RAL-TR-2004-016

CCLRC

Numerical Analysis Group Progress Report January 2002 - December 2003

Iain S. Duff (Editor)

April 30, 2004

Council for the Central Laboratory of the Research Councils

Library and Information Services
CCLRC Rutherford Appleton Laboratory
Chilton Didcot
Oxfordshire OX11 0QX
UK
Tel: +44 (0)1235 445384
Fax: +44(0)1235 446403
Email: library@rl.ac.uk

CCLRC reports are available online at:
http://www.clrc.ac.uk/Activity/ACTIVITY=Publications;SECTION=225;

ISSN 1358-6254

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

RAL-TR-2004-016

Numerical Analysis Group Progress Report January 2002 - December 2003

Iain S. Duff (Editor)

ABSTRACT

We discuss the research activities of the Numerical Analysis Group in the Computational Science and Engineering Department at the Rutherford Appleton Laboratory of CLRC for the period January 2002 to December 2003. This work was supported by EPSRC grant R46441 until October 2003 and grant S42170 thereafter.

Keywords: sparse matrices, direct methods, iterative methods, ordering techniques, stopping criteria, numerical linear algebra, large-scale optimization, Harwell Subroutine Library, HSL, GALAHAD, CUTEr

AMS(MOS) subject classifications: 65F05, 65F50.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxon OX11 0QX

April 30, 2004.

Contents

Personnel in Numerical Analysis Group

Staff

Iain Duff.
Group Leader. Sparse matrices and vector and parallel computers and computing.

Mario Arioli
Numerical linear algebra, numerical solution of PDEs, error analysis.

Nick Gould.
Optimization and nonlinear equations particularly for large systems.

Jennifer Scott.
Sparse linear systems and sparse eigenvalue problems.

Kath Vann.
Administrative and secretarial support.

Consultants

Mike Hopper Support for Harwell Subroutine Library and for TSSD.

John Reid HSL, sparse matrices, automatic differentiation, and Fortran.

Visitors

Andy Conn (IBM) Optimization.

Bob Gate (Dundee) Optimization.

Yifan Hu (CLRC Daresbury) Sparse linear systems.

Michal Kočvara (Prague) Optimization.

Felix Kwok (Stanford University) Combining iterative and direct methods.

Gianmarco Manzini (CNR Pavia) Iterative methods.

Jorge Nocedal (Northwestern) Optimization.

Daniel Ruiz (ENSEEIHT) Linear algebra.

Philippe Toint (University of Namur) Optimization.

1 Introduction (I. S. Duff)

This report covers the period from January 2002 to December 2003 and describes work performed by the Numerical Analysis Group within the Computational Science and Engineering Department at the CLRC Rutherford Appleton Laboratory. This work was supported by EPSRC grant R46441 until October 2003 and grant S42170 thereafter.

The details of our activities are documented in the following pages. These words of introduction are intended merely to provide additional information on activities that are not appropriate for the detailed reports.

Perhaps the most exciting event during the last two years has been the renewal of our main EPSRC research grant from October 2003 for a period of four years. The responsive mode panel recommended that we were inspected by a visiting panel and this duly happened on 17th June 2003. The good immediate feedback was later confirmed in writing and the EPSRC were commendably swift in issuing our new grant after receipt of this positive report.

We were very pleased to organize a celebration in Oxford in December 2002 to commemorate the 80th birthday of Alan Curtis. Alan had been the Group Leader of Applied Mathematics at Harwell for many years and it was great to see both himself and so many of his contemporaries at the birthday celebrations.

There were no staff changes in the period of the report although Mario was promoted to Band 3 and both Jennifer Scott and Kath Vann increased their hours slightly after the award of the new Grant.

The support and development of HSL (formerly the Harwell Subroutine Library) continues to be one of our major activities. There was a release of HSL at the beginning of 2002 and another is scheduled for the summer of 2004. The HSL marketing effort continues to be supported by Lawrence Daniels and his team from Hyprotech, even though they have now left AEA and are owned by AspenTech Inc. We are still able to employ John Reid as a consultant using HSL funds. We also benefit from the consultancy of Mike Hopper who helps us both in typesetting and the ongoing commitment to higher software standards.

We maintain our close links with the academic community in Britain and abroad. Iain and Nick continue as Visiting Professors at Strathclyde University and Edinburgh University, respectively. Most members of the Group gave presentations at the Dundee Numerical Analysis meeting in 2003, with Mario giving an invited talk. We had several visitors during the period, including Andy Conn, Michal Kočvara, Gianmarco Manzini, Jorge Nocedal and Philippe Toint, whose visit was supported by an EPSRC Grant. Our CASE student with Dundee, Bob Gate, successfully defended his PhD. Iain was on the jury for the PhD theses of Bruno Carpentieri, Jean Christophe Rioual, Christof Vömel, and Julien Langou in Toulouse, and was a joint supervisor of Bruno and Christof. Bruno was awarded the Léopold Escande Prize for the best thesis at ENSEEIHT. Jennifer was the external examiner for the PhD thesis of Amanda Cooper of the University of Ulster. Nick gave a series of lectures in the LMS-EPSRC Numerical Analysis Summer School in Durham in 2002, and his lecture notes were published by Springer Verlag. In addition, he continues to act as external course assessor and examiner for the Open University's optimization module and is on the advisory board for the MSc in Bath. Iain and Nick are both on the Mathematics College of the EPSRC. Nick was a co-producer of 2003's UK Landscape document on Numerical Analysis, with Nick Higham and Endre Süli.

We continue our close association with Oxford University through the Joint Computational Mathematics and Applications Seminar series and have hosted several talks at RAL through that programme (see Section 7). Nick Trefethen, the professor of Numerical Analysis at Oxford University, has made an office available to the Group that has been used for visits by all Group members, significantly by Nick who visits on a regular basis. Nick taught an MSc course on nonlinear optimization at Oxford in Trinity term of 2002. Iain and Jennifer gave an MSc course at Oxford on Direct Methods for Sparse Matrices in the Michaelmas term of 2002.

Nick's collaboration with Toint and others continues to expand the theory and practice of large-scale optimization. He has developed new algorithms and codes for nonlinear feasibility problems and general nonlinear programming. The resulting packages have been, or will be, added to the nonlinearly constrained optimization library GALAHAD, which was first released in April 2002. He is also collaborating on algorithms for a rival nonlinear programming package KNITRO. Nick was an invited speaker at conferences in Leipzig, Toronto and Minneapolis, and gave seminars at the Universities of Bath, Dundee, Essex and Edinburgh. He has been involved in educating both students and users of the benefits of using optimization techniques, and has written a number of lecture notes/survey articles with this in mind. In addition to his Oxford MSc course, he taught a similar one in Edinburgh in 2003. Nick joined the editorial board of the ACM Transactions on Mathematical Software in 2002. He was a member of the Beale-Orchard-Hays prize committee for the Mathematical Programming Society in 2003. Nick is co-organiser of a series of annual joint Numerical Analysis Days with the University of Bath, which first met in Bath in 2002 and subsequently at RAL in 2003.

Jennifer has continued with her national and international collaborations. Although she has continued her short-hours working, she remains so productive that it is easy to forget this fact. She has spent a significant amount of her time in this period designing and developing a coarse-grained parallel solver for large sparse highly unsymmetric linear systems. She has developed efficient techniques for preordering matrices to the form required for using this solver. She has also worked with Nick on an extensive exercise to compare the performance of direct solvers on large sparse symmetric linear systems. This study has encompassed both HSL and non-HSL solvers and should prove an invaluable guide to those who need to solve such systems. Jennifer presented invited talks at AspenWorld 2002, Bath, Daresbury and Kingston, and continues to coordinate our joint seminar series with Oxford University.

Mario was successful in a review promotion to Band 3 in 2003 and complements well the skills of other team members, particularly with his knowledge of partial differential equations and his enthusiasm for the minutiae of error analysis. He gave invited talks at conferences in Dundee and Bari. He gave seminars at Bath, Strathclyde, and CERFACS. The ERCIM Working Group Applications of Numerical Mathematics in Science, of which Mario has been chairman from 2002, asked Mario to be the co-ordinator and person responsible for the Marie-Curie RTN proposal Applications of Computational Mathematics and Statistics in Science and Technology (NUMAS). This proposal is discussed in Section 5.2.

Iain still leads a project at the European Centre for Research and Advanced Training in Scientific Computation (CERFACS) at Toulouse in France (see Section 5.3). His research interests continue to be in all aspects of sparse matrices, including more recently iterative methods as well as direct methods, and in the exploitation of parallel computers. He is an Editor of the IMA Journal of Numerical Analysis, the Vice-President of the IMA for learned society affairs, editor of the IMANA Newsletter, and IMA representative on the International Committee that oversees the ICIAM international conferences on applied mathematics. He was elected to the Board of Trustees of SIAM in the USA. He gave lectures at summer schools in Canberra, Kentucky, and Neuchatel and was workshop coordinator for a meeting in Copper Mountain, Colorado. He has been on the Programme and Organizing Committee for several international meetings. He gave a plenary invited talk at the Householder meeting in Peebles and has given invited talks at meetings in Calais, Canberra, Milovy, Neuchatel, Oxford and Vancouver and has presented seminars in Argonne, Brisbane, Dublin, Knoxville, LBNL, Liverpool, and Thailand. The work on the book with John Reid also progressed well during this period, and it is hoped to have a draft with OUP during 2004.

We have tried to subdivide our activities to facilitate the reading of this report. This is to some extent an arbitrary subdivision since much of our work spans these subdivisions. Our main research areas and interests lie in numerical linear algebra, and nonlinear systems and optimization. We are particularly concerned with large-scale systems when the matrix or system is sparse or structured. We discuss the solution of linear systems by direct methods in Section 2 and by iterative techniques in Section 3. Work on optimization is considered in Section 4. We group some miscellaneous topics in Section 5. Much of our research and development results in high quality advanced mathematical software, and we report on developments with HSL in Section 6. Lists of seminars (in the joint series with Oxford), technical reports, and publications are given in Sections 7, 8, and 9, respectively. Current information on the activities of the Group and on Group members can be found through page `http://www.cse.clrc.ac.uk/nag` of the World Wide Web.

2 Direct methods

2.1 A numerical evaluation of sparse direct symmetric solvers. Part I: HSL solvers. N. I. M. Gould and J. A. Scott

In recent years a number of new direct solvers for the solution of large sparse, symmetric linear systems of equations have been added to the mathematical software library HSL. These include solvers that are designed for the solution of positive-definite systems as well as those that are principally intended for solving indefinite problems. The available choice can make it difficult for users to know which solver is the most appropriate for their use. This study aimed to compare the alternatives (see Table 2.1.1) on a significant set of large test examples from many different application areas and, as far as is possible, to make recommendations concerning the efficacy of the various packages.

Code	Description
MA27	Sparse symmetric linear solver. Multifrontal algorithm. Minimum degree ordering.
MA47	Sparse symmetric indefinite linear solver.
MA55	Variable band symmetric positive-definite linear solver.
MA57	Sparse symmetric linear solver. Multifrontal algorithm. Approximate minimum degree ordering.
MA62	Sparse symmetric positive-definite linear solver for equations in elemental form. (Uni)frontal algorithm.
MA67	Sparse symmetric indefinite linear solver. Analyse-factorize code.

Table 2.1.1: HSL codes used in our numerical experiments.

In collecting test data we imposed only two conditions: each matrix had to be of order greater than $10,000$ and to be available to other users. We included all matrices of which we are aware that satisfy these conditions and are in publicly available sparse matrix test collections. The final test set $\mathcal{T}$ comprised 88 positive-definite problems and 61 numerically indefinite problems and included problems from linear programming, structural engineering, computational fluid dynamics, acoustics, and financial modelling.

Performance profiles were used to evaluate and compare the performance of the set $\mathcal{S}$ of HSL solvers on the test set $\mathcal{T}$. Suppose that a given code $i \in \mathcal{S}$ reports a statistic $s_{ij} \geq 0$ when run on example $j \in \mathcal{T}$, and that the smaller this statistic the better the software is considered to be. For example, s_{ij} might be the CPU time required to solve problem j using code i. For all problems $j \in \mathcal{T}$, we wanted to compare the performance of code i with the performance of the best solver in the set $\mathcal{S}$.

For $j \in \mathcal{T}$, let $\hat{s}_j = \min\{s_{ij}; i \in \mathcal{S}\}$. Then for $\alpha \geq 1$ and each $i \in \mathcal{S}$ we define

$$k(s_{ij}, \hat{s}_j, \alpha) = \begin{cases} 1 & \text{if } s_{ij} \leq \alpha \hat{s}_j \\ 0 & \text{otherwise.} \end{cases}$$

The *performance profile* (see Dolan and Moré, 2002) of code i is the function

$$p_i(\alpha) = \frac{\sum_{j \in \mathcal{T}} k(s_{ij}, \hat{s}_j, \alpha)}{|\mathcal{T}|}, \quad \alpha \geq 1.$$

Thus $p_i(1)$ gives the fraction of the examples in the test set for which code i is the most effective (according to the statistic s_{ij}), $p_i(2)$ gives the fraction for which it is within a factor of 2 of the best, and $\lim_{\alpha \longrightarrow \infty} p_i(\alpha)$ gives the fraction for which the solver succeeded.

In this study, the statistics used were the CPU times required by the different phases of the solver, the number of nonzero entries in the matrix factor, and the total memory used by the solver. Our findings are presented in detail in Gould and Scott (2003*a*) and Gould and Scott (2003*b*). The broad conclusions were that the best general-purpose HSL package for solving sparse symmetric systems is currently `MA57`, particularly for very large problems. The approximate minimum-degree ordering proved to be more effective than the traditional minimum-degree algorithm but for large, positive-definite problems, it was generally advantageous to use a nested dissection ordering (which is not currently offered as part of the `MA57` package). A weakness of `MA57` is that it offers no out-of-core working. We feel that this should be an important future development.

For indefinite problems, we found that a tiny pivot tolerance can often be better than the default (reducing factorization and solve times and producing sparser factors). In addition, scaling can offer significant improvements. However, the robustness of the solvers for the large, indefinite case is a cause for concern. Four of the 61 indefinite problems were not solved by any HSL code.

References

E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, **91**(2), 201–213, 2002.

N.I.M. Gould and J.A. Scott. Complete results for a numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations. Numerical Analysis Internal Report 2003-2, Rutherford Appleton Laboratory, 2003*a*. Available from www.numerical.rl.ac.uk/reports/reports.shtml.

N.I.M. Gould and J.A. Scott. A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations. Technical Report RAL-TR-2003-019, Rutherford Appleton Laboratory, 2003*b*.

2.2 A numerical evaluation of sparse direct symmetric solvers. Part II: all solvers. N. I. M. Gould, Y. Hu, and J. A. Scott

This study is an extension of the comparison of the sparse symmetric direct HSL solvers reported on in Section 2.1. Our aim was to compare the best general-purpose HSL solver `MA57` with as many other sparse symmetric solvers as possible. The solvers are listed in

Table 2.2.1. A number have parallel versions (and may even have been written primarily as parallel codes); this study considered only serial codes and serial versions of parallel solvers.

Code	Date/version	Authors
BCSLIB-EXT	11.2001, v4.1	The Boeing Company
MA57	03.2003	I.S. Duff, HSL
MUMPS	11.2003, v4.3.2	P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, and J. Koster
Oblio	12.2003, v0.7	F. Dobrian and A. Pothen
PARDISO	12.2003	O. Schenk and K. Gärtner
SPOOLES	1999, v2.2	C. Ashcraft and R. Grimes
SPRSBLKLLT	1997, v0.5	E.G. Ng and B.W. Peyton
TAUCS	08.2003, v2.2	S. Toledo
UMFPACK	04.2003, v4.1	T. Davis
WSMP	2003, v1.9.8	A. Gupta and M. Joshi, IBM

Table 2.2.1: Solvers used in our numerical experiments.

