Universität zu Köln

Prof. Dr. A. Klawonn
Dipl.-Math. M. Lanser

Sommersemester 2012

26. April 2012

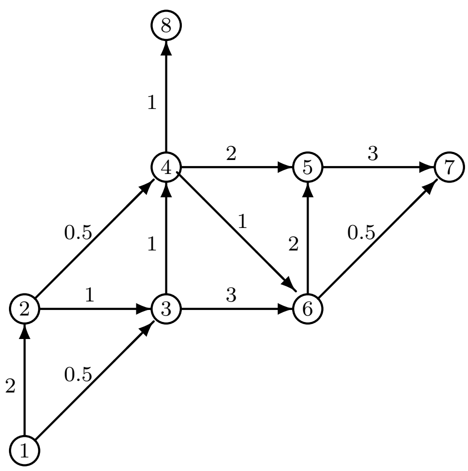
3.Übung zur Numerischen Mathematik I

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

Aufgabe 1: (8 + 4 = 12 Punkte)
Betrachten Sie den unten stehenden, gerichteten Graphen. Fassen Sie diesen Graphen als schematische Darstellung eines elektrischen Netzwerks auf. An den Kanten ist die jeweilige elektrische Leitfähigkeit angegeben. An Knoten 1 befinde sich eine Quelle, die 2 Ampère in das Netzwerk einspeist. An den Knoten 7 und 8 befinden sich Senken, die jeweils 1 Ampère Strom aufnehmen.

(i) Stellen Sie analog zur Vorlesung die Matrizen A und C sowie den Vektor f des Systems $A^TCA = f$ auf, das dieses Netzwerk beschreibt. Geben Sie dazu eine Skizze des Systems mit ab, in der Sie die von Ihnen gewählte Nummerierung der Kanten angeben $(y_1, \ldots, y_{12})$.

(ii) Berechnen Sie die Potentiale indem Sie den \-Operator aus Matlab zum Lösen des Gleichungssystems verwenden. Geben Sie dabei A und C vor und berechnen Sie das System mit dem Computer. Erden Sie dabei den Knoten 2.
Hinweis: Um die j-te Zeile der Matrix B zu streichen, können Sie den Befehl "$B(j,:) = [\ ];$" verwenden. Analog können Sie die j-te Spalte streichen, indem Sie "$B(:,j) = [\ ];$" verwenden.



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.

(i) 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.

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

Aufgabe 3 (Programmieraufgabe) (5 + 3 + 6 + 2 + 4 = 20 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-Newtonverfahren zur Berechnung der der Nullstellen von f wie folgt:

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

(ii) Werten Sie Ihre f- und J_f-Funktionen in den Punkten $[0;0]$, $[0;1]$ und $[1;0]$ aus. (Bitte diary 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 *epsilon* erhalten und die Nullstelle x^* und die Anzahl der benötigten Iterationen *nits* ausgeben.

(iv) Die folgende Funktion soll Ihnen als Ausgangspunkt dienen, um ein Bild vom Konvergenzverhalten Ihres Newtonverfahrens zu zeichnen.

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

ImageMatrix = zeros(length(yrange), length(xrange));

for i=1:length(xrange)
    for j=1:length(yrange)
        % [x;y] ist Gitterpunkt in einem Gitter mit den
        % Raendern xrange und yrange.
        x=xrange(i);
```

```
        y=yrange(j);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Hier wird Ihre newton2d-Funktion aufgerufen! %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        [x_star, n_it] = newton2d([x;y], 1e-6);

        % Schreibe die Anzahl der bis zur Konvergenz
        % (oder bis zum Abbruch) benoetigten Iterationen in
        % eine Matrix.

        ImageMatrix(length(yrange)-j+1, i) = n_it;
    end
end

clf; % Loesche momentan aktive Grafik

% Setze aktive Farbtabelle
colormap(gray); % Schwarz = wenige Iterationen
                % Weiss = viele Iterationen
% colormap(hot); % Huebschere Farbtabelle
image(ImageMatrix); % Stelle die ImageMatrix als Bild dar

axis off; % Schalte Achsen aus

end
```

(v) Rufen Sie die Funktion *draw_nit* mit unterschiedlichen Genauigkeiten wie folgt auf:
draw_nit([-3:1:3] , [-3:1:3]) und *draw_nit([-3:0.01:3] , [-3:0.01:3])*.
Die zweite Variante könnte etwas Zeit brauchen. Bitte drucken Sie die entstehende Grafik aus und geben Sie sie mit ab. Markieren Sie in der Grafik die Punkte, an denen Sie Nullstellen vermuten. Sie können natürlich auch die Ergebnisse Ihres Newton-Verfahrens nutzen, um die Nullstellen zu finden.

Abgabedatum: 03.05.2012, 10 Uhr, Abgabe der Ausarbeitung in den Kasten im Keller des MI.