

4. Projekt

im Fach

Numerische Optimierung

Juli 2020

Maximilian Gaul

## Aufgabe 1

Siehe GlobNewton.m.

## Aufgabe 2

Siehe auch Projekt\_4.m.

Für die Himmelblau-Funktion

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

gelten folgende Ableitungen

$$\nabla f(x_1, x_2) = \begin{bmatrix} 2(x_1^2 + x_2 - 11) \cdot 2x_1 + 2(x_1 + x_2^2 - 7) \\ 2(x_1^2 + x_2 - 11) + 2(x_1 + x_2^2 - 7) \cdot 2x_2 \end{bmatrix}$$
$$H_f(x_1, x_2) = \begin{bmatrix} 4(x_1^2 + x_2 - 11) + 8x_1^2 & 4x_1 + 4x_2 \\ 4x_1 + 4x_2 & 4(x_1 + x_2^2 - 7) + 8x_2^2 \end{bmatrix}$$

Schritt	$\mathbf{x}$	$f(x)$
1	$[0.00, 0.00]^T$	170.0
2	$[1.75, 2.75]^T$	32.26
3	$[3.76, 2.22]^T$	31.69
4	$[3.19, 1.96]^T$	1.31
5	$[3.02, 1.99]^T$	0.01
$\vdots$	$\vdots$	$\vdots$
15	$[3.00, 2.00]^T$	$1.10 \cdot 10^{-26}$

Abbildung 1: Verlauf von GlobNewton für  $f$  bei einer Genauigkeit von  $10^{-12}$

Schritt	$\mathbf{x}$	$f(x)$
1	$[-1.20, 1.00]^T$	125.11
2	$[-2.87, 3.87]^T$	27.30
3	$[-2.80, 3.29]^T$	1.05
4	$[-2.80, 3.14]^T$	0.00
5	$[-2.81, 3.13]^T$	$9.83 \cdot 10^{-7}$
$\vdots$	$\vdots$	$\vdots$
12	$[-2.81, 3.13]^T$	$4.10 \cdot 10^{-29}$

Abbildung 2: Verlauf von GlobNewton für  $f$  bei einer Genauigkeit von  $10^{-12}$

Für die 2D Rosenbrock-Funktion

$$g(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

gelten die Ableitungen

$$\nabla g(x_1, x_2) = \begin{bmatrix} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\ 200(x_2 - x_1^2) \end{bmatrix}$$

$$H_g(x_1, x_2) = \begin{bmatrix} 800x_1^2 - 400(x_2 - x_1^2) + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

Schritt	$\mathbf{x}$	$g(\mathbf{x})$
1	$[0.00, 0.00]^T$	1.00
2	$[0.25, 0.00]^T$	0.95
3	$[0.31, 0.09]^T$	0.48
4	$[0.52, 0.22]^T$	0.46
5	$[0.57, 0.32]^T$	0.19
$\vdots$	$\vdots$	$\vdots$
15	$[1.00, 1.00]^T$	$8.21 \cdot 10^{-28}$

Abbildung 3: Verlauf von GlobNewton für  $g$  bei einer Genauigkeit von  $10^{-12}$

Schritt	$\mathbf{x}$	$g(\mathbf{x})$
1	$[-1.20, 1.00]^T$	24.20
2	$[-1.18, 1.38]^T$	4.73
3	$[-0.93, 0.81]^T$	4.09
4	$[-0.78, 0.59]^T$	3.23
5	$[-0.46, 0.11]^T$	3.21
$\vdots$	$\vdots$	$\vdots$
12	$[1.00, 1.00]^T$	$4.93 \cdot 10^{-28}$

Abbildung 4: Verlauf von GlobNewton für  $g$  bei einer Genauigkeit von  $10^{-12}$

### Aufgabe 3

Die Hesse-Matrizen der beiden Funktionen  $f$  und  $g$  ist stetig und kontinuierlich, d.h. es kann in beiden Fällen vom Zutreffen der Lipschitz-Bedingung

$$\|H(x) - H(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

ausgegangen werden. Weiterhin enthalten beide Funktionen keine mehrfachen Nullstellen durch die das Newton-Verfahren gebremst werden könnte. Aufgrundessen konvergieren beide Funktionen lokal-quadratisch (sollte die

