CONTENTS

List of Figures xiii

List of Tables xv

List of Algorithms xix

List of Symbols xxi

List of Acronyms xxiii

1. Introduction 1
- 1.1. Motivation . . . 1
- 1.2. Outline of the Thesis . . . 3
- 1.3. Target Computer Architectures . . . 6
- 1.4. Related own Publications . . . 7

2. Mathematical Basics of the Generalized Eigenvalue Problem 9
- 2.1. Invariant Subspaces and Deflation . . . 10
- 2.2. Generalized Schur Decomposition . . . 12
- 2.3. Reordering of Eigenvalues . . . 14
- 2.4. Transformation of Eigenvalues . . . 16
  - 2.4.1. Reciprocal Function . . . 17
  - 2.4.2. Scale Function . . . 17
  - 2.4.3. Shift Function . . . 18
  - 2.4.4. Cayley Transformation . . . 18

3. Spectral Divide and Conquer Algorithms 21
- 3.1. Basic Concept . . . 23
  - 3.1.1. Construction of Q and Z from a Deflating Subspace . . . 24
  - 3.1.2. Decoupling Residual . . . 27
  - 3.1.3. Infinite Eigenvalues . . . 28
- 3.2. Deflating Subspaces via the Matrix Sign Function . . . 30
  - 3.2.1. Generalized Matrix Sign Function . . . 32
  - 3.2.2. Spectral Splitting . . . 33
  - 3.2.3. The Divide-Shift-and-Conquer Algorithm . . . 35
- 3.3. Deflating Subspaces via the Matrix Disc Function . . . 40
  - 3.3.1. Inverse Free Iteration . . . 44
  - 3.3.2. The Divide-Scale-and-Conquer Algorithm . . . 46

4. Implementation on Current Hardware **51**
4.1. Divide-Shift/Scale-and-Conquer Algorithms . . . 51
4.1.1. Recursion Free Variant . . . 53
4.1.2. Level-Set Traversing of the Recursion Tree . . . 57
4.1.3. Parallelization of the Recursion Tree . . . 60
4.1.4. Handling Trivial Problems . . . 61
4.1.5. Computation of Deflating Subspaces and Common Operations . . . 63
4.2. Computation of the Generalized Matrix Sign Function . . . 67
4.2.1. Naive Implementation . . . 68
4.2.2. Reducing the Numerical Effort using One-Sided Transformations . . . 69
4.2.3. Reducing the Numerical Effort using Two-Sided Transformations . . . 76
4.2.4. Gauss-Jordan Elimination and GPU acceleration . . . 79
4.2.5. Integrating Gauss-Jordan Elimination and Generalized Matrix Sign Function 90
4.2.6. Summary . . . 90
4.3. Computation of the Matrix Disc Function . . . 91
4.3.1. CPU Implementation . . . 92
4.3.2. CPU Implementation with Directed Acyclic Graph Scheduling . . . 94
4.3.3. Block Upper Triangularization using Generalized Householder Transformations . . . 101
4.3.4. GPU Implementation with Storage Efficient QR Decomposition . . . 104

5. Numerical Experiments for the Eigenvalue Problem **115**
5.1. Hardware and Software Setup . . . 115
5.1.1. Compute Servers . . . 116
5.1.2. Implementation . . . 117
5.1.3. Reference Results and Data Sources . . . 118
5.2. Performance of the Building Blocks . . . 119
5.2.1. Gauss-Jordan Elimination . . . 119
5.2.2. Generalized Matrix Sign Function . . . 124
5.2.3. QR Decompositions . . . 126
5.2.4. Inverse Free Iteration . . . 129
5.3. Computing the Generalized Schur Decomposition . . . 132
5.3.1. Divide-Shift-and-Conquer . . . 134
5.3.2. Divide-Scale-and-Conquer . . . 144

6. Application to Sylvester-Type Matrix Equations **151**
6.1. Mathematical Basics . . . 152
6.2. Classic Direct Solution Techniques . . . 154
6.2.1. Bartels-Stewart Algorithm . . . 155
6.2.2. One- and Two-Solve Schemes . . . 156
6.2.3. Recursive Blocking . . . 159
6.2.4. Matrix Sign Function Solvers . . . 161
6.3. Block Algorithms and their Implementation . . . 163
6.3.1. Improving the level-3 Implementation . . . 167
6.3.2. Efficient Solution of the small Subproblems . . . 169
6.3.3. Explicit Parallelization using OpenMP . . . 175
6.4. Triangular Solvers and the Approximated Schur Decomposition . . . 177

7. Numerical Experiments for the Matrix Equations **181**
7.1. Hardware and Software Setup 181
7.2. Performance of Triangular Solvers 182
7.3. Solution of Generalized Lyapunov Equations 190

8. Conclusions **195**

A. Explanatory Experiments **201**
A.1. Examples in Chapter 4 201
A.2. Examples in Chapter 6 203

Bibliography **205**