();
while (t)
{
    printf("Co sua ma tran f khong(c/k)?");
    scanf("%c",&tl);
    if (toupper(tl)=='C')
    {
        printf("Cho chi so hang can sua : ");
        scanf("%d",&i);
        printf("f[%d] = ",i);
        scanf("%f",&f[i]);
        fflush();
    }
    if (toupper(tl)=='K')
        t=0;
}
printf("\n");
printf("Ma tran f");
printf("\n");
for (i=1;i<=n;i++)
    printf("f[%d] = %10.5f\n",i,f[i]);
printf("\n");

```

```

for (i=1;i<=n;i++)
    x0[i]=0.0;
x0[1]=1.0;
printf("Cho so lan lap l = ");
scanf("%d",&l);
epsi=1e-5;
for (i=1;i<=n;i++)
{
    c=1.0/a[i][i];
    for (j=1;j<=n;j++)
        if (j!=i)
            a[i][j]*=c;
    f[i]*=c;
    a[i][i]=0.0;
}
k=1;
t=0;
do
{
    for (i=1;i<=n;i++)
    {
        x1[i]=f[i];
        for (j=1;j<=n;j++)
            x1[i]=x1[i]-a[i][j]*x0[j];
    }
    s=0.0;
    for (i=1;i<=n;i++)
        s=s+fabs(x1[i]-x0[i]);
    if (s>=epsi)
        for (i=1;i<=n;i++)
            x0[i]=x1[i];
    if (s<epsi)
    {
        t=1;
        printf("\n");
        printf("Phep lap hoi tu sau %d buoc tinh",k);
        printf("\n");
        printf("NGHIEM CUA HE");
        printf("\n");
        for (i=1;i<=n;i++)
            printf("x[%d] = %10.5f\n",i,x1[i]);
    }
    k=k+1;
    if (k>l)
    {
        t=1;
        printf("Phep lap khong hoi tu sau %d buoc tinh",k-1);
    }
}
while (t==0);

```



