

Rozwiązanie zadania N2

Krzysztof Waniak

Rozwiązać układ równań:

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

```
#include<stdio.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("%lf",a)
#define karetka printf("\n")
#define karetka2 printf("\n\n")
#define space printf(" ")

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

void gb(int na,int nband,double mb[4][8],double wb[8],double wa[8])
{
    /*
    podprogram korzysta tylko z wektora, który zawiera diagonalę, jak wiemy,
    reszta pustej macierzy nie jest nazbyt potrzebna
    */
    int i,j,k,nn;
    double q;
    nn = (nband+1)/2;

    for( i=1;i<=na-1; i++)
    {
        for( k=2;k<=nn; k++)
        {
            if ((k+i-1)<= na)
            {
                q= (mb[nn-k+1][k+i-1]) / (mb[nn][i]);
                for( j=nn+1; j<=nband; j++)
                {
                    mb[j-k+1][k+i-1] = mb[j-k+1][k+i-1]-mb[j][i]*q;
                }
                wb[k+i-1] = wb[k+i-1]-wb[i]*q;
            }
        }
    }
}
```

```

for( i=na;i>=1;i--)
{
    for( j=i+1;j<=na; j++)
    {
        if ((nn+j-i) <= nband)
        {
            wb[i] = wb[i]-mb[nn+j-i][i]*wa[j];
        }
    }

    wa[i] = wb[i]/mb[nn][i];
}
}

int main(void)
{
    FILE *fdane, *fwynik;
    int i, j, d, na = 7, nband = 3;
    double mb[4][8];
    double wb[8];
    /*wektor wynikow*/
    double wa[8];
    /*wektor niewiadomych*/
    char plik_a[30];
    char plik_b[30];

    for(i = 0; i <= 7; i++)
    {
        wb[i] = 0.0 + i;
        wa[i] = 0.0;
        for(j = 0; j <= 3; j++)
        {
            mb[j][i] = 0.0;
        }
    }

    wyp(Podaj nazwe pliku wejscowego);
    scanf("%s", &plik_a[0]);

    while(nieistnieje(plik_a))
    {
        karetk;
        wyp(Taki plik nie istnieje);
        wyp(Wprowadz nazwe istniejacego pliku);
        karetk;
        scanf("%s", &plik_a[0]);
    }
    karetk;
    wyp(Podaj nazwe pliku wyjscowego);
    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;
}

```

```

fdane = fopen(plik_a, "r");
for(i = 1; i <= 7 ; i++)
{
    for(j = 1; j <= 3; j++)
    {
        fscanf(fdane, "%lf", &mb[j][i]);
    }
}

fclose(fdane);

karetka;
gb( na, nband, mb, wb, wa);

fwynik = fopen(plik_b, "w");
for(i = 1; i <= 7; i++)
{
    fprintf(fwynik, "x%i = %f \n", i, wa[i]);
    printf("x%i = %f", i, wa[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;

return 0;
}

```

Wynik działania programu:

```

x1 = 0.166789
x2 = 0.332842
x3 = 0.501841
x4 = 0.659794
x5 = 0.858984
x6 = 0.904271
x7 = 1.523932

```