Mathematics in Industry 16
The European Consortium for Mathematics in Industry

Bastiaan Michielsen
Jean-René Poirier *Editors*

Scientific Computing in Electrical Engineering

SCEE 2010

Springer

MATHEMATICS IN INDUSTRY **16**

THE EUROPEAN CONSORTIUM
FOR MATHEMATICS IN INDUSTRY

For further volumes:
http://www.springer.com/series/4650

Bastiaan Michielsen
Jean-René Poirier
Editors

Scientific Computing in Electrical Engineering SCEE 2010

With 176 Figures, 115 in color and 36 Tables

Springer

Editors

Bastiaan Michielsen
ONERA
2 avenue Édouard Belin
31055 Toulouse
France
Bastiaan.Michielsen@onera.fr

Jean-René Poirier
LAPLACE-ENSEEIHT
2 rue Charles Camichel
31071 Toulouse
France
poirier@laplace.univ-tlse.fr

ISBN 978-3-642-22452-2 e-ISBN 978-3-642-22453-9
DOI 10.1007/978-3-642-22453-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011942998

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book presents an account of the Scientific Computing in Electrical Engineering conference, SCEE 2010, which took place in Toulouse, September 2010. The SCEE series of conferences covers many aspects of mathematics applied to electrical engineering, including electronics, electrical networks and electromagnetics. It started as a national meeting in 1997 in Germany and the first two meetings were organised under the auspices of the Deutscher Mathematiker Vereinigung. The title SCEE appeared for the first time in 2000 and since then the conference has been held every other year and in different European countries.

The organisation of the 8th conference was provided by the Toulouse branch of Onera, the French aerospace laboratory, and the ENSEEIHT, situated in the heart of Toulouse, was kind enough to make one of its lecture halls available. This 8th edition of the SCEE conference was further sponsored by

ABB,	Switzerland	http://www.abb.com
AWR,	Finland	http://web.awr.com
MunEDA,	Germany	http://www.muneda.com
CST,	Germany	http://www.cst.com

Their financial and material support is gratefully acknowledged.

The scientific programme of the conference was organised by the programme committee, which consisted of:

Dr. Andreas Blaszczyk (ABB Corporate Research, Switzerland)
Prof. Gabriela Ciuprina (Polytehnica University of Bucharest, Romania)
Dr. Georg Denk (Infineon, Germany)
Prof. Michael Günther (University of Wuppertal, Germany)
Dr. Jan ter Maten (NXP Semiconductors, The Netherlands)
Ir. Bastiaan Michielsen (Onera, France)
Prof. Ursula van Rienen (University of Rostock, Germany)
Prof. Vittorio Romano (University of Catania, Italy)
Dr. Janne Roos (Helsinki University of Technology, Finland)
Prof. Wil Schilders (TU/e & NXP Semiconductors, The Netherlands)
Prof. Thomas Weiland (TU Darmstadt & CST, Germany)

The programme committee attended to the reviewing of the proposed contributions. This resulted in 41 plenary talks and 34 poster presentations. The programme committee was also responsible for inviting specialist speakers to introduce the sessions. We were happy to have the following invited talks (in order of their appearance at the conference):

- Guillaume Sylvand (EADS IW, France),
  "From quasi-static to high frequencies: An overview of numerical simulation at EADS"
- Tim Davis (University of Florida, USA),
  "Sparse matrix methods for circuit simulation problems"
- Heidi Thornquist (Sandia National Laboratories, USA),
  "Advances in Parallel Transistor-Level Circuit Simulation"
- Maurizio Repetto (Politecnico Torino, Italy),
  "Tonti diagrams and algebraic methods for the solution of coupled problems"
- Patrick Dular (Université de Liège, Belgium),
  "Magnetic model refinement via coupling of finite element subproblems"
- Naoufel Ben Abdallah (Université Paul Sabatier), who was our invited speaker for the device modelling session, passed away in the summer of 2010, just two months before the conference. His time slot in the conference programme was left unfilled
- Jörg Ostrowski (ABB, Switzerland),
  "Transient Full Maxwell Computation of Slow Processes"
- Helmut Gräb (TU München, Germany),
  "From Sizing over Design Centering and Pareto Optimization to Tolerance Pareto Optimization of Electronic Circuits"
- Joost Rommes (NXP, The Netherlands),
  "Challenges in model order reduction for industrial problems"

All authors of papers accepted for presentation at the conference were also invited, in a second round, to propose a contribution for publication in this post-conference book. The programme committee organised the final reviewing and you will find the 47 selected papers in the main part of this book.

As editors of this book and members of the local organising committee, we would like to thank all the authors who contributed to the conference and, later, to this book for their good work. We thank the reviewers for having gone twice through all the proposed contributions and for all their constructive remarks. We also thank Onera and the ENSEEIHT for having made this work possible and the staff of Springer Verlag for their patience in getting this book to the press.

Toulouse, France

Bastiaan Michielsen, Onera
Jean-René Poirier, ENSEEIHT

Contents

Part I Mathematical Methods

Sparse Matrix Methods for Circuit Simulation Problems 3
Timothy A. Davis and E. Palamadai Natarajan

Some Remarks on A Priori Error Estimation for ESVDMOR 15
Peter Benner and André Schneider

Block Preconditioning Strategies for High Order Finite Element Discretization of the Time-Harmonic Maxwell Equations 25
Matthias Bollhöfer and Stéphane Lanteri

From Sizing over Design Centering and Pareto Optimization to Tolerance Pareto Optimization of Electronic Circuits 35
Helmut Gräb

Importance Sampling for Determining SRAM Yield and Optimization with Statistical Constraint 39
E.J.W. ter Maten, O. Wittich, A. Di Bucchianico, T.S. Doorn, and T.G.J. Beelen

Effective Numerical Computation of Parameter Dependent Problems 49
Lennart Jansen and Caren Tischendorf

Analytical Properties of Circuits with Memristors 59
Ricardo Riaza

Scattering Problems in Periodic Media with Local Perturbations 69
Therese Pollok, Lin Zschiedrich, and Frank Schmidt

Part II Computational Electromagnetics

From Quasi-static to High Frequencies: An Overview of Numerical Simulation at EADS 81
Guillaume Sylvand

Transient Full Maxwell Computation of Slow Processes 87
J. Ostrowski, R. Hiptmair, F. Krämer, J. Smajic, and T. Steinmetz

A Frequency-Robust Solver for the Time-Harmonic Eddy Current Problem 97
Michael Kolmbauer and Ulrich Langer

