7th AMM Advanced solvers for modern architectures: Program

Updated: Tuesday 12th November, 2019 13:42

Monday	
11:00 - 11:30	Registration
11:30 - 11:45	Welcome ceremony
11:45 - 12:35	**Eike Müller**
12:35 - 14:00	Lunch
14:00 - 14:30	**Harald Köstler - ExaStencils**
14:30 - 15:00	**Mirco Altenbernd - EXADUNE**
15:00 - 15:30	**Johannes Rentrop - EXAHD**
15:30 - 16:15	Coffee break
16:15 - 17:05	**Laura Grigori**
17:05 - 17:35	**Nils-Arne Dreier - EXADUNE**
19:00	Conference dinner

Tuesday	
9:30 - 10:20	**Erin Carson**
10:20 - 10:50	**Oliver Rheinbach - EXASTEEL**
10:50 - 11:20	Coffee break
11:20 - 11:50	**Nils Kohl - TERRA-NEO**
11:50 - 12:20	**Dominik Ernst - ESSEX-II**
12:20 - 14:00	Lunch
14:00 - 14:50	**Andreas Frommer**
14:50 - 15:20	**Bruno Lang - ESSEX-II**
15:20 - 16:00	Coffee break
16:00 - 16:50	**Stefano Zampini**
16:50 - 17:40	**Andrew Barker**

Wednesday	
9:30 - 10:20	**Nicole Spillane**
10:20 - 10:50	**Matthias Bolten - ExaStencils**
10:50 - 11:20	Coffee break
11:20 - 12:10	**Wim Vanroose**
12:10 - 12:30	Closing Remarks
12:30 -	Lunch

Fast semi-implicit DG solvers for fluid dynamics: hybridization and multigrid preconditioners

Eike Müller, *University of Bath*

For problems in Numerical Weather Prediction, time to solution is critical. Semi-implicit time-stepping methods can speed up geophysical fluid dynamics simulations by taking larger model time-steps than explicit methods. This is possible since semi-implicit integrators treat the fast (but physically less important) waves implicitly. As a consequence, the time-step size is not restricted by an overly tight CFL condition. A disadvantage of this approach is that a large system of equations has to be solved repeatedly at every time step. However, using an suitably preconditioned iterative method significantly reduces the computational cost of this solve, potentially making a semi- implicit scheme faster overall.

A good spatial discretisation is equally important. Higher-order Discontinuous Galerkin (DG) methods are known for having high arithmetic intensity and can be parallelised very efficiently, which makes them well suited for modern HPC hardware. Unfortunately, the arising linear system in semi-implicit timestepping is difficult to precondition since the numerical flux introduces off diagonal artificial diffusion terms. Those terms prevent the traditional reduction to a Schur-complement pressure equation. This issue can be avoided by using a hybridised DG discretisation, which introduces additional flux-unknowns on the facets of the grid and results in a sparse elliptic Schur-complement problem. Recently Kang, Giraldo and Bui-Thanh [1] solved the resultant linear system with a direct method. However, since the cost grows with the third power of the number of unknowns, this becomes impractical for high resolution simulations.

We show how this issue can be overcome by construcing a non-nested geometric multigrid preconditioner similar to [2] instead. We demonstrate the effectiveness of the multigrid method for the non- linear shallow water equations, an important model system in geophysical fluid dynamics. With our solvers semi-implicit IMEX time- steppers become competitive with standard explicit Runge Kutta methods. Hybridisation and reduction to the Schur-complement system is implemented in the Slate language [3], which is part of the Firedrake Python framework for solving finite element problems via code generation.

[1] Kang, Giraldo, Bui-Thanh (2019): "IMEX HDG-DG: a coupled implicit hybridized discontinuous Galerkin (HDG) and explicit discontinuous Galerkin (DG) approach for shallow water systems" Journal of Computational Physics, 109010, arXiv:1711.02751

[2] Cockburn, Dubois, Gopalakrishnan, Tan (2014): "Multigrid for an HDG method", IMA Journal of Numerical Analysis 34(4):1386-1425

[3] Gibson, Mitchell, Ham, Cotter, (2018): "A domain-specific language for the hybridization and static condensation of finite element methods." arXiv preprint arXiv:1802.00303.

