# Electronic Transactions on Numerical Analysis
# Volume 21, 2005

## Contents

**Abstract.**
This paper introduces a new model to describe the flow of data from update to frontal matrix in the unsymmetric multifrontal method for solving sparse linear systems. The model is based on the elimination tree of an unsymmetric matrix and consists of the edges in this tree together with some cross edges.
By symmetrically renumbering the rows and columns of the coefficient matrix using a tree-based postordering, we can permute the matrix into a bordered block triangular form while preserving the elimination tree. The model simplifies when the matrix has this form, which suggests that a multifrontal implementation based on it should be simpler as well.
We also extend the model to handle pivoting for stability; compare it with others used in the unsymmetric multifrontal method; and point out the implications for parallelism.

**Key Words.**
sparse matrix factorization, sparse LU decomposition, elimination tree

**AMS(MOS) Subject Classifications.**
65F05, 65F50

**Abstract.**
Any matrix $A$ of dimension $m \times n$ ($m \geq n$) can be reduced to upper triangular form by multiplying by a sequence of $mn - n(n+1)/2$ appropriately chosen rotation matrices. In this work, we address the question of whether such a factorization exists when the set of allowed rotation planes is restricted. We introduce the rotation graph as a tool to devise elimination orderings in QR factorizations. Properties of this graph characterize sets of rotation planes that are sufficient (or sufficient under permutation) and identify rotation planes to add to complete a deficient set. We also devise a constructive way to determine all feasible rotation sequences for performing the QR factorization using a restricted set of rotation planes. We present applications to quantum circuit design and parallel QR factorization.

**Key Words.**
QR decomposition, Givens rotations, plane rotations, parallel QR decomposition, quantum circuit design, qudits

**AMS(MOS) Subject Classifications.**
65F25, 81R05, 65F05, 65F20, 81P68

**28** Obtaining bounds on the two norm of a matrix from the splitting lemma. *Doron Chen, John R. Gilbert, and Sivan Toledo*.

**Abstract.**
The splitting lemma is one of the two main tools of *support theory*, a framework for bounding the condition number of definite and semidefinite preconditioned linear systems. The splitting lemma allows the analysis of a complicated system to be partitioned into analyses of simpler systems. The other tool is the symmetric-product-support lemma, which provides an explicit spectral bound on a preconditioned matrix.

The symmetric-product-support lemma shows that under suitable conditions on the null spaces of $A$ and $B$, the finite eigenvalues of the pencil $(A, B)$ are bounded by $\|W\|_2^2$, where $U = VW$, $A = UU^T$, and $B = VV^T$. To apply the lemma, one has to construct a $W$ satisfying these conditions, and to bound its 2-norm.

In this paper we show that in all its existing applications, the splitting lemma can be viewed as a mechanism to bound $\|W\|_2^2$ for a given $W$. We also show that this bound is sometimes tighter than other easily-computed bounds on $\|W\|_2^2$, such as $\|W\|_F^2$ and $\|W\|_1 \|W\|_\infty$. The paper shows that certain regular splittings have useful algebraic and combinatorial interpretations. In particular, we derive six separate algebraic bounds on the 2-norm of a real matrix; to the best of our knowledge, these bounds are new.

**Key Words.**
matrix norm bounds, two-norm, norm bounds for sparse matrices, splitting lemma, support theory, support preconditioning

**AMS(MOS) Subject Classifications.**
15A60, 65F10, 65F35, 65F50

**47** Communication balancing in parallel sparse matrix–vector multiplication. *Rob H. Bisseling and Wouter Meesen*.

**Abstract.**
Given a partitioning of a sparse matrix for parallel matrix–vector multiplication, which determines the total communication volume, we try to find a suitable vector partitioning that balances the communication load among the processors. We present a new lower bound for the maximum communication cost per processor, an optimal algorithm that attains this bound for the special case where each matrix column is owned by at most two processors, and a new heuristic algorithm for the general case that often attains the lower bound. This heuristic algorithm tries to avoid raising the current lower bound when assigning vector components to processors. Experimental results show that the new algorithm often improves upon the heuristic algorithm that is currently implemented in the sparse matrix partitioning package Mondriaan. Trying both heuristics combined with a greedy improvement procedure solves the problem optimally in most practical cases. The vector partitioning problem is proven to be NP-complete.

**Key Words.**
vector partitioning, matrix–vector multiplication, parallel computing, sparse matrix, bulk synchronous parallel

**AMS(MOS) Subject Classifications.**
05C65, 65F10, 65F50, 65Y05

**66** Addressing the stochastic nature of scientific computations via dynamic loop scheduling. *Ioana Banicescu and Ricolindo L. Cariño*.

**Abstract.**
In general, scientific applications are large, computationally intensive, irregular and data-parallel. One of the main performance degradation factors when running scientific applications in parallel and distributed environments is load imbalance that may rise from a wide range of characteristics, due to problem, algorithmic, and systemic factors. Moreover, due to the stochastic behavior of many scientific applications, the load imbalance can unexpectedly occur at runtime, leading to difficulties in making decision about optimal partitioning, allocation and scheduling. In recent years, since parallel loops are prevalent and a major source of parallelism in scientific computations, many dynamic loop scheduling algorithms have been developed to improve performance via load balancing. These algorithms address the unpredictable behavior of computations, and many of these are based on probabilistic analyses. These have proven to be effective in improving performance of important applications used in scientific computing by addressing the entire spectrum of factors that lead to load imbalance. This paper proposes the use of a dynamic loop scheduling approach for repartitioning in scientific applications that contain computationally intensive parallel loops. A new partition is computed and data is redistributed during the execution of a parallel loop. This approach addresses load imbalance factors as they arise during a computational step, in contrast to the traditional load balancing approach of determining a new partition and migrating data after a computational step. Preliminary performance tests of an application that implements the proposed approach validate the feasibility and the effectiveness of the approach.

