

III. 10 Optimierungsprobleme

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

(i) Strukturoptimierung (Born-Oppenheimer Näherung)

Für den Hamiltonoperator des Moleküls

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

Wird die Kernposition fixiert

R_a ! Born-Oppenheimer: Die Masse der Kerne ist sehr viel größer als die der Elektronen \Rightarrow langsame Bewegung

- * Daher wird die Kernposition R_a fixiert, an dem Punkt in dem die minimale Gesamtenergie verliert.
(Adiabatische Behandlung, Elektron folgen den Kernen schnell)
- * Bestimmung der Kernposition \Rightarrow Optimierungsproblem bzgl. der Kernposition \Rightarrow Benötigt wird $E(R_a)$ und Algorithmen
- * Taylorentwicklung um Gleichgewichtsposition R_a^0

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

existiert in quadratisches Potential \Rightarrow harmonische Oszillatoren
 \Rightarrow Phononen. M wird über die Hessesche Matrix beschrieben

(kann durch finite Differenz an E[...] brechen werden)

(ii) Basisatz optimierung

Zur Ermittlung STO oder STO enthaltene Parameter:

STO:

$$\chi_{r, h_1, l, m}(r, \theta, \varphi) = N Y_{lm}(\theta, \varphi) r^{n-1} e^{-\frac{r}{\lambda}}$$

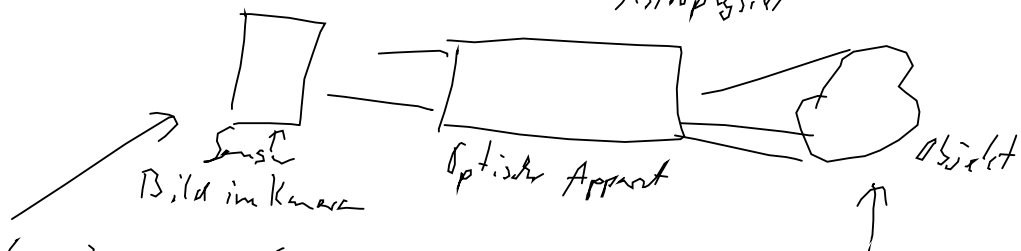
Dieses Zeta χ wurde für den Ansatz der Wellenfunktion optimiert und zwar

$$\langle E(r_1, \dots, r_n) \rangle$$

Funktion der entsprechenden Basisatzparameter!

Diese Funktion wurde numerisch breitet und dann optimiert.

(iii) Beispiel aus der modernen Physik (Exp. Physik, Laserphysik, Astrophysik)



$$B(x, y) = \int dx' \int dy' \text{PSF}(x-x', y-y') O(x', y')$$

Point Spread Funktion
beschreibt wie ein Punkt des Objekts
in Bild abgebildet wird.

← Punkt wird verschmiert! PSF macht Bildverschmierung
Fassen wir $O(x', y')$ als Verteilung auf, so kann z.B.

$$E(\theta) = \left\| \int dx' \int ds' \text{PSF}(x-x', s-s') O(x', s) - B_{\text{exp}}(x, s) \right\|_2$$

\Rightarrow wird $O(x, s)$ variiert und dabei $E(\theta)$ minimiert.

Kann man in gewissern Sinne Unsicherheit reduzieren
(Eine von vielen Möglichkeiten der Deconvolutionalgorithmen)

\Rightarrow Eine Übersicht über Optimierungsalgorithmen wäre schön!

Algorithmen für Optimierungsprobleme

Bevor es los geht, eine Banche:

Quadratische Optimierungsprobleme

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

\uparrow wird gesucht

keinen über lineares Gleichungssystem gelöst werden:

Dann

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

Gradient
muss verschwinden

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

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

(Adjektiv)
 x_{ij}^{-1}

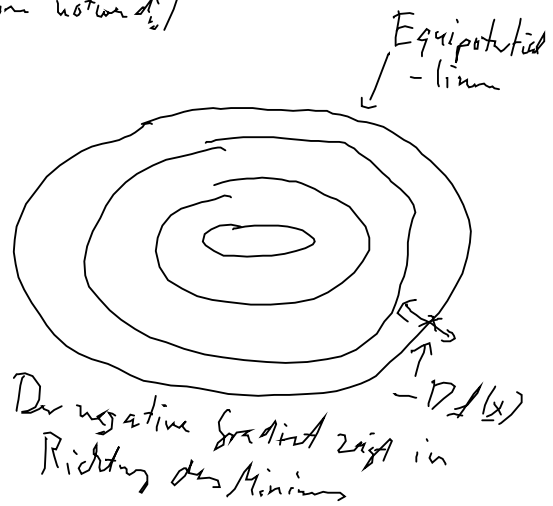
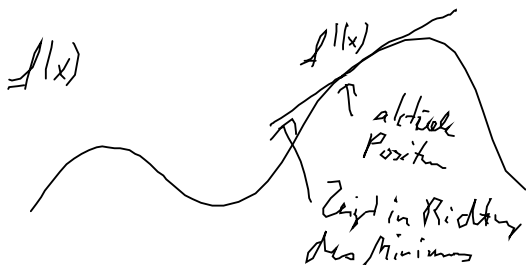
$$\Rightarrow \sum_{i'} x_{i'j} a_{i'} = y_j$$

