

PMAA16

**6-8 July 2016
Bordeaux
France**

Table of contents

MS - Computation of large numbers of eigenpairs of large sparse matrices	6
CORK: Compact Rational Krylov solvers for nonlinear eigenvalue problems, Karl Meerbergen [et al.]	7
Optimizing least-squares rational filters for solving interior eigenvalue problems, Edoardo Di Napoli [et al.]	8
Inexact FEAST subspace iteration for solving large sparse eigenvalue problems, Brendan Gavin [et al.]	9
Latest Developments in PRIMME to Compute Many Interior Eigenpairs in Large Hermitian Matrices, Eloy Romero Alcalde [et al.]	10
A quadrature-based parallel eigensolver for large-scale simulations, Tetsuya Sakurai [et al.]	11
"TraceMIN: A Scalable Parallel Algorithm for Large Sparse Symmetric Eigenvalue Problems", Ahmed Sameh	12
Computing invariant subspaces in quantum chemistry, Joost Vandevondele	13
A Projected Preconditioned Conjugate Gradient Algorithm for Computing a Large number of Eigenpairs, Eugene Vecharynski [et al.]	15
MS - Task-based scientific library on top of runtime systems	16
The ParalleX Execution Model and HPX-5 Runtime System for Scalable Libraries, Thomas Sterling	17
Scalable Task-Based Software at Petascale and Beyond, Martin Berzins	18
Exploiting Kepler architecture in sparse direct solver with runtime systems, Mathieu Faverge	19
Concurrency is not complexity: A use case with PaRSEC, George Bosilca	20
Task-based sparse Cholesky solver on top of runtime system, Florent Lopez [et al.]	21
Hierarchical Computations on Manycore Architectures, Hatem Ltaief	22

qr_mumps: a runtime-based Sequential Task Flow parallel solver, Alfredo Buttari [et al.]	23
Towards Highly Parallel and Compute-bound Computation of Selected Eigenvectors given a Matrix in Schur Form, Lars Karlsson [et al.]	24
MS - Algorithms and applications of Krylov methods and preconditioning	26
Preconditioned Krylov solvers for non-linear Schrödinger equation with application in Superfluids and Superconductors, Wim Vanroose [et al.]	27
Multilevel Variable-Block Schur-Complement Based Preconditioning for Large CFD Computation, Bruno Carpentieri [et al.]	28
MPI-GPU parallelism in preconditioned Krylov solvers for block-tridiagonal matrices, Alejandro Lamas Daviña [et al.] .	30
Parallelization of the Rational Arnoldi Algorithm, Mario Berljafa [et al.]	31
CP - Sparse direct methods	32
Scaling a Multifrontal Solver, Bob Lucas [et al.]	33
Efficient parallelization of direct solvers for isogeometric analysis, Maciej Paszynski [et al.]	34
Dissection : A direct solver with kernel detection for finite element matrices on multi-core supercomputers, Atsushi Suzuki [et al.]	36
CP - Applications	38
Optimizing linear operators within FETI DDM, Václav Hapla [et al.]	39
Development of a Multi-physics Code with Adaptive Mesh Refinement, William Dai	40
Hybrid programming in the legacy code AVBP using OMPs, Cyril Fournier [et al.]	42
High Intensity Matrix Polynomial Solvers for the Heat and Poisson Equations., Thierry Dumont [et al.]	43
CP - Eigensolver	45
Approximating Functionals of Hermitian Matrix Product Operators via a Block Lanczos Method, Moritz August [et al.] .	46
ChASE: Chebyshev Accelerated Subspace iteration Eigensolver library on heterogeneous architectures, Edoardo Di Napoli [et al.]	47

CP - High performance computing	49
Increasing Arithmetic Intensity using Stencil Compilers on Many-Core Architectures, Simplice Donfack [et al.]	50
A Parallel Algorithm for Scalable Bloom Filters Matrix, Zhu Wang	51
Parallel solver for shifted systems, Nela Bosner [et al.]	52
MS - Parallel SVD/EVD solvers for large dense matrices	53
Computation of the CS and the indefinite CS decomposition, Sanja Singer	54
Accelerating the SVD using GPUs, Mark Gates [et al.]	55
Asymptotic Quadratic Convergence of the Parallel Block-Jacobi EVD Algorithm for Hermitian Matrices, Gabriel Oksa [et al.]	56
High-Performance Parallelization Method of DSYRK for SVD and other Matrix Computations on Xeon Phi, Shuhei Kudo [et al.]	57
Preprocessing Algorithm for the Generalized SVD on the Graphics Processing Units, Goran Flegar	58
ELPA: Algorithmic extensions and optimization, Michael Rippl	59
An accelerated tridiagonal divide-and-conquer algorithm on distributed memory architectures, Shengguo Li [et al.]	60
GPU-hosted SVD computations for large scale hierarchical matrices, George Turkiyyah [et al.]	62
MS - Advanced topics for EVP and SVD towards future scientific simulation	63
Parallel dense eigenvalue solver and SVD solver for post-petascale computing systems, Toshiyuki Imamura	64
A real-valued method for solving complex symmetric linear systems arising in contour integral eigensolver, Yasunori Futamura [et al.]	66
A new implementation of the orthogonal QD algorithm and its application, Kinji Kimura	68
Performance analysis of the quadruple precision eigensolver library QPEigenK on the K computer, Yusuke Hirota [et al.]	70
MS - Fast sparse selected inversion techniques	72
Applications of trace estimation techniques, Yousef Saad . . .	73
Parallel Selected Inversion of Asymmetric Matrices, Mathias Jacquelin [et al.]	74
Takahashi formulas and the generalized Takahashi structure, Klaus Gaertner, Olaf Schenk	75
Highly parallel stochastic estimation of the diagonal of inverse matrices, Fabio Verbosio [et al.]	76

MS - Multilevel parallel linear solvers	77
Multilevel Spectral Analysis of H-Matrices, Yuval Harness [et al.]	78
Fast Methods applied to BEM Solvers for industrial problems, Guillaume Sylvand	79
Performance and scalability of a parallel Block Low-Rank multifrontal solver, Patrick Amestoy [et al.]	81
Approximate factorization preconditioning with HSS matrices and random sampling, Pieter Ghysels [et al.]	82
Feedback in the use of multilevel parallel linear solvers for thermomechanical studies, Olivier Boiteau [et al.]	83
Designing an efficient and scalable block low-rank direct solver for large scale clusters, Xavier Lacoste [et al.]	85
Parallel hierarchical solver, Eric Darve [et al.]	86
A massively parallel sparse linear solver ESPRESO, Lubomir Riha [et al.]	87
MS - Efficient computation of inner eigenvalues of large sparse matrices	89
Efficient subspace iteration with Chebyshev-type filtering, Martin Galgon [et al.]	90
Parallel methods for computing interior eigenvalues in linear and nonlinear eigenvalue problems in SLEPc, Jose E. Roman [et al.]	91
Filtered thick restart Lanczos algorithm and the EVSL package, Yousef Saad	92
Block Krylov-type complex moment-based nonlinear eigensolver with hierarchical parallelism, Akira Imakura [et al.] . .	93
MS - Sparse matrix and tensor computations	95
High Performance Parallel Sparse Tucker Decompositions, Oguz Kaya [et al.]	96
An Exploration of Optimization Algorithms for High Performance Tensor Completion, Shaden Smith [et al.]	97
An Empirical Study of Sparse BLAS on Emerging Heterogeneous Processors, Weifeng Liu [et al.]	99
Making effective sparse matrix–vector multiplication both portable and usable, Albert-Jan Yzelman	100
CP - Preconditioning	102
Parallel preconditioners for a higher order time integration methods, Radim Blaheta [et al.]	103
Scaling preconditioners for parallel CG-accelerated projection techniques, Dan Gordon [et al.]	104

Large-Scale Sparse Inverse Covariance Estimation, Matthias Bollhoefer [et al.]	106
CP - Linear algebra	107
Exploring Vectorization Possibilities on the Intel Xeon Phi for Solving Tridiagonal Systems, Ioannis Venetis [et al.] . . .	108
Streamwise numbering for Gauss-Seidel and bidiagonal preconditioners in convection dominated flows, Paula Córdoba [et al.]	110
SpMV Algorithm for High Bandwidth Memory Based Application Specific Hardware Accelerator, Fazle Sadi [et al.] . . .	111
Author Index	112

**MS - Computation of large
numbers of eigenpairs of
large sparse matrices**

CORK: Compact Rational Krylov solvers for nonlinear eigenvalue problems

Karl Meerbergen ^{*† 1}, Roel Van Beeumen , Wim Michiels

¹ KU Leuven – Celestijnenlaan 200A, 3001 Leuven, Belgium, Belgium

We discuss properties of the compact rational Krylov (CORK) methods for solving large-scale nonlinear eigenvalue problems. For many years, linearizations were used for solving polynomial and rational eigenvalue problems. On the other hand, for the general nonlinear case, can first be approximated by a (rational) matrix polynomial and then a convenient linearization is used. However, the major disadvantage of linearization-based methods is the growing memory and orthogonalization costs with the iteration count, i.e., in general they are proportional to the degree of the polynomial. Therefore, the CORK family of rational Krylov methods exploits the structure of the linearization pencils by using a generalization of the compact Arnoldi decomposition. In this way, the extra memory and orthogonalization costs due to the linearization of the original eigenvalue problem are negligible for large-scale problems. Furthermore, we prove that each CORK step breaks down into an orthogonalization step of the original problem dimension and a rational Krylov step on small matrices. We also briefly discuss implicit restarting of the CORK method and how to exploit low rank structure. The CORK method is illustrated with two large-scale examples. Finally, we show relations with a class of contour integration methods.

Keywords: eigenvalue problems, rational Krylov method, linearisation, contour integration

*Speaker

†Corresponding author: karl.meerbergen@cs.kuleuven.be

Optimizing least-squares rational filters for solving interior eigenvalue problems

Edoardo Di Napoli* ^{1,2}, Jan Winkelmann † ³

¹ Juelich Supercomputing Centre – Forschungszentrum Jülich GmbH (JSC) – Wilhelm-Johnen-Straße 52428 Jülich, Germany

² Juelich Aachen Research Alliance – High-performance Computing (JARA-HPC) – Schinkelstrasse 2, 52062 Aachen, Germany, Germany

³ Aachen Institute for Advanced Study in Computational Engineering Science - RWTH Aachen (AICES - RWTH) – Schinkelstrasse 2, 52062 Aachen, Germany, Germany

FEAST-like solvers have gained popularity for interior eigenproblems. Originally derived from the Cauchy’s integration formula, a spectral projector is approximated using numerical integration. This results in a matrix-valued rational transfer function that is used in a linear system solve to dampen the unwanted eigenvalues. The poles of the transfer function depend on the integration rule that is used.

Recent approaches interpret the spectral projector not as a numerical contour integration, but directly as a rational transfer function. In our novel approach we apply an optimization based approach to the transfer function, resulting in a non-convex Least-Squares problem. We show some results that mitigate the non-convexity of the problem.

An important advantage of our approach is the ability to add further constraints. Rational transfer functions with poles near the real axis may lead to accuracy or performance problems when applying the function to a matrix. Constrained optimization can effectively eliminate this problem.

We provide comparisons showing that this approach performs better than existing approaches.

Keywords: Rational filters, least squares, eigensolver, optimization

*Corresponding author: e.di.napoli@fz-juelich.de

†Speaker

Inexact FEAST subspace iteration for solving large sparse eigenvalue problems

Brendan Gavin ¹, Eric Polizzi * ¹

¹ University of Massachusetts Amherst (UMass Amherst) – United States

The FEAST Eigenvalue algorithm uses a shift-invert strategy that requires solving multiple linear systems with shifts in the complex plane in order to find the eigenvectors of a matrix whose eigenvalues that lie inside a given interval. Similarly to other shift-invert algorithms, solving the linear systems represents the most challenging part of the computation. We present an innovative modification of FEAST that finds interior eigenpairs without solving linear systems by building on the inexact shifted subspace iteration work of A. Spence et al. By using only a few iterations of iterative refinement per subspace update, we are able to apply the FEAST algorithm to problems where the direct solution of the linear systems is too costly, but without requiring the use of matrix factorizations or preconditioner matrices. This work applies to finding the eigenvalue decomposition of both Hermitian and non-Hermitian matrices, as well as to finding the singular value decomposition.

Keywords: FEAST

*Speaker

Latest Developments in PRIMME to Compute Many Interior Eigenpairs in Large Hermitian Matrices

Eloy Romero Alcalde * ¹, Andreas Stathopoulos ¹

¹ College of William and Mary – Williamsburg, Virginia, United States

PRIMME is a high-performance, standalone library dedicated to compute partial eigendecompositions of large, sparse Hermitian matrices with methods that have schemes close to the Davidson-type methods. Examples of popular methods implemented in PRIMME are Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG), Generalized Davidson with CG restarting (GD+k) and Jacobi-Davidson. These methods exhibit clear advantages under memory limitations and when preconditioner is available. However, because of the cost of keeping the new solutions orthogonal against the already converged eigenvectors, their performance does not scale linearly with respect to the number of eigenvalues.

In this talk we discuss two approaches to reduce the impact of the orthogonalization in the context of finding all eigenvalues in a region. One is spectrum slicing, in which the target region is split into subintervals treated independently. In this way, eigenvectors are only kept orthogonalized against other eigenvectors corresponding to the same subinterval. The accuracy of the computed eigenpairs controls the orthogonality between eigenvectors in different subintervals. The other approach is reducing the number of outer iterations by using a polynomial preconditioning technique. Concretely we show results using least-squares filters.

Keywords: Davidson, spectrum slicing, polynomial filtering

*Speaker

A quadrature-based parallel eigensolver for large-scale simulations

Tetsuya Sakurai *¹, Yasunori Futamura¹, Akira Imakura¹

¹ University of Tsukuba – Japan

Large-scale eigenvalue problems arise in wide variety of scientific and engineering applications such as nano-scale materials simulation, vibration analysis of automobiles, analysis of big data, etc. In such situations, high performance parallel eigensolvers are required to exploit the full capability of large-scale computing environments. In this talk, we present a parallel method, the Sakurai-Sugiura method (SSM), for solving large-scale interior eigenvalue problems. This method is derived using a numerical quadrature of matrix inverses with several quadrature points on the complex plane. A hierarchical structure of the method is used to utilize a large amount of computational resources. SSM can avoid global communication between computing nodes because of its hierarchical structure, however this parallel structure causes a larger memory requirement, and limits the problem size. We show some techniques to reduce the memory requirement of SSM to attain solutions of large-scale problems. A rational type stochastic estimation method for eigenvalue density is also used. We show numerical experiments with problems in electronic structure calculations, lattice quantum chromodynamics and shell model calculations of nucleus confirming the efficiency of the proposed method.

Keywords: parallel eigensolver, contour integral, eigenvalue density

*Speaker

TraceMIN: A Scalable Parallel Algorithm for Large Sparse Symmetric Eigenvalue Problems

Ahmed Sameh * ¹

¹ Purdue University [West Lafayette] (Purdue) – Hovde Hall, 610 Purdue Mall, West Lafayette, IN 47907, United States

The trace-minimization scheme (TraceMin) proved to be a robust and scalable parallel algorithm for obtaining few of the smallest eigenpairs of large sparse symmetric eigenvalue problems $A x = \lambda B x$, where A is symmetric and B is symmetric positive definite. Although the first version of this algorithm was published in 1982 (more than three decades ago) exhaustive tests have shown that TraceMin is quite competitive to many eigensolvers introduced since the late 1990s. In this presentation we outline simple modifications of TraceMin that allows computing large number of eigenpairs belonging to any intermediate interval in the spectrum.

Keywords: Symmetric eigenvalue problems, Trace minimization, Solving large saddle point problems, Parallel computing.

*Speaker

Computing invariant subspaces in quantum chemistry

Joost Vandevondele * ¹

¹ ETH Zurich (ETHZ) – HIT G 41.4 Wolfgang-Pauli-Strasse 27 8093 Zurich, Switzerland, Switzerland

Eigenvalue problems are considered central in quantum chemistry. Indeed, the matrix representation of the Schroedinger Equation and related theories such as Hartree-Fock or density functional theory, are naturally expressed in such a form. We will focus on the case of density functional theory, where the corresponding problem requires the computation of a large fraction (typical 1-50%) of the eigenvalues and eigenvectors, of matrices can be large (up 10 million rows) and sparse (a few thousand non-zeros per row). Furthermore, due to the use of non-orthogonal basis sets, the eigenvalue problem is generalized. Additionally, the problem is non-linear, i.e. is commonly solved in a self-consistent fashion. We will discuss two different approach to effectively deal with this problem that both exploit the fact that only the subspace spanned by the lowest eigenvectors is needed, and not the individual eigenvalues and eigenvectors. The first approached employs direct minimization with respect to the vectors spanning the space, with a reformulation of the orthonormality constraint.[1] This method scales cubically with matrix size (for a fixed fraction of eigenvectors), and is thus suitable for smaller matrices. The second approach represents the subspace by a projector matrix, but exploits the fact that this projector matrix is sparse.[2] The projector is obtained from the matrix sign function, which is computed using sparse linear algebra, in particular matrix matrix multiplication. The latter operation has been implemented in a sparse matrix library[3], which is massively parallel and GPU enabled[4]. Aspects of this implementation will be discussed as well.

