RAL-TR-2000-001

Numerical Analysis Group Progress Report
January 1998 - December 1999

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 1998 to December 1999.

Keywords: sparse matrices, frontal and multifrontal methods, numerical linear algebra, large-scale eigenvalue computations, large-scale optimization, Fortran, Harwell Subroutine Library, HSL

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

Current reports available by anonymous ftp to ftp.numerical.rl.ac.uk in directory pub/reports. This report is in file duffRAL2000001.ps.gz. Report also available through URL `http://www.numerical.rl.ac.uk/reports/reports.html`.

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

February 14, 2000

Contents

Personnel in Numerical Analysis Group

Staff

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

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

Jacko Koster (February 1998 until June 30th 1999).
PARASOL Project.

Jean-Yves L'Excellent (May 1999).
PARASOL Project.

Karl Meerbergen (from May 1st 1998).
Large-scale eigenvalue problems.

Petr Plecháč (until August 31st 1998).
PARASOL Project.

John Reid (until August 31st 1998, then consultant for HSL).
Sparse matrices and development of the Fortran programming language.

Jennifer Scott.
Sparse linear systems and sparse eigenvalue problems.

Kath Vann (from November 1999).
Administrative and secretarial support.

Visitors and Attached Staff

Mike Hopper (Consultant)	Support for Harwell Subroutine Library and for TSSD.
Andy Conn (IBM Yorktown Heights)	Optimization.
Andreas Findling (NEC Stuttgart)	Software tuning.
Yifan Hu (CLRC Daresbury)	Sparse linear systems.
Rich Lehoucq (SANDIA National Laboratories)	Numerical linear algebra.
Frédéric Magoulès (ONERA)	Domain decomposition.
Jorge Nocedal (Northwestern University)	Optimization.
Philippe Toint (University of Namur)	Optimization.

1 Introduction (I. S. Duff)

This report covers the period from January 1998 to December 1999 and describes work performed by the Numerical Analysis Group within the Computational Science and Engineering Department at the CLRC Rutherford Appleton Laboratory.

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.

Two years ago, we reported that the past period had seen substantial organizational changes and this has remained true throughout the period of this report. This time the Department not only changed its name but split into the Information Technology Department and the Computational Science and Engineering Department with our Group being included in the latter. Since Paul Durham is the Director of the Department this has strengthened even further our links with the Daresbury Laboratory campus of CLRC. We have recently established links with the ISIS people at Rutherford and have discussed their use of HSL codes. In the reorganization, we lost the secretarial support of Linda Miles, who left CLRC for a career break, and had no secretarial support for nearly half of this year until Kath Vann joined us in November 1999.

However, perhaps the most dramatic aspect of our life in this period has been our happily successful efforts at establishing a more stable funding regime for the Group. The complications of doing this have seemed to us often artificial and we are becoming quite accomplished at jumping through the variety of hoops presented to us. Towards the end of 1998, we prepared for a renewal of our Grant from the Cross Programmes Group of the EPSRC only to learn when it was too late to submit a further application that this was now deemed inappropriate and we should have to apply directly to the Mathematics Programme. A Grant extension application to Cross Programmes was then hastily prepared and, after appropriate review, continued our funding until September 30th 1999. We would like to acknowledge the help of Alasdair Rose and Vanessa Johnson of the Mathematics Programme and Maggie Wilson of the Cross Programmes Group for their help in negotiating these minefields. This then enabled us to prepare an application for a Programme Grant for four years which was submitted to the Mathematics Programme but was reviewed also by other Programmes and by a panel that visited us on site at Rutherford. Both the Visiting Panel and the Mathematics responsive mode panel gave us very favourable reviews and our Grant was awarded, although we have some conditions on dissemination to be satisfied before it is extended from two to four years ... yet more hoops.