$\Rightarrow \underline{x} - a = \underline{y}$
 kann über $\underline{a} = \underline{x}^{-1} \cdot \underline{y}$ gelöst werden, falls \underline{x}
 invertierbar ist. Aber das muss nicht sein,
 das wird es schwierig. Auch schwierig bei großen Matrizen.
Methoden mit Gradienten:

Steepest descent (einfachste Methode)

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

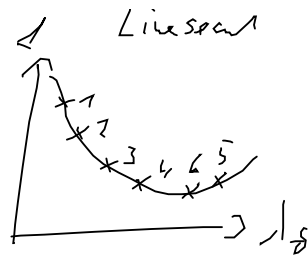
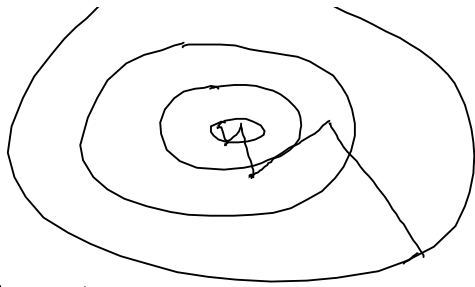
Idee 1D



Idee für das Vorgehen:

- 1) Starten an Punkt x_0
- 2) Berechnen $g = -\nabla f(x_0)$, das wird unsere Suchrichtung.
- 3) Lineare such $f(x_0 + \lambda g)$ berechnen für verschiedene λ , so klein wie f klein wird, steigt f so höher an.
 Setze $x_0^{(n)} = x_0^{(n-1)} + \lambda^{(n)} g$, gehe wieder zu 2

Illustration falls $\nabla f(x_0) \neq 0$, sonst stopps



- Vorteile:
- Bei guter Liniensort wird steepest descent immer den Funktionswert reduzieren (Gradienten sind dem Minimum zu naher)
 - Sehr einfach
 - Nur ein Gradientenmess gespeichert werden.

- Nachteile:
- Zwei aufeinander folgende Liniensorten stellen sich recht oft auf
 - => Teilweise wird die erreichte Optimierung in einer Dimension wieder zunicht gemacht.
 - Der Algorithmus oscilliert um das minimale Wert
 - Guter Liniensort teuer, oft muss sehr lange angesetzt
 - Nahe dem Minimum nimmt die Konvergenzrate ab (flack)

Hauptanwendung: Schnell nahe an das Minimum^{zu} kommen und dann übernimmt die bessere Methode.

Conjugate gradient Methode

Hauptproblem von steepest descent, was das teilweise rückgängig machen der Optimierung im nächsten Schritt.

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

$$\underset{\substack{\uparrow \\ \text{Richter}}}{d_i} = \underset{\substack{\uparrow \\ \text{Aktueller negativer} \\ \text{Gradient}}}{g_i} + \beta_i \underset{\substack{\uparrow \\ \text{letzte Richter}}}{d_{i-1}}$$

Das ist so gebaut, dass für quadratische Oberflächen kein Gradientenrichtungswechsel der vorherigen Richtung entsteht.

Verschiedene Möglichkeiten für β_i :

Fletcher-Reeves: $\beta_i^{FR} = \frac{g_i \cdot g_i}{g_{i-1} \cdot g_{i-1}}$

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

Hestenes-Hinzel:

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

Komplett super funktioniert CG bei nicht-quadratischen Oberflächen nicht, daher ist ein Neustart notwendig
 2. B misst man in wie weit die aufeinanderfolgenden Gradienten nicht orthogonal sind

Werden anhand von quadratischen Funktionen entwickelt, so dass $d_i \cdot H \cdot g_i = 0$, aber d zu g konjugiert
 Dort sind alle äquivalent, im allgemeinen unterschiedliche Verfahren

(PR funktioniert besser PostA)

Bemerkungen

- Sehr viel ~~das~~ bessere Konvergenz als steepest descent
- Precondition auf die Variablen (Koordinatentransformall kann helfen)
- Nur zwei Verfahren müssen verglichen werden (einer mehr als Steepest Descent)
- Der Gradient muss in beide Methoden berechnet werden, nicht immer ist das analytisch oder in schneller Zeit möglich.