Depolarization of Electromagnetic Waves from Bare Soil Surfaces 107
Naheed Sajjad, Ali Khenchaf, and Arnaud Coatanhay

Two Finite-Element Thin-Sheet Approaches in the Electro-Quasistatic Formulation 117
Jens Trommler, Stephan Koch, and Thomas Weiland

Mode Selecting Eigensolvers for 3D Computational Models 127
Bastian Bandlow and Rolf Schuhmann

Magnetic Model Refinement via a Coupling of Finite Element Subproblems 137
Patrick Dular, Ruth V. Sabariego, Laurent Krähenbühl, and Christophe Geuzaine

Substrate Modeling Based on Hierarchical Sparse Circuits 143
Daniel Ioan, Gabriela Ciuprina, and Ioan-Alexandru Lazăr

A Boundary Conformal DG Approach for Electro-Quasistatics Problems 153
A. Fröhlcke, E. Gjonaj, and T. Weiland

Optimization of the Current Density Distribution in Electrochemical Reactors 163
Florin Muntean, Alexandru Avram, Johan Deconinck, Marius Purcar, Vasile Topa, Calin Munteanu, Laura Grindei, and Ovidiu Garvasuc

Streamer Line Modeling 173
Thomas Christen, Helmut Böhme, Atle Pedersen, and Andreas Blaszczyk

A Discontinuous Galerkin Formalism to Solve the Maxwell-Vlasov Equations. Application to High Power Microwave Sources 183
Laura Pebernet, Xavier Ferrieres, Vincent Mouysset, François Rogier, and Pierre Degond

Part III Coupled Problems

Tonti Diagrams and Algebraic Methods for the Solution of Coupled Problems 195
Fabio Freschi and Maurizio Repetto

Soliton Collision in Biomembranes and Nerves- A Stability Study 205
Revathi Appali, Benny Lautrup, Thomas Heimburg, and Ursula van Rienen

Nonlinear Characterization and Simulation of Zinc-Oxide Surge Arresters 213
Frank Denz, Erion Gjonaj, and Thomas Weiland

Behavioural Electro-Thermal Modelling of SiC Merged PiN Schottky Diodes 223
M. Zubert, M. Janicki, M. Napieralska, G. Jablonski, L. Starzak, and A. Napieralski

A Convergent Iteration Scheme for Semiconductor/Circuit Coupled Problems 233
Giuseppe Alì, Andreas Bartel, Markus Brunk, and Sebastian Schöps

Multirate Time Integration of Field/Circuit Coupled Problems by Schur Complements 243
Sebastian Schöps, Andreas Bartel, and Herbert De Gersem

Part IV Circuit and Device Modelling and Simulation

Advances in Parallel Transistor-Level Circuit Simulation 257
Heidi K. Thornquist and Eric R. Keiter

Sensitivity-Based Steady-State Mismatch Analysis for RF Circuits 267
Fabrice Veersé, Joël Besnard, and Hubert Filiol

Modelling and Simulation of Forced Oscillators with Random Periods 275
Roland Pulch

Initialization of HB Oscillator Analysis from Transient Data 285
Mikko Hulkkonen, Mikko Honkala, Jarmo Virtanen, and Martti Valtonen

Robust Periodic Steady State Analysis of Autonomous Oscillators Based on Generalized Eigenvalues 293
R. Mirzavand Boroujeni, E.J.W. ter Maten, T.G.J Beelen, W.H.A. Schilders, and A. Abdipour

Mutual Injection Locking of Oscillators under Parasitic Couplings 303
M.M. Gourary, S.G. Rusakov, S.L. Ulyanov, and M.M. Zharov

Time Domain Simulation of Power Systems with Different Time Scales 313
Valeriu Savcenco, Bertrand Haut, E. Jan W. ter Maten, and Robert M.M. Mattheij

Adaptive Wavelet-Based Method for Simulation of Electronic Circuits 321
Kai Bittner and Emira Dautbegovic

Modeling and Simulation of Organic Solar Cells 329
Carlo de Falco, Antonio Iacchetti, Maddalena Binda, Dario Natali, Riccardo Sacco, and Maurizio Verri

Numerical Simulation of a Hydrodynamic Subband Model for Semiconductors Based on the Maximum Entropy Principle 339
G. Mascali and V. Romano

Inverse Doping Profile of MOSFETs via Geometric Programming 347
Yiming Li and Ying-Chieh Chen

Numerical Simulation of Semiconductor Devices by the MEP Energy-Transport Model with Crystal Heating 357
Vittorio Romano and Alexander Rusakov

Part V Model Order Reduction

Challenges in Model Order Reduction for Industrial Problems 367
Joost Rommes

On Approximate Reduction of Multi-Port Resistor Networks 377
M.V. Ugryumova, J. Rommes, and W.H.A. Schilders

Improving Model-Order Reduction Methods by Singularity Exclusion 387
Pekka Miettinen, Mikko Honkala, Janne Roos, and Martti Valtonen

Partitioning-Based Reduction of Circuits with Mutual Inductances 395
Pekka Miettinen, Mikko Honkala, Janne Roos, and Martti Valtonen

Model Order Reduction of Parameterized Nonlinear Systems by Interpolating Input-Output Behavior 405
Michael Striebel and Joost Rommes

On the Selection of Interpolation Points for Rational Krylov Methods 415
E. Fatih Yetkin and Hasan Dağ

Discrete Empirical Interpolation in POD Model Order Reduction of Drift-Diffusion Equations in Electrical Networks 423
Michael Hinze and Martin Kunkel

Model Order Reduction for Complex High-Tech Systems 433
Agnieszka Lutowska, Michiel E. Hochstenbach, and Wil H.A. Schilders

Parametric Model Order Reduction by Neighbouring Subspaces..... 443
Kynthia Stavrakakis, Tilmann Wittig, Wolfgang Ackermann, and Thomas Weiland

Author Index..... 453

Index..... 455

Introduction

Electromagnetic interactions are the only fundamental interactions which can be easily manipulated on the macroscopic scale. As such, electromagnetic interactions play a major role in modern life. Technological advances in information processing machines, from mobile telephones to car information systems and personal computers to cash machines, are perhaps the ones that most immediately come to mind. The advances of electromagnetic technology show no sign of stopping and will continue to influence not only our daily life but also the way we do research.

The role of applied mathematics in this process is important in two complementary ways. Firstly, mathematical analysis of the fundamental models from theoretical physics, as well as the engineering models derived from them, is essential for the proper understanding of the nature of the phenomena themselves. Understanding physical phenomena means knowing some abstract, concise, organising model for them. Secondly, the mathematics of numerical computations with the said models is essential for the reliability of the conclusions we draw concerning these phenomena. In a certain way, we always seek to master the physical phenomena we encounter, first by trying to predict them and then by trying to influence the course of events.

