Sparse Days

June 17-18th, 2013

CERFACS, Toulouse, France

Monday, June 17th 2013

9.30 - 10.00 Registration and welcome coffee

Session I

10.00 - 10.30 *Using multiple breadth-first search to find separators of a graph*
C. Ashcraft (LSTC, USA)

10.30 - 11.00 *Scalable matrix computations on large scale-free graphs using 2D graph partitioning*
E. Boman (Sandia National Laboratories, USA)

11.00 - 11.15 Coffee break

11.15 - 11.45 *A robust limited-memory incomplete Cholesky factorization*
J. Scott (Rutherford Appleton Laboratory (RAL), UK)

11.45 - 12.15 *Communication avoiding ILU(0) preconditioner*
S. Moufawad (INRIA Rocquencourt, France)

Lunch

Session II

14.00 - 14.30 *Sparse inverse incidence matrices in the factorization of indefinite matrices*
W. Schilders (TU Eindhoven, The Netherlands)

14.30 - 15.00 *Solving sparse linear systems in a many-core environment*
M. Byckling (CSC - IT Centre for Science Ltd., Finland)

15.00 - 15.30 *Improvements to the Augmented Block Cimmino Distributed method*
M. Zenadi (ENSEEIHT-IRIT, France)

15.30 - 16.00 Coffee break

Session III

16.00 - 16.30 *A Sparse BLAS implementation using the "Recursive Sparse Blocks" layout*
M. Martone (Max Planck Institute for Plasma Physics, Garching, Germany)

16.30 - 17.00 *MKL Sparse BLAS: performance optimizations on modern architectures*
S. Pudov (Intel, Russia)

17.00 - 17.30 *Recovery policies for Krylov solver resiliency*
M. Zounon (INRIA Bordeaux Sud-Ouest, France)

19.30 Banquet at the restaurant Le Pôvre Yves in Toulouse.

Tuesday, June 18th 2013

Session IV

9.15 - 9.45 *Preconditioning PDE-constrained optimization problems*
A. Wathen (Oxford University, UK)

9.45 - 10.15 *GMRES convergence for non-normal matrices*
G. Meurant (France)

10.15 - 10.45 Coffee break

Session V

10.45 - 11.15 *Multilevel low rank approximation preconditioners*
Y. Saad (University of Minnesota, USA)

11.15 - 11.45 *Block Low-Rank (BLR) approximations to improve multifrontal sparse solvers*
C. Weisbecker (ENSEEIHT-IRIT, France)

11.45 - 12.05 *Towards an optimal-order approximate sparse factorization exploiting data-sparseness in separators*
X. S. Li (Lawrence Berkeley National Laboratory, USA)

12.05 - 12.35 *A direct-iterative hybrid block linear solver for discontinuous-Galerkin finite-element equations*
S. Murphy (CERFACS and University of Nottingham, France and UK)

Lunch

Session VI

14.00 - 14.30 *Intel direct sparse solver for clusters, a research project for solving large sparse systems of linear algebraic equations on clusters*
A. Kalinkin (Intel, Russia)

14.30 - 15.00 *Work stealing and granularity optimizations for a sparse solver on manycores*
X. Lacoste (INRIA Bordeaux Sud-Ouest, France)

15.00 - 15.30 *Parallel design and performance of nested filtering factorization preconditioner*
L. Qu (INRIA Rocquencourt, France)

15.30 - 16.00 *Approximate inverse preconditioning with adaptive dropping*
J. Kopal (Technical University of Liberec, Czech Republic)

Closure

June 17th - Session I

Using multiple breadth-first search to find separators of a graph
C. Ashcraft (Livermore Software Technology Corporation, USA).

Automatic nested dissection of George and Liu (1981) used the level structures generated by a breadth-first search from a pseudo-peripheral vertex. The nodes of each level form a separator of the graph. The level structure maps the graph into a compressed graph who has a weighted tridiagonal adjacency matrix. They found the best partition of this compressed graph, and mapped it back to the original graph.