As in the first study, performance profiles were used to evaluate the performance of the solvers. Additionally, the same large set of test problems and the same statistics were used. Reports on this study and its findings will appear shortly.

2.3 Task scheduling for the MUMPS package (P. R. Amestoy, I. S. Duff, S. Pralet, and C. Vömel)

MUMPS, a MUltifrontal Massively Parallel Solver (Amestoy, Duff, Koster and L'Excellent, 2001) was originally developed with the support of the EU LTR project PARASOL and continues to be maintained and developed with Version 4.3 being released in July 2003. This includes a code for complex systems and for sequential operation. MUMPS is distributed on almost a daily basis from its main web site at http://www.enseeiht.fr/lima/apo/MUMPS. The main research work done on MUMPS over this reporting period was on task scheduling and task assignment.

The asynchronous distributed memory multifrontal solver MUMPS exploits two kinds of parallelism when a sparse matrix is factorized. A first natural source of parallelism is established by independent branches of the assembly tree. Furthermore, tree nodes with a large enough block to update can be updated in parallel by splitting the update between several slaves of the master that is factorizing the block of fully summed variables, and the root node can be treated in parallel if it is big enough. MUMPS uses dynamic data structures and dynamic scheduling of computational tasks to accommodate extra fill-in in the factors due to numerical considerations (not taken into account during the analysis step). This dynamic approach also allows the parallel code to cope with load variations on the processors and we have investigated and developed this over the last two years.

While the *master* processor of each node in the tree is chosen during the analysis phase, the *slaves* for the parallel update of large contribution blocks are only chosen during the factorization phase. This dynamic task scheduling takes place in order to balance the work load of the processors at run-time. Problems arise from offering too much freedom to the dynamic scheduling. If every processor is a candidate for a slave then, on each processor, enough workspace has to be reserved during the analysis phase for the corresponding computational tasks. However, during the factorization, typically not all processors are actually needed as slaves (and, on a large number of processors, only a very few are needed), so the prediction of the required workspace will be overestimated. Thus the size of the problems that can be solved is reduced unnecessarily because of this difference between the prediction and allocation of memory in the analysis phase and the memory actually used during the factorization.

With the concept of *candidate processors* it is possible to address this issue. The concept originates in an algorithm presented by Pothen and Sun (1993) and extends efficiently to MUMPS. For each node that requires slaves to be chosen dynamically during the factorization because of the size of its contribution block, we introduce a limited set of processors from which the slaves can be selected. While the master previously chose slaves from among all less loaded processors, the freedom of the dynamic scheduling can be reduced so that the slaves are only chosen from the candidates. This effectively allows us to exclude all non-candidates from the estimation of workspace during the analysis phase and leads to a more realistic prediction of workspace needed. Furthermore, the candidate concept allows us to structure the computation better since we can explicitly restrict the choice of the slaves to a certain group of processors and enforce a 'subtree-to-subcube' mapping principle.

Our new approach significantly improves the scalability of the solver in terms of execution time and storage. By comparison with the previous version of MUMPS, we demonstrate the efficiency and the scalability of the new algorithm on up to 512 processors. Our test cases include matrices from regular 3D grids and irregular ones from real-life applications (Amestoy, Duff and Vömel, 2002). We show some of the results of our new strategy in Table 2.3.1 where the problem size scales with the number of processors and the new code is clearly far more scalable than the old.

	Cubic grids (ND)			Rectangular grids (ND)		
Processors	flops	old	new	flops	old	new
1	3.6e+09	19.1	18.7	2.2e+09	13.5	13.1
16	5.9e+10	18.8	19.8	3.6e+10	13.8	13.2
32	1.1e+11	25.8	22.2	6.8e+10	15.5	15.3
48	1.8e+11	28.7	30.4	9.0e+10	14.2	14.8
64	2.2e+11	30.7	25.6	1.2e+11	17.6	16.8
128	4.4e+11	45.6	33.0	2.4e+11	33.5	20.3
256	9.1e+11	109.1	43.0	3.8e+11	45.2	18.4
512	1.7e+12	421.9	64.0	7.1e+11	195.5	24.3

Table 2.3.1: Performance of old and new versions of the LDL^T MUMPS factorization (time in seconds on a CRAY T3E).

We have also extended this task scheduling to exploit clusters of SMPs. Our main target type of computer architecture can be defined as a two-level architecture where a number of nodes are connected by a network and each node is composed of a number of identical processors sharing a common memory (that is, is an SMP node). The work in Amestoy et al. (2002) implicitly assumed that our target computer was a distributed memory computer with uniform memory access and uniform cost of communication. We showed the limitations of this approach on a machine with a two-level architecture and we indicated how we can remedy these limitations in Amestoy, Duff, Pralet and Vömel (2003). Our modifications of the algorithms affect both the symbolic factorization and the numerical factorization phase. Our experiments on the IBM SP from CINES (Montpellier) with 16 processors per SMP node and up to 128 processors show that we can significantly reduce both the amount of inter-node communication and the factorization time. For example, on 3D cubic grid problems we can reduce the factorization time on 64 processors from 204 seconds to 145 seconds. The algorithms have been integrated into Version 4.3 of MUMPS.

Correspondence or enquiries about MUMPS can be conducted using the email address mumps@cerfacs.fr.

References

P. R. Amestoy, I. S. Duff, and C. Vömel. Task scheduling in an asynchronous distributed memory multifrontal solver. Technical Report RAL-TR-2002-033, Rutherford Appleton Laboratory, 2002.

P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Analysis and Applications*, **23**(1), 15–41, 2001.

P. R. Amestoy, I. S. Duff, S. Pralet, and C. Vömel. Adapting a parallel sparse direct solver to architectures with clusters of SMPs. *Parallel Computing*, **29**(11-12), 1645–1668, 2003.

A. Pothen and C. Sun. A mapping algorithm for parallel sparse Cholesky factorization. *SIAM J. Scientific Computing*, **14**(5), 1253–1257, 1993. Timely Communication.

2.4 Symmetric weighted matching and preselection of 2×2 pivots in indefinite multifrontal solvers (I. S. Duff, J. R. Gilbert, and S. Pralet)

We have studied techniques for scaling and choosing pivots when using multifrontal methods in the LDL^T factorization of symmetric indefinite matrices, where L is a lower triangular matrix and D is a block diagonal matrix with 1×1 and 2×2 blocks.

For the LU factorization of a matrix A, `MC64` (Duff and Koster, 1999, Duff and Koster, 2001) can be used to get a maximum weighted matching so that the corresponding permutation will place large entries on the diagonal. The matrix can then be scaled so that diagonal entries have modulus one and off-diagonals have modulus less than or equal to one. This has been found to greatly improve the numerical stability of the subsequent LU factorization.

If, however, `MC64` is applied to a symmetric matrix the resulting permutation will not normally preserve symmetry. We examined ways in which `MC64` can be used while still preserving symmetry. Then an ordering, for example, AMD (Amestoy, Davis and Duff, 1996), can be computed on the permuted matrix to get a symmetric permutation in order to decrease the fill-in in the factors. We have also experimented with using `MC64` to predetermine suitable 2×2 pivots and using AMD on a reduced graph where nodes correspond to the 1×1 and 2×2 preselected pivots.

We have used these techniques with two symmetric multifrontal codes `MA47` (Duff and Reid, 1996) and `MA57` (Duff, 2002) and our initial results have been very encouraging.

We plan to present our preliminary results at a meeting in San Francisco in February 2004. A technical report of our results will be available shortly.

References

P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Analysis and Applications*, **17**(4), 886–905, 1996.

I. S. Duff. MA57 – A new code for the solution of sparse symmetric indefinite systems. Technical Report RAL-TR-2002-024, Rutherford Appleton Laboratory, Oxfordshire, England, 2002. To appear in *ACM Trans. Math. Softw.*

I. S. Duff and J. Koster. The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM J. Matrix Analysis and Applications*, **20**(4), 889–901, 1999.

I. S. Duff and J. Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J. Matrix Analysis and Applications*, **22**(4), 973–996, 2001.

I. S. Duff and J. K. Reid. Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems. *ACM Trans. Math. Softw.*, **22**(2), 227–257, 1996.

2.5 Ordering techniques for stable block diagonal forms for unsymmetric parallel sparse solvers (I. S. Duff, Y. Hu, and J. A. Scott)

One approach to efficiently solving very large sparse linear systems of equations $Ax = b$ is to reorder the system matrix to bordered block diagonal form and then to solve the block system in parallel. We are interested in developing new algorithms for rapidly ordering unsymmetric systems to singly bordered block diagonal (SBBD) form. That is, we want to order A to the form

$$PAQ = A_{SB} = \left(\begin{array}{ccccc} A_{11} & & & & C_1 \\ & A_{22} & & & C_2 \\ & & \ldots & & . \\ & & & A_{NN} & C_N \end{array} \right), \qquad (2.5.1)$$

where the rectangular blocks on the diagonal A_{ll} are $m_l \times n_l$ matrices with $m_l \geq n_l$ and $\sum_{k=1}^{N} m_l = n$ (the order of A), and the border blocks C_l are $m_l \times k$ with $k \ll n_l$. The block diagonals lead to subproblems that can be solved independently, leaving an interface problem that links the subproblems to be solved to complete the solution of the original problem. For the method to work well in parallel, the interface problem needs to be small compared with n so that communication between the blocks is minimized. Recently, a number of solvers that implement direct algorithms based on this coarse-grained parallel approach have been developed for HSL. These include `HSL_MP43` (Scott, 2001) and `HSL_MP48` (Duff and Scott, 2002) (see Section 6).

The MONET algorithm (Hu, Maguire and Blake, 2000) is designed for ordering highly unsymmetric chemical process engineering problems to SBBD form. HSL offers a Fortran 95 implementation of the MONET algorithm as routine `HSL_MC66`. For highly unsymmetric problems and $N \leq 32$, `HSL_MC66` produces well-balanced SBBD forms with narrow borders. However, when compared with the total time required to solve the linear system, `HSL_MC66` is relatively expensive. This prompted us to try and develop alternative algorithms. The algorithms of Hu and Scott are based on computing either a vertex separator or a wide separator of the symmetrized matrix $A^T + A$. A graph partitioning tool such as the well-known METIS package is used to partition the graph of $A^T + A$. A vertex separator is extracted from the output; it is optionally widened to a wide separator and then used to partition the matrix. Hu and Scott report on a number of variants of this separator approach and compare their performance with that of MONET (Hu and Scott, 2003).

Identifier	n	$\hat{n}$	
bayer01	57735	57916	(0.31)
lhr71c	70304	70764	(0.65)
10cols	29496	29602	(0.36)
circuit_4	80209	80924	(0.89)
scircuit	170998	171327	(0.19)
venkat50	62424	63544	(1.79)

Table 2.5.1: The order of the stretched matrix $\hat{A}_{SB}$ ($N = 8$). The figures in parentheses are the percentage increases in the order of the system.

More recently, we have considered a two-phase approach. In the first phase, A is ordered to doubly bordered block diagonal (DBBD) form A_{DB} and then in the second phase, row stretching is used to obtain an SBBD form $\hat{A}_{SB}$ of order $\hat{n}$. Stretching is a sparse matrix preprocessing technique that makes matrices sparser but, at the same time, larger (Grcar, 1990). In Table 2.5.1, n is compared with $\hat{n}$ for a number of test examples ($N = 8$). The results are encouraging since the small increase in the order of the system is unlikely to add a significant overhead to the time needed to solve the system.

Identifier	n	SEP_VS(ND)	MONET	Two-phase
bayer01	57735	458	**254**	431
lhr71c	70304	**883**	990	1163
10cols	29496	313	**279**	**263**
circuit_4	80209	19305	16417	**1107**
scircuit	170998	1274	4353	**581**
venkat50	62424	2492	2536	**2232**

Table 2.5.2: The size of the border in the 8-block SBBD form computed by the different algorithms.

In Table 2.5.2, border sizes for the two-phase approach are compared with those obtained using MONET and the separator method of Hu and Scott (denoted by SEP_VS(ND)). The narrowest borders and those that are within 5 percent of the best are highlighted in bold. The results show that no one approach always gives the narrowest borders. However, for (nearly) symmetric problems (including scircuit and venkat50) the two-phase approach outperforms the SEP_VS(ND) method and we found that MONET only produces significantly smaller borders for a subset of our highly unsymmetric test problems. Further results will shortly be available (Duff and Scott, 2004).

References

I.S. Duff and J.A. Scott. A parallel direct solver for large sparse highly unsymmetric linear systems. Technical Report RAL-TR-2002-033, Rutherford Appleton Laboratory, 2003.

I.S. Duff and J.A. Scott. A parallel direct solver for large sparse highly unsymmetric linear systems. Technical Report, Rutherford Appleton Laboratory. To appear, 2004.

J. F. Grcar. Matrix stretching for linear equations. Technical Report SAND90-8723, Sandia National Laboratories, Albuquerque, 1990.

Y.F. Hu and J.A. Scott. Ordering techniques for singly bordered block diagonal forms for unsymmetric parallel sparse direct solvers. Technical Report RAL-TR-2003-020, Rutherford Appleton Laboratory, 2003.

Y.F. Hu, K.C.F. Maguire, and R.J. Blake. A multilevel unsymmetric matrix ordering for parallel process simulation. *Computers in Chemical Engineering*, **23**, 1631–1647, 2000.

J.A. Scott. The design of a portable parallel frontal solver for chemical process engineering problems. *Computers in Chemical Engineering*, **25**, 1699–1709, 2001.

2.6 A parallel direct solver for large sparse highly unsymmetric linear systems (I. S. Duff and J. A. Scott)

The HSL code `MA48` was developed in the early 1990s by Duff and Reid (1996) for solving unsymmetric sparse linear systems $Ax = b$. The code has been highly successful and has been incorporated into a number of commercial software packages. However, the size of problem `MA48` can solve is limited by the amount of computer memory available for holding the matrix A and its factors. To solve larger problems, as well as to solve problems more quickly, our aim was to develop a parallel version of `MA48`. Because `MA48` is a well-established code that represents substantial programming effort and expertise, we were anxious to exploit the existing code as far as possible in developing a parallel version. The idea behind our parallel approach was to preorder the matrix A to SBBD form (2.5.1) and then apply a direct solver to each of the diagonal blocks in parallel. Solving the interface problem that links the subproblems completes the solution. We have already used this approach in the development of the parallel frontal solvers `HSL_MP42`, `HSL_MP43`, and `HSL_MP62`, and the experienced gained from these codes has been used in developing our new general-purpose parallel direct solver, `HSL_MP48`.

For portability, `HSL_MP48` is written in Fortran 90 and uses MPI for message passing. It may be used on shared or distributed memory machines and may be run on a single process or on up to N processes, where N is the number of blocks in the SBBD form. In practice, we have found that $N \leq 16$ generally gives the best results; for larger N solving the interface problem (which is currently performed on a single process using `MA48`) can become a bottleneck and limit the speedup that is achieved. HSL_MP48 allows the matrix data to be "shared" between the processes, with each process only requiring access to the submatrices assigned to it. Furthermore, at the end of the submatrix factorization, the user may optionally choose to write the matrix factors to unformatted sequential files. This increases the size of problem that can be solved.

