Computerphysik WS 2015/2016

Prof. Dr. Roland Netz, FU Berlin

Übungsblatt 6: Interpolation und lineare Ausgleichsrechnung

Alexander Schlaich, Johann Hansing
23. November 2015

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Sonntag, 29.11., 24:00 Uhr.**

Aufgabe 6: Streuquerschnitt aus Neutronenstreuung (20 Punkte)

Wir betrachten in dieser Aufgabe Messdaten aus einem Streuexperiment, in dem Neutronen an einem Atomkern gestreut werden (Tabelle 1). Auftretende Messfehler wurden dabei durch mehrere sukzessive Messungen abgeschätzt. Ihre Aufgabe wird im folgenden darin bestehen, Werte des Streuquerschnittes zwischen den experimentellen Messpunkten zu bestimmen und daraus die Position des Resonanzpeaks und dessen Halbwertsbreite zu bestimmen.

Die Halbwertsbreite einer Funktion mit einem Maximum ist die Differenz zwischen den beiden Argumentwerten, für die die Funktionswerte auf die Hälfte des Maximums abgesunken sind, anschaulich also die „Breite bei halber Höhe“.

Tabelle 1: Experimenteller Streuquerschnitt aus Neutronenstreuung. Messfehler wurden aus mehreren sukzessiven Messungen abgeschätzt.

E (MeV)	0	25	50	75	100	125	150	175	200
σ (mb)	10.6	16.0	45.0	83.5	52.8	19.9	10.85	8.25	4.7
$\Delta\sigma$ (mb)	1.386	2.09	3.85	2.2	1.43	1.76	0.055	2.156	0.671

Bitte achten Sie bei allen Plots auf Achsenbeschriftungen und Legende!

Aufgabe 6.1: Polynom-Interpolation (10 Punkte)

Eine mathematisch einfache Interpolationsmethode ist die Polynom-Interpolation. Bei einem Polynomgrad n ist

$$P_n(x) = \sum_{i=0}^{n} a_i x^i.$$

Das Interpolationspolynom ist durch eine Punktmenge $\{(x_i,f_i), i \in \{0,1,\ldots,n\}, x_i,f_i \in \mathbb{R}\}$ und die Bedingungen $P_n(x_i) = f_i,\ \ 0 \leq i \leq n$ eindeutig bestimmt. Zur Bestimmung des eindeutigen Polynoms implementieren Sie in dieser Aufgabe zweierlei Methoden, die sich in ihrer Effizienz unterscheiden.

Naive Polynom-Interpolation

Für einen gegebenen Datensatz (x_i, f_i) kann folgende Matrix M konstruiert werden:

$$\begin{bmatrix} {x_0}^0 & {x_0}^1 & ... & {x_0}^n \\ {x_1}^0 & .. & .. & {x_1}^n \\ . & . & . & . \\ . & & & \\ {x_n}^0 & .. & .. & {x_n}^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ .. \\ .. \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ .. \\ .. \\ f_n \end{bmatrix} \tag{1}$$

Aus der Lösung des Gleichungssystems $\mathbf{M}\cdot\mathbf{a} = \mathbf{f}$ ergeben sich die Koeffizienten a_i, die zur Auswertung von $P_n(x)$ benötigt werden.

- 6.1.1 (2 Punkte): Implementieren Sie die naive Polynom-Interpolation durch das Lösen des linearen Gleichungssystems. Verwenden Sie hierzu `np.linalg.solve()`.
- 6.1.2 (1 Punkt): Erstellen Sie einen Plot, in welchem Sie das Interpolationspolynom (als Linie) zusammen mit den Messdaten (Punktue mit Fehlerbalken) darstellen. Datenpunkte mit Fehlerbalken können sie mit `matplotlib.pyplot.errorbar` erzeugen.

  Verwenden Sie für alle Plots einen Wertebereich für die x-Achse mit $x \in \{0,1,2,3,\ldots,200\}$.
- 6.1.3 (1 Punkt): Eine häufig verwendete Methode um Überschwinger bei hohen Ordnungen des Interpolationspolynoms zu vermeiden, ist beispielsweise nur jeden 2. oder 3. Datenpunkt zu interpolieren. Fügen Sie Ihrem Plot aus 6.1.2 das Interpolationspolynom hinzu, bei dem nur jeder 2. Datenpunkt zur Bestimmung des Polynoms verwendet wird. (Jedes n-te Element eines `np.array a` bekommen Sie mit der Notation `a[::n]`).

  Welches neue Problem taucht nun auf?
- 6.1.4 (3 Punkte): Programmieren sie eine Funktion `hwb(x,y)` zur Bestimmung der Halbwertsbreite eines Resonanzpeaks. Die Funktion sollte als Argumente eine Liste mit x-Werten und eine Liste mit y-Werten nehmen. In der Praxis übergeben Sie `hwb(x,y)` die x und y Werte ihres Plots aus Aufgabe 6.1.2. Dokumentieren Sie im Detail die Funktionsweise ihres Algorithmus. *Hinweis:* Implementieren Sie den folgenden Pseudocode

```
hwb(x,y):
ymaxWert = max(y)
ymaxIndex = where( y == ymaxWert )
ikleiner = where( y < ymaxWert / 2 )
liIndex = max({ i in ikleiner | i < ymaxIndex })
reIndex = min({ i in ikleiner | i > ymaxIndex })
return x[ymaxIndex], (x[reIndex] - x[liIndex])
```

  Verwenden Sie `numpy.max()` und `numpy.where()`.

