

III. 10 Optimierungsprobleme

Im Zusammenhang mit Problemen in der Quantenmechanik stellen sich oft Optimierungsprobleme z. B.:

(i) Strukturoptimierung (Born-Oppenheimer-Näherung)

Für den Hamiltonoperator der Moleküle!

$$H_e = \sum_i \frac{p_i^2}{2m_e} - \sum_a \sum_i \frac{1}{4\pi\epsilon_0} \frac{Z_a}{|r_i - R_a|} + \sum_{i,j} \frac{1}{8\pi\epsilon_0} \frac{1}{|r_i - r_j|} + \sum_{a,b} \frac{1}{8\pi\epsilon_0} \frac{Z_a Z_b}{|R_a - R_b|}$$

bisher fixiert.

Wurden die Kernpositionen angenommen!

Begründung: Born-Oppenheimer-Näherung:

Die Masse der Kerne ist sehr viel größer, als die der Elektronen \Rightarrow langsame Bewegung.

• Daher wird die Kernposition R_a fixiert, an den Punkt fixiert, an dem es die minimale Gesamtenergie gibt! (Adiabatische Behandlung Elektronen folgen den Kernen schnell)

• Bestimmung der Kernposition \Rightarrow Optimierungsproblem bzgl. Kernposition \Rightarrow Beantwortet man

$E(R_e)$ und Algorithmus.

• Taylorentwicklung um Gleichgewichtsposition R_a^0

$$E(R_a^0 + \Delta R_a^0) = \sum_{i,j} M_{ij} (\Delta R_a^0)_i (\Delta R_a^0)_j$$

ergibt ein quadratisches Potential \Rightarrow harmonische Oszillatoren

\Rightarrow Phonon. μ wird über die Hessesche Matrix bestimmt

(kann durch finite Differenzen $\underline{F}(-)$ berechnet werden)

(ii) Basis satz optimierung

Zur Erinnerung, STO oder STO enthalten Parameter

STO: $\chi_{s,m,l,m}(r, \theta, \varphi) = N Y_{l,m}(\theta, \varphi) r^{n-1} e^{i\varphi l}$

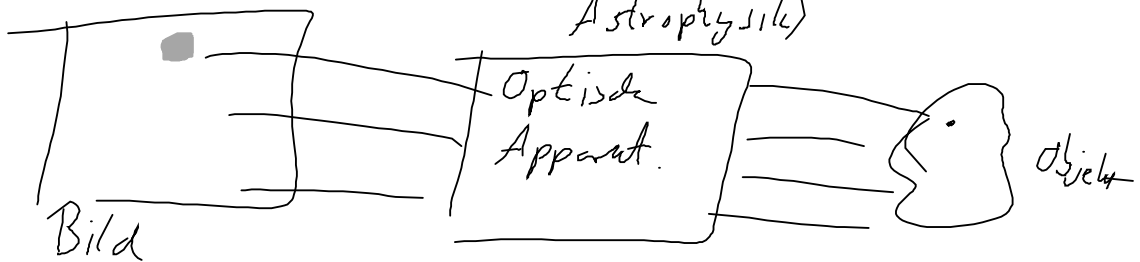
\uparrow
Zeta

Dieser Zeta ξ wurde für den Ansatz der Wellenfunktion optimiert und zur

$$\langle E(\xi_1, \dots, \xi_n) \rangle$$

Diese Funktion wurde numerisch berechnet und dann optimiert.

(iii) Beispiel aus der normalen Physik
(Exp. Physik, Imaging; Astrophysik)



$$B(x, y) = \int dx' \int dy' \underbrace{PSF(x-x', y-y')}_{\text{Point Spread Funktion}} O(x', y')$$

beschreibt wie ein Punkt des Objekts im Bild abgebildet wird.
PSF macht Bild unscharf.

Fassen wir $O(x', y')$ als Vektor auf, so kann es

$$E(0) = \left\| \int dx' \int dy' \text{PSF}(x-x', y-y') O(x', y') - \text{Bexp}(x, y) \right\|_2$$

\Rightarrow wird $O(x, y)$ variert und dabei $E(0)$ minimiert

Kann man in gewissen Grenzen Unschärfe reduzieren.

