Serie 11

Abgabedatum: Fr. 19.05, in den Übungsgruppen

Forum: https://forum.math.ethz.ch/c/spring-23/numerische-methoden-phys/

Webpage: https://metaphor.ethz.ch/x/2023/fs/401-2664-00L/

1. *Blitzortung*

Um während eines Gewitters in der Schweiz die Blitzeinschläge zu lokalisieren sind in 13 grossen Städten Detektoren verteilt. Jeder Blitz erzeugt eine breitbandige Störung im elektromagnetischen Feld der Erde. Diese *sferic* genannten Signale werden von den Detektoren erkannt und an eine zentrale Stelle weitergeleitet. Anhand der unterschiedlichen Empfangszeiten kann mit minimal 4 Detektoren die Position der elektrischen Entladung bestimmt werden.

Die Koordinaten der Detektoren $\underline{x}_i, i = 1 \ldots 13$ sind exakt bekannt. Die Ankunftszeiten $t_i, i = 1 \ldots 13$ sind mit Messfehlern von ungefähr 10 Mikrosekunden behaftet. Die Positionen $\underline{x}_i$ sowie die Ankunftszeiten t_i liegen in SI-Einheiten vor. Gesucht ist für jeden Impuls die Position $\underline{x}$ wo der Blitz eingeschlagen hat.

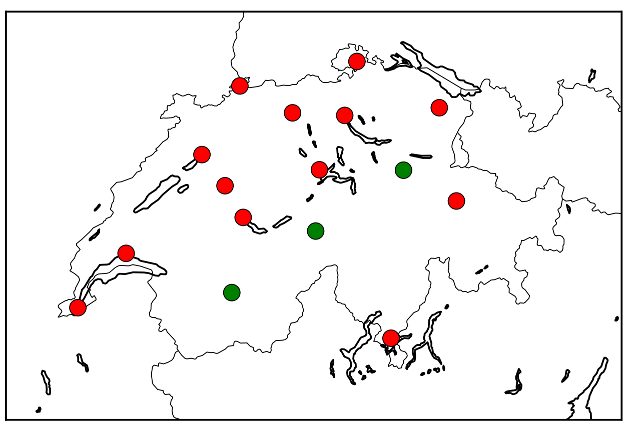

Abbildung 1: Messstationen (rot), Blitzereignisse (grün). (Plot mit Matplotlib Basemap generiert.)

Für jedes **Paar** von Detektoren gilt, dass der Abstand zum Event $\|\underline{x} - \underline{x}_i\|$ gleich der Differenz der Ankunftszeiten mal die Lichtgeschwindigkeit c ist

$$\|\underline{x} - \underline{x}_i\| - \|\underline{x} - \underline{x}_j\| = c\,(t_i - t_j) \qquad \text{für alle } \textbf{Paare } i, j, \tag{1}$$

wobei $c = 299792458.0$ m/s.

Bitte wenden!

a) Was für eine Methode würden Sie verwenden um das überbestimmte Gleichungssystem (1) zu lösen?

b) Implementieren Sie eine geeignete Methode zur Lösung von (1) in `Python`. Verwenden Sie $\underline{x}_0 = [600000, 100000]$ als Startwert für $\underline{x}$. Die Koordinaten (in Meter) sowie die Messergebnisse (Laufzeiten in Sekunden) befinden sich im Template.

Hinweis: Verwenden Sie das Template `blitzortung_template.py`.

2. *Morse Potential aus Messwerten*

Ein Experimentator modelliert ein Molekül mit dem Morse Potential

$$M(x) := D\left(\exp(-2\beta(x-r)) - 2\exp(\beta(x-r))\right),$$

mit Parametern $D > 0$, $\beta > 0$ und $r > 0$. Die fehlerbehafteten Messungen (x_i, M_i) für Quecksilber liegen uns vor. Im Folgenden wollen wir daraus Werte für die Parameter mit Hilfe der nichtlinearen Ausgleichsrechnung

$$D, \beta, r = \operatorname{argmin}_{D', \beta', r'} \frac{1}{2} \|F(D', \beta', r')\|_2^2$$

mit

$$F(D, \beta, r) := \begin{bmatrix} M(x_0) - M_0 \\ \vdots \\ M(x_N) - M_N \end{bmatrix}$$

bestimmen.

Hinweis: Verwenden Sie das Template `morse_fit_template.py`

a) Implementieren Sie eine Funktion `lstsq` zur Lösung eines *linearen* Ausgleichsproblems $\mathbf{A}\underline{x} = \underline{b}$. Dabei soll die QR-Zerlegung von $\mathbf{A}$ verwendet werden.

b) Implementieren Sie nun den Gauss-Newton Algorithmus in der Funktion `gauss_newton`. Verwenden Sie `lstsq` aus obiger Teilaufgabe zur Lösung der linearen Teilprobleme.

Hinweis: Falls Sie obige Aufgabe nicht gelöst haben, so verwenden Sie notfalls die Funktion `lstsq` aus `numpy.linalg`.

c) Geben Sie die gefundenen Werte der Parameter aus und plotten Sie das Residuum $|M(x_i) - M_i|$ gegen x_i für $i = 0, \ldots, N$.

3. *Ausgleichsrechnung*

Das Template-File `datafittmpl.py` liest die 14-tägigen Mittelwerte der gemeldeten Corona Fälle in der Schweiz zwischen 25.02.2020 und 14.06.2021 im Array `ma`. Gesucht ist eine lineare Kombination von 7 Gauss-Funktionen mit Parameter q_j und s_j

$$g_j(t) = \exp\left(-\frac{(t-q_j)^2}{s_j^2}\right),$$

die die Daten in `ma` im Sinne der kleinsten Quadrate approximiert.

Siehe nächstes Blatt!

a) Wie viele Parameter werden insgesamt optimiert?

b) Handelt es sich um ein lineares oder um ein nicht-lineares Ausgleichsproblem?

c) Ergänzen Sie `datafittmpl.py` mit einer Funktion `g(q,s,t)`, die die Werte einer einzigen solchen Gauss-Funktion berechnet.

d) Implementieren Sie eine Funktion `f(x,t)`, die die Werte einer solchen linearen Kombination berechnet; dabei enthällt `x` alle Parameter des Problems.

e) Implementieren Sie die Zielfunktion `F(x,t,m)`, wobei `x` = Parameter, `t` = Zeitpunkte der Messungen (Tage) und `m =` gemessene Daten sind.

f) Verwenden Sie die Plots der Daten `ma` und der Gauss-Funktionen um geeignete Startwerte für die Parameter in den 7 Wellen zu bestimmen.
Hinweis:
`numpy.concatenate((a,b,c))` *erzeugt ein* `array` *mit den Werten in den* `array`*s* `a, b` *und* `c`.

g) Lösen Sie das Ausgleichsproblem mitttles standard Löser in `scipy`. Plotten Sie die Daten `ma`, die Anfangsapproximation und die gefundene Lösung in einem Bild.

h) Kann man das Problem nur mit 3 solchen Wellen vernünftig lösen?

4. *Parameterbestimmung*

Lösen Sie das Problem der Parameterbestimmung der Reaktionsraten der Reaktionskette $A \to B \to C$, die wir in der Vorlesung besprochen haben.

Verwenden Sie dazu als "echte" Reaktionsraten $1,1$ und $20,1$ und geben Sie die Lösung in minimize mit BFGS, TNC und in least-squares mit TRF, LM aus.