Lagrange-Verfahren

Ein Polynom vom Grad n, welches durch $n+1$ Datenpunkte verläuft, lässt sich alternativ mit der Lagrange-Formel konstruieren:

$$P_n(x) = \sum_{i=0}^{n} f_i l_i(x)$$

mit

$$l_i(x) = \frac{x-x_0}{x_i-x_0} \cdot \frac{x-x_1}{x_i-x_1} \cdot \frac{x-x_2}{x_i-x_2} \cdot \dots \cdot \frac{x-x_{i-1}}{x_i-x_{i-1}} \cdot \frac{x-x_{i+1}}{x_i-x_{i+1}} \cdot \dots \cdot \frac{x-x_n}{x_i-x_n} = \prod_{\substack{j=0,\\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}$$

- 6.1.5 (3 Punkte): Implementieren Sie die Lagrange-Formel zum Interpolieren der Daten und plotten Sie das Interpolationspolynom zusammen mit dem Interpolationspolynom aus 6.1.2, sowie mit den Messdaten um sich zu vergewissern, dass beide Polynome identisch sind.

  Beschreiben Sie kurz die Vorteile der Lagrange-Methode gegenüber der naiven Interpolation.

6.2 Interpolation mit kubischen Splines (3 Punkte)

Oftmals ist es hilfreicher, anstatt eines Interpolationspolynoms hoher Ordnung stückweise Interpolationsfunktionen zu definieren. Fordert man dann die Stetigkeit der zweiten Ableitungen, so erhält man für ein Polynom dritten Grades kubische Splines. In Python können Splines einfach berechnet werden mittels `scipy.interpolate.UnivariateSpline`.

- 6.2.1 (2 Punkte): Machen Sie sich anhand der Dokumentation mit `UnivariateSpline` vertraut und erstellen Sie eine Interpolation der Messpunkte ohne Berücksichtigung der Messfehler. Um einen kubischen Spline ohne *smoothing* und *weighting* zu erhalten sollten ihre Parameter `k = 3, w = None` und `s = 0` sein.

  Erstellen Sie einen Plot wie in 6.1.2 mit den Messdaten und dem Interpolationsspline.

- 6.2.2 (1 Punkt): Ermitteln Sie mit Ihrer Funktion aus 6.1.4 die Halbwertsbreite und Resonanzenergie und vergleichen Sie Ihre Werte.

6.3 Lineare Ausgleichsrechnung (7 Punkte)

Der Streuquerschnitt folgt aus quantenmechanischer Rechnung (Fermis Goldene Regel) als Breit-Wigner Funktion,

$$\sigma = \frac{\sigma_0}{(E-E_r)^2 + \gamma^2/4}, \tag{2}$$

wobei σ_0 der maximale Streuquerschnitt, E_r die Resonanzenergie und γ die Halbwertsbreite sind.

- 6.3.1 (2 Punkte): Zeigen Sie, dass für den Kehrwert des Streuquerschnitts $1/\sigma$ die Funktion (2) auf ein lineares Ausgleichsproblem der Form

$$f(x) = a_1 x^2 + a_2 x + a_3 \tag{3}$$

  führt. Wie verhalten sich a_1,a_2,a_3 zu σ_0, E_r, γ?

- 6.3.2 (2 Punkte): Lösen Sie das lineare Ausgleichsproblem (3) mit dem Datensatz $\{E_i, \sigma_i^{-1}\}$, indem Sie eine Funktion schreiben, welche das Normalengleichungssystem

$$\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y} \tag{4}$$

  löst.

- 6.3.3 (1 Punkt): Stellen Sie in einem Plot ihr Resultat für $f(x)$ und σ_i^{-1} dar sowie in einem weiteren Plot den Kehrwert, also $f^{-1}(x)$ und die Werte σ_i. Was beobachten Sie?

Im Folgenden sollen nun die unterschiedlichen Standardabweichungen $\Delta\sigma_i$ der Werte σ_i berücksichtigt werden. Dazu werden die Werte A_{ij} und y_i entpsrechend gewichtet,

$$A_{ij} = f_j(x_i) \cdot w_i \qquad \text{und} \qquad y_i = {}^{1}/_{\sigma_i} \cdot w_i, \tag{5}$$

wobei in unserem Fall $w_i = {\sigma_i}^2/_{\Delta\sigma_i}$ die Gewichte sind.

- 6.3.4 (2 Punkte): Wiederholen Sie den Fit aus 6.3.2, wobei Sie zusätzlich zu den Messwerten noch deren Standardabweichungen $\Delta\sigma_i$ verwenden.

  Hinweis: In Gleichung (5) haben wir bereits die korrekte Fehlerfortpflanzung berücksichtigt.

  Stellen Sie Ihr Resultat für $f^{-1}(x)$ und die Messwerte σ_i in einem Plot dar und beschreiben Sie Ihre Beobachtung.

  Welche Werte erhalten Sie für E_r und γ?