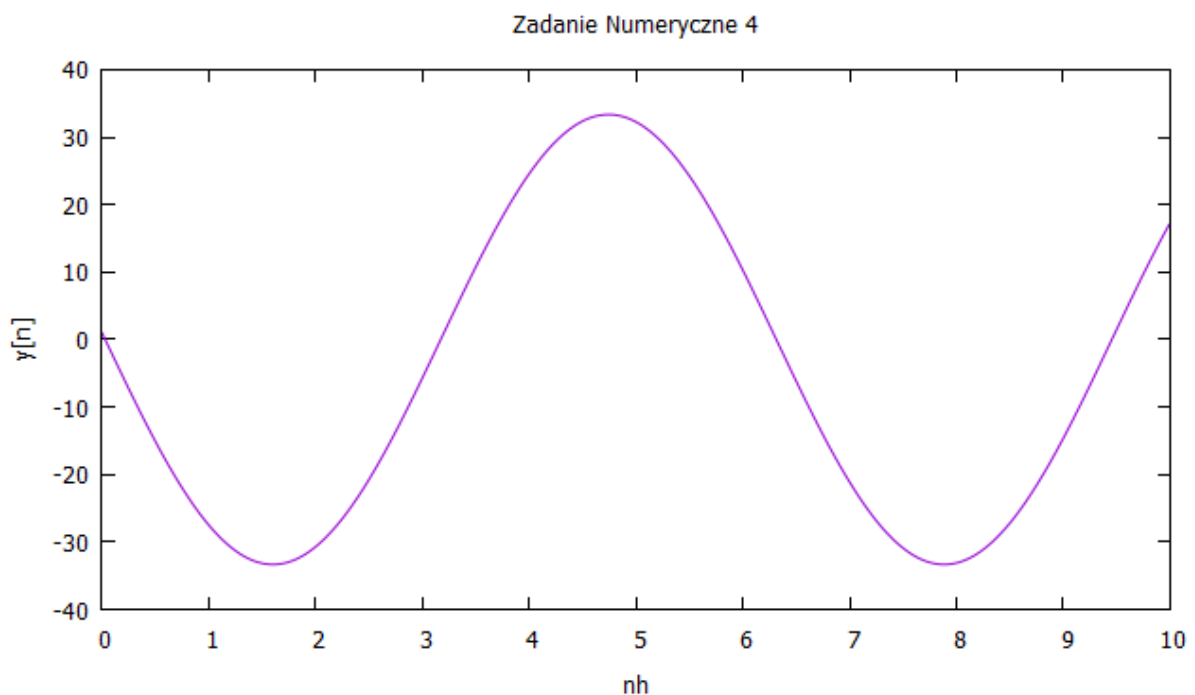


#### Zadanie Numeryczne 4

Zadanie polegało na rozwiązaniu układu równań. Wiersze zostały tak przestawiane aby otrzymać podany układ.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 4 & 0 & 0 \\ 1 & -1.9999 & 1 & 0 \\ 0 & 1 & -1.9999 & 1 \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

Następnie w sposób jawny zostały wyliczone pierwsze trzy wyrazy  $y_0$   $y_1$   $y_2$ , które zostały wykorzystane do obliczenia reszty wyrazów. Aby licząc wyrazy w pętli odwoływać się do wcześniejszych dwóch wyrazów.



Dzięki zastosowaniu tablic zamiast wektorów udało się uniknąć czasochłonnych operacji `push_back`. Do potęgowania nie używam funkcji „`pow`” tylko wielokrotnego mnożenia. Do wypisania wyników nie używam „`cout`” tylko „`printf`”. Dzięki tym zabiegom udało się uzyskać lepszą złożoność obliczeniową.

```

#include <stdio.h>
#include <math.h>
#include <fstream>
#include <iostream>

using namespace std;
const int N = 1000;
const double h = 0.01;
const double h2 = h * h;

int main()
{
    double y[N];

    y[0] = 1;
    y[1] = (-4.0) / (-6.0 + h2);
    y[2] = -1.0 - (-2+h2) * y[1];

    for (int i = 2; i < N - 1; i++) {
        y[i + 1] = (-1.0)*y[i - 1] + (2.0*y[i]) - (h2*y[i]);
    }

    for (int i = 0; i < N - 1; i++) {
        printf("%f \n", y[i]);
    }

    /*
    fstream plik;
    plik.open("ZN4.txt", ios::out);

    if (plik.good() == true)
    {
        for (int n = 0; n <= N - 1; n++)
        {
            plik << n * h << " " << y[n] << endl;
        }
    }
    else
    {
        printf("Nie moge otworzyc pliku ZN4.txt do zapisu!\n");
        exit(1);
    }
    plik.close();
    */

    return 0;
}

```