3. Projekt: Newton-Verfahren

Abgabe bis spätestens 20.1.2019 in Moodle, Vorführung der Programme am 21./22.1. in der Übung, gerne schon am 14./15.1. .

Aufgabenstellung:

1. Implementieren Sie das Newton-Verfahren zur Bestimmung der Nullstelle einer Funktion $f : \mathbb{R}^n \supset U \to \mathbb{R}^n$ als eine MATLAB-Funktion. Dabei sollen die Funktion f und die erste Ableitung (Jacobi-Matrix) J_f – also eine matrixwertige Funktion – als Parameter übergeben werden (als sogenannte *function handle*, analog zu einem Funktions-Pointer in C), sowie ein Startvektor, die maximale Anzahl der Iterationen, und eine Fehlertoleranz. Zurückgegeben werden soll die approximative Lösung sowie die Anzahl der benötigten Iterationen.

   - Verwenden Sie zur Lösung der linearen Gleichungssysteme Ihren Löser aus dem 1. Projekt.
   - Wählen Sie als Fehlerschätzer die euklidische Norm des Zuwachses $\|x^{(k+1)} - x^{(k)}\|_2$.
   - Testen Sie die Implementierung mit der folgenden Funktion $f : \mathbb{R}^2 \to \mathbb{R}^2$,

   $$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2 - 4 \\ x_2 e^{-x_1} - 2 \end{pmatrix} \tag{1}$$

   Wählen Sie geeignete Startvektoren $x^{(0)} \in \mathbb{R}^2$ um beide Nullstellen zu finden (siehe Vorlesung, veranschaulichen Sie sich die Nullstellen als Schnittpunkte der durch die beiden nichtlinearen Gleichungen definierten Kurven; erstellen Sie dazu eine geeignete Grafik). Wie viele Iterationen benötigen Sie bei unterschiedlichen Toleranzen von $10^{-2} \dots 10^{-10}$? Finden sie auch Startvektoren, für die das Newtonverfahren nicht konvergiert?

2. **Newtonfraktale:** Hat eine Funktion mehrere Nullstellen, so bezeichnet man als den *Einzugsbereich* einer Nullstelle x^* die Menge aller Startvektoren $x^{(0)}$, für die das Newton-Verfahren gegen x^* konvergiert. Schon für einfache Funktionen ist es schwer vorhersagbar, welcher Startvektor bei konvergentem Newton-Verfahren zu welcher Nullstelle führt (erstaunlicherweise eben nicht unbedingt zu der nächstgelegenen). Die Einzugsmengen sind oft sogenannte *Fraktale*. Untersuchen Sie folgendes Beispiel: Berechnung der 3-ten Einheitswurzeln von 1, d.h. die Nullstellen (in $\mathbb{C}$) der Gleichung $z^3 = 1$ (also $1, -\frac{1}{2} + \frac{\sqrt{3}}{2}, -\frac{1}{2} - \frac{\sqrt{3}}{2}$ mit $i^2 = -1$). Gehen Sie dabei wie folgt vor (auch wenn es wesentlich effizientere Methoden gibt, wenn man im Komplexen rechnet):

   - Überlegen Sie, dass sich das Problem als Nullstellenproblem in $\mathbb{R}^2$ (komplexe Ebene; $z = x + iy \equiv (x, y)^T \in \mathbb{R}^2$) wie folgt formulieren lässt:

   $$f((x,y)^T) = \begin{pmatrix} x^3 - 3xy^2 - 1 \\ -y^3 + 3x^2 y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

   und berechnen Sie die Jacobi-Matrix J_f; implementieren Sie f und J_f als Matlab-Funktionen.
   - Legen Sie ein äquidistantes Gitter von Punkten $(x_i, y_j) \in [-1,1] \times [-1,1]$, $i, j = 1, 2, \dots, n$ fest, die als Startwerte für die Newton-Iteration verwendet werden sollen.

- Führen Sie die Newton-Iteration (mit dem von Ihnen implementierten Programm) für jeden dieser Punkte (x_i, y_j) als Startwert durch und belegen Sie in einer $n \times n$-Matrix M den entsprechenden Eintrag M_{ij} mit einer ganzen Zahl wie folgt: $M_{ij} = 0$, falls keine Konvergenz, $M_{ij} = 1$, falls Konvergenz gegen die 1. Nullstelle, $M_{ij} = 2$, falls Konvergenz gegen die 2. Nullstelle, usw. .
- Die Werte der Matrix M werden jetzt als Farb-Code für ein kleines eingefärbtes Quadrat (Pixel = Picture element) am Punkt (x_i, y_j) interpretiert. Dies geschieht in Matlab zum Beispiel mit dem Befehl `image(x,y,M)` .

  Beispiel-Code:

  ```
  nStart = 10;               % Anzahl der Startwerte in einer Richtung
  x = linspace(1,-1,nStart); % nStart aequidistante Punkte, wie [-1:2/(nStart-1):1]
  y = linspace(-1,1,nStart);

  ...

  root   = newtonSystemSimple(@fz3,@dfz3,[x(i);y(j)], tol, maxIt);
  ...
  M(j,i) = ...    % 0, 1, 2, oder 3 ; x nach rechts, also Spaltenindex ...
  ...
  M = 20*M;       % Umskalieren, da es 64 Farb-Codes 0,... 63 gibt
  image(x,y,M)
  colorbar   % Legende fuer die Farben
  ```

- Erstellen Sie ein paar „bunte Bilder“: unterschiedliche Auflösung (Anzahl der Startwerte), Zoom in einen interessanten Bereich ... . Vorsicht: nicht zu viele Startwerte (maximal ca. 500×500), wenig Iterationen, `tol = 1.e-2` ausreichend. Interpretieren Sie das Ergebnis (lokale Konvergenz, Einzugsgebiete ...).
- weitere Hinweise in der Übung, informieren Sie sich in der Matlab-Help (oder Octave-Doku) über die verwendeten Befehle / Funktionen.

Projektbericht und Benotung

- Siehe 1./2. Projekt
- Dokumentieren Sie auch Ihre Ergebnisse und Überlegungen; gehen Sie insbesondere in Ihrem Bericht auf die gestellten Fragen ein
- Verwenden Sie einen mathematischen Formelsatz (z.B. Latex Formelsatz, bzw. Word/OpenOffice Formeleditor)
- Beschriften Sie alle Grafiken vollständig (Achsen, Legende, etc.); Bilder/Grafiken in den Bericht
- Der Bericht muss als PDF abgegeben werden!
- 2 Seiten