(Eine von vielen Möglichkeiten der deconvoluten Algorithmus)

\Rightarrow Eine über sieht über Optimierungsalgorithmen wie oben!

Algorithmen für Optimierungsproblem

Beweis es lösbar ist, eine Beweismenge

Quadratische Optimierungsproblem

z.B.
$$F_{\text{err}}(\underline{a}) = \sum_j \left(y_j - \sum_i a_i x_{ij} \right)^2$$

↳ $\sum_i a_i x_{ij}$ wird gesucht.
kann über ein lineares Gleichungssystem gelöst werden.

Dann

$$\frac{\partial F_{\text{err}}(\underline{a})}{\partial a_i} = -2 \sum_j x_{ij} (y_j - \sum_{i'} a_{i'} x_{i'j}) \stackrel{!}{=} 0$$

$$\Rightarrow \sum_{i', j} x_{ij} x_{i'j} a_{i'} = \sum_j x_{ij} y_j$$

$$\sum_j x_{ij} \sum_{i'} x_{i'j} a_{i'} = \sum_j x_{ij} y_j$$

$$\Rightarrow \sum_{i'} x_{i'j} a_{i'} = y_j$$

$$\Rightarrow \underline{X} \cdot \underline{a} = \underline{y}$$

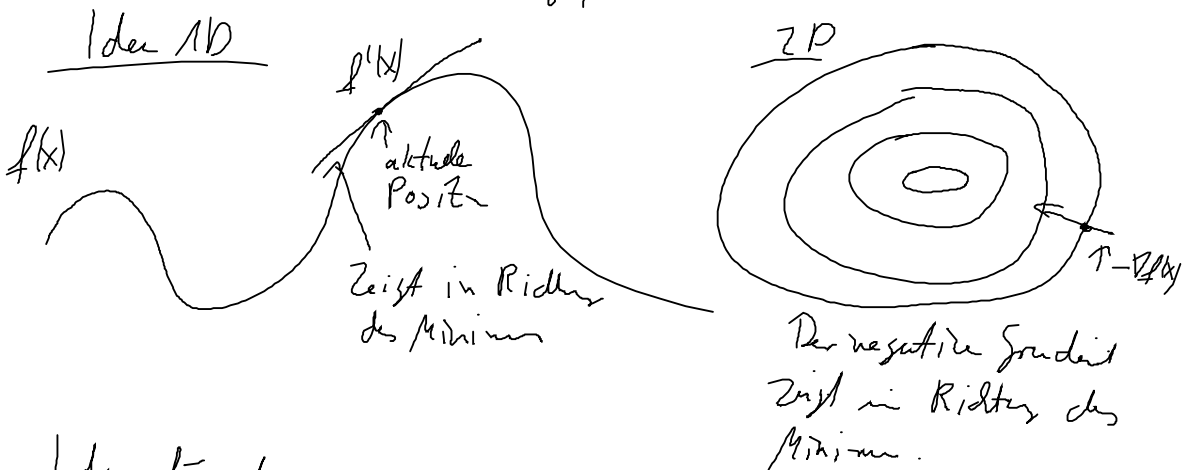
Kann über $\underline{a} = \underline{X}^{-1} \cdot \underline{y}$ gelöst werden.
falls \underline{X} invertierbar ist

Kann man auch lösen ohne Invertierung, wichtig bei großen Matrizen.

Methoden mit Gradienten:

Steepest descent (einfachste Method -)

Bemerkung: Wir suchen in folgenden Minimum ein
Funktion $f(x)$, die folgende Methode finden in der
Regel nur lokale Minima!
(Zusätzliche Strategien für globale Minima
notwendig)

