Tensorstruktur der Zellmatrizen bei finiten Elementen

Enes Witwit
Universität Heidelberg

29. Mai 2017

Contents

1. Einleitung
2. Theorie
3. Pseudoinverse
4. Effiziente Berechnung
5. Resultate

1 Einleitung

2 Theorie

3 Pseudoinverse

4 Effiziente Berechnung

5 Resultate

Hochleistungsrechnen

Ziel Löse ein sehr komplexes Problem.

Lösungsansatz Teile das komplexe Problem auf in Subprobleme (Parallelisierung).

Initial-Problem

$$v = A(u)$$

A, möglicherweise nichtlinearer, finite Elemente Operator, der Vektor u als Input nimmt.

Probleme

- A wird unter Umständen sehr groß $\rightarrow$ Speicherplatz.
- A liegt nicht mehr im Cache $\rightarrow$ Abrufen der Elemente von A zeitintesiv.
- Berechnung des Matrix-Vektor-Produkts komplex

Divide and Conquer

Nach [MK12] können wir die Ursprungsgleichung umformen zu

$$v = A(u) = \sum_{k=1}^{n_{cells}} P_k^T A_k P_k u \,.$$

P_k kümmert sich um die Einordnug der lokalen Freiheitsgrade in die globalen Freiheitsgrade.

$$\begin{aligned} v_k &= A_k u_k \\ A_k^{-1} v_k &= u_k \end{aligned}$$

Inverse/Pseudoinverse

1. Tensorstruktr und Summenfaktorisierung.
2. Singulärwertzerlegung höherer Ordnung (HOSVD).

1. Einleitung
2. Theorie
3. Pseudoinverse
4. Effiziente Berechnung
5. Resultate

Higher Order Singular Value Decomposition

Higher Order Singular Value Decomposition

Definition Tensor

Ein Tensor ist eine multidimensionale Matrix $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.
Die Ordnung ist die Anzahl der Dimensionen, in diesem Fall N.

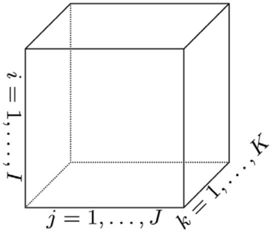

Abbildung: Tensor dritter Ordnung $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ [TK09, 456]

Tensoren: Definitionen und Eigenschaften

Bemerkung Tensor-Charakteristiken
Es sei $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ ein Tensor.

a) Den Tensor $\boldsymbol{\mathcal{X}}$ nennt man kubisch genau dann, wenn $I_i = I_j$ für alle i, j.
b) Einen kubischen Tensor nennt man supersymmetrisch genau dann, wenn die Elemente des Tensors konstant bleiben unter jeglicher Permutation der Indizes.
c) Einen Tensor nennt man stückweise symmetrisch, wenn die Elemente konstant bleiben unter der Permutation von mindestens zwei Indizes.

Definition Diagonal
Den Tensor $\boldsymbol{\mathcal{X}}$ nennt man diagonal, wenn $x_{i_1,\dots,i_N} \neq 0$ genau dann wenn
$i_1 = \cdots = i_N$.

Definition Diagonal
Den Tensor $\boldsymbol{\mathcal{X}}$ nennt man diagonal, wenn $x_{i_1,\dots,i_N} \neq 0$ genau dann wenn
$i_1 = \cdots = i_N$.

Definition Faser
Eine Faser ist das multidimensionale Analog zu Matrixspalten und Matrixzeilen. Wir definieren eine Faser, indem wir jeden Index abgesehen von einem festhalten.

Tensor-Entfaltung

Wir wollen unseren Tensor als Matrix darstellen.

Tensor-Entfaltung

Wir wollen unseren Tensor als Matrix darstellen.

- $Mode - n$ Entfaltung von $\boldsymbol{\mathcal{X}}$ wird mit $\mathbf{X}_{(n)}$ notiert.
- Ordnet die $mode - n$ Fasern in die Spalten der Ergebnismatrix.
- Formal ist es eine Abbildung des Indize N-tupels $(i_1, \ldots, i_N)$ auf das Matrixindize-Tupel (i_n, j)

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \text{ mit } J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m \,.$$

Tensor-Matrix Produkt

Definition $n - mode$ Produkt
Das n-mode Produkt des Tensors $\boldsymbol{\mathcal{X}}$ mit einer Matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ wird mit $\boldsymbol{\mathcal{X}} \times_n \mathbf{U}$ notiert. Die Ergebnismatrix hat die Größe $I_1 \times \dots I_{n-1} \times J \times I_{n+1} \times \dots I_N$

