Übungsserie 10

Abgabe KW 20

Scannen Sie ihre manuellen Lösungen für die Aufgaben 1 und 2 in die Dateien *Name_ Vorname_ Klasse_ S10_ Aufg1.pdf* bzw. *Name_ Vorname_ Klasse_ S10_ Aufg2.pdf* und fassen Sie diese mit Ihrer MATLAB-Funktion *Name_ Vorname_ Klasse_ S* und dem Skript *Name_ Vorname_ Klasse_ S10_ Aufg3b*.m in einer ZIP-Datei *Name_ Vorname_ Klasse_ S10.zip* zusammen. Laden Sie dieses File vor der Übungsstunde nächste Woche auf OLAT hoch.

Aufgabe 1 (ca. 45 Minuten):

Gegeben ist das lineare Gleichungssystem

$$Ax = b \text{ mit } A = \begin{pmatrix} 8 & 5 & 2 \\ 5 & 9 & 1 \\ 4 & 2 & 7 \end{pmatrix} \text{ und } b = \begin{pmatrix} 19 \\ 5 \\ 34 \end{pmatrix}.$$

a) Überprüfen Sie, ob das obige System bzgl. dem Jacobi-Verfahren konvergiert.

b) Berechnen Sie auf vier Stellen nach dem Komma die Näherung $x^{(3)}$ mit dem Jacobi-Verfahren ausgehend vom Startvektor $x^{(0)} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$. Schreiben Sie alle benötigten Matrizen sowie die verwendete Iterationsgleichung explizit auf. Die Iterationen selber führen Sie aber natürlich mit MATLAB durch.

c) Wie gross ist gemäss der a-posteriori Abschätzung der absolute Fehler von $x^{(3)}$?

d) Schätzen Sie a-priori die Anzahl Iterationsschritte ab, damit der berechnete Näherungsvektor in jeder Kompnente maximal um 10^{-4} von der exakten Lösung $x = (2, -1, 4)^T$ abweicht.

e) Wiviele Iterationsschritte würden Sie a-priori benötigen, wenn Sie als Startvektor nicht $x^{(0)}$ sondern $x^{(2)}$ aus b) verwenden würden?

Aufgabe 2 (ca. 30 Minuten):

Wiederholen Sie die obige Aufgabe, diesmal für das Gauss-Seidel Verfahren. Sie dürfen (ausnahmsweise) die Inverse von $D + L$ benutzen (müssen aber nicht, wenn Sie nicht wollen).

Aufgabe 3 (ca. 75 Minuten):

a) Implementieren Sie das Jacobi- und Gauss-Seidel-Verfahren zusammen in einer Funktion als `[xn, n, n2] = Name_Vorname_Klasse_S10_Aufg3a(A,b,x0,tol,opt)`. Sie können dabei die Matrix-Funktionen von MATLAB benutzen, (z.B. `triu(A), diag(diag(A)), tril(A), (D+L)^(-1)`), ohne aber `A^(-1)`zu berechnen. Dabei soll `xn` der Iterationsvektor nach `n` Iterationen sein, zusätzlich soll `n2` die Anzahl benötigter Schritte gemäss der a-priori Abschätzung angeben. Über den Parameter `opt` soll gesteuert werden, ob das Jacobi- oder das Gauss-Seidel Verfahren zur Anwendung kommt. Überlegen Sie sich, wie die Abbruchbedingung für Ihre while-Schleife lauten muss, um die Iteration bei Erreichen einer vorgegebenen Fehlertoleranz `tol` abzubrechen. Sie werden dafür die Norm brauchen: `norm(...,inf)`.Achten sie darauf, dass Sie Matrizen, die Sie in ihrer Funktion nicht mehr brauchen, gleich mit dem Befehl `clear` wieder löschen, um Speicher freizugeben.

b) Schreiben Sie ein kurzes Skript `Name_Vorname_Klasse_S10_Aufg3b.m.` Testen Sie damit die Laufzeit Ihres Programmes für Jacobi- und Gauss-Seidel getrennt im Vergleich zum Gauss-Verfahren, welches Sie in Serie 7 implementiert hatten (siehe *Name_ Vorname_Klasse_S7_Aufg2.m*). Verwenden Sie dafür die folgenden Werte für $A, b, x0$ und tol:

```
>> A = diag(diag(ones(3000)*4000))+ones(3000);
>> x = [1:1:1500,1500:-1:1]';
>> b = A*x;
>> x0 = zeros(3000,1);
>> tol = 1e-4;
```

Den Zeitvergleich können Sie dabei analog wieder mit tic() und toc() messen, z.B.

```
>> t1 = tic; [xn, n, n2] = Name_Klasse_S10_Aufg3a(A,b,x0,tol,Jacobi); toc(t1)
```

Wieviel länger braucht die Gauss-Zerlegung? Achtung: lassen Sie die Gauss-Zerlegung nur laufen, wenn Sie Ihren Computer für einige Minuten nicht für anderes brauchen. Schreiben Sie die gemessenen Werte als Kommentar in Ihr Programm.