On each of the diagonal blocks, HSL_MP48 uses a modified version of MA50, the code that lies at the heart of MA48, to perform the partial factorization. The main change we needed to make to MA50 was to prevent pivoting on columns with entries belonging to more than one block. In addition, the Schur complement remaining at the end of the factorization of the diagonal blocks had to be extracted from the data structures and passed to the processor responsible for the interface problem. Although implementing the changes was non-trivial, most of the MA50 code was unaltered. The modified routines are presently internal to the HSL_MP48 package but may later form the basis for a user-callable package.

Identifier	Order	MA48	HSL_MP48 ($N = 8$)						
			Number of processors						
			1	2		4		8	
4cols	11770	2.47	0.67	0.41	*1.63*	0.29	*2.31*	0.23	*2.91*
lhr14c	14270	7.23	8.87	4.88	*1.82*	2.87	*3.09*	1.74	*5.09*
10cols	29496	16.4	2.75	1.60	*1.72*	0.93	*2.96*	0.65	*4.23*
lhr34c	35152	24.2	30.1	16.2	*1.85*	9.80	*3.07*	5.89	*5.11*
bayer01	57735	6.37	4.23	2.39	*1.77*	1.48	*2.86*	0.97	*4.36*
lhr71c	70304	50.6	71.2	39.8	*1.79*	22.3	*3.19*	12.4	*5.74*

Table 2.6.1: Timings for HSL_MP48 for chemical process engineering test problems.

In Table 2.6.1, wallclock timings (in seconds) for HSL_MP48 run on up to 8 processors of an SGI Origin 2000 are presented, together with timings for MA48 run on a single processor. The timings are for factorizing the matrix and then solving for a single right-hand side. The numbers in italics are the speedups for HSL_MP48 compared with using a single processor. We see that good speedups are achieved and, for some problems, HSL_MP48 on a single processor is sometimes faster than MA48. Further results are given in Duff and Scott (2002). The package HSL_MP48 is available now and will be included in HSL 2004.

References

I.S. Duff and J.K. Reid. The design of MA48, a code for the direct solution of sparse unsymmetric linear systems of equations. *ACM Trans. Mathematical Software*, **22**, 187–226, 1996.

I.S. Duff and J.A. Scott. A parallel direct solver for large sparse highly unsymmetric linear systems. Technical Report RAL-TR-2002-033, Rutherford Appleton Laboratory, 2002.

2.7 HSL_MC73: A fast multilevel spectral and profile reduction code (Y. Hu and J. A. Scott)

In recent years there has been considerable interest in spectral orderings for use in graph partitioning, image analysis, profile and wavefront reduction algorithms for sparse symmetric matrices, and the study of range-dependent random graphs. Spectral orderings are dependent upon the computation of the eigenvector corresponding to the smallest non-zero eigenvalue of the Laplacian matrix associated with the graph of the problem, the so-called Fiedler vector. If the original problem is sparse then the Laplacian matrix is sparse, symmetric positive-semidefinite, and of the same order as the problem. Computing the Fiedler vector using a Lanczos algorithm is expensive for large problems; an alternative is a multilevel approach (Barnard and Simon, 1994).

Even with a multilevel implementation, computing a spectral ordering for use as a profile reduction ordering is still significantly more expensive than using a heuristic algorithm such as Reverse Cuthill-McKee or the Sloan method. This motivated the development by Hu and Scott (2001) of an efficient multilevel algorithm for profile and wavefront reduction that in terms of quality is competitive with the hybrid Sloan algorithm of Kumfert and Pothen (1997) but is faster since it avoids the need for any spectral information.

We have designed and developed a new flexible software package that implements both the multilevel spectral algorithm and the multilevel profile reduction algorithm. Both algorithms are included in a single package because they employ similar techniques and the software needed to implement them contains common elements; it is thus efficient for software development and maintenance to incorporate both within a single package. The new code, HSL_MC73, is written in Fortran 95 and will be included in HSL 2004. Full details of the code, together with numerical results that illustrate its performance when used to compute spectral reorderings for undirected range-dependent random graphs, are given in the report by Hu and Scott (2003).

References

S.T. Barnard and H.D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience*, **6**, 101–117, 1994.

Y.F. Hu and J.A. Scott. A multilevel algorithm for wavefront reduction. *SIAM J. Scientific Computing*, **23**, 1352–1375, 2001.

Y.F. Hu and J.A. Scott. HSL_MC73 : A fast multilevel spectral and profile reduction code Technical Report RAL-TR-2003-036, Rutherford Appleton Laboratory, 2003.

G. Kumfert and A. Pothen. Two improved algorithms for envelope and wavefront reduction. *BIT*, **37:3**, 559–590, 1997.

2.8 Cholesky factorization of a dense matrix with high performance in packed storage (John Reid, Fred Gustavson, Jerzy Waśniewski, John Gunnels, and Bjarne Andersen)

In designing the Level 3 BLAS, Dongarra, Du Croz, Duff and Hammarling (1990) chose not to address packed storage schemes for symmetric matrices because 'such storage schemes do not seem to lend themselves to partitioning into blocks ... Also packed storage is required much less with large memory machines available today'. The aim of this work was to investigate whether packing is possible without substantial loss of performance.

We took the same starting point as that of LINPACK and LAPACK, with the upper (or lower) triangular part of the matrix being stored by columns. Following LINPACK and LAPACK, we overwrite the given matrix by its Cholesky factor.

2.8.1a. Lower Packed Format

0									
1	10								
2	11	19							
3	12	20	27						
4	13	21	28	34					
5	14	22	29	35	40				
6	15	23	30	36	41	45			
7	16	24	31	37	42	46	49		
8	17	25	32	38	43	47	50	52	
9	18	26	33	39	44	48	51	53	54

2.8.1b. Lower Blocked Hybrid Format

0									
1	2								
3	4	5							
6	7	8	27						
9	10	11	28	29					
12	13	14	30	31	32				
15	16	17	33	34	35	45			
18	19	20	36	37	38	46	47		
21	22	23	39	40	41	48	49	50	
24	25	26	42	43	44	51	52	53	54

Figure 2.8.1: Lower Packed and Blocked Hybrid Formats.

In the lower packed form, the lower-triangular part of the matrix is held by columns, as illustrated on the left of Figure 2.8.1. After careful consideration of alternatives, we have chosen to sort each block column to be ordered by rows. This is illustrated for block size $nb = 3$ on the right of Figure 2.8.1. The sort can be performed efficiently with the help of a buffer of size $n \times nb$ when the order is n and the block size is nb.

We use the notation A_{ij} and L_{ij} for the blocks of the given matrix and its Cholesky factor. The key calculation, performed by the BLAS routine _GEMM, is

$$A_{ij} = A_{ij} - L_{ik} L_{jk}^T.$$

This will be efficient if the block size is chosen so that the blocks fit comfortably in Level-1 cache.

If the number of blocks is not very large, that is, if n is not much larger than nb, the factorization of the diagonal blocks is also important. By making temporary copies in full format, we can use the LAPACK routine _POTRF, but this is unsatisfactory since it itself is a block algorithm and calls _POTF2 for its diagonal blocks and this employs Level 2 BLAS. We have therefore written our own 'kernel' Cholesky code that uses a block size kb designed to fit in registers and have found that the block size $kb = 2$ is suitable. This code, written in Standard Fortran has proved to be remarkably fast on all six of our test platforms. It significantly outperforms the vendors' codes on all but the IBM Power4, where it is only marginally slower.

n	40	100	250	640	1600	4000
Packed LAPACK	747	1043	1059	1037	638	635
Vendor Packed LAPACK	1750	2658	3107	3773	3969	3836
Full LAPACK	440	1390	2562	3495	3901	4010
Vendor Full LAPACK	1492	2486	3454	3832	4162	4327
Packed Recursive+	170	593	1586	2621	3434	3943
Packed Recursive	181	618	1652	2700	3523	3980
Packed Hybrid+	878	2085	3211	3974	4188	4275
Packed Hybrid	1006	2334	3441	4149	4266	4309

Table 2.8.1: Mflops, Cholesky factorizations, $nb = 100$, IBM Power4.

For our main performance tests we varied n between 40 and 4000 and ran on six platforms that are in wide use. In Table 2.8.1, we show some factorization speeds on the IBM Power4 computer. The first row shows the performance of the LAPACK code _PPTRF when compiled with full optimization and calling the vendor-supplied BLAS. This performance deteriorates markedly as n increases beyond 640. We believe that this is because Level 2 BLAS are being used. None of the other codes have this defect. The ESSL library contains an equivalent code for the lower packed format and we show its speed in the second line of the table.

The next two lines show the speeds of comparable codes for the full format and provide our benchmark. We note that the vendor codes are much faster for small n, which is probably because the LAPACK code uses Level 2 BLAS (_POTF2) to factorize the blocks on the diagonal, but are only slightly faster for $n \geq 1000$ where the speed of _GEMM is of prime importance.

There are two rows for each of the recursive (Andersen, Gustavson and Waśniewski, 2001) and hybrid formats, according to whether the overheads of rearrangement to this format are included. We do not include rearrangement of the factor back to ordinary packed format since the recursive or hybid format is more suitable for forward and back substitution.

The recursive algorithms achieve performance that approaches that of the LAPACK full codes when the order is large. This is because both are then doing most of their work in significant calls of the Level 3 BLAS _GEMM. However, for smaller n, their performance is poor, probably because of the larger ratio of procedure calls to actual computation.

The hybrid algorithm is much faster than the recursive algorithm for small n, is significantly faster for medium n, and is slightly faster for large n.

If we compare the packed hybrid code with the compiled full LAPACK code, we see that it is always faster, and significantly so for small n. We see this as very encouraging. Furthermore, it is slightly faster than the vendor packed code except for small n. It is sometimes faster than the vendor full code and would have been faster for all $n{\geq}1600$ if we had switched to $nb{=}200$ at $n{=}1600$.

A RAL report will shortly be available that describes this work in more detail.

References

B.S. Andersen, F.G. Gustavson, and J. Waśniewski. A Recursive Formulation of Cholesky Factorization of a Matrix in Packed Storage. *ACM Trans. Math. Softw.*, **27**(2), 214–244, Jun 2001.

J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Softw.*, **16**, 1–17, 1990.

3 Iterative methods

3.1 Backward error analysis and stopping criteria for Krylov space methods (M. Arioli, D Loghin, and A. Wathen)

The finite-element method approximates the weak form of a coercive elliptic partial differential equation defined within a Hilbert space by a linear system of equations

$$\mathbf{Au} = \mathbf{b} \tag{3.1.1}$$

where $\mathbf{A}$ is an $N \times N$ positive definite (not necessarily symmetric) matrix.

We denote by $\mathcal{H}$ the space $\mathbb{R}^N$ with the scalar product based on a symmetric and positive-definite matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$:

$$(\mathbf{x}, \mathbf{y})_{\mathbf{H}} = \mathbf{x}^T\mathbf{H}\mathbf{y}.$$

The Krylov methods approximate iteratively the solution $\mathbf{u}$ of (3.1.1) by $\mathbf{u}^{(k)}$ computed by minimizing either a norm of the residual $\mathbf{b}-\mathbf{A}\mathbf{u}^{(k)}$ or a suitable norm of the error $\mathbf{u}-\mathbf{u}^{(k)}$ on a Krylov space. When using an iterative method, we normally incorporate a stopping criterion based on the a posteriori component-wise or norm-wise backward error theory of Arioli, Duff and Ruiz (1992).

Owing to the special structure of the space $\mathcal{H}$, we must modify the usual backward error analysis taking into account the norm in $\mathcal{H}$. On the basis of the backward error analyses presented in Rigal and Gaches (1967), for the finite-dimensional case, and in Arioli, Noulard and Russo (2001) for a general Banach space, a good candidate stopping criterion could be

$$\text{IF} \ \ \|\mathbf{A}\mathbf{u}^{(k)} - \mathbf{b}\|_{\mathbf{H}^{-1}} \leq \eta \|\mathbf{b}\|_{\mathbf{H}^{-1}} \ \ \text{THEN STOP} \ , \tag{3.1.2}$$

with $\eta < 1$ an a priori threshold fixed by the user.

For the conjugate gradient method, we assume $\mathbf{A}$ is symmetric and positive definite, and that $\mathbf{H} = \mathbf{A}$. Using this equality, it follows that (3.1.2) is equivalent to the stopping criteria:

$$\text{IF} \ \ \|\mathbf{u}^{(k)} - \mathbf{u}\|_{\mathbf{H}} \leq \eta \|\mathbf{u}\|_{\mathbf{H}} \ \ \text{THEN STOP} \ . \tag{3.1.3}$$

We need to add, within the conjugate gradient algorithm, some tool for estimating the value $e_{\mathbf{A}}^{(k)} = (\mathbf{u} - \mathbf{u}^{(k)})^T\mathbf{A}(\mathbf{u} - \mathbf{u}^{(k)}) = \mathbf{r}^{(k)T}\mathbf{A}^{-1}\mathbf{r}^{(k)}$ at each step k and we must also estimate $\mathbf{b}^T\mathbf{A}^{-1}\mathbf{b} = \mathbf{u}^T\mathbf{A}\mathbf{u}$.

Arioli (2003), Meurant (1999*b*), and Strakoš and Tichý (2002) presented several techniques that can be used to estimate $e_{\mathbf{A}}^{(k)}$ and $\mathbf{b}^T\mathbf{A}^{-1}\mathbf{b}$. In particular, the value of $e_{\mathbf{A}}^{(k)}$ can be estimated using the rule of Hestenes and Stiefel (1952). The Hestenes and Stiefel rule computes a lower bound ξ_k for $e_{\mathbf{A}}^{(k)}$ that is equal to the bound computed by the Gauss rule proposed by Golub and Meurant (1997). Moreover, Strakoš and Tichý

(2002) proved that the Hestenes and Stiefel rule is numerically stable when finite-precision arithmetic is used.

Under the assumption that $e_{\mathbf{A}}^{(k+d)} << e_{\mathbf{A}}^{(k)}$, where the integer d denotes a suitable delay, the Hestenes and Stiefel estimate ξ_k can then be computed very cheaply using the information computed during the conjugate gradient method.

Moreover, Arioli (2003) and Meurant (1999*b*) proved that, introducing a preconditioner, the energy norm of the preconditioned problem is equal to $e_{\mathbf{A}}^{(k)}$.

Finally, the choice of η will depend on the properties of the problem that we want to solve, and, in practical cases, η can frequently be much larger than ε, the roundoff unit of the computer's finite-precision arithmetic. Arioli (2003) suggested that a reasonable choice for η, when (3.1.1) is obtained from a finite-element approximation of a two dimensional partial differential equation, could be $\eta \approx h$, where h is the maximum diameter of an element in $\mathcal{T}_h$ of the given mesh. Arioli (2003) has also proved that using this choice for η within (3.1.2), the error between the exact solution of the partial differential equation and the function built using $\mathbf{u}^{(k)}$ and the basis functions of the finite elements used to approximate the problem, measured with the continuous norm, is of order $\mathcal{O}(h)$.

Arioli, Loghin and Wathen (2003) have analysed the non-symmetric case where $\mathbf{A} = \mathbf{H}$ is no longer true. In this case, (3.1.2) cannot be seen as a straightforward relation among the errors and their measure in the norm of $\mathcal{H}$. However, the norm of the symmetric part of A^{-1} and the dual norm are equivalent with constants independent of the dimension of the discrete problem (Arioli et al., 2003).