[1] Vandevondele, J; Hutter, J. 2003. An efficient orbital transformation method for electronic structure calculations. *JOURNAL OF CHEMICAL PHYSICS* 118 (10): 4365-4369. <http://dx.doi.org/10.1063/1.1543154>.

[2] Vandevondele J; Borstnik U; Hutter J; 2012 Linear scaling self-consistent field calculations for millions of atoms in the condensed phase. *JOURNAL OF CHEMICAL THEORY AND COMPUTATION* 8(10): 3565-3573 <http://dx.doi.org/10.1021/ct200897x>

[3] Borstnik U; Vandevondele J; Weber V; Hutter J; 2014 Sparse Matrix

*Speaker

Multiplication: the distributed block-compressed sparse row library PAR-
ALLEL COMPUTING 40(5-6): 47-58 <http://dx.doi.org/10.1016/j.parco.2014.03.012>
[4] Schütt O; Messmer P; Hutter J; VandeVondele J GPU-Accelerated Sparse
Matrix-Matrix Multiplication for Linear Scaling Density Functional Theory
in Electronic Structure Calculations on Graphics Processing Units: From
Quantum Chemistry to Condensed Matter Physics (eds R. C. Walker and A.
W. Götz), John Wiley and Sons, Ltd, Chichester, UK.
<http://dx.doi.org/10.1002/9781118670712.ch8>

A Projected Preconditioned Conjugate Gradient Algorithm for Computing a Large number of Eigenpairs

Eugene Vecharynski ^{*† 1}, Chao Yang ¹, John Pask ²

¹ Lawrence Berkeley National Laboratory (LBNL) – Lawrence Berkeley National
Lab MS 50F Cyclotron Rd., Berkeley, CA 94720, United States

² Lawrence Livermore National Laboratory (LLNL) – Lawrence Livermore
National Laboratory 7000 East Avenue • Livermore, CA 94550, United States

We present a Projected Preconditioned Conjugate Gradient (PPCG) algorithm for computing a relatively large number of lowest eigenvalues of a Hermitian matrix. The algorithm performs fewer Rayleigh-Ritz calculations, and as a result is substantially faster than standard approaches. We will discuss a number of practical issues for implementing PPCG, and demonstrate its performance in Kohn-Sham density functional theory based electronic structure calculations.

Keywords: preconditioned eigensolvers, Rayleigh, Ritz method, electronic structure calculations

*Speaker

†Corresponding author: vecharynski@lbl.gov

**MS - Task-based scientific
library on top of runtime
systems**

The ParalleX Execution Model and HPX-5 Runtime System for Scalable Libraries

Thomas Sterling * ¹

¹ Indiana University Center for Research in Extreme Scale Technologies (IU
CREST) – United States

The combined challenges of parallel efficiency and scalability have motivated recent work in dynamic adaptive execution such as a number of Asynchronous Multi-Tasking systems (AMT) implemented primarily as runtime system libraries. The advanced HPX-5 runtime system is derived from the abstract ParalleX execution model derived to address the interrelated sources of performance degradation of starvation, latency, overheads, and the delays of waiting due to contention for shared logical and physical resources (SLOWER). ParalleX replaces the CSP model with a hierarchical global name space within which event driven computation, and futures/dataflow based synchronization enable multiple-threaded computation to support dynamic adaptive computation. The HPX-5 runtime system (now in release 2.3) is an experimental library that embodies the principles of the ParalleX execution model to enable guided computing through introspection of system and application runtime status information rather than limited to conventional ballistic computing. Libraries and applications have been developed on top of HPX-5 demonstrating the value of dynamic scheduling, lightweight parallelism, over-subscription, fine-grained synchronization, load balancing, and critical path analysis. A new version of the MTL linear algebra library, the new DASHMM library for fast multipole methods and Barnes-Hut N-body computations, dynamic methods such as PIC, AMR, and Wavelets computations, and graph processing techniques as well as others have been implemented and evaluated across a diversity of MPPs and commodity clusters of a wide range of scales. The presentation will describe the principles of ParalleX and features of the HPX-5 concentrating on the delivered capabilities and results for representative applications.

*Speaker

Scalable Task-Based Software at Petascale and Beyond

Martin Berzins ^{*† 1}

¹ SCI Institute - University of Utah – United States

Task-based software which uses adaptive and asynchronous execution is viewed as a possible software paradigm for post-petascale architectures. Task based approaches often combine a programming model with a runtime system.

One such system used in the Utah Uintah software is described. The emphasis is on the generality of the approach and on the design of the runtime system in a way that allows for scaling of complex engineering applications to run at the limits of todays computer architectures. An additional level of complexity arises from dealing with portability across very different future architectures consisting of GPUs, Xeon Phi, conventional and low power cpus. This approach is currently being addressed in Uintah through a substantial rewrite to accomodate the Kokkos portable performance layer. Preliminary results with Kokkos will be presented.

*Speaker

†Corresponding author: mb@sci.utah.edu

Exploiting Kepler architecture in sparse direct solver with runtime systems.

Mathieu Faverge *^{1,2}

¹ Bordeaux INP - Inria Bordeaux - Sud-Ouest

² Innovative Computing Laboratory (ICL) – TN, USA

Sparse direct solvers is a time consuming operation required by many scientific applications to simulate physical problems. By its important overall cost, many studies tried to optimize the time to solution of those solvers on multi-core and distributed architectures. More recently, many works have addressed heterogeneous architectures to exploit accelerators such as GPUs or Intel Xeon Phi with interesting speedup. Despite researches towards generic solutions to efficiently exploit those accelerators, their hardware evolution requires continual adaptation of the kernels running on those architectures. The recent Nvidia architectures, as Kepler, present a larger number of parallel units thus requiring more data to feed every computational units. A solution considered to supply enough computation has been to study problems with large number of small computations. The batched BLAS libraries proposed by Intel, Nvidia, or the University of Tennessee are examples of this solution. We discuss in this talk the use of the variable size batched matrix-matrix multiply to improve the performance of a the PaStiX sparse direct solver. Indeed, this kernel suits the supernodal method of the solver, and the multiple updates of variable sizes that occur during the numerical factorization. Performance results on a spectrum of matrices with different properties will be presented.

Keywords: sparse direct solver, GPU, runtime system, DAG

*Speaker

Concurrency is not complexity: A use case with PaRSEC

George Bosilca * ¹

¹ Innovative Computing Laboratory, University of Tennessee, Knoxville (ICL) – United States

The exponential growth of computer power over the last decades is transforming many traditionally experimental fields, such as chemistry, and biology, into simulation-driven sciences. This progress is challenged by two fundamental trends: 1) the increasing need for predictive simulation characterized by higher-fidelity - as well as high-cost and high-complexity - models, and 2) the ongoing shift from serial to massively concurrent computer architectures, characterized by increasingly complex storage hierarchies, simpler but vastly more numerous data-parallel compute cores, and the essential importance of power consumption and fault tolerance. Their transition will not be complete without a revolutionary progress of modern computational science to counterbalance these architectural complexities. PaRSEC, a distributed task-based runtime, provides a possible answer to this challenge, by shifting the focus in parallel algorithms, from control to data flow, improving portability and productivity.

Keywords: runtime, distributed, task, heterogeneous

*Speaker

Task-based sparse Cholesky solver on top of runtime system

Florent Lopez *¹, Iain Duff¹, Jonathan Hogg¹

¹ Numerical Analysis Group - STFC Rutherford Appleton Laboratory –
Numerical Analysis Group STFC Rutherford Appleton Laboratory Chilton,
Didcot OX11 0QX, UK, United Kingdom

In this talk we present the implementation of a task-based sparse Cholesky solver on top of runtime system. To achieve this, we use two different programming models: a Sequential Task Flow (STF) model and Parametrized Task Graph (PTG) model. We first present an STF-based implementation of our code using both the StarPU runtime system and the OpenMP 4.0 standard and then, we move to a PTG model using the PaRSEC runtime system. We compare these implementations against the state-of-the-art MA87 solver from the HSL library to assess our approach in terms of performance and scalability on shared-memory multicore architectures. Additionally, we present a porting of our code on GPUs architectures to show the benefit of exploiting runtime systems in the context of heterogeneous systems.

Keywords: Sparse Matrices, Cholesky, Runtime system, GPU

*Speaker

Hierarchical Computations on Manycore Architectures

Hatem Ltaief * ¹

¹ KAUST – Saudi Arabia

Algorithms that would fully exploit extreme scale hardware will have at worst $O(N \log N)$ operation complexity for problem size N , so that the parallel weak scaling on P processors has a hope of achieving $O(\log P)$ time complexity. Many practically globally coupled linear systems can aim at this desirable complexity because they are "data-sparse," a class that includes hierarchically low-rank matrices (H-matrices). We are motivated by H-matrices arising from spatial correlations, integral equations, and PDEs following processes of Schur complementation. Compressing to and manipulating high dimensional operators in hierarchically low-rank format requires a large number of small BLAS operations, which are superlinear only in the local rank parameter, assumed small. Some of the required operations are still missing in libraries for emerging multicore and GPGPU processing elements, and efficient data structures and iterators – imitating the tree-based operations of fast multipole – must also be defined. Placing the power of hierarchically low-rank operations in the hands of users will therefore require significant new library development. We will describe the motivation and current state of development of the KBLAS and HBLAS components of the Hierarchical Computations on Manycore Architectures (HiCMA) library being developed in the Extreme Computing Research Center at KAUST.

Keywords: hierarchical, GPU, linear algebra, high performance computing

*Speaker

qr_mumps: a runtime-based Sequential Task Flow parallel solver

Alfredo Buttari * ¹, Emmanuel Agullo ², Abdou
Guermouche ², Florent Lopez ³

¹ Institut de recherche en informatique de Toulouse (IRIT) – Université Toulouse I [UT1] Capitole, Université des Sciences Sociales - Toulouse I, Institut National Polytechnique de Toulouse - INPT, Université Paul Sabatier [UPS] - Toulouse III, CNRS : UMR5505, Université Toulouse le Mirail - Toulouse II, Université Toulouse I (UT1) Capitole, Université Paul Sabatier (UPS) - Toulouse III – 118 Route de Narbonne, F-31062 Toulouse Cedex 9, France

² HiePACS (INRIA Bordeaux - Sud-Ouest) – CNRS : UMR5800, Université de Bordeaux, INRIA, CERFACS – INRIA Bordeaux Sud-Ouest, HiePACS project team, 200 avenue de la Vieille Tour, 33405 Talence cedex, France, France

³ Rutherford Appleton Laboratory - Didcot (RAL) – Science and Technology Facilities Council Rutherford Appleton Laboratory Harwell Science and Innovation Campus Didcot OX11 0QX, United Kingdom

qr_mumps is a parallel, direct solver for sparse linear systems based on the multifrontal QR factorization. Parallelism is achieved using a Sequential Task Flow (STF) based model on top of the StarPU runtime system. In this talk we will show how STF parallelism can be applied to a sparse, direct solver and how the use of a modern runtime system allows for the portable and efficient implementation of complex algorithms that can improve its performance and scalability as well as its memory consumption. Finally we will discuss the details of porting the qr_mumps solver on GPU-equipped architectures. Experimental results on multiple architectures will assess the portability of the proposed approach.

Keywords: sparse linear algebra, direct solvers, runtime systems

*Speaker

Towards Highly Parallel and Compute-bound Computation of Selected Eigenvectors given a Matrix in Schur Form

Lars Karlsson ^{*† 1}, Björn Adlerborn ¹, Carl Christian Kjelgaard Mikkelsen ¹, Bo Kågström ¹

¹ Umeå University – Sweden

Given a dense n -by- n matrix in Schur form, a user-specified set of computed eigenvalues (a list of diagonal indices of the Schur matrix), and optionally a unitary transformation matrix, the problem of interest is to compute left and/or right eigenvectors of the selected eigenvalues in a robust manner. Optionally, the computed eigenvectors are then back-transformed by the provided transformation matrix. The robust implementation available in LAPACK today (via the subroutines `xTREVC`) is based on an algorithm by Edward Anderson (LAWN#36, 1991) and can be described as follows. For each selected eigenvalue, a particular triangular system is solved while carefully avoiding overflow by scaling. Each computed eigenvector is then optionally back-transformed by a matrix-vector multiply. Both steps (back substitution and back transformation) are memory-bound. The LAPACK subroutine only exploits parallelism implicitly via multithreaded BLAS. Thus the parallelism available through the simultaneous processing of several selected eigenvalues is not exploited. Another option for improving the parallelism of the LAPACK implementation would be to use Batched BLAS for the many back transformations.

Gates, Haidar, and Dongarra (2014) recently extended the approach taken in LAPACK. First, they observed that the eigenvectors associated with eigenvalues adjacent on the diagonal of the Schur matrix can be back-transformed simultaneously using a single matrix-matrix multiply with very little overhead. This replaces matrix-vector multiplies with matrix-matrix multiplies and thereby improves the arithmetic intensity of the back transformation step. Second, they offloaded all back transformations to a GPU and overlapped both the computation and the communication between CPU and GPU. Third, they performed the independent back substitutions on the

*Speaker

†Corresponding author: larsk@cs.umu.se

CPU in parallel. While a significant improvement, the back substitution step is still memory-bound and sequential, the computation as a whole is limited to a shared memory node, and the amount of parallelism that can be exploited is limited to the number of selected eigenvalues, which can be anything between 1 and the matrix size n .

We provide three extensions to the robust back substitution algorithm by Anderson (LAWN#36, 1991) that makes it compute-bound and more suitable for parallelization. The first extension provides support for multiple right-hand sides by solving for several eigenvalues at the same time, which moves the algorithm towards a compute-bound regime. Secondly, the original element-wise algorithm is reformulated into a blocked algorithm, which leads to coarser-grained parallelism with a tunable number of synchronization points and thereby makes the algorithm more suitable for parallelization. Finally, the control flow of the original algorithm relies on a set of non-trivial data-dependent conditional statements. We have found them simple enough to permit vectorization across the multiple right-hand sides. This adds vector-level parallelism and also reduces the overheads of conditional statements.

By parallelizing across the selected eigenvalues and within each individual back substitution and back transformation step, our algorithm can utilize more cores than previously published algorithms. The computation can be mapped to a runtime system that manages the parallelism across multiple distributed memory nodes as well as between CPUs and attached accelerators/GPUs.

Future work includes support for the real case, where two-by-two blocks (corresponding to complex conjugate eigenvalue pairs) along the diagonal complicate the back substitution step. It remains to investigate if the ideas of vectorization and blocking carry over from the complex to the real case. Extensions to the generalized eigenvalue problem are also planned.

**MS - Algorithms and
applications of Krylov
methods and preconditioning**

Preconditioned Krylov solvers for non-linear Schrödinger equation with application in Superfluids and Superconductors

Wim Vanroose * ¹, Michiel Wouters[†] ¹

¹ University of Antwerp – Belgium

In this talk we discuss a scalable solver for the non-linear Schrödinger equation with parameters. This equation appears in a multitude of complex physics problems such as superconductors and superfluids. In this application it is important to explore the solution landscape as a function of the parameters. We illustrate how the Non-linear Schrödinger equation can be efficiently preconditioned using multigrid and deflation. Having access to the lowest eigenvalues of the operator leads to a powerful method where the Krylov subspace can be recycled during the parameter exploration.

Keywords: Deflation, Multigrid, parametric problems

*Speaker

[†]Corresponding author: michiel.wouters2@uantwerpen.be

Multilevel Variable-Block Schur-Complement Based Preconditioning for Large CFD Computation

Bruno Carpentieri ^{*† 1}, Masha Sosonkina[‡]

¹ School of Science and Technology, Nottingham Trent University (NTU) –
Burton Street Nottingham NG1 4BU Great Britain, United Kingdom

Sparse matrices arising from the solution of systems of partial differential equations may often exhibit a fine-grained block structure when several unknown physical quantities are associated with the same grid point. Examples arise in the finite element discretization of the Navier-Stokes equations for turbulent flows analysis, in the boundary element discretization of the Helmholtz equations in electromagnetic scattering applications and in the study of the mechanisms underlying cardiac electrical dynamics modelled by the Bidomain equations, to name a few. If variables assigned to each grid point are numbered consecutively, the matrix arising from the discretization may have a block structure, with the presence of small and usually dense nonzero blocks in the pattern, due to the mutual coupling of the variables at the same node. We refer to this form of blocking as a perfect block ordering. On the other hand, when the matrix is general unstructured, it is sometimes possible to compute imperfect block orderings by treating some zero entries of the matrix as nonzero elements, with a little sacrifice of memory, and grouping together sets of rows and columns having a similar nonzero structure.

In all these situations, it is natural to consider block forms of multi-elimination methods that can exploit any available block structure, either perfect or imperfect. A clear advantage is to store the matrix as a collection of blocks using the variable block compressed sparse row (VBCSR) format, saving column indices and pointers for the block entries. On indefinite problems, computing with blocks instead of single elements enables a better control of pivot breakdowns, near singularities, and other possible sources of numerical instabilities. Block Incomplete LU (ILU) solvers may be used instead of

*Speaker

†Corresponding author: bruno.carpentieri@ntu.ac.uk

‡Corresponding author: msosonki@odu.edu

pointwise ILU methods as local solvers. A full block implementation may be unravelled based on higher level optimized Basic Linear Algebra Subroutines (BLAS), having better flops to memory ratios on modern cache-based computer architectures. Finally, grouping variables in clusters, the Schur complement is smaller and the last reduced system may be better conditioned and easier to solve.