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{U})_{i_1 \dots i_{n-1} \, j \, i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \dots i_N} \, u_{j \, i_n}$$

Tensor-Matrix Produkt

Definition $n - mode$ Produkt
Das n-mode Produkt des Tensors $\boldsymbol{\mathcal{X}}$ mit einer Matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ wird mit $\boldsymbol{\mathcal{X}} \times_n \mathbf{U}$ notiert. Die Ergebnismatrix hat die Größe $I_1 \times \ldots I_{n-1} \times J \times I_{n+1} \times \ldots I_N$

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{U})_{i_1 \ldots i_{n-1}\, j\, i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \ldots i_N} u_{j\, i_n}$$

Jedes $n - mode$ Produkt kann mit Hilfe von entfalteten Tensoren äquivalent ausgedrückt werden.

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{U} \Longleftrightarrow \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$$

Higher Order Singular Value Decomposition

Ziel Sinnvolle Zerlegung eines Tensors.

- Für Matrizen gibt es die Singulärwertzerlegung. ($M = U\Sigma V^T$)
- Für Tensoren haben wir die Higher Order Value Decomposition (HOSVD) oder auch Tucker Decomposition genannt.
- Die HOSVD ist eine multidimensionale Hauptkomponentenanalyse.
- Zerlegt Tensor in einen Kerntensor (Core Tensor), welcher das pendant zum Σ ist und mehreren orthogonalen Matrizen.

Higher Order Singular Value Decomposition

Allgemein ist die HOSVD des Tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ gegeben durch

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 A^{(1)} \ldots \times_N A^{(N)} .$$

Man kann äquivalent die HOSVD, wie in [TK09, 462], auch mit entfalteten Tensoren wie folgt angeben

$$\mathbf{X}_{(n)} = A^{(n)} \mathbf{G}_{(n)} \left(A^{(N)} \otimes \ldots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)} \right)^T .$$

Beispiel

Es sei $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$. Dann kann man den Tensor $\boldsymbol{\mathcal{X}}$ zerlegen in

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{G}} \times_1 A \times_2 B \times_3 C \,, \tag{1}$$

wobei $A \in \mathbb{R}^{I \times P}$, $B \in \mathbb{R}^{J \times Q}$ und $C \in \mathbb{R}^{K \times R}$ die orthogonalen Faktormatrizen sind. Der Tensor $\mathcal{G}$ bezeichnet den Kerntesor und zeigt wie hoch die Korrelation zwischen den verschiedenen Komponenten ist.

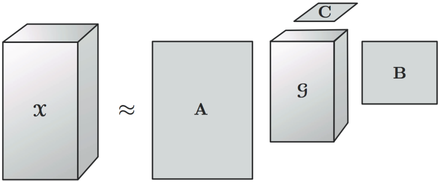

Abbildung: HOSVD eines Tensors dritter Ordnung [TK09, 475]

Berechnung

Die Berechnung der HOSVD von $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ funktioniert wie folgt:

1. Berechne die mode-k Entfaltungen $\mathbf{X}_{(k)}$ für alle k.
2. Berechne die Singulärwertzerlegung $\mathbf{X}_{(k)} = U_k \Sigma_k V_k^T$ und speichere U_k.
3. Der Kerntensor $\boldsymbol{\mathcal{G}}$ ergibt sich aus der Projektion des Tensors auf die Tensorbasis geformt von den Faktormatrizen $\{ U_k \}_{k=1}^N$ also $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{X}} \times_{n=1}^N U_n^T$.

Kronecker Produkt

Lemma (Invertieren des Kronecker Produkts)
Es seien $A \in \mathbb{R}^{i \times i}$ und $B \in \mathbb{R}^{j \times j}$ invertierbar, so ist auch $(A \otimes B)$ invertierbar.