The stopping criteria (3.1.2) can be then replaced with:

$$\text{IF} \quad \|\mathbf{b} - \mathbf{u}^{(k)}\|_{\mathbf{A}^{-1}} \leq \eta \|\mathbf{u}\|_{\mathbf{H}} \sqrt{C_2} \quad \text{THEN STOP} . \tag{3.1.4}$$

Arioli et al. (2003) proved that choosing $\eta = \mathcal{O}(h)$ within (3.1.4), the error between the exact solution of a partial differential equation and the function built using $\mathbf{u}^{(k)}$ and the basis functions of the finite elements used to approximate the continuous problem, measured with the continuous norm, is of order $\mathcal{O}(h)$.

References

M. Arioli (2003), A stopping criterion for the conjugate gradient algorithm in a finite-element method framework, *Numer. Math. DOI: 10.1007/s00211-003-0500-y*, 1–24.

M. Arioli, I. S. Duff, and D. Ruiz (1992), Stopping criteria for iterative solvers, *SIAM J. Matrix Anal. Appl.* **13**(1), 138–144.

M. Arioli, D. Loghin, and A. Wathen (2003), Stopping criteria for iterations in finite-element methods, Technical Report RAL-TR-2003-009, RAL.

M. Arioli, E. Noulard, and A. Russo (2001), Stopping criteria for iterative methods: Applications to PDE's, *CALCOLO* **38**, 97–112.

G. Golub and G. Meurant (1997), Matrices, moments and quadrature II; how to compute the norm of the error in iterative methods, *BIT* **37**, 687–705.

M. Hestenes and E. Stiefel (1952), Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards* **49**, 409–436.

G. Meurant (1999*b*), Numerical experiments in computing bounds for the norm of the error in the preconditioned conjugate gradient algorithm', *Numerical Algorithms* **22**, 353–365.

J. L. Rigal and J. Gaches (1967), 'On the compatibility of a given solution with the data of a linear system, *J. Assoc. Comput. Mach.* **14**(3), 543–548.

Z. Strakoš and P. Tichý (2002), On error estimation by conjugate gradient method and why it works in finite-precision computations, *Electronic Transactions on Numerical Analysis* **13**, 56–80.

3.2 A Chebyshev-based two-stage iterative method (M. Arioli and D. Ruiz)

Arioli and Ruiz (2002) analyse the solution of large, sparse, and ill-conditioned linear systems of equations,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \ , \tag{3.2.1}$$

by an iterative method based on a two-phase approach. Analogously to direct methods, the method includes a preliminary "factorisation" phase followed by a "cheap" solution phase, both based only on numerical tools that are usually exploited in iterative methods and which, in addition, offer the possibility of keeping the matrix in implicit form only requiring matrix-vector products.

Let $\widehat{\mathbf{A}} = \mathbf{Q}\mathbf{A}$, where $\mathbf{Q}$ is a given preconditioner, be the spd iteration matrix. Unfortunately, several of the most commonly used preconditioners transform the matrix $\mathbf{A}$ to a matrix whose eigenvalues are in a relatively small number of clusters but which is still ill-conditioned.

During the first phase, Arioli and Ruiz (2002) start with an initial set of s randomly generated vectors (s is the block-size), and use Chebyshev polynomials (Hageman and Young, 1981) in $\widehat{\mathbf{A}}$ to "*damp*" within these vectors the eigenfrequencies associated with all the eigenvalues in some predetermined range. They fix a positive number $\lambda_{\min}(\widehat{\mathbf{A}}) < \mu < \lambda_{\max}(\widehat{\mathbf{A}})$, and decide to compute all the eigenvectors associated with all the eigenvalues in the range $[\lambda_{\min}(\widehat{\mathbf{A}}), \mu]$. The computation of $\lambda_{\max}(\widehat{\mathbf{A}})$ is usually not too difficult, and in some cases a sharp upper-bound may be already available through some *a priori* knowledge of the numerical properties of $\widehat{\mathbf{A}}$. If the eigenvalues are well clustered, the number of remaining eigenvalues in the interval $[\lambda_{\min}(\widehat{\mathbf{A}}), \mu]$, with reasonable μ (for example $\lambda_{\max}/100$, or $\lambda_{\max}/10$), should be small compared to the size of the linear system.

The use of Chebyshev polynomials can be interpreted as a *filtering* tool that increases the degree of collinearity with some selected eigenvectors. Then, after "*filtering*" the initial starting vectors, a set of s vectors with eigencomponents below a certain level ε for those eigenvalues in the range $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$, and relatively much bigger eigencomponents linked with the smallest eigenvalues in $\widehat{\mathbf{A}}$, is obtained.

Of course, the *a priori* choice of the block-size s may not correspond to or at least be greater than, the number k of remaining eigenvalues outside the interval $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$.

To overcome the difficulties related to the case $k > s$, Arioli and Ruiz (2002) propose using a Block-Lanczos type of approach to build a Krylov basis starting with these filtered vectors and to stop when appropriate. One of the main drawbacks with a Lanczos/Block-Lanczos algorithm is that it does not maintain the nice property of the filtered vectors, and gradually (and rather quickly, indeed) the Lanczos vectors may again have eigencomponents all about the same level.

To maintain the level of the unwanted eigenfrequencies in the orthonormal block Krylov basis under ε, Arioli and Ruiz (2002) propose to perform, at each Block-Lanczos iteration, a few extra Chebyshev iterations on the newly generated Block Lanczos vectors $\mathbf{V}^{(k+1)}$. In this way, "*near*" invariance can be maintained.

It must be mentioned, beforehand, that the nice property of the block-Krylov spaces, which makes the projected matrix $\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}$ block-tridiagonal, is lost after re-filtering the current block $\mathbf{V}^{(k+1)}$. Therefore, it is necessary to orthogonalize the filtered vectors against all the previously constructed ones at each iteration, as for example in an Orthodir process.

Once the *near*-invariant subspace linked to the smallest eigenvalues is obtained, it can be used for the computation of further solutions. First, one performs an oblique projection of the initial residual ($\hat{\mathbf{r}}_\mathbf{0} = \hat{\mathbf{b}} - \widehat{\mathbf{A}}\mathbf{x}_\mathbf{0}$) onto this *near* invariant subspace in order to get the eigencomponents in the solution corresponding to the smallest eigenvalues. To compute the remaining part of the solution vector $\widehat{\mathbf{A}}\mathbf{x}_\mathbf{2} = \hat{\mathbf{r}}_\mathbf{1}$, one can then use the classical Chebyshev algorithm with eigenvalue bounds given by μ and $\lambda_{\max}(\widehat{\mathbf{A}})$ as explained by Hageman and Young (1981, Chapter 4).

It is also possible to iterate on that solution phase, and improve the solution with iterative refinement in the usual way.

In Arioli and Ruiz (2002) the authors extensively discuss the choice of the filtering level ε, the block size s, and the cut-off parameter μ as a function of the eigenvalue distribution of $\widehat{\mathbf{A}}$.

Table 3.2.1 summarizes the results of the comparison of the number of Chebyshev filtering steps for different values of ε and μ obtained on a numerical test based on the Rutherford-Boeing collection. The rapid change in the Chebyshev rate of convergence with smaller values of μ induces much more Chebyshev filtering steps at each iteration, and this could only be counterbalanced by a very strong reduction in the total number of iterations in the first phase of the algorithm. In other words, it is worth reducing the value of μ only if there is a very strong clustering of eigenvalues in the spectrum of the iteration matrix $\widehat{\mathbf{A}}$ and if the change in μ helps to reduce by a large amount the dimension of the invariant subspace that will be approximated.

	Number of Chebyshev Iterations					
	$\mu = \lambda_{\max}/5$		$\mu = \lambda_{\max}/10$		$\mu = \lambda_{\max}/100$	
Block Krylov Iteration	Value of ε		Value of ε		Value of ε	
	10^{-14}	10^{-8}	10^{-14}	10^{-8}	10^{-14}	10^{-8}
Start	35 + 2	20 + 3	51 + 4	30 + 3	165 + 14	96 + 14
1	10	9	15	15	59	56
2	11	11	20	18	88	90
3	16	15	32	30	165	96
4	25	20	43	30	-	-
5	35	20	-	-	-	-

In the case $\mu = \lambda_{\max}/5$, there are 33 eigenvectors to capture.
In the case $\mu = \lambda_{\max}/10$, there are 26 eigenvectors to capture.
In the case $\mu = \lambda_{\max}/100$, there are 19 eigenvectors to capture.

Table 3.2.1: Comparison of the number of Chebyshev filtering steps for different values of the filtering level and different bounds for the damping interval. (Block Lanczos/Orthodir with block size 6).

References

M. Arioli and D. Ruiz (2002), A Chebyshev-based two-stage iterative method as an alternative to the direct solution of linear systems, Technical Report RAL-TR-2002-021, RAL.

L. A. Hageman and D. M. Young (1981), *Applied Iterative Methods*, Academic Press, New York and London.

3.3 Underground 3-D flow modelling and HSL routines (M. Arioli and G. Manzini)

Underground 3-D flow modelling plays a key role in several physical phenomena and engineering processes including oil reservoir exploitation and underground water remediation. One application is the simulation of the decontamination process around the Stráz pod Ralskem uranium mine in the Czech Republic. Several algorithms are compared on a set of 3-D test problems arising from this problem (Arioli, Maryška, Rozložník and Tůma, 2001).

Darcy's law describes the relationship between the pressure $p(\mathbf{x})$ (the "total head") and the velocity field $\mathbf{u}(\mathbf{x})$ (the "visible effect") in ground-water flow. In particular, Darcy's law relates the vector field $\mathbf{u}$ to the scalar field p via the permeability tensor $K(\mathbf{x})$ which accounts for the soil characteristics, and the divergence of $\mathbf{u}$ to the source-sink term $f(\mathbf{x})$. Let Ω be a simply connected, bounded, polygonal domain in $\mathbb{R}^2$, defined by a closed piecewise linear connected curve $\Gamma = \Gamma_D \cup \Gamma_N$, and $\mathbf{n}$ be the external normal to Γ. Darcy's equations are given by

$$\left\{ \begin{array}{rcll} \mathbf{u}(\mathbf{x}) & = & -K(\mathbf{x}) \operatorname{grad} p(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \operatorname{div} \mathbf{u}(\mathbf{x}) & = & f(\mathbf{x}), & \mathbf{x} \in \Omega, \end{array} \right. \tag{3.3.1}$$

with boundary conditions

$$\begin{array}{rcll} p(\mathbf{x}) & = & g_D(\mathbf{x}), & \mathbf{x} \in \Gamma_D, \\ \mathbf{u} \cdot \mathbf{n} & = & g_N(\mathbf{x}), & \mathbf{x} \in \Gamma_N. \end{array} \tag{3.3.2}$$

The vector field $\mathbf{u}$ and the scalar field p are unknown in the interior part of Ω. In order to solve Darcy's law, mixed finite-element approximation techniques are used. This leads to the solution of an augmented, nonsingular, and sparse system of linear equations

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{b} \end{bmatrix}, \tag{3.3.3}$$

where $\mathbf{M}$ is a $n \times n$ symmetric and positive definite matrix, and $\mathbf{A}^T$ the $m \times n$ full rank divergence matrix with entries equal to either 1, -1, or 0.

The HSL codes `MA47` and `MA57` are two efficient and robust direct packages designed for solving symmetric sparse systems. Table 3.3.1 shows the number of nonzero entries required to store the matrix factors and the CPU times required to solve the augmented system for both `MA57` and `MA47` on test problems related to the Stráz pod Ralskem uranium mine. For very large problems, the augmented systems may involve several million unknowns. In such cases, the memory requirements of direct solvers can be prohibitive. For these systems, it is preferable instead to apply a specialized version of the classical null space algorithm for the minimisation of linearly constrained quadratic forms. This has the advantage of preserving the physical meaning of the computed velocity field $\mathbf{u}$ because it naturally conserves flux. Null space algorithms compute bases of the null space defined by the flow conservation equation. Given such a basis, a reduced linear system in the null space may be solved using the conjugate gradient algorithm, in which required matrix-vector products are computed implicitly.

The HSL package `MA49` can be used to compute an orthogonal null-space basis. In particular, `MA49` avoids the explicit computation of the full orthogonal matrix $\mathbf{H}$, storing the information necessary to compute it as a product of sparse elementary matrices, and thus all the matrix-vector products required by the algorithm can be computed implicitly. Importantly, in this case the number of steps performed by the conjugate gradient method is independent of the mesh size used for the discretization (Arioli and Manzini, 2002). This approach may also be useful and efficient for nonlinear variants of Darcy's equation in which the permeability tensor depends on the velocity field $\mathbf{u}$; in this case, `MA49` is employed only once and the factors are used within a nonlinear conjugate gradient method applied to the problem. However, for very large problems, the storage of the sparse factor matrices necessary to compute the orthogonal null-space basis can again be prohibitive. For this reason, Arioli and Manzini (2001) have proposed a novel approach based on network programming techniques. Arioli and Manzini (2002) analyse the use of shortest path tree algorithms to identify a basis of the null space using simple permutation matrices. Because negligible additional storage is required, the resulting code is extremely competitive for very large problems. In particular, when the test problem has a unit square domain and $K(\mathbf{x})$ has a random distribution with its values ranging from 1 to 10^{-12}, the proposed approach is competitive in terms of CPU time with `MA47`. In particular, on a mesh with 155746 triangles (389874 degrees of freedom) `MA47` requires 35 seconds to solve the problem, `MA57` requires 20 seconds, and our approach requires 36 seconds. When the mesh was refined taking the number of triangles up to 1002499 and the number of degrees of freedom up to 2507711, both `MA47` and `MA57` were unable to solve the system because they required more than 1Gbyte (the size of the RAM memory on the computer used for the experiments) to store the matrix factors. Our approach solved the problem in 732 seconds.

References

M. Arioli and G. Manzini, Null space algorithm and spanning trees in solving Darcy's equation, Technical Report RAL-TR-2002-026, RAL. To appear in BIT-Numerical Mathematics

M. Arioli and G. Manzini. A null space algorithm for mixed finite-element approximations of Darcy's equation. *Commun. Numer. Meth. Engng*, **18**, 2002.

M. Arioli, J. Maryška, M. Rozložník, and M. Tůma. Dual variable methods for mixed-hybrid finite-element approximation of the potential fluid flow problem in porous media. Technical Report RAL-TR-2001-023, Rutherford Appleton Laboratory, Oxfordshire, 2001.

M. Arioli and G. Manzini. A network programming approach in solving Darcy's equations by mixed finite-element methods. Technical Report RAL-TR-2001-037, Rutherford Appleton Laboratory, Oxfordshire, 2001.

3.4 Efficient parallel iterative solvers for the solution of large dense linear systems arising from the boundary element method in electromagnetism (G. Alléon, B. Carpentieri, I. S. Duff, L. Giraud, J. Langou, E. Martin, and G. Sylvand)

In recent years, there has been a significant amount of work on the simulation of electromagnetic wave propagation phenomena, addressing various topics ranging from

Name	$m+n$	`MA47`		`MA57`	
	# nonzeros	# entries	CPU Time	# entries	CPU Time
SIT100	10262 34094	1467237	2.6 sec	821500	0.7 sec
K1_SAN	67759 303364	7104285	14 sec	6567837	9 sec
OLESNIK0	88263 402623	9356219	23 sec	8427827	12 sec
D_PRETOK	182730 885416	40182401	257 sec	33700651	87 sec
TURON_M	189924 912345	42586618	277 sec	31248608	71 sec

Table 3.3.1: Comparison of `MA47` and `MA57`.

radar cross section to electromagnetic compatibility, to absorbing materials, and antenna design. To address these problems the Maxwell equations are often solved in the frequency domain leading to singular integral equations of the first kind. The discretization by the boundary element method (BEM) results in linear systems with dense complex matrices which are very challenging to solve. In this project we propose preconditioning strategies for the iterative solution of these systems.

