Eigenwertprobleme in Matlab

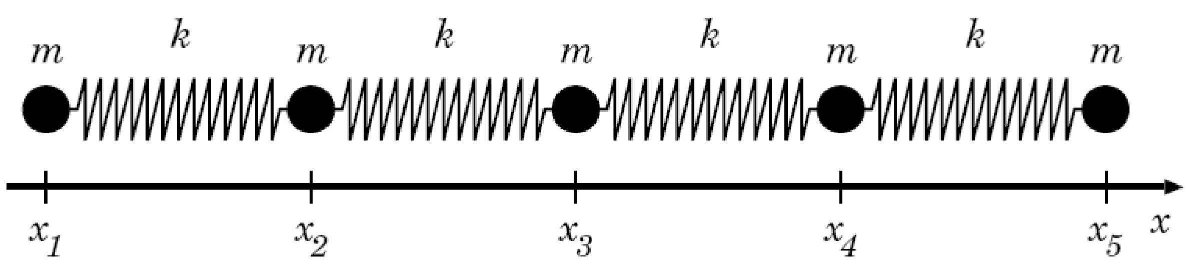
Aufgabe: Implementieren eines Eigenwertproblemes (EWP) und dessen Lösung in Matlab am Beispiel eines Systems von n gekoppelten harmonischen Oszillatoren.

Ziel: Kennenlernen des Eigenwertsolvers
Visualisierungen in Matlab
Aufstellen von Eigenwertproblemen aus Differentialgleichungen

Benötigte Funktionen:

- *[V,D]=**eig**(M)* … liefert die Eigenvektoren V und die dazugehörigen Eigenwerte in einer Diagonalmatrix D
- ***plot()***, ***diag()***, ***ones()***, aus vorangegangenen Seminaren
- ***linspace**(start,end,n)*…kreieren eines Vektors von start bis end in n linearen Schritten

Veranschaulichung:



1. Aufstellen des Eigenwertproblemes.

$$\begin{aligned}
m\ddot{x}_1 &= k(x_2 - x_1 - l)\\
m\ddot{x}_2 &= k(x_3 - x_2 - l) - k(x_2 - x_1 - l)\\
m\ddot{x}_3 &= k(x_4 - x_3 - l) - k(x_3 - x_2 - l)\\
m\ddot{x}_4 &= k(x_5 - x_4 - l) - k(x_4 - x_3 - l)\\
m\ddot{x}_5 &= -k(x_5 - x_4 - l).
\end{aligned}$$

Überführung des Differentialgleichungssystems in eine Matrixform.

$$\left.\begin{array}{l}\omega = \sqrt{k/m} \\ y_n = x_n - nl\end{array}\right\} \begin{pmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \\ \ddot{y}_4 \\ \ddot{y}_5 \end{pmatrix} = -\omega^2 \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$

Gesucht wird die Lösung in Form von harmonischen Funktionen. (analog der Lösung für den einzelnen h.O.)

$$\vec{y}(t) = (A \cos \Omega t + B \sin \Omega t)\vec{e}$$

Damit lässt sich das Problem auf eine Matrixoperation reduzieren.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \vec{e} =: \mathcal{M}\vec{e} = \frac{\Omega^2}{\omega^2}\vec{e},$$

Dabei entspricht $^{\Omega^2}/_{\omega^2}$ dem Eigenwert λ mit dem dazugehörigen Eigenvektor $\vec{e}$ **.**
Diese Diagonalform der Matrix M soll in Matlab implementiert werden. Dafür wird die Eigenwertroutine ***eig()*** verwendet.

2. Algorithmus initialisieren

- Kreieren einer *nxn* Matrix mit -1 auf den beiden Nebendiagonalen und 2 auf der Hauptdiagonale bis auf das (1,1) und das (n,n)-te Matrixelement (=1)
- Lösen des Eigenwertproblems unter Ausgabe der Eigenwerte und des Eigenvektors
- Erweiterung des Programmes zur Visualisierung der Eigenzustände des Systems