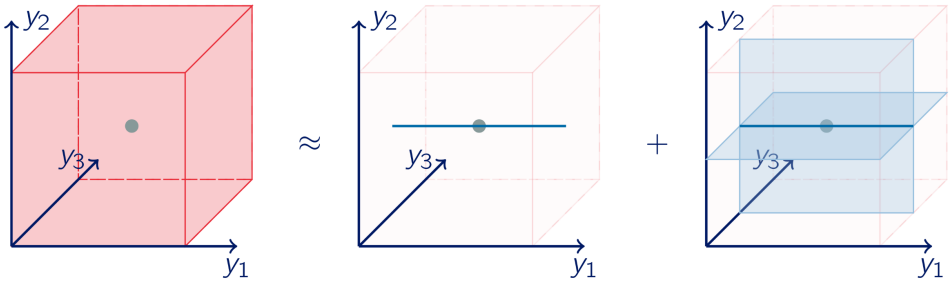
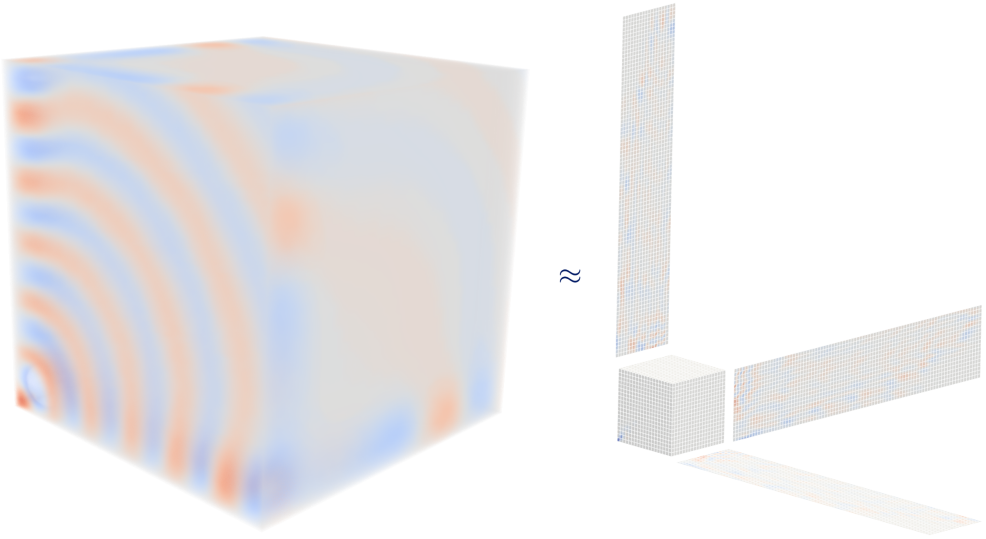
Efficient numerical approximation of solutions to high-dimensional partial differential equations

with applications in option pricing and scattering problems

Jacob Snoeijer





Promotoren **Prof. dr. Karel in 't Hout — Prof. dr. Wim Vanroose**

Proefschrift voorgelegd tot het behalen van de graad van doctor in de wetenschappen – wiskunde
Faculteit Wetenschappen — Antwerpen, 2022

Universiteit Antwerpen

Universiteit Antwerpen

Faculteit Wetenschappen
doctor in de wetenschappen – wiskunde

Efficient numerical approximation of solutions to high-dimensional partial differential equations

with applications in option pricing and scattering problems

Proefschrift voorgelegd tot het behalen van de graad van
doctor in de wetenschappen – wiskunde
aan de Universiteit Antwerpen te verdedigen door

Jacob Snoeijer

Promotoren
Prof. dr. Karel in 't Hout
Prof. dr. Wim Vanroose

Antwerpen, 30 augustus 2022

Jury

Voorzitter
Prof. dr. Benny Van Houdt, University of Antwerp, Belgium

Promotoren
Prof. dr. Karel in 't Hout, University of Antwerp, Belgium
Prof. dr. Wim Vanroose, University of Antwerp, Belgium

Leden
Prof. dr. Hans Vande Sande, University of Antwerp, Belgium
Prof. dr. ir. Kees Oosterlee, Utrecht University, The Netherlands
Prof. dr. Michèle Vanmaele, Ghent University, Belgium

Contact
Jacob Snoeijer

Universiteit Antwerpen
Faculteit Wetenschappen
Departement Wiskunde
Middelheimlaan 1, 2020 Antwerpen, België
M: jacob.snoeijer@uantwerpen.be

Nederlandstalige samenvatting

Veel geobserveerde fenomenen om ons heen kunnen wiskundig beschreven worden middels (partiële) differentiaalvergelijkingen (PDVen). In veel gevallen is een analytische oplossing van een dergelijke PDV niet bekend en daarom zal een numerieke benadering van de oplossing voor die PDV gezocht moeten worden.

In dit proefschrift beschouwen we efficiënte numerieke benaderingen voor oplossingen van hoog-dimensionale problemen. In veel toepassingen zijn drie (plaats)dimensies voor fysische problemen voldoende, maar er bestaan ook toepassingsgebieden waar hoog-dimensionale problemen, met een dimensie veel groter dan drie, van nature voorkomen. Die problemen komen bijvoorbeeld voor in de financiële wiskunde waar opties geprijsd en grieken geschat moeten worden. Voor het prijzen van een basketoptie wordt een Black–Scholes vergelijking opgelost met een (plaats)dimensie gelijk aan het aantal onderliggende goederen in een mand met aandelen, de basket. Een tweede toepassingsgebied waar eenvoudig hoog-dimensionale problemen kunnen voorkomen zijn zogenaamde verstrooiingsproblemen in bijvoorbeeld foton-ionisatie. Bij foton-ionisatie is er een systeem van atomen (of moleculen) dat onder invloed van een lichtstraal uit elkaar valt en waarbij enkelvoudige- of meervoudige-ionisatie kan ontstaan. De ontsnappingshoek van die elektronen kan gemeten worden en komt overeen met een kansverdeling, de 'far field map'. Dit wordt beschreven met de amplitude van een golf in de ontsnappingsrichting. Deze golffunctie kan beschreven worden als de oplossing van een Helmholtz vergelijking met een plaatsafhankelijk golfgetal. De dimensie van dit probleem groeit met het aantal beschouwde elektronen in het systeem.

Er zijn diverse standaardtechnieken voor het numeriek oplossen van dergelijke PDVen, maar die zijn praktisch alleen mogelijk voor laag-dimensionale problemen. Als de dimensie van de problemen groter wordt, dan wordt het bepalen van numerieke benaderingen voor de oplossingen van de PDVen te rekenintensief. Daarom beschouwen we in dit proefschrift efficiënte manieren om de numerieke oplossing van de voorkomende PDVen te benaderen.

De eerste benaderingstechniek die we beschouwen is voorgesteld door Reisinger en Wittum [71] en gebaseerd op de hoofdcomponentenanalyse, ofwel 'principal component analysis' (PCA), van de covariantiematrix. In veel financiële toepassingen blijkt dat de eigenwaarde behorende bij de eerste hoofdcomponent veel groter is dan alle overige eigenwaarden. Voor een goede benadering van de oplossing kunnen alle overige hoofdcomponenten niet volledig genegeerd worden, maar de eerste-orde correctietermen zijn wel voldoende om een goede analytische benadering te verkrijgen voor de exacte oplossing van een hoog-dimensionale Black-Scholes PDV. Voor deze analytische benadering moeten in totaal slechts een één-dimensionale PDV en $(d-1)$ twee-dimensionale PDVen opgelost worden.

In het eerste deel van dit proefschrift zullen we deze benaderingstechniek nader bestuderen en de discretisatiefouten analyseren voor het prijzen van Europese-stijl basketopties. Vervolgens

kan deze benaderingstechniek verder uitgebreid worden, waarmee eveneens Bermuda-stijl en Amerikaanse-stijl basketopties geprijsd kunnen worden. De eerlijke prijs voor een Bermuda-stijl optie wordt beschreven middels een Black-Scholes PDV zoals ook voor een Europese-stijl optie, maar in aanvulling daarop kan op een vast aantal afgesproken tijdspunten de optie eenmalig vroegtijdig uitgeoefend worden. Dit resulteert in een aanvullende optimale uitoefenconditie die opgenomen moet worden in de tijdsdiscretisatie. Deze optimale uitoefenconditie kan een inconsistentie opleveren voor gebruik in de PCA-gebaseerde benadering. Ook de discretisatiefouten voor het prijzen van Bermuda-stijl basketopties worden in dit proefschrift nader geanalyseerd.

In plaats van op een eindig aantal uitoefenmomenten kunnen Amerikaanse-stijl opties op elk gewenst moment eenmalig uitgeoefend worden. Dit kan geformuleerd worden met een partieel differentiaal complementariteitsprobleem (PDCP). De PCA-gebaseerde aanpak kan uitgebreid worden om ook de oplossing van dergelijke PDCPen numeriek te benaderen. Verschillende tijdsdiscretisatiemethoden worden beschouwd. Tevens wordt een vergelijking gemaakt met een alternatieve benaderingstechniek, de comonotone aanpak. Deze comonotone aanpak blijkt een lineaire combinatie van twee speciale gevallen van de PCA-gebaseerde aanpak te zijn.

Ten slotte wordt besproken dat de PCA-gebaseerde aanpak ook gebruikt kan worden voor het bepalen van de grieken, ofwel de partiële afgeleide van de optieprijs naar een zekere variabele. Zo kunnen zowel de Deltas als de Gammas benaderd worden. Dit wordt geïllustreerd met enkele numerieke voorbeelden voor Europese-, Bermuda- en Amerikaanse-stijl basketopties.

Het tweede deel van dit proefschrift gaat over een andere aanpak om de oplossing van een hoog-dimensionale differentiaalvergelijking te beschrijven. In deze aanpak wordt de rang voor de numerieke oplossing van die vergelijking beperkt. Van bijvoorbeeld een twee-dimensionaal probleem kan de oplossing op het beschouwde plaatsrooster voorgesteld worden middels een matrix. Van deze matrix kunnen de singuliere waarden bepaald worden door een singuliere waarden decompositie (SVD). Er wordt opgemerkt dat de oplossing van de Helmholtz vergelijking die we beschouwen voor de verstrooiingsproblemen een lage rang heeft. Er blijken dus slechts een beperkt aantal singuliere waarden relevant te zijn. Dus in plaats van het oplossen van een differentiaalvergelijking op het volledige rooster kunnen we deze differentiaalvergelijking projecteren op een ruimte opgespannen door factormatrices verkregen vanuit de singuliere waarden decompositie van de oplossing. Hiermee verkrijgen we een nieuwe differentiaalvergelijking voor een lage-rang factormatrix en is dus effectief het aantal onbekenden gereduceerd. De vergelijking voor deze factormatrix kan gerelateerd worden aan vergelijkingen die eveneens opgelost worden in de coupled channel techniek. Deze lage-rang benadering in twee dimensies kan uitgebreid worden naar benaderingen voor oplossingen van hoog-dimensionale problemen.

Er wordt dan ook een korte introductie over de representatie van hoog-dimensionale data door middel van tensoren gegeven. Er bestaan verschillende tensordecomposities die gebruikt kunnen worden om deze lage-rank tensoren te beschrijven, zoals de Canonical Polyadic (CP)-tensordecompositie en de Tucker-tensordecompositie. In dit proefschrift presenteren we een alternerende projectiemethode om direct numeriek de lage-rang factoren te bepalen voor een oplossing van een hoog-dimensionaal Helmholtz probleem. Numerieke experimenten tonen inderdaad goede resultaten die enkelvoudige, dubbele en drievoudige ionisatie kunnen beschrijven met slechts een lage-rang benadering voor het hoog-dimensionaal Helmholtz

probleem.

Ten slotte verkennen we of deze alternerende projectiemethode ook gebruikt kan worden voor het oplossen van tijdsafhankelijke differentiaalvergelijkingen. Als alternatieve methode bestaat in elk geval de dynamische lage-rang integrator van Lubich en anderen [47]. Deze methode kan worden geïnterpreteerd als het oplossen van een optimalisatieprobleem. Op basis daarvan worden in dit proefschrift nog enkele alternatieve methoden geformuleerd. Een numerieke vergelijking van alle besproken methoden laat een zekere potentie zien voor methoden als de dynamische lage-rang integrator en mogelijk ook voor de alternerende projectiemethode. Echter, aanvullend onderzoek zal nodig zijn om een efficiënte numerieke methode te formuleren voor het benaderen van lage-rang oplossingen voor stijve problemen.

Dankwoord

Na een goede periode van alweer bijna zes jaar ligt hier nu een schriftelijke weerslag van diverse elementen die de afgelopen jaren in mijn onderzoek voorbij zijn gekomen. Het was een tijd waarin veel dingen bekeken zijn, nader onderzocht werden en nieuwe resultaten gevonden zijn.

Inhoudelijk zal ik daar in dit dankwoord niet iets over zeggen; daarvoor zou de rest van dit proefschrift zeker nog eens bekeken of gelezen kunnen worden. Ik noem hier bewust ook kijken: wiskunde mag dan misschien bekend staan om haar vergelijkingen, maar veel kan toch ook visueel voorgesteld of ondersteund worden met diverse leuke of verhelderende figuren. Mogelijk leest dit proefschrift daarmee soms ook bijna net zo vlot als een prentenboek.

Mocht het lezen ervan toch niet altijd even vlot gaan, dan illustreert dat ook een andere parallel met het onderzoek in de afgelopen jaren. Ongetwijfeld zullen velen de ervaring delen dat onderzoek soms soepel verloopt en op andere tijden juist iets moeizamer gaat. Toch is deze periode in alle tijden leerzaam geweest en is dit een uitgelezen plek om sommige mensen nog eens met name te bedanken voor hun betrokkenheid, inbreng of welke rol ze de afgelopen jaren dan ook bij het onderzoek gehad hebben.

Als eerste betreft dat de beide promotoren voor het doctoraatsonderzoek, Karel en Wim. Allebei hebben jullie een heel eigen karakter en daarmee ook een eigen manier van werken. Toch maakt juist die eigenheid het ook een hele mooie combinatie om jullie beide als promotor gehad te mogen hebben. Ik herinner mij nog goed dat een notitie nog maar amper ter tafel lag (of later op een scherm verscheen) en Karel had de eerste typo in een of ander subscript uit een vergelijking alweer zien staan. Karel, ik heb je zorgvuldigheid en precisie in alles wat je deed altijd zeer kunnen waarderen. Verder lijkt het wel of Wim een grenzeloze belangstelling heeft voor allerlei verschillende onderwerpen en overal associaties of verbanden ziet. Dat is uiteindelijk ook weer een vruchtbare basis gebleken voor diverse nieuwe onderzoeksuggesties. Wim, ik heb je brede belangstelling en vermogen verbanden te zien zeer kunnen waarderen.

Als mandaatsassistent heb ik ook voor diverse cursussen in zowel de bachelor- als de masteropleiding de oefeningensessies mogen geven of programmeeropdrachten kunnen begeleiden. Het is mooi om studenten in hun studie te begeleiden en te zien dat ze onderwerpen eigen maken. Naast de cursussen van mijn beide promotoren was ik onder andere ook vaste assistent bij het Computerpracticum. Daarbij verdient Stijn ook zeker nog een bijzondere vermelding. Onder zijn leiding leerden we studenten tijdens die cursus de allereerste beginselen van het programmeren in diverse programmeeromgevingen. Ik heb het mooi gevonden om samen met je een training Mathematica te volgen en inzichten daaruit erbij op te nemen in die cursus.

Verder zijn er nog de collega's met wie ik in die tijd het bureau M.G.328 gedeeld heb; Nick, Michiel en Pieter. Dank voor het gezelschap en de gesprekken over alles wat ook maar ter sprake kwam. Ook de andere collega's van de toegepaste wiskunde met wie veel middagpauzes gedeeld zijn zullen niet ongenoemd blijven: Michiel, Siegfried, Maarten, Lynn, Jeffrey, Valérie en Lise.

Tot slot een woord van dank aan de overige leden van de individuele doctoraatscommissie en de doctoraatsjury. Allereerst aan Benny voor het voorzitten van de commissie en de opvolging gedurende deze tijd. De individuele gesprekken met Hans waren altijd nuttig en interessant om ook weer op een nieuwe manier naar het onderzoek te kijken. Michèle bleek tijdens een conferentie in A Coruña in hetzelfde hotel te verblijven, waardoor we toen al eens genoeglijk hebben kunnen doorpraten over diverse onderwerpen. Tot slot heb ik Kees de afgelopen jaren al meerdere keren gezien en gesproken; je hartelijke belangstelling tijdens diverse conferenties is altijd erg gewaardeerd.

Nogmaals, een hartelijk dank aan allen die op welke wijze dan ook betrokken geweest zijn bij de totstandkoming en afronding van dit doctoraat.

Jacob Snoeijer
Antwerpen, augustus 2022

Contents

List of Symbols and Notations

Abbreviations

FLOP	FLoating Point Operation
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PDCP	Partial Differential Complementarity Problem
SDE	Stochastic Differential Equation
MC	Monte Carlo (simulation)
LSMC	Least Squares Monte Carlo (by Longstaff and Schwartz [52])
CI	Confidence Interval
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
RK	Runge-Kutta (method)
CN	Crank–Nicolson (scheme)
ADI	Alternating Direction Implicit (scheme)
EP	Explicit Payoff method
IT	Ikonen–Toivanen (splitting technique)
CP	Canonical Polyadic (decomposition)
TT	Tensor Train (decomposition)
ECS	Exteriour Complex Scaling
KSL	Version of dynamical low-rank integrator, see Alg. 10
KKT	Karush–Kuhn–Tucker (conditions)

Discretization paramters

m	Number of (internal) spatial discretization points
N	Number of temporal discretization points
`Npaths`	Number of simulated paths in a Monte Carlo simulation
n_i	Total number of spatial discretization points in i-th direction ($i = 1, 2, \ldots, d$)
r_i	Rank (of a tensor) in i-th direction ($i = 1, 2, \ldots, d$)

General symbols

$\mathbb{N}$	Integers
$\mathbb{R}$	Real valued numbers
$\mathbb{C}$	Complex valued numbers
x, y, z	Scalars (lower case standard font)
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	Vectors (lower case bold font)
X, Y, Z	Random variables (upper case standard font)
$\boldsymbol{A}, \boldsymbol{D}_{xx}$	Matrices (upper case bold font)
$\mathcal{M}, \mathcal{X}$	Tensors (upper case calligraphic bold font)
$\boldsymbol{Y}_{(k)}$	k-th unfolding of tensor $\mathcal{Y}$
$X \sim \mathcal{N}(\mu,\ \sigma^2)$	Random variable X is a normally distributed with mean μ and variance σ^2
$\mathbb{E}[X]$	Expected value of random variable X
$N(\cdot)$	Cumulative normal density function
$\mathcal{L}\mathcal{M}$	Linear operator $\mathcal{L}$ applied on tensor $\mathcal{M}$
$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$	Lagrangian function, see Eqn. 9.31

Vector, matrix and tensor operations

$\mu[\boldsymbol{A}]$	Logarithmic matrix norm
$\overline{\boldsymbol{A}}$	Complex conjugate of matrix
$\boldsymbol{A}^{\mathsf{T}}$	Transpose of matrix
$\boldsymbol{A}^{\mathsf{H}}$	Transposed complex conjugate of matrix
$\boldsymbol{z} = \boldsymbol{x} * \boldsymbol{y}$	Elementwise product of vectors
$\boldsymbol{Z} = \boldsymbol{x} \circ \boldsymbol{y}$	Outer product of vectors
$\boldsymbol{Z} = \boldsymbol{X} \circ \boldsymbol{Y}$	Hadamard product of matrices, see Def. 4
$\boldsymbol{Z} = \boldsymbol{X} \otimes \boldsymbol{Y}$	Kronecker product of matrices, see Def. 5
$\boldsymbol{Z} = \boldsymbol{X} \odot \boldsymbol{Y}$	Khatri-Rao product of matrices, see Def. 6
$\mathcal{Y} = \mathcal{X} \times_k \boldsymbol{A}$	Tensor times matrix product, see Sec. 7.2.2

Helmholtz problem specific symbols

k_0^2	Constant wave number
χ	Space-dependent part of wave number
u_{sc}	Solution to Helmholtz problem
Δ	Laplace operator

Option pricing specific symbols

K	Strike price
T	Maturity time
E	(Finite) number of exercise times for Bermudan option
τ	Standard time (forward)
t	Time till maturity; $t = T - \tau$
d	Number of assets in a basket
$\phi(\boldsymbol{s})$	Payoff function
$\boldsymbol{S_0} \in \mathbb{R}^d$	Spot prices of assets in basket
$\boldsymbol{\sigma} \in \mathbb{R}^d$	Volatilities of assets in basket
$\boldsymbol{\omega} \in \mathbb{R}^d$	Weights of assets in basket
$\boldsymbol{\rho} \in \mathbb{R}^{d \times d}$	Correlation matrix for underlying assets
$\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$	Covariance matrix; $\Sigma_{ij} = \sigma_i \rho_{ij} \sigma_j$ for $i, j = 1, 2, \ldots, d$
$u(\boldsymbol{s}, t)$	Fair value of a basket option
$\widetilde{u}(\boldsymbol{s}, t)$	PCA-based approximation to fair value of a option
$u^{\mathrm{up}}(\boldsymbol{s}, t)$	Comonotonic upper bound for fair value of a option
$u^{\mathrm{low}}(\boldsymbol{s}, t)$	Comonotonic lower bound for fair value of a option
$u^{\mathrm{app}}(\boldsymbol{s}, t)$	Comonotonic approximation to fair value of a option
$\Delta_k(\boldsymbol{s}, t)$	k-th Delta of a basket option ($k = 1, 2, \ldots, d$)
$\widetilde{\delta}_k(\boldsymbol{s}, t)$	PCA-based approximation to k-th Delta of a basket option ($k = 1, 2, \ldots, d$)
$\Gamma_{kl}(\boldsymbol{s}, t)$	k, l-th Gamma of a basket option ($k, l = 1, 2, \ldots, d$)
$\widetilde{\gamma}_{kl}(\boldsymbol{s}, t)$	PCA-based approximation to k, l-th Gamma of a basket option ($k, l = 1, 2, \ldots, d$)

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

A lot of phenomena observed around us can be described in terms of mathematical problems or equations. Although the computational power to numerically solve these problems has extensively increased over the past decades, the mathematical equations to solve have become more and more complicated. This thesis is about efficient numerical approximation of such challenging problems and goes under the title *'efficient numerical approximation of solutions to high-dimensional partial differential equations – with applications in option pricing and scattering problems.'* In the following, we take a first, closer look at the keywords and provide an overview of the content of this thesis.

A thesis in mathematics often deals with *solutions* and *equations*. An unknown quantity of interest, let us call it u, is a solution to a mathematical problem that is formulated in terms of one or more equations. Mathematics plays a crucial role in modeling phenomena observed in eg. physics, chemistry or financial markets.

An important mathematical tool to model a wide range of phenomena is a *partial differential equation* (PDE). The unknown quantity u depends on an independent real variable, let us call it x. If the unknown quantity u can also change over time t, then we call it time-dependent. Partial differential equations are equations that relate the unknown dependent quantity $u(x,t)$, the independent variables (such as x and time t) and the dependence of u on the independent variables (such as the partial derivatives $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial t}$). Of course, next to x, one can introduce additional independent variables which will increase the dimension of the problem. For example, if x represents a one-dimensional space-coordinate then one can also introduce y and z to describe phenomena in three space dimensions.

This thesis is not about modeling by partial differential equations, but focuses on the effective numerical solution of PDEs that arise in different application areas. An (semi-)closed analytical solution to the PDEs under consideration is almost always unknown, so one is led to the *numerical approximation* of the solution to these PDEs. To get insight in the quality of this numerical approximation an analysis of its error (measured in a certain norm) is important.

For a lot of physical applications three-dimensional problems arise naturally. In the present thesis, however, we deal with *high-dimensional problems*. Such problems, with high dimension, appear in various application areas. In Part I of this thesis, we consider *applications in option*

pricing. Valuation of financial derivative products, like basket options, easily results in PDEs with an arbitrarily large dimension. For example, for a basket option the dimension of the relevant Black-Scholes PDE is equal to the number of assets in that basket. Thus a basket option on all assets weighted in an index such as the BEL-20 leads to a 20-dimensional problem.

A second application area where *high-dimensional problems* appear naturally is subject of Part II of this thesis. There we consider *applications in scattering* such as atomic and molecular breakup reactions. For example the helium atom, He, with two electrons is the simplest system on which double ionization might occur. But, as the number of electrons increases also the dimension of the Helmholtz problem to describe this multiple-ionization increases accordingly. Moreover, automatic selection of basis functions, instead of the use of spherical harmonics in (8.4), yields already a six-dimensional problem for the helium atom.

We finish with the first keyword: *efficient*. A standard numerical discretization of the high-dimensional partial differential equations considered in this thesis is infeasible. This is known in the literature as the curse of dimensionality. Let d denote the dimension of the PDE and assume one discretizes each direction with $n = 100$ unknowns. Then, for example, a $d = 5$ dimensional PDE is discretized with $n^5 = 10^{10}$ unknowns. Thus just to store only one representation in double-precision takes already more than 74.5GB of memory. Apart from infeasible memory consumption also the number of floating point operations (FLOPs) or computing time is enormous, even on state-of-the-art computer architectures. In this thesis we shall investigate two alternative approximation techniques to get around the curse of dimensionality.

1.1 Approximating solutions to high-dimensional PDEs

The aim of this thesis is to study and develop efficient numerical methods to approximate solutions to high-dimensional PDEs. The approximation techniques under consideration are based upon two, distinct approaches.

The first idea replaces a single high-dimensional partial differential operator by a linear combination of multiple low-dimensional partial differential operators. The d-dimensional Black–Scholes operator in (3.2) contains a covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$. Inspired by the *principal component analysis* (PCA), well-known for example in statistics, Reisinger & Wittum [71] suggest a transformation of the covariance matrix $\mathbf{\Sigma}$. In financial applications the eigenvalue corresponding to the first principal component is often dominant and this observation will be exploited in this first approximation approach. It turns out that neglecting all other principal components does not yield a good approximation but adding first-order corrections yields a good PCA-based approximation for the Black–Scholes operator. The main advantage of this PCA-based approximation approach is that an analytical approximation to the solution of the Black–Scholes PDE is obtained in terms of the solutions to a one-dimensional PDE and $(d-1)$ two-dimensional PDEs. These PDEs are independent of each other and can therefore be solved in parallel. The PCA-based approximation approach is the subject of Part I in this thesis.

The second approach restricts the rank of the solution of a differential equation and derives a differential equation for the low-rank components of that low-rank solution. For example, a numerical representation of the solution of a two-dimensional problem on a certain grid can be represented by a matrix. From that matrix a *singular value decomposition* (SVD) can be computed to obtain the singular values with the left- and right singular vectors. It is observed that the solution of certain Helmholtz problems that appear in scattering problems are of low rank. Thus instead of solving a differential equation on a full grid the differential equation is projected on the space spanned by the other factor matrices. This leads to an equation for the remaining low-rank factor of the solution. The equation for this low-rank factor can be related to equations that arise in the coupled channel technique. This idea for two-dimensional problems can be extended to larger dimensional problems where we obtain a low-rank Tucker tensor representation of the solution. This projection approach can be extended to solve for the low-rank factors of solutions to time-dependent PDEs. This could result in an alternative for the dynamical low-rank integrator by Lubich et al. [47]. The approach to directly solve for low-rank factors of the solution to high-dimensional differential equations is the subject of Part II.

1.2 Outline of the thesis

The outline of this thesis is as follows. Chapter 2 starts with a short introduction to option valuation. Further the famous Black–Scholes model [3] and the considered financial options through this thesis are introduced. The standard discretization of the Black–Scholes PDE and some remediation for the non-smoothness of the initial condition are given. As an alternative to the PDE-based methods for option valuation also a short introduction in Monte Carlo simulation based methods is presented.

In Chapter 3 the PCA-based approximation approach is introduced and applied to valuate European-style basket options. Further a discussion on the spatial and temporal discretization is given and concluded with a rigorous stability analysis for the spatial and temporal discretization. That chapter contains some numerical experiments where the total error in the discretization and the asymptotic runtime of the PCA-based approximation approach are analyzed.

In Chapter 4 the PCA-based approximation approach is extended to Bermudan-style basket options. The fair value of a Bermudan-style basket option is the solution to a Black–Scholes PDE where at certain time frames an optimal exercise condition is imposed. This optimal exercise condition may cause some inconsistencies in the PCA-based approximation approach. Similar to the European-style basket option, the error in the discretization is analyzed.

In Chapter 5 the PCA-based approximation approach is extended such that the solution to partial differential complementarity problems (PDCPs) can be approximated. These problems arise in valuation of American-style basket options. Different temporal discretizations are considered. Also a comparison with the comonotonic approach is made, where it is observed that the comonotonic approach can be seen as a linear combination of special cases of the PCA-based approximation approach. A numerical comparison between these

two methods is given in that chapter.

As the last chapter in the first part about option valuation, we consider in Chapter 6 an extension of the PCA-based approximation approach to approximate the Greeks. Two PCA-based approximation approaches are formulated to estimate the Deltas. Further also a PCA-based approximation approach for the Gammas is derived. The methods are illustrated with numerical examples for European-, Bermudan- and American-style basket options.

With Chapter 7 the second part of the thesis starts with an introduction to high-dimensional data representation using tensors. A short overview about tensors and different tensor decompositions such as the *Canonical Polyadic (CP) decomposition* and the *Tucker tensor decomposition* is given. Also an outlook about the promising *Tensor Train decomposition* is included.

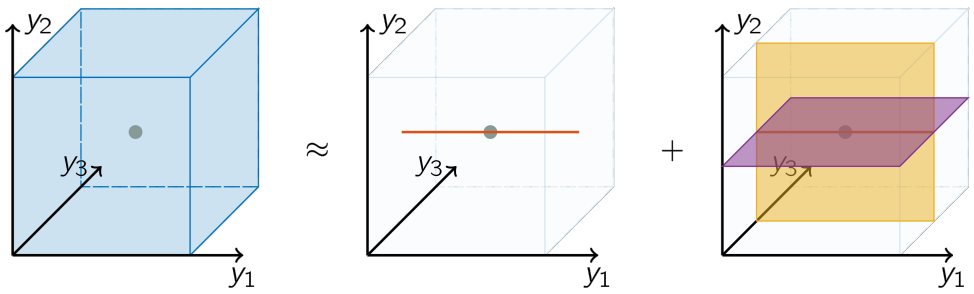
In Chapter 8 we show that scattering solutions for single-, double- and triple-ionization problems can be approximated by low-rank matrices and tensors. An alternating projection method is used to directly solve for low-rank factors of the solution to high-dimensional Helmholtz problems without the need to solve a large linear system. Numerical experiments are shown to validate this approach.

In Chapter 9 an exploration to extend the alternating projection method is done. Instead of solving for the low-rank factor matrices of linear time-independent problems, such as the Helmholtz problem, in this chapter possibilities to solve for the low-rank factor matrices of time-dependent problems are explored. A literature review of the existent dynamical low-rank integrator is given. That method can be interpreted as solving an optimization problem. This leads to some alternative ideas and algorithms to solve for the low-rank factors of a time-dependent solution of a PDE. Also the alternating projection method of Chapter 8 is extended to solve for the low-rank factors of the solution of time-dependent problems. A numerical comparison between the derived methods is given, which show some potential for certain methods. Additional research is needed to arrive at an efficient numerical method to approximate the low-rank factors of solutions to stiff partial differential equations.

Finally, Chapter 10 summarizes the conclusions of this thesis and gives an outlook for possible further research.

Part I

Financial mathematics and option valuation



Chapter 2

Introduction to option valuation and Black-Scholes model

Chapter summary:

In this chapter we give an introduction to option valuation and the Black–Scholes model [3].

In practice, option valuation of different kind of options is done via partial differential equations or via Monte Carlo simulations. In this chapter we give a general introduction for both techniques. Although numerical methods that solve partial differential equations are studied in this thesis, a short introduction in option valuation using simulations is given.

Especially the estimation of the Greeks using pathwise derivatives [5] is reviewed together with the valuation of American-style options using a method proposed by Longstaff and Schwartz [52].

Finally, an outlook for the rest of this first part of the thesis is given where we numerically approximate solutions to partial differential equations for high-dimensional Black–Scholes problems.

2.1 Introduction to option valuation

In this first part of the thesis we will mainly focus on option valuation as widely used in financial markets and studied in financial mathematics. A financial option is a financial derivative product depending on an underlying asset, for example a stock of a company.

A financial option is a security or contract between two parties that gives the holder the right to buy (i.e. a *call option*) or sell (i.e. a *put option*), from the writer, an underlying asset subject to some contract parameters until a specified moment in time, called the *maturity time* T. Remark that the holder has the right but not the obligation to buy or sell the

underlying asset. An example of a contract parameter is the prescribed *strike price* K for which the asset can be sold or bought.

A European-style option can only be exercised at maturity time T while an American-style option can be exercised once at any time between today (as the time of inception of the option) and maturity. In this thesis we will also consider Bermudan-style[1] options and this style of options can be seen as something between European- and American-style options. A Bermudan-style option can be exercised once at one of a finite number of possible exercise dates between today and maturity as prescribed in the contract.

Instead of trading an option on a single asset, it is also possible to trade an option on multiple assets, also called a basket of assets, for example the weighted assets in an index. In this thesis we will consider *basket options* on a (weighted) arithmetic average of correlated assets.

Because the holder of an option has a right but no obligation to buy or sell an asset, the option has financial value. Indeed, an option can be seen as an insurance against a large raise (in case of a call option) or a large drop (in case of a put option) in the asset price. A natural question that arises is then: *'what is the fair value of an option?'*.

At maturity time $\tau = T$ the value of an option is known and prescribed by the *payoff* function $\phi(s)$, where s is the price of the underlying asset. Assume that at time $\tau = T$ the price of the underlying asset is given by S_T. At maturity the holder can compare the *price* S_T with the strike price K of the option and choose to exercise the option (if it is financially beneficial) or not. In the last case the option is worthless. For a vanilla option the payoff function is given by

$$\phi(s) = \begin{cases} \max(K - s,\ 0) & \text{(put option)} \\ \max(s - K,\ 0) & \text{(call option)} \end{cases} \tag{2.1}$$

where $s > 0$ is the asset price.

Of course S_T is not known today, so to answer the question about the fair value of the option today one has to make an assumption about the underlying asset price S_τ, where $\tau \in (0,\ T]$ denotes the time, with $\tau = 0$ being the time of inception of the option. It is common in the present literature to model the underlying asset price S_τ with a stochastic differential equation (SDE)

$$\mathrm{d}S_\tau = r(\tau)S_\tau \mathrm{d}\tau + \sigma(\tau)S_\tau \mathrm{d}W_\tau. \tag{2.2}$$

Here $r(\tau) \geq 0$ is the given risk-free interest rate at time τ, $\sigma(\tau) > 0$ is the given volatility at time τ and W is a standard Brownian motion under the risk-neutral measure. Further, the initial asset price $S_0 > 0$ is given.

The solution to the stochastic differential equation (2.2) is known and given by

$$S_\tau = S_0 \exp\left(\int_0^\tau \left(r(s) - \tfrac{1}{2}\sigma^2(s) \right) \mathrm{d}s + \int_0^\tau \sigma(s) \mathrm{d}W_s \right). \tag{2.3}$$

[1] Apart from the fact that Bermuda is located between Europe and America, these names for the different option styles do not have any geographical meaning.

2.1.1 Black–Scholes model

To successfully answer the question about the fair value of a vanilla option Black and Scholes [3] and Merton [57] developed a model that is currently well-known as the Black–Scholes model.

To arrive at the Black–Scholes formula for the fair value of a European-style option in terms of the price of the underlying asset some 'ideal conditions' in the market for the asset and for the option are made, see also [3, 35]:

1. There are no riskless arbitrage opportunities.
2. The short-term interest rate r is known and is constant over time.
3. The asset price follows a geometric Brownian motion in continuous time with a variance rate proportional to the square of the asset price. Thus the distribution of possible asset prices at the end of any finite interval is log-normal. The volatility σ of the return on the asset is constant.
4. The are no dividends payed during the life of the derivative.
5. There are no transaction costs in buying or selling the asset or the option. All assets are perfectly divisible.
6. It is possible to borrow and lend any fraction of cash, at the short-term interest rate.
7. It is possible to buy or sell any fraction of the asset. There are no penalties to short selling. (A seller who does not own a security will simply accept the price of the security from a buyer, and will agree to settle with the buyer on some future date by paying him an amount equal to the price of the security on that date.)

Thus in the Black–Scholes model it is assumed that the asset price S_τ follows a stochastic process where $r(\tau) \equiv r$ and $\sigma(\tau) \equiv \sigma$ are constant. Then, the exact solution in (2.3) reduces to

$$S_\tau = S_0 \exp\left(\left(r - \tfrac{\sigma^2}{2}\right)\tau + \sigma\sqrt{\tau}Z\right) \tag{2.4}$$

for $\tau \in (0,\ T]$ and where Z is a standard normal random variable [31, 35].

Assuming $S_0 > 0$ yields

$$\ln\left(\frac{S_\tau}{S_0}\right) = \left(r - \tfrac{\sigma^2}{2}\right)\tau + \sigma\sqrt{\tau}Z. \tag{2.5}$$

Thus $\ln\left(\frac{S_\tau}{S_0}\right)$ is normally distributed with mean $\left(r - \frac{\sigma^2}{2}\right)\tau$ and variance $\sigma^2\tau$; hence S_τ is log-normally distributed. Using this, the expected value of S_τ (under the risk-neutral measure) is given by

$$\mathbb{E}[S_\tau] = S_0 e^{r\tau}. \tag{2.6}$$

The fair value $u(s,t)$ of an option at time till maturity $t = T - \tau$ and $S_\tau = s$ is given by the expected value of a random variable

$$u(s,t) = \mathbb{E}\left[e^{-rt}\phi(S_T)\right]\big|_{S_\tau = s}, \tag{2.7}$$

where $s > 0$ denotes the asset price and the payoff function $\phi(s)$ is given by the contract.

As an alternative, the fair value $u(s,t)$ of an option can also be seen as the solution to a partial differential equation (PDE). Again, assume an asset price process following (2.2) under the Black–Scholes model, thus with $r(\tau) \equiv r$ and $\sigma(\tau) \equiv \sigma$ constant. Then Itô's Lemma [35, 44] can be used to derive a PDE for the unique deterministic function $u(s,t)$ that describes the fair value of the option at time till maturity $t = T - \tau$:

$$\frac{\partial u(s,t)}{\partial t} = \tfrac{1}{2}\sigma^2 s^2 \frac{\partial^2 u(s,t)}{\partial s^2} + rs\frac{\partial u(s,t)}{\partial s} - ru(s,t) \tag{2.8}$$

for $s > 0$ and $t \in (0,\ T]$. Equation (2.8) is well-known as the *Black–Scholes partial differential equation*.

2.2 Option valuation via partial differential equations

The Black–Scholes PDE (2.8) is supplemented with the initial condition

$$u(s, t = 0) = \phi(s). \tag{2.9}$$

To complete the model, also a boundary condition for $s = 0$ has to be imposed. At $s = 0$ both the diffusion and convection terms in (2.8) cancel. So, one can impose a Dirichlet boundary condition equivalent to the boundary condition that the PDE (2.8) is also satisfied at $s = 0$:

$$u(s = 0, t) = e^{-rt}\phi(0). \tag{2.10}$$

The semi-closed analytic solution to this PDE for a vanilla European-style option is known, see e.g. [3]:

$$u(s,t) = sN(d_1) - Ke^{-rt}N(d_2)\,,$$

with $s > 0, t \in (0,\ T]$ and

$$d_1 = \frac{\ln\left(\frac{s}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)t}{\sigma\sqrt{t}},$$
$$d_2 = d_1 - \sigma\sqrt{t},$$

where $N(\cdot)$ is the cumulative normal density function. But for Bermudan- or American-style options and basket options in general such formulas for a semi-closed analytic solution are generally lacking in the literature. Therefore, we study efficient and stable numerical methods to approximate the fair values of these type of options.

2.2.1 Discretization

To discretize PDE (2.8), the spatial variable $s \in [0,\ \infty)$ has to be limited to a finite domain. Therefore, a parameter $S_{\max}$ is introduced such that $s \in [0,\ S_{\max}]$, where the parameter

$S_{\max}$ is chosen very large, e.g. between $S_{\max} = 4K$ and $S_{\max} = 8K$. This requires an additional boundary condition at $s = S_{\max}$, for example a linear boundary condition can be choosen:

$$\left.\frac{\partial^2 u(s,t)}{\partial s^2}\right|_{s=S_{\max}} = 0. \tag{2.11}$$

Then the spatial variable s can be discretized on e.g. a uniform mesh $s_i = ih$, with $i = 0, 1, \ldots, m$, and mesh width $h = \frac{S_{\max}}{m}$, where m is the number of spatial discretization points in the domain. Based upon a Taylor expansion, finite difference approximations to $\frac{\partial^2 u}{\partial s^2}$ and $\frac{\partial u}{\partial s}$ can be used (see also e.g. [36]) to discretize PDE (2.8) on this mesh. In this thesis second-order finite difference schemes are employed, so the truncation error is $\mathcal{O}\left(h^2\right)$. Implementing also boundary conditions (2.10) and (2.11) the semi-discrete system

$$\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{u}(t) + \boldsymbol{g}(t) \tag{2.12}$$

is obtained, where the i-th entry of $\boldsymbol{u} \in \mathbb{R}^m$ represents the approximation to the solution $u(s_i, t)$. Further $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is given and represents the discretized Black–Scholes operator and $\boldsymbol{g}(t) \in \mathbb{R}^d$ is a given time-dependent vector that depends on the Dirichlet boundary condition (2.10).

Next, the time variable t is discretized on a uniform temporal mesh $t_i = i\Delta t$, with step size $\Delta t = \frac{T}{N}$, where integer $N \geq 1$ is the number of time steps. For the semi-discrete one-dimensional Black–Scholes PDE the temporal discretization is often done by the θ-method with parameter $\theta \in [0,\ 1]$ (for other time integration methods, see also [36]). Using the θ-method with $\theta = \frac{1}{2}$ one obtains the well-known Crank–Nicolson (CN) scheme. Under natural assumptions on the semi-discrete system (2.12), it is unconditionally stable and has a global temporal error $\mathcal{O}\left(\Delta t^2\right)$.

The total error of the spatial and temporal discretization with respect to the exact solution is defined by

$$E(m,\ N) = \|\boldsymbol{u}_N - \boldsymbol{u}^*\|_\infty \tag{2.13}$$

where $\boldsymbol{u}_N \in \mathbb{R}^m$ denotes the vector of the numerical approximation to the solution at time $t = N\Delta t = T$ and $\boldsymbol{u}^* \in \mathbb{R}^m$ denotes the vector with i-th entry equal to the value $u(s_i, T)$, where $u(s, t)$ is the exact solution to PDE (2.8).

2.2.2 Cell averaging and Backward Euler damping

Although a second-order discretization for both space and time is used for the Black–Scholes PDE (2.8) the total discretization error can have an irregular convergence behaviour. This is related to the non-smoothness of the initial condition. Indeed at strike K the payoff function ϕ is continuous but not differentiable. Finite difference approximations assume sufficient smoothness of the pertinent function, which is violated for most payoff functions ϕ, such as given in (2.1). This will lead to an irregular behaviour in the spatial error, where strong oscillations are observed. A smoothing technique like *cell averaging* can be applied to remedy this undesirable behaviour [50, 66].

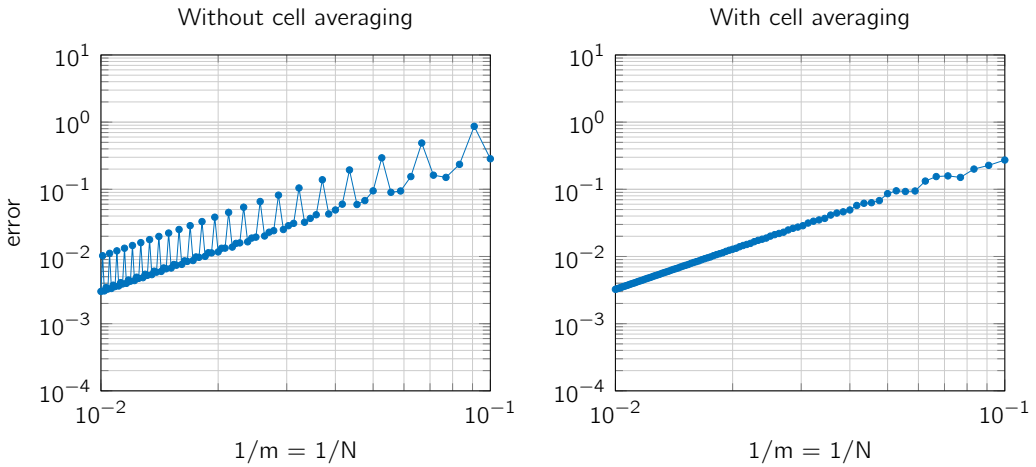


Figure 2.1: Total discretization error for European-style vanilla call option without (left) and with (right) cell averaging as described in Example 2.2.1.

Instead of a pointwise representation $\boldsymbol{\phi} \in \mathbb{R}^m$ of the payoff function $\phi(s)$ on the spatially discretized mesh, its pointwise evaluation nearest to strike K is replaced by an average over a certain cell. Let $i \in \{1, 2, \ldots, m\}$ denote the index such that $|s_i - K|$ is minimal. Then the value ϕ_i in vector $\boldsymbol{\phi}$ is replaced by an integral

$$\phi_i = \frac{1}{h} \int_{s_{i-\frac{1}{2}}}^{s_{i+\frac{1}{2}}} \phi(s) \mathrm{d}s, \tag{2.14}$$

where intermediate mesh points $s_{i-\frac{1}{2}}$ and $s_{i+\frac{1}{2}}$ are defined by

$$s_{i-\frac{1}{2}} := \frac{s_{i-1} + s_i}{2},$$
$$s_{i+\frac{1}{2}} := \frac{s_i + s_{i+1}}{2}.$$

Example 2.2.1. Consider the example of valuation of a European vanilla call option with $r = 0.03$, $\sigma = 0.2$, $T = 1$, $K = 100$ and discretize with $m = N$ space and time-discretization points where $S_{\max} = 4K$. The total discretization error (2.13) with respect to the exact solution for a call option without and with cell averaging is shown in Figure 2.1. It is clear that cell averaging of the payoff function can reduce the oscillation observed in the total discretization error of this numerical approximation to the fair value of an option.

In most cases the total discretization error will be dominated by the spatial discretization error. But also the temporal error can suffer from the non-smoothness of the initial condition ϕ which may affect the convergence behaviour of time integration methods like the Crank–Nicolson scheme. To alleviate this, *Rannacher timestepping* [67] (also known as Backward Euler damping) can be applied. This is done by replacing the first timestep from $t = 0$ to $t = \Delta t$ of the time integration method by two timesteps of length $\frac{\Delta t}{2}$ using the Backward Euler scheme.

2.2.3 Curse of dimensionality

The Black–Scholes PDE (2.8) for vanilla options can be extended to basket options[2]. Therefore, consider a basket with d assets and let $u(\boldsymbol{s},t) = u(s_1, s_2, \ldots, s_d, t)$ be the fair value of a European-style basket option if at time till maturity $t = T - \tau$ the i-th asset price equals s_i where $i = 1, 2, \ldots, d$. Then u satisfies the d-dimensional Black–Scholes PDE [62, 84]

$$\frac{\partial u}{\partial t}(\boldsymbol{s},t) = \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sigma_i\sigma_j\rho_{ij}s_is_j\frac{\partial^2 u}{\partial s_i \partial s_j}(\boldsymbol{s},t) + \sum_{i=1}^{d} rs_i\frac{\partial u}{\partial s_i}(\boldsymbol{s},t) - ru(\boldsymbol{s},t) \tag{2.15}$$

whenever $(\boldsymbol{s},t) \in (0,\infty)^d \times (0,T]$. Further $r \geq 0$ is the given risk-free interest rate, $\sigma_i > 0$ (with $i = 1, 2, \ldots, d$) are the given volatilities and $\boldsymbol{\rho} = (\rho_{ij}) \in \mathbb{R}^{d\times d}$ is the correlation matrix, where $i, j = 1, 2, \ldots, d$, that describes the correlation between the underlying assets.

The PDE (2.15) is also satisfied if $s_i = 0$ for any given i, thus at the boundary of the spatial domain. Next, we have the initial condition given by the payoff function

$$u(\boldsymbol{s},0) = \phi(\boldsymbol{s}), \tag{2.16}$$

whenever $\boldsymbol{s} \in (0,\infty)^d$.

For a put-on-average basket option the payoff function is given by

$$\phi(\boldsymbol{s}) = \max\left(K - \sum_{i=1}^{d}\omega_i s_i,\ 0\right), \tag{2.17}$$

where the prescribed weights $\omega_i > 0$ $(i = 1, 2, \ldots, d)$ are fixed, given by the contract and such that $\sum_{i=1}^{d}\omega_i = 1$.

The number of unknowns in the spatial discretization on the (truncated) domain $(0, S_{\max}]^d$ grows exponentially in the dimension d. Indeed, if m is the number of discretization points per asset in the domain, then the total number of discretization points equals m^d. This approach is feasible for $d = 1, 2$ and $d = 3$, but when d becomes moderate or large, e.g. $d \geq 5$, then numerically solving PDE (2.15) with a reasonably fine spatial mesh becomes impractical.

Example 2.2.2. As an example to illustrate this exponential dependence of the runtime on the dimension we consider the valuation of a European-style basket put-on-average option with the parameters of Set A by Reisinger and Wittum [71] as given in Appendix A. For lower dimensional problems we restrict Set A to the first d assets. The measured total runtime for solving the d-dimensional Black–Scholes PDE for dimensions $d \in \{2, 3, 4, 5\}$ is shown in Figure 2.2. In this example the number of time steps N is taken equal to the number of discretization points per asset m. Further a model to predict the total runtime is fit on the data to describe the asymptotic behaviour. Asymptotically, but before some out-of-memory artifacts appear, the total runtime scales indeed approximately $\mathcal{O}(Nm^d)$.

[2]For example, one can think of an option on all assets weighed in an index, such as the BEL20 where one has $d = 20$ assets in a basket.

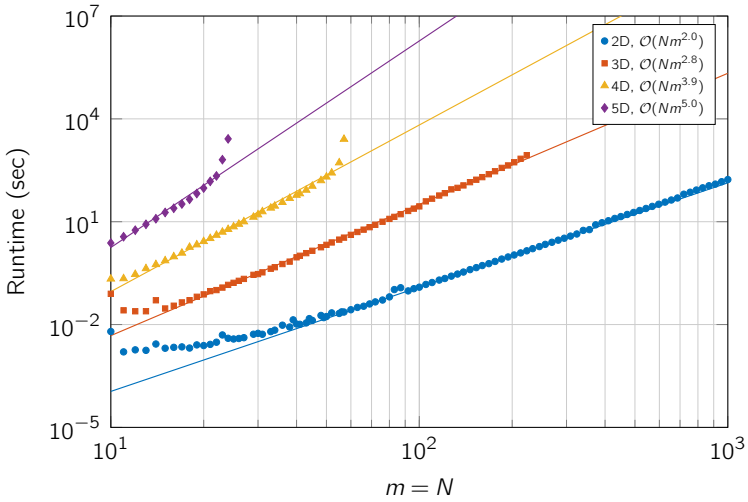


Figure 2.2: Total runtime for solving the d-dimensional Black–Scholes PDE for dimensions $d \in \{2,3,4,5\}$ using a standard spatial discretization of the d-dimensional space.

2.3 Option valuation via Monte Carlo simulation

For basket options with a moderate or large number of assets, the standard approach for numerically solving a high-dimensional PDE like (2.15) is computationally too expensive. As an alternative, Monte Carlo (MC) simulations are often used to valuate these basket options. The price processes of the underlying assets are simulated along different paths, where the number of simulated paths is denoted by `Npaths`.

Although this thesis mainly focuses on solving partial differential equations, here a short introduction in Monte Carlo simulations and Least Squares Monte Carlo (LSMC) simulations is given. A main advantage of Monte Carlo simulations is that the computational cost scales approximately linearly in the number of assets.

2.3.1 Option and Delta values for European-style basket option

For a basket with d assets we model the underlying asset price process S_τ^i (for $i = 1,2,\dots,d$) similar to (2.2), with constant risk-free interest rate r and constant volatilities σ_i (where $i = 1,2,\dots,d$), using a multidimensional geometric Brownian motion which is given by a system of stochastic differential equations (SDEs):

$$\mathrm{d}S_\tau^i = rS_\tau^i \mathrm{d}\tau + \sigma_i S_\tau^i \mathrm{d}W_\tau^i \tag{2.18}$$

for $0 < \tau \leq T$ and $i = 1,2,\dots,d$. Here W^i $(i = 1,2,\dots,d)$ is a multidimensional standard Brownian motion with given correlation matrix $\boldsymbol{\rho} = (\rho_{ij})_{i,j=1}^d$. The stochatic variables S_τ^i for asset price processes S^i (with $i = 1,2,\dots,d$) can be represented by a d-dimensional vector $\boldsymbol{S_\tau} = (S_\tau^i)_{i=1}^d$. Further, the initial asset prices $S_0^i > 0$ are given.

To generate the d correlated random normal variables for the Brownian motion the Cholesky factorization $\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C} = \boldsymbol{\rho}$ can be used, where $\boldsymbol{C}$ is an upper triangular matrix [33]. Now, the d-dimensional row vector describing the Brownian increments $\boldsymbol{W_\tau}$ for the d assets can be described as

$$\boldsymbol{W_\tau} = \sqrt{\tau}\boldsymbol{Z}\boldsymbol{C},$$

where $\boldsymbol{Z}$ is a row vector representing a d-dimensional standard normal random variable.

The solution to the system of stochastic differential equations (2.18) is given by

$$S_\tau^i = S_0^i \exp\left(\left(r - \tfrac{1}{2}\sigma_i^2\right)\tau + \sigma_i W_\tau^i\right), \tag{2.19}$$

for $i = 1, 2, \ldots, d$ and $\tau \in (0,\ T]$.

Written in vector notation, this is given by

$$\boldsymbol{S_\tau} = \boldsymbol{S_0} * \exp\left(\left(r\boldsymbol{e} - \frac{\boldsymbol{\sigma} * \boldsymbol{\sigma}}{2}\right)\tau + \boldsymbol{\sigma} * \boldsymbol{W_\tau}\right), \tag{2.20}$$

for $\tau \in (0,\ T]$. Here $\boldsymbol{\sigma}$ and $\boldsymbol{S_\tau} \in \mathbb{R}^d$ are interpreted as row vectors. Further the notation $\boldsymbol{x} * \boldsymbol{y}$ is used for the element wise product between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. The function $\exp(\cdot)$ applied on a vector is taken element wise and the vector $\boldsymbol{e} = [1, 1, \ldots, 1] \in \mathbb{R}^d$.

2.3.1.1 Option value for European-style basket option

Let $u(\boldsymbol{s}, t) = u(s_1, s_2, \ldots, s_d, t)$ denote the fair value of a European-style basket option if at time till maturity $t = T - \tau$ the i-th asset price equals s_i, with $i = 1, 2, \ldots, d$. Similar to (2.7), the fair value $u(\boldsymbol{s}, t)$ of a basket option can also be seen as the expected value of a random variable P_t:

$$u(\boldsymbol{s}, t) = \mathbb{E}\left[\underbrace{e^{-rt}\phi(\boldsymbol{S_T})}_{P_t}\right]\Bigg|_{\boldsymbol{S_\tau} = \boldsymbol{s}}, \tag{2.21}$$

where $\boldsymbol{s} \in (0,\ \infty)^d$ denotes the vector with asset prices and the payoff function $\phi(\boldsymbol{s})$ is given by the contract.

Define the one-dimensional random variable $\overline{S}_T$ as the pertinent linear combination of the random variables for the values of the different assets:

$$\overline{S}_T = \sum_{i=1}^{d} \omega_i S_T^i. \tag{2.22}$$

The payoff function (2.17) for $\boldsymbol{S_T}$ reduces to the one-asset payoff function for a vanilla put option as given in (2.1) with the value of the asset given by $\overline{S}_T$ in (2.22).

A Monte Carlo simulation for valuation of this type of basket options under a multidimensional geometric Brownian motion is given in Algorithm 1 (especially lines 7 and 10).

Example 2.3.1. As an example for valuation of a European-style basket put-on-average option we consider Set A as given in Appendix A by Reisinger and Wittum [71].

Algorithm 1: Monte Carlo simulation for option valuation and estimation of Delta-k (with $k = 1, 2, \ldots, d$) of a European-style basket option under Black–Scholes model.

Given $\boldsymbol{S_0} \in \mathbb{R}^d$, K, r, $\boldsymbol{\sigma} \in \mathbb{R}^d$, $\boldsymbol{\rho} \in \mathbb{R}^{d \times d}$, T, $\texttt{Npaths}$, $\phi(\boldsymbol{s})$, $\frac{\partial \phi}{\partial s_k}(\boldsymbol{s})$;
$\boldsymbol{C}^\mathsf{T}\boldsymbol{C} = \texttt{chol}\,[\boldsymbol{\rho}]$;
for $i = 1, 2, \ldots, \texttt{Npaths}$ **do**
  $\boldsymbol{Z}^{(i)} \sim [\mathcal{N}(0,\ 1),\ \mathcal{N}(0,\ 1),\ \ldots,\ \mathcal{N}(0,\ 1)]$;
  $\boldsymbol{W}_T^{(i)} = \sqrt{T}\,\boldsymbol{Z}^{(i)}\boldsymbol{C}$;
  $\boldsymbol{S}_T^{(i)} = \boldsymbol{S_0} * \exp\left(\left(r\boldsymbol{e} - \frac{1}{2}\boldsymbol{\sigma} * \boldsymbol{\sigma}\right)T + \boldsymbol{\sigma} * \boldsymbol{W}_T^{(i)}\right)$;
  $P_T^{(i)} = e^{-rT}\phi\left(\boldsymbol{S}_T^{(i)}\right)$;
  $\Delta_k^{(i)} = e^{-rT}\frac{S_T^k}{S_0^k}\frac{\partial \phi}{\partial s_k}\left(\boldsymbol{S}_T^{(i)}\right)$;
end
Option value $P_T \quad = \frac{1}{\texttt{Npaths}} \sum_{i=1}^{\texttt{Npaths}} P_T^{(i)}$;
Delta-k value $\Delta_k \quad = \frac{1}{\texttt{Npaths}} \sum_{i=1}^{\texttt{Npaths}} \Delta_k^{(i)}$;

We consider the valuation of this basket option with $\boldsymbol{S_0} = (K, K, \ldots, K) \in \mathbb{R}^d$. Using a PCA-based approximation approach[3] the reference option value today, $P_T^* = 0.17577$, is obtained.

The Monte Carlo simulation is done with $\texttt{Npaths} \geq 1$ simulated paths. The estimations for the mean and the variance of the option value are given by [31]

$$\begin{aligned} \overline{P_T} &= \frac{1}{\texttt{Npaths}} \sum_{i=1}^{\texttt{Npaths}} P_T^{(i)}, \\ \overline{V}^2 &= \frac{1}{\texttt{Npaths} - 1} \sum_{i=1}^{\texttt{Npaths}} \left(P_T^{(i)} - \overline{P_T}\right)^2, \end{aligned} \tag{2.23}$$

where $P_T^{(i)}$ is the option value along the i-th path.

By the Central Limit Theorem, the error of the estimated mean with respect to the real mean, approximated by $P_T^* - \overline{P_T}$, behaves like $\mathcal{N}\left(0,\ \frac{V^2}{\texttt{Npaths}}\right)$, where V^2 is the unknown variance of the random variable P_T. Thus for a large number of simulated paths, the Monte Carlo estimation $\overline{P_T}$ approximates P_T^* with a *standard error* $\frac{V}{\sqrt{\texttt{Npaths}}}$. Hence, the standard error decays $\mathcal{O}(1/\sqrt{\texttt{Npaths}})$ when $\texttt{Npaths}$ increases.

Using $\overline{V}$ as estimation for V in the standard error, the 95% *confidence interval* (CI) for P_T^* is estimated by

$$95\%\ \text{CI} \approx \left[\overline{P_T} - \frac{1.96V}{\sqrt{\texttt{Npaths}}},\ \overline{P_T} + \frac{1.96V}{\sqrt{\texttt{Npaths}}}\right]. \tag{2.24}$$

The obtained Monte Carlo estimations for the fair option value and the PCA-based reference value for an increasing number of simulated paths $\texttt{Npaths}$ are shown in Figure 2.3a.

[3] For the details about this approach we refer to Chapter 3.

Next, the difference with respect to the reference value and the Monte Carlo standard error are shown in Figure 2.3c. Indeed, the standard error and the difference with respect to the reference value behave $\mathcal{O}(1/\sqrt{\texttt{Npaths}})$. We remark that this rather slow decay of the standard error for Monte Carlo methods can be improved using for example Multilevel Monte Carlo methods [22, 23].

2.3.1.2 Deltas for European-style basket option

Besides the fair value of an option, in financial practice also the Greeks are quantities of main interest. The Greeks describe the sensitivity of the option value to a change in one of the underlying financial parameters. For each asset in the basket we have a Greek Delta. Thus, for the k-th asset, the Delta-k is defined by $\Delta_k(\boldsymbol{s},t) = \frac{\partial u(\boldsymbol{s},t)}{\partial s_k}$, for $k = 1,2,\ldots,d$.

We will estimate Delta-k for European-style basket options by pathwise derivatives, a well-known technique by Broadie and Glasserman [5] for estimating the Greeks of options. This technique can be applied in particular to estimate the Deltas for basket options. As an example we illustrate this for the Greek Delta-k, but other Greeks can be estimated in a similar manner.

Using (2.21) and the definition of Delta-k it follows that

$$\Delta_k(\boldsymbol{s},t) = \frac{\partial u(\boldsymbol{s},t)}{\partial s_k} = \mathbb{E}\left[e^{-rt} \sum_{i=1}^{d} \frac{\partial \phi(\boldsymbol{S_T})}{\partial s_i} \frac{\partial S_T^i}{\partial s_k} \right] \Bigg|_{\boldsymbol{S_\tau}=\boldsymbol{s}}, \tag{2.25}$$

where S_T^i denotes the i-th entry of the vector with random variables $\boldsymbol{S_T}$.

The derivative of S_τ^i with respect to s_k is given by

$$\frac{\partial S_\tau^i}{\partial s_k} = \begin{cases} \exp\left(\left(r - \frac{1}{2}\sigma_i^2\right)\tau + \sigma_i W_\tau^i\right) = \frac{S_\tau^k}{S_0^k} & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases} \tag{2.26}$$

for $i,k = 1,2,\ldots,d$.

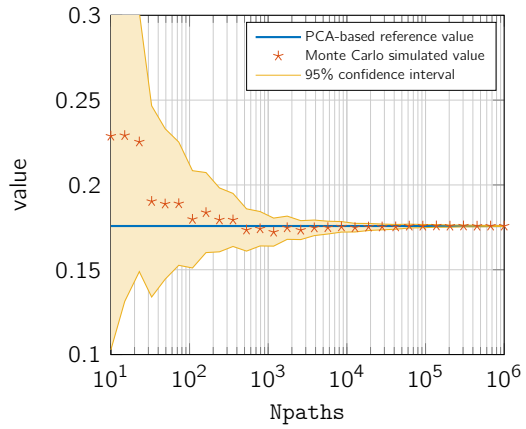
Further, the partial derivatives of the payoff function ϕ as given in (2.17) are

$$\frac{\partial \phi(\boldsymbol{s})}{\partial s_i} = \begin{cases} -\omega_i & \text{if } K > \sum_{j=1}^{d} \omega_j s_j \\ 0 & \text{if } K \leq \sum_{j=1}^{d} \omega_j s_j \end{cases}, \tag{2.27}$$

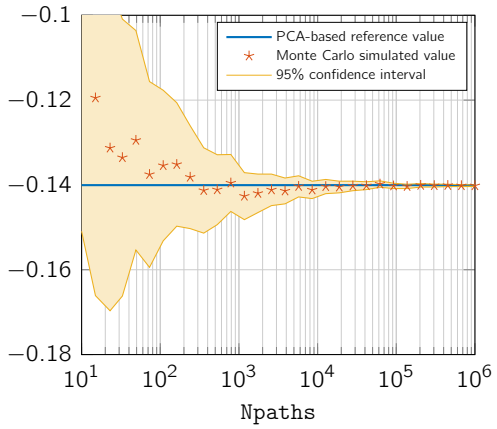
for $i = 1,2,\ldots,d$. Thus the Greek Delta-k can be expressed as the expectation of a differentiated payoff function

$$\Delta_k(\boldsymbol{s},t) = \frac{\partial u(\boldsymbol{s},t)}{\partial s_k} = \mathbb{E}\left[e^{-rt} \frac{S_T^k}{S_0^k} \frac{\partial \phi(\boldsymbol{S_T})}{\partial s_k} \right] \Bigg|_{\boldsymbol{S_\tau}=\boldsymbol{s}}. \tag{2.28}$$

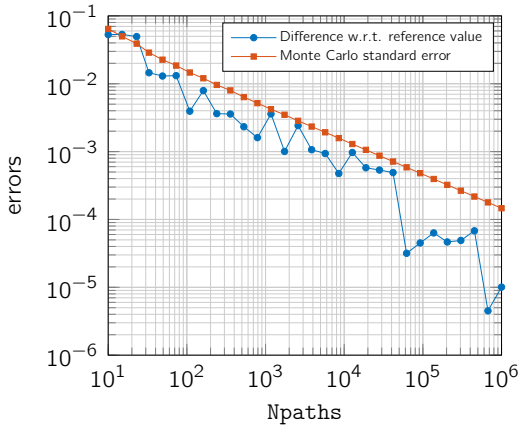
Thus, with exactly the same set of asset price paths it is possible to estimate both the fair value of a basket option and its Deltas. The Monte Carlo simulation for estimation of Delta-k for this basket option is given in Algorithm 1 (especially lines 8 and 11).



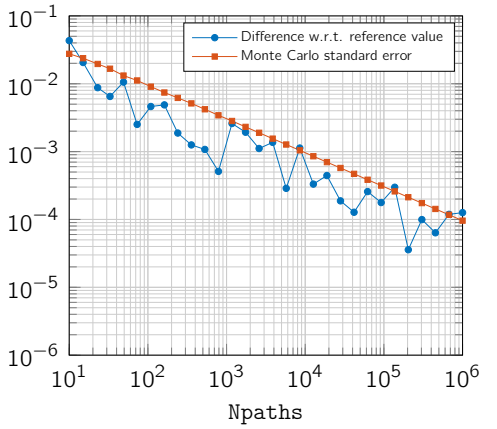
(a) Option value for put-on-average European basket option.



(b) Delta-k value for put-on-average European basket option.



(c) Errors in option value for put-on-average European basket option.



(d) Errors in Delta-k value for put-on-average European basket option.

Figure 2.3: Fair option value (left) and Delta-k, with $k = 1$, estimation (right) of put-on-average European-style basket option under the Black–Scholes model using Monte Carlo simulation.

Example 2.3.1 (continued)**.** Consider again Set A as given in Appendix A. For Delta-1 the reference value at $\boldsymbol{S_0} = (K, K, \ldots, K) \in \mathbb{R}^d$ obtained by a PCA-based approximation approach is given by $\Delta_1^* = -0.14005$. The obtained Monte Carlo estimations for the value of this Delta based on (2.28) and its difference with respect to this reference value are shown in Figures 2.3b and 2.3d.

The results indicate indeed that the pathwise derivative technique can be successfully applied to estimate the Deltas for European-style basket options using standard Monte Carlo simulation. It converges to the same values for Delta-k as obtained using a PDE reference method. Again, the standard error and the difference with respect to the reference value decay with $\mathcal{O}(1/\sqrt{\texttt{Npaths}})$, where `Npaths` is the number of simulated paths.

2.3.2 Option value for American-style vanilla option

For American-style vanilla options, Longstaff and Schwartz [52] present an algorithm that uses simulations to approximate their fair value[4]. The intuition behind this approach comes from the observation that the holder of an American-style option optimally compares, at any given time instant, the payoff from immediate exercise with the expected payoff from continuation. The key insight of the approach by Longstaff and Schwartz is that the conditional expectation of the payoff from continuing can be estimated from cross-sectional information available from the simulation. The conditional expectation function for the expected payoff from continuation can be seen as a least squares solution over specific data. By estimating the conditional expectation at each time instant an optimal exercise strategy along each simulation path can be formulated and used to valuate the American-style option. This technique is also called *Least Squares Monte Carlo* (LSMC) approach in the literature. In [52, Section 1] an illustrative numerical example is presented to explain the Longstaff–Schwartz approach to valuate American-style vanilla options.

We mention that the fair value of an American-style vanilla option can again be expressed as an expectation of a discounted payoff. Assume that the American-style option for a given asset price path is optimally exercised at time $\tau^* \in [0,\ T]$. Then, the fair value of the option can then be written as an expectation [24, 75]

$$u(s,t) = \mathbb{E}\left[\underbrace{e^{-r(t-t^*)}\phi(S_{\tau^*})}_{P_t}\right]\Bigg|_{S_\tau = s}, \tag{2.29}$$

where $t^* = T - \tau^*$ depends on the asset price path.

Example 2.3.2. As an example, choose $r = 0.05$, $\sigma = 0.25$, $S_0 = 100$, $K = 100$ and $T = 1$. For approximating the American-style put option price, we take $E = 100$ equidistant exercise points in time. As reference value, the corresponding Bermudan-style option value approximation from the Black–Scholes PDE discretization of Section 2.2 with $m = N = 10^4$

[4] Actually the fair value of a Bermudan-style vanilla option is estimated, but when the (finite) number of exercise times for a Bermudan-style option increases the Bermudan-style option value converges to the American-style option value.

is employed[5] (including cell averaging and backward Euler damping). With this approach the reference value $P_T^* = 7.96833$ is obtained. The estimated option values and errors are shown in Figures 2.4a and 2.4c.

The results confirm, as expected, that the Longstaff–Schwartz method can be used to accurately valuate American-style options. The standard error behaves similarly as for a Monte Carlo method to valuate European-style options. The difference with respect to the reference value seems to level off when the number of simulated paths increases. Up to a certain number of simulated paths, the Longstaff–Schwartz method seems to generate an estimation for the American-style vanilla option value that lies close to the option value obtained by the PDE discretization approach. From this comparison, it is not clear which of the two approaches has the largest error that causes this observed difference. We mention that it is well-known that the choice of basis functions for the regression may have some impact on the accuracy of the Longstaff–Schwartz method.

2.3.3 Delta value for American-style vanilla option

The Longstaff–Schwartz approach [52] for valuation of American-style vanilla options can be combined with pathwise derivatives [5] for estimating the Greeks similar to European-style options.

Observe that the Longstaff–Schwartz method constructs an optimal exercise strategy for American-style options. According to that optimal exercise strategy for all simulated paths the payoff is evaluated, discounted and averaged to approximate the expectation of the payoff for the option.

Instead of using the discounted payoff function for the computation of the expected value under the optimal exercise strategy, one can also use an other function. To approximate the Delta of an option using pathwise derivatives in the European-style context, it has been observed (cf. Section 2.3.1.2) that this is just the expectation of an other function over the same simulation [24]. Applying this idea here and compute

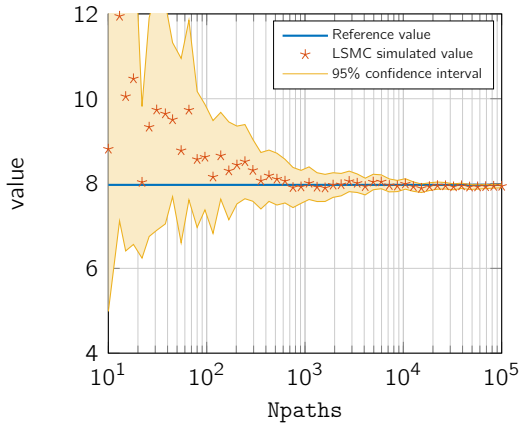
$$\Delta(s,t) = \frac{\partial u(s,t)}{\partial s} = \mathbb{E}\left[e^{-r(t-t^*)} \frac{\mathrm{d}\phi(S_{\tau^*})}{\mathrm{d}s} \frac{S_{\tau^*}}{S_0} \right] \Bigg|_{S_\tau = s}, \tag{2.30}$$

where $t^* = T - \tau^*$ depends on the asset price path. This gives, with exactly the same set of asset price paths, the pathwise approximation to the Delta of an American-style option.

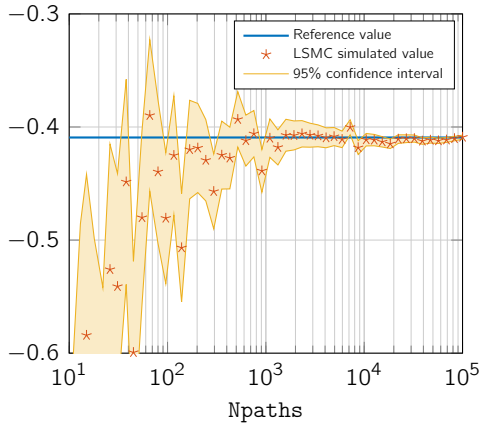
Example 2.3.2 (continued)**.** The reference value for the Delta of an American-style option is approximated by the Delta value for a Bermudan-style option with $E = 100$ equidistant exercise points in time. The reference value is obtained from the PDE discretization in a similar way as in Example 2.3.2 and is given by $\Delta^* = -0.40928$.

The obtained Monte Carlo estimations for the Delta value and the difference with respect to this reference value are shown in Figures 2.4b and 2.4d.

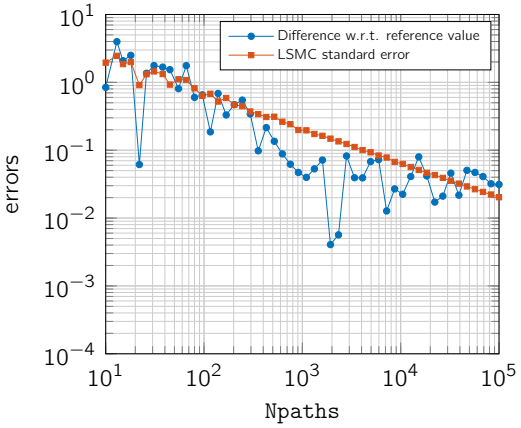
[5] In Section 2.2 only the valuation of European-style options is presented. This approach can be extended to valuate Bermudan-style options by explicitly imposing the optimal exercise condition at the exercise points in time. For the valuation of Bermudan (basket) option we refer to Chapter 4.



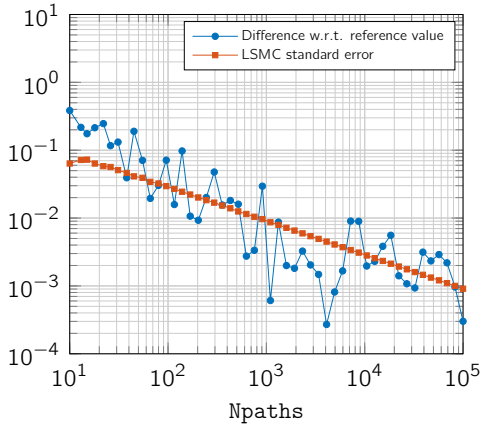
(a) Option value for American vanilla put option.



(b) Delta for American vanilla put option.



(c) Errors in option value for American vanilla put option.



(d) Errors in Delta value for American vanilla put option.

Figure 2.4: Fair option value (left) and Delta estimation (right) of an American-style vanilla put option under the Black–Scholes model using the Longstaff–Schwartz approach.

The numerical results indicate indeed that the pathwise derivative technique can successfully be combined with the Longstaff–Schwartz method to estimate the Greeks for American-style vanilla options. Again, the standard error and the difference with respect to the reference value decay with $\mathcal{O}(1/\sqrt{\texttt{Npaths}})$, where `Npaths` is the number of simulated paths.

2.3.4 Option and Delta values for American-style basket option

Finally, similar to Section 2.3.1 for European-style basket options, we consider the valuation and estimation of option and Delta-k values for American-style basket options. For a basket with d assets we model the underlying asset price process S_τ^i with $\tau \in [0,\ T]$ (for $i = 1, 2, \ldots, d$) again using a multidimensional geometric Brownian motion, given by (2.18).

The method by Longstaff and Schwartz for American-style options on one asset, needs only some minor modifications to adapt to American-style basket options. For the fixed prescribed weights ω_i in the payoff (2.17), consider the one-dimensional, average price process $\overline{S}_\tau$ given in (2.22). The payoff (2.17) is then clearly given by just the payoff for a vanilla put option (2.1) on the average price.

In this modification for American-style basket options one can use the one-dimensional averaged stock prices $\overline{S}_\tau$ at time τ and the corresponding discounted cash flow received at time τ in the regression used in the Longstaff–Schwartz approach [92].

In the following example, we observe that it yields remarkably well approximations.

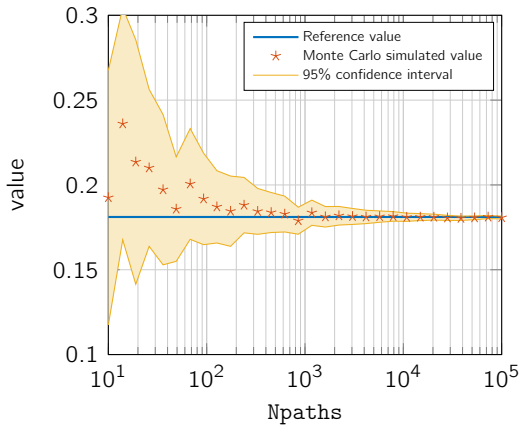
Example 2.3.1 (continued)**.** Consider again the financial parameters of Set A in Appendix A. Let us estimate the fair option value and Delta-k of an American-style basket option. With exactly the same random paths for the multi-asset price process the values of the option and Delta-k can be estimated.

The reference values for the option and the Deltas at $\boldsymbol{S_0} = (K, K, \ldots, K) \in \mathbb{R}^d$ are obtained using PCA-based PDE methods[6]. The obtained reference value for the option is given by $P_T^* = 0.18110$ and the reference value for Delta-1 is given by $\Delta_1^* = -0.14624$.

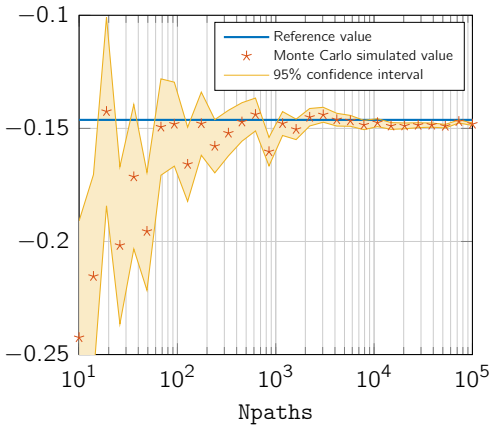
For this option the fair value and the Delta-1 value with the differences with respect to their reference values are shown in Figure 2.5. The results suggest that the Longstaff–Schwartz method can also be applied to valuate American-style basket options. Moreover, the pathwise derivative technique can also be used to estimate the Deltas for American-style basket options.

As mentioned already in the discussion of the Longstaff–Schwartz method for American-style vanilla options also with the American-style basket options it is observed that the difference with respect to the reference solution seems to level off when the number of simulated paths increases. This behaviour is observed for both the estimation of the option and the Delta value.

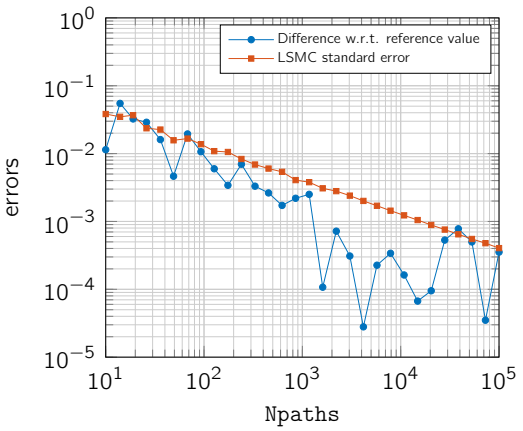
[6] For the details about this method we refer to Chapters 5 and 6.



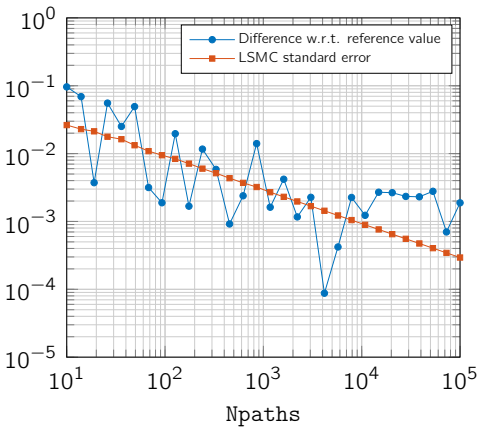
(a) Option value for put-on-average American basket option.



(b) Delta-k value for put-on-average American basket option.



(c) Errors in option value for put-on-average American basket option.



(d) Errors in Delta-k value for put-on-average American basket option.

Figure 2.5: Fair option value (left) and Delta-k, with $k = 1$, estimation (right) of put-on-average American-style basket option under the Black–Scholes model using the Longstaff–Schwartz approach.

2.4 Outlook

In this chapter we have reviewed different types and styles of financial options. Further the Black–Scholes equation is introduced and we presented some basic techniques to numerically valuate an option using both partial differential equations and Monte Carlo simulation. Further the pathwise derivative technique is discussed to estimate the Greeks.

In practice both PDE methods and Monte Carlo simulations are used to valuate options[7]. Main advantages of PDE methods for valuation of options are the well-understood convergence behaviour and the Greeks that appear naturally in the PDE. A drawback for valuation of basket options by the numerical solution of PDEs is that standard methods are computationally too expensive when the number of assets in a basket is moderate or large.

As an alternative approach for valuation of basket options, where the number of assets is large, Monte Carlo simulation can be applied. This approach is often straightforward to implement and the computational cost increases linearly in the number of assets. Besides the fact that the results using Monte Carlo simulation are probabilistic, a well-known drawback of Monte Carlo simulation is that a lot of simulations are needed and the convergence is rather slow in the number of simulated paths.

In the remainder of this first part of the thesis we study an analytical technique by Reisinger and Wittum [71] to approximate the solution of a high-dimensional PDE as given in (2.15) by a linear combination of solutions to low-dimensional PDEs. This opens up the possibility to numerically valuate basket options with a moderate number of assets using numerical PDE methods.

We mention also the numerical discretization technique based on *sparse grids* [6] that effectively reduces the number of grid points compared to a full grid solution. This technique has been applied for the numerical valuation of basket options (e.g. in [51]) but the number of grid points on a sparse grid is still too large for a high number of assets.

Instead, in the following we will focus on the Principal Component Analysis based (PCA-based) approximation approach by Reisinger and Wittum [71]. This will yield an analytical approximation to the value of the solution of (2.15). This analytical approximation consists a linear combination of solutions to PDEs with low dimension. In typical financial applications, solutions to just one- and two-dimensional PDEs are sufficient. This leads to an approximation technique with a computational cost that is only linear in the number of assets in the basket.

In Chapter 3 we introduce the PCA-based approximation approach of [71] and apply it to numerically valuate European-style basket options. In Chapters 4 and 5 we extend this PCA-based approximation approach to numerically valuate Bermudan- and American-style basket options. The PCA-based approximation approach can also be used to numerically estimate the Greeks for European-, Bermudan- and American-style basket options and that will be the topic of Chapter 6. Much attention will be given to the convergence behaviour of the numerical approximations.

[7] Two other important, contemporary approaches are numerical integration and machine learning, which have not been discussed in this chapter.

Chapter 3

European-style basket options

Chapter summary:

In this chapter we study the principal component analysis based approach introduced by Reisinger and Wittum for the approximation of European-style basket option values via high-dimensional partial differential equations (PDEs).

This PCA-based approximation approach requires the solution of just a limited number of low-dimensional PDEs.

Next, an efficient discretization of the pertinent PDEs is presented and a rigorous stability analysis is given for the spatial and temporal discretizations.

This approximation approach with an efficient discretization leads to a favourable convergence behaviour.

The content of this chapter is mainly based on published work in [41] and [39].

3.1 Introduction

In this chapter we introduce the principal component analysis based (PCA-based) approximation approach introduced by Reisinger and Wittum [71] that deals with the valuation of European-style basket options.