3.4.1 Combining fast multipole techniques and approximate inverse preconditioners for large calculations in electromagnetism (B. Carpentieri, I. S. Duff, L. Giraud and G. Sylvand)

Our primary focus is on the design of an efficient parallelizable preconditioner for solving the dense complex systems arising in electromagnetic calculations using preconditioned Krylov methods. In that respect, we consider an approximate inverse method based on the Frobenius-norm minimization. The preconditioner is constructed from a sparse approximation of the dense coefficient matrix, and the patterns both for the preconditioner and for the coefficient matrix are computed *a priori* using geometric information from the mesh. We describe how such a preconditioner can be naturally implemented in a parallel code that implements the multipole technique for the matrix-vector product calculation. We investigate the numerical scalability of our preconditioner on realistic industrial test problems and show that it exhibits some limitations on very large problems of size close to one million unknowns. To improve its robustness on those large problems we propose an embedded iterative scheme that combines nested GMRES solvers with different fast multipole computations. We show through extensive numerical experiments that this new scheme is extremely robust at affordable memory and CPU costs for the solution of very large and challenging problems. We show a summary of our results in Table 3.4.1 on a set of standard test problems. These clearly show the benefit of using the embedded scheme. The Compaq Alpha server on which these runs were made is a cluster of Symmetric Multi-Processors, each node of which consists of four DEC Alpha processors (EV 6, 1.3 GFlops peak) that share 512 MB of memory. More details can be found in Carpentieri, Duff, Giraud and Sylvand (2003) from which this table is taken.

3.4.2 Using spectral low rank preconditioners for large electromagnetic calculations (I. S. Duff, L. Giraud, J. Langou, and E. Martin)

For solving large dense complex linear systems that arise in electromagnetic calculations, we perform experiments using a general purpose spectral low rank update preconditioner (Carpentieri, Duff and Giraud, 2003) in the context of the GMRES method preconditioned by an approximate inverse preconditioner. The goal of the spectral preconditioner is to improve the convergence properties by shifting by one the smallest eigenvalues of the original preconditioned system.

Numerical experiments have been performed on parallel distributed memory computers, using a Fast Multipole code (Sylvand, 2002), to illustrate the efficiency of this technique on large and challenging real–life industrial problems. We show the results for a complete monostatic calculation for our four main test problems in Table 3.4.2 where the advantage of using the spectral low rank correction ($M_{SLRU(k)}$) compared to the Frobenius norm preconditioning (M_{Frob}) is clearly seen. More details on this work are available in Duff, Giraud, Langou and Martin (2003) from which this table is taken.

Aircraft				
Size	GMRES(∞)		FGMRES(∞,60)	
	Iter	Time	Iter	Time
94704	746	2h 9m	23+1320	2h 30m
213084	973	7h 19m	30+1740	6h 11m
591900	1461	16h 42m$^{(64)}$	43+2520	12h*
1160124	M.L.E.$^{(64)}$	> 1d	43+2520	14 h 28m**
Cobra				
Size	GMRES(∞)		FGMRES(∞,60)	
	Iter	Time	Iter	Time
60695	369	26m	21+600	17m
179460	353	1h 11m	18+510	38m

Table 3.4.1: Number of matrix-vector products and elapsed time required to converge. The tests were run on 8 processors of the Compaq machine, except those marked with $^{(k)}$, that were run on k processors. GMRES(∞) is the GMRES algorithm without any restarting and FGMRES(∞,60) is the FGMRES algorithm with a restart of 60 in the inner loop but no restarting in the outer.

References

G. Alléon, B. Carpentieri, I. S. Duff, L. Giraud, E. Martin, and G. Sylvand. Efficient parallel iterative solvers for the solution of large dense linear systems arising from the boundary element method in electromagnetism. Technical Report TR/PA/03/65, CERFACS, Toulouse, France, 2003.

			M_{Frob}		$M_{SLRU(k)}$	
Geometry	Order	# procs.	# iter	times	# iter	times
Cetaf	8	5 391	16 391	1 h 40 m	5 349	47 m
Airbus	32	23 676	87 121	46 h	47 385	18h 40 m
Cobra	32	60 695	29 777	21 h	16 921	8h 30 m
Almond	32	104 793	34 375	25 h 30 m	21 273	14h 40 m

Table 3.4.2: Cost for a complete monostatic calculation.

B. Carpentieri, I. S. Duff, and L. Giraud. A class of spectral two-level preconditioners. *SIAM J. Scientific Computing*, **25**(2), 749–765, 2003.

B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand. Combining fast multipole techniques and an approximate inverse preconditioner for large parallel electromagnetics calculations. Technical Report RAL-TR-2003-024, Rutherford Appleton Laboratory, Oxfordshire, England, 2003.

I. S. Duff, L. Giraud, J. Langou, and E. Martin. Using spectral low rank preconditioners for large electromagnetic calculations. Technical Report RAL-TR-2003-023, Rutherford Appleton Laboratory, Oxfordshire, England, 2003.

G. Sylvand. *Résolution Itérative de Formulation Intégrale pour Helmholtz 3D : Applications de la Méthode Multipôle à des Problèmes de Grande Taille.* PhD thesis, Ecole Nationale des Ponts et Chaussées, 2002.

3.5 The Sparse BLAS (I. S. Duff, M. Heroux, R. Pozo, and C. Vömel)

We are delighted to report the release, during this reporting period, of the new Basic Linear Algebra Subprogram (BLAS) Standard developed and defined by the BLAS Technical Forum (2002). This involved many extensions to the earlier standards, including new functionalities, mixed precision BLAS, and sparse BLAS. A technical description of these can be found in Blackford, Demmel, Dongarra, Duff, Hammarling, Henry, Heroux, Kaufman, Lumsdaine, Petitet, Pozo, Remington and Whaley (2002). Our main contribution to this effort was in the design, implementation, and testing of the sparse BLAS.

The design of the sparse BLAS is discussed in the paper by Duff, Heroux and Pozo (2002). This consists of a set of kernels providing basic operations for sparse matrices and vectors, including the multiplication of a dense vector or a set of dense vectors by a sparse matrix. A principal goal of the Sparse BLAS standard is to aid in the development of iterative solvers for large sparse linear systems by specifying interfaces for a high-level description of vector and matrix operations for the algorithm developer while leaving enough freedom for vendors to provide the most efficient implementation of the underlying algorithms for their specific architectures.

The Sparse BLAS standard defines interfaces and bindings for the three target languages: C, Fortran 77 and Fortran 95. Our Fortran 95 implementation is intended as a reference model for the Sparse BLAS. The design is based on the idea of matrix handles so that the user need not be concerned with the details of the underlying storage schemes. The software implementation has been published as a TOMS algorithm (Duff and Vömel, 2002).

References

L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Softw.*, **28**(2), 135–151, 2002.

BLAS Technical Forum. Special Issue: On Basic Linear Algebra Subprograms Technical BLAST Forum Standard -I and II. *The International Journal of High Performance Computing Applications*, **16**(1–2), 2002.

I. S. Duff and C. Vömel. Algorithm 818: a reference model implementation of the Sparse BLAS in Fortran 95. *ACM Trans. Math. Softw.*, **28**(2), 268–283, 2002.

I. S. Duff, M. A. Heroux, and R. Pozo. An overview of the Sparse Basic Linear Algebra Subprograms: the new standard from the BLAS Technical Forum. *ACM Trans. Math. Softw.*, **28**(2), 239–267, 2002.

4 Optimization

4.1 GALAHAD (N. I. M. Gould, D. Orban and Ph. L. Toint)

We have long been known for our nonlinear optimization package, LANCELOT (Conn, Gould and Toint, 1992). Since this was definitely showing its age, we have now replaced this by a library of freely-available Fortran 90 optimization packages. Rather than providing a single nonlinear programming package, we have decided instead to produce a library, GALAHAD (Gould, Orban and Toint, 2003), of optimization-related packages. The first release of the library occurred in April 2002, and key components included

- QPA, an working-set method for finding a critical point of a (nonconvex) quadratic function over a polyhedral region, (see Gould and Toint, 2002),
- QPB, a usually-superior interior-point method for the same problem (see Conn, Gould, Orban and Toint, 2000, and Gould and Toint, 2002*b*),
- LSQP, an interior-point method for minimizing a linear or separable convex quadratic function over a polyhedral region,
- PRESOLVE, a presolve utility for simplifying linear and quadratic programming (see Gould and Toint, 2002*b*),
- LANCELOT B, an updated version of the old package,
- GLTR, a method for minimizing a quadratic function within or on a (scaled) ball (the ℓ_2-trust-region subproblem) based on Gould, Lucidi, Roma and Toint (1999), and
- a variety of sparse-matrix manipulation tools.

A subsequent sub-release (July 2003) added the FILTRANE package for solving nonlinear feasibility problems, described in Section 4.5, to the library.

LANCELOT B offers a number of improvements over its predecessor, but is still far from state-of-the-art. New features include

- the automatic allocation of workspace,
- a non-monotone descent strategy to be used by default (a variety of non-monotone history lengths are possible),
- the optional use of Moré and Toraldo (1991)-type projections during the subproblem solution phase,
- an interface to Lin and Moré's (1999) public domain incomplete Cholesky factorization package ICFS for use as a preconditioner, and
- the optional use of structured trust regions to better model structured problems (see Conn, Gould, Sartenaer and Toint, 1996).

The main reason for extending LANCELOT's life is as a prototype for what may be achieved using Fortran 90 in preparation for future replacement GALAHAD SQP or interior-point solvers, since the problem data structure is unlikely to change. However, as can be seen from Figure 4.1.1, some of the new options mentioned above perform favourably compared to the earlier A version of the code.

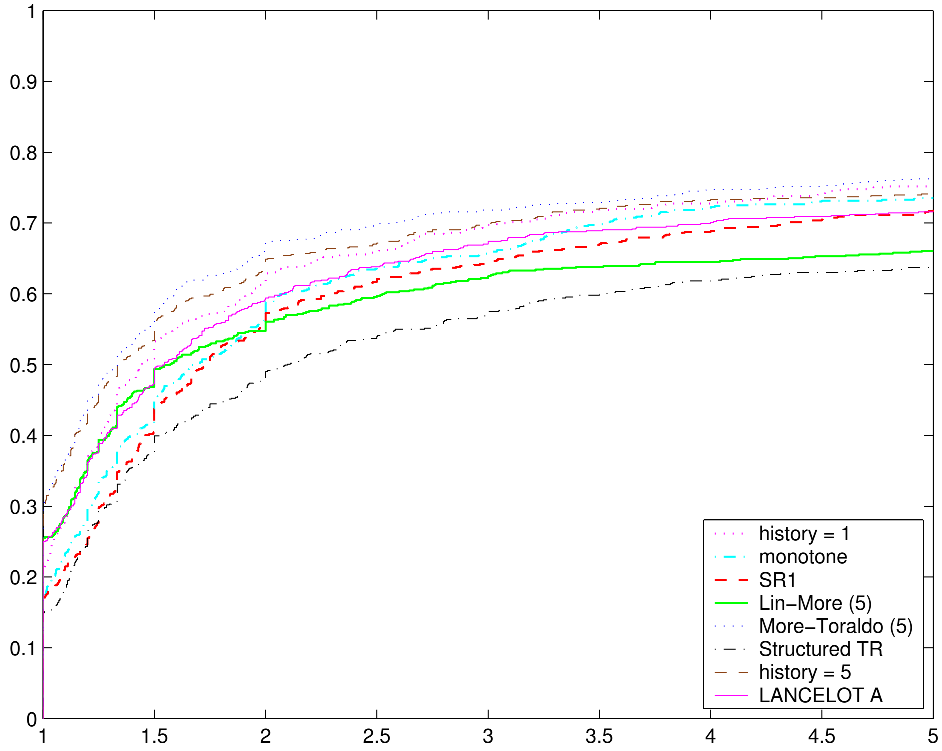

Figure 4.1.1: CPU time performance profile (Dolan and Moré, 2002) for LANCELOT B options. The horizontal axis gives the argument α, while the vertical axis records $p_i(\alpha)$, the percentage of times option i is within a fraction α of the best option on each test problem in the CUTEr test set, for each of the competing options, i.

The GALAHAD library is coded entirely in Fortran 90, and is thread-safe. It is fully documented, and capable of supporting multi-platform, simultaneous use (within a Unix-like environment). GALAHAD is downloadable without-charge from its Web page,

`http://galahad.rl.ac.uk/galahad-www` .

The full scope of the package(s) is described in Gould et al. (2003). GALAHAD has been downloaded roughly 200 times since its release in April 2002.

References

A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, **87**(2), 215–249, 2000.

A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint. Convergence properties of minimization algorithms for convex constraints using a structured trust region. *SIAM Journal on Optimization*, **6**(4), 1059–1086, 1996.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for Large-scale Nonlinear Optimization (Release A).* Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.

E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, **91**(2), 201–213, 2002.

N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, **9**(2), 504–525, 1999.

N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software*, **29**(4), 352–372, 2003.

N. I. M. Gould and Ph. L. Toint. An iterative working-set method for large-scale non-convex quadratic programming. *Applied Numerical Mathematics* **43** (1–2), 109–128 (2002).

N. I. M. Gould and Ph. L. Toint. Preprocessing for quadratic programming. Technical Report RAL-TR-2002-001, Rutherford Appleton Laboratory, Oxfordshire, 2002b. To appear, *Mathematical Programming.*

N. I. M. Gould and Ph. L. Toint. Numerical methods for large-scale non-convex quadratic programming. *In* 'Trends in Industrial and Applied Mathematics'. (A. H. Siddiqi and M. Kočvara, eds) Kluwer Academic Publishers, 149–179 2002

C. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, **9**(4), 1100–1127, 1999.

J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, **1**(1), 93–113, 1991.

4.2 CUTEr, an Optimization Testing Environment (N. I. M. Gould, D. Orban and Ph. L. Toint)

Our other freely-available optimization package CUTE (see Bongartz, Conn, Gould and Toint, 1995) metamorphosed into CUTEr during 2002. We gave a full synopsis of the scope and design of CUTEr in the last progress report, and merely summarise recent developments here.

CUTEr provides subroutine-level interfaces between a large collection of optimization test problems and a variety of optimization solvers, both ours and others. This new release is characterized by

- a set of new tools, including a unified facility to report the performance of the various optimization packages being tested,
- a set of new interfaces to additional optimization packages,
- extra, larger test examples,
- some Fortran 90/95 and C/C++ support, and
- AMPL and Matlab interfaces.

In addition the package has the following features;

- a complete redesign of organization of the various files that make up the environment, now allowing concurrent installations on a single machine and shared installations on a network, and
- a new simplified and automated installation procedure, but
- the restriction of the environment to UNIX systems.

CUTEr is freely downloadable from its Web page,

`http://cuter.rl.ac.uk/cuter-www` .

The full scope of the package(s) is described in Gould, Orban and Toint (2003).

References

I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and Unconstrained Testing Environment. *ACM Transactions on Mathematical Software*, **21**(1), 123–160, 1995.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, **29**(4), 373–394, 2003.

4.3 An interior-point ℓ_1-penalty method for nonlinear optimization (N. I. M. Gould, D. Orban and Ph. L. Toint)

While an updated version of LANCELOT appears in GALAHAD, our longer-term goal is to replace this by a more powerful interior-point or SQP approach. We have reservations about the applicability of the latter for large-scale problems (see Gould, 2003), but are very enthusiastic about the former.

