Prof. Dr. Stefan Funken
Dipl.-Math Markus Bantle
Institut für Numerische Mathematik
Universität Ulm

Übungsblatt 5

(Besprechung Mo. 26.11. 2011)

Aufgabe 19 *(Implizites Q-Theorem)*

Sei $A \in \mathbb{C}^{n\times n}$ und seien $Q = (q_1, \ldots, q_n), U = (u_1, \ldots, u_n)$ unitär, so dass $H = Q^H AQ$ und $G = U^H AU$ in Hessenbergform sind. Zeigen Sie, falls $q_1 = u_1$ und H unreduziert ist, dass gilt:

$$q_j = z_j u_j \text{ mit } |z_j| = 1 \text{ und}$$
$$|h_{j,j-1}| = |g_{j,j-1}|, \ j = 2, \ldots, n.$$

Das bedeutet, sobald die erste Spalte von Q festgelegt ist, sind alle weiteren Spalten der Matrix Q, welche A auf Hessenbergform bringt, bis auf eine Konstante vom Betrage eins eindeutig bestimmt.

Aufgabe 20 *(Matlab QR-Algorithmus)*

(a) Schreiben Sie eine Matlab-Funktion `function A = HessRed(A)`, in der die Matrix A mit Hilfe von Householder Spiegelungen auf Hessenberg Form reduziert wird. Sie dürfen dazu die Funktion `HouseholderVector.m`, welche Sie von der Homepage herunterladen können, verwenden. Die Funktion berechnet zu einer Matrix-Spalte x den zugehörigen Householder-Vektor.

(b) Testen Sie ihre Funktion für die Matrix

$$\begin{pmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{pmatrix}.$$

Lassen Sie in jedem Schritt die Transformationsmatrix Q_i ausgeben. Lassen Sie zudem in jedem Schritt die transformierte Matrix $Q_i \cdots Q_1 A Q_1^T \cdots Q_i$ als Sternchenbild ausgeben. Berechnen Sie am Ende die komplette Transformationsmatrix Q.

(c) Schreiben Sie eine Funktion `[H,Q,R] = hessqr(H,maxit)`, die das QR-Verfahren für eine Matrix in Hessenbergform durchführt.

Die QR-Zerlegung in jedem Schritt soll hierbei mit Hilfe von Givens-Rotationen durchgeführt werden. Achten Sie darauf nur die nötigen Informationen abzuspeichern.

(d) Schreiben Sie ein Skript `main.m`, in dem Sie ihre Funktionen aus Teil (a) und (b) für eine zufällige, symmetrische Matrix testen.

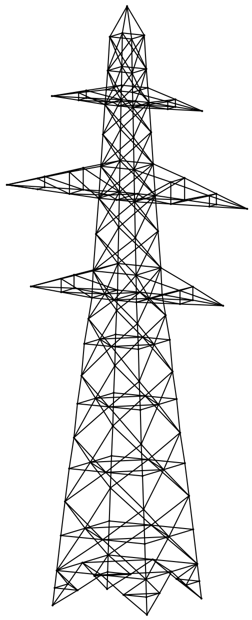
Aufgabe 21 *(Nullstellen von Polynomen)*

Sei $f : \mathbb{C} \to \mathbb{C}$ definiert durch $f(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_0$ wobei $a_0, \ldots a_{n-1} \in \mathbb{C}$ und $n \in \mathbb{N}$. Zeigen Sie mit Hilfe des Satzes von Gerschgorin:
Wenn z eine Nullstelle von f ist, so gilt

$$|z| \leq \max\left\{1, \sum_{i=0}^{n-1} a_i\right\}.$$

Hinweis: Betrachten Sie die Begleitmatrix zum Polynom f.

Aufgabe 22 *(Eigenschwingungen eines Hochspannungsmastes)*



Ziel dieser Aufgabe ist es die Eigenschwingungen eines Hochspannungsmastes (siehe Abbildung) zu berechnen. Der Mast ist durch $N_C = 167$ Knotenpunkte und $N_E = 503$ Verbindungsträger gegeben. Für die Rechnung berücksichtigen wir für jedes Balkenelement Biegungen in 2 Ebenen $w(x)$ und $v(x)$, Längsdehnungen $u(x)$ und Torsionen, gegeben durch den Torsionswinkel $\theta(x)$. Hier ist $x \in \mathbb{R}^3$ Punkt des Balkenelements. Für jedes Balkenelement ist die potenzielle Energie gegeben als Funktional der Biegungen, Dehnungen und Torsionen. Ziel ist es nun, die gesamte potentielle Energie des Systems zu minimieren. Diskretisiert man nun die Funktionale für alle Balkenelemente mit Hilfe der Methode der Finiten Elemente (Numerik 4), so erhält man ein Minimierungsproblem der Form