Basket options constitute a popular type of financial derivatives and possess a payoff depending on a weighted average of different assets. In general, exact valuation formulas for such options are not available in the literature in semi-closed analytic form. Therefore, the development and analysis of efficient approximation methods for their fair values is of much importance.

In this chapter, we consider the valuation of European-style basket options through partial differential equations (PDEs). If d denotes the number of different assets in the basket, then the pertinent PDE is d-dimensional. In this thesis, we are interested in the situation where d is medium or large, for example $d \geq 5$. It is well-known that this renders the application of standard discretization methods for PDEs impractical, due to the curse of dimensionality.

For this style basket options, an effective approach has been introduced by Reisinger and Wittum [71] and next studied in, e.g., Reisinger and Wissmann [68, 69, 70] and in our recent papers [39, 41]. This approach is based on a principal component analysis (PCA) and yields an approximation formula for the value of the basket option that requires the solution of a limited number of only low-dimensional PDEs.

A *European-style basket option* is a financial contract that gives the holder the right to buy or sell a prescribed weighted average of d assets at a prescribed maturity date T for a prescribed strike price K. We assume in this thesis the well-known Black–Scholes model. Thus, the asset prices S_τ^i (with $i = 1, 2, \ldots, d$) for $\tau \in [0,\ T]$ evolve according to a multidimensional geometric Brownian motion, which is given (under the risk-neutral measure) by the system of stochastic differential equations (SDEs)

$$dS_\tau^i = rS_\tau^i d\tau + \sigma_i S_\tau^i dW_\tau^i \quad (0 < \tau \leq T,\ 1 \leq i \leq d). \tag{3.1}$$

Here τ is time, with $\tau = 0$ representing the time of inception of the option, $r \geq 0$ is the given risk-free interest rate, $\sigma_i > 0$ $(i = 1, 2, \ldots, d)$ are the given volatilities and W^i $(i = 1, 2, \ldots, d)$ is a multidimensional standard Brownian motion with given correlation matrix $\boldsymbol{\rho} = (\rho_{ij})_{i,j=1}^d$. Further, the initial asset prices $S_0^i > 0$ $(i = 1, 2, \ldots, d)$ are given. In essentially all financial applications, the correlation matrix is full.

Let $u(\boldsymbol{s}, t) = u(s_1, s_2, \ldots, s_d, t)$ be the fair value of a European-style basket option if at time till maturity $t = T - \tau$ the i-th asset price equals s_i $(i = 1, 2, \ldots, d)$. Financial mathematics theory yields that u satisfies the d-dimensional time-dependent PDE [62, 84]

$$\frac{\partial u}{\partial t}(\boldsymbol{s}, t) = \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d \sigma_i \sigma_j \rho_{ij} s_i s_j \frac{\partial^2 u}{\partial s_i \partial s_j}(\boldsymbol{s}, t) + \sum_{i=1}^d r s_i \frac{\partial u}{\partial s_i}(\boldsymbol{s}, t) - r u(\boldsymbol{s}, t) \tag{3.2}$$

whenever $(\boldsymbol{s}, t) \in (0, \infty)^d \times (0, T]$.

The PDE (3.2) is also satisfied if $s_i = 0$ for any given i, thus at the boundary of the spatial domain. At maturity time of the option its fair value is known and specified by the particular option contract.

If ϕ is the given payoff function of the option, then one has the initial condition

$$u(\boldsymbol{s}, 0) = \phi(\boldsymbol{s}) \tag{3.3}$$

whenever $\boldsymbol{s} \in (0, \infty)^d$.

In this thesis, we shall consider the class of basket put options. These have a payoff function given by

$$\phi(\boldsymbol{s}) = \max\left(K - \sum_{i=1}^d \omega_i s_i,\ 0 \right) \tag{3.4}$$

with prescribed weights $\omega_i > 0$ $(i = 1, 2, \ldots, d)$ such that $\sum_{i=1}^{d} \omega_i = 1$.

The outline of this chapter is as follows. Following Reisinger and Wittum [71], in Section 3.2.1 a convenient coordinate transformation is applied to the PDE (3.2) for European-style basket options by means of a spectral decomposition of the covariance matrix. This way, a d-dimensional time-dependent PDE for a transformed option value function is obtained in which each coefficient is directly proportional to one of the eigenvalues. Using a minor assumption on the covariance matrix in Section 3.2.1.3 a proof for Dirichlet boundary conditions for the transformed domain is given. In Section 3.2.2, the feature that the transformed option value function is obtained in a form where each coefficient is directly proportional to one of the eigenvalues is used. This feature is exploited to derive a principal component analysis (PCA) based approximation approach. The key property of this approximation is that it is determined by just a limited number of one- and two-dimensional PDEs. In Section 3.3, an efficient discretization of the one- and two-dimensional PDEs for European-style basket options is described, which employs finite differences on a nonuniform spatial grid followed by the Brian and Douglas Alternating Direction Implicit (ADI) scheme on a uniform temporal grid. In Section 3.4, a rigorous stability analysis is given for the spatial and temporal discretizations defined in Section 3.3. In Section 3.5 we study in detail the error in the discretization described in Section 3.3 for the PCA-based approximation and observe a favourable, near second-order convergence behaviour. Next, a runtime comparison is included to demonstrate the computational advantage of the PCA-based approximation approach. The final Section 3.6 presents our conclusions and outlook.

3.2 PCA-based approximation approach

3.2.1 Coordinate transformation

In this section we apply two subsequent coordinate transformations to the PDE (3.2) for a European-style basket option. We assume here that the elementary functions $\ln(\cdot)$, $\exp(\cdot)$, $\tan(\cdot)$, $\arctan(\cdot)$ are taken componentwise whenever their argument is a vector.

3.2.1.1 Transformation to a problem with coefficients proportional to eigenvalues

The covariance matrix $\boldsymbol{\Sigma} = \left(\Sigma_{ij}\right) \in \mathbb{R}^{d\times d}$ is given by $\Sigma_{ij} = \sigma_i \rho_{ij} \sigma_j$ for $i, j = 1, 2, \ldots, d$. Let $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$ denote a real diagonal matrix of eigenvalues of $\boldsymbol{\Sigma}$ and $\boldsymbol{Q}$ a real orthogonal matrix of eigenvectors of $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathsf{T}}$.

Then, following [71], we apply the coordinate transformation

$$\boldsymbol{x}(\boldsymbol{s}, t) = \boldsymbol{Q}^{\mathsf{T}} \left(\ln(\boldsymbol{s}/K) - \boldsymbol{b}(t)\right), \tag{3.5}$$

where $\boldsymbol{b}(t) = (b_1(t), b_2(t), \ldots, b_d(t))^{\mathsf{T}}$ with $b_i(t)$ for $1 \le i \le d$ to be determined.

The partial derivatives of the transformation given in (3.5) are given by

$$\begin{aligned} \frac{\partial x_i}{\partial s_j} &= q_{ji}\frac{1}{s_j} \\ \frac{\partial x_i}{\partial t} &= -\sum_{j=1}^{d} q_{ji} b_i'(t). \end{aligned} \tag{3.6}$$

where $i,j = 1,2,\ldots,d$. Let the function v be defined by

$$u(\boldsymbol{s},t) = v(\boldsymbol{x}(\boldsymbol{s},t),t).$$

Then, using the chain rule the first derivative of $u(\boldsymbol{s},t)$ to t can be written as

$$\frac{\partial u(\boldsymbol{s},t)}{\partial t} = \frac{\partial v(\boldsymbol{x},t)}{\partial t} - \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial v(\boldsymbol{x},t)}{\partial x_i} q_{ji} b_j'(t). \tag{3.7}$$

Further, the first derivative of $u(\boldsymbol{s},t)$ to s_j (with $j = 1,2,\ldots,d$) is given by

$$\frac{\partial u(\boldsymbol{s},t)}{\partial s_j} = \frac{1}{s_j}\sum_{i=1}^{d} q_{ji}\frac{\partial v(\boldsymbol{x},t)}{\partial x_i}. \tag{3.8}$$

Finally, some more calculations for the second derivative of $u(\boldsymbol{s},t)$ to s_i and s_j for $i,j = 1,2,\ldots,d$ yields

$$\frac{\partial^2 u(\boldsymbol{s},t)}{\partial s_i \partial s_j} = \begin{cases} \frac{1}{s_i}\frac{1}{s_j}\sum_{k=1}^{d}\sum_{l=1}^{d} q_{ik}q_{jl}\frac{\partial^2 v(\boldsymbol{x},t)}{\partial x_k \partial x_l}, & \text{for } i \neq j, \\ \frac{1}{s_i^2}\left(\sum_{k=1}^{d}\sum_{l=1}^{d} q_{ik}q_{il}\frac{\partial^2 v(\boldsymbol{x},t)}{\partial x_k \partial x_l} - \sum_{l=1}^{d} q_{il}\frac{\partial v(\boldsymbol{x},t)}{\partial x_l}\right) & \text{for } i = j. \end{cases} \tag{3.9}$$

An easy calculation yields that v satisfies

$$\frac{\partial v}{\partial t}(\boldsymbol{x},t) - \sum_{i,j=1}^{d} q_{ij} b_i'(t)\frac{\partial v(\boldsymbol{x},t)}{\partial x_j} = \frac{1}{2}\sum_{k=1}^{d}\lambda_k \frac{\partial^2 v}{\partial x_k^2}(\boldsymbol{x},t) + \sum_{i,j=1}^{d}\left(r - \tfrac{1}{2}\sigma_i^2\right) q_{ij}\frac{\partial v(\boldsymbol{x},t)}{\partial x_j} - r v(\boldsymbol{x},t) \tag{3.10}$$

whenever $\boldsymbol{x} \in \mathbb{R}^d$, $t \in (0,T]$. Thus, choosing $b_i(t)$ with $i = 1,2,\ldots,d$ such that

$$b_i'(t) = \tfrac{1}{2}\sigma_i^2 - r \tag{3.11}$$

leads to a pure diffusion equation for v, without mixed derivative terms, and with a simple reaction term:

$$\frac{\partial v}{\partial t}(\boldsymbol{x},t) = \frac{1}{2}\sum_{k=1}^{d}\lambda_k \frac{\partial^2 v}{\partial x_k^2}(\boldsymbol{x},t) - r v(\boldsymbol{x},t), \tag{3.12}$$

whenever $\boldsymbol{x} \in \mathbb{R}^d$, $t \in (0,T]$.

The ordinary differential equation (ODE) for (3.11) for b_i has a simple solution

$$b_i(t) = b_i(0) + (\tfrac{1}{2}\sigma_i^2 - r)t, \tag{3.13}$$

for $i = 1,2,\ldots,d$. We choose[1] $b_i(0) = 0$ for $i = 1,2,\ldots,d$, thus $\boldsymbol{b}(t)$ in (3.5) is elementwise given by

$$b_i(t) = \left(\tfrac{1}{2}\sigma_i^2 - r\right)t, \tag{3.14}$$

for $i = 1,2,\ldots,d$.

[1] We remark that it is possible to make other choices here, e.g. $b_i(0) = \ln(s_i/K)$ or $b_i(T) = 0$.

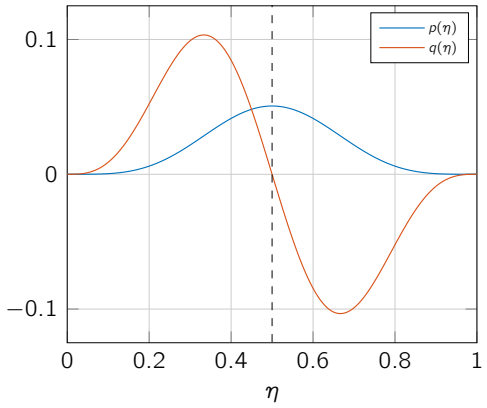


Figure 3.1: Plot of functions $p(\eta)$ and $q(\eta)$ used in (3.17).

3.2.1.2 Additional transformation to a unit-cube

Following [71], we apply a second coordinate transformation, which maps the spatial domain $\mathbb{R}^d$ onto the d-dimensional open unit cube $D = (0,\ 1)^d$,

$$\boldsymbol{y}(\boldsymbol{x}) = \tfrac{1}{\pi}\arctan(\boldsymbol{x}) + \tfrac{1}{2}. \tag{3.15}$$

The partial derivative of the transformation given in (3.15) is given by

$$\frac{\partial y_i}{\partial x_i} = \frac{1}{\pi}\frac{1}{x_i^2+1} \tag{3.16}$$

where $i = 1, 2, \ldots, d$. Let the function w be defined by

$$v(\boldsymbol{x}, t) = w(\boldsymbol{y}(\boldsymbol{x}), t).$$

Then it is readily seen that

$$\frac{\partial w}{\partial t}(\boldsymbol{y}, t) = \sum_{k=1}^{d} \lambda_k \left[p(y_k)\frac{\partial^2 w}{\partial y_k^2}(\boldsymbol{y}, t) + q(y_k)\frac{\partial w}{\partial y_k}(\boldsymbol{y}, t)\right] - r w(\boldsymbol{y}, t) \tag{3.17}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$, $t \in (0,\ T]$ with

$$p(\eta) = \frac{1}{2\pi^2}\sin^4(\pi\eta), \quad q(\eta) = \frac{1}{\pi}\sin^3(\pi\eta)\cos(\pi\eta) \quad \text{for } \eta \in \mathbb{R}.$$

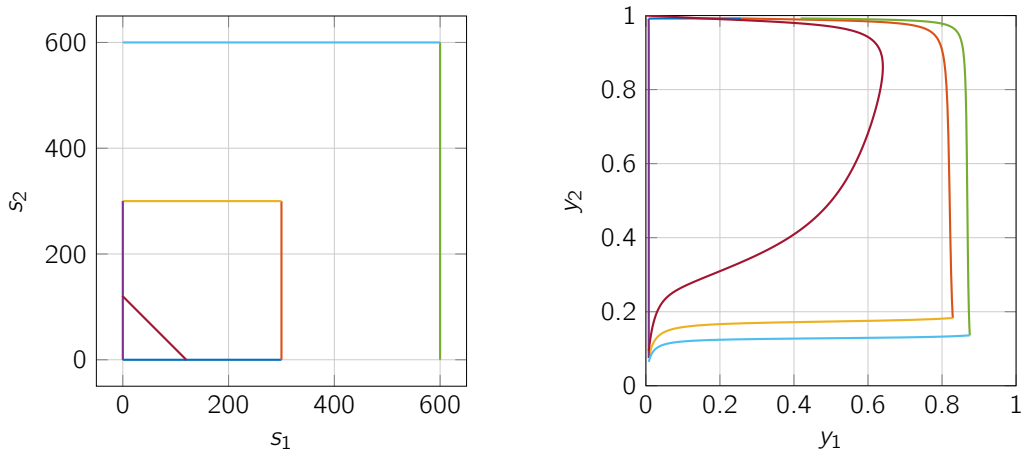
These functions are plotted in Figure 3.1. The PDE (3.17) is a convection-diffusion-reaction equation without mixed derivatives. Let ψ denote the transform of the payoff function ϕ,

$$\psi(\boldsymbol{y}, t) = \phi(K\exp[\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}(t)]) \quad \text{with} \quad \boldsymbol{x} = \tan\left[\pi(\boldsymbol{y} - \tfrac{1}{2})\right] \tag{3.18}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$, $t \in [0, T]$.

Then for (3.17) one has the initial condition

$$w(\boldsymbol{y}, 0) = \psi(\boldsymbol{y}, 0). \tag{3.19}$$



(a) s coordinates

(b) y coordinates

Figure 3.2: Visualization of a rectangular domain in s-coordinates transformed to y-coordinates using the coordinate transformation presented in Section 3.2.1.

At the boundary ∂D of the spatial domain $D = (0,1)^d$ we shall consider a Dirichlet condition. Therefore we make a minor assumption that each column of the matrix $\boldsymbol{Q}$ satisfies one of the following two conditions:

(a) All its entries are strictly positive;

(b) It has both a strictly positive and a strictly negative entry.

For any given $k \in \{1, 2, \ldots, d\}$ such that the k-th column of $\boldsymbol{Q}$ satisfies condition (a) there holds

$$w(\boldsymbol{y}, t) = K e^{-rt} \tag{3.20}$$

whenever $\boldsymbol{y} \in \partial D$ with $y_k = 0$ and $t \in (0,\ T]$. On the complementary part of ∂D a homogeneous Dirichlet condition is valid.

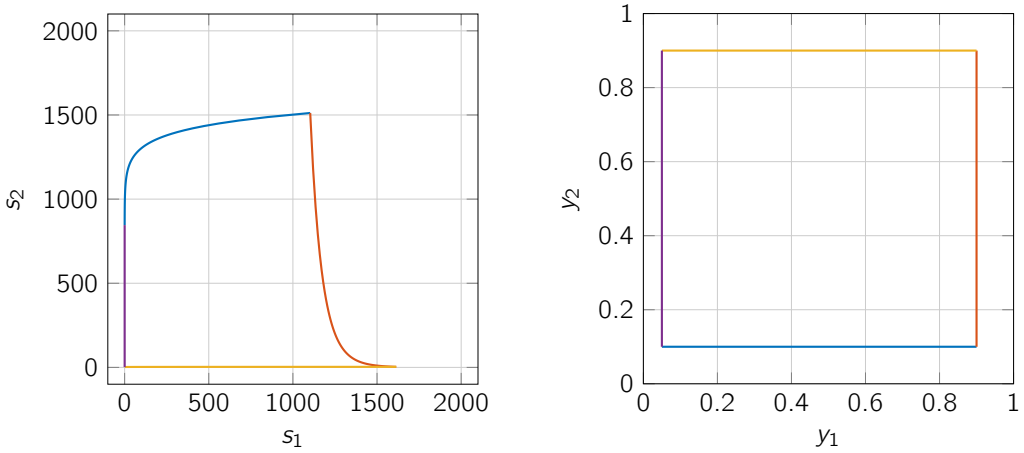
3.2.1.3 Proof for Dirichlet boundary condition for (3.17)

A short proof of the result given in this section can also be found in [41]. Consider the following minor assumption on the matrix $\boldsymbol{Q}$ of eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$.

Assumption 1. *Each column of $\boldsymbol{Q}$ satisfies one of the following two conditions:*

1. *all its entries are strictly positive;*
2. *it has both a strictly positive and a strictly negative entry.*

Then we have the following result, formulated as Lemma 1:



(a) s coordinates

(b) y coordinates

Figure 3.3: Visualization of a rectangular domain in y-coordinates transformed to s-coordinates using the coordinate transformation presented in Section 3.2.1.

Lemma 1. *Let the function ψ be given by (3.18) with ϕ defined by (3.4). Let $k \in \{1, 2, \ldots, d\}$, $t \in [0,\ T]$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_d)^{\mathsf{T}}$ with fixed $y_j \in (0,\ 1)$ whenever $j \neq k$.*

If the k-th column of $\boldsymbol{Q}$ satisfies condition 1, then $\psi(\boldsymbol{y}, t) \to K$ as $y_k \downarrow 0$.

If the k-th column of $\boldsymbol{Q}$ satisfies condition 2, then $\psi(\boldsymbol{y}, t) \to 0$ as $y_k \downarrow 0$.

Finally, $\psi(\boldsymbol{y}, t) \to 0$ as $y_k \uparrow 1$.

Proof. Let $\boldsymbol{x} = \tan\left[\pi(\boldsymbol{y} - \frac{1}{2})\right]$ and $\boldsymbol{s} = K \exp\left[\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}(t)\right]$, so that $\psi(\boldsymbol{y}, t) = \phi(\boldsymbol{s})$.

Suppose first $y_k \downarrow 0$. Then $x_k \to -\infty$.
If the k-th column of $\boldsymbol{Q}$ satisfies condition 1, then all entries of $\boldsymbol{Q}\boldsymbol{x}$ tend to $-\infty$. Consequently, all entries of $\boldsymbol{s}$ tend to zero and thus $\phi(\boldsymbol{s}) \to K$.
If the k-th column of $\boldsymbol{Q}$ satisfies condition 2, then the entries of $\boldsymbol{Q}\boldsymbol{x}$ go to either $-\infty$ or $+\infty$ with at least one entry that tends to $+\infty$. It follows that the entries of $\boldsymbol{s}$ go to either zero or $+\infty$ with at least one entry that tends to $+\infty$, and therefore $\phi(\boldsymbol{s}) \to 0$.

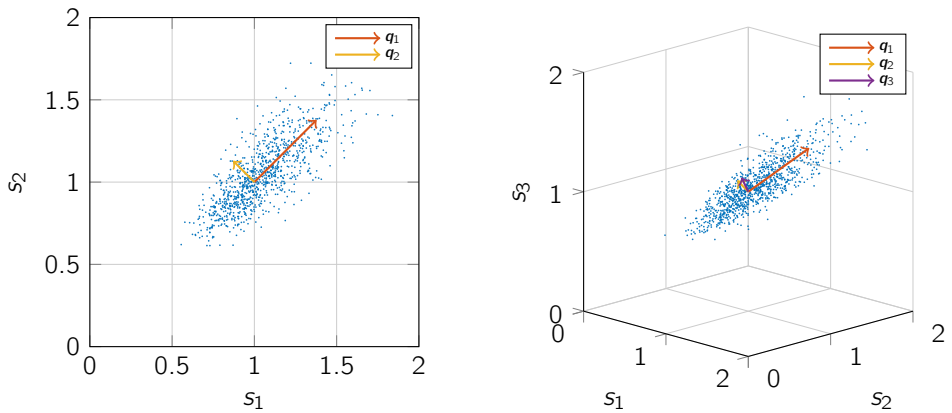
Suppose next $y_k \uparrow 1$. Then $x_k \to +\infty$ and the entries of $\boldsymbol{Q}\boldsymbol{x}$ go to either $+\infty$ or $-\infty$ with at least one entry that tends to $+\infty$. Hence, $\phi(\boldsymbol{s}) \to 0$. □

For any given $k \in \{1, 2, \ldots, d\}$ the diffusion and convection coefficients $p(y_k)$ and $q(y_k)$ in (3.17) vanish as $y_k \downarrow 0$ or $y_k \uparrow 1$. Accordingly, (3.17) is also satisfied on each boundary part

$$\left\{\boldsymbol{y} : \boldsymbol{y} = (y_1, y_2, \ldots, y_d)^{\mathsf{T}} \text{ with } y_k = \delta \text{ and } y_j \in (0,\ 1) \text{ whenever } j \neq k\right\}$$

for $\delta \in \{0, 1\}$.

Also the initial condition (3.19) hold on each such boundary part, upon taking the relevant limit value for $\psi(\boldsymbol{y}, t)$ given by Lemma 1. On each part where this limit value equals K,



(a) PCA (two dimensions) (b) PCA (three dimensions)

Figure 3.4: Visualization of multiple simulations for two (left) and three (right) correlated assets values under the Black–Scholes model. Further the principal components (or scaled eigenvectors of covariance matrix $\boldsymbol{\Sigma}$) are shown. In this example there is clearly one dominant eigenvector/eigenvalue $\boldsymbol{q}_1$.

the solution (3.20) is obtained, and on each part where the limit value equals zero, the zero solution holds. This yields the Dirichlet boundary condition for the PDE (3.17) stated before.

3.2.2 PCA-based approximation approach for European basket option

Assume the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ are ordered such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$. In many financial applications it holds that λ_1 is dominant, that is, λ_1 is much larger than λ_2.

In view of this observation, Reisinger and Wittum [71] introduced a PCA-based approximation approach of the exact solution w to the d-dimensional PDE (3.17). To this purpose, regard w also as a function of the eigenvalues and write $w(\boldsymbol{y}, t;\, \boldsymbol{\lambda})$ with $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_d)^{\mathsf{T}}$. Let

$$\widehat{\boldsymbol{\lambda}} = (\lambda_1, 0, \ldots, 0)^{\mathsf{T}}$$
$$\delta\boldsymbol{\lambda} = \boldsymbol{\lambda} - \widehat{\boldsymbol{\lambda}} = (0, \lambda_2, \ldots, \lambda_d)^{\mathsf{T}}.$$

Under sufficient smoothness, a first-order Taylor expansion of w at $\widehat{\boldsymbol{\lambda}}$ yields

$$w(\boldsymbol{y}, t;\, \boldsymbol{\lambda}) \approx w(\boldsymbol{y}, t;\, \widehat{\boldsymbol{\lambda}}) + \sum_{l=2}^{d} \delta\lambda_l \frac{\partial w}{\partial \lambda_l}(\boldsymbol{y}, t;\, \widehat{\boldsymbol{\lambda}}). \tag{3.21}$$

The partial derivative $\partial w / \partial \lambda_l$ (for $2 \leq l \leq d$) can be approximated by a forward finite difference,

$$\frac{\partial w}{\partial \lambda_l}(\boldsymbol{y}, t;\, \widehat{\boldsymbol{\lambda}}) \approx \frac{w(\boldsymbol{y}, t;\, \widehat{\boldsymbol{\lambda}} + \delta\lambda_l \boldsymbol{e}_l) - w(\boldsymbol{y}, t;\, \widehat{\boldsymbol{\lambda}})}{\delta\lambda_l}, \tag{3.22}$$

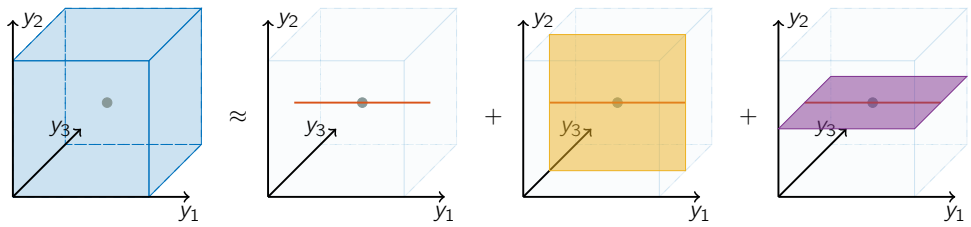


Figure 3.5: A visualization of the domains for the PDEs in the PCA-based approximation.

where $\boldsymbol{e}_l$ denotes the l-th standard basis vector in $\mathbb{R}^d$. From (3.21) and (3.22), it follows that

$$w(\boldsymbol{y}, t;\ \boldsymbol{\lambda}) \approx w(\boldsymbol{y}, t;\ \widehat{\boldsymbol{\lambda}}) + \sum_{l=2}^{d} \left[w(\boldsymbol{y}, t;\ \widehat{\boldsymbol{\lambda}} + \delta\lambda_l \boldsymbol{e}_l) - w(\boldsymbol{y}, t;\ \widehat{\boldsymbol{\lambda}}) \right].$$

Write

$$w^{(1)}(\boldsymbol{y}, t) = w(\boldsymbol{y}, t;\ \widehat{\boldsymbol{\lambda}})$$
$$w^{(1,l)}(\boldsymbol{y}, t) = w(\boldsymbol{y}, t;\ \widehat{\boldsymbol{\lambda}} + \delta\lambda_l \boldsymbol{e}_l).$$

Then the *PCA-based approximation* reads

$$w(\boldsymbol{y}, t) \approx \widetilde{w}(\boldsymbol{y}, t) = w^{(1)}(\boldsymbol{y}, t) + \sum_{l=2}^{d} \left[w^{(1,l)}(\boldsymbol{y}, t) - w^{(1)}(\boldsymbol{y}, t) \right] \tag{3.23}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$ and $t \in (0,\ T]$.

By construction, $w^{(1)}$ satisfies the PDE (3.17) with λ_k being set to zero for all $k \neq 1$, and $w^{(1,l)}$ satisfies (3.17) with λ_k being set to zero for all $k \notin \{1, l\}$. Thus $w^{(1)}$ and $w^{(1,l)}$ satisfy essentially a one- and two-dimensional PDE. The PDE (3.17) for $w^{(1)}$ and $w^{(1,l)}$ is completed by the same initial and boundary conditions as for w, given above. We write

$$\widetilde{u}(\boldsymbol{s}, t) = \widetilde{w}(\boldsymbol{y}(\boldsymbol{x}(\boldsymbol{s}, t)), t)$$

for the PCA-based approximation in the original coordinates.

As an example, a three dimensional visualization of the domains on which these essentially low-dimensional PDEs for $w^{(1)}$ and $w^{(1,l)}$ (with $l = 2, 3$) need to be solved, as used in the PCA-based approximation, is shown in Figure 3.5.

In financial practice, one is often interested in the option value at inception in the single point $\boldsymbol{s} = \boldsymbol{S_0}$, where $\boldsymbol{S_0} = (S_0^1, S_0^2, \ldots, S_0^d)^\mathsf{T}$ is the vector of initial (spot) asset prices. Let

$$\boldsymbol{Y_0} = \boldsymbol{y}(\boldsymbol{x}(\boldsymbol{S_0}, T)) \in (0,\ 1)^d$$

denote the corresponding point in the y-domain with elements $\boldsymbol{Y_0} = (Y_0^1, Y_0^2, \ldots, Y_0^d)^\mathsf{T}$.

Then $w^{(1)}(\boldsymbol{Y_0}, T)$ can be acquired by solving a one-dimensional PDE on the line segment L_1 in the y-domain that is parallel to the y_1-axis and passes through $y = \boldsymbol{Y_0}$. Hence, y_k can be fixed at the value Y_0^k whenever $k \neq 1$.

Next, $w^{(1,l)}(\boldsymbol{Y_0}, T)$, for $2 \leq l \leq d$, can be acquired by solving a two-dimensional PDE on the plane segment P_l in the y-domain that is parallel to the $(y_1,\ y_l)$-plane and passes through $y = \boldsymbol{Y_0}$. Hence, in this case, y_k can be fixed at the value Y_0^k whenever $k \notin \{1, l\}$.

Determining the PCA-based approximation $\widetilde{w}(\boldsymbol{Y_0}, T) = \widetilde{u}(\boldsymbol{S_0}, T)$ thus requires solving just 1 one-dimensional PDE and $d-1$ two-dimensional PDEs. This clearly constitutes a major computational advantage, compared to solving the full d-dimensional PDE at once whenever d is medium or large. Notice further that the different terms in the approximation (3.23) can be computed in parallel independently of each other. Then the total computational time equals approximately that of solving just 1 two-dimensional PDE.

We remark that instead of (3.21) also higher-order Taylor expansions of w at $\widehat{\boldsymbol{\lambda}}$ can be used to derive higher-order PCA-based approximations for the fair value of an option. This can reduce the error made in the PCA-based approximation, but comes also with an additional cost of solving also higher-dimensional PDEs.

A rigorous error analysis of the PCA-based approximation relevant to European-style basket options has been given by Reisinger and Wissmann [69]. In particular, under mild assumptions, these authors showed that $w - \widetilde{w} = \mathcal{O}\left(\lambda_2^2\right)$ in the maximum norm.

3.3 Discretization

To arrive at the values $w^{(1)}(\boldsymbol{Y_0}, T)$ and $w^{(1,l)}(\boldsymbol{Y_0}, T)$ (for $2 \leq l \leq d$) in the approximation $\widetilde{w}(\boldsymbol{Y_0}, T)$ of $w(\boldsymbol{Y_0}, T)$ we perform a finite difference discretization of the pertinent one- and two-dimensional PDEs on a (Cartesian) nonuniform spatial grid, followed by a suitable implicit time discretization.

3.3.1 Spatial discretization

Let $\kappa_0 = \frac{1}{2}$ and $\kappa_1 > 0$. Note that with the choice of $b(t)$ in (3.14) the point $(\kappa_0, \kappa_0, \ldots, \kappa_0)^\mathsf{T}$ in the y-domain corresponds to the point $(K, K, \ldots, K)^\mathsf{T}$ in the s-domain if $t = 0$.

For any given $k \in \{1, 2, \ldots, d\}$ a nonuniform mesh $0 = y_{k,0} < y_{k,1} < \ldots < y_{k,m+1} = 1$ in the k-th spatial direction, with m mesh points in the interior of the domain, is defined by (see e.g. [40])

$$y_{k,i} = \varphi(\xi_i) \quad \text{with } \xi_i = \xi_{\min} + i\Delta\xi, \quad \Delta\xi = \frac{\xi_{\max} - \xi_{\min}}{m+1} \quad (\text{for } i = 0, 1, \ldots, m+1),$$

with

$$\varphi(\xi) = \kappa_0 + \kappa_1 \sinh(\xi) \quad (\text{for } \xi_{\min} \leq \xi \leq \xi_{\max})$$

and

$$\xi_{\min} = -\sinh^{-1}(\kappa_0/\kappa_1)$$
$$\xi_{\max} = \sinh^{-1}((1-\kappa_0)/\kappa_1).$$

Remark that $\xi_{\max} = -\xi_{\min}$ since $\kappa_0 = \frac{1}{2}$. The parameter κ_1 controls the fraction of mesh points that lie in the neighborhood of κ_0. We make the heuristic choice $\kappa_1 = \frac{1}{40}$.

The above mesh is smooth in the sense that there exist constants $C_0, C_1, C_2 > 0$ (independent of i and m) such that the mesh widths $\Delta y_{k,i} = y_{k,i} - y_{k,i-1}$ satisfy

$$C_0\,\Delta\xi \le \Delta y_{k,i} \le C_1\,\Delta\xi \quad \text{and} \quad |\Delta y_{k,i+1} - \Delta y_{k,i}| \le C_2\,(\Delta\xi)^2 .$$

The spatial derivatives in (3.17) are discretized using central finite difference schemes. Let $f : \mathbb{R} \to \mathbb{R}$ be any given smooth function, let $\cdots < \eta_{i-1} < \eta_i < \eta_{i+1} < \cdots$ be any given smooth mesh and denote the mesh widths by $h_i = \eta_i - \eta_{i-1}$. Then second-order approximations to the first and second derivatives are given by

$$f'(\eta_i) \approx \beta_{i,-1}\,f(\eta_{i-1}) + \beta_{i,0}\,f(\eta_i) + \beta_{i,1}\,f(\eta_{i+1}),$$

$$f''(\eta_i) \approx \gamma_{i,-1}\,f(\eta_{i-1}) + \gamma_{i,0}\,f(\eta_i) + \gamma_{i,1}\,f(\eta_{i+1}),$$

with

$$\beta_{i,-1} = \frac{-h_{i+1}}{h_i(h_i+h_{i+1})}\ ,\ \beta_{i,0} = \frac{h_{i+1}-h_i}{h_i h_{i+1}}\ ,\ \beta_{i,1} = \frac{h_i}{h_{i+1}(h_i+h_{i+1})}\ ,$$

and

$$\gamma_{i,-1} = \frac{2}{h_i(h_i+h_{i+1})}\ ,\ \gamma_{i,0} = \frac{-2}{h_i h_{i+1}}\ ,\ \gamma_{i,1} = \frac{2}{h_{i+1}(h_i+h_{i+1})}\ .$$

The above two finite difference formulas are applied with $\eta_i = y_{k,i}$ for $1 \le i \le m$ and $1 \le k \le d$.

Semidiscretization of the PDE for $w^{(1,l)}$ on the plane segment P_l leads to a system of ordinary differential equations (ODEs) of the form

$$\boldsymbol{w}'(t) = (\boldsymbol{A_1} + \boldsymbol{A_l})\,\boldsymbol{w}(t) + \boldsymbol{g}(t) \tag{3.24}$$

for $t \in (0,\ T]$. Here $\boldsymbol{w}(t)$ is the vectorization of $w^{(1,l)}(t)$ on the plane segment P_l. So, $\boldsymbol{w}(t)$ is a vector of dimension m^2 and $\boldsymbol{A_1}$, $\boldsymbol{A_l}$ are given $m^2 \times m^2$ matrices that are tridiagonal (possibly up to permutation), commute and correspond to, respectively, the first and the l-th spatial direction. Further, $\boldsymbol{g}(t) = \boldsymbol{g_1}(t) + \boldsymbol{g_l}(t)$ is a given vector of dimension m^2, which is obtained from the Dirichlet boundary condition stated at the end of Section 3.2.1. The ODE system (3.24) is completed by an initial condition

$$\boldsymbol{w}(0) = \boldsymbol{w_0}$$

where the vector $\boldsymbol{w_0}$ is determined by the function $\psi(\cdot, 0)$ on P_l with the function ψ defined by (3.18).

The payoff function ϕ given by (3.4) is continuous but not everywhere differentiable, and hence, this also holds for the function ψ given by (3.18). It is well-known that the non-smoothness of the payoff function can have an adverse impact on the convergence behaviour of the spatial discretization. To alleviate this, we employ cell averaging near the points of nonsmoothness in defining the initial vector $\boldsymbol{w_0}$, see e.g. [40, 50, 66].

3.3.2 Temporal discretization

For the temporal discretization of the ODE system (3.24), a common Alternating Direction Implicit (ADI) method is used. Let a step size $\Delta t = T/N$ with integer $N \geq 1$ be given and define temporal grid points $t_n = n\Delta t$ for $n = 0, 1, \ldots, N$. Then the familiar second-order Brian and Douglas ADI scheme for two-dimensional PDEs yields approximations $\boldsymbol{w_n} \approx \boldsymbol{w}(t_n)$ that are successively defined for $n = 1, 2, \ldots, N$ by

$$\begin{cases} \boldsymbol{z_0} = \boldsymbol{w}_{n-1} + \Delta t\,(\boldsymbol{A_1} + \boldsymbol{A_I})\,\boldsymbol{w}_{n-1} + \Delta t\,\boldsymbol{g}(t_{n-1}), \\ \boldsymbol{z_1} = \boldsymbol{z_0} + \frac{1}{2}\Delta t\boldsymbol{A_1}(\boldsymbol{z_1} - \boldsymbol{w}_{n-1}) + \frac{1}{2}\Delta t(\boldsymbol{g_1}(t_n) - \boldsymbol{g_1}(t_{n-1})), \\ \boldsymbol{z_2} = \boldsymbol{z_1} + \frac{1}{2}\Delta t\boldsymbol{A_I}(\boldsymbol{z_2} - \boldsymbol{w}_{n-1}) + \frac{1}{2}\Delta t(\boldsymbol{g_I}(t_n) - \boldsymbol{g_I}(t_{n-1})), \\ \boldsymbol{w_n} = \boldsymbol{z_2}. \end{cases} \tag{3.25}$$

In the scheme (3.25) a forward Euler predictor stage is followed by two implicit but unidirectional corrector stages, which serve to stabilize the predictor stage. The two linear systems in each time step can be solved very efficiently by employing a priori LU factorizations of the pertinent two matrices. The number of floating-point operations per time step is then directly proportional to the number of spatial grid points, i.e. m^2, which is optimal.

As for the spatial discretization, also the convergence of the temporal discretization can be adversely affected by the nonsmooth payoff function. To alleviate this, we apply backward Euler damping at the initial time, also known as Rannacher time stepping, that is, the first time step is replaced by two half steps with the backward Euler method, see, e.g., [40, 67].

Discretization of the PDE for $w^{(1)}$ on the line segment L_1 is done analogously to the above. Then a semidiscrete system

$$\boldsymbol{w}'(t) = \boldsymbol{A_1}\,\boldsymbol{w}(t) + \boldsymbol{g_1}(t) \tag{3.26}$$

is obtained with $\boldsymbol{w}(t)$ and $\boldsymbol{g_1}(t)$ vectors of dimension m and $\boldsymbol{A_1}$ is an $m \times m$ tridiagonal matrix. Temporal discretization is performed by the Crank–Nicolson scheme with backward Euler damping. Recall that the Crank–Nicolson scheme can be regarded as a special case of the Brian and Douglas scheme, which is seen upon setting $\boldsymbol{A_I}$ and $\boldsymbol{g_I}$ both equal to zero in (3.25).

3.4 Stability analysis

The favourable rigorous stability results for the spatial and temporal discretizations as discussed in this section has been proven in [41].

In this section stability results are presented for the spatial and temporal discretizations given in Section 3.3. To this purpose we employ the logarithmic matrix norm. This is defined, for any given square matrix $\boldsymbol{A}$, by the limit

$$\mu[\boldsymbol{A}] = \lim_{t \downarrow 0} \frac{\|\boldsymbol{I} + t\boldsymbol{A}\| - 1}{t}, \tag{3.27}$$

where $\|\cdot\|$ denotes any given matrix norm that is induced by a vector norm $|\cdot|$ and $\boldsymbol{I}$ is the identity matrix. The next theorem provides a key property of the logarithmic norm:

Theorem 2 (Key property of logarithmic norm, see e.g. [36]). *Let $\omega \in \mathbb{R}$. Then*

$$\mu[\boldsymbol{A}] \le \omega \quad \Longleftrightarrow \quad \|e^{t\boldsymbol{A}}\| \le e^{t\omega}$$

for all $t \ge 0$.

By virtue of Theorem 2, a linear semidiscrete system of ODEs with matrix $\boldsymbol{A}$ is *stable* in the norm $|\cdot|$ whenever $\mu[\boldsymbol{A}]$ can be bounded by a moderate constant ω uniformly in the spatial mesh.

Write $\eta_i = y_{k,i}$ and $h_i = \eta_i - \eta_{i-1}$ as before. Let $H_i = h_i + h_{i+1}$ and consider the $m \times m$ diagonal matrix $\boldsymbol{H}$ given by

$$\boldsymbol{H} = \tfrac{1}{2}\mathrm{diag}\,(H_1, H_2, \ldots, H_m)\,.$$

For vectors $\boldsymbol{v}$ of dimension m^k we define $|\boldsymbol{v}|_{\boldsymbol{H}} = |\boldsymbol{H}^{1/2}\boldsymbol{v}|_2$ (if $k=1$) and $|\boldsymbol{v}|_{\boldsymbol{H}} = |(\boldsymbol{H}\otimes\boldsymbol{H})^{1/2}\boldsymbol{v}|_2$ (if $k=2$), where $\otimes$ is the Kronecker product. Thus $|\cdot|_{\boldsymbol{H}}$ constitutes a naturally scaled Euclidean vector norm. For $m^k \times m^k$ matrices $\boldsymbol{A}$, let the induced matrix norm and logarithmic matrix norm be denoted by $\|\boldsymbol{A}\|_{\boldsymbol{H}}$ and $\mu_{\boldsymbol{H}}[\boldsymbol{A}]$, respectively ($k = 1, 2$).

The following theorem is a direct consequence of [90, Theorems 3 and 4], which generalize two results from [43]. It yields that the semidiscrete systems derived in Section 3.3 are stable in $|\cdot|_{\boldsymbol{H}}$.

Theorem 3. *Let $\kappa_0 = \frac{1}{2}$ and $\kappa_1 > 0$. Then there exists a constant $\omega > 0$ (independent of $m \ge 1$, $\lambda_1, \lambda_2, \ldots, \lambda_d \ge 0$ and $r \ge 0$) such that*

$$\mu_{\boldsymbol{H}}[\boldsymbol{A_1}] \le \lambda_1 \omega$$

for (3.26) *and*

$$\mu_{\boldsymbol{H}}[\boldsymbol{A_1} + \boldsymbol{A_l}] \le (\lambda_1 + \lambda_l)\omega$$

for (3.24).

For any given $\kappa_1 > 0$, it is readily seen using [90] that constant $\omega < 4$, which is indeed moderate. We next consider the stability of the temporal discretizations of the semidiscrete systems from Section 3.3. Let

$$R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \quad (z \in \mathbb{C}).$$

Then the stability matrices for the Crank–Nicolson discretization of (3.26) and the Brian and Douglas discretization of (3.24) are given by $\boldsymbol{B_1}$ and $\boldsymbol{B}$, respectively, with

$$\boldsymbol{B_1} = R(\Delta t \boldsymbol{A_1}) \quad \text{and} \quad \boldsymbol{B} = R(\Delta t \boldsymbol{A_1})\, R(\Delta t \boldsymbol{A_l})\,,$$

where for (3.24) it has been used that $\boldsymbol{A_1}$ and $\boldsymbol{A_l}$ commute. Stability of the temporal discretizations concerns power boundedness of $\boldsymbol{B_1}$ and $\boldsymbol{B}$ with constants uniformly in the spatial mesh and the time step. We have the following positive result.

Theorem 4. *Let $\kappa_0 = \frac{1}{2}$, $\kappa_1 > 0$ and let constant $\omega > 0$ be given by Theorem 3. Then*

$$\|\boldsymbol{B}_1^n\|_{\boldsymbol{H}} \le e^{2\lambda_1 \omega T}$$
$$\|\boldsymbol{B}^n\|_{\boldsymbol{H}} \le e^{2(\lambda_1+\lambda_l)\omega T}$$

whenever $n \ge 0$, $\Delta t > 0$, $0 \le t_n \le T$ and, respectively, $\Delta t \lambda_1 \omega \le 1$ and $\Delta t(\lambda_1 + \lambda_l)\omega \le 1$.

Proof. For (3.26) there holds $\mu_{\boldsymbol{H}}[\Delta t \boldsymbol{A}_1] \le \Delta t \lambda_1 \omega$. By applying a well-known result[2] due to von Neumann, see e.g. [36, Theorem I.2.11], we obtain

$$\|\boldsymbol{B}_1\|_{\boldsymbol{H}} = \|R(\Delta t \boldsymbol{A}_1)\|_{\boldsymbol{H}} \le R(\Delta t \lambda_1 \omega)$$

whenever $1 - \frac{1}{2}\Delta t \lambda_1 \omega > 0$. It is easily verified that

$$R(\zeta) \le 1 + 2\zeta \quad \text{whenever } \zeta \in \mathbb{R},\ 0 \le \zeta \le 1.$$

Hence, if $n \ge 0$, $\Delta t > 0$, $0 \le t_n \le T$ and $\Delta t \lambda_1 \omega \le 1$, then

$$\|\boldsymbol{B}_1^n\|_{\boldsymbol{H}} \le (1 + 2\Delta t \lambda_1 \omega)^n \le e^{2\lambda_1 \omega T}.$$

Next, for (3.24) there holds $\mu_{\boldsymbol{H}}[\Delta t \boldsymbol{A}_k] \le \Delta t \lambda_k \omega$ $(k = 1, l)$ and the bound on $\|\boldsymbol{B}^n\|_{\boldsymbol{H}}$ follows completely analogously. □

In view of Theorem 4, the temporal discretizations from Section 3.3 are stable in $|\cdot|_{\boldsymbol{H}}$ under a minor condition on the time step, which is independent of the spatial mesh.

Our final stability result deals with the maximum norm. It is first shown that the sequence of spatial mesh widths in the interval [0, 1] is monotonically decreasing up to the midpoint $\frac{1}{2}$ (and, by symmetry, monotonically increasing beyond this point).

Lemma 2. *Let $\kappa_0 = \frac{1}{2}$, $\kappa_1 > 0$. Then $h_{i+1} \le h_i$ whenever $\eta_i \le \frac{1}{2}$.*

Proof. For each given i, there holds $h_i = \varphi(\xi_i) - \varphi(\xi_{i-1}) = \varphi'(\varepsilon_i)\Delta\xi$ with certain $\varepsilon_i \in (\xi_{i-1}, \xi_i)$. Since the function φ' is monotonically decreasing on $(-\infty, 0]$, it follows that $h_{i+1} \le h_i$ whenever $\eta_{i+1} \le \frac{1}{2} = \varphi(0)$. In the remaining case, where $\eta_i \le \frac{1}{2} < \eta_{i+1}$, one readily obtains $h_{i+1} \le h_i$ by using symmetry of the mesh about the point $\frac{1}{2}$. □

The following theorem reveals the favourable result that the semidiscrete systems from Section 3.3 are contractive in the maximum norm $|\cdot|_\infty$.

Theorem 5. *Let $\kappa_0 = \frac{1}{2}$, $\kappa_1 > 0$. Then $\mu_\infty[\boldsymbol{A}_1] \le 0$ for (3.26) and $\mu_\infty[\boldsymbol{A}_1 + \boldsymbol{A}_l] \le 0$ for (3.24).*

[2] This is to be distinguished from the von Neumann stability analysis that is relevant only to normal matrices.

Proof. For any square matrix $\boldsymbol{A} = (a_{ij})$ it holds that $\mu_\infty[\boldsymbol{A}] = \max_i \left(a_{ii} + \sum_{j \neq i} |a_{ij}|\right)$, see e.g. [36]. For (3.26) we have $\boldsymbol{A} = \boldsymbol{A_1}$ with

$$\begin{aligned} a_{i,i-1} &= \lambda_1 \frac{2p(\eta_i) - h_{i+1} q(\eta_i)}{h_i H_i}, \\ a_{i,i} &= \lambda_1 \frac{-2p(\eta_i) + (h_{i+1} - h_i) q(\eta_i)}{h_i h_{i+1}} - r, \\ a_{i,i+1} &= \lambda_1 \frac{2p(\eta_i) + h_i q(\eta_i)}{h_{i+1} H_i} \end{aligned}$$

and $a_{ij} = 0$ whenever $|i-j| \geq 2$.

We prove that the off-diagonal entries of $\boldsymbol{A}$ are all nonnegative. It is directly seen, using the definitions of p and q, that $a_{i,i-1} \geq 0$ and $a_{i,i+1} \geq 0$ if and only if the following conditions hold

$$\begin{aligned} \pi h_{i+1} &\leq \tan(\pi \eta_i) \qquad (\text{whenever } 0 < \eta_i < \tfrac{1}{2}) \\ \pi h_i &\leq -\tan(\pi \eta_i) \quad (\text{whenever } \tfrac{1}{2} < \eta_i < 1). \end{aligned}$$

Since $\tan(\zeta) \geq \zeta$ (for $0 \leq \zeta < \frac{\pi}{2}$) and $\tan(\zeta) \leq \zeta - \pi$ (for $\frac{\pi}{2} < \zeta \leq \pi$), the above conditions are satisfied if

$$\begin{aligned} h_{i+1} &\leq \eta_i \qquad (\text{whenever } 0 < \eta_i < \tfrac{1}{2}) \\ h_i &\leq 1 - \eta_i \quad (\text{whenever } \tfrac{1}{2} < \eta_i < 1). \end{aligned}$$

Applying Lemma 2 yields $h_{i+1} \leq h_i \leq \eta_i$ whenever $0 < \eta_i < \frac{1}{2}$. Next, by symmetry of the mesh, it also holds that $h_i \leq 1 - \eta_i$ whenever $\frac{1}{2} < \eta_i < 1$. Hence, all off-diagonal entries of the matrix $\boldsymbol{A_1}$ are nonnegative and we arrive, by employing the above formula for the logarithmic maximum norm, at $\mu_\infty[\boldsymbol{A_1}] = -r \leq 0$.

For (3.24), the result follows completely analogously, using the subadditivity of the logarithmic norm, that is, $\mu_\infty[\boldsymbol{A_1} + \boldsymbol{A_l}] \leq \mu_\infty[\boldsymbol{A_1}] + \mu_\infty[\boldsymbol{A_l}]$. □

3.5 Numerical experiments

3.5.1 Discretization error of PCA-based approximation approach

In this section we investigate by ample numerical experiments the error of the discretization described in Section 3.3 of the PCA-based approximation $\widetilde{u}(\boldsymbol{S_0}, T)$ defined in Section 3.2.2. We consider the six parameter sets for the basket option and the underlying asset price model as given in Appendix A.

We consider a European-style basket option and study the absolute error in the discretization of $\widetilde{u}(\boldsymbol{S_0}, T)$ at the point $\boldsymbol{S_0} = (K, K, \ldots, K)^\mathsf{T}$.

Table 3.1 displays our reference values for the PCA-based approximation $\widetilde{u}(\boldsymbol{S_0}, T)$ for the European-style basket put option. These values have been obtained by applying the PDE discretization from Section 3.3 with $m = N = 1000$ spatial and temporal grid points.

Set	$\widetilde{u}(\boldsymbol{S_0}, T)$
A	0.17577
B	0.83257
C	0.77065
D	9.46550
E	9.10039
F	8.76358

Table 3.1: Reference values $\widetilde{u}(\boldsymbol{S_0}, T)$ for European-style basket put options.

In the case of Set A, Reisinger & Wittum [71] obtain the approximation $w(\boldsymbol{Y_0}, T) \approx 0.1759$ for the European-style basket option. So this result from the literature agrees well with our numerical value for that set.

We next study, for the European-style basket put options for Sets A–F, the absolute error in the discretization described in Section 3.3 of the PCA-based approximation $\widetilde{u}(\boldsymbol{S_0}, T)$ in function of $m = N = 10, 11, 12, \ldots, 100$. To determine the error of the discretization for the PCA-based approximation, the reference values from Table 3.1 are used.

Figure 3.6 displays for Sets A–F the absolute error in the discretization of $\widetilde{u}(\boldsymbol{S_0}, T)$ versus $1/m$. As the main observation, Figure 3.6 clearly indicate (near) second-order convergence of the discretization error in all six cases. This is a very favourable result. Additional experiments indicate that the error stems essentially from the spatial discretization (and not the temporal discretization).

For Sets A and D, the error drop in the (less important) region $m \leq 20$ is somewhat surprising, but it is easily explained from a change of sign in the error. Except for this the error behaviour is always found to be regular and second-order.

3.5.2 Runtime comparison with respect to full grid discretization

Similar to Example 2.2.2, we can compute the runtime needed for the PCA-based approximation approach applied to Set A. The different measured runtimes are shown in Figure 3.7. Further, again a model for the asymptotic behaviour of the runtime is fit on the data. Due to the computational expensive cell averaging and backward Euler damping in the regime with smaller number of discretization points, the asymptotic behaviour of the PCA-based approximation approach is not yet completely visible. Neglecting the, asymptotically not important, cell averaging and backward Euler damping shows indeed the expected asymptotic behaviour of the runtime that scales $\mathcal{O}\left(dNm^2\right)$. The linear dependence of the dimension is clearly visible in the measured runtimes of the PCA-based approximation approach applied to Set D, E and F, as shown in Figure 3.8.

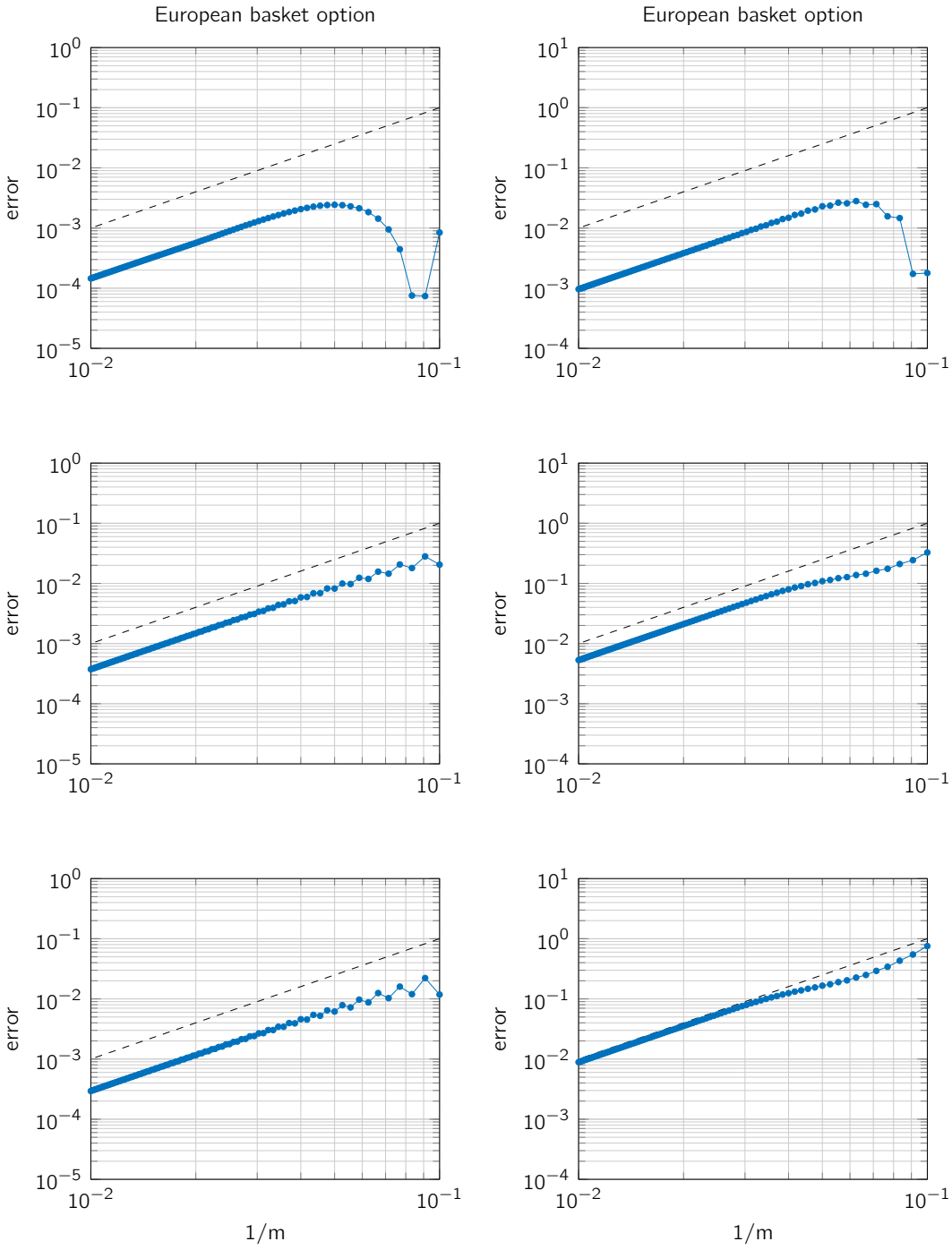


Figure 3.6: Discretization error for PCA-based approximation $\widetilde{u}(\boldsymbol{S_0}, T)$ in Set A, B and C (left; top to bottom) and D, E, and F (right; top to bottom). Reference line (dashed) included for second-order convergence.

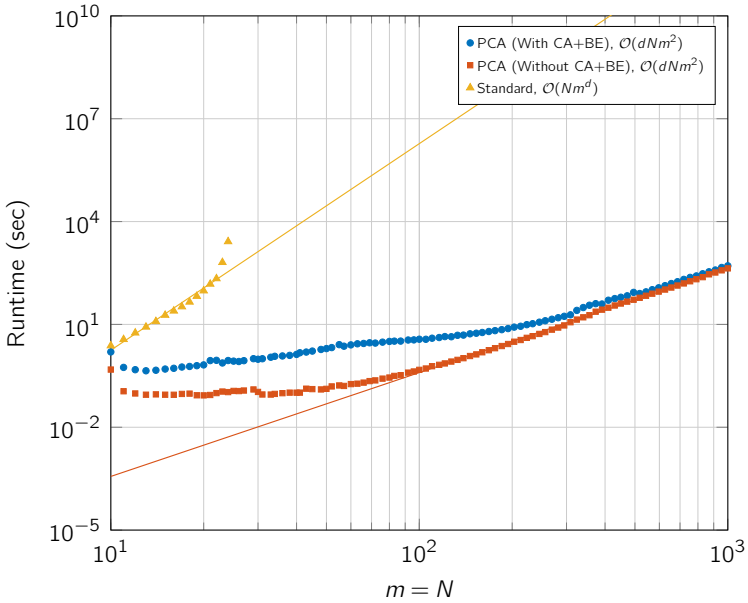


Figure 3.7: Comparison of total runtime for numerical solving the 5-dimensional Black–Scholes PDE using a standard spatial discretization and the runtime for approximating this solution using the PCA-based approximation approach.

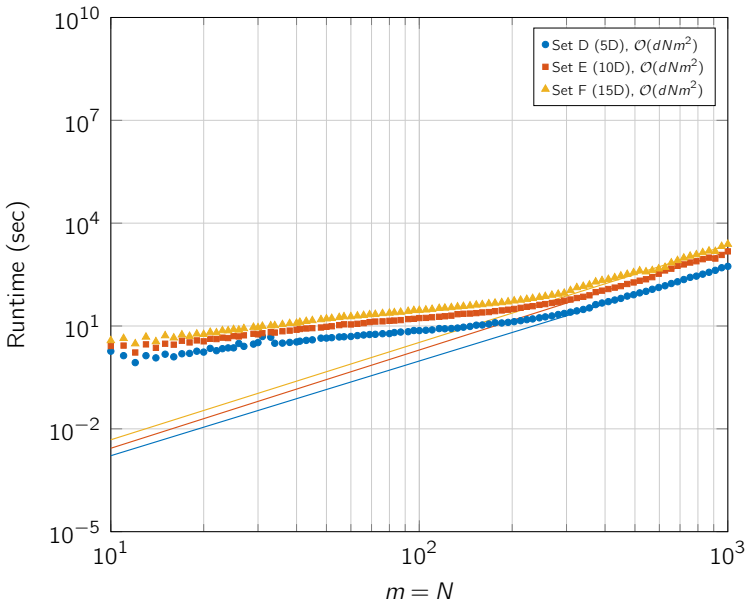


Figure 3.8: Comparison of total runtime for approximating the solution of a d-dimensional Black–Scholes PDE of Sets D, E and F, with $d \in \{5, 10, 15\}$, using the PCA-based approximation approach.

3.6 Conclusions and outlook

In this chapter we have investigated the PCA-based approximation approach by Reisinger & Wittum [71] for the valuation of European-style basket options. This approximation approach is highly effective as it requires the solution of only a limited number of low-dimensional PDEs.

By numerical experiments the favourable result is shown that a common discretization of these PDE problems leads to a second-order convergence behaviour in space and time.

This promising result gives ideas to apply this method also to Bermudan-style and American-style basket options, which is subject of Chapters 4 and 5. Further it is interesting to explore the possibilities to evaluate one or more of the Greeks for an basket option, like the Deltas. This will be subject of Chapter 6.

Chapter 4

Bermudan-style basket options

Chapter summary:

In this chapter we study the principal component analysis (PCA) based approach introduced by Reisinger & Wittum [71] for the approximation of Bermudan-style basket option values via partial differential equations (PDEs). This highly efficient approximation approach requires the solution of only a limited number of low-dimensional PDEs complemented with optimal exercise conditions.

It is demonstrated by ample numerical experiments that a common discretization of the pertinent PDE problems yields a second-order convergence behaviour in space and time, which is as desired. It is also found that this behaviour can be somewhat irregular, and insight into this phenomenon is obtained.

The content of this chapter is based on published work in '*Numerical valuation of Bermudan basket options via partial differential equations*' by Karel in 't Hout and Jacob Snoeijer, [41].

4.1 Introduction

This chapter deals with the valuation of Bermudan-style basket options. Basket options have a payoff depending on a weighted average of different assets. Semi-closed analytic valuation formulas are generally lacking in the literature for these options. Consequently, research into efficient and stable methods for approximating their fair values is of much interest.

Up to now three main approaches have been considered in the literature for the approximate valuation of financial options. The first approach is by Monte Carlo methods. These estimate the expected discounted payoff value by computing sample means. In particular we mention in the present context the stochastic grid bundling method by Jain & Oosterlee

[45]. The second approach is by numerical integration, which is employed in for example the Carr–Madan method [7] and the COS method of Fang & Oosterlee [20]. The third approach is to numerically solve a time-dependent partial differential equation (PDE) that holds for the option value.

The valuation of basket options gives rise to a d-dimensional time-dependent PDE. Here the spatial dimension d equals the number of different assets in the basket. Our interest in this chapter is in the situation where the dimension is large, say $d \geq 5$. It is well-known that this leads to a very challenging task and only few effective computational methods are available in the literature. Recently, research has started into the application of deep neural networks for high-dimensional PDEs, see Sirignano & Spiliopoulos [77].

In the present chapter we shall investigate a principal component analysis based approximation approach introduced by Reisinger & Wittum [71] and subsequently studied in e.g. [68, 69, 70] that renders this task feasible. In particular, Reisinger & Wissmann [68] applied this approach to Bermudan contracts, namely for Bermudan swaptions in the LIBOR market model.

A *Bermudan-style basket option* is a financial contract that provides the holder the right to buy or sell a given weighted average of d assets for a specified price K at one from a specified finite set of exercise times $\tau_1 < \tau_2 < \cdots < \tau_E = T$ with $\tau_1 > 0$.

Again, we assume in this chapter the well-known Black–Scholes model, similar to Chapter 3. To describe the Bermudan-style specific characteristics of this option, let $\alpha_e = T - \tau_{E-e}$ for $e = 0, 1, \ldots, E-1$ and $\alpha_E = T$. Then the fair value function u of a Bermudan-style basket option satisfies the PDE (3.2), with the natural boundary condition, on each time interval (α_{e-1}, α_e) for $e = 1, 2, \ldots, E$. Next, the initial condition (3.3) holds and for $e = 1, 2, \ldots, E-1$ one has

$$u(\boldsymbol{s}, \alpha_e) = \max\left(\phi(\boldsymbol{s}),\ \lim_{t \uparrow \alpha_e} u(\boldsymbol{s}, t)\right) \tag{4.1}$$

whenever $\boldsymbol{s} \in (0, \infty)^d$. Condition (4.1) stems from the early exercise feature of Bermudan-style options and represents the optimal exercise condition. Notice that it is nonlinear.

In the present chapter we shall consider the class of Bermudan-style basket put options with payoff function of the form (3.4).

An outline of the rest of this chapter is as follows. Following Reisinger & Wittum [71], we first apply in Section 4.2.1 a useful coordinate transformation to (3.2) by using a spectral decomposition of the pertinent covariance matrix. This leads to a d-dimensional time-dependent PDE for a transformed option value function w in which each coefficient is directly proportional to one of the eigenvalues. In Section 4.2.2 this feature is employed to define a principal component analysis (PCA) based approximation $\widetilde{w}$ to w. The key property of $\widetilde{w}$ is that it is defined by only a limited number of one- and two-dimensional PDEs. In Section 4.2.3 a note on the optimal exercise condition is given. Section 4.3 describes a common discretization of the one- and two-dimensional PDE problems by means of finite differences on a suitable nonuniform spatial grid followed by the Brian and Douglas ADI scheme on a uniform temporal grid. In view of the nonsmoothness of the payoff function, cell averaging and backward Euler damping are applied.

The main contribution of this chapter is given in Section 4.4. Extensive numerical experiments are presented where we study in detail the error of the discretization described in Section 4.3 of the PCA-based approximation $\widetilde{w}$ defined in Section 4.2.2 for Bermudan-style basket options. Six financial parameter sets from the literature are considered, with number of assets $d \in \{5, 10, 15\}$. A second-order convergence behaviour is observed, which is as desired. It is also found that this behaviour can be somewhat irregular. Additional numerical experiments are performed that yield insight into this phenomenon. Section 4.5 contains our conclusions and outlook.

4.2 PCA-based approximation approach

4.2.1 Coordinate transformation

For a detailed presentation of the coordinate transformation that is used for European- and Bermudan-style basket options we refer to Section 3.2 in the chapter about European-style basket options.

The coordinate transformation results in, see also (3.17):

$$\frac{\partial w}{\partial t}(\boldsymbol{y}, t) = \sum_{k=1}^{d} \lambda_k \left[p(y_k) \frac{\partial^2 w}{\partial y_k^2}(\boldsymbol{y}, t) + q(y_k) \frac{\partial w}{\partial y_k}(\boldsymbol{y}, t) \right] - r w(\boldsymbol{y}, t) \tag{4.2}$$

whenever $\boldsymbol{y} \in (0,1)^d$, $t \in (\alpha_{e-1}, \alpha_e)$, $1 \le e \le E$ with

$$p(\eta) = \frac{1}{2\pi^2} \sin^4(\pi\eta), \quad q(\eta) = \frac{1}{\pi} \sin^3(\pi\eta)\cos(\pi\eta) \quad \text{for } \eta \in \mathbb{R}.$$

Clearly, the PDE (4.2) is a convection-diffusion-reaction equation without mixed derivative terms. Recall that the function ψ in (3.18) is defined by

$$\psi(\boldsymbol{y}, t) = \phi(K \exp[\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}(t)]) \quad \text{with} \quad \boldsymbol{x} = \tan\left[\pi(\boldsymbol{y} - \tfrac{1}{2})\right] \tag{4.3}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$, $t \in [0, T]$. Then for (4.2) one has the initial condition

$$w(\boldsymbol{y}, 0) = \psi(\boldsymbol{y}, 0) \tag{4.4}$$

together with the optimal exercise condition

$$w(\boldsymbol{y}, \alpha_e) = \max\left(\psi(\boldsymbol{y}, \alpha_e),\ \lim_{t \uparrow \alpha_e} w(\boldsymbol{y}, t) \right) \tag{4.5}$$

for $\boldsymbol{y} \in (0,1)^d$ and $e = 1, 2, \ldots, E-1$.

At the boundary ∂D of the spatial domain $D = (0,1)^d$ we shall consider a Dirichlet condition. In Section 3.2.1.3 the details of its derivation are provided, where the minor assumption formulated in Assumption 1 on the matrix $\boldsymbol{Q}$ is made. For any given $k \in \{1, 2, \ldots, d\}$ such that the entries of the k-th column of $\boldsymbol{Q}$ are all strictly positive there holds

$$w(\boldsymbol{y}, t) = K e^{-r(t - \alpha_{e-1})} \tag{4.6}$$

whenever $\boldsymbol{y} \in \partial D$ with $y_k = 0$ and $t \in (\alpha_{e-1}, \alpha_e)$, $1 \le e \le E$. On the complementary part of ∂D a homogeneous Dirichlet condition is valid.

4.2.2 PCA-based approximation approach for Bermudan basket option

The PCA-based approximation approach for the Bermudan-style basket options follows exactly the same path as for European-style basket options as discussed in Section 3.2.2.

Recall (3.23), then the *PCA-based approximation* for Bermudan-style basket options reads

$$w(\boldsymbol{y},t) \approx \widetilde{w}(\boldsymbol{y},t) = w^{(1)}(\boldsymbol{y},t) + \sum_{l=2}^{d} \Big[w^{(1,l)}(\boldsymbol{y},t) - w^{(1)}(\boldsymbol{y},t) \Big] \tag{4.7}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d,\ t \in (\alpha_{e-1}, \alpha_e),\ 1 \le e \le E$.

By definition, $w^{(1)}$ satisfies the PDE (4.2) with λ_k being set to zero for all $k \neq 1$ and $w^{(1,l)}$ satisfies (4.2) with λ_k being set to zero for all $k \notin \{1, l\}$, which is completed by the same initial condition, optimal exercise condition and boundary condition as for w.

We formally write

$$\begin{aligned} \widetilde{u}(\boldsymbol{s},t) &= \widetilde{w}(\boldsymbol{y}(\boldsymbol{x}(\boldsymbol{s},t)),t) \\ u^{(1)}(\boldsymbol{s},t) &= w^{(1)}(\boldsymbol{y}(\boldsymbol{x}(\boldsymbol{s},t)),t) \\ u^{(1,l)}(\boldsymbol{s},t) &= w^{(1,l)}(\boldsymbol{y}(\boldsymbol{x}(\boldsymbol{s},t)),t) \end{aligned} \tag{4.8}$$

for the PCA-based approximation and its terms in the original coordinates.

In financial applications one is often interested in the option value at inception in the single point $\boldsymbol{S_0} = (S_0^1, S_0^2, \ldots, S_0^d)^\mathsf{T}$ is the vector of known asset prices. Let

$$\boldsymbol{Y_0} = \boldsymbol{y}(\boldsymbol{x}(\boldsymbol{S_0}, T)) \in (0,\ 1)^d$$

denote the corresponding point in the y-domain with elements $\boldsymbol{Y_0} = (Y_0^1, Y_0^2, \ldots, Y_0^d)^\mathsf{T}$.

Then $w^{(1)}(\boldsymbol{Y_0}, T)$ can be obtained by solving a one-dimensional PDE on the line segment L_1 in the y-domain that is parallel to the y_1-axis and passes through $y = \boldsymbol{Y_0}$. In other words, y_k can be fixed at the value Y_0^k whenever $k \neq 1$.

Next, $w^{(1,l)}(\boldsymbol{Y_0}, T)$ with $2 \le l \le d$ can be obtained by solving a two-dimensional PDE on the plane segment P_l in the y-domain that is parallel to the $(y_1,\ y_l)$-plane and passes through $y = \boldsymbol{Y_0}$. Thus, in this case, y_k can be fixed at the value Y_0^k whenever $k \notin \{1, l\}$.

In view of the above key observation, computing the PCA-based approximation (4.7) for $(\boldsymbol{y}, t) = (\boldsymbol{Y_0}, T)$ requires solving just 1 one-dimensional PDE and $d-1$ two-dimensional PDEs. This clearly yields a main computational advantage compared to solving the full d-dimensional PDE whenever d is large.

We mention that the PCA-based approximation approach described above is directly extended to other types of multi-asset payoffs, such as for rainbow options. This requires only straightforward modifications to the initial, boundary and optimal exercise conditions.

4.2.3 A note regarding the optimal exercise condition

Let $1 \le e \le E-1$ and write $\psi_e(\boldsymbol{y}) = \psi(\boldsymbol{y}, \alpha_e)$. Let $\boldsymbol{y} \in L_1$, which forms the intersection of L_1 and $P_2, \ldots, P_d$. By the optimal exercise condition (4.5), the natural approximation to $w(\boldsymbol{y}, t)$ at $t = \alpha_e$ based on $\widetilde{w}$ is

$$\begin{aligned} w(\boldsymbol{y}, \alpha_e) &\approx \max\left(\psi_e(\boldsymbol{y}),\ \lim_{t \uparrow \alpha_e} \widetilde{w}(\boldsymbol{y}, t)\right) \\ &= \lim_{t \uparrow \alpha_e} \max\left(\psi_e(\boldsymbol{y}),\ \widetilde{w}(\boldsymbol{y}, t)\right) \\ &= \lim_{t \uparrow \alpha_e} \max\left(\psi_e(\boldsymbol{y}),\ w^{(1)}(\boldsymbol{y}, t) + \sum_{l=2}^{d} \left[w^{(1,l)}(\boldsymbol{y}, t) - w^{(1)}(\boldsymbol{y}, t)\right]\right). \end{aligned}$$

On the other hand, by construction of $w^{(1)}$ and $w^{(1,\,l)}$ for $2 \le l \le d$, we have

$$\begin{aligned} w(\boldsymbol{y}, \alpha_e) &\approx \widetilde{w}(\boldsymbol{y}, \alpha_e) \\ &= w^{(1)}(\boldsymbol{y}, \alpha_e) + \sum_{l=2}^{d} \left[w^{(1,l)}(\boldsymbol{y}, \alpha_e) - w^{(1)}(\boldsymbol{y}, \alpha_e)\right] \\ &= \lim_{t \uparrow \alpha_e} \Big(\max\left(\psi_e(\boldsymbol{y}),\, w^{(1)}(\boldsymbol{y}, t)\right) + \sum_{l=2}^{d} \left[\max\left(\psi_e(\boldsymbol{y}),\, w^{(1,l)}(\boldsymbol{y}, t)\right) - \max\left(\psi_e(\boldsymbol{y}),\, w^{(1)}(\boldsymbol{y}, t)\right)\right]\Big). \end{aligned}$$

It may hold that

$$\widetilde{w}(\boldsymbol{y}, \alpha_e) \ne \max\left(\psi_e(\boldsymbol{y}),\ \lim_{t \uparrow \alpha_e} \widetilde{w}(\boldsymbol{y}, t)\right), \tag{4.9}$$

and hence, the PCA-based approximation $\widetilde{w}$ does not satisfy the optimal exercise condition.

4.3 Discretization

To arrive at the values $w^{(1)}(\boldsymbol{Y_0}, T)$ and $w^{(1,l)}(\boldsymbol{Y_0}, T)$ (for $2 \le l \le d$) in the approximation $\widetilde{w}(\boldsymbol{Y_0}, T)$ of $w(\boldsymbol{Y_0}, T)$ we perform a finite difference discretization of the pertinent one- and two-dimensional PDEs on a (Cartesian) nonuniform spatial grid, followed by a suitable implicit time discretization.

4.3.1 Spatial discretization

The spatial discretization is exactly the same as discussed in Section 3.3.1 for European-style basket options. Semidiscretization of the PDE for $w^{(1,l)}$ on the plane segment P_l leads to a system of ordinary differential equations (ODEs) of the form

$$\boldsymbol{w}'(t) = (\boldsymbol{A_1} + \boldsymbol{A_l})\, \boldsymbol{w}(t) + \boldsymbol{g}(t) \tag{4.10}$$

for $t \in (\alpha_{e-1}, \alpha_e)$, $1 \le e \le E$. Here $\boldsymbol{w}(t)$ is the vectorization of $w^{(1,l)}(t)$ on the plane segment P_l. So, $\boldsymbol{w}(t)$ is a vector of dimension m^2 and $\boldsymbol{A_1}$, $\boldsymbol{A_l}$ are given $m^2 \times m^2$ matrices

that are tridiagonal (possibly up to permutation), commute and correspond to, respectively, the first and the l-th spatial direction. Further, $\boldsymbol{g}(t) = \boldsymbol{g_1}(t) + \boldsymbol{g_l}(t)$ is a given vector of dimension m^2, which stems from the Dirichlet boundary condition. The ODE system (4.10) is completed by an initial condition

$$\boldsymbol{w}(0) = \boldsymbol{\psi_0}$$

and, for $1 \le e \le E-1$, an optimal exercise condition

$$\boldsymbol{w}(\alpha_e) = \max\left(\boldsymbol{\psi_e},\ \lim_{t\uparrow\alpha_e} \boldsymbol{w}(t)\right). \tag{4.11}$$

Here the vector $\boldsymbol{\psi_e}$ is determined by the function $\psi(\cdot,\alpha_e)$ on P_l for $0 \le e \le E-1$. The maximum of any given two vectors is to be taken componentwise.

The payoff function ϕ given by (3.4) is continuous but not everywhere differentiable, and hence, this also holds for the function ψ given by (4.3). It is well-known that the non-smoothness of the payoff function can have an adverse impact on the convergence behaviour of the spatial discretization. To alleviate this, we employ cell averaging near the points of nonsmoothness in defining the initial vector $\boldsymbol{\psi_0}$, see e.g. [40, 50, 66].

4.3.2 Temporal discretization

For the temporal discretization of the ODE system (4.10) a standard Alternating Direction Implicit (ADI) method is applied. Consider a given step size $\Delta t = T/N$ with integer $N \ge E$ and define temporal grid points $t_n = n\Delta t$ for $n = 0, 1, \ldots, N$.

Assume that $\alpha_e = t_{n_e}$ for some integer n_e whenever $e = 1, 2, \ldots, E-1$. Let $\boldsymbol{w_0} = \boldsymbol{\psi_0}$ and

$$\mathcal{N} = \{n_1, n_2, \ldots, n_{E-1}\}.$$

Application of the familiar second-order Brian and Douglas ADI scheme for two-dimensional PDEs leads to an approximation $\boldsymbol{w_n} \approx \boldsymbol{w}(t_n)$ that is successively defined for $n = 1, 2, \ldots, N$ by

$$\begin{cases} \boldsymbol{z_0} = \boldsymbol{w}_{n-1} + \Delta t\,(\boldsymbol{A_1} + \boldsymbol{A_l})\,\boldsymbol{w}_{n-1} + \Delta t\,\boldsymbol{g}(t_{n-1}), \\ \boldsymbol{z_1} = \boldsymbol{z_0} + \frac{1}{2}\Delta t\,\boldsymbol{A_1}(\boldsymbol{z_1} - \boldsymbol{w}_{n-1}) + \frac{1}{2}\Delta t(\boldsymbol{g_1}(t_n) - \boldsymbol{g_1}(t_{n-1})), \\ \boldsymbol{z_2} = \boldsymbol{z_1} + \frac{1}{2}\Delta t\,\boldsymbol{A_l}(\boldsymbol{z_2} - \boldsymbol{w}_{n-1}) + \frac{1}{2}\Delta t(\boldsymbol{g_l}(t_n) - \boldsymbol{g_l}(t_{n-1})), \\ \boldsymbol{w_n} = \begin{cases} \boldsymbol{z_2} & \text{if } n \notin \mathcal{N} \\ \max(\boldsymbol{\psi_e},\ \boldsymbol{z_2}) & \text{if } n = n_e \in \mathcal{N} \end{cases}. \end{cases} \tag{4.12}$$

In the scheme (4.12) a forward Euler predictor stage is followed by two implicit but unidirectional corrector stages, which serve to stabilize the predictor stage. The two linear systems in each time step can be solved very efficiently by using a priori LU factorizations of the pertinent matrices. The number of floating-point operations per time step is then directly proportional to the number of spatial grid points m^2, which is optimal.

Like for the spatial discretization, also the convergence behaviour of the temporal discretization can be adversely effected by the nonsmooth payoff function. To remedy this, backward Euler damping (or Rannacher time stepping [67]) is applied at initial time as well as at each

Set	European	Bermudan ($E = 10$)
A	0.17577	0.18041
B	0.83257	1.05537
C	0.77065	0.99277
D	9.46550	9.81201
E	9.10039	9.44701
F	8.76358	9.11013

Table 4.1: Reference values $\widetilde{u}(\boldsymbol{S_0}, T)$ for European- and Bermudan-style basket put options.

exercise date, that is, with $n_0 = 0$, the time step from t_{n_e} to t_{n_e+1}, is replaced by two half steps of the backward Euler method for $e = 0, 1, \ldots, E-1$.

Finally, discretization of the PDE for $w^{(1)}$ on the line segment L_1 is performed completely analogously to the above. Then a semidiscrete system

$$\boldsymbol{w}'(t) = \boldsymbol{A_1}\boldsymbol{w}(t) + \boldsymbol{g}_1(t) \tag{4.13}$$

is obtained with $\boldsymbol{w}(t)$ and $\boldsymbol{g}_1(t)$ vectors of dimension m and $\boldsymbol{A_1}$ is an $m \times m$ tridiagonal matrix. Temporal discretization is done using the Crank–Nicolson scheme with backward Euler damping.

4.4 Numerical experiments

In this section we investigate by ample numerical experiments the error of the discretization described in Section 4.3 of the PCA-based approximation $\widetilde{u}(\boldsymbol{S_0}, T)$ defined in Section 4.2.2. We consider the six parameter sets for the basket option and the underlying asset price model as defined in Appendix A.

If not otherwise specified, we consider a Bermudan-style basket option with $E = 10$ equidistant exercise times $\tau_i = i\frac{T}{E}$ with $i = 1, 2, \ldots, E$ and study the absolute error in the discretization of $\widetilde{u}(\boldsymbol{S_0}, T)$ at the point $\boldsymbol{S_0} = (K, K, \ldots, K)^{\mathsf{T}}$. For comparison, also the results for a European-style basket option is included in the experiments. The number of time steps is taken as $N = m$ for the European-style option and $N = E\lceil m/E \rceil$ for the Bermudan-style option.

Table 4.1 provides reference values for $\widetilde{u}(\boldsymbol{S_0}, T)$, which have been computed by using the PCA-based approximation approach and choosing $m = 1000$. The estimated maximal absolute error in this reference values is approximately $5 \cdot 10^{-5}$.

In the case of Sets B and C, Jain & Oosterlee [45] obtain, using the stochastic grid bundling method, the approximations $u(\boldsymbol{S_0}, T) \approx 1.06$ and $u(\boldsymbol{S_0}, T) \approx 1.00$, respectively, for the Bermudan-style basket option. Clearly, these approximations from the literature agree well with our corresponding values for $\widetilde{u}(\boldsymbol{S_0}, T)$ given in Table 4.1.

