Tensor-Train Decomposition for High-Dimensional Economic Models

Johannes Brumm Jakob Hußmann

EEA-ESEM 2024

Karlsruhe Institute of Technology

Table of contents

1. Introduction
2. Method
3. Economic Application
4. Conclusion

Introduction

Motivation

Quantitative Macro Models

Quantitative Macro Models

1. **Heterogeneous Agents**:
   overlapping generations, continuum of agents, discrete types.

Quantitative Macro Models

1. **Heterogeneous Agents**:
   overlapping generations, continuum of agents, discrete types.
2. **Multiple Assets**:
   liquid-/illiquid assets, housing.

Motivation

Quantitative Macro Models

1. **Heterogeneous Agents**:
   overlapping generations, continuum of agents, discrete types.
2. **Multiple Assets**:
   liquid-/illiquid assets, housing.
3. **Aggregate Risk**:
   realistically large shocks.

Quantitative Macro Models

1. **Heterogeneous Agents**:
   overlapping generations, continuum of agents, discrete types.
2. **Multiple Assets**:
   liquid-/illiquid assets, housing.
3. **Aggregate Risk**:
   realistically large shocks.

High-Accuracy Solutions

Motivation

Quantitative Macro Models

1. **Heterogeneous Agents**:
   overlapping generations, continuum of agents, discrete types.
2. **Multiple Assets**:
   liquid-/illiquid assets, housing.
3. **Aggregate Risk**:
   realistically large shocks.

High-Accuracy Solutions

- **Global Solution Techniques**: when shocks are large.

Motivation

Quantitative Macro Models

1. **Heterogeneous Agents**:
   overlapping generations, continuum of agents, discrete types.
2. **Multiple Assets**:
   liquid-/illiquid assets, housing.
3. **Aggregate Risk**:
   realistically large shocks.

High-Accuracy Solutions

- **Global Solution Techniques**: when shocks are large.
- **Curse of Dimensionality**: infeasible in high dimensions!

Related Literature

Standard Methods

grid size grows exponentially in dimensions. Infeasible for $d > 3$.

Related Literature

Standard Methods

grid size grows exponentially in dimensions. Infeasible for $d > 3$.

Advanced Methods

Related Literature

Standard Methods

grid size grows exponentially in dimensions. Infeasible for $d > 3$.

Advanced Methods

1. Sparse Grids: Brumm and Scheidegger (2017), Krueger and Kubler (2006), ...
    - reduce grid sizes, focus on important points
    - good in medium dimensions
    - grids still grow to an overwhelming size for $d > 20$

Related Literature

Standard Methods

grid size grows exponentially in dimensions. Infeasible for $d > 3$.

Advanced Methods

1. Sparse Grids: Brumm and Scheidegger (2017), Krueger and Kubler (2006), ...
   - reduce grid sizes, focus on important points
   - good in medium dimensions
   - grids still grow to an overwhelming size for $d > 20$
2. Neural Nets: Azinovic et al. (2022), Fernandez-Villaverde et al. (2020), ...
   - can handle large problems $d > 20$
   - black-box optimization
   - convergence depends on hyper-parameters

Related Literature

Standard Methods

grid size grows exponentially in dimensions. Infeasible for $d > 3$.

Advanced Methods

1. **Sparse Grids**: Brumm and Scheidegger (2017), Krueger and Kubler (2006), ...
    - reduce grid sizes, focus on important points
    - good in medium dimensions
    - grids still grow to an overwhelming size for $d > 20$
2. **Neural Nets**: Azinovic et al. (2022), Fernandez-Villaverde et al. (2020), ...
    - can handle large problems $d > 20$
    - black-box optimization
    - convergence depends on hyper-parameters

$\rightarrow$ Still room for improvement!

This Paper

Novel **global solution technique** based on the **Tensor-Train Decomposition**.

Method

The Solution Technique in a Nutshell

The Solution Technique in a Nutshell

1. **Time Iteration**:
   fixed-point solution approach using non-linear solvers.

The Solution Technique in a Nutshell

1. **Time Iteration**:
   fixed-point solution approach using non-linear solvers.
2. **Selection of Points**:
   Sample from the ergodic set (no curse of dimensionality!)

The Solution Technique in a Nutshell

1. **Time Iteration**:
   fixed-point solution approach using non-linear solvers.
2. **Selection of Points**:
   Sample from the ergodic set (no curse of dimensionality!)
3. **Policy Approximations**:
   High-dimensional off-grid interpolation ($\rightarrow$ high complexity!)