Our recently developed variable block algebraic recursive multilevel solver (VBARMS) incorporates compression techniques during the factorization to detect fine-grained dense structures in the linear system automatically, without any user's knowledge of the underlying problem, and exploits them to improve the overall robustness and throughput of the multilevel iterative solver. Exposing dense matrix blocks during the factorization may lead to more efficient and numerically stable parallel solvers.

In this talk we present a performance analysis of VBARMS against the parallel implementation of the ARMS method provided in the pARMS package. We illustrate its remarkable efficiency for solving block structured linear systems arising from an implicit Newton-Krylov formulation of the Reynolds Averaged Navier Stokes equations in turbulent incompressible flow analysis past a three-dimensional wing, in combination with conventional parallel global solvers such as in particular the Restricted Additive Schwarz preconditioner. We show how the performance of VBARMS improves on hardware accelerators by revealing a high-degree of the parallelism. Finally, we report on ongoing experiments that use a different compression for the coefficient matrix and for the Schur Complement matrix, to improve the robustness and to decrease the factorization costs of the VBARMS method.

Keywords: Krylov Subspace Methods, Multilevel Iterative Solvers, Parallel Preconditioning, Block Methods, Navier, Stokes Equations, Computational Fluid Dynamics

MPI-GPU parallelism in preconditioned Krylov solvers for block-tridiagonal matrices

Alejandro Lamas Daviña ^{*† 1}, Jose E. Roman ¹

¹ Universitat Politècnica de Valencia (UPV) – Spain

We consider the computation of a few eigenpairs of a generalized eigenvalue problem $Ax = \lambda Bx$ with block-tridiagonal matrices, not necessarily symmetric, in the context of projection methods. For interior eigenvalues, the shift-and-invert transformation technique will require solving linear systems in each iteration of the eigensolver. This can be computed directly by means of an LU factorization or in an iterative way with preconditioned Krylov methods (inexact shift-and-invert). In this work we compare the use of these methods to solve the linear systems.

In order to solve the block-tridiagonal system, the BCYCLIC (block oriented cyclic reduction) and SPIKE algorithms have been selected and combined, and some of their variants have been used as preconditioners in the inexact shift-and-invert case.

A multi-GPU solution code has been created by means of MPI, in the context of SLEPc, the Scalable Library for Eigenvalue Problem Computations, and several implementations of the algorithms have been developed making use of different CUDA libraries, such as CUBLAS and MAGMA, that efficiently perform the required dense operations on the GPU.

A study of scalability has been carried out in a GPU-equipped cluster, and the performance results for the CUDA implementations and their CPU counterparts, and a comparison between the different method approaches and algorithms will be presented and discussed.

Keywords: Eigenvalue problem, block tridiagonal matrices, MPI, GPU

*Speaker

†Corresponding author: alejandro.lamas@dsic.upv.es

Parallelization of the Rational Arnoldi Algorithm

Mario Berljafa * ¹, Stefan Guettel ¹

¹ The University of Manchester – United Kingdom

The rational Arnoldi algorithm is a popular method in scientific computing used to construct an orthonormal basis of a rational Krylov space. Each basis vector is a rational matrix function times the starting vector. Rational functions possess a partial fraction expansion which often allows to compute several basis vectors simultaneously. However, this parallelism may cause instability due to the orthogonalization of ill-conditioned bases. We present and compare continuation strategies to minimize these effects, and discuss our (parallel) HPC implementation.

Keywords: rational Arnoldi, parallel

*Speaker

CP - Sparse direct methods

Scaling a Multifrontal Solver

Bob Lucas * ¹, Roger Grimes , Clement Weisbecker

¹ Livermore Software Technology Corporation (LSTC) – 7374 Las Positas Road,
Livermore, CA 94550, United States

The end of Dennard scaling in the last decade has led to an exponential growth in the number of processor cores being used to solve large systems of linear equations in science and engineering. Direct methods are often required, and evolving a multifrontal code to keep pace with the increasing number of cores, and the changes within them, has been challenging. This talk discusses some of the transformations that have been required. Reordering is now often the computational bottleneck, and is increasingly avoided, even if that leads to less efficient factorizations and triangular solves. With the increasing width of SIMD extensions to microprocessors (e.g., AVX), use of vendor-tuned functions such as DGEMM is increasingly important. Memory bandwidth is not keeping up the processing power of multicore processors, requiring rethinking fundamental data structures to minimize data movement. Message passing is required to scale beyond one shared memory system, while multithreading is needed to avoid local memory bottlenecks on individual MPI ranks. Tiled frontal matrices have had to be introduced for factorization to scale beyond $O(100)$ cores. Finally, because processing performance is scaling faster than memory volume, we are introducing block low-rank approximations to reduce storage required as well as operations performed.

Keywords: sparse matrix solver, parallel processing

*Speaker

Efficient parallelization of direct solvers for isogeometric analysis

Maciej Paszynski * ¹, Grzegorz Gurgul ², Daniel Garcia ³,
David Pardo ³

¹ AGH University of Science and Technology [Pologne] – 30 Mickiewicza Av.
30-059 Krakow, Poland

² AGH University of Science and Technology [Pologne] – 30 Mickiewicza Av.
30-059 Krakow, Poland

³ Basque Center for Applied Mathematics (BCAM) – Alameda Mazarredo, 14,
48009 Bilbao, Bizkaia, Spain, Spain

Isogeometric analysis (IGA) is a modern computational method introduced by Hughes et al. [1]. The method utilizes either tensor product based grids with B-spline basis functions, or adaptive 3D grids with T-spline basis functions (other variations also exist). While IGA produces high-accuracy approximations of the solutions of a large variety of partial differential equations, it also increases the computational cost per unknown of the multi-frontal direct solver (MFDS) [2,3,4,5].

In this presentation, we propose a new approach for the optimization of sequential and parallel direct solver algorithms for IGA. The main idea is to introduce some new basis functions that decouple the already existing basis functions, and as a result, make the global matrix larger but more disconnected.

We have analyzed this method for the case of sequential MFDS executed over uniform IGA grids in [6], showing the superior results of our proposed approach. In this presentation, we discuss the performance of parallel MFDS for IGA refined with additional basis functions.

To illustrate this approach we present two numerical examples. The first one is dedicated to large uniform grids, and the second one is devoted to adaptive grids resulting from the discretization of singular solutions. In both cases, we observe that our enriched systems can be solved using a direct solver (both sequentially and in parallel) in a fraction of the time required to solve the original IGA system. Specifically, the new system can often be solved in less than 10% of the time needed to solve the original IGA system. Moreover, since our systems are built by enriching the original IGA spaces, our solutions also exhibit an smaller best approximation error.

*Speaker

Acknowledgement. This work is supported by National Science Centre, Poland grant no. DEC-2015/17/B/ST6/01867

- [1] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer methods in applied mechanics and engineering* 194(39) (2005) 4135-4195.
- [2] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, V.M. Calo, The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213-216 (2012) 353-361
- [3] M. Wozniak, M. Paszynski, D. Pardo, L. Dalcin, V.M. Calo, Computational cost of isogeometric multi-frontal solvers on parallel distributed memory machines *Computer Methods in Applied Mechanics and Engineering*, 284 (2015) 971-987
- [4] M. Wozniak, K. Kuznik, M. Paszynski, V.M. Calo, D. Pardo Computational cost estimates for parallel shared memory isogeometric multi-frontal solvers, *Computers and Mathematics with Applications*, 67(10) (2014) 1864-1883.
- [5] M. Paszynski, *Fast Solvers for Mesh-Based Computations*, Taylor & Francis, CRC Press (2016)
- [6] D. Garcia, D. Pardo, L. Dalcin, M. Paszynski, D. Pardo, V.M. Calo, Refined Isogeometric Analysis (rIGA): Fast Direct Solvers by Controlling Continuity, submitted to *Computer Methods in Applied Mechanics and Engineering* (2016).

Keywords: isogeometric finite element method, parallel multifrontal direct solvers

Dissection : A direct solver with kernel detection for finite element matrices on multi-core supercomputers

Atsushi Suzuki ^{*† 1}, François-Xavier Roux ²

¹ Cybermedia Center, Osaka University – Machikaneyama, Toyonaka, Osaka, 560-0043, Japan

² Laboratoire Jacques-Louis Lions (LJLL) – INRIA, Université Paris VII - Paris Diderot, CNRS : UMR7598, Université Pierre et Marie Curie (UPMC) - Paris VI – B.C. 187 75252 Paris Cedex 05, France

A direct solver for large-scale finite element matrices is developed. The unsymmetric sparse matrix is assumed to be structurally symmetric with non-zero entries and to have an LDU factorization with a symmetric partial pivoting. It could be indefinite and/or non-invertible. During symbolic factorization phase, the matrix is decomposed into mixture of sparse and dense sub-matrices by a nested bisection, which is realized by METIS or SCOTCH graph partitioning library. Numeric factorization phase consists of block strategy to use level 3 BLAS routines efficiently.

The sparse sub-matrix is reordered into a block tridiagonal structure with variable size by reverse Cuthill-McKee ordering. The dense sub-matrix is decomposed into a union of blocks with fixed size 'b' to use rank-b update during the factorization procedure. However, this block strategy affects pivoting procedure deeply because search range of diagonal entries to find the largest absolute value is limited. Actually the sub-matrix may not be factorized only with 1x1 block of a partial symmetric pivoting. As a remedy, a given threshold is used to introduce postponing computation concerning suspicious null pivots. During factorization of a block, when a diagonal entry becomes smaller than the previous one with ratio given by the threshold, factorization is terminated and the factorization of reset of the entries is postponed.

The Schur complement is constructed from these suspicious null pivots and is examined by a factorization with 1x1 and 2x2 pivoting and by a robust kernel detection algorithm based on measurement of residuals with orthogonal projections onto supposed image spaces. The kernel detection algorithm

*Speaker

†Corresponding author: Atsushi.Suzuki@cas.cmc.osaka-u.ac.jp

can resolve rank-deficiency without computing eigenvalues nor singular values. It is only based on LDU factorization but it uses higher precision arithmetic, e.g. quadruple precision, to distinguish numerical round-off errors that occurred during factorization of the whole sparse matrix from ones during the kernel detection procedure itself. This algorithm is a direct extension of symmetric one in DOI:10.1002/nme.4729. Matrices by finite element discretization for elasticity and flow problems usually have small size of the kernel. Hence the cost of this kernel detection does not deteriorate parallel performance.

At the top of the bisection tree, the factorization of a dense matrix needs to be parallelized, especially in multi-core environment. The established techniques of construction of a task-dependency tree is used. Since our task-dependency tree of factorization is rather simple, the critical path of each dissection level is easily found by a heuristic way. The tasks are scheduled in a static way with some remained dynamic parts to reduce load imbalance due to under- or over-estimated complexity of actual implementation of BLAS libraries and some environmental noise from processes of the operating system. This technique of asynchronous task execution can reduce idle time of processors drastically. Dissection code is written by C++ and POSIX threads library is used to realize asynchronous execution of tasks on shared memory architecture. A static data structure analyzed during symbolic phase can use level 3 BLAS routines efficiently. Usage of optimized sequential BLAS library can hide machine dependency and as a result, the code runs on both a super-scalar CPU with large cache memory and a modern vector CPU. NEC SX-ACE has 4 vector cores, where peak performance of each vector core running at 1GHz clock is 64GFlop/s. By using level 3 BLAS library, e.g. DTRSM and DGEMM, it needs not to write vector directives explicitly. Dissection has competitive performance to Intel Pardiso on Intel Xeon processor. It achieves twice faster computation on SX-ACE than IvyBridge Xeon processor. Detailed numerical results by Xeon multi-core processor and by SX-ACE vector processor will be reported for some practical finite element matrices from three-dimensional flow problems.

Keywords: Direct solver, sparse matrix, finite element, rank, deficiency, asynchronous parallelization, multi, core computation

CP - Applications

Optimizing linear operators within FETI DDM

Václav Hapla ^{*† 1}, David Horák ¹, Jakub Kružík ¹

¹ IT4Innovations National Supercomputing Center, VSB - Technical University of Ostrava (IT4I) – Czech Republic

Our team belongs to the research programme HPC Libraries and Supercomputing for Industry at IT4Innovations National Supercomputing Center (Czech Republic). The team is focused on developing highly scalable algorithms for the solution of linear and non-linear problems arising from different engineering applications. One of the main application areas is structural mechanics. As the fundamental parallelization technique, domain decomposition methods (DDM) of the FETI type are used. For solving the resulting discretized and decomposed problems, linear system or quadratic programming (QP) solvers are employed in case of linear elasticity or contact problems, respectively. These methods are implemented within our software projects ESPRESO and PERMON, which have shown parallel scalability up to tens of thousands of cores and billions of unknowns. In this talk, we focus on optimized implementations of several special linear operators used within the FETI methods using besides others the communication hiding and communication avoiding (CA/CH) techniques, developed within the EXA2CT (EXascale Algorithms and Advanced Computational Techniques) project, belonging to European Exascale Projects.

Keywords: PERMON, ESPRESO, domain decomposition, DDM, FETI, linear operator, communication avoiding and hiding, CA/CH

*Speaker

†Corresponding author: vaclav.hapla@vsb.cz

Development of a Multi-physics Code with Adaptive Mesh Refinement

William Dai ^{*† 1}

¹ Los Alamos National Laboratory (LANL) – Los Alamos National Laboratory
Mail Stop T082 Los Alamos, NM 87545 USA, United States

Typical Eulerian methods have difficulty in keeping sharp material interfaces, leaving questions as to what is correct physical behavior or artifact of numerical methods. This sometimes limits the Eulerian methods' usefulness for certain types of calculations. With adaptive mesh refinement (AMR), and treatment of material interfaces, the range of calculations for which the Eulerian approach can be applied has the potential to be greatly expanded. For this purpose, we have been working on the Roxane project for seven years. The Roxane project covers a variety of physics in standard geometries of one, two, and three dimensions, including hydrodynamics, volume fraction material advection, material mixing, elastic-plastic strength models, magnetohydrodynamics, 3-T radiation diffusion, detonation shock dynamics, HE burn models, etc.

To best resolve material interfaces, we refine any mixed cell as well as any pure cell whose neighboring cell contains different materials. In the treatment of multi-materials, it is often found that a major fraction of computer memory is used to store the zeros of material data in large simulations. Because of this, it is difficult to run large simulations with many materials. To overcome this difficulty, we have developed a compressed material data structure in the project, so that only non-zero material values are stored in memory.

To reduce the number of communication in parallel computer environments, we combine all the communications at the beginning of each time step. To meet the need of different numerical algorithms, the number of ghost cells surrounding the part of mesh of each computer processor could be changed at any time during a simulation.

*Speaker

†Corresponding author: dai@lanl.gov

In this presentation, in addition to AMR, parallel strategy, and IO package, we will particularly describe a numerical approach for solving nonlinear 3-T radiation diffusion equations. Although AMR can reduce the number of materials within a mixed cell, it cannot eliminate mixed cells in simulations. Treatment of mixed cells is critical for many problems. Accurate estimate of the temperature of each material within mixed cells is very important for the calculation of certain subsequent physics packages. Different temperatures of the materials within mixed cells may result in very different outcomes. Furthermore, although tables of equation of state (EOS) of pure material are available, tables for mixture of material are not available. Therefore, to better resolve the sub-cell structure, we decompose each of mixed cells to a set of sub-cells in two and three dimensions through material interface reconstruction so that each sub-cell contains only one material. The sub-cells thus generated are general polygons two dimensions and polyhedrons in three dimensions.

3-T Radiation diffusion equations are solved on these general polygons and polyhedrons. For systems of multi-materials with dramatically different material properties, the correct treatment for the discontinuity of material properties is important. We use the governing physics to obtain the effective diffusion coefficient across a material interface for flux calculations on polyhedral meshes. For a system with dramatically different materials, formulations good for steady states are important even for time-dependent problems. This often challenges the second order accuracy (in time) of numerical schemes. We applied a rare time-stepping technique to have the both properties. Another important aspect in numerical simulations for 3-T radiation equations is the numerical treatment for interaction between radiation and material. The 3-T radiation diffusion equations are often solved through operator splitting. In our approach radiation and material are fully coupled, and three temperatures are updated simultaneously. The numerical scheme to be presented is fully nonlinear. From the scheme to be presented, we will show how well or bad some other less perfect schemes, such as operator-splitting, linearization, incorrect treatments of material discontinuity, and first order accuracy.

Keywords: nonlinear, radiation, diffusion, AMR, interface

Hybrid programming in the legacy code AVBP using OMPs

Cyril Fournier ^{*† 1}, Gabriel Staffelbach ¹, Isabelle D'ast ¹

¹ Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS) – CERFACS – France

The AVBP code, co-developed by CERFACS and IFPEN, is the state of the art parallel CFD Navier-Stokes solver for reactive compressible flows based on 3D fully unstructured meshes. It is widely used for large eddy simulation of combustion in aeronautical engines, rockets and piston engines. Its flat MPI implementation scales up-to 296 kcores on the BlueGene Q platform.

