

# Rozwiązanie zadania N7

Krzysztof Waniak

Znaleźć wartości własne macierzy z dokładnością  $10^{-8}$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & -1 \end{bmatrix}$$

Wartości własne macierzy znalezione za pomocą metody iteracyjnej QR.

Kod programu:

```
#include<stdio.h>
#include<math.h>
#include<ctype.h>          /* zawiera F_OK itp.    */
#include<unistd.h>         /* zawiera funkcje access(), usleep() */
#define wyp(a) printf(##a "\n")
#define wyp2(a) printf(##a)
#define wypisz(a) printf("%9.5f",a)
#define wypisz2(a) printf("%i",a)
#define karetka printf("\n")
#define karetka2 printf("\n\n")

/* Sprawdzanie, czy wejściowy plik nie istnieje, jeśli nie istnieje zwraca
wartosc "TRUE" */
int nieistnieje(const char* nazwa)
{
    return access(nazwa, F_OK);
}

void QR(double A[][3], double Q[][3], double R[][3])
{
    int i,j,k;
    double u2[3];
    double u3[3];
    double v1[3];
    double v2[3];
    double v3[3];
    double proj1[3];
    double proj2[3];
    double norma;
    double suma = 0;
    double suma1 = 0;

    for(i = 0; i<3; i++)
    {
        v1[i] = A[i][0];
    }

    for(i = 0; i<3; i++)
    {
        norma += v1[i]*v1[i];
    }

    norma = sqrt(norma);
```

```

for(i = 0; i<3; i++)
{
    Q[i][0] = v1[i]/norma;
}

norma = 0;
for(i = 0; i<3; i++)
    v2[i] = A[i][1];

for(i = 0; i<3; i++)
{
    suma += v1[i]*v2[i];
    suma1 += v1[i]*v1[i];
}

for(i = 0; i<3; i++)
{
    proj1[i] = (suma/suma1)*v1[i];
}

for(i = 0; i<3; i++)
{
    u2[i] = v2[i] - proj1[i];
}

for(i = 0; i<3; i++)
{
    norma += u2[i]*u2[i];
}

norma = sqrt(norma);

for(i = 0; i<3; i++)
{
    Q[i][1] = u2[i]/norma;
}

norma = 0;

suma = 0;
suma1 = 0;
for(i = 0; i<3; i++)
{
    v3[i] = A[i][2];
}

for(i = 0; i<3; i++)
{
    suma += v1[i]*v3[i];
    suma1 += v1[i]*v1[i];
}

for(i = 0; i<3; i++)
{
    proj1[i] = (suma/suma1)*v1[i];
}

suma = 0;
suma1 = 0;

for(i = 0; i<3; i++)

```

```

    {
        suma += u2[i]*v3[i];
        suma1 += u2[i]*u2[i];
    }

    for(i = 0; i<3; i++)
    {
        proj2[i] = (suma/suma1)*u2[i];
    }

    for(i = 0; i<3; i++)
    {
        u3[i] = v3[i] - proj1[i] - proj2[i];
    }

    for(i = 0; i<3; i++)
    {
        norma += u3[i]*u3[i];
    }

    norma = sqrt(norma);

    for(i = 0; i<3; i++)
    {
        Q[i][2] = u3[i]/norma;
    }

/* tworzenie macierzy R */
    suma = 0;
    for(j = 0; j<3; j++)
    {
        for(k = 0; k<3; k++)
        {
            for(i = 0; i<3; i++)
            {
                suma += Q[i][k]*A[i][j];
            }
            R[k][j] = suma;
            suma = 0;
        }
    }
}

void mnoz(double R[][3], double Q[][3], double B[][3])
{
    double suma = 0;
    int i,j,k;

    for(j = 0; j<3; j++)
    {
        for(k = 0; k<3; k++)
        {
            for(i = 0; i<3; i++)
            {
                suma += R[k][i]*Q[i][j];
            }
            B[k][j] = suma;
            suma = 0;
        }
    }
}

```

```

void przypisz(double b[][3], double b1[][3])
{
    int i,j;
    for(i = 0; i<3; i++)
    {
        for(j = 0; j<3; j++)
        {
            b1[i][j] = b[i][j];
        }
    }
}

int main()
{
    int i;
    FILE *fwynik;
    char plik_b[30];
    int d;
    double a[3][3];
    double q[3][3];
    double r[3][3];
    double b[3][3];
    double b1[3][3];

    wyp(Podaj nazwe pliku wyjsciowego:);
    scanf("%s", &plik_b[0]);
    while(!nieistnieje(plik_b))
    {
        karetk;
        wyp(Taka nazwa pliku juz istnieje);
        wyp(Wprowadz inna nazwe pliku);
        karetk;
        scanf("%s", &plik_b[0]);
    }
    karetk;
    a[0][0]=1;
    a[0][1]=2;
    a[0][2]=3;
    a[1][0]=2;
    a[1][1]=4;
    a[1][2]=5;
    a[2][0]=3;
    a[2][1]=5;
    a[2][2]=-1.0;

    do
    {
        QR(a,q,r);
        mnoz(q,r,b);
        mnoz(r,q,b1);
        przypisz(b1,a);
    }while( fabs(b1[0][0] - b[0][0]) > 0.00000001 );

    fwynik = fopen(plik_b, "w");
    wyp(Wartosci wlasne:);
    karetk;
    for(i = 0; i < 3; i++)
    {
        fprintf(fwynik,"%1.5f \n", b1[i][i]);
    }
}

```

```

        wypisz(b1[i][i]);
        karetka;
    }
    fclose(fwynik);

    karetka2;
    d=0;
    wyp2(Wynik obliczony i zapisany jako:);
    space;
    while(plik_b[d]!='\0')    printf("%c",plik_b[d++]);
    karetka2;

    /*system("pause");*/
    return 0;
}

```

Wynik działania programu zapisany do pliku wynik.txt:

Działanie aplikacji:

```

[waniak@ibisnet N7]$ make run
./N7
Podaj nazwe pliku wyjsciowego:
w.txt

Wartosci wlasne:

8.548513
-4.574087
0.025574

Wynik obliczony i zapisany jako: w.txt

```