// Autor: Daniel Szarek

//====================

Zawartość folderu z zadaniem:

- main.cpp – Kod z rozwiązaniem zadania napisanym w języku C++
- Makefile – Do uruchomienia programu main.cpp (Makefile oferuje możliwość uruchomienia programu za pomocą komendy 'make run', usunięcia plików po uruchomieniu programu komendą 'make clean' oraz możliwość utworzenia paczki .tar.gz z zawartością foldera z zadaniem za pomocą komendy 'make tar'
- Opracowanie.pdf – Ten dokument z opracowaniem zadania N5
- wykres.gnu – Skrypt do programu gnuplot do utworzenia wykresu podanego w opracowaniu
- wykres.png – Gotowy wykres, zrobiony przez skrypt
- wyniki.txt – Plik tekstowy z wynikami programu main.cpp

Do zrealizowania zadania wykorzystałem GSL - GNU Scientific Library, w Makefile zawarłem komendy do uruchomienia zadania na swojej maszynie w systemie Ubuntu, w celu uruchomienia zadania na swoim sprzęcie należy w prawidłowy sposób zainstalować bibliotekę GSL oraz zmienić scieżkę INCLUDEPATH1 dla swojej maszyny.

Treść Zadania:

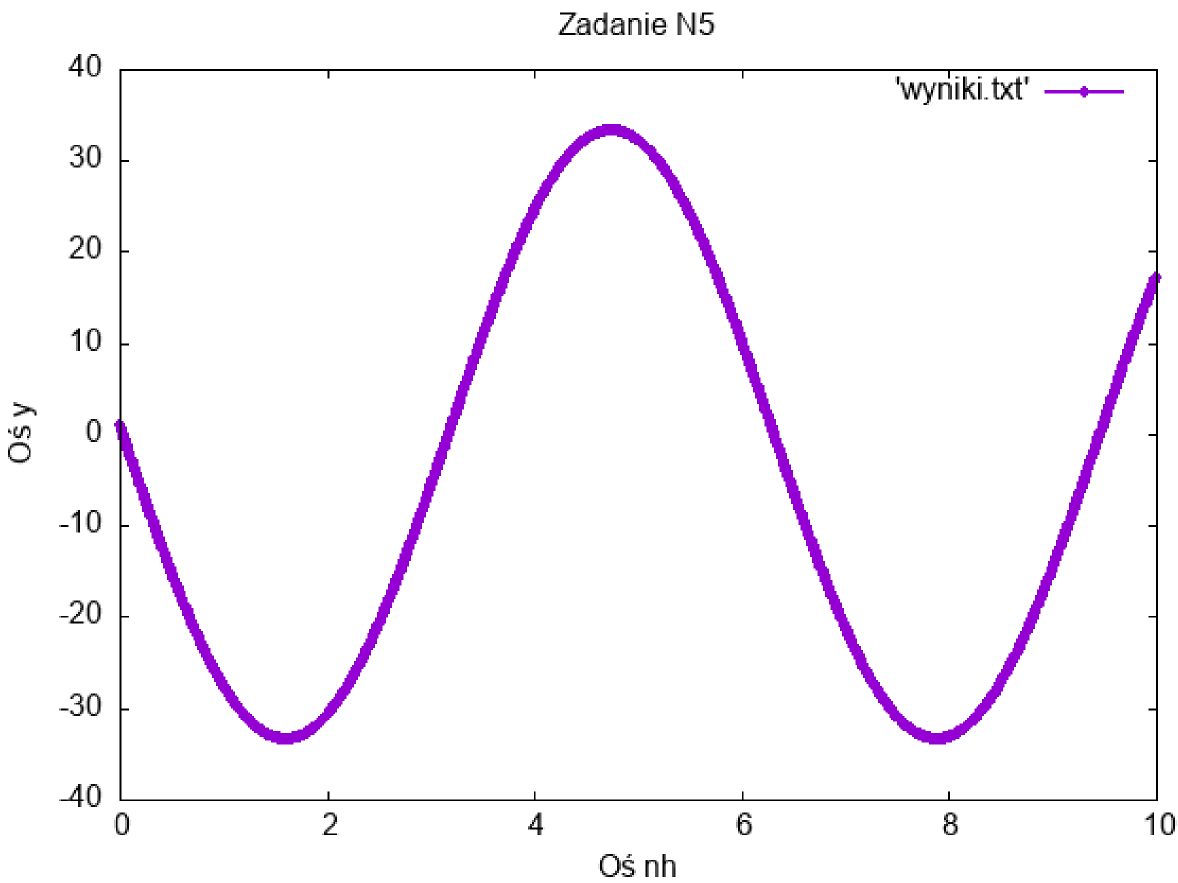
N5 *Zadanie numeryczne*
Rozwiązać układ $(N+1)\times(N+1)$ równań postaci

$$\begin{cases} y_0 & = 1 \\ (D_2 y)_n + y_n & = 0\,, \quad n = 1\ldots(N-1) \\ -3y_0 + 4y_1 + y_2 & = 0\,, \end{cases} \tag{5}$$

gdzie $N = 1000$, $h = 0.01$, a

$$(D_2 y)_n = \frac{y_{n-1} - 2y_n + y_{n+1}}{h^2}. \tag{6}$$

Rozwiązanie w postaci graficznej:



Omówienie:

W ramach rozwiązania zadania skorzystałem z wskazówki podanej w treści zadania.

Wskazówka: Poprzestawiać układ równań tak, aby wyliczyć pierwsze wyrazy w sposób jawny.