Following this line of reasoning, we can be a bit more precise on the role of mathematics in industry. Existence and uniqueness results are important to exactly identify what can be considered as a "consistent model" and what not. Convergence results are essential for being able to decide whether an obtained conclusion is reliable or not. In the end, the purpose is to replace time-consuming and costly real experiments with more time-efficient and cheaper simulated experiments. This ideal situation has not yet been reached in every domain but scientific computation is already an indispensable part of industrial design cycles.

As for the vast domain of electromagnetics-related technological development, one can distinguish different application domains of mathematics. The most obvious one is the computation of the electromagnetic field itself. The possibility to construct numerical representations of electromagnetic fields corresponding to given space-time distributions of electric charge has grown from the construction of elementary "analytical" solutions of canonical problems in the nineteenth century into an abundance of discretisation-based algorithms, where, roughly speaking, the concept

of a point-wise converging approximation using global expansions (the analytic functions) has been replaced by globally converging approximations using local expansions. This has created a completely new and extremely vast domain of research where there seems to be no limit to the geometrical complexity for which solutions can be obtained using modern computers.

Although one may question the applicability of the system of Maxwell equations as a correct model for physics on the atomic and subatomic scale, there is no doubt that it is successful as a generic model for macroscopic electromagnetic phenomena. This however does not mean that all modelling necessarily involves solving the Maxwell equations. From a certain point of view there is a hierarchy of derived models. The most often used model derivation method is asymptotic in nature. The fundamental Maxwellian model is a space-time model, but, more often than not, in engineering applications one is interested in single frequency states. These have only an asymptotical meaning but, in fact, the asymptote is rather easily attained and it pays off to work with the so-called frequency domain Maxwell equations, which determine the complex valued vectorial amplitudes (phasors) of the time harmonic fields.

Another extremely useful asymptotic is the "quasi-static" one, which describes the field behaviour in configurations where propagation delay is negligible. This asymptotic provides the link between macroscopic electromagnetic field theory and the models for electronic circuits. Starting from the microscopic point of view, where one should replace the Maxwell equations by quantum physics models of matter and interaction, one can also climb up to the macroscopic level by using different sorts of asymptotics. The constitutive coefficients one uses in the macroscopic Maxwell equations can be obtained through statistical quantum physics, though it is also possible to make do with a phenomenological model in accordance with irreversible thermodynamics.

It should be said that all of these different models, fundamental or derived, have their own specificities. Even though certain models are derived from for example the time-domain Maxwell equations, they require a separate mathematical study. A satisfying unique-existence analysis of a time domain does not necessarily clear up the situation of the corresponding problem in the time harmonic case, a reduced two-dimensional version of the problem or a quasi-static problem. All of these problem classes present their own set of difficulties and solution methods. The same thing can be said of problems of (statistical) quantum physics and their derived models. This implies that the different communities, each focussing on one specific type of model, show a tendency to isolate themselves from the others. This is unfortunate, because it neglects the fact that we need all the models together to better understand the way things work and in order to make more reliable predictions about how new products will behave, once the designs are realised.

The SCEE series of conferences was created to counteract this isolation. The idea was and is to organise the communication between researchers working on all the above-mentioned aspects of electromagnetic and electronic phenomena, from the detailed functioning of electronic devices, where the dynamics of electrically charged particles in solid state or plasmas are studied, to network theory and

circuit simulation, to macroscopic electromagnetic field, for which the modelling of machines requires considering the coupling of mechanical and thermophysical phenomena with electromagnetic ones, as well as the radiation, propagation and reception of electromagnetic waves.

Outline of the Book

In this book the readers will find papers on many of the aspects discussed above. The book is divided into five parts. The first part and the last part are both more or less generic. This first part is called *Mathematical Methods* and presents contributions which, although linked to some application domain, are closer to general applied mathematics than those in the other parts. The last part is called *Model Order Reduction*. Techniques of model order reduction (MOR) can often find use in several application domains and can be based on an application-independent analysis of a system of equations. However, some MOR methods depend on a very special property only appearing in one given class of applications.

The remaining three parts present contributions from applied mathematics, which are more closely related to their respective application domains. The second part, *Computational Electromagnetics*, examines computational methods in macroscopic electromagnetic field theory. The contributions in the third part, *Coupled Problems*, are concerned with multi-physics modelling. The part *Circuit and Device Modelling and Simulation*, deals with mathematics applied to circuit simulation, i.e. electromagnetic interactions on the scale of electronic systems, as well as with modelling on the scale of the interior of the electronic devices themselves.

Each part has its own introduction, which serves to situate the various contributions in the overall context sketched above.

Part I
Mathematical Methods

Introduction

This part is concerned with general mathematical methods of interest for numerical modelling in electrical engineering. As, in the end, any numerical modelling is based on finite dimensional models, numerical linear algebra is a common subject of interest to the whole community. As such, the first four papers show some aspects of this vast domain of research.

The first contribution in this part was written by T. Davis (an invited speaker at the conference) and E. Palamadai Natarajan. It is concerned with sparse matrices, especially those arising with the differential algebraic equations (DAE) used in circuit simulation problems. Sparse methods based on operations on dense sub-matrices, such as multi-frontal methods, are not effective in these cases. A software package named KLU, which was specifically written to exploit the properties of sparse circuit matrices, is presented as are results comparing it with other packages for circuit simulation.

In the next contribution, P. Benner and A. Schneider discuss a priori error estimation for singular value decomposition-based model order reduction methods. Proven error estimates are a necessary first step before a fully automatic application of such approaches can be relied on. This work presents steps towards a global a priori error estimation for this class of algorithms.

The contribution by M. Bollhöfer and S. Lanteri discusses block pre-conditioning for the solution of large linear systems resulting from the discretisation of the time-harmonic Maxwell equations. The proposed strategies combine principles from incomplete factorisation methods with complex shift of the diagonal entries of the underlying system matrices. Numerical results are presented for electromagnetic wave propagation problems in homogeneous and heterogeneous media.

The following three contributions present mathematical methods related to circuit design and analysis. The contribution by Gräb (an invited speaker at the conference) provides an overview of multi-objective sizing tasks in electronic circuit design. It is shown how statistically distributed and range-valued parameters can

be included in yield optimisation and design centering. In addition, accounting for parameter tolerances by multi-objective Pareto optimisation is presented.

