1. Projekt: Nichtlineare Ausgleichsprobleme

Abgabe der Implementierung: bis 11.5.2019 in moodle (23:55); Abnahme am 15.5. bzw. 16.5. in der Übung (Ausführlichkeit nach Sichtung der Abgaben auf Moodle, insbesondere, wenn ich noch keinen Eindruck von ihrem erfolgreichen Arbeiten an dem Projekt in den Übungen bekommen habe).

Aufgabenstellung:

(a) Implementieren Sie in MATLAB das Gauß-Newton-Verfahren zur Lösung des nichtlinearen Ausgleichsproblems

$$\|F(x)\|_2 = \min$$

für eine differenzierbare Funktion $F : \mathbb{R}^n \supset U \to \mathbb{R}^m$ mit $m \geq n$. Dabei sollen die auftretenden linearen Ausgleichsprobleme mit Hilfe der QR-Zerlegung gelöst werden. Es kann vorausgesetzt werden, dass die Jacobi-Matrix der Funktion F stets vollen Rang hat, also $\mathrm{Rang}(J_F(x)) = n$.

(b) Testen Sie ihre Implementierung an einem einfachen nichtlinearen (oder auch linearen) Ausgleichsproblem (z.B. $F_i(x) = y_i - x_1 \sin(x_2 t_i)$, siehe Vorlesung).

(c) Lösen Sie mit Ihrer Implementierung das folgende nichtlineare Ausgleichsproblem: In dem Forschungsartikel *Growth Patterns and Scaling Laws Governing AIDS Epidemic in Brazilian Cities* von F.J. Antonio et al (2014), siehe `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4207789/#pone.0111015` wird versucht, einige Aussagen über die Ausbreitung von AIDS in Brasilien in den Jahren 1980-2012 zu treffen. Insbesondere werden die vorhandenen Daten der Anzahl von neuen AIDS-Diagnosen pro Jahr durch geeignete kontinuierliche Wachstumsfunktionen beschrieben. Ihre Aufgabe ist es, die Abbildung 2(a) aus dem Artikel zu reproduzieren. Wie in wissenschaftlichen Veröffentlichungen nicht unüblich, werden die verwendeten Daten öffentlich zur Verfügung gestellt. In der Datei `DataAIDSBrazil.csv` sind diese auf Moodle zu finden. Sie können diese Datei aber auch direkt auf der Webseite des Artikels unter *Associated data* herunterladen. Verwenden Sie die beiden folgenden Wachstumsfunktionen als Modellfunktionen mit jeweils 3 zu bestimmenden Parametern:

- Logistische Funktion (Sigmoidfunktion):

  $$y(t) = \frac{x_1}{1 + x_2 e^{-x_3(t-t_0)}},$$

  Dies ist die allgemeine Lösung der logistischen DGL $\dot{y} = \lambda y \left(1 - \frac{y}{K}\right)$, mit $x_1 = K$, $x_2 = C$, $x_3 = \lambda$.

- Gompertzfunktion

  $$y(t) = x_1 e^{-x_2 e^{(-x_3 * (t-t_0))}}$$

  Auch diese Funktion wird für die Beschreibung von Populationsdynamiken verwendet, z.B. auch Tumorwachstum.

Im vorliegenden Fall ist $t_0 = 1980$. Die Parametrisierung ist etwas anders als die in dem oben zitierten Artikel. In der Datendatei sind die Zahlen von neuen AIDS-Diagnosen pro Jahr in über 5000 Städten in Brasilien für den Zeitraum von 1980-2012 zusammengestellt. Außerdem befinden sich in der ersten Zeile die Jahreszahlen und in den ersten fünf Spalten Informationen zu den Städten (in Matlab können sie die Daten direkt mit Doppelklick anschauen, für einen

Import in Excel verwendet man die Auswahl *Daten* und dann *Externe Daten abrufen*. Auf Moodle liegt aber auch ein Excelsheet.)

Mit den folgenden Befehlen können Sie die numerischen Daten (ab Spalte 6 und Zeile 2) einlesen:

```
data = csvread('DataAidsBrazil.csv',1,5);
yi = sum(data,1);  % ueber alle Staedte summieren
ti = [1980:2012];
```

Wir betrachten in diesem Projekt nur die jährlichen Gesamtzahlen der AIDS-Diagnosen, daher die Aufsummation. Anmerkung: es ist am einfachsten, mit einer neuen Zeitvariablen $\tau = t - t_0$ zu arbeiten, also mit den Jahreszahlen $[0 - 32]$ und $\tau_0 = 0$ alle Berechnungen durchzuführen. Erst bei der Visualisierung der Daten und Kurven, müssen diese dann wieder nach rechts verschoben werden ins Intervall $[1980 - 2012]$.