The Solution Technique in a Nutshell

1. **Time Iteration**:
   fixed-point solution approach using non-linear solvers.
2. **Selection of Points**:
   Sample from the ergodic set (no curse of dimensionality!)
3. **Policy Approximations**:
   High-dimensional off-grid interpolation ($\rightarrow$ high complexity!)

   $\Rightarrow$ Tensor-Train Decomposition

Approximation

$$\hat{f}(\vec{x}) = \sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m} A[i_1, ..., i_d] \cdot \left(\phi_{i_1}(x_1) \cdot \ldots \cdot \phi_{i_d}(x_d) \right)$$

Approximation

- Products of one-dimensional basis-functions ϕ

$$\hat{f}(\vec{x}) = \sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m} A[i_1, ..., i_d] \cdot \left(\phi_{i_1}(x_1) \cdot \ldots \cdot \phi_{i_d}(x_d)\right)$$

Approximation

- Products of one-dimensional basis-functions ϕ
- m basis-functions in each dimension $1, \ldots, d$

$$\hat{f}(\vec{x}) = \sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m} A[i_1, ..., i_d] \cdot \Big(\underbrace{\phi_{i_1}(x_1)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } 1}} \cdot \ldots \cdot \underbrace{\phi_{i_d}(x_d)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } d}} \Big)$$

Approximation

- Products of one-dimensional basis-functions ϕ
- m basis-functions in each dimension $1, \ldots, d$
- Combination $i_1, \ldots, i_d$ weighted with $A[i_1, \ldots, i_d]$

$$\hat{f}(\vec{x}) = \sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m} \underbrace{A[i_1, ..., i_d]}_{\substack{\text{tensor of} \\ \text{weights}}} \cdot \left(\underbrace{\phi_{i_1}(x_1)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } 1}} \cdot \ldots \cdot \underbrace{\phi_{i_d}(x_d)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } d}} \right)$$

Approximation

- Products of one-dimensional basis-functions ϕ
- m basis-functions in each dimension $1, \ldots, d$
- Combination $i_1, \ldots, i_d$ weighted with $A[i_1, \ldots, i_d]$
- Sum up over all permutations, Richter et al. (2023)

$$\hat{f}(\vec{x}) = \underbrace{\sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m}}_{\substack{\text{sum over all} \\ \text{dimensions } 1, \ldots, d \\ \text{and bases } 1, \ldots, m}} \underbrace{A[i_1, ..., i_d]}_{\substack{\text{tensor of} \\ \text{weights}}} \cdot \Big(\underbrace{\phi_{i_1}(x_1)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } 1}} \cdot \ldots \cdot \underbrace{\phi_{i_d}(x_d)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } d}} \Big)$$

Approximation

- Products of one-dimensional basis-functions ϕ
- m basis-functions in each dimension $1, \ldots, d$
- Combination $i_1, \ldots, i_d$ weighted with $A[i_1, \ldots, i_d]$
- Sum up over all permutations, Richter et al. (2023)

$$\hat{f}(\vec{x}) = \underbrace{\sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m}}_{\substack{\text{sum over all} \\ \text{dimensions } 1, \ldots, d \\ \text{and bases } 1, \ldots, m}} \underbrace{A[i_1, ..., i_d]}_{\substack{\text{tensor of} \\ \text{weights}}} \cdot \Big(\underbrace{\phi_{i_1}(x_1)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } 1}} \cdot \ldots \cdot \underbrace{\phi_{i_d}(x_d)}_{\substack{\text{basis function } i_1 \\ \text{in dimension } d}} \Big)$$

$\rightarrow$ A is an order-d tensor with m^d entries!

Handling the exponential size of A

Handling the exponential size of A

1. **Exploit Structure**:
   Symmetry, smoothness, linearity ($\rightarrow$ latent low rank!)

Handling the exponential size of A

1. **Exploit Structure**:
   Symmetry, smoothness, linearity ($\rightarrow$ latent low rank!)
2. **Decomposition**:
   Decompose A into a sequence of smaller objects with lower rank.

Handling the exponential size of A

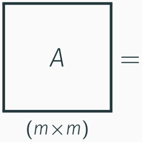
1. **Exploit Structure**:
   Symmetry, smoothness, linearity ($\rightarrow$ latent low rank!)
2. **Decomposition**:
   Decompose A into a sequence of smaller objects with lower rank.
3. **Oseledets (2011)**:
   Decompose order-d tensor of size m^d into a sequence of d order-3 tensors with size mr^2.

