The main contribution of this thesis is to demonstrate that graphics processors (GPUs) as representatives of emerging many-core architectures are very well-suited for the fast and accurate solution of large sparse linear systems of equations, using parallel multigrid methods on heterogeneous compute clusters. Such systems arise for instance in the discretisation of (elliptic) partial differential equations with finite elements. We report on at least one order of magnitude speedup over highly-tuned conventional CPU implementations, without sacrificing neither accuracy nor functionality. In more detail, this thesis includes the following contributions:

Single precision floating point computations may be insufficient for the class of problems considered in this thesis. We revisit mixed precision iterative refinement techniques to not only increase the accuracy of computed results, but also to increase the efficiency of the solution process. Both on CPUs and on GPUs, we demonstrate a significant performance improvement without loss of accuracy compared to computing in high precision only.

We present efficient parallelisation techniques for multigrid solvers on graphics hardware, in particular for numerically strong smoothers and preconditioners that are suitable for highly anisotropic grids and operators. For instance, an efficient formulation of the cyclic reduction algorithm to solve tridiagonal systems is developed. In view of hardware-oriented numerics, we carefully analyse the trade-off between numerical and runtime performance for inexact parallelisation techniques that decouple some of the inherently sequential characteristics of strong smoothing operators.

For large-scale established software frameworks, the re-implementation tailored to novel hardware platforms is often prohibitively expensive. We develop a 'minimally invasive' approach to integrate support for co-processor hardware like GPUs into FEAST, a finite element discretisation and solver toolbox. Our technique has the major advantage that applications built on top of the toolbox do not have to be changed at all to benefit from co-processor acceleration. The approach is evaluated for benchmark problems in linearised elasticity and stationary laminar flow computed on large-scale GPU-enhanced clusters. Good speedup factors and near-ideal weak scalability are observed. The achievable speedup is analysed and a theoretical speedup model is presented.

Finally, we provide a historical overview of scientific computing on graphics hardware since the early beginnings in 2001/2002, when GPGPU was an obscure research topic pursued by few, to the widespread adoption nowadays. We discuss the evolution of the hardware and the programming model, and provide a comprehensive bibliography of publications related to PDE simulations on GPUs.

Keywords:
Scientific Computing
Hardware-Oriented Numerics
Finite Elements
Multigrid Solvers
Krylov Subspace Solvers
Iterative Solvers
Tridiagonal Solvers
Cyclic Reduction
Mixed Precision Iterative Refinement
Double-Single Precision Emulation
Ill-Conditioned Linear Systems
Large Sparse Linear Systems
Locally Structured Systems

Poisson Problem
Stationary Laminar Flow
Computational Fluid Dynamics
Linearised Elasticity
Computational Solid Mechanics

High-Performance Computing
Heterogeneous Computing
Hybrid Computing
Graphics Processors (GPUs)
GPGPU
GPU Computing
NVIDIA CUDA

Domain Decomposition
Parallel Computing
Fine-Grained Parallelism
Multicolour Parallelisation