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}\,.$$

Für die Moore Penrose Pseudoinversen gilt analog

$$(A \otimes B)^{+} = A^{+} \otimes B^{+}\,.$$

Lemma (Transponieren)
Es seien A, B beliebige Matrizen. Es gilt

$$(A \otimes B)^T = A^T \otimes B^T .$$

Lemma (Matrixprodukt und Kronecker Produkt)
Es seien A, B, C, D Matrizen, deren Matrizenprodukte AC und BD definiert sind. Dann gilt

$$AC \otimes BD = (A \otimes B)(C \otimes D).$$

1. Einleitung
2. Theorie
3. Pseudoinverse
4. Effiziente Berechnung
5. Resultate

Lokale Massenmatrix

$$M_{ik} = \int\limits_T \varphi_i(\mathbf{x})\, \varphi_j(\mathbf{x})\, d\mathbf{x}$$

Elementsteifigkeitsmatrix der Laplace Bilinearform

$$V_{ij} = \int\limits_T \nabla\varphi_i(\mathbf{x})\, \nabla\varphi_j(\mathbf{x}) d\mathbf{x}$$

- T sei die Referenzzelle für Rechtecke
- $\varphi_i(\mathbf{x})$ sei eine zweidimensionale reelle Basisfunktion des diskreten Raumes V_n mit $\mathbf{x} = (x, y)$.

Tensorstruktur der Ansatzfunktionen

$$\varphi_i^{2D}(\mathbf{x}) = \varphi_{i_1+(N+1)i_2}^{2D}(x,y) = \varphi_{i_1}^{1D}(x)\varphi_{i_2}^{1D}(y),$$

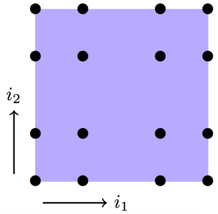

Abbildung: [Tea, 3]

Ziel Definiere Index-Transformation p, sodass $p(i) = (i_1, i_2)$.

Index-Transformation

Das Inverse der Transformation ist gegeben durch

$$p^{-1}(i_1, i_2) = i_1 + (N+1)i_2 = i \tag{2}$$

Extrahiere i_1 durch

$$\begin{aligned} i \ (mod(N+1)) &= p^{-1}(i_1, i_2) \ (mod(N+1)) \\ &= i_1 + \underbrace{(N+1)i_2}_{0} \ (mod(N+1)) \\ &= i_1 \end{aligned}$$

Nutze (2) um i_2 zu erhalten

$$i_2 = \frac{i - i_1}{N+1}.$$

Index-Transformation

$$\boxed{p(i) = \left(i \pmod{N+1}, \frac{i - (i \pmod{N+1})}{N+1}\right)}$$

Tensorstruktur der Massenmatrix

Es seien $\mathbf{x}_q = (x_{q1}, x_{q2})$ die Stützstellen und $\mathbf{w}_q = w_{q1} w_{q2}$ die Gewichte der Gauss Quadratur.

$$M_{ij} = \int\limits_T \varphi_i(\mathbf{x})\, \varphi_j(\mathbf{x})\, d\mathbf{x}$$

Tensorstruktur der Massenmatrix

Es seien $\mathbf{x}_q = (x_{q1}, x_{q2})$ die Stützstellen und $\mathbf{w}_q = w_{q1} w_{q2}$ die Gewichte der Gauss Quadratur.

$$\begin{aligned} M_{ij} &= \int\limits_T \varphi_i(\mathbf{x})\,\varphi_j(\mathbf{x})\,d\mathbf{x} \\ &\approx \sum_{q=1}^{Q} \mathbf{w}_q \; \varphi_i(\mathbf{x}_q)\,\varphi_j(\mathbf{x}_q) \end{aligned}$$

Tensorstruktur der Massenmatrix

Es seien $\mathbf{x}_q = (x_{q1}, x_{q2})$ die Stützstellen und $\mathbf{w}_q = w_{q1} w_{q2}$ die Gewichte der Gauss Quadratur.

$$
\begin{aligned}
M_{ij} &= \int\limits_T \varphi_i(\mathbf{x})\, \varphi_j(\mathbf{x})\, d\mathbf{x} \\
&\approx \sum_{q=1}^{Q} \mathbf{w}_q \;\varphi_i(\mathbf{x}_q)\, \varphi_j(\mathbf{x}_q) \\
&= \sum_{q_1=1}^{Q_{1D}} \sum_{q_2=1}^{Q_{1D}} \varphi_{i_1}(x_{q1})\varphi_{i_2}(x_{q2})\varphi_{j_1}(x_{q1})\varphi_{j_2}(x_{q2})\, w_{q1} w_{q2} \\
&= \sum_{q_1=1}^{Q_{1D}} w_{q1}\varphi_{i_1}(x_{q1})\varphi_{j_1}(x_{q1}) \sum_{q_2=1}^{Q_{1D}} w_{q2}\varphi_{i_2}(x_{q2})\varphi_{j_2}(x_{q2})\,.
\end{aligned}
$$

Definiere

- Es sei $\mathcal{N}$ eine Matrix mit $\mathcal{N}_{iq} = \varphi_i(\mathbf{x}_q)$.
- Es sei $\mathcal{W}$ eine Matrix mit $\mathcal{W}_{ii} = \mathbf{w}_i$, sonst Nullen.

