

# **Sprawozdanie 3**

**Metody Numeryczne - Laboratoria**

## **Temat laboratoriów:**

**Iteracyjne rozwiązywanie układu równań liniowych  
metodą Jakobiego**



**WFIS AGH**

**25.03.2021**

**Łukasz Wajda**

## 1. Wstęp teoretyczny

Kolejną nowo poznaną metodą rozwiązywania układów równań linowych jest metoda Jakobiego, która pozwala w sposób iteracyjny rozwiązać układy równań typu  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ . Aby jej użyć posłużyliśmy się równaniem różniczkowym ruchu ciała:

$$\frac{d^2x}{dt^2} = -\omega^2x - \beta V + F_0 \sin(\Omega t), \quad (1)$$

poddanego działaniu siły sprężystej  $(-\omega^2x)$ , siły tarcia  $(-\beta V)$  zależnej od prędkości oraz siły wymuszającej ruch  $(F_0 \sin(\Omega t))$ .

Po wprowadzeniu siatki, której węzłami są chwile czasowe:

$$x(t) = x \cdot t_i, \quad t_i = h \cdot i, \quad (2)$$

gdzie  $h$  to krok czasowy, oraz przejściu z pochodnych na ilorazy różnicowe otrzymaliśmy równanie (3).

$$x_{i-1} - 2x_i + x_{i+1} + \omega^2 h^2 x_i + \beta h(x_{i+1} - x_i) = F_0 \sin(\Omega h i) h^2. \quad (3)$$

Aby zapisać równanie w postaci przypominającej funkcję liniową przyjęliśmy,  $a_1 = 1$ ,  $a_2 = \omega h^2 - 2 + \beta h$ ,  $a_3 = 1 + \beta h$ ,  $b_i = F_0 \sin(\Omega h i) h^2$ , wtedy równanie otrzymało postać:

$$a_1 x_{i-1} + a_2 x_i + a_3 x_{i+1} = b_i. \quad (4)$$

Określamy warunki początkowe:  $x_0 = 1$ ,  $V_0 = 0$  i zapisujemy układ w postaci równania macierzowego  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & \dots & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_n \end{bmatrix} \quad (5)$$

## 2. Zadanie

Zadanie polegało na wyznaczeniu wektora wynikowego  $\mathbf{x}$ , jego kolejnych przybliżeń posługując się metodą Jakobiego. Początkowo korzystamy z rozkładu macierzy  $\mathbf{A}$  na trzy części: macierz trójkątną dolną ( $\mathbf{L}$ ), diagonalną ( $\mathbf{D}$ ) oraz trójkątną górną ( $\mathbf{U}$ ). Uzyskujemy więc:  $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ , a metoda Jacobiego tworzy kolejne wektory wyników ze wzoru:

$$x_{i+1} = \mathbf{D}^{-1} \cdot (\mathbf{b} - (\mathbf{L} + \mathbf{U}) \cdot \mathbf{x}_i). \quad (6)$$

Powyższy wzór w zadaniu reprezentuje się w następującej postaci:

$$x_n[i] = \frac{1}{d_0[i]} (b[i] - d_1[i]x_s[i-1] - d_2[i]x_s[i-2]), \quad i = 0, 1, 2, \dots, n. \quad (7)$$

Wektory n-elementowe  $d_0, d_1, d_2$  to przekątne macierzy:

$$d_0 = [1, 1, a_3, a_3, \dots, a_3], \quad (8)$$

$$d_1 = [0, -1, a_2, a_2, \dots, a_2], \quad (9)$$

$$d_2 = [0, 0, a_1, a_1, \dots, a_1]. \quad (10)$$

Określono również parametry:  $\omega = 1$ ,  $n = 2000$  i  $h = 0.02$ . Obliczanie kolejnych przybliżeń należało zakończyć aż do uzyskania zadanej dokładności (mniejszej od  $\varepsilon = 1 \cdot 10^{-6}$ ) wyrażająca się wzorem:

$$\varepsilon > |s_n - s_s|, \quad (11)$$

Gdzie:

$s_n = \sum_i (x_n[i])^2$ ,  $x_n$  – przybliżony wynik z i-tej iteracji,

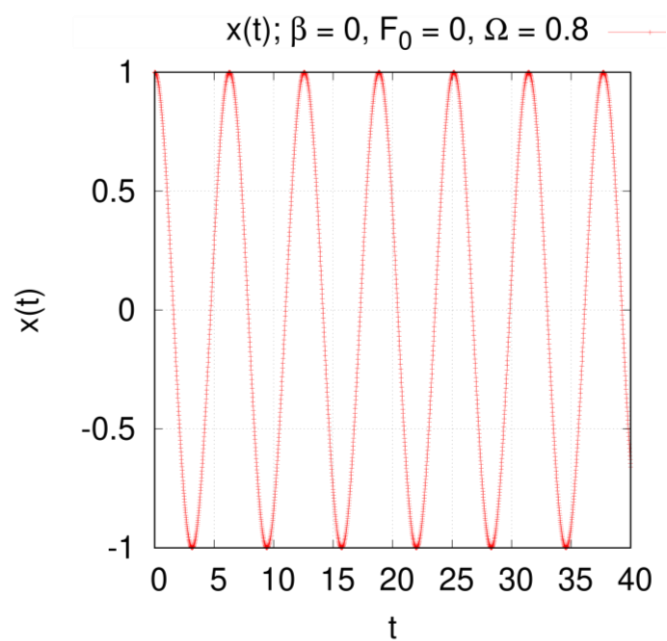
$s_s = \sum_i (x_s[i])^2$ ,  $x_s$  – przybliżony wynik z i-1 iteracji.

Warunek zapewnia, że metoda automatycznie zakończy działanie, gdy zawartość wektora wyników nie zmienia się już znacząco z iteracji na iterację.

### 3. Wyniki

W pierwszej kolejności zaimplementowaliśmy macierz trójkątną  $A$ , alokując pamięć dla sześciu wektorów zmiennoprzecinkowych:  $\vec{b}, \vec{d_0}, \vec{d_1}, \vec{d_2}, \vec{x_s}$  oraz  $\vec{x_n}$  i wypełniając je wartościami. Uruchamiamy program dla 3 różnych wartości:  $\beta$ ,  $F_0$  i  $\Omega$ . We wszystkich przypadkach metoda Jacobiego wykonała 2002 iteracje przed uzyskaniem zbieżności. Wyniki uznano za zbieżne, gdy spełniona jest nierówność (11). Poniższe wykresy uzyskano przy wykorzystaniu programu Gnuplot oraz przy następujących parametrach:

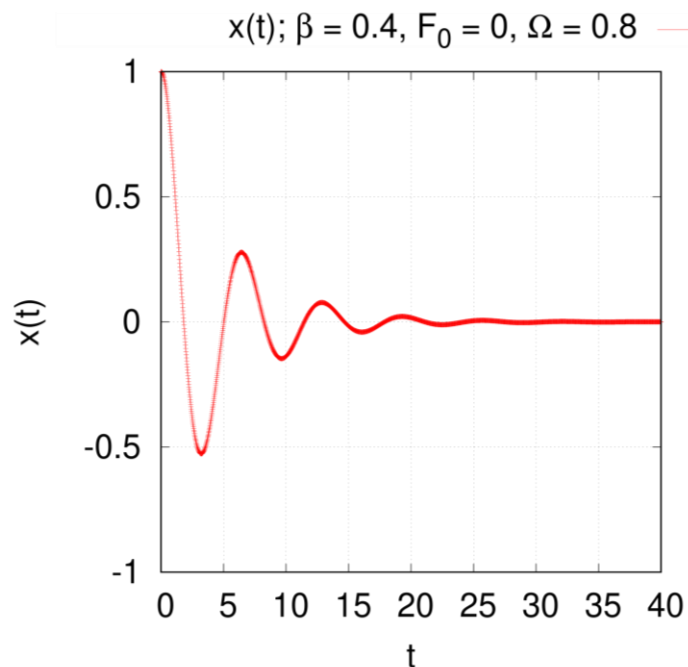
- Wariant 1  
 $\beta = 0.0, F_0 = 0.0, \Omega = 0.8$



Rysunek 1: Wykres  $x(t)$

Wykres z Rysunku 1 przedstawia jak przy braku tłumienia i braku siły wymuszającej wychylenie ciała zmienia się w czasie. Wykres reprezentuje równomierne drgania co pokrywa się z oczekiwaniami.

