PROF. DR. BERND SIMEON
CHRISTIAN GOBERT
THOMAS MÄRZ

Numerik

SS 2009

Lösungsvorschläge zu Blatt 1

Lösungsvorschlag zu Aufgabe 1 *Numerische Berechnung der Jacobimatrix*
Die Jacobi-Matrix hat folgende Struktur:

$$DF(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{pmatrix}.$$

a) Wir approximieren $\frac{\partial F}{\partial x_k}(x) \in \mathbb{R}^n$ durch einen Vorwärtsdifferenzenquotient

$$\frac{\partial F}{\partial x_k}(x) \approx \Delta_k F(x) = \frac{1}{h} \cdot \Big(F(x + he_k) - F(x)\Big),$$

wobei e_k den k-ten euklidischen Einheitsvektor bezeichnet. Der Algorithmus sieht dann so aus:

```
h0=sqrt(eps); y=f(x); n=length(y);
Df=zeros(n); E=eye(n);

for k=1:n
    h=max(1,abs(x(k)))*h0;
    Df(:,k)=(f(x+h*E(:,k))-y)/h;
end
```

Erklärung: Ähnlich wie bei Hausaufgabe H5 in Numerik I ergibt sich auch hier bei Approximation der Ableitungen durch Vorwärtsdifferenzen ein Fehler der Ordnung $\mathcal{O}(h + \text{eps}/h)$, der sich aus Approximations- und Rundungsfehleranteil zusammensetzt. Die optimale Wahl $h_0 = \sqrt{\text{eps}}$ liefert einen Gesamtfehler von der Ordnung $\mathcal{O}(\sqrt{\text{eps}})$, also bei t-stelliger Arithmetik etwa $t/2$ Stellen Genauigkeit. Die Skalierung `h=max(1,abs(x(k)))*h0` verhindert eine zu kleine Störung im Fall $x >> 1$.

Feedback-Strategie für die Wahl von h. Damit der Auslöschungsfehler bei der Berechung von $\Delta_k F(x)$ nicht zu groß wird, fordert man

$$\|F(x + \hat{h}e_k) - F(x)\| \approx \|F(x)\| \sqrt{\text{eps}} \qquad (*)$$

d.h. bei t-stelliger Arithmetik sollen höchstens etwa $t/2$ führende Ziffern in der Differenz ausgelöscht werden. Wurde der Störparameter h zu klein gewählt, ist also

$$\|F(x + he_k) - F(x)\| << \|F(x)\| \sqrt{\text{eps}},$$

so erhält man wegen $\|F(x+\hat{h}e_k)-F(x)\| \doteq \|(D_kF)(x)\|\cdot\hat{h} \doteq \|\Delta_kF(x)\|\cdot\hat{h}$ durch

$$\hat{h} = \frac{\|F(x)\|\sqrt{\text{eps}}}{\|\Delta_kF(x)\|} = \frac{\|F(x)\|\sqrt{\text{eps}}}{\|F(x+he_k)-F(x)\|}\cdot h$$

eine größere Störung $\hat{h}$, für die Gleichung $(*)$ näherungsweise erfüllt ist. Im MATLAB-Code ist die `for`-Schleife zu ersetzen durch

```
for k=1:n
    h=max(1,abs(x(k)))*h0
    Df(:,k)=(f(x+h*E(:,k))-y)/h;

    if ( norm(Df(:,k))*h < norm(y)*h0 )
        hneu=norm(y)*h0/norm(Df(:,k))
        Df(:,k)=(f(x+hneu*E(:,k))-y)/hneu;
    end
end
```

b) Aus der Definition der Jacobi-Matrix folgt unmittelbar: wenn T_F tridiagonal ist, ist es auch die Jacobi-Matrix DF.
Wir betrachten nun ein Beispiel: Sei $n=9$ und T_F tridiagonal, dann ist

$$F(x) = \begin{pmatrix} F_1(x_1,x_2) \\ F_2(x_1,x_2,x_3) \\ F_3(x_2,x_3,x_4) \\ F_4(x_3,x_4,x_5) \\ \vdots \\ F_8(x_7,x_8,x_9) \\ F_9(x_8,x_9) \end{pmatrix}, \qquad DF(x) = \begin{pmatrix} \star & \star & & & & \\ \star & \star & \star & & 0 & \\ & \star & \star & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \ddots & \ddots & \star \\ & & & & \star & \star \end{pmatrix}.$$

Die Idee zur effizienten Berechnung von $DF(x)$:
x_1 kommt nur in F_1 und F_2 vor, x_2 kommt nur in F_1, F_2 und F_3 vor, x_3 kommt nur in F_2, F_3 und F_4 vor, usw., d.h. allgemein x_i nur in F_{i-1}, F_i, F_{i+1}. Deshalb störe *simultan* x_1, x_4, x_7 sowie x_2, x_5, x_8 und x_3, x_6, x_9. Dies entspricht je einer Funktionsauswertung von F. Dazu kommt die Berechnung von $Z = F(x)$.
Der Algorithmus hat dann folgende Gestalt:

```
h0=sqrt(eps);
y=f(x);
n=length(y);

Df=zeros(n);
E=eye(n);

h=max(1,abs(x))*h0;
for j=1:3
    x1=x; x1(j:3:n)=x(j:3:n)+h(j:3:n);
    y1(:,j)=f(x1);
end

for k=1:n
    k1=max(k-1,1); kn=min(k+1,n);
    Df(k1:kn,k)=(y1(k1:kn,mod(k-1,3)+1) - y(k1:kn))/h(k);
end
```

Lösungsvorschlag zu Aufgabe 2 *QR-Zerlegung*

a) Man partitioniert die Matrix $\tilde{Q}$ in $\tilde{Q} = (Q\ q)$. Ausmultiplizieren von $\tilde{Q}\tilde{R} = (A\ b)$ ergbit

$$QR = A, \quad Qr + \rho q = b$$

Mulitpliziert man die rechte Gleichung mit Q^T, so ergibt sich wegen $Q^TQ = I \in \mathbb{R}^{n \times x}$ und $Q^Tq = 0$

$$r = Q^T b$$

Laut Vorlesung gilt für die Lösung x des linearen Ausgleichproblems

$$Rx = Q^T b = r,$$

also

$$b - Ax = b - QRx = b - Qr = \rho q.$$

Da $||q||_2 = 1$ und $\rho \geq 0$ gilt also

$$||b - Ax||_2 = \rho.$$

b)
```
function x = LinAusgleich(A,b)

[Q,R] = qr([A b],0);
    x = R(1:n,1:n)\R(1:n,n+1)


return
```