**Key Words.**
dynamic load balancing, data repartitioning, loop scheduling

**AMS(MOS) Subject Classifications.**
68W10

**81** Locality of reference in sparse Cholesky factorization methods. *Elad Rozin and Sivan Toledo*.

**Abstract.**
This paper analyzes the cache efficiency of two high-performance sparse Cholesky factorization algorithms: the multifrontal algorithm and the left-looking algorithm. These two are essentially the only two algorithms that are used in current codes; generalizations of these algorithms are used in general-symmetric and general-unsymmetric sparse triangular factorization codes. Our theoretical analysis shows that while both algorithms sometimes enjoy a high level of data reuse in the cache, they are incomparable: there are matrices on which one is cache efficient and the other is not, and vice versa. The theoretical analysis is backed up by detailed experimental evidence, which shows that our theoretical analyses do predict cache-miss rates and performance in practice, even though the theory uses a fairly simple cache model. We also show, experimentally, that on matrices arising from finite-element structural analysis, the left-looking algorithm consistently outperforms the multifrontal algorithm. Direct cache-miss measurements indicate that the difference in

performance is largely due to differences in the number of level-2 cache misses that the two algorithms generate. Finally, we also show that there are matrices where the multifrontal algorithm may require significantly more memory than the left-looking algorithm. On the other hand, the left-looking algorithm never uses more memory than the multifrontal one.

**Key Words.**
Cholesky factorization, sparse cholesky, multifrontal methods, cache-efficiency, locality of reference

**AMS(MOS) Subject Classifications.**
15A23, 65F05, 65F50, 65Y10, 65Y20

**107** Finding nonoverlapping substructures of a sparse matrix. *Ali Pinar and Virginia Vassilevska*.

**Abstract.**
Many applications of scientific computing rely on sparse matrix computations, thus efficient implementations of sparse matrix kernels are crucial for the overall efficiency of these applications. Due to the low compute-to-memory ratio and irregular memory access patterns, the performance of sparse matrix kernels is often far away from the peak performance on modern processors. Alternative matrix representations have been proposed, where the matrix $A$ is split into $A_d$ and $A_s$, so that $A_d$ contains all dense blocks of a specified form in the matrix, and $A_s$ contains the remaining entries. This facilitates using dense matrix kernels on the entries of $A_d$, producing better memory performance. We study the problem of finding a maximum number of nonoverlapping rectangular dense blocks in a sparse matrix. We show that the maximum nonoverlapping dense blocks problem is NP-complete by a reduction from the maximum independent set problem on cubic planar graphs. We also propose a $2/3$-approximation algorithm for $2 \times 2$ blocks that runs in linear time in the number of nonzeros in the matrix. We discuss alternatives to rectangular blocks such as diagonal blocks and cross blocks and present complexity analysis and approximation algorithms.

**Key Words.**
memory performance, memory-efficient data structures, high-performance computing, sparse matrices, independent sets, NP-completeness, approximation algorithms

**AMS(MOS) Subject Classifications.**
65F50, 65Y20, 05C50, 05C69, 68Q17,68W25

**125** The influence of random number generators on graph partitioning algorithms. *Ulrich Elsner*.

**Abstract.**
Many modern heuristic algorithms rely at least in some part on random numbers. Using graph partitioning algorithms as an example, we demonstrate the often underestimated influence of these random-number generators on the result of the heuristic algorithms. Even methods that rely on random numbers mostly as a tie-breaking tool can deliver very different results depending only on the starting value of the pseudo-random number generator.

**Key Words.**
graph partitioning, heuristic algorithms, pseudo random-number generator

**AMS(MOS) Subject Classifications.**
05C85, 90C59, 94C15, 65Y99, 68R10

**134** Optimality-preserving elimination of linearities in Jacobian accumulation. *Uwe Naumann and Jean Utke*.

**Abstract.**
We consider a mathematical function that is implemented in a high-level programming language such as C or Fortran. This function is assumed to be differentiable in some neighborhood of a set of input arguments. For available local partial derivatives of the arithmetic operators and intrinsic functions provided by the programming language, the Jacobian of the function at the given arguments can be accumulated by using the chain rule. This technique is known as automatic differentiation of numerical programs.

Under the above assumptions the values of the local partial derivatives are well defined for given values of the inputs. A code for accumulating the Jacobian matrix that is based on the chain rule takes these partial derivatives as input and computes the nonzero entries of the Jacobian using only scalar multiplications and additions. The exploitation of the associativity of the chain rule or, equivalently, the algebraic properties of the corresponding field $(\mathbb{R}, *, +)$ – in particular, associativity of the multiplication and distributivity – to minimize the number of multiplications leads to a combinatorial optimization problem that is widely conjectured to be NP-hard. Several heuristics have been developed for its approximate solution. Their efficiency always depends on the total number of partial derivatives.

Linearities in the function lead to constant partial derivatives that do not depend on the input values. We present a specialized constant folding algorithm to decrease the size of the combinatorial problem in order to increase the efficiency of heuristics for its solution. Moreover, we show that this algorithm preserves optimality in the sense that an optimal solution for the reduced problem yields an objective value no worse than that of an optimal solution for the original problem.

**Key Words.**
Jacobian accumulation, linearities, constant folding

**AMS(MOS) Subject Classifications.**
90C27, 26B10, 68N99