Metody numeryczne - N2

DAMIAN PORADYŁO

N2 *Zadanie numeryczne*

Rozwiązać układ z poprzedniego zadania (zestaw 1 zad 4,5) poprawnie (optymalnie) implementując algorytm do rozwiązywania układu z macierzą trójdiagonalną (może być algorytm Thomasa). W zadaniu proszę przyjąć $N = 1000$. Wynik przedstawić w postaci graficznej (wykres: oś x: nh oś y wartości rozwiązania y_n).

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & a_3 & b_3 & c_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}$$

$$d_0 = b_0, \qquad d_n = b_n - l_n c_{n-1}, \qquad l_n = \frac{a_n}{d_{n-1}},$$

$$z_0 = f_0, \qquad z_n = f_n - l_n z_{n-1},$$

$$y_N = \frac{z_n}{d_N}, \qquad y_n = \frac{z_n - c_n y_{n+1}}{d_n},$$

$$b_0 = b_N = f_N = 1, \qquad c_0 = a_N = 0, \qquad f_{0:N-1} = 0,$$

$$b_n = -\frac{2}{h^2}, \qquad a_n = c_n = \frac{1}{h^2}, \qquad h = \frac{1}{N}$$

1 OPIS METODY

Metoda Thomasa idealnie nadaje się do rozwiązywania równań trójprzekątniowych. W odróżnieniu do np. ***eliminacji Gaussa*** jest znacznie szybsza. Jej złożoność jest rzędu LINIOWEGO, wynosi **O(N)**.

Mając macierz trójprzekątniową(jak w treści zadania):

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & a_3 & b_3 & c_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}$$

Możemy zapisać jako: $a_i y_{i-1} + b_i y_i + c_i y_{i+1} = f_i$

Gdzie: $a_1 = c_n = 0$ a także $i = 1, 2, 3, 4, \dots \dots N$

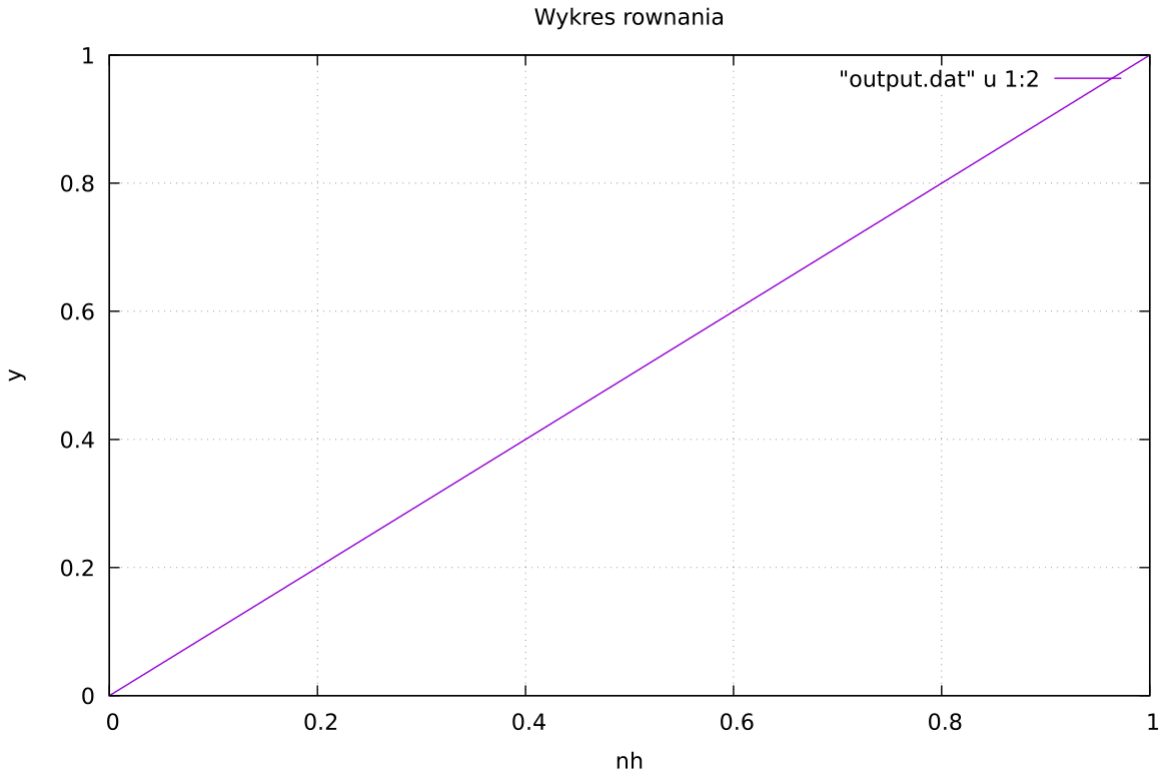
Algorytm Thomasa sprowadza się do wyzerowania współczynników $a_i: i = 2, 3, 4 \dots N$, tak aby ostatnie równanie było postaci $b_N y_N = f_N$, co pozwoli wyznaczyć y_N. Mając dane y_N można wyznaczyć y_{N-1}, oczywiście przy założeniu $a_{N-1} = 0$, tak postępując można rozwiązać cały układ równań.

Na co warto zwrócić uwagę to również fakt, że tak macierz takiego układu równań może być przechowywana w pamięci komputera przy pomocy trzech tablic jednowymiarowych, co jak wiemy dodatkowo zmniejsza zużycie pamięci.

2 WYKRES

Wykres stworzony w programie **gnuplot** w celu zwizualizowania wyników. Znajduje się on również w osobnym pliku w katalogu z programem pod nazwą **wykres.pdf**

Dla **N = 1000**



Rysunek 1 Wykres

3 WYNIKI

Wyniki przedstawiające wynik programu znajdują się w pliku „dane.dat”. W pierwszej kolumnie są wartości **nh** w drugiej natomiast wyniki **y**.

4 KOD PROGRAMU

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

#define N 1000
double h = 1.0/N;

void initMatrix(double *c, double *b, double *a, double *f) {
    for(int i=0; i<=N; i++) {
        if(i>=1) a[i] = c[i] = 1.0/(h*h);
        if(i<N)  c[i] = 1.0/(h*h);
        b[i] = -(2.0/(h*h));
        f[i] = 0;
    }
    b[0] = b[N] = 1;
    c[0] = a[N] = 0;
    f[N] = 1;
}
void solve(double *a, double *b, double *c, double *y, double *f) {
    for(int i=1; i<=N; i++) {
        double l = a[i] / b[i-1];
        a[i] = 0;
        b[i] -= l * c[i-1];
        f[i] -= l * f[i-1];
    }
    y[N] = f[N] / b[N];
    for(int i=N-1; i>=0; i--) {
        y[i] = (f[i] - c[i] * y[i+1])/b[i];
    }
}
void printResult(double *y) {
    for(int n=0; n<=N; n++) printf("%.10lf\t%.10lf\n", n*h, y[n]);
}

int main(int argc, char const *argv[]) {
    double b[N+1], c[N], a[N+1], y[N+1], f[N+1];

    initMatrix(c, b, a, f);
    solve(a,b,c,y,f);
    printResult(y);

    return 0;
}
```

Kod 1 Kod programu

5 URUCHOMIENIE

>> make run

W skład zestawu wchodzą:

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