Hinweise zur Implementierung

(i) Implementieren Sie das Gauß-Newton-Verfahren als eine MATLAB-Funktion, die als Eingabe einen Startvektor für die Iteration, jeweils einen Function Handle für die Funktion F, die ausgewertet werden muss, sowie für deren Jacobi-Matrix J_F, eine Fehler-Toleranz und die maximale Anzahl von Iterationen erwartet, und die berechnete approximative Lösung sowie die Anzahl der benötigten Iterationen zurückliefert. Als Fehlerschätzer, der gegen die übergebene Toleranz getestet wird, soll die 2-Norm des Residuums $J_F(x)^T F(x)$ verwendet werden (warum ?).

(ii) Verwenden Sie zunächst die von Matlab/Octave bereitgestellte Funktion für die QR-Zerlegung und erst, wenn alle Test erfolgreich waren, in einem zweiten Schritt ihre eigene Implementierung der QR-Zerlegung mit Householder-Reflexionen.

(iii) Implementieren Sie für die Aufgabenteile (b) und (c) die entsprechenden Funktionen, die die Auswertung von F und von J_F liefern. Schreiben Sie außerdem jeweils ein MATLAB-Skript, in dem das nichtlineare Ausgleichsproblem (b) bzw. (c) gelöst und die erhaltene Lösung zusammen mit den Datenpunkten grafisch dargestellt wird. Verwenden Sie plausible Startwerte. Für den Teil (c) geht man dabei wie folgt vor: Plotten sie zunächst die Datenpunkte und denken Sie sich eine geeignete S-Kurve, die die Datenpunkte gut approximiert. Sie können dann für diese S-Kurve ungefähr angeben:

- Zeitpunkt der größten Steigung $\tau_{\max}$
- Größte Steigung (ist am Wendepunkt): s
- Asymptotischer Wert für $\tau \to \infty$: $y_{\max}$.

Für die beiden betrachteten Modellfunktionen kann man Nachrechnen, dass für die exakte Kurve gelten würde:

- Logistische Funktion:

$$x_1 = y_{\max}, \quad x_3 = \frac{4s}{y_{\max}}, \quad x_2 = e^{x_3 \tau_{\max}}.$$

Aus den geschätzen Werten für $\tau_{\max}$, s und $y_{\max}$ erhalten Sie damit Startwerte für die drei Paramter $x = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})^T$

- Grompertzfunktion:

$$x_1 = y_{\max}, \quad x_3 = \frac{e^1\, s}{x_1}, \quad x_2 = e^{x_3 \tau_{\max}}.$$

Abnahmekriterien (Implementierung)

Zur Erinnerung: Die erfolgreiche Bearbeitung der drei Programmieraufgaben ist Voraussetzung für die Teilnahme an der Klausur. Die Implementierung kann alleine oder zu zweit erfolgen. Das Programm muss in der Übung vorgeführt und seine Funktionsweise erklärt werden und vorher auf Moodle hochgeladen werden (als zip-Datei `nachname_vorname.zip`). Bitte geben Sie im Testskript am Anfang als Kommentar an, ob - bzw. mit wem - Sie das Programm zusammen erstellt haben.
Es müssen alle Quellen (auch von Teilen des Programm-Codes) angegeben werden. Die Implementierung muss im wesentlichen eigenständig erfolgen, d.h. es sind nur minimale Übernahmen von anderen erlaubt. Insbesondere ist es nicht zulässig, ein anderes nicht selbsterstelltes Programm beziehungsweise Programmteile lediglich anzupassen. Wenn Sie Funktionsweises ihres Programmes nicht erklären können, gehe ich davon aus, dass Sie diese nicht eigenständig erstellt haben.
Ein von mir bemerkter Betrugsversuch bedeutet, dass keine weitere Möglichkeit zur Projektabgabe besteht - sie können also dann weder an dieser noch einer der anderen Projektaufgaben und auch nicht an der Klausur teilnehmen. Als Betrugsversuch gilt die Übernahme - auch in Teilen - von nicht selbst erstelltem Programmiercode. Ich werde dieses Semester Plagiate wesentlich ernster nehmen als in Numerik I.
Also fangen Sie bald an :-) . Es ist ein interessantes Projekt! Ich fand es jedenfalls interessant, als ich es vor mehreren Tagen gemacht habe und überprüft habe, ob auch wirklich alles mit den Daten funktioniert. Insofern ist es brandneu an der Beuth - was den Anwendungsfall angeht. Austausch und Diskussion untereinander ist natürlich erwünscht. Aber kein Einseitiges Übernehmen der Arbeit von anderen! Fragen in den Übungen sind willkommen.