We extend this idea to use a pair of vertices, and drop breadth-first searches from each of them independently. The difference of their levels forms a map of vertices to "half-level" sets. This difference map induces a compressed graph with a weighted pentadiagonal adjacency matrix. Again, we find the best partition of this compressed graph, and map it back to the original graph.

Separators obtained from single level sets (George and Liu 1981) are minimal — each separator vertex is adjacent to two domains. Separators obtained from half-level sets need not be minimal, and some sort of improvement step must follow to find a minimal separator.

George and Liu (1981) executed a sequence of breadth-first searches to find a pseudo-peripheral node. One can do the same to find a "pseudo-diameter" pair of vertices. The last two in the sequence to find a peripheral node are a good pair.

In contrast, execute multiple independent breadth–first searches from a set of random nodes. Now take the difference of the level structures of each pair of nodes and evaluate its best partition. If we have k level structures, then we have $k(k-1)/2$ pairs, and so for the work of k breadth–first searches we can generate $k(k-1)/2$ partitions and choose the best.

A MPP implementation is challenging due to the number of collective communications. We describe an approach linear in the diameter of the graph, linear in the maximum "width" of a non-minimal separator, but independent of the number of breadth–first searches.

This is joint work with Roger Grimes (Livermore Software Technology Corporation).

Scalable matrix computations on large scale-free graphs using 2D graph partitioning
E. Boman (Sandia National Laboratory, USA).

Scalable parallel computing is essential for processing large scale-free (power-law) graphs. The distribution of data across processes becomes important on distributed-memory computers with thousands of cores. It has been shown that two-dimensional layouts (edge partitioning) can have significant advantages over traditional one-dimensional layouts. However, simple 2D block distribution does not use the structure of the graph, and more advanced 2D partitioning methods are too expensive for large graphs. We propose a new two-dimensional partitioning algorithm that combines graph partitioning with 2D block distribution. The computational cost of the algorithm is essentially the same as 1D graph partitioning. We study the performance of sparse matrix-vector multiplication (SpMV) for scale-free graphs from the web and social networks using several different partitioners and both 1D and 2D data layouts. We show that SpMV run time is reduced by exploiting the graph's structure. Contrary to popular belief, we observe that current graph and hypergraph partitioners often yield relatively good partitions on scale-free graphs. We demonstrate that our new 2D partitioning method consistently outperforms the other methods considered, for both SpMV and an eigensolver, on matrices with up to 1.6 billion nonzeros using up to 16,384 cores.

A robust limited-memory incomplete Cholesky factorization

J. Scott (Rutherford Appleton Laboratory (RAL), UK).

Incomplete Cholesky (IC) factorizations have long been an important tool in the armoury of methods for the numerical solution of large sparse symmetric linear systems $Ax = b$. This talk focuses on the design and development of a new robust and efficient general-purpose incomplete Cholesky factorization package HSL_MI28, which is available within the HSL mathematical software library. It implements a limited memory approach that exploits ideas from the positive semidefinite Tismenetsky-Kaporin modification scheme and, through the incorporation of intermediate memory, is a generalisation of the widely-used ICFS algorithm of Lin and More. Both the sparsity density of the incomplete factor and the amount of memory used in its computation are under the users control. The performance of HSL_MI28 is demonstrated using extensive numerical experiments involving a large set of test problems arising from a wide range of real-world applications. They illustrate the significant advantage of employing a modest amount of intermediate memory and that, with limited memory, high quality yet sparse general-purpose preconditioners are obtained.

This is joint work with M. Tuma (Institute of Computer Science, Academy of Sciences of the Czech Republic).