Hesse-Matrix eine Abstiegsrichtung liefern). Global gesehen konvergiert das Newton-Verfahren je nach Schrittweitenstrategie (ob effizient oder nicht) und Startwert entweder gar nicht aufgrund zu kleiner Schrittweiten (z.B. normales Armijo-Verfahren) oder zumindest nur superlinearer. Die lokale quadratische Konvergenz der Himmelblau-Funktion kann man in (1) und (2) zwischen Schritt 3 und 4 bzw. 2 und 3 gut erkennen. Da beide Funktionen nicht quadratisch sind, konvergiert das Verfahren nicht in einem einzigen Schritt.

Bei Quasi-Newton-Verfahren mit approximierter Hesse-Matrix und effizienter Schrittweitenstrategie kann man global gesehen von einer superlinearen Konvergenz für beide Funktionen  $f$  und  $g$  ausgehen. Im Gegensatz zum reinen Newton-Verfahren kann man die Update-Formeln der Hesse-Matrix so wählen, dass eine Abstiegsrichtung entsteht. Broyden et al. haben 1973 in *On the Local and Superlinear Convergence of Quasi-Newton Methods* gezeigt, dass die Fehler in der Approximation von  $H_k$  begrenzt sind und sich nicht unbeschränkt erhöhen und daraus die superlineare Konvergenz abgeleitet werden kann.

Weiterhin sind beide Funktionen nicht quadratischer Natur ansonsten könnte die Schrittweite ggf. exakt berechnet werden.

#### Aufgabe 4

Das Optimierungsproblem

$$\min -2x_1 - 3x_2 - 4x_3$$

unter den Nebenbedingungen

$$x_1 + x_2 + x_3 \leq 4$$

$$3x_2 + x_3 \leq 6$$

$$x_1 \leq 2$$

$$x_3 \leq 3$$

$$x_i \geq 0, i \in \{1, 2, 3\}$$

hat folgende Normalform

$$\min -2x_1 - 3x_2 - 4x_3$$

$$x_1 + x_2 + x_3 + x_4 = 4 \text{ (I)}$$

$$3x_2 + x_3 + x_5 = 6 \text{ (II)}$$

$$x_1 + x_6 = 2 \text{ (III)}$$

$$x_3 + x_7 = 3 \text{ (IV)}$$

$$x_i \geq 0, i \in \{1, 2, 3, 4, 5, 6, 7\} \text{ (V)}$$

## Aufgabe 5

	$\begin{bmatrix} 2 \\ 1 \\ 3 \\ -3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 3 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 3 \\ 0 \\ 3 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \\ 2 \\ 0 \\ 2 \end{bmatrix}$
(I)	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
(II)	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
(III)	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
(IV)	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
(V)	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

Die Vektoren  $x^{(3)}$ ,  $x^{(4)}$  und  $x^{(5)}$  sind gültige Basisvektoren während  $x^{(1)}$  einen negativen Eintrag enthält sowie nicht alle Nebenbedingungen erfüllt.  $x^{(2)}$  erfüllt ebenfalls nicht alle Nebenbedingungen.

## Aufgabe 6

Das Optimierungsproblem lässt sich in Matrixschreibweise als lineares Gleichungssystem der Form  $Ax = b$  schreiben

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot x = \begin{bmatrix} 4 \\ 6 \\ 2 \\ 3 \end{bmatrix}$$

Aus dem angegebenen Basisvektor  $x = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \\ 4 \\ 0 \\ 1 \end{bmatrix}$  kann man die Indexmengen

$B = \{1, 3, 5, 7\}$  und  $N = \{2, 4, 6\}$  ablesen.  $B$  enthält die Indizes bei denen  $x_i \neq 0$  sind während  $N$  gerade die Einträge enthält, bei denen  $x_i = 0$  sind. Daraus wiederum kann man  $A_B$  und  $A_N$  bilden

$$A_B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad A_N = \begin{bmatrix} 1 & 1 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

die gerade die Spalten aus  $A$  enthalten, die in der jeweiligen Indexmenge angegeben sind.