The next contribution by E.J.W. ter Maten et al. studies importance sampling as a means of achieving efficient Monte Carlo sampling that also properly covers tails of distributions. An optimal upper bound is derived for the number of samples needed to efficiently obtain an accurate fail probability. The contribution by L. Jansen and C. Tischendorf is concerned with the analysis of parameter-dependent differential algebraic equations. The authors show how to take benefit from the smoothness of the solution as a function of the parameters, in order to efficiently find the solutions for a range of parameter values.

The last two contributions in this part present a mathematical analysis of two specific modelling problems, one from circuit theory and one from electromagnetic field theory. R. Riaza's contribution is concerned with a new lumped circuit element, called a memristor, which is characterised by a nonlinear charge-flux relation. Some analytical properties of semi-state models of the corresponding memristive circuits are studied in terms of differential algebraic equations. More specifically, the geometric index of the DAEs arising in so-called branch-oriented analysis methods is considered. In the last paper of this part, T. Pollok et al. discuss scattering problems for the Helmholtz equation in periodic configurations. The authors develop a general mathematical analysis, valid in any dimension, and algorithms for the handling of periodic structures with local defects.

Sparse Matrix Methods for Circuit Simulation Problems

Timothy A. Davis and E. Palamadai Natarajan

Abstract Differential algebraic equations used for circuit simulation give rise to sequences of sparse linear systems. The matrices have very peculiar characteristics as compared to sparse matrices arising in other scientific applications. The matrices are extremely sparse and remain so when factorized. They are permutable to block triangular form, which breaks the sparse LU factorization problem into many smaller subproblems. Sparse methods based on operations on dense submatrices (supernodal and multifrontal methods) are not effective because of the extreme sparsity. KLU is a software package specifically written to exploit the properties of sparse circuit matrices. It relies on a permutation to block triangular form (BTF), several methods for finding a fill-reducing ordering (variants of approximate minimum degree and nested dissection), and Gilbert/Peierls' sparse left-looking LU factorization algorithm to factorize each block. The package is written in C and includes a MATLAB interface. Performance results comparing KLU with SuperLU, Sparse 1.3, and UMFPACK on circuit simulation matrices are presented. KLU is the default sparse direct solver in the Xyce^{TM}circuit simulation package developed by Sandia National Laboratories.

1 Overview

The KLU software package is specifically designed for solving sequences of unsymmetric sparse linear systems that arise from the differential-algebraic equations used to simulate electronic circuits. Two aspects of KLU are essential for these

T.A. Davis (✉)
Department of Computer and Information Science and Engineering. University of Florida, FL, USA
e-mail: davis@cise.ufl.edu

E. Palamadai Natarajan
Ansys. Inc., USA
e-mail: ekanathan@gmail.com

B. Michielsen and J.-R. Poirier (eds.), *Scientific Computing in Electrical Engineering SCEE 2010*, Mathematics in Industry 16, DOI 10.1007/978-3-642-22453-9_1,

problems: (1) a permutation to block upper triangular form [15, 17], and (2) an asymptotically efficient left looking LU factorization algorithm with partial pivoting [18]. KLU does not exploit supernodes, since the factors of circuit simulation matrices are far too sparse as compared to matrices arising in other applications (such as finite-element methods).

Circuit simulation involves many different tasks for which KLU is useful:

1. DC operating point analysis, where BTF ordering is often helpful. Convergence in DC analysis is critical in that it is typically the first step of a higher level analysis such as transient analysis.
2. Transient analysis, which requires a fast and accurate sparse LU factorization. The sparse linear factorization/solve stages typically dominate the run-time of transient analyses of post-layout circuits with a large number of parasitic devices.
3. Harmonic balance analysis, which is typically solved using Krylov based iterative methods, since the Jacobian representing all the harmonics is huge and cannot be solved with a direct method. KLU is useful in factor/solve stages involving the pre-conditioner.

Section 2 describes the characteristics of circuit matrices, which motivate the design of the KLU algorithm. Section 3 gives a brief description of the algorithm. A more detailed discussion may be found in [24]. Performance results of KLU in comparison with SuperLU [12], Sparse 1.3 [21, 22], and UMFPACK [4, 6, 7] are presented in Sect. 4. An extended version of this paper appears in [11].

In this paper, $|A|$ denotes the number of nonzeros in the matrix A.

2 Characteristics of Circuit Matrices

Circuit matrices arise from Newton's method applied to the differential-algebraic equations representing the underlying circuit [23]. A modified nodal analysis is typically used, resulting in a sequence of linear systems with unsymmetric sparse coefficient matrices with identical nonzero pattern (ignoring numerical cancellation). Circuit matrices exhibit certain unique characteristics for which KLU is designed, which are not generally true of matrices from other applications:

1. Circuit matrices are extremely sparse and remain so when factorized. The ratio of floating-point operation (flop) count over $|L + U|$ is much smaller than matrices from other applications (even for comparable values of $|L + U|$). A set of columns in L with identical or similar nonzero pattern is called a *supernode* [12]. Supernodal and multifrontal methods obtain high performance by exploiting supernodes via dense matrix kernels (the BLAS, [13]). Because their nodal interconnection is highly dissimilar and their fill-in is so low, the supernodes in circuit matrices typically have very few columns. Dense matrix kernels are not effective when used on very small matrices, and thus supernodal/multifrontal methods are not suitable for circuit matrices.

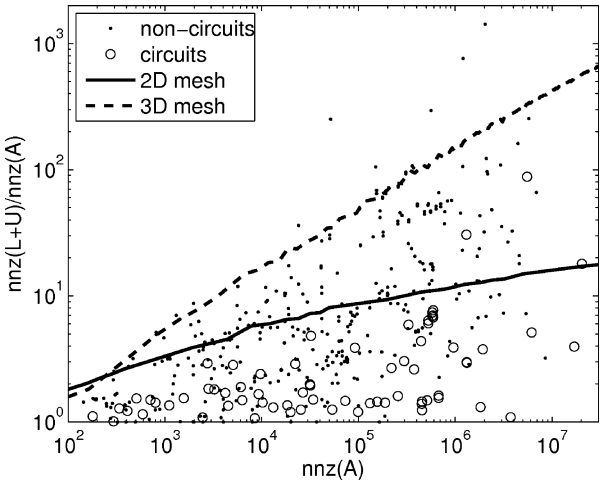

Fig. 1 Fill-in factor versus the number of nonzeros in the largest irreducible block

2. Nearly all circuit matrices are permutable to a block triangular form. In DC operating point analysis, capacitors are open and hence node connectivity is broken in the circuit. This helps in creating many small strongly connected components in the corresponding graph, and the resulting permuted matrix is block triangular with many small blocks. However in transient simulation, capacitors are not open and hence the nodes of the circuit are mostly reachable from each other. This often leads to one large diagonal block when permuted to BTF form, but still a large number of small blocks due to the presence of independent and controlled sources.