Code generation for HPC

Harald Köstler - ExaStencils, *Friedrich-Alexander-Universität Erlangen-Nürnberg*

In recent years the principle of separation of concerns has increasingly been investigated for simulation software. Applications scientists want to be able to easily run the software for different input data and configurations, but they also want to include new physical models or modify existing ones. Additionally, they sometimes have also experience which discretizations or numerical algorithms. Framework developers are familiar with modern software design and low-level architecture-specific optimizations instead. From the computer science perspective, this separation of concerns can be supported by providing interfaces with different levels of abstraction for different roles. Code generation technology can then be used as a tool to automatically map the abstract descriptions to concrete code and thus to improve the development of simulation software. After intense research in the last years, code generation is now mature enough to find its way into real-world applications and existing software frameworks and to offer embedded or external domain-specific languages (DSL) as an interface to users. We will present two examples for successful use of code generation technology, the external DSL ExaStencils, and the embedded DSL pystencils. Applications include non-Newtonian fluids, Ocean simulation, and material sciences.

Checkpoint/Restart for Iterative Solvers utilising Lossy Compression

Mirco Altenbernd - EXADUNE, *University of Stuttgart*

Fault-tolerance is one of the major challenges of extreme-scale computing. There is broad consensus that future leadership-class machines will exhibit a substantially reduced mean-time-between-failure (MTBF) compared to today's systems because of the expected increase in the number of components without commensurately improving the reliability per component. The resilience challenge at scale is best summarized as, faults and failures are likely to become the norm rather than the exception: Any simulation run will be compromised without inclusion of resilience techniques into the underlying software stack and system. Therefore we investigate recovery approaches for partially lost data in iterative solvers using lossy compressed checkpoints and data reconstruction techniques.

Multigrid methods are optimal iterative solvers for elliptic problems and are widely used as preconditioners or even standalone solvers, their hierarchy is often readily available. This hierarchy provides a straightforward compression scheme by projecting the data to a coarser level. Apart from that, there exist more advanced compression techniques like SZ, a powerful compressor which is specially designed as a floating-point lossy data compressor. It has been already used to improve classical checkpointing approaches for the recovery of time-marching systems. SZ compression, unlike multigrid compression, allows users to prescribe accuracy targets and by this is more easily adaptable to the needs of the iterative solver. For both compression techniques, we evaluate their usability for restoring partially lost data, e.g. due to node-losses, of the iterative approximation during a linear solve. We compare different recovery approaches from simple value-wise replacement without post-processing up to solving local auxiliary problems and their efficiency. For the efficiency evaluation, we focus mainly on compression rate, numerical overhead and necessary communication. Furthermore, we investigate the checkpoint-frequency.

Massively Parallel High-dimensional Simulation with the Sparse Grid Combination Technique

Johannes Rentrop - EXAHD, *University of Bonn*

Solvers that use a regular full grid discretization suffer from the curse of dimensionality, i.e. from the exponential dependence of the number of degrees of freedom on the mesh width. In problem settings with a medium or high number of dimensions, this imposes a severe bottleneck for simulations even on the largest supercomputers. One method to alleviate this problem is the sparse grid combination technique. It is an extrapolation method to create a solution on a so called sparse grid. Here, the number of grid points are reduced in such a way that only $\mathcal{O}(h^{-1}(\log h^{-1})^{d-1})$ instead of $\mathcal{O}(h^{-d})$ grid points are needed. For functions from Sobolov spaces with dominating mixed smoothness, the asymptotic approximation error only decreases slightly, from $\mathcal{O}(h^2)$ to $\mathcal{O}(h^2(\log h^{-1})^{d-1})$ in case of piecewise linear approximation.

