Prof. Dr. Dimitri Kuzmin
Dipl.-Math. Justus Klipstein
Numerische Mathematik für Physiker und Ingenieure

Sommersemester 2018
Übungsblatt 9

Abgabe der **Aufgaben bis Donnerstag, 05.07.2018, 16:15 Uhr vor der Vorlesung** (einzeln, zu zweit oder dritt) in den Briefkasten Ihrer Übungsgruppe:

Gruppe	Termin	Übungsleiter	Briefkasten
2	Montag 10:00-12:00	Korinna Rosin	107
3	Montag 12:00-14:00	Mirco Arndt	108
4	Montag 14:00-16:00	Marina Bangert	109
5	Montag 16:00-18:00	Marina Bangert	109
6	Dienstag 14:00-16:00	Justus Klipstein	108
7	Dienstag 16:00-18:00	Dr. Fatma Ibrahim	110

Die Briefkästen befinden sich im Foyer des Mathematikgebäudes. Bitte vermerken Sie unbedingt auf jeder Abgabe Ihren Namen und Ihre Übungsgruppe. Eine Abgabe der theoretischen Übungsaufgaben per E-Mail ist nicht möglich.
Abgabe der **Programmieraufgaben bis Donnerstag, 12.07.2018, 24:00 Uhr** digital im Moodle-Arbeitsraum der Veranstaltung. Bei Abgabe zu zweit oder dritt bitte nur einmal einreichen und im Kommentar den Namen und E-Mail Adresse des/der Koautors/Koautorin/Koautoren nennen. Sie können die Aufgaben in MatLab bzw. Octave schreiben. Bitte den "Header" als Kommentar in die Programme einfügen.

Aufgabe 9.1 (Jacobi- und Gauß-Seidel-Verfahren | 2+2+4 Punkte)

Gegeben sei das lineare Gleichungssystem

$$\underbrace{\begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 2 \\ 1 & 2 & 5 \end{bmatrix}}_{=:A} \mathbf{x} = \underbrace{\begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}}_{=:\mathbf{b}}.$$

Es soll mit iterativen Verfahren der Form

$$P\mathbf{x}_{n+1} = N\mathbf{x}_n + \mathbf{b}$$

approximativ gelöst werden (wobei $A = P - N$).

a) Geben Sie jeweils die Matrizen P und N, welche dem Jacobi- und Gauß-Seidel Verfahren entsprechen, explizit an.

b) Warum konvergieren bei diesem Problem beide Verfahren für jeden Startwert $x_0 \in \mathbb{R}^3$?

c) Führen Sie die ersten zwei Schritte für beide Verfahren mit dem Startwert $\mathbf{x} = (0, 0, 0)^\top$ aus.

Aufgabe 9.2 (Jacobi- und Gauß-Seidel-Verfahren | 2+6 Puntkte)

Betrachten Sie die Matrix

$$A := \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 \end{bmatrix}.$$

a) Ist das Gauss-Seidel-Verfahren bezüglich A konvergent?

b) Sei $b \in \mathbb{R}^3$ beliebig. Berechnen Sie für das Jacobi-Verfahren, angewendet auf das lineare Gleichungssystem $Ax = b$, eine Kontraktionszahl ρ, d. h. ein ρ mit

$$\left\|e^{n+1}\right\|_\infty \leq \rho \left\|e^n\right\|_\infty \qquad \forall n \in \mathbb{N},$$

wobei $e^{n+1} := x^{n+1} - x^*$ und $x^* = A^{-1}b$. Wie viele Schritte N des Jacobi-Verfahrens sind erforderlich, um den Anfangsfehler e^0 um einen Faktor von 10^{-6} zu reduzieren?