A typical nonlinear programming problem is to

$$\underset{x\in\mathbf{R}^n}{\text{minimize}} \quad f(x) \tag{4.3.1a}$$

$$\text{such that} \quad c_{\mathcal{E}}(x) = 0 \;\text{ and }\; c_{\mathcal{I}}(x) \geq 0, \tag{4.3.1b}$$

involving a mixture of smooth, general, possibly nonlinear and nonconvex, equality and inequality constraints. Although we had originally thought that the basic Sℓ_1QP approach would be easy to develop once we had good quadratic programming codes—these had arisen as part of the initial release of GALAHAD, see Section 4.1—we soon became more sceptical once we appreciated the difficulties caused by the convergence of the latter to local rather than global solutions. We thus rapidly changed tack, and now consider the alternative described in this section to be more promising.

Our main idea is to embed (4.3.1) into a higher dimensional subspace such that its feasible set has a non-empty interior, allowing the use of interior-point methods for its numerical solution. The reformulation that we use is to

$$\begin{array}{ll} \underset{x,s}{\text{minimize}} & \phi\mathbf{s}S(x,s;\nu) \overset{\text{def}}{=} f(x) + \nu\displaystyle\sum_{i\in\mathcal{E}}[c_i(x)+2s_i] + \nu\displaystyle\sum_{i\in\mathcal{I}} s_i \\ \text{subject to} & c_i(x)+s_i \geq 0 \;\text{ and }\; s_i \geq 0, \;\text{ for all }\; i \in \mathcal{E}\cup\mathcal{I}, \end{array} \tag{4.3.2}$$

involving so-called “elastic” variables s and a penalty parameter ν. This problem has only *inequality* constraints, and it is trivial to pick s sufficiently large so that (x,s) is strictly feasible for (4.3.2), which is a distinguishing advantage of the formulation. A further advantage is that the reformulation (4.3.2) is surprisingly regular in the sense that the well known Mangasarian-Fromowitz constraint qualification condition holds at feasible pairs (x,s), and not just at local solutions. We also note that the reformulation relaxes the shape of the constraints, promoting larger steps and easing the nonlinearity of the strictly feasible set in the neighbourhood of a solution—other reformulations have been suggested by Armand (2002), Mayne and Polak (1976) and Tits, Wächter, Bakhtiari, Urban, and Lawrence (2002).

Problem (4.3.2) may then be tackled using interior-point techniques, (approximately) minimizing a sequence of *logarithmic barrier* functions

$$\phi\mathbf{s}S(x,s;\nu) - \mu\sum_{i\in\mathcal{E}\cup\mathcal{I}}\log(c_i(x)+s_i) - \mu\sum_{i\in\mathcal{E}\cup\mathcal{I}}\log s_i,$$

for a decreasing sequence $\{\mu_k\}$ of positive barrier parameters whose limit is zero and a possibly increasing sequence $\{\nu_k\}$ of positive penalty parameters. Exactness of the penalty function eliminates the need to drive this last sequence to infinity. Global and fast local convergence of the proposed scheme have been established, based on previous analysis of interior-point methods for nonconvex problems (see, Conn, Gould, Orban and Toint, 2000, Gould, Orban, Sartenaer and Toint, 2001, and Gould, Orban, Sartenaer and Toint, 2002). Further improvements of the resulting algorithm include making the elastic variables s implicit and other numerically helpful heuristics. Theoretical aspects of this approach are fully described in Gould, Orban and Toint (2003c), although this work is still on-going and the resulting code is not yet ready for GALAHAD.

References

P. Armand. A quasi-Newton penalty barrier method for convex minimization problems. Technical Report 2002-01, Laboratoire d'Arithmétique, de Calcul Formel et d'Optimisation, Université de Limoges, Limoges, France, 2002.

A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, **87**(2), 215–249, 2000.

N. I. M. Gould. Some reflections on the current state of active-set and interior-point methods for constrained optimization. *SIAG/Optimization Views-and-News*, **14**(1), 2–7, 2003.

N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. *SIAM Journal on Optimization*, 11(4):974–1002, 2001.

N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Componentwise fast convergence in solving nonlinear equations. *Mathematical Programming, Series B*, 92(3):481–508, 2002.

N. I. M. Gould, D. Orban, and Ph. L. Toint. An interior-point ℓ_1-penalty method for nonlinear optimization. Technical Report RAL-TR-2003-022, Rutherford Appleton Laboratory, Oxfordshire, 2003.

D. Q. Mayne and E. Polak. Feasible directions algorithms for optimisation problems with equality and inequality constraints. *Mathematical Programming*, 11(1):67–80, 1976.

A. L. Tits, A. Wächter, S. Bakhtiari, T. J. Urban, and C. T. Lawrence. A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. Technical Report TR 2002-29, Institute for Systems Research, University of Maryland, College Park, USA, 2002.

4.4 Filter Methods (N. I. M. Gould and Ph. L. Toint)

Another class of modern techniques for nonlinear optimization are the so-called "filter" methods, Such methods were first proposed by Fletcher and Leyffer (2002) as a means of assessing the suitability of steps computed by sequential-quadratic-programming (SQP) methods. Their primary aim is to avoid the use of merit functions, since it is far from obvious how best to combine the objective and constraints. Filter methods instead treat the objective and constraints as independent objects, and essentially assess the suitability of an SQP step by rejecting it only if neither the objective nor constraint violation improves following the step. Although a general purpose SQP Filter method is necessarily far more complicated than this simple idea, there is strong evidence that the approach is worthwhile, and offers more flexibility than other merit-function based approaches.

Unfortunately, convergence of the basic SQP Filter method depends upon being able to solve the step-finding quadratic programming (QP) subproblem. Since in general this is a non-convex optimization problem, it is unreasonable in practice to hope to be able to do so in every case. Thus our research was based on alternatives that do not require the exact solution of the QP subproblem.

One way to do this is to relax the requirements on the step, but to insist that the step is constructed as the sum of two components, one of which aims towards (linearized) feasibility, and the other towards objective-function decrease. Both components have to be chosen to ensure "Cauchy-like" decrease conditions so familiar in trust-region methods, but fortunately there are good methods to guarantee this. The global convergence of just such a scheme was established in Fletcher, Gould, Leyffer, Toint and Wächter (2002).

An alternative in which the SQP step is attempted first, but in which the Fletcher et al. (2002) method is used as a fall-back is also possible, and has been analysed by Gould and Toint (2001). Another possibility is to weaken further the requirement that the filter points be themselves monotonically improving so long as there is an overall monotonic trend. Again such a framework is amenable to global convergence analysis (see, Gould and Toint, 2001).

References

R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, **91**(2), 239–269, 2002.

R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Waechter. Global convergence of a trust-region SQP-filter algorithm for nonlinear programming. *SIAM Journal on Optimization*, 13(3):635–659, 2002.

N. I. M. Gould and Ph. L. Toint. Global convergence of a hybrid trust-region SQP-filter algorithm for general nonlinear programming. Technical Report RAL-TR-2001-033, Rutherford Appleton Laboratory, Oxfordshire, 2001.

N. I. M. Gould and Ph. L. Toint. Global convergence of a hybrid trust-region SQP-filter algorithm for general nonlinear programming. Technical Report RAL-TR-2003-003, Rutherford Appleton Laboratory, Oxfordshire, 2003.

4.5 Nonlinear feasibility problems and FILTRANE (N. I. M. Gould, S. Leyffer and Ph. L. Toint)

In this project, we considered the solution of the general smooth feasibility problem, that is the problem of finding a vector x satisfying the constraints

$$c_{\mathcal{E}}(x) = 0 \;\text{ and }\; c_{\mathcal{I}}(x) \geq 0.$$

If such a point cannot be found, it is at least desirable to find a local minimizer of the constraint violation. We choose here to consider the Euclidean norm of these violations, that is to find a local minimizer of the function $\min_x f(x) = \frac{1}{2}\|\theta(x)\|^2$, where we define

$$\theta(x) \stackrel{\rm def}{=} \left(\begin{array}{c} c_{\mathcal{E}}(x) \\ [c_{\mathcal{I}}(x)]_- \end{array} \right),$$

with $[c_{\mathcal{I}}(x)]_- = \min[0, c_{\mathcal{I}}(x)]$, the minimum being taken componentwise. An important special case of this problem is when $\mathcal{I} = \emptyset$, which reduces to a system of smooth nonlinear equations. This problem is therefore not only fairly general, but also practically important because a large number of applications can be cast in this form. Moreover, solving the feasibility problem may also occur as a subproblem within more complicated contexts, such as the "restoration" phase in the solution of the nonlinear programming problem using filter methods (see, for example, Fletcher and Leyffer, 2002).

For simplicity of exposition, we assume that the problem only contains nonlinear equations. In this case, we may build two distinct local quadratic models of $f(x)$ in the neighbourhood of a given iterate x_k: the Gauss-Newton model and the full second-order Newton model which includes an additional term involving the curvature of the equality constraints.

In our method, we have chosen to compute the step s_k by minimizing one of these models in some region surrounding the current iterate x_k, defined by the constraint $\|s_k\| \le \tau_k \Delta_k$, where Δ_k is a trust-region radius which is updated in the usual trust-region manner. The parameter $\tau_k \ge 1$ allows for steps that potentially extend much beyond the limit of the trust region itself, in the case where convergence seems satisfactory. The solution of the subproblem of minimizing the model subject to the trust-region constraint is computed approximately using the Generalized Lanczos Trust-Region method (Gould, Lucidi, Roma and Toint, 1999) as implemented in the GLTR module of GALAHAD. Once the step s_k has been computed, we define the *trial point* $x_k^+ = x_k + s_k$ and consider the question of deciding whether or not it is acceptable as our next iterate x_{k+1}. The filter is used to answer this question. In order to define what we mean, we first say that a point x_1 *dominates* a point x_2 whenever $|\theta_i(x_1)| \le |\theta_i(x_2)| \;$ for all $\; i \in \mathcal{E}$. Thus, if iterate x_{k_1} dominates iterate x_{k_2}, the latter is of no real interest to us since x_{k_1} is at least as good as x_{k_2} for each i. All we need to do now is to remember iterates that are not dominated by other iterates using a structure called a filter. A *filter* is a list $\mathcal{F}$ of m-tuples of the form $(|\theta_{1,k}|, \ldots, |\theta_{m,k}|)$ such that, broadly speaking, for $k \ne \ell$, $|\theta_{i,k}| < |\theta_{i,\ell}|$ for at least one i. Filter methods then accept the new trial iterate x_k^+ if it is not dominated by any other iterate in the filter. In order to avoid cycling, and assuming the trial point is acceptable in that sense, we may wish to add it to the filter, so as to avoid other iterates that are worse. This may however cause an existing filter value θ_ℓ to be dominated. If this happens, we simplify later comparisons by removing θ_ℓ from the filter.

If the trial point is not acceptable for the filter, it may nevertheless be acceptable for the usual trust-region mechanism. Our algorithm therefore attempts to combine the filter and trust-region acceptability criteria to allow a potentially larger set of trial points to be accepted. Overall the method appears to be both theoretically sound and practically an improvement over existing methods. See Gould, Leyffer and Toint (2003) and Gould and Toint (2003b, 2003c) for more details.

The resulting Fortran 90 module FILTRANE (see Gould and Toint, 2003) was added to GALAHAD in June 2003.

References

R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, **91**(2), 239–269, 2002.

N. I. M. Gould, S. Leyffer and Ph. L. Toint. A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares. Technical Report RAL-TR-2003-004, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2003. To appear, *SIAM Journal on Optimization.*

N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

N. I. M. Gould and Ph. L. Toint. FILTRANE, a Fortran 95 filter-trust-region package for solving systems of nonlinear equalities, nonlinear inequalities and nonlinear least-squares problems. Technical Report RAL-TR-2003-017, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2003.

N. I. M. Gould and Ph. L. Toint. The filter idea and its application to the nonlinear feasibility problem. In D. Griffiths and A. Watson, editors, *Proceedings 20th Biennial Conference on Numerical Analysis*, 73-79, University of Dundee, Scotland, 2003b.

N. I. M. Gould and Ph. L. Toint. How mature is nonlinear optimization? In J. Hall, editor, *ICIAM 2003 Proceedings, Sydney*, (to appear), Philadelphia, USA, 2003. SIAM.

4.6 The asymptotic convergence of interior-point and other related methods (N. I. M. Gould, D. Orban, A. Sartenaer and Ph. L. Toint)

While the global convergence of a class of interior-point methods that includes QPB from GALAHAD had already been established by Conn et al. (2000), its asymptotic convergence behaviour was unknown, although presumed, from practical experience, to be fast.

In Gould et al. (2001), we formally established the superlinear convergence of the iterates to a critical point of the underlying nonlinear program under standard assumptions. Most interestingly, not only was it possible to show a normwise superlinear (and effectively quadratic) rate, but that the same rate was almost always achievable for each primal-dual component of the solution. This resolved a long standing open question as to whether there were methods for which this were possible, especially since this is not always the case for standard SQP methods. The result is of some importance since in some applications only a subset of the minimization variables are of interest, and it might be wasteful to wait for the convergence of the uninteresting ones.

While we originally derived these results for interior-point methods, they actually apply in much more general contexts. For example, suppose we wish to solve a system of nonlinear equations

$$F(x) = 0 \tag{4.6.1}$$

by instead solving a related parameterized system

$$F(x) = u_k \tag{4.6.2}$$

for a sequence of vectors $\{u_k\}$ converging to zero—the particular case of interior-point methods is when F represents the first-order optimality (KKT) conditions and each component $(u_k)_i = \mu_k$ for some sequence of decreasing barrier parameters $\{\mu_k\}$.

Suppose that x_k is an approximate root of (4.6.2) in the sense that $F(x_k) - u_k$ is "reasonably small" relative to u_k. In this case, under reasonable assumptions on the derivatives of F, if $\{u_k\}$ is reduced superlinearly to zero, a single Newton step from x_k will result in a point x_{k+1} for which $F(x_{k+1}) - u_{k+1}$ is also "reasonably small". Thus reducing u_k in this way, pulls the iterates towards a root of (4.6.1), and the fact that we are using Newton's method—albeit on (4.6.2) rather than (4.6.1)—leads to the superlinear convergence of the $\{x_k\}$. Moreover the convergence is actually componentwise since the iterates approach their limit along a well defined trajectory rather than "randomly" as would result if Newton's method were applied directly to (4.6.1). Such an approach is made rigorous by Gould et al. (2002).

References

A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, **87**(2), 215–249, 2000.

N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. *SIAM Journal on Optimization*, 11(4):974–1002, 2001.

N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Componentwise fast convergence in solving nonlinear equations. *Mathematical Programming, Series B*, 92(3):481–508, 2002.

4.7 Sequential linear-quadratic programming for huge optimization problems (R. H. Byrd, N. I. M. Gould, J. Nocedal and R. Waltz)

While conventional SQP and interior-point methods have proved to be most effective at solving medium- to large-scale nonlinear optimization problems—say those involving up to 10^5 variables and/or constraints—it is difficult to imagine at this stage how they will generally cope with huge examples (say with 10^7 or more variables/constraints). The difficulty is simply that a (non-convex) quadratic program is too complicated a subproblem for huge problems. Such a restriction is not present for linear programs (LPs) of the same size, problems involving 10^7 unknowns are routinely solved by both simplex and interior-point methods. While this suggests that a more naive sequential linear programming (SLP) approach may be possible, SLP is notoriously slow.