For the project EXAHD within the DFG priority program Software for Exascale Computing, we developed a massively parallel software framework for the combination technique, which makes use of both a coarse layer of parallelism offered by the combination technique as well as a fine layer of parallelism used by the solver it is coupled to. This allows us to scale better to the huge number of cores available on the next generation of supercomputers. Our method also allows an intrinsic algorithm-based approach to fault tolerance. This talk provides an overview of the theory behind and implementation of the framework as well as advancements made within the EXAHD project in applying it to the gyrokinetic plasma turbulence code GENE.

Robust linear solvers based on enlarged Krylov methods and multilevel preconditioners

Laura Grigori, *INRIA Paris*

This talk focuses on solving large sparse linear systems of equations arising from the discretization of PDEs with strongly heterogeneous coefficients by using preconditioned Krylov subspace solvers. It discusses two challenges that we need to address for achieving scalability and robustness in linear solvers. Scalability relies on reducing global synchronizations between processors, while also increasing the arithmetic intensity on one processor. Robustness relies on ensuring that the condition number of the preconditioned matrix is bounded. We will first focus on enlarged Krylov subspace methods, an approach that relies on building a larger Krylov subspace that captures the smallest eigenvalues/eigenvectors of the system matrix and leads to a faster convergence. On a parallel computer, it allows to reduce the communication while increasing the arithmetic intensity. We will then focus on multilevel domain decomposition methods. We present a recent multilevel additive Schwarz method that relies on building a hierarchy of robust coarse spaces that transfer spectral information from one level to the following. This leads to a preconditioner that guarantees that the condition number of the preconditioned matrix is bounded at every level of the hierarchy. Numerical results on large scale computers, in particular for linear systems arising from solving linear elasticity problems, will discuss the efficiency of the proposed methods.

This is a joint work with H. Al Daas, P. Jolivet, O. Tissot, and P. H. Tournier.

Strategies for vectorized Block Conjugate Gradient methods

Nils-Arne Dreier - EXADUNE, *University of Münster*

Block Krylov methods are used to improve the convergence rate while solving linear systems with multiple right hand sides. Moreover many of the challenges occurring in HPC are mitigated - they are reducing the communication overhead, are well-posed for explicit vectorization, and have a better utilization of the memory bandwidth, especially in matrix based code.

In this talk we present a generalization of the Block Conjugate Gradients method, which allows to balance the blocking overhead for a fixed number of right hand sides. To implement this methods we build up on the SIMD abstractions of the DUNE framework. Finally we present a performance analysis of the building blocks of the method which gives us an idea how to choose a good block size.

Accelerating the Solution of Linear Systems via Multiprecision Arithmetic

Erin Carson, *Charles University*

Support for floating point arithmetic in multiple precisions is becoming increasingly common in emerging architectures. For example, half precision is now available on the NVIDIA V100 GPUs, on which it runs twice as fast as single precision with a proportional savings in energy consumption. Further, the NVIDIA V100's half-precision tensor cores can provide up to a 16x speedup over double precision.

We present a general algorithm for solving an n-by-n nonsingular linear system $Ax = b$ based on iterative refinement in three precisions. The working precision is combined with possibly different precisions for solving for the correction term and for computing the residuals. Our rounding error analysis of the algorithm provides sufficient conditions for convergence and bounds for the attainable normwise forward error and normwise and componentwise backward errors, generalizing and unifying many existing rounding error analyses for iterative refinement.

We show further that by solving the correction equations by GMRES preconditioned by the LU factors the restriction on the condition number can be weakened to allow for the solution of systems which are extremely ill-conditioned with respect to the working precision. Compared with a standard $Ax = b$ solver that uses LU factorization in single precision, these results suggest that on architectures for which half precision is efficiently implemented it will be possible to solve certain linear systems $Ax = b$ in less time and with greater accuracy.

We present recent performance results on the latest GPU architectures which show that this approach can result in practical speedups and also discuss recent work in extending this approach to iterative refinement for least squares problems.