The following experiment illustrates the low fill-in properties of circuit matrices. As of March 2010, the University of Florida Sparse Matrix Collection [10] contains 491 matrices that are real, square, unsymmetric, and have full structural rank[1] (excluding matrices tagged as subsequent matrices in sequences of matrices with the same size and pattern). Of these 491 matrices, 81 are from circuit or power network simulation. Figure 1 plots the fill-in factor ($|L + U|/|A|$ versus $|A|$) for each matrix, using `lu` in MATLAB (R2010a). If the matrix is reducible to block triangular form, only the largest block is factorized for this experiment (found via `dmperm` [5]). For comparison, the two lines in Fig. 1 are 2D and 3D square meshes as ordered by METIS [20], which obtains the asymptotically optimal ordering for regular meshes.

The fill-in factor for circuit matrices stays remarkably low as compared to matrices from other applications. Very few circuit matrices experience as much fill-in as 2D or 3D meshes.

[1] A matrix has full structural rank if a permutation exists so that the diagonal is zero-free.

The properties of circuit matrices demonstrated here indicate that they should be factorized via an asymptotically efficient non-supernodal sparse LU method, which motivates the KLU algorithm discussed in the next Section.

3 KLU Algorithm

KLU performs the following steps when solving the first linear system in a sequence.

1. The matrix is permuted into block triangular form (BTF). This consists of two steps: an unsymmetric permutation to ensure a zero free diagonal using maximum transversal [14, 15], followed by a symmetric permutation to block triangular form by finding the strongly connected components of the graph [16, 17, 26]. A matrix with full rank permuted to block triangular form looks as follows:
$$PAQ = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ & A_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & A_{nn} \end{bmatrix}$$
2. Each block A_{kk} is ordered to reduce fill. The Approximate Minimum Degree (AMD) ordering [1,2] on $A_{kk} + A_{kk}^T$ is used by default. The user can alternatively choose COLAMD [8, 9], an ordering provided by CHOLMOD (such as nested dissection based on METIS [20]), or any user-defined ordering algorithm that can be passed as a function pointer to KLU. Alternatively, the user can provide a permutation to order each block.
3. Each diagonal block is scaled and factorized using our implementation of Gilbert/ Peierls' left looking algorithm with partial pivoting [18]. A simpler version of the same algorithm is used in the LU factorization method in the CSparse package, `cs_lu` [5] (but without the pre-scaling and without a BTF permutation). Pivoting is constrained to within each diagonal block, since the factorization method factors each block as an independent problem. No pivots can ever be selected from the off-diagonal blocks.
4. The system is solved using block back substitution.

For subsequent factorizations for matrices with the same nonzero pattern, the first two steps above are skipped. The third step is replaced with a simpler left-looking method that does not perform partial pivoting (a *refactorization*). This allows the depth-first-search used in Gilbert/Peierls' method to be skipped, since the nonzero patterns of L and U are already known.

When the BTF form is exploited, entries outside the diagonal blocks do not need to be factorized, requiring no work and causing no fill-in. Only the diagonal blocks need to be factorized.

The final system of equations to be solved after ordering and factorization with partial pivoting can be represented as

$$(PRAQ)Q^T x = PRb \tag{1}$$

where P represents the row permutation due to the BTF and fill-reducing ordering and partial pivoting, and Q represents the column permutation due to just the BTF and fill-reducing ordering. The matrix R is a diagonal row scaling matrix (discussed below). Let $(PRAQ) = LU + F$ where LU represents the factors of all the blocks collectively and F represents the entire off diagonal region. Equation (1) can now be written as

$$x = Q(LU + F)^{-1}(PRb). \tag{2}$$

The block back substitution in (2) can be better visualized as follows. Consider a simple 3-by-3 block system

$$\begin{bmatrix} L_{11}U_{11} & F_{12} & F_{13} \\ 0 & L_{22}U_{22} & F_{23} \\ 0 & 0 & L_{33}U_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}. \tag{3}$$

The equations corresponding to the above system are

$$L_{11}U_{11}x_1 + F_{12}x_2 + F_{13}x_3 = b_1 \tag{4}$$

$$L_{22}U_{22}x_2 + F_{23}x_3 = b_2 \tag{5}$$

$$L_{33}U_{33}x_3 = b_3 \tag{6}$$

In block back substitution, we first solve (6) for x_3, and then eliminate x_3 from (5) and (4) using the off-diagonal entries. Next, we solve (5) for x_2 and eliminate x_2 from (4). Finally we solve (4) for x_1.

The core of the Gilbert/Peierls factorization algorithm used in KLU is solving a lower triangular system $Lx = b$ with partial pivoting where L, x and b are all sparse. It consists of a symbolic step to determine the non-zero pattern of x and a numerical step to compute the values of x. This lower triangular solution is repeated n times during the entire factorization (where n is the size of the matrix) and each solution step computes a column of the L and U factors. The importance of this factorization algorithm is that the time spent in factorization is proportional to the number of floating point operations performed. The entire left looking algorithm is described in the algorithm below.

The lower triangular solve is the most expensive step and includes a symbolic and a numeric factorization step. Let $b = A(:, k)$, the kth column of A. Let G_L be the directed graph of L with n nodes. The graph G_L has an edge $j \rightarrow i$ iff $l_{ij} \neq 0$. Let $\mathscr{B} = \{i \,|\, b_i \neq 0\}$ and $\mathscr{X} = \{i \,|\, x_i \neq 0\}$ represent the set of nonzero indices in b and x respectively. Now the nonzero pattern $\mathscr{X}$ is given by

Algorithm 1 Left-looking LU factorization

```
L = I
for k = 1 to n do
    solve Lx = A(:,k) for x
    do partial pivoting on x
    U(1:k,k) = x(1:k)
    L(k:n,k) = x(k:n)/U(k,k)
end for
```

$$\mathscr{X} = Reach_{G_L}(\mathscr{B}) \tag{7}$$

$Reach_G(i)$ denotes all nodes in a graph G reachable via paths starting at node i. $Reach(S)$ applied to a set S is the union of $Reach(i)$ for all nodes $i \in S$. Equation (7) states that the nonzero pattern $\mathscr{X}$ is computed by the determining the vertices in G_L that are reachable from the vertices of the set $\mathscr{B}$.