Dann können wir die Massenmatrix schreiben als

$$M = \underbrace{\mathcal{N}\mathcal{W}}_{:=\mathcal{W}_N}\mathcal{N}^T. = \mathcal{W}_N\mathcal{N}^T$$

Definiere

- Es sei $\mathcal{N}$ eine Matrix mit $\mathcal{N}_{iq} = \varphi_i(\mathbf{x}_q)$.
- Es sei $\mathcal{W}$ eine Matrix mit $\mathcal{W}_{ii} = \mathbf{w}_i$, sonst Nullen.

Dann können wir die Massenmatrix schreiben als

$$M = \underbrace{\mathcal{N}\mathcal{W}}_{:=\mathcal{W}_N}\mathcal{N}^T. = \mathcal{W}_N\mathcal{N}^T$$

Nutze Tensorstruktur der Ansatzfunktionen

$$\mathcal{N} = \mathcal{N}^{1D} \otimes \mathcal{N}^{1D}$$
$$\mathcal{W}_N = \mathcal{W}_N^{1D} \otimes \mathcal{W}_N^{1D}$$

Tensorstruktur der Massenmatrix

$$
\begin{aligned}
M &= \mathcal{W}_N \mathcal{N}^T \\
&= (\mathcal{W}_N^{1D} \otimes \mathcal{W}_N^{1D})(\mathcal{N}^{1D} \otimes \mathcal{N}^{1D})^T \\
&= (\mathcal{W}_N^{1D} \otimes \mathcal{W}_N^{1D})((\mathcal{N}^{1D})^T \otimes (\mathcal{N}^{1D})^T) \\
&= (\mathcal{W}_N^{1D} (\mathcal{N}^{1D})^T) \otimes (\mathcal{W}_N^{1D} (\mathcal{N}^{1D})^T)
\end{aligned}
$$

Pseudoinverse der Massenmatrix

$$
\begin{aligned}
M^{+} &= [(\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T) \otimes (\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T)]^{+} \\
&= (\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T)^{+} \otimes (\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T)^{+}
\end{aligned}
$$

Tensorstruktur der Laplace Bilinearform

$$
\begin{aligned}
V_{ij} &= \int_T \nabla\varphi_i(\mathbf{x})\,\nabla\varphi_j(\mathbf{x})d\mathbf{x} \\
&= \int_T (\partial_{x_1}\varphi_i(\mathbf{x})\partial_{x_1}\varphi_j(\mathbf{x})) + (\partial_{x_2}\varphi_i(\mathbf{x})\partial_{x_2}\varphi_j(\mathbf{x}))\,d\mathbf{x} \\
&= \int_T (\partial_{x_1}\varphi_i(\mathbf{x})\partial_{x_1}\varphi_j(\mathbf{x})) + (\partial_{x_2}\varphi_i(\mathbf{x})\partial_{x_2}\varphi_j(\mathbf{x}))\,d\mathbf{x} \\
&= \int_T \partial_{x_1}\varphi_i(\mathbf{x})\partial_{x_1}\varphi_j(\mathbf{x})d\mathbf{x} + \int_T \partial_{x_2}\varphi_i(\mathbf{x})\partial_{x_2}\varphi_j(\mathbf{x})\,d\mathbf{x} \\
&= \underbrace{\sum_{q=1}^{(N+1)^2} \mathbf{w}_q\partial_{x_1}\varphi_i(\mathbf{x}_q)\partial_{x_1}\varphi_j(\mathbf{x}_q)}_{K^1} + \underbrace{\sum_{q=1}^{(N+1)^2} \mathbf{w}_q\partial_{x_2}\varphi_i(\mathbf{x}_q)\partial_{x_2}\varphi_j(\mathbf{x}_q)}_{K^2}
\end{aligned}
$$