```

    getch();
}

```

§6. PHƯƠNG PHÁP LẬP GAUSS-SEIDEL

Phương pháp lập Gauss-Seidel được cải tiến từ phương pháp lập đơn. Nội dung cơ bản của phương pháp là ở chỗ khi tính nghiệm xấp xỉ thứ $(k+1)$ của ẩn x_i ta sử dụng các xấp xỉ thứ $(k+1)$ đã tính của các ẩn x_1, \dots, x_{i-1} . Giả sử đã cho hệ: $\mathbf{AX} = \mathbf{B}$ và ta có nghiệm:

$$x_i = \beta_i + \sum_{j=1}^n \alpha_{ij} x_j \quad (i=1, \dots, n)$$

Lấy xấp xỉ ban đầu tùy ý $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$ và tất nhiên ta cố gắng lấy chúng tương ứng với x_1, x_2, \dots, x_n (càng gần càng tốt). Tiếp theo ta giả sử rằng đã biết xấp xỉ thứ k $x_i^{(k)}$ của nghiệm. Theo Seidel ta sẽ tìm xấp xỉ thứ $(k+1)$ của nghiệm theo các công thức sau:

$$\begin{aligned} x_1^{(k+1)} &= \beta_1 + \sum_{j=1}^n \alpha_{1j} x_j^{(k)} \\ x_2^{(k+1)} &= \beta_2 + \alpha_{21} x_1^{(k+1)} + \sum_{j=2}^n \alpha_{2j} x_j^{(k)} \\ &\dots\dots \\ x_i^{(k+1)} &= \beta_i + \sum_{j=1}^{i-1} \alpha_{ij} x_j^{(k+1)} + \sum_{j=i}^n \alpha_{ij} x_j^{(k)} \\ &\dots\dots \\ x_n^{(k+1)} &= \beta_n + \sum_{j=1}^{n-1} \alpha_{nj} x_j^{(k+1)} + \alpha_{nn} x_n^{(k)} \end{aligned}$$

Thông thường phương pháp Gauss - Seidel hội tụ nhanh hơn phương pháp lập đơn nhưng tính toán phức tạp hơn. Để dễ hiểu phương pháp này chúng ta xét một ví dụ cụ thể:

Cho hệ phương trình:

$$\begin{cases} 10x_1 + x_2 + x_3 = 12 \\ 2x_1 + 10x_2 + x_3 = 13 \\ 2x_1 + 2x_2 + 10x_3 = 14 \end{cases}$$

nghiệm đúng của hệ là $(1, 1, 1)$

Ta đưa về dạng thuận tiện cho phép lập:

$$\begin{cases} x_1 = 1,2 - 0,1x_2 - 0,1x_3 \\ x_2 = 1,3 - 0,2x_1 - 0,1x_3 \\ x_3 = 1,4 - 0,2x_2 - 0,2x_1 \end{cases}$$

Lấy $x_1^{(0)} = 1,2$; $x_2^{(0)} = 0$; $x_3^{(0)} = 0$;

Sử dụng phương pháp lập Seidel ta có:

$$\begin{cases} x_1^1 = 1,2 - 0,1 \times 0 - 0,1 \times 0 = 1,2 \\ x_2^1 = 1,3 - 0,2 \times 1,2 - 0,1 \times 0 = 1,06 \\ x_3^1 = 1,4 - 0,2 \times 1,2 - 0,2 \times 1,06 = 0,948 \end{cases}$$

$$\begin{cases} x_1^2 = 1,2 - 0,1 \times 1,06 - 0,1 \times 0,948 = 0,9992 \\ x_2^2 = 1,3 - 0,2 \times 0,9992 - 0,1 \times 0,948 = 1,00536 \\ x_3^2 = 1,4 - 0,2 \times 0,9992 - 0,2 \times 1,00536 = 0,999098 \end{cases}$$

và cứ thế tiếp tục. Chương trình mô tả thuật toán Gauss - Seidel như sau:

Chương trình 10-8

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 6

void main()
{
    float b[max],x[max];
    float a[max][max];
    int i,j,k,n,dem,t1;
    float t,s,d,w,epsi;
    char tl;

    clrscr();
    printf("Cho so an so 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 a ma ban da nhap\n");
    printf("\n");
    for (i=1;i<=n;i++)
    {
        for (j=1;j<=n;j++)
            printf("%10.5f",a[i][j]);
        printf("\n");
    }
    printf("\n");
    t1=1;
    fflush();
    while (t1)
    {
        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]);
                fflush();
            }
            if (toupper(tl)=='K')
                t1=0;
        }
        printf("Ma tran a\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");
        printf("Cho cac phan tu cua ma tran b : \n");
        for (i=1;i<=n;i++)
        {
            printf("b[%d] = ",i);
            scanf("%f",&b[i]);
        }
        printf("\n");
        printf("Ma tran b");
        printf("\n");
        for (i=1;i<=n;i++)
            printf("b[%d] = %10.5f\n",i,b[i]);
        printf("\n");
        t1=1;
        fflush();
        while (t1)
        {
            printf("Co sua ma tran b khong(c/k)?");
            scanf("%c",&tl);
            if (toupper(tl)=='C')
            {
                printf("Cho chi so hang can sua : ");
                scanf("%d",&i);
                printf("b[%d] = ",i);
                scanf("%f",&b[i]);
                fflush();
            }
            if (toupper(tl)=='K')
                t1=0;
        }
        printf("\n");
        printf("Ma tran b");
        printf("\n");
        for (i=1;i<=n;i++)
            printf("b[%d] = %10.5f\n",i,b[i]);
        printf("\n");

```

```

printf("Cho so lan lap k : ");
scanf("%d",&k);
printf("\n");
w=1;
epsi=1e-8;

for (i=1;i<=n;i++)
    x[i]=0.0;
dem = 0;
do
{
    dem=dem+1;
    for (i=1;i<=n;i++)
    {
        s=0.0;
        for (j=1;j<=n;j++)
            s=s+a[i][j]*x[j];
        d=x[i];
        x[i]=(1-w)*d+w*(-s+a[i][i]*d+b[i])/a[i][i];
        t=fabs(d-x[i]);
    }
}
while ((dem<=k)&&(t>epsi*fabs(x[n])));
if (t<epsi*fabs(x[n]))
{
    printf("Nghiem sau %d lan lap la :\n",dem);
    for (i=1;i<=n;i++)
        printf("x[%d] = %12.8f\n",i,x[i]);
}
else
{
    printf("Khong dat do chinh xac sau %d lan lap\n",k);
    printf("Nghiem cua lan lap cuoi cung la : \n");
    for (i=1;i<=n;i++)
        printf("x[%d] = %12.8f\n",i,x[i]);
}
getch();
}

```

§7. PHƯƠNG PHÁP CRAMER

Một trường hợp riêng của hệ phương trình , trong đó số phương trình bằng số ẩn , nghĩa là hệ có dạng :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n \end{cases}$$

trong đó $A = (a_{ij})$ là một ma trận không suy biến . một hệ phương trình như vậy có tên là *hệ thống Cramer*

Định lí Cramer : Hệ thống Cramer có nghiệm duy nhất được cho bởi công thức :

$$x_i = \frac{|A^{(i)}|}{|A|} \quad (i=1, \dots, n)$$

trong đó $A^{(i)}$ là ma trận nhận được từ A bằng cách thay cột thứ i bởi cột $B=[b_1, \dots, b_n]^T$

