Universität Konstanz
FB Mathematik & Statistik
Prof. Dr. M. Junk
J. Budday

Ausgabe: 14.01.2013

Abgabe: 21.01.2013
bis 9.55 Uhr in die
Briefkästen vor F441

Übungen zur Veranstaltung
Numerik partieller Differentialgleichungen

Blatt 03

Aufgabe 1: M-Matrizen

Eine Matrix $A \in \mathbb{R}^{n \times n}$ heißt *zeilendiagonaldominant*, wenn für alle $i \in \{1, \ldots, n\}$ gilt:

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

Gilt für alle $i \in \{1, \ldots, n\}$ sogar die strikte Ungleichung, so heißt A *streng zeilendiagonaldominant*.

(a) Zeigen Sie: Ist A eine M-Matrix, so gilt $a_{ii} > 0$ für alle $i \in \{1, \ldots, n\}$.

(b) Zeigen Sie: Ist A eine streng zeilendiagonaldominante L_0-Matrix mit $a_{ii} > 0$ für alle $i \in \{1, \ldots, n\}$, so ist A eine M-Matrix.

(c) Implementieren Sie jeweils eine Routine zur Überprüfung, ob bei einer gegebenen Matrix A die Eigenschaften einer L_0-Matrix bzw. einer M-Matrix vorliegen. (Ausgabe: 1 = true, 0 = false)

Aufgabe 2: Randwertaufgabe

Für $\Omega := (0, 1)^2$ betrachten wir die folgende Randwertaufgabe:

$$\begin{cases} \Delta u(x, y) = g(x, y) \ , \ (x, y) \in \Omega \\ u(x, y) = \gamma(x, y) \ , \ (x, y) \in \partial\Omega \end{cases}$$

Angenommen die Lösung u dieses Problems sei durch $u(x, y) = P_1(x)P_2(y)$ mit zwei Polynomen P_1, P_2 mit $\deg P_i \leq 3$ für $i = 1, 2$ gegeben. Zeigen Sie, dass in diesem Fall die Lösung des Verfahrens aus Abschnitt 1.4 immer der exakten Lösung entspricht.

- bitte wenden -

Aufgabe 3: Krummlinige Ränder

Wir betrachten erneut die Randwertaufgabe aus Aufgabe 2, jedoch nun für den Einheitskreis $\Omega := \{(x, y)^T \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}$. Geben Sie eine interessante Lösung u der Randwertaufgabe vor und bestimmen Sie die Funktionen g und γ so, dass u tatsächlich Lösung dieser Randwertaufgabe ist. Ziel ist es nun, diese Randwertaufgabe über der vorgegebenen krummlinig berandeten Menge Ω numerisch zu lösen. Für die Implementierung sollten Sie folgende Hinweise unbedingt beachten:

- Betten Sie den Einheitskreis Ω in ein Quadrat Q ein, das an allen 4 Seiten einen echt positiven Abstand zu $\partial\Omega$ aufweist. Erzeugen Sie mit

  ```
  [X,Y]=meshgrid(...)
  ```

  auf Q ein Gitter, das den Nullpunkt enthält, und finden Sie mit Hilfe des `find`-Befehls durch `I=find(X.^2+Y.^2<1)` die Nummern jener Gitterpunkte, die in Ω liegen.

- Zur Nummerierung der Gitterpunkte über Ω wird es hilfreich sein, die Nummer eines Gitterpunktes in die zugehörigen Indizes der Gittermatrix umrechnen zu können und umgekehrt. Programmieren Sie hierfür mit den Nummern in `I`, den Befehlen `find`, `ind2sub` und `sub2ind` die im Skript eingeführte Funktion ν und ihre Inverse ν^{-1} inline in Ihrem Code. (Bemerkung: $\nu(i, j)$ soll [ ] sein, wenn $x_{ij} \notin \Omega$. Beachten Sie, dass gilt: $x_{ij} = (ih, jh)$)

- Als nächstes müssen aus den gefundenen Gitterpunkten die numerischen Randpunkte ausfindig gemacht werden, also jene die zwar in Ω liegen, ihnen dort aber in mindestens einer Raumrichtung ein Gitternachbar aus Ω fehlt. Die einfachste Möglichkeit ist zunächst eine Schleife über alle Ω-Gitterpunkte zu programmieren, in der die Existenz der 4 Nachbarn überprüft wird. Die Nummern der Nachbarn erhalten Sie mit den Funktionen ν und ν^{-1} (z.B. den östlichen Nachbarn mit `[i,j] =` ν^{-1}`(k)` und ν`(i+1,j)`).

- Stellen Sie das zu lösende Gleichungssystem auf und behandeln Sie die Randpunkte dabei wie es im Skript für solche Fälle beschrieben wurde. Überprüfen Sie Ihre Systemmatrix mit den in Aufgabe 1(c) implementierten Routinen.

- Verwenden Sie zur Visualisierung Ihrer numerischen Lösung das gesamte Gitter über Q. Damit die Lösung nachher jedoch nur direkt über Ω visualisiert wird und Matlab am Rand von Ω keine graphischen Übergänge interpoliert, gehen Sie wie folgt vor:

  Erzeugen Sie eine dem numerischen Q-Gitter entsprechend große Matrix mit `NaN`-Einträgen. Wenn also mit

  ```
  [X,Y]=meshgrid(linspace(a,b,n),linspace(c,d,m))
  ```

  das Gitter erzeugt wurde, so gelingt dies einfach durch `Z=NaN(m,n)`. Sind in `I` jene Nummern der Gitterpunkte abgespeichert worden, die über den `find`-Befehl als Punkte aus Ω identifiziert wurden, und in `u` die zugehörigen Werte Ihrer numerischen Lösung, so ersetzen Sie mit `Z(I)=u` die entsprechenden `NaN`-Einträge über Ω durch Ihre Lösungswerte. Plotten Sie nun mit Hilfe von `surf(X,Y,Z)`, so wird das graphische Interpolieren aufgrund der `NaN`-Einträge außerhalb von Ω verhindert.