D-MATH
Dr. Vasile Gradinaru
Dr. Adrian Montgomery Ruf



# Serie 12

**Best before:** Di. 26.05. / Mi. 27.05, in den Übungsgruppen

**Koordinatoren:** Adrian Montgomery Ruf, HG G 54.1, `adrian.ruf@sam.math.ethz.ch`

**Webpage:** `http://metaphor.ethz.ch/x/2020/fs/401-1662-10L/#exercises`

**1.** *Polynomfit und die Runge-Funktion*

Die Runge-Funktion ist definiert als:

$$f(x) := \frac{1}{1+x^2}\,. \tag{1}$$

Wir wollen diese Funktion auf dem Interval $[-5, 5]$ mit einem Polynom $P_n(x)$ von Grad $n$ approximieren. Dazu schreiben wir ein lineares Ausgleichsproblem mit $m$ gleichmässig in $[-5, 5]$ verteilten Punkten $\{(x_i, f(x_i))\}_{i=1}^{m}$ wie folgt:

$$\mathbf{A}\underline{c} = \underline{b} \tag{2}$$

wobei $\underline{c}$ die $n + 1$ Koeffizienten des Polynoms $P_n$ sind.

**a)** Finden Sie die Matrix $\mathbf{A}$ und die rechte Seite $\underline{b}$.

**b)** Lösen Sie das lineare Ausgleichsproblem für Polynome mit Grad $2 \leq n \leq 14$ und jeweils $m = 20$ und $m = 40$.

*Hinweis:* Diese Aufgabe besitzt kein template.

**2. Kernaufgabe: Global Positioning System**

Modellierung

Global Positioning System ist ein globales Navigationssatellitensystem zur Positionsbestimmung und Zeitmessung. GPS basiert auf Satelliten, die mit codierten Radiosignalen ständig ihre aktuelle Position und die genaue Uhrzeit ausstrahlen. Aus den Signallaufzeiten können GPS-Empfänger dann ihre eigene Position und Geschwindigkeit berechnen. Theoretisch reichen dazu die Signale von drei Satelliten aus, welche sich oberhalb ihres Abschaltwinkels befinden müssen, da daraus die genaue Position und Höhe bestimmt werden kann. In der Praxis haben aber GPS-Empfänger keine Uhr, die genau genug ist, um die Laufzeiten korrekt zu messen. Deshalb wird das Signal eines vierten Satelliten benötigt, mit dem dann auch die genaue Zeit im Empfänger bestimmt werden kann. [a]
Das Signal des Satelliten $i$ enthält seine Zeit und Position:

$$[t_{s,i},\ x_{s,i},\ y_{s,i},\ z_{s,i}]$$

Wir können $N$ Satelliten empfangen und speichern die Messwerte zeilenweise in $\mathbf{X} \in \mathbb{R}^{N\times 4}$. Die Zeit des Empfängers $t_r$ ist im Vergleich zur Signallaufzeit nur unzureichend

**Bitte wenden!**

bekannt und muss ebenso wie die Position bestimmt werden, d.h. wir suchen:

$$\underline{x} := [t_r,\ x_r,\ y_r,\ z_r].$$

Dazu minimieren wir die Residuen $r_i$, $i = 1 \dots N$ gegeben durch folgende Gleichung:

$$\boxed{r_i := -c^2(t_r - t_{s,i})^2 + (x_r - x_{s,i})^2 + (y_r - y_{s,i})^2 + (z_r - z_{s,i})^2}$$

wobei $c$ die Lichtgeschwindigkeit im Vakuum darstellt.

[a] Aus Wikipedia: `http://de.wikipedia.org/wiki/GPS`

Aufgabenstellung

**a)** Implementieren Sie den Residuenvektor $F(\underline{x}, \mathbf{X}) : \mathbb{R}^4 \times \mathbb{R}^{N\times 4} \to \mathbb{R}^N$.

**b)** Implementieren Sie die Jacobi-Matrix $J(\underline{x}, \mathbf{X}) : \mathbb{R}^4 \times \mathbb{R}^{N\times 4} \to \mathbb{R}^{N\times 4}$.

**c)** Lösen Sie das nichtlineare Ausgleichsproblem mit dem Gauss-Newton Verfahren.

**d)** Die Genauigkeit der Positionsbestimmung hängt vom Messfehler in der Signallaufzeit, der Anzahl sichtbarer Satelliten und ihrer Verteilung ab. Die Funktion `random_sampling` im Code Template simuliert den Ausfall von Satelliten, was auftritt wenn Satelliten in einer gewissen Himmelsrichtung von einem Hindernis verdeckt werden.

Kommentieren Sie die Beobachtungen für die mit `random_sampling` erstellten Plots mit verschiedenen Toleranzen in der Zeitmessung $\varepsilon_t = 10^{-10}, 10^{-8}, 10^{-7}, 10^{-6}$.

*Hinweis:* Verwenden Sie das template `gps.py`

**3.** *Lineare Ausgleichsrechnung mit linearen Nebenbedingung*

Seien die Matrizen $A$, $C$ und die Vektoren $b$, $d$:

$$A = \begin{bmatrix} 5 & -1 & -1 & 6 & 4 & 0 \\ -3 & 1 & 4 & -7 & -2 & -3 \\ 1 & 3 & -4 & 5 & 4 & 7 \\ 0 & 4 & -1 & 1 & 4 & 5 \\ 4 & 2 & 3 & 1 & 6 & -1 \\ 3 & -3 & -5 & 8 & 0 & 2 \\ 0 & -1 & -4 & 4 & -1 & 3 \\ -5 & 4 & -3 & -2 & -1 & 7 \\ 3 & 4 & -3 & 6 & 7 & 7 \end{bmatrix}, \quad b = \begin{bmatrix} -4 \\ 1 \\ -2 \\ 3 \\ 3 \\ 0 \\ -1 \\ 3 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 3 & -2 & 3 & 8 & 0 \\ -3 & 0 & 0 & 1 & 9 & 4 \\ -2 & 3 & -2 & 4 & 17 & 4 \end{bmatrix}, \quad d = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}$$

Berechnen Sie die Lösung $x_0$ des lineares Ausgleichsproblems $Cx = d$. Schlagen Sie eine andere Lösung $x_g$ von $Cx = d$ vor. Berechnen Sie die Lösung $x_a$ des lineares Ausgleichsproblems $Ax = b$.

Berechnen Sie die Lösung $x_m$ des lineares Ausgleichsproblems $Ax = b$ mit der linearen Nebenbedingung $Cx = d$.

**Siehe nächstes Blatt!**

Geben Sie für $x$ in $\{x_m, x_0, x_g, x_a\}$ aus:

- $x$
- die Residuen $r_A(x) = |b - Ax|_2$ und $r_C(x) = |d - Cx|_2$
- die Normen $|x|_2$

Welche Eigenschaften besitzen $x_m$, $x_0$, $x_g$ und $x_a$ jeweils.

*Hinweis:* Verwenden Sie das template `lstsqLinConstr.py`.

**4.** *Potenzmethoden*

Im Template `power_method.py` ist eine Matix $\mathbf{A}$ definiert. Verwenden Sie die Potenzmethoden der Vorlesung um den grössten und kleinsten Eigenwert, sowie den Eigenwert welcher am nächsten an 40 liegt zu berechnen.