2 Statische Optimierung mit Beschränkungen

In dieser Übung soll das Minimum der Kostenfunktion $f(\mathbf{x})$

$$\begin{aligned} &\min_{\mathbf{x}} \; f(\mathbf{x}) \\ \text{u.B.v.} \quad &\mathbf{g}(\mathbf{x}) = \mathbf{0} \\ &\mathbf{h}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{2.1}$$

bezüglich der Optimierungsvariablen $\mathbf{x}$ unter Gleichungsbeschränkungen $\mathbf{g}(\mathbf{x})$ und Ungleichungsbeschränkungen $\mathbf{h}(\mathbf{x})$ berechnet werden. Zur Vorbereitung für die Übung werden folgende Punkte empfohlen:

1. Machen Sie sich mit der MATLAB-Routine `fmincon` zur Lösung von beschränkten Optimierungsproblemen und den darin verwendeten Algorithmen vertraut.

2. Lesen Sie die Theorie zu beschränkten Optimierungsproblemen, allen voran die Lösung mittels SQP-Verfahren, durch.

Folgende Aufgaben sollen in der Übungseinheit gelöst werden:

1. Programmieren Sie in MATLAB eine Funktion `xmin = callsqp(x0)`, die mittels des *SQP*-Verfahrens das Optimierungsproblem (2.1) löst. Verwenden Sie hierfür die von MATLAB zur Verfügung gestellte Funktion `quadprog`, um das unterlagerte quadratische Programm zu lösen. Stellen Sie den Wert der Kostenfunktion f, deren Gradienten $\mathbf{df}$ und Hessematrix $\mathbf{ddf}$ in einer Funktion `[f,df,ddf] = calcf(x)` und die Werte der Ungleichungsbeschränkungen $\mathbf{h}$ und der Gleichungsbeschränkungen $\mathbf{g}$ sowie der entsprechenden Jacobimatrizen $\mathbf{dh} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ und $\mathbf{dg} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ in einer Funktion `[h,g,dh,dg,ddh,ddg] = calchg(x)` zur Verfügung. `ddh` und `ddg` bezeichnen geeignete Datenstrukturen, die die Hessematrizen $\left(\nabla^2 h_i\right)(\mathbf{x})$ und $\left(\nabla^2 g_i\right)(\mathbf{x})$ enthalten. Verwenden Sie die konstante Schrittweite $\alpha_k = 1$.

2. Testen Sie Ihre Funktion anhand der *Styblinski-Tang*-Funktion

$$f(x_1, \ldots, x_n) = \frac{1}{2} \sum_{i=1}^{n} \left(x_i^4 - 16x_i^2 + 5x_i \right)$$

für $n = 2$ Optimierungsvariablen. Verwenden Sie außerdem die Ungleichungsbeschränkung

$$\bar{h}(x_1, x_2) = (x_1 - 2.75)^2 + (x_2 - 2.75)^2 \geq 1, \tag{2.2}$$

wie in Abbildung 2.1 dargestellt.

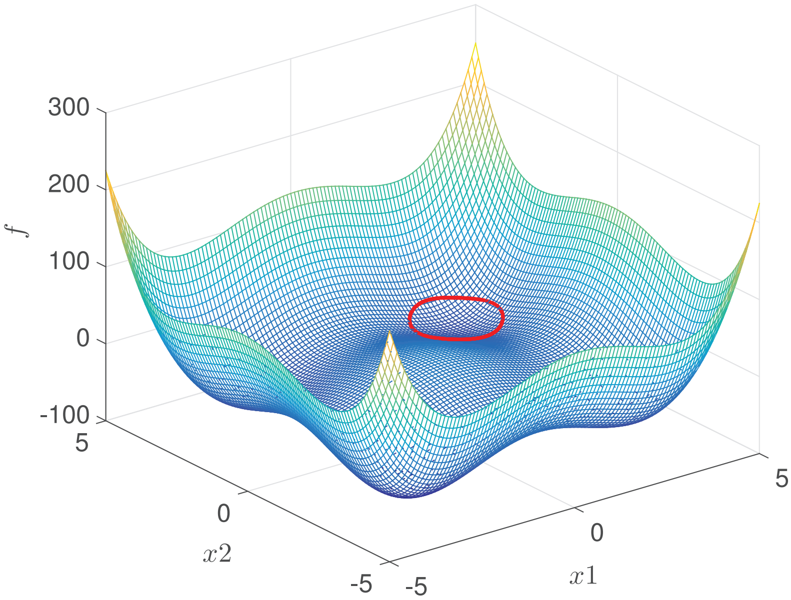

Abbildung 2.1: *Styblinski-Tang*-Funktion mit der Ungleichungsbeschränkung (2.2).

Vergleichen Sie die Ergebnisse Ihrer Implementierung mit dem Algorithmus `sqp` der MATLAB-Funktion `fmincon`. Verwenden Sie hierbei verschiedene Startwerte und lassen Sie sich den Verlauf der Iterationen in die Kostenfunktion einzeichnen.

- Konvergieren die Verfahren zu einem globalen/lokalen Minimum?
- Werden die Beschränkungen eingehalten?
- Welche Methode weist die besseren Konvergenzeigenschaften auf? Wie könnte man die Konvergenz verbessern?

3. Finden Sie mittels `fmincon` und eines Ihnen geeignet erscheinenden Verfahrens ein (lokales) Minimum der in der Funktion `[f,df] = obj_func(x)` zur Verfügung gestellten Kostenfunktion unter Berücksichtigung von nichtlinearen Beschränkungen in Form der Funktion `[h,g,dh,dg] = constr_func(x)`. Implementieren Sie zusätzlich die Beschränkungen

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \mathbf{x} = \begin{bmatrix} \pi \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.3}$$

Beachten Sie dazu folgende Hinweise:

- Die Anzahl der Optimierungsvariablen beträgt $n = 250$.
- Verwenden Sie als Startpunkt der Iterationen $\mathbf{x}_0 = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix}^{\mathrm{T}}$.
- Sie können die Ergebnisse mittels der Funktion `plotsol(x)` darstellen.