Exasteel - Computational Scale-Bridging with Million-way Concurrency

Oliver Rheinbach - EXASTEEL, *TU Freiberg*

In the EXASTEEL project, computational homogenization using the FE^2 approach is combined with fast nonlinear domain decomposition solvers for multiscale simulations with million-way parallelism. The results obtained in the project EXASTEEL on large supercomputers are discussed, using structured as well as unstructured meshes for the representative volume elements (RVEs). Highly scalable domain decomposition methods of the nonlinear FETI-DP are also discussed as well as recent results on reducing the energy to solution.

HyTeG: Software Design for Extreme-Scale Finite Element Multigrid Solvers

Nils Kohl - TERRA-NEO, *Friedrich-Alexander-Universität Erlangen-Nürnberg*

Insightful, finely resolved simulations of physical models such as Earth-mantle convection require the solution of systems of equations of enormous size. A global resolution of the Earth-mantle of about 1km results in more than a trillion (10^{12}) unknowns. Only solvers with optimal complexity - such as multigrid methods - can achieve that scalability. In this talk we present the HPC framework HyTeG that implements parallel and matrix-free finite-element multigrid solvers for extreme-scale simulations as they are required for modern geophysical applications. We combine excellent performance, scalability and geometric flexibility through structured refinement of unstructured meshes and fully distributed domain partitioning.

Performance Engineering for Tall & Skinny Matrix Multiplications on GPUs

Dominik Ernst - ESSEX-II, *Friedrich-Alexander-Universität Erlangen-Nürnberg*

General matrix-matrix multiplications (GEMM) in vendor- supplied BLAS libraries are best optimized for square matrices but often show bad performance for tall & skinny matrices, which are much taller than wide. Nvidias current CUBLAS implementation delivers only a fraction of the potential performance (as given by the roofline model) in this case. We describe the challenges and key properties of an implementation that can achieve perfect performance. We further evalu- ate different approaches of parallelization and thread distribution, and devise a flexible, configurable mapping scheme. A code generation approach enables a simultaneously flexible and specialized implementation with autotuning. This results in perfect performance for a large range of matrix sizes in the domain of interest, and at least 2/3 of maximum performance for the rest on an Nvidia Volta GPGPU.

Analysis and Performance of Block Krylov Methods for Matrix Functions

Andreas Frommer, *Bergische Universität Wuppertal*

To compute $f(A)b_i, A \in \mathbb{C}^{n\times n}, b \in \mathbb{C}^n, i = 1, \ldots, k$, block Krylov methods can be particularly efficient. The reason is that one approximates from the sum of all individual Krylov spaces, and this richer subspace may allow to get good approximations at earlier stages. In addition, *block reduction* can occur when individual subspaces start to have non-trivial intersection.

On HPC architectures, block methods have the additional advantage that A must be loaded from memory only once for an entire matrix-block vector multiplication AB, $B = [b_1 \mid \ldots \mid b_k]$. This can have significant impact on the wall clock time. However, block Krylov methods also come with an additional cost, typically spent in orthogonalizations and other additional arithmetic, resulting in an overhead of $\mathcal{O}(k^2n + k^3)$.

In this talk we present a methodology for the convergence analysis of block Krylov subspace methods, including restarting. Our approach allows to also consider variants of block Krylov subspace methods which differ from the classical scheme in the way they recombine the individual Krylov subspaces. Our general approach is based on the notion of a *block scalar product* which takes it values in a $*$-subalgebra of $\mathbb{C}^{k\times k}$. This methodology will then be applied to matrix functions $f(A)B$ where f can be represented as a Cauchy integral or a Cauchy-Stieltjes function.

The BEAST eigensolver

Bruno Lang - ESSEX-II, *Bergische Universität Wuppertal*

We report on recent improvements to BEAST, an eigensolver based on subspace iteration and Rayleigh–Ritz extraction. To achieve high performance, BEAST provides three variants for the projection step, double and single precision computations, and allows switching between these at runtime. It exposes several layers of parallelism, and we present numerical experiments demonstrating scalablility to large numbers of cores.