However, the increasing computational capability on the emerging many-core node requires to revisit the parallel paradigm to cope with the hierarchical architectures. In this talk, we will discuss the choices made to move towards hybrid MPI+X implementation. More particularly, we will detail how OmpSs, developed by BSC (Barcelona Supercomputing Center) has been introduced and analyze the achieved performance in a complex CFD simulation.

Keywords: MPI, OmpSs, Task, based programming, Manycore

*Speaker

†Corresponding author: cyril.fournier@cerfacs.fr

High Intensity Matrix Polynomial Solvers for the Heat and Poisson Equations.

Thierry Dumont * ¹, Thomas Guillet , Raphael Prat ,
Violaine Louvet

¹ Institut Camille Jordan, CNRS UMR 5208. (ICJ) – Université Claude Bernard-Lyon I - UCBL (FRANCE) – Université Lyon 1. 69622 Villeurbanne cedex., France

Heat equation and Poisson equations are basic blocks of many numerical methods for partial differential equations (PDE). These two equations, which could be considered as simple are actually numerical bottlenecks in many applications like fluid mechanics, plasma physics and so on, as obtaining fast solvers is always challenging, at least in dimension 3.

We will show that fast and precise parallel solvers are obtained when two conditions are fulfilled: 1) use explicit high order stabilized methods, 2) perform arithmetic intensive matrix vector products obtained from high order discretizations.

Concerning the heat equation (and more generally parabolic PDE's), there is a growing interest since some years in explicit stabilized Runge-Kutta (RK) methods, particularly those known as Rock methods. Being explicit RK methods, Rock methods reduce to the application at each time step of a matrix polynomial when the parabolic PDE is linear; they can be used if the discrete matrix has a real negative spectrum of eigenvalues, like in the case of the heat equation. Then, compared to the classical implicit approach, the linear algebra is much simplified, and the methods are parallel, which is a first necessary, but not sufficient condition to obtain performing applications. To build fast solvers, one must also be able to combine high arithmetic intensity matrix vector product with an unexpensive spatial discretizations of the PDE: for this the growingly popular Discontinuous Galerkin (DG) methods are a natural choice.

In the first part of our talk we will describe the Rock methods and the DG discretization we have used, and show the performances of the code we have developed for the heat equation in dimension 3. Explicit stabilized RK methods share a common history with Chebyshev methods for the solution of linear systems and Chebyshev preconditioners for iterative methods. In

*Speaker

the second part of our talk, after recalling briefly this history, we will present our numerical experiments on the Poisson equation, in dimension 3, using Conjugate Gradient, Chebyshev based preconditioning and a well chosen Discontinuous Galerkin based discretization, which is a key to obtain a high arithmetic intensity, together with a high order approximation.

Keywords: Poisson equation, heat equation, stabilized explicit methods, Matrix polynomial, arithmetic intensity

CP - Eigensolver

Approximating Functionals of Hermitian Matrix Product Operators via a Block Lanczos Method

Moritz August ^{*} ¹, Mari Carmen Bañuls ², Thomas Huckle ¹

¹ Technical University of Munich (TUM) – Technische Universität München
Arcisstrasse 21 80333 München Germany, Germany

² Max-Planck-Institut für Quantenoptik (MPQ) – Hans-Kopfermann-Str. 1
Garching, Germany

We present a method to approximate functions $\text{Tr}_f(A)$ of very high-dimensional hermitian matrices A represented as Matrix Product Operators (MPOs). Our method is based on a reformulation of a block Lanczos algorithm in tensor network format. We state important properties of the method and show how to adapt the basic Lanczos algorithm to the tensor network formalism to allow for high-dimensional computations. Additionally, we give an analysis of the complexity of our method and provide numerical evidence that it yields good approximations of the trace-norm and the entropy of MPOs while being robust towards truncations.

Keywords: Lanczos, Tensor Networks, Matrix Product Operators, Big Data, High, Dimensional, Quantum Information

^{*}Speaker

ChASE: Chebyshev Accelerated Subspace iteration Eigensolver library on heterogeneous architectures

Edoardo Di Napoli * ^{1,2}, Jan Winkelmann ³, Paul Springer

¹ Juelich Supercomputing Centre – Forschungszentrum Jülich GmbH (JSC) –
Wilhelm-Johnen-Straße 52428 Jülich, Germany

² Juelich Aachen Research Alliance – High-performance Computing
(JARA-HPC) – Schinkelstrasse 2, 52062 Aachen, Germany, Germany

³ Aachen Institute for Advanced Study in Computational Engineering Science -
RWTH Aachen (AICES - RWTH) – Schinkelstrasse 2, 52062 Aachen, Germany,
Germany

We propose to step away from the black-box approach and allow the eigensolver to accept as much information as it is available from the application. Such a strategy implies that the resulting library is tailored to the specific application, or class of applications, and loose generality of usage. On the other hand, the resulting eigensolver maximally exploits knowledge from the application and become very efficient. With this general strategy in mind, we present here a version of a Chebyshev Accelerated Subspace iteration Eigensolver (ChASE) which targets extremal eigenpairs of dense eigenproblems. In particular, ChASE focuses of on a class of applications resulting in having to solve sequences of eigenvalue problems where adjacent problems possess a certain degree of correlation. A typical example of such applications is Density Functional Theory where the solution to a non-linear partial differential equation is worked out by generating and solving dozens of algebraic eigenvalue problems in a self- consistent fashion over dozens of iterations. Similarly, any non-linear eigenvalue problem, which can be solved by the method of successive linearization, gives rise to sequences of correlated algebraic eigenproblems that are the target of ChASE. We re-design the eigensolver so as to minimize its complexity and have better control of its numerical features. Following the algorithm optimizations, we strive to adopt a strategy leading to an implementation that would lends itself to high-performance parallel computing and avoid, at the same time, issues related to portability to heterogeneous architectures. We achieve such a goal

*Speaker

by implementing parallel kernels for the modular tasks of the eigensolver using programming strategies out of MPI, OpenMP, and CUDA.

Keywords: Iterative eigensolver, dense eigenproblems, heterogeneous computing platforms, optimization, Density Functional Theory

CP - High performance computing

Increasing Arithmetic Intensity using Stencil Compilers on Many-Core Architectures

Simplicio Donfack *¹, Patrick Sanan², Olaf Schenk³,
Bram Reys⁴, Wim Vanroose⁴

¹ Inria – L’Institut National de Recherche en Informatique et en Automatique
(INRIA) – France

² USI Lugano, Institute of Computational Science (USI) – Switzerland

³ USI Lugano, Institute of Computational Science (USI) – Institute of
Computational Science Università della Svizzera italiana Via Giuseppe Buffi 13
6900 Lugano, Switzerland

⁴ University of Antwerp – Belgium

In time-stepping methods kernels of subsequent matrix vector multiplications can increase the arithmetic intensity. By tiling the computational domain, or in general, reorganizing loops, multiple applications of the same operation can be executed per memory read. Many Krylov subspace solvers for linear systems of equations are similarly based on a single multiplication of a sparse matrix A with a vector v . Inspired by polynomial preconditioning, we increase the arithmetic intensity of this Krylov subspace building block by replacing matrix A with a higher-degree matrix polynomial $p_m(A)$. This allows for a better use of vector units and as a consequence shows better speed-ups on the latest hardware. As the number of Krylov subspace iterations required for convergence increases with problem size, and the time for each global collective operation increases with machine scale, Krylov subspace solves in large-scale applications can constitute a significant fraction of the overall time. In this talk, we demonstrate the impact of stencil compilers and polynomial matrix Krylov subspace iterations and implement, evaluate, and optimize a high-arithmetic intensity formulation of CG as a high-performance, distributed-memory Krylov solver.

Keywords: stencil compilers, arithmetic intensity, Krylov subspace solvers

*Speaker

A Parallel Algorithm for Scalable Bloom Filters Matrix

Zhu Wang * ¹

¹ Data Communication Technology Research Institute (DCTRI) – China

Bloom filters are used to support membership query of large scale dataset. The algorithm can be very efficient in time and space and introduces a very small false positive rate. Scalable Bloom filters can further be applied in the case in which the size of the dataset is not stable, but increases by time. When the dataset is large, scalable Bloom filters form a matrix that represents the items in the set. In this paper, we present a parallel algorithm for querying items in the scalable Bloom filters matrix (SBFM). We analyse the cost of each Bloom filter lookup and find the false positive rate of the Bloom filters in the SBFM. Through experiment evaluation, we conclude that the most time consuming procedure of the algorithm is the calculation of hash positions. With the deduction of the execution probability in the bloom filters, we define the sequence of the threads when calculating hash positions. Then the entire lookup process is shown in the paper. We further use theoretical analysis to find the theoretical performance of the algorithm. Then experiments are executed to verify our theoretical results. The experiments proves to be in good accordance with the theoretical analysis.

*Speaker

Parallel solver for shifted systems

Nela Bosner ^{*† 1}, Zvonimir Bujanović ¹

¹ Department of Mathematics, Faculty of Science, University of Zagreb –
Bijenicka cesta 30, 10000 Zagreb, Croatia

We propose a combination of a hybrid CPU-GPU and a pure GPU algorithm for solving shifted linear systems with multiple right sides, for a large number of shifts. Such problems appear in control theory when evaluating the transfer function or as a part of an algorithm performing interpolatory model reduction, as well as when computing numerical solution of a large linear system of ODE's.

The new algorithm for solving systems of the form $(A - \sigma I)X = B$, for many different $\sigma \in \{C\}$ simultaneously, consists of two phases. In the first phase, we reduce the generally full system matrix $A \in \mathbb{R}^{n \times n}$, and the full right-hand side matrix $B \in \mathbb{R}^{n \times m}$, to a suitable form, which enables us to solve the systems with far less computational effort. This reduction is done only once, regardless of the number of shifts: A is transformed to a so-called m -Hessenberg form and B is made upper-triangular. For the first transformation, we introduce a highly parallel CPU-GPU hybrid algorithm. The algorithm is blocked; individual blocks are being reduced by the CPU, and the necessary updates of the rest of the matrix are split among many cores of CPU and GPU. To enhance parallelization, the reduction and the update computation is overlapped.

In the second phase, the reduced m -Hessenberg-triangular systems are repeatedly being solved for given batches of shifts. This solver is implemented entirely on the GPU, and it annihilates the m subdiagonals of the system matrix simultaneously for all shifts in the batch. The most demanding part of this algorithm are the RQ factorizations of many m -Hessenberg matrices independently. Hence, each factorization is run by a different block of threads, while the updates mostly rely on cuBLAS routines.

Benefits of such load distribution are demonstrated by numerical experiments: on our platform, both parallel algorithms outperform their CPU-bound counterparts by the factor of 3.5 for larger dimensions.

Keywords: GPU algorithms, shifted systems, m , Hessenberg reduction

*Speaker

†Corresponding author: nela@math.hr

**MS - Parallel SVD/EVD
solvers for large dense
matrices**

Computation of the CS and the indefinite CS decomposition

Sanja Singer * ¹

¹ University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture
– Ivana Lučića 5, 10000 ZAGREB, CROATIA Phone:)(+385 1) 6168 215 Fax:
(+ 385 1) 6156 940, Croatia

Accurate computation of the cosine-sine (CS) decomposition can be a step forward to make block one-sided Jacobi method significantly faster. Instead of ordinary, elementwise one-sided Jacobi algorithm, fast block version of the algorithm relies on the accurate CS computation. Similarly, the same is valid for the indefinite cosine-sine (JCS) decomposition, as the part of the one-sided hyperbolic block-Jacobi algorithm.

Computation of both, CS and JCS, can be organized as two SVDs (computed in parallel), of either diagonal or off-diagonal blocks, followed by the appropriate QR factorizations.

Joint work with Vedran Novaković.

Keywords: cosine'sine decomposition, SVD, parallel computing

*Speaker

Accelerating the SVD using GPUs

Mark Gates ^{*† 1}, Azzam Haidar ¹, Stan Tomov ¹, Jack Dongarra ^{1,2,3}

¹ Innovative Computing Laboratory, University of Tennessee (ICL) – Suite 203
Claxton, 1122 Volunteer Blvd, Knoxville, TN 37996, United States

² University of Manchester, Manchester – United Kingdom

³ Oak Ridge National Laboratory (ORNL) – Oak Ridge, TN, United States

The increasing gap between memory bandwidth and computation speed necessitates developing new algorithms to take full advantage of today's high performance computers. For dense matrices, the classic algorithm for the singular value decomposition (SVD) uses a one stage reduction to bidiagonal form, which is limited in performance by the memory bandwidth. We turn instead to a two stage reduction that first reduces the matrix to band form using high performance Level 3 BLAS, then reduces the band matrix to bidiagonal form using optimized, cache-friendly kernels with dynamic scheduling. This removes the memory bandwidth limitation, decreases communication and synchronizations, and increases the computational intensity. While the number of floating point operations increases, this increase is offset by the more efficient execution and reduced data transfers. Moreover, we take advantage of accelerators, such as GPUs and coprocessors, to further improve performance. We accelerate all stages of the SVD – QR factorization, reduction to bidiagonal, bidiagonal SVD using divide-and-conquer, and back-transformation of singular vectors – providing a significant speedup compared to existing multi-core and GPU-based SVD algorithms.

Keywords: Singular value decomposition, SVD, GPU

*Speaker

†Corresponding author: mgates3@icl.utk.edu

Asymptotic Quadratic Convergence of the Parallel Block-Jacobi EVD Algorithm for Hermitian Matrices

Gabriel Oksa * ¹, Yusaku Yamamoto[†] ², Marian Vajtersic[‡]
₃

¹ Institute of Mathematics, Slovak Academy of Sciences – Bratislava, Slovakia

² Dept. Communication Engineering and Informatics, The University of
Electro-Communications, – Tokyo, Japan

³ Dept. Computer Sciences, University of Salzburg – Austria

We provide the proof of the asymptotic quadratic convergence of the parallel block-Jacobi EVD algorithm for Hermitian matrices with well-separated eigenvalues (including the multiple ones) as well as clusters of eigenvalues. Individual 2×2 block subproblems are chosen using the parallel dynamic ordering (its greedy implementation).

Keywords: Block Jacobi method, parallel dynamic ordering, Hermitian matrix, asymptotic quadratic convergence

*Speaker

[†]Corresponding author: yusaku.yamamoto@uec.ac.jp

[‡]Corresponding author: marian@cosy.sbg.ac.at

High-Performance Parallelization Method of DSYRK for SVD and other Matrix Computations on Xeon Phi

Shuhei Kudo ^{*† 1}, Yusaku Yamamoto ¹

¹ The University of Electro-Communications – 1-5-1 Chofugaoka, Chofu, Tokyo
182-8585, Japan

In this talk, we discuss parallelization methods of a BLAS routine DSYRK. DSYRK is a variant of matrix multiplication which calculate $C = AA^T$, where A is an $m \times n$ real matrix. It is frequently used in matrix computations such as the singular value decomposition. Since C is a symmetric matrix, the computation of the upper or lower triangular part of C can be skipped. On the other hand, when parallelizing DSYRK, it is more difficult to achieve good load balance than in the case of standard (non-symmetric) matrix multiplication because the divided computational regions have more irregular shapes. This becomes more problematic on recent high-performance many-core architecture CPUs which have much larger degree of parallelism. We therefore implemented DSYRK using a couple of parallelization methods for Xeon Phi (Knights Corner) and analyzed the performance results. In this talk, we will describe the implementation details and the results of the performance analysis. Performance comparisons with other existing implementations are also presented.

Keywords: Xeon Phi, DSYRK, Level, 3 BLAS, Knights Corner

*Speaker

†Corresponding author: shuhei-kudo@outlook.jp

Preprocessing Algorithm for the Generalized SVD on the Graphics Processing Units

Goran Flegar ^{*† 1}

¹ University of Zagreb, Faculty of Science, Department of Mathematics –
Bijenička cesta 30, 10000 Zagreb, Croatia

We describe a parallelization of the preprocessing step of Page's algorithm for the generalized singular value decomposition on the graphics processing units. A blocking algorithm that takes advantage of the GPU's architecture is used to compute the required QR decompositions. A similar algorithm is employed to compute the QR decompositions with column pivoting by using a tournament selection scheme to select blocks of linearly independent columns.

Finally, the results of numerical tests performed on our algorithm are presented and the algorithm is compared to the Lapack xGGSVP routines.

Keywords: GPGPU, Generalized SVD, QR Decomposition, Column Pivoting, Complete Orthogonal Decomposition, Tournament Selection

*Speaker

†Corresponding author: gflegar@student.math.hr

ELPA: Algorithmic extensions and optimization

Michael Rippl * ¹

¹ Fakultät für Informatik – TU München ; Lehrstuhl für wissenschaftliches Rechnen ; Boltzmannstr. 3 ; D-85748 Garching, Germany

The solution of symmetric eigenproblems plays a key role in many computational simulations. Especially in ab-initio molecular dynamics it is by far the most expensive part.

ELPA is one of the leading libraries in the computation of parts or the whole eigenspektrum on large scale compute clusters. The project finished in 2011 and some progress was still going on till now. Based on comparisons with other parallel solvers we will discuss future improvements of the ELPA code. A further goal is to exploit the computing power of modern heterogeneous architectures. Additionally we will present further algorithmic ideas that might improve the performance and the applicability of the ELPA library.

