Fehlerliste von 4. Übung

Krendl Wolfgang

Februar 2011

4 Numerische Lösung von linearen Gleichungssystemen und Optimierungsproblemen

4.1 Iterative Lösung von linearen Gleichungssystemen

- Nicht alle Elemente der Matrix abspeichern, sondern nur die der Haupt- und der beiden Nebendiagonalen.
- Fehlerhafte Berechnung des Lastvektors. In den Fällen wo für die Berechnung des Lastvektors die Gauß-1 Formel verwendet wird, gilt z.B. in Matalab

```
fh(1) = h/2*(sin(k*pi*h/2);
for i = 1:1:n
    fh(i+1) =  h/2*(sin(k*pi*(x(i)-h/2))) + sin(k*pi*(x(i)+h/2)));
end
fh(n+1) = h/2*(sin(k*pi*h*(n-1/2));
```

  d.h. $f(i+1) = \tilde{f}(\frac{i}{n})$.

- Bei zu langsamer Konvergenz, sollten verschiedene Startvektoren x_0 bzw. verschiedene Werte für ω im SOR- Verfahren getestet werden.
- Beim SOR- Verfahren wurde bei der Berechnung der $(k+1)$- ten Iterierten anstatt

$$\begin{aligned}\tilde{x}_i^{(k+1)} &= \frac{1}{a_{i,i}}\Big(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}\Big),\\ x_i^{(k+1)} &= x_i^{(k)} + \omega\big(\tilde{x}_i^{(k+1)} - x_i^{(k)}\big),\\ &\quad \text{für} \quad i = 1,\dots,n,\end{aligned}$$

  wurde

$$\begin{aligned}\tilde{x}_i^{(k+1)} &= \frac{1}{a_{i,i}}\Big(b_i - \sum_{j=1}^{i-1} a_{i,j}\tilde{x}_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}\Big),\\ &\quad \text{für} \quad i = 1,\dots,n,\\ x^{(k+1)} &= x^{(k)} + \omega\big(\tilde{x}^{(k+1)} - x^{(k)}\big),\end{aligned}$$

  implementiert.

4.2 Ein restrigiertes Optimierungsproblem

- Aufschreiben der Lagrange Funktion:

  $$\mathbb{L}(x_1, x_2, x_3, \lambda_1, \lambda_2) := x_1 - x_2 - x_3 + \lambda_1(x_1^2 + 2x_2^2 - 1) + \lambda_2(3x_1 - 4x_3)$$

- Angabe des *Satzes 4.16: (Notwendige Extremalbedingungen)*,und Prüfung der Voraussetzungen:
  - $m = 2 < n = 3$
  - Für jedes lokale Extremum $x^\star$ von f unter der Nebenbedingung $c(x) = 0$ gilt

    $$rang(c'(x^\star)) = \begin{pmatrix} 2x_1^\star & 4x_2^\star & 0 \\ 3 & 0 & -4 \end{pmatrix} = m = 2.$$

    da der Fall $x_1^\star = x_2^\star = 0$ nicht auftretten kann, weil sonst $c_1(x^\star) = -1 \neq 0$.

- Angabe des Kuhn-Tucker Systems:

  $$\nabla_{x,\lambda}\mathbb{L}(x, \lambda) = 0$$
  $$\Updownarrow$$
  $$\begin{aligned} \frac{\partial \mathbb{L}}{\partial x_1}(x, \lambda) &= 1 + 2\lambda_1 x_1 + 3\lambda_2 = 0 \\ \frac{\partial \mathbb{L}}{\partial x_2}(x, \lambda) &= -1 + 4\lambda_1 x_2 = 0 \\ \frac{\partial \mathbb{L}}{\partial x_3}(x, \lambda) &= -1 - 4\lambda_2 = 0 \\ \frac{\partial \mathbb{L}}{\partial \lambda_1}(x, \lambda) &= x_1^2 + 2x_2^2 - 1 = 0 \\ \frac{\partial \mathbb{L}}{\partial \lambda_2}(x, \lambda) &= 3x_1 - 4x_3 = 0 \end{aligned}$$

- Händisches Lösen des Kuhn-Tucker Systems.
- Argumentation, warum das lokale Minimum, auch ein globales Minimum ist: *Für jede konvexe Funktion f gilt, dass jedes lokale Minimum auch ein globales Minimum ist.*

4.3 Ein Optimalsteuerproblem

Dieses Beispiel wurde leider von fast keinem Studenten gelöst. Im folgenden sind grob die Antworten auf die einzelnen Teilaufgaben gegeben:

