Temat: Rozwiązywanie układów równań liniowych metodami iteracyjnymi			
Wykonał: Marcin Fabrykowski	Wydział: FiIS	Kierunek Inf. Stos.	Grupa: grupa 3

1. Metoda Jakobiego
   Mając równanie Ax=b,
   gdzie A=L+D+U,
   przy czym:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ l_{21} & 0 & 0 \\ l_{31} & l_{32} & 0 \end{bmatrix} + \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} + \begin{bmatrix} 0 & u_{12} & u_{13} \\ 0 & 0 & u_{23} \\ 0 & 0 & 0 \end{bmatrix} \tag{1}$$

   co po podstawieniu daje:
   $Lx + Dx + Ux = b$

$$\begin{bmatrix} 0 & 0 & 0 \\ l_{21} & 0 & 0 \\ l_{31} & l_{32} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & u_{12} & u_{13} \\ 0 & 0 & u_{23} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{2}$$

   przekszatałcając...
   $Dx = b - (L + U)x$

$$\begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} - \left[\begin{bmatrix} 0 & 0 & 0 \\ l_{21} & 0 & 0 \\ l_{31} & l_{32} & 0 \end{bmatrix} + \begin{bmatrix} 0 & u_{12} & u_{13} \\ 0 & 0 & u_{23} \\ 0 & 0 & 0 \end{bmatrix} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3}$$

   mnożąc obie strony przez D^{-1} otrzymamy koncowy wzór:

$x^{(x+)1} = -D^{-1}(L+U)x^{(i)} + D^{-1}b$

$$\begin{bmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ x_3^{(i+1)} \end{bmatrix} = -\begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{a_{22}} & 0 \\ 0 & 0 & \frac{1}{a_{33}} \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & 0 \\ l_{21} & 0 & 0 \\ l_{31} & l_{32} & 0 \end{bmatrix} + \begin{bmatrix} 0 & u_{12} & u_{13} \\ 0 & 0 & u_{23} \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ x_3^{(i)} \end{bmatrix} - \begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{a_{22}} & 0 \\ 0 & 0 & \frac{1}{a_{33}} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4}$$

2. Wykonanie ćwiczenia

   Celem ćwiczenia jest znalezienie rozwiązania równania różniczkowego:

$$\frac{d^x}{dt^2} = -\omega^2 x - \beta V + F_0 \sin(\Omega t) \tag{5}$$

Zamieniając drugą pochodną na symetryczny trójpunktowy iloraz różnicowy otrzymamy:

$$\frac{x_{i-1} - 2x_i + x_{i+1}}{h^2} = -\omega^2 x_i - \beta V_i + F_0 \sin(\Omega h i) \tag{6}$$

$$x_{i-1} - 2x_i + x_{i+1} + \omega^2 h^2 x_i + \beta h(x_{i+1} - x_i) = F_0 \sin(\Omega h i) h^2 \tag{7}$$

co zapisując ogólnie przyjmuje postać:

$$a_1 x_{i-1} + a_2 x_i + a_3 x_{i+1} = b_i \tag{8}$$

gdzie:

$$\begin{aligned} a_1 &= 1 \\ a_2 &= \omega^2 h^2 - 2 - \beta h \\ a_3 &= 1 + \beta h \\ b_i &= F_0 \sin(\Omega h i) h^2 \end{aligned}$$

przyjmując dodatkowo warunki początkowe:

$$\begin{aligned} x(t=0) &= x_0 = 1 \\ V(t=0) &= V_0 = 0 \end{aligned}$$

możemy zapisac całość w postaci macierzowej:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & ... & 0 \\ -1 & 1 & 0 & 0 & 0 & ... & 0 \\ a_1 & a_2 & a_3 & 0 & 0 & ... & 0 \\ 0 & a_1 & a_2 & a_3 & 0 & ... & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & ... & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ b_2 \\ ... \\ b_n \end{bmatrix} \tag{9}$$

Ogólny wzór na wyznaczenie wektora x to:

$$x^{(x+)1} = -D^{-1}(L+U)x^{(i)} + D^{-1}b$$

jednak wykorzystamy fakt, że nasza macierz jest rzadka i U jest macierzą zerową, a D jest dwuprzekątną i przekształcimy wzór do postaci:

$$x^{(x+1)} = D^{-1}\left(b - d_1x^{(x)} - d_2x^{(x)}\right)$$

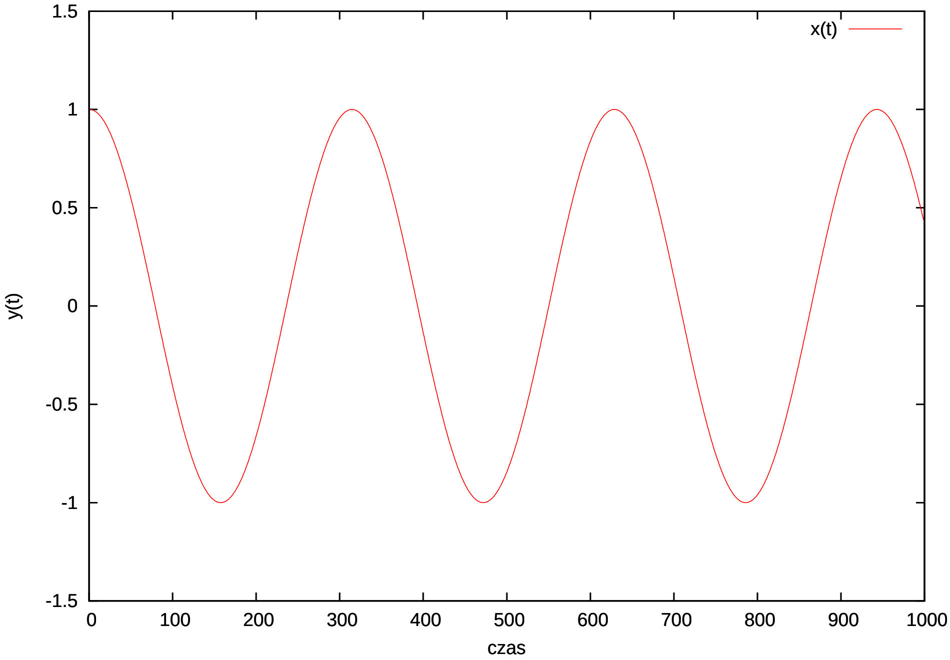
Program wykonujący to ćwiczenie:

Listing 1: main.cpp

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <math.h>
#include <stdio.h>
using namespace std;
int main(int argc,char* argv[] )
{
        if(argc<6)
        {
                cout<<"Usage: ./main n h B F0 Omega"<<endl;
                return -1;
        };
        float *d0,*d1,*d2,*x,*b;
        float h=atof(argv[2]);
        int n=atoi(argv[1]);
        float w=1;
        float B=atof(argv[3]),F=atof(argv[4]),Omega=atof(argv[5]);
        float a1=1;
        float a2=w*w*h*h-B*h-2;
        float a3=(1+B*h);
        d0=new float[n];
        d1=new float[n];
        d2=new float[n];
        b=new float[n];
        x=new float[n];
        memset(d0,0,n);
        memset(d1,0,n);
        memset(d2,0,n);
```

```
        memset(x,0,n);
        memset(b,0,n);
        int i;
        for(i=0;i<n;i++)
        {
                d0[i]=a3;
                d1[i]=a2;
                d2[i]=a1;
                b[i]=F*sin(Omega*h*i)*h*h;
        };
        d0[0]=1.0;
        d1[0]=0.0;
        d2[0]=0.0;

        d0[1]=1.0;
        d1[1]=-1.0;
        d2[1]=0.0;

        b[0]=1;
        b[1]=0;
        x[0]=1;
        x[1]=1;

        int j;
        for(i=2;i<n;i++)
        {
                x[i]=1/d0[i]*(b[i]-d2[i]*x[i-2]-d1[i]*x[i-1]);
        };

        FILE* plik=fopen("dane.dat","w");
        for(i=0;i<n;i++)
        {
                fprintf(plik,"%d %f\n",i,x[i]);
        };
        fclose(plik);
        delete [] d0;
        delete [] d1;
        delete [] d2;
        delete [] x;
        delete [] b;
};
```

Rysunek 1: Plot1



Naszym celem było wyznaczenie rozwiązań przy parametrach:

$$V_0 = 0$$
$$x_0 = 1$$
$$\omega = 1$$
$$n = 1000$$
$$h = 0.02$$

oraz

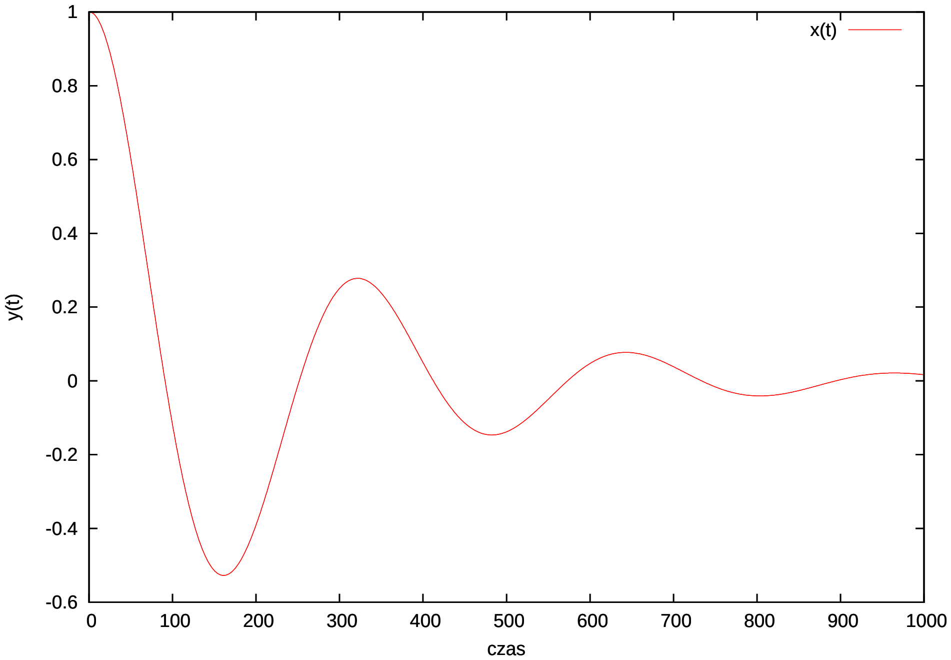
(a) $\beta = 0.0, F_0 = 0.0, \Omega = 0.8$
wycznik czego przedstawia rys. 1

(b) $\beta = 0.4, F_0 = 0.0, \Omega = 0.8$
wynik czego przedstawia rys. 2

(c) $\beta = 0.4, F_0 = 1.0, \Omega = 0.8$
czego wynik przedstawia rys. 3

3. Wnioski
Biorąc pod uwagę liczbę operacji wykonywanych podczas działania tego algorytmu, łatwo zauważyć, że jest on przystosowany do macierzy rzadkich

Rysunek 2: Plot2



Rysunek 3: Plot3