*Speaker

An accelerated tridiagonal divide-and-conquer algorithm on distributed memory architectures

Shengguo Li ^{*† 1}, Shuliang Lin ², Francois-Henry Rouet ³,
Liu Jie ², Huang Chun ², Gao Xingyu ⁴

¹ National University of Defense Technology (CHINA) (NUDT) – China

² National University of Defense Technology (NUDT) – China

³ Lawrence Berkeley National Laboratory – United States

⁴ Institute of Applied Physics and Computational Mathematics - IACM (Beijing, China)) – China

Efficient and reliable rank-structured matrix computations have been an intense area of recent research. Rank-structured matrices (also referred to as data sparse matrices) have numerically low-rank off-diagonal blocks. Detecting and compressing these blocks, typically using a rank-revealing factorization, is the key to designing efficient algorithms for a wide range of applications. Rank-structured techniques have been used to solve integral equations (more specifically, dense linear systems arising from the Boundary Element Method), computing the roots of polynomials, developing sparse direct solvers and preconditioners for matrices arising from the Finite Element Method, solving eigenvalue and singular value problems, etc. Several classes of rank-structured matrices have appeared in the literature, e.g., H and H^2 matrices, and Hierarchically Semi-Separable matrices (HSS), among others. In this talk, we investigate using HSS matrices to accelerate the Tridiagonal Divide-and-Conquer (TDC) algorithm used in ScaLAPACK to solve eigenvalue problems. We compare our accelerated version not only against the version in ScaLAPACK, but also against the algorithms implemented in the ELPA library (Eigenvalue solvers for Petascale Applications).

The central idea of our work is that some intermediate eigenvector matrices that appear in the TDC algorithm are rank-structured, and they can be approximated accurately and efficiently using HSS representations. The main cost of the TDC algorithm is in computing eigenvectors, which relies on matrix-matrix multiplication. Using HSS representations for intermediate matrices, standard matrix-matrix multiplication is replaced with fast HSS multiplications. Recently, the authors proposed to use HSS matrices to accelerate the TDC algorithm implemented in LAPACK. Experiments

*Speaker

†Corresponding author: nudt1sg@nudt.edu.cn

on shared-memory systems showed significant speed-ups with respect to the version of MKL implemented within Intel MKL (up to 6x). The authors also extended this idea to bidiagonal and banded Divide-and-Conquer algorithms for SVD calculations.

In this work, we extend the techniques used in our previous works to the distributed memory environment, and we accelerate the TDC algorithm in ScaLAPACK. We rely on the STRUMPACK library (STRUctured Matrices PACKage) package to handle HSS matrix construction and HSS matrix multiplications. STRUMPACK implements dense and sparse direct solvers and preconditioners. It relies on HSS techniques based on randomized sampling. The dense component of STRUMPACK is built on top of BLACS and ScaLAPACK, and the interface for the HSS construction and HSS matrix-matrix product features is very similar to that of ScaLAPACK. In the talk, we will describe in detail how to change ScaLAPACK routines to explore the low-rank structure, and how to use STRUMPACK to update the eigenvectors.

Numerous numerical results have been performed to compare the proposed Parallel Accelerated DC (PADC) algorithm against the version in ScaLAPACK. All the results were obtained on the Tianhe 2 supercomputer. It turns out that PADC is faster than the version of ScaLAPACK implemented in Intel MKL for some matrices with few deflations using hundreds of processes. We also compared PADC with ELPA and obtained similar speed-ups. With our current implementation, we obtain speed-ups over 2x against both libraries, but gains tend to disappear when using more than a few hundred processes. We will address this issue in the presentation and we will suggest ways of improving both PADC and STRUMPACK.

Keywords: Eigenvalue, Divide and Conquer, Scalapack, Distributed memory architecture

GPU-hosted SVD computations for large scale hierarchical matrices

George Turkiyyah *^{1,2}, Wajih Boukaram¹, Hatem Ltaief¹,
David Keyes¹

¹ KAUST – Saudi Arabia

² American University of Beirut – Lebanon

Large dense matrices are ubiquitous in scientific computing but present challenges in extreme computing environments because of their prohibitive memory requirements. However, many of these dense matrices that arise in science and engineering applications have a structure consisting of blocks, of various sizes, that have relatively low numerical rank. This structure can be systematically exploited to approximate a dense matrix more compactly, in a hierarchical and accuracy-controllable manner. The effective compression of these hierarchical matrices is needed both during their initial generation and when performing matrix operations on them. The compression is performed with generalized SVD procedures adapted to the structure of hierarchical matrices. In this work, we describe algorithms for hierarchical SVDs and show their implementation on single and multi GPU systems. The operations involved in these algorithms include randomized and batched SVDs of small dense blocks, as well as batched GEMM and QR operations, all performed at multiple levels of granularity. The algorithms have optimal complexity in the matrix size and result in representations that requires storage only linear in the matrix dimension. The algorithms are algebraic and do not necessarily require knowledge of the underlying analytical properties of the problem from which the matrix originated. In addition, matrix-vector multiplication and other BLAS-operations can be performed directly on the compressed representations in an efficient manner as our demonstration results show.

*Speaker

**MS - Advanced topics for
EVP and SVD towards
future scientific simulation**

Parallel dense eigenvalue solver and SVD solver for post-petascale computing systems

Toshiyuki Imamura * ¹

¹ RIKEN Advanced Institute for Computational Science (RIKEN AICS) – 7-1-26
Minatojima-minimi-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan

EigenExa is a parallel dense eigenvalue solver toward post-petascale supercomputer systems. We have developed the EigenExa library on the K computer, which has more than 10 PFLOPS computational power and is currently number one system in Japan, since 2012. The library is already ported and released as Open Source Software not only for the K-computer but several supercomputer platforms. With the scientific collaboration between computer centers, we support EigenExa currently on six hardware platforms, Fujitsu SPARC64 (VIIIfx, IXfx, and XIfx), IBM BlueGene/Q, NEC SX-ACE, and Intel x86-64 general processors. The library is taken advantage of by typical modern numerical simulation fields such as quantum chemistry, material science, drug design, structure analysis and big data science. By using the full system of the K computer (82.944nodes, 663.552cores, totally), we examined the feasibility of the numerical algorithm and parallel implementation up to a ultra-scale diagonalisation with a one million dimensional matrix. The test run was done within approximately one hour for diagonalization and another one hour for a validation check. We examined that residual error and orthogonal error (defined by $\max \|Ax - \lambda x\|/N\|A\|$ and $\|XX^{\dagger} - I\|/\sqrt{\{N\}}$, respectively) are acceptable values less than 10^{-10} . Even, the matrix size is one million; still we observed that there is room for the optimization by detailed analysis.

The most significant point to improve the performance is to reduce or hide the communication overhead accumulated through running the EigenExa library. This point to cover up the overhead behind computation is a weak technique because it is only applicable to large problems but the smaller case is unhelpful. We recognized this situation and introduced an idea of communication avoidance. We modified the conventional Dongarra-Sorensen Householder tridiagonalization algorithm. Communication Avoidance reduces the number of issues of collective communications, especially MPI_Allreduce's which calculate the inner products of a set of vectors. This modified solver results in a real performance improvement by the time, in

*Speaker

fact, we can reduce up to 25 percent of the elapsed time. This idea can be applied to the block Householder tridiagonalization algorithm, which is adopted in most of the modern eigenvalue solvers as well as the EigenExa library. For the design of communication-avoiding block Householder tridiagonalization, we need to investigate the new algorithms to generate a block-Householder reflector by Y.Yamamoto and G.Ballerd. The algorithm is based on QR factorization, which we can choose appropriate parallel implementation with respect to the communication cost. In fact, we have selected the CholeskyQR2 algorithm in the current release. The most recent version of the EigenExa library supports two types of eigenvalue problems; the standard type and the generalized type. Communication avoidance is also able to apply the bidiagonalization for singular value decomposition. As presented in the tridiagonalization, we can expect to improve the performance of the most time-consuming part in SVD calculation.

Towards the post-petasclae computing era, we also investigate the efficient utilization of accelerator, such as GPU, MIC, and FPGA. Communication avoidance and adapting the emerging future hardwares will be presented in the mini-symposium.

Keywords: eigenvalue solver, dense matrix, parallel computer, post, petascale computing

A real-valued method for solving complex symmetric linear systems arising in contour integral eigensolver

Yasunori Futamura * ¹, Yuto Inoue ¹, Tetsuya Sakurai ¹

¹ University of Tsukuba – Japan

In this study we consider real-symmetric definite generalized eigenvalue problems and consider computing interior eigenvalues and corresponding eigenvectors. Real-symmetric definite generalized eigenvalue problems arise in various fields in scientific computing such as electronic structure calculations and vibration analysis using the finite element method.

In this study, we propose a real-valued method for solving a certain type of complex symmetric linear system arising in the algorithms of contour integral eigensolvers.

Contour integral eigensolvers such as the Sakurai-Sugiura method and the FEAST algorithm have been developed in the last decade and have recently attracted attentions because of their coarse-grained parallelism. This parallelism comes from the fact that linear systems solved in the algorithm of the eigensolvers are independent of each other. However, the coefficient matrices of the linear systems are complex symmetric even if the matrices of the generalized eigenvalue problem are real.

For complex symmetric linear system, several Krylov subspace methods were proposed such as the conjugate orthogonal conjugate gradient method and the conjugate orthogonal conjugate residual method. Though the methods could work well by using good preconditioner, they do not fully exploit the special structures of the coefficient matrix of the linear system in the contour integral eigensolver because the above Krylov methods are designed for general complex symmetric matrices. Moreover, the convergence analysis of the Krylov methods for complex symmetric matrices is not simple compared to that of Krylov methods for real-symmetric matrices and complex Hermitian matrices.

Our method is an iterative method and is derived by extending a study

*Speaker

by Axelsson et al. in 2000. In our method, all operations are performed in real-valued form though the inverse of a real-symmetric matrix appears in the iteration. The convergence rate of our method can easily be analyzed by the spectrum of the original eigenvalue problem.

In the presentation, we show the derivation and an analysis of our method, and numerical experiments using eigenproblems from practical applications.

Keywords: Contour integral eigensolver, generalized eigenvalue problem, complex symmetric matrix

A new implementation of the orthogonal QD algorithm and its application

Kinji Kimura * ¹

¹ Graduate School of Informatics, Kyoto University – Japan

The Golub-Kahan-Lanczos(GKL) bidiagonalization algorithm with the thick-restart is one of the most efficient algorithms for obtaining singular pairs of large scale matrices. The GKL algorithm requires a full singular value decomposition(SVD) solver for bidiagonal matrices in the thick-restart process. If the solver can compute more accurate SVD, the number of iterations in the GKL algorithm decreases. Similarly, in the Sakurai-Sugiura method for computing eigenpairs on large scale matrices, it is required to generate orthonormal basis of the Krylov subspace using SVD. Singular vectors corresponding to non-zero singular values form an orthonormal basis of the Krylov subspace. If singular values computed do not have relative accuracy, we cannot distinguish easily whether singular values are zero or not. Thus, the solver in both the GKL algorithm and the Sakurai-Sugiura method must be able to compute "relatively accurate" singular values. In addition, the algorithm is constituted by a sequence of orthogonal transformations for computing accurate singular vectors.

The modified QR algorithm proposed by Demmel and Kahan, and the orthogonal QD algorithm with shift (oqds algorithm) proposed by von Matt is performed by using sequences of orthogonal transformations. It is to be noted that there is no guarantee that the QR algorithm can compute tiny singular values accurately. On the other hand, the oqds algorithm is guaranteed to compute "relatively accurate" singular values theoretically. However, the code of the oqds algorithm is not opened. Thus, in this talk, we propose four techniques for the implementation of the oqds algorithm. First, we improve to generate orthogonal transformations. The oqds algorithm employs the Givens rotation and the generalized Givens rotation. The Givens rotation is actualized by the BLAS1 subroutine "drotg", and the generalized Givens rotation by the original subroutine "drotg3". Instead of using "drotg", we can use the "dlartg" subroutine in LAPACK to actualize the Givens rotation. This subroutine is slower, but it is more accurate than "drotg". Moreover, the generalized Givens rotation can be redefined by the original Givens rotation. Thus, we can apply "dlartg" to the generalized

*Speaker

Givens rotation.

Secondly, we incorporate a new shift of origin into the oqd algorithm in order to accelerate its convergence. It is known that the convergence of the oqds algorithm is accelerated more effectively when the shift value is closer to the smallest singular value on each iteration. Acceleration of convergence reduces the number of iterations. The smaller number of iterations provides accurate SVD since accumulation of rounding errors can be suppressed. In the conventional method, Laguerre's and Newton's methods are used to compute the shift value. However, the shift value computed does not sufficiently accelerate the convergence. Thus, we propose new shift strategies. The new shift strategies compute bounds of the smallest singular value by the generalized Rutishauser's shift strategy, the Collatz's bound, and the Kato-Temple inequality. The shift values computed by the proposed shift strategies are much closer to the smallest singular value than that computed by the conventional shift strategy.

Thirdly, instead of the stopping criteria proposed by von Matt, we adopt the more efficient stopping criteria which have already been used by the "dlasq2" subroutine and the "dbdsqr" subroutine in LAPACK.

Fourthly, for computing the summation of shift values, the double-double arithmetic is employed.

The results of numerical experiments indicate that the improved oqds algorithm performs more accurate SVD than that of the modified QR algorithm. Moreover, we evaluate the performance of the improved oqds algorithm in the GKL algorithm.

The improved oqds algorithm can be parallelized using the same technique as that used by the modified QR algorithm implemented in ScaLAPACK.

Keywords: orthogonal QD algorithm, relatively accurate singular value, TRGKL

Performance analysis of the quadruple precision eigensolver library QPEigenK on the K computer

Yusuke Hirota * ¹, Toshiyuki Imamura ²

¹ RIKEN Advanced Institute for Computational Science (RIKEN AICS) – Japan

² RIKEN Advanced Institute for Computational Science (RIKEN AICS) – 7-1-26 Minatojima-minimi-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan

In this talk, we present the performance analysis results of the eigensolvers in the quadruple precision eigensolver library QPEigenK on the K computer. In many scientific applications, it is required to solve large dense standard eigenvalue problems. Generally, the computed eigenvalues and eigenvectors become less accurate as the matrix dimension increases due to rounding errors. For example, the result of the eigenvalue problem of 375,000 dimensional matrix in double precision arithmetic has only a few digit of accuracy. Thus high accuracy eigensolvers are required and the performance of the solvers are important for the scientific applications.

QPEigen_s and QPEigen_sx are quadruple precision eigensolvers for a real symmetric matrix in the library QPEigenK. The solvers QPEigen_s and QPEigen_sx compute eigenvalues and eigenvectors through a tridiagonal matrix and a pentadiagonal one respectively. In the solvers, the quadruple precision arithmetic are performed without FMA operations thus each quadruple arithmetic requires 35 double precision arithmetic. The solvers are parallelized in MPI/OpenMP hybrid fashion. However, due to the technical reasons, the backward transformation is not parallelized in threads.

We evaluated the performance of the solvers QPEigen_s, QPEigen_sx, and eigen_s on the K computer (each node has an octa-core CPU, 128 GFLOPS). The solver eigen_s computes eigenvalues and eigenvectors via tridiagonal matrix with double precision arithmetic and is more optimized for the K computer than the quadruple solvers.

The total execution time and its detailed breakdown to compute all eigenvalues and eigenvectors of real symmetric matrices ($n = 5,000, 10,000$) are measured. The execution time of QPEigen_s for the matrix ($n = 10,000$) on 256 nodes and on 16,384 nodes are 1,146 seconds and 118.5 seconds respectively. The execution times are 61 times and 19 times longer than

*Speaker

the execution time of `eigen_s` on 256 nodes. The performance gap between `eigen_s` and `QPEigen_s` on 256 nodes is reasonable since the quadruple precision solver required 35 times more operations and the back transformation, which is the computationally most significant part, is not parallelized in threads. The gap becomes smaller as the number of available node increases since the quadruple precision solvers show better scalability than the double precision solver `eigen_s`.

In the execution of `QPEigen_s` on 4096 node, more than half of the execution time is spend in the computation in back transformation. Thus the thread parallelization of the back transformation is essential for the acceleration of the quadruple solvers. If the backward transformation of `QPEigen_s` is ideally thread parallelized (i.e. the computation in backward transformation is accelerated 8 times), the execution time of `QPEigen_s` for solving eigenproblem ($n = 10,000$) on 4,096 nodes is 65.3 seconds and then more than half of the total execution time is consumed in communication in tridiagonalization and back transformation. Therefore the reduction of the communication time is essential for further speedup. For example, the communication avoiding technique for tridiagonalization (Imamura et al., 2015) may be effective.

Keywords: eigenvalue problem, quadruple precision, `QPEigenK`, K computer

MS - Fast sparse selected inversion techniques

Applications of trace estimation techniques

Yousef Saad * ¹

¹ University of Minnesota – United States

A well-known technique used in statistical methods is to estimate the trace of some matrix via sampling. For example, one can estimate the trace of $\exp(A)$ by computing $w = \exp(A)v$ for many vectors v , and the mean of the inner products of v and w will yield an approximation of the trace under some conditions. This basic technique has found uses in areas as diverse as quantum physics, statistics, and numerical linear algebra. We will discuss the extension of this idea to the problem of estimating the diagonal of the inverse of a matrix and then consider other problems including estimating eigenvalue counts in intervals, computing spectral densities, and estimating ranks of matrices. A few non-standard applications such as subspace tracking in signal processing, and the problem of estimating the log-determinant of positive definite matrices will be also briefly discussed.

