Universität zu Köln

Dr. Martin Lanser

Wintersemester 2019/2020

06. November 2019

4.Übung zur Algorithmischen Mathematik und Programmieren

Hinweis: Beachten Sie alle Abgabeformalitäten, die auf dem ersten Übungszettel angegeben wurden.

Aufgabe 1: (3 + 3 + 2 = 8 Punkte)
Das Newton-Verfahren ist nicht immer problemlos anwendbar, z. B. wenn die Funktion f, deren Nullstelle gesucht wird, nicht überall stetig differenzierbar ist. Betrachten wir dazu die nichtlineare Gleichung

$$f(x) = \ln(x) + 1.$$

a) Führen Sie per Hand oder mit MATLAB Programm die ersten zwei Iterationen des Newton-Verfahrens zur Bestimmung einer Nullstelle von f aus; einmal den Startwert $x^{(0)} = 1$ und einmal mit $x^{(0)} = 0.5$. Was fällt Ihnen auf?

b) Das **Sekantenverfahren** vermeidet Schwierigkeiten mit der Ableitung $f'(x)$ durch eine Approximation mit Hilfe des Differenzenquotienten

$$f'(x^{(k)}) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}.$$

Leiten Sie das eindimensionale Sekantenverfahren her, indem Sie im eindimensionalen Newton-Verfahren die Ableitung durch oben genannte Approximation ersetzen. Geben Sie das Sekantenverfahren in Form eines **Pseudocodes** an.

c) Geben Sie eine grafische Interpretation des Sekantenverfahrens mittels einer Skizze an. Orientieren Sie sich dazu an der Skizze zum Newton-Verfahren aus der Vorlesung.

Aufgabe 2: (6 + 2 = 8 Punkte)
Es sei das nichtlineare Gleichungssystem

$$F(x) = F(x_1, x_2) = \begin{pmatrix} x_1^3 + x_2^3 - 4 \\ x_1^3 - x_2^3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

gegeben.

a) Zeigen Sie, dass das Newton-Verfahren zur Lösung von $F(x) = 0$ für jeden Startvektor $(x_1^{(0)}, x_2^{(0)})^T \in [1,2] \times [1,2] \subset \mathbb{R}^2$ konvergiert.

b) Führen Sie zwei Iterationsschritte des Newton-Verfahrens durch, ausgehend vom Startvektor $(x_1^{(0)}, x_2^{(0)})^T = (1,1)^T$.

Aufgabe 3 (8 Punkte)
Untersuchen Sie das Konvergenzverhalten des Newton-Verfahrens aus Programmieraufgabe 1 von diesem Blatt mit dem dort gegebenen f. Nutzen Sie dazu folgende MATLAB Funktion zur Visualisierung:

```
%Diese Funktion gibt ein Bild der Konvergenzgeschwindigkeit aus
function draw_nits(xrange,yrange,tol)

ImageMatrix=zeros(length(xrange),length(yrange));
ns=zeros(length(xrange)*length(yrange),2);

for i=1:length(xrange)
    for j=1:length(yrange)
        %[x,y] soll nun immer ein Punkt im von xrange und yrange
        %definierten Gitter sein

        x=xrange(i);
        y=yrange(j);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %AUFRUF IHRER NEWTON FKT  %
        %kann z.B wie folgt heissen%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%
        [x_null,n_it]=newton2d([x;y],tol);

        %merke Nullstelle
        ns((i-1)*length(yrange)+j,:)=x_null';

        %Schreibe bis zur Konvergenz benoetigte Anzahl in Matrix
        ImageMatrix(length(yrange)-j+1,i)=n_it;
    end
end

clf;

%Setze Farbtabelle und plotte
colormap(hot);
image(ImageMatrix);

axis off;

%plotte noch die Nullstellen
figure(2);
hold on;
for i=1:length(xrange)*length(yrange)
   plot(ns(i,1),ns(i,2),'X','MarkerSize', 24);
end

end
```

Rufen Sie die Funktion mit unterschiedlichen Genauigkeiten auf: *draw_nits([-3:0.1:3] , [-3:0.1:3], 1e-1)* und *draw_nits([-3:0.1:3] , [-3:0.1:3], 1e-6)*. Beschreiben Sie das Ergebnis. Welche Aussage haben die beiden erstellten Grafiken? Zoomen Sie auch einmal für beide Genauigkeiten nahe an eine der Nullstellen heran! Sollte Ihr Rechner nicht zu lange dafür benötigen, versuchen Sie einmal mir erhöhter Auflösung *draw_nits([-3:0.01:3] , [-3:0.01:3], 1e-6)*.

Programmieraufgabe 1 (10 Punkte)
Sei

$$f : \mathbb{R}^2 \to \mathbb{R}^2, \quad f(x,y) = \begin{pmatrix} x^3 - 3xy^2 \\ -y^3 + 3x^2y + 1 \end{pmatrix} .$$

Implementieren Sie ein 2D-Newton-Verfahren zur Berechnung der Nullstellen von f wie folgt:

(i) Implementieren Sie die Funktion f und die Jacobimatrix $Df : \mathbb{R}^2 \to \mathbb{M}^{2\times 2}$ von f jeweils in einer eigenen .m-Dateien.

(ii) Werten Sie Ihre f- und Df-Funktionen in den Punkten $[0; 0]$, $[0; 1]$ und $[1; 0]$ aus. (Bitte Ergebnisse ausdrucken). Sie sollten jeweils einen 2×1-Vektor, bzw. eine 2×2-Matrix erhalten.

(iii) Implementieren Sie ein 2D-Newtonverfahren zur Approximation der Nullstellen von f. Ihr Verfahren sollte als Eingabeparameter einen Startvektor $x^{(0)}$ und eine Abbruchgenauigkeit *tol* erhalten, die Nullstelle x^* und die Anzahl der benötigten Iterationen *nits* ausgeben.

Abgabedatum: 14.11.2019, 12 Uhr, Abgabe der Ausarbeitung in den Kasten im MI.