Figures 4.1 and 4.2 display for Sets A, B, C and D, E, F, respectively, the absolute error in the discretization of $\widetilde{u}(\boldsymbol{S_0}, T)$ versus $1/m$ for all $m = 10, 11, 12, \ldots, 100$. Here both

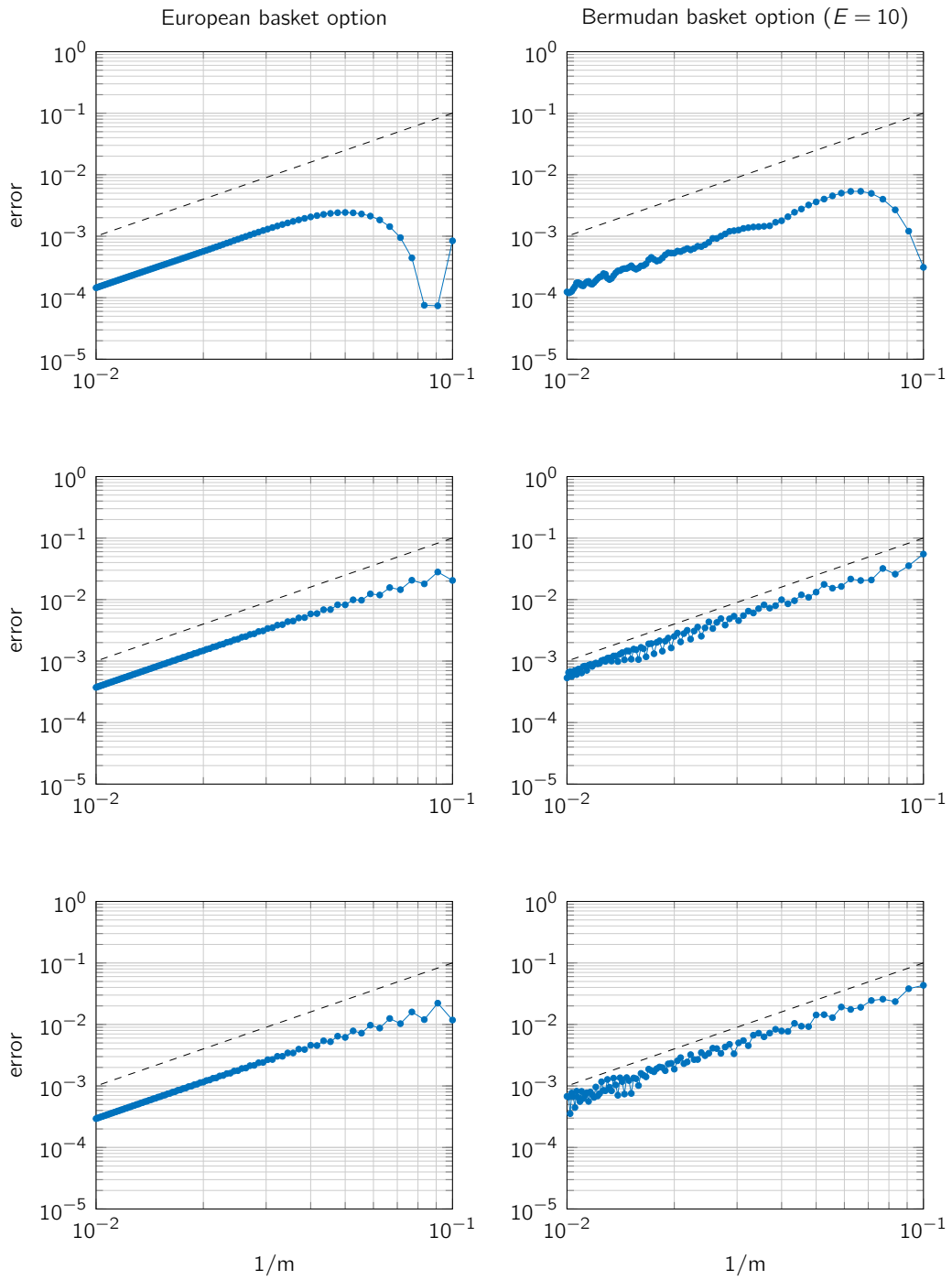


Figure 4.1: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ in Set A (top), B (middle) and C (bottom). Left: European-style basket option. Right: Bermudan-style basket option. Reference line (dashed) included for second-order convergence.

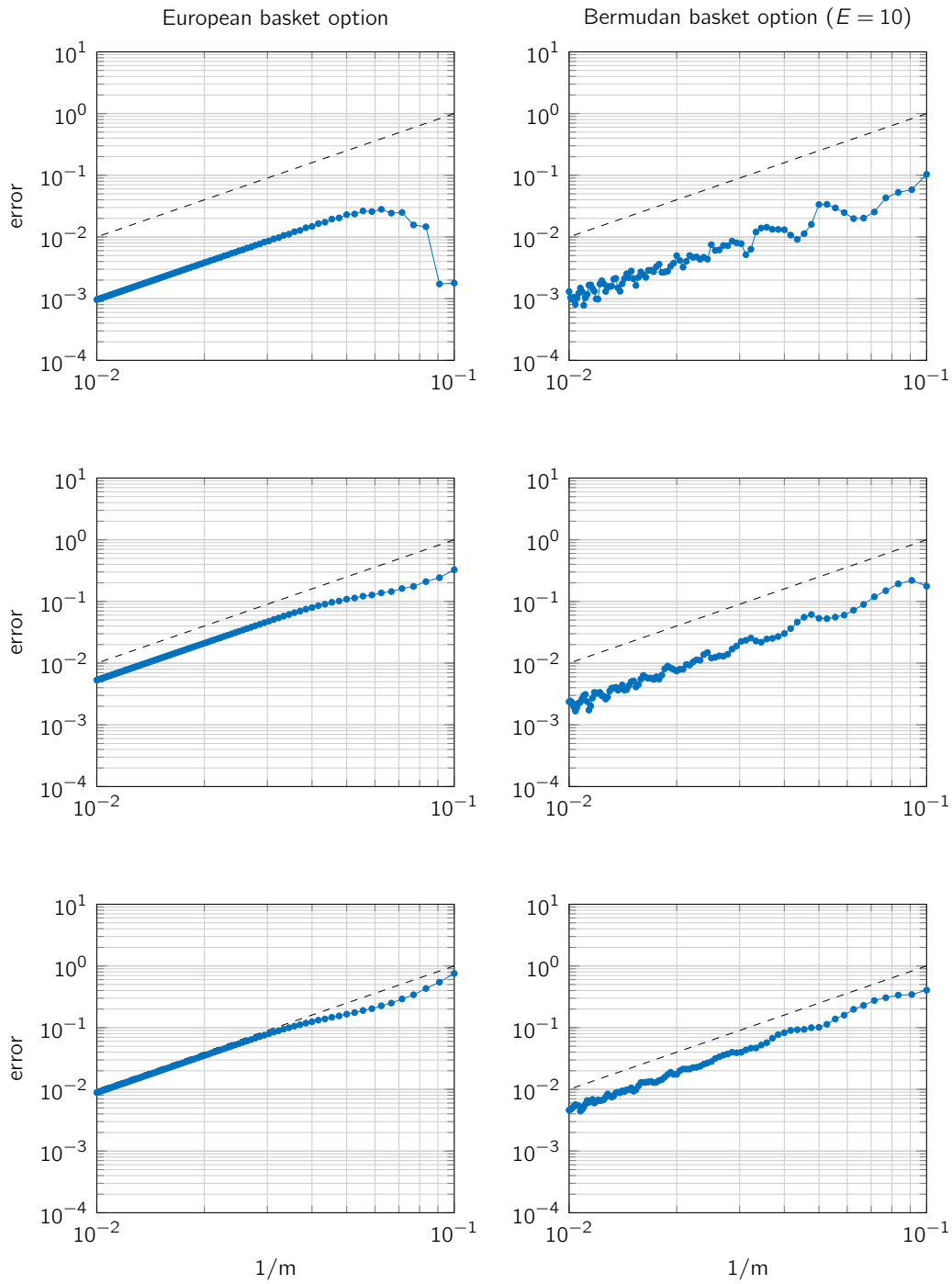


Figure 4.2: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ in Set D (top), E (middle) and F (bottom). Left: European-style basket option. Right: Bermudan-style basket option. Reference line (dashed) included for second-order convergence.

Set	European	Bermudan ($E = 10$)
A	0.18061	0.18407
B	1.00043	1.17792
C	0.94368	1.11902
D	9.57526	9.90055
E	9.31614	9.62111
F	9.07415	9.36091

Table 4.2: Reference values for leading term $u^{(1)}(\boldsymbol{S_0}, T)$ in PCA-based approximation for European- and Bermudan-style basket put options.

the European- and Bermudan-style basket options are considered. The favourable result is observed that the discretization error is always bounded from above by cm^{-2} with a moderate constant c, which is as desired.

As already mentioned in Section 3.5, for the European-style basket option in Sets A and D, we remark that the error drop in the (less important) region $m \leq 20$ that corresponds to a change of sign. Besides this the behaviour of the discretization error is always seen to be regular.

For the Bermudan-style basket option the observed error behaviour is less regular, in particular in the interesting region of large values m. To gain more insight into this phenomenon, we have computed separately the discretization error for the leading term $u^{(1)}(\boldsymbol{S_0}, T)$ and for the correction term $\sum_{l=2}^{d} \left[u^{(1,l)}(\boldsymbol{S_0}, T) - u^{(1)}(\boldsymbol{S_0}, T)\right]$ in $\widetilde{u}(\boldsymbol{S_0}, T)$, see (4.7) and (4.8).

Reference values for the leading term are given in Table 4.2. The obtained result for Sets A, B, C and D, E, F is shown in Figures 4.3 and 4.4, respectively.

It is clear that, with one minor exception in the case of Set D, the error for the leading term behaves regularly and the error for the correction term is small compared to this. For the Bermudan-style basket option, however, the behaviour of the discretization error for the correction term is rather irregular.

A subsequent study shows that for any given l the error $e^{(1,l)}(m)$ is always very close to the error $e^{(1)}(m)$, which is as expected, but the difference can be both positive and negative, leading to an irregular behaviour of the difference $e^{(1,l)}(m) - e^{(1)}(m)$. This is exacerbated when summing these differences up over $l = 2, 3, \ldots, d$. Hence, the irregular behaviour of the error for the correction term can adversely affect the regular behaviour of the error for the leading term.

We remark that this has been observed in many other experiments we performed for the Bermudan-style basket option, for example for other points $\boldsymbol{S_0}$, for other numbers of exercise times $E \geq 2$ and for other dimensions $d \geq 3$ and for other covariance matrices Σ, having $\lambda_1 \gg \lambda_2 > \cdots > \lambda_d > 0$.

Application of the backward Euler method in all time steps, which is unconditionally contractive in the maximum norm, does not yield an improvement.

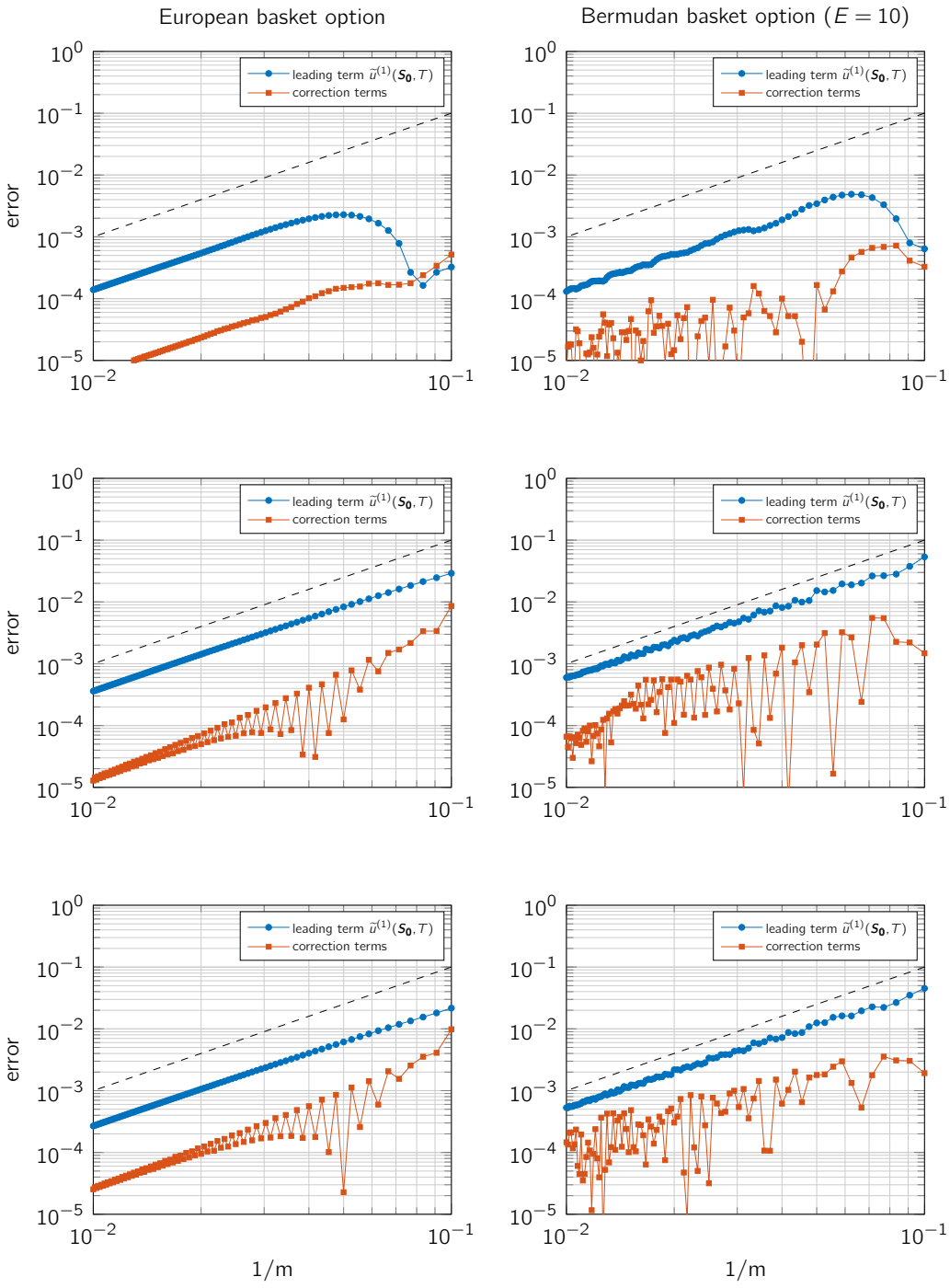


Figure 4.3: Discretization error for leading term $\widetilde{u}^{(1)}(\boldsymbol{S_0}, T)$ and the correction terms in Set A (top), B (middle) and C (bottom). Left: European-style basket option. Right: Bermudan-style basket option. Reference line (dashed) included for second-order convergence.

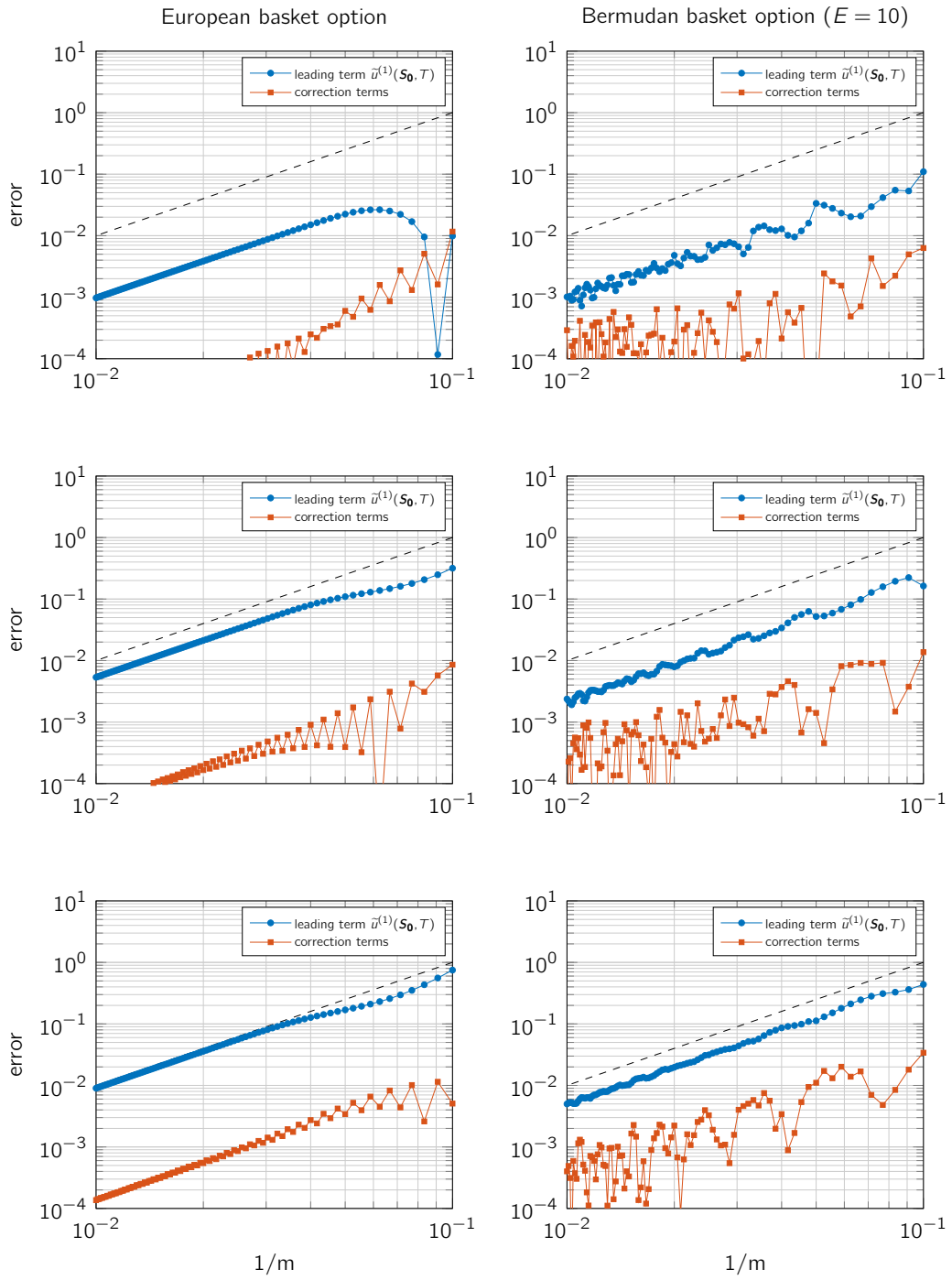


Figure 4.4: Discretization error for leading term $\widetilde{u}^{(1)}(\boldsymbol{S_0}, T)$ and the correction terms in Set D (top), E (middle) and F (bottom). Left: European-style basket option. Right: Bermudan-style basket option. Reference line (dashed) included for second-order convergence.

Indeed, also for different number of exercise times $E \geq 2$ the irregular behaviour of the discretization error of $\widetilde{u}(\boldsymbol{S_0}, T)$ is observed. In a small demonstration for all Sets A–F the number of exercise times are chosen as $E \in \{1, 2, 4, 8\}$. In all these cases new reference values of the Bermudan-style basket option are computed. Again, this is done using the PCA-based approximation approach and choosing $m = 1000$. These reference values are used in a numerical study of the discretization error of $\widetilde{u}(\boldsymbol{S_0}, T)$. The results are shown in Figures 4.5 (for set A and D), 4.6 (for set B and C) and 4.7 (for set E and F). It is clear that the irregular behaviour of the discretization error of $\widetilde{u}(\boldsymbol{S_0}, T)$ starts to appear when $E > 1$. This is exactly the case where the optimal exercise condition is introduced and the Bermudan-style basket option differs from an European-style basket option. Especially visible in the cases of Set A, D and E one might think of the irregular behaviour of the discretization error as some kind of oscillations that are damped and/or amplified by increasing the number of exericse times E.

We attribute the above phenomenon to the spatial nonsmoothness of the exact Bermudan option value function at the early exercise times.

4.5 Conclusions

In this chapter we have investigated the PCA-based approximation approach by Reisinger & Wittum [71] for the valuation of Bermudan-style basket options. This approximation approach is highly effective as it requires the solution of only a limited number of low-dimensional PDEs, supplemented with optimal exercise conditions.

By numerical experiments the favourable result is shown that a common discretization of these PDE problems leads to a second-order convergence behaviour in space and time. It is also observed that this convergence behaviour can be somewhat irregular. Insight into this phenomenon is obtained by regarding the total discretization error as a superposition of discretization errors for the leading term and the correction term.

More research has to be done to explain this irregular behaviour and to determine a suitable remedy for it. The note in Section 4.2.3 maybe helpful in finding a possible source for this irregularity.

Another topic for future research concerns a rigorous analysis of the error in the PCA-based approximation approach for Bermudan-style basket options. Reisinger and Wissmann [69] have given a rigorous analysis of the error in the PCA-based approximation relevant to European-style basket options. These results will be important to extend it also to Bermudan-style basket options.

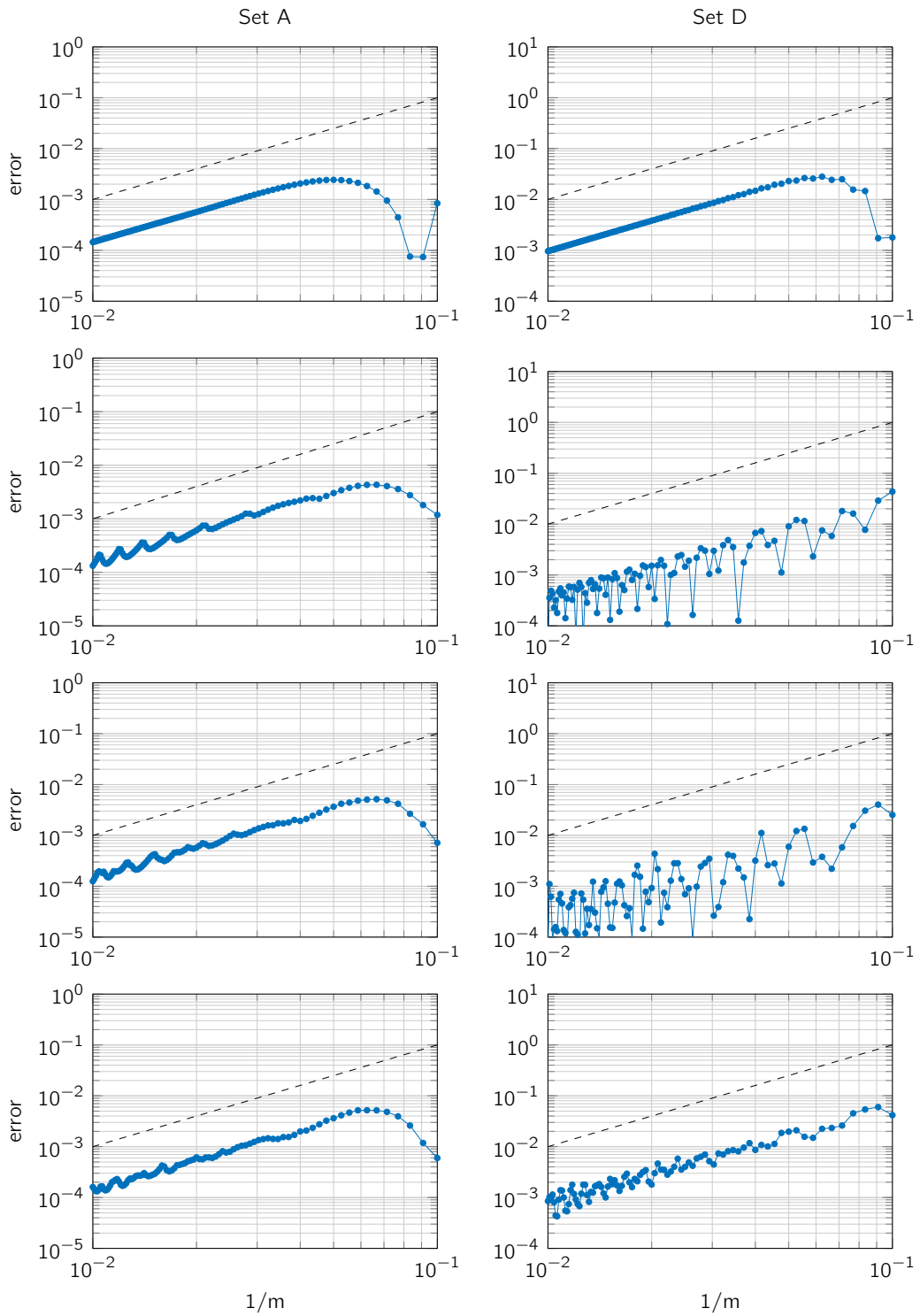


Figure 4.5: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ of a Bermudan-style basket option with number of exercise times $E = 1$ (top), $E = 2$ (top-middle), $E = 4$ (bottom-middle) and $E = 8$ (bottom). Left: set A. Right: set D. Reference line (dashed) included for second-order convergence.

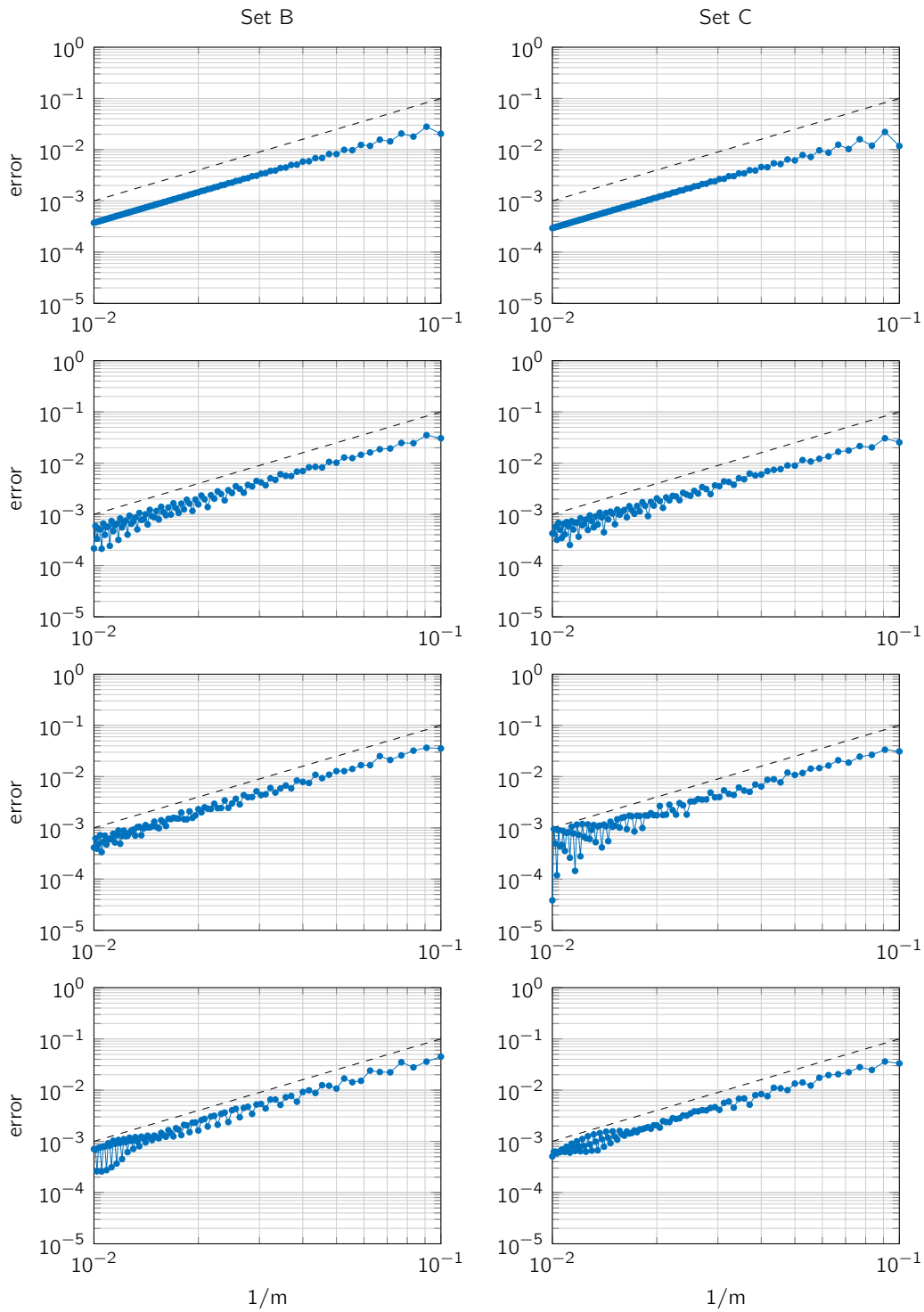


Figure 4.6: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ of a Bermudan-style basket option with number of exercise times $E = 1$ (top), $E = 2$ (top-middle), $E = 4$ (bottom-middle) and $E = 8$ (bottom). Left: set B. Right: set C. Reference line (dashed) included for second-order convergence.

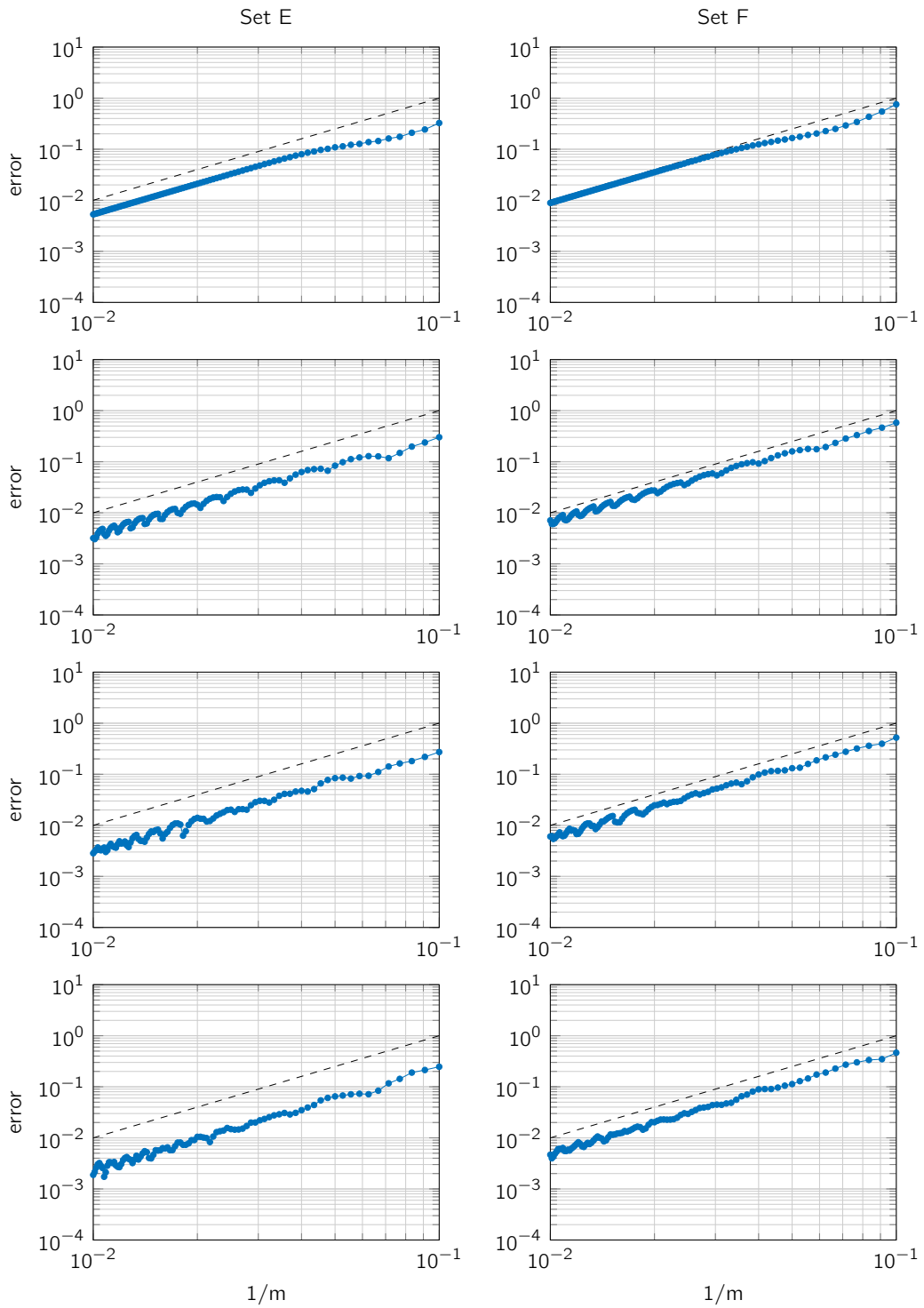


Figure 4.7: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ of a Bermudan-style basket option with number of exercise times $E = 1$ (top), $E = 2$ (top-middle), $E = 4$ (bottom-middle) and $E = 8$ (bottom). Left: set E. Right: set F. Reference line (dashed) included for second-order convergence.

Chapter 5

American-style basket options

Chapter summary:

In this chapter we study the principal component analysis based approach introduced by Reisinger and Wittum (2007) and the comonotonic approach considered by Hanbali and Linders (2019) for the approximation of American-style basket option values via multidimensional partial differential complementarity problems (PDCPs).

Both approximation approaches require the solution of just a limited number of low-dimensional PDCPs. It is demonstrated by ample numerical experiments that they define approximations that lie close to each other.

Next, an efficient discretization of the pertinent PDCPs is presented that leads to a favourable convergence behaviour.

The content of this chapter is mainly based on published work in '*Numerical valuation of American basket options via partial differential complementarity problems*' by Karel in 't Hout and Jacob Snoeijer, [39].

5.1 Introduction

In this chapter, we consider the valuation of American-style basket options through partial differential complementarity problems (PDCPs). If d denotes the number of different assets in the basket, then the pertinent PDCP is d-dimensional. In this chapter, we are interested in the situation where d is medium or large, say $d \geq 5$. It is well-known that this renders the application of standard discretization methods for PDCPs impractical, due to the curse of dimensionality.

In the literature, an alternative useful approach has been investigated that employs the idea of comonotonicity. For European-style basket options, this comonotonic approach has been developed notably by Kaas et al. [46], Dhaene et al. [17, 18], Vyncke et al. [91], Deelstra et

al. [15, 16] and Chen et al. [9, 10]. Recently, an extension to American-style basket options has been presented by Hanbali and Linders [29], who consider a comonotonic approximation formula that requires the solution of just two one-dimensional PDCPs. In this chapter we shall study and compare the PCA-based and comonotonic approaches for the effective valuation of American-style basket options. To our knowledge, our paper [39] is the first paper where these two, different but related, approaches are jointly investigated.

We assume in this chapter the well-known Black–Scholes model, as introduced in Chapter 3.

An *American-style basket option* is a financial contract that gives the holder the right to buy or sell a prescribed weighted average of d assets for a prescribed strike price K at any given single time up to and including a prescribed maturity time T. The fair value function u of an American-style basket option satisfies the (nonlinear) d-dimensional time-dependent PDCP

$$\begin{aligned} u(\boldsymbol{s},t) &\geq \phi(\boldsymbol{s}), \\ \frac{\partial u}{\partial t}(\boldsymbol{s},t) &\geq \mathcal{A}u(\boldsymbol{s},t), \\ (u(\boldsymbol{s},t)-\phi(\boldsymbol{s}))\left(\frac{\partial u}{\partial t}(\boldsymbol{s},t)-\mathcal{A}u(\boldsymbol{s},t)\right) &= 0 \end{aligned} \tag{5.1}$$

whenever $(\boldsymbol{s},t) \in (0,\infty)^d \times (0,T]$. Here $\mathcal{A}$ denotes the Black–Scholes operator, see (3.2):

$$\mathcal{A}u(\boldsymbol{s},t) = \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sigma_i\sigma_j\rho_{ij}s_is_j\frac{\partial^2 u}{\partial s_i \partial s_j}(\boldsymbol{s},t) + \sum_{i=1}^{d} rs_i\frac{\partial u}{\partial s_i}(\boldsymbol{s},t) - ru(\boldsymbol{s},t). \tag{5.2}$$

If ϕ is the given payoff function of the option, then the PDCP (5.1) is provided with the initial condition

$$u(\boldsymbol{s},0) = \phi(\boldsymbol{s}) \tag{5.3}$$

whenever $\boldsymbol{s} \in (0,\infty)^d$. Further, (5.1) also holds if $s_i = 0$ for $i = 1,2,\ldots,d$. In this chapter, we shall consider the class of basket put options, with a payoff as given in (3.4).

The outline of this chapter is as follows. In Section 5.2, the PCA-based approximation approach, for European-style basket options discussed in Section 3.2.2, is extended to American-style basket options. This gives rise to an approximation that is defined by a limited number of one- and two-dimensional PDCPs. The discretization of the one- and two-dimensional PDEs for European-style basket options as discussed in Section 3.3 is adapted in Section 5.3 to the pertinent PDCPs for American-style basket options, where the basic explicit payoff (EP) approach as well as the more advanced Ikonen–Toivanen (IT) splitting technique are considered. Section 5.4 collects results from the literature on the comonotonic approach for valuing European- and American-style basket options. We consider the same comonotonic approximation as Hanbali and Linders [29], which is determined by just two one-dimensional PDEs (for the European-style basket option) or PDCPs (for the American-style basket option). Section 5.5 contains the main contribution of this chapter. In this section, we perform ample numerical experiments and obtain the positive result that the PCA-based approximation and comonotonic approaches yield approximations to the option value that always lie close to each other for both European- and American-style basket put

options. We next study in detail the error in the discretization described in Section 5.3 for the PCA-based and comonotonic approximations and observe a favourable, near second-order convergence behaviour. The final Section 5.6 presents our conclusions and outlook.

5.2 PCA-based approximation approach

For the presentation of the coordinate transformation that is used for European- and American-style basket options we refer to Section 3.2 in the chapter about European-style basket options.

As also given in (3.17), this transformation results in the transformed PDE

$$\frac{\partial w}{\partial t}(\boldsymbol{y},t) = \mathcal{B}w(\boldsymbol{y},t) := \sum_{k=1}^{d} \lambda_k \left[p(y_k)\frac{\partial^2 w}{\partial y_k^2}(\boldsymbol{y},t) + q(y_k)\frac{\partial w}{\partial y_k}(\boldsymbol{y},t)\right] - rw(\boldsymbol{y},t) \tag{5.4}$$

whenever $\boldsymbol{y} \in (0,1)^d,\ t \in (0,\ T]$ with

$$p(\eta) = \frac{1}{2\pi^2}\sin^4(\pi\eta)\,, \quad q(\eta) = \frac{1}{\pi}\sin^3(\pi\eta)\cos(\pi\eta) \quad \text{for } \eta \in \mathbb{R}.$$

The PDE (5.4) is a convection-diffusion-reaction equation without mixed derivatives. Let ψ denote the transform of the payoff function ϕ,

$$\psi(\boldsymbol{y},t) = \phi(K\exp[\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}(t)]) \quad \text{with} \quad \boldsymbol{x} = \tan\left[\pi(\boldsymbol{y} - \tfrac{1}{2})\right] \tag{5.5}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d,\ t \in [0,\ T]$. Then for (5.4) one has the initial condition

$$w(\boldsymbol{y},0) = \psi(\boldsymbol{y},0). \tag{5.6}$$

Applying the coordinate transformation from Section 3.2.1 to the PDCP (5.1) for the value function u of an American-style basket option, directly yields the following PDCP for the transformed function w,

$$\begin{aligned} w(\boldsymbol{y},t) &\geq \psi(\boldsymbol{y},t), \\ \frac{\partial w}{\partial t}(\boldsymbol{y},t) &\geq \mathcal{B}w(\boldsymbol{y},t), \\ (w(\boldsymbol{y},t) - \psi(\boldsymbol{y},t))\left(\frac{\partial w}{\partial t}(\boldsymbol{y},t) - \mathcal{B}w(\boldsymbol{y},t)\right) &= 0 \end{aligned} \tag{5.7}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d,\ t \in (0,\ T]$ with function ψ defined by (5.5) and initial condition (5.6). As for the European-style options, a Dirichlet condition is taken at the boundary of the spatial domain $D = (0,1)^d$. For any given $k \in \{1,2,\ldots,d\}$ such that the entries of the k-th column of $\boldsymbol{Q}$ are all strictly positive there holds

$$w(\boldsymbol{y},t) = K \tag{5.8}$$

whenever $\boldsymbol{y} \in \partial D$ with $y_k = 0$ and $t \in (0,T]$. Notice that, compared to (3.20), the discount factor $\exp(-rt)$ is absent in (5.8). On the complementary part of ∂D, a homogeneous Dirichlet condition is valid.

The PCA-based approximation for the American-style basket option value function w is given by (3.23), where by definition $w^{(1)}$ satisfies the PDCP (5.7) with λ_k being set to zero for all $k \neq 1$, and $w^{(1,l)}$ satisfies (5.7) with λ_k being set to zero for all $k \notin \{1, l\}$.

5.3 Discretization

Semidiscretization of the pertinent one- and two-dimensional PDCPs in the case of American-style basket options follows along the same lines as described in Section 3.3 for the corresponding PDEs in the case of European-style basket options. The relevant boundary condition (5.8) is now independent of time, and hence, the same holds for $\boldsymbol{g}$.

Semidiscretization of the PDCP for $w^{(1,l)}$ on the plane segment P_l yields

$$\begin{aligned} \boldsymbol{w}(t) &\geq \boldsymbol{\psi}(t), \\ \boldsymbol{w}'(t) &\geq (\boldsymbol{A_1} + \boldsymbol{A_l})\, \boldsymbol{w}(t) + \boldsymbol{g}, \\ (\boldsymbol{w}(t) - \boldsymbol{\psi}(t))^{\mathsf{T}} \left(\boldsymbol{w}'(t) - (\boldsymbol{A_1} + \boldsymbol{A_l})\, \boldsymbol{w}(t) - \boldsymbol{g}\right) &= 0 \end{aligned} \tag{5.9}$$

for $t \in (0,\ T]$ and $\boldsymbol{w}(0) = \boldsymbol{w_0}$. Here $\boldsymbol{\psi}(t)$ is a vector of dimension m^2 that is determined by the function $\psi(\cdot, t)$ on P_l. Inequalities for vectors are to be understood componentwise.

For the temporal discretization of the semidiscrete PDCP (5.9) we consider two adaptations of the Brian and Douglas ADI scheme (3.25). They both generate successive approximations $\widehat{\boldsymbol{w}}_{\boldsymbol{n}}$ to $\boldsymbol{w}(t_n)$ for $n = 1, 2, \ldots, N$ with $\widehat{\boldsymbol{w}}_{\boldsymbol{0}} = \boldsymbol{w_0}$.

The first adaptation is elementary and follows the so-called explicit payoff (EP) approach,

$$\begin{cases} \boldsymbol{z_0} = \widehat{\boldsymbol{w}}_{n-1} + \Delta t\, (\boldsymbol{A_1} + \boldsymbol{A_l})\, \widehat{\boldsymbol{w}}_{n-1} + \Delta t \boldsymbol{g}, \\ \boldsymbol{z_1} = \boldsymbol{z_0} + \frac{1}{2}\Delta t \boldsymbol{A_1} (\boldsymbol{z_1} - \widehat{\boldsymbol{w}}_{n-1}), \\ \boldsymbol{z_2} = \boldsymbol{z_1} + \frac{1}{2}\Delta t \boldsymbol{A_l} (\boldsymbol{z_2} - \widehat{\boldsymbol{w}}_{n-1}), \\ \overline{\boldsymbol{w}}_{\boldsymbol{n}} = \boldsymbol{z_2}, \\ \widehat{\boldsymbol{w}}_{\boldsymbol{n}} = \max\{\overline{\boldsymbol{w}}_{\boldsymbol{n}},\ \boldsymbol{\psi_n}\}. \end{cases} \tag{5.10}$$

Here $\boldsymbol{\psi_n} = \boldsymbol{\psi}(t_n)$ and the maximum of two vectors is to be taken componentwise. The adaptation (5.10) can be regarded as first carrying out a time step by ignoring the American constraint and next applying this constraint explicitly.

The second adaptation is more advanced and employs the Ikonen–Toivanen (IT) splitting technique [28, 37, 38],

$$\begin{cases} \boldsymbol{z_0} = \widehat{\boldsymbol{w}}_{n-1} + \Delta t\, (\boldsymbol{A_1} + \boldsymbol{A_l})\, \widehat{\boldsymbol{w}}_{n-1} + \Delta t \boldsymbol{g} + \Delta t \widehat{\boldsymbol{\mu}}_{n-1}, \\ \boldsymbol{z_1} = \boldsymbol{z_0} + \frac{1}{2}\Delta t \boldsymbol{A_1} (\boldsymbol{z_1} - \widehat{\boldsymbol{w}}_{n-1}), \\ \boldsymbol{z_2} = \boldsymbol{z_1} + \frac{1}{2}\Delta t \boldsymbol{A_l} (\boldsymbol{z_2} - \widehat{\boldsymbol{w}}_{n-1}), \\ \overline{\boldsymbol{w}}_{\boldsymbol{n}} = \boldsymbol{z_2}, \\ \widehat{\boldsymbol{w}}_{\boldsymbol{n}} = \max\{\overline{\boldsymbol{w}}_{\boldsymbol{n}} - \Delta t \widehat{\boldsymbol{\mu}}_{n-1},\ \boldsymbol{\psi_n}\}, \\ \widehat{\boldsymbol{\mu}}_{n} = \max\{0,\ \widehat{\boldsymbol{\mu}}_{n-1} + (\boldsymbol{\psi_n} - \overline{\boldsymbol{w}}_{\boldsymbol{n}})/\Delta t\}. \end{cases} \tag{5.11}$$

with $\widehat{\boldsymbol{\mu}}_{\mathbf{0}} = 0$. The auxiliary vector $\widehat{\boldsymbol{\mu}}_{\boldsymbol{n}}$ is often called a Lagrange multiplier. A useful interpretation of this adaptation is given in [42], where is observed that this IT-splitting technique can be seen as an additional (algebraic) Douglas splitting.

The vector $\widehat{\boldsymbol{w}}_{\boldsymbol{n}}$ and the auxiliary vector $\widehat{\boldsymbol{\mu}}_n$ are computed in two parts. In the first part, an intermediate vector $\overline{\boldsymbol{w}}_{\boldsymbol{n}}$ is computed. In the second part, $\overline{\boldsymbol{w}}_{\boldsymbol{n}}$ and $\widehat{\boldsymbol{\mu}}_{n-1}$ are updated to $\widehat{\boldsymbol{w}}_{\boldsymbol{n}}$ and $\widehat{\boldsymbol{\mu}}_{\boldsymbol{n}}$ by a certain simple, explicit formula.

The obtained accuracy for the adaptation by the IT approach is generally better than by the EP approach, see, e.g., [40, 42] and also Section 5.5 below. A virtue of both adaptations (5.10), (5.11) is that the computational cost per time step is essentially the same as that for the standard Brian and Douglas ADI scheme as given in (3.25).

5.4 Comonotonic approach

In a variety of papers in the literature, the concept of comonotonicity has been employed for arriving at efficiently computable approximations as well as upper and lower bounds for option values. For European-style basket options, relevant references to the comonotonic approach are, notably, Kaas et al. [46], Dhaene et al. [17, 18], Vyncke et al. [91], Deelstra et al. [15, 16] and Chen et al. [9, 10]. Recently, an extension to American-style basket options has been considered by Hanbali and Linders [29]. In this section, we review results obtained with the comonotonic approach and applied in loc. cit. Here the assumption has been made that the payoff function ϕ is convex, which is satisfied by (3.4), and that all correlations in the SDE system (3.1) are nonnegative.

5.4.1 Comonotonic approach for European-style baskets

It follows from [46] that an upper bound for the European-style basket option value function u is acquired by setting all correlations in (3.1) equal to one, i.e., $\rho_{ij} = 1$ for all $i, j = 1, 2, \ldots, d$. Denote this upper bound by u^{up}. Consider the same coordinate transformations as in Section 3.2.1 and denote the obtained transformed functions by v^{up} and w^{up}. The pertinent covariance matrix $\boldsymbol{\Sigma}^{\mathrm{up}} = \left(\sigma_i \sigma_j\right)_{i,j=1}^{d}$ has single nonzero eigenvalue $\lambda^{\mathrm{up}} = \sum_{i=1}^{d} \sigma_i^2$.

Hence, the function v^{up} satisfies the one-dimensional PDE

$$\frac{\partial v^{\mathrm{up}}}{\partial t}(\boldsymbol{x}, t) = \frac{1}{2}\lambda^{\mathrm{up}} \frac{\partial^2 v^{\mathrm{up}}}{\partial x_1^2}(\boldsymbol{x}, t) - r v^{\mathrm{up}}(\boldsymbol{x}, t) \tag{5.12}$$

whenever $\boldsymbol{x} \in \mathbb{R}^d$, $t \in (0,\ T]$. Next, the function w^{up} satisfies the one-dimensional PDE

$$\frac{\partial w^{\mathrm{up}}}{\partial t}(\boldsymbol{y}, t) = \mathcal{B}^{\mathrm{up}} w^{\mathrm{up}}(\boldsymbol{y}, t) := \lambda^{\mathrm{up}} \left[p(y_1) \frac{\partial^2 w^{\mathrm{up}}}{\partial y_1^2}(\boldsymbol{y}, t) + q(y_1) \frac{\partial w^{\mathrm{up}}}{\partial y_1}(\boldsymbol{y}, t) \right] - r w^{\mathrm{up}}(\boldsymbol{y}, t)$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$, $t \in (0,\ T]$. The same initial and boundary conditions apply as in Section 3.2.1, using the pertinent function ψ^{up}.

It turns out that the upper bound above is, in general, rather crude. In the comonotonic approach, accurate lower bounds for the European-style basket option value have been derived, however. We consider here the lower bound chosen in [29], which has been motivated by results obtained in [16, 46]. Let $\nu_i \in (0,\ 1]$ be given by

$$\nu_i = \frac{\sum_{j=1}^{d} \omega_j S_0^j \rho_{ij} \sigma_j}{\sqrt{\sum_{j=1}^{d} \sum_{k=1}^{d} \omega_j \omega_k S_0^j S_0^k \rho_{jk} \sigma_j \sigma_k}} \quad \text{for } 1 \leq i \leq d. \tag{5.13}$$

The lower bound is acquired upon replacing the volatility σ_i by $\nu_i \sigma_i$ for $1 \leq i \leq d$ and subsequently setting in (3.1) all correlations equal to one. Denote this bound by u^{low} and the corresponding transformed functions by v^{low} and w^{low}. Then, with $\lambda^{\text{low}} = \sum_{i=1}^{d} (\nu_i \sigma_i)^2$, the function v^{low} satisfies the one-dimensional PDE

$$\frac{\partial v^{\text{low}}}{\partial t}(\boldsymbol{x}, t) = \frac{1}{2} \lambda^{\text{low}} \frac{\partial^2 v^{\text{low}}}{\partial x_1^2}(\boldsymbol{x}, t) - r v^{\text{low}}(\boldsymbol{x}, t)$$

whenever $\boldsymbol{x} \in \mathbb{R}^d,\ t \in (0,\ T]$. Next, the function w^{low} satisfies the one-dimensional PDE

$$\frac{\partial w^{\text{low}}}{\partial t}(\boldsymbol{y}, t) = \mathcal{B}^{\text{low}} w^{\text{low}}(\boldsymbol{y}, t) := \lambda^{\text{low}} \left[p(y_1) \frac{\partial^2 w^{\text{low}}}{\partial y_1^2}(\boldsymbol{y}, t) + q(y_1) \frac{\partial w^{\text{low}}}{\partial y_1}(\boldsymbol{y}, t) \right] - r w^{\text{low}}(\boldsymbol{y}, t)$$

whenever $\boldsymbol{y} \in (0,\ 1)^d,\ t \in (0,\ T]$. The same initial and boundary conditions apply as in Section 3.2.1, using the pertinent function ψ^{low}.

Clearly, the comonotonic upper as well as lower bound can be viewed as obtained upon replacing in the PDE (3.2) the covariance matrix $\boldsymbol{\Sigma}$ by a certain matrix of rank one. For the lower bound, this rank-one matrix is given by $\boldsymbol{\Sigma}^{\text{low}} = \xi \xi^{\mathsf{T}}$ with (eigen)vector $\xi = (\nu_1 \sigma_1, \nu_2 \sigma_2, \ldots, \nu_d \sigma_d)^{\mathsf{T}}$ and single nonzero eigenvalue $\lambda^{\text{low}} = \xi^{\mathsf{T}} \xi$.

Based on a result by Vyncke et al [91], a specific linear combination of the comonotonic lower and upper bounds has been considered in [29], which approximates the value of a European-style basket option. This *comonotonic approximation* reads

$$u^{\text{app}}(\boldsymbol{S_0}, T) = z u^{\text{low}}(\boldsymbol{S_0}, T) + (1 - z) u^{\text{up}}(\boldsymbol{S_0}, T), \tag{5.14}$$

where $z \geq 0$ is given by

$$z = \frac{c - b}{c - a}$$

with

$$a = \sum_{i=1}^{d} \sum_{j=1}^{d} \omega_i \omega_j S_0^i S_0^j \left(e^{\nu_i \nu_j \sigma_i \sigma_j T} - 1 \right),$$

$$b = \sum_{i=1}^{d} \sum_{j=1}^{d} \omega_i \omega_j S_0^i S_0^j \left(e^{\rho_{ij} \sigma_i \sigma_j T} - 1 \right),$$

$$c = \sum_{i=1}^{d} \sum_{j=1}^{d} \omega_i \omega_j S_0^i S_0^j \left(e^{\sigma_i \sigma_j T} - 1 \right).$$

5.4.2 Comonotonic approach for American-style baskets

In [29], the authors next proposed (5.14) as an approximation to the value of an American-style basket option, where u^{low} and u^{up} are now defined via the solutions w^{low} and w^{up} to the PDCP (5.7) with $\mathcal{B}$ replaced by $\mathcal{B}^{\text{low}}$ and $\mathcal{B}^{\text{up}}$, respectively, and function ψ replaced by ψ^{low} and ψ^{up}, respectively. We remark that, to our knowledge, it is an open question in the literature at present whether these functions u^{low} and u^{up} form actual lower and upper bounds for the American-style basket option value.

For the numerical solution of the pertinent PDEs and PDCPs, in [29] a finite difference method was applied in space and the explicit Euler method in time, with the EP approach for American-style basket options. In the following, we shall employ the spatial and temporal discretizations described in Section 5.3. In particular this allows for much less time steps than is required, in view of stability, by the explicit Euler method.

5.5 Numerical experiments

In this section, we perform ample numerical experiments. Our main aims are to determine whether the PCA-based and comonotonic approaches define approximations to European- and American-style basket put option values that lie close to each other, and next, to gain insight into the error of the discretizations described in Section 5.3 in computing these approximations.

We consider two parts of experiments, depending on the parameter sets chosen for the basket option and underlying asset price model. In the first part we choose the same six parameter sets A–F as defined Appendix A. In the second part we shall select parameter sets similar to those in [29].

Our first numerical experiment concerns the two adaptations of the temporal discretization scheme to PDCPs by the EP and IT approaches as described in Section 5.3 for American-style options. Consider Set A, B and E and $\boldsymbol{S_0} = (K, K, \ldots, K)^{\mathsf{T}}$. For a fixed number of spatial grid points, given by $m = 100$, we study the absolute error in the two pertinent discretizations of the PCA-based and comonotonic approximations $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ in function of the number of time steps $N = 10, 11, 12, \ldots, 100$.

Figure 5.1 displays for these American-style basket options the obtained errors with respect to the values computed for a large number of time steps, $N = 1000$. Note that these errors do not contain the error due to spatial discretization, but only due to the temporal discretization. Figure 5.1 clearly illustrates that, in the PCA-based as well as the comonotonic case, the IT approach yields a (much) smaller error than the EP approach for any given N. For Set B the error behaviour is somewhat irregular for the PCA-based approximation with the IT approach, but for both other sets the errors behave regular in function of the number of time steps. Further, the observed order of convergence for IT is approximately 1.5, whereas for EP it is only approximately 1.0. The better performance of IT compared to EP is well-known in the literature, see, e.g., [38, 40, 42]. Accordingly, in the following, we shall always apply the IT approach.

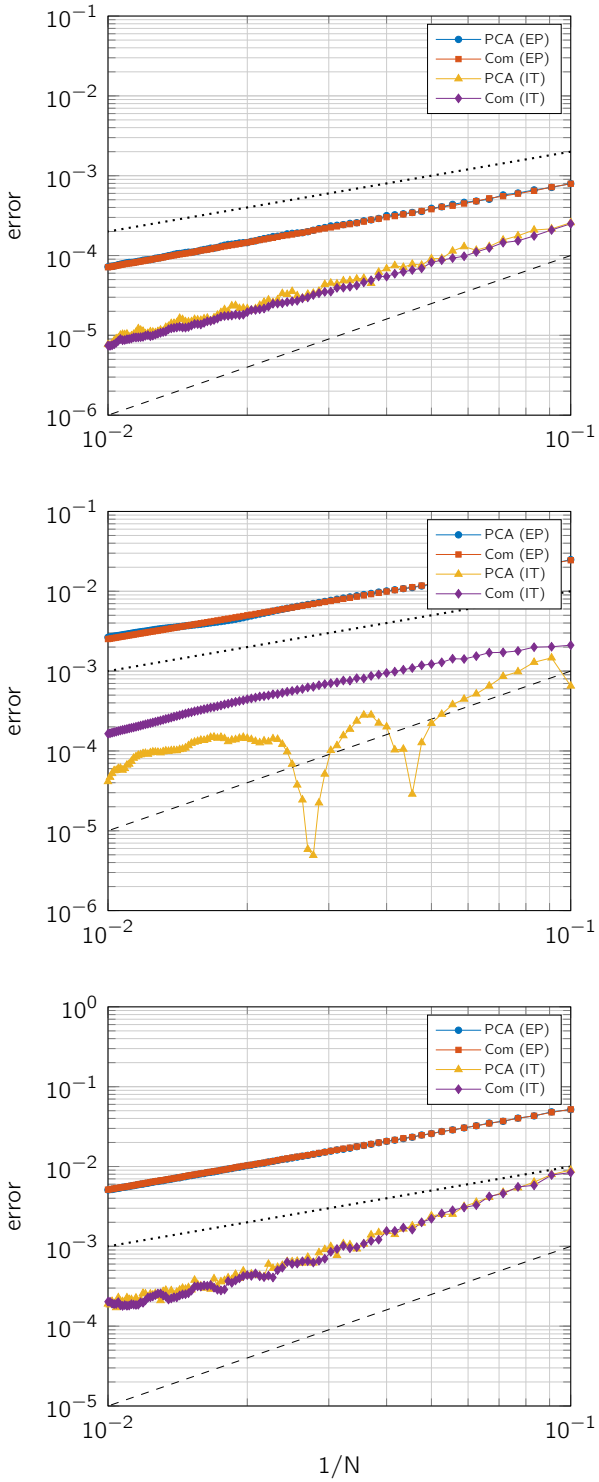


Figure 5.1: Error with respect to the semidiscrete values for $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ in cases A (top), B (middle) and E (bottom) if $m = 100$ for American-style basket options. Two reference lines included for first-order convergence (dotted) and second-order convergence (dashed).

Set	$\widetilde{u}(\boldsymbol{S_0}, T)$	$u^{\text{app}}(\boldsymbol{S_0}, T)$	$u^{\text{low}}(\boldsymbol{S_0}, T)$
A	0.17577	0.17583	0.17577
B	0.83257	0.84125	0.83942
C	0.77065	0.78083	0.77955
D	9.46550	9.46570	9.46523
E	9.10039	9.10128	9.09974
F	8.76358	8.76554	8.76255

Table 5.1: Reference values $\widetilde{u}(\boldsymbol{S_0}, T)$, $u^{\text{app}}(\boldsymbol{S_0}, T)$, $u^{\text{low}}(\boldsymbol{S_0}, T)$ for European-style basket put options for Set A–F.

Set	$\widetilde{u}(\boldsymbol{S_0}, T)$	$u^{\text{app}}(\boldsymbol{S_0}, T)$	$u^{\text{low}}(\boldsymbol{S_0}, T)$
A	0.18110	0.18120	0.18114
B	1.07928	1.08615	1.08431
C	1.01641	1.02435	1.02306
D	9.86176	9.86206	9.86159
E	9.49645	9.49774	9.49620
F	9.15935	9.16219	9.15920

Table 5.2: Reference values $\widetilde{u}(\boldsymbol{S_0}, T)$, $u^{\text{app}}(\boldsymbol{S_0}, T)$, $u^{\text{low}}(\boldsymbol{S_0}, T)$ for American-style basket put options for Set A–F.

Let $\boldsymbol{S_0} = (K, K, \ldots, K)^{\mathsf{T}}$ as above. Table 5.1 displays our reference values for the PCA-based and comonotonic approximations $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$, respectively, as well as the lower bound $u^{\text{low}}(\boldsymbol{S_0}, T)$ for the European-style basket put option. These values have been obtained by applying the PDE discretization from Section 3.3 with $m = N = 1000$ spatial and temporal grid points. Clearly, the positive result holds that, for each given set, the two approximations and the lower bound lie close to each other.

Similarly, Table 5.2 shows our reference values for $\widetilde{u}(\boldsymbol{S_0}, T)$, $u^{\text{app}}(\boldsymbol{S_0}, T)$, $u^{\text{low}}(\boldsymbol{S_0}, T)$ for the American-style basket put option. These values have been obtained by applying the PDCP discretization from Section 5.3 and $m = N = 1000$. We find the favourable result that also in the American case, for each given set, the PCA-based and comonotonic approximations lie close to each other. Recall that, at present, it is not clear whether $u^{\text{low}}(\boldsymbol{S_0}, T)$ forms an actual lower bound in this case.

We next study, for European- and American-style basket put options and Sets A–F, the absolute error in the discretization described in Sections 3.3 and 5.3 of the PCA-based and comonotonic approximations $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ in function of $m = N = 10, 11, 12, \ldots, 100$. To determine the error of the discretization for the PCA-based and comonotonic approximations, the corresponding reference values from Tables 5.1 and 5.2 are used.

Figures 5.2 and 5.3 display for Sets A, B, C and D, E, F, respectively, the absolute error in the discretization of $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ versus $1/m$, where the left column concerns the European-style option and the right column the American-style option.

As a main observation, Figures 5.2 and 5.3 clearly indicate (near) second-order convergence

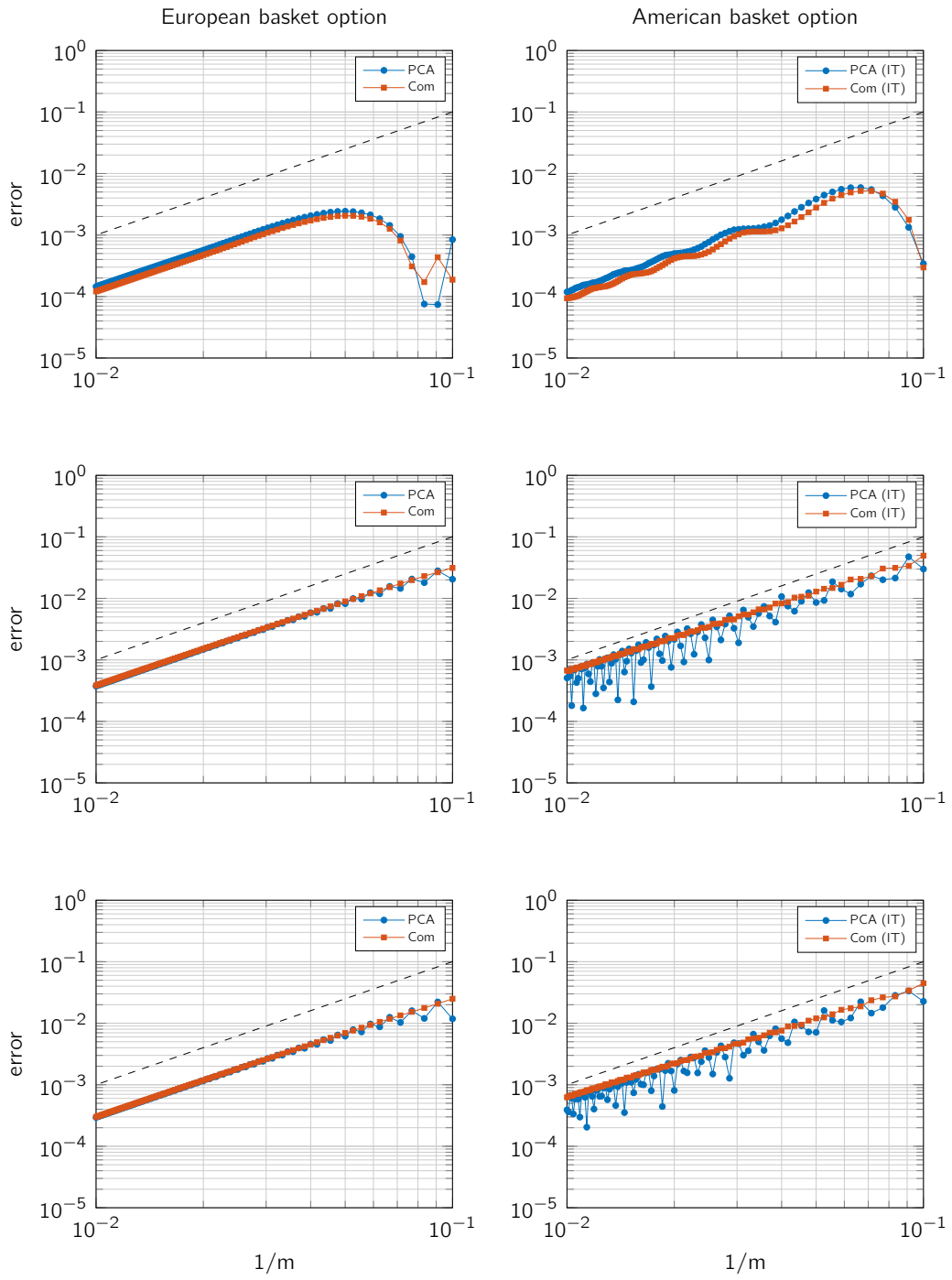


Figure 5.2: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ in cases A (top), B (middle) and C (bottom). Left: European-style basket option. Right: American-style basket option. Reference line (dashed) included for second-order convergence.

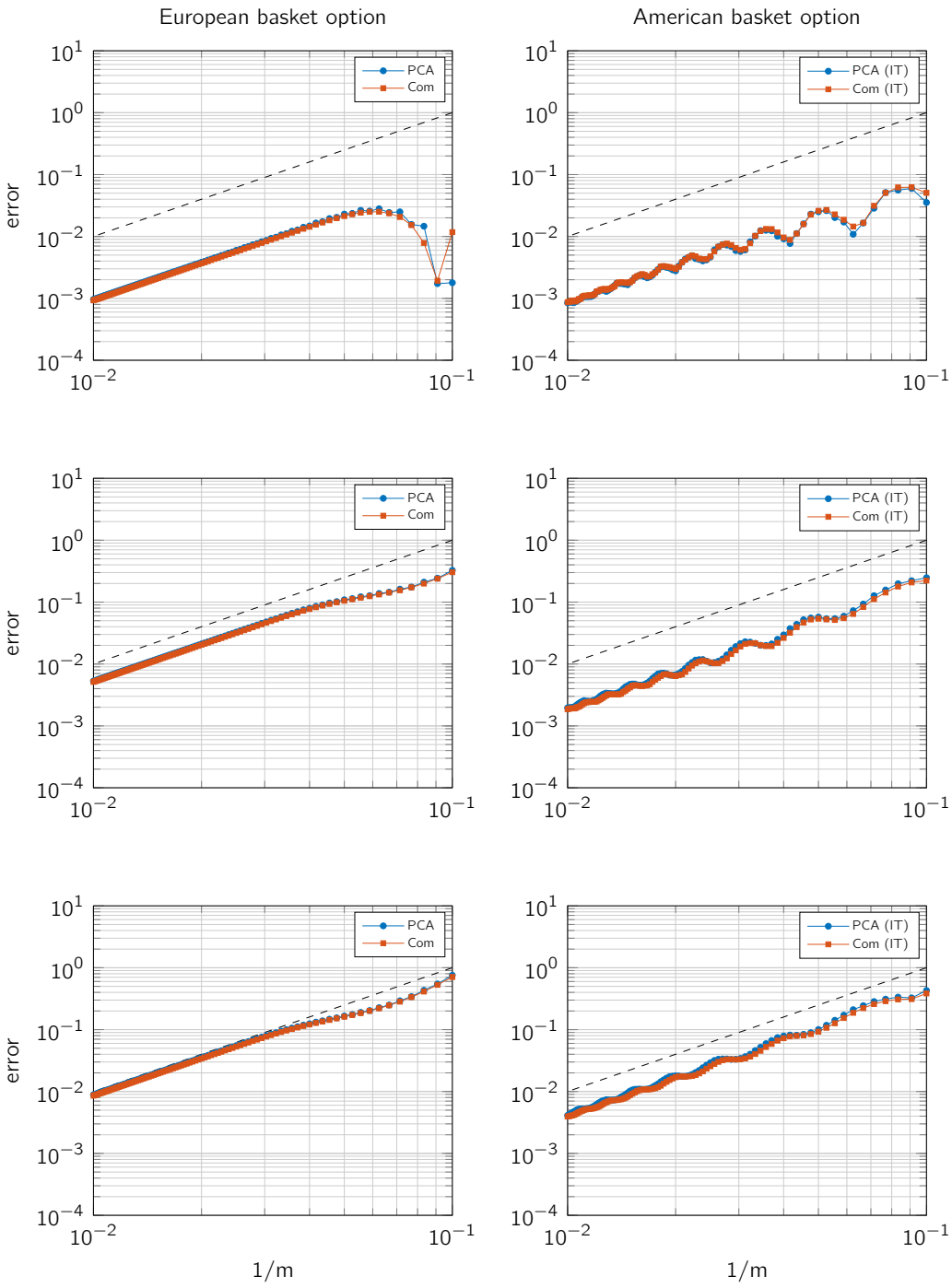


Figure 5.3: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ in cases D (top), E (middle) and F (bottom). Left: European-style basket option. Right: American-style basket option. Reference line (dashed) included for second-order convergence.

of the discretization error in all cases, that is, for all Sets A–F, for both the European- and American-style basket options, and for both the PCA-based and comonotonic approximations. This is a very favourable result. Additional experiments indicate that the error stems essentially from the spatial discretization (and not the temporal discretization).

For the European-style option and Sets A and D, we remark that the error drop in the (less important) region $m \leq 20$ corresponds to a change of sign. Besides this, in the case of the European-style basket option, the behaviour of the discretization error is always seen to be regular.

For the American-style option, it is found that the discretization error often behaves somewhat less regular, with oscillations occurring. A similar phenomenon has also been observed and studied for Bermudan-style basket options (see Section 4.4 or [41]) and is attributed to the spatial nonsmoothness of the exact option value function at the early exercise boundary.

In the following we consider the second part of experiments and choose parameter sets inspired by those from [29]. Here a basket put option with $d = 8$ equally weighted underlying assets is taken and $\boldsymbol{S_0} = (40, 40, \ldots, 40)^\mathsf{T}$. Next, the strike $K \in \{35, 40, 45\}$ and the maturity time $T \in \{0.5, 1, 2\}$. For the interest rate we choose[1] $r = 0.05$. Finally, the volatilities are given by

$$\boldsymbol{\sigma} = (\sigma_i)_{i=1}^8 = \begin{pmatrix} \sigma_1 & 0.6 & 0.1 & 0.9 & 0.3 & 0.7 & 0.8 & 0.2 \end{pmatrix}$$

with $\sigma_1 \in \{0.3, 0.9\}$. We select correlation $\rho_{ij} = 0.8$ for all $i \neq j$. Then, for the pertinent two covariance matrices, the first eigenvalue is dominant. In particular, there holds

$$\begin{aligned} \sigma_1 = 0.3: &\qquad \boldsymbol{\lambda} = (\lambda_i)_{i=1}^8 = \begin{pmatrix} 2.1398 & 0.1461 & 0.1101 & 0.0796 & \ldots \end{pmatrix}, \\ \sigma_1 = 0.9: &\qquad \boldsymbol{\lambda} = (\lambda_i)_{i=1}^8 = \begin{pmatrix} 2.7299 & 0.1620 & 0.1396 & 0.1076 & \ldots \end{pmatrix}. \end{aligned}$$

Further, the relevant matrices of eigenvectors $\boldsymbol{Q}$ satisfy the assumption as stated in Assumption 1.

Tables 5.3 and 5.4 show our reference values for $\widetilde{u}(\boldsymbol{S_0}, T)$, $u^{\text{app}}(\boldsymbol{S_0}, T)$, $u^{\text{low}}(\boldsymbol{S_0}, T)$ for the European- and American-style basket put option, respectively, which have been obtained in the same way as above. Again, we find the favourable result that, for each given parameter set and each given (European- or American-style) option, these three values lie close to each other.

Figure 5.4 displays, analogously to Figures 5.2 and 5.3, the absolute error in the discretization of $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ for the (representative) three parameter sets given by $T \in \{0.5, 1, 2\}$, $K = 40$, $\sigma_1 = 0.3$. The outcomes again indicate a favourable, second-order convergence result. The regularity of the error behaviour is seen to decrease as the maturity time T increases. We note that for $T = 2$ this behaviour is partly explained from a (near) vanishing error when $m \approx 20$.

[1] This differs from [29] where the rate $r = 0.01$ is taken, but then American-style option values are often close to their European counterpart, which is less interesting.

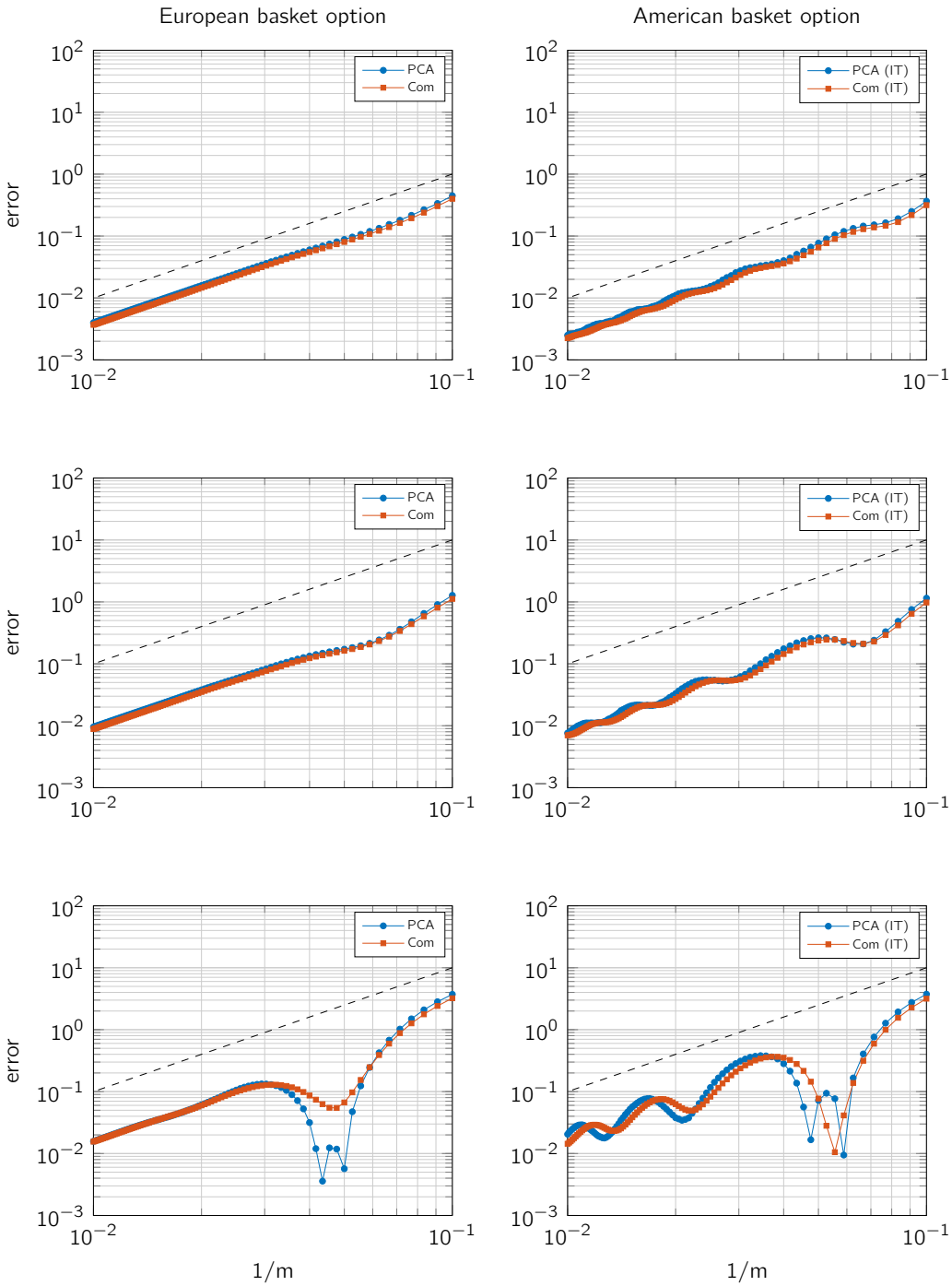


Figure 5.4: Discretization error for $\widetilde{u}(\boldsymbol{S_0}, T)$ and $u^{\text{app}}(\boldsymbol{S_0}, T)$ in cases $T = 0.5$ (top), $T = 1$ (middle) and $T = 2$ (bottom) where $K = 40$, $\sigma_1 = 0.3$. Left: European-style basket option. Right: American-style basket option. Reference line (dashed) included for second-order convergence.

T	K	σ_1	$\widetilde{u}(\boldsymbol{S_0}, T)$	$u^{\text{app}}(\boldsymbol{S_0}, T)$	$u^{\text{low}}(\boldsymbol{S_0}, T)$
0.5	35	0.3	2.13020	2.13271	2.12954
		0.9	2.74982	2.75307	2.74963
	40	0.3	4.40336	4.40715	4.40328
		0.9	5.14582	5.15003	5.14595
	45	0.3	7.45442	7.45827	7.45427
		0.9	8.21316	8.21738	8.21313
1	35	0.3	3.35805	3.36599	3.35620
		0.9	4.23834	4.24750	4.23731
	40	0.3	5.78199	5.79261	5.78114
		0.9	6.79656	6.80770	6.79599
	45	0.3	8.75406	8.76551	8.75329
		0.9	9.82315	9.83486	9.82235
2	35	0.3	4.71159	4.73545	4.70532
		0.9	5.89254	5.91682	5.88742
	40	0.3	7.20593	7.23607	7.20149
		0.9	8.54494	8.57378	8.54048
	45	0.3	10.08246	10.11611	10.07862
		0.9	11.51843	11.54974	11.51371

Table 5.3: Reference values $\widetilde{u}(\boldsymbol{S_0}, T)$, $u^{\text{app}}(\boldsymbol{S_0}, T)$, $u^{\text{low}}(\boldsymbol{S_0}, T)$ for European-style basket put options for alternative testset.

5.6 Conclusions

The valuation of American-style basket options via d-dimensional PDCPs constitutes a notoriously challenging task whenever the number of assets d is medium or large. In this chapter, we have studied an extension of the PCA-based approach by Reisinger and Wittum [71] to valuate American-style basket options. This approximation approach is highly effective, as the numerical solution of only a limited number of low-dimensional PDCPs is required. In addition, we have considered the comonotonic approach, which was developed for basket options notably in [9, 10, 15, 16, 17, 18, 46, 91]. We have studied the comonotonic approximation formula for American-style basket option values recently examined in Hanbali and Linders [29]. The comonotonic approach is also highly effective, since it requires the numerical solution of just two one-dimensional PDCPs. In this chapter these two, different but related, approaches are jointly investigated.

For the discretization of the pertinent PDCPs, we apply finite differences on a nonuniform spatial grid followed by the Brian and Douglas ADI scheme on a uniform temporal grid and selected the Ikonen–Toivanen (IT) technique [28, 37, 38] to efficiently handle the complementarity problem in each time step.

As a first main result, we find in ample numerical experiments that the PCA-based and comonotonic approaches always yield approximations to the value of an American-style (as well as European-style) basket option that lie close to each other.

T	K	σ_1	$\widetilde{u}(\boldsymbol{S_0}, T)$	$u^{\text{app}}(\boldsymbol{S_0}, T)$	$u^{\text{low}}(\boldsymbol{S_0}, T)$
0.5	35	0.3	2.17006	2.17293	2.16973
		0.9	2.79440	2.79840	2.79494
	40	0.3	4.50018	4.50506	4.50118
		0.9	5.24177	5.24795	5.24387
	45	0.3	7.64424	7.65063	7.64670
		0.9	8.38729	8.39562	8.39142
1	35	0.3	3.48012	3.48874	3.47879
		0.9	4.37236	4.38280	4.37246
	40	0.3	6.01652	6.02870	6.01717
		0.9	7.03281	7.04676	7.03498
	45	0.3	9.14612	9.16072	9.14867
		0.9	10.19561	10.21256	10.20013
2	35	0.3	5.06452	5.08982	5.05865
		0.9	6.27930	6.30536	6.27500
	40	0.3	7.78521	7.81748	7.78222
		0.9	9.14045	9.17258	9.13855
	45	0.3	10.94634	10.98327	10.94585
		0.9	12.36710	12.40399	12.36770

Table 5.4: Reference values $\widetilde{u}(\boldsymbol{S_0}, T)$, $u^{\text{app}}(\boldsymbol{S_0}, T)$, $u^{\text{low}}(\boldsymbol{S_0}, T)$ for American-style basket put options for alternative testset.

As a next main result, we observe near second-order convergence of the discretization error in all numerical experiments for both the PCA-based and comonotonic approaches for American-style (as well as European-style) basket options.

At this moment it is still open which (if any) of the two approaches, PCA-based or comonotonic, is to be preferred for the approximate valuation of American-style basket options on $d \geq 5$ assets. In particular, whereas in our experiments the two approaches always define approximations that lie close to each other, it is not clear at present which approach (if any) generally yields the smallest error with respect to the exact option value.The comonotonic approach requires less computational work than the PCA-based approach, but both are computationally cheap.

A further investigation into the PCA-based and comonotonic approaches, both experimental and analytical, will be the subject of future research. This concerns the open question above as well as their fundamental properties, such as convergence, and their range of applications.

Chapter 6

Approximation of the Greeks

Chapter summary:

In this chapter we study the principal component analysis (PCA) based approach introduced by Reisinger & Wittum [71] for the approximation of the Greeks for European-, Bermudan- and American-style basket option values via partial differential equations (PDEs) or partial differential complementarity problems (PDCPs). This highly efficient approximation approach requires the solution of only a limited number of low-dimensional PDEs complemented with optimal exercise conditions.

We discuss two versions of a PCA-based approximation approach to approximate the Deltas. One of these approaches can also be extended to approximate the Gammas. The first PCA-based approximation approach uses terms that are already computed for option valuation. The second PCA-based approximation approach is inspired by pathwise derivatives [5], a concept well-known to estimate the Greeks from Monte Carlo simulation.

Numerical examples illustrate the convergence behaviour of the error for the considered methods. Similar to the PCA-based approximation approaches to valuate options some irregularities in the convergence of the discretization error are visible, but overall again a nearly second-order convergence behaviour is found.

6.1 Introduction

This chapter deals with the approximation of the Greeks for European-, Bermudan- and American-style basket options. Besides the valuation of the fair value of an option, in financial practice also the Greeks are quantities of main interest. The Greeks describe the sensitivity of the option value to a change in one of the underlying financial parameters. For example the Deltas (denoted by Δ) and Gammas (denoted by Γ) are important Greeks and

can be seen as partial derivatives of the fair option value u with respect the underlying asset value:

$$\text{Delta (for asset } k\text{)}: \ \Delta_k = \frac{\partial u}{\partial s_k},$$
$$\text{Gamma (for assets } k \text{ and } l\text{)}: \ \Gamma_{kl} = \frac{\partial^2 u}{\partial s_k \partial s_l}.$$

As mentioned in the previous chapters, European-, Bermudan- and American-style basket options constitute a popular type of financial derivatives and possess a payoff depending on a weighted average of different assets. In general, exact valuation formulas for such options are not available in the literature in semi-closed analytic form. Moreover, computing the Greeks for such type of options can become even more complicated. Thus the development and analysis of efficient approximation methods for their fair values and their Greeks is of much importance.

In this chapter, we consider the numerical approximation of the Greeks for European-, Bermudan and American-style basket options through partial differential equations (PDEs) or partial differential complementarity problems (PDCPs). If d denotes the number of different assets in the basket, then the pertinent PDE is d-dimensional. In this chapter, we are interested in the situation where d is medium or large, say $d \geq 5$. It is well-known that this renders the application of standard discretization methods for PDEs impractical, due to the curse of dimensionality.

For the valuation of these styles of basket options, an effective approach has been introduced by Reisinger and Wittum [71] and next studied in, e.g., Reisinger and Wissmann [68, 69, 70] and in our recent papers [39, 41]. This approach is based on principal component analysis (PCA) and yields an approximation formula for the fair value of the basket option that requires the solution of a limited number of only low-dimensional PDEs.

In this chapter this PCA-based approximation approach for the fair value of an option is also used to numerically approximate the Greeks for high-dimensional European-, Bermudan and American-style basket options under the Black–Scholes model, where we mainly focus on approximating the Deltas for these type of options.

The outline of this chapter is as follows. In Section 6.2 the computed derivatives (i.e. Greeks) of solutions to PDEs are used to approximate the Greeks, one of the advantages of PDE-based methods. We exploit this feature and show that from the PCA-based approximation approach for the fair value of the option also the Deltas of that option can be readily approximated. This leads to the first PCA-based version to approximate the Deltas. With the cost of solving some additional two-dimensional PDEs this version can be extended to approximate also the Gammas. Inspired by the pathwise derivative method [5], as widely used in Monte Carlo simulation to estimate Greeks, in Section 6.3 a second PCA-based version to approximate the Greeks is derived similar to the pathwise derivative method. In Section 6.4 satisfactory numerical results for approximating the Greeks with both versions for European-, Bermudan- and American-style basket options are obtained. The results for the Deltas are also compared with results obtained using (Least Squares) Monte Carlo simulation by Longstaff–Schwartz approach. In Section 6.5 conclusions and a discussion are given.