The support and development of the Harwell Subroutine Library (HSL) is one of our major activities. There have been no releases of either HSL or the NAg-marketed Harwell

Sparse Matrix Library during the period of this report. The HSL marketing effort from AEA Technology PLC has again seen changes of personnel and has moved into the care of Nick Brealey of the Electromagnetics Department at Culham Laboratory. Since we now employ John Reid as a consultant using HSL funds, the continued sales of the Library are even more important to us than formerly. We have benefited greatly from the consultancy of Mike Hopper who helped 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 John continue as Visiting Professors at Strathclyde University and RMCS Shrivenham, respectively, and Nick was recently appointed a Visiting Professor at the University of Edinburgh. All members of the Group gave contributed talks at the Dundee Numerical Analysis meeting in 1999, Iain co-hosted a visitor from China (Zhong-Zhi Bai) with Andy Wathen from Oxford on an EPSRC grant, and Nick and Iain are supervising a CASE student (Carsten Keller), again with Andy Wathen. Jennifer and Karl co-hosted a visitor from USA (Rich Lehoucq), again with Andy. Iain was on the jury for the PhD thesis of Serge Gratton and for the habilitation theses of Patrick Amestoy and of Annick Sartenaer in Toulouse, Jennifer was the external examiner for the PhD thesis of Stuart Hawkins at Bath, Karl was on the jury for the PhD thesis of Gorik De Samblanx in K.U. Leuven in Belgium, and Nick was the external examiner for the PhD thesis of Bobby Cheng from Manchester.

Most of our visitors stayed only for a short time although the interaction with them has been quite intense. 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 8).

John has continued to combine his interests in Fortran and sparse matrices giving several talks on these topics during the last two years. He completed his work on scaling in least-squares problems, worked with Jennifer on frontwidth minimization, developed the sparse linear programming driver `LA04`, and spoke on the PARASOL interface at a meeting in Greenwich. He attended and spoke at the ISO WG5 meeting in Trollatten (Sweden). John has developed his work with Numrich on co-array Fortran and spoke about this at meetings at Heathrow and Oxford. He has given courses on Fortran 90 at RAL and RMCS Shrivenham. He has continued with his collaboration on automatic differentiation with groups at Shrivenham, the University of Hertfordshire, and NAg Ltd. John was obliged to retire on age grounds in August 1998 but happily we have been able to retain his partial services through the use of HSL funds so he continues as an active Group member, naturally focussed primarily on issues relating to HSL.

Nick's collaboration with Conn and Toint continues to expand the theory and practice of large-scale optimization and, during the period of this report, they completed their magnum opus, a 966 page book on trust-region methods that will be published by SIAM next year. Nick was also appointed as a Visiting Professor of Mathematics at the University of Edinburgh in November 1998 and has already had a visit there in that capacity. In addition to the book and closely related research, he has developed algorithms and codes for the solution of large-scale quadratic programming problems both using barrier-function methods and active-set techniques. Both approaches are likely to be used in his new nonlinearly constrained optimization package GALAHAD that will eventually supersede the well known and highly successful augmented Lagrangian package LANCELOT. His work with Conn and Toint has occasioned visits to CERFACS in Toulouse and Namur in Belgium. Nick was an invited speaker at conferences at Erice and Cambridge, gave seminars in Dundee, Edinburgh, and Oxford, and was an organizer and speaker at the Foundations of Computational Mathematics Meeting (FoCM 99) at Oxford. We are delighted to record his Individual Merit promotion to Band 2 this June.

Jennifer has continued with her national and international collaborations and her qualities were well recognized by promotion to Band 3 in July 1998. Although she has continued her short-hours working, she remains so productive that it is easy to forget this fact. Much of her work in this period has been on enhancing the performance of frontal solvers and using them in practical industrial problems. She has developed very effective ordering techniques for use with frontal methods both when the matrix is assembled and when it is held as a set of element matrices. She has completed an MPI harness for using the frontal code on distributed memory machines that has proven very effective on small numbers of processors and has collaborated with AEA Technology, NEC, WS Atkins, and chemical engineers in using frontal methods to great effect on their practical industrial problems. Jennifer continues to coordinate our joint seminar series with Oxford University. She was a coordinator for the EPSRC Summer School in Leicester, gave an invited talk at a workshop of physicists in Strasbourg, and visited Stuttgart as part of a collaboration with NEC (see Section 2.7). She spoke at an ERCIM Executive meeting at RAL and the Householder XIV Symposium in Canada. She presented seminars at Oxford and Portsmouth, and has attended meetings in Edinburgh, London, Manchester, and Oxford.

