

Übungsblatt 5: Mehrdimensionale nichtlineare Nullstellensuche

Markus Mietтинен, Julian Kappler
13. November 2015

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Sonntag, 22.11., 24:00 Uhr.**

Die Lösungen sollten in Form eines IPython Notebooks (*.ipynb) abgegeben werden. Bitte achten Sie dabei auf eine sinnvolle Benennung aus der Ihr Name hervorgeht. Zur Abgabe schicken Sie die Lösungsdatei im Anhang einer Email an Ihren Tutor.

Aufgabe 5: Iterative Lösung von nichtlinearen Gleichungssystemen (20 Punkte)

Im Folgenden soll das mehrdimensionale Newton-Verfahren auf ein “physikalisches” Problem angewendet werden: N repulsive, d.h. sich gegenseitig abstoßende Teilchen, in einem 1-dimensionalen Potential. Wir suchen nach statischen, also kräftefreien, Lösungen des Problems. Solche Lösungen entsprechen den Konfigurationen, gegen die ein solches System typischerweise strebt.

Das Wechselwirkungspotential zwischen zwei Teilchen $i \neq j$ an Positionen x_i, x_j sei

$$V_r(x_i, x_j) = \frac{k_r}{(x_i - x_j)^2}, \quad (1)$$

wobei $k_r = 1.0$ die Stärke der Wechselwirkung festlegt.

Ein Teilchen an der Stelle x spürt zusätzlich das externe Potential

$$V_w(x) = k_w x^2, \quad (2)$$

wobei $k_w = 2.1$ die Stärke dieses harmonischen Potentials bestimmt.

- 5.1 (1 Punkt): Stellen Sie das Potential V_w für $x \in (-5, 5)$ und das Potential $V_r(x_i, x_j)$ für $0.1 < |x_i - x_j| < 3$ in jeweils einem Plot dar.

Legen Sie ein `numpy.array` an, das die Koordinaten von vier Teilchen

$$\vec{x} = (-2., -1., 0., 0.5) \quad (3)$$

speichert. Visualisieren Sie diese Teilchen als Punkte $(x_i, V_w(x_i))$ in Ihrem Plot von V_w .

Hinweis: Im Folgenden sollten Teilchenpositionen immer auf diese Weise dargestellt werden.

- 5.2 (2 Punkte): Angenommen, das System besteht aus N Teilchen, welche sich an Positionen $\vec{x} \in \mathbb{R}^N$ befinden. Berechnen Sie analytisch, welche Kraft $F_i(\vec{x})$ auf das i -te Teilchen wirkt.

Hinweis:

$$F_i(\vec{x}) = -\frac{\partial U(\vec{x})}{\partial x_i} \quad (4)$$

wobei

$$U(\vec{x}) = \sum_i V_w(x_i) + \frac{1}{2} \sum_{i \neq j} V_r(x_i, x_j) \quad (5)$$

das Gesamtpotential des Systems ist.

- 5.3 (2 Punkte): Schreiben Sie nun eine Funktion, welche die Kräfte $\vec{F}(\vec{x})$, die auf N Teilchen an Positionen \vec{x} wirken, berechnet.

Ihre Funktion sollte die Teilchenpositionen (ein `numpy.array`) als Argument nehmen, und ein `numpy.array` mit den dazugehörigen Kräften zurückgeben.

Welche Kräfte wirken auf die vier Teilchen aus Teilaufgabe 5.1?

- 5.4 (2 Punkte): Für das Newton-Verfahren benötigen wir die Jacobi-Matrix

$$J_{ij} = \frac{\partial F_i}{\partial x_j}, \quad i, j \in \{1, \dots, N\}, \quad (6)$$

der Funktion $\vec{F}(\vec{x})$. Berechnen Sie die Komponenten dieser Matrix analytisch.

- 5.5 (2 Punkte): Implementieren Sie nun eine Funktion, welche die Jacobi-Matrix J berechnet.

Eingabeparameter sollten die Teilchenpositionen sein, Rückgabewert die Jacobi-Matrix als `numpy.array`.

Welche Jacobimatrix ergibt sich für die Teilchen aus Teilaufgabe 5.1?

- 5.6 (2 Punkte): Berechnen Sie die ersten 2 Iterationsschritte des mehrdimensionalen Newtonverfahrens für vier Teilchen.

Verwenden Sie als Startvektor $\vec{x}^{(0)}$ die Positionen aus Teilaufgabe 5.1 und visualisieren Sie die Positionen $\vec{x}^{(0)}$, $\vec{x}^{(1)}$, $\vec{x}^{(2)}$ in je einem separaten Plot zusammen mit dem Potential V_w .

- 5.7 (2 Punkte): Implementieren Sie nun das mehrdimensionale Newton-Verfahren, um die kräftefreien Konfigurationen von N Teilchen zu bestimmen.

Ihr Programm sollte als Argumente die Anfangsteilchenpositionen $\vec{x}^{(0)}$, sowie optional die gewünschte Genauigkeit ϵ (Standardwert 10^{-3}) und eine maximale Anzahl an Iterationsschritten $N_{iter,max}$ (Standardwert 1000) nehmen. Die resultierende Konfiguration sowie die Anzahl an benötigten Iterationsschritten sollte zurückgegeben werden.

Hinweis: Zur Lösung des entsprechenden linearen Gleichungssystems können sie `numpy.linalg.solve` verwenden. Als Konvergenzkriterium sollte $(\vec{x}^{(k+1)} - \vec{x}^{(k)})^2 < \epsilon^2$ gewählt werden.

- 5.8 (2 Punkte): Berechnen Sie mithilfe Ihrer Funktion aus Teilaufgabe 5.7 die kräftefreie Konfiguration der vier Teilchen aus Teilaufgabe 5.1.

Visualisieren Sie ihr Resultat wie in Teilaufgabe 5.1.

Berechnen Sie auch mithilfe Ihrer Funktion aus Teilaufgabe 5.3 die auf die Teilchen wirkenden Kräfte.

- 5.9 (3 Punkte): Implementieren Sie eine Funktion, die das Gesamtpotential des Systems als Funktion der Teilchenpositionen berechnet.

Implementieren Sie nun eine Variation der Funktion aus Teilaufgabe 5.7, die in jedem Iterationsschritt das Gesamtpotential berechnet und am Ende eine Liste dieser Potentiale zurückgibt.

Stellen Sie das Potential als Funktion des Iterationsschrittes für die vier Teilchen aus Teilaufgabe 5.1 dar.

- 5.10 (2 Punkte): Berechnen Sie nun mithilfe Ihrer Funktion aus Teilaufgabe 5.9 das Gesamtpotential der kräftefreien Konfiguration für jeweils

$$N \in \{2, 4, 8, 16, 32, 64, 128\} \quad (7)$$

Teilchen.

Stellen Sie dieses Potential als Funktion der Teilchenanzahl in einem doppellogarithmischen Plot dar.

Ist das Potential für große N proportional zu N^α ? Wenn ja, für welches α ?

Hinweis: Die Anfangspositionen der Teilchen sollten zwischen -10 und 10 gewählt werden, wobei keine zwei Teilchen an derselben Position starten sollten. Zum Erstellen von zufälligen Startpositionen können Sie `numpy.random.random` verwenden.