6.2 PCA-based approximation of Greeks (version 1)

In the Black–Scholes PDE (3.2) the Deltas and Gammas appear already as terms in that PDE. In general, this is also one of the advantages of PDE-based methods for approximating the Greeks; the Deltas and Gammas can be obtained with almost no additional computational costs in addition to valuation of the option.

Also for approximation of the Greeks one can use the PCA-based approximation approach for valuation of options such that only numerical solutions to low-dimensional PDEs are used. Recall, the PCA-based approximation of the fair value of a European-style basket option is given by $\widetilde{w}$ in (3.23) or

$$w(\boldsymbol{y},t) \approx \widetilde{w}(\boldsymbol{y},t) = w^{(1)}(\boldsymbol{y},t) + \sum_{l=2}^{d} \Big[w^{(1,l)}(\boldsymbol{y},t) - w^{(1)}(\boldsymbol{y},t) \Big] \tag{6.1}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$ and $t \in (0,\ T]$.

Here $w^{(1)}(\boldsymbol{y},t)$ and $w^{(1,l)}(\boldsymbol{y},t)$ are solutions to the PDE given by (3.17) or

$$\frac{\partial w}{\partial t}(\boldsymbol{y},t) = \sum_{k=1}^{d} \lambda_k \left[p(y_k) \frac{\partial^2 w}{\partial y_k^2}(\boldsymbol{y},t) + q(y_k) \frac{\partial w}{\partial y_k}(\boldsymbol{y},t) \right] - r w(\boldsymbol{y},t) \tag{6.2}$$

whenever $\boldsymbol{y} \in (0,\ 1)^d$, $t \in (0,\ T]$ and λ_k is set to zero for $k \neq 1$ or $k \notin \{1, l\}$.

6.2.1 PCA-based approximation of Deltas (version 1)

The PCA-based approximation (6.1) contains sufficient terms to approximate the Deltas for the European-style basket option. Indeed, see also (3.6) and (3.16), the Greek Delta-k with $k = 1, 2, \ldots, d$, denoted by $\Delta_k(\boldsymbol{s},t)$, is given by

$$\begin{aligned} \Delta_k(\boldsymbol{s},t) = \frac{\partial u(\boldsymbol{s},t)}{\partial s_k} &= \sum_{i=1}^{d} \frac{\partial w(\boldsymbol{y},t)}{\partial y_i} \frac{\partial y_i}{\partial x_i} \frac{\partial x_i}{\partial s_k} \\ &= \frac{1}{s_k} \sum_{i=1}^{d} q_{ki} \frac{1}{\pi} \frac{1}{x_i^2 + 1} \frac{\partial w(\boldsymbol{y},t)}{\partial y_i}, \end{aligned} \tag{6.3}$$

for $(\boldsymbol{s},t) \in (0,\infty)^d \times (0,T]$, where transformations (3.5) and (3.15) are used.

By construction of the PCA-based approximation (see also Section 3.2), $w^{(1)}(\boldsymbol{y},t)$ satisfies the PDE (6.2) with λ_k being set to zero for all $k \neq 1$, and $w^{(1,l)}(\boldsymbol{y},t)$ satisfies (6.2) with λ_k being set to zero for all $k \notin \{1, l\}$. Thus, solving these PDEs for $w^{(1)}(\boldsymbol{y},t)$ and $w^{(1,l)}(\boldsymbol{y},t)$, one obtains approximations to $\frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1}$, $\frac{\partial w^{(1,l)}(\boldsymbol{y},t)}{\partial y_1}$ and $\frac{\partial w^{(1,l)}(\boldsymbol{y},t)}{\partial y_l}$ with $l = 2, 3, \ldots, d$.

Observe that multiple approximations to $\frac{\partial w(\boldsymbol{y},t)}{\partial y_1}$ are available

$$\begin{aligned} &\frac{\partial w(\boldsymbol{y},t)}{\partial y_1} \approx \frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1}, \\ &\frac{\partial w(\boldsymbol{y},t)}{\partial y_1} \approx \frac{\partial w^{(1,l)}(\boldsymbol{y},t)}{\partial y_1} \quad \text{for } l = 2, 3, \ldots, d. \end{aligned}$$

To approximate $\frac{\partial w(\boldsymbol{y},t)}{\partial y_1}$ one can use a linear combination similar to the PCA-based approximation itself:

$$\frac{\partial w(\boldsymbol{y},t)}{\partial y_1} \approx \frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1} + \sum_{l=2}^{d} \left[\frac{\partial w^{(1,l)}(\boldsymbol{y},t)}{\partial y_1} - \frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1} \right]. \tag{6.4}$$

For the partial derivatives $\frac{\partial w(\boldsymbol{y},t)}{\partial y_l}$ with $l = 2,3,\ldots,d$ only one approximation is known in the PCA-based approximation and that is used to approximate $\frac{\partial w(\boldsymbol{y},t)}{\partial y_l}$:

$$\frac{\partial w(\boldsymbol{y},t)}{\partial y_l} \approx \frac{\partial w^{(1,l)}(\boldsymbol{y},t)}{\partial y_l}, \tag{6.5}$$

with $l = 2,3,\ldots,d$.

Hence, all terms in (6.3) are approximated in terms of the PCA-based approximation and an approximation for $\Delta_k(\boldsymbol{s},t)$ with $k = 1,2,\ldots,d$ is derived.

6.2.2 PCA-based approximation of Gammas (version 1)

To approximate the Greek Gamma-(k,l), denoted by $\Gamma_{kl}(\boldsymbol{s},t)$ with $k,l = 1,2,\ldots,d$, based on the option valuation using the PCA-based approximation approach, similar ideas as for the Deltas can be applied. Indeed, see also (3.9) and (3.16), $\Gamma_{kl}(\boldsymbol{s},t)$ is given by

$$\begin{aligned}
\Gamma_{kl}(\boldsymbol{s},t) &= \frac{\partial^2 u(\boldsymbol{s},t)}{\partial s_k \partial s_l} \\
&= \begin{cases}
\frac{1}{s_k}\frac{1}{s_l} \sum_{i=1}^{d}\sum_{j=1}^{d} q_{ki} q_{lj} \frac{1}{\pi^2} \frac{1}{x_i^2+1} \frac{1}{x_j^2+1} \frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_i \partial y_j}, & \text{for } k \neq l, \\
\frac{1}{s_k^2} \left(\sum_{i=1}^{d}\sum_{j=1}^{d} q_{ki} q_{kj} \frac{1}{\pi^2} \frac{1}{x_i^2+1} \frac{1}{x_j^2+1} \frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_i \partial y_j} - \sum_{i=1}^{d} q_{ki} \frac{1}{\pi} \frac{1}{x_i^2+1} \frac{\partial w(\boldsymbol{y},t)}{\partial y_i} \right) & \text{for } k = l.
\end{cases}
\end{aligned} \tag{6.6}$$

for $(\boldsymbol{s},t) \in (0,\infty)^d \times (0,T]$, where transformations (3.5) and (3.15) are used.

By construction of the PCA-based approximation, $w^{(1)}(\boldsymbol{y},t)$ satisfies PDE (6.2) with λ_i being set to zero for all $i \neq 1$, and $w^{(1,j)}(\boldsymbol{y},t)$ satisfies (6.2) with λ_i being set to zero for all $i \notin \{1,j\}$. Solving these PDEs for $w^{(1)}(\boldsymbol{y},t)$ and $w^{(1,j)}(\boldsymbol{y},t)$, yields approximations to $\frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1}$, $\frac{\partial^2 w^{(1)}(\boldsymbol{y},t)}{\partial y_1^2}$, $\frac{\partial w^{(1,j)}(\boldsymbol{y},t)}{\partial y_1}$, $\frac{\partial^2 w^{(1,j)}(\boldsymbol{y},t)}{\partial y_1^2}$, $\frac{\partial w^{(1,j)}(\boldsymbol{y},t)}{\partial y_j}$ and $\frac{\partial^2 w^{(1,j)}(\boldsymbol{y},t)}{\partial y_j^2}$. with $j = 2,3,\ldots,d$.

Multiple approximations to $\frac{\partial w(\boldsymbol{y},t)}{\partial y_1}$ and $\frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_1^2}$ are available and a linear combination similar to the PCA-based approximation is used to approximate these terms

$$\begin{aligned}
\frac{\partial w(\boldsymbol{y},t)}{\partial y_1} &\approx \frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1} + \sum_{j=2}^{d} \left[\frac{\partial w^{(1,j)}(\boldsymbol{y},t)}{\partial y_1} - \frac{\partial w^{(1)}(\boldsymbol{y},t)}{\partial y_1} \right], \\
\frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_1^2} &\approx \frac{\partial^2 w^{(1)}(\boldsymbol{y},t)}{\partial y_1^2} + \sum_{j=2}^{d} \left[\frac{\partial^2 w^{(1,j)}(\boldsymbol{y},t)}{\partial y_1^2} - \frac{\partial^2 w^{(1)}(\boldsymbol{y},t)}{\partial y_1^2} \right].
\end{aligned} \tag{6.7}$$

For the partial derivatives $\frac{\partial w(\boldsymbol{y},t)}{\partial y_j}$ and $\frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_j^2}$ with $j = 2,3,\ldots,d$ just one approximation is known and used to approximate the appropriate derivatives

$$\begin{aligned} \frac{\partial w(\boldsymbol{y},t)}{\partial y_j} &\approx \frac{\partial w^{(1,j)}(\boldsymbol{y},t)}{\partial y_j}, \\ \frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_j^2} &\approx \frac{\partial^2 w^{(1,j)}(\boldsymbol{y},t)}{\partial y_j^2}, \end{aligned} \tag{6.8}$$

with $j = 2,3,\ldots,d$.

In contrast to approximation of the Deltas in the previous section for the Gammas some necessary terms are not approximated in the PCA-based approximation approach for the option value. Indeed, for approximation of the Gammas also the cross-derivatives $\frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_i \partial y_j}$ with $i \neq j$ are needed but unknown from the (first-order) PCA-based approximation as given in (6.1). Instead of a first-order approximation in (3.21) also higher-order approximations could be used. This can reduce the error of the PCA-based approximation approach for option valuation but it comes with a significant additional cost of solving three-dimensional PDEs.

As an alternative to approximating these cross-derivatives one can also solve additional two-dimensional PDEs for $w^{(i,j)}(\boldsymbol{y},t)$ that satisfies (6.2) with λ_n being set to zero for all $n \notin \{i,j\}$. So, with the cost of solving $\frac{(d-1)(d-2)}{2}$ additional two-dimensional PDEs (where d is the number of assets in the basket option) it is possible to approximate

$$\frac{\partial^2 w(\boldsymbol{y},t)}{\partial y_i \partial y_j} \approx \frac{\partial^2 w^{(i,j)}(\boldsymbol{y},t)}{\partial y_i \partial y_j}, \tag{6.9}$$

and approximation the Gammas $\Gamma_{kl}(\boldsymbol{s},t)$ for $k,l = 1,2,\ldots,d$ using (6.6).

Remark 1. For Bermudan- and American-style basket options a similar approximation approach for the Greeks can be derived as already discussed for European-style basket options. For a detailed discussion about valuation of Bermudan- and American-style basket options we refer to Chapters 4 and 5. For the valuation of Bermudan-style basket options the optimal exercise condition (4.11) has to be implemented and for the valuation of American-style basket options the PDE as given in (5.4), originally in (3.17), changes to a PDCP as given in (5.7).

6.3 PCA-based approximation of Greeks (version 2)

Inspired by the pathwise derivative method [5], which is well-known for estimating the Greeks using Monte Carlo simulation, one can formulate an alternative PCA-based approach to approximate the Deltas by differentiating the PDE (3.2) and initial condition (3.3) with respect to an s_k, with $k = 1,2,\ldots,d$.

Let us consider a European-style basket option and differentiate PDE (3.2) for option value

u with respect to s_k, with $k = 1, 2, \ldots, d$. This yields a new PDE for Delta-k:

$$\begin{aligned}\frac{\partial \Delta_k}{\partial t}(\boldsymbol{s}, t) &= \frac{\partial}{\partial s_k}\left(\frac{\partial u}{\partial t}(\boldsymbol{s}, t)\right) \\ &= \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sigma_i\sigma_j\rho_{ij}s_i s_j\frac{\partial^3 u}{\partial s_i \partial s_j \partial s_k}(\boldsymbol{s}, t) + \sum_{i=1}^{d}\sigma_i\sigma_k\rho_{ik}s_i\frac{\partial^2 u}{\partial s_i \partial s_k}(\boldsymbol{s}, t) + \sum_{i=1}^{d} r s_i\frac{\partial^2 u}{\partial s_i \partial s_k}(\boldsymbol{s}, t).\end{aligned}$$

for $(\boldsymbol{s}, t) \in (0, \infty)^d \times (0, T]$.

Rearranging terms yields the following PDE for Delta-k:

$$\frac{\partial \Delta_k}{\partial t}(\boldsymbol{s}, t) = \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sigma_i\sigma_j\rho_{ij}s_i s_j\frac{\partial^2 \Delta_k}{\partial s_i \partial s_j}(\boldsymbol{s}, t) + \sum_{i=1}^{d}(\sigma_i\sigma_k\rho_{ik} + r)\, s_i\frac{\partial \Delta_k}{\partial s_i}(\boldsymbol{s}, t). \tag{6.10}$$

for $(\boldsymbol{s}, t) \in (0, \infty)^d \times (0, T]$.

For the initial condition also the payoff function (3.4) has to be differentiated with respect to s_k, so

$$\Delta_k(\boldsymbol{s}, 0) = \frac{\partial \phi}{\partial s_k}(\boldsymbol{s}) = \begin{cases} -\omega_k & \text{if } \phi(\boldsymbol{s}) > 0, \\ 0 & \text{if } \phi(\boldsymbol{s}) = 0, \end{cases} \tag{6.11}$$

whenever $\boldsymbol{s} \in (0, \infty)^d$.

This initial condition is discontinuous in a $(d-1)$-dimensional space where $K = \sum_{i=1}^{d}\omega_i s_i$. Besides that, observe that this PDE with initial condition for $\Delta_k(\boldsymbol{s}, t)$ has a similar form as the PDE for $u(\boldsymbol{s}, t)$. It is again a convection-diffusion equation with exactly the same structure in the diffusion term.

6.3.1 PCA-based approximation of Deltas (version 2)

Because PDE (6.10) for Delta-k has exactly the same structure in the diffusion term as PDE (3.2) for the option value, a similar PCA-based approximation approach as discussed in Section 3.2 will exist for this pathwise derivative-inspired approach of approximating Delta-k.

We will consider the new PCA-based approach to approximate Delta-k obtained from PDE (6.10). Assume that the elementary functions $\ln(\cdot)$, $\exp(\cdot)$, $\tan(\cdot)$, $\arctan(\cdot)$ are taken componentwise whenever their argument is a vector.

Consider the covariance matrix $\boldsymbol{\Sigma} = \left(\Sigma_{ij}\right) \in \mathbb{R}^{d\times d}$ which is elementwise given by $\Sigma_{ij} = \sigma_i\rho_{ij}\sigma_j$ for $i, j = 1, 2, \ldots, d$. The spectral decomposition is given by $\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathsf{T}}$, where $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$ is an orthogonal matrix with eigenvectors of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d) \in \mathbb{R}^{d\times d}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{\Sigma}$.

Similar to Section 3.2, consider the coordinate transformation

$$\boldsymbol{x}(\boldsymbol{s}, t) = \boldsymbol{Q}^{\mathsf{T}}\left(\ln(\boldsymbol{s}/K) - \boldsymbol{b}(t)\right), \tag{6.12}$$

where $\boldsymbol{b}(t) = (b_1(t), b_2(t), \ldots, b_d(t))^{\mathsf{T}}$ with $b_i(t)$ for $1 \le i \le d$ to be determined.

The partial derivatives of the transformation given in (6.12) are given by

$$\frac{\partial x_i}{\partial s_j} = q_{ji}\frac{1}{s_j}$$
$$\frac{\partial x_i}{\partial t} = -\sum_{j=1}^{d} q_{ji} b_i'(t).$$

where $b_i'(t) = \frac{\mathrm{d}b_i}{\mathrm{d}t}(t)$ for a function $b_i(t)$ to be determined for $i = 1, 2, \ldots, d$.

Let the function δ_k be defined by

$$\Delta_k(\boldsymbol{s}, t) = \delta_k(x(\boldsymbol{s}, t), t).$$

Then using the chain rule the first derivative of $\Delta_k(\boldsymbol{s}, t)$ to t can be written as

$$\frac{\partial \Delta_k(\boldsymbol{s}, t)}{\partial t} = \frac{\partial \delta_k(\boldsymbol{x}, t)}{\partial t} - \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial \delta_k(\boldsymbol{x}, t)}{\partial x_i} q_{ji} b_j'(t). \tag{6.13}$$

Next, the first derivative of $\Delta_k(\boldsymbol{s}, t)$ to s_j (with $j = 1, 2, \ldots, d$) is given by

$$\frac{\partial \Delta_k(\boldsymbol{s}, t)}{\partial s_j} = \frac{1}{s_j}\sum_{i=1}^{d} q_{ji}\frac{\partial \delta_k(\boldsymbol{x}, t)}{\partial x_i}. \tag{6.14}$$

Finally, the second derivative of $\Delta_k(\boldsymbol{s}, t)$ to s_i and s_j for $i, j = 1, 2, \ldots, d$ is given by

$$\frac{\partial^2 \Delta_k(\boldsymbol{s}, t)}{\partial s_i \partial s_j} = \begin{cases} \frac{1}{s_i}\frac{1}{s_j}\sum_{l=1}^{d}\sum_{m=1}^{d} q_{il} q_{jm} \frac{\partial^2 \delta_k(\boldsymbol{x}, t)}{\partial x_l \partial x_m}, & \text{for } i \neq j, \\ \frac{1}{s_i^2}\left(\sum_{l=1}^{d}\sum_{m=1}^{d} q_{il} q_{im} \frac{\partial^2 \delta_k(\boldsymbol{x}, t)}{\partial x_l \partial x_m} - \sum_{m=1}^{d} q_{im} \frac{\partial \delta_k(\boldsymbol{x}, t)}{\partial x_m}\right) & \text{for } i = j. \end{cases} \tag{6.15}$$

An easy calculation yields that δ_k satisfies

$$\frac{\partial \delta_k}{\partial t}(\boldsymbol{x}, t) = \frac{1}{2}\sum_{l=1}^{d} \lambda_l \frac{\partial^2 \delta_k}{\partial {x_l}^2}(\boldsymbol{x}, t) + \sum_{i,l=1}^{d} \left(\sigma_i \sigma_k \rho_{ik} + r - \tfrac{1}{2}\sigma_i^2 + b_i'(t)\right) q_{il} \frac{\partial \delta_k}{\partial x_l}(\boldsymbol{x}, t). \tag{6.16}$$

whenever $\boldsymbol{x} \in \mathbb{R}^d$, $t \in (0, T]$.

Thus, there is still a degree of freedom left, which can be used to reduce this PDE to a pure diffusion problem. Choose $b_i(0) = 0$, which leads with the ODE for $b_i(t)$

$$b_i'(t) = \tfrac{1}{2}\sigma_i^2 - r - \sigma_i \sigma_k \rho_{ik} \tag{6.17}$$

to a simple expression for $\boldsymbol{b}(t)$, which is elementwise given by

$$b_i(t) = \left(\tfrac{1}{2}\sigma_i^2 - r - \sigma_i \sigma_k \rho_{ik}\right) t.$$

This leads to a pure diffusion equation for δ_k, without mixed derivative terms:

$$\frac{\partial \delta_k}{\partial t}(\boldsymbol{x}, t) = \frac{1}{2}\sum_{l=1}^{d} \lambda_l \frac{\partial^2 \delta_k}{\partial x_l^2}(\boldsymbol{x}, t), \tag{6.18}$$

whenever $\boldsymbol{x} \in \mathbb{R}^d,\ t \in (0, T]$.

It is convenient to perform a second coordinate transformation, which maps the spatial domain $\mathbb{R}^d$ onto the d-dimensional open unit cube $D = (0,\ 1)^d$,

$$\boldsymbol{y}(\boldsymbol{x}) = \tfrac{1}{\pi} \arctan(\boldsymbol{x}) + \tfrac{1}{2}. \tag{6.19}$$

This last transformation and the PCA-based approximation for Delta-k are done is exactly the same way as described in Section 3.2.2.

Remark 2. In principle the same technique based on the pathwise method as discussed for Delta-k can also be applied to derive a new PDE with initial condition for Gamma-(k, l), with $k, l = 1, 2, \ldots, d$. But, because the payoff function (3.4) is not twice differentiable it is not clear how to define an effective initial value problem for Gamma-(k, l).

Remark 3. Also the approximation of Delta-k for Bermudan- and American-style basket options using this pathwise derivative-based approach is not clear. For example, it is not clear how to effectively implement the optimal exercise condition (4.11) in approximating the Deltas for Bermudan-style basket options. Also for American-style basket options it is unclear if a PDCP for the Deltas is valid.

6.4 Numerical experiments

In this section some numerical examples demonstrate the potential of the two different versions of the PCA-based approaches to approximate the Greeks. The different PCA-based approximations to the Deltas and Gammas are denoted by $\widetilde{\delta}_k(\boldsymbol{s}, t)$ and $\widetilde{\gamma}_{kl}(\boldsymbol{s}, t)$, respectively.

For a European-style basket option we can compare the approximation of Δ_k (for $k = 1, 2, \ldots, d$) using version 1 and version 2. The obtained reference values for the PCA-based methods are compared with the (Least Squares) Monte Carlo approach where pathwise derivatives are used to estimate the Deltas. For Bermudan- and American-style basket options only version 1 of the PCA-based approximation approach is applicable to approximate the Greeks Δ_k (for $k = 1, 2, \ldots, d$) and Γ_{kl} (for $k, l = 1, 2, \ldots, d$).

In this section a numerical study of the error in the total discretization of the different PCA-based approximations $\widetilde{\delta}_k(\boldsymbol{s}, t)$ and $\widetilde{\gamma}_{kl}(\boldsymbol{s}, t)$ for respectively the Deltas and the Gammas of European-, Bermudan- and American-style basket options is done.

As an example, consider Set A as defined in Appendix A. The reference values are computed using the pertinent versions of the PCA-based approximation approaches for Deltas $\delta_k(\boldsymbol{S_0}, T)$ and Gammas $\gamma_{kl}(\boldsymbol{S_0}, T)$ of the European-, Bermudan and American-style basket put options. Here we choose $\boldsymbol{S_0} = (K, K, \ldots, K)^\mathsf{T}$. These reference values have been obtained by using the PCA-based approximation approach with $m = N = 1000$ spatial and temporal grid points for European- and American-style basket options. For Bermudan-style basket options we use $E = 10$ equidistant exercise times $\tau_i = i\frac{T}{E}$ with $i = 1, 2, \ldots, E$. The number of spatial grid points $m = 1000$ and the number of temporal gridpoints is given by $N = E\lceil m/E \rceil$ for the Bermudan-style option.

For comparison also a Monte Carlo simulation is done to estimate the Deltas of the European-, Bermudan- and American-style basket option via pathwise derivatives. For the estimation of the Deltas for Bermudan- and American-style basket options the Least Squares Monte Carlo method by Longstaff and Schwartz [52] is used. Gobet [24] showed that the concept of pathwise derivatives to estimate the Deltas for European-style options using Monte Carlo methods can be extended to estimate the Deltas for American-style options.

Tables 6.1a, 6.1b and 6.1c show the obtained reference values for the available PCA-based approximations of $\delta_k(\boldsymbol{S_0}, T)$ for Deltas of the European-, Bermudan- and American-style basket put option of Set A. Considering the results for European-style basket options then for all Deltas the positive result holds that the different approximations agree for the first two or three digits. The differences between the approximations is almost always below 1% of the Delta value. For the Bermudan- and American-style basket options the differences between the considered methods increases slightly to 3% and 7% of the Delta value, but also in these cases the different approximations agree for the first digits. We remark that at least the LSMC values may contain a certain bias or error due to regressions that may not be performing well. Similar kind of remarks on the LSMC values are also made by e.g. [29].

Similarly, Tables 6.2a, 6.2b and 6.2c show the obtained reference values for the approximation of $\gamma_{kl}(\boldsymbol{S_0}, T)$ using the PCA-based approximation of the option value for Gammas of the European-, Bermudan- and American-style basket put option of Set A. Due to a lack of good alternative reference values for these Gammas it is hard to make some quantitative statements on the quality of this approximation of the Gammas using the PCA-based approximation. Qualitatively one can observe that with some exceptions all Gammas are positive. In the cases where the Gamma is negative the values are close to zero. Further, in absolute value this is also smaller than the observed differences in approximating the Deltas.

As a second part of this numerical study, consider the absolute error in the PCA-based discretizations $\widetilde{\delta}_k(\boldsymbol{S_0}, T)$ (for $k = 1, 2, \ldots, d$) and $\widetilde{\gamma}_{kl}(\boldsymbol{S_0}, T)$ (for $k, l = 1, 2, \ldots, d$) at the point $\boldsymbol{S_0} = (K, K, \ldots, K)^\mathsf{T}$.

In Figure 6.1 the discretization error with respect to the computed reference values of Set A for the Deltas is shown. Clearly, for the PCA-based approximation approach (version 1) the approximation of Δ_k indicates second-order convergence of the discretization error for European-, Bermudan- and American-style basket options. Furthermore, the convergence behaviour for the Deltas of the European-style basket option is smooth, as expected from the error behaviour for the option valuation of European-style basket options itself. As expected from the discretization errors for valuation of the option the results are less regular for Bermudan- and American-style basket options. But clearly, one observes again nearly second-order convergence of the discretization error.

Finally, the PCA-based approximation approach (version 2) is only applicable for approximating the Deltas of European-style basket options, so no results are available for Bermudan- and American-style basket options. For European-style basket options second order convergence behaviour is observed, although this behaviour is not smooth. This can be explained by the non-smoothness of the initial condition. In this case the initial condition is a step function, which is not continuous anymore. Cell averaging of this initial condition is applied, but due to the lack of continuity this becomes computationally more expensive.

Set A	Version 1	Version 2	Monte Carlo
Δ_1	-0.14145	-0.13999	-0.14008
Δ_2	-0.02166	-0.02271	-0.02273
Δ_3	-0.01987	-0.02049	-0.02051
Δ_4	-0.09462	-0.09395	-0.09401
Δ_5	-0.08389	-0.08403	-0.08407

(a) European-style basket put options using Set A. For the Monte Carlo method $\texttt{Npaths} = 10^7$ is used and the antithetic paths are added.

Set A	Version 1	Version 2	Monte Carlo
Δ_1	-0.14673	–	-0.14605
Δ_2	-0.02308	–	-0.02375
Δ_3	-0.02113	–	-0.02141
Δ_4	-0.09859	–	-0.09826
Δ_5	-0.08740	–	-0.08766

(b) Bermudan-style basket put options. For the Least Squares Monte Carlo method $\texttt{Npaths} = 10^6$ is used and the antithetic paths are added.

Set A	Version 1	Version 2	Monte Carlo
Δ_1	-0.14710	–	-0.15373
Δ_2	-0.02318	–	-0.02501
Δ_3	-0.02121	–	-0.02255
Δ_4	-0.09886	–	-0.10350
Δ_5	-0.08765	–	-0.09222

(c) American-style basket put options. For the Least Squares Monte Carlo method $\texttt{Npaths} = 5 \cdot 10^5$ is used and the antithetic paths are added.

Table 6.1: Reference values $\widetilde{\delta}_k(\boldsymbol{S_0}, T)$ for Delta-k of European-, Bermudan- and American-style basket put options of Set A using PCA-based approximation approach version 1 and version 2.

k \ l	**1**	**2**	**3**	**4**	**5**
1	0.13853	-0.00250	-0.00467	0.11982	0.10806
2	-0.00250	0.00317	0.04323	0.03970	0.04751
3	-0.00467	0.04323	0.04246	0.01480	0.01952
4	0.11982	0.03970	0.01480	0.06145	0.03663
5	0.10806	0.04751	0.01952	0.03663	0.02092

(a) European-style basket put option.

k \ l	**1**	**2**	**3**	**4**	**5**
1	0.14585	0.00028	-0.00155	0.12472	0.11215
2	0.00028	0.00516	0.04398	0.04158	0.04898
3	-0.00155	0.04398	0.04339	0.01664	0.02096
4	0.12472	0.04158	0.01664	0.06675	0.04086
5	0.11215	0.04898	0.02096	0.04086	0.02549

(b) Bermudan-style basket put option.

k \ l	**1**	**2**	**3**	**4**	**5**
1	0.14599	0.00028	-0.00153	0.12475	0.11218
2	0.00028	0.00512	0.04394	0.04159	0.04899
3	-0.00153	0.04394	0.04335	0.01662	0.02093
4	0.12475	0.04159	0.01662	0.06681	0.04090
5	0.11218	0.04899	0.02093	0.04090	0.02555

(c) American-style basket put option.

Table 6.2: Reference values $\widetilde{\gamma}_{kl}(\boldsymbol{S_0}, T)$ for Gamma-(k,l) of European-, Bermudan- and American-style basket put option of Set A using PCA-based approximation approach version 1.

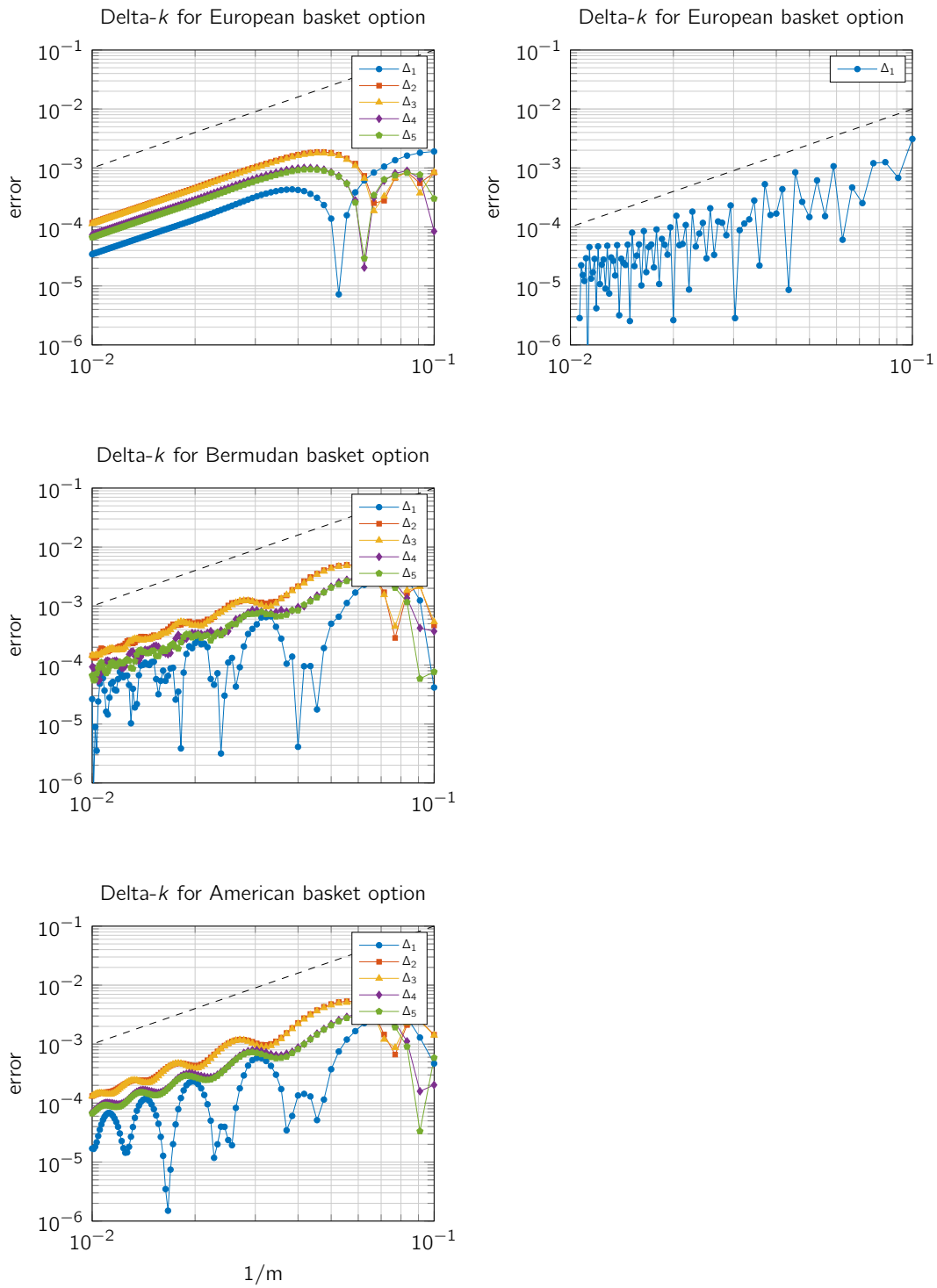


Figure 6.1: Discretization error for PCA-based approximation $\widetilde{\delta}_k(\boldsymbol{S_0}, T)$ in Set A using version 1 (left) or version 2 (right). The Deltas are computed for European- (top), Bermudan- (middle) and American- (bottom) style basket options. Reference line (dashed) included for second-order convergence.

Finally, in Figure 6.2 the discretization error for Δ_k and Γ_{kk} using PCA-based approximation approach version 1 for Set A are shown. Again in all cases nearly second order convergence is observed for both the Deltas as the Gammas. This gives some first indication that approximation of both the Deltas and the Gammas using the PCA-based approximation approach should be possible.

6.5 Conclusions

The approximation of the Greeks, already the Deltas and Gammas of European-, Bermudan- and American-style basket options is a challenging task when the number of assets d in the basket is medium or large. In this chapter we considered some extensions of the PCA-based approximation approach by Reisinger and Wittum [71] to approximation also some Greeks. This PCA-based approximation approach is very effective, because this approximation requires only the numerical solution of a limited number of low-dimensional PDEs (or PDCPs, if an American-style basket option is considered).

We studied this PCA-based approximation approach for the fair value of an option and observed that with minimal additional costs also approximations for the Deltas can be derived. This can be applied to European-, Bermudan- and American-style basket options. When some additional low-dimensional PDEs (or PDCPs, if an American-style basket option is considered) are solved also approximations to the Gammas can be derived.

As an alternative approach a PDE is derived similar to the pathwise derivative method [5] that is widely used in Monte Carlo methods to approximate the Greeks. We observed that this PDE for Deltas and Gammas has a similar form as the Black–Scholes PDE and a similar PCA-based method for these Greeks exists. We remarked that this is currently only applicable for Deltas of European-style basket options and a further investigation is needed to find a similar method for Deltas of Bermudan- and American-style basket options. Also the initial condition for the Gammas is non-trivial which makes this approach also in a current stage not applicable for approximating the Gammas, even not for European-style basket options.

Finally, this chapter gives a numerical proof of principle to use PCA-based approximation approaches also for approximation of the Greeks. A rigorous analysis of the error made in approximation of the Greeks using the PCA-based approximation is also still an open question for further research.

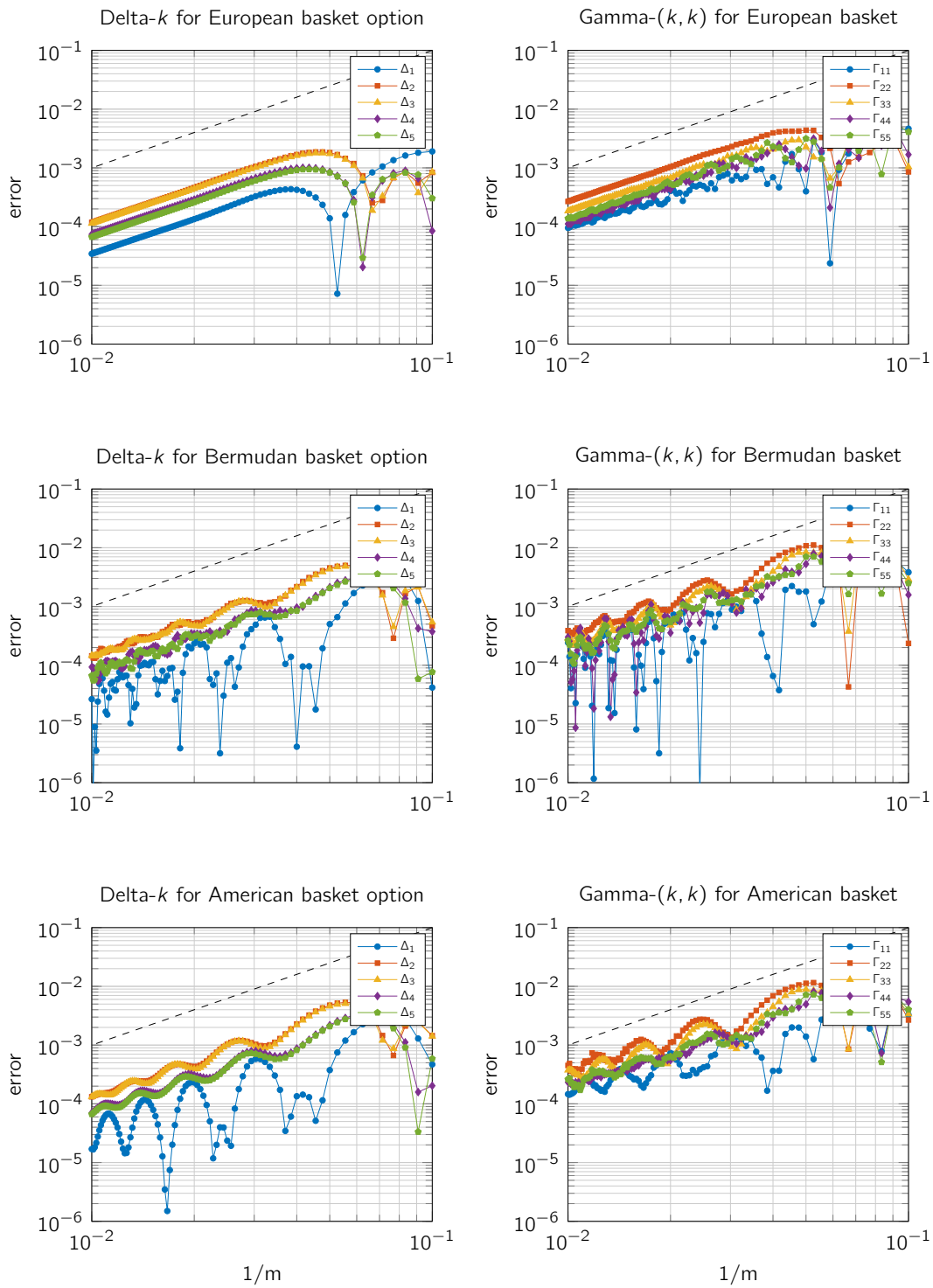
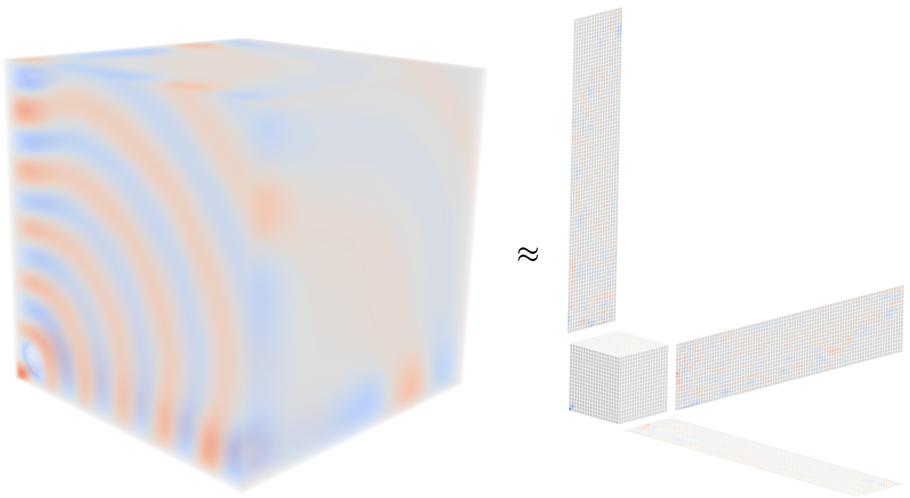


Figure 6.2: Discretization error for PCA-based approximation (version 1) of $\widetilde{\delta}_k(\boldsymbol{S_0}, T)$ (left) and $\widetilde{\gamma}_{kk}(\boldsymbol{S_0}, T)$ (right) in Set A. The Deltas and Gammas are computed for European- (top), Bermudan- (middle) and American- (bottom) style basket options. Reference line (dashed) included for second-order convergence.

Part II

Tensor approximations and scattering problems



Chapter 7

Introduction to high-dimensional data representation using tensors

Chapter summary:

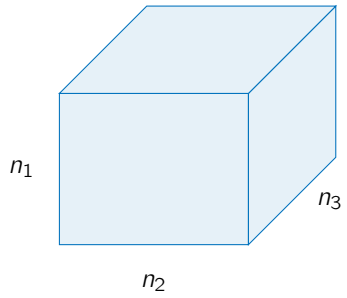
This chapter gives an overview about extending vectors and matrices to high-dimensional arrays and represent this high-dimensional data using tensors. Further, some preliminaries, notation, properties and tensor operations are discussed.

The extension of the singular value decomposition (SVD) for matrices to tensors leads to two classes of tensor decompositions, i.e. the Canonical Polyadic decomposition and the Tucker tensor decomposition. The CP-decomposition constructs a rank-r decomposition for a tensor while the Tucker tensor decomposition constructs orthonormal factor matrices.

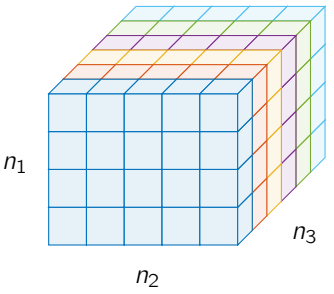
This introduction to tensors is mainly based on the survey paper from Kolda and Bader [14, 48, 49] and further details, results and references about tensors can be found there.

7.1 Introduction

In many applications, tensor representations are used to describe real-valued data and almost all results are presented for real-valued tensors, see for example [48]. The tensors in the applications that we will consider in this second part of the thesis are often complex-valued. Therefore, this introduction is also used to mention the concepts in the context of complex-valued tensors. These extensions are almost always trivial due to the definition of the tensor operations in terms of underlying matrix operations on the tensor representations.

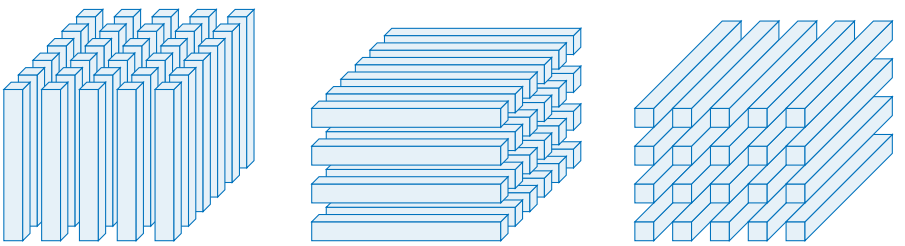


(a) Visualization of a three-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$.



(b) Example color coding for frontal slices of tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$.

Figure 7.1: A visualization of a three-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. A tensor can be seen as a block of data. In the right figure, if each cube represents a number then this tensor has dimensions $n_1 = 4, n_2 = 5$ and $n_3 = 6$.



(a) mode-1 fibers (b) mode-2 fibers (c) mode-3 fibers

Figure 7.2: A visualization of the mode-1, mode-2 and mode-3 fibers of a three-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ where data is stored in respectively columns, rows or tubes.

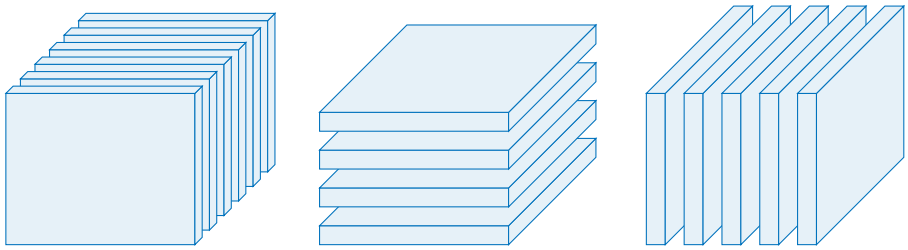
7.2 High-dimensional data representation

A vector $\boldsymbol{v} \in \mathbb{C}^{n_1}$ can be seen as a one-dimensional array where a collection of data is stored in a single column. Further, a matrix $\boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2}$ is a two-dimensional object where a collection of data is stored in rows and columns. A tensor can be seen as further generalization where a collection of data is stored in an high-dimensional object with more than two dimensions.

For example, a three-dimensional tensor[1] $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ as visualized in Figure 7.1 can be seen as a cube of data. The data in this tensor can be stored in columns (mode-1 fibers), rows (mode-2 fibers) or tubes (mode-3 fibers), as visualized in Figure 7.2. Another useful interpretation is to view a tensor as a collection of *slices*, as shown in Figure 7.3.

In general for a d-dimensional tensor $\mathcal{X}$ one has a collection of data that can be stored in *fibers* for each direction, i.e. mode-k fibers for $k = 1, 2, \ldots, d$. The number of dimensions of a tensor is also known as the *order* of a tensor.

[1] In this thesis we will use the convention to write scalars x in a standard lowercase font, vectors $\boldsymbol{v}$ in a bold lowercase font, matrices $\boldsymbol{A}$ in a bold uppercase font and tensors $\mathcal{G}$ in a bold calligraphic uppercase font.



(a) Frontal slices (b) Horizontal slices (c) Lateral slices

Figure 7.3: A visualization of the frontal, horizontal and lateral slices of a three-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$.

Definition 1. *The* inner product *of two equally sized tensors* $\mathcal{X}$ *and* $\mathcal{Y} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ *generalizes the definition for vectors and matrices. The inner product of two tensors is defined as the sum of the elementwise product of the entries, thus*

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} x_{i_1 i_2 \cdots i_d} \overline{y_{i_1 i_2 \cdots i_d}}, \tag{7.1}$$

where $\overline{y}$ *denotes the complex conjugate of* y.

Definition 2. *The* (Frobenius-)norm *of a tensor* $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ *generalizes the Frobenius norm for matrices, thus*

$$\|\mathcal{X}\| := \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} x_{i_1 i_2 \cdots i_d} \overline{x_{i_1 i_2 \cdots i_d}}}. \tag{7.2}$$

Definition 3. *A* d*-dimensional tensor* $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ *is a* rank-one tensor *if it can be written as the outer product of* d *vectors, thus*

$$\mathcal{X} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)} \circ \cdots \circ \boldsymbol{u}^{(d)}, \tag{7.3}$$

where $\circ$ *represents the vector outer product and* $\boldsymbol{u}^{(k)} \in \mathbb{C}^{n_k}$ *for* $k = 1, 2, \ldots, d$. *Thus each element of tensor* $\mathcal{X}$ *can be written as product of the entries of the corresponding vectors*

$$x_{i_1, i_2, \ldots, i_d} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(d)}_{i_d},$$

for all $i_k = 1, 2, \ldots, n_k$ *and* $k = 1, 2, \ldots, d$.

Almost all tensor operations are defined in terms of different kind of matrix products. Especially the following matrix products and notations are used [78]:

Definition 4. *The* Hadamard product, *denoted by* $\boldsymbol{X} \circ \boldsymbol{Y}$, *of two equally sized matrices* $\boldsymbol{X}$ *and* $\boldsymbol{Y} \in \mathbb{C}^{M \times N}$ *is a matrix* $\boldsymbol{Z} \in \mathbb{C}^{M \times N}$, *which is defined by the elementwise product*

$$z_{ij} = x_{ij} y_{ij},$$

where $i = 1, 2, \ldots, M$ *and* $j = 1, 2, \ldots, N$.

Definition 5. *The* Kronecker product, *denoted by* $\boldsymbol{X} \otimes \boldsymbol{Y}$, *of two matrices* $\boldsymbol{X} \in \mathbb{C}^{K \times L}$ *and* $\boldsymbol{Y} \in \mathbb{C}^{M \times N}$ *is a matrix* $\boldsymbol{Z} \in \mathbb{C}^{KM \times LN}$ *which is defined by*

$$\begin{aligned}\boldsymbol{Z} = \boldsymbol{X} \otimes \boldsymbol{Y} &= \begin{bmatrix} x_{11}\boldsymbol{Y} & x_{12}\boldsymbol{Y} & \cdots & x_{1L}\boldsymbol{Y} \\ x_{21}\boldsymbol{Y} & x_{22}\boldsymbol{Y} & \cdots & x_{2L}\boldsymbol{Y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{K1}\boldsymbol{Y} & x_{K2}\boldsymbol{Y} & \cdots & x_{KL}\boldsymbol{Y} \end{bmatrix}, \\ &= \begin{bmatrix} \boldsymbol{x}_1 \otimes \boldsymbol{y}_1 & \boldsymbol{x}_1 \otimes \boldsymbol{y}_2 & \boldsymbol{x}_1 \otimes \boldsymbol{y}_3 & \cdots & \boldsymbol{x}_2 \otimes \boldsymbol{y}_1 & \boldsymbol{x}_2 \otimes \boldsymbol{y}_2 & \cdots & \boldsymbol{x}_L \otimes \boldsymbol{y}_{N-1} & \boldsymbol{x}_L \otimes \boldsymbol{y}_N \end{bmatrix}.\end{aligned}$$

Definition 6. *The* Khatri-Rao product, *denoted by* $\boldsymbol{X} \odot \boldsymbol{Y}$, *of two matrices* $\boldsymbol{X} \in \mathbb{C}^{K \times N}$ *and* $\boldsymbol{Y} \in \mathbb{C}^{M \times N}$ *is a matrix* $\boldsymbol{Z} \in \mathbb{C}^{KM \times N}$ *which is defined by the matching column Kronecker product*

$$\boldsymbol{Z} = \boldsymbol{X} \odot \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{x}_1 \otimes \boldsymbol{y}_1 & \boldsymbol{x}_2 \otimes \boldsymbol{y}_2 & \cdots & \boldsymbol{x}_N \otimes \boldsymbol{y}_N \end{bmatrix}.$$

Some useful properties of Hadamard, Kronecker, and Khatri-Rao products and their pseudo-inverse [25] $\boldsymbol{A}^\dagger$ of matrix $\boldsymbol{A}$ are given by [78]:

$$\begin{aligned}(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) &= (\boldsymbol{AC} \otimes \boldsymbol{BD}), \\ (\boldsymbol{A} \otimes \boldsymbol{B})^{\mathsf{T}} &= \boldsymbol{A}^{\mathsf{T}} \otimes \boldsymbol{B}^{\mathsf{T}}, \\ (\boldsymbol{A} \otimes \boldsymbol{B})^{\dagger} &= \boldsymbol{A}^{\dagger} \otimes \boldsymbol{B}^{\dagger}, \\ (\boldsymbol{A} \odot \boldsymbol{B})^{\mathsf{T}}(\boldsymbol{A} \odot \boldsymbol{B}) &= \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right) \circ \left(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B}\right), \\ (\boldsymbol{A} \odot \boldsymbol{B})^{\dagger} &= \left[\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right) \circ \left(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B}\right)\right]^{\dagger} (\boldsymbol{A} \odot \boldsymbol{B})^{\mathsf{T}}.\end{aligned}$$

7.2.1 Tensor unfoldings to matrices

Tensors can be *unfolded* to matrices, also called *matricization* or *flattening*, where the elements of a tensor are reordered to a certain matrix. For example, the mode-k unfolding of a tensor stores the mode-k fibers of that tensor into columns of a matrix. Given a tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ the k-th unfolding of that tensor is a matrix denoted by

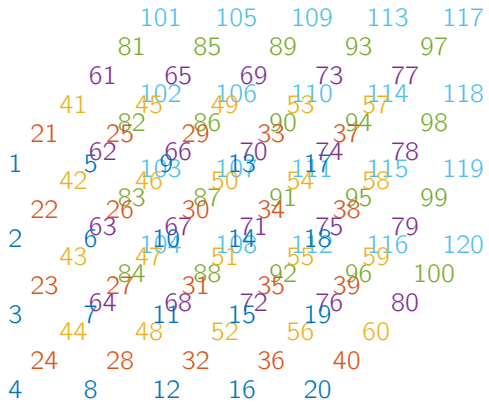
$$\boldsymbol{X}_{(k)} \in \mathbb{C}^{n_k \times n_1 n_2 \cdots n_{k-1} n_{k+1} \cdots n_d}.$$

Example 7.2.1. Consider a tensor $\mathcal{X} = \mathbb{R}^{n_1 \times n_2 \times n_3}$ with $n_1 = 4, n_2 = 5$ and $n_3 = 6$. Observe that the structure of this tensor can be visualized as shown in Figure 7.1. In this example we will use a color coding for values in different frontal slices, as indicated in Figure 7.1b.

Define $\boldsymbol{n} = (n_1,\ n_2,\ n_3)^{\mathsf{T}}$.

Let the tensor $\mathcal{X}$ be given by $\mathcal{X} = \texttt{reshape}[1 : n_1 n_2 n_3,\ \boldsymbol{n}]$ or

$$\mathcal{X} =$$



Then the first, second and third unfolding $\boldsymbol{X}_{(1)} \in \mathbb{R}^{n_1 \times n_2 n_3}$, $\boldsymbol{X}_{(2)} \in \mathbb{R}^{n_2 \times n_1 n_3}$ and $\boldsymbol{X}_{(3)} \in \mathbb{R}^{n_3 \times n_1 n_2}$ of tensor $\mathcal{X}$ are given by:

$$\boldsymbol{X}_{(1)} = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 & 65 & 69 & 73 & 77 & 81 & 85 & 89 & 93 & 97 & 101 & 105 & 109 & 113 & 117 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 & 66 & 70 & 74 & 78 & 82 & 86 & 90 & 94 & 98 & 102 & 106 & 110 & 114 & 118 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 & 67 & 71 & 75 & 79 & 83 & 87 & 91 & 95 & 99 & 103 & 107 & 111 & 115 & 119 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 & 68 & 72 & 76 & 80 & 84 & 88 & 92 & 96 & 100 & 104 & 108 & 112 & 116 & 120 \end{bmatrix},$$

$$\boldsymbol{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 21 & 22 & 23 & 24 & 41 & 42 & 43 & 44 & 61 & 62 & 63 & 64 & 81 & 82 & 83 & 84 & 101 & 102 & 103 & 104 \\ 5 & 6 & 7 & 8 & 25 & 26 & 27 & 28 & 45 & 46 & 47 & 48 & 65 & 66 & 67 & 68 & 85 & 86 & 87 & 88 & 105 & 106 & 107 & 108 \\ 9 & 10 & 11 & 12 & 29 & 30 & 31 & 32 & 49 & 50 & 51 & 52 & 69 & 70 & 71 & 72 & 89 & 90 & 91 & 92 & 109 & 110 & 111 & 112 \\ 13 & 14 & 15 & 16 & 33 & 34 & 35 & 36 & 53 & 54 & 55 & 56 & 73 & 74 & 75 & 76 & 93 & 94 & 95 & 96 & 113 & 114 & 115 & 116 \\ 17 & 18 & 19 & 20 & 37 & 38 & 39 & 40 & 57 & 58 & 59 & 60 & 77 & 78 & 79 & 80 & 97 & 98 & 99 & 100 & 117 & 118 & 119 & 120 \end{bmatrix},$$

$$\boldsymbol{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\ 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\ 81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 & 91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100 \\ 101 & 102 & 103 & 104 & 105 & 106 & 107 & 108 & 109 & 110 & 111 & 112 & 113 & 114 & 115 & 116 & 117 & 118 & 119 & 120 \end{bmatrix}.$$

7.2.2 Tensor multiplication by matrices

Tensors can be *multiplied by matrices*, which leads to the so called k-mode products. Consider again the tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$. This tensor can be multiplied in mode-k by a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n_k}$. This tensor times matrix product is denoted by

$$\mathcal{Y} = \mathcal{X} \times_k \boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2 \times \cdots n_{k-1} \times m \times n_{k+1} \times \cdots \times n_d}$$

and yields again a tensor. Observe that the number of unknowns in the k-th direction may change, depending on the dimensions of matrix $\boldsymbol{A}$. Elementwise this operation is defined by

$$y_{i_1, i_2, \cdots, i_{k-1}, j, i_{k+1}, \cdots, i_d} = \sum_{i_k=1}^{n_k} x_{i_1, i_2, \ldots, i_d} \overline{a_{j i_k}}, \tag{7.4}$$

or, using the k-th unfolding of tensor $\mathcal{X}$ the k-th unfolding of this product is given by

$$\boldsymbol{Y}_{(k)} = \overline{\boldsymbol{A}} \boldsymbol{M}_{(k)}. \tag{7.5}$$

To finish this short introduction about tensor multiplication by matrices, we recall some useful properties for the tensor times matrix product, originally already listed for real-valued matrices:

Proposition 1 ([49, Proposition 3.4])**.** *Let tensor $\mathcal{G} \in \mathbb{C}^{r_1 \times r_2 \times \ldots r_d}$.*

1. *Given matrices $\boldsymbol{A} \in \mathbb{C}^{N \times r_n}$ and $\boldsymbol{B} \in \mathbb{C}^{M \times r_m}$, then*

$$\mathcal{G} \times_n \boldsymbol{A} \times_m \boldsymbol{B} = (\mathcal{G} \times_n \boldsymbol{A}) \times_m \boldsymbol{B} = (\mathcal{G} \times_m \boldsymbol{B}) \times_n \boldsymbol{A} \quad (n \neq m). \tag{7.6}$$

2. *Given matrices $\boldsymbol{A} \in \mathbb{C}^{N \times r_k}$ and $\boldsymbol{B} \in \mathbb{C}^{K \times N}$ then*

$$\mathcal{G} \times_k \boldsymbol{A} \times_k \boldsymbol{B} = \mathcal{G} \times_k (\boldsymbol{B}\boldsymbol{A}). \tag{7.7}$$

3. *If $\boldsymbol{A} \in \mathbb{C}^{K \times r_k}$ is unitary, i.e. $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{I}$, then*

$$\mathcal{M} = \mathcal{G} \times_k \boldsymbol{A} \Rightarrow \mathcal{G} = \mathcal{M} \times_k \boldsymbol{A}^{\mathsf{H}}. \tag{7.8}$$

7.3 Tensor rank and tensor decompositions

Before we start with some preliminaries about different tensor decompositions, it is important to mention that these tensor decompositions have properties that can be seen as high-dimensional extensions of the singular value decomposition (SVD) for matrices.

For a given complex-valued matrix $\boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2}$ the SVD of matrix $\boldsymbol{A}$ is given by $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$, where $\boldsymbol{U} \in \mathbb{C}^{n_1 \times r}$ and $\boldsymbol{V} \in \mathbb{C}^{n_2 \times r}$ have orthonormal columns and $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ is a diagonal matrix. Further $r \leq \min(n_1, n_2)$ is called the rank of the matrix.

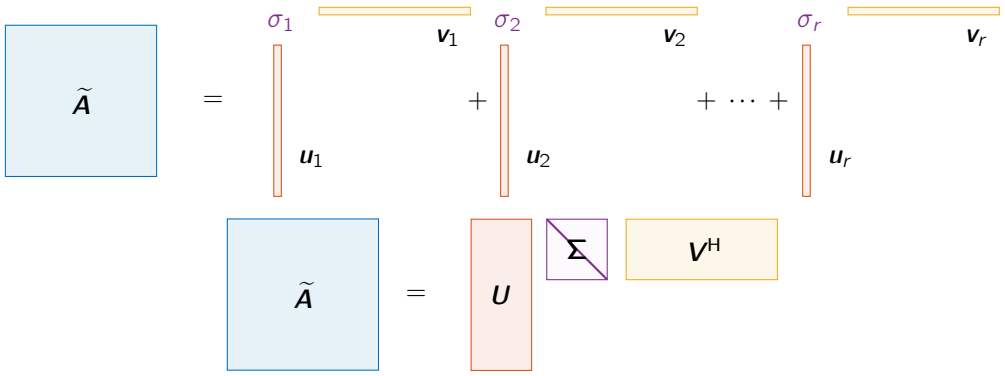


Figure 7.4: Visualization of the singular value decomposition of a low-rank matrix $\widetilde{\boldsymbol{A}}$.

It is known that the SVD of $\boldsymbol{A}$ satisfies the following two different properties simultaneously:

1. The SVD is a rank-r decomposition. Thus, the best low-rank approximation $\widetilde{\boldsymbol{A}}$ to $\boldsymbol{A}$ with rank $r \leq \min(n_1, n_2)$ is given by

$$\boldsymbol{A} \approx \widetilde{\boldsymbol{A}} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{H}} = \sum_{i=1}^{r} \sigma_i (\boldsymbol{u}_i \circ \boldsymbol{v}_i), \tag{7.9}$$

   where $\circ$ denotes the outer product of two vectors and $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the i-th columns of respectively matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. The diagonal matrix $\boldsymbol{\Sigma}$ has entries σ_i on its diagonal.

   This property of constructing a rank-r decomposition can be maintained for tensor decompositions and this will lead to the *Canonical Polyadic (CP) decomposition*, as presented in Section 7.3.1.

2. The SVD constructs orthonormal mode matrices. Thus, the factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ satisfy the following identities

$$\begin{aligned} \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} &= \boldsymbol{I}, \\ \boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} &= \boldsymbol{I}. \end{aligned} \tag{7.10}$$

   This property can be maintained for high-dimensional data and lead to the *Tucker tensor decomposition*, as presented in Section 7.3.2.

The singular value decomposition of matrix $\boldsymbol{A}$ computes the eigenvalue decomposition of the matrices $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}$ that arise in the normal equation for a linear system with $\boldsymbol{A}$. Observe that in two dimensions both conditions are satisfied with the matrix SVD. But, for high-dimensional tensors choosing one of these two properties will lead to a different tensor decomposition and only one property can be satisfied.

In practice, the Canonical Polyadic decomposition is often applied to tensors for data interpretation. Although this is an interesting field and with a range of applications in this thesis we will mainly focus on the Tucker tensor decomposition. The Tucker tensor decomposition is often used for data compression, due to the orthogonality property of the factor matrices. This compression is also exactly our application if we derive equations to approximate the low-rank solutions to (partial) differential equations.

7.3.1 Canonical Polyadic decomposition

The Canonical Polyadic (CP) decomposition preserves property (7.9) and factorizes a d-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ as the sum of rank-one tensors, as defined in (7.3). Thus with the CP decomposition one wants to represent a tensor $\mathcal{X}$ with a rank-r tensor $\widetilde{\mathcal{X}}$:

$$\mathcal{X} \approx \widetilde{\mathcal{X}} = \sum_{i=1}^{r} \sigma_i \left(\boldsymbol{u}_i^{(1)} \circ \boldsymbol{u}_i^{(2)} \circ \cdots \circ \boldsymbol{u}_i^{(d)} \right), \tag{7.11}$$

where $\circ$ is the outer (or Kronecker) product for vectors, rank r is a positive integer, σ_i is a weight and vector $\boldsymbol{u}_i^{(j)} \in \mathbb{C}^{n_j}$ is normalized for $i = 1, 2, \ldots, r$ and $j = 1, 2, \ldots, d$. The vectors $\boldsymbol{u}_i^{(j)}$ can be seen as columns of matrix $\boldsymbol{U}^{(j)} = \left[\boldsymbol{u}_1^{(j)}, \boldsymbol{u}_2^{(j)}, \cdots, \boldsymbol{u}_r^{(j)} \right]$ with $j = 1, 2, \ldots, d$.
The matrices $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(d)}$ are called *factor matrices*.

Elementwise the Canonical Polyadic decomposition is given by

$$x_{i_1, i_2, \ldots, i_d} \approx \widetilde{x}_{i_1, i_2, \ldots, i_d} = \sum_{i=1}^{r} \sigma_i u_{i i_1}^{(1)} u_{i i_2}^{(2)} \cdots u_{i i_d}^{(d)}, \tag{7.12}$$

where $i_k = 1, 2, \ldots, n_k$ for $k = 1, 2, \ldots, d$. A visualization of the CP decomposition for a three-dimensional tensor is shown in Figure 7.5.

Using the factorization of a tensor $\widetilde{\mathcal{X}}$ as given in (7.11), the matrix unfoldings of $\widetilde{\mathcal{X}}$ can be given in terms of Khatri-Rao products[2]:

$$\widetilde{\boldsymbol{X}}_{(k)} = \overline{\boldsymbol{U}^{(k)}} \boldsymbol{D} \left(\bigodot_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}^{(l)} \right)^{\mathsf{H}} \tag{7.13}$$

where $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{\sigma})$ is a diagonal matrix with the values σ_i on its diagonal.

For example, if $\mathcal{X}$ is a real-valued three-dimensional tensor the unfoldings are given by

$$\begin{aligned}
\widetilde{\boldsymbol{X}}_{(1)} &= \boldsymbol{U}^{(1)} \boldsymbol{D} \left(\boldsymbol{U}^{(3)} \odot \boldsymbol{U}^{(2)} \right)^{\mathsf{T}}, \\
\widetilde{\boldsymbol{X}}_{(2)} &= \boldsymbol{U}^{(2)} \boldsymbol{D} \left(\boldsymbol{U}^{(3)} \odot \boldsymbol{U}^{(1)} \right)^{\mathsf{T}}, \\
\widetilde{\boldsymbol{X}}_{(3)} &= \boldsymbol{U}^{(3)} \boldsymbol{D} \left(\boldsymbol{U}^{(2)} \odot \boldsymbol{U}^{(1)} \right)^{\mathsf{T}}.
\end{aligned}$$

7.3.1.1 The CP-rank of a tensor

The *Canonical Polyadic-rank*, *CP-rank* or *rank* of tensor $\mathcal{X}$, denoted by $r = \operatorname{rank}(\mathcal{X})$, is defined as the smallest number of rank-one tensors that is needed to obtain equality in the approximation (7.11). In contrast to the matrix rank, for the CP-rank there is no algorithm to determine the CP-rank of a given tensor. Moreover, this problem is NP-hard [48].

[2]Here, the notation $\bigodot_{l=d}^{1} \boldsymbol{U}^{(l)}$ should be interpreted as $\bigodot_{l=d}^{1} \boldsymbol{U}^{(l)} = \boldsymbol{U}^{(d)} \odot \boldsymbol{U}^{(d-1)} \odot \cdots \odot \boldsymbol{U}^{(1)}$.

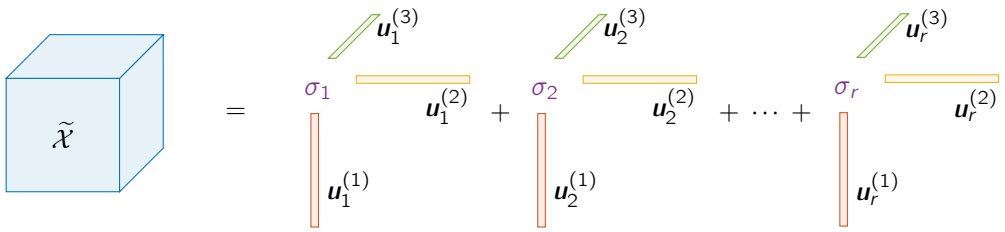


Figure 7.5: A visualization of the Canonical Polyadic decomposition of a three-dimensional tensor $\widetilde{\mathcal{X}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ with rank r.

7.3.1.2 Storage costs for CP-decomposition

One of the advantages of the Canonical Polyadic decomposition is that it can describe high-dimensional, but low rank, tensors, with $n^d := n_1 n_2 \cdots n_d$ elements using only a small number of parameters. Indeed, the number of parameters to store the CP decomposition is only $\mathcal{O}(dnr)$. Here we use the convention $n := \sqrt[d]{n^d}$ and n^d as defined before. The *linear dependence* of the number of parameters on the dimension d of the representation of the tensor clearly breaks the curse of dimensionality.

7.3.1.3 Computing the CP-decomposition

Given a tensor $\mathcal{X}$, it is possible to compute an rank-r CP-decomposition $\widetilde{\mathcal{X}}$ of $\mathcal{X}$. To obtain low-rank matrix approximations, one can compute an SVD and select the first r singular values and vectors to obtain the best rank-r approximation of a matrix. Such a relation does not exist for the CP-decomposition of tensors. To obtain a rank-r CP-decomposition the alternating least squares method (ALS) is used [8, 30, 48].

Consider the tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ and compute the rank-r CP-decomposition $\widetilde{\mathcal{X}}$ that minimizes

$$\min_{\widetilde{\mathcal{X}}} \left\| \mathcal{X} - \widetilde{\mathcal{X}} \right\|, \tag{7.14}$$

where

$$\widetilde{\mathcal{X}} = \sum_{i=1}^{r} \sigma_i \left(\boldsymbol{u}_i^{(1)} \circ \boldsymbol{u}_i^{(2)} \circ \cdots \circ \boldsymbol{u}_i^{(d)} \right).$$

Here, σ_i for $i = 1, 2, \ldots, r$ and $\boldsymbol{U}^{(l)}$ with $l = 1, 2 \ldots, d$ are the unknowns.

The alternating least squares method iterates over $k = 1, 2, \ldots, d$ to solve for factor matrix $\boldsymbol{U}^{(k)}$, where matrices $\boldsymbol{U}^{(l)}$ with $l \neq k$ are fixed. Thus, the minimization problem (7.14) reduces to a linear least squares problem. When one solves for $\boldsymbol{U}^{(k)}$, with $k = 1, 2, \ldots, d$ the minimization problem written in the k-th unfolding reduces to

$$\min_{\widehat{\boldsymbol{U}}^{(k)}} \left\| \boldsymbol{X}_{(k)} - \widehat{\boldsymbol{U}}^{(k)} \left(\bigodot_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}^{(l)} \right)^{\mathsf{T}} \right\|_F$$

Algorithm 2: Alternating least squares algorithm to compute the rank-r CP decomposition $\widetilde{\mathcal{X}}$ of tensor $\mathcal{X}$.

Given: general tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ and rank $r \in \mathbb{N}$;
Construct initial guess for $\boldsymbol{U}^{(k)}$ with $k = 1, 2, \ldots, d$;
while *not converged* **do**
    for $k = 1, 2, \ldots d$ **do**
        Compute $\widehat{\boldsymbol{U}}^{(k)}$ using (7.15);
        Normalize columns of $\widehat{\boldsymbol{U}}^{(k)}$ s.t. $\widehat{\boldsymbol{U}}^{(k)} = \boldsymbol{U}^{(k)} \mathrm{diag}(\boldsymbol{\sigma})$;
    end
end
$\mathcal{X} \approx \widetilde{\mathcal{X}} = \sum_{i=1}^{r} \sigma_i \left(\boldsymbol{u}_i^{(1)} \circ \boldsymbol{u}_i^{(2)} \circ \cdots \circ \boldsymbol{u}_i^{(d)} \right)$;

where $\widehat{\boldsymbol{U}}^{(k)} := \boldsymbol{U}^{(k)} \mathrm{diag}(\boldsymbol{\sigma})$. Hence, the least squares solution to this problem is given by

$$\begin{aligned} \widehat{\boldsymbol{U}}^{(k)} &= \boldsymbol{X}_{(k)} \left[\left(\bigodot_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}^{(l)} \right)^{\mathsf{T}} \right]^{\dagger}, \\ &= \boldsymbol{X}_{(k)} \left(\bigodot_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}^{(l)} \right) \left(\bigodot_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}^{(l)\mathsf{T}} \boldsymbol{U}^{(l)} \right)^{\dagger}. \end{aligned} \tag{7.15}$$

This algorithm to compute the rank-r CP decomposition is summarized in Algorithm 2. A practical and efficient implementation of this algorithm is given in e.g. [79].

7.3.2 Tucker tensor decomposition

An alternative class of tensor decompositions can be obtained by preserving property (7.10). The Tucker tensor decomposition writes a low-rank tensor $\widetilde{\mathcal{X}} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ into a small core tensor $\mathcal{G} \in \mathbb{C}^{r_1 \times r_2 \times \cdots \times r_d}$ multiplied by orthonormal factor matrices $\boldsymbol{U}_i \in \mathbb{C}^{n_i \times r_i}$ (with $i = 1, 2, \ldots, d$) along each mode:

$$\begin{aligned} \mathcal{X} \approx \widetilde{\mathcal{X}} &= \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_d \boldsymbol{U}_d, \\ &= \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_d=1}^{r_d} g_{i_1, i_2, \ldots, i_d} \left(\boldsymbol{u}_{i_1}^{(1)} \circ \boldsymbol{u}_{i_2}^{(2)} \circ \cdots \circ \boldsymbol{u}_{i_d}^{(d)} \right). \end{aligned} \tag{7.16}$$

where $\boldsymbol{u}_{i_k}^{(k)}$ represents the i_k-th column of factor matrix $\boldsymbol{U}_k$.

With this Tucker tensor decomposition, the low-rank tensor $\widetilde{\mathcal{X}}$ can be written as a linear combination of (at most) $r^d := r_1 r_2 \cdots r_d$ rank-one tensors as outer products of the different columns of the factor matrices. Elementwise the Tucker tensor decomposition is given by

$$x_{i_1, i_2, \ldots, i_d} \approx \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \cdots \sum_{j_d=1}^{r_d} g_{j_1 j_2 \cdots j_d} \left(u_{1 i_1 j_1} u_{2 i_2 j_2} \cdots u_{d i_d j_d} \right), \tag{7.17}$$

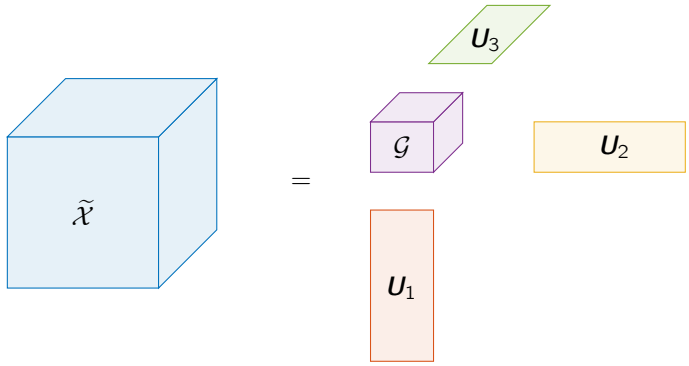


Figure 7.6: A visualization of the Tucker tensor decomposition of a three-dimensional tensor $\widetilde{\mathcal{X}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ with multi-linear rank $\boldsymbol{r} \in \mathbb{N}^d$.

where $i_k = 1, 2, \ldots, n_k$ for $k = 1, 2, \ldots, d$.

Note that a tensor $\mathcal{M}$ given in a Tucker tensor decomposition is not unique. It is possible to choose arbitrary unitary matrices $\boldsymbol{Q}_i \in \mathbb{C}^{r_i \times r_i}$ and represent $\mathcal{M}$ as:

$$\begin{aligned} \mathcal{M} &= \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_d \boldsymbol{U}_d \\ &= \mathcal{G} \times_1 \boldsymbol{U}_1 \boldsymbol{Q}_1^{\mathsf{H}} \boldsymbol{Q}_1 \times_2 \boldsymbol{U}_2 \boldsymbol{Q}_2^{\mathsf{H}} \boldsymbol{Q}_2 \times \cdots \times_d \boldsymbol{U}_d \boldsymbol{Q}_d^{\mathsf{H}} \boldsymbol{Q}_d \\ &= (\mathcal{G} \times_1 \boldsymbol{Q}_1 \times_2 \boldsymbol{Q}_2 \times \cdots \times_d \boldsymbol{Q}_d) \times_1 \boldsymbol{U}_1 \boldsymbol{Q}_1^{\mathsf{H}} \times_2 \boldsymbol{U}_2 \boldsymbol{Q}_2^{\mathsf{H}} \times \cdots \times_d \boldsymbol{U}_d \boldsymbol{Q}_d^{\mathsf{H}} \end{aligned} \tag{7.18}$$

which is also a Tucker tensor decomposition. So, without changing the tensor $\mathcal{M}$ it is always possible to use a representation with unitary factor matrices $\boldsymbol{U}_i$. The Tucker tensor decomposition for a three-dimensional tensor is visualized in Figure 7.6.

7.3.2.1 The k-rank and multilinear of a tensor

Consider a d-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$. The k-rank (with $1 \leq k \leq d$) of this tensor, denoted by $r_k = \operatorname{rank}_k(\mathcal{X})$, is defined as the column rank of the matrix $\boldsymbol{X}_{(k)}$, i.e.

$$r_k = \operatorname{rank}_k(\mathcal{X}) = \operatorname{rank}\left(\boldsymbol{X}_{(k)}\right), \tag{7.19}$$

where the vector $\boldsymbol{r} = (r_1, r_2, \ldots, r_d)^{\mathsf{T}} \in \mathbb{N}^d$ is called the *multilinear rank* of the tensor. Note that the different k-ranks of a tensor are not necessary equal for all $k = 1, 2, \ldots, d$.

7.3.2.2 Storage costs for Tucker tensor decomposition

For tensors that can be described by a low multilinear rank $\boldsymbol{r}$, also the Tucker tensor decomposition clearly reduced the total number of parameters. But, due to the existence of the core tensor, with $r^d := r_1 r_2 \cdots r_d$ unknowns, the Tucker tensor format has still a number of parameters that depends exponential on the dimension d. Indeed the number of parameters to store a tensor in the Tucker tensor decomposition is $\mathcal{O}\left(r^d + dnr\right)$ where $n := \sqrt[d]{n_1 n_2 \cdots n_d}$.

Thus the number of parameters is still exponential in the dimension d. Indeed, n^d unknowns in the full rank tensor are exchanged for r^d unknowns in the core tensor. The only advantage of this construction is that possibly $r^d \ll n^d$. Thus the Tucker tensor decomposition can be beneficial for three-dimensional problems, but when the dimension d further increases another tensor decomposition has to be considered.

7.3.2.3 Unfolding a Tucker tensors

Assume that the Tucker tensor decomposition of a d-dimensional tensor $\mathcal{X}$ is known and given by $\mathcal{X} \approx \mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_d \boldsymbol{U}_d$. Then, a useful explicit expression for the tensor unfolding in terms of these matrices is known; the k-th unfolding of a tensor $\mathcal{M}$ represented in Tucker tensor format is given by:

$$\boldsymbol{M}_{(k)} = \overline{\boldsymbol{U}_k} \boldsymbol{G}_{(k)} \left(\bigotimes_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}_l \right)^{\mathsf{H}} . \tag{7.20}$$

As a special case, we remark that also vectorization can be seen as a certain unfolding and is written as

$$\mathtt{vec}\,[\mathcal{M}] = (\boldsymbol{U}_d \otimes \boldsymbol{U}_{d-1} \otimes \cdots \boldsymbol{U}_1)\,\mathtt{vec}\,[\mathcal{G}] = \left(\bigotimes_{l=d}^{1} \boldsymbol{U}_l \right) \mathtt{vec}\,[\mathcal{G}]\,. \tag{7.21}$$

7.3.2.4 Computing a Tucker tensor decomposition

A Tucker tensor decomposition of an arbitrary tensor $\mathcal{X}$ can be constructed using a sequence of singular value decompositions of the unfolded matrices, also known as Higher-order SVD [14, 86, 87] or HOSVD. Recall that for a Tucker tensor decomposition one computes the core tensor $\mathcal{G}$ and the factor matrices $\boldsymbol{U}_k$ with $k = 1, 2, \ldots, d$:

$$\mathcal{X} \approx \mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_d \boldsymbol{U}_d. \tag{7.22}$$

Further, the k-th unfolding of tensor $\mathcal{X}$ is given by

$$\boldsymbol{X}_{(k)} = \overline{\boldsymbol{U}_k} \boldsymbol{G}_{(k)} \left(\bigotimes_{\substack{l=d \\ l \neq k}}^{1} \boldsymbol{U}_l \right)^{\mathsf{H}} . \tag{7.23}$$

Thus, an SVD of $\boldsymbol{X}_{(k)}$ yields exactly the factor matrix $\overline{\boldsymbol{U}_k}$ with the orthonormality property $\boldsymbol{U}_k^{\mathsf{H}} \boldsymbol{U}_k = \boldsymbol{I}$. This can be repeated for all $k = 1, 2, \ldots, d$.

Finally, using (7.8) the core tensor $\mathcal{G}$ can be computed. The HOSVD algorithm is summarized by Algorithm 3.

As mentioned before, a Tucker tensor decomposition is not unique. This gives in principle possibilities choose some transformations to simplify the structure of the core tensor $\mathcal{G}$ in some sense. This was already observed by Tucker [86] and many others [48].

Algorithm 3: The Higher-order SVD algorithm to compute the Tucker tensor decomposition of tensor $\mathcal{X}$.

Given: general tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ and multilinear rank $\boldsymbol{r} \in \mathbb{N}^d$;
for $k = 1, 2, \ldots d$ **do**
  $\left[\boldsymbol{U}_k, \boldsymbol{\Sigma}_k, \boldsymbol{V}_k\right] = \mathrm{svd}\left[\boldsymbol{X}_{(k)}\right]$;
end
$\mathcal{G} = \mathcal{X} \times_1 \boldsymbol{U}_1^{\mathsf{H}} \times_2 \boldsymbol{U}_2^{\mathsf{H}} \times \cdots \times_d \boldsymbol{U}_d^{\mathsf{H}}$;

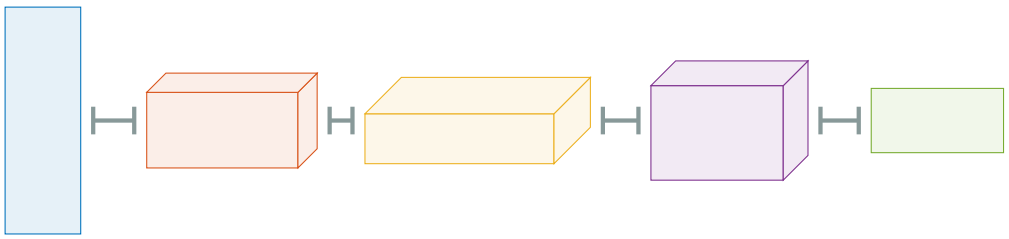


Figure 7.7: A visualization of the Tensor Train decomposition of a five-dimensional tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_5}$. All tensors have a dimension of at most three, where the size of tensor $\mathcal{G}_k$ equals $r_{k-1} \times n_k \times r_k$ for $k = 1, 2, \ldots, d$.

7.3.2.5 High-dimensional Tucker extension: Tensor Train decomposition

Because the Tucker tensor decomposition is less efficient to compress high-dimensional data, due to the exponentially dependence of the number of parameters on the dimension d, other tensor decompositions are developed and studied. For example the Tensor Train decomposition by Oseledets [64] is a possibility to maintain a certain kind of 'orthogonal factors' combined with a number of parameters that do not scale exponentially in the dimension. With the Tensor Train format the original tensor $\mathcal{X}$ is represented by a network of low-dimensional and small tensors, in this case a linear tensor network, as visualized in Figure 7.7.

Elementwise the Tensor Train decomposition is given by

$$x_{i_1, i_2, \ldots, i_d} \approx \sum_{\alpha_1=1}^{r_1} \sum_{\alpha_2=1}^{r_2} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} G_1(\alpha_0, i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \cdots G_d(\alpha_{d-1}, i_d, \alpha_d) \tag{7.24}$$

where r_k, with $k = 1, 2, \ldots, d$, are the ranks of certain auxiliary matrices, often called *compression ranks* or *TT-ranks*. Further $\alpha_0 = \alpha_d = 1$ and $i_k = 1, 2, \ldots, n_k$ for $k = 1, 2, \ldots, d$. Observe that $\boldsymbol{G_1}$ and $\boldsymbol{G_d}$ are matrices with size $n_1 \times r_1$ and $r_{d-1} \times n_d$ respectively. All other cores $\mathcal{G}_k$ with $k = 2, 3, \ldots, d-1$ are three-dimensional tensors with dimensions $r_{k-1} \times n_k \times r_k$.