Karl joined us from Leuven in Belgium at the beginning of May 1998 and has been very active in establishing contacts with collaborators in Europe and local universities and pursuing research on eigensystem problems. He was also involved in an industrial contract with his previous employers in using domain decomposition techniques within frequency response calculations for acoustic modelling in car interiors. His main efforts have been on the solution of quadratic eigenproblems and the development of a rational Lanczos method. He has co-authored a chapter in a forthcoming book on eigensystem templates that will be published by SIAM. He has established collaborations with De Samblanx at KU Leuven, Magoulès at ONERA, and Spence at Bath. He was an invited speaker at the Eighth EPSRC Summer School at Leicester, presented a talk at the Householder XIV Symposium in Whistler, and gave seminars at Bath, Daresbury, Oxford, Utrecht, and Warwick. He was awarded an honorary mention in the Householder Prize for the best thesis in numerical linear algebra from 1996 to 1999.

The EU LTR Project PARASOL (see Section 6.2) finished on June 30th 1999 and, as one might expect, its last year saw much activity. In the context of this Project, we had three people working at Rutherford during the period. Petr Plecháč worked on ordering schemes, including the graph partitioning packages METIS and RALPAR, until he left for an assistant professorship at the University of Delaware in August 1998. Petr attended meetings in Greenwich and Manchester to give talks on the Project and presented an invited talk at the GAMM meeting in Kiel and a seminar at Imperial College. He also was an invited speaker at the EPSRC Summer School in Leicester and visited CERFACS to work with Patrick Amestoy and Jean-Yves L'Excellent on MUMPS. Jacko Koster joined us from CERFACS where he had already spent two months as a postdoc on the PARASOL Project. He took the lead role in package integration and design of the test driver and stayed with us until the end of the Project. He attended all of the PARASOL working meetings and is now working as a postdoc at Parallab in Bergen, where he is primarily concerned with using MUMPS within the Parallab domain decomposition code DDM. While at RAL, Jacko was also involved in further work on matrix preprocessing (Section 3.2) and in the project with NEC (Section 2.7). Jean-Yves L'Excellent, who was working for CERFACS in a subcontract with Patrick Amestoy at ENSEEIHT-IRIT, visited us for the month of May 1999 and worked closely with Jacko. After the end of the Project, Jean-Yves got a job at NAg Ltd in Oxford.

Iain still leads a project at the European Centre for Research and Advanced Training in Scientific Computation (CERFACS) at Toulouse in France (see Section 6.1). Iain is an Editor of the IMA Journal of Numerical Analysis, an Honorary Secretary of the IMA, editor of the IMANA Newsletter, chairman of the IMA Programme Committee, was chairman of the Adjudicating Committee for the Fox Prize Meeting in June 1999, IMA representative on the CCIAM International Committee that oversees the ICIAM international conferences on applied mathematics, a member of the International Scientific Programme Committee for ICIAM '99, and is on the Mathematics College of the EPSRC. He has a Grant to work with Xiaoye Li at NERSC, Berkeley on sparse solution techniques. In high performance computing, he has given tutorials at SC'98 (Orlando, Florida) and EuroPar'99 (Toulouse) and gave lectures at summer schools at Orsay in Paris and in Natal, Brazil and was workshop coordinator for a meeting in Copper Mountain, Colorado. He completed his book with Dongarra, Sorensen, and van der Vorst on numerical linear algebra on high performance computers that was published by SIAM in 1998. Iain was the general chairman for the EuroPar'99 meeting that attracted over 400 people to Toulouse in September 1999 and has been on the Programme and Organizing Committee for several international meetings including the SIAM Parallel Processing Conference in San Antonio and an IMA meeting in Oxford. He has given invited talks at meetings in Belfast, Dalian (China), Edinburgh, Glasgow, Helsinki, Plovdiv (Bulgaria), Toulouse, and Umeå. He was a guest of the Chinese Academy of Sciences in Beijing. He has presented contributed talks and seminars in Gainesville (Florida), Gothenberg, ICIAM'99 (Edinburgh), Knoxville, Lexington, Minneapolis, NAg Ltd, NERSC, San Antonio, Strathclyde, and Whistler (Canada), and gave some postgraduate lectures at Strathclyde.