$$
\begin{aligned}
K_{ij}^1 &= \sum_{q=1}^{(N+1)^2} \mathbf{w}_q \partial_{x_1} \varphi_i(\mathbf{x}_q) \partial_{x_1} \varphi_j(\mathbf{x}_q) \\
&= \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} w_{q1} w_{q2} \partial_{x_1} \varphi_{i1}(x_{q1}) \varphi_{i2}(x_{q2}) \partial_{x_1} \varphi_{j1}(x_{q1}) \varphi_{j2}(x_{q2}) \\
&= \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} w_{q1} w_{q2} \varphi_{i1}'(x_{q1}) \varphi_{i2}(x_{q2}) \varphi_{j1}'(x_{q1}) \varphi_{j2}(x_{q2}) \\
&= \sum_{q_1=1}^{N} w_{q1} \varphi_{i1}'(x_{q1}) \varphi_{j1}'(x_{q1}) \sum_{q_2=1}^{N} w_{q2} \varphi_{i2}(x_{q2}) \varphi_{j2}(x_{q2})
\end{aligned}
$$

Definiere

- Es sei $\widehat{\mathcal{N}}^{1D}$ eine Matrix mit $\widehat{\mathcal{N}}_{ik}^{1D} = \varphi_i'^{1D}(x_k)$
- Dementsprechend ist $\widehat{\mathcal{W}}_N^{1D}$ eine Matrix, die aufgebaut ist wie $\mathcal{W}_N^{1D}$, mit dem Unterschied, dass sie die Evaluationen der ersten Ableitungen der Ansatzfunktionen beinhaltet.

Definiere

- Es sei $\widehat{\mathcal{N}}^{1D}$ eine Matrix mit $\widehat{\mathcal{N}}^{1D}_{ik} = \varphi'^{1D}_i(x_k)$
- Dementsprechend ist $\widehat{\mathcal{W}}^{1D}_N$ eine Matrix, die aufgebaut ist wie $\mathcal{W}^{1D}_N$, mit dem Unterschied, dass sie die Evaluationen der ersten Ableitungen der Ansatzfunktionen beinhaltet.

Dann können wir K_1, K_2 schreiben als

$$K_1 = (\widehat{\mathcal{W}}^{1D}_N(\widehat{\mathcal{N}}^{1D})^T) \otimes (\mathcal{W}^{1D}_N(\mathcal{N}^{1D})^T)$$
$$K_2 = (\mathcal{W}^{1D}_N(\mathcal{N}^{1D})^T) \otimes (\widehat{\mathcal{W}}^{1D}_N(\widehat{\mathcal{N}}^{1D})^T)$$

$$\boxed{V = \widehat{\mathcal{W}}_N\widehat{\mathcal{N}}^T \otimes \mathcal{W}_N\mathcal{N}^T + \mathcal{W}_N\mathcal{N}^T \otimes \widehat{\mathcal{W}}_N\widehat{\mathcal{N}}^T.}$$

Problem bei Laplace

Problem Die Addition in der Tensorstruktur macht unseren ersten Ansatz hinfällig.
Idee Vereinfache die Form durch geeignete Basiswahl und dann sehen wir weiter. Wir wählen geeignete Basis, sodass $\mathcal{W}_N \mathcal{N}^T = I_n$.
Folgende Basispolynome bieten sich an:

$$\varphi_i^{1D}(x_k) = \frac{1}{\sqrt{w_i}} l_i(x_k) = \begin{cases} \frac{1}{\sqrt{w_i}} & \text{, wenn } i = k \\ 0 & \text{, sonst.} \end{cases}$$

Die Funktion $l_i(x_k)$ bezeichnet das $i - te$ Lagrange Polynom.

Zeige

$$\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T = \mathcal{N}^{1D}\mathcal{W}^{1D}(\mathcal{N}^{1D})^T = I_n$$

- $\mathcal{N}$ ist Diagonalmatrix mit Einträgen $\dfrac{1}{\sqrt{w_i}}$
- Es gilt $\mathcal{N} = \mathcal{N}^T$.
- $\mathcal{W}$ Diagonalmatrix mit Einträgen w_i in der Diagonalen.

In der Ergebnismatrix steht dann in der Diagonalen

$$(\frac{1}{\sqrt{w_i}})^2 w_i = 1\,,$$

sonst Nullen.

- Definiere $A := \widehat{\mathcal{W}}_N^{1D}(\widehat{\mathcal{N}}^{1D})^T$.
- Wähle Basis so, dass $\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T = I_n$ gilt.

