Metody numeryczne - N3

DAMIAN PORADYŁO

N3 *Zadanie numeryczne*

Rozwiązać układ $(N+1)\ x\ (N+1)$ równań postaci:

$$\begin{cases} y_0 & = 1 \\ (D_2 y)_n + y_n & = 0, \\ y_{N-1} - 2y_N + y_0 & = 0 \end{cases} \qquad n = 1 \dots (N-1)$$

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

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

Rozwiązanie przedstawić w graficznie (nh, y_n).

1 OPIS METODY

Układ równań po przekształceniu do macierzy przedstawia się następująco:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & h^2-2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & h^2-2 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & h^2-2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 & h^2-2 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Jak możemy zauważyć, macierz ta jest lekko zaburzoną macierzą trójdiagonalną. Problemem jest tu element [0][N] w którym zamiast 0 znajduje się 1. Taki układ idealnie nadaje się na zastosowanie wzoru **Shermana-Morrisona**.

Wzór Shermana-Morrsiona (niech $A_1 = A + uv^T$):

$$A_1{}^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

Jak łatwo się domyślić, nie będziemy stosować tego wzoru bezpośrednio do obliczenia $A_1{}^{-1}$.

W tym przypadku skupimy się tak naprawdę na rozwiązaniu dwóch poniższych równań (z macierzą trójdiagonalną). Robimy to za pomocą **Metody Thomasa.**

$$Az = b \qquad \text{z czego wynika} \qquad z = A^{-1}b$$

$$Aq = u \qquad \text{z czego wynika} \qquad q = A^{-1}u$$

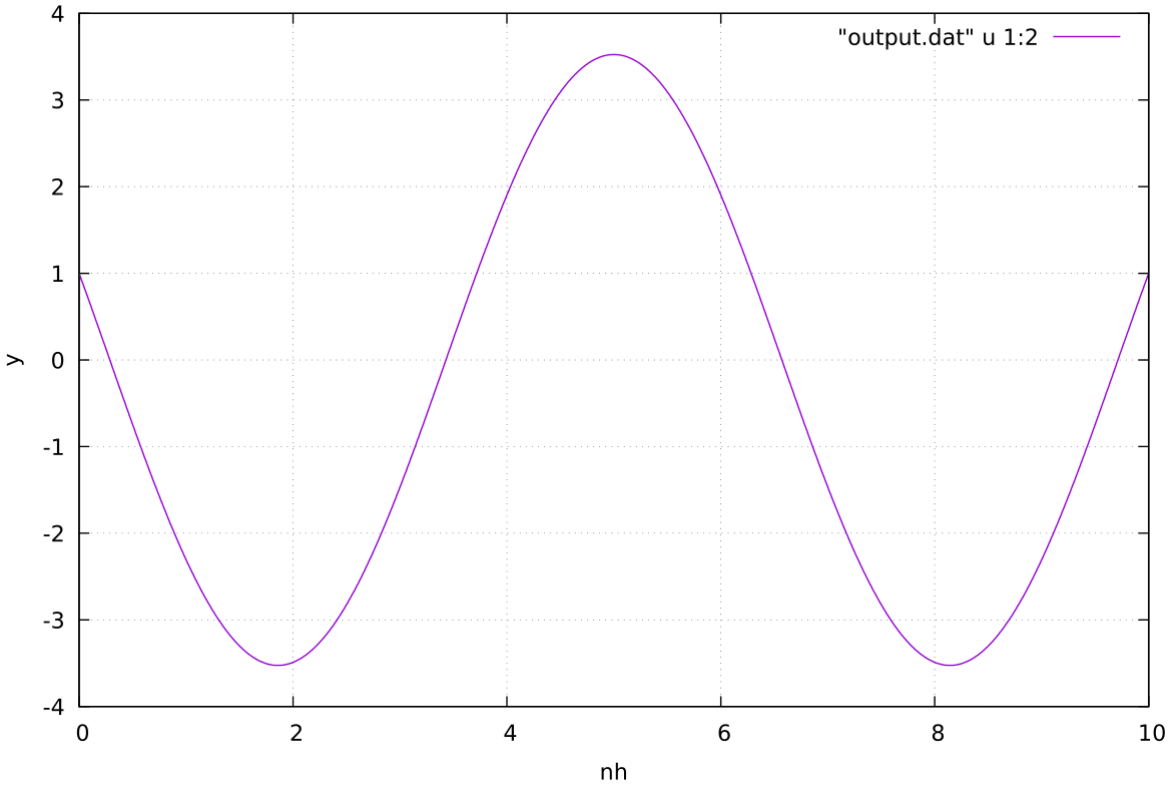
Teraz rozwiązanie przyjmuje postać:

$$w = z - \frac{v^Tz}{1 + v^Tq}q$$

Gdzie: v^Tz oraz $1 + v^Tq$ są jakimiś liczbami a natomiast z i q to wektory.

Złożoność obliczeniowa w tym przypadku wynosi $O(n)$

2 WYKRES



Wykres 1 Wykres zależności y od nh

3 WYNIKI

Wyniki na podstawie których został wygenerowany wykres znajdują się w paczce z programem w pliku pod nazwą: „**output.dat**”

```cpp
#include <iostream>
#include <cstdio>

#define N 1000
#define h 0.01
#define h_square h*h

void initMatrix(double *a, double *b, double *c, double *f) {
    b[0] = 1.0;
    c[0] = 0;
    f[0] = 1.0;
    for(int i=0; i<N; i++) {
        a[i] = 1.0;
        if(i>0) {
            b[i] = (h_square - 2.0);
            c[i] = 1;
            f[i] = 0;
        }
    }
    c[N-1] = 1.0;
    b[N] = -2.0;
}
void metodaThomasa(double *a, double *b, double *c, double *u, double *f) {
    for(int i = 1; i < N; i++){
        double m = a[i] / b[i-1];
        a[i] = 0.0;
        b[i] -= m*c[i-1];
        f[i] -= m*f[i-1];
    }
    u[N] = f[N]/b[N];
    for (int k = N; k >= 0 ; k--) {
        u[k] = (f[k] - c[k]*u[k+1])/b[k];
    }
}

void solve(double *wynik, double *u, double *z, double *v) {
    double vUp = 0;
    double zDown = 0;

    for(int i = 0; i<N+1; i++){
        vUp += v[i]*u[i];
        zDown += v[i]*z[i];
    }
    zDown += 1.0;
    for(int i=0; i<N+1; i++) {
          wynik[i] = u[i] - (vUp/zDown) * z[i];
    }
}

void printArray(double *array) {
    for (int i = 0; i <= N; i++) {
         printf("%.10f %.10f \n",  (i)*h, array[i]);
    }
}

int main() {
    double a[N], b[N+1], c[N], f[N+1], u[N+1], z[N+1], wynik[N+1];
    double u_V[N+1] = {0};
    double V[N] = {0};
    u_V[N] = 1.0;
    V[0] = 1.0;

    initMatrix(a,b,c,f);
    metodaThomasa(a,b,c,u,f);
    metodaThomasa(a,b,c,z,u_V);
    solve(wynik, u, z, V);
    printArray(wynik);


}
```

5 URUCHOMIENIE

>> make run

W skład zestawu wchodzą:

- Plik programu
- Wygenerowany wykres
- Plik zawierający otrzymane wyniki
- Opracowanie
- Makefile