

2. Projekt

***Quasi-Newton-Verfahren & Gauß-Newton-Verfahren***

im Fach

Numerische Optimierung

Juni 2020

Maximilian Gaul

## **Aufgabe 1**

Siehe Programmcode in Project2.m.

## Aufgabe 2

$$\begin{aligned}
 & I - \frac{s(As)^T}{s^T As} + \frac{A^{-1}yy^T}{y^T s} + \frac{(s - A^{-1}y)s^T A + s(s - A^{-1}y)^T A}{y^T s} - \frac{(s - A^{-1}y)s^T A}{y^T ss^T As} + \frac{(s - A^{-1}y)s^T yy^T + s(s - A^{-1}y)^T yy^T}{(y^T s)^2} - \frac{(s - A^{-1}y)^T y_{ss}^T A}{(y^T s)^2 s^T As} + \frac{(s - A^{-1}y)^T y_{ss}^T A_{ss}^T A}{(y^T s)^2 s^T As} - \frac{(s - A^{-1}y)^T y_{ss}^T A_{ss}^T A}{(y^T s)^2 y^T s} \\
 & I - \frac{ss^T A}{s^4 As} + \frac{A^{-1}yy^T}{y^T s} + \frac{ss^T A - A^{-1}ys^T A + s(s^T - (A^{-1}y)^T)A}{y^T s} - \frac{ss^T A_{ss}^T A - A^{-1}ys^T A_{ss}^T A + s(s^T - (A^{-1}y)^T)A_{ss}^T A}{y^T ss^T As} + \frac{ss^T yy^T - A^{-1}ys^T yy^T + s(s^T - (A^{-1}y)^T)yy^T}{(y^T s)^2} - \frac{(s^T - (A^{-1}y)^T)y_{ss}^T A}{(y^T s)^2 s^T As} + \frac{(s^T - (A^{-1}y)^T)y_{ss}^T A_{ss}^T A}{(y^T s)^2 y^T s} - \frac{(s^T - (A^{-1}y)^T)y_{ss}^T yy^T}{(y^T s)^2 y^T s} \\
 & I - \frac{ss^T A}{s^4 As} + \frac{A^{-1}yy^T}{y^T s} + \frac{2ss^T A - A^{-1}ys^T A - sy^T}{y^T s} - \frac{2ss^T A - A^{-1}ys^T A - sy^T}{y^T ss^T As} \cdot ss^T A + \frac{2ss^T - A^{-1}ys^T - sy^T}{(y^T s)^2} \cdot ss^T A + \frac{(s^T - y^T(A^{-1})^T)y_{ss}^T A_{ss}^T A}{(y^T s)^2 s^T As} - \frac{(s^T - y^T(A^{-1})^T)y_{ss}^T yy^T}{(y^T s)^2 y^T s}
 \end{aligned}$$

### Aufgabe 3

Wenn die Suchrichtung des BFGS Verfahrens:

$$d = -B \cdot \nabla f(x)$$

keine Abstiegsrichtung ist, d.h. die Bedingung:

$$\nabla f(x)^T \cdot d < 0$$

nicht erfüllt ist, muss das Verfahren 'resettet' werden. In diesem Fall bietet es sich an, die Suchrichtung auf den negativen Gradienten zu setzen:

$$d = -\nabla f(x)$$

Da nun die Abstiegsrichtung nicht mehr zur approximierten Inversen der Hesse-Matrix  $B$  passt, muss diese ebenfalls für den nächsten Schritt neu bestimmt werden. Hierzu bieten sich verschiedene Möglichkeiten an:

- Wie beim Start des BFGS-Verfahrens  $B = I$  setzen
  - Hierbei geht jeglicher berechnete Fortschritt verloren, es handelt sich um einen recht naiven Ansatz
- Die Hesse-Matrix einmalig aus Differenzenquotienten des Gradienten bestimmen und anschließend invertieren
  - Hoher Rechenaufwand von  $\mathcal{O}(n^2)$  für die Hesse-Matrix und nochmal  $\mathcal{O}(n^3)$  für das Invertieren
  - Bisher berechneter Fortschritt geht ebenfalls verloren aber die Approximation der Hesse-Matrix ist sehr genau
- Man könnte, wie in den Vorlesungsfolien beschrieben,  $\frac{y^T s}{y^T y} \cdot I_n$  als positiv-definitive Matrix verwenden