Robust domain decomposition with PETSc: PCBDDC and PCHPDDM

Stefano Zampini, *KAUST*

In this talk, we will discuss robust domain decomposition preconditioners available in the PETSc library, namely the Balancing Domain Decomposition by Constraints and the additive Schwarz methods, which can both be endowed with adaptive coarse spaces to robustify the convergence of Krylov methods. We will review the algorithms and give an overview of their implementation details. Numerical results for different classes of linear systems arising from different discretization methods will be also discussed.

Fast matrix-free solvers and preconditioners for emerging hardware architectures

Andrew Barker, *Lawrence Livermore National Lab*

Emerging hardware architectures feature a large degree of intrinsic parallelism and concurrency, which makes them well-suited to high-order finite element methods. Explicit matrix assembly for such methods is prohibitively expensive, so that matrix-free alternatives are essential to realize good performance on emerging architectures. Tensorized matrix-free algorithms designed for the combination of high-order finite elements and modern hardware have already had a large impact on explicit time simulations, which require fast operator-vector multiplies, but implicit or semi-implicit algorithms require also solvers and preconditioners, which are not as well developed in the matrix-free setting. In this talk, we explore some preconditioners for unassembled linear systems which require as little discretization information as possible, so they can be thought of as analogous to algebraic multigrid in the assembled case. We explore future directions for such solvers and present some examples of their practical use in real simulations.

Domain Decomposition Methods with Adaptive Multipreconditioning

Nicole Spillane, *École Polytechnique*

Domain decomposition methods are a family of parallel solvers for large linear systems. They all share the idea of approximating the inverse of some matrix by a sum of local inverses (in the so-called subdomains). I will present some classical domain decomposition methods, their limitations and some recent efforts to improve their robustness and scalability so that they can be applied to problems arising from real life simulations.

More precisely, I will introduce the GenEO coarse space and how it evolved into the method of adaptive multipreconditioning. This is a modification of the iterative solver (the preconditioned conjugate gradient algorithm). Instead of one single preconditioner, a family of preconditioners is applied at each iteration, each corresponding to one of the subdomains. This significantly increases the size of the minimization space and consequently accelerates convergence. I will introduce the method, discuss its analysis and show numerical results obtained in collaboration with C. Bovet, P. Gosselet, A. Parret-Fréaud and D.J. Rixen.

Improving parallel structured multigrid methods with block smoothers

Matthias Bolten - ExaStencils, *Bergische Universität Wuppertal*

While multigrid methods show an excellent scaling behavior depending on the number of cores only logarithmically, on large-scale supercomputers this logarithmic dependency is visible. To reduce influence of this dependence, the amount of work being carried out on the coarse levels has to be reduced. This can be achieved by applying aggressive coarsening, effectively reducing the number of coarse levels. While this is easily possible in a geometric multigrid setting, the overall performance of the method will deteriorate, as the size of the coarse space is reduced. To retain a good convergence rate, the smoothing procedure has to be improved. We propose to use block smoothers to accomplish this, at the same time this results in a higher locality of the operations performed and thus in a better exploitation of modern computer architectures. The proposed methods cannot be analyzed using standard Local Fourier Analysis techniques, therefore we extended the available analysis methodology to cover the block case, as well.

Pipelined Krylov methods

Wim Vanroose, *Universiteit Antwerpen, Department of Mathematics*

Delays in communication and synchronisation are dominating over the timing of the compute tasks in many parallel applications. How can we reorganise Krylov methods when dot-products take too long to complete? An answer is given by pipelined Krylov methods. These are variants of Krylov methods were the different compute and communication tasks are executed simultaneously by using pipelining. In this talk we give an introduction, the derivation of pipelined Krylov methods and an analysis of the rounding error propagation.