Dann können wir das Matrix-Vektor Produkt Vu darstellen durch

$$y = Vu = [(A \otimes I_n) + (I_n \otimes A)]u.$$

- Definiere $A := \widehat{\mathcal{W}}_N^{1D}(\widehat{\mathcal{N}}^{1D})^T$.
- Wähle Basis so, dass $\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T = I_n$ gilt.

Dann können wir das Matrix-Vektor Produkt Vu darstellen durch

$$y = Vu = [(A \otimes I_n) + (I_n \otimes A)]u.$$

Transformiere u und y zu Matrizen U und Y mit Hilfe der Index-Transformation $p(\cdot)$.

$$Y = VU = [(A \otimes I_n) + (I_n \otimes A)]U = AUI_n^T + I_nUA^T = AU + UA^T.$$

- Definiere $A := \widehat{\mathcal{W}}_N^{1D}(\widehat{\mathcal{N}}^{1D})^T$.
- Wähle Basis so, dass $\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T = I_n$ gilt.

Dann können wir das Matrix-Vektor Produkt Vu darstellen durch

$$y = Vu = [(A \otimes I_n) + (I_n \otimes A)]u.$$

Transformiere u und y zu Matrizen U und Y mit Hilfe der Index-Transformation $p(\cdot)$.

$$Y = VU = [(A \otimes I_n) + (I_n \otimes A)]U = AUI_n^T + I_n UA^T = AU + UA^T.$$

Lyapunow Gleichung

$$Y = AU + UA^T.$$

Lokaler Massetensor

$$M_{i1,i2,j1,j2} = \int\limits_T \varphi_{i1}(x_1)\varphi_{i2}(x_2)\varphi_{j1}(x_1)\varphi_{j2}(x_2)\, d(x_1, x_2)$$

Lokaler Laplace Tensor

$$V_{i1,i2,j1,j2} = \int\limits_T \varphi'_{i1}(x_1)\varphi_{i2}(x_2)\varphi'_{j1}(x_1)\varphi_{j2}(x_2) + \varphi_{i1}(x_1)\varphi'_{i2}(x_2)\varphi_{j1}(x_1)\varphi'_{j2}(x_2)\, d(x_1, x_2).$$

$\rightarrow$ Was ist die Pseudoinverse eines Tensors?

Moore Penrose Pseudoinverse

Definition (Moore Penrose Pseudoinverse)

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^T = AA^+$
4. $(A^+A)^T = A^+A$

Probleme

1. Wir haben kein Tensor-Tensor Produkt.
2. Wir wissen nicht was die Transponierte eines Tensors ist.

Tensor Operationen

Definition (Tensor-Tensor Produkt)
Es seien $\boldsymbol{\mathcal{X}}^1 \in \mathbb{R}^{I_1 \times I_2 \times I_1 \times I_2}$ und $\boldsymbol{\mathcal{X}}^2 \in \mathbb{R}^{I_1 \times I_2 \times I_1 \times I_2}$ Tensoren. Dann definieren wir das Produkt dieser beiden Tensoren elementweise wie folgt

$$ttp(\boldsymbol{\mathcal{X}}^1, \boldsymbol{\mathcal{X}}^2)_{i_1,i_2,j_1,j_2} = \sum_{j_1=1}^{I_1} \sum_{j_2=1}^{I_2} \boldsymbol{\mathcal{X}}^1_{i_1,i_2,j_1,j_2} \boldsymbol{\mathcal{X}}^2_{j_1,j_2,k_1,k_2} \tag{3}$$

Tensor Operationen

Definition (Transponierte eines Tensors)
Es seien $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_1 \times I_2}$ ein Tensor. Dann definieren wir die Transponierte des Tensors elementweise wie folgt

$$\mathcal{X}^T_{i_1\, i_2\, i_1\, j_2} = \mathcal{X}^T_{p^{-1}(i_1,i_2)p^{-1}(j_1,j_2)} = \mathcal{X}_{p^{-1}(j_1,j_2)p^{-1}(i_1,i_2)} = \mathcal{X}_{j_1\, j_2\, i_1\, i_2}$$

Moore Penrose Pseudoinverse für Tensoren

Definition (Moore Penrose Pseudoinverse für Tensoren)

1. $ttp(A, ttp(A^+, A)) - A \quad = 0$
2. $ttp(A^+, ttp(A, A^+)) - A^+ \quad = 0$
3. $(ttp(A, A^+))^T - ttp(A, A^+) = 0$
4. $(ttp(A^+, A))^T - ttp(A^+, A) = 0$