   $\rightarrow$ **Tensor-Train Decomposition**

Tensor-Train Decomposition – 2D Special Case

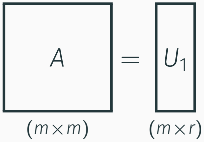
Tensor-Train Decomposition – 2D Special Case

- Approximate $(m \times m)$-matrix A



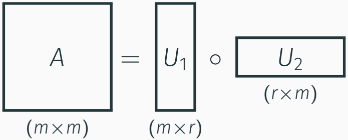
Tensor-Train Decomposition – 2D Special Case

- Approximate $(m \times m)$-matrix A
- outer-product of $(m \times r)$-matrices U_1, U_2



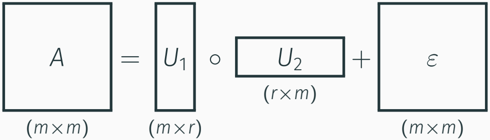
Tensor-Train Decomposition – 2D Special Case

- Approximate $(m \times m)$-matrix A
- outer-product of $(m \times r)$-matrices U_1, U_2



Tensor-Train Decomposition – 2D Special Case

- Approximate $(m \times m)$-matrix A
- outer-product of $(m \times r)$-matrices U_1, U_2
- error ε low if A has low rank



Matrix Completion

Matrix Completion

$$(m \times r) \circ (r \times m)$$

Matrix Completion

$$\overbrace{(m \times r)}^{U_1} \circ \overbrace{(r \times m)}^{U_2}$$

Generalization

Matrix Completion

$$\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Generalization

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$(m \times r) \qquad\qquad (r \times m)$$

Matrix Completion

$$\underbrace{\overbrace{(m \times r)}^{U_1} \circ \overbrace{(r \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\overbrace{(m \times r)}^{U_1} \qquad \overbrace{(r \times m)}^{U_3}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\overbrace{(m \times r)}^{U_1} \circ (r \times m \times r) \circ \overbrace{(r \times m)}^{U_3}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\overbrace{(m \times r)}^{U_1} \circ \overbrace{(r \times m \times r)}^{U_2} \circ \overbrace{(r \times m)}^{U_3}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\overbrace{(m \times r)}^{U_1} \circ \overbrace{(r \times m \times \cancel{r})}^{U_2} \circ \overbrace{(\cancel{r} \times m)}^{U_3}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\overbrace{(m \times r)}^{U_1} \circ \underbrace{\overbrace{(r \times m \times \cancel{r})}^{U_2} \circ \overbrace{(\cancel{r} \times m)}^{U_3}}_{(r \times m \times m)}$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\overbrace{(m \times \cancel{r})}^{U_1} \circ \underbrace{\overbrace{(r \times m \times \cancel{r})}^{U_2} \circ \overbrace{(\cancel{r} \times m)}^{U_3}}_{(\cancel{r} \times m \times m)}$$

Generalization

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \underbrace{\overbrace{(r \times m \times \cancel{r})}^{U_2} \circ \overbrace{(\cancel{r} \times m)}^{U_3}}_{(\cancel{r} \times m \times m)}}_{(m \times m \times m)}$$

Generalization

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \underbrace{\overbrace{(r \times m \times \cancel{r})}^{U_2} \circ \overbrace{(\cancel{r} \times m)}^{U_3}}_{(\cancel{r} \times m \times m)}}_{(m \times m \times m)} \approx A$$

Matrix Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \overbrace{(\cancel{r} \times m)}^{U_2}}_{(m \times m)} \approx A$$

Tensor Completion

$$\underbrace{\overbrace{(m \times \cancel{r})}^{U_1} \circ \underbrace{\overbrace{(r \times m \times \cancel{r})}^{U_2} \circ \overbrace{(\cancel{r} \times m)}^{U_3}}_{(\cancel{r} \times m \times m)}}_{(m \times m \times m)} \approx A$$

same principle for $d > 3$!

