Sprawozdanie – Laboratorium nr 2

Iteracyjne rozwiązywanie układu równań liniowych metodą Jacobiego

Mikołaj Marchewa, 25 marca 2020

1. Wstęp teoretyczny

Do wykonania zadań z laboratorium wykorzystaliśmy metodę Jacobiego, która umożliwia w sposób iteracyjny rozwiązać układy równań typu $A \cdot x = b$. Metoda ta wykorzystuje wektor rozwiązań x i w kolejnych iteracjach przybliża wynik, aż do uzyskania zadanej dokładności φ, która liczona jest z następującego wzoru:

$$\varphi = |s_n - s_s|, \tag{1}$$

gdzie:

$s_n = \sum_i (x_n[i])^2$,

$s_s = \sum_i (x_s[i])^2$,

x_n - przybliżony wynik z *i*-tej iteracji,

x_s - przybliżony wynik z *i-1* iteracji.

Aby obliczyć kolejne przybliżenia wyniku korzystamy z rozkładu macierzy *A* na trzy części: macierz trójkątną dolną, diagonalną oraz trójkątną górną - oznaczane kolejno: *L, D, U.* Uzyskujemy więc: $A = L + D + U$, a metoda Jacobiego tworzy kolejne wektory wyników ze wzoru:

$$x_{k+1} = D^{-1} \cdot (b - (L + U) \cdot x_k). \tag{2}$$

2. Zadanie do wykonania

2.1 Opis problemu

Zadanym problemem tych ćwiczeń laboratoryjnych było rozwiązanie równania różniczkowego opisującego ruch ciała pod wpływem siły sprężystości ($-\omega^2 x$), z uwzględnieniem siły tarcia ($-\beta V$) oraz danej siły wymuszenia ($F_0 \cdot \sin(\Omega t)$):

$$\frac{d^2x}{dt^2} = -\omega^2 x - \beta V + F_0 \cdot \sin(\Omega t). \tag{3}$$

Powyższe równanie na potrzeby obliczeń musimy przekształcić do postaci pozwalającej na uzależnienie wychylenia od kolejnych chwil czasowych:

$$x(t) = x \cdot t_i, \quad t_i = h \cdot i, \tag{4}$$

gdzie *h* to zadany krok czasowy.

Zastępujemy drugą pochodną wychylenia po czasie, a także prędkość ciała (pierwszą pochodną położenia po czasie) ilorazem różnicowym. Po podstawieniu do równania (3) i uporządkowaniu jego wyrazów uzyskaliśmy poniższy wzór:

$$a_1 x_{i-1} + a_2 x_i + a_3 x_{i+1} = b_i, \tag{5}$$

gdzie:

$$a_1 = 1,$$

$$a_2 = \omega^2 \cdot h^2 - 2 - \beta \cdot h,$$

$$a_3 = 1 + \beta \cdot h,$$

$$b_i = F_0 \cdot \sin(\Omega \cdot h \cdot i) \cdot h^2.$$

Równanie (5), po przyjęciu odpowiednich warunków początkowych $(x(t=0) = 1, \mathrm{V(t=0)} = 0)$, zapisujemy w postaci równania macierzowego $A \cdot x = b$, gdzie:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & a_1 & a_2 & a_3 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & a_1 & a_2 & a_3 \end{pmatrix}, \quad x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \\ F_0 \cdot \sin(\Omega \cdot t_1) \cdot h^2 \\ F_0 \cdot \sin(\Omega \cdot t_2) \cdot h^2 \\ F_0 \cdot \sin(\Omega \cdot t_3) \cdot h^2 \\ F_0 \cdot \sin(\Omega \cdot t_4) \cdot h^2 \\ \vdots \\ F_0 \cdot \sin(\Omega \cdot t_{n-3}) \cdot h^2 \\ F_0 \cdot \sin(\Omega \cdot t_{n-2}) \cdot h^2 \\ F_0 \cdot \sin(\Omega \cdot t_{n-1}) \cdot h^2 \end{pmatrix}.$$

Ponieważ *A* jest macierzą trójprzekątnikową możemy zapisać ją przy użyciu jedynie trzech wektorów:

$$d_0 = [1, 1, a_3, a_3, \ldots, a_3],$$

$$d_1 = [0, -1, a_2, a_2, \ldots, a_2],$$

$$d_2 = [0, 0, a_1, a_1, \ldots, a_1].$$

Teraz dysponując metodą Jacobiego rozwiązywania układów równań liniowych, wyznaczamy kolejne przybliżenie wyniku korzystając z przekształconego wzoru (2):

$$x_n[i] = \frac{1}{d_0[i]} \cdot (b[i] - (d_1[i] \cdot x_s[i-1] + d_1[i] \cdot x_s[i-2])), \tag{6}$$

aż do uzyskania założonej dokładności.