Also in a Tensor Train context the tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ can be reshaped to a matrix $\boldsymbol{X}_k \in \mathbb{C}^{\prod_{i=1}^{k} n_i \times \prod_{i=k+1}^{d} n_i}$ where the elements of tensor $\mathcal{X}$ in certain dimensions are stacked in columns of matrix $\boldsymbol{A}_k$. Actually this unfolding is just a reshape of a tensor to a matrix:

$$\boldsymbol{X}_k = \mathtt{reshape}\left[\mathcal{X}, \prod_{i=1}^{k} n_i, \prod_{i=k+1}^{d} n_i\right]. \tag{7.25}$$

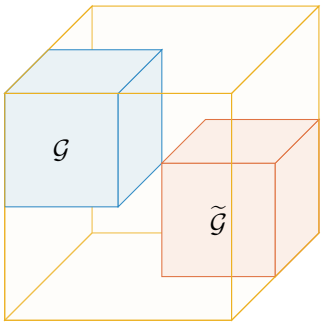


Figure 7.8: Visualization of a block super-diagonal core tensor $\mathcal{H}$.

The existence of a Tensor Train decomposition of a certain TT-rank is given by [64] or formulated as

Theorem 6 ([64, Theorem 2.1])**.** *If the rank of the unfolding matrix $\boldsymbol{X}_k$ of a d-dimensional tensor $\mathcal{X}$ is given by*

$$rank(\boldsymbol{X}_k) = r_k \tag{7.26}$$

then there exists a Tensor Train decomposition (7.24) *with TT-ranks of at most r_k for $k = 1, 2, \ldots, d$.*

The number of parameters for the Tensor Train format are $\mathcal{O}\left(dnr^2\right)$ where $n := \sqrt[d]{n_1 n_2 \cdots n_d}$ and $r := \sqrt[d]{r_1 r_2 \cdots r_d}$.

7.3.3 Linear operators applied on Tucker tensors

Consider two tensors $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ given in Tucker tensor format with multi-linear ranks $\boldsymbol{r}$ and $\widetilde{\boldsymbol{r}}$. Let these tensors be given by

$$\mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_d \boldsymbol{U}_d \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d},$$
$$\widetilde{\mathcal{M}} = \widetilde{\mathcal{G}} \times_1 \widetilde{\boldsymbol{U}_1} \times_2 \widetilde{\boldsymbol{U}_2} \times \cdots \times_d \widetilde{\boldsymbol{U}_d} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}.$$

The *addition of tensors* $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ is trivially defined as the elementwise addition. But when $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ are given in Tucker tensor format then the Tucker tensor representation of the sum has to be constructed explicitly, as given by

$$\mathcal{M} + \widetilde{\mathcal{M}} = \mathcal{H} \times_1 \boldsymbol{V}_1 \times_2 \boldsymbol{V}_2 \times \cdots \times_d \boldsymbol{V}_d,$$

with

$$\boldsymbol{V}_i = \begin{bmatrix} \boldsymbol{U}_i & \widetilde{\boldsymbol{U}_i} \end{bmatrix} \in \mathbb{C}^{n_i \times (r_i + \widetilde{r_i})},$$

where $i = 1, 2, \ldots, d$ and a block super-diagonal core $\mathcal{H} \in \mathbb{C}^{(r_1+\widetilde{r_1}) \times (r_2+\widetilde{r_2}) \times \cdots \times (r_d+\widetilde{r_d})}$ with $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ as block-tensors on the main diagonal of the core tensor $\mathcal{H}$ [81]. A visualization of this block super-diagonal core tensor is given in Figure 7.8.

In general, with this construction the orthogonality of the factor matrices is lost. Further, the multi-linear rank of the core tensor increases with each tensor addition. Orthogonality can be restored by re-orthogonalization of the factor matrices using an QR-decomposition and multiplying the resulting matrix $\boldsymbol{R}$ into the core tensor.

The inner product of two tensors in Tucker format can be computed in terms of smaller matrix products:

$$\begin{aligned}\left\langle \mathcal{M}, \widetilde{\mathcal{M}} \right\rangle &= \mathrm{vec}\left[\mathcal{M}\right]^{\mathsf{H}} \mathrm{vec}\left[\widetilde{\mathcal{M}}\right] \\ &= \mathrm{vec}\left[\mathcal{G}\right]^{\mathsf{H}} \left(\bigotimes_{l=d}^{1} \boldsymbol{U}_l^{\mathsf{H}}\right) \left(\bigotimes_{l=d}^{1} \widetilde{\boldsymbol{U}}_l\right) \mathrm{vec}\left[\widetilde{\mathcal{G}}\right] \\ &= \mathrm{vec}\left[\mathcal{G}\right]^{\mathsf{H}} \left(\bigotimes_{l=d}^{1} \boldsymbol{U}_l^{\mathsf{H}} \widetilde{\boldsymbol{U}}_l\right) \mathrm{vec}\left[\widetilde{\mathcal{G}}\right] \\ &= \mathrm{vec}\left[\mathcal{G} \bigtimes_{l=1}^{d} \widetilde{\boldsymbol{U}}_l^{\mathsf{H}} \boldsymbol{U}_l\right]^{\mathsf{H}} \mathrm{vec}\left[\widetilde{\mathcal{G}}\right] \\ &= \left\langle \mathcal{G} \bigtimes_{l=1}^{d} \widetilde{\boldsymbol{U}}_l^{\mathsf{H}} \boldsymbol{U}_l, \widetilde{\mathcal{G}} \right\rangle.\end{aligned} \tag{7.27}$$

Hence, computation of the norm of a tensor in Tucker format with unitary factor matrices reduces to the norm of the core tensor:

$$\|\mathcal{M}\| = \langle \mathcal{M}, \mathcal{M} \rangle = \mathrm{vec}\left[\mathcal{G} \bigtimes_{l=1}^{d} \boldsymbol{U}_l^{\mathsf{H}} \boldsymbol{U}_l\right]^{\mathsf{H}} \mathrm{vec}\left[\mathcal{G}\right] = \mathrm{vec}\left[\mathcal{G}\right]^{\mathsf{H}} \mathrm{vec}\left[\mathcal{G}\right] = \|\mathcal{G}\|. \tag{7.28}$$

Consider a linear operator[3] $\mathcal{L}$ on a Tucker tensor $\mathcal{M}$ where the linear operator has a Kronecker structured matrix representation. Thus, the application of the linear operator $\mathcal{L} : \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_d} \to \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_d}$ on a tensor can be represented by a matrix-vector product with $\boldsymbol{L} \in \mathbb{C}^{n^d \times n^d}$ where $n^d := \prod_{i=1}^{d} n_i$ and a vectorized tensor. Here

$$\boldsymbol{L} = \sum_{i=1}^{R} \boldsymbol{L}_{d,i} \otimes \boldsymbol{L}_{d-1,i} \otimes \cdots \otimes \boldsymbol{L}_{1,i} \tag{7.29}$$

with $\boldsymbol{L}_{k,i} \in \mathbb{C}^{n_k \times n_k}$ for $i = 1, 2, \ldots, R$ such that

$$\mathcal{F} = \mathcal{L}\mathcal{M} \quad \Leftrightarrow \quad \mathrm{vec}\left[\mathcal{F}\right] = \boldsymbol{L}\mathrm{vec}\left[\mathcal{M}\right]. \tag{7.30}$$

Hence, using (7.21) as vectorization of a tensor the application of a Kronecker structured linear operator $\mathcal{L}$ to a Tucker tensor $\mathcal{M} = \mathcal{G} \bigtimes_{i=1}^{d} \boldsymbol{U}_i$ is given by

$$\mathcal{L}\mathcal{M} = \sum_{i=1}^{R} \mathcal{G} \times_1 \boldsymbol{L}_{1,i}\boldsymbol{U}_1 \times_2 \boldsymbol{L}_{2,i}\boldsymbol{U}_2 \times_3 \ldots \times_d \boldsymbol{L}_{d,i}\boldsymbol{U}_d. \tag{7.31}$$

[3]Note: both (linear) operators and tensors are denoted in a calligraphic font. From the context it will be clear if a (linear) operator is meant or a tensor. (Linear) operators are typically denoted by $\mathcal{A}, \mathcal{B}$ or $\mathcal{L}$ while tensors are typically denoted by $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{F}, \mathcal{G}, \mathcal{H}$ or $\mathcal{M}$.

Chapter 8

Low rank approximation of solutions for linear time-independent PDEs

Chapter summary:

Atomic and molecular breakup reactions, such as multiple-ionization, are described by a driven Schrödinger equation. This equation is equivalent to a high-dimensional Helmholtz equation and it has solutions that are outgoing waves, emerging from the target. We show that these waves can be described by a low-rank approximation. For two-dimensional problems this is a matrix product of two low-rank matrices and for three-dimensional problems it is a low-rank tensor decomposition. We propose an iterative method that solves, in an alternating projection way, for these low-rank components of the scattered wave. We illustrate the method with examples in two and three dimensions.

The content of this chapter is submitted in the paper '*Solving for the low-rank tensor components of a scattering wave function*' by Jacob Snoeijer and Wim Vanroose, [80].

8.1 Introduction

An in-coincidence experiment measures simultaneously the outgoing momenta of multiple products of a microscopic reaction [74]. It is an instrument that can study the correlations in reactions involving multiple particles. In double ionization, for example, a single photon ionizes, simultaneously, two electrons and the outgoing momenta of both particles are captured [1]. The reaction probes the correlation between two electrons in, for example, a chemical bound at the moment of photon impact. The outgoing wave of the two electrons is described by a six-dimensional correlated wave and results in a cross section that depends on four angles, the directions of the first and the second electron.

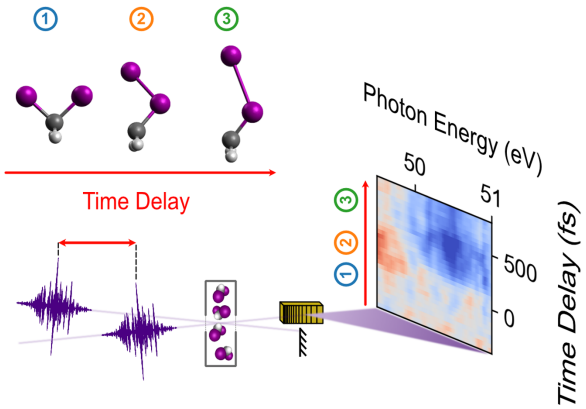


Figure 8.1: Example setup of experiments with free-electron lasers (figure taken from Desy).

Free-electron lasers, and similar experiments around the world, are expected to generate a wealth of scattering data. This will result in high-dimensional forward and inverse wave problems that need to be solved to interpret the data. A sketch of the setup of these kind of experiments is shown in Figure 8.1.

The experimental cross sections are often smooth functions as a function of the angles. Similarly, some parts of the scattering solution, such as single ionization, is localized a limited subspace of the possible full solution domain. The scattering solution can then be described by a low-rank wave function, a product of one-particle bound states with scattering waves in the other coordinates.

This chapter introduces a low-rank representation for the scattering solutions, not only for the single ionization but also for double and triple ionization waves that appear in breakup reactions.

We also propose and analyze an alternating direction algorithm that directly solves for the low-rank components that describe the solution. This reduces a large-scale linear system to smaller, low-dimensional, scattering problems that are solved in a iterative sequence. The proposed method can be generalized to high-dimensional scattering problems where a low-rank tensor decomposition is used to represent the full scattering wave function.

Efficient low-rank tensor representations are used in quantum physics for quite some time already [34, 58]. They are also used in the applied mathematics literature to approximate high-dimensional problems, for a review see [26, 27, 48]. Methods such ALS [32], DMRG [63], and AMEn [19] use in alternating directions, a small linear system to determine the low-rank components of a tensor decomposition. These innovations have not found their application in computational scattering theory.

To calculate cross sections, from first principles, we start from a multi-particle Schrödinger equation. The equation is reformulated into a driven Schrödinger equation with an unknown scattering wave function and a right hand side that describes the excitation, for example, a dipole operator working on the initial state.

Since the asymptotic behaviour of a scattering function for multiple charged particles is in many cases unknown, absorbing boundary conditions [2, 76] are used. Here, an artificial layer is added to the numerical domain that dampens outgoing waves. The outgoing wave boundary conditions are then replaced with homogeneous Dirichlet boundary conditions at the end of the artificial layer. This boundary do not require any knowledge about the asymptotic behaviour, which becomes very complicated for these multiple charged particles.

The resulting equation is discretized on a grid and results in a large, sparse indefinite linear system. It is typically solved by a preconditioned Krylov subspace method [12]. However, the preconditioning techniques for indefinite systems are not as efficient as preconditioners for symmetric and positive definite systems. Solving the resulting equation is still a computationally expensive task, often requiring a distributed calculation on a supercomputer.

To compare the resulting theoretical cross sections with experimental data, a further post-processing step is necessary. The cross section is the farfield map and this is calculated through integrals of the scattering wave function, which is the solution of the linear system, and a Greens function [56].

The main result of the chapter is that we show that scattering waves that describe multiple ionization can be represented by a low-rank tensor. We first show this for a two-dimensional wave and then generalize the results to three-dimensional waves. The methodology can be generalized to higher dimensional waves.

The outline of this chapter is a follows. In Section 8.2 we review the methodology that solves the forward scattering problem. It results in a driven Schrödinger equation with absorbing boundary conditions. From the solution we can extract the cross section using an integral. In Section 8.3 we illustrate, in two dimensions, that the solution can be approximated by a truncated low-rank approximation. We also show that these low-rank components can be calculated directly with an iterative method. In Section 8.4 we show that this methodology generalizes to three- and higher-dimensional problems. A truncated Tucker tensor decomposition is used to determine the low-rank components with a similar iterative method. A discussion of some numerical results and a comparison of the different presented versions of the method is given in Section 8.5. Finally, in Section 8.6, we summarize some conclusions and discuss some possible extensions of the presented method.

8.2 State of the art

This section summarizes the methodology that solves forward break-up problems with charged particles. The methodology is developed in a series of papers [56, 73] and applied to solve the impact-ionization problem [72] and double ionization of molecules [88, 89]. These methods are being extended to treat, for example, water [83].

The helium atom, He, as visualized in Figure 8.2, is the simplest system on which double ionization might occur [4]. It has two electrons with coordinates $\boldsymbol{r}_1 \in \mathbb{R}^3$ and $\boldsymbol{r}_2 \in \mathbb{R}^3$ relative to the nucleus positioned at origin $\mathbf{0}$. The driven Schrödinger equation for $u(\boldsymbol{r}_1, \boldsymbol{r}_2) \in \mathbb{C}^2$

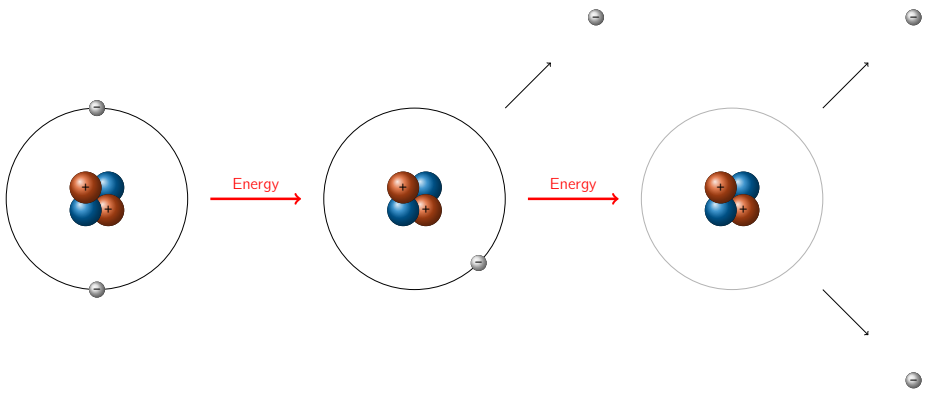


Figure 8.2: Single and double ionization for Helium atom.

then reads

$$\left(-\frac{\hbar^2}{m_e}\frac{1}{2}\Delta_{r_1}-\frac{\hbar^2}{m_e}\frac{1}{2}\Delta_{r_2}-\frac{Ze^2}{4\pi\epsilon_0\|r_1\|}-\frac{Ze^2}{4\pi\epsilon_0\|r_2\|}+\frac{e^2}{4\pi\epsilon_0\|r_1-r_2\|}-E\right)u(r_1,r_2)=\mu\cdot\phi_0(r_1,r_2), \tag{8.1}$$

forall $r_1, r_2 \in \mathbb{R}^3$, where Ze is the nuclear charge of the atom, thus for the helium atom one has $Z=2$. Further, $\hbar=\frac{h}{2\pi}$, where h is the Planck constant [J·s], m_e the mass of a stationary electron [kg], e is the elementary charge [C] and ϵ_0 is the vacuum permittivity constant $\left[\frac{s^2\cdot C^2}{m^3\cdot kg}\right]$.

Here, $u(r_1,r_2)$ is a probability amplitude that yields a probability density for the electron in the far field. The right hand side is the dipole operator μ working on the ground state ϕ_0, the eigenstate with the lowest energy λ_0. The operators $-\frac{1}{2}\Delta_{r_1}$ and $-\frac{1}{2}\Delta_{r_2}$ are the Laplacian operators for the first and second electron and model the kinetic energy. The nuclear attraction scales with $-1/\|r_1\|$ and $-1/\|r_2\|$ and the electron-electron repulsion scales with $1/\|r_1-r_2\|$.

The total energy $E=\hbar\nu+\lambda_0$ is the energy deposited in the system by the photon, $\hbar\nu$, and the energy λ_0 of the ground state. If the $E>0$, both electrons can escape simultaneously from the system. The solution $u(r_1,r_2)$ then represents a six-dimensional wave emerging from the nucleus.

The equation can be interpreted as a Helmholtz equation with a space-dependent wave number, $k^2(r_1,r_2)$,

$$\left(-\Delta_{6D}-k^2(r_1,r_2)\right)u(r_1,r_2)=f(r_1,r_2), \tag{8.2}$$

where $r_1, r_2 \in \mathbb{R}^3$.

In this chapter we prefer to write this Helmholtz equation as

$$\left(-\Delta_{6D}-k_0^2(1+\chi(r_1,r_2))\right)u(r_1,r_2)=f(r_1,r_2), \tag{8.3}$$

where $r_1, r_2 \in \mathbb{R}^3$, k_0^2 is a constant wave number, in this case related to the total energy E, and a space-dependent function $\chi:\mathbb{R}^6\to\mathbb{R}$, that goes to zero if $\|r_1\|\to\infty$ or $\|r_1\|\to\infty$ that represents all the potentials.

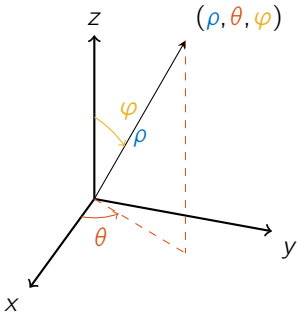


Figure 8.3: A point in 3D space given in spherical coordinates.

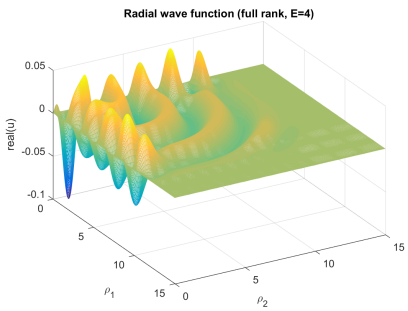


Figure 8.4: Example of a radial wave function $u_{l_1 m_1, l_2 m_2}(\rho_1, \rho_2)$.

8.2.1 Expansion in spherical waves and absorbing boundary conditions

For small atomic and molecular systems, where spherical symmetry is relevant, the system is typically written in spherical coordinates and expanded in spherical harmonics. With $\boldsymbol{r}_1(\rho_1, \theta_1, \varphi_1)$ and $\boldsymbol{r}_2(\rho_2, \theta_2, \varphi_2)$ we can write

$$u(\boldsymbol{r}_1, \boldsymbol{r}_2) = \sum_{l_1=0}^{\infty} \sum_{m_1=-l_1}^{l_1} \sum_{l_2=0}^{\infty} \sum_{m_2=-l_2}^{l_2} u_{l_1 m_1, l_2 m_2}(\rho_1, \rho_2) Y_{l_1 m_1}(\theta_1, \varphi_1) Y_{l_2 m_2}(\theta_2, \varphi_2), \tag{8.4}$$

where $Y_{l_1 m_1}(\theta_1, \varphi_1)$ and $Y_{l_2 m_2}(\theta_2, \varphi_2)$ are spherical harmonics, the eigenfunctions of the angular part of a three-dimensional Laplacian in spherical coordinates. In practice the sum in Equation (8.4) is truncated. The expansion is then a low-rank, truncated, tensor decomposition of a six-dimensional tensor describing the solution.

For each combination of l_1, m_1, l_2 and m_2, the radial function $u_{l_1 m_1, l_2 m_2}(\rho_1, \rho_2)$ describes an outgoing wave that depends on the distances ρ_1 and ρ_2 of the two electrons to the nucleus.

A coupled equation that simultaneously solves for all the $u_{l_1 m_1, l_2 m_2}(\rho_1, \rho_2)$'s is found by inserting the truncated sum in (8.1), multiplying with $Y^*_{l_1 m_1}(\theta_1, \varphi_1)$ and $Y^*_{l_2 m_2}(\theta_2, \varphi_2)$ and integrating over all the angular coordinates,

$$\begin{aligned} \left(-\frac{1}{2}\frac{d^2}{d\rho_1^2} + \frac{l_1(l_1+1)}{2\rho_1^2} - \frac{1}{2}\frac{d^2}{d\rho_2^2} + \frac{l_2(l_2+1)}{2\rho_2^2} + V_{l_1 m_1 l_2 m_2}(\rho_1, \rho_2) - E \right) u_{l_1 m_1, l_2 m_2}(\rho_1, \rho_2) \\ + \sum_{l_1', m_1', l_2', m_2'} V_{l_1 m_1 l_2 m_2, l_1' m_1' l_2' m_2'}(\rho_1, \rho_2) u_{l_1' m_1', l_2' m_2'}(\rho_1, \rho_2) = f_{l_1 m_1, l_2 m_2}(\rho_1, \rho_2). \end{aligned} \tag{8.5}$$

For all l_1, m_1, l_2, m_2. Further, $\rho_1, \rho_2 \in [0, \infty)$ and boundary conditions $u(\rho_1 = 0, \rho_2) = 0$ for all $\rho_2 \geq 0$ and $u(\rho_1, \rho_2 = 0) = 0$ for all $\rho_1 \geq 0$ are applied.

The equation (8.5) is typically discretized on a spectral elements quadrature grid [73].

To reflect the physics, where electrons are emitted from the system, outgoing wave boundary conditions need to be applied at the outer boundaries. There are many ways to implement outgoing wave boundary conditions. Exterior complex scaling (ECS) [76] for example, is

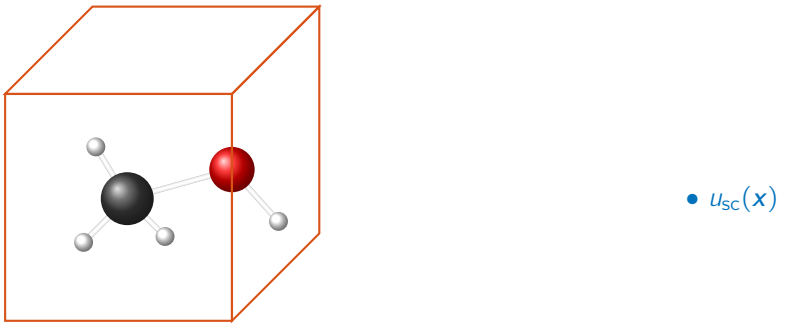


Figure 8.5: The near-field and far-field around a molecule. The detector to measure $u_{sc}(\boldsymbol{x})$ is located in the far-field, on a large distance from the molecule.

frequently used in the computational atomic and molecular physics literature. In the computational electromagnetic scattering a perfectly matched layer (PML) [2] is used, which can also be interpreted as a complex scaled grid [11].

8.2.2 Calculation of the amplitudes

To correctly predict the probabilities of the arriving particles at the detector, we need the amplitudes of the solution far away from the molecule. These are related to the asymptotic amplitudes of the wave functions.

Let us go back to the formulation with the Helmholtz equation, as given in (8.3). Suppose that we have solved the following Helmholtz equation with absorbing boundary conditions, in any representation,

$$\left(-\Delta_{6D} - k_0^2\,(1+\chi(\boldsymbol{x}))\right) u_{sc}(\boldsymbol{x}) = f(\boldsymbol{x}), \tag{8.6}$$

for all $\boldsymbol{x} \in [-b,\ b]^d$ and where f is only non-zero on the real part of the grid $[-b,\ b]^d \subset \mathbb{R}^d$. Similarly, $\chi(\boldsymbol{x})$ is only non-zero on the box $[-b,\ b]^d$.

The calculation of the asymptotic amplitudes requires the solution $u_{sc}(\boldsymbol{x})$ for an $\boldsymbol{x}$ outside of the box $[-b,\ b]^d$. To that end, we reorganize equation (8.6), after we have solved it, as follows

$$\left(-\Delta_{6D} - k_0^2\right) u_{sc} = f + k_0^2\,\chi\,u_{sc}. \tag{8.7}$$

The right hand side of (8.7) is now only non-zero on $[-b,\ b]^d$, since both f and χ are only non-zero there. Furthermore, since we have solved (8.6) we also know u_{sc} on $[-b,\ b]^d$. So the full right hand side of (8.7) is known. The remaining left hand side of (8.7) is now a Helmholtz equation with a constant wave number k_0^2. For this equation the Greens function is known analytically. Thus, for all $\boldsymbol{x} \in \mathbb{R}^d$ we have

$$\begin{aligned} u_{sc}(\boldsymbol{x}) &= \int G(\boldsymbol{x},\boldsymbol{y})\left(f(\boldsymbol{y}) + k_0^2\,\chi(\boldsymbol{y})u_{sc}(\boldsymbol{y})\right)\mathrm{d}\boldsymbol{y} \\ &= \int_{[-b,\ b]^d} G(\boldsymbol{x},\boldsymbol{y})\left(f(\boldsymbol{y}) + k_0^2\,\chi(\boldsymbol{y})u_{sc}(\boldsymbol{y})\right)\mathrm{d}\boldsymbol{y}, \end{aligned} \tag{8.8}$$

where f and χ are limited to $[-b,\ b]^d$ thus we can truncate the integral to the box $[-b,\ b]^d$.

This methodology was successfully applied to calculate challenging break up problems, see for example [72].

8.2.3 Single ionization versus double ionization

Let us discuss the qualitative behaviour of the solution for single and double ionization. To illustrate the behaviour, we truncate the partial wave expansion as given in (8.4) to the first term (i.e. $l_1 = l_2 = 0$). This is known as the s-wave expansion. The six-dimensional wave function is then approximated by

$$u(\boldsymbol{r}_1, \boldsymbol{r}_2) \approx u(\rho_1, \rho_2) Y_{00}(\theta_1, \varphi_1) Y_{00}(\theta_2, \varphi_2). \tag{8.9}$$

The radial wave, $u(\rho_1, \rho_2)$, then fits a two-dimensional Helmholtz equation

$$\left(-\frac{1}{2}\frac{d^2}{d\rho_1^2} - \frac{1}{2}\frac{d^2}{d\rho_2^2} + V_1(\rho_1) + V_2(\rho_2) + V_{12}(\rho_1, \rho_2) - E\right) u(\rho_1, \rho_2) = f(\rho_1, \rho_2), \tag{8.10}$$

for all $\rho_1, \rho_2 \in [0, \infty)$ and where $V_1(\rho_1)$ and $V_2(\rho_2)$ represents the one-particle potentials and $V_{12}(\rho_1, \rho_2)$ the two-particle repulsion. This model is known as a s-wave or Temkin-Poet model [65, 85].

Before the photo-ionization, the atom is in a two-particle ground state. In this s-wave model, it is the eigenstate of

$$\left(-\frac{1}{2}\frac{d^2}{d\rho_1^2} - \frac{1}{2}\frac{d^2}{d\rho_2^2} + V_1(\rho_1) + V_2(\rho_2) + V_{12}(\rho_1, \rho_2)\right) \phi_0(\rho_1, \rho_2) = \lambda_0 \phi_0(\rho_1, \rho_2). \tag{8.11}$$

with the lowest energy. Simultaniously, there are one-particle states that are eigenstates of

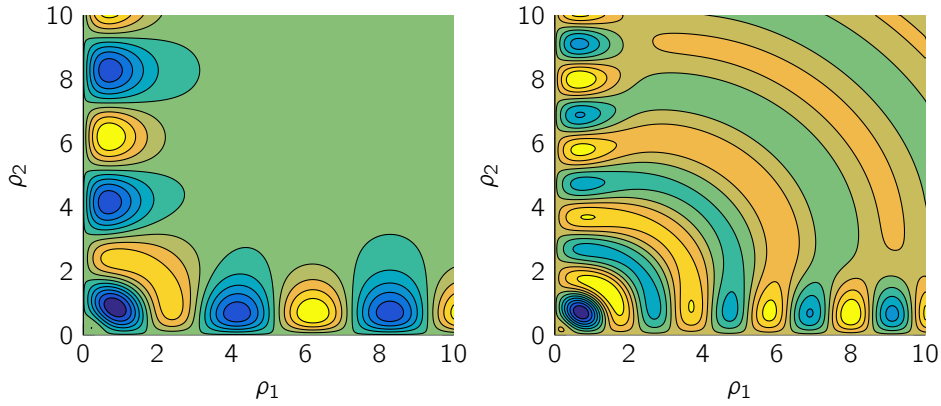
$$\left(-\frac{1}{2}\frac{d^2}{d\rho_1^2} + V_1(\rho_1)\right) \phi_i(\rho_1) = \mu_i \phi_i(\rho_1), \tag{8.12}$$

and

$$\left(-\frac{1}{2}\frac{d^2}{d\rho_2^2} + V_2(\rho_2)\right) \varphi_i(\rho_2) = \nu_i \varphi_i(\rho_2). \tag{8.13}$$

When (8.10) is solved with the energy $E = \hbar\nu + \lambda < 0$, there is only single ionization. Only one of the two coordinates ρ_1 or ρ_2 can become large and the solution, as can be seen in Figure 8.6a, is localized along both axis. The solution is a product of an outgoing wave in one coordinate and a bound state in the other coordinate. For example, along the ρ_2-axis, the solution is described by $A_i(\rho_2)\phi_i(\rho_1)$, where $A_i(\rho_2)$ is a one-dimensional outgoing wave, with an energy $E - \mu_i$ and $\phi_i(\rho_1)$ is a bound state of (8.12) in the first coordinate with energy μ_i. Similary, there is a wave, along the ρ_1 axis, that is an outgoing wave of the form $B_i(\rho_1)\varphi_i(\rho_2)$, with a scattering wave in the first coordinate, ρ_1, and a bound state in the second coordinate ρ_2, where $\varphi_i(\rho_2)$ is the solution of (8.13).

When (8.10) is solved with energy $E = \hbar\nu + \lambda \geq 0$ there is also double ionization and both coordinates ρ_1 and ρ_2 can become large. We see, in Figure 8.6b, a (spherical) wave in



(a) Single ionization (b) Double ionization

Figure 8.6: Left: When the energy $E = \hbar\nu + \lambda_0 < 0$, there is only single ionization. The solution is then localized along the edges, where the solution is a combination of an outgoing wave in the ρ_1 and a bound state in ρ_2, or vice-versa. Right: For energy $E > 0$, there is, in addition to single ionization with solution localized along the edges, a double ionization wave where both coordinates can become large.

the middle of the domain, where both coordinates can be become large. To describe this solution the full coordinate space is necessary. Note that these solutions still show single ionization along the axes. Even for $E > 0$, one particle can take away all the energy and leave the other particle as a bound state.

8.2.4 Coupled channel model for single ionization waves

In this section, we write the single ionization solution as a low-rank decomposition and derive the equations for the low-rank components. When there is only single ionization, the total wave can be written as

$$u(\rho_1, \rho_2) = \sum_{m=1}^{M} \phi_m(\rho_1) A_m(\rho_2) + \sum_{l=1}^{L} B_l(\rho_1) \varphi_l(\rho_2), \tag{8.14}$$

where $\phi_m(\rho_1)$ and $\varphi_l(\rho_2)$ are the bound state eigenstates, defined in (8.12) and (8.13). The first term is localized along the ρ_2-axis, the second term is localized along the ρ_1-axis with $\mu_i < 0$ and $\nu_i < 0$.

As discussed in [12], this expansion is not unique. We can add multiples of $\gamma_m \varphi_m(\rho_2)$ to $A_m(\rho_2)$ and simultaneously subtract $\gamma_l \phi_l(\rho_1)$ from $B_l(\rho_1)$ without contaminating the result. Indeed, for any choice of $\gamma_i \in \mathbb{C}$ and $L = M$ it holds that

$$u(\rho_1, \rho_2) = \sum_{m=1}^{M} \phi_m(\rho_1) \left(A_m(\rho_2) + \gamma_i \varphi_m(\rho_2)\right) + \sum_{l=1}^{L} \left(B_l(\rho_1) - \gamma_i \phi_l(\rho_1)\right) \varphi_l(\rho_2)) = u(\rho_1, \rho_2). \tag{8.15}$$

To make the expansion unique, [12] chooses to select $A_i \perp \varphi_j$ when $j \geq i$ and $B_j \perp \phi_i$ when $i \geq j$. In this chapter, we choose to make the functions in the set $\{\phi_{m \in \{1,\ldots,M\}}, B_{l \in \{1,\ldots,L\}}\}$

orthogonal. We also assume that

$$V_{12}(\rho_1,\rho_2) \approx \sum_{m=1}^{M}\sum_{l=1}^{L} \phi_m(\rho_1)\varphi_l(\rho_2) \iint \phi_m^*(\rho_1)\varphi_l^*(\rho_2)V_{12}(\rho_1,\rho_2)\mathrm{d}\rho_1\mathrm{d}\rho_2.$$

Given a function $f(\rho_1,\rho_2)$, a right hand side, we can now derive the equations for A_m and B_l. When we insert the low-rank decomposition of the expansion (8.14), in the two-dimensional Helmholtz equation (8.10), multiply with ϕ_m^* and integrate over ρ_1 to find

$$(H_2+\mu_m-E)A_m(\rho_2)+\sum_{k=1}^{M}\left(\int_0^\infty \phi_m^*(\rho_1)V_{12}(\rho_1,\rho_2)\phi_k(\rho_1)\mathrm{d}\rho_1\right)A_k(\rho_2)=\int_0^\infty \phi_m^*(\rho_1)f(\rho_1,\rho_2)\mathrm{d}\rho_1$$

for $m=1,2,\ldots,M$ and $\rho_2\in[0,\infty)$. We have used that $\phi_m \perp B_l$ to eliminate the second term in the expansion (8.14).

Similarly, for B_l, we multiply with φ_l^* and integrate over ρ_2 to find

$$(H_1+\nu_l-E)B_l(\rho_1)+\sum_{k=1}^{L}\left(\int_0^\infty \varphi_l^*(\rho_2)V_{12}(\rho_1,\rho_2)\varphi_k(\rho_2)\mathrm{d}\rho_2\right)B_k(\rho_1)=\int_0^\infty \varphi_l^*(\rho_2)\tilde{f}(\rho_1,\rho_2)\mathrm{d}\rho_2$$

for $l=1,2,\ldots,L$ and $\rho_1\in[0,\infty)$. Where the right hand side function f is slightly changed to $\tilde{f}$ to correct for $\phi_m(\rho_1)A_m(\rho_2)$ terms that are not eliminated when multiplied with φ_l^*.

8.3 Low-rank matrix representation of a 2D wave function that includes both single and double ionization

8.3.1 Low rank of the double ionization solution

We now discuss the main result of the chapter and derive a coupled channel equation that gives a low-rank approximation for the double ionization wave function, as shown in Figure 8.6b.

Although the full coordinate space is necessary to describe the double ionization wave function, the rank of this double ionization wave function is low, see Figure 8.7. A numerical verification with increasing low-rank approximations of this double-ionization wave function is shown in Figure 8.8. The different contour plots illustrate indeed that a low-rank decomposition of the wave function could be sufficient to describe the full double ionization wave function of Figure 8.6b.

In Section 8.2.4 we have shown that the single ionization wave can be represented by a low-rank decomposition. In this section, we show that also the double ionization wave can be written as a similar low-rank decomposition.

We first illustrate that the solution of a two-dimensional driven Schrödinger equation that contains both single and double ionization, it is a solution of (8.10) with $E>0$, can be

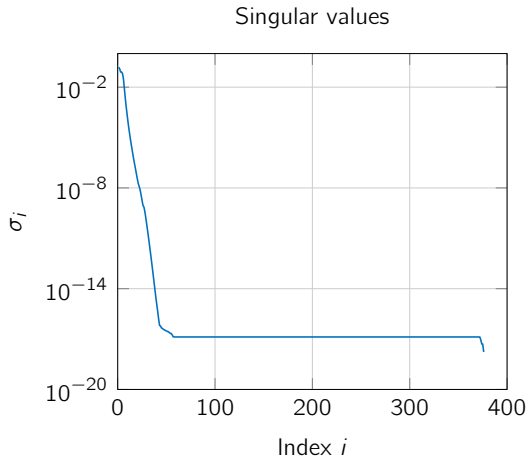
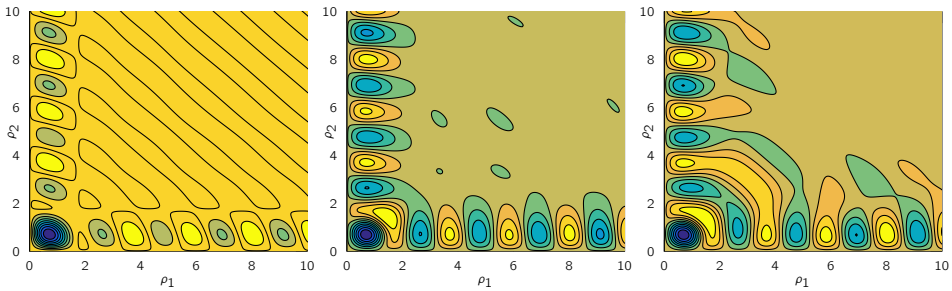
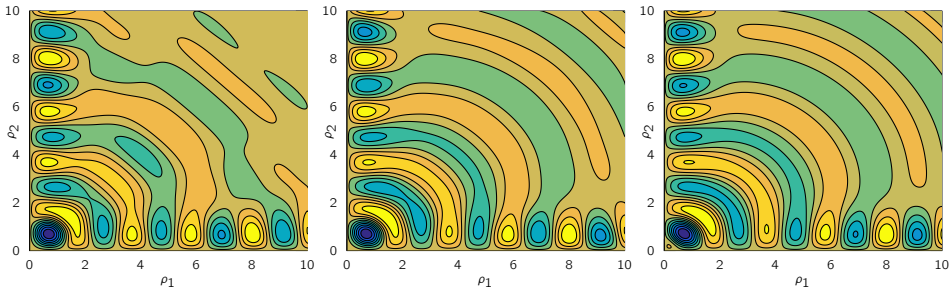


Figure 8.7: Plot of singular values of double ionization wave function.



(a) Rank-1 approximation (b) Rank-2 approximation (c) Rank-3 approximation



(d) Rank-4 approximation (e) Rank-5 approximation (f) Full rank wave

Figure 8.8: Contour plots of the double ionization wave function (bottom, right) and low-rank approximations for increasing rank.

represented by a similar low-rank decomposition. In Figure 8.9 we solve a representative Helmholtz equation on a uniform mesh with a space-dependent wave number, $k(\rho_1, \rho_2)$, in the first quadrant where $\rho_1 \geq 0$ and $\rho_2 \geq 0$. The equation is

$$\left(-\Delta_{2D} - k^2(\rho_1, \rho_2)\right) u_{sc}(\rho_1, \rho_2) = f(\rho_1, \rho_2), \tag{8.16}$$

where Δ_{2D} is the two-dimensional Laplacian and the solution u_{sc} satisfies homogeneous boundary conditions $u_{sc}(\rho_1, 0) = 0$ for all $\rho_1 \geq 0$ and $u_{sc}(0, \rho_2) = 0$ for all $\rho_2 \geq 0$. On the other boundaries outgoing boundary conditions are imposed.

The right hand side $f(\rho_1, \rho_2)$ has a support that is limited to $[0,\ b]^2 \subset \mathbb{R}^2_+$, i.e. $f(\rho_1, \rho_2) = 0$, for all $\rho_1 \geq b$ or $\rho_2 \geq b$.

The wave number $k(\rho_1, \rho_2)$ can be split in a constant part, k_0^2 and variable part $\chi(\rho_1, \rho_2)$. The variable part is also only non-zero on $[0, b)^2$

$$k^2(\rho_1, \rho_2) = \begin{cases} k_0^2\,(1 + \chi(\rho_1, \rho_2)) & \text{if} \quad \rho_1 < b \quad \text{and} \quad \rho_2 < b, \\ k_0^2 & \text{if} \quad \rho_1 \geq b \quad \text{or} \quad \rho_2 \geq b. \end{cases} \tag{8.17}$$

The domain is extended with exterior complex scaling (ECS) absorbing boundary condition [56].

The wave function u_{sc} is discretized on the two-dimensional mesh and can be represented by a matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$. One can compute the singular decomposition of this matrix, $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$ with $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{C}^{n \times n}$ where $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the singular values σ_i on the diagonal.

The results are shown in Figure 8.9 and show that the singular values decrease rapidly. Thus the wave function can efficiently be approximated by a truncated representation,

$$\boldsymbol{A} \approx \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{H}}, \tag{8.18}$$

where $\boldsymbol{u}_i \in \mathbb{C}^n$ are columns of $\boldsymbol{U}$ and $\boldsymbol{v}_i \in \mathbb{C}^n$ are rows from $\boldsymbol{V}$ and σ_i are largest r singular values. Thus $\boldsymbol{A}$ is approximated by it low-rank representation with rank r. This truncated decomposition drops all contributions with $\sigma_i < \tau$ below a threshold τ, for example the expected discretization error.

Finally, Figure 8.10 illustrates that a low-rank approximation to the wave function is sufficient calculate an accurate approximation to the far field or the cross section.

8.3.2 Determining the low-rank components directly

In the example of the previous section, we have first calculated a matrix representation of the solution, $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, and then approximated it by low-rank components. The aim is now to develop a method that calculates, directly, these components without first calculating the full solution $\boldsymbol{A}$. This approach avoids expensive calculations.

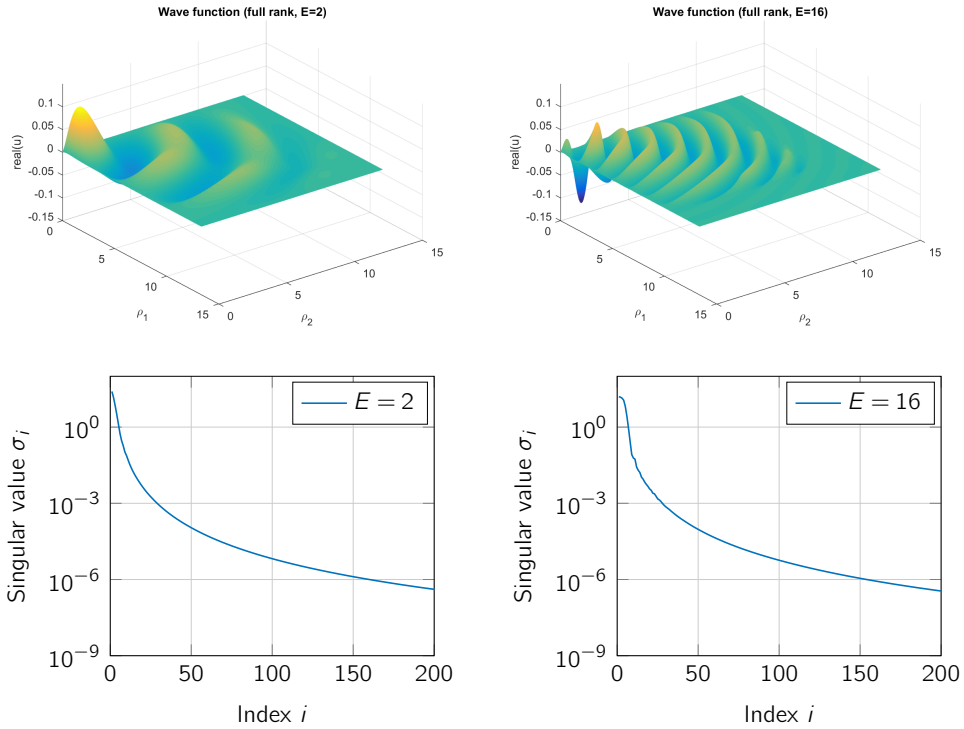


Figure 8.9: The top figures show the representative double ionization wave function for two different energies (i.e. $E = 2$ and $E = 16$). The singular values of the matrix-representation of the discretized functions are shown in the bottom figures. We show the solution of a two-dimensional Helmholtz equation with a space-dependent wave number $k^2(\rho_1, \rho_2)$ given by $k^2(\rho_1, \rho_2) = E - e^{-|\rho_1 - \rho_2|}$. The right hand side $f(\rho_1, \rho_2)$ on the finite domain $[0, b)^2$, with $b = 10$, is given by $f(\rho_1, \rho_2) = -e^{-\rho_1^2 - \rho_2^2}$. Finite difference discretization is done on a uniform mesh with $M = 1000$ interior mesh points per direction. At the boundaries $x = b$ and $y = b$ the domain is extended with exterior complex scaling under an angle $\frac{\pi}{6}$ where 33% additional discretization points are added, so $n = 1333$.

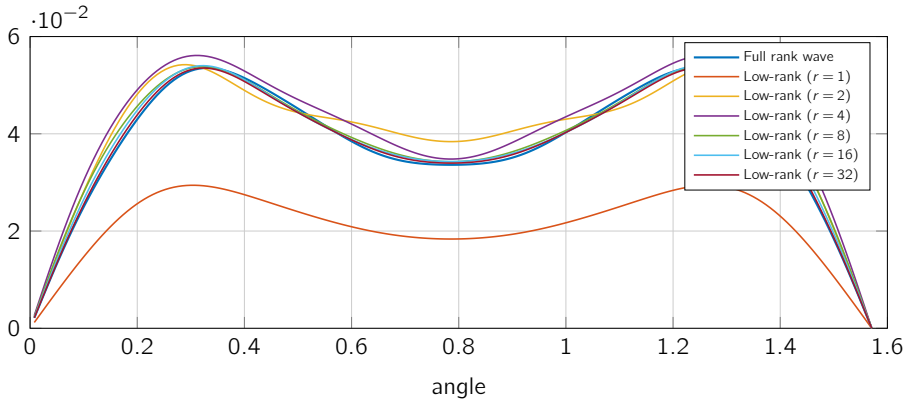


Figure 8.10: The cross section computed from low-rank approximations of the wave function.

We start from the discretized two-dimensional Helmholtz equation as given in (8.16). In matrix form this is given by

$$-\boldsymbol{D}_{xx}\boldsymbol{A} - \boldsymbol{A}\boldsymbol{D}_{yy}^{\mathsf{T}} - \boldsymbol{K} \circ \boldsymbol{A} = \boldsymbol{F}, \tag{8.19}$$

where $\boldsymbol{D}_{xx} \in \mathbb{C}^{n\times n}$ and $\boldsymbol{D}_{yy} \in \mathbb{C}^{n\times n}$ are sparse matrices that represent the discretization of the second derivatives, $\boldsymbol{K}$ is the matrix that represents the space-dependent wave number, $k^2(\rho_1, \rho_2)$, on the grid and $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is the matrix that describes the unknown partial wave. The right hand side $\boldsymbol{F} \in \mathbb{C}^{n\times n}$ is given. The Hadamard product, $\circ$, multiplies the matrices point wise, element by element.

We now make the approximation $\boldsymbol{A} \approx \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}$, with low-rank matrices $\boldsymbol{U} \in \mathbb{C}^{n\times r}$ and $\boldsymbol{V} \in \mathbb{C}^{n\times r}$ where $r \ll n$ and write

$$-\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}} - \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy} - \boldsymbol{K} \circ \left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right) = \boldsymbol{F}. \tag{8.20}$$

We start with a guess for $\boldsymbol{V} \in \mathbb{C}^{n\times r}$ with orthogonal columns such that $\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \boldsymbol{I}_r$. We can now multiply (8.20) from the right by $\boldsymbol{V}$ and obtain

$$-\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} - \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{V} - \left(\boldsymbol{K} \circ \left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right)\boldsymbol{V} = \boldsymbol{F}\boldsymbol{V}. \tag{8.21}$$

where $\boldsymbol{U} \in \mathbb{C}^{n\times r}$ is now the remaining unknown.

We use the vectorizing identities

$$\begin{aligned} &\operatorname{vec}[\boldsymbol{A} \circ \boldsymbol{B}] = \operatorname{vec}[\boldsymbol{A}] \circ \operatorname{vec}[\boldsymbol{B}] \quad &&\text{for } \boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{l\times p}, \\ &\operatorname{vec}[\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}] = \left(\boldsymbol{B}^{\mathsf{T}} \otimes \boldsymbol{A}\right)\operatorname{vec}[\boldsymbol{X}] \quad &&\text{for } \boldsymbol{A} \in \mathbb{C}^{k\times l}, \boldsymbol{X} \in \mathbb{C}^{l\times m} \text{ and } \boldsymbol{B} \in \mathbb{C}^{m\times n}, \end{aligned} \tag{8.22}$$

to obtain

$$-(\boldsymbol{I} \otimes \boldsymbol{D}_{xx})\operatorname{vec}[\boldsymbol{U}] - \left((\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{V})^{\mathsf{T}} \otimes \boldsymbol{I}\right)\operatorname{vec}[\boldsymbol{U}] - \operatorname{vec}\left[\left(\boldsymbol{K} \circ \left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right)\boldsymbol{V}\right] = \operatorname{vec}[\boldsymbol{F}\boldsymbol{V}]. \tag{8.23}$$

The last term of the left hand side can be simplified and written as

$$\begin{aligned}\mathrm{vec}\left[\left(\boldsymbol{K}\circ\left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right)\boldsymbol{V}\right] &= \left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\mathrm{vec}\left[\boldsymbol{K}\circ\left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right],\\ &= \left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\left(\mathrm{vec}\left[\boldsymbol{K}\right]\circ\mathrm{vec}\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right]\right),\\ &= \left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}\right]\right)\mathrm{vec}\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right],\\ &= \left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}\right]\right)\left(\left(\boldsymbol{V}^{\mathsf{H}}\right)^{\mathsf{T}}\otimes\boldsymbol{I}\right)\mathrm{vec}\left[\boldsymbol{U}\right].\end{aligned} \tag{8.24}$$

This results in

$$\left[-\left(\boldsymbol{I}\otimes\boldsymbol{D}_{xx}\right)-\left(\left(\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{V}\right)^{\mathsf{T}}\otimes\boldsymbol{I}\right)-\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}\right]\right)\left(\left(\boldsymbol{V}^{\mathsf{H}}\right)^{\mathsf{T}}\otimes\boldsymbol{I}\right)\right]\mathrm{vec}\left[\boldsymbol{U}\right]=\mathrm{vec}\left[\boldsymbol{F}\boldsymbol{V}\right]. \tag{8.25}$$

This is a linear system for the remaining unknown columns of the matrix $\boldsymbol{U}\in\mathbb{C}^{n\times r}$.

In (8.18), we have approximated $\boldsymbol{A}$ as $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$ where $\boldsymbol{U},\boldsymbol{V}\in\mathbb{C}^{n\times r}$ and $\boldsymbol{\Sigma}\in\mathbb{R}^{r\times r}$ are truncated matrices. With an orthogonal guess for $\boldsymbol{V}$, we solve for a $\boldsymbol{U}$ in (8.25). Since we approximate $\boldsymbol{A}$ now by the product $\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}$, hence $\boldsymbol{U}$, the solution of (8.25), includes the diagonal matrix with singular values.

We can now do a QR decomposition of $\boldsymbol{U}$ to arrive at a guess for the orthogonal matrix $\boldsymbol{U}$.

The next step is to improve the guess for $\boldsymbol{V}$ in a similar way. The equation (8.20) becomes, when we multiply from the left by $\boldsymbol{U}^{\mathsf{H}}$,

$$-\boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}-\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}-\boldsymbol{U}^{\mathsf{H}}\left(\boldsymbol{K}\circ\left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right)=\boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}. \tag{8.26}$$

Using the vectorizing identities (8.22), this results in

$$\left[-\left(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}\right)-\left(\boldsymbol{D}_{yy}^{\mathsf{T}}\otimes\boldsymbol{I}\right)-\left(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\right)\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}\right]\right)\left(\boldsymbol{I}\otimes\boldsymbol{U}\right)\right]\mathrm{vec}\left[\boldsymbol{V}^{\mathsf{H}}\right]=\mathrm{vec}\left[\boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}\right], \tag{8.27}$$

where we use that

$$\begin{aligned}\boldsymbol{U}^{\mathsf{H}}\left(\boldsymbol{K}\circ\left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right) &= \left(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\right)\mathrm{vec}\left[\boldsymbol{K}\right]\circ\mathrm{vec}\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right]\\ &= \left(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\right)\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}\right]\right)\left(\boldsymbol{I}\otimes\boldsymbol{U}\right)\mathrm{vec}\left[\boldsymbol{V}^{\mathsf{H}}\right].\end{aligned}$$

Combining the equations (8.25) and (8.27) we can now propose an algorithm that updates $\boldsymbol{U}$ and $\boldsymbol{V}$ in an alternating way. The steps are described in the Algorithm 4.

8.3.3 Comparison between coupled channel and a low-rank decomposition

We now compare the coupled channel approach from Section 8.2.4 with the low-rank approach from Section 8.3.2. In the coupled channel calculation we use the eigenfunctions of one-particle subsystems as a basis for the expansion. After integration over one of the coordinates this results in a coupled set of one-dimensional equations. Let us illustrate that equation (8.25) and (8.27) reduce to a the coupled channel equations when we choose the eigenfunctions, from (8.12) and (8.13), as columns for $\boldsymbol{U}$ or $\boldsymbol{V}$.

Algorithm 4: Solve for the low-rank matrix decomposition of the solution $\boldsymbol{A} \approx \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}$ of a two-dimensional Helmholtz problem with space-dependent wave number.

Choose $\boldsymbol{V} \in \mathbb{C}^{n \times r}$ as initial guess;
$[\boldsymbol{V}, \boldsymbol{R}] = \mathtt{qr}\,[\boldsymbol{V}, 0]$;
while *not converged* **do**
  Solve $\left[-(\boldsymbol{I} \otimes \boldsymbol{D}_{xx}) - \left((\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{V})^{\mathsf{T}} \otimes \boldsymbol{I}\right) - (\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{diag}\,(\mathtt{vec}\,[\boldsymbol{K}])\left((\boldsymbol{V}^{\mathsf{H}})^{\mathsf{T}} \otimes \boldsymbol{I}\right)\right] \mathtt{vec}\,[\boldsymbol{U}] = \mathtt{vec}\,[\boldsymbol{F}\boldsymbol{V}]$;
  $[\boldsymbol{U}, \boldsymbol{R}] = \mathtt{qr}\,[\boldsymbol{U},\ 0]$;
  Solve $\left[-(\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}) - (\boldsymbol{D}_{yy}^{\mathsf{T}} \otimes \boldsymbol{I}) - (\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}})\,\mathrm{diag}\,(\mathtt{vec}\,[\boldsymbol{K}])\,(\boldsymbol{I} \otimes \boldsymbol{U})\right] \mathtt{vec}\left[\boldsymbol{V}^{\mathsf{H}}\right] = \mathtt{vec}\left[\boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}\right]$;
  $[\boldsymbol{V}, \boldsymbol{R}] = \mathtt{qr}\,[\boldsymbol{V},\ 0]$;
end
$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{R}^{\mathsf{H}}\boldsymbol{V}^{\mathsf{H}}$;

Let use take a look at equations (8.25) and (8.27). Further, assume that the discretized space-dependent wave number is given by

$$\boldsymbol{K} = E\boldsymbol{I} \otimes \boldsymbol{I} - \boldsymbol{I} \otimes V_1(\boldsymbol{x}) - V_2^{\mathsf{T}}(\boldsymbol{y}) \otimes \boldsymbol{I} - V_{12}(\boldsymbol{x}, \boldsymbol{y}).$$

For (8.25) assume that the columns of $\boldsymbol{V}$ are the eigenstates of $-\boldsymbol{D}_{yy} + V_2(\boldsymbol{y})$ with eigenvalues ν_i. We can then write

$$-(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{diag}\,(\mathtt{vec}\,[\boldsymbol{K}])\left((\boldsymbol{V}^{\mathsf{H}})^{\mathsf{T}} \otimes \boldsymbol{I}\right) = -E\boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes V_1(\boldsymbol{x}) + \left(\boldsymbol{V}^{\mathsf{H}}V_2(\boldsymbol{y})\boldsymbol{V}\right)^{\mathsf{T}} \otimes \boldsymbol{I} + (\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{diag}\,(\mathtt{vec}\,[V_{12}(\boldsymbol{x}, \boldsymbol{y})])\left((\boldsymbol{V}^{\mathsf{H}})^{\mathsf{T}} \otimes \boldsymbol{I}\right).$$

Then equation (8.25) becomes

$$\Big[(\boldsymbol{I} \otimes (-\boldsymbol{D}_{xx} + V_1(\boldsymbol{x}) - \boldsymbol{I})) + \left(\left(\boldsymbol{V}^{\mathsf{H}}(-\boldsymbol{D}_{yy} + V_2(\boldsymbol{y}))\boldsymbol{V}\right)^{\mathsf{T}} \otimes \boldsymbol{I}\right) + (\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{diag}\,(\mathtt{vec}\,[V_{12}(\boldsymbol{x}, \boldsymbol{y})])\left((\boldsymbol{V}^{\mathsf{H}})^{\mathsf{T}} \otimes \boldsymbol{I}\right)\Big] \mathtt{vec}\,[\boldsymbol{U}] = \mathtt{vec}\,[\boldsymbol{F}\boldsymbol{V}].$$

When we use that the columns of $\boldsymbol{V}$ are eigenfunctions of $-\boldsymbol{D}_{yy} + V_2(\boldsymbol{y})$ with eigenvalues ν_i, equation (8.25) becomes

$$\Big[\left(\mathrm{diag}\,(\boldsymbol{\nu}) \otimes \boldsymbol{I} + \boldsymbol{I} \otimes (-\boldsymbol{D}_{xx} + V_1(\boldsymbol{x}) - E\boldsymbol{I})\right) + (\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{diag}\,(\mathtt{vec}\,[V_{12}(\boldsymbol{x}, \boldsymbol{y})])\left((\boldsymbol{V}^{\mathsf{H}})^{\mathsf{T}} \otimes \boldsymbol{I}\right)\Big] \mathtt{vec}\,[\boldsymbol{U}] = \mathtt{vec}\,[\boldsymbol{F}\boldsymbol{V}].$$

The term $(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{diag}\,(\mathtt{vec}\,[V_{12}(\boldsymbol{x}, \boldsymbol{y})])\left((\boldsymbol{V}^{\mathsf{H}})^{\mathsf{T}} \otimes \boldsymbol{I}\right)$ couples the columns of $\boldsymbol{U}$. It should be interpreted as a discretized version of $\int \varphi_i^*(\rho_2)V_{12}(\rho_1, \rho_2)\varphi_j(\rho_2)\mathrm{d}\rho_2$, where we integrate over one of the coordinates. The columns of $\boldsymbol{V}$ are the φ_i represented on a integration grid.

However, in general, the columns are $\boldsymbol{V}$ are not eigenfunctions of the operator. The term $\mathrm{diag}\,(\boldsymbol{\nu}) \otimes \boldsymbol{I}$ then becomes a matrix that also couples the different components of $\boldsymbol{U}$.

Similarly, for (8.27) assume that the columns of $\boldsymbol{U}$ are the eigenstates of $-\boldsymbol{D}_{xx} + V_1(\boldsymbol{x})$. We can then write

$$-\left(\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\right)\mathrm{diag}\,(\mathtt{vec}\,[\boldsymbol{K}])\,(\boldsymbol{I} \otimes \boldsymbol{U}) = -E\,\boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}V_1(\boldsymbol{x})\boldsymbol{U} + V_2^{\mathsf{T}}(\boldsymbol{y}) \otimes \boldsymbol{I} + \left(\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\right)\mathrm{diag}\,(\mathtt{vec}\,[V_{12}(\boldsymbol{x}, \boldsymbol{y})])\,(\boldsymbol{I} \otimes \boldsymbol{U}).$$

Then equation (8.27) becomes

$$\Big[\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\big(-\boldsymbol{D}_{xx}+V_1(\boldsymbol{x})\big)\boldsymbol{U}+\big(-\boldsymbol{D}_{yy}+V_2(\boldsymbol{y})-E\boldsymbol{I}\big)^{\mathsf{T}}\otimes\boldsymbol{I} + \big(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\big)\operatorname{diag}\big(\operatorname{vec}\left[V_{12}(\boldsymbol{x},\boldsymbol{y})\right]\big)\left(\boldsymbol{I}\otimes\boldsymbol{U}\right)\Big]\operatorname{vec}\left[\boldsymbol{V}^{\mathsf{H}}\right]=\operatorname{vec}\left[\boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}\right].$$

When we use that the columns of $\boldsymbol{U}$ are eigenfunctions of $-\boldsymbol{D}_{xx}+V_1(\boldsymbol{x})$ with eigenvalues μ_i, equation this becomes

$$\Big[\boldsymbol{I}\otimes\operatorname{diag}\left(\boldsymbol{\mu}\right)+\left(-\boldsymbol{D}_{yy}+V_2(\boldsymbol{y})-E\boldsymbol{I}\right)^{\mathsf{T}}\otimes\boldsymbol{I}+\big(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\big)\operatorname{diag}\big(\operatorname{vec}\left[V_{12}(\boldsymbol{x},\boldsymbol{y})\right]\big)\left(\boldsymbol{I}\otimes\boldsymbol{U}\right)\Big]\operatorname{vec}\left[\boldsymbol{V}^{\mathsf{H}}\right] = \operatorname{vec}\left[\boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}\right].$$

The term $\big(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\big)\operatorname{diag}\big(\operatorname{vec}\left[V_{12}(\boldsymbol{x},\boldsymbol{y})\right]\big)\left(\boldsymbol{I}\otimes\boldsymbol{U}\right)$ couples the columns of $\boldsymbol{V}$ and should be interpreted as a discretized version of $\int\phi_i^*(\rho_1)V_{12}(\rho_1,\rho_2)\phi_j(\rho_1)\mathrm{d}\rho_1$, where we integrate over one of the coordinates. The columns of $\boldsymbol{U}$ are the ϕ_i represented on a integration grid. In general, the columns are $\boldsymbol{U}$ are not eigenfunctions of the operator. The term $\boldsymbol{I}\otimes\operatorname{diag}\left(\boldsymbol{\mu}\right)$ then becomes a matrix that also couples the different components of $\boldsymbol{V}$.

Thus in short, in each iteration of the alternating method we are solving generalized coupled channel equations.

8.3.4 Convergence with projection operators

We will now write both linear systems, (8.25) for $\operatorname{vec}\left[\boldsymbol{U}\right]$, and, (8.27) for $\operatorname{vec}\left[\boldsymbol{V}\right]$, as projection operators applied to the residual of the matrix equation, (8.19). For sufficiently large rank k the alternating projection approach will converge to a solution with zero residual.

We denote by $\boldsymbol{L}$ the discretized two-dimensional Helmholtz operator on the full grid

$$\boldsymbol{L}=\left(\boldsymbol{I}\otimes\left(-\boldsymbol{D}_{xx}\right)\right)+\left(\left(-\boldsymbol{D}_{yy}\right)\otimes\boldsymbol{I}\right)-\operatorname{diag}\left(\operatorname{vec}\left[\boldsymbol{K}\right]\right)\left(\boldsymbol{I}\otimes\boldsymbol{I}\right). \tag{8.28}$$

We can now explicitly write equation (8.25) in terms of projections and this linear operator:

$$\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\operatorname{vec}\left[\boldsymbol{U}\right]=\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\operatorname{vec}\left[\boldsymbol{F}\right]. \tag{8.29}$$

The residual matrix, $\boldsymbol{R}$, is given by

$$\boldsymbol{R}=\boldsymbol{F}-\left(-\boldsymbol{D}_{xx}\right)\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}-\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\left(-\boldsymbol{D}_{yy}\right)+\boldsymbol{K}\circ\left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right). \tag{8.30}$$

In vector form this reads, using that $\boldsymbol{U}$ is a solution of equation (8.29),

$$\begin{aligned}
\operatorname{vec}\left[\boldsymbol{R}\right] &= \operatorname{vec}\left[\boldsymbol{F}\right]-\left(\overline{\boldsymbol{V}}\otimes\left(-\boldsymbol{D}_{xx}\right)+\left(-\boldsymbol{D}_{yy}^{\mathsf{T}}\right)\overline{\boldsymbol{V}}\otimes\boldsymbol{I}-\operatorname{diag}\left(\operatorname{vec}\left[\boldsymbol{K}\right]\right)\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\right)\operatorname{vec}\left[\boldsymbol{U}\right],\\
&= \Big(\boldsymbol{I}-\left(\overline{\boldsymbol{V}}\otimes\left(-\boldsymbol{D}_{xx}\right)+\left(-\boldsymbol{D}_{yy}^{\mathsf{T}}\right)\overline{\boldsymbol{V}}\otimes\boldsymbol{I}-\operatorname{diag}\left(\operatorname{vec}\left[\boldsymbol{K}\right]\right)\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\right)\left[\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\right]^{-1}\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\Big)\operatorname{vec}\left[\boldsymbol{F}\right]\\
&= \Big(\boldsymbol{I}-\boldsymbol{L}\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\left[\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\right]^{-1}\left(\boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\right)\Big)\operatorname{vec}\left[\boldsymbol{F}\right]\\
&= P_{\boldsymbol{V}}\operatorname{vec}\left[\boldsymbol{F}\right],
\end{aligned}$$

where $P_{\boldsymbol{V}}$ is given by

$$\begin{aligned} P_{\boldsymbol{V}} &:= \boldsymbol{I} - \boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\left[\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\right]^{-1}\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right) \\ &:= \boldsymbol{I} - \boldsymbol{X}. \end{aligned} \tag{8.31}$$

The operator $P_{\boldsymbol{V}}$ is a projection operator. Indeed, observe that the terms between the two inverses cancel, in the next equation, against one of the inverse factors:

$$\begin{aligned} \boldsymbol{X}^2 &= \boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\left[\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\right]^{-1}\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\left[\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\right]^{-1}\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right) \\ &= \boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\left[\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right)\boldsymbol{L}\left(\overline{\boldsymbol{V}} \otimes \boldsymbol{I}\right)\right]^{-1}\left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right) \\ &= \boldsymbol{X}. \end{aligned}$$

We then have that $P_{\boldsymbol{V}}^2 = (1-\boldsymbol{X})(1-\boldsymbol{X}) = 1 - 2\boldsymbol{X} + \boldsymbol{X}^2 = 1 - \boldsymbol{X} = P_{\boldsymbol{V}}$.

This projection operator removes all components from the residual matrix that can be corrected by the subspace spanned by $\boldsymbol{V}$. It is similar as a deflation operator, often used in preconditioning [21].

A similar derivation results in a projection operator $P_{\boldsymbol{U}}$ for the update of $\boldsymbol{V}$:

$$\begin{aligned} P_{\boldsymbol{U}} &= \boldsymbol{I} - \boldsymbol{L}\left(\boldsymbol{I} \otimes \boldsymbol{U}\right)\left[\left(\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{L}\left(\boldsymbol{I} \otimes \boldsymbol{U}\right)\right]^{-1}\left(\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\right) \\ &= \boldsymbol{I} - \boldsymbol{Y}. \end{aligned} \tag{8.32}$$

This is again a projection operator.

So, Algorithm 4 repeatedly projects the residual matrix, $\boldsymbol{R}$, on a subspace. Alternating between a subspace that is orthogonal to the subspace spanned by the columns of $\boldsymbol{V}^{(k)}$ at iteration k and a subspace that is orthogonal to the columns of $\boldsymbol{U}^{(k)}$ at iteration k. The residual matrix $\boldsymbol{R}^{(k)}$ after k iterations is the result of a series of projections

$$\boldsymbol{R}^{(k)} = P_{\boldsymbol{U}^{(k)}} P_{\boldsymbol{V}^{(k)}} P_{\boldsymbol{U}^{(k-1)}} P_{\boldsymbol{V}^{(k)}} \ldots P_{\boldsymbol{U}^{(0)}} P_{\boldsymbol{V}^{(0)}} \boldsymbol{R}^{(0)}. \tag{8.33}$$

It is similar to the *method of Alternating Projections* [93] that goes back to Neumann [59], where a solution is projected on two alternating subspaces resulting in a solution that lies in the intersection between the two spaces. However, here the columns for $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$ are changing each iteration.

However, when the rank of $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$ is sufficiently large, the only intersection between the changing subspaces is the $\boldsymbol{0}$ matrix. So the residual converges to 0.

8.4 Solving for the low-rank tensor approximation of 3D Helmholtz equations

Algorithm 4, as introduced in Section 8.3.2 for two-dimensional problems, can be extended to higher dimensions. We illustrate this extension for the three-dimensional Helmholtz problems. First, we will discuss the problem with a constant wave number and then extend the results to space-dependent wave numbers.

We solve the Helmholtz equation with a constant or space-dependent wave number, $k^2(\rho_1,\rho_2,\rho_3)$, in the first quadrant, where $\rho_1 \geq 0, \rho_2 \geq 0$ and $\rho_3 \geq 0$. The driven equation is given by

$$\left(-\Delta_{3D} - k^2(\rho_1,\rho_2,\rho_3)\right) u_{sc}(\rho_1,\rho_2,\rho_3) = f(\rho_1,\rho_2,\rho_3), \tag{8.34}$$

where Δ_{3D} is the three-dimensional Laplacian and the solution u_{sc} satisfies homogeneous boundary conditions $u_{sc}(0,\rho_2,\rho_3) = 0$ for all $\rho_2,\rho_3 \geq 0$, $u_{sc}(\rho_1,0,\rho_3) = 0$ for all $\rho_1,\rho_3 \geq 0$ and $u_{sc}(\rho_1,\rho_2,0) = 0$ for all $\rho_1,\rho_2 \geq 0$. On the other boundaries outgoing boundary conditions are applied.

The right hand side $f(\rho_1,\rho_2,\rho_3)$ has a support that is limited to $[0,b]^3 \subset \mathbb{R}^3_+$, i.e. $f(\rho_1,\rho_2,\rho_3) = 0$, for all $\rho_1 \geq b$, $\rho_2 \geq b$ or $\rho_3 \geq b$.

The wave number $k^2(\rho_1,\rho_2,\rho_3)$ can be split in a constant part, k_0^2, and variable part $\chi(\rho_1,\rho_2,\rho_3)$. The variable part is also only non-zero on $[0,b)^3$

$$k^2(\rho_1,\rho_2,\rho_3) = \begin{cases} k_0^2\left(1+\chi(\rho_1,\rho_2,\rho_3)\right) & \text{if} \quad \rho_1 < b \quad \text{and} \quad \rho_2 < b \quad \text{and} \quad \rho_3 < b, \\ k_0^2 & \text{if} \quad \rho_1 \geq b \quad \text{or} \quad \rho_2 \geq b \quad \text{or} \quad \rho_3 \geq b. \end{cases} \tag{8.35}$$

The domain is extended with exterior complex scaling (ECS) absorbing boundary condition [56].

The wave function is discretized on a three-dimensional mesh with $n_1 \times n_2 \times n_3$ unknowns and can be represented by a Tucker tensor decomposition [86]. The Tucker tensor $\mathcal{M} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ with multi-linear rank $\boldsymbol{r} = (r_1, r_2, r_3)$ is given by

$$\mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \in \mathbb{C}^{n_1 \times n_2 \times n_3}. \tag{8.36}$$

Here the tensor $\mathcal{G} \in \mathbb{C}^{r_1 \times r_2 \times r_3}$ is called the core tensor and the factor matrices $\boldsymbol{U}_i \in \mathbb{C}^{n_i \times r_i}$ have orthonormal columns for $i = 1,2,3$. Here r_i refers to the rank for each direction and n_i to the number of mesh points in each direction. So, to store this tensor only one core tensor and d factor matrices need to be stored, so the storage costs[1] scales $\mathcal{O}\left(r^d + dnr\right)$.

Let $\mathcal{L}$ be the discretization of the three-dimensional Helmholtz operator as given in (8.34). Observe that the operator $\mathcal{L}$ can be written as a sum of Kronecker-products, where the matrix representation $\boldsymbol{L}$ is of the following form

$$\boldsymbol{L} = -\boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{D}_{xx} - \boldsymbol{I} \otimes \boldsymbol{D}_{yy} \otimes \boldsymbol{I} - \boldsymbol{D}_{zz} \otimes \boldsymbol{I} \otimes \boldsymbol{I} - \operatorname{diag}\left(\operatorname{vec}[\mathcal{K}]\right) \boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{I}. \tag{8.37}$$

Here $\boldsymbol{D}_{xx} \in \mathbb{C}^{n_1 \times n_1}$, $\boldsymbol{D}_{yy} \in \mathbb{C}^{n_2 \times n_2}$ and $\boldsymbol{D}_{zz} \in \mathbb{C}^{n_3 \times n_3}$ are sparse matrices that represent the discretization of the second derivatives and $\mathcal{K}$ is a tensor that represents the constant or space-dependent wave number, $k^2(\rho_1,\rho_2,\rho_3)$, discretized on the grid.

8.4.1 Helmholtz equation with constant wave number

First, consider the three-dimensional Helmholtz problem with a constant wave number, so $k^2(\rho_1,\rho_2,\rho_3) \equiv k^2$. The application of the Helmholtz operator $\mathcal{L}$ on tensor $\mathcal{M}$ in Tucker

[1] Here, we used the notation $r^d := \prod_{i=1}^d r_i$ and $r := \sqrt[d]{r^d}$, to deal with possible unequal number of discretization points n_i or ranks r_i in different directions for a Tucker tensor.

tensor format is given by

$$\begin{aligned}\mathcal{L}\mathcal{M} &= \mathcal{F} \\ \mathcal{L}\mathcal{M} &= -\mathcal{G}\times_1 \boldsymbol{D}_{xx}\boldsymbol{U}_1\times_2\boldsymbol{U}_2\times_3\boldsymbol{U}_3 \\ &\quad -\mathcal{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{D}_{yy}\boldsymbol{U}_2\times_3\boldsymbol{U}_3 \\ &\quad -\mathcal{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{U}_2\times_3\boldsymbol{D}_{zz}\boldsymbol{U}_3 \\ &\quad -k^2\mathcal{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{U}_2\times_3\boldsymbol{U}_3 \\ &= \mathcal{F}\end{aligned} \tag{8.38}$$

where $\boldsymbol{U}_i^{\mathsf{H}}\boldsymbol{U}_i = \boldsymbol{I}$ for $i = 1,2,3$ and $\mathcal{F}$ is a tensor representation of the right hand side function f discretized on the grid.

8.4.1.1 Solving for product of basis functions and core terms (version 1)

Similar to the two-dimensional case, we can derive equations to iteratively solve for the factors $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$. To derive the equations for $\boldsymbol{U}_1$ we start from (8.38) and multiply with $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ in the second and third direction, respectively:

$$\mathcal{L}\mathcal{M}\times_2\boldsymbol{U}_2^{\mathsf{H}}\times_3\boldsymbol{U}_3^{\mathsf{H}} = \mathcal{F}\times_2\boldsymbol{U}_2^{\mathsf{H}}\times_3\boldsymbol{U}_3^{\mathsf{H}}.$$

For a review of tensors, tensor decompositions and tensor operations like this tensor-times-matrix product denoted by the symbol $\times_i$ we refer to Chapter 7 or [48].

Using explicitly the Tucker tensor representation for $\mathcal{M}$ and that the columns of $\boldsymbol{U}_i$ are orthonormal, the following expression is derived for the Helmholtz operator applied on a tensor in Tucker format

$$\mathcal{L}\mathcal{M}\times_2\boldsymbol{U}_2^{\mathsf{H}}\times_3\boldsymbol{U}_3^{\mathsf{H}} = \mathcal{G}\times_1(-\boldsymbol{D}_{xx}-k^2\boldsymbol{I})\boldsymbol{U}_1 - \mathcal{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2 - \mathcal{G}\times_1\boldsymbol{U}_1\times_3\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3.$$

Writing this tensor equation in the first unfolding leads to a matrix equation, recall the first unfolding is given by $\boldsymbol{M}_{(1)} = \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}(\boldsymbol{U}_3\otimes\boldsymbol{U}_2)^{\mathsf{H}}$, see also [48]:

$$\overline{(-\boldsymbol{D}_{xx}-k^2\boldsymbol{I})\boldsymbol{U}_1}\boldsymbol{G}_{(1)} - \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\left(I\otimes\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}} - \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes I\right)^{\mathsf{H}} = \boldsymbol{F}_{(1)}(\boldsymbol{U}_3\otimes\boldsymbol{U}_2). \tag{8.39}$$

To solve this equation for $\boldsymbol{U}_1$, it is written in vectorized form as

$$\begin{aligned}\left\{\boldsymbol{I}\otimes\overline{(-\boldsymbol{D}_{xx}-k^2\boldsymbol{I})} + \left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)} - \overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes\boldsymbol{I}\right)}\right]\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}}_{\boldsymbol{X}_1}\right] \\ = \operatorname{vec}\left[\boldsymbol{F}_{(1)}(\boldsymbol{U}_3\otimes\boldsymbol{U}_2)\right].\end{aligned} \tag{8.40}$$

Observe that this is a square system with $n_1\times r_2 r_3$ unknowns, where the solution in matrix form $\boldsymbol{X}_1$ could have, in general, a rank $r > r_1$. In a similar way, equations for $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are derived by multiplying (8.38) with the other factor matrices in the appropriate directions:

$$\begin{aligned}\left\{\boldsymbol{I}\otimes\overline{(-\boldsymbol{D}_{yy}-k^2\boldsymbol{I})} + \left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}_1\right)} - \overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes\boldsymbol{I}\right)}\right]\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_2}\boldsymbol{G}_{(2)}}_{\boldsymbol{X}_2}\right] \\ = \operatorname{vec}\left[\boldsymbol{F}_{(2)}(\boldsymbol{U}_3\otimes\boldsymbol{U}_1)\right],\end{aligned} \tag{8.41}$$

Algorithm 5: Solve for the low-rank tensor decomposition of the solution $\mathcal{M}$ of a three-dimensional Helmholtz with constant wave number (version 1).

```
[G, U1, U2, U3] = hosvd(initial guess);
while not converged do
    for i = 1, 2, 3 do
        Solve for X_i = Ū_i G_(i) ∈ ℂ^{n × r^{d-1}} using (8.40), (8.41) or (8.42);
        Ū_i G_(i) = qr[X_i(:, 1 : r_i), 0];
    end
end
G = reconstruct[G_(i), i];
M = G ×_1 U_1 ×_2 U_2 ×_3 U_3;
```

and

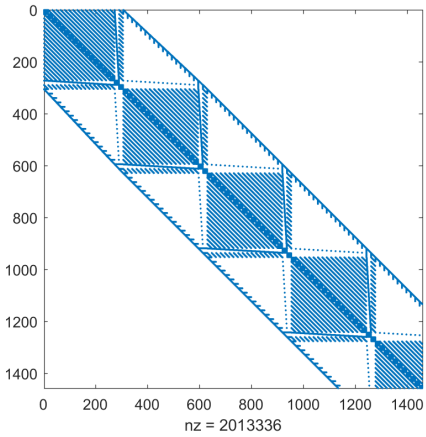
$$\left\{\boldsymbol{I}\otimes\overline{(-\boldsymbol{D}_{zz}-k^2\boldsymbol{I})}+\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}_1\right)}-\overline{\left(\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\otimes\boldsymbol{I}\right)}\right]\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_3}\boldsymbol{G}_{(3)}}_{\boldsymbol{X}_3}\right] \tag{8.42}$$
$$=\operatorname{vec}\left[\boldsymbol{F}_{(3)}\left(\boldsymbol{U}_2\otimes\boldsymbol{U}_1\right)\right].$$

Alternating between solving for $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ using (8.40), (8.41) or (8.42) results in algorithm that approximates the low-rank solutions for three-dimensional problems as given in (8.34). This algorithm is summarized in Algorithm 5. Also in the three-dimensional case the orthogonality of the columns of $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are maintained by additional QR factorizations. Observe that we solve for a large matrix $\boldsymbol{X}_i \in \mathbb{C}^{n_i \times r_1 r_2 r_3 / r_i}$. So, in general the rank of this matrix could be $\min(n_i,\ r_1 r_2 r_3 / r_i)$. But it is also known that $\boldsymbol{X}_i = \overline{\boldsymbol{U}_i}\boldsymbol{G}_{(i)}$ which leads to the fact that the rank of $\boldsymbol{X}_i$ should be at most r_i. Selecting the first r_i columns of $\boldsymbol{X}_i$ and computing its QR decomposition is sufficient to derive a new orthonormal basis as factor matrix $\overline{\boldsymbol{U}}_i$.

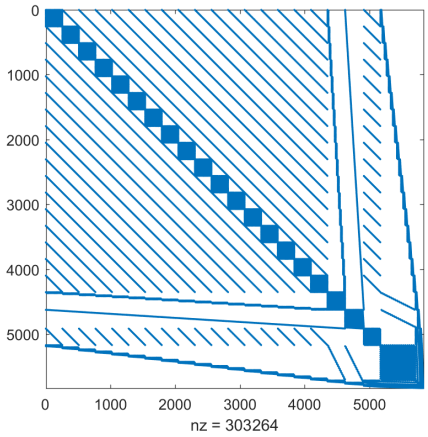
Finally, observe that solving for $\boldsymbol{X}_i$ using (8.40), (8.41) or (8.42) is computationally not efficient. In all iterations, we solve for a total of $\mathcal{O}\left(dnr^{d-1}\right)$ unknowns, while there are only $\mathcal{O}\left(r^d + dnr\right)$ unknowns in the Tucker tensor factorization. Furthermore, solving equations (8.40), (8.41) and (8.42) is also expensive. Indeed, computing a symmetric reverse Cuthill-McKee permutation of the system matrix one observes a matrix with a bandwidth $\mathcal{O}\left(r^{d-1}\right)$. For example when $d = 3, n_i = n = 168, r_i = r = 18$ one obtains the sparsity pattern on the diagonal of the matrix as shown in Figure 8.11a. So solving a system as given in (8.40), (8.41) or (8.42) has a computational cost of $\mathcal{O}\left(nr^{2(d-1)}\right)$.

8.4.1.2 Solving for the basis functions and the core tensor separately (version 2)

To circumvent solving the large systems in (8.40), (8.41) and (8.42), we can pre-compute the QR factorization of the unfolding of the core tensor, $\boldsymbol{G}_{(i)}$, and project the equations onto the obtained $\boldsymbol{Q}_i$. Indeed, this will further reduce the number of unknowns in these linear systems to exactly the number of unknowns that are needed for the factor matrices $\overline{\boldsymbol{U}}_i$, for $i = 1, 2, 3$.



(a) Sparsity pattern of the top of the symmetric reverse Cuthill-McKee permutation of the system matrix to solve for $\boldsymbol{X}_i$ using (8.40). Note: only the first 4.5 of the 168 blocks are shown.



(b) Sparsity pattern of the symmetric reverse Cuthill-McKee permutation of the system matrix to solve for $\boldsymbol{G}_{(1)}$ using (8.46).

Figure 8.11: Sparsity patterns of the symmetric reverse Cuthill-McKee permutation of certain system matrices ($d = 3, n = 168, r = 18$).