The reachability problem is solved using a depth-first search. During the depth-first search, the Gilbert/ Peierls algorithm computes the *topological* order of $\mathscr{X}$. If the nodes of a directed acyclic graph are written out in topological order from left to right, then all edges in the graph would point to the right. If $Lx = b$ is solved in topological order, all numerical dependencies are satisfied. The natural order 1, 2, ..., n is one such ordering (since the matrix L is lower triangular), but any topological ordering will suffice. That is, x_j must be computed before x_i if there is a path from j to i in G_L. Since the depth-first graph traversal produces $\mathscr{X}$ in topological order as an intrinsic by-product, the solution of $Lx = b$ can be computed using the algorithm below. Sorting the nodes in $\mathscr{X}$ to obtain the natural ordering could take more time than the number of floating-point operations, so this is skipped. The computation of $\mathscr{X}$ and x both take time proportional to the floating-point operation count.

Algorithm 2 Solve $Lx = b$ where L, x and b are sparse

```
X = Reach_{G_L}(B)
x = b
for j ∈ X in any topological order do
    x(j+1:n) = x(j+1:n) − L(j+1:n, j)x(j)
end for
```

4 Performance Comparisons with Other Solvers

Five different sparse LU factorization techniques are compared:

1. KLU with default parameter settings: BTF enabled, the AMD fill-reducing ordering applied to $A + A^T$, and a strong preference for pivots selected from the diagonal.

Table 1 The thirteen test matrices with the highest run times

Matrix	Entire matrix		Largest block		Rows in	Singletons
	Rows $\times 10^3$	Nonzeros $\times 10^3$	Rows $\times 10^3$	Nonzeros $\times 10^3$	2nd largest block	$\times 10^3$
Raj1	263.7	1,300.3	263.6	1,299.6	5	0.2
ASIC_680k	682.9	2,639.0	98.8	526.3	2	583.8
rajat24	358.2	1,947.0	354.3	1,923.9	172	3.4
TSOPF_RS_b2383_c1	38.1	16,171.2	4.8	31.8	654	0.0
TSOPF_RS_b2383	38.1	16,171.2	4.8	31.8	654	0.0
rajat25	87.2	606.5	83.5	589.8	57	3.4
rajat28	87.2	606.5	83.5	589.8	57	3.4
rajat20	86.9	604.3	83.0	587.5	57	3.6
ASIC_320k	321.8	1,931.8	320.9	1,314.3	6	0.3
ASIC_320ks	321.7	1,316.1	320.9	1,314.3	6	0.1
rajat30	644.0	6,175.2	632.2	6,148.3	7	11.7
Freescale1	3,428.8	17,052.6	3,408.8	16,976.1	19	0.0

2. KLU with default parameters, except that BTF is disabled. For most matrices, using BTF is preferred, but in a few cases the BTF pre-ordering can dramatically increase the fill-in in the LU factors.
3. SuperLU 3.1 [12], using non-default diagonal pivoting preference and ordering options identical to KLU (but without BTF).[2] These options typically give the best results for circuit matrices. SuperLU is a supernodal variant of the Gilbert/Peierls' left-looking algorithm used in KLU.
4. UMFPACK [4, 6, 7] with default parameters. In this mode, UMFPACK evaluates the symmetry of the nonzero pattern and selects either the AMD ordering on $A + A^T$ and a strong diagonal preference, or it uses the COLAMD ordering with no preference for the diagonal. For most circuit simulation matrices, the AMD ordering is used. UMFPACK is a right-looking multifrontal algorithm that makes extensive use of BLAS kernels.
5. Sparse 1.3 [21, 22], the sparse solver used in SPICE3f5, the latest version of SPICE.[3]

The University of Florida Sparse Matrix Collection [10] includes 81 real square unsymmetric matrices or matrix sequences (only the first matrix in each sequence is considered here) arising from the differential algebraic equations used in SPICE-like circuit simulation problems, or from power network simulation. All five methods were tested an all 81 matrices, except for two matrices too large for any method on the computer used for these tests (a single-core 3.2 GHz Pentium 4 with 4 GB of RAM). The thirteen matrices requiring the most amount of time to analyze, factorize, and solve (as determined by the fastest method for each matrix) are shown in Table 1. All of the matrices come from a transient analysis, since the run time

[2] Threshold partial pivoting tolerance of 0.001 to give preference to the diagonal, the SuperLU symmetric mode, and the AMD ordering on $A + A^T$.

[3] http://bwrc.eecs.berkeley.edu/Classes/icbook/SPICE/

for KLU is very low for matrices arising from a DC analysis. The table lists the matrix name followed by the size of the whole matrix and the largest block in the BTF form (the dimension and the number of nonzeros). The last two columns list the dimension of the second-largest block, and the number of 1-by-1 blocks, respectively.

A performance profile compares the relative run times of multiple methods on a set of test problems. Let the relative run time of a method on a particular problem be equal to its run time for that problem divided by the fastest run time of any method for that problem. A relative run time of 1.0 means that the method is the fastest for that problem among the methods being compared; 2.0 means that it took twice the time as the fastest method. The x axis of a performance profile is this relative run time. The y axis of a performance profile is the number of problems. A point (x, y) is plotted if a method has a relative run time of x (or less) for y problems in the test set.

The performance profiles of the four methods are shown in Fig. 2. It excludes the symbolic ordering and analysis, since this step is done just once for a whole sequence of matrices. Note that the x axis of Fig. 2 is a log scale. For most matrices, KLU (with BTF) is the fastest method. In the worst case (the Raj1 matrix) it is 26 times slower than SuperLU, but this is because the permutation to BTF used by KLU causes fill-in to dramatically increase.

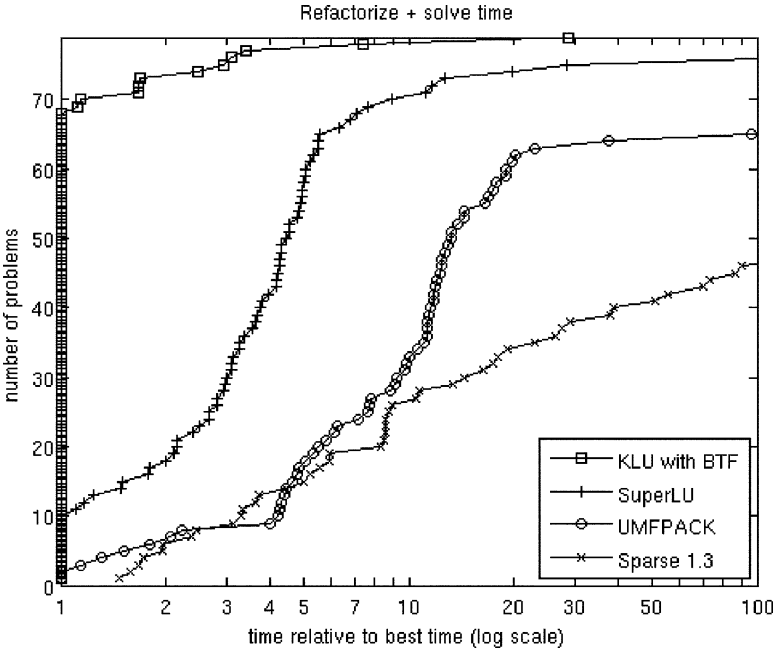

Fig. 2 Performance profile of refactorize+solve time

Table 2 Analyze+factorize+solve time in seconds, and relative fill-in ($|L+U|/|A|$) for KLU. Run times within 25% of the fastest are shown in bold. A dash is shown if the method ran out of memory