General Case

Tensor-Train Decomposition

General Case

$$A \approx U_1 \circ U_2 \circ \ldots \circ U_{d-1} \circ U_d$$

Tensor-Train Decomposition

General Case

$$\overbrace{A}^{m\times\ldots\times m} \approx \overbrace{U_1}^{m\times r} \circ \overbrace{U_2}^{r\times m\times r} \circ \ldots \circ \overbrace{U_{d-1}}^{r\times m\times r} \circ \overbrace{U_d}^{r\times m}$$

Tensor-Train Decomposition

General Case

$$\underbrace{\overbrace{A}^{m\times\ldots\times m}}_{\mathcal{O}(m^d)} \approx \underbrace{\overbrace{U_1}^{m\times r} \circ \overbrace{U_2}^{r\times m\times r} \circ \ldots \circ \overbrace{U_{d-1}}^{r\times m\times r} \circ \overbrace{U_d}^{r\times m}}_{\mathcal{O}(mdr^2)}$$

General Case

$$\underbrace{\overbrace{A}^{m\times\ldots\times m}}_{\mathcal{O}(m^d)} \approx \underbrace{\overbrace{U_1}^{m\times r} \circ \overbrace{U_2}^{r\times m\times r} \circ \ldots \circ \overbrace{U_{d-1}}^{r\times m\times r} \circ \overbrace{U_d}^{r\times m}}_{\mathcal{O}(mdr^2)}$$

Scaling Example

General Case

$$\underbrace{\overbrace{A}^{m\times\ldots\times m}}_{\mathcal{O}(m^d)} \approx \underbrace{\overbrace{U_1}^{m\times r} \circ \overbrace{U_2}^{r\times m\times r} \circ \ldots \circ \overbrace{U_{d-1}}^{r\times m\times r} \circ \overbrace{U_d}^{r\times m}}_{\mathcal{O}(mdr^2)}$$

Scaling Example

d	#A	$\sum$ #U
2	100	100
5	10^5	850
10	10^{10}	2100
100	10^{100}	24600

Table 1: Object size for different d with $m = 10$ and $r = 5$.

Back to the Approximation

Back to the Approximation

Approximation

Approximation

- Replace A by its decomposition $U_1, \ldots, U_d$

Approximation

- Replace A by its decomposition $U_1, \ldots, U_d$
- Sequential contractions from end to end

Approximation

- Replace A by its decomposition $U_1, \ldots, U_d$
- Sequential contractions from end to end

$$\hat{f}(x) = U_1 \circ \ldots \circ \left(\left(U_{d-1} \circ \left(U_d \circ \phi(x_d) \right) \circ \phi(x_{d-1}) \right) \circ \ldots \circ \phi(x_1) \right)$$

Approximation

- Replace A by its decomposition $U_1, \ldots, U_d$
- Sequential contractions from end to end

$$\hat{f}(x) = U_1 \circ \ldots \circ \left(\left(U_{d-1} \circ (U_d \circ \phi(x_d)) \circ \phi(x_{d-1}) \right) \circ \ldots \circ \phi(x_1) \right)$$

Computing the Decomposition

Back to the Approximation

Approximation

- Replace A by its decomposition $U_1, \ldots, U_d$
- Sequential contractions from end to end

$$\hat{f}(x) = U_1 \circ \ldots \circ \left(\Big(U_{d-1} \circ \big(U_d \circ \phi(x_d) \big) \circ \phi(x_{d-1}) \Big) \circ \ldots \circ \phi(x_1) \right)$$

Computing the Decomposition

- Non-linear approximation problem

Approximation

- Replace A by its decomposition $U_1, \ldots, U_d$
- Sequential contractions from end to end

$$\hat{f}(x) = U_1 \circ \ldots \circ \left(\left(U_{d-1} \circ \left(U_d \circ \phi(x_d) \right) \circ \phi(x_{d-1}) \right) \circ \ldots \circ \phi(x_1) \right)$$

Computing the Decomposition

- Non-linear approximation problem
- Decomposed in a sequence of linear problems, Holtz et al. (2012)

Economic Application

Model Setup

Model Setup

- **Large Scale OLG**: with yearly calibration, portfolio choice, endogenous labor.

Model Setup

- Large Scale OLG: with yearly calibration, portfolio choice, endogenous labor.
- NK-Production: with Rotemberg price adjustment costs and union bargaining.

Model Setup

- Large Scale OLG: with yearly calibration, portfolio choice, endogenous labor.
- NK-Production: with Rotemberg price adjustment costs and union bargaining.
- Government: faces large spending shocks, 20% of GPD every decade.

Model Setup

- **Large Scale OLG**: with yearly calibration, portfolio choice, endogenous labor.
- **NK-Production**: with Rotemberg price adjustment costs and union bargaining.
- **Government**: faces large spending shocks, 20% of GPD every decade.
- **Aggregate Risk**: persistent large shocks.

Model Setup

- **Large Scale OLG**: with yearly calibration, portfolio choice, endogenous labor.
- **NK-Production**: with Rotemberg price adjustment costs and union bargaining.
- **Government**: faces large spending shocks, 20% of GPD every decade.
- **Aggregate Risk**: persistent large shocks.