Pseudoinverse der HOSVD

Die HOSVD eines Tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ ergibt

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_{n=1}^{4} U^{(n)}.$$

Die Pseudoinverse des Tensors $\mathcal{X}$ erhält man über

$$\boxed{\boldsymbol{\mathcal{X}}^{+} = \boldsymbol{\mathcal{G}}^{+} \times_{n=1}^{4} {U^{(n)}}^{T}.}$$

1 Einleitung

2 Theorie

3 Pseudoinverse

4 Effiziente Berechnung

5 Resultate

Kronecker Produkt

Wir wollen folgende zwei Aufgaben effizient lösen:

1. $z = (\mathcal{B} \otimes \mathcal{A})y$
2. $z = (\mathcal{C} \otimes \mathcal{B} \otimes \mathcal{A})v$

$z = (\mathcal{B} \otimes \mathcal{A}) y$

$$\mathcal{B} \times \mathcal{A} = \begin{pmatrix} b_{11}\mathcal{A} & \dots & b_{1q}\mathcal{A} \\ \vdots & \ddots & \vdots \\ b_{p1}\mathcal{A} & \dots & b_{pq}\mathcal{A} \end{pmatrix}.$$

1. Geordnete Indexierung
   $y = (y_1, y_2, \dots, y_n, \dots, \dots, y_{(q-1)n+1}, y_{(q-1)n+2}, \dots, y_{qn})^T$
2. Definiere $y^{(i)} = (y_1, y_2, \dots, y_{in})^T$
3. Berechne $w^{(i)} = \mathcal{A} y^{(i)}$
4. Berechne $z_j^{(k)} = \sum_{i=1}^{q} b_{ki} w_j^{(i)}$

for i=1 < N **do**
  for j= 1 < N **do**
    $w_j^{(i)} = y_1^{(i)} a_{j1} + \cdots + y_n^{(i)} a_{jn}$
  end for
end for
for k=1 < N **do**
  for j= 1 < N **do**
    $z_j^{(k)} := w_j^{(1)} b_{k1} + \cdots + w_j^{(q)} b_{kq}$
  end for
end for

$z = (\mathcal{C} \otimes \mathcal{B} \otimes \mathcal{A})v$

$$
\begin{pmatrix}
c_{11} b_{11} \mathcal{A} & \dots & c_{11} b_{1N} \mathcal{A} & \dots & \dots & c_{1N} b_{11} \mathcal{A} & \dots & c_{1N} b_{1N} \mathcal{A} \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
c_{11} b_{N1} \mathcal{A} & \dots & c_{11} b_{NN} \mathcal{A} & \dots & \dots & c_{1N} b_{N1} \mathcal{A} & \dots & c_{1N} b_{NN} \mathcal{A} \\
c_{21} b_{11} \mathcal{A} & \dots & c_{21} b_{1N} \mathcal{A} & \dots & \dots & c_{2N} b_{11} \mathcal{A} & \dots & c_{2N} b_{1N} \mathcal{A} \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
c_{N1} b_{N1} \mathcal{A} & \dots & c_{N1} b_{NN} \mathcal{A} & \dots & \dots & c_{NN} b_{N1} \mathcal{A} & \dots & c_{NN} b_{NN} \mathcal{A}
\end{pmatrix}
$$

for k=1 < N **do**
  for i= 1 < N **do**
    for j= 1 < N **do**
      $w_k(i,j) = \mathcal{V}(i,j,1)a_{k1} + \cdots + \mathcal{V}(i,j,N)a_{kN}$
    end for
  end for
end for
for k=1 < N **do**
  for i= 1 < N **do**
    for j= 1 < N **do**
      $\mathcal{W}_{i,j}(k) := w_j(k,1)b_{i1} + \cdots + w_j(k,N)b_{iN}$
    end for
  end for
end for
for k=1 < N **do**
  for i= 1 < N **do**
    for j= 1 < N **do**
      $\mathcal{Z}(i,j,k) = \mathcal{W}_{j,k}(1)c_{i1} + \cdots + \mathcal{W}_{j,k}(N)c_{iN}$
    end for
  end for
end for

Kerntensor invertieren

$$\boldsymbol{\mathcal{X}}^{+} = \boldsymbol{\mathcal{G}}^{+} \times_{n=1}^{4} U^{(n)^{T}}.$$

Wie schwer ist es die Pseudoinversen des Kerntensor $\mathcal{G}$ zu finden?
Kerntensor ist in der Regel vollbesetzt.
Aber der Kerntensor enthält zwei Arten von Zahlen. Sehr kleine Zahlen kleiner als 10^{-10} und Zahlen größer 1.
Idee
Setzte alle kleinen Zahlen auf 0 $\rightarrow$ Der Kerntensor ist nun diagonal und sehr einfach zu invertieren.