Let us discuss the details. We start again from equation (8.39) and use the QR factorization of $\boldsymbol{G}_{(1)}^{\mathsf{H}}$:

$$\boldsymbol{Q}_1 \boldsymbol{R}_1^{\mathsf{H}} = \mathtt{qr}\left[\boldsymbol{G}_{(1)}^{\mathsf{H}}\right].$$

This yields

$$\overline{(-\boldsymbol{D}_{xx} - k^2\boldsymbol{I})\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}} - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(I \otimes \boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}} - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes I\right)^{\mathsf{H}} = \boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right).$$

Post multiplication of this equation by $\boldsymbol{Q}_1$ yields

$$\overline{(-\boldsymbol{D}_{xx} - k^2\boldsymbol{I})\boldsymbol{U}_1}\boldsymbol{R}_1 - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(I \otimes \boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}}\boldsymbol{Q}_1 - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes I\right)^{\mathsf{H}}\boldsymbol{Q}_1 = \boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)\boldsymbol{Q}_1.$$

To solve this equation for $\boldsymbol{U}_1$, it is written is vectorized form as

$$\left\{\boldsymbol{I} \otimes \overline{(-\boldsymbol{D}_{xx} - k^2\boldsymbol{I})} + \boldsymbol{Q}_1^{\mathsf{T}}\left[-\overline{\left(\boldsymbol{I} \otimes \boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)} - \overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes \boldsymbol{I}\right)}\right]\overline{\boldsymbol{Q}_1} \otimes \boldsymbol{I}\right\}\mathtt{vec}\left[\underbrace{\overline{\boldsymbol{U}_1}\boldsymbol{R}_1}_{\boldsymbol{X}_1}\right] = \mathtt{vec}\left[\boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)\boldsymbol{Q}_1\right]. \tag{8.43}$$

In a similar way, the update equations for $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are derived by multiplying (8.38) with the other factor matrices in the appropriate dimensions and using the QR factorizations of

$\boldsymbol{G}_{(i)}^{\mathsf{H}}$:

$$\left\{\boldsymbol{I}\otimes\overline{(-\boldsymbol{D}_{yy}-k^2\boldsymbol{I})}+\boldsymbol{Q}_2^{\mathsf{T}}\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}_1\right)}-\overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes\boldsymbol{I}\right)}\right]\overline{\boldsymbol{Q}_2}\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_2}\boldsymbol{R}_2}_{\boldsymbol{X}_2}\right] = \operatorname{vec}\left[\boldsymbol{F}_{(2)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_1\right)\boldsymbol{Q}_2\right] \tag{8.44}$$

and

$$\left\{\boldsymbol{I}\otimes\overline{(-\boldsymbol{D}_{zz}-k^2\boldsymbol{I})}+\boldsymbol{Q}_3^{\mathsf{T}}\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}_1\right)}-\overline{\left(\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\otimes\boldsymbol{I}\right)}\right]\overline{\boldsymbol{Q}_3}\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_3}\boldsymbol{R}_3}_{\boldsymbol{X}_3}\right] = \operatorname{vec}\left[\boldsymbol{F}_{(3)}\left(\boldsymbol{U}_2\otimes\boldsymbol{U}_1\right)\boldsymbol{Q}_3\right]. \tag{8.45}$$

All these equations are cheap to solve. Indeed, $\operatorname{vec}\left[\overline{\boldsymbol{U}_i}\boldsymbol{R}_i\right]$ has length $n_i r_i$. Computing a symmetric reverse Cuthill-McKee permutation of these system matrices one observes a matrix with a bandwidth $\mathcal{O}(r)$, so solving these equations has a computational cost $\mathcal{O}\left(nr^2\right)$.

Of course, this only updates the factor matrices as basis vectors in each direction. As a single final step, we still have to compute the core tensor $\mathcal{G}$. This will be the computationally most expensive part.

The core tensor $\mathcal{G}$ can be obtained by multiplying (8.38) with all the d factor matrices in the matching directions. Unfolding this equation in a certain direction (eg. the first folding) leads again to a matrix equation. In vectorized form, it is given by

$$\left\{\boldsymbol{I}\otimes\overline{\boldsymbol{U}_1^{\mathsf{H}}\left(-\boldsymbol{D}_{xx}-k^2\boldsymbol{I}\right)\boldsymbol{U}_1}+\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)}-\overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes\boldsymbol{I}\right)}\right]\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\boldsymbol{G}_{(1)}\right] = \operatorname{vec}\left[\boldsymbol{U}_1^{\mathsf{T}}\boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)\right]. \tag{8.46}$$

Indeed, considering again an example where $d=3, n_i=n=168, r_i=r=18$ one obtains a matrix with a sparsity pattern that is shown in Figure 8.11b. Hence, this matrix has not a limited bandwidth anymore. It coupled all functions to all other functions. Although this equation has to be solved only once in the algorithm, when the rank increases, it will rapidly dominate the computational cost of this algorithm.

8.4.1.3 Efficient combination of versions 1 and 2 into new algorithm (version 3)

In the first version of the algorithm, see Section 8.4.1.1 and Algorithm 5, an update for $\boldsymbol{G}_{(i)}$ is computed for each direction in each iteration. This leads to a too expensive algorithm. Then we changed the algorithm such that the costs for the updates in each direction is reduced, see Section 8.4.1.2 and Algorithm 6. But, in that version almost all information for a full update of core tensor $\mathcal{G}$ is lost. Therefore a final, but potentially too expensive, equation needs to be solved.

Observe that the expensive computation for the full core tensor, in version 2, can now be replaced by a single solve per iteration as done in version 1. This leads to a third version of the algorithm. It avoids repeatedly solving the large systems (like version 1) and it

Algorithm 6: Solve for the low-rank tensor decomposition of the solution $\mathcal{M}$ of a three-dimensional Helmholtz problem with constant wave number (version 2).

$[\mathcal{G}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3] = \text{hosvd(initial guess)};$
while *not converged* **do**
  for *i = 1, 2, 3* **do**
    $\boldsymbol{Q}_i \widetilde{\boldsymbol{R}} = \texttt{qr}\left[\boldsymbol{G}_{(i)}^{\mathsf{H}}, 0\right];$
    Solve for $\boldsymbol{X}_i = \overline{\boldsymbol{U}_i}\boldsymbol{R}_i \in \mathbb{C}^{n_i \times r_i}$ using (8.43), (8.44) or (8.45);
    $\overline{\boldsymbol{U}_i}\boldsymbol{R}_i = \texttt{qr}\left[\boldsymbol{X}_i, 0\right];$
    $\mathcal{G} = \texttt{reconstruct}\left[\boldsymbol{R}_i \boldsymbol{Q}_i^{\mathsf{H}},\ i\right];$
  end
end
Solve for $\boldsymbol{G}_{(1)} \in \mathbb{C}^{r_1 \times r_2 r_3}$ using (8.46);
$\mathcal{G} = \texttt{reconstruct}\left[\boldsymbol{G}_{(1)},\ 1\right];$
$\mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3;$

does not solve too expensive systems (like version 2). The computational complexity of this algorithm is equal to the complexity of version 1, so $\mathcal{O}\left(nr^{2(d-1)}\right)$. Furthermore, the systems that need to be solved, each iteration, have exactly the same number of unknowns as the representation of the tensor in low-rank Tucker tensor format. In summary, this final version of the algorithm is given by Algorithm 7.

8.4.1.4 Numerical comparison of three versions for 3D Helmholtz equation

Consider a three-dimensional domain $\Omega = [-10,\ 10]^3$ that is discretized with $M = 100$ equidistant mesh points per direction in the interior of the domain. The domain is extended with exterior complex scaling to implement the absorbing boundary conditions. Hence, in total there are $n = n_x = n_y = n_z = 168$ unknowns per direction. As constant wave number we use $k^2 = 4$ and a right hand side $f(\rho_1, \rho_2, \rho_3) = -e^{-\rho_1^2 - \rho_2^2 - \rho_3^2}$. By symmetry, we expect a low-rank factorization with a equal rank in each direction, so we fix $r = r_x = r_y = r_x$.

The convergence of the residuals of the three versions are given in the left column of Figure 8.12. It is clear that all three versions converge to a good low-rank approximation of the full solution. By increasing the maximal attainable rank r, a better low-rank solution is obtained, as expected. Remarkably, for $r = 30$, in version 2, the final residual is larger then the residuals obtained by both other algorithms while the compute-time for version 2 is larger than the other algorithms.

The compute-time for the most time-consuming parts in the different versions of the algorithm can be measured as a function of the maximal attainable rank r. For the three versions of the algorithm the runtimes are shown in the right column of Figure 8.12. For all parts the expected and measured dependence on the rank r are given. For all versions of the algorithm 10 iterations are applied.

Comparing the total runtime for the three different versions one obtains results as shown in Figure 8.13. Indeed, as expected version 3 is approximately 3 times faster than version 1

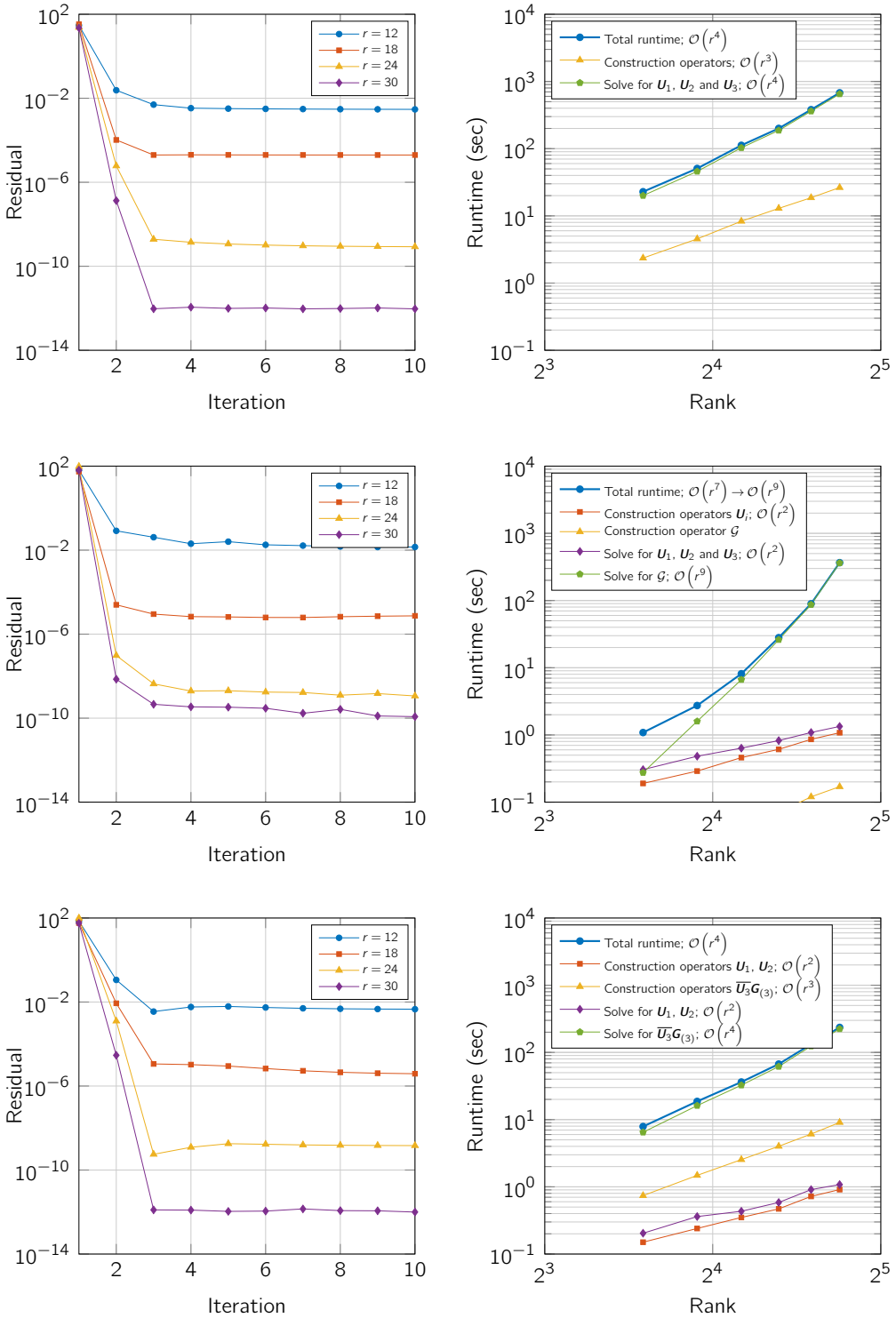


Figure 8.12: Left: Plot of residual per iteration for constant wave number in three-dimensional Helmholtz problem. Right: Plot of runtime of most time consuming parts for constant wave number in three-dimensional Helmholtz problem. Both problems have $M = 100$. Top: Algorithm 5 (version 1), middle: Algorithm 6 (version 2), bottom: Algorithm 7 (version 3).

Algorithm 7: Solve for the low-rank tensor decomposition of the solution $\mathcal{M}$ of a three-dimensional Helmholtz problem with constant wave number (version 3).

$[\mathcal{G}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3] = \text{hosvd(initial guess)};$
while *not converged* **do**
  for $i = 1, 2$ **do**
    $\boldsymbol{Q}_i \widetilde{\boldsymbol{R}} = \mathrm{qr}\left[\boldsymbol{G}_{(i)}^{\mathsf{H}}, 0\right];$
    Solve for $\boldsymbol{X}_i = \overline{\boldsymbol{U}_i}\boldsymbol{R}_i \in \mathbb{C}^{n_i \times r_i}$ using (8.43) or (8.44);
    $\overline{\boldsymbol{U}_i}\boldsymbol{R}_i = \mathrm{qr}\left[\boldsymbol{X}_i, 0\right];$
    $\mathcal{G} = \texttt{reconstruct}\left[\boldsymbol{R}_i \boldsymbol{Q}_i^{\mathsf{H}},\ i\right];$
  end
  Solve for $\boldsymbol{X}_3 = \overline{\boldsymbol{U}_3}\boldsymbol{G}_{(3)} \in \mathbb{C}^{n_3 \times r^{d-1}}$ using (8.42);
  $\overline{\boldsymbol{U}_3}\boldsymbol{G}_{(3)} = \mathrm{qr}\left[\boldsymbol{X}_3, 0\right];$
  $\mathcal{G} = \texttt{reconstruct}\left[\boldsymbol{G}_{(3)},\ 3\right];$
end
$\mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3;$

and the runtime scales similar in rank r. Further, for small rank r version 2 is faster than both other versions. But when the rank increases the expensive solve for the core tensor $\mathcal{G}$ starts to dominate the runtime. The total runtime will increase dramatically.

8.4.2 Projection operator for constant wave number

Also in three dimensions we can write the linear systems (8.40) for $\boldsymbol{U}_1$, (8.41) for $\boldsymbol{U}_2$ and (8.42) for $\boldsymbol{U}_3$ as projection operators applied to the residual of the tensor equation, (8.38).

Consider a tensor $\mathcal{M}$ in Tucker format and factorized as $\mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$, with unknowns $\mathcal{G}, \boldsymbol{U}_1, \boldsymbol{U}_2$ and $\boldsymbol{U}_3$. Discretization of (8.34) leads to a linear operator $\mathcal{L}$ applied on tensors. Its matrix representation $\boldsymbol{L}$ has a sum of Kronecker products structure, as given in (8.37).

Solving for an unknown factors $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ or $\boldsymbol{U}_3$ (and the core-tensor $\mathcal{G}$) using (8.40), (8.41) or (8.42) can be interpreted as a projection operator applied on the residual. For example, (8.40) can be interpreted as

$$\overline{\left(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I}\right) \boldsymbol{L} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I}\right)} \mathrm{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right] = \left(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I}\right) \mathrm{vec}\left[\boldsymbol{F}_{(1)}\right].$$

The residual, in tensor format, is given by

$$\begin{aligned}\mathcal{R} &= \mathcal{F} - \mathcal{L}\mathcal{M}, \\ &= \mathcal{F} - \mathcal{G} \times_1 \left(-\boldsymbol{D}_{xx} - k^2 \boldsymbol{I}\right) \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 + \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{D}_{yy} \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 + \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{D}_{zz} \boldsymbol{U}_3.\end{aligned}$$

Writing this tensor equation in the first unfolding leads to the following matrix equation

$$\boldsymbol{R}_{(1)} = \boldsymbol{F}_{(1)} - \overline{\left(-\boldsymbol{D}_{xx} - k^2 \boldsymbol{I}\right) \boldsymbol{U}_1} \boldsymbol{G}_{(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)^{\mathsf{H}} + \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}} + \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)} \left(\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)^{\mathsf{H}},$$

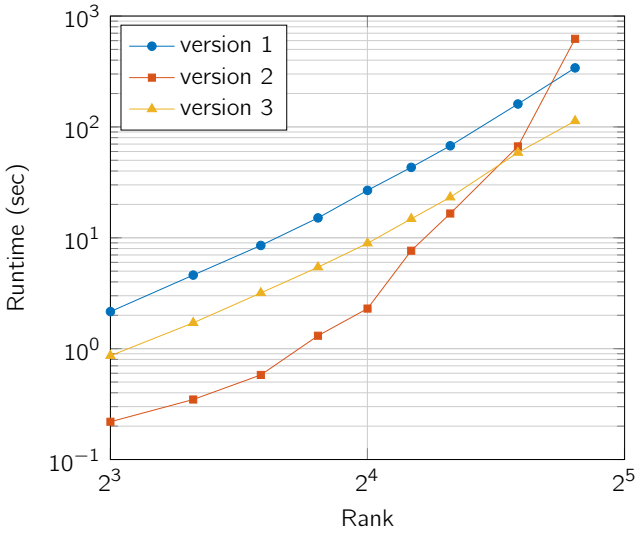


Figure 8.13: Plot of runtime for 4 iteration with constant wave number in three dimensions using the three different algorithms ($M = 100$).

which can be vectorized as

$$\begin{aligned}
\mathrm{vec}\left[\boldsymbol{R}_{(1)}\right] &= \mathrm{vec}\left[\boldsymbol{F}_{(1)}\right] - \left(\overline{(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes (-\boldsymbol{D}_{xx} - k^2\boldsymbol{I}))} - \overline{(\boldsymbol{U}_3 \otimes \boldsymbol{D}_{yy}\boldsymbol{U}_2 \otimes \boldsymbol{I})} - \overline{(\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right) \mathrm{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right] \\
&= \mathrm{vec}\left[\boldsymbol{F}_{(1)}\right] - \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\mathrm{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right] \\
&= \mathrm{vec}\left[\boldsymbol{F}_{(1)}\right] - \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\left[\overline{(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I})\,\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right]^{-1}(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I})\,\mathrm{vec}\left[\boldsymbol{F}_{(1)}\right] \\
&= P_{23}\mathrm{vec}\left[\boldsymbol{F}_{(1)}\right],
\end{aligned}$$

where operator P_{23} is given by

$$\begin{aligned}
P_{23} &= \boldsymbol{I} - \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\left[\overline{(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I})\,\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right]^{-1}(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I}) \\
&= \boldsymbol{I} - \boldsymbol{X}.
\end{aligned} \tag{8.47}$$

This operator P_{23} is indeed a projection operator. Observe that the terms between the two inverses cancel against one of the inverse factors:

$$\begin{aligned}
\boldsymbol{X}^2 &= \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\left[\overline{(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I})\,\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right]^{-1}(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I})\,\overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\left[\overline{(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I})\,\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right]^{-1}(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I}) \\
&= \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\left[\overline{(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I})\,\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right]^{-1}(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I}) \\
&= \boldsymbol{X}.
\end{aligned}$$

This operator is a natural extension to higher dimensions of the two dimensional operators as derived in Section 8.3.4.

A similar derivation results in projection operators P_{13} and P_{12} for the updates in $\boldsymbol{U}_2$ and

$\boldsymbol{U}_3$, respectively.

$$\begin{aligned}
P_{23} &= \boldsymbol{I} - \overline{\boldsymbol{L}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I}\right)} \left[\overline{\left(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I}\right) \boldsymbol{L}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I}\right)}\right]^{-1} \left(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I}\right), \\
P_{13} &= \boldsymbol{I} - \overline{\boldsymbol{L}\left(\boldsymbol{U}_3 \otimes \boldsymbol{I} \otimes \boldsymbol{U}_1\right)} \left[\overline{\left(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{I} \otimes \boldsymbol{U}_1^{\mathsf{H}}\right) \boldsymbol{L}\left(\boldsymbol{U}_3 \otimes \boldsymbol{I} \otimes \boldsymbol{U}_1\right)}\right]^{-1} \left(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{I} \otimes \boldsymbol{U}_1^{\mathsf{T}}\right), \\
P_{12} &= \boldsymbol{I} - \overline{\boldsymbol{L}\left(\boldsymbol{I} \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1\right)} \left[\overline{\left(\boldsymbol{I} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{U}_1^{\mathsf{H}}\right) \boldsymbol{L}\left(\boldsymbol{I} \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1\right)}\right]^{-1} \left(\boldsymbol{I} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{U}_1^{\mathsf{T}}\right).
\end{aligned} \tag{8.48}$$

The successive application of these projection operators on the residual results in an updated residual that lies in the intersection of all subspaces.

8.4.3 Helmholtz equation with space-dependent wave number

The presented algorithms with constant wave number can be extended to space-dependent wave numbers. So, let us consider a three-dimensional Helmholtz problem where $\mathcal{K} = k^2(\rho_1, \rho_2, \rho_3)$ represents the space-dependent wave number on the discretized mesh.

Further, we assume that a Canonical Polyadic decomposition of the space-dependent wave number tensor $\mathcal{K}$ is known, i.e.

$$\mathcal{K} = \sum_{i=1}^{s} \sigma_i \left(\boldsymbol{v}_i^{(1)} \circ \boldsymbol{v}_i^{(2)} \circ \cdots \circ \boldsymbol{v}_i^{(d)}\right), \tag{8.49}$$

where $s \in \mathbb{N}_+$ is the CP-rank of $\mathcal{K}$ and $\boldsymbol{v}_i^{(j)} \in \mathbb{C}^{n_j}$ for $i = 1, 2, \ldots, s; j = 1, 2, \ldots, d$ are vectors. Further, σ_i is a tensor generalization of a singular value and $\circ$ denotes the vector outer product.

The application of the space-dependent Helmholtz operator $\mathcal{L}$ on tensor $\mathcal{M}$ is given by

$$\begin{aligned}
\mathcal{L}\mathcal{M} &= \mathcal{F} \\
\mathcal{L}\mathcal{M} &= -\mathcal{G} \times_1 \boldsymbol{D}_{xx}\boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \\
&\quad - \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{D}_{yy}\boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \\
&\quad - \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{D}_{zz}\boldsymbol{U}_3 \\
&\quad - \mathcal{K} \circ (\mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3) \\
&\quad = \mathcal{F},
\end{aligned} \tag{8.50}$$

where $\boldsymbol{U}_i^{\mathsf{H}}\boldsymbol{U}_i = \boldsymbol{I}$ for $i = 1, 2, 3$ and $\mathcal{F}$ is a tensor representation of the right hand side function f discretized on the used grid. Here $\circ$ denotes the Hadamard product for tensors.

In a similar way as in the three-dimensional constant wave number case, we can derive equations to iteratively solve for the factors $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$. We start from (8.50) and multiply with $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ in the second and third direction, respectively. Using that the columns of $\boldsymbol{U}_i$ are orthonormal, the following expression is derived:

$$\begin{aligned}
\mathcal{L}\mathcal{M} \times_2 \boldsymbol{U}_2^{\mathsf{H}} \times_3 \boldsymbol{U}_3^{\mathsf{H}} &= -\mathcal{G} \times_1 \boldsymbol{D}_{xx} \\
&\quad - \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2 \\
&\quad - \mathcal{G} \times_1 \boldsymbol{U}_1 \times_3 \boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3 \\
&\quad - [\mathcal{K} \circ (\mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3)] \times_2 \boldsymbol{U}_2^{\mathsf{H}} \times_3 \boldsymbol{U}_3^{\mathsf{H}}.
\end{aligned}$$

Written in the first unfolding, the multiplication with $\boldsymbol{U}_2^{\mathsf{H}}$ and $\boldsymbol{U}_3^{\mathsf{H}}$ in, respectively, the second and third direction is equivalent to post-multiplication with the matrix

$$\left(\boldsymbol{I}\otimes\boldsymbol{U}_2^{\mathsf{H}}\right)^{\mathsf{H}}\left(\boldsymbol{U}_3^{\mathsf{H}}\otimes\boldsymbol{I}\right)^{\mathsf{H}}=\left(\boldsymbol{U}_3^{\mathsf{H}}\otimes\boldsymbol{U}_2^{\mathsf{H}}\right)^{\mathsf{H}}=\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right).$$

Most of the terms are equal to the case where we had a constant wave number, see also (8.38). Let us focus on the last term that contains the Hadamard product with the space-dependent wave number, i.e.:

$$\mathcal{K}\circ\left(\mathcal{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{U}_2\times_3\boldsymbol{U}_3\right). \tag{8.51}$$

For Hadamard products of tensors, $\mathcal{Z}=\mathcal{X}\circ\mathcal{Y}$, the following property for the k-th unfolding holds $\boldsymbol{Z}_{(k)}=\boldsymbol{X}_{(k)}\circ\boldsymbol{Y}_{(k)}$. Thus, written in the first unfolding (8.51) is given by

$$\begin{aligned}&\boldsymbol{K}_{(1)}\circ\boldsymbol{M}_{(1)}\\&\boldsymbol{K}_{(1)}\circ\left(\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)^{\mathsf{H}}\right).\end{aligned} \tag{8.52}$$

As the Hadamard product-term (8.52) is written in the first unfolding and multiplication with $\boldsymbol{U}_2^{\mathsf{H}}$ and $\boldsymbol{U}_3^{\mathsf{H}}$ in respectively the second and third dimension results in

$$\left[\underbrace{\boldsymbol{K}_{(1)}}_{\boldsymbol{K}}\circ\Big(\underbrace{\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}}_{\boldsymbol{U}}\underbrace{\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)^{\mathsf{H}}}_{\boldsymbol{V}^{\mathsf{H}}}\Big)\right]\underbrace{\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)}_{\boldsymbol{V}}. \tag{8.53}$$

The derivation of the other terms of (8.50) are equal to the constant wave number case.

The equation written in the first unfolding leads to the following matrix equation:

$$\begin{aligned}-\overline{\boldsymbol{D}_{xx}\boldsymbol{U}_1}\boldsymbol{G}_{(1)}-\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\left(I\otimes\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}}-\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes I\right)^{\mathsf{H}}\\-\left[\boldsymbol{K}_{(1)}\circ\left(\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)^{\mathsf{H}}\right)\right]\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)=\boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right).\end{aligned} \tag{8.54}$$

Vectorization of the last term, i.e. (8.53), results again in an expression for the space-dependent wave number of the form $\left(\boldsymbol{K}\circ\left(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right)\right)\boldsymbol{V}$, similar to the two-dimensional case which was given in (8.21). Using again (8.24), the vectorization of this expression is given by

$$\left(\boldsymbol{U}_3^{\mathsf{T}}\otimes\boldsymbol{U}_2^{\mathsf{T}}\otimes\boldsymbol{I}\right)\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}_{(1)}\right]\right)\overline{\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{I}\right)}\mathrm{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right]. \tag{8.55}$$

Because $\mathcal{K}$ is known in a Canonical Polyadic tensor (CP tensor) decomposition[2], as given in (8.49), we have

$$\mathrm{diag}\left(\mathrm{vec}\left[\boldsymbol{K}_{(1)}\right]\right)=\sum_{i=1}^{s}\sigma_i\mathrm{diag}\left(\boldsymbol{v}_i^{(3)}\right)\otimes\mathrm{diag}\left(\boldsymbol{v}_i^{(2)}\right)\otimes\mathrm{diag}\left(\boldsymbol{v}_i^{(1)}\right).$$

So, using the CP tensor representation of the space-dependent wave number the vectorization in (8.55) simplifies even further:

$$\sum_{i=1}^{s}\sigma_i\left(\boldsymbol{U}_3^{\mathsf{T}}\otimes\boldsymbol{U}_2^{\mathsf{T}}\otimes\boldsymbol{I}\right)\left[\mathrm{diag}\left(\boldsymbol{v}_i^{(3)}\right)\otimes\mathrm{diag}\left(\boldsymbol{v}_i^{(2)}\right)\otimes\mathrm{diag}\left(\boldsymbol{v}_i^{(1)}\right)\right]\overline{\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{I}\right)}\mathrm{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right],$$

[2]Otherwise a Canonical Polyadic tensor decomposition can be computed using for example an CP-ALS algorithm [48].

which reduces to

$$\underbrace{\sum_{i=1}^{s} \sigma_i \left(\boldsymbol{U}_3^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(3)}\right) \overline{\boldsymbol{U}_3}\right) \otimes \left(\boldsymbol{U}_2^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(2)}\right) \overline{\boldsymbol{U}_2}\right) \otimes \left(\mathrm{diag}\left(\boldsymbol{v}_{i_1}^{(1)}\right)\right)}_{K_1} \mathtt{vec}\left[\overline{\boldsymbol{U}_1} \boldsymbol{G}_{(1)}\right].$$

In this way the K_1 operator is defined and can be applied to $\mathtt{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right]$. Observe that this expansion is only advantageous if the space-dependent wave number has low rank, which is typical the case for our applications.

In a similar way, the K_2 and K_3 operators can be derived:

$$\begin{aligned}
K_1 &= \sum_{i=1}^{s} \sigma_i \left(\boldsymbol{U}_3^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(3)}\right) \overline{\boldsymbol{U}_3}\right) \otimes \left(\boldsymbol{U}_2^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(2)}\right) \overline{\boldsymbol{U}_2}\right) \otimes \left(\mathrm{diag}\left(\boldsymbol{v}_i^{(1)}\right)\right), \\
K_2 &= \sum_{i=1}^{s} \sigma_i \left(\boldsymbol{U}_3^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(3)}\right) \overline{\boldsymbol{U}_3}\right) \otimes \left(\boldsymbol{U}_1^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(1)}\right) \overline{\boldsymbol{U}_1}\right) \otimes \left(\mathrm{diag}\left(\boldsymbol{v}_i^{(2)}\right)\right), \\
K_3 &= \sum_{i=1}^{s} \sigma_i \left(\boldsymbol{U}_2^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(2)}\right) \overline{\boldsymbol{U}_2}\right) \otimes \left(\boldsymbol{U}_1^{\mathsf{T}} \mathrm{diag}\left(\boldsymbol{v}_i^{(1)}\right) \overline{\boldsymbol{U}_1}\right) \otimes \left(\mathrm{diag}\left(\boldsymbol{v}_i^{(3)}\right)\right).
\end{aligned} \tag{8.56}$$

So, we find the following linear system to solve for $\mathtt{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right]$:

$$\begin{aligned}
\left\{-\boldsymbol{I} \otimes \overline{\boldsymbol{D}_{xx}} - \left[\overline{\left(\boldsymbol{I} \otimes \boldsymbol{U}_2^{\mathsf{H}} \boldsymbol{D}_{yy} \boldsymbol{U}_2\right)} - \overline{\left(\boldsymbol{U}_3^{\mathsf{H}} \boldsymbol{D}_{zz} \boldsymbol{U}_3 \otimes \boldsymbol{I}\right)}\right] \otimes \boldsymbol{I} - K_1\right\} \mathtt{vec}\left[\underbrace{\overline{\boldsymbol{U}_1} \boldsymbol{G}_{(1)}}_{\boldsymbol{X}_1}\right] \\
= \mathtt{vec}\left[\boldsymbol{F}_{(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)\right].
\end{aligned} \tag{8.57}$$

Observe this is a square system with $n_1 \times r_2 r_3$ unknowns (where the solution in matrix form $\boldsymbol{X}_1$ is typical for rank $r > r_1$). In a similar way, update equations for $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are derived by multiplying (8.50) with the other factor matrices in the appropriate directions:

$$\begin{aligned}
\left\{-\boldsymbol{I} \otimes \overline{\boldsymbol{D}_{yy}} + \left[-\overline{\left(I \otimes \boldsymbol{U}_1^{\mathsf{H}} \boldsymbol{D}_{xx} \boldsymbol{U}_1\right)} - \overline{\left(\boldsymbol{U}_3^{\mathsf{H}} \boldsymbol{D}_{zz} \boldsymbol{U}_3 \otimes \boldsymbol{I}\right)}\right] \otimes \boldsymbol{I} - K_2\right\} \mathtt{vec}\left[\underbrace{\overline{\boldsymbol{U}_2} \boldsymbol{G}_{(2)}}_{\boldsymbol{X}_2}\right] \\
= \mathtt{vec}\left[\boldsymbol{F}_{(2)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_1\right)\right],
\end{aligned} \tag{8.58}$$

and

$$\begin{aligned}
\left\{-\boldsymbol{I} \otimes \overline{\boldsymbol{D}_{zz}} + \left[-\overline{\left(\boldsymbol{I} \otimes \boldsymbol{U}_1^{\mathsf{H}} \boldsymbol{D}_{xx} \boldsymbol{U}_1\right)} - \overline{\left(\boldsymbol{U}_2^{\mathsf{H}} \boldsymbol{D}_{yy} \boldsymbol{U}_2 \otimes I\right)}\right] \otimes I - K_3\right\} \mathtt{vec}\left[\underbrace{\overline{\boldsymbol{U}_3} \boldsymbol{G}_{(3)}}_{\boldsymbol{X}_3}\right] \\
= \mathtt{vec}\left[\boldsymbol{F}_{(3)} \left(\boldsymbol{U}_2 \otimes \boldsymbol{U}_1\right)\right].
\end{aligned} \tag{8.59}$$

Alternating between solving for $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ using (8.57), (8.58) or (8.59) results in an algorithm to approximate low-rank tensor solutions for three-dimensional Helmholtz problems as given in (8.34). Also in this case the orthogonality of the columns of $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are maintained by additional QR factorizations. So, we derive the algorithm as formulated in Algorithm 8. The generalization for dimensions $d > 3$ is straightforward.

Algorithm 8: Solve for the low-rank tensor decomposition of the solution $\mathcal{M}$ of a three-dimensional Helmholtz problem with space-dependent wave number (version 1).

```
[G, U1, U2, U3] = hosvd(initial guess);
[Σ, V1, V2, V3] = cp_als(K);
while not converged do
    for i = 1, 2, 3 do
        Compute K_i using (8.56);
        Solve for X_i = conj(U_i) G_(i) ∈ C^{n_i × r^{d-1}} using (8.57), (8.58) or (8.59);
        conj(U_i) G_(i) = qr[X_i(:, 1 : r_i), 0];
    end
end
G = reconstruct[G_(i), i];
M = G ×_1 U_1 ×_2 U_2 ×_3 U_3;
```

Similar to the discussion for the constant wave number algorithms, observe that we solve again for a large matrix $\boldsymbol{X}_i \in \mathbb{C}^{n_i \times r_1 r_2 r_3 / r_i}$. So, in general the rank of this matrix could be $\min(n_i,\ r_1 r_2 r_3 / r_i)$. But it is also known that $\boldsymbol{X}_i = \overline{\boldsymbol{U}_i}\boldsymbol{G}_{(i)}$ leads to the fact that the rank of $\boldsymbol{X}_i$ should be at most r_i. So selecting the first r_i columns of $\boldsymbol{X}_i$ and computing its QR decomposition is sufficient to derive a new orthonormal basis as factor matrix $\overline{\boldsymbol{U}}_i$.

Algorithm 8 is exactly the space-dependent wave number equivalent of Algorithm 5. The same ideas can be applied to derive space-dependent wave number alternatives of the algorithms corresponding to version 2 and version 3. Again, to circumvent solving large systems, we can pre-compute the QR factorization of $\boldsymbol{G}_{(i)}$ and project these equations onto the obtained $\boldsymbol{Q}_i$. Indeed, this will reduce the number of unknowns in these linear systems to exactly the number of unknowns as needed for the factor matrices $\overline{\boldsymbol{U}}_1$ and $\overline{\boldsymbol{U}}_2$.

To discuss the details we start again from equation (8.54) and use the QR factorization of $\boldsymbol{G}_{(1)}^{\mathsf{H}}$, given by

$$\boldsymbol{Q}_1 \boldsymbol{R}_1^{\mathsf{H}} = \mathrm{qr}\left[\boldsymbol{G}_{(1)}^{\mathsf{H}}\right].$$

This yields

$$\begin{aligned} -\overline{\boldsymbol{D}_{xx}\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}} - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(I \otimes \boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}} - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes I\right)^{\mathsf{H}} \\ - \left[\boldsymbol{K}_{(1)} \circ \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)^{\mathsf{H}}\right]\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right) = \boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right). \end{aligned}$$

Post multiplication of the left hand side of this equation by $\boldsymbol{Q}_1$ yields

$$\begin{aligned} -\overline{\boldsymbol{D}_{xx}\boldsymbol{U}_1}\boldsymbol{R}_1 - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(I \otimes \boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)^{\mathsf{H}}\boldsymbol{Q}_1 - \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes I\right)^{\mathsf{H}}\boldsymbol{Q}_1 \\ - \left[\boldsymbol{K}_{(1)} \circ \overline{\boldsymbol{U}_1}\boldsymbol{R}_1\boldsymbol{Q}_1^{\mathsf{H}}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)^{\mathsf{H}}\right]\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)\boldsymbol{Q}_1 = \boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)\boldsymbol{Q}_1. \end{aligned}$$

To solve this equation for $\boldsymbol{U}_1$, it is written in vectorized form as

$$\left\{-\boldsymbol{I}\otimes\overline{\boldsymbol{D}_{xx}}+\boldsymbol{Q}_1^{\mathsf{T}}\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\right)}-\overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes\boldsymbol{I}\right)}\right]\overline{\boldsymbol{Q}_1}\otimes\boldsymbol{I}-\boldsymbol{Q}_1^{\mathsf{T}}K_1\overline{\boldsymbol{Q}_1}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_1}\boldsymbol{R}_1}_{\boldsymbol{X}_1}\right] = \operatorname{vec}\left[\boldsymbol{F}_{(1)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\right)\boldsymbol{Q}_1\right]. \tag{8.60}$$

In a similar way, the update equations for $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are derived by multiplying (8.50) with the other factor matrices in the appropriate dimensions and using the QR factorizations of $\boldsymbol{G}_{(i)}^{\mathsf{H}}$:

$$\left\{-\boldsymbol{I}\otimes\overline{\boldsymbol{D}_{yy}}+\boldsymbol{Q}_2^{\mathsf{T}}\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}_1\right)}-\overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\boldsymbol{D}_{zz}\boldsymbol{U}_3\otimes\boldsymbol{I}\right)}\right]\overline{\boldsymbol{Q}_2}\otimes\boldsymbol{I}-\boldsymbol{Q}_2^{\mathsf{T}}K_2\overline{\boldsymbol{Q}_2}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_2}\boldsymbol{R}_2}_{\boldsymbol{X}_2}\right] = \operatorname{vec}\left[\boldsymbol{F}_{(2)}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_1\right)\boldsymbol{Q}_2\right], \tag{8.61}$$

and

$$\left\{-\boldsymbol{I}\otimes\overline{\boldsymbol{D}_{zz}}+\boldsymbol{Q}_3^{\mathsf{T}}\left[-\overline{\left(\boldsymbol{I}\otimes\boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}_1\right)}-\overline{\left(\boldsymbol{U}_2^{\mathsf{H}}\boldsymbol{D}_{yy}\boldsymbol{U}_2\otimes\boldsymbol{I}\right)}\right]\overline{\boldsymbol{Q}_3}\otimes\boldsymbol{I}\right\}\operatorname{vec}\left[\underbrace{\overline{\boldsymbol{U}_3}\boldsymbol{R}_3}_{\boldsymbol{X}_3}\right]-\boldsymbol{Q}_3^{\mathsf{T}}K_3\overline{\boldsymbol{Q}_3} = \operatorname{vec}\left[\boldsymbol{F}_{(3)}\left(\boldsymbol{U}_2\otimes\boldsymbol{U}_1\right)\boldsymbol{Q}_3\right]. \tag{8.62}$$

All these equations are cheap to solve. Indeed, $\operatorname{vec}\left[\overline{\boldsymbol{U}_i}\boldsymbol{R}_i\right]$ has length $n_i r_i$. Computing a symmetric reverse Cuthill-McKee permutation of the system matrix one observes a matrix with a bandwidth $\mathcal{O}(r)$, so solving these equations has a computational cost $\mathcal{O}\left(nr^2\right)$.

Alternating between solving for $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ using (8.60), (8.61) or (8.59) results again in an algorithm to approximate low-rank solutions for three-dimensional space-dependent Helmholtz problems. Also in this case the orthogonality of the columns of $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are maintained by additional QR factorizations. So, we derive the algorithm as formulated in Algorithm 9. Algorithm 9 is exactly the space-dependent wave number equivalent of Algorithm 7.

8.4.4 Projection operator for space-dependent wave number

Consider a tensor $\mathcal{M}$ in Tucker tensor format and factorized as $\mathcal{M}=\mathcal{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{U}_2\times_3\boldsymbol{U}_3$, with unknowns $\mathcal{G},\boldsymbol{U}_1,\boldsymbol{U}_2$ and $\boldsymbol{U}_3$. Discretization of (8.34) with a space-dependent wave number leads to a linear operator $\mathcal{L}$ applied on tensors. Its matrix representation $\boldsymbol{L}$ has again a structure as given in (8.37).

Solving for the unknown factors $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ or $\boldsymbol{U}_3$ (and the core-tensor $\mathcal{G}$) using (8.57), (8.58) or (8.59) can, again, be interpreted as a projection operator applied on the residual. For example, (8.57) can be interpreted as

$$\overline{\left(\boldsymbol{U}_3^{\mathsf{H}}\otimes\boldsymbol{U}_2^{\mathsf{H}}\otimes\boldsymbol{I}\right)\boldsymbol{L}\left(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{I}\right)}\operatorname{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right]=\left(\boldsymbol{U}_3^{\mathsf{T}}\otimes\boldsymbol{U}_2^{\mathsf{T}}\otimes\boldsymbol{I}\right)\operatorname{vec}\left[\boldsymbol{F}_{(1)}\right], \tag{8.63}$$

Algorithm 9: Solve for the low-rank tensor decomposition of the solution $\mathcal{M}$ of a three-dimensional Helmholtz problem with space-dependent wave number (version 3).

$[\mathcal{G}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3] = \text{hosvd(initial guess)};$
$[\boldsymbol{\Sigma}, \boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3] = \text{cp\_als}(\mathcal{K});$
while *not converged* **do**
  for $i = 1, 2$ **do**
    Compute $\mathcal{K}_i$ using (8.56);
    $\boldsymbol{Q}_i \widetilde{\boldsymbol{R}} = \texttt{qr}\left[\boldsymbol{G}_{(i)}^{\mathsf{H}}, 0\right];$
    Solve for $\boldsymbol{X}_i = \overline{\boldsymbol{U}_i}\boldsymbol{R}_i \in \mathbb{C}^{n_i \times r_i}$ using (8.60) or (8.61);
    $\overline{\boldsymbol{U}_i}\boldsymbol{R}_i = \texttt{qr}\left[\boldsymbol{X}_i, 0\right];$
    $\mathcal{G} = \texttt{reconstruct}\left[\boldsymbol{R}_i \boldsymbol{Q}_i^{\mathsf{H}},\ i\right];$
  end
  Compute $\mathcal{K}_3$ using (8.56);
  Solve for $\boldsymbol{X}_3 = \overline{\boldsymbol{U}_3}\boldsymbol{G}_{(3)} \in \mathbb{C}^{n_3 \times r^{d-1}}$ using (8.59);
  $\overline{\boldsymbol{U}_3}\boldsymbol{G}_{(3)} = \texttt{qr}\left[\boldsymbol{X}_3, 0\right];$
  $\mathcal{G} = \texttt{reconstruct}\left[\boldsymbol{G}_{(3)},\ 3\right];$
end
$\mathcal{M} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3;$

The residual in tensor format is given by

$$\begin{aligned}\mathcal{R} &= \mathcal{F} - \mathcal{L}\mathcal{M}, \\ &= \mathcal{F} + \mathcal{G} \times_1 \boldsymbol{D}_{xx}\boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \\ &\quad + \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{D}_{yy}\boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \\ &\quad + \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{D}_{zz}\boldsymbol{U}_3 \\ &\quad + \mathcal{K} \circ (\mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3)\,.\end{aligned} \tag{8.64}$$

Writing this tensor equation in the first unfolding leads to the following matrix equation

$$\begin{aligned}\boldsymbol{R}_{(1)} = \boldsymbol{F}_{(1)} &+ \overline{\boldsymbol{D}_{xx}\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2)^{\mathsf{H}} \\ &+ \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\,(\boldsymbol{U}_3 \otimes \boldsymbol{D}_{yy}\boldsymbol{U}_2)^{\mathsf{H}} \\ &+ \overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\,(\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes \boldsymbol{U}_2)^{\mathsf{H}} \\ &+ \boldsymbol{K}_{(1)} \circ \left(\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2)^{\mathsf{H}}\right)\end{aligned} \tag{8.65}$$

which can be matricized as

$$\text{vec}\left[\boldsymbol{R}_{(1)}\right] = \text{vec}\left[\boldsymbol{F}_{(1)}\right] - \left(-\overline{(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{D}_{xx})} - \overline{(\boldsymbol{U}_3 \otimes \boldsymbol{D}_{yy}\boldsymbol{U}_2 \otimes \boldsymbol{I})} - \overline{(\boldsymbol{D}_{zz}\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})} - \text{diag}\left(\text{vec}\left[\boldsymbol{K}_{(1)}\right]\right)\overline{(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right)\text{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right].$$

Rewriting this results in exactly the same structure and projection operator as in the constant wave number case:

$$\begin{aligned}\text{vec}\left[\boldsymbol{R}_{(1)}\right] &= \ldots \\ &= \text{vec}\left[\boldsymbol{F}_{(1)}\right] - \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\text{vec}\left[\overline{\boldsymbol{U}_1}\boldsymbol{G}_{(1)}\right] \\ &= \text{vec}\left[\boldsymbol{F}_{(1)}\right] - \overline{\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\left[\overline{(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I})\,\boldsymbol{L}\,(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I})}\right]^{-1}(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I})\,\text{vec}\left[\boldsymbol{F}_{(1)}\right] \\ &= P_{23}\text{vec}\left[\boldsymbol{F}_{(1)}\right]\end{aligned}$$

where projection operator P_{23} is similar to the projector in the constant wave number case, see (8.47), and now given by

$$\begin{aligned} P_{23} &= \boldsymbol{I} - \overline{\boldsymbol{L}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I}\right)} \left[\overline{\left(\boldsymbol{U}_3^{\mathsf{H}} \otimes \boldsymbol{U}_2^{\mathsf{H}} \otimes \boldsymbol{I}\right) \boldsymbol{L}\left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{I}\right)}\right]^{-1} \left(\boldsymbol{U}_3^{\mathsf{T}} \otimes \boldsymbol{U}_2^{\mathsf{T}} \otimes \boldsymbol{I}\right) \\ &= \boldsymbol{I} - \boldsymbol{X}. \end{aligned} \tag{8.66}$$

A similar derivation results in projection operators P_{13} and P_{12} for the updates in $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$, respectively. Both are also the same as in the constant wave number case, as given in (8.48).

8.5 Numerical results

In this section, we demonstrate the promising results of the derived algorithms with some numerical experiments in two and three dimensions. Furthermore, we consider discretizations of the Helmholtz equation with constant and space-dependent wave numbers.

8.5.1 2D Helmholtz problem with space-dependent wave number

First, we consider a two-dimensional Helmholtz problem with a space-dependent wave number given by $k^2(\rho_1, \rho_2) = 2 + e^{-\rho_1^2 - \rho_2^2}$.

For this example the two-dimensional domain $\Omega = [-10,\ 10]^2$ is discretized with $M = 1000$ equidistant mesh points per direction in the interior of the domain. Further it is extended with exterior complex scaling to implement the absorbing boundary conditions. In total, the number of discretization points per directions equals $n = n_1 = n_2 = 1668$. As external force $f(\rho_1, \rho_2) = -e^{-\rho_1^2 - \rho_2^2}$ is applied.

In this space-dependent wave number example it is known that the matrix representation of the semi-exact solution of the Helmholtz equation on the full grid has a low rank. Indeed, approximating the semi-exact solution with a low-rank matrix with rank $r = 17$ is in this case sufficient to obtain an error below the threshold $\tau = 10^{-6}$.

Starting with an random (orthonormalized) initial guess for $\boldsymbol{V}^{(0)} \in \mathbb{C}^{n \times r}$ only a small number of iterations of Algorithm 4 is needed to obtain an error similar to the specified threshold τ. As shown in Figure 8.14 both the residual and the error with respect to the semi-exact solution decay in only a few iterations (i.e. in this example 4-8 iterations) to a level almost similar to the expected tolerance.

The singular values of the approximation $\boldsymbol{A}^{(k)} = \boldsymbol{U}^{(k)} \boldsymbol{R}^{(k)^{\mathsf{H}}} \boldsymbol{V}^{(k)^{\mathsf{H}}}$ in iteration i can be computed and are shown for increasing iterations in Figure 8.14. As expected the low- rank approximations recover the singular values of the full grid semi-exact solution. In fact $\boldsymbol{R}^{(k)}$ converges towards $\mathrm{diag}(\sigma_i)$.

The numerical rank of the matrix representation of the solution of a Helmholtz problem with a space-dependent wave number is unknown in advance. But, the presented algorithm

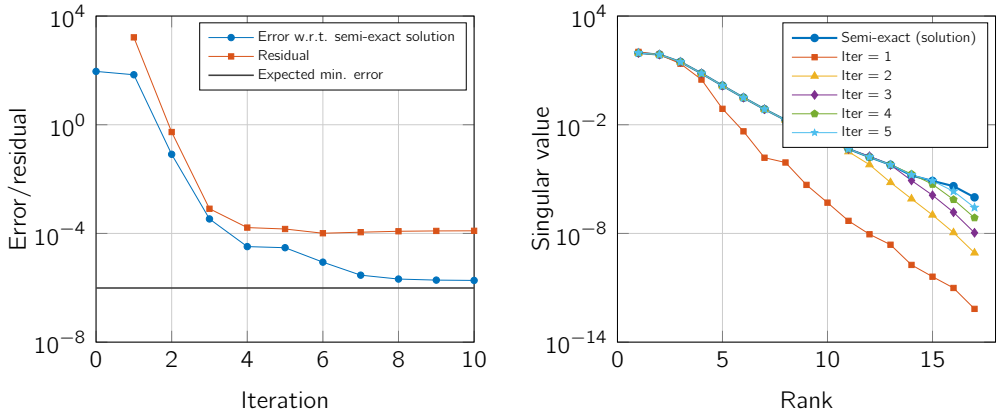


Figure 8.14: Plot of error and residual (left) and singular values (right) per iteration for space-dependent wave number in a two-dimensional Helmholtz problem ($M = 1000, r = 17$).

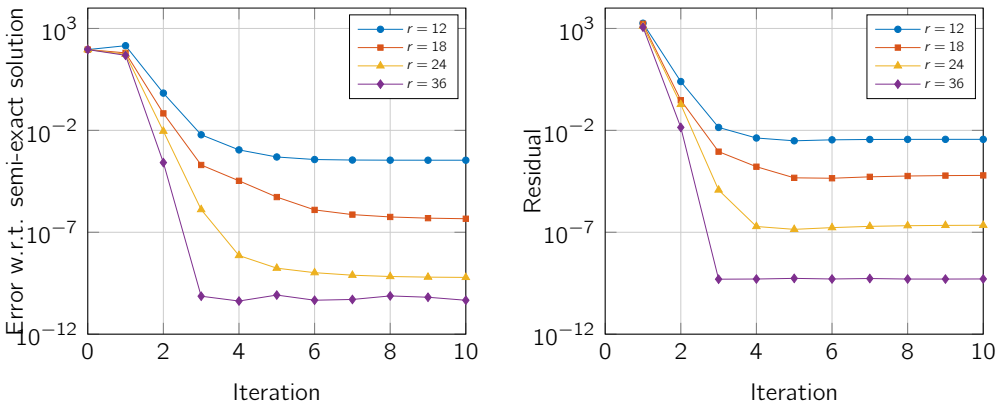


Figure 8.15: Plot of errors (left) and residuals (right) per iteration for space-dependent wave number in a two-dimensional Helmholtz problem with increasing ranks ($M = 1000$)

is stable with respect to over- and underestimation of the numerical rank of the solution. Figure 8.15 shows both the error and residual per iteration and illustrates this statement by approximating the same semi-exact solution with increasing ranks $r \in \{12, 18, 24, 36\}$.

In contrast to the constant coefficient wave number case the convergence with space-dependent wave number depends also on the maximal attainable rank. For increasing maximal attainable ranks the number of needed iterations decreases. This is especially observed when the error is considered, but it can also be seen in the figure where the residuals are shown, Figure 8.15.

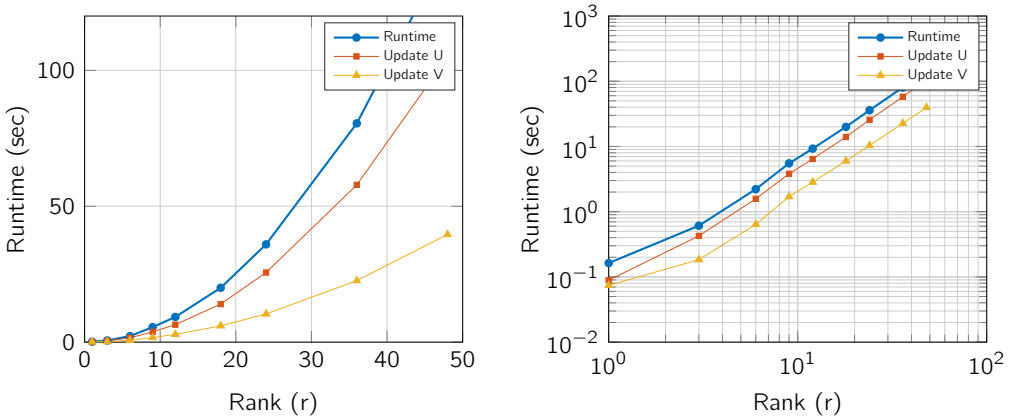


Figure 8.16: Plot of runtime for 10 iterations for space-dependent wave number in a two-dimensional Helmholtz problem with increasing ranks ($M = 1000$). Left: linlin-scale, right: loglog-scale.

8.5.2 3D Helmholtz problem with space-dependent wave number

In this example we solve a three-dimensional Helmholtz problem with a space-dependent wave number discretized on a DVR-grid [73]. All three versions of the three-dimensional algorithm for space-dependent wave numbers can successfully be applied.

First, to reduce computational cost of construction of the operators K_1, K_2 and K_3, see (8.56), a CP-decomposition of the space-dependent wave number is constructed. As shown in Figure 8.17 the space-dependent wave number can be well-approximated by a small number of rank-1 tensors. For the examples discussed in this section we used a CP-rank $s = 32$ to approximate this space-dependent wave number. Hence, the error in approximating the wave number is approximately $\mathcal{O}\left(10^{-4}\right)$.

For all three versions of the algorithm we use 10 iterations of the algorithm to converge to the low-rank solution. For example if we compute the low-rank solution (with $r = r_x = r_y = r_z = 16$) the residual after each iteration for all algorithms is shown in Figure 8.18.

If we increase the maximal attainable rank r of the low-rank approximation, indeed the residual decreases as shown in Figure 8.19a. The residual for version 1 and version 3 are good, while version 2 cannot compete with both other versions by reducing the residual as far as the other versions. Therefore version 1 or version 3, as given in Algorithm 8 or Algorithm 9 are preferred.

Considering the runtimes of the three versions, similar results as before are observed. In this experiment with `orderDvr` $= 7$ the number of gridpoints equals to $n = 41$. For version 1 and 3, again a runtime of $\mathcal{O}\left(nr^4\right)$ is observed. The runtime for version 2 splits into two parts: $\mathcal{O}\left(nr^2 + r^9\right)$. Due to the small rank r and the large number of iterations in these examples algorithm 2 is the fastest version. The runtimes for version 1 and version 3 differ indeed approximately a factor d, which makes version 3 better then version 1. The runtimes

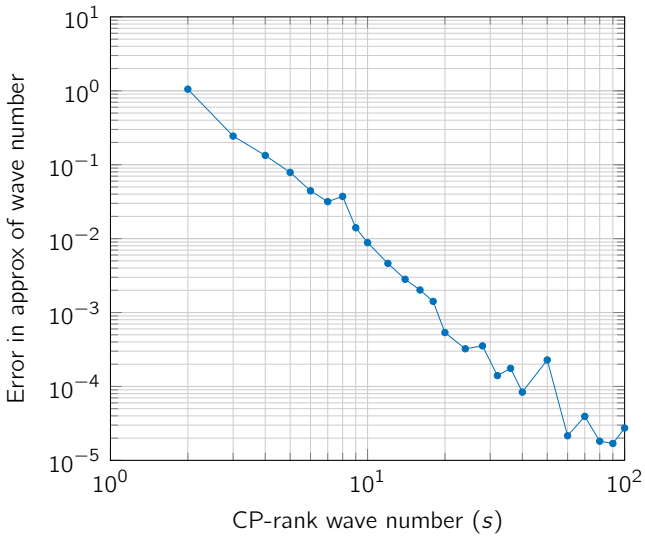


Figure 8.17: Low rank CP approximation to space-dependent wave number for three-dimensional Helmholtz problem.

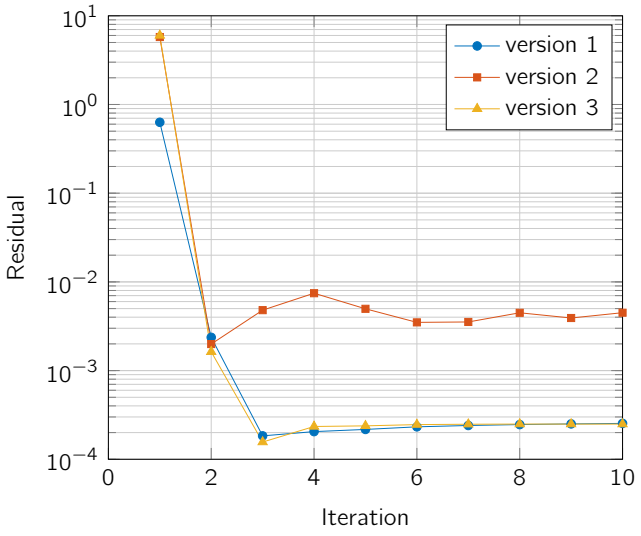
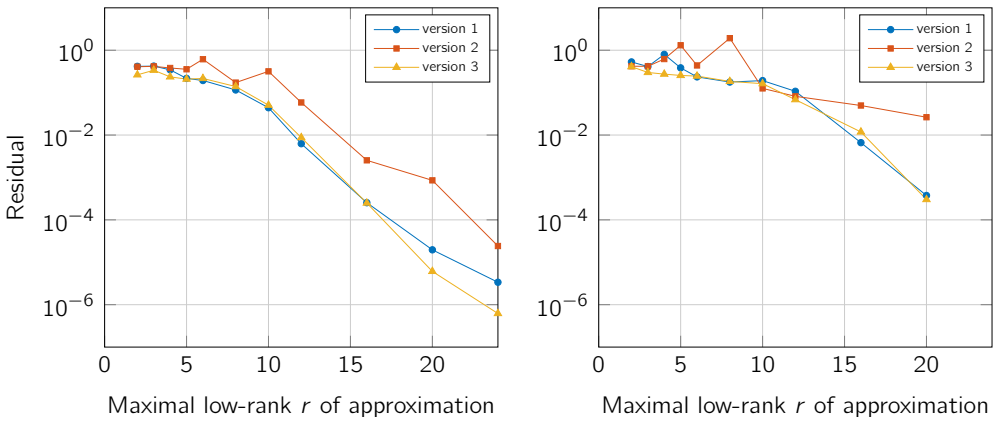


Figure 8.18: Residual per iteration for version 1, version 2 and version 3 of three-dimensional Helmholtz problem with space-dependent wave number (`orderDvr` $= 7$, $r = 16, s = 32$).



(a) `orderDvr` = 7 (b) `orderDvr` = 14

Figure 8.19: Residual after iteration 10 iterations for all three versions of algorithm with low-rank wave number $s = 32$.

with `orderDvr` = 7 (i.e. $n = 41$) are shown in Figure 8.20a and with `orderDvr` = 14 (i.e. $n = 90$) are shown in Figure 8.20b.

Comparing the runtimes for `orderDvr` = 7 and `orderDvr` = 14 we see for version 2 (when the rank gets larger) indeed approximately the same runtime independent of `orderDvr`. Also versions 1 and 3 consume approximately twice as much time which is as expected by the linear dependence on n for both algorithms.

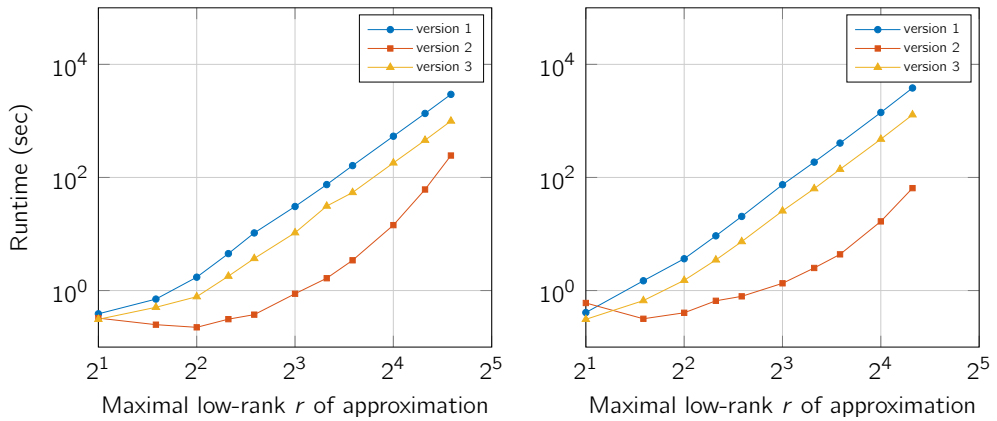
An impression of the low-rank approximation to the wave function is shown in Figures 8.21b and 8.22. In this impression the single, double and triple ionization are visible and can be represented by a low-rank wave function.

8.6 Discussion and conclusions

In this chapter we have analyzed the scattering solutions of a driven Schrödinger equation. These describe a break-up reaction where a quantum system is fragmented into multiple fragments. These problems are equivalent to solving a Helmholtz equation with space-dependent wave numbers.

We have shown, first in two dimensions and also in three dimensions, that the wave function of multiple ionization can be well approximated by a low-rank solution. In two dimensions, the waves can be represented as a product of two low-rank matrices. In three dimensions the waves can be represented with a low-rank Tucker tensor decomposition.

We propose a method that determines these low-rank components of the solution directly. We write the solution as a product of low-rank components and assume that a guess for all but one component is given. We then write a linear system for the remaining unknown



(a) `orderDvr` $= 7$ (b) `orderDvr` $= 14$

Figure 8.20: Runtime of 10 iterations for all three versions of algorithm for 3D Helmholtz with space-dependent wave number of low-rank $s = 32$.

component. This is then repeated until each of the components is updated.

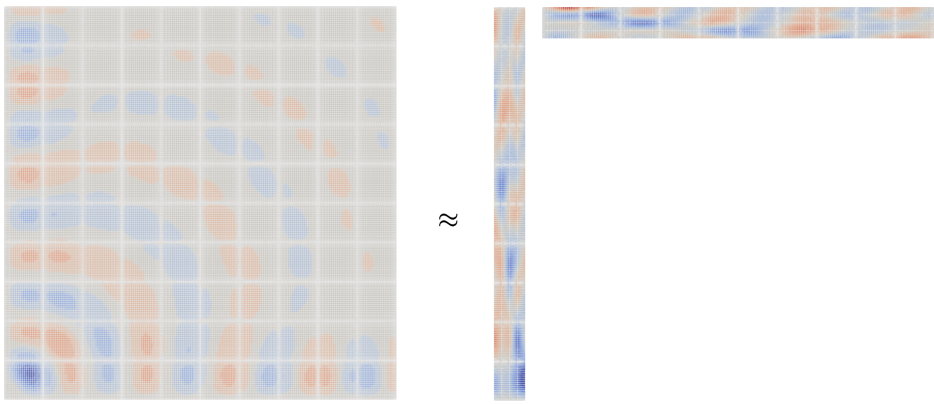
This procedure can be interpreted as a series of projections of the residual on a subspace and a correction within that subspace.

In theory, the generalization for dimensions $d > 3$ is straightforward. But for dimensions $d > 3$ it starts to be beneficial to change to a Tensor Train factorization [64]. It is expected that similar strategy can also be applied to tensors in Tensor Train format.

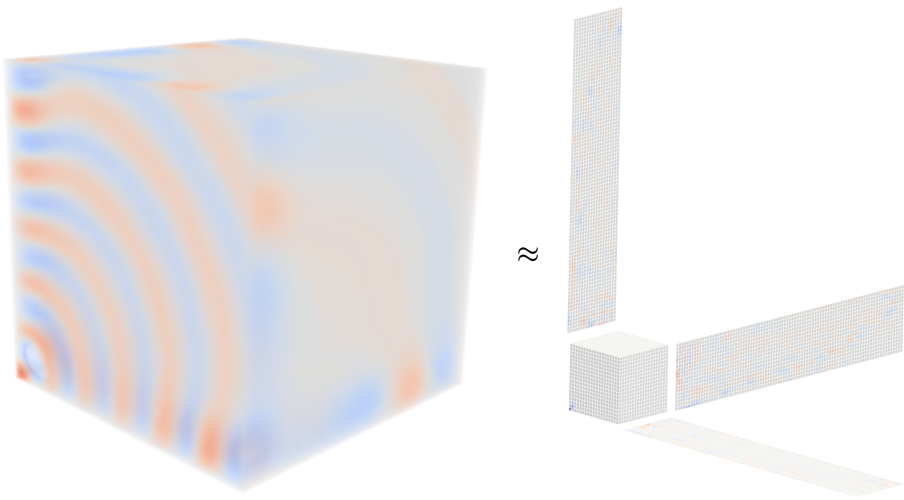
As demonstrated by the numerical experiments, the presented algorithms are able to exploit the low-rank structure of the solutions. This gives the advantage to reduce the number of unknowns and shorten the computational time to solve the Helmholtz equation.

In two dimensions, the low-rank representation of the solution can be represented by only $2nr$ unknowns instead of the full grid of n^2 unknowns. Also the linear systems to solve per iteration have only nr unknowns.

In high-dimensional Helmholtz equations, the low-rank Tucker tensor decomposition represents the solution with $\mathcal{O}\left(r^d + dnr\right)$ unknowns. So, the total number of unknowns is reduced, but it is still exponential in the dimension d. For increasing dimensions this leads, again, to systems with a number of unknowns exponential in d. Maybe other Tucker-like tensor decompositions with a number of unknowns only polynomial in d can resolve this problem and make the presented algorithm also applicable for higher dimensions.



(a) Impression of a low-rank matrix approximation in two dimenions.



(b) Impression of a low-rank tensor approximation in three dimensions.

Figure 8.21: Impressions of a low-rank approximation of a matrix and a Tucker tensor representing the wave function as solution to a two- and three-dimensional Helmholtz problem with a space-dependent wave number.

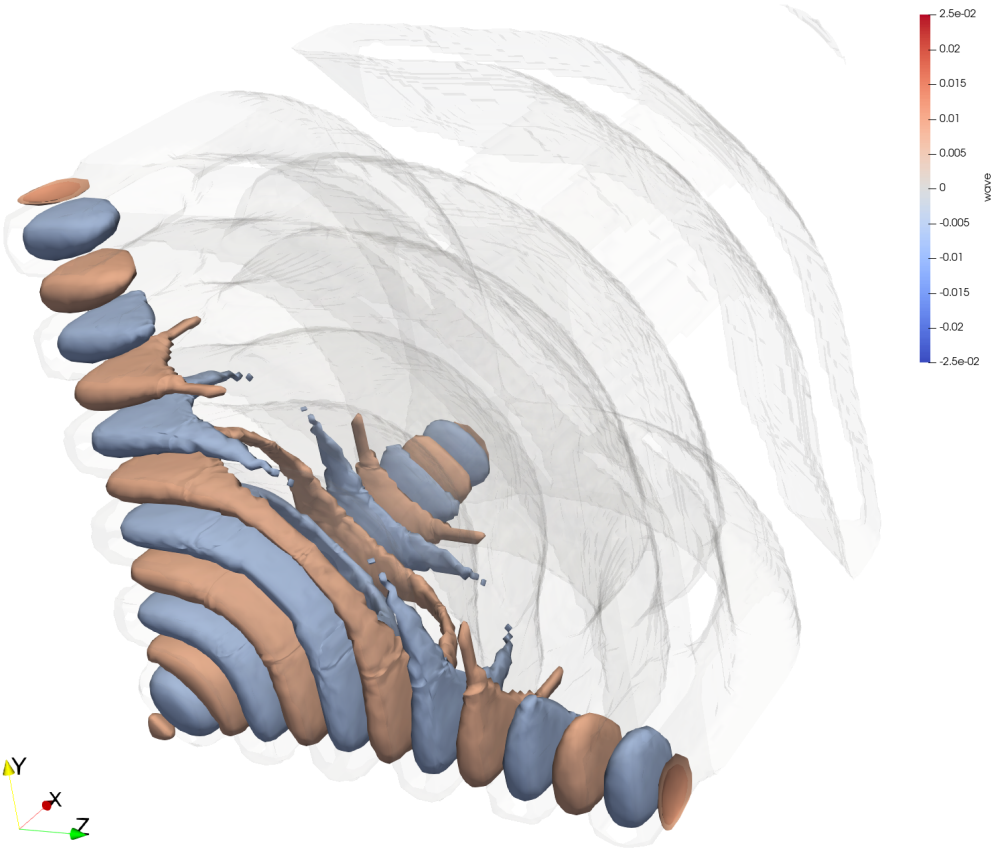


Figure 8.22: Visualization of a three-dimensional wave as low-rank approximation to the Helmholtz problem with space-dependent wave number with single, double and triple ionization.

Chapter 9

Low-rank approximation of solutions for linear time-dependent PDEs

Chapter summary:

In this chapter we study low-rank approximations of solutions for linear time-dependent partial differential equations. The dynamical low-rank integrator by Lubich et al. [47] is studied. Further it is observed that this can be seen as solving an optimization problem.

In this chapter we study different alternative optimization problems for the low-rank factors and formulate algorithms to solve these optimization problems. The alternative algorithms can both use explicit and implicit time integration such that also stiff differential equations could efficiently be solved.

Also the alternating algorithm that was used to solve for the low-rank factors of the Helmholtz problem in [80] is extended to solve for the low-rank factors of time-dependent PDEs.

A numerical study shows some preliminary results and give an overview about promising algorithms that could be used to solve for low-rank factors of time-dependent PDEs.

9.1 Introduction and motivation

In Chapter 8 we presented an alternating algorithm to efficiently solve for the low-rank components of the solution to a Helmholtz problem. In that chapter the problems were time-independent but in this chapter we explore possibilities to derive algorithms for low-rank approximations to solutions of time-dependent problems.

One might expect that low-rank components for solutions to, for example, diffusion problems could also be determined. Indeed, consider the Laplace transform (see eg. [82]) applied on

the time variable t of a function $u(x,t)$:

$$U(x,s) = \int_0^\infty u(x,t)e^{-st}\mathrm{d}t \tag{9.1}$$

where $x \in (0,\infty)$ and $t > 0$.

Let us consider a one-dimensional diffusion model problem with initial- and boundary condition as given by

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= \frac{\partial^2 u(x,t)}{\partial x^2}, \\ u(x=0,t) &= \alpha, \\ u(x,t=0) &= g(x), \end{aligned} \tag{9.2}$$

where $x \in (0,\infty)$ and $t > 0$. So, we model, for example, the temperature of an insulated infinite long one-dimensional pipe. At one end the temperature is fixed at α and an initial temperature profile is given by $g(x)$.

Applying the Laplace transform, some useful transforms are given by (see eg. [82])

$$\begin{aligned} u(x,t) = c \quad &\rightarrow \quad U(x,s) = \frac{c}{s} \quad \text{where } c \in \mathbb{R} \text{ is a constant}, \\ \frac{\partial u(x,t)}{\partial t} \quad &\rightarrow \quad sU(x,s) - u(x,0), \\ \frac{\partial^2 u(x,t)}{\partial x^2} \quad &\rightarrow \quad \frac{\partial^2 U(x,s)}{\partial x^2}, \end{aligned} \tag{9.3}$$

where $U(\cdot,s)$ is the Laplace transform of $u(\cdot,t)$.

So, writing the Laplace transform of $u(x,t)$ in (9.2) yields an ordinary differential equation (ODE) for $U(x,s)$ as given by

$$\begin{aligned} -\frac{\partial^2 U(x,s)}{\partial x^2} + sU(x,s) &= g(x), \\ U(x=0,s) &= \frac{\alpha}{s}. \end{aligned} \tag{9.4}$$

A discretized version of (9.1) changes the integral to an infinite sum. So, if $U(x,s)$ is low-rank and it is a linear combination of functions $u(x,t)e^{-st}$ for $t \in (0,\infty)$ it is likely that the function $u(x,t)$ is also low-rank over time t. That leads to the subject of this chapter where we search for stable evolution equations for the low-rank factors of the solution of a time-dependent partial differential equation (PDE).

The outline of this chapter is as follows. In Section 9.2 we review the *dynamical low-rank* integrator by Lubich et al. [47, 53, 54, 61]. In Section 9.3 it is observed that this approach can be seen as an optimization problem and KKT-conditions are derive for it. In Section 9.4 we explore an alternative two-factor matrix factorization, formulate an optimization problem and we derive KKT-conditions for it. Instead of solving one system for all factors an alternative approach similar to the alternating algorithms of Chapter 8 is presented in Section 9.5. With this method separate linear systems for the factor matrices are solved. In

Section 9.6 we apply the different discussed algorithms to two-dimensional model diffusion and Schrödinger problems. This section presents some numerical results and starts a discussion about possibilities for the methods. Further two promising algorithms are compared in a second experiment with a Schrödinger model problem. In Section 9.7 a summary and preliminary conclusions are given. Further an outlook for improvements and ideas for possibilities to extend the methods to higher-dimensional problems is given.

9.2 Review of the dynamical low-rank integrator

In a variety of applications one is interested in an approximation to time-dependent representations $\boldsymbol{H}(t) \in \mathbb{R}^{n_x \times n_y}$, for varying time t, of solutions to partial differential equations. For the applications considered in this chapter the matrix $\boldsymbol{H}(t)$ is the unknown solution of a matrix differential equation, where $f(t, \boldsymbol{H})$ is a known function:

$$\dot{\boldsymbol{H}}(t) = f(t, \boldsymbol{H}(t)). \tag{9.5}$$

The number of unknowns in these matrices can grow extensively when the dimensions n_x and n_y increase. But it is observed that in the applications under consideration the rank of matrix $\boldsymbol{H}(t)$ is low over time t. Therefore, low-rank approximations of these large matrices are considered to reduce the number of unknowns.

For this class of applications Lubich et al. introduced the *dynamical low-rank approximation* [47, 54] where matrix $\boldsymbol{H}(t)$ is approximated by a low-rank matrix $\boldsymbol{Y}(t)$, with rank $r \ll \min(n_x,\ n_y)$ such that

$$\left\|\dot{\boldsymbol{Y}}(t) - \dot{\boldsymbol{H}}(t)\right\|_F \quad \text{or} \quad \left\|\dot{\boldsymbol{Y}}(t) - f(t, \boldsymbol{Y}(t))\right\|_F \tag{9.6}$$

is minimized. Here $\|\cdot\|_F$ stands for the Frobenius norm; thus for a matrix $\boldsymbol{A}$ the Frobenius norm is given by $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$ (or $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j} a_{ij}\overline{a_{ij}}}$ if matrix $\boldsymbol{A}$ has complex valued entries).

This problem is complemented with an initial condition, for example $\boldsymbol{Y}(t_0) = \boldsymbol{H}(t_0)$, if the chosen rank r in the dynamical low-rank approximation is sufficient to describe $\boldsymbol{H}(t_0)$. Otherwise using a singular value decomposition (SVD) of $\boldsymbol{H}(t_0)$ the r largest singular values with the corresponding singular vectors can be chosen to obtain the best rank-r approximation for $\boldsymbol{H}(t_0)$ [25].

Hence $\boldsymbol{Y}(t)$ is the low-rank solution to a nonlinear differential equation for rank-r matrices such that the defect in the differential equation for a full-rank solution is minimized. The dynamical low-rank approximation is a numerical integration technique that is explicit and does not suffer from ill-conditioning of matrices arising in the differential equation [47]. Lubich presents a projector splitting algorithm that leads to a simple and less expensive time-stepping algorithm and is robust under ill-conditioning.

9.2.1 The dynamical low-rank integrator

A rank-r approximation of a matrix can be obtained by a singular value decomposition. For a fixed rank r, the low-rank approximation to $\boldsymbol{H}(t) \in \mathbb{C}^{n_x \times n_y}$, where $r \ll \min(n_x,\ n_y)$, is denoted in a unique way by

$$\boldsymbol{H}(t) \approx \boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}, \tag{9.7}$$

where $\boldsymbol{U}(t) \in \mathbb{C}^{n_x \times r}$ and $\boldsymbol{V}(t) \in \mathbb{C}^{n_y \times r}$ with both r orthonormal columns and $\boldsymbol{S}(t) \in \mathbb{R}^{r \times r}$ is a diagonal matrix containing the r largest singular values.

To slightly weaken the conditions on this factorization we keep the constraints on $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ but for $\boldsymbol{S}(t) \in \mathbb{C}^{r \times r}$ it does not necessary need to be a diagonal matrix, but it is sufficient that it is an invertible matrix. Hence this factorization is not unique. Indeed, for example choose $\widetilde{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{P}$ and $\widetilde{\boldsymbol{V}} = \boldsymbol{V}\boldsymbol{Q}$ where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are orthonormal $r \times r$ matrices. The same matrix $\boldsymbol{Y}(t)$ is obtained by choosing a new matrix $\widetilde{\boldsymbol{S}} = \boldsymbol{P}^{\mathsf{H}}\boldsymbol{S}\boldsymbol{Q}$:

$$\boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} = \underbrace{\boldsymbol{U}(t)\boldsymbol{P}(t)}_{\widetilde{\boldsymbol{U}}(t)}\underbrace{\boldsymbol{P}(t)^{\mathsf{H}}\boldsymbol{S}(t)\boldsymbol{Q}(t)}_{\widetilde{\boldsymbol{S}}(t)}\underbrace{\boldsymbol{Q}(t)^{\mathsf{H}}\boldsymbol{V}(t)^{\mathsf{H}}}_{\widetilde{\boldsymbol{V}}^{\mathsf{H}}(t)}.$$

Instead, a unique decomposition in the tangent space at $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ will be used [47]. Denote by $\mathcal{V}_{n,r}$ the set of r orthonormal vectors in $\mathbb{C}^n$. The tangent space at $\boldsymbol{U}(t) \in \mathcal{V}_{n_x,r}$ is given by

$$\mathcal{T}_{\boldsymbol{U}(t)}\mathcal{V}_{n_x,r} = \left\{\dot{\boldsymbol{U}}(t) \in \mathbb{C}^{n_x \times r} : \dot{\boldsymbol{U}}(t)^{\mathsf{H}}\boldsymbol{U}(t) + \boldsymbol{U}(t)^{\mathsf{H}}\dot{\boldsymbol{U}}(t) = 0\right\}.$$

Consider the extended tangent map of $(\boldsymbol{S},\boldsymbol{U},\boldsymbol{V}) \mapsto \boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$, where $\mathrm{so}(r)$ denotes the space of skew-symmetric $r \times r$ matrices:

$$\begin{aligned}\mathbb{C}^{r\times r} \times \mathcal{T}_{\boldsymbol{U}(t)}\mathcal{V}_{n_x,r} \times \mathcal{T}_{\boldsymbol{V}(t)}\mathcal{V}_{n_y,r} &\to \mathcal{T}_{\boldsymbol{Y}(t)}\mathcal{M}_r \times \mathrm{so}(r) \times \mathrm{so}(r),\\ \left(\dot{\boldsymbol{S}},\dot{\boldsymbol{U}},\dot{\boldsymbol{V}}\right) &\mapsto \left(\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}},\ \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}},\ \boldsymbol{V}^{\mathsf{H}}\dot{\boldsymbol{V}}\right).\end{aligned}$$

Lubich et al. mentioned that the dimensions of the vector spaces on both sides agree, it has a zero null-space and the map is an isomorphism [47]. Hence, all tangent matrices are of the form

$$\dot{\boldsymbol{Y}}(t) = \dot{\boldsymbol{U}}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}},$$

where $\dot{\boldsymbol{S}}(t) \in \mathbb{C}^{r\times r}, \dot{\boldsymbol{U}}(t) \in \mathcal{T}_{\boldsymbol{U}(t)}\mathcal{V}_{n_x,r}$ and $\dot{\boldsymbol{V}}(t) \in \mathcal{T}_{\boldsymbol{V}(t)}\mathcal{V}_{n_y,r}$. Further, imposing the gauge conditions or orthogonality constraints

$$\boldsymbol{U}^{\mathsf{H}}(t)\dot{\boldsymbol{U}}(t) = 0 \quad \text{and} \quad \boldsymbol{V}^{\mathsf{H}}(t)\dot{\boldsymbol{V}}(t) = 0 \tag{9.8}$$

yields a uniquely defined $\dot{\boldsymbol{S}}(t), \dot{\boldsymbol{U}}(t)$ and $\dot{\boldsymbol{V}}(t)$ determined by $\dot{\boldsymbol{Y}}(t)$.

The minimization condition (9.6) on the tangent space can be seen as an orthogonal projection, i.e. find $\dot{\boldsymbol{Y}}(t) \in \mathcal{T}_{\boldsymbol{Y}(t)}\mathcal{M}_r$ (where $\mathcal{M}_r$ represents the rank-r matrices) that satisfies

$$\left\langle \dot{\boldsymbol{Y}}(t) - \dot{\boldsymbol{H}}(t),\ \Delta\boldsymbol{Y}(t)\right\rangle = 0 \quad \text{for all} \quad \Delta\boldsymbol{Y}(t) \in \mathcal{T}_{\boldsymbol{Y}(t)}\mathcal{M}_r. \tag{9.9}$$

With this formulation differential equations for the factors in (9.7) can be derived:

Proposition 2 ([47, Proposition 2.1]). *For a rank-r matrix* $\boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ *with non-singular* $\boldsymbol{S}(t) \in \mathbb{C}^{r\times r}$ *and with* $\boldsymbol{U}(t) \in \mathbb{C}^{n_x\times r}$ *and* $\boldsymbol{V}(t) \in \mathbb{C}^{n_y\times r}$ *both with orthonormal columns the minimization in* (9.6) *or* (9.9) *is equivalent to* $\dot{\boldsymbol{Y}}(t) = \dot{\boldsymbol{U}}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}}$, *where*

$$\begin{aligned}\dot{\boldsymbol{U}}(t) &= \left(\boldsymbol{I} - \boldsymbol{U}(t)\boldsymbol{U}(t)^{\mathsf{H}}\right)\dot{\boldsymbol{H}}(t)\boldsymbol{V}(t)\boldsymbol{S}(t)^{-1}\\ \dot{\boldsymbol{V}}(t) &= \left(\boldsymbol{I} - \boldsymbol{V}(t)\boldsymbol{V}(t)^{\mathsf{H}}\right)\dot{\boldsymbol{H}}(t)^{\mathsf{H}}\boldsymbol{U}(t)\boldsymbol{S}(t)^{-\mathsf{H}}\\ \dot{\boldsymbol{S}}(t) &= \boldsymbol{U}(t)^{\mathsf{H}}\dot{\boldsymbol{H}}(t)\boldsymbol{V}(t).\end{aligned} \tag{9.10}$$

In theory this system of differential equations for the factors can be solved numerically. But both $\dot{\boldsymbol{U}}(t)$ and $\dot{\boldsymbol{V}}(t)$ depend on the inverse of $\boldsymbol{S}(t)$. A singularity in $\boldsymbol{S}(t)$ will break these update equations. Indeed, (nearly) singularity of $\boldsymbol{S}(t)$ is a reasonable scenario because it is related to the actual rank of the solution. Because the actual rank is unknown, and underestimation will lead to a loss of accuracy, it is reasonable to arrive in situations where the actual rank is over-estimated.

Lubich presents with the dynamical low-rank approximation a method to circumvent this problem. Condition (9.9) can be seen as a differential equation on rank-r matrices:

$$\dot{\boldsymbol{Y}}(t) = P(\boldsymbol{Y}(t))\dot{\boldsymbol{H}}(t), \tag{9.11}$$

where $P(\boldsymbol{Y})$ is the (solution dependent) orthogonal projector onto the tangent space $\mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$. The projector has a simple representation for $\boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$, as stated in [47, Lemma 4.1] and [54]:

Lemma 3 ([47, Lemma 4.1]). *The orthogonal projection onto the tangent space* $\mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$ *at* $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} \in \mathcal{M}_r$ *is given by*

$$P(\boldsymbol{Y}) = I - P^{\perp}(\boldsymbol{Y}) \quad \textit{with} \quad P^{\perp}(\boldsymbol{Y})\boldsymbol{Z} = P_{\boldsymbol{U}}^{\perp}\boldsymbol{Z}P_{\boldsymbol{V}}^{\perp}$$

where $\boldsymbol{Z} \in \mathbb{C}^{n_x\times n_y}$.