Communication avoiding ILU(0) preconditioner

S. Moufawad (INRIA Rocquencourt, France).

We present a communication avoiding ILU(0) preconditioner for solving large systems of linear equations by using iterative Krylov subspace methods. Recent research has focused on communication avoiding Krylov subspace methods based on so called s-step methods. However there is no communication avoiding preconditioner yet, and this represents a serious limitation of these methods. Our preconditioner allows to perform s iterations of the iterative method with no communication, through ghosting some of the input data and performing redundant computation. It thus reduces data movement by a factor s between different levels of the memory hierarchy in a serial computation and between different processors in a parallel computation. To avoid communication, an alternating reordering algorithm is introduced, that requires the input matrix to be ordered by using nested dissection or kway partitioning. We show that the reordering does not affect the convergence rate of the ILU(0) preconditioned system as compared to nested dissection and kway ordering, while it reduces data movement and should improve the expected time needed for convergence.

This is joint work with L. Grigori (INRIA Rocquencourt, France).

June 17th - Session II

Sparse inverse incidence matrices in the factorization of indefinite matrices

W. Schilders (TU Eindhoven, The Netherlands).

Some time ago, we developed a new factorization method for indefinite matrices. This can also be used as a basis for preconditioning indefinite linear systems which arise in many problems like least-squares, saddle-point and electronic circuit simulations. Here we consider its application to resistor network modeling. The sparsity of the matrix blocks in our factorization depends on the sparsity of the inverse of a permuted incidence matrix. We introduce three different possible permutations, and determine the permutation which leads to the sparsest inverse of the incidence matrix. We show several examples of the application of this technique to problems from the electronics industry.

This is joint work with S. Lungten and J. Maubach (TU Eindhoven, The Netherlands).

Solving sparse linear systems in a many-core environment

M. Byckling (CSC - IT Centre for Science Ltd., Finland).

We consider solving sparse linear systems in a many-core environment. In a typical HPC cluster, the number of cores per node is steadily increasing. In addition, with new hardware architectures such as the Intel Xeon Phi, the number of available threads has been increased by almost one order of magnitude, with the expense of a single thread being much slower than a traditional core.

We study if traditional techniques, both direct and iterative, to solve sparse linear systems are at all applicable in such environments. In particular, we consider parallel preconditioners based on sparse approximate inverses. Can sparse flops be free?

This is joint work with M. Huhtanen (University of Oulu, Finland).

References

[1] M. Byckling and M. Huhtanen. Approximate factoring of the inverse. *Numerische Mathematik*. 1–22. 2010.

[2] E. Chow. A Priori Sparsity Patterns for Parallel Sparse Approximate Inverse Preconditioners. *SIAM J. Sci. Comput.*. 21(5):1804–1822. 1999.

[3] I. S. Duff and K. Kaya. Preconditioners based on Strong Components. *CERFACS Technical Report*. 1–32. 2010.

[4] I. S. Duff and J. Koster. On Algorithms For Permuting Large Entries to the Diagonal of a Sparse Matrix. *SIAM J. Matrix Anal. Appl.*. 22(4):973–996. 2001.

[5] D. Fritzsche and A. Frommer and D. B. Szyld Extensions of Certain Graph-based Algorithms for Preconditioning *SIAM Journal on Scientific Computing*. 29(5):2144–2161. 2007.

Improvements to the Augmented Block Cimmino Distributed method

M. Zenadi (ENSEIIHT-IRIT, France).

The Augmented Block Cimmino Distributed method is an evolution of the block Cimmino algorithm. We augment the original system such that the partitions of the matrix are mutually orthogonal. Iteration matrix issued from such a system is an orthogonal projector, this makes the method converge in a single iteration. However, it requires building an auxiliary system and solving it using a direct solver. In this talk we study the different approaches to solve the auxiliary system by building it in parallel then solving it using a direct solver, or by solving it using conjugate gradients iterations without building it.