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 frontal or multifrontal methods in Section 2 and other numerical linear algebra activities in Section 3. Work on optimization is considered in Section 4 and activities on the Fortran programming language in Section 5. We group some miscellaneous topics in Section 6. Much of our research and development results in high quality advanced mathematical software, and we report on our computer infrastructure and software developments in Section 7. Lists of seminars (in the joint series with Oxford), technical reports, and publications are given in Sections 8, 9, and 10, respectively. Current information on the activities of the Group and on Group members can be found through page `http://www.cse.clrc.ac.uk/Group/CSENAG` of the World Wide Web.

2 Frontal and multifrontal methods

2.1 MUMPS - a distributed memory multifrontal solver (P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent and J. Koster)

The European project PARASOL (see Section 6.2) had as its main goal the design and development of a library of scalable sparse matrix solvers for distributed memory computers. The CLRC Rutherford Appleton Laboratory, CERFACS, and ENSEEIHT (both Toulouse, France), were responsible for the direct solvers. In this context, we have developed a MUltifrontal Massively Parallel Solver (MUMPS) (Amestoy, Duff, L'Excellent and Koster, 1999, Amestoy, Duff and L'Excellent, 2000).

MUMPS has been designed to solve symmetric positive definite, general symmetric, and unsymmetric linear systems whose coefficient matrices are possibly rank deficient. The MUMPS package uses a multifrontal approach to factorise the matrix (Duff and Reid, 1983, Duff and Reid, 1984). Similar to serial HSL solvers, the parallel MUMPS package solves in three main steps: an analysis step, a factorization step and a solution step.

MUMPS achieves high performance by exploiting two kinds of parallelism: tree parallelism that comes from the sparsity of the problem and node parallelism from dense matrix kernels. 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. MUMPS overlaps computation with communication by using asynchronous communication.

In the years 1998 and 1999, there were several important upgrades to the MUMPS package. Figure 2.1 lists the major releases of the package for partners in the PARASOL project. The continuous development and incorporation of new features requested by the partners over the past two years has resulted in a software package (currently MUMPS version 4.1.1) that is unique amongst sparse direct solvers.

So far, the MUMPS software has mainly been used for solving problems from the industrial partners in the PARASOL project. Typical PARASOL test cases are from application areas such as computational fluid dynamics, structural mechanics, modelling compound devices, modelling ships and mobile offshore platforms, industrial processing of complex non-Newtonian liquids, and modelling car bodies and engine components. Table 2.2 shows the performance of the MUMPS factorization and solution phases on a symmetric positive definite matrix (provided by MSC) that comes from the modelling of an inline skater. The matrix is of order 503, 712 and has 18.7 million nonzeros in its lower triangular part. The factorization requires 1.4×10^{11} floating-point operations and the factors contain 175 million entries. The largest problem we have solved to date is a model of an AUDI

Version	Important new features	Release month
1.0	MPI version using only tree parallelism.	5/97
2.0	Uses both node and tree parallelism.	2/98
	Uses ScaLAPACK at root node of elimination tree.	
	Better data management enables solution of larger problems.	
	Uses PARASOL interface (host-node paradigm).	
2.1	Version for symmetric positive definite matrices	5/98
2.2	Available with Fortran 90 interface:	9/98
	Ability to handle general symmetric matrices.	
	Hybrid host version (including serial code).	
	Basic version of rank estimate and null-space.	
2.2	Available with PARASOL interface:	10/98
	Includes orderings based on METIS and other graph partitioning strategies.	
3.1	Version with entry for unassembled matrices.	1/99
	Improved strategies for rank estimation and null-space.	
3.2	Capability of handling distributed matrix input.	2/99
4.0	Final code for PARASOL partners.	4/99
	Better tuned/interface for other PARASOL codes.	
	Ability to return a Schur complement matrix.	
4.1.1	Current version.	12/99