Research Questions

Economic Application

Model Setup

- **Large Scale OLG**: with yearly calibration, portfolio choice, endogenous labor.
- **NK-Production**: with Rotemberg price adjustment costs and union bargaining.
- **Government**: faces large spending shocks, 20% of GPD every decade.
- **Aggregate Risk**: persistent large shocks.

Research Questions

1. Impact of large shocks on debt, inflation and taxation?

Economic Application

Model Setup

- **Large Scale OLG**: with yearly calibration, portfolio choice, endogenous labor.
- **NK-Production**: with Rotemberg price adjustment costs and union bargaining.
- **Government**: faces large spending shocks, 20% of GPD every decade.
- **Aggregate Risk**: persistent large shocks.

Research Questions

1. Impact of large shocks on debt, inflation and taxation?
2. How are expenses financed depending on the policy?

Model Setup

- **Large Scale OLG**: with yearly calibration, portfolio choice, endogenous labor.
- **NK-Production**: with Rotemberg price adjustment costs and union bargaining.
- **Government**: faces large spending shocks, 20% of GPD every decade.
- **Aggregate Risk**: persistent large shocks.

Research Questions

1. Impact of large shocks on debt, inflation and taxation?
2. How are expenses financed depending on the policy?
3. Intergenerational impact of different policy?

Computational Problem

Computational Problem

State Space

- 60 Generations
- two types of assets
- three aggregate shocks
- five policy parameters

Computational Problem

State Space

- 60 Generations
- two types of assets
- three aggregate shocks
- five policy parameters

Policy Functions

- 120 individual policies
- 4 aggregate policies

Computational Problem

State Space

- 60 Generations
- two types of assets
- three aggregate shocks
- five policy parameters

Policy Functions

- 120 individual policies
- 4 aggregate policies

→ Computationally very challenging!

Computational Problem

State Space

- 60 Generations
- two types of assets
- three aggregate shocks
- five policy parameters

Policy Functions

- 120 individual policies
- 4 aggregate policies

$\rightarrow$ Computationally very challenging!

$\rightarrow$ The proposed method solves the problem in 2h on a PC!

Results

- Average relative error in consumption 0.014%
- Average market clearing error 0.036%
- Q99 consumption error $< 0.08\%$

	Q99	AVG	Q50
Euler-Equation 1	7.40E-04	1.31E-04	8.03E-05
Euler-Equation 2	7.97E-04	1.40E-04	8.40E-05
Labor Supply	1.13E-03	3.43E-04	2.57E-04
Phillips-Curve	1.03E-03	2.08E-04	1.63E-04
Market Clearing	1.29E-03	3.61E-04	2.35E-04

Conclusion

Conclusion

Conclusion

1. We present a **new global solution technique** for high-dimensional economic problems.

Conclusion

1. We present a **new global solution technique** for high-dimensional economic problems.
2. We exploit the **Tensor-Train Decomposition** for efficient approximations.

Conclusion

1. We present a **new global solution technique** for high-dimensional economic problems.
2. We exploit the **Tensor-Train Decomposition** for efficient approximations.
3. We solve a high-dimensional **economic application** with **high accuracy**, at low **computational cost**.

References

Azinovic, M., Gaegauf, L., & Scheidegger, S. (2022).**Deep equilibrium nets.** *International Economic Review*, *63*(4), 1471–1525.

Brumm, J., & Scheidegger, S. (2017).**Using adaptive sparse grids to solve high-dimensional dynamic models.** *Econometrica*, *85*(5), 1575–1612.

Fernandez-Villaverde, J., Nuno, G., Sorg-Langhans, G., & Vogler, M. (2020).**Solving high-dimensional dynamic programming problems using deep learning.** *Working Paper*.

Holtz, S., Rohwedder, T., & Schneider, R. (2012).**The alternating linear scheme for tensor optimization in the tensor train format.** *SIAM Journal on Scientific Computing*, *34*(2), A683–A713.

References ii

Krueger, D., & Kubler, F. (2006).**Pareto-improving social security reform when financial markets are incomplete!?** *American Economic Review*, *96*(3), 737–755.

Oseledets, I. V. (2011).**Tensor-train decomposition.** *SIAM Journal on Scientific Computing*, *33*(5), 2295–2317.

Richter, L., Sallandt, L., & Nüsken, N. (2023). **From continuous-time formulations to discretization schemes: Tensor trains and robust regression for bsdes and parabolic pdes.**