Tensor-Vektor Produkt

Definition Tensor-Vektor Produkt
Es sei $\mathcal{A} \in \mathbb{R}^{N\times N\times N\times N}$ und $U \in \mathbb{R}^{N\times N}$. Dann ist $Y = tvp(\mathcal{A}, U)$ gegeben durch

$$Y_{i_1\, i_2} = \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} \mathcal{A}_{i_1\, i_2 j_1 j_2} U_{j_1 j_2}$$

$\rightarrow$ **Komplexität** $O(N^4)$.

Pseudoinversen-Tensor Vektor Produkt

1.Möglichkeit

$$tvp(\mathcal{X}^{+}, U) = \underbrace{tvp(\mathcal{G}^{+} \times_{n=1}^{4} U^{(n)^{T}}, U)}_{O(2N^4+8N^5+N^6)}$$

2.Möglichkeit

$$\mathcal{X}^{+}_{(n)} u = \underbrace{U^{(n)} \mathcal{G}^{+}_{(n)} (U^{(N_1)} \otimes U^{(N_2)} \otimes U^{(N_3)}) u}_{O(8N^4+8N^5+N^6+N^{10})} .$$

Massenmatrix

$$M^+ u = (\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T)^+ \otimes (\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T)^+ u$$

1. Berechnung des Matrix-Matrix Produkts $O(N^3)$
2. Berechnung der Pseudoinversen/Inversen mittels SVD oder Gauß Algorithmus von $\mathcal{W}_N^{1D}(\mathcal{N}^{1D})^T \in \mathbb{R}^{N \times N}$ ist von Komplexität $O(N^3)$
3. Matrix-Vektor Produkt dank unseres Algorithmus Komplexität von $O(4N^3)$

Insgesamt $O(6N^3)$.

Laplace Bilinearform

Lyapunow Gleichung

$$Y = AU + UA^T.$$

Lösung ist gegeben durch

$$U = \int_0^\infty e^{A\tau}(-V)e^{A^T\tau}\,d\tau.$$

Lyapunow Gleichung

- Naive Berechnung hat Komplexität von $O(N^6)$
- Bartels-Stewart Algorithmus [uYZ03] liefert uns eine Komplexität von $O(N^3)$.

1 Einleitung

2 Theorie

3 Pseudoinverse

4 Effiziente Berechnung

5 Resultate

Resultate

Wir haben zwei Möglichkeiten betrachtet die Pseudoinverse der zu untersuchunden Bilinearformen herzuleiten

1. Ausnutzung der Tensorstruktur und Herleitung einer einfachen Form
2. Universellen Ansatz über die HOSVD

Die beiden Methoden haben ihre Vor und Nachteile. Die erste Methode:

- nicht flexibel
- individueller Lösungsansatz für jede Bilinearform
- Komplexitätsklasse $O(N^3)$

Die zweite Methode über die HOSVD:

- flexibilität
- Komplexitätsklasse vom $O(N^6)$

Zukunft

In Zukunft kann man

- Mehr Bilinearformen auf Tensorstrukturen untersuchen
- Das Kommunikationsverhalten des $n - mode$ Produkt mit anderen Operatoren näher betrachten
- Effizientere Matrix-Matrix Multiplikation
- Unsere definierte Entfaltung bezüglich der HOSVD untersuchen

$$\mathcal{X}^{(p)}v = [(U^{(1)} \otimes U^{(2)})\mathcal{G}^{(p)}(U^{(3)} \otimes U^{(4)})]v$$

Framework für Testzwecke und Inspiration

https://github.com/Oldbridgegeek/pcvt

Katharina Kormann Martin Kronbichler.
A generic interface for parallel cell-based finite element operator application.
Elsevier, 2012.

Efficient evaluation of weak forms in discontinuous Galerkin methods.

Brett Bader Tamara Kolda.
Tensor Decompositions and Applications.
SIAM, 2009.

Danny Sorensen und Yunkai Zhou.
Direct Methods for Matrix Sylvester and Lyapunov Gleichungen.
2003.