We are most attracted to an idea of Fletcher and Sainz de la Maza (1989) in which the SLP iteration is accelerated by solving an equality-constrained quadratic program (EQP) whose constraints are determined from a prediction of the optimal active ones from the LP. The advantage is that the combinatorially "hard" QP is replaced by the easier LP-EQP combination. While this is a simple and elegant idea, this begs such questions as what to do if the LP is infeasible or unbounded, or if the same is true for the EQP.

We have recently been considering just such an SLP-EQP approach in which a linear approximation to an ℓ_1-exact penalty function is minimized within a (polyhedral) trust-region, and in which the resulting EQP is approximately minimized within a second (elliptical) trust-region. The advantages here are that both subproblems are feasible, and both have bounded solutions. The "art" is in devising a mechanism to adjust the two trust-region radii to ensure both global and fast local convergence.

As a first stage, we have developed a general two-trust-region framework for which we have established global convergence (see Byrd, Gould, Nocedal and Waltz, 2003). Our next goal is to investigate the local convergence of such methods, and we are heartened by numerical experience (see Byrd, Gould, Nocedal and Waltz, 2002) which suggests that fast local convergence is possible with just such a framework.

Full details of the implementation of our software package **SLIQUE**, along with comparative tests on the CUTEr test set are given by Byrd et al. (2002). Although there are still many outstanding practical issues, such as how to manage the penalty parameter in the ℓ_1-penalty function, how to solve and truncate the EQP, and even how accurately to solve the LP, we believe that our approach will be capable of solving huge problems. Since we believe that collaboration is the best way to tackle huge problems, the project also acts as a friendly and useful bridge between the competing GALAHAD and KNITRO software development teams.

References

R. H.. Byrd, N. I. M. Gould, J. Nocedal and R. Waltz. An active set algorithm for nonlinear programming using linear programming and equality constrained subproblems. Technical Report RAL-TR-2002-032, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2002. To appear, *Mathematical Programming*.

R. H.. Byrd, N. I. M. Gould, J. Nocedal and R. Waltz. On the convergence of successive linear programming algorithms. Technical Report RAL-TR-2003-013, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2003.

R. Fletcher and E. Sainz de la Maza. Nonlinear programming and nonsmooth optimization by successive linear programming. *Mathematical Programming*, 43(3):235-256, 1989.

5 Miscellaneous Activities

5.1 The computing environment within the Group (N. I. M. Gould)

Our policy of upgrading the Group's workstations has continued over the past two years. Although the Group's "high-performance" machine remains a Compaq Alpha DS20 dual EV6-processor server, we have taken the opportunity to replace all of our aging Sun equipment by Linux-based Dell PCs. In particular, a dual-processor Precision 460 with 4Gbytes of RAM now acts both as our Web, ftp and file server and as a general-purpose machine for running larger jobs. Group members each have Dimension 2650s with large disks for local work. Our dependence on BITD central file servers has been severed, and we now back up files automatically each night to the RAL E-science tape store. The transition from Solaris to Linux has been, in the main, straightforward, although there have been hardware teething troubles with both the fileserver and one of the smaller machines. We plan to keep at least one of our old Suns simply so that we can verify HSL codes on a Sun compiler and operating systems.

We still benefit from other public CCLRC machines, in particular the HPCx IBM Regatta multiprocessor system. We continue to have access to a number of Fortran 95 compilers, mostly on our own machines. We have also made use of MPI, and associated parallel language support systems, on both our own machines and on those provided by HPCx. In combination with our Grant application, we obtained computing time on other national facilities, specifically the CSAR machines (SGI Origin 2000 and CRAY T3E) at Manchester.

5.2 ERCIM (M. Arioli)

As stated in each ERCIM News, The European Research Consortium for Informatics and Mathematics (ERCIM) is an organisation dedicated to the advancement of European research and development, in information technology and applied mathematics. Its national member institutions aim to foster collaborative work within the European research community and to increase co-operation with European industry.

ERCIM started in 1989 with the three Laboratories CWI (Amsterdam), GMD (Germany), and INRIA (France) and were joined by RAL in the following year. There are now members from 13 countries in the EU and Eastern Europe.

In the early days there were quite active groups and ERCIM meetings in mathematics, but we have been concerned in recent years that the M of ERCIM was becoming neglected. Thus there were two initiatives launched in 2001 to rectify this.

One was an ERCIM Working Group on Numerical Linear Algebra and Statistics coordinated by Erricos Kontoghiorghes (Université de Neuchatel, Switzerland) and Bernard Philippe (IRISA, France) to which both Mario and Iain belong and which has had two meetings in 2001, one of which was attended by Mario.

Nation	Organizations
Austria	University of Salzburg
Belgium	KU Leuven, University of Namur
Czech Rep.	ICS-AS CR
Denmark	Royal Veterinary and Agricultural University
France	INRIA/IRISA, CERFACS, INPT/IRIT
Germany	University of Erfurt, University of Dortmund
Greece	University of Patras
Italy	IMATI-CNR, IMATI-CNR,IAC-CNR, University La Sapienza Roma
Luxembourg	Centre Henri Tudor
Norway	University of Bergen
Slovakia	Slovak Academy of Sciences
Sweden	Linkoping University
Switzerland	ETH Zurich, University of Basel
Cyprus	University of Cyprus
UK	RAL and Universities of Cardiff, Manchester, Oxford, and Strathclyde

Table 5.2.1: NUMAS Participants (administrative manager is ERCIM)

The other ERCIM Group is coordinated by Mario and has a wider mathematical remit involving several ERCIM institutions interested in Applications of Numerical Mathematics in Science (ANMS). Although we anticipate that many application areas will benefit from the results and activities of the working group, it will focus on the following four areas:

- Numerical Linear Algebra.

- Numerical Solution of Differential Equations.

- Continuous Optimization and Optimal Control.

- Large Scale Scientific Computing.

As chairman of the ANMS working group, Mario co-ordinates its activities for the ERCIM fellowship programme (advertisement on the web, selection of the topics), he maintains the home page of the group and stimulates the participants to propose common projects within the European Framework Programme 6 (FP6). In this respect, his role as chairman enhances the European visibility of CCLRC and improves relations with other numerical analysis groups in other ERCIM organizations. This activity and the possibility in the close future of hosting ERCIM fellowships could be beneficial to the HPCx programme run by the Computational Science and Engineering Department in terms of its European image and could attract more co-operation between European partners and CSE groups in the field of computational science.

Within this framework, the ANMS working group asked Mario to be the co-ordinator and person responsible for the Marie-Curie RTN proposal entitled Applications of Computational Mathematics and Statistics in Science and Technology (NUMAS). This proposal aims at extending the activities of the ANMS working group and of the Matrix Computations and Statistics (MCS) ERCIM working group, to other British and European Universities and Institutions not participating in ERCIM.

The several European institutions that are participating in this proposal are given in Table 5.2.1.

The web site for the Working Group is
`http://www.numerical.rl.ac.uk/ercim/WGanms.html`

5.3 CERFACS (I. S. Duff)

Iain has continued to lead a project at CERFACS on Parallel Algorithms and several of the contributions to this report reflect interactions with that team.

The main areas of research in the Parallel Algorithms Project are the development and tuning of kernels for numerical linear algebra, the solution of sparse systems using direct methods or iterative methods or a combination of the two, heterogeneous computing including the use of PVM and MPI, large least-squares calculations with applications to data assimilation, large eigensystem calculations, optimization, and the reliability of computations. Other activities of the Project include advanced training by both courses and research. Two short international meetings were hosted by the Parallel Algorithms Project during the period of this report and were attended by members of the Group.

The home page for CERFACS is `http://www.cerfacs.fr` and current information on the Parallel Algorithms Project can be found on page `http://www.cerfacs.fr/algor/`. Full details on the activities of the Parallel Algorithms Team for the last two years can be found in the reports referenced below.

References

The Parallel Algorithms Project. Scientific Report for 2002. Technical Report TR/PA/02/124, CERFACS, Toulouse, France, 2002.

The Parallel Algorithms Project. Scientific Report for 2003. Technical Report TR/PA/03/135, CERFACS, Toulouse, France, 2003.

5.4 The Grid-TLSE project (I. S. Duff)

In the context of large sparse calculations, the Parallel Algorithms Group at CERFACS is involved as one of the leading partners of a ACI-Grid project (funded by the French Ministry of Research from December 2002 until November 2005). The main scientists involved in this project are: P. Amestoy (ENSEEIHT-IRIT), M. Buvry (ENSEEIHT-IRIT), M. Daydé (ENSEEIHT-IRIT) who is the project coordinator, I.S. Duff, L. Giraud

(CERFACS), J.Y. L'Excellent (INRIA-ENSL), M. Pantel (ENSEEIHT-IRIT), and C. Puglisi (ENSEEIHT-IRIT).

This project will use the grid at several levels and is called Grid-TLSE, for Test for Large Systems of Equations. It will add new services to the Grid and use the grid capabilities to implement these services. The main services will be mainly to:

- provide the users with automatic expertise on sparse direct solvers using matrices either from the data base or provided by the user (a natural follow up step will be to extend this to iterative solvers).

- make available to the scientific community a set of test problems through a data base. The set of examples will grow dynamically as users submit new problems that are integrated within the data set.

A prototype was developed in 2003 and was made available to the industrial partners that are the end-users (CEA, CNES, EADS, EDF, IFP). The specification phase is still ongoing, it is conducted by CERFACS and ENSEEIHT and involves other academic Labs (LABRI, Bordeaux; INRIA-ENS, Lyon) as well as industrial partners. More information on the project can be found from the URL `http://www.enseeiht.fr/lima/tlse`

An agreement is being prepared between CCLRC and this project so that HSL routines will form the backbone of the direct codes that can be accessed and experimented on within this project.

6 HSL (Harwell Subroutine Library) (J. K. Reid)

6.1 Collaboration with Hyprotech and AspenTech

The collaboration with Lawrence Daniels, Iain Strachan and Pascale Hicklin was stable and happy throughout the period, though in the meantime their company was acquired by AspenTech Inc., so the time has been stressful for them.

The fixed annual fee for rights to incorporate HSL software in their packages continues and this is enough to support John Reid's consultancy.

In collaboration with our former colleague Alan Curtis, Lawrence has been working on a much improved version of the solver for stiff ordinary differential equations or differential algebraic equations. More details are given in Section 6.3.1.

6.2 HSL 2002 and HSL Archive

HSL 2002 and the accompanying HSL Archive was released at the start of the period and remained current through the rest of it.

HSL 2002 is our fully-supported main library. While this is a commercial product, it is also available without charge to UK academics for teaching and academic research purposes. This is a direct result of much of our core funding being provided by grants from the Engineering and Physical Science Research Council (R46441 and S42170).

The HSL Archive comprises older packages that were part of previous releases of HSL, many of which have been superseded by more modern codes. The HSL Archive packages are not completely frozen since we aim to correct any errors that come to light, but it is not our intention to develop any of its packages further. The split allows us to focus our attention in the packages of HSL 2002.

Access to both HSL 2002 and the HSL Archive is available through the HSL web page

`http://www.cse.clrc.ac.uk/Activity/HSL`

6.3 HSL 2004

The next release of HSL is planned for Summer 2004 and will be called HSL 2004. Significant work on the following new packages was performed during the reporting period.

6.3.1 HSL_DC05 (Lawrence Daniels)

`HSL_DC05` is a suite of Fortran 95 procedures for the solution of a system of ordinary differential equations (ODE) or differential algebraic equations (DAE) of index 1:

$$\mathbf{F}(t, \mathbf{Y}, \mathbf{Y}') = \mathbf{0}, \tag{6.3.1}$$

where $\mathbf{Y}' = \mathrm{d}\mathbf{Y}/\mathrm{d}t$. It provides a powerful optional method of avoiding the incorrect occurrence of negative values for solution components that should remain positive throughout. Some of the components of $\mathbf{F}$ and $\mathbf{Y}$ are purely algebraic. If there are N_{alg} such algebraic equations and variables, we can split $\mathbf{Y}$ into two vectors: $\mathbf{Y}_1$ of length N_{alg} and $\mathbf{Y}_2$ of length $(N_{\text{eq}} - N_{\text{alg}})$, where N_{eq} is the total number of equations (algebraic and differential) in equation (6.3.1). In turn, equation (6.3.1) can be written as:

$$\mathbf{F}_1(t, \mathbf{Y}_1, \mathbf{Y}_2) = \mathbf{0}, \tag{6.3.2}$$

$$\mathbf{F}_2(t, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_2') = \mathbf{0}, \tag{6.3.3}$$

where $\mathbf{F}_1$ and $\mathbf{F}_2$ are vectors of length N_{alg} and $(N_{\text{eq}} - N_{\text{alg}})$, respectively. Components $\mathbf{Y}_1$ of $\mathbf{Y}$ are purely algebraic, their derivatives $\mathbf{Y}_1'$ being absent from equations (6.3.2) and (6.3.3). The system must be of index no higher than 1, i.e. it must be nonsingular in the sense that equation (6.3.2) can in principle be solved for $\mathbf{Y}_1$ and equation (6.3.3) for $\mathbf{Y}_2'$. `HSL_DC05` cannot solve DAE systems of index greater than 1, which present added difficulties not present in those of index 1.

The solution starts from given initial values of $\mathbf{Y}_2$ at $t = T$, using a variable order Backward Differentiation Formula (BDF) method, also known as Gear's method (Gear, 1970) of order 1 to 5. The method automatically chooses step-size and order of integration formulae, and is especially efficient on stiff systems in particular those arising from mass action kinetics. `HSL_DC05` has been designed to handle DAE problems, which can be regarded in some sense as limiting cases of ODE problems of infinite stiffness. The methods used in `HSL_DC05` are not too inefficient to be acceptable on many non-stiff problems. Function evaluation and linear algebra are carried out in the calling program by using reverse communication, which gives the user full flexibility over access to the problem data and choice of method for solving the linear equations.

6.3.2 MA51 extension for the determinant of a sparse matrix (John Reid)

`MA51` is an auxiliary package for use following a call to `MA48` or `MA50` that performed an LU factorization of a general sparse matrix. It was originally designed for the singular or rectangular case to identify which rows and which columns are treated specially. It may now also be asked to compute the determinant of a square matrix. It finds the parity of the permutations that have been applied and hence the sign of the determinant. To avoid overflow or underflow, the logarithm of the determinant is found as the sum of the logarithms of the absolute values of the diagonal entries of the triangular factors.

6.3.3 MC67 Hager's algorithms for reducing the profile of a symmetric matrix (John Reid and Jennifer Scott)

Given the sparsity pattern of an $n \times n$ symmetric matrix $\mathbf{A}$ and a symmetric permutation that reduces the profile of $\mathbf{A}$, this routine computes a new symmetric permutation with a smaller profile.

The down exchange algorithm of Hager (2002) involves cyclic permutations that correspond to the successive exchange of rows $(k, k+1)$, $(k+1, k+2)$, ..., $(l-1, l)$

of the permuted matrix and interchanging the corresponding columns. For a given k, the value of l that most reduces the profile is found. A pass over the matrix with k taking the values $n-1$, $n-2$, ..., 1 is performed; l is calculated for each k and, if this gives a profile reduction, the corresponding permutation is applied. `MC67B/BD` performs one or more such passes, stopping if there is no improvement in the profile or if the improvement is small.

Hager's up exchange is similar, with the direction reversed. For a given k, rows and columns $(k, k-1)$, $(k-1, k-2)$, $\dots$, $(l+1, l)$, are exchanged, again with the value of l that most reduces the profile. A pass over the matrix is performed with k taking the values 2, 3, ..., n. `MC67C/CD` performs one or more such passes, again stopping if there is no improvement in the profile or if the improvement is small.