Matrix	KLU+BTF		KLU no BTF		SuperLU	Sparse 1.3
	Fill	Time	Fill	Time	Time	Time
Raj1	40.3	111.0	5.5	**4.6**	**4.2**	3,038.9
ASIC_680ks	2.6	**5.0**	2.7	7.2	**4.6**	818.1
ASIC_680k	2.1	**5.8**	2.1	7.4	**5.8**	8,835.1
rajat24	28.7	119.0	3.3	**6.0**	13.9	–
TSOPF_RS_b2383_c1	1.3	**6.5**	2.1	71.8	34.9	–
TSOPF_RS_b2383	1.3	**6.5**	2.1	72.0	34.2	–
rajat25	6.7	**8.5**	35.2	31.7	37.2	2,675.4
rajat28	6.9	**9.1**	28.4	25.4	50.0	3,503.0
rajat20	7.0	**9.1**	35.2	31.3	40.5	4,314.1
ASIC_320k	2.5	30.4	42.9	447.5	**18.1**	7,908.2
ASIC_320ks	3.2	36.6	3.2	36.4	**21.5**	684.9
rajat30	5.1	73.0	3.2	**23.8**	**22.5**	–
Freescale1	3.9	**86.8**	3.9	**85.6**	–	–

Table 3 Refactorize+solve time in seconds

Matrix	KLU+BTF	KLU no BTF	SuperLU	Sparse 1.3
	Time	Time	Time	Time
Raj1	94.4	**3.0**	**3.3**	127.4
ASIC_680ks	**3.9**	5.4	**3.5**	256.7
ASIC_680k	**4.6**	**5.1**	**4.6**	835.8
rajat24	91.2	**3.7**	12.4	–
TSOPF_RS_b2383_c1	**5.2**	40.8	10.9	–
TSOPF_RS_b2383	**5.1**	41.0	10.9	–
rajat25	**6.7**	27.0	36.8	374.4
rajat28	**7.3**	21.8	49.6	512.7
rajat20	**7.3**	26.8	40.2	657.1
ASIC_320k	28.7	429.0	**17.1**	870.1
ASIC_320ks	35.0	35.0	**20.7**	182.0
rajat30	60.5	**18.6**	**19.6**	–
Freescale1	**70.5**	**70.6**	–	–

The time for the thirteen largest matrices is shown in Tables 2 and 3. The fastest run times and run times within 25% of the fastest are shown in bold. A dash is shown if the method ran out of memory.

For sparse Cholesky factorization, the flops per $|L|$ ratio is an accurate predictor of the relative performance of a BLAS-based supernodal method versus a non-supernodal method. If this ratio is 40 or higher, `chol` in MATLAB (and `x=A\b` for sparse symmetric positive definite matrices) automatically selects a supernodal

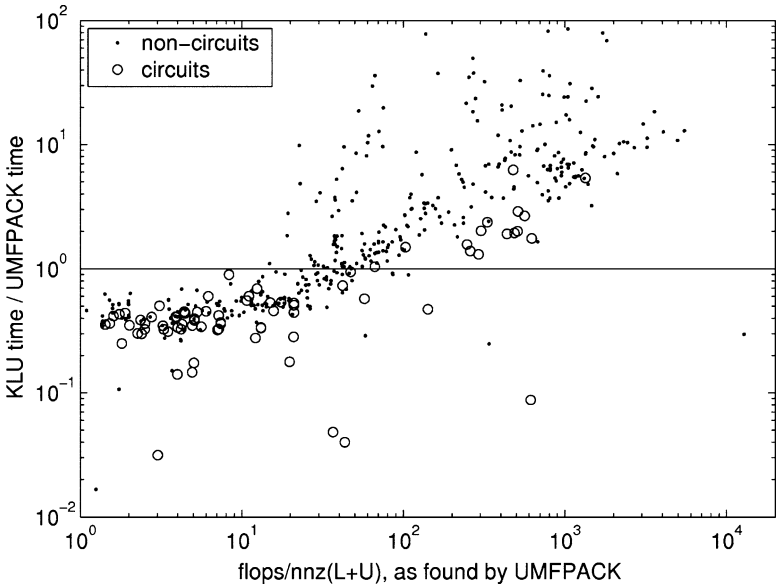

Fig. 3 Relative performance of KLU versus UMFPACK as a function of flops/$|L + U|$

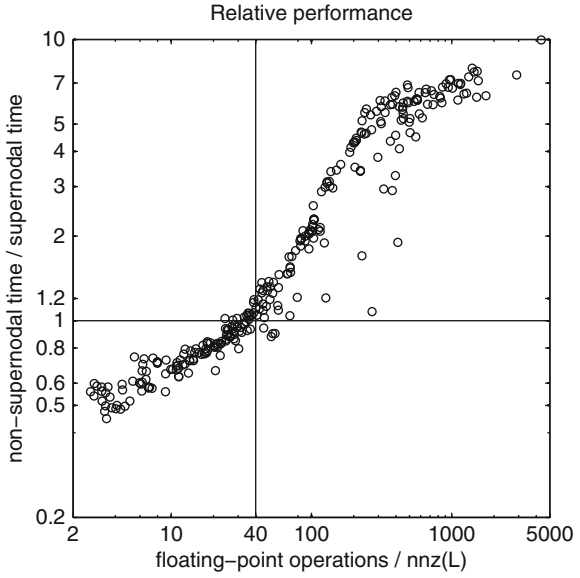

Fig. 4 Relative supernodal (BLAS-based) and non-supernodal (not BLAS-based) performance for sparse Cholesky

solver. Otherwise, a non-supernodal solver is used [3]. A similar comparison is shown in Fig. 3 between KLU and UMFPACK. If the matrix is reducible, only the largest block is factorized. Figure 4 shows the results for sparse Cholesky factorization from [3].