June 17th - Session III

A Sparse BLAS implementation using the "Recursive Sparse Blocks" layout

M. Martone (Max Planck Institute for Plasma Physics, Garching, Germany).

Sparse matrix computations arise in many scientific computing problems and often operations such as Sparse Matrix-Vector Multiplication (SPMV) can dominate the total computation time. These computations are generally characterized by irregular memory accesses and low compute-to-load/store ratio. Recent general purpose CPUs (cache based, shared memory parallel) perform poorly, because of relative high latency in cache/memory access and decreasing memory-to-compute bandwidths' ratio. Recent experiments suggest that by combining cache-blocking techniques (leading to space-filling-curve block layout of data), and coarse-grained shared memory parallelism it is possible to execute SPMV of large sparse matrices more than twice as fast than with Intel MKL's industry-standard, highly optimized CSR (Compressed Sparse Rows) implementation. Additionally, in contrast to the CSR format, our format (RSB — Recursive Sparse Blocks) enables a higher parallelism of operations like transposed or symmetric SPMV or sparse triangular solve. This talk will introduce main ideas, results and perspectives in using the RSB format in a Sparse BLAS (Basic Linear Algebra Subroutines) context, according to the recently released "librsb" library.

MKL Sparse BLAS: performance optimizations on modern architectures

S. Pudov (Intel, Russia).

Sparse BLAS was integrated into Intel (R) MKL 8.0 and it was continuously optimized for upcoming architectures. MKL Sparse BLAS supports many sparse formats and is comprised of Fortran-style interfaces. The interfaces are very universal; they support a lot of interesting features like usage sparse sub-matrices, etc. Also computational kernels make almost no assumptions about sparse data storage except for several cases (e.g. solving routines). Current MKL key functionality demonstrates substantial performance results. But this all-mode approach resulted in limited numbers of supported optimization algorithms. It is well known that effective optimization of sparse computations requires preliminary information about input matrices: the more information is available the better algorithm can be chosen. This information is usually time-consuming and cannot be found for every function call. Traditionally computations are divided into two steps analysis and execution so matrix is analyzed only once and this information is used in further function calls. For iterative methods and for algorithms with the same matrix sparsity structure this approach can provide a lot of benefits. Hereby current MKL Sparse BLAS universal interfaces come into conflict with optimizations on modern parallel many-core architectures because they are unable to take into account sparse matrix specifics. To resolve this issue a research project has been initiated in MKL for developing new Sparse BLAS interfaces that should support two-steps execution mode for providing auto-tuning functionality. We mainly investigate SpMV routines as they are most actively used. Several experiments on modern many-core architectures have been performed. Performance results are very promising but there are several options for final interface implementation to be chosen based on customer usage analysis.

Recovery policies for Krylov solver resiliency

M. Zounon (INRIA Bordeaux Sud-Ouest, France).

The advent of exascale machines will require the use of parallel resources at an unprecedented scale with potentially billions of computing units leading to a high rate of hardware faults. High Performance Computing applications that aim at exploiting all these resources will thus need to be resilient, i.e., being able to eventually compute a correct output in presence of core faults. Contrary to checkpointing techniques or Algorithm Based Fault Tolerant (ABFT) mechanisms, strategies based on interpolation for recovering lost data do not require extra work or memory when no fault occurs. We apply this latter strategy to Krylov iterative solvers for systems of linear equations, which are often the most computational intensive kernels in HPC simulation

codes. We propose and discuss several variants able to possibly handle multiple simultaneous faults. We study the impact of the recovery methods, the fault rate and the number of processors on the resilience of the most popular solvers that are CG, GMRES and BiCGStab solvers.

June 18th - Session IV

Preconditioning PDE-constrained optimization problems

A. Wathen (Oxford University, UK).

Many control problems for PDEs can be expressed as Optimization problems with the relevant PDEs acting as constraints. As is being discovered in other areas such as multi-physics, there seem to be distinct advantages to tackling such constrained Optimization problems 'all-at-once' or with a 'one-shot' method. That is, decoupling of the overall problem in some loosely coupled iterative fashion appears to be a rather poorer approach than to compute on the fully coupled problem.

