MATLAB Tutorium

Inhalt der Vorlesung

Teil I

- Modellierung eines Tragwerks
- Lösung von linearen ODEs

Teil II

- Shift-Strategien
- Deflationsalgorithmus
- Berechnung der Eigenvektoren mittels QR-Iteration

MATLAB Tutorium

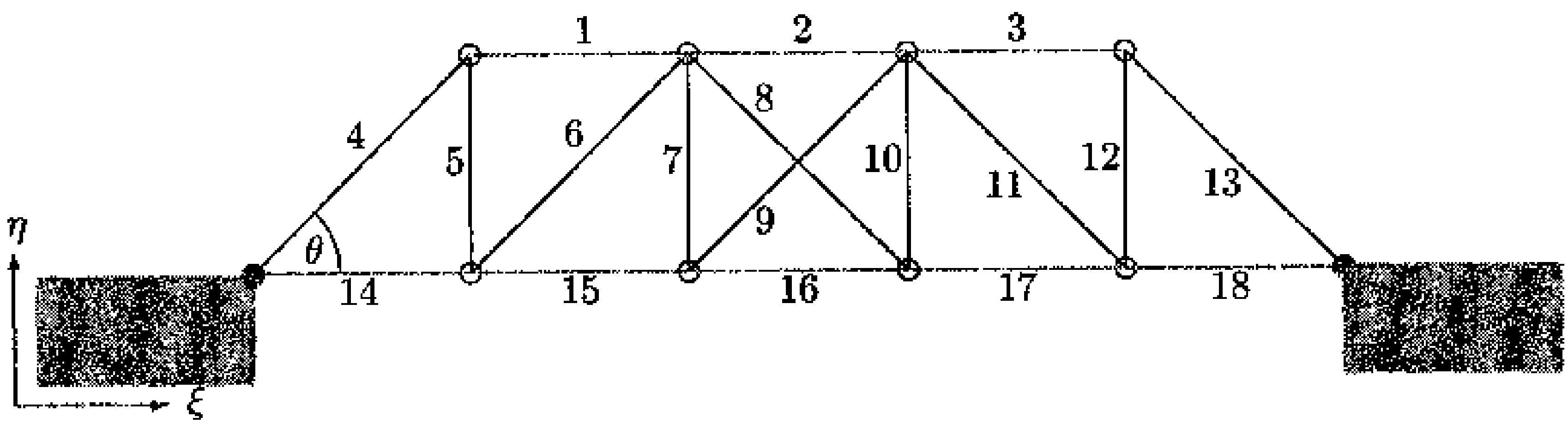


Figure 1: Tragwerk

MATLAB Tutorium

Bei der Modellierung von Stabwerken, entstehen folgende Systeme:

$$p = Kx$$

mit $K = B^T EB$, Verschiebungsvektor x und Lastvektor p und Diagonalmatrix E mit den Materialeigenschaften.
Betrachtet man nun das dynamische System:

$$M\ddot{x}(t) = -Kx(t)$$

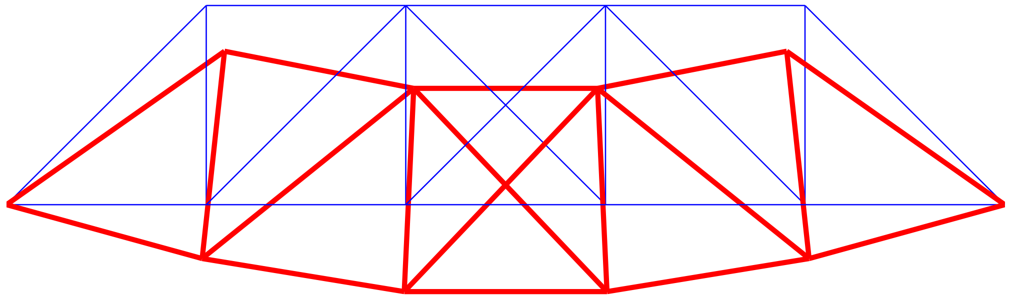
Der Ansatz

$$x(t) = e^{\lambda t} v$$

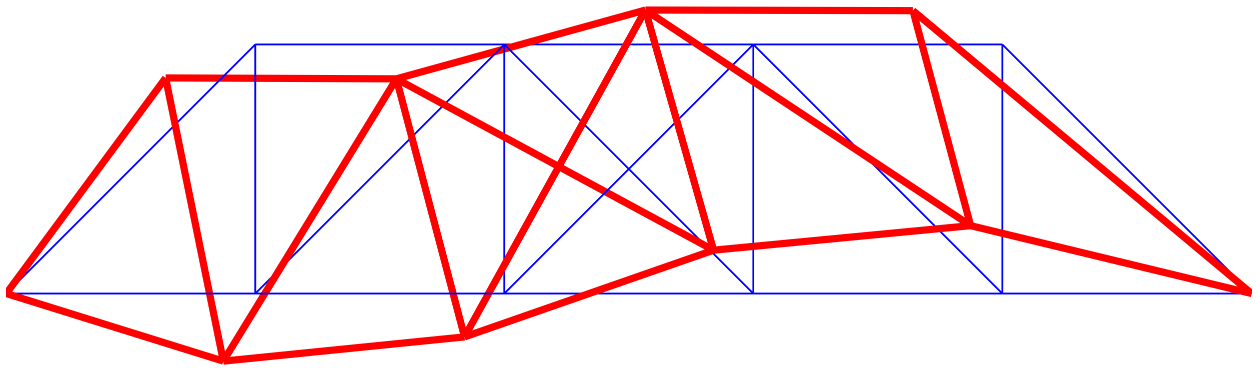
führt auf ein Eigenwertproblem der Form: $-\lambda^2 w = Aw$.

MATLAB Tutorium

1. Eigenschwingung: $-\lambda^2=0.35986$

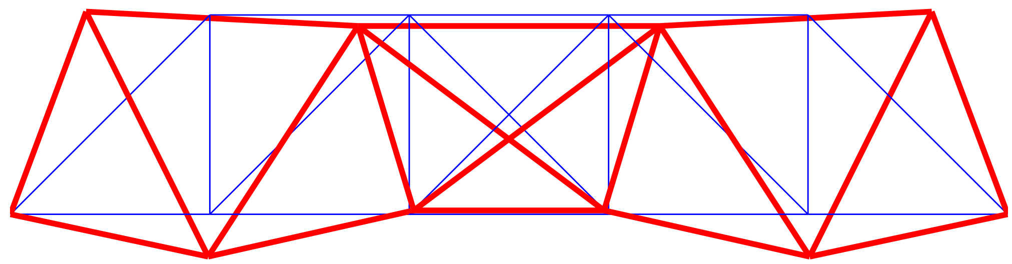


3. Eigenschwingung: $-\lambda^2 = 2.8027$

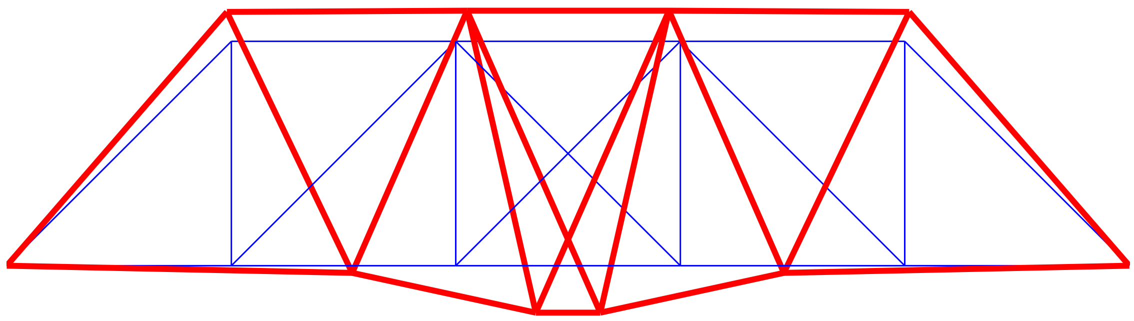


MATLAB Tutorium

6. Eigenschwingung: $-\lambda^2 = 9.2956$



8. Eigenschwingung: $-\lambda^2$=15.4855



MATLAB Tutorium

Shift Strategien

- Algorithmen zur numerischen Approximation von einzelnen oder mehreren Eigenwerten, welche einen Shift benötigen:

  a) **inverse Vektoriteration**

  b) **QR-Iteration mit festem Shift**

  c) **QR-Iteration mit Deflation**

- Problem:
  Wie kann der Shift gewählt werden, um eine möglichst gute Konvergenz zu erzwingen?