Parametry zadania: $V_0 = 0, x_0 = 1, \omega = 1, \mathrm{n} = 2000, \mathrm{h} = 0.02$ oraz szukana dokładność wyniku: $\varphi = 10^{-6}$. Rozwiązania znaleźliśmy następujących dla trzech przypadków:

1) $\beta = 0.0, F_0 = 0.0,\ \Omega = 0.8,$
2) $\beta = 0.4, F_0 = 0.0,\ \Omega = 0.8,$
3) $\beta = 0.4, F_0 = 0.1,\ \Omega = 0.8.$

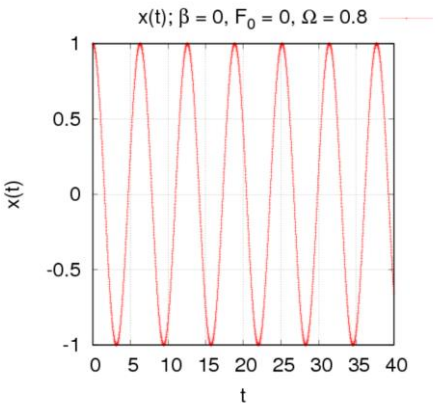
2.2 Wyniki

Do uzyskania poniższych rezultatów użyłem uprzednio napisanego programu w języku **C,** a także programu *Gnuplot,* generującego wykresy. Do spełnienia warunku dokładności, dla każdego z przypadków wykonała się taka sama liczba iteracji (Rys.1).

```
8marchewa@taurus:~/MN/lab03$ ./main
iter number = 2002
```

Rys.1 Liczba iteracji programu do spełnienia warunku dokładności

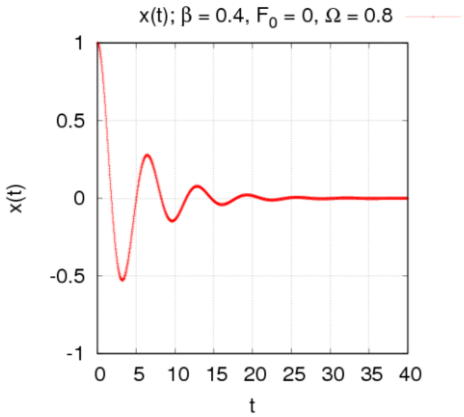
1)



Rys.2 Wykres zależności wychylenia ciała od czasu dla danych 1)

Wykres z *Rys.2* przedstawia jak w czasie zmieniało się wychylenie ciała. Rezultat pokrywa się z oczekiwanym – brak tłumienia, brak siły wymuszającej.

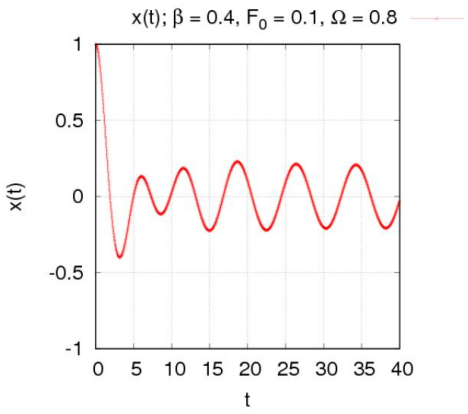
2)



Rys.3 Wykres zależności wychylenia ciała od czasu dla danych 2)

Na wykresie z *Rys.3* oczekiwaliśmy regularnego zmniejszania się wychylenia ciała w czasie – zgodnie z wprowadzonymi danymi. Rezultat jest więc zadowalający.

3)



Rys.4 Wykres zależności wychylenia ciała od czasu dla danych 3)

Ostatni przypadek to tłumienie oraz dodanie siły wymuszającej do siły sprężystości działającej na ciało. Tu również wynik pokrywa się z założeniami.

3. Wnioski

Metoda Jacobiego, ze względu na swoją prostotę, świetnie nadaje się do implementacji w programach komputerowych. Jednakże jej sporym minusem jest konieczność przechowywania w pamięci komputera dwóch wektorów wyników – pochodzącego z *i*-tej oraz *i - 1* iteracji. Taki sposób jest więc mało efektywny.

Rozwiązaniem tego problemu jest modyfikacja wzoru (6) na uzyskiwanie kolejnych wektorów wyników tak, by uzależnić kolejne wartości wektora wyników x_{i+1} od tych uzyskanych w poprzednich iteracjach x_i oraz x_{i-1}. Ta szybka korekta programu skutkuje otrzymanie zadanej dokładności wyniku już w drugiej iteracji obliczeń. Jest więc to nieporównywalnie efektywniejszy rezultat od metody Jacobiego. Dodatkowo zmniejsza się liczba przechowywanych wektorów w pamięci o jeden.

Tak zmodyfikowana metoda Jacobiego nosi nazwę metody ***Gaussa–Seidela.***