III.10 Optimierungsprobleme

Im Zusammenhang mit den Problem in der Quantenmechanik stellen sich öfters Optimierungsprobleme z.B.

(i) Strukturoptimierung (Born-Oppenheimer Näherung)

Für den Hamiltonoperator der Moleküle

$$H_e = \sum_i^{N_e} -\frac{\hbar^2 \Delta_i}{2m_e} - \sum_a^{N_{kern}} \sum_i^{N_e} \frac{1}{4\pi\varepsilon_0}\frac{Z_a e^2}{|\underline{r}_i - \underline{R}_a|} + \sum_{i\neq j}^{N_e} \frac{1}{8\pi\varepsilon_0}\frac{1 e^2}{|\underline{r}_i-\underline{r}_j|} + \sum_{ab}^{N_{kern}} \frac{1}{8\pi\varepsilon_0}\frac{e^2 Z_a Z_b}{|\underline{R}_a-\underline{R}_b|}$$

Werden die Kernposition fixiert R_i! Born-Oppenheimer: Die Masse der Kerne ist sehr viel größer, als die der Elektronen $\Rightarrow$ langsame Bewegung

* Daher wird die Kernposition $\underline{R}_a$ fixiert, an dem Punkt in dem die minimale Gesamtenergie vorliegt.
  (Adiabatische Behandlung, Elektronen folgen den Kern schnell)
* Bestimmung der Kernposition $\Rightarrow$ Optimierungsproblem bezgl. der Kernposition $\Rightarrow$ Benötigt wird $E(R_a)$ und Algorithmus
* Taylorentwicklung um Gleichgewichtsposition $\underline{R}_a^0$

$$E(\{\underline{R}_a^0 + \underline{\Delta R}_a\}) = \sum_{ij} M_{ij} (\Delta R_a^0)_i (\Delta R_a^0)_j$$

ergibt ein Quadratisches Potential $\Rightarrow$ harmonischer Oszillator $\Rightarrow$ Phononen. M wird über die Hesse'sche Matrix bestimmt.

(kann durch finite Differenzen aus E(...) berechnet werden)

(ii) Basissatzoptimierung

Zur Erinnerung STO oder GTO enthalten Parameter:

STO:

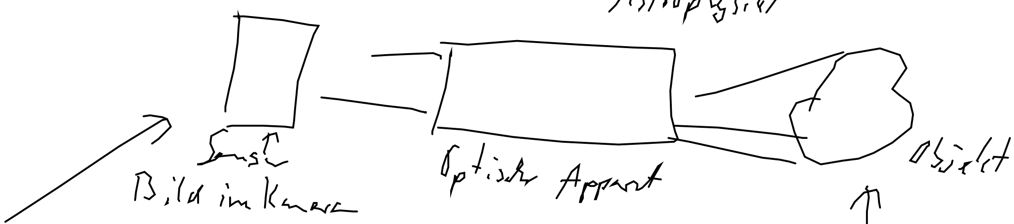
$$\chi_{\zeta,n,l,m}(r,\vartheta,\varphi) = N\, Y_{lm}(\vartheta,\varphi)\, r^{n-1}\, e^{-\underbrace{\zeta}_{\text{Zeta}} r}$$

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

$$\langle E(\underbrace{\zeta_1,\dots,\zeta_n}_{\text{Funktion der entsprechenden Basissatzparameter!}})\rangle$$

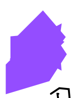
Diese Funktion wurde numerisch berechnet und dann optimiert.

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



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

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



Punkt wird verschmiert! PSF macht Bild unscharf

Fassen wir $O(x',y')$ als Vektor auf, so kann z.B.

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

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

Kann man in gewissen Grenzen Unschärfe reduzieren

(Einer von vielen Möglichkeit der Deconvolutionalgorith)

$\Rightarrow$ Eine Übersicht über Optimierungsalgorithmen wäre schön!

Algorithmen für Optimierungsprobleme

Bevor es los geht, eine Bemerkung:

Quadratische Optimierungsproblem

z.B. $Err(\underline{a}) = \sum_j \left(y_j - \sum_i^N a_i x_{ij}\right)^2$ ← wird gesucht

könnte über lineares Gleichungssystem gelöst werden:

Denn

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

Gradient muss verschwinden

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

$$\sum_j x_{ij} \sum_{i'} x_{i'j} a_{i'} = \sum_j x_{ij} y_j \qquad (\text{beliebige } x_{ij})$$

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

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

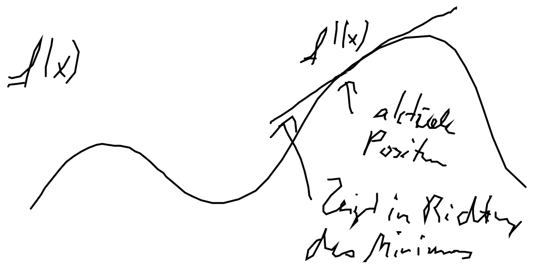
Kann über $\underline{a} = \underline{\underline{X}}^{-1} \cdot \underline{y}$ gelöst werden, falls $\underline{\underline{X}}$ invertierbar ist. Aber das muss nicht sein, dann wird es schwierig. Auch schwierig bei großen Matrizen.