Table 2.1: The versions of MUMPS in 1998/1999.

crankshaft. The corresponding linear system is symmetric positive definite and of order 943, 695 with more than 39 million entries in its lower triangular part. With the best ordering of the unknowns that we tried, MUMPS created 1.4 billion entries in the factors and required 5.9×10^{12} floating-point operations for the factorization.

number of	elapsed time	
processors	factorization	solution
1	723	18.5
2	385	10.7
4	222	8.8
8	151	5.0
12	97	4.4
16	68	4.2
32	62	4.4

Table 2.2: Factorisation and solution time (in seconds) for MUMPS on the INLINE500K test case (503, 712 unknowns) on an SGI Origin 2000 (195Mhz) machine.

The MUMPS software is written in Fortran 90. It requires MPI for message passing and makes use of BLAS, LAPACK, BLACS, and ScaLAPACK subroutines. MUMPS was developed and tested on an IBM SP2, an SGI Power Challenge, and an SGI Origin 2000. The software has recently been ported to a Cray T3E.

References

P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods in Appl. Mech. Eng.*, to appear, 2000.

P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. Technical Report RAL-TR-1999-059, Rutherford Appleton Laboratory, 1999.

I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear systems. *ACM Trans. Math. Softw.*, **9**, 302–325, 1983.

I. S. Duff and J. K. Reid. The multifrontal solution of unsymmetric sets of linear systems. *SIAM J. Scientific and Statistical Computing*, **5**, 633–641, 1984.

2.2 Ordering algorithms for the MUMPS solver (P. Plecháč)

Within the PARASOL project, Petr worked on the implementation of new ordering strategies for the MUMPS solver. In collaboration with Iain and Ron Fowler (RAL), he developed six ordering strategies which have been implemented in the PARASOL interface.

The package of ordering algorithms provides strategies based on local heuristics (minimum degree and its modifications), on global heuristics (nested dissection and multilevel spectral methods), and combinations of these two approaches. The new ordering algorithms include the HALO-AMD ordering and dynamic control of the subgraph size in the graph partitioning phase. These have been designed in collaboration with Iain and Patrick Amestoy and their efficiency was tested on many examples. Petr also developed and implemented a general test deck (GPMAT) that allows the testing of different ordering strategies for large sparse matrices on different platforms. The PARASOL interface was extended to include the library of ordering algorithms so that they can be used by the MUMPS solver in the PARASOL library.

The ordering algorithms included in the PARASOL library and accessible through the PARASOL interface are:

1. approximate minimum degree [AMD],
2. multilevel bisection and multiple minimum degree on subgraphs [OE0],
3. multilevel bisection and approximate minimum degree on subgraphs [OE1],
4. multilevel bisection and HALO-AMD on subgraphs [OE2],
5. multilevel bisection, vertex separator and AMD on subgraphs [ON0],
6. multilevel bisection, multi-sector and HALO-AMD on subgraphs [RL0],
7. multilevel bisection, refinement and HALO-AMD on subgraphs [RL1].

The implementation of algorithms of the [OEx] family is based on the METIS library (Karypis and Kumar, 1997) while the algorithms denoted as [RLx] were developed in collaboration with Ron Fowler and are based on the RALPAR package (Fowler and Greenough, 1998).

The PARASOL version of the GPMAT library has been successfully tested on PARASOL test problems. In particular, the algorithm [ON0] was found to be extremely efficient for large test cases and significantly reduced both the number of floating-point operations and the fill-in for the MUMPS solver.

References

R. F. Fowler and C. Greenough. RALPAR - RAL Mesh Partitioning Program. Version 2.0. Technical Report RAL-TR-98-025, Rutherford Appleton Laboratory, 1998.

G. Karypis and V. Kumar. METIS: Unstructured graph partitioning and sparse matrix ordering system. Technical Report TR 97-061, Department of Computer Science, University of Minnesota, Minnesota, 1997.