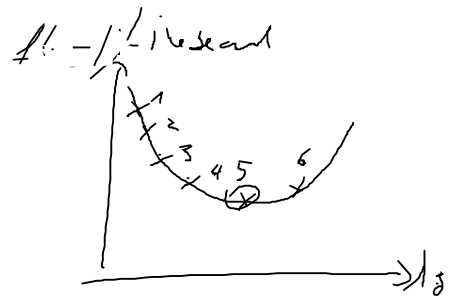
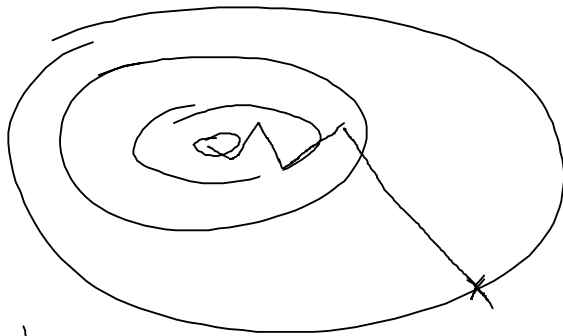


Idee für das
Vorgehen:

- 1) Starte am Punkt x_0
- 2) Berechne $\underline{g} = -\nabla f(x_0)$, das wird unsere Suchrichtung
- 3) Lineare such. berechne $f(x_0 + \lambda \underline{g})$
für verschiedene λ , so lange f abnimmt
erhöhe λ weiter, sobald f steigt linear auf.

Illustration

$$\text{Setze } \underline{x}^{(n)} = \underline{x}^{(n-1)} + \lambda^{(n)} \underline{g}^{(n)}$$



Vorteile

- Bei guter Linesearch wird steapest descent immer der Funktionswert reduziert.
(Garantie sich dem Minimum zu nähern)
- sehr einfach
- Nur ein Gradient muss bestimmt werden.

Nachteile

- Zwei aufeinander folgende Linesearchs stehen senkrecht aufeinander
 \Rightarrow teilweise wird die erreichte Optimierung in einer Dimension wieder zurück gemacht.
- Der Algorithmus oszilliert entlang des optimalen Weges.
- Nahe des Minimums nimmt die Konvergenzrate ab (flacher)

Hauptanwendung:

Schnell nahe an ein Minimum zu kommen, dann bessere Methode.

Conjugate gradient Methode

Hauptproblem von Steepest Descent,
wir unsere Optimierung im nächsten Schritt teilweise
rückgängig gemacht haben:

Hier wird versucht, dass die Richtungen sich nicht
zu abrupt ändern:

$$\underbrace{d_i}_{\text{Richtung}} = \underbrace{g_i}_{\text{Aktuelle negative Gradient}} + \beta_i \underbrace{d_{i-1}}_{\text{letzte Richtung}}$$

Das ist so konstruiert, dass für quadratische
Oberflächen, keine Gradientenrichtungen entlang
vorheriger Richtungen entstehen.

Verschiedene Möglichkeiten für β_i :

Fletcher-Ravens (FR) $\beta_i^{FR} = \frac{g_i \cdot g_i}{g_{i-1} \cdot g_{i-1}}$

Polak-Ribiere (PR) $\beta_i^{PR} = \frac{g_i \cdot (g_i - g_{i-1})}{g_{i-1} \cdot g_{i-1}}$

Hestenes-Stiefel $\beta_i^{HS} = \frac{g_i \cdot (g_i - g_{i-1})}{d_{i-1} \cdot (g_i - g_{i-1})}$

Wurde anhand einer quadratischen Fkt. entwickelt, so
dass $d_i^T \cdot H \cdot g_i = 0$, also d_i zu g_i konjugiert.

Dort sind alle Verfahren äquivalent, im allgemeinen unterschiedlich
Vektoren.

Komplett super funktioniert CG bei nicht-quadratischen
Oberflächen nicht, daher ist ein Neustart notwendig.

(z.B. misst man in wie weit die aufeinanderfolgenden Gradienten
nicht orthogonal sind (PR funktioniert besser nach Restart))

- Bemerkung :
- Sehr viel bessere Konvergenz als steepest descent.
 - Preconditioning auf die Variablen (Koordinaten-
transformiert) kann helfen.
 - Nur zwei Vektoren müssen gespeichert
werden (ein mehr als Steepest Descent)
 - Der Gradient muss bei beiden Methoden
berechnet werden, nicht immer ist das
analytisch oder in schneller Zeit möglich.