Program do zadania Nr.3:

1. Eliminacja Gaussa:

```cpp
#include <iostream>
#include <cmath>

using namespace std;

double** Matrix() {
    double matrix[5][5]{
            { -116.66654, 583.33346 , -333.33308 , 100.00012 , 100.00012},
            { 583.33346 , -116.66654 , -333.33308, 100.00012 , 100.00012},
            { -333.33308, -333.33308 , 133.33383 , 200.00025 , 200.00025},
            { 100.00012 , 100.00012  , 200.00025 , 50.000125 , -649.99988},
            { 100.00012 , 100.00012 , 200.00025 , -649.99988 , 50.000125}};
    auto** retMatrix = new double*[5];
    for (int i = 0; i < 5; ++i) {
        retMatrix[i] = new double[5];
    }
    for(int i = 0; i < 5; i++) {
        for(int j = 0; j < 5; j++) {
            retMatrix[i][j] = matrix[i][j];
        }
    }
    return retMatrix;
}

double* gaussElimination(double **tab, double *res, int n) {
    for(int i = 0; i < n; i++) {
        double div = tab[i][i];

        for(int j = i; j < n; j++) {
            tab[i][j] /= div;
        }
        res[i] /= div;
        if(i+1 < n) {
            for(int k = i+1; k < n; k++) {
                double mul = tab[i][i]*tab[k][i];
                for(int j = i; j < n; j++) {
                    tab[k][j] -= mul*tab[i][j];
                }
                res[k] -= mul*res[i];
            }
```

2. Back substitution i obliczanie normy wektorów:

```
            }
        }
    }
    return res;
}

void back_substitution(double **tab, double* res, int n) {
    for(int i = n-1; i > 0; i--) {
        for(int j = i-1; j>= 0; j--) {
            res[j] -= tab[j][i]*res[i];
        }
    }
}

void printResults(double* res, int n) {
    for(int i = 0; i < n; i++) {
        printf("z%d = %.12f  ", i+1, res[i]);
    }
    printf("\n");
}

double* sub(const double* a,const double* b){
    auto* c = new double[5];
    for (int i = 0; i < 5; i++) {
        c[i] = a[i] - b[i];
    }
    return c;
}

double getNorma(const double* normaM){
    double norma = 0;
    for (int i = 0; i < 5; ++i) {
        norma+= pow(normaM[i], 2);
    }
    return sqrt(norma);
}

int main() {
    double *b1{new double[5]{-0.33388066, 1.08033290, -0.98559856, 1.31947922, -0.09473435}};
    double *b2{new double[5]{-0.33388066, 1.08033290, -0.98559855, 1.32655028, -0.10180541}};
    double *b3{new double[5]{0.72677951, 0.72677951, -0.27849178, 0.96592583, 0.96592583}};
    double *b4{new double[5]{0.73031505, 0.73031505, -0.27142071, 0.96946136, 0.96946136}};
    int i = 5;
    cout.precision(15);
    double Norma1 = getNorma(sub(b1,b2));
    double Norma2 = getNorma(sub(b3,b4));
    double **arr0 = Matrix();
    double *z1 = gaussElimination(arr0, b1, i);
    back_substitution(arr0, b1, i);
    double *z2 = gaussElimination(Matrix(), b2, i);
    back_substitution(arr0, b2, i);
    double Norma3 = (getNorma(sub(z1,z2))/Norma1);
    double *z3 = gaussElimination(Matrix(), b3, i);
    back_substitution(arr0, b3, i);
    double *z4 = gaussElimination(Matrix(), b4, i);
    back_substitution(arr0, b4, i);
    double Norma4 = (getNorma(sub(z3,z4))/Norma2);

    cout << "A*z = b1: " << endl;
    printResults(z1,i);
    cout << "A*z = b2: " << endl;
    printResults(z2,i);
    cout << "A*z = b3: " << endl;
    printResults(z3,i);
    cout << "A*z = b4: " << endl;
    printResults(z4,i);

    cout << "||b1-b2|| = " << Norma1 << endl;
    cout << "||b3-b4|| = "<< Norma2 << endl;
    cout << "||z1-z2|| / ||b1-b2|| = "<< Norma3 << endl;
    cout << "||z3-z4|| / ||b3-b4|| = "<< Norma4 << endl;
}
```

Wyniki do zadania Nr.3:

```
A*z = b1:
z1 = 0.001982859550  z2 = -0.000037445535  z3 = -0.000219649469  z4 = 0.000240550981  z5 = -0.001779754105
A*z = b2:
z1 = 0.001985368955  z2 = -0.000034936130  z3 = -0.000214630646  z4 = 0.000253161908  z5 = -0.001787346206
A*z = b3:
z1 = 354.885181398806  z2 = 354.885181398806  z3 = 709.768197734129  z4 = 354.883432436151  z5 = 354.883432436151
A*z = b4:
z1 = 358.434024577320  z2 = 358.434024577319  z3 = 716.865884091143  z4 = 358.432275548128  z5 = 358.432275548128
||b1-b2|| = 0.00999998895235883
||b3-b4|| = 0.0100000030944946
||z1-z2|| / ||b1-b2|| = 0.00159517667351682
||z3-z4|| / ||b3-b4|| = 1003.7641154378
```

Komentarze do zadania Nr.3:

W tym zadaniu było użyto metody eliminacji Gaussa i Back substitution. Metoda ta sprowadza macierz danego układu równań do macierzy jednostkowej lub macierzy trójkątnej, a później szukałem normy wektorów – podanych w treści zadania.

1. Eliminacja Gaussa i Back Substitution

$\mathbf{z_i} = \mathbf{A}^{-1}\mathbf{b_i}$ gdzie $i = 1, 2, 3, 4.$ - możemy zapisać jako: A*z = b_i i rozwiązujemy te cztery równania na „z" za pomocą eliminacji Gaussa i metody Back Substitution.

Przekształcenie macierzy do macierzy trójkątnej górnej odbywa się etapowo w dwóch krokach, które realizowane są dla wartości i zmieniającej się od 1 do n:

1. Podzielić i-ty wiersz macierzy przez odwrotność jej argumentu ai,i w następujący sposób:

$$W_i' = \frac{W_i}{a_{i,i}}$$

2. Dla każdego wiersza o indeksie k takim że i<k≤n odjąć otrzymany wiersz Wi' przemnożony przez argument ai,k:

$$W_k' = W_k + W_i' \cdot a_{k,i}$$

Aby zredukować do zera argumenty macierzy znajdujące się nad przekątną główną należy operację odejmowania przeprowadzić dla k∈{1, 2, ..., n}{i}.

Niewiadome o indeksach i=n-1, n-2, ... , 1 obliczyć można za pomocą następującego wzoru:

$$x_i = b_{i,n+1} - \sum_{k=i+1}^{n} b_{i,k} \cdot x_k$$

2. Norma wektorów

Normą wektora nazywamy funkcję ‖·‖ : V → R, spełniającą następujące warunki: (x,y∈V):

1.‖x‖>0 ∧ ‖x‖= 0⇔x= 0.
2.‖αx‖=|α|·‖x‖α∈C.
3.‖x+y‖ <= ‖x‖+‖y‖.

Norma wektora jest nieujemną liczbą rzeczywistą, przy czym jedynym wektorem o zerowej normie jest wektor zerowy.

Dla rozwiązywania norm wektorów podanych w treści zadania, na początku skorzystałem z funkcji sub() dla odejmowania znaczeń b1-b2 itd.. Dalej korzystamy z:

Dla x = $\begin{matrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \end{matrix}$: $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2}$