The use of iterative methods for the relavant linear algebra is crucial here since the overall dimensions (including the Optimization and PDE) are usually very large, but matrix vector products as required in Krylov subspace methods such as MINRES are still readily computed. The work to ensure rapid convergence is in preconditioning and it is this topic that we will mostly focus on in this lecture.

GMRES convergence for non-normal matrices

G. Meurant (France).

At Sparse Days 2012 we explained how an expression for the GMRES residual norms can be found for normal matrices. This expression involved products of differences of eigenvalues and the projections of the right-hand side on the eigenvectors. In this talk we will show how this technique can be extended to non-normal matrices. For diagonalizable matrices we obtain an expression involving products of differences of eigenvalues, the right-hand side and determinants of submatrices of the matrix of the eigenvectors. Things are more tricky when the matrix is not diagonalizable. Using the Jordan canonical form we will explain what are the difficulties and we will provide some partial results about the residual norms. A case that can be completely handled is when the matrix is a single scaled Jordan block. Then elegant expressions for the residual norms can be obtained.

June 18th - Session V

Multilevel low rank approximation preconditioners

Y. Saad (University of Minnesota, USA).

A new class of preconditioning methods for solving linear systems of equations will be introduced. These methods, which are based on exploiting low-rank approximations to certain matrices, have a number of appealing features. They handle indefiniteness quite well and they are amenable to SIMD compuations such those inherent to GPUs. We will first describe the methods for Symmetric Positive Definite model problems arising from Finite Difference discretizations of PDEs. We will then show how to extend them to general situations, by exploiting the domain decomposition framework.

Block Low-Rank (BLR) approximations to improve multifrontal sparse solvers

C. Weisbecker (ENSEEIHT-IRIT, France).

Matrices coming from elliptic Partial Differential Equations (PDEs) have been shown to have a low-rank property: well defined off-diagonal blocks of their Schur complements can be approximated by low-rank products. In the multifrontal context, this can be exploited within the fronts in order to obtain a substantial reduction of the memory requirement and an efficient

way to perform many of the basic dense algebra operations. Many hierarchical representations have been proposed in the literature to benefit from these low-rank compressions, mainly the H and HSS matrices. We propose a flexible, flat format, called Block Low-Rank, designed to be easily adapted to any robust algebraic multifrontal solver without conceding the versatility of such a solver. The memory requirement and the operation count of the multifrontal process could be divided by a factor of at least 2 and 5, respectively, on various applicative problems such as structural mechanics and time-harmonic wave equations for full waveform inversion.

Towards an optimal-order approximate sparse factorization exploiting data-sparseness in separators

X.S. Li (Lawrence Berkeley National Laboratory, USA).

Nested dissection ordering and its graph partitioning generalization give rise to an ordered sequence of separators of (roughly) geometrically decreasing size. The fill-ins in the LU factros are confined in the parts of the matrix associated with the separators. In particular, the diagonal blocks associated with the separators are fully dense in the factors and they contribute to the dominant terms in the costs of the storage for the factors and the flops for computing the factors. Employing the data-sparse representations or compressions, such as low-rank approximation, for these separator blocks can drastically lower the overall factorization costs both in memory and flops. The low rankness can appear in many matrices from discretized PDEs.

Recently, we have been investigating fast and stable algorithms for one type of data-sparse representation called hierarchically semi-separable (HSS) structure, and using them in the sparse LU factorization. In this talk, we will show that both in theory and in practice, the HSS-embedded saprse LU factorization has much lower complexity than the traditional factorization algorithm. The complexity of this class of algorithms is closely related to the numerical ranks of the separator blocks which vary with the PDEs. For elliptic problems, we can achieve large amount of compression and the resulting factorization can be used as a nearly optimal-order direct solver. For wider classes of problems, the approximate factorization can be used as a preconditioner. We will show performance results from direct solvers as well as preconditioners for a wide range of problems. We will also illustrate the potential of such solvers/preconditioners being used for the extreme-scale computers and problem size.