Dopatrzyłem się, że ostatnim równaniu w naszym układzie równań, podanym poniżej.

$$\begin{cases} y_0 & = 1 \\ (D_2 y)_n + y_n & = 0\,, \quad n = 1 \ldots (N-1) \\ -3y_0 + 4y_1 + y_2 & = 0\,, \end{cases}$$

Występują elementy podane w lewym dolnym rogu macierzy A, która będzie naszą macierzą układu równań.

$$-3y_0 + 4y_1 + y_2 \quad = 0\,,$$

W celu obliczenia kolejnych wartości macierzy z A korzystam ze wzoru (D2y)n + yn = 0, implementacje n = 1 oraz n = 2, pokazuje poniżej.

$y_0 = 1$, $y_{1000} = 0$, bo $N = 1000$

Obliczamy pierwszy wiersz macierzy:

$$\frac{y_0 - 2y_1 + y_2}{h^2} + y_1 = \frac{y_0}{h^2} - \frac{2y_1}{h^2} + y_1 + \frac{y_2}{h^2}$$

Stąd otrzymujemy:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ 1/h^2 & -2/h^2+1 & 1/h^2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Obliczamy drugi wiersz macierzy:

$$\frac{y_1 - 2y_2 + y_3}{h^2} + y_2 = \frac{y_1}{h^2} - \frac{2y_2}{h^2} + y_2 + \frac{y_3}{h^2}$$

Stąd otrzymujemy

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ 1/h^2 & -2/h^2+1 & 1/h^2 & 0 & \cdots \\ 0 & 1/h^2 & -2/h^2+1 & 1/h^2 & 0 \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Wartości dla kolejnych n do N-1, będziemy wyznaczać w sposób analogiczny jak powyżej. Pamiętamy o tym, że dla ostatniego wiersza będziemy mieli inną implementację podaną we wzorze, stąd otrzymamy poniższą macierz układu równań A.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{h^2} & \frac{-2}{h^2}+1 & \frac{1}{h^2} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{h^2} & \frac{-2}{h^2}+1 & \frac{1}{h^2} & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & \vdots & \ddots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{h^2} & \frac{-2}{h^2}+1 & \frac{1}{h^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{h^2} & \frac{-2}{h^2}+1 & \frac{1}{h^2} \\ -3 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Nie zapominam o wspomnieniu o wektorach y oraz b, które będą stanowić integralną część naszego zadania.

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix}$$

$$b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Teraz korzystając ze wskazówki, utworzyłem nową macierz w programie nazwaną również macierzą A, podaną poniżej. Macierz ta jako wiersze będzie przyjmować dwa pierwsze wiersze oraz ostatni wiersz macierzy A układu równań. Takim zabiegiem otrzymamy układ równań 3 na 3, gdzie prostym będzie wyznaczenie wartości dla y1 oraz y2.

W programie operuje na wektorach, stąd podana powyżej macierz A układu równań, będzie reprezentowana za pomocą wektorów, stąd w opracowaniu występuje kolizja oznaczeń z macierzą 3 na 3, ale w samym programie nie.

Do utworzenia oraz rozwiązania macierzy 3 na 3 korzystam z funkcji tworzenieMacierzy3na3 oraz rozwiazanieUkladuRownan3na3.

Macierz A układu 3 na 3 będzie wyglądać w sposób następujący.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1/h^2 & -2/h^2+1 & 1/h^2 \\ -3 & 4 & 1 \end{bmatrix}$$

W programie utworzyłem na potrzebę rozwiązania macierzy 3 na 3, dwa wektory o rozmiarach 3, które będą reprezentowały nasze 3 pierwsze wartości dla y, nazwałem ten wektor yTemp oraz wektor b, który będzie reprezentował 2 pierwsze elementy oraz N-ty element wektora b, nazwałem ten wektor bTemp.

$$yTemp = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$bTemp = \begin{bmatrix} 1 & & b_0 \\ 0 & & b_1 \\ 0 & & b_N \end{bmatrix}$$

Następnie z podanej powyżej macierzy 3 na 3 wyliczam wartości y1 oraz y2, które wykorzystam w funkcji rozwiazanieOstateczne, do wyliczenia w sposób rekurencyjny pozostałych wartości wektora y.

```
// Funkcja w celu w celu rozwiazania wartosci zmiennych wektora y

void rozwiazanieOstateczne(const gsl_vector *diag, const gsl_vector *e, const gsl_vector *f, const gsl_vector *b, const gsl_vector *yTemp, gsl_vector *y){

    // Wyluskanie wartosci y z rozwiazania ukladu 3 na 3

    double wartoscYTemp1 = gsl_vector_get(yTemp, 1);
    double wartoscYTemp2 = gsl_vector_get(yTemp, 2);

    // Przypisanie wartosci z rozwiazania ukladu 3 na 3 do glownego rozwiazania

    gsl_vector_set(y, 0, gsl_vector_get(yTemp, 0));
    gsl_vector_set(y, 1, wartoscYTemp1);
    gsl_vector_set(y, 2, wartoscYTemp2);

    // Utworzenie zmiennych pomocniczych w celu rozwiazania rekurencyjnego

    double zmienna1, zmienna2, zmienna3, wynik;

    for(int i = 2; i < N; i++){
        zmienna1 = gsl_vector_get(e, i - 1);
        zmienna2 = gsl_vector_get(diag, i);
        zmienna3 = gsl_vector_get(f, i - 1);
        wynik = (gsl_vector_get(b, i) - zmienna1 * wartoscYTemp1 - zmienna2 * wartoscYTemp2) / zmienna3;
        gsl_vector_set(y, i + 1, wynik);
        wartoscYTemp1 = wartoscYTemp2;
        wartoscYTemp2 = wynik;
    }

}
```