Um herauszufinden, welche dieser Methoden am besten geeignet ist (d.h. die richtig Lösung in der kürzesten Zeit findet), wird die Laufzeit des inversen BFGS-Verfahrens bestimmt. Startwerte:

- N-dim. Rosenbrock-Funktion:  $\begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}$
- Himmelblau-Funktion:  $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$
- Bazaraa-Shetty:  $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$

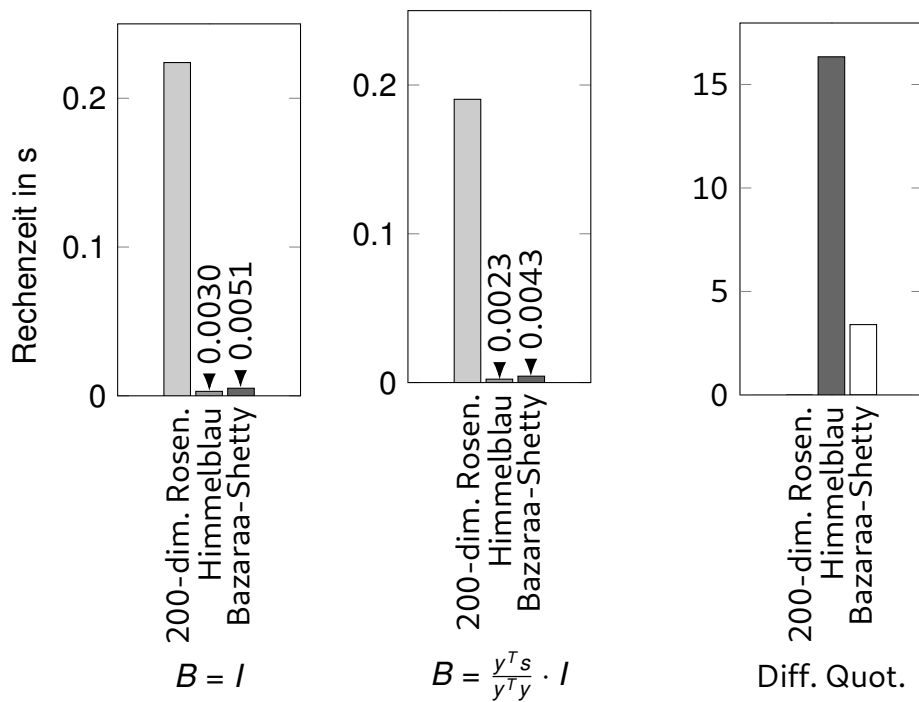


Abbildung 1: Vergleich der Rechenzeit für eine Genauigkeit von  $10^{-8}$  gemittelt über 100 Durchläufe (Intel Core i3-7100U, 8GB RAM, Windows 10 64-Bit)

Alle drei Funktionen profitieren von  $B = \frac{y^T s}{y^T y} \cdot I$ , daher habe ich diesen Weg in meiner Implementierung gewählt.

#### Aufgabe 4

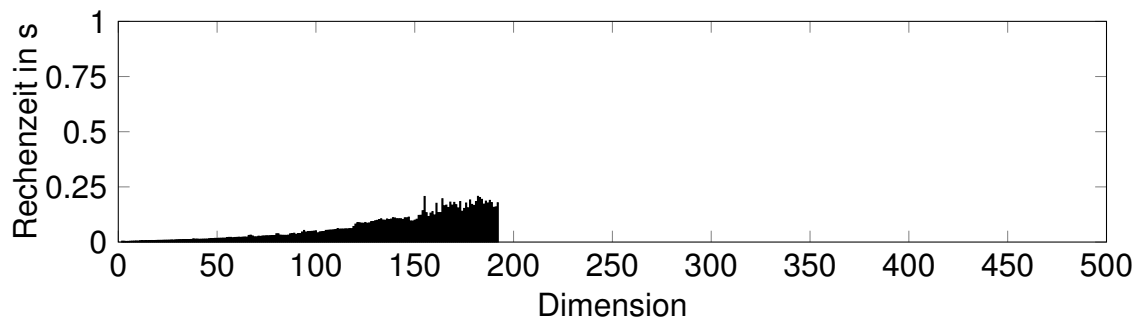


Abbildung 2: Rechenzeit der N-dimensionalen Rosenbrock-Funktion gemittelt über 100 Durchläufe