Keywords: trace estimators, spectral density, log, determinant

*Speaker

Parallel Selected Inversion of Asymmetric Matrices

Mathias Jacquelin * ¹, Chao Yang ¹, Lin Lin ²

¹ Lawrence Berkeley National Laboratory (LBNL) – Lawrence Berkeley National
Lab MS 50F Cyclotron Rd., Berkeley, CA 94720, United States

² University of California (UC Berkeley) – 119 California Hall Berkeley, CA
94720-1500, United States

In this talk, we will present a parallel implementation of the selected inversion algorithm for asymmetric matrices, PSelInv. We will discuss the scheduling techniques and optimizations that allow PSelInv to deliver a high level of performance on large scale platforms. Additionally, we will present how asynchronous point to point communications can be arranged to further improve the strong scaling of the algorithm.

Keywords: linear algebra, sparse matrix, selected inversion

*Speaker

Takahashi formulas and the generalized Takahashi structure

Klaus Gaertner, Olaf Schenk * ¹

¹ Università della Svizzera italiana (USI) – Institute of Computational Sciences
Via Giuseppe Buffi 13 CH-6904 Lugano, Switzerland

Computing selected elements of the inverse of a structural symmetric sparse matrix A can be done efficiently for the sparse subset Z by the Takahashi recurrence. The computational complexity is that of the LU factorization for A . The technique can be extended to treat the three factor product $X = A^{-1}YB^{-1}$, where A, Y, B are given sparse matrices with some additional properties. Special cases of interest are $B = A^T$, the congruence transformation (e.g. quantum physical evolution processes described by the NEGF techniques) or $B = A$, describing the parameter dependence of the inverse with respect to parameters p_i entering $A(y_{ijk}) = da_{ij}/dp_k$. We give an explicit prove that X can be computed by Takahashi like techniques with the complexity of the LU factorization of A .

Based on the proof special aspects of the related algorithm, especially its parallelization, will be discussed. The algorithm will be made available via the sparse linear solver PARDISO.

Keywords: sparse matrices, computation of selected elements of the inverse, quantum scattering problem, sensitivity of the inverse matrix.

*Speaker

Highly parallel stochastic estimation of the diagonal of inverse matrices

Fabio Verbosio * ¹, Matthias Bollhoefer ², Olaf Schenk ¹

¹ Università della Svizzera italiana, Institute of Computational Science (ICS) –
Institute of Computational Science Università della Svizzera Italiana Via G. Buffi
13 6900 Lugano Switzerland, Switzerland

² TU Braunschweig, Institute for Computational Mathematics (TUBS) – TU
Braunschweig Institute for Computational Mathematics Pockelsstr. 14 38106
Braunschweig, Germany

In recent years, the increasing accuracy of elaborated mathematical models has caused a boost in both the complexity of numerical methods and the size of linear systems used to approach these models. The solution of such systems and, therefore, the inversion of the matrices involved, reached prohibitive costs in terms of memory and complexity. For this reasons, the computation of selected entries of inverse matrices acquires primary importance in several applications, from risk management to nano-electronic device simulation and genomic prediction.

Among the suitable methods, stochastic techniques are applied whenever an estimation up to some precision of some of the entries of an inverse matrix is needed. Such techniques are based on iterative methods and own therefore very high potential for scaling.

We present a highly parallel stochastic estimation algorithm for the extraction of the diagonal of the inverse of sparse matrices representing stencil operators on bi- and tridimensional grid and we show scalability results for a distributed memory implementation of the algorithm based on asynchronous and collective communication (MPI, GASPI). Furthermore, we present an application on both symmetric and non-symmetric matrices arising from the computation of the retarded Green's function in the non-equilibrium Green's function method, used in the design of nanoscale electronic devices.

Keywords: stochastic estimator, selective inversion, parallel computing, high performance computing, Green's functions

*Speaker

MS - Multilevel parallel linear solvers

Multilevel Spectral Analysis of H-Matrices

Yuval Harness ^{*† 1}, Emmanuel Agullo^{‡ 1}, Luc Giraud^{§ 1},
Eric Darve ²

¹ INRIA (INRIA) – INRIA – France

² Institute for Computational and Mathematical Engineering (iCME) – Huang
Building 475 Via Ortega Stanford University Stanford, CA 94305-4042, United
States

In this talk we will present a new multilevel spectral analysis for hierarchical matrices. A detailed estimate of the condition number of the preconditioned system in the strong hierarchical symmetric case will be analyzed and demonstrated. Similarities and differences with the algebraic multilevel iteration (AMLI) method will be explored and discussed. As an application we will consider the accuracy of the analysis and the efficacy of the condition number estimation on dense Schur complement matrices that arise in the solution of large scale sparse linear systems via parallel hybrid solver that combines direct sparse solvers with domain decomposition approach. Numerical and parallel experiments will be presented to evaluate the advantages and drawbacks of the new approach.

This work is a joint effort between Professor Darve's group at Stanford and the Inria HiePACS team.

Keywords: Hierarchical solvers, Spectral bounds, Algebraic multilevel methods, Preconditioning

*Speaker

†Corresponding author: yuval.harness@inria.fr

‡Corresponding author: emmanuel.agullo@inria.fr

§Corresponding author: giraud@cerfacs.fr

Fast Methods applied to BEM Solvers for industrial problems

Guillaume Sylvand * ¹

¹ Airbus Group Innovations / Inria (AGI/Inria) – INRIA, Airbus Group Innovations – France

Fast Methods applied to BEM Solvers for industrial problems. For the numerical simulation of wave propagation in acoustics, Airbus Group Innovations relies on integral equations solved with the Boundary Elements Method, leading to the need to solve dense linear systems. In this talk, we intend to present two families of fast solvers (Fast Multipole Method and Hmat method) that can be used on these systems. We propose to underline their similarities, their connections and their differences, to present their complementarity in future high performance solvers and to illustrate their performances on industrial class applications.

Airbus Group Innovations is the Airbus Group research center, dedicated to upstream research applied to all Business Units (Airbus, Airbus Helicopters, Airbus Defence and Space). The applied mathematics team has developed over the years a software family called Aseris destined to solve various acoustic and electromagnetism propagation problems using integral equations and boundary elements methods. This software suite is used in design and research department to work on noise reduction, CEM, furtivity, etc. The advantages of integral equations and BEM solver are well known: mainly accuracy, and simpler (surfacic) mesh. The main algorithm drawback is the need to cope with a dense matrix whose size can be quite large for wave propagation problems, where the mesh step is governed by the wavelength of the physical problem treated (in frequency domain). For example, acoustic problems on a full size aircraft at 20 000 Hz (upper limit of audible frequencies) can involve more than 10^8 unknowns. Solving such linear systems with standard method is just impossible (storage would require 80.000 terabytes of disk, factorization would take 100 years on all Airbus HPC facilities). Since the late 90's, fast methods have been introduced to deal with these limitations. First, the Fast Multipole Method (FMM) allowed to compute fast matrix-vector products (in $O(n \log^2(n))$) instead of $O(n^2)$ for the standard algorithm), and hence to design fast solvers using iterative methods. Lately, H-mat methods have gained wide acceptance by introducing fast direct solvers, allowing to solve systems in $O(n \log^2(n))$ –

*Speaker

or less – without the hassle of using iterative solvers (unknown convergence rate and difficulty to find a good preconditionner). H-mat is a lossy, hierarchical storage scheme for matrices that, along with an associated arithmetic, provides a rich enough set of approximate operations to perform the matrix addition, multiplication, factorization (e.g. LU or LDLT) and inversion. It allows the construction of a fast direct solver with complexity $O(n \log^2(n))$ in some cases, which is especially important for BEM applications as it gracefully handles a large number of Right-Hand Sides (RHS). It also provides a kernel-independent fast solver, allowing one to use the method for different physics. Airbus Group Innovations has recently implemented the H-mat arithmetic and successfully applied it to a wide range of industrial applications in electromagnetism and acoustics. Furthermore, these algorithms are hard to efficiently parallelize, as the very scarce literature on the subject shows. We developed a parallel solver that goes beyond the aforementioned reference, using innovative techniques on top of a state-of-the-art runtime system StarPU. This enables the solving of very large problems, with a very good efficiency. In this presentation, we show some results on the accuracy of this method on several challenging applications, and its fast solving time and efficient use of resources.

High performance solvers have been implemented into Aseris software that allows to run large scale industrial applications. H-mat solver has to be preferred for medium problems especially with a large number of RHS. At the moment, FMM solver remains the reference solver for huge problems.

Keywords: Fast solver, Fast multipole method, FMM, H, matrix, task based, runtime

Performance and scalability of a parallel Block Low-Rank multifrontal solver

Patrick Amestoy* ¹, Alfredo Buttari[†] ², Jean-Yves
L'excillant[‡] ³, Theo Mary ^{§¶} ⁴

¹ Institut de recherche en informatique de Toulouse (IRIT) – Institut National Polytechnique de Toulouse - INPT – 118 Route de Narbonne, F-31062 Toulouse Cedex 9, France

² Institut de recherche en informatique de Toulouse (IRIT) – CNRS : UMR5505 – 118 Route de Narbonne, F-31062 Toulouse Cedex 9, France

³ INRIA - Laboratoire de l'Informatique du Parallélisme (LIP) – École Normale Supérieure (ENS) - Lyon – 46 Allée d'Italie 69364 LYON CEDEX 07, France

⁴ Institut de recherche en informatique de Toulouse (IRIT) – Université Paul Sabatier (UPS) - Toulouse III – 118 Route de Narbonne, F-31062 Toulouse Cedex 9, France

We consider the use of the Block Low-Rank format (BLR) to solve real-life industrial problems with low-rank sparse direct solvers. The Block Low-Rank approach provides significant gains compared to full-rank solvers on practical applications. In this talk, we present several variants of the BLR multifrontal factorization, depending on the strategies used to perform the updates in the frontal matrices and on the approaches to handle numerical pivoting. In our numerical experiments, the MUMPS library is used to compare and analyze the performance of each BLR variant in a parallel (MPI+OpenMP) setting on a variety of applications.

Keywords: Block Low Rank, multifrontal, sparse direct solver

*Corresponding author: Patrick.Amestoy@enseeiht.fr

†Corresponding author: alfredo.buttari@enseeiht.fr

‡Corresponding author: jean-yves.l'excillant@ens-lyon.fr

§Speaker

¶Corresponding author: theo.mary@enseeiht.fr

Approximate factorization preconditioning with HSS matrices and random sampling

Pieter Ghysels ^{*} ¹, Xiaoye Li ¹

¹ Lawrence Berkeley National Laboratory (LBNL) – Lawrence Berkeley National
Lab MS 50F Cyclotron Rd., Berkeley, CA 94720, United States

We present an effective preconditioner for linear systems that arise from PDE discretizations. The preconditioner is constructed from an incomplete factorization, based on a multifrontal version of classical Gaussian elimination. The fill-in introduced during the factorization is compressed or approximated by rank-structured matrices, i.e., matrices with sub-blocks that are of low rank. We consider Hierarchically Semi-Separable (HSS) matrices, a specific type of rank-structured or hierarchical matrices. The compression of matrix sub-blocks into a low-rank product representation in the HSS format is performed with a novel randomized sampling technique. We apply the incomplete factorization as a preconditioner for GMRES or BiCGStab and compare with a number of other common preconditioners such as ILU and AMG. We look at linear and nonlinear elasticity problems, Maxwell's equation and a number of large-scale applications which are of key importance to the DOE. Our solver/preconditioner is released under a BSD license as a package called STRUMPACK. It supports MPI+OpenMP parallelism, single and double precision, real and complex numbers and 64 bit integers.

Keywords: factorization, preconditioning, hierarchical matrix, low, rank, HSS

^{*}Speaker

Feedback in the use of multilevel parallel linear solvers for thermomechanical studies

Olivier Boiteau ^{*† 1}, Natacha Béreux ^{*}₂

¹ EDF Lab (EDF R&D) – EDF – EDF Lab Paris-Saclay EDF

R&D/SINETICS/I23 7, bd Gaspard Monge 91120 Palaiseau, France

² EDF Lab (EDF R&D) – EDF – EDF Lab Paris-Saclay EDF R&D/AMA 7, bd Gaspard Monge 91120 PALAISEAU, France

EDF guarantees the technical and economic capabilities of its electricity production means, from the conception to the end of operational life. The safety and availability requirements infer the need to verify the correct operation, maintenance and replacements conditions of machines through numerical simulation. The analysis of real behaviour and the risks associated with industrial structures requires knowledge over a full range of physical models: thermomechanics with Code_Aster[1], electromagnetics with Code_Carmel[2], hydrodynamics with TELEMAC system[3] etc.

Often, EDF has opted for developing its own in-house codes. Thanks to this choice, we often remain free to choose their basic components. One of these components is particularly crucial for the time consumption and memory requirement of our codes: the sparse linear system management. For more than 10 years we have chosen to limit our research and development on this topic. When it is possible, considering our software industrial requirement and end-users' expectations and operational constraints, it is usually well worth to entrust this crucial algorithm step to external packages, such as MUMPS[5]. and PETSc[4].

This talk gives some feedback in terms of performance, robustness and flexibility in the daily use of these powerful products in our thermomechanical studies. It also aims to highlight the recent use of block low-rank sparse factorization and multigrid preconditioning in this context.

Joint work with the MUMPS team.

[1] <http://www.code-aster.org>

[2] <http://code-carmel.univ-lille1.fr>

[3] <http://opentelemac.org>

*Speaker

†Corresponding author: olivier.boiteau@edf.fr

- [4] <http://www.mcs.anl.gov/petsc/>
[5] <http://mumps.enseeiht.fr>

Keywords: Sparse linear solver, Block low rank compression, Multigrid preconditioners, MUMPS, PETSc, Multifrontal solver, Preconditioner, Krylov solver, thermomechanics

Designing an efficient and scalable block low-rank direct solver for large scale clusters

Xavier Lacoste * ¹, David Goudin ¹, Cedric Augonnet ¹

¹ Commissariat à l’Energie Atomique et aux Energies Alternatives (CEA-CESTA)
– Université de Franche-Comté – CESTA, 33116 Le Barp Cedex, France, France

The resolution of electromagnetics problems with integral equation formulation leads to solving large dense linear systems that require a lot of memory and computational power. In order to address large problems, we enhanced our Full-MPI direct solver with a block low-rank compression method which dramatically reduces memory and computational power requirements. In this talk, we will show how we derived a block low-rank version of our existing solver, and will detail the various optimizations we used to increase its scalability on a large-scale cluster. For example by improving the performance of not only the most common kernel that rarely occurs but induce a high load imbalance between nodes. We also used an hybrid programming MPI/OpenMP model to reduce the load imbalance. And, finally we introduced algorithmic improvements to remove synchronizations and improve the execution of the critical path.

Keywords: Dense Linear Algebra, Direct solver, Block Low Rank

*Speaker

Parallel hierarchical solver

Eric Darve * ¹, Hadi Pouransari ¹, Chao Chen ¹, Erik Boman ², Siva Rajamanickam ²

¹ Stanford – United States

² Sandia National Laboratories [Albuquerque] (SNL) – PO Box 5800
Albuquerque, NM 87185, United States

Hierarchical solvers are linear solvers with reduced computational cost and memory requirements. They are based on the low-rank compression of certain off-diagonal blocks, which allows a trade-off between accuracy and computational time. They offer many similarities with multigrid solvers and as such rely on a set of multilevel grids, that are increasingly coarser. Compared to multigrid, the pattern of computation is similar. In a parallel implementation, we can process each grid from fine to coarse, with operators that transfer information between a child and its parent grid. Similarly, communication is required to exchange data at the boundary of each processor subdomain. Despite these similarities with multigrid, hierarchical solvers have unique features such as the fact that they are based on a (sparsified) LU or Cholesky factorization, and the low-rank compression of fill-ins in the matrix. As a result, unique optimizations are required to reach good scalability on a parallel computer. We will present details on the algorithm and its parallel implementation using MPI. Benchmarks will be presented on various problems resulting from the discretization of partial differential equations.

Keywords: parallel linear solver, multilevel solver, hierarchical solver

*Speaker

A massively parallel sparse linear solver ESPRESO

Lubomir Riha ¹, Alexandros Markopoulos * ¹, Tomas Brzobohaty ¹, Ondrej Meca ¹, Tomas Kozubek ¹

¹ IT4Innovations National Supercomputing Centre (IT4Innovations) – 6231, Studentska, 708 33 Ostrava - Poruba, Czech Republic

ESPRESO is a highly efficient parallel solver which contains the HTFETI method suitable for parallel machines with tens or hundreds of thousands of cores. The solver is able to run on massively parallel machines with thousands of compute nodes and hundreds of thousands of CPU cores. ESPRESO also supports modern many-core accelerators.

We are currently developing several versions of the solver. ESPRESO CPU is a CPU version using sparse representation of system matrices. ESPRESO MIC is an Intel Xeon Phi accelerated version working with dense representation of system matrices in the form of Schur complement. ESPRESO GPU is a GPU accelerated version working with dense structures. Support for sparse structures using cuSolver is under development.