4.3.1 FEM - Diskretisierung

Für das diskretisierte Optimalsteuerproblem erhält man:

$$\min_{(\underline{y},\underline{u}) \in \mathbb{R}^{(n_s+1)\times(n_c+1)}} \frac{1}{2}(M_S(\underline{y} - \underline{y}_d), \underline{y} - \underline{y}_d)_{\mathbb{R}^{(n_s+1)}} + \frac{\alpha}{2}(M_C\underline{u}, \underline{u})_{\mathbb{R}^{(n_c+1)}}$$

mit NB: $A\underline{y} = M_{SC}\underline{u}$,

wobei gilt:

- $M_S = [(\phi_j, \phi_i)_{L_2(\Omega)}]_{i,j=0,\dots,n_s}$
- $M_C = [(\phi_j, \phi_i)_{L_2(\Omega)}]_{i,j=0,\dots,n_c}$
- $M_{SC} = [(\phi_j, \phi_i)_{L_2(\Omega)}]_{i=0,\dots,n_s,j=0,\dots,n_c}$
- $A = [(\phi_j, \phi_i)_{L_2(\Omega)} + (\nabla\phi_j, \nabla\phi_i)_{L_2(\Omega)}]_{i,j=0,\dots,n_s}$
- $\underline{y}_d = (y_d(x_i))_{i=0,\dots,n_c}$

4.3.2 KKT - System

Lagrangefunktion:

$$\mathbb{L}(\underline{\mathrm{y}}, \underline{\mathrm{u}}, \underline{\lambda}) := \frac{1}{2}(M_S(\underline{\mathrm{y}} - \underline{\mathrm{y}}_d), \underline{\mathrm{y}} - \underline{\mathrm{y}}_d)_{\mathbb{R}^{(n_s+1)}} + \frac{\alpha}{2}(M_C\underline{\mathrm{u}}, \underline{\mathrm{u}})_{\mathbb{R}^{(n_c+1)}} + (\underline{\lambda}, M_{SC}\underline{\mathrm{u}} - A\underline{\mathrm{y}})_{\mathbb{R}^{n_s}}$$

Kuhn-Tucker System:

$$\begin{aligned}
\nabla_{(\underline{\mathrm{y}},\underline{\mathrm{u}},\underline{\lambda})}\mathbb{L}(\underline{\mathrm{y}}, \underline{\mathrm{u}}, \underline{\lambda}) &= 0 \\
&\Updownarrow \\
\nabla_{\underline{\mathrm{y}}}\mathbb{L}(\underline{\mathrm{y}}, \underline{\mathrm{u}}, \underline{\lambda}) &= M_S(\underline{\mathrm{y}} - \underline{y}_d) - A^T\underline{\lambda} = 0 \\
\nabla_{\underline{\mathrm{u}}}\mathbb{L}(\underline{\mathrm{y}}, \underline{\mathrm{u}}, \underline{\lambda}) &= \alpha M_C\underline{\mathrm{u}} + M_{SC}^T\underline{\lambda} = 0 \\
\nabla_{\underline{\lambda}}\mathbb{L}(\underline{\mathrm{y}}, \underline{\mathrm{u}}, \underline{\lambda}) &= A\underline{\mathrm{y}} - M_{SC}\underline{\mathrm{u}} = 0
\end{aligned}$$

Zur Bestimung von $(\underline{\mathrm{y}}, \underline{\mathrm{u}}, \underline{\lambda})$, gilt es also das folgende lineare Gleichungssystem zu lösen:

$$\underbrace{\begin{pmatrix} \alpha M_C & 0 & M_{SC}^T \\ 0 & M_S & -A^T \\ M_{SC} & -A & 0 \end{pmatrix}}_{S} \begin{pmatrix} \underline{u} \\ \underline{y} \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} 0 \\ M_S\underline{y}_d \\ 0 \end{pmatrix} \tag{4.1}$$

Bemerkung:

1. Die Systemmatrix S ist eine **dünnbesetzte**, **indefinite** Matrix, und somit das PCG- Verfahren nicht anwendbar. Ein mögliches Verfahren zur Lösung dieses indefiniten Systems, ist das Präkonditionierte MINRES (Minimal Residual) Verfahren.

2. Für den Fall $n = n_s = n_c$, vereinfacht sich das obrige System 4.1 zu:

$$\underbrace{\begin{pmatrix} \frac{1}{\alpha}M & A^T \\ A & -M \end{pmatrix}}_{S} \begin{pmatrix} \underline{\lambda} \\ \underline{y} \end{pmatrix} = \begin{pmatrix} 0 \\ -M_S\underline{y}_d \end{pmatrix}$$

   wobei

   - $M = M_S = M_C = M_{SC}$, und
   - $\underline{u} = -\frac{1}{\alpha}\underline{\lambda}$.