MATLAB Tutorium

Inverse Vektoriteration

- **Inverse Vektoriteration:**

  $x^{(0)}$ mit $\|x^{(0)}\|_2 = 1$
  μ Schätzung des gesuchten Eigenwerts
  $\lambda^{(0)} = \mu$
  for $k = 0, 1, \ldots$ **do**
  Löse $(A - \mu Id)y^{(k+1)} = x^{(k)}$
  $x^{(k+1)} := \frac{y^{(k+1)}}{\|y^{(k+1)}\|_2}$
  $\lambda^{(k)} := \frac{1}{\left(x^{(k)}\right)^T y^{(k+1)}} + \mu$
  end for

- Je näher die Schätzung μ bei dem gesuchten Eigenwert λ liegt, desto schneller konvergiert die inverse Vektoriteration.

MATLAB Tutorium

A priori Abschätzungen

- $A \in \mathbb{C}^{n\times n}$, dann gilt: $|\lambda| \leq \|A\|$, $\forall \lambda \in \sigma(A)$, $\|\cdot\|$ ist eine konsistente Matrixnorm

- Gerschgorin-Kreise:

  $$\sigma(A) \subset \mathcal{S}_R,\ \mathcal{S}_R = \bigcup_{i=1}^{n} R_i, \quad R_i = \left\{ z \in \mathbb{C} \,\middle|\, |z - a_{ii}| \leq \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}| \right\}$$

- Wegen $\sigma(A) = \sigma\left(A^T\right)$ gilt:

  $$\sigma(A) \subset \mathcal{S}_C,\ \mathcal{S}_C = \bigcup_{j=1}^{n} C_j, \quad C_j = \left\{ z \in \mathbb{C} \,\middle|\, |z - a_{jj}| \leq \sum_{\substack{i=1\\ j\neq i}}^{n} |a_{ij}| \right\}$$

MATLAB Tutorium

Gerschgorin-Theoreme:

- **Erstes Gerschgorin-Theorem:** Für eine gegebene Matrix $A \in \mathbb{C}^{n \times n}$ gilt: $\forall \lambda \in \sigma(A)$, $\lambda \in \mathcal{S}_R \bigcap \mathcal{S}_C$.

- **Zweites Gerschgorin-Theorem:** Seien

$$\mathcal{S}_1 = \bigcup_{i=1}^{m} R_i, \quad \mathcal{S}_2 = \bigcup_{i=m+1}^{n} R_i.$$

  Ist $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, so enthält $\mathcal{S}_1$ genau m Eigenwerte von A, jeder entsprechen seiner algebraischen Vielfachheit gezählt, während die restlichen Eigenwerte in $\mathcal{S}_2$ enthalten sind.

- **Drittes Gerschgorin-Theorem:** Sei $A \in \mathbb{C}^{n \times n}$ eine irreduzible Matrix. Ein Eigenwert $\lambda \in \sigma(A)$ kann nicht auf dem Rand von $\mathcal{S}_R$ liegen, es sei denn er liegt auf dem Rand eines jeden Kreises R_i, für $i = 1, \ldots, n$.

MATLAB Tutorium

Beispiel:

$$A = \begin{pmatrix} 30 & 1 & 2 & 3 \\ 4 & 15 & -4 & -2 \\ -1 & 0 & 3 & 5 \\ -3 & 5 & 0 & -1 \end{pmatrix}$$

