

Rozwiązanie zadania N5

Krzysztof Waniak

Dla układu równań $Ax = b$, gdzie:

$$A = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}$$

Znaleźć rozwiązanie za pomocą metody Gaussa-Seidla z wybraną dokładnością.

Kod programu:

```
#include<stdio.h>
#include<stdlib.h>
#include<gsl/gsl_linalg.h>
#include<gsl/gsl_blas.h>
#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(" ")

int main(void)
{
    /* rozwiazane za pomoca funkcji bibliotecznych GSL */
    double a0[] = { 4.0, 0.0, 0.0, 0.0, 4.0, 0.0, 0.0, 0.0, 4.0 };
    double a1[] = { 0.0, -1.0, 0.0, -1.0, 0.0, -1.0, 0.0, -1.0, 0.0 };
    double x[] = { 0.0, 0.0, 0.0 };
    double r[] = { 2.0, 6.0, 2.0 };
    double pomocniczy_wek_2[3];
    double wektor[9];
    double pomocniczy_wek_1[3];

    gsl_matrix_view D = gsl_matrix_view_array(a0, 3, 3);
    gsl_matrix_view R = gsl_matrix_view_array(a1, 3, 3);
    gsl_matrix_view DO = gsl_matrix_view_array(wektor, 3, 3);
    gsl_matrix_view Temp = gsl_matrix_view_array(pomocniczy_wek_1, 3, 1);
    gsl_matrix_view X = gsl_matrix_view_array(x, 3, 1);
    gsl_matrix_view Btmp = gsl_matrix_view_array(pomocniczy_wek_2, 3, 1);

    gsl_permutation *permutacja;
    int p0, p1, p2, p3, dokladnosc;
    p3 = 0;
    permutacja = gsl_permutation_alloc(3);
    gsl_linalg_LU_decomp(&D.matrix, permutacja, &p0);
    gsl_linalg_LU_invert(&D.matrix, permutacja, &DO.matrix);

    karetka;
    wyp(Podaj zadana dokladnosc:);
    space;
    scanf("%i", &dokladnosc);

    while(p3 != dokladnosc)
    {
```

```

        gsl_blas_dgemm(CblasNoTrans, CblasNoTrans, 1.0, &R.matrix,
&X.matrix, 0.0, &Temp.matrix);

        for(p1 = 0; p1 < 3; ++p1)
        {
            for(p2 = 0; p2 < 1; ++p2)
            {
                pomocniczy_wek_2[p1] = r[p1] -
gsl_matrix_get(&Temp.matrix, p1, p2);
            }
        }
        Btmp = gsl_matrix_view_array(pomocniczy_wek_2, 3, 1);
        gsl_blas_dgemm(CblasNoTrans, CblasNoTrans, 1.0, &DO.matrix,
&Btmp.matrix, 0.0, &X.matrix);
        ++p3;
    }

    printf("x(%i) = \n", p3);

    for(p1 = 0; p1 < 3; ++p1)
    {
        for(p2 = 0; p2 < 1; ++p2)
        {
            if(p2 == 0)
            {
                printf("%lf\n", gsl_matrix_get(&X.matrix,p1,p2));
            }
            else
            {
                printf("%lf ",gsl_matrix_get(&X.matrix,p1,p2));
            }
        }
    }

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

```