Hager's recommendation is to perform a sequence of pairs of down then up passes until a given number has been performed or a pair yields no further improvement. The given number is under the user's control. We found that it was rare in test cases for there to be much improvement after 5 passes, so we have set the default value to 5. We also offer facilities to perform up then down pairs or passes of one kind only.

6.3.4 HSL_MC73 Fiedler vector and profile reduction of a symmetric sparse matrix (Yifan Hu and Jennifer Scott)

Let A be an $n \times n$ matrix with a symmetric sparsity pattern. `HSL_MC73` has entries to compute the (approximate) Fiedler vector of the unweighted or weighted Laplacian matrix of A and to compute a symmetric permutation that reduces the profile and wavefront of A by using a multilevel algorithm. A number of profile reduction algorithms are offered:

(1) The multilevel algorithm of Hu and Scott (2001) (referred to here as the multilevel Sloan algorithm),

(2) A multilevel spectral ordering algorithm, and

(3) A hybrid algorithm that refines the multilevel spectral ordering (2) using `MC60`.

In each case, an option exists to refine the computed ordering using the Hager exchange algorithm (`MC67`).
If Hager exchanges are not employed, the orderings computed using (1) and (3) generally yield smaller profiles and wavefronts than the spectral ordering (2). For some problems, (1) yields smaller profiles and wavefronts than (3), but for others the converse is true. Algorithm (1) is faster than (3). Using Hager exchanges can substantially increase the ordering cost but can give worthwhile reductions in the profile and wavefront.

6.3.5 MI27 Preconditioned conjugate gradients for a sparse symmetric positive-definite linear system (Mario Arioli and Gianmarco Manzini)

This package uses the conjugate gradient method to solve the $n \times n$ symmetric positive definite linear system $\mathbf{Au} = \mathbf{b}$, optionally using a preconditioning matrix supplied by the user. If $\mathbf{M} = \mathbf{U}^T\mathbf{U}$ is the preconditioning matrix, the routine actually solves the preconditioned system

$$\overline{\mathbf{A}}\mathbf{y} = \overline{\mathbf{b}}, \tag{6.3.4}$$

with $\overline{\mathbf{A}} = \mathbf{U}^{-T}\mathbf{A}\mathbf{U}^{-1}$ and $\overline{\mathbf{b}} = \mathbf{U}^{-T}\mathbf{b}$ and recovers the solution $\mathbf{u} = \mathbf{U}^{-1}\mathbf{y}$.

The novel feature of this code is that several different stopping criteria are provided, including one based on the norm

$$\|\mathbf{r}\|_{\mathbf{A}^{-1}} = (\mathbf{r}^T\mathbf{A}^{-1}\mathbf{r})^{1/2}$$

of the residual

$$\mathbf{r} = \mathbf{A}\mathbf{u}^{(k)} - \mathbf{b}.$$

Reverse communication is used for preconditioning operations and the products of **A** with a vector **z**.

6.3.6 HSL_MP48 Parallel code to solve a general sparse set of equations (Iain Duff and Jennifer Scott)

The module HSL_MP48 solves sets of $n \times n$ unsymmetric linear systems of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, in parallel using Gaussian elimination. The matrix **A** must have been preordered to singly-bordered block-diagonal form

$$\begin{pmatrix} \mathbf{A_{11}} & & & & \mathbf{C_1} \\ & \mathbf{A_{22}} & & & \mathbf{C_2} \\ & & \cdots & & . \\ & & & \mathbf{A_{NN}} & \mathbf{C_N} \end{pmatrix}.$$

MPI is used for message passing.

A partial LU decomposition is performed on each of the submatrices $(\mathbf{A_{ll}} \quad \mathbf{C_l})$ separately. Once all possible eliminations have been performed, for each submatrix there remains a Schur complement matrix $\mathbf{F_l}$. The variables that remain are called interface variables and the interface matrix **F** is formed by summing the matrices $\mathbf{F_l}$. Gaussian elimination is used to factorize **F**, using the HSL sparse direct solver MA48. Block forward elimination and back substitution completes the solution.

The user's matrix data may optionally be held in unformatted sequential files. In addition, **L** and **U** factors for the submatrices may optionally be written to sequential files. This reduces main memory requirements when the number N of submatrices is greater than the number of processes used.

The HSL package HSL_MC66 may be used for preordering the matrix to singly-bordered block-diagonal form.

References

C. W. Gear. *Numerical initial value problems in ordinary differential equations.* Prentice-Hall, Englewood Cliffs, New Jersey, 1970.

W. W. Hager. Minimizing the profile of a symmetric matrix. *SIAM J. Scientific Computing*, **23**(5), 1799–1816, 2002.

Y. F. Hu and J. A. Scott. A multilevel algorithm for wavelength reduction. *SIAM J. Scientific Computing*, **23**(4), 1352–1375, 2001.

7 Seminars

31 January 2002 Dr Ivan Graham (Bath) Iterative methods for PDE eigenproblems.

21 February 2002 Dr Alexander Meeraus (GAMS Development Corporation) Algebraic modelling systems and mathematical programming.

25 April 2002 Dr Stefano Salvini (NAG Ltd) SMP parallelism: current achievements, future challenges.

13 June 2002 Professor Arne Drud (ARKI Consulting and Development) Some complexity issues in sparse LU factorisation.

14 November 2002 Dr Andrew Cliffe (Serco) Computation of periodic orbits for the Navier Stokes equations.

5 December 2002 Alan Curtis Day in Oxford.

13 February 2003 Dr Tony Garratt (AspenTech) Numerical issues arising in dynamic optimisation of process modelling applications.

13 March 2003 Dr Stefan Scholtes (Cambridge) Combinatorial structures in nonlinear programming.

8 May 2003 Dr Shaun Forth (Shrivenham) Elimination automatic differentiation for Jacobian calculation.

19 June 2003 Professor Philippe Toint (Namur) A filter method for the nonlinear feasibility problem.

26 September 2003 Bath-RAL Numerical Analysis Day at RAL.

6 November 2003 Dr Eric Fraga (UCL) Robust numerical methods for computer aided process plant design.

13 November 2003 Dr Patrick Bosander (COMSOL Ltd) Multiphysics modelling in FEMLAB.

8 Reports issued in 2002-2003

We give a full listing of Rutherford Technical Reports issued during the period of this Progress Report. The other report listings, from organizations with which we collaborate, only include reports not already included as RAL reports. All of our current technical reports are publicly accessible via the internet from

`http://www.numerical.rl.ac.uk/reports/reports.html`.

Rutherford Reports

RAL-TR-2002-001 Preprocessing for quadratic programming. N.I.M. Gould and Ph.L. Toint.

RAL-TR-2002-009 CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited. N.I.M. Gould, D. Orban, and Ph.L. Toint.

RAL-TR-2002-010 Numerical Analysis Group Progress Report. January 2000 - December 2001. I. S. Duff (Editor).

RAL-TR-2002-012 Parallel frontal solvers for large sparse linear systems. J.A. Scott.

RAL-TR-2002-014 GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. N.I.M. Gould, D. Orban, and Ph.L. Toint.

RAL-TR-2002-016 Sparse symmetric preconditioners for dense linear systems in electromagnetism. B. Carpentieri, I.S. Duff, L. Giraud, and M. Magolu monga Made.

RAL-TR-2002-018 Algorithm 818: a reference model implementation of the Sparse BLAS in Fortran 95. I. S. Duff and C. Vömel.

RAL-TR-2002-019 A class of incomplete orthogonal factorization methods. II: implementation and results. A.T. Papadopoulos, I.S. Duff, and A.J. Wathen.

RAL-TR-2002-020 A class of spectral two-level preconditioners. B. Carpentieri, I.S. Duff, and L. Giraud.

RAL-TR-2002-021 A Chebyshev-based two-stage iterative method as an alternative to the direct solution of linear systems. M. Arioli and D. Ruiz.

RAL-TR-2002-024 MA57 - A new code for the solution of sparse symmetric definite and indefinite systems. I.S. Duff.

RAL-TR-2002-026 Null space algorithm and spanning trees in solving Darcy's equation. M. Arioli and G. Manzini.

RAL-TR-2002-028 Task scheduling in an asynchronous distributed memory multifrontal solver. P.R. Amestoy, I.S. Duff, and C. Voemel.

RAL-TR-2002-029 Jacobian code generated by source transformation is as efficient as hand coding. S.A. Forth, M. Tadjouddine, J.D. Pryce, and J.K. Reid.

RAL-TR-2002-030 The new features of Fortran 2000. J.K. Reid.

RAL-TR-2002-031 An introduction to algorithms for nonlinear optimization. N.I.M. Gould and S. Leyffer.

RAL-TR-2002-032 An active-set algorithm for nonlinear programming using linear programming and equality constrained subproblems. R.H. Byrd, N.I.M. Gould, J. Nocedal, and R.A. Waltz.

RAL-TR-2002-033 A parallel direct solver for large sparse unsymmetric linear systems. I.S. Duff and J.A. Scott.

RAL-TR-2002-034 A stopping criterion for the conjugate gradient algorithm in a finite-element method framework. M. Arioli.

RAL-TR-2003-003 Global convergence of a non-monotone trust-region filter algorithm for nonlinear programming. N.I.M. Gould and Ph. L. Toint.

RAL-TR-2003-004 A multidimensional filter algorithm for nonlinear equations and nonlinear least squares. N.I.M. Gould, S. Leyffer, and Ph. L. Toint.

RAL-TR-2003-009 Stopping criteria for iterations in finite-element methods. M. Arioli, D. Loghin and A. J. Wathen.

RAL-TR-2003-013 On the convergence of successive linear programming algorithms. R.H. Byrd, N.I.M. Gould, J. Nocedal, and R.A. Waltz.

RAL-TR-2003-017 FILTRANE, a Fortran 95 filter-trust-region package for solving nonlinear feasibility problems. N.I.M. Gould and Ph. L. Toint.

RAL-TR-2003-019 A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations. N.I.M. Gould and J.A. Scott.

RAL-TR-2003-021 A Chebyshev-based two-stage iterative method as an alternative to the direct solution of linear systems. M. Arioli and D. Ruiz.

RAL-TR-2003-020 Ordering techniques for singly bordered block diagonal forms for unsymmetric parallel direct solvers. Y. Hu and J.A. Scott.

RAL-TR-2003-022 An interior-point l_1-penalty method for nonlinear optimization. N.I.M. Gould, D. Orban, and Ph. L. Toint.

RAL-TR-2003-023 Using spectral low rank preconditioners for large electromagnetic calculations. I.S. Duff, L. Giraud, J. Langou, and E. Martin.

RAL-TR-2003-024 Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations. B. Carpentieri, I.S. Duff, L. Giraud, and G. Sylvand.

RAL-TR-2003-026 Null space algorithm and spanning trees in solving Darcy's equation. M. Arioli and G. Manzini.

RAL-TR-2003-034 A stopping criterion for the conjugate gradient algorithm in a finite-element method framework. M. Arioli.

RAL-TR-2003-036 A fast multilevel Fiedler and profile reduction code. Y. Hu and J.A. Scott.

CERFACS Reports

TR/PA/02/13 General CUTEr documentation. N.I.M. Gould, D. Orban, and Ph.L. Toint.

TR/PA/02/14 General SifDec documentation. N.I.M. Gould, D. Orban, and Ph.L. Toint.

TR/PA/03/05 Adapting a parallel sparse direct solver to SMP architectures. P. Amestoy, I.S. Duff, S. Pralet, and C. Vömel.

TR/PA/03/14 Impact of the implementation of MPI point-to-point communications on the performance of two general sparse solvers. P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, and X.S. Li.

TR/PA/03/65 Efficient parallel iterative solvers for the solution of large dense linear systems arising from the boundary element method in electromagnetism. G. Alléon, B. Carpentieri, I. S. Duff, L. Giraud, J. Langou, E. Martin, and G. Sylvand.

9 External Publications in 2002-2003

P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and Xiaoye S. Li. Impact of the implementation of MPI point-to-point communications on the performance of two general sparse solvers. *Parallel Computing*, **29**(7), 833–849, 2003.

P. R. Amestoy, I. S. Duff, S. Pralet, and C. Vömel. Adapting a parallel sparse direct solver to architectures with clusters of smps. *Parallel Computing*, **29**(11-12), 1645–1668, 2003.

M. Arioli and G. Manzini. A null space algorithm for mixed finite-element approximations of Darcy's equation. *Commun. Numer. Meth. Engng*, **18**, 645–657, 2002.

M. Arioli and G. Manzini. Null space algorithm and spanning trees in solving Darcy's equation. *BIT*, **43**, 839–848, 2003.

L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Softw.*, **28**(2), 135–151, 2002.

B. Carpentieri, I. S. Duff, and L. Giraud. A class of spectral two-level preconditioners. *SIAM J. Scientific Computing*, **25**(2), 749–765, 2003.

I. S. Duff and C. Vömel. Algorithm 818: a reference model implementation of the Sparse BLAS in Fortran 95. *ACM Trans. Math. Softw.*, **28**(2), 268–283, 2002.

I. S. Duff and C. Vömel. Incremental norm estimation for dense and sparse matrices. *BIT*, **42**(2), 300–322, 2002.

I. S. Duff, M. A. Heroux, and R. Pozo. An overview of the Sparse Basic Linear Algebra Subprograms: the new standard from the BLAS Technical Forum. *ACM Trans. Math. Softw.*, **28**(2), 239–267, 2002.

R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter. Global convergence of a trust-region SQP-filter algorithm for nonlinear programming. *SIAM Journal on Optimization*, **13**(3), 635–659, 2002.

N. I. M. Gould. Some reflections on the current state of active-set and interior-point methods for constrained optimization. *SIAG/Optimization Views-and-News*, **14**(1), 2–7, 2003.

N. I. M. Gould and S. Leyffer. An introduction to algorithms for nonlinear optimization. *in* A. W. C. (J. F. Blowey and T. Shardlow, eds, 'Frontiers in Numerical Analysis (Durham 2002)', pp. 109–197, Heidelberg, Berlin, New York, Springer Verlag, 2003.

N. I. M. Gould and Ph. L. Toint. An iterative working-set method for large-scale non-convex quadratic programming. *Applied Numerical Mathematics*, **43**(1–2), 109–128, 2002.

N. I. M. Gould and Ph. L. Toint. Numerical methods for large-scale non-convex quadratic programming. *in* A. H. Siddiqi and M. Kočvara, eds, ‘Trends in Industrial and Applied Mathematics’, pp. 149–179, Dordrecht, The Netherlands, Kluwer Academic Publishers, 2002.

N. I. M. Gould and Ph. L. Toint. Global convergence of a hybrid trust-region sqp-filter algorithm for general nonlinear programming. *in* E. W. Sachs and R. Tichatschke, eds, ‘System Modelling and Optimization XX’, pp. 23–54, Dordrecht, The Netherlands, Kluwer Academic Publishers, 2003.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited. *ACM Trans. Math. Softw.*, **29**(4), 373–394, 2003.

N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Trans. Math. Softw.*, **29**(4), 353–372, 2003.

N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Componentwise fast convergence in the solution of full-rank systems of nonlinear equation. *Mathematical Programming, Series B*, **92**(3), 481–508, 2002.

J.K. Reid and J.A. Scott. Implementing Hager’s exchange methods for matrix profile reduction. *ACM Trans. Math. Softw.*, **28**, 1–15, 2002.

J.A. Scott. Parallel frontal solvers for large sparse linear systems. *ACM Trans. Math. Softw.*, **29**, 395–417, 2003.