FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Fachbereich Elektrotechnik
und Informationstechnik

Prof. Georg Hoever

SS 2022

2. Praktikum zur Höhere Mathematik 2 für (Wirtschafts-)Informatik

Ziel dieses Praktikums ist eine Implementierung des Newtonverfahrens.

Dabei wird die Klasse `CMyVektor` des ersten Praktikums weiter verwendet.

1. Aufgabe

Um bequem mit Matrizen $A \in \mathbb{R}^{m\times n}$ arbeiten zu können, soll eine Klasse `CMyMatrix` implementiert werden:

- Implementieren Sie die Informationen über die Dimensionen und die Einträge als private Attribute.
- Implementieren Sie (public-)Methoden, um
  - eine Matrix einer bestimmten Dimension anzulegen,
  - eine bestimmte Komponente der Matrix zu setzen,
  - eine bestimme Komponente der Matrix zurückzugeben.
- Implementieren Sie eine (public-)Methode `CMyMatrix invers()`, die
  - bei einer 2×2-Matrix A mit $\det A \neq 0$ die Inverse A^{-1} mittels der Formel

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

    liefert,
  - ansonsten eine Fehlermeldung liefert und zum Programmabbruch führt.

Implementieren Sie ferner eine überladene Operator-Funktion

```
CMyVektor operator*(CMyMatrix A, CMyVektor x)
```

die eine Matrix-Vektor-Multiplikation realisiert.

2. Aufgabe

Zu einer Funktion $f : \mathbb{R}^m \to \mathbb{R}^n$ soll die Jacobi-Matrix an einer Stelle $\vec{x} \in \mathbb{R}^m$ berechnet werden:

- Implementieren Sie eine Funktionen

  `CMyMatrix jacobi(CMyVektor x, CMyVektor (*funktion)(CMyVektor x))`,

  der man im ersten Parameter die Stelle $\vec{x}$ und im zweiten Parameter die Funktion f als Funktionspointer übergibt, und die die Jacobi-Matrix $J_f = f'(\vec{x})$ numerisch durch entsprechende Differenzenquotienten zu festem $h = 10^{-4}$ berechnet.

- Testen Sie die Berechnung an

$$f : \mathbb{R}^4 \to \mathbb{R}^3, \quad f(x_1, x_2, x_3, x_4) \;=\; \begin{pmatrix} x_1 x_2 \; \mathrm{e}^{x_3} \\ x_2 x_3 x_4 \\ x_4 \end{pmatrix}, \qquad \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 3 \end{pmatrix}.$$

3. Aufgabe

Zu einer Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$ soll ausgehend von einer Stelle $\vec{x} \in \mathbb{R}^n$ das Newtonverfahren zur Bestimmung einer Nullstelle von f durchgeführt werden.

- Implementieren Sie das entsprechende Verfahren unter Benutzung der Klassen `CMyVektor` und `CMyMatrix`.
- Entsprechend der Implementierung der `invers`-Methode braucht das Verfahren nur für den Fall $n = 2$ zu funktionieren.
- Nutzen Sie wieder einen Funktions-Pointer zur Angabe der Funktion f.
- Führen Sie die Newton-Iteration durch, bis $\|f(\vec{x})\| < 10^{-5}$ ist, oder bis 50 Schritte gemacht wurden.
- Testen Sie das Verfahren an

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

  mit Startwert $\left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$.

Für Interessierte:

Die Suche nach einer lokalen Extremstelle einer Funktion $f : \mathbb{R}^n \to \mathbb{R}$ kann man auffassen als Suche nach einer Nullstelle von $g : \mathbb{R}^n \to \mathbb{R}^n,\ g(\vec{x}) = \operatorname{grad} f(\vec{x})$.

- Schreiben Sie eine Funktion, die den Gradienten zu der Testfunktion $f : \mathbb{R}^2 \to \mathbb{R}$ von Praktikum 1 zurückgibt, und wenden Sie darauf das Newtonverfahren an.

  Testen Sie verschiedene Startwerte und vergleichen Sie die Ergebnisse (Konvergenzpunkt und dazu nötige Schrittanzahl) des Gradientenverfahrens und des Newton-Verfahrens.

  Sie werden feststellen, dass das Newton-Verfahren beim Startwert $(0.2, -2.1)^T$, der beim ersten Praktikum angegeben ist, zu einem völlig anderen Punkt konvergiert. Das liegt daran, dass man noch recht weit von der Optimalstelle, die man beim Gradientenverfahren erreicht, entfernt ist, und das Newtonverfahren sich nach einem Schritt weit von der Stelle entfernt. Das Funktionsgebirge hat viele Gipfel und Täler (betrachten Sie sich das Gebirge z.B. mittels der Geogebra-Visualisierungen, die auf meiner Homepage verlinkt sind!), und das Newtonverfahren konvergiert dann gegen eine der Extremstellen irgendwo.

  Wenn Sie als Startwert allerdings z.B. den Wert nehmen, der im Gradientenverfahren nach dem fünften Schritt erreicht ist, braucht das Newton-Verfahren nur noch drei Schritte bis zur Optimalstelle, während das Gradientenverfahren noch 15 Schritte benötigt.

  Entsprechend ist es manchmal sinnvoll, robuste aber langsame mit schnellen aber nicht sicheren Verfahren zu mischen: Mit dem robusten Verfahren berechnet man erste Annäherungen, und wenn man dann „nah genug“ ist, wechselt man auf das schnelle Verfahren.

- Wenn Sie die Funktionen $s(a, m)$ bzw. $s(c, \lambda)$ zur Parameteranpassung aus dem Teil „Für Interessierte“ des ersten Praktikums implementiert haben, können Sie auch hierzu jeweils eine Funktion schreiben, die den jeweiligen Gradienten zurückgibt, und darauf das Newtonverfahren anwenden.

  Bei der exponentiellen Anpassung ($s(c, \lambda)$) können Sie beobachten, dass Sie bei Startwerten, die nicht zu weit von der Optimalstelle entfernt sind, mit dem Newtonverfahren die Optimalstelle deutlich schneller erreichen als mit dem Gradientenverfahren.

  Bei der linearen Anpassung ($s(a, m)$) können Sie beobachten, dass Sie mit dem Newtonverfahren bei vernünftigen Startwerten sogar schon nach nur einem Schritt die Optimalstelle erreichen.

Das liegt daran, dass die Anpassungsfunktion

$$s(a,m) \;=\; \sum_{k=1}^{N}(f(x_k)-y_k)^2 \quad \text{zu} \quad f(x)=mx+a$$

eine in den Parametern a und m quadratische Funktion ist. Der Gradient ist damit eine lineare Funktion. Genauso wie man bei der Anwendung des eindimensionalen Newtonverfahrens auf eine Gerade deren Nullstelle nach einem Schritt erreicht (da das Newtonverfahren ja genau die Nullstelle der linearen Näherung bestimmt, die bei einer Geraden mit der Funktion übereinstimmt), ist es auch im Mehrdimensionalen: Das Newtonverfahren bestimmt die Nullstelle der linearen Näherung, die bei einer linearen Funktion mit der Funktion übereinstimmt.