Contents

List of Figures **xi**

List of Tables **xv**

List of Algorithms **xvii**

1 Introduction **1**
1.1 Background and motivation 1
1.2 Objectives 4
1.3 Structure of the document 4

2 High-performance computing **7**
2.1 Hardware evolution 8
2.2 Parallel architectures 11
2.2.1 Central control mechanism 11
2.2.2 Communication mechanism 12
2.3 Programming models and parallel software 13
2.3.1 Threads and OpenMP 13
2.3.2 Message Passing Interface 14
2.3.3 Software 17
2.4 Hardware accelerators 21
2.4.1 Integrated circuits 22
2.4.2 Manycore processors 22
2.4.3 Graphics processing units 23
2.5 Performance indicators 33
2.5.1 Execution time 33
2.5.2 Speedup 34
2.5.3 Efficiency 35
2.5.4 Scalability 35
2.5.5 FLOPs/s 35

3 Eigenvalue problems **37**
3.1 Methods to compute eigenvalues . . . 38
3.1.1 Direct methods . . . 38
3.1.2 Iterative methods . . . 39
3.2 Krylov methods for eigenvalue problems . . . 41
3.2.1 Krylov-Schur . . . 43
3.3 Scientific computing software . . . 44
3.3.1 PETSc . . . 46
3.3.2 SLEPc . . . 50
3.4 Solving eigenproblems with GPUs . . . 53
3.4.1 The PARISO code . . . 55
3.4.2 Optimization of spin eigenanalysis . . . 56
3.4.3 Acceleration with graphics processors . . . 62
3.5 Conclusions . . . 68

4 Block-tridiagonal eigenvalue problems **71**
4.1 Matrix-vector product . . . 73
4.2 Shift and invert: Linear systems . . . 73
4.2.1 Thomas algorithm . . . 73
4.2.2 Block cyclic reduction . . . 75
4.2.3 Spike . . . 77
4.3 Block cyclic tridiagonal structures . . . 80
4.3.1 Schur complement . . . 80
4.4 Parallel implementations . . . 81
4.4.1 Matrix-vector product . . . 81
4.4.2 Direct linear solvers . . . 84
4.5 Numerical experiments . . . 92
4.5.1 Single process executions . . . 93
4.5.2 Multi-process executions . . . 98
4.6 Conclusions . . . 109

5 Matrix functions **113**
5.1 Dense methods for matrix functions . . . 114
5.1.1 Square root . . . 115
5.1.2 Sign . . . 118
5.1.3 Exponential . . . 118
5.2 Krylov methods for matrix functions . . . 119
5.2.1 Restarted Arnoldi . . . 120
5.3 Numerical experiments . . . 121
5.3.1 Computational evaluation of dense solvers . . . 121
5.3.2 Computational evaluation of sparse solvers . . . 127
5.4 Conclusions . . . 134

6 Conclusions **137**

Bibliography **141**