This is joint work with A. Napov, F.-H. Rouet, S. Wang, and J. Xia.

A direct-iterative hybrid block linear solver for discontinuous-Galerkin finite-element equations

S. Murphy (CERFACS, University of Nottingham, France and UK).

Discontinuous-Galerkin (DG) finite-element methods are widely used for obtaining higher order numerical solutions to partial differential equations in areas such as electrodynamics, fluid dynamics and mathematical physics. Frequently the most computationally intensive step when implementing a DG method is finding the solution to a large sparse structurally symmetric block matrix with dense blocks, which is not necessarily numerically symmetric or definite. The pattern of the block structure corresponds directly to the finite-element mesh over the spatial domain with a diagonal block for each element and blocks above and below the diagonal for each interface between two neighbouring elements. The size of the diagonal blocks correspond to the order of approximation for each element, which leads to irregular block sizes when using *p*- and *hp*-refinement.

We introduce a hybrid linear solver, designed for DG methods, that combines aspects of iterative, dense direct and sparse direct methods. At one extreme, the solver uses `GMRES` pre-conditioned by dense solves on the subdomains corresponding to each element in the mesh. Adjusting a control parameter permits the size of these subdomains to increase by agglomerating neighbouring subdomains. When these subdomains become sufficiently large, the solver switches from using dense code to using sparse code. At the other extreme, there is only one subdomain, and therefore the strategy can be viewed on a sliding scale from a fully iterative method, through a hybrid method, to a fully direct sparse linear solver. A further parameter controls the overlap between neighbouring subdomains with the contributions from the overlapping regions summed in the manner of an additive-Schwarz algorithm. A novel feature of

our approach is to use the `ANALYSE` phase of a multifrontal solver on the block structure of the matrix to determine which elements to include in each subdomain.

We look at results obtained from the solution of the indefinite saddle-point matrix arising from a DG discretisation for Stokes equations. We consider the results of using the solver as an alternative to the more common strategy of considering the pressure and velocity components for the full domain and using an approximate Schur complement for the pressure component. We use our solver to obtain a solution for both pressure and velocity variables on a collection of subdomains of various sizes with varying amounts of overlap. We present results from a parallel implementation of the solver, and compare its performance with a selection of established direct and iterative linear solvers.

This is joint work with I. Duff (CERFACS, RAL).

June 18th - Session VI

Intel direct sparse solver for clusters

A. Kalinkin (Intel, Russia).

This research explores the Intel Direct Sparse Solver for Clusters package which implements a direct method to solve equation $Ax = b$ with sparse matrix A on a cluster. This method has been developed by Intel and is based on "multifrontal" approach to Cholesky decomposition to achieve good work balance on a large number of processes. The parallelization approach is based on a dependency tree and consists of using MPI for communications between cluster nodes and using OpenMP directives to use shared memory parallelism available within a node.

Work stealing and granularity optimizations for a sparse solver on manycores

X. Lacoste (INRIA Bordeaux Sud-Ouest, France).

The current trend in computer architecture shows an escalation in the number, as well as in the heterogeneity, of the computing resources. In order to remain efficient, algorithms, especially those in critical domains such as linear algebra, need to be adapted to match the latest advances in computer architecture.

PaStiX is a sparse parallel direct solver, based on a dynamic scheduler for modern hierarchical architectures. In this presentation, we study the replacement of the internal highly specialized scheduling in PaStiX by two generic runtime frameworks: PaRSEC and StarPU.

The task graph of the factorization step will be made available to the two runtimes, providing them with the opportunity to optimize it in order to minimize the execution time and maximize the resource utilization (including accelerators). We present the expression of the factorization in a task based manner, the kernels used for the GPU computations and the DAG algorithms used with the two runtimes. A comparative study of the performance of the supernodal solver with the three different schedulers is performed on different architectures. The results demonstrate that these generic DAG-based runtimes can obtain performance similar to specialized schedulers while providing uniform and portable programming interfaces.