In order to solve real engineering problems, we are developing a FEM/BEM library that enables database files from ANSYS simulation software to be imported and all inputs required by the FETI or HTFETI solver generated. In addition, we are developing an interface to ELMER that allows ESPRESO to be used as its linear solver. This integration is done through API that can be used as an interface to many other applications.

The Hybrid FETI method implemented in ESPRESO is a combination of FETI and FETI-DP method. As it is known, FETI-DP approach requires an identification of the corner nodes (CN) to enforce partial gluing across neighboring subdomains. Moreover, the subset of CN, in some cases, is not arbitrary. This step usually requires passing to the solver beside the matrices and vectors also mesh information. The talk will show a different implementation of FETI-DP based on algebraic operations only to avoid information extraction from the mesh.

In latest tests, ESPRESO library solved a linear elasticity problem with 70 billions of unknowns, and heat transfer problem with 124 billions of unknowns.

*Speaker

Keywords: domain decomposition method, Hybrid FETI method, massive parallelization, singular matrix, factorization

**MS - Efficient computation
of inner eigenvalues of large
sparse matrices**

Efficient subspace iteration with Chebyshev-type filtering

Martin Galgon ¹, Lukas Krämer ¹, Bruno Lang ^{* 1},
Andreas Alvermann ², Holger Fehske ², Andreas Pieper ²,
Georg Hager ³, Moritz Kreutzer ³, Faisal Shahzad ³,
Gerhard Wellein ³, Achim Basermann ⁴, Melven
Röhrig-Zöllner ⁴, Jonas Thies ⁴

¹ University of Wuppertal – Germany

² University of Greifswald – Germany

³ Erlangen Regional Computing Center – Germany

⁴ German Aerospace Center (DLR) – Germany

Shift-invert and other methods for computing inner eigenvalues often require the solution of linear systems. This may become a problem if the linear systems are very ill-conditioned and the matrix dimension precludes the use of direct solvers. Then eigensolvers with polynomial acceleration become particularly attractive because they avoid the solution of linear systems.

We discuss techniques for increasing the efficiency of subspace iteration with polynomial filtering. These include reducing the polynomial degree by adapting Chebyshev-based filter functions, optimizing the size of the search space, and using tailored high-performance computational kernels.

*Speaker

Parallel methods for computing interior eigenvalues in linear and nonlinear eigenvalue problems in SLEPc

Jose E. Roman * ¹, Carmen Campos ¹

¹ Universitat Politècnica de Valencia (UPV) – Spain

SLEPc provides a collection of solvers and tools for computing the partial solution of various types of eigenvalue problems, including linear, polynomial and nonlinear, on parallel computers. In this talk we discuss a few recent additions that are relevant when computing interior eigenvalues. For symmetric-definite eigenproblems, $Ax = \lambda Bx$, in which all eigenvalues are real, SLEPc is able to compute all eigenvalues contained in a given interval. This is done with a spectrum slicing technique based on inertia (via indefinite triangular factorizations of $A - \sigma B$ computed at strategic shifts σ) that sweeps the interval using Lanczos to compute eigenvalues in chunks until there is guarantee that all eigenvalues have been found. We present an extension that allows the computation to be done with hierarchical parallelism, in a divide-and-conquer scheme that splits the interval of interest in several subintervals. Data is moved transparently from the parent communicator to the sub-communicators, and vice versa. We also provide some optimizations for the case that a sequence of similar eigenproblems is solved with this technique.

For polynomial eigenvalue problems, $P(\lambda)x = 0$, where $P(\cdot)$ is a matrix polynomial, we have implemented a parallel Jacobi-Davidson solver that incorporates a deflation strategy so that several interior eigenvalues can be computed in an effective way. The algorithm is based on work by Effenberger [SIMAX 34(3):1231-1256, 2013]. For the general nonlinear eigenvalue problem, $T(\lambda)x = 0$, where $T(\cdot)$ is an analytic matrix function, we present a parallel implementation of the NLEIGS method by Güttel et al., [SISC 36(6):A2842-A2864, 2014], which is based on a (rational) Krylov iteration operating on a companion-type linearization of a rational interpolant of the nonlinear function. Our implementation uses a tensorized representation of the built subspace basis in order to be memory-efficient.

*Speaker

Filtered thick restart Lanczos algorithm and the EVSL package

Yousef Saad * ¹

¹ University of Minnesota – United States

This talk will be about two different strategies for extracting extreme or interior eigenvalues of large sparse (Hermitian) matrices. The first is based on a polynomial filtering technique. This general approach can be quite efficient in the situation where the matrix-vector product operation is inexpensive and when a large number of eigenvalues is sought, as is the case in electronic structure calculations for example. However, its competitiveness depends critically on a good implementation. The method presented relies on a combination of the Lanczos algorithm with partial reorthogonalization and polynomial filtering based on least-squares polynomials. The second approach we discuss represents ongoing work based on using domain-decomposition type techniques. This approach relies on spectral Schur complements combined with Newton's iteration. This method is particularly appealing for interior eigenvalue problems.

Keywords: Lanczos algorithm, Thick restart, polynomial filtering, rational filtering.

*Speaker

Block Krylov-type complex moment-based nonlinear eigensolver with hierarchical parallelism

Akira Imakura * ¹, Tetsuya Sakurai ¹

¹ University of Tsukuba – Japan

In this talk, we consider a method for finding all eigenvalues located in a certain region on the complex plain and their corresponding eigenvectors of nonlinear eigenvalue problems. As a powerful algorithm for solving this kind of interior-type generalized eigenvalue problems, Sakurai and Sugiura have proposed the complex moment-based eigensolver that constructs certain complex moment matrices constructed by a contour integral. The most time-consuming part of the complex moment-based eigensolver is to solve linear systems in each quadrature point of a numerical integral. Since these linear systems can be solved independently, this method has high parallel efficiency. Now, there are several improvements and variants for solving generalized eigenvalue problems and also for solving nonlinear eigenvalue problems.

The most well used complex moment-based eigensolvers are the block SS-RR method and the FEAST eigensolver. These methods construct the subspace by the contour integral and extract the target eigenpairs by directly applying the Rayleigh-Ritz procedure. Here, we note that the block SS-RR method is characterized as using high order complex moments and the FEAST eigensolver is premised on using as the subspace iteration method.

Recently, as improvements of the block SS-RR method for solving generalized eigenvalue problems, we proposed block Krylov-type complex moment-based eigensolvers which are based on the block Arnoldi/Lanczos methods of standard eigenvalue problems. Using the block Arnoldi/Lanczos procedure, these methods can avoid directly applying Rayleigh-Ritz procedure. This leads to reduce the computational costs for extracting the eigenpairs from the subspace constructed by the contour integral.

In this talk, we extend the idea to the nonlinear eigenvalue problems and

*Speaker

propose a block Krylov-type complex moment-based nonlinear eigensolver. We also evaluate the performance of the proposed method compared with the traditional complex moment-based nonlinear eigensolvers.

Keywords: Nonlinear eigenvalue problems, complex moment based eigenvalues, Krylov subspace.

MS - Sparse matrix and tensor computations

High Performance Parallel Sparse Tucker Decompositions

Oguz Kaya * ¹, Bora Uçar† ²

¹ École normale supérieure de Lyon (ENS LYON) – École Normale Supérieure (ENS) - Lyon – 15 parvis René Descartes - BP 7000 69342 Lyon Cedex 07, France

² ROMA (ENS Lyon / CNRS / Inria Grenoble Rhône-Alpes) – CNRS : UMR5668, Laboratoire d’informatique du Parallélisme, École Normale Supérieure (ENS) - Lyon, INRIA – Laboratoire de l’Informatique du Parallélisme 46 Allée d’Italie 69364 Lyon, France

Tensor methods have increasingly been employed to better analyze datasets with many features and higher dimensionality than matrix-based methods. Tucker tensor decomposition has successfully been applied to real-world problems such as web search, hyperlink analysis of web pages, and recommender systems, albeit being computationally expensive for large datasets. This talk focuses on an efficient computation and parallelization of the Higher Order Orthogonal Iteration (HOOI) algorithm to compute the Tucker decomposition of very big sparse tensors and enable tensor-based analysis of such datasets. We investigate the parallelization of the major steps of the HOOI algorithm such as tensor-times-matrix-chain multiply and truncated SVD operations. We then examine reducing the load imbalance and the communication cost of the parallel algorithm for better scalability. Finally, we present scalability results up to 4096 cores on 256 nodes of an IBM BlueGene/Q supercomputer of the MPI+OpenMP parallel implementation of the algorithm. This is a joint work with Bora Uçar.

Keywords: tucker decomposition, sparse tensor, parallel algorithms, combinatorial scientific computing

*Speaker

†Corresponding author: bora.ucar@ens-lyon.fr

An Exploration of Optimization Algorithms for High Performance Tensor Completion

Shaden Smith ¹, Jongsoo Park ², George Karypis ^{*† 1}

¹ Computer Science & Engineering Department, University of Minnesota – 4-192 KHKH, 200 Union St SE, Minneapolis, MN 55455, United States

² Intel – United States

Many domains rely on multi-way data, which are variables that interact in three or more dimensions, or modes. An electronic health record is an interaction between variables such as a patient, symptoms, diagnosis, medical procedures, and outcome. Similarly, how much a customer will like a product is an interaction between the customer, product, and the context in which the purchase occurred (e.g., date of purchase or location). Analyzing multi-way data can provide valuable insights about the underlying relationships of the interacting variables. Utilizing these insights, a doctor would be more equipped to reach a provide a successful treatment and a retailer would be able to better recommend products that meet the customer’s needs and preferences. Tensors are a natural way of representing multi-way data. Tensor completion is the problem of estimating or recovering missing values of a tensor. For example, discovering phenotypes in electronic health records is improved by tensor completion due to missing and noisy data. Similarly, predicting how a customer will rate a product under some context can be thought of as estimating a missing value in a tensor.

Multi-way data analysis follows the assumption that the data of interest follows a low-rank model that can be discovered. Tensor factorization is a technique that reduces a tensor to a low-rank representation, which can then be used by applications or domain experts. Tensor completion is often accomplished by finding a low-rank tensor factorization for the known data, and if a low-rank model exists then it can be used to predict the unknown data. A subtle, but important constraint is that the factorization must only capture the non-zero (or observed) entries of the tensor. The remaining entries are treated as missing values, not actual zeros as is often the case in other sparse tensor and matrix operations.

Tensor completion is challenging on modern processors for several reasons. Modern architectures have lower ratios of memory bandwidth to com-

*Speaker

†Corresponding author: karypis@cs.umn.edu

pute capabilities, which is detrimental to tensors which have highly unstructured access patterns and three or more indices per non-zero value. Furthermore, processors have more parallelism and load balance is difficult to achieve because tensors do not have uniformly distributed non-zeros and often have a combination of long, sparse modes (e.g., patients or customers) and short, dense modes (e.g., medical procedures or temporal information).

The high performance computing community has addressed some of these challenges in recent years, with research spanning both shared-memory and distributed-memory systems. However, the techniques and optimizations that underlie these methods are applied to factorizations that are not suitable for tensor completion due to the treatment of missing entries.

In this work, we explore the task of high performance tensor completion with three popular optimization algorithms: alternating least squares (ALS), stochastic gradient descent (SGD), and coordinate descent (CCD++). We address issues on shared- and distributed-memory systems such as memory- and operation-efficient algorithms, cache locality, load balance, and communication. Our contributions include: i) Hybrid MPI+OpenMP implementations of ALS, SGD, and CCD++ that utilize compressed tensor representations in order to improve cache locality and reduce memory consumption and the number of FLOPs performed; ii) A distributed-memory SGD algorithm that combines stratification and asynchronous updates to improve scalability; iii) A method of load balancing in the presence of sparse and dense modes; iv) An experimental evaluation with several real-world datasets on up to 1024 cores. Our ALS and CCD++ algorithms are 153x and 21.4x faster than state-of-the-art parallel methods, respectively. This effectively reduces solution time from hours to seconds; v) We show that depending on the underlying parallel architecture and the characteristics of the desired solution, the best performing optimization method varies.

An Empirical Study of Sparse BLAS on Emerging Heterogeneous Processors

Weifeng Liu ^{*† 1}, Brian Vinter ¹

¹ University of Copenhagen – Denmark

In recent years, low-throughput GPUs have been integrated onto the same chip as the CPU. AMD APUs, Intel CPU-GPU SoCs and nVidia Tegra are representatives in this trend. The newest hardware progress, such as unified virtual address space and shared last level caches, makes tightly coupled CPU-GPU heterogeneous processors a promising tool for scientific computing. This talk will focus on our empirical study of performance behaviors of sparse BLAS routines (e.g., SpTRANS, SpMV, SpTRSV and SpGEMM) on emerging heterogeneous processors. A performance comparison with modern multi-core and many-core processors will also be presented.

Keywords: Sparse BLAS, Heterogeneous processors, GPU, Shared virtual memory

*Speaker

†Corresponding author: weifeng@nbi.dk

Making effective sparse matrix–vector multiplication both portable and usable

Albert-Jan Yzelman ^{*† 1}

¹ Huawei Technologies France – Huawei Technologies – France

Past research into efficient formats for sparse matrices has identified three major performance aspects: maximising bandwidth use, cache reuse, and data locality. On any architecture, whether they be contemporary CPUs, accelerators such as GPUs, or co-processors like the Intel Xeon Phi, the data structure additionally is tuned to make maximum use of their respective parallelisation capabilities: multi-threading, vectorisation, or both.

This led to a veritable plethora of data structures that, in practice, remain mostly unused. For reasons of portability and usability, the de facto data structure remains to be Compressed Row Storage (CRS, also known as CSR), or even the simple coordinate format (COO); while both CRS and COO have disadvantages are theoretically described, that are furthermore and repeatedly have shown to lead to practical and noticeable loss in performance. Are we thus perhaps in a case of premature optimisation, and should we focus on optimising CRS and COO instead of trying to replace them? Or is it that performance should not be the main metric for success, and that future research should address more the portability and usability aspects of their proposed solutions?

While the state of the art is exploring both directions, this talk focuses on the latter: we explore the high-level trade-offs on the possible data structures when taking into account bandwidth, cache, and locality from a performance perspective; we present a framework for dealing with vectorisation from a generic point of view; and we discuss various use cases, specifically, how to hide the complexity of high-performance data structures from application programmers.

Keywords: sparse matrix, matrix, vector multiplication, cache, oblivious, multi, threading, vectorisation, parallelisation, data structure, CRS, CSR, COO,

*Speaker

†Corresponding author: albertjan.yzelman@huawei.com

iterative solvers

CP - Preconditioning

Parallel preconditioners for a higher order time integration methods

Radim Blaheta ^{*† 1}, Owe Axelsson ¹, Tomas Lubier ¹

¹ Institute of Geonics AS CR (IGN) – Studentska 1768, 708 00 Ostrava-Poruba, Czech Republic

The contribution considers nonstationary Darcy flow or poroelasticity problems discretized in space by mixed finite elements and in time by stable higher order methods. Parallelizable preconditioners for iterative solution of linear systems arising within the time steps are suggested and analysed. The analysis is illustrated by numerical experiments with third order Runge-Kutta time integration method.

Keywords: higher order time integration, parallelizable preconditioners, poroelasticity

*Speaker

†Corresponding author: blaheta@ugn.cas.cz

Scaling preconditioners for parallel CG-accelerated projection techniques

Dan Gordon ^{*† 1}, Rachel Gordon ²

¹ Dept. of Computer Science – University of Haifa, Haifa 34988, Israel

² Dept. of Aerospace Engineering – The Technion, Haifa 32000, Israel

We consider the problem of solving strongly indefinite linear systems that arise from a variety of domains, such as circuit simulation, CFD, acoustic and thermal problems, and many others. The matrices of such systems often have a highly irregular structure with an extremely large variance in both the size and the number of elements in the rows and columns. Such problems gave rise to sophisticated preconditioning techniques, which often aim to place the largest element on the diagonal. Another problem with huge matrices is that the ILU-type preconditioners are inherently sequential and their parallel versions are not as useful in all cases.

We examine the runtime behavior of several CG variants, such as CGNR, CGNE, CGMN (and its block-parallel version CARP-CG), combined with various row and column scalings. We also introduce a new variant: CG acceleration of a method originally introduced in the context of image reconstruction from projections [Censor, Gordon & Gordon, PARCO 2001]. This method is called "CARP1" in the CARP paper [Gordon & Gordon, SISC 2005], and also "DROP" in [Censor et al., SISC 2008]. CARP1 is just CARP with blocks consisting of a single equation, but its CG acceleration, called CARP1-CG has not been studied before. CARP1 incorporates a mechanism which takes account of the number of elements in every column.

CARP-CG has been used successfully on many problematic linear systems arising from partial differential equations (PDEs), such as convection-dominated problems, the Helmholtz equation at high frequencies, problems with discontinuous coefficients, and problems with large off-diagonal elements. More recently, CARP-CG was found to be useful for solving difficult linear systems arising from the solution of eigenvalue problems, and for solving the elastic wave equation in the frequency domain. Even though CARP-CG is essentially a domain decomposition (DD) method, its application to large 3D wave problems does not exhibit the problem of artificial reflections at subdomain boundaries; a problem common to other DD methods.