$$\frac{1}{2}\mathbf{u^T S u} - \frac{1}{2}\lambda \mathbf{u^T M u} \to \min \qquad \Longleftrightarrow \qquad \mathbf{Su} = \lambda \mathbf{Mu},$$

also ein Eigenwertproblem der Form

$$\mathbf{M^{-1} S u} = \lambda \mathbf{u}. \tag{1}$$

Der Lösungsvektor $\mathbf{u} \in \mathbb{R}^{\mathbf{6N_C}}$ enthält für jeden Knotenpunkt die Verschiebung in x, y und z-Richtung, den Torsionswinkel sowie die Ableitung der Verschiebung in y und z-Richtung. Die Matrix $\mathbf{S}$ heißt Steifigkeitsmatrix und enthält für jeden Knotenpunkt einen 6×6-Block (also $\mathbf{A} \in \mathbb{R}^{\mathbf{6N_C \times 6N_C}}$). Die Matrix $\mathbf{M} \in \mathbb{R}^{\mathbf{6N_C \times 6N_C}}$ heißt Massenmatrix.
Durch Lösen des obigen Eigenwertproblems erhalten wir also Eigenwerte und zugehörige Eigenvektoren, welche für jeden Knotenpunkt unter anderem die Verschiebungen in die drei Koordinaten-Richtungen enthalten.

(a) Laden Sie die Datei fem3d_stab.m von der Homepage herunter. Diese soll in den folgenden Aufgabenteilen vervollständigt werden.

1. Datenstrukturen für die Knoten und Balkenelemente

Um die Geometrie des Hochspannungsmastes im Computer zu speichern, verwenden wie die Matrizen `coordinates` und `elements`. Die $N_C \times 3$ Matrix `coordinates` enthält die Koordinaten aller Knotenpunkte (jede Zeile entspricht einem Punkt). Die $N_E \times 4$ Matrix `elements` enthält für jedes Balkenelement die Nummer des Anfangs- und Endpunktes, sowie die Breite und Tiefe des jeweiligen Balkens.
Die vier Knoten, an denen der Mast im Boden verankert ist, bewegen sich natürlich nicht. Die Nummern der vier Knoten sind im Vektor `bdry` gespeichert. Uns interessiert also nur die Verschiebung in den "freien Knoten". Der Vektor `freenodes` enthält die Indizes der Einträge im Vektor $\mathbf{u}$, welche zu freien Knoten gehören (für jeden freien Knoten also 6 Einträge).

(b) Berechnen Sie den Vektor `freenodes` (Zeilen 23-26).

2. Eigenwertproblem lösen und Verschiebung zur k-ten Eigenschwingung extrahieren

(c) Lösen Sie das Eigenwertproblem (1) und sortieren Sie die Eigenvektoren und Eigenwerte so, dass die Eigenwerte absteigend geordnet sind.

(d) Extrahieren Sie aus dem Eigenvektor zum k-ten Eigenwert (für $k \in \{1, 4\}$) die drei Verschiebungen für jeden Knoten und speichern Sie diese im Vektor `u` (also ist `u`$\in \mathbb{R}^{3N_C}$). Skalieren Sie `u` so, dass die maximale Verschiebung den Betrag `scal` hat.

3. Eigenschwingung darstellen
Um die Eigenschwingung darzustellen, wollen wir eine Animation erzeugen. Hierfür berechnen wir für $t \in [0, 2\pi]$ (`t = linspace(0,2*pi,nFrames)`) die Auslenkung zum Zeitpunkt t_k in jedem Knoten x_i durch $\tilde{x}_i(t) = x_i + \sin(t_k) \cdot u$. Führen Sie für jeden Zeitschritt folgende Schritte aus:

(e) Zeichnen Sie die Knoten-Punkte x_i (mit `plot3d`).

(f) Zeichnen Sie die Balkenelemente des unausgelenkten Mastes (mit `plot3d`, Sie brauchen eine `for`-Schleife über alle Balkenelemente).

(g) Zeichnen Sie die Knotenpunkte und Balkenelemente des ausgelenkten Mastes.

Für jeden Zeitschritt wird dann das aktuelle Bild mit Hilfe von `getframe` gespeichert. Anschließend können dann alle Bilder nacheinander als Video mit dem Befehl `movie` wiedergegeben werden.
Zusatzaufgabe:
Mit Hilfe der Befehle `avifile` und `addframe` kann man die Animation als avi-Datei abspeichern. Ergänzen Sie Ihr Programm so, dass ein avi-Video des schwingenden Mastes erzeugt wird.