This is joint work with .

Parallel design and performance of nested filtering factorization preconditioner

L. Qu (INRIA Rocquencourt, France).

In this talk, we present the parallel design and performance of the nested filtering factorization preconditioner (NFF). This preconditioner can be used for solving linear systems arising from the discretization of a system of PDEs on unstructured grids. It is based on a recursive decomposition that exploits a bordered block diagonal structure of the input matrix, obtained priorly by using graph partitioning techniques. It also allows to preserve several directions of interest of the input matrix to alleviate the effect of low frequency modes on the convergence of iterative methods. Due to its recursive formulation, NFF has limited memory requirements and

it is also naturally suitable for hierarchical parallel machines. We show experimentally its convergence rate and its time to solution on a boundary value problem with highly heterogeneous coefficients, discretized on three-dimensional grids.

This is joint work with L. Grigori (INRIA Rocquencourt, France).

Approximate inverse preconditioning with adaptive dropping

J. Kopal (Technical University of Liberec, Czech Republic).

The contribution deals with an approximate inverse preconditioning for the conjugate gradient method. The main goal is to compute the decomposition that would be sparse and reliable at the same time. In particular, the generalized Gram–Schmidt process with an adaptive dropping strategy is considered.

Assume the system of linear algebraic equations in the form $Ax = b$, where A is symmetric and positive definite. Symmetrically preconditioned system can be written in the form

$$\tilde{Z}^T A \tilde{Z} y = \tilde{Z}^T b, \qquad x = \tilde{Z} y,$$

where $\tilde{Z}$ is the factor of the approximation $\tilde{Z}\tilde{Z}^T$ to A^{-1}, that plays the role of the preconditioner.

The generalized Gram–Schmidt process in exact arithmetic provides matrices Z and U, so that $U^T U = (Z^{(0)})^T A Z^{(0)}$, $Z^T A Z = I$, and $ZU = Z^{(0)}$. The columns of the matrix $Z^{(0)}$ are initial vectors that are A-orthogonalized against previously computed vectors. Matrix U contains the orthogonalization coefficients. It is clear that U is the Cholesky factor of $A = U^T U$ for $Z^{(0)} = I$. In practice, the effects of the finite precision arithmetic should be considered. The corresponding bounds for the norms $\|\bar{Z}^T A \bar{Z} - I\|$, $\|\bar{Z}\bar{U} - I\|$, $\|\bar{U}^T \bar{U} - (Z^{(0)})^T A Z^{(0)}\|$ were derived in [1], where the approximate quantities (obtained by dropping) are denoted with an extra tilde.

The results in [1] motivate a new scheme where the accuracy of computed column vectors in $\tilde{Z}$ reflect the level of errors committed throughout previous factorization steps. Namely, the entries that have magnitudes smaller than the current level of the error can be dropped in any case. This dropping level is a function of the theoretical error bound for orthogonalization scheme. The whole scheme can be interpreted as a computation with the roundoff unit much larger than the standard $u = \epsilon/2$. Our implementation uses additional techniques that can significantly enhance numerical properties of the preconditioner and its sparsity, namely pivoting and diagonal scaling. In particular, these techniques support to achieve the main goal by approximate minimization of the conditioning of $\bar{U}$.

The theoretical results will be accompanied by carefully chosen experiments that demonstrate usefulness of the approach. We hope that the developed algorithm may extend scope of applicability of the considered type of approximate inverse preconditioners.

This is joint work with M. Rozložník (Technical University of Liberec, Liberec) and M. Tůma (Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague).

Acknowledgement: This work was supported by Grant Agency the Czech Republic under the project 108/11/0853 and the project P201/13-06684S of the Grant Agency of the Czech Republic.

References

[1] M. Rozložník, M. Tůma, A. Smoktunowicz, J. Kopal: "Numerical stability of orthogonalization methods with a non-standard inner product", BIT Numerical Mathematics, 52 (4), pp 1035-1058, 2012.