*Speaker

†Corresponding author: gordon@cs.haifa.ac.il

The efficacy of CARP-CG on such problems is due to the fact that it is a CG acceleration of CARP, which is a block-parallel version of KACZ - the Kaczmarz algorithm. KACZ is SOR on the system $AA^T y = b, x = A^T y$, so it inherently normalizes the equations. A key reason for the robustness of KACZ is that after the normalization of A , all elements on the diagonal of AA^T are 1, while the off-diagonal elements are < 1 . The robustness of KACZ carries over to its parallelization and acceleration in CARP-CG.

We present several simple schemes for the acceleration and improved accuracy of the above algorithms. These schemes are various combinations of row and column scalings. The row scaling consists of dividing each equation by the L_2 -norm of its coefficients, which we call equation normalization (EN). EN has also been shown to be useful for GMRES and Bi-CGSTAB on problems with discontinuous coefficients [Gordon & Gordon, JCAM 2010].

We also use two types of column scalings: the first one, called column normalization (CN), consists of dividing each column by the L_2 -norm of the column elements. The second type of column scaling is sparsity-oriented: each column is divided by the number of its nonzeros; this scaling is called CS. These scalings can also be combined, for example, EN-CN-CS.

The results of this study show that the above CG methods can benefit very significantly from these scalings and their combination. For a given number of iterations, relative residual results can be improved on some problems by several orders of magnitude. Equivalently, relative residual goals can be reached in a significantly shorter time.

Keywords: Circuit problems, CG acceleration, CARP_CG, row scaling, column scaling, sparsity_oriented scaling.

Large-Scale Sparse Inverse Covariance Estimation

Matthias Bollhoefer^{* 1}, Olaf Schenk^{† 2}

¹ TU Braunschweig, Institute for Computational Mathematics (TUBS) – TU Braunschweig Institute for Computational Mathematics Pockelsstr. 14 38106 Braunschweig, Germany

² USI Lugano, Institute of Computational Science (USI) – Institute of Computational Science Università della Svizzera italiana Via Giuseppe Buffi 13 6900 Lugano, Switzerland

The estimation of large sparse inverse covariance matrices is an ubiquitous statistical problem in many application areas such as mathematical finance or geology or many others. Numerical approaches typically rely on the maximum likelihood estimation or its negative log-likelihood function. When the Gaussian mean random field is expected to be sparse, regularization techniques which add a sparsity prior have become popular to address this issue. Recently a quadratic approximate inverse covariance method (QUIC) [1] has been proposed. The hallmark of this method is its super-linear to quadratic convergence which makes this algorithm to be among the most competitive methods. In this paper we present a sparse version of this method and we will demonstrate that using advanced sparse matrix technology the sparse version of QUIC is easily able to deal with problems of size one million within a few minutes on modern multicore computers.

[1] C.-J. Hsieh, M.A. Sustik, I.S. Dhillon, and P.K. Ravikumar, Sparse inverse covariance matrix estimation using quadratic approximation, in Advances in Neural Information Processing Systems, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds., vol. 24, Neural Information Processing Systems Foundation, 2011, pp. 2330-2338.

Keywords: inverse covariance estimation, sparse matrices

*Speaker

†Corresponding author: olaf.schenk@usi.ch

CP - Linear algebra

Exploring Vectorization Possibilities on the Intel Xeon Phi for Solving Tridiagonal Systems

Ioannis Venetis ^{*† 1}, Alexandros Nakos ¹, Alexandros
Kouris ¹, Efstratios Gallopoulos ¹

¹ University of Patras [Patras] – 265 04 Rio Patras, Greece

Calculating the solution of a tridiagonal system is a problem that has attracted the attention of many researchers, since it commonly appears as a computational kernel in the context of larger problems. Even though this problem is of only linear sequential computational complexity there is a wealth of properties that have inspired innovative parallel solutions; see e.g. [2]. The introduction of coprocessors like the Intel Xeon Phi and the NVIDIA GPUs has led to algorithms that try to exploit the characteristics of these architectures in order to further speedup the calculation and the accuracy of the computed solution. For example, in [1] the Spike algorithm for banded systems [3] is combined with a special block diagonal pivoting strategy to produce an algorithm for general tridiagonal systems. The algorithm has been mapped on the GPU architecture providing very good performance. Similarly, we have recently presented g-Spike, an algorithm that solves general tridiagonal systems. g-Spike also builds on the Spike partitioning, but uses orthogonal factorizations via Givens reductions in order to increase robustness. We showed that g-Spike leads to high performance when implemented in CUDA for NVIDIA GPUs as well as on the Intel Xeon Phi; cf. [5] and [4]. Regarding the latter, we note that the main characteristics of the Xeon Phi that allow applications to achieve high performance are its large number of general purpose execution cores (up to 61, each one with 4-way HyperThreading) and its 512-bit wide vectorization registers. Despite the performance gains achieved in our previous work due to vectorization, we found that significant further improvements can be obtained. For example, data layout can be reorganized so as to move elements that are accessed together into contiguous memory addresses, hence further assisting vectorization. Such an approach could prove beneficial if the cost of data movement can be amortized among multiple uses of the same tridiagonal matrix in the context of a larger application. Furthermore, it is quite common for such systems to have multiple right-hand sides, that can

*Speaker

†Corresponding author: venetis@ceid.upatras.gr

be processed in a vectorized manner. We finally consider the more special but rather frequent case of matrices that are symmetric positive definite or diagonally dominant matrices and discuss algorithms that are suitable for the aforementioned coprocessor architecture.

- [1] L.-W. Chang, J.A. Stratton, H.S. Kim, and W.-M.W. Hwu. A scalable, numerically stable, high-performance tridiagonal solver using GPUs. In Proc. Int'l. Conf. High Performance Computing, Networking Storage and Analysis, SC '12, pages 27:1-27:11, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.
- [2] E. Gallopoulos, B. Philippe, and A.H. Sameh. *Parallelism in Matrix Computations*. Springer, 2015.
- [3] A.H. Sameh and D.J. Kuck. On stable parallel linear system solvers. *J. Assoc. Comput. Mach.*, 25(1):81-91, January 1978.
- [4] I. E. Venetis, A. Sobczyk, A. Kouris, A. Nakos, N. Nikoloutsakos, and E. Gallopoulos. A general tridiagonal solver for coprocessors: Adapting g-Spike for the Intel Xeon Phi. In Proc. of 2015 Int'l. Conf. on Parallel Computing (ParCo 2015) (to appear), Edinburgh, UK, Sept. 2015.
- [5] I.E. Venetis, A. Kouris, A. Sobczyk, E. Gallopoulos, and A.H. Sameh. A direct tridiagonal solver based on Givens rotations for GPU architectures. *Parallel Computing*, 49:101-116, Nov. 2015.

Keywords: tridiagonal system, vectorization, Intel Xeon Phi

Streamwise numbering for Gauss-Seidel and bidiagonal preconditioners in convection dominated flows

Paula Córdoba * ¹, Guillaume Houzeaux ¹, Juan Carlos Cajas ¹

¹ Barcelona Supercomputing Center (BSC-CNS) – Edificio Nexus II, Campus Nord UPC, Jordi Girona 29-31, 08034 Barcelona, Spain

Choosing the correct preconditioner when solving complex physical problems for both, applied fields and basic research may be a hard task and several aspects have to be taken into account. One of them is the physics of the problem studied, as the weights of the sparse matrix highly depend on this. In the present contribution, the construction, implementation and results of a closely-related-to-the-physics preconditioners for convection dominated problems is studied. In this case, the information propagates mainly in the direction of advection. Then if the discretization of the problem is done using a finite element method, it can be shown that the main contribution in every row to a certain node of the resultant matrix, apart from the diagonal term, comes from the previous neighbouring node which has the closest direction to the velocity of the flow direction. Thereby, a mesh node numbering along the flow direction (streamwise direction) is proposed in such a way that the main weights of each node will lie on the first sub-diagonal. Knowing this, several numerical examples in two and three dimensions have been tested using both, Gauss-Seidel and Bidiagonal preconditioning together Krylov subspace methods, in particular the GMRES and BiCGSTAB solvers are used. The examples have been executed in sequential and in parallel and compared between them. For the parallel case, the parallelization is done with MPI. At this level a Jacobi preconditioner is used on the interface nodes and the Gauss-Seidel and Bidiagonal ones are only used to precondition the interior nodes, as the interfaces cut the advection lines.

Keywords: Sparse linear systems, Preconditioning, Gauss Seidel, Bidiagonal, Krylov Subspace methods, Streamwise direction

*Speaker

SpMV Algorithm for High Bandwidth Memory Based Application Specific Hardware Accelerator

Fazle Sadi ^{*† 1}, Franz Franchetti , Larry Pileggi

¹ Department of Electrical and Computer Engineering - Carnegie Mellon University (CMU) – Pittsburgh, Pennsylvania 15213-3890, United States

Sparse Matrix-Vector multiplication (SpMV) is one of the most important kernels for numerous scientific applications. However, SpMV is inherently a memory bound problem due to its very low ratio of computation to memory access. This renders commercial off-the-shelf (COTS) architectures unsuitable for SpMV as the main memory bandwidth is insufficient for available computation resources on these platforms, well known as the memory wall problem. On top of that, SpMV requires random access into a memory space which is far too big for cache. Hence, it becomes difficult to utilize the main memory bandwidth which is already scarce.

With the advent of 3D IC technology, stacked DRAM such as High Bandwidth Memory (HBM) can be deemed as a solution to the memory wall problem. However, full utilization of this high bandwidth in an energy efficient manner still remains as a challenge for COTS architectures. On the other hand, advanced circuit level techniques such as Logic in Memory (LiM) based hardware accelerators are recently being used to provide low power solution to big data problems. Moreover, interposer technology has become available to provide high speed channel between compute core and memory.

Given these advancements in hardware, we propose a new algorithm for large SpMV problems which is specially optimized to fully exploit the underlying micro-architecture and overall system capabilities. This algorithm is implemented in two steps. The key feature of the first step is that it converts all the main memory random access into streaming access. This reduces the overall data transfer volume significantly and ensures full utilization of the memory bandwidth. On top of that, we propose a meta-data compression technique, namely Variable Length Delta Index (VLDI), to decrease the data transfer volume even further. VLDI is particularly effective

*Speaker

†Corresponding author: fsadi@cmu.edu

for sparse matrices where meta-data to payload ratio is high, e.g. sparse bit matrices.

After the first step, the SpMV problem effectively converts into a big merge-sort problem. In the second step, we demonstrate a novel fast merge-sort algorithm, namely Propagated Active Path (path) merge-sort. This algorithm helps to consume the all the bandwidth provided by HBM and maintain high throughput at the merge-sort core end. PAP merge-sort is independent of the problem size. Hence, the problem size can be easily scaled without sacrificing performance and efficiency. Furthermore, the hardware implementation of PAP merge-sort is enhanced to provide wide output interface. This increases the throughput of the computation core linearly which is especially helpful for systems with multiple HBM.

The overall hardware system for the proposed SpMV algorithm is also demonstrated. It constitutes of application specific LiM based accelerator core, eDRAM (embedded DRAM) scratchpad, 3D stacked DRAM and interposer platform. To test the effectiveness of the proposed SpMV algorithm on the accelerator system, we investigate the performance and energy efficiency of various COTS architectures using finely tuned standard libraries such as Intel's Math Kernel Library (MKL). Our experimental results show that the proposed algorithm, along with the data compression and fast merge-sort technique, implemented on the application specific hardware can achieve at least two orders of magnitude improvement in performance and energy efficiency over the available COTS architectures.

Keywords: Sparse matrix vector, merge sort, accelerator, logic in memory, meta data compression

Author Index

- Adlerborn, Björn, 24
Agullo, Emmanuel, 23, 78
Alvermann, Andreas, 90
Amestoy, Patrick, 81
Augonnet, Cedric, 85
August, Moritz, 46
Axelsson, Owe, 103
- Béreux, Natacha, 83
Bañuls, Mari Carmen, 46
Basermann, Achim, 90
Berljafa, Mario, 31
Berzins, Martin, 18
Blaheta, Radim, 103
Boiteau, Olivier, 83
Bollhoefer, Matthias, 76, 106
Boman, Erik, 86
Bosilca, George, 20
Bosner, Nela, 52
Boukaram, Wajih, 62
Brzobohaty, Tomas, 87
Bujanović, Zvonimir, 52
Buttari, Alfredo, 23, 81
- Córdoba, Paula, 110
Cajas, Juan Carlos, 110
Campos, Carmen, 91
Carpentieri, Bruno, 28
Chen, Chao, 86
Chun, Huang, 60
- D’Ast, Isabelle, 42
Dai, William, 40
Darve, Eric, 78, 86
Di Napoli, Edoardo, 8, 47
Donfack, Simplicie, 50
Dongarra, Jack, 55
Duff, Iain, 21
Dumont, Thierry, 43
- Faverge, Mathieu, 19
Fehske, Holger, 90
Flegar, Goran, 58
Fournier, Cyril, 42
Franchetti, Franz, 111
- Futamura, Yasunori, 11, 66
- Galgon, Martin, 90
Gallopoulos, Efstratios, 108
Garcia, Daniel, 34
Gates, Mark, 55
Gavin, Brendan, 9
Ghysels, Pieter, 82
Giraud, Luc, 78
Gordon, Dan, 104
Gordon, Rachel, 104
Goudin, David, 85
Grimes, Roger, 33
Guermouche, Abdou, 23
Guettel, Stefan, 31
Guillet, Thomas, 43
Gurgul, Grzegorz, 34
- Hager, Georg, 90
Haidar, Azzam, 55
Hapla, Václav, 39
HARNESS, Yuval, 78
Hirota, Yusuke, 70
Hogg, Jonathan, 21
Horák, David, 39
Houzeaux, Guillaume, 110
Huckle, Thomas, 46
- Imakura, Akira, 11, 93
Imamura, Toshiyuki, 64, 70
Inoue, Yuto, 66
- Jacquelin, Mathias, 74
Jie, Liu, 60
- Kågström, Bo, 24
Karlsson, Lars, 24
Karypis, George, 97

Kaya, Oguz, 96
 Keyes, David, 62
 Kimura, Kinji, 68
 Kjelgaard Mikkelsen, Carl Christian,
 24
 Kouris, Alexandros, 108
 Kozubek, Tomas, 87
 Krämer, Lukas, 90
 Kreutzer, Moritz, 30
 Kružík, Jakub, 39
 Kudo, Shuhei, 57

 L'Excellent, Jean-Yves, 81
 Lacoste, Xavier, 85
 Lamas Daviña, Alejandro, 30
 Lang, Bruno, 90
 Li, Shengguo, 60
 Li, Xiaoye, 82
 Lin, Lin, 74
 Lin, Shuliang, 60
 Liu, Weifeng, 99
 Lopez, Florent, 21, 23
 Louvet, Violaine, 43
 Ltaief, Hatem, 22, 62
 Luber, Tomas, 103
 Lucas, Bob, 33

 Markopoulos, Alexandros, 87
 Mary, Theo, 81
 Meca, Ondrej, 87
 Meerbergen, Karl, 7
 Michiels, Wim, 7

 Nakos, Alexandros, 108

 Oksa, Gabriel, 56
 Olaf Schenk, Klaus Gaertner,, 75

 Pardo, David, 34
 Park, Jongsoo, 97
 Pask, John, 15
 Paszynski, Maciej, 34
 Pieper, Andreas, 90
 Pileggi, Larry, 111
 Polizzi, Eric, 9

 Pouransari, Hadi, 86
 Prat, Raphael, 43

 Röhrig-Zöllner, Melven, 90
 Rajamanickam, Siva, 86
 Reps, Bram, 50
 Riha, Lubomir, 87
 Rippl, Michael, 59
 Roman, Jose E., 30, 91
 Romero Alcalde, Eloy, 10
 Rouet, Francois-Henry, 60
 Roux, François-Xavier, 36

 Saad, Yousef, 73, 92
 Sadi, Fazle, 111
 Sakurai, Tetsuya, 11, 66, 93
 Sameh, Ahmed, 12
 Sanan, Patrick, 50
 Schenk, Olaf, 50, 76, 106
 Shahzad, Faisal, 90
 Singer, Sanja, 54
 Smith, Shaden, 97
 Sosonkina, Masha, 28
 Springer, Paul, 47
 Staffelbach, Gabriel, 42
 Stathopoulos, Andreas, 10
 Sterling, Thomas, 17
 Suzuki, Atsushi, 36
 Sylvand, Guillaume, 79

 Thies, Jonas, 90
 Tomov, Stan, 55
 Turkiyyah, George, 62

 Uçar, Bora, 96

 Vajtersic, Marian, 56
 Van Beeumen, Roel, 7
 VandeVondele, Joost, 13
 Vanroose, Wim, 27, 50
 Vecharynski, Eugene, 15
 Venetis, Ioannis, 108
 Verbosio, Fabio, 76
 Vinter, Brian, 99

 Wang, Zhu, 51

Weisbecker, Clement, 33
Wellein, Gerhard, 90
Winkelmann, Jan, 8, 47
Wouters, Michiel, 27

Xingyu, Gao, 60

Yamamoto, Yusaku, 56, 57
Yang, Chao, 15, 74
Yzelman, Albert-Jan, 100