Furthermore, the projector has a simple representation and is given by

$$P(\boldsymbol{Y})\boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}, \tag{9.12}$$

where $\boldsymbol{Z} \in \mathbb{C}^{n_x\times n_y}$.

Proof. Indeed, using the gauge conditions it holds that

$$\dot{\boldsymbol{Y}} = \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}.$$

Writing out the expressions for the update equations in Proposition 2 leads to

$$\begin{aligned}\dot{\boldsymbol{Y}} &= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\left(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}\right)^{\mathsf{H}}\\ &= \dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\end{aligned}$$

Using the notation $P_{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}$, $P_{\boldsymbol{V}} = \boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}$, $P_{\boldsymbol{U}}^{\perp} = I - P_{\boldsymbol{U}}$ and $P_{\boldsymbol{V}}^{\perp} = I - P_{\boldsymbol{V}}$ this can be rewritten as

$$\begin{aligned}\dot{\boldsymbol{Y}} &= \dot{\boldsymbol{H}}P_{\boldsymbol{V}} - P_{\boldsymbol{U}}\dot{\boldsymbol{H}}P_{\boldsymbol{V}} + P_{\boldsymbol{U}}\dot{\boldsymbol{H}}\\ &= P(\boldsymbol{Y})\dot{\boldsymbol{H}},\end{aligned}$$

which holds for every matrix $\boldsymbol{Z} = \dot{\boldsymbol{H}}(t)$. □

Observe that all terms in (9.12) are in the tangent space $\mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$. For example

$$\begin{aligned} P(\boldsymbol{Y})\left(\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}\right) &= \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} \\ &= \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} \\ &= \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}. \end{aligned} \tag{9.13}$$

So $\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} \in \mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$. In a similar way one can find also $\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} \in \mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$ and $\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z} \in \mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$.

Note that $P_{\boldsymbol{U}}$ is the orthogonal projector onto the range $\mathcal{R}(\boldsymbol{Y})$ of $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}$ and $P_{\boldsymbol{V}}$ is the orthogonal projector onto the range $\mathcal{R}\left(\boldsymbol{Y}^{\mathsf{H}}\right)$. So the projector (9.12) can also be given in terms of orthogonal projectors onto ranges:

$$P(\boldsymbol{Y})\boldsymbol{Z} = \boldsymbol{Z}P_{\mathcal{R}\left(\boldsymbol{Y}^{\mathsf{H}}\right)} - P_{\mathcal{R}(\boldsymbol{Y})}\boldsymbol{Z}P_{\mathcal{R}\left(\boldsymbol{Y}^{\mathsf{H}}\right)} + P_{\mathcal{R}(\boldsymbol{Y})}\boldsymbol{Z}. \tag{9.14}$$

9.2.2 Abstract formulation of the integrator

Let us first assume that $\dot{\boldsymbol{H}}(t)$ is explicitly known[1]. Further let $\boldsymbol{Y}_0 \approx \boldsymbol{H}(t_0)$ be a rank-r approximation of the initial condition. Because of the simple representation of the projector of the form (9.14) a step of the standard Lie-Trotter splitting of (9.11) from $t = t_0$ to $t = t_1 = t_0 + \Delta t$ is given by

1. Solve the initial value problem

$$\begin{aligned} \dot{\boldsymbol{Y}}_1(t) &= \dot{\boldsymbol{H}}(t)P_{\mathcal{R}\left(\boldsymbol{Y}_1^{\mathsf{H}}\right)} \\ \boldsymbol{Y}_1(t_0) &= \boldsymbol{Y}_0. \end{aligned} \tag{9.15}$$

   on the interval $t_0 \leq t \leq t_1$.

2. Solve the initial value problem

$$\begin{aligned} \dot{\boldsymbol{Y}}_2(t) &= -P_{\mathcal{R}(\boldsymbol{Y}_2)}\dot{\boldsymbol{H}}(t)P_{\mathcal{R}\left(\boldsymbol{Y}_2^{\mathsf{H}}\right)} \\ \boldsymbol{Y}_2(t_0) &= \boldsymbol{Y}_1(t_1). \end{aligned} \tag{9.16}$$

   on the interval $t_0 \leq t \leq t_1$.

3. Solve the initial value problem

$$\begin{aligned} \dot{\boldsymbol{Y}}_3(t) &= P_{\mathcal{R}(\boldsymbol{Y}_3)}\dot{\boldsymbol{H}}(t) \\ \boldsymbol{Y}_3(t_0) &= \boldsymbol{Y}_2(t_1). \end{aligned} \tag{9.17}$$

   on the interval $t_0 \leq t \leq t_1$.

4. Finally $\boldsymbol{Y}_1 := \boldsymbol{Y}_3(t_1)$ approximates $\boldsymbol{Y}(t_1)$ as the solution to (9.11) at time $t = t_1$.

[1] In this chapter we are actually interested in $\boldsymbol{H}(t)$ where it is the unknown solution to a partial differential equation, hence $\dot{\boldsymbol{H}}(t)$ is not explicitly known. But for example in applications where $\boldsymbol{H}(t)$ represents a series of moving images both the images $\boldsymbol{H}(t)$ and updates $\dot{\boldsymbol{H}}(t)$ are known and one wants to exploit the sparsity of the updates to obtain low-rank approximations to the images over time.

This method is only first-order accurate. Lubich et al. mention that all these differential equations can be solved exactly as described in the following lemma:

Lemma 4 ([54, Lemma 3.1])**.** *The solution of* (9.15) *is given by*

$$\begin{aligned} \boldsymbol{Y}_1(t) &= \boldsymbol{U}_1(t)\boldsymbol{S}_1(t)\boldsymbol{V}_1^{\mathsf{H}}(t), \\ \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{U}_1(t)\boldsymbol{S}_1(t)\right) &= \dot{\boldsymbol{H}}(t)\boldsymbol{V}_1(t), \\ \dot{\boldsymbol{V}}_1(t) &= 0. \end{aligned} \tag{9.18}$$

The solution of (9.16) *is given by*

$$\begin{aligned} \boldsymbol{Y}_2(t) &= \boldsymbol{U}_2(t)\boldsymbol{S}_2(t)\boldsymbol{V}_2^{\mathsf{H}}(t), \\ \dot{\boldsymbol{S}}_2(t) &= -\boldsymbol{U}_2(t)^{\mathsf{H}}\dot{\boldsymbol{H}}(t)\boldsymbol{V}_2(t), \\ \dot{\boldsymbol{U}}_2(t) &= 0 \\ \dot{\boldsymbol{V}}_2(t) &= 0. \end{aligned} \tag{9.19}$$

The solution of (9.17) *is given by*

$$\begin{aligned} \boldsymbol{Y}_3(t) &= \boldsymbol{U}_3(t)\boldsymbol{S}_3(t)\boldsymbol{V}_3^{\mathsf{H}}(t), \\ \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{V}_3(t)\boldsymbol{S}_3(t)^{\mathsf{H}}\right) &= \dot{\boldsymbol{H}}(t)^{\mathsf{H}}\boldsymbol{U}_3(t), \\ \dot{\boldsymbol{U}}_3(t) &= 0. \end{aligned} \tag{9.20}$$

The solution of the differential equations (9.18), (9.19) *and* (9.20) *are given by*

$$\begin{aligned} \boldsymbol{U}_1(t)\boldsymbol{S}_1(t) &= \boldsymbol{U}_1(t_0)\boldsymbol{S}_1(t_0) + (\boldsymbol{H}(t) - \boldsymbol{H}(t_0))\boldsymbol{V}_1(t_0), \\ \boldsymbol{S}_2(t) &= \boldsymbol{S}_2(t_0) - \boldsymbol{U}_2(t_1)^{\mathsf{H}}(\boldsymbol{H}(t) - \boldsymbol{H}(t_0))\boldsymbol{V}_2(t_0), \\ \boldsymbol{V}_3(t)\boldsymbol{S}_3(t)^{\mathsf{H}} &= \boldsymbol{V}_3(t_0)\boldsymbol{S}_3(t_0)^{\mathsf{H}} + (\boldsymbol{H}(t) - \boldsymbol{H}(t_0))^{\mathsf{H}}\boldsymbol{U}_3(t_0). \end{aligned} \tag{9.21}$$

Proof. See also: [54, Lemma 3.1].

Since (9.13) results in $\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}, \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}}$ and $\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{Z} \in \mathcal{T}_{\boldsymbol{Y}}\mathcal{M}_r$ it follows that the solutions to (9.15), (9.16) and (9.17) all stay of rank r.

Therefore $\boldsymbol{Y}_i(t)$ can indeed be factorized as $\boldsymbol{Y}_i(t) = \boldsymbol{U}_i(t)\boldsymbol{S}_i(t)\boldsymbol{V}_i(t)^{\mathsf{H}}$, for $i = 1, 2, 3$ where $\boldsymbol{S}_i(t)$ is an invertible matrix and $\boldsymbol{U}_i(t)$ and $(V)_i(t)$ both have orthonormal columns.

Thus $\dot{\boldsymbol{Y}}_1(t) = (\boldsymbol{U}_1(t)\boldsymbol{S}_1(t))'\boldsymbol{V}_1^{\mathsf{H}} + \boldsymbol{U}_1(t)\boldsymbol{S}_1(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}}$. From (9.15) one has $\dot{\boldsymbol{Y}}_1(t) = \dot{\boldsymbol{H}}(t)\boldsymbol{V}_1\boldsymbol{V}_1^{\mathsf{H}}$ which is satisfied when (9.18) holds. Similar results holds for $\dot{\boldsymbol{Y}}_2(t)$ and $\dot{\boldsymbol{Y}}_3(t)$. □

9.2.3 Practical algorithm for the dynamical low-rank integrator

This abstract formulation of the integrator leads to the following practical algorithm. Given the low-rank factorization (9.7) of a rank-r matrix $\boldsymbol{H}(t_0) \approx \boldsymbol{Y}_0 = \boldsymbol{U}_0\boldsymbol{S}_0\boldsymbol{V}_0^{\mathsf{H}}$ and an explicitly known increment $\Delta\boldsymbol{H} = \boldsymbol{H}(t_1) - \boldsymbol{H}(t_0)$. A step of the integrator is presented in [54]:

1. Define and factorize $\boldsymbol{K}_1$

$$\begin{aligned} \boldsymbol{K}_1 &= \boldsymbol{U}_0 \boldsymbol{S}_0 + \Delta \boldsymbol{H} \boldsymbol{V}_0, \\ \boldsymbol{U}_1 \widehat{\boldsymbol{S}_1} &= \mathrm{qr}\left[\boldsymbol{K}_1\right], \end{aligned} \tag{9.22}$$

   with $\widehat{\boldsymbol{S}_1} \in \mathbb{R}^{r \times r}$ and $\boldsymbol{U}_1 \in \mathbb{R}^{n_x \times r}$ has orthonormal columns.

2. Define $\widetilde{\boldsymbol{S}_0}$

$$\widetilde{\boldsymbol{S}_0} = \widehat{\boldsymbol{S}_1} - \boldsymbol{U}_1^{\mathsf{H}} \Delta \boldsymbol{H} \boldsymbol{V}_0. \tag{9.23}$$

3. Define and factorize $\boldsymbol{L}_1$

$$\begin{aligned} \boldsymbol{L}_1 &= \boldsymbol{V}_0 \widetilde{\boldsymbol{S}_0}^{\mathsf{H}} + \Delta \boldsymbol{H}^{\mathsf{H}} \boldsymbol{U}_1, \\ \boldsymbol{V}_1 \boldsymbol{S}_1^{\mathsf{H}} &= \mathrm{qr}\left[\boldsymbol{L}_1\right], \end{aligned} \tag{9.24}$$

   with $\boldsymbol{S}_1 \in \mathbb{R}^{r \times r}$ and $\boldsymbol{V}_1 \in \mathbb{R}^{n_y \times r}$ has orthonormal columns.

4. Application of these steps, in this sequential order also called K, S and L-steps by the name of this auxiliary matrices, leads to the computation of a rank-r factorization of

$$\boldsymbol{H}(t_1) \approx \boldsymbol{Y}_1 = \boldsymbol{U}_1 \boldsymbol{S}_1 \boldsymbol{V}_1^{\mathsf{H}}. \tag{9.25}$$

   Observe that this $\boldsymbol{Y}_1$ is actually the same rank-r matrix as obtained in the abstract formulation in Section 9.2.2.

Lubich mentioned already that this splitting is only first order accurate and higher order approximations can be obtained by standard composition techniques such as symmetric splitting or Strang splitting.

This practical algorithm can also be extended to the case where $\boldsymbol{H}(t)$ is the unknown solution to a matrix differential equation (9.5). Lubich suggests to replace $\Delta \boldsymbol{H} = \boldsymbol{H}(t_1) - \boldsymbol{H}(t_0)$ by an expression that resembles an explicit Runge-Kutta method,

$$\Delta \boldsymbol{H} := \mathcal{RK}\left[f,\ \boldsymbol{Y}_0,\ t_0,\ \Delta t\right]. \tag{9.26}$$

In this notation f represents a function as given in (9.5), $\boldsymbol{Y}_0 = \boldsymbol{U}_0 \boldsymbol{S}_0 \boldsymbol{V}_0^{\mathsf{H}}$ represents a low-rank factorization of the solution at time $t = t_0$ and Δt is the step size. For example, explicit Euler is given by:

$$\Delta \boldsymbol{H} := \Delta t f(t_0, \boldsymbol{Y}_0).$$

In the numerical experiments we will use the classical Runge-Kutta method [36] (RK-4) as given by:

$$\begin{aligned} \boldsymbol{Q}_1 &= \Delta t f\left(t_0,\ \boldsymbol{Y}_0\right), \\ \boldsymbol{Q}_2 &= \Delta t f\left(t_0 + \tfrac{1}{2}\Delta t,\ \boldsymbol{Y}_0 + \tfrac{1}{2}\boldsymbol{Q}_1\right), \\ \boldsymbol{Q}_3 &= \Delta t f\left(t_0 + \tfrac{1}{2}\Delta t,\ \boldsymbol{Y}_0 + \tfrac{1}{2}\boldsymbol{Q}_2\right), \\ \boldsymbol{Q}_4 &= \Delta t f\left(t_0 + \Delta t,\ \boldsymbol{Y}_0 + \boldsymbol{Q}_3\right), \\ \Delta \boldsymbol{H} &:= \tfrac{1}{6}\left(\boldsymbol{Q}_1 + 2\boldsymbol{Q}_2 + 2\boldsymbol{Q}_3 + \boldsymbol{Q}_4\right). \end{aligned} \tag{9.27}$$

Algorithm 10: Lubich's dynamical low-rank KSL-algorithm for 2D problems.

Given: a low-rank approximation to $\boldsymbol{H}(t_0) \approx \boldsymbol{Y}(t_0) = \boldsymbol{U}(t_0)\boldsymbol{S}(t_0)\boldsymbol{V}(t_0)^{\mathsf{H}}$;
for $t = t_0,\ t_0 + \Delta t,\ t_0 + 2\Delta t,\ \ldots, T - \Delta t$ **do**
  $\boldsymbol{U}_0 = \boldsymbol{U}(t),\ \boldsymbol{S}_0 = \boldsymbol{S}(t),\ \boldsymbol{V}_0 = \boldsymbol{V}(t)$;
  $\Delta\boldsymbol{H} := \mathcal{RK}\left[f,\ \boldsymbol{U}_0\boldsymbol{S}_0\boldsymbol{V}_0^{\mathsf{H}},\ t,\ \Delta t\right]$;
  $\boldsymbol{K}_1 := \boldsymbol{U}_0\boldsymbol{S}_0 + \Delta\boldsymbol{H}\boldsymbol{V}_0$;
  $\boldsymbol{U}_1\widehat{\boldsymbol{S}_1} = \mathrm{qr}\left[\boldsymbol{K}_1\right]$;
  $\widetilde{\boldsymbol{S}_0} := \widehat{\boldsymbol{S}_1} - \boldsymbol{U}_1^{\mathsf{H}}\Delta\boldsymbol{H}\boldsymbol{V}_0$;
  $\boldsymbol{L}_1 := \boldsymbol{V}_0\widetilde{\boldsymbol{S}_0}^{\mathsf{H}} + \Delta\boldsymbol{H}^{\mathsf{H}}\boldsymbol{U}_1$;
  $\boldsymbol{V}_1\boldsymbol{S}_1^{\mathsf{H}} = \mathrm{qr}\left[\boldsymbol{L}_1\right]$;
  $\boldsymbol{U}(t + \Delta t) = \boldsymbol{U}_1,\ \boldsymbol{S}(t + \Delta t) = \boldsymbol{S}_1,\ \boldsymbol{V}(t + \Delta t) = \boldsymbol{V}_1$;
end

Observe that only *explicit* Runge-Kutta methods can be used here, because projections of $\Delta\boldsymbol{H}$ onto low-rank factors $\boldsymbol{V}_0$ and $\boldsymbol{U}_1$ need to be computed efficiently, see eg. (9.22), (9.23) and (9.24). In case one is interested in, for example, pure diffusion problems the spatial discretization yields stiff matrix differential equations. Therefore, the presented KSL-algorithm is then only stable and applicable under a strong constraint on the timestep Δt due to the explicit nature of the time integration method used in (9.26) for these K-, S-, and L-steps.

The algorithm, also known as the KSL-algorithm, is summarized in Algorithm 10.

In [54] a remarkable exactness result is stated for the KSL-algorithm:

Theorem 7 ([54, Theorem 4.1])**.** *Suppose $\boldsymbol{H}(t)$ has a rank of at most r for all $t \geq t_0$. Using the initial value $\boldsymbol{Y}_0 = \boldsymbol{H}(t_0)$ the splitting algorithm of Section 9.2.3 is exact, thus $\boldsymbol{Y}_1 = \boldsymbol{H}(t_1)$.*

So, this means that in case $\boldsymbol{H}(t)$ is the unknown solution to a matrix differential equation we can expect for the KSL-algorithm with a sufficiently large rank r a similar convergence behaviour as the time integrator that is used to numerically approximate that underlying differential equations for K, S and L.

9.2.4 Remarks on stability of the dynamical low-rank integrator

To analyze the stability of the dynamical low-rank integrator, let us consider a model problem where we take a two-dimensional diffusion equation with homogeneous Dirichlet boundary conditions on a bounded domain Ω. The partial differential equation is then given by

$$\frac{\partial h}{\partial t}(x, y, t) = d_{11}\frac{\partial^2 h(x, y, t)}{\partial x^2} + d_{22}\frac{\partial^2 h(x, y, t)}{\partial y^2}, \tag{9.28}$$

for some diffusion constants $d_{11} \geq 0$ and $d_{22} \geq 0$. For simplicity take $d_{11} = d_{22} = 1$.

Semi-discretization is done on a uniform spatial grid with $n_x \times n_y$ unknowns, so a numerical approximation to the solution $h(x, y, t)$ in the meshpoints defined by that mesh can be

represented by a matrix $\boldsymbol{H}(t) \in \mathbb{R}^{n_x \times n_y}$. The differential equation can now be written in the following matrix form

$$\dot{\boldsymbol{H}}(t) = \boldsymbol{D}_{xx}\boldsymbol{H}(t) + \boldsymbol{H}(t)\boldsymbol{D}_{yy}^{\mathsf{T}}$$

where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{yy}$ are sparse matrices that represent the (finite difference) discretization of the differential operators in both directions.

Applied to a low-rank approximation $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with rank r this leads to

$$\dot{\boldsymbol{H}}(t) = \boldsymbol{D}_{xx}\boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}.$$

Thus, when we drop the time-dependent argument, the equation for $\dot{\boldsymbol{U}}(t)$ in (9.10) gives

$$\begin{aligned}
\dot{\boldsymbol{U}} &= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{S}^{-1} \\
&= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\left(\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right)\boldsymbol{V}\boldsymbol{S}^{-1} \\
&= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\boldsymbol{V}\boldsymbol{S}^{-1} \\
&= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\boldsymbol{V}\boldsymbol{S}^{-1} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\boldsymbol{V}\boldsymbol{S}^{-1} \\
&= \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U}.
\end{aligned}$$

Now, we can introduce a map w that takes a matrix $\boldsymbol{U} \in \mathbb{C}^{n_x \times r}$ and maps it to

$$\boldsymbol{W} := \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U},$$

to describe the right hand side of this evolution equation.

A perturbation $\dot{\boldsymbol{U}}$ in $\boldsymbol{U}$ leads to a perturbation $\Delta\boldsymbol{W}$ in $\boldsymbol{W}$. This perturbation is, up to first order terms, given by

$$\Delta\boldsymbol{W} = \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\dot{\boldsymbol{U}} + \left(\boldsymbol{I} - \dot{\boldsymbol{U}}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \left(\boldsymbol{I} - \boldsymbol{U}\dot{\boldsymbol{U}}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U}. \tag{9.29}$$

Consider the example where $\boldsymbol{U}$ consists of two columns, i.e. $r = 2$. Then $\boldsymbol{U}, \dot{\boldsymbol{U}}$ and $\Delta\boldsymbol{W} \in \mathbb{C}^{n_x \times 2}$. We consider the eigenvectors $\boldsymbol{Z} \in \mathbb{C}^{n_x \times 2}$ that are invariant under these perturbations:

$$\left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{Z} + \left(\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{Z}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} = \lambda\boldsymbol{Z}.$$

Write these eigenvectors $\boldsymbol{Z}$ as linear combination of the columns of $\boldsymbol{U}$, thus $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{2\times 2}$. Using this expression one can obtain

$$\left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{\alpha} + \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{\alpha}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} = \lambda\boldsymbol{U}\boldsymbol{\alpha}.$$

This equation can be projected onto the columns of $\boldsymbol{U}$, so the following equation is derived

$$\begin{aligned}
\boldsymbol{U}^{\mathsf{H}}\left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{\alpha} + \boldsymbol{U}^{\mathsf{H}}\left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{\alpha}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \boldsymbol{U}^{\mathsf{H}}\left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} &= \lambda\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{\alpha} \\
\left(\boldsymbol{U}^{\mathsf{H}} - \boldsymbol{\alpha}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} + \left(\boldsymbol{U}^{\mathsf{H}} - \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{U}^{\mathsf{H}}\right)\boldsymbol{D}_{xx}\boldsymbol{U} &= \lambda\boldsymbol{\alpha} \\
2\boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U} - \left(\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathsf{T}}\right)\boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U} &= \lambda\boldsymbol{\alpha}.
\end{aligned}$$

To simplify notation we define $\widetilde{\boldsymbol{D}} := \boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}$ and write a matrix eigenvalue problem:

$$2\widetilde{\boldsymbol{D}} - \left(\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathsf{T}}\right)\widetilde{\boldsymbol{D}} = \lambda\boldsymbol{\alpha}.$$

Recall that $\boldsymbol{D}_{xx}$ is a (finite difference) discretization of the second derivative operator thus all eigenvalues are negative. Therefore also the eigenvalues of $\widetilde{\boldsymbol{D}}$ are negative.

Written as classical eigenvalue problem, this reads

$$2\widetilde{d}_{ij} - \sum_k \alpha_{ik}\widetilde{d}_{kj} - \sum_k \alpha_{ki}\widetilde{d}_{kj} = \lambda\alpha_{ij} \quad \forall i,j,$$

where we used the convention that $\widetilde{d}_{ij}$ represents the i,j-th element of matrix $\widetilde{\boldsymbol{D}}$.

Written as a linear eigenvalue system for $\mathtt{vec}[\boldsymbol{\alpha}]$ this results in

$$\begin{bmatrix} 0 & -\widetilde{d}_{21} & -\widetilde{d}_{21} & 0 \\ -2\widetilde{d}_{12} & 2\widetilde{d}_{12} - \widetilde{d}_{22} & -\widetilde{d}_{22} & 0 \\ 0 & -\widetilde{d}_{11} & 2\widetilde{d}_{21} - \widetilde{d}_{11} & -2\widetilde{d}_{21} \\ 0 & -\widetilde{d}_{12} & -\widetilde{d}_{12} & 0 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \alpha_{21} \\ \alpha_{22} \end{bmatrix} = \lambda \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \alpha_{21} \\ \alpha_{22} \end{bmatrix}.$$

Assuming that $\widetilde{\boldsymbol{D}}$ is symmetric leads to the following eigenvalues λ

$$\lambda = \begin{cases} 0, \\ 2\widetilde{d}_{12}, \\ \frac{1}{2}\left(-\widetilde{d}_{11} + 2\widetilde{d}_{12} - \widetilde{d}_{22} + \sqrt{\left(-\widetilde{d}_{11} + 2\widetilde{d}_{12} - \widetilde{d}_{22}\right)^2 + 16\widetilde{d}_{12}}\right), \\ \frac{1}{2}\left(-\widetilde{d}_{11} + 2\widetilde{d}_{12} - \widetilde{d}_{22} - \sqrt{\left(-\widetilde{d}_{11} + 2\widetilde{d}_{12} - \widetilde{d}_{22}\right)^2 + 16\widetilde{d}_{12}}\right). \end{cases}$$

Hence, at least one of these eigenvalues is positive which makes this evolution equation unstable when one wants to apply it to a pure diffusion problem.

This is also somewhat expected, because the Laplace operator can be seen as a smoother. So high frequencies are smoothed out which can also lead to a reduction of the rank. But this conflicts with the requirement that the columns of $\boldsymbol{U}(t)$ should stay normalized over time and span a space with a fixed dimension r.

9.3 Dynamical low-rank as optimization problem

The equations for $\dot{\boldsymbol{U}}(t), \dot{\boldsymbol{V}}(t)$ and $\dot{\boldsymbol{S}}(t)$ in (9.10) can also be viewed as solving an optimization problem [60]. In this formulation it is again assumed that $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ is low-rank with rank r. So the matrix $\boldsymbol{S}(t) \in \mathbb{C}^{r\times r}$ is non-singular and $\boldsymbol{U}(t) \in \mathbb{C}^{n_x\times r}$ and $\boldsymbol{V}(t) \in \mathbb{C}^{n_y\times r}$ both have orthonormal columns.

In this section we will derive the Karush–Kuhn–Tucker conditions, or KKT conditions. The KKT conditions, are (first-order) necessary conditions for a local optimizer $\boldsymbol{x}^* \in \mathbb{R}^n$ of a general optimization problem

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\ &\text{s.t.} \quad g_i(\boldsymbol{x}) = 0, \quad \text{for } i = 1,2,\ldots,p. \end{aligned} \tag{9.30}$$

where f is called the objective function. Here we restrict the attention to only equality constraints for this optimization problem. Define the *Lagrangian function* $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ for (9.30) by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x}). \tag{9.31}$$

Observe that at the optimal solution $\boldsymbol{x}^*$ there are quantities $\boldsymbol{\lambda}^*$ (where $\boldsymbol{\lambda} \in \mathbb{R}^p$ are called the Lagrange multipliers) such that

$$\nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0. \tag{9.32}$$

For an optimal solution this condition is necessary, but not sufficient. Indeed, if we want to minimize the objective function f also a solution that maximizes the objective function satisfies (9.32), thus it is not a sufficient condition.

The first-order necessary conditions, or KKT conditions, for optimization problem (9.30) are defined in eg. [60, Theorem 12.1]:

Theorem 8 ([60, Theorem 12.1])**.** *Suppose that $\boldsymbol{x}^*$ is a local solution of* (9.30) *and that the functions f and g_i are continuously differentiable and that the linear independence constraint qualification holds at $\boldsymbol{x}^*$. Then there is a Lagrange multiplier vector $\boldsymbol{\lambda}^* \in \mathbb{R}^p$ with components λ_i^* for $i = 1, 2, \ldots, p$ such that the following conditions are satisfied at $(\boldsymbol{x}^*,\ \boldsymbol{\lambda}^*)$:*

$$\begin{aligned} \nabla_x \mathcal{L}(\boldsymbol{x}^*,\ \boldsymbol{\lambda}^*) &= 0 \\ g_i(\boldsymbol{x}^*) &= 0, \quad \text{for } i = 1, 2, \ldots, p. \end{aligned}$$

9.3.1 Explicit evaluation of PDE constraint in optimization problem

We can now derive the KKT conditions for an optimization problem that solves for $\dot{\boldsymbol{U}}(t), \dot{\boldsymbol{V}}(t)$ and $\dot{\boldsymbol{S}}(t)$ similar to (9.10). The result is stated as the following lemma:

Lemma 5. *Given $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ the KKT-conditions of*

$$\begin{aligned} &\min \left\| \dot{\boldsymbol{U}}(t) \right\|_F + \left\| \dot{\boldsymbol{V}}(t) \right\|_F \\ s.t. \quad &\dot{\boldsymbol{H}}(t) = \dot{\boldsymbol{U}}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} \end{aligned} \tag{9.33}$$

for a given $\dot{\boldsymbol{H}}(t)$, $\boldsymbol{U}(t), \boldsymbol{S}(t)$ and $\boldsymbol{V}(t)$ are

$$\begin{aligned} 2\dot{\boldsymbol{U}}(t) + \boldsymbol{\lambda}(t)\boldsymbol{V}(t)\boldsymbol{S}(t)^{\mathsf{H}} &= 0 \\ 2\dot{\boldsymbol{V}}(t) + \boldsymbol{\lambda}(t)^{\mathsf{H}}\boldsymbol{U}(t)\boldsymbol{S}(t) &= 0 \\ \boldsymbol{U}(t)^{\mathsf{H}}\boldsymbol{\lambda}(t)\boldsymbol{V}(t) &= 0 \\ \dot{\boldsymbol{U}}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} &= \dot{\boldsymbol{H}}(t) \end{aligned} \tag{9.34}$$

or, eliminating the Lagrange multiplier:

$$\begin{aligned} \boldsymbol{U}(t)^{\mathsf{H}}\dot{\boldsymbol{U}}(t) &= 0 \\ \boldsymbol{V}(t)^{\mathsf{H}}\dot{\boldsymbol{V}}(t) &= 0 \\ \dot{\boldsymbol{U}}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} &= \dot{\boldsymbol{H}}(t) \end{aligned} \tag{9.35}$$

Proof. Let us drop the time-dependent argument in the notation.

The Lagrangian function is given by

$$\mathcal{L}(\dot{\boldsymbol{U}},\dot{\boldsymbol{S}},\dot{\boldsymbol{V}}) = \sum_{i,j=1}^{n_x,r} \dot{u}_{ij}^2 + \sum_{i,j=1}^{n_y,r} \dot{v}_{ij}^2 - \sum_{k,l=1}^{n_x,n_y} \lambda_{kl} \left(\dot{h}_{kl} - \sum_{m,n=1}^{r,r} \dot{u}_{km} s_{mn} v_{ln} - \sum_{m,n=1}^{r,r} u_{km} \dot{s}_{mn} v_{ln} - \sum_{m,n=1}^{r,r} u_{km} s_{mn} \dot{v}_{ln} \right).$$

The partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{U}}$ are given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{u}_{ij}} &= 2\dot{u}_{ij} + \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \sum_{m=1}^{r} \sum_{n=1}^{r} \lambda_{kl} \frac{\partial \dot{u}_{km}}{\partial \dot{u}_{ij}} s_{mn} v_{ln} \\ &= 2\dot{u}_{ij} + \sum_{l=1}^{n_y} \sum_{n=1}^{r} \lambda_{il} s_{jn} v_{ln} \\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{U}}} &= 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda} \boldsymbol{V} \boldsymbol{S}^{\mathsf{H}}. \end{aligned} \tag{9.36}$$

Further, the partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{V}}$ are given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{v}_{ij}} &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \sum_{m=1}^{r} \sum_{n=1}^{r} \lambda_{kl} u_{km} s_{mn} \frac{\partial \dot{v}_{ln}}{\partial \dot{v}_{ij}} \\ &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x} \sum_{m=1}^{r} \lambda_{ki} u_{km} s_{mj} \\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{V}}} &= 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}} \boldsymbol{U} \boldsymbol{S}. \end{aligned} \tag{9.37}$$

Finally, the partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{S}}$ are given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{s}_{ij}} &= \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \sum_{m=1}^{r} \sum_{n=1}^{r} \lambda_{kl} u_{km} \frac{\partial \dot{s}_{mn}}{\partial \dot{s}_{ij}} v_{ln} \\ &= \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \lambda_{kl} u_{ki} v_{lj} \\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{S}}} &= \boldsymbol{U}^{\mathsf{H}} \boldsymbol{\lambda} \boldsymbol{V}. \end{aligned} \tag{9.38}$$

Indeed, setting the partial derivatives w.r.t. $\dot{\boldsymbol{U}}, \dot{\boldsymbol{V}}$ and $\dot{\boldsymbol{S}}$ to zero yields the KKT conditions as given in (9.34).

Observe that the equality from (9.38)

$$\boldsymbol{U}(t)^{\mathsf{H}} \boldsymbol{\lambda}(t) \boldsymbol{V}(t) = 0 \tag{9.39}$$

can be used to eliminate the Lagrange multiplier in (9.36) and (9.37). Therefore, the equation from $\dot{\boldsymbol{U}}$ is pre-multiply with $\boldsymbol{U}^{\mathsf{H}}$ and equation from $\dot{\boldsymbol{V}}$ is pre-multiplied with $\boldsymbol{V}^{\mathsf{H}}$ to obtain:

$$\begin{aligned} 0 &= 2\boldsymbol{U}^{\mathsf{H}} \dot{\boldsymbol{U}} + \boldsymbol{U}^{\mathsf{H}} \boldsymbol{\lambda} \boldsymbol{V} \boldsymbol{S}^{\mathsf{H}} = \boldsymbol{U}^{\mathsf{H}} \dot{\boldsymbol{U}}, \\ 0 &= 2\boldsymbol{V}^{\mathsf{H}} \dot{\boldsymbol{V}} + \boldsymbol{V}^{\mathsf{H}} \boldsymbol{\lambda}^{\mathsf{H}} \boldsymbol{U} \boldsymbol{S} = \boldsymbol{V}^{\mathsf{H}} \dot{\boldsymbol{V}}, \end{aligned}$$

which results in the conditions as given in (9.35). □

Indeed, the necessary conditions to satisfy this minimization problem are exactly the gauge conditions as stated in (9.8). Using $\boldsymbol{V}(t)^{\mathsf{H}}\dot{\boldsymbol{V}}(t) = 0 = \dot{\boldsymbol{V}}(t)^{\mathsf{H}}\boldsymbol{V}(t)$ and the vectorization identity

$$\operatorname{vec}[\boldsymbol{AXB}] = \left(\boldsymbol{B}^{\mathsf{T}} \otimes \boldsymbol{A}\right) \operatorname{vec}[\boldsymbol{X}]$$

for matrices $\boldsymbol{A}, \boldsymbol{X}$ and $\boldsymbol{B}$ with appropriate sizes, equations (9.35) can be written to

$$\begin{aligned}
\left(\boldsymbol{I} \otimes \boldsymbol{U}(t)^{\mathsf{H}}\right) \operatorname{vec}\left[\dot{\boldsymbol{U}}(t)\right] &= 0 \\
\left(\boldsymbol{V}(t)^{\mathsf{T}} \otimes \boldsymbol{I}\right) \operatorname{vec}\left[\dot{\boldsymbol{V}}(t)^{\mathsf{H}}\right] &= 0 \\
\left(\overline{\boldsymbol{V}(t)}\boldsymbol{S}(t)^{\mathsf{T}} \otimes \boldsymbol{I}\right) \operatorname{vec}\left[\dot{\boldsymbol{U}}(t)\right] + \left(\overline{\boldsymbol{V}(t)} \otimes \boldsymbol{U}(t)\right) \operatorname{vec}\left[\dot{\boldsymbol{S}}(t)\right] + \left(\boldsymbol{I} \otimes \boldsymbol{U}(t)\boldsymbol{S}(t)\right) \operatorname{vec}\left[\dot{\boldsymbol{V}}(t)^{\mathsf{H}}\right] &= \operatorname{vec}\left[\dot{\boldsymbol{H}}(t)\right].
\end{aligned}$$

So, this results in a large linear system for the unknowns $\dot{\boldsymbol{U}}(t), \dot{\boldsymbol{V}}(t)$ and $\dot{\boldsymbol{S}}(t)$:

$$\underbrace{\begin{bmatrix} \overline{\boldsymbol{V}(t)}\boldsymbol{S}(t)^{\mathsf{T}} \otimes \boldsymbol{I} & \overline{\boldsymbol{V}(t)} \otimes \boldsymbol{U}(t) & \boldsymbol{I} \otimes \boldsymbol{U}(t)\boldsymbol{S}(t) \\ \boldsymbol{I} \otimes \boldsymbol{U}(t)^{\mathsf{H}} & 0 & 0 \\ 0 & 0 & \boldsymbol{V}(t)^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{J}(t)} \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{U}}(t)\right] \\ \operatorname{vec}\left[\dot{\boldsymbol{S}}(t)\right] \\ \operatorname{vec}\left[\dot{\boldsymbol{V}}(t)^{\mathsf{H}}\right] \end{bmatrix} = \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}(t)\right] \\ 0 \\ 0 \end{bmatrix} \tag{9.40}$$

where $\boldsymbol{J}(t)$ is a tall matrix with dimensions $(n_x n_y + 2r^2) \times (n_x r + r^2 + n_y r)$.

Using an explicit Runge-Kutta approximation for $\dot{\boldsymbol{H}}(t)$ as given in (9.26) this leads to a linear system of equations for the updates $\dot{\boldsymbol{U}}, \dot{\boldsymbol{V}}$ and $\dot{\boldsymbol{S}}$.

To solve this linear system a direct method that solves the normal equations can be used. To explicitly analyze the normal equations, consider $\boldsymbol{J}(t)^{\mathsf{H}}$ and drop the time-dependent argument:

$$\boldsymbol{J}(t)^{\mathsf{H}} = \boldsymbol{J}^{\mathsf{H}} = \begin{bmatrix} \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} & \boldsymbol{I} \otimes \boldsymbol{U} & 0 \\ \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}} & 0 & 0 \\ \boldsymbol{I} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}} & 0 & \overline{\boldsymbol{V}} \otimes \boldsymbol{I} \end{bmatrix}.$$

We use the property that the product of Kronecker products of matrices with appropriate dimensions is given by $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC} \otimes \boldsymbol{BD})$ [25]. Then $\boldsymbol{J}^{\mathsf{H}}\boldsymbol{J}$ is given by

$$\boldsymbol{J}^{\mathsf{H}}\boldsymbol{J} = \begin{bmatrix} \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} & \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{U} & \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}\boldsymbol{S} \\ \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}} & \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} & \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} \\ \overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}} & \overline{\boldsymbol{V}} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} & \boldsymbol{I} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} + \overline{\boldsymbol{V}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}.$$

When $\boldsymbol{U}$ and $\boldsymbol{V}$ have orthogonal columns then $\boldsymbol{J}^{\mathsf{H}}\boldsymbol{J}$ reduces even further to

$$\boldsymbol{J}^{\mathsf{H}}\boldsymbol{J} = \begin{bmatrix} \overline{\boldsymbol{S}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} & \overline{\boldsymbol{S}} \otimes \boldsymbol{U} & \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}\boldsymbol{S} \\ \boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}} & \boldsymbol{I} \otimes \boldsymbol{I} & \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{S} \\ \overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}} & \overline{\boldsymbol{V}} \otimes \boldsymbol{S}^{\mathsf{H}} & \boldsymbol{I} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{S} + \overline{\boldsymbol{V}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}.$$

The right hand side of the normal equations is given by

$$\boldsymbol{J}^{\mathsf{H}} \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}(t)\right] \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \left(\overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right) \operatorname{vec}\left[\dot{\boldsymbol{H}}\right] \\ \left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}}\right) \operatorname{vec}\left[\dot{\boldsymbol{H}}\right] \\ \left(\boldsymbol{I} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\right) \operatorname{vec}\left[\dot{\boldsymbol{H}}\right] \end{bmatrix} = \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}}\right] \\ \operatorname{vec}\left[\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\boldsymbol{V}\right] \\ \operatorname{vec}\left[\boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\right] \end{bmatrix}.$$

Algorithm 11: Solving the normal equations for the KKT-conditions using the factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with an explicit time stepping scheme for 2D problems.

```
Given: a low-rank approximation to H(t_0) ≈ Y(t_0) = U(t_0)S(t_0)V(t_0)^H;
for t = t_0, t_0 + Δt, t_0 + 2Δt, ..., T − Δt do
    Solve (9.40), i.e. (9.41), for U̇, Ṡ and V̇;
    U(t + Δt) = U(t) + U̇;
    S(t + Δt) = S(t) + Ṡ;
    V(t + Δt) = V(t) + V̇;
end
```

Thus, the normal equations to solve (9.40) are summarized by

$$\begin{bmatrix} \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} & \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{U} & \overline{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}\boldsymbol{S} \\ \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}} & \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} & \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} \\ \overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}} & \overline{\boldsymbol{V}} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} & \boldsymbol{I} \otimes \boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} + \overline{\boldsymbol{V}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{U}}\right] \\ \operatorname{vec}\left[\dot{\boldsymbol{S}}\right] \\ \operatorname{vec}\left[\dot{\boldsymbol{V}}^{\mathsf{H}}\right] \end{bmatrix} = \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}}\right] \\ \operatorname{vec}\left[\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\boldsymbol{V}\right] \\ \operatorname{vec}\left[\boldsymbol{S}^{\mathsf{H}}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\right] \end{bmatrix}. \tag{9.41}$$

This algorithm is summarized in Algorithm 11.

We remark that the KSL-algorithm solves for unknowns $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{S}, \boldsymbol{S}$ and $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{S}^{\mathsf{H}}$. Thus the last equation of (9.35) can be written as

$$\begin{aligned} \dot{\boldsymbol{U}}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\boldsymbol{S}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} &= \dot{\boldsymbol{H}}(t) \\ \dot{\boldsymbol{K}}(t)\boldsymbol{V}(t)^{\mathsf{H}} - \boldsymbol{U}(t)\dot{\boldsymbol{S}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{K}}(t)^{\mathsf{H}} &= \dot{\boldsymbol{H}}(t). \end{aligned}$$

Further in the KSL-algorithm the orthogonality of the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ is not enforced in the update for the auxiliary matrices $\boldsymbol{K}$ and $\boldsymbol{L}$ but imposed afterwards with an QR-factorization.

9.3.2 Implicit evaluation of PDE constraint in optimization problem

In this section we will explore possibilities for an implicit evaluation of $\dot{\boldsymbol{H}}(t)$ instead of an explicit evaluation of the PDE constraint in the optimization problem as we have seen in the previous section.

As an example, consider a linear differential operator on a two-dimensional domain, such as the Laplace operator Δ_2, applied on a function $h(x, y, t)$. Semi-discretization is done on a uniform spatial grid with $n_x \times n_y$ unknowns, so a numerical approximation to the function $h(x, y, t)$ in the meshpoints defined by that mesh can be represented by a matrix $\boldsymbol{H}(t) \in \mathbb{R}^{n_x \times n_y}$.

Further, given a rank-r matrix factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ of that function discretized on the grid where the columns of $\boldsymbol{U}(t) \in \mathbb{C}^{n_x \times r}$ and $\boldsymbol{V}(t) \in \mathbb{C}^{n_y \times r}$ are orthonormal and $\boldsymbol{S} \in \mathbb{C}^{r \times r}$ is an invertible matrix.

Example 9.3.1. Consider the two-dimensional heat equation as model problem with homogeneous Dirichlet boundary conditions on a bounded domain Ω. The partial differential

equation is then given by

$$\frac{\partial h}{\partial t}(x,y,t) = d_{11}\frac{\partial^2 h(x,y,t)}{\partial x^2} + 2d_{12}\frac{\partial^2 h(x,y,t)}{\partial x \partial y} + d_{22}\frac{\partial^2 h(x,y,t)}{\partial y^2}, \tag{9.42}$$

for some diffusion constants $d_{11} \geq 0$, $d_{12} \geq 0$ and $d_{22} \geq 0$. For simplicity we take $d_{11} = d_{22} = 1$ and $d_{12} = 0$.

Thus, the differential equation (9.42) can now be written in matrix form as given by

$$\dot{\boldsymbol{H}}(t) = \boldsymbol{D}_{xx}\boldsymbol{H}(t) + \boldsymbol{H}(t)\boldsymbol{D}_{yy}^{\mathsf{T}}$$

where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{yy}$ are sparse matrices that represent the (finite difference) discretization of the differential operators in both directions. Observe that $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{yy}$ are symmetric.

Let us introduce the notation $\Delta[\boldsymbol{H}(t)]$ that denotes the sum of these two matrix-matrix products as discretization of the Laplace operator:

$$\dot{\boldsymbol{H}}(t) = \Delta[\boldsymbol{H}(t)] := \boldsymbol{D}_{xx}\boldsymbol{H}(t) + \boldsymbol{H}(t)\boldsymbol{D}_{yy}^{\mathsf{T}}. \tag{9.43}$$

The integral form of a function $\boldsymbol{H}(t)$ is given by

$$\boldsymbol{H}(t) = \boldsymbol{H}(t_0) + \int_{t_0}^{t} \dot{\boldsymbol{H}}(s)\mathrm{ds}.$$

Observe that in this case we can exchange the order of integration and differentiation, so we can write

$$\int_{t_0}^{t} \dot{\boldsymbol{H}}(s)\mathrm{ds} = \int_{t_0}^{t} \Delta\left[\boldsymbol{U}(s)\boldsymbol{S}(s)\boldsymbol{V}(s)^{\mathsf{H}}\right]\mathrm{ds} = \Delta\left[\int_{t_0}^{t} \boldsymbol{U}(s)\boldsymbol{S}(s)\boldsymbol{V}(s)^{\mathsf{H}}\mathrm{ds}\right].$$

Thus, starting from a rank-r factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ and defining a new rank-r approximation of this linear differential equation for $\boldsymbol{H}(t+\Delta t)$ one has to satisfy

$$\boldsymbol{U}(t+\Delta t)\boldsymbol{S}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}} - \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} = \Delta\left[\int_{t}^{t+\Delta t} \boldsymbol{U}(s)\boldsymbol{S}(s)\boldsymbol{V}(s)^{\mathsf{H}}\mathrm{ds}\right],$$

where the last integral can be evaluated in different ways. For example, in an explicit or implicit way given by:

$$\Delta\left[\int_{t}^{t+\Delta t} \boldsymbol{U}(s)\boldsymbol{S}(s)\boldsymbol{V}(s)^{\mathsf{H}}\mathrm{ds}\right] \approx \begin{cases} \Delta t\Delta\left[\boldsymbol{U}(t)\boldsymbol{S}(s)\boldsymbol{V}(t)^{\mathsf{H}}\right] \\ \Delta t\Delta\left[(1-\theta)\boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \theta\boldsymbol{U}(t+\Delta t)\boldsymbol{S}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}}\right] \\ \Delta t\Delta\left[\boldsymbol{U}(t+\Delta t)\boldsymbol{S}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}}\right] \end{cases}$$

with $\theta \in [0,1]$. We remark that the first and last example are a special cases of the expression in the middle. Select $\theta = 0$ to obtain the first example and $\theta = 1$ for the last example.

About the notation, we mention that the Δ-symbol is used in this expression for both Δt as symbol for a time step and in $\Delta[\cdot]$ as symbol for the discretized differential operator. The meaning of this Δ-symbol should be clear from the context.

Now we can extend the formulation of KKT-conditions using an explicit evaluation of the PDE constraint in Section 9.3.1 with the idea of implicit time integration for the factor matrices. As an example we will use the θ-method for time integration, so

$$\Delta\left[\int_t^{t+\Delta t} \boldsymbol{U}(s)\boldsymbol{S}(s)\boldsymbol{V}(s)^{\mathsf{H}}\mathrm{d}s\right] \approx \Delta t\Delta\left[(1-\theta)\boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \theta\boldsymbol{U}(t+\Delta t)\boldsymbol{S}(t+dt)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}}\right] \tag{9.44}$$

with $\theta \in [0,1]$.

The optimization problem to find increments $\dot{\boldsymbol{U}}, \dot{\boldsymbol{S}}$ and $\dot{\boldsymbol{V}}$ such that $\boldsymbol{U}(t+\Delta t) = \boldsymbol{U}(t) + \dot{\boldsymbol{U}}$, $\boldsymbol{S}(t+\Delta t) = \boldsymbol{S}(t) + \dot{\boldsymbol{S}}$ and $\boldsymbol{V}(t+\Delta t) = \boldsymbol{V}(t) + \dot{\boldsymbol{V}}$ is formulated as:

$$\begin{aligned} &\min \left\|\dot{\boldsymbol{U}}\right\|_F + \left\|\dot{\boldsymbol{V}}\right\|_F \\ &\text{s.t.} \quad \boldsymbol{H}(t+\Delta t) - \boldsymbol{H}(t) = \Delta t\Delta\left[(1-\theta)\boldsymbol{H}(t) + \theta\boldsymbol{H}(t+\Delta t)\right]. \end{aligned} \tag{9.45}$$

So, using the factorization of $\boldsymbol{H}(t)$, the definition for the increments and dropping the time-dependent arguments for the factors (i.e. $\boldsymbol{U} = \boldsymbol{U}(t), \boldsymbol{S} = \boldsymbol{S}(t)$ and $\boldsymbol{V} = \boldsymbol{V}(t)$) we derive the following optimization problem:

$$\begin{aligned} &\min \left\|\dot{\boldsymbol{U}}\right\|_F + \left\|\dot{\boldsymbol{V}}\right\|_F \\ &\text{s.t.} \quad \left(\boldsymbol{U}+\dot{\boldsymbol{U}}\right)\left(\boldsymbol{S}+\dot{\boldsymbol{S}}\right)\left(\boldsymbol{V}+\dot{\boldsymbol{V}}\right)^{\mathsf{H}} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} = \Delta t\Delta\left[(1-\theta)\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \theta\left(\boldsymbol{U}+\dot{\boldsymbol{U}}\right)\left(\boldsymbol{S}+\dot{\boldsymbol{S}}\right)\left(\boldsymbol{V}+\dot{\boldsymbol{V}}\right)^{\mathsf{H}}\right]. \end{aligned}$$

Linearization[2] of the constraint yields

$$\begin{aligned} &\min \left\|\dot{\boldsymbol{U}}\right\|_F + \left\|\dot{\boldsymbol{V}}\right\|_F \\ &\text{s.t.} \quad \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} = \Delta t\Delta\left[\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \theta\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \theta\boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \theta\boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}\right]. \end{aligned}$$

Rearranging terms in the constraint yields

$$\begin{aligned} &\min \left\|\dot{\boldsymbol{U}}\right\|_F + \left\|\dot{\boldsymbol{V}}\right\|_F \\ &\text{s.t.} \quad \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} - \theta\Delta t\Delta\left[\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}\right] = \Delta t\Delta\left[\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\right]. \end{aligned} \tag{9.46}$$

Example 9.3.2 (continuing example 9.3.1)**.** For the Laplace operator in this example the optimization problem (9.46) is given by

$$\begin{aligned} &\min \left\|\dot{\boldsymbol{U}}\right\|_F + \left\|\dot{\boldsymbol{V}}\right\|_F \\ &\text{s.t.} \quad \Delta t\left(\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right) = \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} \\ &\quad -\theta\Delta t\left(\boldsymbol{D}_{xx}\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{D}_{xx}\boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} + \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right). \end{aligned}$$

Further the differential operators $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{yy}$ are symmetric.

Lemma 6. *The KKT conditions of optimization problem* (9.46) *are given by*

$$\begin{aligned} 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda}\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}} - \theta\Delta t\Delta[\boldsymbol{\lambda}]\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}} &= 0 \\ 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} - \theta\Delta t\Delta[\boldsymbol{\lambda}]^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} &= 0 \\ \boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t\boldsymbol{U}^{\mathsf{H}}\Delta[\boldsymbol{\lambda}]\boldsymbol{V} &= 0 \\ \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} - \theta\Delta t\Delta\left[\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}\right] &= \Delta t\Delta\left[\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\right], \end{aligned} \tag{9.47}$$

[2] Instead of linearization of the constraints one can also derive the KKT-conditions for these non-linear constrains and linearize the KKT-conditions afterwards.

where the differential operators in $\Delta[\cdot]$ need to be symmetric.

Eliminating the Lagrange multiplier, the compact KKT-conditions are given by:

$$\begin{aligned} \dot{\boldsymbol{U}}^{\mathsf{H}}\dot{\boldsymbol{U}} &= 0 \\ \dot{\boldsymbol{V}}^{\mathsf{H}}\dot{\boldsymbol{V}} &= 0 \\ \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} - \theta\Delta t\Delta\left[\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}\right] &= \Delta t\Delta\left[\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\right]. \end{aligned} \tag{9.48}$$

Proof. The application of the linear operator $\Delta[\cdot]$ on a space-discretized function $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ can be seen (or generalized) as a finite sum of s terms where matrix $\boldsymbol{H}(t)$ is pre- and post-multiplied by some operator matrices

$$\Delta\left[\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\right] = \sum_s \mathbb{A}^{(s)}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\mathbb{B}^{(s)}, \tag{9.49}$$

where $\mathbb{A}^{(s)}$ and $\mathbb{B}^{(s)}$ are proxies for the symmetric differential operators; further $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{S}}$ and $\widetilde{\boldsymbol{V}}^{\mathsf{H}}$ are proxies for $\boldsymbol{U}, \dot{\boldsymbol{U}}, \boldsymbol{S}, \dot{\boldsymbol{S}}, \boldsymbol{V}$ or $\dot{\boldsymbol{V}}$.

Thus the Lagrangian function is given by

$$\mathcal{L}(\dot{\boldsymbol{U}}, \dot{\boldsymbol{S}}, \dot{\boldsymbol{V}}) = \sum_{i,j=1}^{n_x,r} \dot{u}_{ij}^2 + \sum_{i,j=1}^{n_y,r} \dot{v}_{ij}^2 - \sum_{k,l}^{n_x,n_y} \lambda_{kl} \begin{pmatrix} \dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}} \\ \quad - \theta\Delta t \sum_s \mathbb{A}^{(s)}\left(\dot{\boldsymbol{U}}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\boldsymbol{S}\dot{\boldsymbol{V}}^{\mathsf{H}}\right)\mathbb{B}^{(s)} \\ \quad - \Delta t \sum_s \mathbb{A}^{(s)}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}\mathbb{B}^{(s)} \end{pmatrix}_{kl}.$$

The entries of a (generalized) term $\left(\mathbb{A}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\mathbb{B}\right)_{kl}$ are given by

$$\left(\mathbb{A}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\mathbb{B}\right)_{kl} = \sum_{m,n,p,q}^{n_x,r,r,n_y} \mathbb{A}_{km}\widetilde{u}_{mn}\widetilde{s}_{np}\widetilde{v}_{qp}\mathbb{B}_{ql}. \tag{9.50}$$

The partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{U}}$ are given by

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\dot{u}_{ij}} &= 2\dot{u}_{ij} + \sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\sum_{n=1}^{r}\sum_{p=1}^{r}\sum_{q=1}^{n_y}\lambda_{kl}\left(\mathbb{I}_{km}\frac{\partial\dot{u}_{mn}}{\partial\dot{u}_{ij}}s_{np}v_{qp}\mathbb{I}_{ql} - \theta\Delta t\sum_s \mathbb{A}^{(s)}_{km}\frac{\partial\dot{u}_{mn}}{\partial\dot{u}_{ij}}s_{np}v_{qp}\mathbb{B}^{(s)}_{ql}\right) \\ &= 2\dot{u}_{ij} + \sum_{l=1}^{n_y}\sum_{p=1}^{r}\lambda_{il}s_{jp}v_{lp} - \theta\Delta t\sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{p=1}^{r}\sum_{q=1}^{n_y}\lambda_{kl}\sum_s \mathbb{A}^{(s)}_{ki}s_{jp}v_{qp}\mathbb{B}^{(s)}_{ql} \\ \frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{U}}} &= 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda}\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}} - \theta\Delta t\sum_s \mathbb{A}^{(s)^{\mathsf{T}}}\boldsymbol{\lambda}\mathbb{B}^{(s)^{\mathsf{T}}}\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}}. \end{aligned}$$

Using that $\mathbb{A}^{(s)}$ and $\mathbb{B}^{(s)}$ are symmetric for all s this leads to the following representation of the partial derivative for $\boldsymbol{U}$:

$$\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{U}}} = 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda}\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}} - \theta\Delta t\Delta[\boldsymbol{\lambda}]\boldsymbol{V}\boldsymbol{S}^{\mathsf{H}}. \tag{9.51}$$

Further, the partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{V}}$ are given by

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \dot{v}_{ij}} &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\sum_{n=1}^{r}\sum_{p=1}^{r}\sum_{q=1}^{n_y}\lambda_{kl}\left(\mathbb{I}_{km}u_{mn}s_{np}\frac{\partial \dot{v}_{qp}}{\partial \dot{v}_{ij}}\mathbb{I}_{ql} - \theta\Delta t\sum_s \mathbb{A}^{(s)}_{km}u_{mn}s_{np}\frac{\partial \dot{v}_{qp}}{\partial \dot{v}_{ij}}\mathbb{B}^{(s)}_{ql}\right)\\ &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x}\sum_{n=1}^{r}\lambda_{ki}u_{kn}s_{nj} - \theta\Delta t\sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\sum_{n=1}^{r}\lambda_{kl}\sum_s \mathbb{A}^{(s)}_{km}u_{mn}s_{nj}\mathbb{B}^{(s)}_{il}\\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{V}}} &= 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} - \theta\Delta t\sum_s \mathbb{B}^{(s)}\boldsymbol{\lambda}^{\mathsf{H}}\mathbb{A}^{(s)}\boldsymbol{U}\boldsymbol{S}.\end{aligned}$$

Using that $\mathbb{A}^{(s)}$ and $\mathbb{B}^{(s)}$ are symmetric for all s this leads to the following representation of the partial derivative for $\boldsymbol{V}$:

$$\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{V}}} = 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S} - \theta\Delta t\Delta[\boldsymbol{\lambda}]^{\mathsf{H}}\boldsymbol{U}\boldsymbol{S}. \tag{9.52}$$

Finally, the partial derivatives of the Lagrangian w.r.t. $\boldsymbol{S}$ are given by

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \dot{s}_{ij}} &= \sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\sum_{n=1}^{r}\sum_{p=1}^{r}\sum_{q=1}^{n_y}\lambda_{kl}\left(\mathbb{I}_{km}u_{mn}\frac{\partial \dot{s}_{np}}{\partial \dot{s}_{ij}}v_{qp}\mathbb{I}_{ql} - \theta\Delta t\sum_s \mathbb{A}^{(s)}_{km}u_{mn}\frac{\partial \dot{s}_{np}}{\partial \dot{s}_{ij}}v_{qp}\mathbb{B}^{(s)}_{ql}\right)\\ &= \sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\lambda_{kl}u_{ki}v_{lj} - \theta\Delta t\sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{p=1}^{r}\sum_{q=1}^{n_y}\lambda_{kl}\sum_s \mathbb{A}^{(s)}_{km}u_{mi}v_{qj}\mathbb{B}^{(s)}_{ql}\\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{S}}} &= \boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t\boldsymbol{U}^{\mathsf{H}}\sum_s \mathbb{A}^{(s)^{\mathsf{T}}}\boldsymbol{\lambda}\mathbb{B}^{(s)^{\mathsf{T}}}\boldsymbol{V}.\end{aligned}$$

Indeed, setting the partial derivatives w.r.t. $\dot{\boldsymbol{U}}, \dot{\boldsymbol{S}}$ and $\dot{\boldsymbol{V}}$ to zero yields indeed the KKT conditions as given in (9.47).

Observe that the equality

$$\boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t\boldsymbol{U}^{\mathsf{H}}\Delta[\boldsymbol{\lambda}]\boldsymbol{V} = 0 \tag{9.53}$$

can be used to eliminate the Lagrange multiplier in (9.47). Therefore the equation from $\dot{\boldsymbol{U}}$ is pre-multiply with $\boldsymbol{U}^{\mathsf{H}}$ and equation from $\dot{\boldsymbol{V}}$ is pre-multiplied with $\boldsymbol{V}^{\mathsf{H}}$ to obtain:

$$\begin{aligned}0 &= 2\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} + \left(\boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t\boldsymbol{U}^{\mathsf{H}}\Delta[\boldsymbol{\lambda}]\boldsymbol{V}\right)\boldsymbol{S}^{\mathsf{H}} = \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}}\\ 0 &= 2\boldsymbol{V}^{\mathsf{H}}\dot{\boldsymbol{V}} + \left(\boldsymbol{V}^{\mathsf{H}}\boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U} - \theta\Delta t\boldsymbol{V}^{\mathsf{H}}\Delta[\boldsymbol{\lambda}]^{\mathsf{H}}\boldsymbol{U}\right)\boldsymbol{S} = \boldsymbol{V}^{\mathsf{H}}\dot{\boldsymbol{V}}\end{aligned}$$

which results in the conditions as given in (9.48). □

So, these KKT-conditions can again be formulated in a large linear system for the unknowns $\dot{\boldsymbol{U}}, \dot{\boldsymbol{S}}$ and $\dot{\boldsymbol{V}}$:

$$\underbrace{\begin{bmatrix}(\boldsymbol{I}-\theta\Delta t\boldsymbol{L})\left(\overline{\boldsymbol{V}}\boldsymbol{S}^{\mathsf{T}}\otimes\boldsymbol{I}\right) & (\boldsymbol{I}-\theta\Delta t\boldsymbol{L})\left(\overline{\boldsymbol{V}}\otimes\boldsymbol{U}\right) & (\boldsymbol{I}-\theta\Delta t\boldsymbol{L})\left(\boldsymbol{I}\otimes\boldsymbol{U}\boldsymbol{S}\right)\\ \boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}} & 0 & 0\\ 0 & 0 & \boldsymbol{V}^{\mathsf{T}}\otimes\boldsymbol{I}\end{bmatrix}}_{J(t)}\begin{bmatrix}\mathrm{vec}\left[\dot{\boldsymbol{U}}\right]\\ \mathrm{vec}\left[\dot{\boldsymbol{S}}\right]\\ \mathrm{vec}\left[\dot{\boldsymbol{V}}^{\mathsf{H}}\right]\end{bmatrix} = \begin{bmatrix}\mathrm{vec}\left[\dot{\boldsymbol{H}}(t)\right]\\ 0\\ 0\end{bmatrix} \tag{9.54}$$

Algorithm 12: Solving the normal equations for the KKT-conditions using the factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with an implicit time stepping scheme for 2D problems.

Given: a low-rank approximation to $\boldsymbol{H}(t_0) \approx \boldsymbol{Y}(t_0) = \boldsymbol{U}(t_0)\boldsymbol{S}(t_0)\boldsymbol{V}(t_0)^{\mathsf{H}}$;
for $t = t_0,\ t_0 + \Delta t,\ t_0 + 2\Delta t,\ , \ldots, T - \Delta t$ **do**
    Solve (9.54) for $\dot{\boldsymbol{U}}, \dot{\boldsymbol{S}}$ and $\dot{\boldsymbol{V}}$;
    $\boldsymbol{U}(t + \Delta t) = \boldsymbol{U}(t) + \dot{\boldsymbol{U}}$;
    $\boldsymbol{S}(t + \Delta t) = \boldsymbol{S}(t) + \dot{\boldsymbol{S}}$;
    $\boldsymbol{V}(t + \Delta t) = \boldsymbol{V}(t) + \dot{\boldsymbol{V}}$;
end

where $\boldsymbol{I}$ is an identity matrix and $\boldsymbol{L}$ is the discretized linear differential operator on the full grid. For example, the two-dimensional Laplace operator on the full grid is given by $\boldsymbol{L}$:

$$\boldsymbol{L} = \boldsymbol{I} \otimes \boldsymbol{D}_{xx} + \boldsymbol{D}_{yy} \otimes \boldsymbol{I}.$$

The matrix $\boldsymbol{J}(t)$ is again a tall matrix with dimensions $(n_x n_y + 2r^2) \times (n_x r + r^2 + n_y r)$. To solve for $\dot{\boldsymbol{U}}, \dot{\boldsymbol{S}}$ and $\dot{\boldsymbol{V}}$ a direct method to solve the normal equations can be used. Using this in a time stepping scheme leads to a new algorithm as summarized in Algorithm 12.

Indeed, this optimization problem and algorithm is a implicit time stepping generalization of the problem and algorithm as discussed in Section 9.3.1. When we choose $\theta = 0$ (and thus use an explicit method for time integration) the KKT-conditions of (9.47) and (9.48) reduce indeed to respectively (9.34) and (9.35).

9.4 Two-factor matrix factorization

Instead of writing a SVD-based three-factor matrix factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ for a rank-r matrix in this section we incorporate matrix $\boldsymbol{S}(t)$ in the factors $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$.

Let us start from the singular value decomposition of the spatial discretization at $t = t_0$, $\boldsymbol{H}(t_0) = \boldsymbol{U}(t_0)\boldsymbol{S}(t_0)\boldsymbol{V}(t_0)^{\mathsf{H}}$, where $\boldsymbol{U}(t_0) \in \mathbb{C}^{n_x \times r}$ and $\boldsymbol{V}(t_0) \in \mathbb{C}^{n_y \times r}$ with both r orthonormal columns and $\boldsymbol{S}(t_0) \in \mathbb{R}^{r \times r}$ is a diagonal matrix. Then, we can write (where the square root of a diagonal matrix is a diagonal matrix with square roots of the diagonal on these entries)

$$\begin{aligned}\boldsymbol{H}(t_0) &= \boldsymbol{U}(t_0)\sqrt{\boldsymbol{S}(t_0)}\sqrt{\boldsymbol{S}(t_0)}\boldsymbol{V}(t_0)^{\mathsf{H}} \\ &= \widetilde{\boldsymbol{U}}(t_0)\widetilde{\boldsymbol{V}}(t_0)^{\mathsf{H}}.\end{aligned}$$

Now it is not longer valid that the columns of $\widetilde{\boldsymbol{U}}(t_0)$ and $\widetilde{\boldsymbol{V}}(t_0)$ are orthonormal. Indeed, $\widetilde{\boldsymbol{U}}(t_0)^{\mathsf{H}}\widetilde{\boldsymbol{U}}(t_0) = \sqrt{\boldsymbol{S}(t_0)}^{\mathsf{H}}\boldsymbol{U}(t_0)^{\mathsf{H}}\boldsymbol{U}(t_0)\sqrt{\boldsymbol{S}(t_0)} = \boldsymbol{S}(t_0)$.

To simplify notation, for the remainder of this section the $\sim$-symbol is dropped above $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$.

9.4.1 Explicit evaluation of PDE constraint in optimization problem

Similar to Section 9.3.1 we can also formulate an optimization problem to solve for the low-rank factors $\boldsymbol{U}(t) \in \mathbb{C}^{n_x \times r}$ and $\boldsymbol{V}(t) \in \mathbb{C}^{n_y \times r}$ assuming that $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$, where the columns of $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ are orthogonal but not orthonormal. The result is stated as the following lemma:

Lemma 7. *Given $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ the KKT-conditions of*

$$\begin{aligned} &\min \left\|\dot{\boldsymbol{U}}(t)\right\|_F + \left\|\dot{\boldsymbol{V}}(t)\right\|_F \\ s.t. \quad &\dot{\boldsymbol{H}}(t) = \dot{\boldsymbol{U}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} \end{aligned} \tag{9.55}$$

for a given $\dot{\boldsymbol{H}}(t)$, $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ are

$$\begin{aligned} 2\dot{\boldsymbol{U}}(t) + \boldsymbol{\lambda}(t)\boldsymbol{V}(t) &= 0 \\ 2\dot{\boldsymbol{V}}(t) + \boldsymbol{\lambda}(t)^{\mathsf{H}}\boldsymbol{U}(t) &= 0 \\ \dot{\boldsymbol{U}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} &= \dot{\boldsymbol{H}}(t) \end{aligned} \tag{9.56}$$

or, eliminating the Lagrange multiplier:

$$\begin{aligned} \boldsymbol{U}(t)^{\mathsf{H}}\dot{\boldsymbol{U}}(t) - \dot{\boldsymbol{V}}(t)^{\mathsf{H}}\boldsymbol{V}(t) &= 0 \\ \dot{\boldsymbol{U}}(t)\boldsymbol{V}(t)^{\mathsf{H}} + \boldsymbol{U}(t)\dot{\boldsymbol{V}}(t)^{\mathsf{H}} &= \dot{\boldsymbol{H}}(t) \end{aligned} \tag{9.57}$$

Proof. Let us drop the time-dependent argument in the notation.

The Lagrangian function is given by

$$\mathcal{L}(\dot{\boldsymbol{U}}, \dot{\boldsymbol{V}}) = \sum_{i,j=1}^{n_x, r} \dot{u}_{ij}^2 + \sum_{i,j=1}^{n_y, r} \dot{v}_{ij}^2 - \sum_{k,l=1}^{n_x, n_y} \lambda_{kl} \left(\dot{h}_{kl} - \sum_{m=1}^{r} \dot{u}_{km} v_{lm} - \sum_{m=1}^{r} u_{km} \dot{v}_{lm} \right).$$

The partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{U}}$ are given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{u}_{ij}} &= 2\dot{u}_{ij} + \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \sum_{m=1}^{r} \lambda_{kl} \frac{\partial \dot{u}_{km}}{\partial \dot{u}_{ij}} v_{lm} \\ &= 2\dot{u}_{ij} + \sum_{l=1}^{n_y} \lambda_{il} v_{lm} \\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{U}}} &= 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda}\boldsymbol{V}. \end{aligned}$$

Further, the partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{V}}$ are given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{v}_{ij}} &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \sum_{m=1}^{r} \lambda_{kl} u_{km} \frac{\partial \dot{v}_{lm}}{\partial \dot{v}_{ij}} \\ &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x} \lambda_{ki} u_{kj} \\ \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{V}}} &= 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U}. \end{aligned}$$

Indeed, setting the partial derivatives w.r.t. $\dot{\boldsymbol{U}}$ and $\dot{\boldsymbol{V}}$ to zero yields indeed the KKT conditions as given in (9.56).

The Lagrange multipliers can be eliminated by pre-multiplication of the equation for $\dot{\boldsymbol{U}}$ with $\boldsymbol{U}^{\mathsf{H}}$ and the equation from $\dot{\boldsymbol{V}}$ can be pre-multiplied by $\boldsymbol{V}^{\mathsf{H}}$ to obtain:

$$\begin{aligned} 0 &= 2\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} + \boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V}, \\ 0 &= 2\boldsymbol{V}^{\mathsf{H}}\dot{\boldsymbol{V}} + \boldsymbol{V}^{\mathsf{H}}\boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U}. \end{aligned}$$

Thus, using the transposed version of the second equation yields

$$\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} = \dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V},$$

which results in the conditions as given in (9.57). □

To solve the KKT-conditions of (9.57) we can write a linear system and solve for $\dot{\boldsymbol{U}}(t)$ and $\dot{\boldsymbol{V}}(t)$. Indeed, the KKT conditions where the Lagrange multiplier is eliminated can be written as:

$$\underbrace{\begin{bmatrix} \overline{\boldsymbol{V}(t)} \otimes \boldsymbol{I} & \boldsymbol{I} \otimes \boldsymbol{U}(t) \\ \boldsymbol{I} \otimes \boldsymbol{U}(t)^{\mathsf{H}} & -\boldsymbol{V}(t)^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{J}(t)} \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{U}}(t)\right] \\ \operatorname{vec}\left[\dot{\boldsymbol{V}}(t)^{\mathsf{H}}\right] \end{bmatrix} = \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}(t)\right] \\ 0 \end{bmatrix}, \tag{9.58}$$

where $\boldsymbol{J}(t)$ is a tall matrix with dimensions $(n_x n_y + r^2) \times r(n_x + n_y)$.

Thus the linear system of (9.58) is highly over-determined. To solve this system the normal equations can be solved using a direct method. To gain more insight into the solution of the normal equations we will explicitly formulate the linear system for these equations. Therefore, consider $\boldsymbol{J}(t)^{\mathsf{H}}$ and drop the time-dependent argument:

$$\boldsymbol{J}(t)^{\mathsf{H}} = \boldsymbol{J}^{\mathsf{H}} = \begin{bmatrix} \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} & \boldsymbol{I} \otimes \boldsymbol{U} \\ \boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}} & -\overline{\boldsymbol{V}} \otimes \boldsymbol{I} \end{bmatrix}.$$

Then $\boldsymbol{J}^{\mathsf{H}}\boldsymbol{J}$ is given by

$$\begin{aligned} \boldsymbol{J}^{\mathsf{H}}\boldsymbol{J} &= \begin{bmatrix} \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} & \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U} - \boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{U} \\ \overline{\boldsymbol{V}} \otimes \boldsymbol{U}^{\mathsf{H}} - \overline{\boldsymbol{V}} \otimes \boldsymbol{U}^{\mathsf{H}} & \boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} + \overline{\boldsymbol{V}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}, \\ &= \begin{bmatrix} \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} & 0 \\ 0 & \boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} + \overline{\boldsymbol{V}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}, \end{aligned}$$

where the right hand side of the normal equations is given by

$$\boldsymbol{J}^{\mathsf{H}} \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}\right] \\ 0 \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I}\right) \operatorname{vec}\left[\dot{\boldsymbol{H}}\right] \\ \left(\boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\right) \operatorname{vec}\left[\dot{\boldsymbol{H}}\right] \end{bmatrix} = \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}\boldsymbol{V}\right] \\ \operatorname{vec}\left[\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\right] \end{bmatrix}.$$

So, we find a decoupled linear system for $\dot{\boldsymbol{U}}$ and $\dot{\boldsymbol{V}}$ with $r(n_x + n_y)$ equations and unknowns:

$$\begin{bmatrix} \boldsymbol{V}^{\mathsf{T}}\overline{\boldsymbol{V}} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} & 0 \\ 0 & \boldsymbol{I} \otimes \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} + \overline{\boldsymbol{V}}\boldsymbol{V}^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{U}}\right] \\ \operatorname{vec}\left[\dot{\boldsymbol{V}}^{\mathsf{H}}\right] \end{bmatrix} = \begin{bmatrix} \operatorname{vec}\left[\dot{\boldsymbol{H}}\boldsymbol{V}\right] \\ \operatorname{vec}\left[\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}\right] \end{bmatrix}. \tag{9.59}$$

Using this in a time stepping scheme leads to a new algorithm as summarized in Algorithm 13.

Algorithm 13: Solving the normal equations for the KKT-conditions using the factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with an explicit time stepping scheme for 2D problems.

Given: a low-rank approximation to $\boldsymbol{H}(t_0) \approx \boldsymbol{Y}(t_0) = \boldsymbol{U}(t_0)\boldsymbol{V}(t_0)^{\mathsf{H}}$;
for $t = t_0,\ t_0 + \Delta t,\ t_0 + 2\Delta t,\ , \ldots, T - \Delta t$ **do**
  Solve (9.58), i.e. (9.59), for $\dot{\boldsymbol{U}}$ and $\dot{\boldsymbol{V}}$;
  $\boldsymbol{U}(t + \Delta t) = \boldsymbol{U}(t) + \dot{\boldsymbol{U}}$;
  $\boldsymbol{V}(t + \Delta t) = \boldsymbol{V}(t) + \dot{\boldsymbol{V}}$;
end

Hence the solution of optimization problem (9.55) can also be formulated as the solution to a Sylvester equation. Starting again from the KKT-conditions as given in (9.57) and pre-multiply the first equation with $\boldsymbol{U}(t)$ and post-multiply the second equation with $\boldsymbol{V}(t)$ yields (where we again dropped the time-dependent argument)

$$\begin{aligned} \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} - \boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V} &= 0, \\ \dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} + \boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V} &= \dot{\boldsymbol{H}}\boldsymbol{V}. \end{aligned}$$

For readability we marked the unknown for the Sylvester equation in blue. Adding these two equations gives

$$\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} + \dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \dot{\boldsymbol{H}}\boldsymbol{V}. \tag{9.60}$$

Analogously, starting from the equations in (9.57) and post-multiplying the first equation with $\boldsymbol{V}(t)^{\mathsf{H}}$ and pre-multiplying the second equation with $\boldsymbol{U}(t)^{\mathsf{H}}$ results in

$$\begin{aligned} \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}} - \dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} &= 0 \\ \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}} &= \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}. \end{aligned}$$

Subtracting the first equation form the second equation we obtain

$$\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}} + \dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} = \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}. \tag{9.61}$$

Combining these new equations result in a new system of decoupled evolution equations for $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$:

$$\begin{aligned} \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} + \dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} &= \dot{\boldsymbol{H}}\boldsymbol{V}, \\ \boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}} + \dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} &= \boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{H}}. \end{aligned} \tag{9.62}$$

We remark that similar projections (if $\boldsymbol{U}$ and $\boldsymbol{V}$ have orthogonal columns) are also used in Section 8.3.2 to directly determine the low-rank components for the solution of an 2D Helmholtz equation.

Both equations in (9.62) are a Sylvester equation which has a general form $\boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{B} = \boldsymbol{C}$, where $\boldsymbol{X}$ is the unknown solution of the equation. The matrices that appear near the time derivatives $\dot{\boldsymbol{U}}(t)$ and $\dot{\boldsymbol{V}}(t)$ should be interpreted as mass matrices, similar to finite element theory.

For the initial state at t_0 it is known that $\boldsymbol{U}(t_0)^{\mathsf{H}}\boldsymbol{U}(t_0) = \boldsymbol{S}(t_0)$ and $\boldsymbol{V}(t_0)^{\mathsf{H}}\boldsymbol{V}(t_0) = \boldsymbol{S}(t_0)$. So, this contains the singular values of the initial state. Because the singular values decay the diagonal elements go to zero, hence these masses go to zero, which can lead to large time derivatives.

Further, also the terms $\boldsymbol{U}(t)\boldsymbol{U}(t)^{\mathsf{H}}$ and $\boldsymbol{V}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ appear in the Sylvester equations. These matrices are not necessary diagonal. However, their eigenvalues are the eigenvalues of $\boldsymbol{S}(t)$ or 0.

Lemma 8. *The eigenvalues of $\boldsymbol{U}(t)\boldsymbol{U}(t)^{\mathsf{H}}$ and $\boldsymbol{V}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ are the eigenvalues of $\boldsymbol{S}(t)$ or 0.*

Proof. Indeed, let $\boldsymbol{W}$ be the eigenmatrix of $\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}$ (where we dropped the time-dependent argument). Then we have

$$\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}. \tag{9.63}$$

If $\boldsymbol{U}$ is perpendicular to $\boldsymbol{W}$ then the eigenvalue is zero. If $\boldsymbol{W}$ is not perpendicular to $\boldsymbol{U}$ then it should lie in the range of $\boldsymbol{U}$. Hence, we can write $\boldsymbol{W} = \boldsymbol{W}_1 + \boldsymbol{W}_2$ where $\boldsymbol{W}_1 = \boldsymbol{U}\boldsymbol{\alpha}$ and $\boldsymbol{W}_2 \perp \boldsymbol{U}$, where $\boldsymbol{\alpha}$ is a coefficient matrix.

Thus, neglecting the perpendicular part, equation (9.63) reduces to

$$\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{\alpha} = \boldsymbol{U}\boldsymbol{\alpha}\boldsymbol{\Lambda}.$$

If the columns of $\boldsymbol{\alpha}$ are the eigenvectors of $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}$ then also the columns of $\boldsymbol{U}\boldsymbol{\alpha}$ are the eigenvectors of $\boldsymbol{U}\boldsymbol{U}^{\mathsf{H}}$ with the eigenvalues $\boldsymbol{\Lambda}$. □

The existence and uniqueness of the solution of a Sylvester equation given by

$$\boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{B} = \boldsymbol{C}, \tag{9.64}$$

where matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$ have dimensions $n \times n, m \times m$ and $n \times m$ respectively is known in literature, and given by e.g. [13]:

Theorem 9 ([13, Theorem 8.2.1])**.** *Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of $\boldsymbol{A}$, and $\mu_1, \mu_2, \ldots, \mu_m$ be the eigenvalues of $\boldsymbol{B}$. Then the Sylvester equation* (9.64) *has a unique solution $\boldsymbol{X}$ if and only if $\lambda_i + \mu_j \neq 0$ for all $i = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, m$. In other words, the Sylvester equation has a unique solution if and only if $\boldsymbol{A}$ and $-\boldsymbol{B}$ do not have a common eigenvalue.*

Since $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} = \boldsymbol{S}$ and $\boldsymbol{B} = \boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \boldsymbol{S}$ have the same eigenvalues there is no problem with uniqueness. However, when both $\boldsymbol{A}$ and $\boldsymbol{B}$ have zero eigenvalues then there is a problem for uniqueness.

9.4.2 Implicit evaluation of PDE constraint in optimization problem

In this section we revisit results from Section 9.4.1 and incorporate the use of an implicit time integration method similar to Section 9.3.2. Maybe this could lead to stable evolution equations for a low-rank factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$, where the columns of $\boldsymbol{U}(t) \in \mathbb{C}^{n_x \times r}$ and $\boldsymbol{V}(t) \in \mathbb{C}^{n_y \times r}$ are orthogonal but not orthonormal.

As an example we will use the θ-method for time integration, so

$$\Delta\left[\int_t^{t+\Delta t} \boldsymbol{U}(s)\boldsymbol{V}(s)^{\mathsf{H}}\mathrm{d}s\right] \approx \Delta t\Delta\left[(1-\theta)\boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}+\theta\boldsymbol{U}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}}\right] \tag{9.65}$$

with $\theta\in[0,1]$. Observe that the Δ-symbol is used in this expression for both Δt as symbol for a time step and in $\Delta[\cdot]$ as symbol for the discretized differential operator.

Now we can again formulate an optimization problem to find increments $\dot{\boldsymbol{U}}$ and $\dot{\boldsymbol{V}}$ such that $\boldsymbol{U}(t+\Delta t)=\boldsymbol{U}(t)+\dot{\boldsymbol{U}}$ and $\boldsymbol{V}(t+\Delta t)=\boldsymbol{V}(t)+\dot{\boldsymbol{V}}$:

$$\begin{aligned}&\min \left\|\dot{\boldsymbol{U}}\right\|_F+\left\|\dot{\boldsymbol{V}}\right\|_F\\ \text{s.t.}\quad &\boldsymbol{H}(t+\Delta t)-\boldsymbol{H}(t)=\Delta t\Delta\left[(1-\theta)\boldsymbol{H}(t)+\theta\boldsymbol{H}(t+\Delta t)\right].\end{aligned}$$

Using the factorization of $\boldsymbol{H}(t)=\boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$, the definition for the increments and dropping the time-dependent arguments for the factors (i.e. $\boldsymbol{U}=\boldsymbol{U}(t)$ and $\boldsymbol{V}=\boldsymbol{V}(t)$) we derive the following nonlinear optimization problem:

$$\begin{aligned}&\min \left\|\dot{\boldsymbol{U}}\right\|_F+\left\|\dot{\boldsymbol{V}}\right\|_F\\ \text{s.t.}\quad &\left(\boldsymbol{U}+\dot{\boldsymbol{U}}\right)\left(\boldsymbol{V}+\dot{\boldsymbol{V}}\right)^{\mathsf{H}}-\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}=\Delta t\Delta\left[(1-\theta)\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}+\theta\left(\boldsymbol{U}+\dot{\boldsymbol{U}}\right)\left(\boldsymbol{V}+\dot{\boldsymbol{V}}\right)^{\mathsf{H}}\right].\end{aligned}$$

We remark that one can linearize the constraint and derive KKT-conditions or first derive (non-linear) KKT-conditions and linearize these conditions afterwards. Here we start with linearization of the constraint, which yields

$$\begin{aligned}&\min \left\|\dot{\boldsymbol{U}}\right\|_F+\left\|\dot{\boldsymbol{V}}\right\|_F\\ \text{s.t.}\quad &\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}+\cancel{\dot{\boldsymbol{U}}\dot{\boldsymbol{V}}^{\mathsf{H}}}=\Delta t\Delta\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}+\theta\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\theta\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}+\cancel{\theta\dot{\boldsymbol{U}}\dot{\boldsymbol{V}}^{\mathsf{H}}}\right].\end{aligned}$$

Finally, rearrange terms yields the following optimization problem for implicit evaluation of the two-factor matrix factorization

$$\begin{aligned}&\min \left\|\dot{\boldsymbol{U}}\right\|_F+\left\|\dot{\boldsymbol{V}}\right\|_F\\ \text{s.t.}\quad &\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}-\theta\Delta t\Delta\left[\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\right]=\Delta t\Delta\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right].\end{aligned} \tag{9.66}$$

Example 9.4.1 (continuing example 9.3.1)**.** For the Laplace operator of this example the optimization problem (9.66) is given by

$$\begin{aligned}&\min \left\|\dot{\boldsymbol{U}}\right\|_F+\left\|\dot{\boldsymbol{V}}\right\|_F\\ \text{s.t.}\quad &\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}-\theta\Delta t\left(\boldsymbol{D}_{xx}\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{D}_{xx}\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}+\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right)=\Delta t\left(\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right).\end{aligned}$$

Lemma 9. *The KKT conditions of optimization problem* (9.66) *are given by*

$$\begin{aligned}2\dot{\boldsymbol{U}}+\boldsymbol{\lambda}\boldsymbol{V}-\theta\Delta t\Delta[\boldsymbol{\lambda}]\boldsymbol{V}&=0\\ 2\dot{\boldsymbol{V}}+\boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U}-\theta\Delta t\Delta[\boldsymbol{\lambda}]^{\mathsf{H}}\boldsymbol{U}&=0\\ \dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}-\theta\Delta t\Delta\left[\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\right]&=\Delta t\Delta\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right],\end{aligned} \tag{9.67}$$

where the differential operators in $\Delta[\cdot]$ need to be symmetric.