Programmieraufgabe 9.1 (Jacobi- und Gauss-Seidel-Verfahren | 4+4+5+5+6+6 Punkte)

Implementieren Sie die iterativen Löser

a) `[x, numit] = my_jacobi(A,b,x0,eps,maxit)`,

b) `[x, numit] = my_jacobi_vector(A,b,x0,eps,maxit)`,

c) `[x, numit] = my_gauss_seidel(A,b,x0,eps,maxit)`,

d) `[x, numit] = my_sor(A,b,x0,eps,maxit,omega)`,

welche die Lösung $\mathbf{x}$ eines linearen Gleichungssystems $A\mathbf{x} = \mathbf{b}$ mittels Jacobi- bzw. Gauss-Seidel (SOR) approximativ lösen. Dabei soll jeweils Folgendes beachtet werden:

- Die Ausgabeargumente seien die approximierte Lösung $\mathbf{x}$ sowie die Anzahl der durchgeführten Iterationen `numit`.
- Das Jacobi-Verfahren soll zweimal implementiert werden. Einmal als zeilenweise arbeitendes Verfahren. Der Löser `my_jacobi_vector(A,b,x0,eps,maxit)` soll zum berechnen der approximierten Lösung hingegen nur Matrix-Vektor-, Matrix-Matrix-Multiplikationen und Vektor-Addition nutzen.
- Die Iteration soll jeweils beendet werden, wenn der *relative Defekt* (bzgl. $\|\cdot\|_2$, also $\left\|Ax^k - b\right\|_2$) einer Iterierten kleiner ist als `eps`-mal der *relative Eingangsdefekt* ($eps \cdot \|A\ x0 - b\| \,/\, \|x0\|$).
- Wird die Anzahl der maximalen Iterationen `maxit` überschritten, soll die Routine eine Warnung erzeugen und die Iteration abbrechen.
- In `my_gauss_seidel` und `my_sor` ist eine Vorwärtssubstitution auf eine Submatrix von A nötig. Verwenden Sie dazu den Matlab Backslash-Lösungsoperator \ .
- Zum invertieren der Teilmatrizen darf der Befehl `invMatrix = inv( Matrix )` benutzt werden.

e) Betrachten Sie nun das Gleichungssystem $A^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$ mit

$$A^{(n)} := \begin{bmatrix} T & -\mathrm{Id} & & & 0 \\ -\mathrm{Id} & T & -\mathrm{Id} & & \\ & \ddots & \ddots & \ddots & \\ & & -\mathrm{Id} & T & -\mathrm{Id} \\ 0 & & & -\mathrm{Id} & T \end{bmatrix}, \qquad T \quad := \begin{bmatrix} 4 & -1 & & & 0 \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ 0 & & & -1 & 4 \end{bmatrix},$$

$A \in \mathbb{R}^{n^2 \times n^2}$, $T \in \mathbb{R}^{n \times n}$, $\mathrm{Id} \in \mathbb{R}^{n \times n}$ die n-Einheitsmatrix und $\mathbf{b}^{(n)} := (1, 1, \ldots, 1)^\top \in \mathbb{R}^{n^2}$. Erstellen Sie eine Funktion `[A,b] = my_test_system(n)`, welche in Abhängigkeit des Aufrufparameter n obige Matrix $A^{(n)}$ und den Vektor $\mathbf{b}^{(n)}$ erstellt und zurückgibt.

f) Testen Sie in einem Skript `PA10` für $n = 10, 11, \ldots, 60$ die iterativen Löser aus Teil **a)** [oder **b)**], **c)** und **d)** an den mittels `my_test_system` erzeugten linearen Gleichungssystemen, in dem Sie in einem Plot die Anzahl der Iterationen der drei Verfahren in Abhängigkeit von n vergleichen (x-Achse: n, y-Achse: Anzahl Iterationen). Verwenden Sie `eps = 1e-6`, `maxit = 1e8` und den Relaxationsparameter $\omega_{\text{opt}} = \frac{2}{1+\sin(\frac{\pi}{n})}$ sowie $\mathbf{x} = \mathbf{b}$ als Startwert. Speichern Sie den erzeugten Plot als `PA10.pdf` oder ähnlichem ab.
Messen Sie für $n = 60$ mit `tic; toc;` die Zeit, die die iterativen Löser aus Teil **a)**, **b)**, **c)** und **d)** zum lösen des linearen Gleichungssystemen benötigen. Geben Sie diese über das `Command Window` (console) aus.