Prof. Dr. Karsten Urban
M.Sc. Florian Kunstmann
Institut für Numerische Mathematik
Universität Ulm

Übungsblatt 4

Besprechung 11.11.2015.

Hinweise: Die Abgabe der Lösungen der Matlab-Aufgaben 11 und 12 erfolgt per Email bis Dienstag, 10.11.2015 18 Uhr an

florian.kunstmann@uni-ulm.de.

Wichtig: Aufgabe 13 (Hochspannungsmast) ist erst bis zum 16.11.2015 23.59 Uhr einzureichen.
Der Betreff sollte lauten "Num3Blatt$x$" (wobei x für die Nummer des Blattes steht). Die Lösungen müssen als **Anhang** an die Email versendet werden. Für jede Programmieraufgabe ist ein zip-file "AufgabeMy" zu erstellen (wobei y für die Nummer der Aufgabe steht), das die nötigen .m-files enthält.

Aufgabe 11 (Hooke-Jeeves Verfahren) (5 Punkte)

(a) Implementieren Sie das Hooke-Jeeves Verfahren in den Funktionen `explore.m` und `progress.m`. Erweitern Sie ferner ihre Funktion `progress.m` um die Möglichkeit, den Fortschreitungsschritt (also die Funktion selbst) rekursiv aufzurufen, solange die Suchrichtung noch optimal ist.

(b) Testen Sie Ihr Verfahren an der Himmelblau Funktion $f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ mit den Startwerten $(6,4), (0,0), (-5,-5), (5,-3)$. Geben Sie zusätzlich am Ende einen Plot der Funktion und den zurückgelegten Suchpfad aus.

(c) Zerlegen Sie die Ebene in Gitterpunkte und lassen Sie für jeden Gitterpunkt das Verfahren laufen (z.B $[-5,5] \times [-5,5]$). Geben Sie jedem der 4 Minimas eine Farbe und markieren Sie den Gitterpunkt (bzw. das dazugehörige Flächenelement) mit der entsprechenden Farbe, gegen welches Minima der Startpunkt konvergiert ist. Plotten Sie Ihr Gebiet mit der entsprechenden Farbkodierung.

(d) Testen Sie den Algorithmus für die Rosenbrock-Funktion

$$f(x,y) = (1-x)^2 + 100(y - x^2)^2$$

mit Startwert $(-0.5, 1.5)$. Lassen Sie wieder den Suchpfad zeichnen und erklären Sie was Sie sehen.

Aufgabe 12 (Nelder-Mead Verfahren) (5 Punkte)

(a) Implementieren Sie den Algorithmus von Nelder und Mead in Matlab. Achten Sie darauf, dass Sie möglichst geschickt programmieren, d.h. z.B. so wenig Funktionsauswertungen wie möglich durchführen. Der Algorithmus soll abbrechen, wenn die maximale Anzahl an Iterationen erreicht ist, oder wenn der Mittelwert der Funktionswerte in den Simplex-Ecken kleiner als eine gegebene Toleranz ist.

(b) Testen Sie den Algorithmus für die Rosenbrock-Funktion

$$f(x,y) = (1-x)^2 + 100(y - x^2)^2$$

mit Start-Simplex

$$\{(-1.2, 1), (-0.23407, 1.25882), (-0.94118, 1.96593)\}.$$

Zeichnen Sie einen Contour-Plot der Funktion und für jede Iteration das aktuelle Simplex in eine Grafik.

(c) Testen Sie den Algorithmus für die Funktion (nach McKinnon)

$$f(x,y) = \begin{cases} 360x^2 + y + y^2 & x < 0, \\ 6x^2 + y + y^2 & x \geq 0, \end{cases}$$

mit Start-Simplex

$$\{(1,1), (0.8, -0.6), (0,0)\}.$$

Zeichnen Sie einen Contour-Plot der Funktion und für jede Iteration das aktuelle Simplex in eine Grafik.

Aufgabe 13 (Eigenschwingung eines Hochspannungsmasten) (15 Punkte)

Wichtig: Diese Aufgabe ist erst bis zum 16.11.2015, 23:59 Uhr einzureichen.

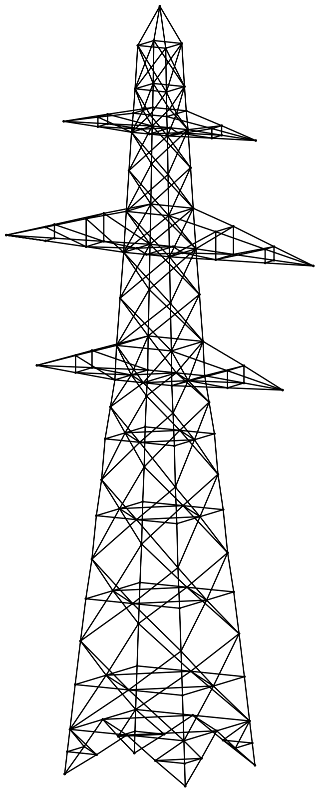

Abbildung 1: Hochspannungsmast

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

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

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 **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 **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.

(e) 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.