Eliminating the Lagrange multiplier, the compact KKT-conditions are given by:

$$\begin{aligned}\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}}-\dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V}&=0\\ \dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}-\theta\Delta t\Delta\left[\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}}+\boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\right]&=\Delta t\Delta\left[\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\right]\end{aligned} \tag{9.68}$$

Proof. The application of the linear operator $\Delta[\cdot]$ on a space-discretized function $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ can be generalized as a finite sum of s terms where $\boldsymbol{H}(t)$ is pre- and post-multiplied by some operator matrices

$$\Delta\left[\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\right] = \sum_s \mathbb{A}^{(s)}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\mathbb{B}^{(s)}, \tag{9.69}$$

where $\mathbb{A}^{(s)}$ and $\mathbb{B}^{(s)}$ are proxies for the symmetric differential operators; further $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}^{\mathsf{H}}$ are proxies for $\boldsymbol{U}, \dot{\boldsymbol{U}}, \boldsymbol{V}$ or $\dot{\boldsymbol{V}}$.

The Lagrangian function is given by

$$\mathcal{L}(\dot{\boldsymbol{U}}, \dot{\boldsymbol{V}}) = \sum_{i,j=1}^{n_x,r} \dot{u}_{ij}^2 + \sum_{i,j=1}^{n_y,r} \dot{v}_{ij}^2 + \sum_{k,l}^{n_x,n_y} \lambda_{kl} \begin{pmatrix} \dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}} \\ \quad - \theta\Delta t \sum_s \mathbb{A}^{(s)}\left(\dot{\boldsymbol{U}}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{U}\dot{\boldsymbol{V}}^{\mathsf{H}}\right)\mathbb{B}^{(s)} \\ \quad - \Delta t \sum_s \mathbb{A}^{(s)}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\mathbb{B}^{(s)} \end{pmatrix}_{kl}.$$

Observe that the kl-th entry of the (generalized) term $\mathbb{A}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\mathbb{B}$ can be written as

$$\left(\mathbb{A}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{V}}^{\mathsf{H}}\mathbb{B}\right)_{kl} = \sum_{m,n,p}^{n_x,r,n_y} \mathbb{A}_{km}\widetilde{u}_{mn}\widetilde{v}_{pn}\mathbb{B}_{pl}. \tag{9.70}$$

The partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{U}}$ are given by

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\dot{u}_{ij}} &= 2\dot{u}_{ij} + \sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\sum_{n=1}^{r}\sum_{p=1}^{n_y}\lambda_{kl}\left(\mathbb{I}_{km}\frac{\partial\dot{u}_{mn}}{\partial\dot{u}_{ij}}v_{pn}\mathbb{I}_{pl} - \theta\Delta t\sum_s \mathbb{A}_{km}^{(s)}\frac{\partial\dot{u}_{mn}}{\partial\dot{u}_{ij}}v_{pn}\mathbb{B}_{pl}^{(s)}\right) \\
&= 2\dot{u}_{ij} + \sum_{l=1}^{n_y}\lambda_{il}v_{lj} - \theta\Delta t\sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{p=1}^{n_y}\lambda_{kl}\sum_s \mathbb{A}_{ki}^{(s)}v_{pj}\mathbb{B}_{pl}^{(s)} \\
\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{U}}} &= 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t\sum_s \mathbb{A}^{(s)^{\mathsf{T}}}\boldsymbol{\lambda}\mathbb{B}^{(s)^{\mathsf{T}}}\boldsymbol{V}.
\end{aligned}$$

Using that $\mathbb{A}^{(s)}$ and $\mathbb{B}^{(s)}$ are symmetric for all s this leads to the following representation of the partial derivative for $\boldsymbol{U}$:

$$\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{U}}} = 2\dot{\boldsymbol{U}} + \boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t\Delta[\boldsymbol{\lambda}]\boldsymbol{V}. \tag{9.71}$$

Further, the partial derivatives of the Lagrangian w.r.t. $\dot{\boldsymbol{V}}$ are given by

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\dot{v}_{ij}} &= 2\dot{v}_{ij} + \sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\sum_{n=1}^{r}\sum_{p=1}^{n_y}\lambda_{kl}\left(\mathbb{I}_{km}u_{mn}\frac{\partial\dot{v}_{pn}}{\partial\dot{v}_{ij}}\mathbb{I}_{pl} - \theta\Delta t\sum_s \mathbb{A}_{km}^{(s)}u_{mn}\frac{\partial\dot{v}_{pn}}{\partial\dot{v}_{ij}}\mathbb{B}_{pl}^{(s)}\right) \\
&= 2\dot{v}_{ij} + \sum_{k=1}^{n_x}\lambda_{ki}u_{kj} - \theta\Delta t\sum_{k=1}^{n_x}\sum_{l=1}^{n_y}\sum_{m=1}^{n_x}\lambda_{kl}\sum_s \mathbb{A}_{km}^{(s)}u_{mj}\mathbb{B}_{il}^{(s)} \\
\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{V}}} &= 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U} - \theta\Delta t\sum_s \mathbb{B}^{(s)}\boldsymbol{\lambda}^{\mathsf{H}}\mathbb{A}^{(s)}\boldsymbol{U}.
\end{aligned}$$

Algorithm 14: Solving the normal equations for the KKT-conditions using the factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with an implicit time stepping scheme for 2D problems.

Given: a low-rank approximation to $\boldsymbol{H}(t_0) \approx \boldsymbol{Y}(t_0) = \boldsymbol{U}(t_0)\boldsymbol{V}(t_0)^{\mathsf{H}}$;
for $t = t_0,\ t_0 + \Delta t,\ t_0 + 2\Delta t,\ \ldots, T - \Delta t$ **do**
  Solve (9.74) for $\dot{\boldsymbol{U}}$ and $\dot{\boldsymbol{V}}$;
  $\boldsymbol{U}(t + \Delta t) = \boldsymbol{U}(t) + \dot{\boldsymbol{U}}$;
  $\boldsymbol{V}(t + \Delta t) = \boldsymbol{V}(t) + \dot{\boldsymbol{V}}$;
end

Using that $\mathbb{A}^{(s)}$ and $\mathbb{B}^{(s)}$ are symmetric for all s this leads to the following representation of the partial derivative for $\boldsymbol{V}$:

$$\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{V}}} = 2\dot{\boldsymbol{V}} + \boldsymbol{\lambda}^{\mathsf{H}}\boldsymbol{U} - \theta\Delta t \Delta[\boldsymbol{\lambda}]^{\mathsf{H}}\boldsymbol{U}. \tag{9.72}$$

Indeed, setting the partial derivatives w.r.t. $\dot{\boldsymbol{U}}$ and $\dot{\boldsymbol{V}}$ to zero yields indeed the KKT conditions as given in (9.67).

Again, the equation from $\dot{\boldsymbol{U}}$ can be pre-multiply with $\boldsymbol{U}^{\mathsf{H}}$ and the equation from $\dot{\boldsymbol{V}}$ can be transposed and post-multiplied with $\boldsymbol{V}$ to obtain:

$$\begin{aligned} 0 &= 2\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} + \boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t \boldsymbol{U}^{\mathsf{H}}\Delta[\boldsymbol{\lambda}]\boldsymbol{V}, \\ 0 &= 2\dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V} + \boldsymbol{U}^{\mathsf{H}}\boldsymbol{\lambda}\boldsymbol{V} - \theta\Delta t \boldsymbol{U}^{\mathsf{H}}\Delta[\boldsymbol{\lambda}]\boldsymbol{V}. \end{aligned}$$

Thus, combining these equations leads to

$$\boldsymbol{U}^{\mathsf{H}}\dot{\boldsymbol{U}} = \dot{\boldsymbol{V}}^{\mathsf{H}}\boldsymbol{V}, \tag{9.73}$$

which results in the conditions as given in (9.68). □

To solve the KKT-conditions of (9.68) one can write a linear system and solve for $\dot{\boldsymbol{U}}(t)$ and $\dot{\boldsymbol{V}}(t)$. Indeed, the KKT conditions where the Lagrange multiplier is eliminated can be written as:

$$\underbrace{\begin{bmatrix} (\boldsymbol{I} - \theta\Delta t\boldsymbol{L})\left(\overline{\boldsymbol{V}(t)} \otimes \boldsymbol{I}\right) & (\boldsymbol{I} - \theta\Delta t\boldsymbol{L})(\boldsymbol{I} \otimes \boldsymbol{U}(t)) \\ \boldsymbol{I} \otimes \boldsymbol{U}(t)^{\mathsf{H}} & -\boldsymbol{V}(t)^{\mathsf{T}} \otimes \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{J}(t)} \begin{bmatrix} \mathtt{vec}\left[\dot{\boldsymbol{U}}(t)\right] \\ \mathtt{vec}\left[\dot{\boldsymbol{V}}(t)^{\mathsf{H}}\right] \end{bmatrix} = \begin{bmatrix} \mathtt{vec}\left[\dot{\boldsymbol{H}}(t)\right] \\ 0 \end{bmatrix} \tag{9.74}$$

where $\boldsymbol{J}(t) \in \mathbb{C}^{(n_x n_y + r^2) \times r(n_x + n_y)}$ is a tall matrix, $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{L}$ is the discretized linear differential operator on the full grid. Solving the normal equations using a direct method in a time stepping scheme leads to a new algorithm as summarized in Algorithm 14.

Indeed, this optimization problem and algorithm is a implicit time stepping generalization of the problem and algorithm as discussed in Section 9.4.1. When we choose $\theta = 0$ (and thus use an explicit method for time integration) the KKT-conditions of (9.67) and (9.68) reduce indeed to respectively (9.56) and (9.57).

9.5 Alternating method to solve for factor matrices

Assume again a two-factor matrix factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$. Instead of solving one large system of coupled equations to obtain an increment for both the factors $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ we can also alternate between solving for $\boldsymbol{U}(t+\Delta t)$ and $\boldsymbol{V}(t+\Delta t)$, similar to the alternating algorithms used to approximate the low-rank solution to the Helmholtz equation, see e.g. Section 8.3.

Example 9.5.1 (continuing example 9.3.1)**.** We consider again the semi-discretization of the heat equation (9.42) with

$$\dot{\boldsymbol{H}}(t) = \Delta[\boldsymbol{H}(t)] := \boldsymbol{D}_{xx}\boldsymbol{H}(t) + \boldsymbol{H}(t)\boldsymbol{D}_{yy}^{\mathsf{T}}$$

Given $\boldsymbol{U}(t), \boldsymbol{V}(t)$ and the discretized differential operator $\Delta[\cdot]$. Using the θ-method for implicit time integration leads to the following matrix equation for $\boldsymbol{U}(t+\Delta t)$ and $\boldsymbol{V}(t+\Delta t)$:

$$\begin{aligned}\boldsymbol{U}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}} - \theta\Delta t\left(\boldsymbol{D}_{xx}\boldsymbol{U}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}} + \boldsymbol{U}(t+\Delta t)\boldsymbol{V}(t+\Delta t)^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right) \\ = \boldsymbol{H}(t) + (1-\theta)\Delta t\left(\boldsymbol{D}_{xx}\boldsymbol{H}(t) + \boldsymbol{H}(t)\boldsymbol{D}_{yy}^{\mathsf{T}}\right) \\ = \boldsymbol{F}(t)\end{aligned} \tag{9.75}$$

with $\theta \in [0,\ 1]$.

For all timesteps t_i with $i = 1, 2, \ldots, N$ an alternating approach to solve for the factors $\boldsymbol{U}(t+\Delta t)$ and $\boldsymbol{V}(t+\Delta t)^{\mathsf{H}}$ can be applied. So, in a certain timestep we start from (9.75) with a guess $\boldsymbol{V}(t+\Delta t) \approx \boldsymbol{V}(t)$, multiply from the right with $\boldsymbol{V}$ and solve for $\boldsymbol{U}(t+\Delta t)$. Thus, we obtain

$$\boldsymbol{U} - \theta\Delta t\left(\boldsymbol{D}_{xx}\boldsymbol{U} + \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\boldsymbol{V}\right) = \boldsymbol{F}\boldsymbol{V}, \tag{9.76}$$

where we used already the orthogonality of the columns of $\boldsymbol{V}$, i.e. $\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \boldsymbol{I} \in \mathbb{R}^{r\times r}$. Vectorizing this equation and rearranging terms yields

$$\left[(\boldsymbol{I}\otimes\boldsymbol{I}) - \theta\Delta t\,(\boldsymbol{I}\otimes\boldsymbol{D}_{xx}) - \theta\Delta t\left(\boldsymbol{V}^{\mathsf{T}}\boldsymbol{D}_{yy}\overline{\boldsymbol{V}}\otimes\boldsymbol{I}\right)\right]\mathrm{vec}\,[\boldsymbol{U}] = \mathrm{vec}\,[\boldsymbol{F}\boldsymbol{V}]\,. \tag{9.77}$$

So, we obtain a linear system of $n_x \times r$ equations for the same amount of unknowns.

In a similar way an update equation for $\boldsymbol{V}(t+\Delta t)$ is derived by pre-multiplying (9.75) with $\boldsymbol{U}^{\mathsf{H}}$, which leads to

$$\boldsymbol{V}^{\mathsf{H}} - \theta\Delta t\left(\boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}} + \boldsymbol{V}^{\mathsf{H}}\boldsymbol{D}_{yy}^{\mathsf{T}}\right) = \boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}. \tag{9.78}$$

Again, we used already the orthogonality of the columns of $\boldsymbol{U}$, i.e. $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} = \boldsymbol{I} \in \mathbb{R}^{r\times r}$. Vectorizing this equation and rearranging terms yields an linear system of $n_y \times r$ equations and unknowns:

$$\left[(\boldsymbol{I}\otimes\boldsymbol{I}) - \theta\Delta t\left(\boldsymbol{I}\otimes\boldsymbol{U}^{\mathsf{H}}\boldsymbol{D}_{xx}\boldsymbol{U}\right) - \theta\Delta t\,(\boldsymbol{D}_{yy}\otimes\boldsymbol{I})\right]\mathrm{vec}\left[\boldsymbol{V}^{\mathsf{H}}\right] = \mathrm{vec}\left[\boldsymbol{U}^{\mathsf{H}}\boldsymbol{F}\right]. \tag{9.79}$$

Alternating between solving for $\boldsymbol{U}$ and $\boldsymbol{V}$ using (9.77) and (9.79) results in an algorithm to approximate low-rank factors for a new time step. To maintain the orthogonality of the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ additional QR-factorizations are included. So, we derived the time integration algorithm as given in Algorithm 15. Remark that this algorithm is similar to Algorithm 4 used to solve for the low-rank matrix decomposition of a solution to a 2D Helmholtz problem.

Algorithm 15: Alternating solve for the low-rank factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with an explicit (if $\theta = 0$) or implicit (if $\theta \neq 0$) time stepping scheme for 2D problems.

Given: the low-rank factors of initial condition $\boldsymbol{U}(0) \in \mathbb{C}^{n_x \times r}$ and $\boldsymbol{V}(0) \in \mathbb{C}^{n_y \times r}$;
for $t = 0, \Delta t,\ 2\Delta t,\ \ldots,\ T - \Delta t$ **do**
  Pre-compute $\boldsymbol{F} = \boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}} + (1-\theta)\Delta t \Delta \left[\boldsymbol{U}(t)\boldsymbol{V}(t)^{\mathsf{H}}\right]$;
  $[\boldsymbol{V}, \boldsymbol{R}] = \mathtt{qr}\left[\boldsymbol{V}(t)\right]$;
  while *not converged* **do**
    Solve (9.77) for $\boldsymbol{U}$;
    $[\boldsymbol{U}, \widetilde{\boldsymbol{R}}] = \mathtt{qr}\left[\boldsymbol{U}\right]$;
    Solve (9.79) for $\boldsymbol{V}^{\mathsf{H}}$;
    $[\boldsymbol{V}, \boldsymbol{R}] = \mathtt{qr}\left[\boldsymbol{V}\right]$;
  end
  $\boldsymbol{U}(t + \Delta t) = \boldsymbol{U}$;
  $\boldsymbol{V}(t + \Delta t) = \boldsymbol{V}\boldsymbol{R}$;
end

9.6 Numerical examples and discussion

In this section we present some small numerical examples and explore the possibilities of the derived methods by applying them to the heat equation as a model problem. This will give some basic insights in the different methods and leads to ideas which type of methods can be useful for further analysis.

As a second numerical example we compare the two promising algorithms and apply them to a Schrödinger model problem where we have a conservation property. A numerical comparison shows good results of these algorithms for the low-rank approximations to the solution of this time-dependent partial differential equation.

9.6.1 Comparison of all algorithms: diffusion model problem

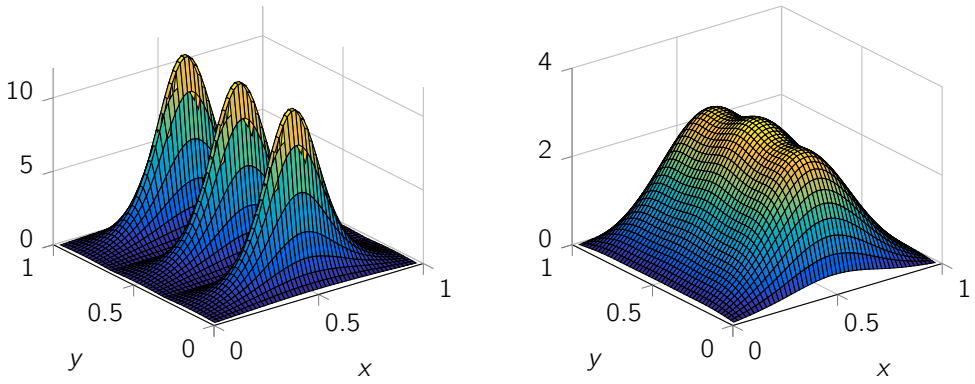
Let us start with a short comparison of all the discussed algorithms in this chapter, i.e. the explicit timestepping methods of Algorithm 10, 11, 12, 15 and also the implicit versions in Algorithm 13, 14, 15. This numerical example is based on Example 9.3.1 where the heat equation as model problem is considered.

Thus, we will numerically solve the partial differential equation

$$\frac{\partial h}{\partial t}(x, y, t) = d_{11}\frac{\partial^2 h(x, y, t)}{\partial x^2} + 2d_{12}\frac{\partial^2 h(x, y, t)}{\partial x \partial y} + d_{22}\frac{\partial^2 h(x, y, t)}{\partial y^2}, \tag{9.80}$$

on a two-dimensional space domain $(x, y) \in \Omega = (0,\ 1)^2$ and $t \in (0,\ T]$ with $T = 0.01$. For simplicity we take diffusion constants $d_{11} = d_{22} = 1$ and $d_{12} = 0$.

The two dimensional domain $\Omega = (0,\ 1)^2$ is discretized with $n_x = n_y = 75$ uniform internal



(a) Initial condition ($t = 0$). (b) Solution ($t = T$).

Figure 9.1: Plot of initial condition and solution for diffusion model problem as given in (9.80).

meshpoints and a standard finite difference scheme is used to discretize the second derivatives. Thus the numerical solution at time t can be represented by a matrix $\boldsymbol{H}(t) \in \mathbb{R}^{n_x \times n_y}$.

For the explicit versions of the algorithms (i.e. Algorithm 10, 11, 13 and 15) the classical RK-4 time integration method (9.27) is used to approximate $\dot{\boldsymbol{H}}(t)$. To obtain stability of this method $N = 225$ timesteps are used for all algorithms to approximate the solution between $t = 0$ and $t = T = 0.01$. For the implicit version of the algorithms (i.e. Algorithm 12, 14 and 15) the Crank-Nicolson time integration method (i.e. $\theta = \frac{1}{2}$) is used to approximate $\dot{\boldsymbol{H}}(t)$.

The applied initial condition for this PDE is of low-rank and given by

$$h(x, y, t = 0) = \sum_{k=1}^{12} \left|\sin(\pi x)\sin(3\pi y)\right|^k. \tag{9.81}$$

A plot of this initial condition and the numerical solution at $t = T = 0.01$ is shown in Figure 9.1. Finally, homogeneous Dirichlet boundary conditions are applied on the boundary of domain Ω.

We remark that, by construction, the initial condition as given in (9.81) is a linear combination of 12 rank-1 functions. Thus, in principle, the rank of this initial condition should be at most 12. Recall that the solution $\boldsymbol{H}(t)$ for all $t \geq 0$ has a certain numerical rank. The rank of the approximations for all methods to this solution can be monitored at all time frames t_i, with $i = 0, 1, \ldots, N$. To approximate the numerical rank the singular values of the approximation on the full discretization grid are computed and the numerical rank is chosen to be the number of singular values larger than a chosen tolerance $\texttt{tol} = 10^{-12}$. Due to the rapid decay of singular values in (9.81) for the initial condition the numerical rank is only $r = 10$, as shown in Figure 9.2a by the reference solution.

In the first part of this experiment for all presented (low-rank) methods no actual constraints on the rank are enforced, i.e. the maximal supported rank of the low-rank methods is chosen equal to the number of meshpoints in the directions. This is an interesting case because the

actual rank of the numerical solution is lower then the maximal supported rank in the low-rank methods. So, if the method performs well it demonstrate some good properties regarding over-estimation of the numerical rank of the solution. As shown in Figure 9.2a almost all presented methods maintain a low-rank approximation over time t. It is observed that only the rank for Algorithms 13 and 14 (i.e. the two-factor matrix factorization with explicit and implicit time integration) increases starting from the first timestep. This may indicate already some errors in these low-rank approximation methods. Indeed, if we consider the error with respect to the reference solution we see again large errors for these two algorithms. In Figures 9.2b and 9.2c the errors of the different algorithms with respect to the reference solution are shown for respectively the RK-4-based and CN-based methods.

Apart from the fact that the two-factor factorization $\boldsymbol{H}(t) = \widetilde{\boldsymbol{U}}(t)\widetilde{\boldsymbol{V}}(t)^{\mathsf{H}}$ as discussed in Section 9.4, thus Algorithms 13 and 14, does not yield good low-rank approximations, all other algorithms have errors almost equal to the underlying time integration method applied on the full grid. So, all these algorithms show in this example robustness under over-estimation of the numerical rank of the actual solution.

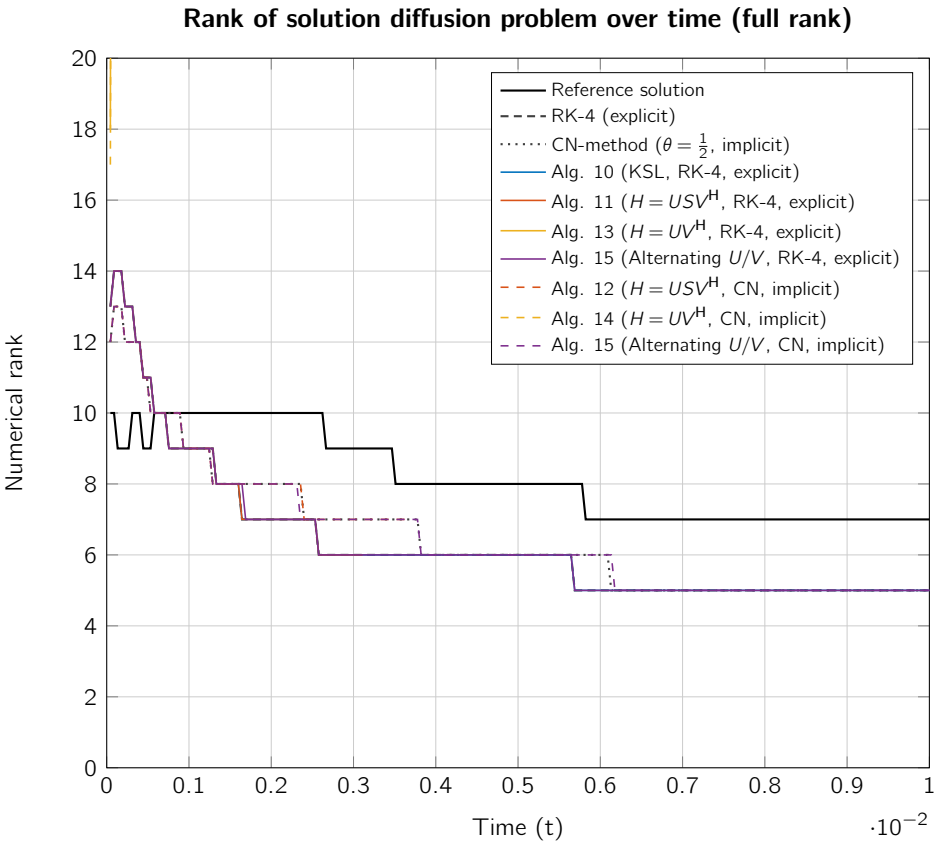
The low-rank approximation methods are developed to reduce the total number of unknowns. Thus instead of a maximal supported rank equal to the number of unknowns per direction the maximal attainable rank is now reduced. For this example pre-knowledge is used to choose the maximal attainable rank equal to $r = 10$. Of course, in general the maximal attainable rank of the solution over time is unknown and some knowledge or heuristics about the solution to the problem need to be known or estimated. The numerical rank of the low-rank solution over time is shown in Figure 9.3a.

Based on the numerical rank of the low-rank approximation it is already clear that Algorithm 11, where the factorization $\boldsymbol{H}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}(t)^{\mathsf{H}}$ with explicit time integration is used, does not obtain any reasonable solution; at least for $t \in (0,\ 0.4)$. All other methods show at least stable results where a low-rank approximation to the solution is obtained.

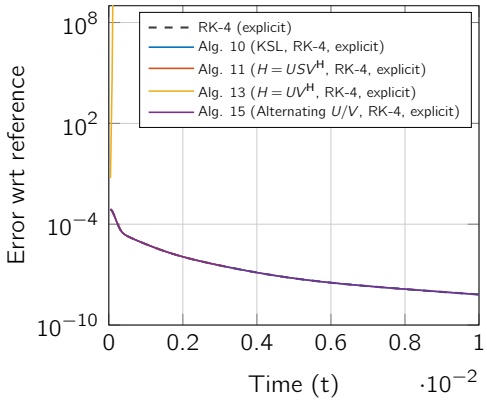
The error for a low-rank approximation can be measured as the difference of the solution by a low-rank method compared with the solution using the reference time integration method. Thus the approximation of the low-rank methods with explicit time integration are compared with the classical RK-4 time method and the implicit low-rank methods are compared with the CN reference solution. If we consider the errors in the low-rank approximations as given in Figures 9.3b and 9.3c for respectively the explicit and implicit time integration methods we see indeed large errors for Algorithms 11 and 12. Again also the low-rank approximation errors for Algorithms 13 and 14 are large.

The KSL-algorithm of Algorithm 10 show only a minor increase of error. This is probably due to the under-estimation of the RK-4 solution on the full mesh at time $t \in (0,\ 0.1)$. Also the Alternating U/V algorithms of Algorithm 15 where an explicit RK-4 method or an implicit CN-method are used show remarkably small errors for to this low-rank approximation. In the Alternating U/V algorithms there is per timestep an inner iteration to solve for $\boldsymbol{U}$ and $\boldsymbol{V}$. For this inner loop two iterations per timestep are used.

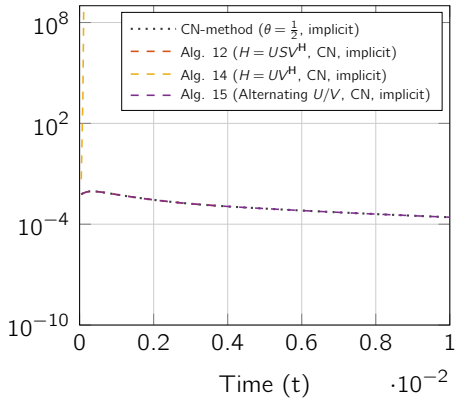
The results of this numerical experiment to explore the low-rank performance of the presented algorithms are summarized in Table 9.1. In this experiment we found that good low-rank approximations to solutions of time dependent diffusion PDEs can be obtained by



(a) Numerical rank of 'low-rank' approximations to (9.80).

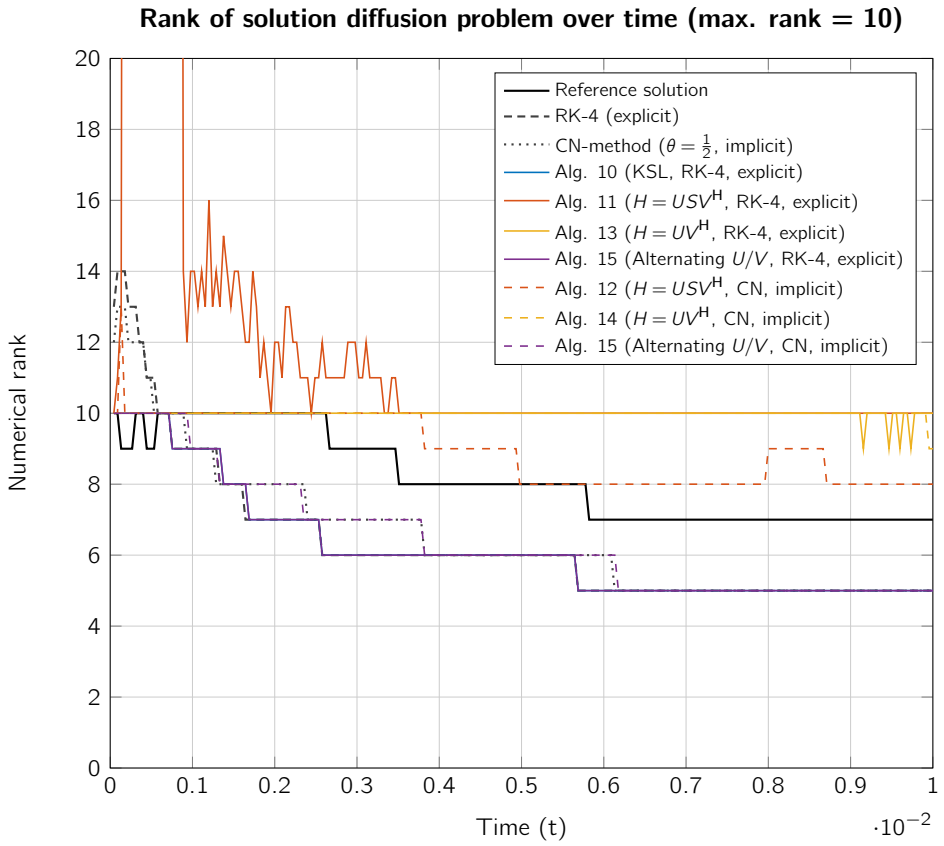


(b) Error of 'low-rank' approximations for explicit algorithms.

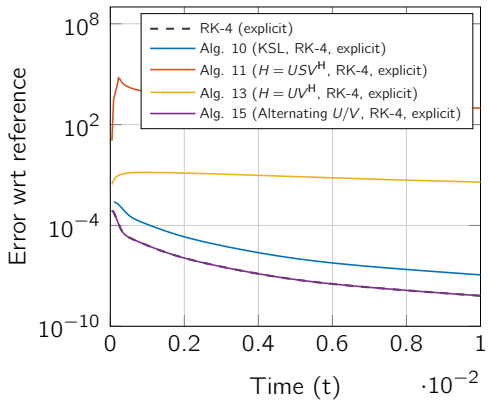


(c) Error of 'low-rank' approximations for implicit algorithms.

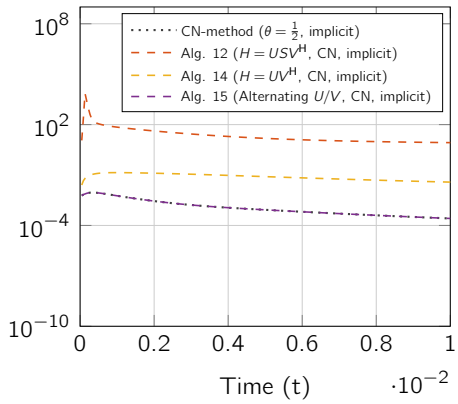
Figure 9.2: Numerical rank (top) and error of 'low-rank' approximations to solution of (9.80) for different explicit (bottom/left) and implicit (bottom/right) algorithms as discussed throughout this chapter. Maximal supported rank is set to the number of spatial discretization points per direction.



(a) Numerical rank of low-rank approximations to (9.80).



(b) Error of low-rank approximations for explicit algorithms.



(c) Error of low-rank approximations for implicit algorithms.

Figure 9.3: Numerical rank (top) and error of low-rank approximations to solution of (9.80) for different explicit (bottom/left) and implicit (bottom/right) algorithms as discussed throughout this chapter. Maximal supported rank is set to $r = 10$.

Algorithm	Explicit/Implicit	Max rank	Low-rank
Alg. 10 (KSL)	Explicit	Yes	Yes
Alg. 11 ($H = USV^{\mathsf{H}}$)	Explicit	Yes	No
Alg. 13 ($H = UV^{\mathsf{H}}$)	Explicit	No	No
Alg. 15 (Alternating U/V)	Explicit	Yes	Yes
Alg. 12 ($H = USV^{\mathsf{H}}$)	Implicit	Yes	No
Alg. 14 ($H = UV^{\mathsf{H}}$)	Implicit	No	No
Alg. 15 (Alternating U/V)	Implicit	Yes	Yes

Table 9.1: Summary of stability of the different algorithms for maximal supported rank and low-rank approximations to solutions of the pure diffusion problem (9.80).

Algorithm 10 (i.e. the KSL-algorithm) or Algorithm 15. We remark that to our knowledge Algorithm 10 can only be used with explicit time integration methods and Algorithm 15 has support for explicit and implicit time integration methods.

9.6.2 Comparison of stable algorithms: Schrödinger model problem

Based on the results of Section 9.6.1, in this section we will do a further comparison of the two promising algorithms for low-rank approximations (i.e. Algorithm 10 and 15). Therefore we consider a second two-dimensional numerical model problem with a conservation property (when the absorbing boundary conditions are neglected).

Let us consider a semi-discretized Schrödinger equation, as given by

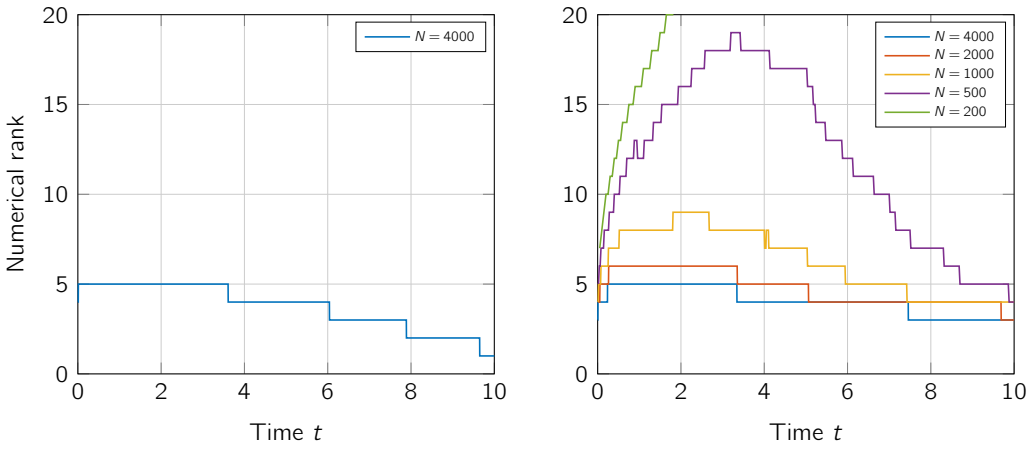
$$\frac{\mathrm{d}\boldsymbol{H}(t)}{\mathrm{d}t} = -i\left(\boldsymbol{D}_{xx}\boldsymbol{H}(t) + \boldsymbol{H}(t)\boldsymbol{D}_{yy}^{\mathsf{T}}\right), \tag{9.82}$$

where $i = \sqrt{-1}$ and $\boldsymbol{H}(t) \in \mathbb{C}^{n_x \times n_y}$ is the (matrix)discretization of the solution $h(x, y, t)$ discretized on a two-dimensional mesh with uniform grid points $x, y \in [-L,\ L]$ with $L = 10$. Further the domain is extended with exteriour complex scaling to implement absorbing boundary conditions [2, 76]. To implement the absorbing boundary conditions an artificial layer is added to the numerical domain that dampens outgoing waves. The outgoing wave boundary conditions are then replaced with homogeneous Dirichlet boundary conditions at the end of the artificial layer. This boundary do not require any knowledge about the asymptotic behaviour, which may be very complicated in different applications.

In this numerical example we consider $M = 200$ discretization points per direction in the interior of the domain. At the boundaries $x \pm L$ and $y = \pm L$ the domain is extended with exterior complex scaling under an angle $\frac{\pi}{6}$ where 33% additional discretization points are added. Thus with $2 \times 33\%$ additional discretization points the total number of discretization points per direction is given by $n_x = n_y = 334$.

In this numerical experiment we will consider the following rank-1 initial condition $\boldsymbol{H}(0) = f(x, y)$ on the discretized mesh:

$$f(x, y) = e^{-x^2 - y^2}. \tag{9.83}$$



(a) Rank of RK-4 solution

(b) Rank of CN solution

Figure 9.4: Numerical rank (i.e. #singular values $> \texttt{tol} = 10^{-12}$) of RK-4 (left) and CN (right) solutions to PDE (9.82) with initial condition $\boldsymbol{H}(0) = f(x, y)$ (2D, $M = 200$). The number of time steps is denoted by N.

For a stable explicit time integration method the number of timesteps is chosen as $N = 4000$. The numerical rank of the RK-4 and CN solution over time is shown in Figure 9.4. Indeed, the rank of the numerical solution is low over time. The Crank-Nicolson method can also be used with larger timesteps and then we see that the rank of the numerical solution (slightly) increases. After a certain time the rank of the numerical solution decreases again.

Recall that Algorithm 10 reduces with a sufficiently large rank to the RK-4 time integration method and the implicit alternating U/V method of Algorithm 15 reduces to the CN-method. Therefore we measure the error due to the low-rank approximation of both methods in terms of the difference with respect to the appropriate full-rank time integration method. These differences over time are shown in Figures 9.5a and 9.5b. Clearly both methods can obtain any desired accurate low-rank approximation to the solution starting from a rank-1 initial condition.

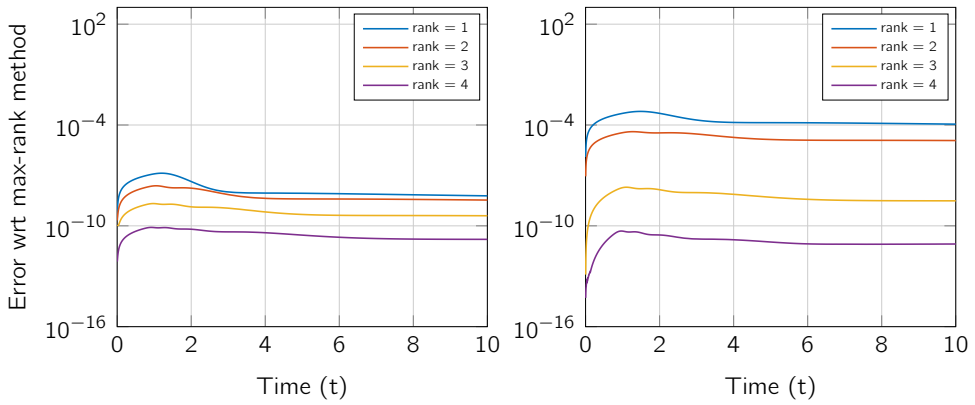
As an alternative we consider also the same model problem, but now with an initial condition given by

$$g(x, y) = e^{-x^2 - y^2 - |x-y|^4}. \tag{9.84}$$

This initial condition represented on this mesh does not have a rank-1 expression. The first singular values of this initial condition are given in Figure 9.6a. The numerical rank (with $\texttt{tol} = 10^{-8}$) of the RK-4 and CN solutions over time is shown in Figure 9.6b.

The errors in the low-rank approximations to the solutions for both Algorithm 10 and 15 are shown in Figure 9.7. Again, the time-dependent solution can be approximated with these low-rank time integration methods.

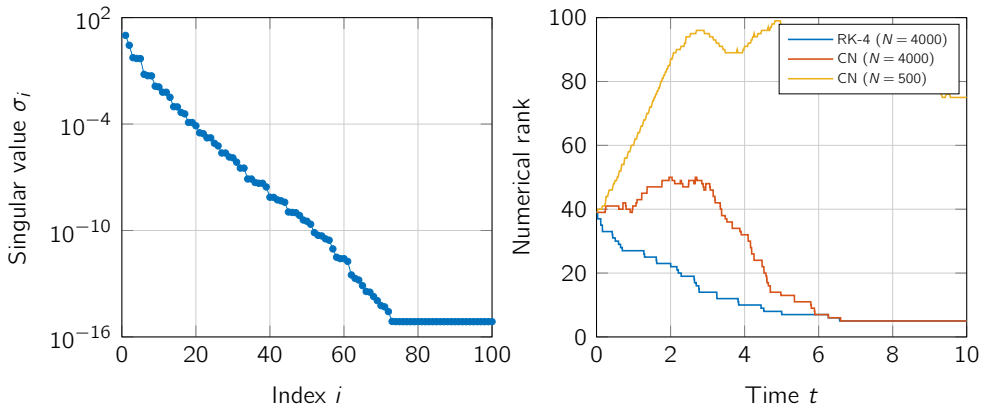
Per timestep the computations of Algorithm 15 are more expensive then the computations for Algorithm 10. The cost of a timestep with the KSL-algorithm is mainly determined by a constant number of (sparse)matrix-vector products, where the cost of a (sparse)matrix-



(a) KSL low-rank approximation (RK-4, explicit)

(b) Alternating U/V low-rank approximation (CN, implicit)

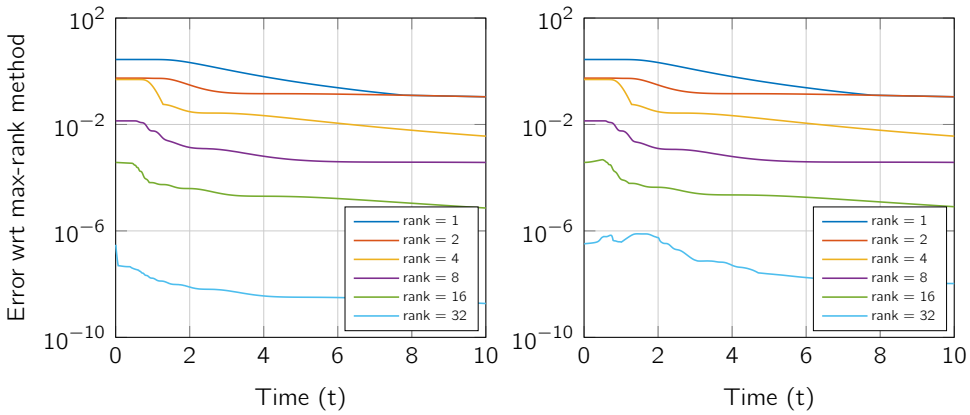
Figure 9.5: Error of different low-rank approximations of Algorithm 10 (KSL, left) and Algorithm 15 (Alternating U/V, right) w.r.t. the full-rank solution over time with initial condition $\boldsymbol{H}(0) = f(x, y)$ where f is given in (9.83) (2D, $M = 200$, $N = 4000$).



(a) Singular values of $g(x, y)$.

(b) Numerical rank of solutions.

Figure 9.6: Initial condition and numerical rank (i.e. #singular values $>$ `tol` $= 10^{-8}$) of RK-4 and CN solution to PDE (9.82) with initial condition $\boldsymbol{H}(0) = g(x, y)$ where g is given in (9.84) (2D, $M = 200$).



(a) KSL low-rank approximation (RK-4, explicit)

(b) Alternating U/V low-rank approximation (CN, implicit)

Figure 9.7: Error of different low-rank approximations of Algorithm 10 (KSL, left) and Algorithm 15 (Alternating U/V, right) w.r.t. the full-rank solution over time with initial condition $\boldsymbol{H}(0) = g(x, y)$ where g is given in (9.84) (2D, $M = 200$, $N = 4000$).

vector scale $\mathcal{O}(r)$ with rank r. For an implicit timestep with Algorithm 15 a constant number of linear systems has to be solved. Because the systems that has to solved are similar to the systems in Chapter 8, the cost for a timestep with the Alternating U/V algorithm scale $\mathcal{O}(r^2)$, where r is the rank.

We can exploit the potential larger timesteps for the Alternating U/V algorithm, but as we have seen in Figures 9.4b and 9.6b the rank of the CN-solution increases when the number of timesteps decreases. Moreover, in the examples considered here the stability condition on the timestep using a RK-4 method is not strong enough to make the additional costs for the Alternating U/V algorithm beneficial. For example, with initial condition g as given in (9.84) we can reduce the number of timesteps for the CN-method to $N = 500$. The error in the low-rank approximations over time is shown in Figure 9.8. Clearly the errors in the low-rank approximations increases and low-rank approximations with larger maximal attainable ranks r are needed to compensate for that.

9.7 Conclusion and outlook

In this chapter a short literature review about Lubich's dynamical low-rank integrator and the KSL-algorithm is given. We found a similar formulation of this algorithm as PDE constraint optimization problem. For different problems the KKT-conditions are derived and algorithms are formulated to solve for evolution equations of low-rank factor matrices where the product of these factors should satisfy a partial differential equation.

Similar to the alternating approach as used in [80] to compute the low-rank factors of the solution of Helmholtz equations also an Alternating U/V algorithm for time integration is discussed.

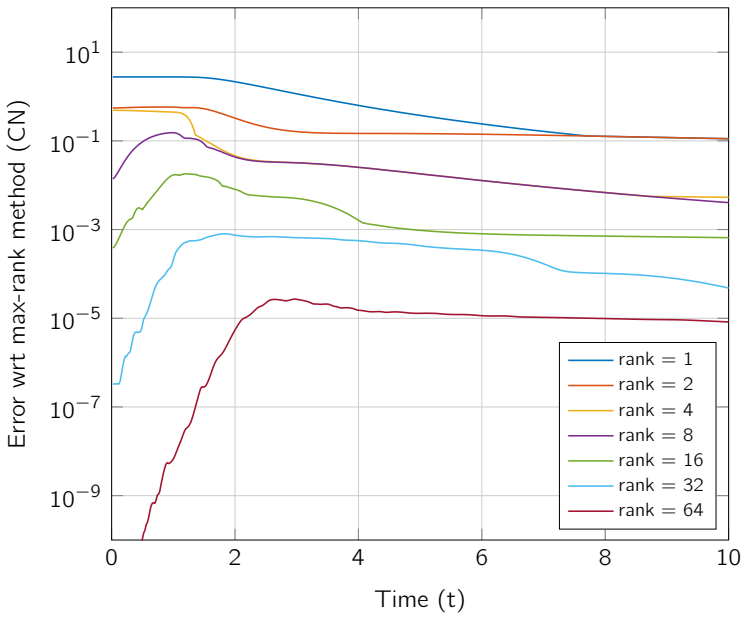


Figure 9.8: Error of different low-rank approximations of Algorithm 14 (Alternating U/V) w.r.t. full-rank solution over time with initial condition $\boldsymbol{H}(0) = g(x, y)$ where g is given in (9.84) (2D, $M = 200$, $N = 500$).

In a first numerical example with a pure diffusion model problem we have seen that both explicit and implicit algorithms that use a three-factor or two-factor matrix factorization are not stable when one coupled system for all factors has to be solved. The alternating or projection based methods (i.e. Algorithm 10 and Algorithm 15) can be used to obtain a time-dependent low-rank factorization of the solution to a partial differential equation.

The Alternating U/V algorithm can also be used together with implicit time integration. At the moment this advantage of the Alternating U/V algorithm does not lead to an efficient algorithm to solve low-rank problems like the Schrödinger model problem. Maybe in contrast to the direct methods that are currently used a good iterative solver exists to efficiently solve the linear systems that appear in the Alternating U/V algorithm. This could potentially reduce the computational cost and make it comparable to the efficient KSL-algorithm.

Also extensions to higher-dimensional problems using, for example, Tucker tensors decompositions are possible, similar to the results for the Helmholtz problem [80]. For the Alternating U/V algorithm this extensions is rather straightforward because that algorithm alternates between solving for all factors separately. Also extensions for the KSL-algorithm in higher-dimensions are already known in literature, such as [55, 64].

Chapter 10

Conclusions and outlook

10.1 Conclusions

In this thesis we studied and developed efficient numerical methods to approximate solutions to high-dimensional PDEs. A standard discretization of partial differential equations gets infeasable when the dimension of the problem becomes larger, eg. for dimensions $d > 3$. We studied approaches to overcome this problem by approximating the differential operator in Part I or to describe the solution explicitly using a low-rank factorization in Part II.

10.1.1 PCA-based approximation approach

The approximation of a differential operator using the principal component analysis (PCA) based approximation approach, as originally presented by Reisinger & Wittum, fits nicely into the Black–Scholes framework. Using the correlations of the underlying assets, the Black–Scholes operator for a d-dimensional basket option can be transformed to a d-dimensional pure diffusion problem. Then, the diffusion coefficients in each direction are equal to the eigenvalues of the covariance matrix of the underlying assets. Because of the intrinsic correlations of assets in a financial market, the first eigenvalue of this covariance matrix will often be dominant. Using a first-order Taylor-expansion in the first eigenvalue, the PCA-based approximation approach for the solution of the Black–Scholes equation is defined. It can be seen as a one-dimensional principal component approximation with first-order correction terms in all other directions. Thus instead of solving a d-dimensional PDE the PCA-based approximation approach approximates the fair option value at a certain point by a linear combination of solutions to 1 one-dimensional and $(d-1)$ two-dimensional PDEs.

The PCA-based approximation approach is in Chapter 3 applied to approximate high-dimensional Black–Scholes PDEs and valuate European-style basket options. We studied in detail the error in the spatial and temporal discretization and observed a favourable, near second-order convergence behaviour. In different numerical experiments we recovered the expected second-order convergence of the total discretization error despite the non-smoothness of the initial condition (which was remediated with cell averaging and backward Euler damping).

In Chapter 4 we extended the PCA-based approximation approach to valuate also Bermudan-style basket options. Therefore the temporal discretization was changed to implement the optimal exercise condition at a finite number of possible exercise times. Numerical experiments similar to what was done for European-style basket options show again nearly second-order convergence of the total discretization error. Compared to the European-style basket options some irregularities or oscillations are observed in the convergence behaviour of the total discretization error. A further numerical study shows that the leading term from the one-dimensional PDE behaves regular and that the correction terms can be both positive and negative, which leads to the observed irregular behaviour. More research has to be done to explain this irregular behaviour and to determine a suitable remedy for it.

To valuate American-style basket options, in Chapter 5 the PCA-based approximation approach was extended even further. The valuation of American-style baskets options requires the solution of a partial differential complementarity problem (PDCP). The PCA-based approximation approach is formulated in terms of solutions to one- and two-dimensional PDCPs. Temporal discretization is done with the Ikonen–Toivanen (IT) splitting technique. Further in that chapter we compared the PCA-based approximation approach with the comonotonic approach. The comonotonic approach was formulated for European- and American-style basket options. The comonotonic approach defines an approximation to the option value through a certain linear combination of an upper and lower bound for the option value. These lower and upper bounds are solutions to one-dimensional PDEs or PDCPs. The upper bound is rather crude. The lower bound is acquired upon replacing the volatility by a specific other value, a value that is based on other theory about comonotonic upper and lower bounds. We observed that these lower and upper bounds are exactly the solutions of the PCA-based approximation approach where the covariance matrix is set to a rank-1 matrix. Numerical experiments confirm also for American-style basket options nearly second order convergence of the total discretization error. Also in case of American-style basket options some irregularities in the convergence behaviour are observed, which may be caused by the non-smoothness of the payoff function that affects the numerical solution in all timesteps due to the optimal exercise condition.

Finally in Chapter 6 the Greeks Delta and Gamma are approximated using the PCA-based approximation approach for European-, Bermudan- and American-style basket options. We observed again second order convergence behaviour for the total discretization error but some oscillations or irregular behaviour are clearly visible. We compared the approximations for the Deltas with an estimation obtained using (Least Squares) Monte Carlo simulation, as introduced in Chapter 2.

10.1.2 Direct approximation of low-rank factors of solutions

The observation that the solution of some differential equations is of low rank can be exploited to efficiently approximate the low-rank factors of that solution.

In Chapter 8 we developed a alternating projection method for the Helmholtz problem that reduces the computational cost of solving the original differential equation when the solution is of low rank. Instead of solving on the full grid, an alternating projection method is used to solve for all low-rank factor matrices separately.

Thus without first solving a large linear system, each of the factors of the low-rank components of a solution can be obtained. Further we linked the equations that has to be solved to obtain the low-rank factor matrices with the equations arising in the coupled channel technique. For this alternating projection method we observed that the errors and residuals decay quickly in each iteration. Thus only a limited number of iterations for the alternating projection algorithm are needed.

We considered atomic and molecular breakup reactions, such as multiple-ionization and solved the Helmholz problem to numerically validate this low-rank approach. The cross sections can accurately be computed with only a low-rank solution.

We presented the concept with a two-dimensional Helmholtz problem and extended it to a three-dimensional problem where we solved for the factor matrices of a Tucker tensor decomposition of the solution. Also numerical results for a three-dimensional problem are shown. In theory, the generalization using a Tucker tensor decomposition for dimensions $d > 3$ is straightforward. But for dimensions $d > 3$ it might be beneficial to change to another tensor factorization, such as a Tensor Train decomposition. This is because the number of unknowns in the Tucker tensor decomposition has still an exponential dependence on the dimension of the problem. It is expected that a similar alternating projection method can be applied to directly solve for the low-rank factors of tensors in the Tensor Train format.

In Chapter 9 we explored some possibilities for different implicit and explicit methods to solve for the low-rank factors of solution to time-dependent partial differential equations. The dynamical low-rank integrator by Lubich is a well-known method from literature. To our knowledge it is only applicable with explicit time integration methods, which makes it less attractive for stiff differential equations. We formulated that algorithm as an optimization problem and derived alternative methods. Numerical experiments show that these alternative methods are not stable.

The alternating projection method as discussed in Chapter 8 can be combined with both implicit and explicit time integration methods. In principle, this yields an alternative to the dynamical low-rank integrator. But the linear systems that has to be solved for the implicit alternating projection method are too expensive to arrive at a computational cost that is similar to the dynamical low-rank integrator, at least in the considered numerical example with a model Schrödinger problem.

10.2 Outlook and further research

To conclude this chapter we give an outlook and some suggestions for further research.

The PCA-based approximation approach is extensively used throughout this thesis but a rigorous analysis of the error in the PCA-based approximation with respect to the fair value of the option is only known for European-style basket options in the literature. We expect that these results can also be extended to Bermudan- and American-style basket options.

Further, we expect that the irregular convergence behaviour in the total discretization error may be related to the (non-smooth) optimal exercise condition that is essential for the

Bermudan- and American-style basket options. But a detailed analysis is still lacking.

For American-style basket options we compared the PCA-based approximation approach with the comonotonic approach and saw that the approximations of both techniques lie close to each other. But at this moment it is still open which (if any) of the two approaches is to be preferred for the approximate valuation of American basket options. In particular, whereas in our experiments the two approaches always define approximations that lie close to each other, it is not clear at present which approach generally yields the smallest error with respect to the exact option value.

Further, a more fundamental question concerns about wider applicability of the PCA-based approximation approach. In the current formulation the PCA-based approximation approach relies extensively on the correlations of the underlying assets in the Black–Scholes model. It is not clear if, and how, this could be generalized to other models, for example with a non-constant volatility as used in the Heston model.

We intended the alternating projection algorithm, from Chapter 8, for the low-rank factors of solutions as a method to solve high-dimensional problems. But approximating high-dimensional data using the Tucker tensor decomposition weakens, but does not solve, the curse of dimensionalty. The Tucker tensor decomposition represents the tensor with $\mathcal{O}\left(r^d + dnr\right)$ unknowns. So, using this decomposition the total number of unknowns is reduced, but it is still exponential in the dimension d. Maybe other orthogonality preserving tensor decompositions, such as the Tensor Train decomposition, with a number of unknowns that is only polynomial in d can resolve this problem and make the alternating projection algorithm also applicable for higher dimensions.

In Chapter 8, a six-dimensional problem is solved by an expansion in spherical harmonics (i.e. the eigenfunctions of part of the operator) that reduces the problem to a coupled two-dimensional problem. We then wrote the two-dimensional components as low-rank matrices. However, directly writing the six-dimensional problem as a low-rank tensor decomposition and solving for these components, might be a more efficient method.

Further we explored different possibilities to solve time-dependent PDEs using a time integration version of the alternating projection method. This method can be combined with, for example, the Crank–Nicolson scheme or θ-method for time integration. This advantage of the alternating projection algorithm does not lead to an efficient algorithm to solve low-rank problems like the Schrödinger model problem. Maybe, in contrast to the direct methods that are currently used, a good iterative solver exists to efficiently solve the linear systems that appear in the alternating projection algorithm. This could potentially reduce the computational cost and make it comparable with the dynamical low-rank integrator. If this could be resolved then also high-dimensional time-dependent PDEs could be solved with these kind of methods.

Appendix A

Parameter sets for numerical experiments with basket options

Set A is given by Reisinger & Wittum [71] and has $d = 5$, $K = 1$, $T = 1$, $r = 0.05$ and

$$\boldsymbol{\rho} = (\rho_{ij})_{i,j=1}^{d} = \begin{pmatrix} 1.00 & 0.79 & 0.82 & 0.91 & 0.84 \\ 0.79 & 1.00 & 0.73 & 0.80 & 0.76 \\ 0.82 & 0.73 & 1.00 & 0.77 & 0.72 \\ 0.91 & 0.80 & 0.77 & 1.00 & 0.90 \\ 0.84 & 0.76 & 0.72 & 0.90 & 1.00 \end{pmatrix},$$

$$\boldsymbol{\sigma} = (\sigma_i)_{i=1}^{d} = \begin{pmatrix} 0.518 & 0.648 & 0.623 & 0.570 & 0.530 \end{pmatrix},$$

$$\boldsymbol{\omega} = (\omega_i)_{i=1}^{d} = \begin{pmatrix} 0.381 & 0.065 & 0.057 & 0.270 & 0.227 \end{pmatrix}.$$

The eigenvalues of the corresponding covariance matrix $\boldsymbol{\Sigma}$ are shown in Figure A.1 and given by

$$(\lambda_i)_{i=1}^{d} = \begin{pmatrix} 1.4089 & 0.1124 & 0.1006 & 0.0388 & 0.0213 \end{pmatrix}$$

Hence, λ_1 is clearly dominant.

Sets B and C are taken from Jain & Oosterlee [45] and have dimensions $d = 10$ and $d = 15$, respectively. Here $K = 40$, $T = 1$, $r = 0.06$ and $\rho_{ij} = 0.25$, $\sigma_i = 0.20$ and $\omega_i = 1/d$ whenever $1 \leq i \neq j \leq d$. Sets B and C have $\lambda_1 = 0.13$ and $\lambda_1 = 0.18$, respectively, and $\lambda_2 = \ldots = \lambda_d = 0.03$. Thus λ_1 is also dominant for these parameter sets.

Sets D, E, F have dimensions $d = 5, 10, 15$, respectively, where $K = 100$, $T = 1$, $r = 0.04$ and $\rho_{ij} = \exp(-\mu|i-j|)$, $\sigma_i = 0.30$ and $\omega_i = 1/d$ whenever $1 \leq i, j \leq d$ with $\mu = 0.0413$. The pertinent correlation structure has been considered in for example Reisinger & Wissmann [68] and leads to rapidly decreasing eigenvalues, as shown in Figure A.1. Sets D, E, F have in particular

$$\begin{aligned} (\lambda_1,\ \lambda_2,\ \lambda_3) =& (0.4218,\ 0.0180,\ 0.0053), \\ & (0.7897,\ 0.0647,\ 0.0187), \\ & (1.1126,\ 0.1337,\ 0.0402), \end{aligned}$$

respectively.

It can be shown that for all six Sets A–F the matrix of eigenvectors $\boldsymbol{Q}$ of $\boldsymbol{\Sigma}$ satisfies Assumption 1.

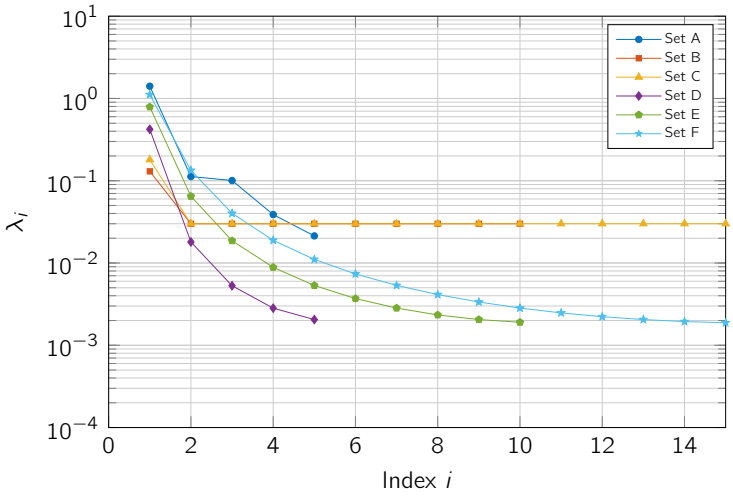


Figure A.1: Plot of eigenvalues λ_i of the covariance matrices $\mathbf{\Sigma}$ in Set A–F.

Bibliography

[1] D. Akoury, K. Kreidi, T. Jahnke, T. Weber, A. Staudte, M. Schoffler, N. Neumann, J. Titze, L. P. H. Schmidt, A. Czasch, et al., *The simplest double slit: interference and entanglement in double photoionization of H2*, Science **318** (2007), no. 5852, 949–952. [Page 109]

[2] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, Journal of computational physics **114** (1994), no. 2, 185–200. [Pages 111, 114, and 182]

[3] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, Journal of political economy **81** (1973), no. 637, 59. [Pages 3, 7, 9, and 10]

[4] J. Briggs and V. Schmidt, *Differential cross sections for photo-double-ionization of the helium atom*, Journal of Physics B: Atomic, Molecular and Optical Physics **33** (2000), no. 1, R1. [Page 111]

[5] M. Broadie and P. Glasserman, *Estimating security price derivatives using simulation*, Management science **42** (1996), no. 2, 269–285. [Pages 7, 17, 20, 77, 78, 81, and 89]

[6] H.-J. Bungartz and M. Griebel, *Sparse grids*, Acta numerica **13** (2004), 147–269. [Page 24]

[7] P. Carr and D. B. Madan, *Option valuation using the fast Fourier transform*, J. Comp. Finan. **2** (1999), 61–73. [Page 46]

[8] J. D. Carroll and J.-J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition*, Psychometrika **35** (1970), no. 3, 283–319. [Page 101]

[9] X. Chen, G. Deelstra, J. Dhaene, D. Linders, and M. Vanmaele, *On an optimization problem related to static super-replicating strategies*, J. Comp. Appl. Math. **278** (2015), 213–230. [Pages 62, 65, and 74]

[10] X. Chen, G. Deelstra, J. Dhaene, and M. Vanmaele, *Static super-replicating strategies for a class of exotic options*, Insur. Math. Econ. **42** (2008), 1067–1085. [Pages 62, 65, and 74]

[11] W. C. Chew and W. H. Weedon, *A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates*, Microwave and optical technology letters **7** (1994), no. 13, 599–604. [Page 114]

[12] S. Cools and W. Vanroose, *A fast and robust computational method for the ionization cross sections of the driven Schrödinger equation using an $\mathcal{O}(n)$ multigrid-based scheme*, Journal of Computational Physics **308** (2016), 20–39. [Pages 111 and 116]

[13] B. Datta, *Numerical methods for linear control systems*, vol. 1, Academic Press, 2004. [Page 172]

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM journal on Matrix Analysis and Applications **21** (2000), no. 4, 1253–1278. [Pages 93 and 104]

[15] G. Deelstra, I. Diallo, and M. Vanmaele, *Bounds for Asian basket options*, J. Comp. Appl. Math. **218** (2008), 215–228. [Pages 62, 65, and 74]

[16] G. Deelstra, J. Liinev, and M. Vanmaele, *Pricing of arithmetic basket options by conditioning*, Insur. Math. Econ. **34** (2004), 55–77. [Pages 62, 65, 66, and 74]

[17] J. Dhaene, M. Denuit, M. Goovaerts, R. Kaas, and D. Vyncke, *The concept of comonotonicity in actuarial science and finance: applications*, Insur. Math. Econ. **31** (2002), 133–161. [Pages 61, 65, and 74]

[18] ———, *The concept of comonotonicity in actuarial science and finance: theory*, Insur. Math. Econ. **31** (2002), 3–33. [Pages 61, 65, and 74]

[19] S. V. Dolgov and D. V. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM Journal on Scientific Computing **36** (2014), no. 5, A2248–A2271. [Page 110]

[20] F. Fang and C. W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comp. **31** (2008), 826–848. [Page 46]

[21] A. Gaul and N. Schlömer, *Preconditioned recycling Krylov subspace methods for self-adjoint problems*, arXiv preprint arXiv:1208.0264 (2012). [Page 125]

[22] M. B. Giles, *Multilevel Monte Carlo path simulation*, Operations research **56** (2008), no. 3, 607–617. [Page 17]

[23] ———, *Multilevel Monte Carlo methods*, Acta numerica **24** (2015), 259–328. [Page 17]

[24] E. Gobet, *Revisiting the Greeks for European and American options*, Stochastic processes and applications to mathematical finance, World Scientific, 2004, pp. 53–71. [Pages 19, 20, and 85]

[25] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 2013. [Pages 96, 151, and 162]

[26] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen **36** (2013), no. 1, 53–78. [Page 110]

[27] W. Hackbusch, *Tensor spaces and numerical tensor calculus*, vol. 42, Springer, 2012. [Page 110]

[28] T. Haentjens and K. in 't Hout, *ADI schemes for pricing American options under the Heston model*, Appl. Math. Fin. **22** (2015), 207–237. [Pages 64 and 74]

[29] H. Hanbali and D. Linders, *American-type basket option pricing: a simple two-dimensional partial differential equation*, Quant. Fin. **19** (2019), 1689–1704. [Pages 62, 65, 66, 67, 72, 74, and 85]

[30] R. A. Harshman et al., *Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis*, (1970). [Page 101]

[31] D. J. Higham, *An introduction to financial option valuation: mathematics, stochastics and computation*, (2004). [Pages 9 and 16]

[32] S. Holtz, T. Rohwedder, and R. Schneider, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM Journal on Scientific Computing **34** (2012), no. 2, A683–A713. [Page 110]

[33] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge university press, 2012. [Page 15]

[34] T. Huckle, K. Waldherr, and T. Schulte-Herbrüggen, *Computations in quantum tensor networks*, Linear Algebra and its Applications **438** (2013), no. 2, 750–781. [Page 110]

[35] J. C. Hull, *Options futures and other derivatives*, Pearson Education India, 2003. [Pages 9 and 10]

[36] W. Hundsdorfer and J. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer, 2003. [Pages 11, 37, 38, 39, and 156]

[37] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Appl. Math. Lett. **17** (2004), 809–814. [Pages 64 and 74]

[38] ———, *Operator splitting methods for pricing American options under stochastic volatility*, Numer. Math. **113** (2009), 299–324. [Pages 64, 67, and 74]

[39] K. J. in 't Hout and J. Snoeijer, *Numerical valuation of American basket options via partial differential complementarity problems.*, Mathematics **9** (2021), no. 13, 1498. [Pages 25, 26, 61, 62, and 78]

[40] K. in 't Hout, *Numerical partial differential equations in finance explained*, Financial Engineering Explained, Palgrave Macmillan UK, 2017. [Pages 34, 35, 36, 50, 65, and 67]

[41] K. in 't Hout and J. Snoeijer, *Numerical valuation of Bermudan basket options via partial differential equations.*, Int. J. Comp. Math. **98** (2021), 829–844. [Pages 25, 26, 30, 36, 45, 72, and 78]

[42] K. in 't Hout and R. Valkov, *Numerical study of splitting methods for American option valuation*, Novel Methods in Computational Finance, Springer, 2017, pp. 373–398. [Pages 65 and 67]

[43] K. in 't Hout and K. Volders, *Stability of central finite difference schemes on non-uniform grids for the Black–Scholes equation.*, Appl. Numer. Math. **59** (2009), 2593–2609. [Page 37]

[44] K. Itô, *On stochastic differential equations*, no. 4, American Mathematical Soc., 1951. [Page 10]

[45] S. Jain and C. Oosterlee, *The stochastic grid bundling method: efficient pricing of Bermudan options and their Greeks*, Appl. Math. Comp. **269** (2015), 412–431. [Pages 46, 51, and 191]

[46] R. Kaas, J. Dhaene, and M. Goovaerts, *Upper and lower bounds for sums of random variables*, Insur. Math. Econ. **27** (2000), 151–168. [Pages 61, 65, 66, and 74]

[47] O. Koch and C. Lubich, *Dynamical low-rank approximation*, SIAM Journal on Matrix Analysis and Applications **29** (2007), no. 2, 434–454. [Pages iii, 3, 149, 150, 151, 152, and 153]

[48] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM review **51** (2009), no. 3, 455–500. [Pages 93, 100, 101, 104, 110, 127, and 136]

[49] T. G. Kolda, *Multilinear operators for higher-order decompositions.*, Tech. report, Sandia National Laboratories, 2006. [Pages 93 and 98]

[50] H.-O. Kreiss, V. Thomée, and O. Widlund, *Smoothing of initial data and rates of convergence for parabolic difference equations*, Communications on Pure and Applied Mathematics **23** (1970), no. 2, 241–259. [Pages 11, 35, and 50]

[51] C. Leentvaar and C. Oosterlee, *On coordinate transformation and grid stretching for sparse grid pricing of basket options*, Journal of Computational and Applied Mathematics **222** (2008), no. 1, 193–209, Special Issue: Numerical PDE Methods in Finance. [Page 24]

[52] F. A. Longstaff and E. S. Schwartz, *Valuing american options by simulation: a simple least-squares approach*, The review of financial studies **14** (2001), no. 1, 113–147. [Pages xiii, 7, 19, 20, and 85]

[53] C. Lubich, *Time integration in the multiconfiguration time-dependent hartree method of molecular quantum dynamics*, Applied Mathematics Research eXpress **2015** (2015), no. 2, 311–328. [Page 150]

[54] C. Lubich and I. V. Oseledets, *A projector-splitting integrator for dynamical low-rank approximation*, BIT Numerical Mathematics **54** (2014), no. 1, 171–188. [Pages 150, 151, 153, 155, and 157]

[55] C. Lubich, B. Vandereycken, and H. Walach, *Time integration of rank-constrained tucker tensors*, SIAM Journal on Numerical Analysis **56** (2018), no. 3, 1273–1290. [Page 186]

[56] C. W. McCurdy, M. Baertschy, and T. Rescigno, *Solving the three-body Coulomb breakup problem using exterior complex scaling*, Journal of Physics B: Atomic, Molecular and Optical Physics **37** (2004), no. 17, R137. [Pages 111, 119, and 126]

[57] R. C. Merton, *Theory of rational option pricing*, The Bell Journal of economics and management science (1973), 141–183. [Page 9]

[58] V. Murg, F. Verstraete, Ö. Legeza, and R. M. Noack, *Simulating strongly correlated quantum systems with tree tensor networks*, Physical Review B **82** (2010), no. 20, 205105. [Page 110]

[59] J. Neumann, *Functional operators vol. ii. the geometry of orthogonal spaces. annals of mathematical studies# 22*, (1950). [Page 125]

[60] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer, 2006. [Pages 159 and 160]

[61] A. Nonnenmacher and C. Lubich, *Dynamical low-rank approximation: applications and numerical experiments*, Mathematics and Computers in Simulation **79** (2008), no. 4, 1346–1357. [Page 150]

[62] C. W. Oosterlee and L. A. Grzelak, *Mathematical modeling and computation in finance*, World Scientific, 2019. [Pages 13 and 26]

[63] I. Oseledets, *DMRG approach to fast linear algebra in the TT-format*, Computational Methods in Applied Mathematics **11** (2011), no. 3, 382–393. [Page 110]

[64] I. V. Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing **33** (2011), no. 5, 2295–2317. [Pages 105, 106, 146, and 186]

[65] R. Poet, *The exact solution for a simplified model of electron scattering by hydrogen atoms*, Journal of Physics B: Atomic and Molecular Physics **11** (1978), no. 17, 3081. [Page 115]

[66] D. M. Pooley, K. R. Vetzal, and P. A. Forsyth, *Convergence remedies for non-smooth payoffs in option pricing*, Journal of Computational Finance **6** (2003), no. 4, 25–40. [Pages 11, 35, and 50]

[67] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numerische Mathematik **43** (1984), no. 2, 309–327. [Pages 12, 36, and 50]

[68] C. Reisinger and R. Wissmann, *Numerical valuation of derivatives in high-dimensional settings via partial differential equation expansions*, J. Comp. Finan. **18** (2015), no. 4, 95–127. [Pages 26, 46, 78, and 191]

[69] ———, *Error analysis of truncated expansion solutions to high-dimensional parabolic PDEs*, ESAIM: M2AN **51** (2017), no. 6, 2435–2463. [Pages 26, 34, 46, 57, and 78]

[70] ———, *Finite difference methods for medium- and high-dimensional derivative pricing PDEs*, High-Performance Computing in Finance: Problems, Methods, and Solutions, Chapman and Hall/CRC, London, 2018, pp. 175–196. [Pages 26, 46, and 78]

[71] C. Reisinger and G. Wittum, *Efficient hierarchical approximation of high-dimensional option pricing problems*, SIAM J. Sci. Comp. **29** (2007), no. 1, 440–458. [Pages i, 2, 13, 15, 24, 25, 26, 27, 29, 32, 40, 43, 45, 46, 57, 74, 77, 78, 89, and 191]

[72] T. Rescigno, M. Baertschy, W. Isaacs, and C. McCurdy, *Collisional breakup in a quantum system of three charged particles*, Science **286** (1999), no. 5449, 2474–2479. [Pages 111 and 115]

[73] T. Rescigno and C. McCurdy, *Numerical grid methods for quantum-mechanical scattering problems*, Physical review A **62** (2000), no. 3, 032706. [Pages 111, 113, and 143]

[74] H. Schmidt-Böcking, R. Dörner, and J. Ullrich, *Coltrims*, Europhysics News **33** (2002), no. 6, 210–211. [Page 109]

[75] S. E. Shreve, *Stochastic calculus for finance ii: Continuous-time models*, vol. 11, Springer, 2004. [Page 19]

[76] B. Simon, *The definition of molecular resonance curves by the method of exterior complex scaling*, Physics Letters A **71** (1979), no. 2-3, 211–214. [Pages 111, 113, and 182]

[77] J. Sirignano and K. Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, J. Comp. Phys. **375** (2018), 1339–1364. [Page 46]

[78] A. Smilde, P. Geladi, and R. Bro, *Multi-way analysis: applications in the chemical sciences*, John Wiley & Sons, 2005. [Pages 95 and 96]

[79] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis, *Splatt: Efficient and parallel sparse tensor-matrix multiplication*, 2015 IEEE International Parallel and Distributed Processing Symposium, IEEE, 2015, pp. 61–70. [Page 102]

[80] J. Snoeijer and W. Vanroose, *Solving for the low-rank tensor components of a scattering wave function.* [Pages 109, 149, 185, and 186]

[81] M. M. Steinlechner, *Riemannian optimization for solving high-dimensional problems with low-rank tensor structure*, Tech. report, EPFL, 2016. [Page 106]

[82] W. A. Strauss, *Partial differential equations: An introduction*, John Wiley & Sons, 2007. [Pages 149 and 150]

[83] Z. L. Streeter, F. L. Yip, R. R. Lucchese, B. Gervais, T. N. Rescigno, and C. W. McCurdy, *Dissociation dynamics of the water dication following one-photon double ionization. i. Theory*, Physical Review A **98** (2018), no. 5, 053429. [Page 111]

[84] D. Tavella and C. Randall, *Pricing financial instruments*, John Wiley & Sons, 2000. [Pages 13 and 26]

[85] A. Temkin, *Nonadiabatic theory of electron-hydrogen scattering*, Physical Review **126** (1962), no. 1, 130. [Page 115]

[86] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika **31** (1966), no. 3, 279–311. [Pages 104 and 126]

[87] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, *A new truncation strategy for the higher-order singular value decomposition*, SIAM Journal on Scientific Computing **34** (2012), no. 2, A1027–A1052. [Page 104]

[88] W. Vanroose, D. A. Horner, F. Martin, T. N. Rescigno, and C. W. McCurdy, *Double photoionization of aligned molecular hydrogen*, Physical Review A **74** (2006), no. 5, 052702. [Page 111]

[89] W. Vanroose, F. Martin, T. N. Rescigno, and C. W. McCurdy, *Complete photo-induced breakup of the H2 molecule as a probe of molecular electron correlation*, Science **310** (2005), no. 5755, 1787–1789. [Page 111]

[90] K. Volders, *Stability of central finite difference schemes on non-uniform grids for 1D partial differential equations with variable coefficients*, AIP Conference Proceedings, vol. 1281, 2010, pp. 1991–1994. [Page 37]

[91] D. Vyncke, M. Goovaerts, and J. Dhaene, *An accurate analytical approximation for the price of a European-style arithmetic Asian option*, Finan. **25** (2004), 121–139. [Pages 61, 65, 66, and 74]

[92] P. Wang, *Computational efficiency and accuracy in the valuation of basket options*, Frontiers in Finance and Economics **6** (2009), no. 1, 1–25. [Page 22]

[93] J. Xu and L. Zikatanov, *The method of alternating projections and the method of subspace corrections in Hilbert space*, Journal of the American Mathematical Society **15** (2002), no. 3, 573–597. [Page 125]

Scientific Résumé

Education

2016 – 2022	**PhD candidate in Mathematics** University of Antwerp, Belgium
2014 – 2016	**Master of Science: Applied Mathematics** Delft University of Technology, The Netherlands *Computational Science and Engineering*
2014 – 2016	**Master of Science: Computational Engineering** Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany *System simulation*
2012 – 2014	**Bachelor of Science: Mathematics** University of Groningen, The Netherlands
2011 – 2014	**Bachelor of Science: Computing Science** University of Groningen, The Netherlands

Teaching assistant

Computer practicum
Bachelor of Mathematics / Bachelor of Physics
Academic year 2016/2017, 2017/2018, 2018/2019, 2019/2020, 2020/2021, 2021/2022

Eindige differentie methoden en financiële wiskunde
Master of Mathematics / Master of Physics
Academic year 2017/18, 2018/19, 2019/20, 2020/21 (PC labs), 2021/22 (PC labs)

Toepassingen van differentiaalvergelijkingen
Master of Mathematics / Master of Physics
Academic year 2020/2021 (PC labs), 2021/2022 (PC labs)

Iteratieve methodes voor lineaire stelsels en eigenwaarden problemen
Bachelor of Mathematics
Academic year 2016/2017, 2018/2019

Numeriek oplossen van gewone differentiaalvergelijkingen
Bachelor of Mathematics
Academic year 2016/2017, 2017/2018, 2018/2019, 2019/2020, 2020/2021, 2021/2022

Numerieke analyse
Bachelor of Mathematics / Bachelor of Computer Science
Academic year 2016/2017, 2017/2018, 2018/2019, 2019/2020, 2021/2022 (PC labs)

Toegepaste lineaire algebra
Bachelor of Chemistry / Bachelor of Biochemistry and Biotechnology
Academic year 2019/2020, 2020/2021, 2021/2022

Co-promotor or jury member master theses
Master of Mathematics

Publications

Karel J. in 't Hout and Jacob Snoeijer, *Numerical valuation of Bermudan basket options via partial differential equations.*, Int. J. Comp. Math. **98** (2021), 829–844.

Karel J. in 't Hout and Jacob Snoeijer, *Numerical valuation of American basket options via partial differential complementarity problems.*, Mathematics **9** (2021), no. 13, 1498.

Jacob Snoeijer and Wim Vanroose, *Solving for the low-rank tensor components of a scattering wave function.*, Under review.

Conference talks and poster presentations

[**ICCF 2022**] *4th International Conference on Computational Finance 2022*
Wuppertal, Germany, June 6-10, 2022 (travel grant)
Title of talk: *Numerical valuation of American basket options via partial differential complementarity problems*.

[**SPRING 2022**] *Annual Spring Meeting of the Dutch-Flemish Scientific Computing Society*
Leuven, Belgium, May 6, 2022
Title of talk: *Solving for the low-rank tensor components of the wave function in scattering problems with multiple ionization*.

[**WSC 2021**] *45th Woudschoten Conference of the Dutch-Flemish Scientific Computing Society*
Zeist, The Netherlands, October 6-8, 2021 (third price)
Poster presentation: *Solving for the low-rank tensor components of the wave function in scattering problems with multiple ionization*.

[**ANLA 2021**] *GAMM Workshop Applied and Numerical Linear Algebra 2021*
Potsdam, Potsdam, September 16-17, 2021
Title of talk: *Solving for the low-rank tensor components of the wave function in scattering problems with multiple ionization*.

[**ICCF 2019**] *3rd International Conference on Computational Finance 2019*
A Coruña, Spain, July 8-12, 2019
Title of talk: *Efficient numerical valuation of high-dimensional basket options via partial differential equations*.

[**WSC 2019**] *44th Woudschoten Conference of the Dutch-Flemish Scientific Computing Society*
Zeist, The Netherlands, October 9-11, 2019
Poster presentation: *Low rank approximations to diffusion dominated partial differential equations*.

[**ICCF 2019**] *3rd International Conference on Computational Finance 2019*
A Coruña, Spain, July 8-12, 2019
Title of talk: *Efficient numerical valuation of high-dimensional basket options via partial differential equations*.

[**G2S3 2019**] *10th Gene Golub SIAM Summer School: High performance data analytics*
Aussois, France, June 17-30, 2019 (selected to participate)
Poster presentation: *Low rank approximations to diffusion dominated partial differential equations*.

[**Math R-Day 2019**] *First Mathematics Research Day*
Antwerp, Belgium, May 15, 2019
Title of talk: *Efficiënte numerieke benaderingen voor eerlijke prijzen voor hoog-dimensionele basket opties*.

[**WSC 2018**] *43th Woudschoten Conference of the Dutch-Flemish Scientific Computing Society*
Zeist, The Netherlands, October 3-5, 2018
Poster presentation: *Numerical valuation of high dimensional Bermudan basket options via partial differential equations*.

[**NUMDIFF 2018**] *15th Conference on the Numerical Solution of Differential and Differential-Algebraic Equations*
Halle, Germany, September 3-7, 2018
Title of talk: *Numerical valuation of Bermudan basket options via partial differential equations*.

[**BMS 2018**] *Belgian Mathematical Society PhD-day*
Ghent, Belgium, May 25, 2018
Poster presentation: *Efficient numerical pricing of basket options via partial differential equations*.

Scientific services

2019 – 2022	**Member of Departementsraad Wiskunde** University of Antwerp
2017 – 2022	**Member of Onderwijscommissie Wiskunde** University of Antwerp
2019	**Co-organizer of Mathematics Research Day at University of Antwerp** First research day for third year bachelor and master students
2017 – 2019	**Co-organizer of Wisku$\mathbb{N}$nde In-$\mathbb{Z}$icht at University of Antwerp** Math workshop for last-year secondary school