Mit

$$A_B^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 1 \end{bmatrix}$$

kann man nun  $\Gamma$  berechnen

$$\Gamma = A_B^{-1} \cdot A_N = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & -1 \\ 2 & -1 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

und  $\beta_B = A_B^{-1} \cdot b = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 1 \end{bmatrix}$ . Mit zusätzlichem  $c = \begin{bmatrix} -2 \\ -3 \\ -4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$  und  $c_B = \begin{bmatrix} -2 \\ -4 \\ 0 \\ 0 \end{bmatrix}$ ,

$c_N = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$  lässt sich  $\xi = \Gamma^T c_B - c_N = \begin{bmatrix} -1 \\ -4 \\ 2 \end{bmatrix}$  berechnen und das Tableau aufstellen (Pivot-Element farbig hinterlegt):

	$x_2$	$x_4$	$x_6$		
$x_1$	0	0	1	2	2
$x_3$	1	1	-1	2	/
$x_5$	2	-1	1	4	4
$x_7$	-1	-1	1	1	1
	-1	-4	2	-12	

$q = 6, p = 7$ , es werden also die Elemente  $x_7$  und  $x_6$  getauscht. Daraus entstehen die neuen Index-Mengen  $B = \{1, 3, 5, 6\}$  und  $N = \{2, 4, 7\}$

Daraus lässt sich wieder bestimmen

$$A_B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} A_N = \begin{bmatrix} 1 & 1 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} A_B^{-1} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 1 \end{bmatrix}$$

und draus wiederum

$$\Gamma = A_B^{-1}A_N = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 1 \\ 3 & 0 & -1 \\ -1 & -1 & 1 \end{bmatrix} \quad \beta_B = A_B^{-1}b = \begin{bmatrix} 1 \\ 3 \\ 3 \\ 1 \end{bmatrix}$$

Mit  $c_B = \begin{bmatrix} -2 \\ -4 \\ 0 \\ 0 \end{bmatrix}$ ,  $c_N = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$  kann man bestimmen  $\xi = \Gamma^T c_B - c_N = \begin{bmatrix} 1 \\ -2 \\ -2 \end{bmatrix}$

und daraus das neue Tableau aufstellen

	$x_2$	$x_4$	$x_7$	
$x_1$	1	1	-1	1
$x_3$	0	0	1	3
$x_5$	3	0	-1	3
$x_6$	-1	-1	1	1
	1	-2	-2	-14

Selbiges erhält man durch die Update-Formel:

	$x_2$	$x_4$	$x_7$	
$x_1$	$0 - \frac{1 \cdot (-1)}{1}$	$0 - \frac{1 \cdot (-1)}{1}$	$\frac{-1}{1}$	$2 - \frac{1 \cdot 1}{1}$
$x_3$	$1 - \frac{-1 \cdot (-1)}{1}$	$1 - \frac{-1 \cdot (-1)}{1}$	$-1 \frac{-1}{1}$	$2 - \frac{-1 \cdot 1}{1}$
$x_5$	$2 - \frac{1 \cdot (-1)}{1}$	$-1 - \frac{1 \cdot (-1)}{1}$	$\frac{-1}{1}$	$4 - \frac{1 \cdot 1}{1}$
$x_6$	$\frac{-1}{1}$	$\frac{-1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$
	$-1 - \frac{2 \cdot (-1)}{1}$	$-4 - \frac{2 \cdot (-1)}{1}$	$-\frac{2}{1}$	$-12 - \frac{2}{1}$

Aktuell ist  $x = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 0 \\ 3 \\ 1 \\ 0 \end{bmatrix}$  mit  $f(x) = -14$ .

## Aufgabe 7

Wenn man in Matlab definiert

```
f = [-2, -3, -4];
A = [1, 1, 1; 0, 3, 1; 1, 0, 0; 0, 0, 1];
lower_bound = [0, 0, 0];
b = [4; 5; 2; 3];
```

und `linprog` so aufruft (primal-simplex hat in meiner Version 2020a nicht funktioniert, konkrete Implementierung siehe `Projekt_4.m`):

```
options = optimoptions("linprog", "OptimalityTolerance", 1e-8,  
    "Algorithm", "dual-simplex");  
linprog(f, A, b, [], [], lower_bound, [], options);
```

dann erhält man  $x = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ 3 \end{bmatrix}$  mit  $f(x) = -\frac{44}{3}$ .