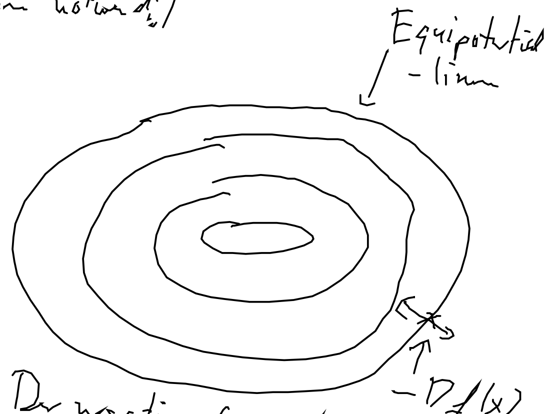
Methoden mit Gradienten:

Steepest descent (einfachste Methode)

Bemerkung: Wir suchen im folgenden ein Minimum einer Funktion $f(\underline{x})$, die folgenden Methoden finden in der Regel nur lokale Minima! (Zusätzliche Strategien sind für globales Minimum notwendig)

Idee 1D



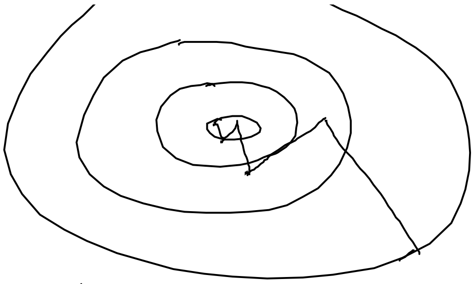


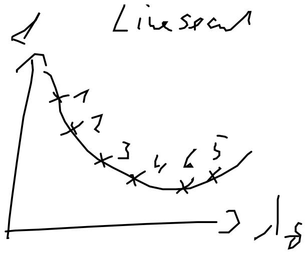
Der negative Gradient zeigt in Richtung des Minimums

Idee für das Vorgehen:

1) Starten auf Punkt $\underline{x}_0$
2) Berechne $\underline{g} = -\nabla f(\underline{x}_0)$, das wird unsere Suchrichtung.
3) Line search $f(\underline{x}_0 + \lambda \underline{g})$ berechnen für verschiedene λ, so lange wie f kleiner wird, steigt f so höre auf.
   Setze $\underline{x}_0^{(n)} = \underline{x}_0^{(n-1)} + \lambda_{(n)} \underline{g}$, gehe wieder zu 2 falls $\nabla f(\underline{x}_0) \neq 0$, sonst stopp

Illustration





Vorteile: · Bei guter Linesearch wird steepest descent immer den Funktionswert reduzieren (Garantie sich dem Minimum zu nähern)

· Sehr einfach

- Nur ein Gradient muss gespeichert werden.

Nachteile · Zwei aufeinander folgende Linesearch stehen senkrecht aufeinander

⇒ teilweise wird die erreichte Optimierung in einer Dimension wieder zunichte gemacht.

· Der Algorithmus oszilliert entlang des minimalen Weges

· guter Linesearch teuer, oft muss sehr kreativ ausgeführt.

· Nahe dem Minimum nimmt die Konvergenzrate ab (flach)

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

Conjugate gradient Methode

Hauptproblem von steepest descent, war das teilweise rückgängig machen der Optimierung im nächsten Schritt.
Hier wird jetzt versucht, dass die Richtung sich nicht abrupt ändert:

$$\underline{d}_i = \underline{g}_i + \beta_i \, \underline{d}_{i-1}$$

↑ Richtung ↑ Aktueller negativer Gradient ↑ letzte Richtung

Das ist so gebaut, dass für quadratische Oberflächen keine Gradientenrichtung entlang der vorherigen Richtung entsteht.

Verschiedene Möglichkeiten für β_i:

Fletcher-Reeves: $$\beta_i^{FR} = \frac{\underline{g}_i \cdot \underline{g}_i}{\underline{g}_{i-1} \cdot \underline{g}_{i-1}}$$

Polak-Ribière (PR): $$\beta_{PR} = \frac{\underline{g}_i \cdot (\underline{g}_i - \underline{g}_{i-1})}{\underline{g}_{i-1} \cdot \underline{g}_{i-1}}$$

Hestenes-Stiefel:

$$\beta_i^{HS} = \frac{\underline{g}_i \cdot (\underline{g}_i - \underline{g}_{i-1})}{\underline{d}_{i-1}^t \cdot (\underline{g}_i - \underline{g}_{i-1})}$$

Wurden anhand von quadratischen Funktion entwickelt, so dass $\underline{d}_i \cdot \underline{H} \cdot \underline{g}_i = 0$, also $\underline{d}$ zu g konjugiert. Dort sind alle equivalent; im allgemeinen unterschiedliche Verhalten

Komplett super funktioniert CG bei nicht-quadratischen Oberflächen nicht, daher ist ein Neustart notwendig.
z.B. misst man inwieweit die aufeinanderfolgenden Gradienten nicht orthogonal sind
(PR funktioniert ~~bes~~ und Restart)

Bemerkung

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