Abgabe der **Theorieaufgaben bis Donnerstag, 07.07.2016, 16:15 Uhr** (einzeln oder zu zweit) in den Briefkasten Ihrer Übungsgruppe. Bitte vermerken Sie auf Ihrer Abgabe Ihre Übungsgruppe sowie Name und Matrikelnummer. Heften Sie mehrere Blätter zusammen.
Abgabe der **Programmieraufgaben bis Donnerstag, 14.07.2016, 16:15 Uhr** digital im Moodle-Arbeitsraum der Veranstaltung. Bei Abgabe zu zweit bitte nur einmal einreichen und im Kommentar den Namen und E-Mail Adresse des/der Koautors/Koautorin nennen. Sie können die Aufgaben in MATLAB bzw. OCTAVE schreiben. Alternativ kann auch in C oder C++ abgegeben werden, allerdings ohne mögliche Hilfestellung. Es kann nur lauffähiger Code bewertet werden.
Bei Fragen zu Vorlesung und Übungen können Sie neben der Übungsgruppe auch das **Tutorium** nutzen, jeweils **Mittwochs von 13:15 bis 14:45** Uhr im CIP-Pool der Mathematik (M946).

Aufgabe 10.1 (Jacobi- und Gauss-Seidel-Verfahren | 2 + 6 Puntkte)

Betrachten Sie die Matrix

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

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?

Aufgabe 10.2 (Gradienten-Verfahren | 6 Punkte)

Berechnen Sie die ersten drei Iterierten des Gradientenverfahrens zur Minimierung von

$$f(x) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top \mathbf{x}$$

für $\mathbf{x} \in \mathbb{R}^2$ und mit

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Verwenden Sie den Startvektor $\mathbf{x}^{(0)} = (0, 0)^\top$.

Für die folgende Programmieraufgabe soll neben JACOBI- und GAUSS-SEIDEL-Verfahren auch das "SOR"-Verfahren (Successive Over Relaxation) implementiert werden. Dieses wird im Skript ab Seite 198 vorgestellt. Dabei wird im t-ten Iteterationsschritt (Notation wie im Skript) zunächst der Gauß-Seidel-Zwischenwert

$$\tilde{x}_j^t = \frac{1}{a_{j,j}} \left(b_j - \sum_{k<j} a_{j,k} x_k^t - \sum_{k>j} a_{j,k} x_k^{t-1} \right)$$

gebildet. Die nächste Iterierte berechnet sich dann als Linearkombination der letzten Iterierten und des Zwischenschritts:

$$x_j^t = \omega \tilde{x}_j^t + (1-\omega) x_j^{t-1}$$

mit dem Relaxationsparameter $\omega \geq 1$. Für $\omega = 1$ ergibt sich also gerade das GAUSS-SEIDEL-Verfahren.

Programmieraufgabe 10.1 (Jacobi- und Gauss-Seidel-Verfahren | 5+6+6+5+3 Punkte)
Implementieren Sie die iterativen Löser

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

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

c) `[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 approximativ lösen. Dabei soll jeweils Folgendes beachtet werden:

- Die Ausgabeargumente seien die approximierte Lösung $\mathbf{x}$ sowie die Anzahl der durchgeführten Iterationen `numit`.
- 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*.
- 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 \.

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

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

$A \in \mathbb{R}^{n^2 \times n^2}$, $T \in \mathbb{R}^{n \times n}$, $\mathrm{Id} \in \mathbb{R}^{n \times n}$ 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.

d) Testen Sie in einem Skript `PA10` für $n = 10, 11, \ldots, 60$ die iterativen Löser aus Teil **a)**, **b)** und **c)** 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` ab.