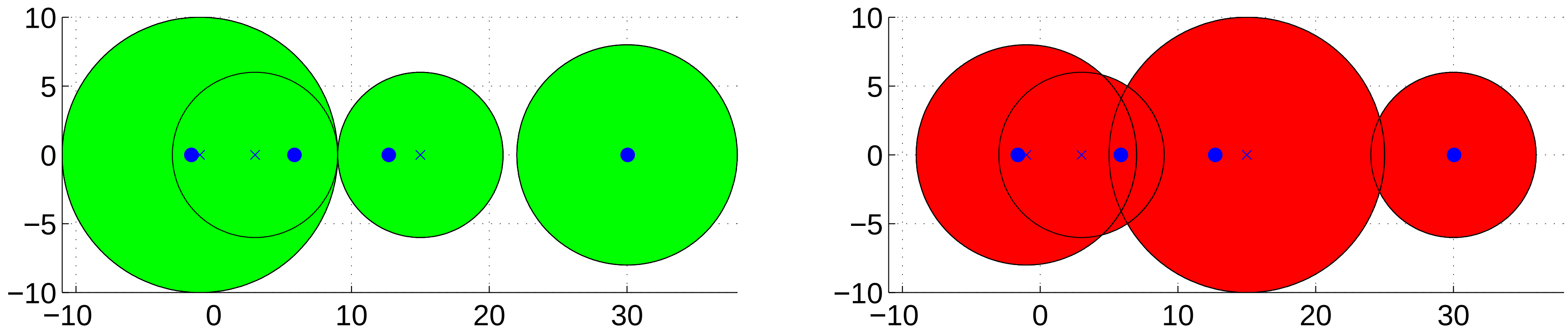


Figure 2: Spaltenkreise (links), Zeilenkreise (rechts), blaue Punkte markieren die Eigenwerte von A, die Kreuzchen markieren die Mittelpunkte der Gerschgorin-Kreise

MATLAB Tutorium

Deflationsalgorithmus mit einfachem Shift (iterativ)

function $QRrayleigh(A)$

$A \in \mathbb{R}^{n \times n}$, A in Hessenbergform

if $n = 1$ **then**

  return $A(1,1)$

end if

$l := 0$, TOL, l_{max}, k_{max}

for $k = n;\ k > 1;\ k--$ **do**

  while $|A_l(k, k-1)| > TOL \cdot (|A_l(k,k)| + |A_l(k-1, k-1)|)$ **and** $l \leq l_{\text{max}}$ **do**

    $\mu^{(l)} := A_l(k,k)$

    $A_l(1:k, 1:k) - \mu^{(l)} Id(1:k, 1:k) =: Q_{l+1} R_{l+1}$

    $A_{l+1}(1:k, 1:k) := R_{l+1} Q_{l+1} + \mu^{(l)} Id(1:k, 1:k)$

  end while

end for

Deflationsalgorithmus mit einfachem Shift (rekursiv)

function $QRrayleigh(A)$
$A \in \mathbb{R}^{n \times n}$, $A_0 = A$, A in Hessenbergform
if $n = 1$ **then**
    return $A(1,1)$
end if
Define $l = 0$, TOL, l_{max}
while $|A_l(n, n-1)| > TOL \cdot (|A_l(n,n)| + |A_l(n-1, n-1)|)$ **and** $l \leq l_{\text{max}}$ **do**
    $\mu^{(l)} := A_l(n,n)$
    $A_l - \mu^{(l)} Id =: Q_{l+1} R_{l+1}$
    $A_{l+1} := R_{l+1} Q_{l+1} + \mu^{(l)} Id$
    $l := l + 1$
end while
if $n = 2$ **then**
    return $A_l(2,2) \cup A_l(1,1)$
else
    $A_{deflation} = A_l(1 : n-1, 1 : n-1)$
    return $A_l(n,n) \cup QRrayleigh(A_{deflation})$
end if

MATLAB Tutorium

Berechnung der Eigenvektoren

- $A \in \mathbb{R}^{n\times n}$, Schur-Zerlegung: $R = Q^H A Q$, R obere Dreiecksmatrix
- Ges: $\mathbf{x} \neq \mathbf{0}$ mit $A\mathbf{x} = \lambda\mathbf{x} \Leftrightarrow \mathbf{R}\mathbf{y} = \lambda\mathbf{y}$ mit $\mathbf{y} = \mathbf{Q^H}\mathbf{x}$
- $$R = \begin{pmatrix} R_{11} & \mathbf{v} & R_{13} \\ 0 & \lambda & \mathbf{w^T} \\ 0 & 0 & R_{33} \end{pmatrix}$$
- $\mathbf{y} = \begin{pmatrix} -\left(R_{11} - \lambda I_{k-1}\right)^{-1}\mathbf{v} \\ 1 \\ 0 \end{pmatrix}$
- $\mathbf{x} = \mathbf{Q}\mathbf{y}$