Như vậy để giải hệ bằng phương pháp Cramer chúng ta lần lượt tính các định thức của ma trận và ma trận thay thế rồi tìm nghiệm theo công thức Cramer. Chương trình sau mô tả thuật toán này,

Chương trình 10-9

```
// Cramer;
#include <conio.h>
#include <stdio.h>
#include <ctype.h>
#define max 50

void main()
{
    float r[max][max], a[max][max];
    float b[max], x[max];
    float delta[max];
    int i, j, k, l, t, n, ok1, ok2, t1;
    float c, d;
    char t1;

    clrscr();
    printf("Cho so an cua phuong trinh 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 a ma ban da nhap\n");
    printf("\n");
    for (i=1; i<=n; i++)
    {
        for (j=1; j<=n; j++)
            printf("%10.5f", a[i][j]);
        printf("\n");
    }
    printf("\n");
    t1=1;
    flushall();
    while (t1)
    {
```

```

printf("Co sua ma tran a khong(c/k)?");
scanf("%c",&t1);
if (toupper(t1)=='C')
{
    printf("Cho chi so hang can sua : ");
    scanf("%d",&i);
    printf("Cho chi so cot can sua : ");
    scanf("%d",&j);
    printf("a[",i,"","j,"]=");
    scanf("%f",&a[i][j]);
    flushall();
}
if (toupper(t1)=='K')
    t1=0;
}
printf("Ma tran a \n");
printf("\n");
for (i=1;i<=n;i++)
{
    for (j=1;j<=n;j++)
        printf("%10.5f",a[i][j]);
    printf("\n");
}
printf("\n");
printf("Cho cac phan tu cua ma tran b : \n");
for (i=1;i<=n;i++)
{
    printf("b[%d] = ",i);
    scanf("%f",&b[i]);
}
printf("\n");
printf("Ma tran b ma ban da nhap\n");
printf("\n");
for (i=1;i<=n;i++)
    printf("b[%d] = %10.5f\n",i,b[i]);
printf("\n");
t1=1;
flushall();
while (t1)
{
    printf("Co sua ma tran b khong(c/k)?");
    scanf("%c",&t1);
    if (toupper(t1)=='C')
    {
        printf("Cho chi so hang can sua : ");
        scanf("%d",&i);
        printf("b[%d] = ",i);
        scanf("%f",&b[i]);
        flushall();
    }
}

```

```

        if (toupper(tl)=='K')
            t1=0;
    }
    printf("\n");
    printf("Ma tran b\n");
    printf("\n");
    for (i=1;i<=n;i++)
        printf("%10.5f",b[i]);
    printf("\n");

//thay b vao tung cot cua a de tinh cac dinh thuc
    for (k=0;k<=n;k++)
    {
        for (i=1;i<=n;i++)
            for (j=1;j<=n;j++)//thay cot b vao a
                if (i==k)
                    r[j][i]=b[j];
                else
                    r[j][i]=a[j][i];
//tinh dinh thuc
        d=1.0;
        i=1;
        ok2=1;
        while (ok2&&(i<=n))
        {
            if (r[i][i]==0.0)
            {
                ok1=1;
                t=t+1;
                while (ok1&&(t<=n))
                    if (r[t][i]!=0)
                    {
                        for (j=i;j<=n;j++)
                        {
                            c=r[i][j];
                            r[i][j]=r[t][j];
                            r[k][j]=c;
                        }
                        d=-d;
                        ok1=0;
                    }
                else
                    t=t+1;
            }
            if (k>n)
            {
                printf("\n");
                printf("*** MA TRAN SUY BIEN ***");
                ok2=0;
                d=0.0;
            }
        }
    }

```

```

    }
    if (r[i][i]!=0)
    {
        c=r[i][i];
        for (j=i+1;j<=n;j++)
            r[i][j]=r[i][j]/c;
        for (t=i+1;t<=n;t++)
        {
            c=r[t][i];
            for (j=i+1;j<=n;j++)
                r[t][j]=r[t][j]-r[i][j]*c;
        }
        i=i+1;
    }
    if (ok2)
        for (i=1;i<=n;i++)
            d=d*r[i][i];
    delta[k]=d;
}
if (delta[0]==0.0)
    printf("He da cho vo nghiem\n");
else
{
    printf("\nNGHIEM CUA HE :\n");
    printf("\n");
    for (i=1;i<=n;i++)
    {
        x[i]=delta[i]/delta[0];
        printf("x[%d] = %10.5f\n",i,x[i]);
    }
}
getch();
}