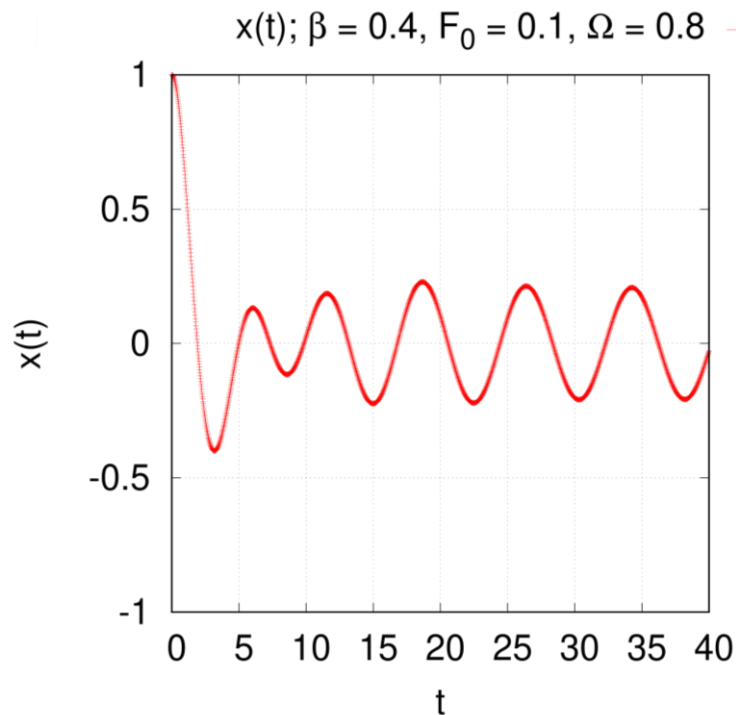
- Wariant 2  
 $\beta = 0.4$ ,  $F_0 = 0.0$ ,  $\Omega = 0.8$



Rysunek 2: Wykres  $x(t)$

Na wykresie z Rysunku 2 otrzymaliśmy regularnie zmniejszające się drgania zanikające w czasie, aż do całkowitego wygaszenia, co jest słuszne z wcześniej postawionymi przypuszczeniami (brak wymuszenia, tłumienie).

- Wariant 3  
 $\beta = 0.4, F_0 = 0.1, \Omega = 0.8$



Rysunek 3: Wykres  $x(t)$

Ostatecznie przy tłumieniu i sile wymuszającej dodanej do siły sprężystości działającej na ciało, otrzymujemy wykres dla którego początkowo następuje lekkie wygaszenie, później amplituda zwiększa się, nie osiąga jednak stanu takiego jak na samym początku, wychylenie od 0 utrzymuje się na stałym poziomie. Wyniki pokrywa się z założeniami.

## 4. Wnioski

Metoda Jakobiego do iteracyjnego rozwiązywania układów równań polega na obliczaniu kolejnych przybliżeń wyniku za pomocą iteracyjnego wzoru. Podczas rozwiązywania zadania skorzystaliśmy z faktu, że mnożona macierz jest macierzą rzadką, dzięki czemu można przechowywać ją w postaci trzech wektorów. Pozwoliło to na zaoszczędzenie używanej pamięci oraz pominięcie zbędnych obliczeń z zerami. Jednak ta metoda nie jest doskonała.

Sporym minusem jest konieczność przechowywania w pamięci dwóch wektorów wyników, pochodzącego z  $i$ -tej oraz  $i - 1$  iteracji. Aby temu zapobiec można skorzystać z metody Gaussa–Seidela. Zapewnia ona zmniejszenie się liczba przechowywanych wektorów wynikowych w pamięci o jeden. Żeby zaimplementować tą metodę w programie wystarczy zmodyfikować wzór (7) do postaci z jednym wektorem, tj.:

$$x[i] = \frac{1}{d_0[i]} (b[i] - d_1[i]x[i - 1] - d_2[i]x[i - 2]).$$

Skutkuje to otrzymaniem zadanej dokładności wyniku już w drugiej iteracji obliczeń.

Na podstawie uzyskanych wykresów i obliczeń, możemy wnioskować, że zadanie zostało prawidłowo wykonane.