These results are remarkable for three reasons:

1. Circuit matrices tend to have a low flop/$|L + U|$ ratio as compared to other matrices.
2. Even when the flop/$|L+U|$ ratio is high enough (200 or more) to justify using the BLAS, the relative performance of a BLAS-based method (UMFPACK) versus KLU is much less than what would be expected if only non-circuit matrices were considered. Thus, circuits not only remain sparse when factorized, even large circuit matrices with higher flops/$|L + U|$ ratios hardly justify the use of the BLAS.
3. The flops/$|L + U|$ ratio for LU factorization (Fig. 3) is not a very accurate predictor of the relative performance of BLAS-based sparse methods as compared to non-BLAS-based methods, as it is for sparse Cholesky factorization (Fig. 4).

5 Summary

KLU has been shown to be an effective solver for the sequences of sparse matrices that arise when solving differential algebraic equations for circuit simulation problems. It is the default sparse solver in Xyce, a circuit simulation package developed by Sandia National Laboratories [19], for which it has been proven to be a robust and reliable solver [25].

Acknowledgements We would like to thank Mike Heroux for coining the name "KLU" and suggesting that we tackle this project in support of the Xyce circuit simulation package developed at Sandia National Laboratories [19, 25]. Portions of this work were supported by the Department of Energy, and by National Science Foundation grants 0203270, 0620286, and 0619080.

References

1. Amestoy, P.R., Davis, T.A., Duff, I.S.: An approximate minimum degree ordering algorithm. SIAM J. Matrix Anal. Appl. **17**(4), 886–905 (1996)
2. Amestoy, P.R., Davis, T.A., Duff, I.S.: Algorithm 837: AMD, an approximate minimum degree ordering algorithm. ACM Trans. Math. Softw. **30**(3), 381–388 (2004)
3. Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S.: Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. ACM Trans. Math. Softw. **35**(3), 1–14 (2008). DOI http://doi.acm.org/10.1145/1391989.1391995
4. Davis, T.A.: Algorithm 832: UMFPACK V4.3, an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. **30**(2), 196–199 (2002)
5. Davis, T.A.: Direct Methods for Sparse Linear Systems. SIAM, Philadelphia, PA (2006)
6. Davis, T.A., Duff, I.S.: An unsymmetric-pattern multifrontal method for sparse LU factorization. SIAM J. Matrix Anal. Appl. **18**(1), 140–158 (1997)
7. Davis, T.A., Duff, I.S.: A combined unifrontal/multifrontal method for unsymmetric sparse matrices. ACM Trans. Math. Softw. **25**(1), 1–19 (1999)

8. Davis, T.A., Gilbert, J.R., Larimore, S.I., Ng, E.G.: Algorithm 836: COLAMD, a column approximate minimum degree ordering algorithm. ACM Trans. Math. Softw. **30**(3), 377–380 (2004)
9. Davis, T.A., Gilbert, J.R., Larimore, S.I., Ng, E.G.: A column approximate minimum degree ordering algorithm. ACM Trans. Math. Softw. **30**(3), 353–376 (2004)
10. Davis, T.A., Hu, Y.: University of Florida sparse matrix collection. ACM Transactions on Mathematical Software, **38**(1), (2011). URL http://www.cise.ufl.edu/sparse/matrices
11. Davis, T.A., Palamadai Natarajan, E.: Algorithm 907: KLU, a direct sparse solver for circuit simulation problems. ACM Trans. Math. Softw. **37**(3), 36:1–36:17 (2010). DOI http://doi.acm.org/10.1145/1824801.1824814. URL http://doi.acm.org/10.1145/1824801.1824814
12. Demmel, J.W., Eisenstat, S.C., Gilbert, J.R., Li, X.S., Liu, J.W.H.: A supernodal approach to sparse partial pivoting. SIAM J. Matrix Anal. Appl. **20**(3), 720–755 (1999)
13. Dongarra, J.J., Du Croz, J., Duff, I.S., Hammarling, S.: A set of level-3 basic linear algebra subprograms. ACM Trans. Math. Softw. **16**(1), 1–17 (1990)
14. Duff, I.S.: Algorithm 575: Permutations for a zero-free diagonal. ACM Trans. Math. Softw. **7**(1), 387–390 (1981)
15. Duff, I.S.: On algorithms for obtaining a maximum transversal. ACM Trans. Math. Softw. **7**(1), 315–330 (1981)
16. Duff, I.S., Reid, J.K.: Algorithm 529: Permutations to block triangular form. ACM Trans. Math. Softw. **4**(2), 189–192 (1978)
17. Duff, I.S., Reid, J.K.: An implementation of Tarjan's algorithm for the block triangularization of a matrix. ACM Trans. Math. Softw. **4**(2), 137–147 (1978)
18. Gilbert, J.R., Peierls, T.: Sparse partial pivoting in time proportional to arithmetic operations. SIAM J. Sci. Statist. Comput. **9**, 862–874 (1988)
19. Hutchinson, S.A., Keiter, E.R., Hoekstra, R.J., Waters, L.J., Russo, T., Rankin, E., Wix, S.D., Bogdan, C.: The Xyce^{TM}parallel electronic simulator – an overview. In: Joubert, G.R., Murli, A., Peters, F.J., Vanneschi, M. (eds.) Parallel Computing: Advances and Current Issues, Proc. ParCo 2001, pp. 165–172. Imperial College Press, London (2002)
20. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**, 359–392 (1998)
21. Kundert, K.S.: Sparse matrix techniques and their applications to circuit simulation. In: Ruehli, A.E. (ed.) Circuit Analysis, Simulation and Design. North-Holland, New York (1986)
22. Kundert, K.S., Sangiovanni-Vincentelli, A.: User's guide: Sparse 1.3. Tech. rep., Dept. of EE and CS, UC Berkeley (1988)
23. Nichols, K., Kazmierski, T., Zwolinski, M., Brown, A.: Overview of SPICE-like circuit simulation algorithms. IEE Proc. Circuits, Devices & Sys. **141**(4), 242–250 (1994)
24. Palamadai Natarajan, E.: KLU - a high performance sparse linear system solver for circuit simulation problems. M.S. Thesis, CISE Department, Univ. of Florida (2005)
25. Sipics, M.: Sparse matrix algorithm drives SPICE performance gains. SIAM News **40**(4) (2007)
26. Tarjan, R.E.: Depth first search and linear graph algorithms. SIAM J. Comput. **1**, 146–160 (1972)