```

§8. HỆ PHƯƠNG TRÌNH SỐ PHỨC

Giả sử ta có một hệ phương trình dạng số phức dạng $AX = B$ trong đó $A = C + jD$, $B = E + jF$ và $X = Y + jZ$. Ta viết lại phương trình dưới dạng :

$$(C + jD)(Y + jZ) = (E + jF)$$

Nhân biểu thức về trái và cân bằng phần thực với phần thực và phần ảo với phần ảo ta nhận được hệ mới :

$$\begin{cases} CY - DZ = E \\ DY + CZ = F \end{cases}$$

Như vậy chúng ta nhận được một hệ gồm $2n$ phương trình số thực. Giải hệ này và kết hợp các phần thực à phần ảo ta nhận được nghiệm của hệ phương trình ban đầu. Chương trình giải hệ phương trình như vậy cho ở dưới đây :

Chương trình 10-10

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <ctype.h>
#define max 20

void main()
{
    int i,j,k,l,n,m;
    float s,t,a[max][max],b[max][max],x[max];

    clrscr();
    printf("Cho so an so cua phuong trinh n = ");
    scanf("%d",&n);
    printf("Cho phan thuc cua cac he so,ke ca ve phai\n");
    for (i=1;i<=n;i++)
        for (j=1;j<=n+1;j++)
        {
            printf("a[%d][%d] = ",i,j);
            scanf("%f",&a[i][j]);
        }
    printf("\n");
    printf("Cho phan ao cua cac he so,ke ca ve phai\n");
    for (i=1;i<=n;i++)
        for (j=1;j<=n+1;j++)
        {
            printf("b[%d][%d] = ",i,j);
            scanf("%f",&b[i][j]);
        }
    for (i=1;i<=n;i++)
        a[i][2*n+1]=a[i][n+1];
    for (i=n+1;i<=2*n;i++)
        a[i][2*n+1]=b[i-n][n+1];
    for (i=n+1;i<=2*n;i++)
        for (j=1;j<=n;j++)
            a[i][j]=b[i-n][j];
    for (i=1;i<=n;i++)
        for (j=n+1;j<=2*n;j++)
            a[i][j]=-b[i][j-n];
    for (i=n+1;i<=2*n;i++)
        for (j=n+1;j<=2*n;j++)
            a[i][j]=a[i-n][j-n];
    m=2*n;
    for (k=1;k<=m-1;k++)
    {
        s=0.0;
```

```

    for (i=k;i<=m;i++)
    {
        t=fabs(a[i][k]);
        if (s<=t)
        {
            s=t;
            l=i;
        }
    }
    for (j=k;j<=m+1;j++)
    {
        s=a[k][j];
        a[k][j]=a[l][j];
        a[l][j]=s;
    }
    if (fabs(a[k][k]/a[l][l])<=1e-08)
    {
        printf("Ma tran suy bien\n");
        getch();
        exit(1);
    }
    for (i=k+1;i<=m;i++)
    {
        s=a[i][k]/a[k][k];
        a[i][k]=0.0;
        for (j=k+1;j<=m+1;j++)
            a[i][j]-=s*a[k][j];
    }
}
x[m]=a[m][m+1]/a[m][m];
for (i=m-1;i>=1;i--)
{
    s=a[i][m+1];
    for (j=i+1;j<=m;j++)
        s-=a[i][j]*x[j];
    x[i]=s/a[i][i];
}
printf("\n");
printf("Nghiem phuc cua he\n");
for (i=1;i<=n;i++)
    if (x[i+n]<0)
        printf("%10.5f-%10.5fj\n",x[i],fabs(x[i+n]));
    else
        printf("%10.5f+%10.5fj\n",x[i],x[i+n]);
getch();
}

```

Dùng chương trình này giải hệ phương trình :

$$\begin{cases} (3+7j)x + (-2+4j)y + (1-3j)z + (4+2j)r = 8+36j \\ (5-6j)x + (2+5j)z + (-3+j)r = 4+10j \\ (4+5j)x + (1+2j)y + (-5-j)z + 6r = 13-3j \\ (2+4j)x + (1-j)y + (2-3j)r = -10+6j \end{cases}$$

Ta nhận được các nghiệm $x = 2 + 3j$; $y = 1 - 2j$; $z = -1 + 4j$ và $r = 1 - j$

Ngoài các phương pháp nêu trên ta thấy rằng từ hệ phương trình $AX = B$ ta có thể tìm nghiệm X của hệ bằng cách viết lại phương trình dưới dạng $X = B/A = A^{-1}B$ với A^{-1} là ma trận nghịch đảo của A . Do vậy trước hết ta cần tìm A^{-1} và sau đó tính tích $A^{-1}B$.