

3. Programmierprojekt zu Algorithmen der Numerischen Mathematik

(LGS und nichtlineare Optimierung)

- a) Programmieren Sie die Methode des steilsten Abstiegs zur Minimierung der Funktion

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, (x_1, x_2) \mapsto 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

mit Startvektor $(-1.2, 1)$ und stellen Sie die ersten 50 Iterierten graphisch dar.

- b) Programmieren Sie die Methode des steilsten Abstiegs zur Lösung eines linearen Gleichungssystems $Ax = b$ mit symmetrisch positiv definiten Matrix A .

Testen Sie Ihre Funktion anhand von

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 20 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_0 = \begin{pmatrix} 20 \\ 1 \end{pmatrix}$$

und stellen Sie die ersten 50 Iterierten graphisch dar.

Hinweis: Besonders anschaulich wird die graphische Darstellung, wenn die Iterierten in ein Höhenprofil von der (zu minimierenden) Funktion f eingezeichnet werden. Das geht zum Beispiel mit

```
hold on;  
[x1,x2] = meshgrid(-20:1:20,-20:1:20);  
contour(x1,x2,x1.^2 + 20*x2.^2,[0:1:400].^2);  
plot(hier sollten die Iterierten stehen,'-*','Linewidth',2)  
hold off;
```

- c) Programmieren Sie das cg-Verfahren zur Lösung eines linearen Gleichungssystems $Ax = b$ mit symmetrisch positiv definiten Matrix A .

Testen Sie Ihre Funktion anhand der Matrix aus Teilaufgabe b) und stellen Sie wieder die Iterierten graphisch dar. Vergleichen Sie die Ergebnisse mit denen aus Teilaufgabe b). Testen Sie Ihre Funktion dann anhand der Matrix generiert durch

```
function A = MatrixGenerator(N)  
A = -4*diag(ones(N^2,1)) - diag(ones(N*(N-1),1),N) - diag(ones(N*(N-1),1),-N);  
for i=0:N-1  
    for j=1:N-1  
        A(j+i*N,j+1+i*N) = -1;  
        A(j+1+i*N,j+i*N) = -1;  
    end  
end
```

mit $N = 4, 20, 40$.

- d) Programmieren Sie vorkonditionierte cg-Verfahren zur Lösung eines linearen Gleichungssystems $Ax = b$ mit symmetrisch positiv definiten Matrix A .

Testen Sie Ihre Funktion zum Beispiel anhand der Matrix aus Teilaufgabe c). Vergleichen Sie die Ergebnisse mit denen aus Teilaufgabe c).

- e) (BONUS) Programmieren Sie das FOM- sowie das GMRES-Verfahren. Testen Sie Ihre Implementierungen anhand des Gleichungssystems

```
load west0479;  
A = west0479;  
b = sum(A,2);
```

Mittels `spy(A)` können Sie sich einen Eindruck von der Struktur der Matrix A verschaffen. Stellen Sie jeweils die Norm der Residuen in Abhängigkeit von der Anzahl der Iterationsschritte dar.

Implementieren Sie auch eine Linksvorkonditionierung für das GMRES-Verfahren, d.h. wenden Sie das Verfahren auf das zu $Ax = b$ äquivalente System

$$B^{-1}Ax = B^{-1}b$$

an. Benutzen Sie anstelle von B die unvollständige LU-Zerlegung

```
[L,U,P] = ilu(A,struct('type','ilutp','droptol',1e-6));
```

und plotten Sie wieder die Residuen.

Besprechung der Programmieraufgabe in den Übungen am 10.07.2013.
Jede Teilaufgabe ist ein Kreuz wert.