2.3 Element resequencing for frontal solvers (J. A. Scott)

A key feature of the frontal method is that, in the innermost loop of the computation, dense linear algebra kernels can be exploited. These are straightforward to vectorize and parallelize and are able to exploit high level BLAS kernels. This makes the method attractive for a wide range of modern computer architectures, including RISC based processors and shared memory parallel processors. However, the performance of the frontal method, in terms of both memory requirements and the number of floating-point operations, is highly dependent on the order in which the elements (or rows, if the matrix is in assembled form) are added into the frontal matrix. The elements (or rows) need to be added in an order that keeps the size of the frontal matrix small. In this section we consider element ordering algorithms and, in the next section, we look at algorithms for ordering rows for a frontal solver.

Element ordering algorithms may be divided into direct and indirect algorithms. Direct algorithms order the elements directly while indirect algorithms use a two-step approach in which the variables are first relabelled and used to resequence the elements; the new variable indices are subsequently discarded. The Harwell Subroutine Library code `MC43` implements both a direct and an indirect ordering algorithm, based on variants of the profile reduction algorithm of Sloan (1986). The code has been in satisfactory use for a decade. Motivated by the findings of Kumfert and Pothen (1997), we considered a number of ways of improving the performance and efficiency of Sloan's algorithm (Reid and Scott, 1999). This work led to improved codes for profile reduction (the Harwell Subroutine Library package `MC60` together with a driver `MC61`) and prompted us to look at revising `MC43` in a similar way. The new element ordering code is called `MC63`. The main differences between `MC43` and `MC63` are:

- The handling of the priority queue. In `MC63`, a switch to a binary heap search is made once the length of the queue exceeds a given threshold. For large problems, this can significantly reduce the time needed to reorder the elements. This is illustrated in Table 2.3.

Table 2.3: A comparison of CPU times for MC43 and MC63 (Sun Ultra).

Identifier	Order	Elements	MC43	MC63
cham	12834	11070	1.34	0.79
fullb	199187	59738	5.45	2.55
mt1	97578	5328	7.55	3.61
shipsec1	140874	41037	16.1	5.72
thread	29736	2176	1.94	0.83
tubu	26573	23446	3.22	1.62

- MC63 includes an option for the user to supply the weights in the priority function. Experiments showed that, for some problems, the weights used by Duff, Reid and Scott (1989) in MC43 are far from optimal.

- A further option in MC63 permits users to provide a global priority vector. Kumfert and Pothen (1997) report that the final ordering can be significantly better if a hybrid algorithm that combines a spectral ordering with the Sloan algorithm is used. Our numerical experiments showed that the hybrid method generally gives smaller profiles than the Sloan algorithm. Some examples illustrating this are given in Table 2.4.

Table 2.4: Root-mean-square wavefronts for the Sloan and hybrid algorithms

Identifier	Order	Elements	Sloan	Hybrid
crplat2	18010	3152	359	242
fullb	199187	59738	2021	1866
lock3491	3416	684	135	104
shipsec1	140874	41037	2494	1555
thread	29736	2176	1962	1858
tubu	26573	23446	447	393

References

I. S. Duff, J. K. Reid, and J. A. Scott. The use of profile reduction algorithms with a frontal code. *Inter. Journal on Numerical Methods in Engineering*, **28**, 2555–2568, 1989.

G. Kumfert and A. Pothen. Two improved algorithms for envelope and wavefront reduction. *BIT*, **18**, 559–590, 1997.

J. K. Reid and J. A. Scott. Ordering symmetric sparse matrices for small profile and wavefront. *Inter. Journal on Numerical Methods in Engineering*, **45**, 1737–1755, 1999.

J. A. Scott. On ordering elements for a frontal solver. *Communications in Numerical Methods in Engineering*, **15**, 309–323, 1999.

S. W. Sloan. An algorithm for profile and wavefront reduction of sparse matrices. *Inter. Journal on Numerical Methods in Engineering*, **23**, 1315–1324, 1986.

2.4 Row ordering for frontal solvers (J. A. Scott)

A number of algorithms have recently been proposed for ordering the rows of unsymmetric matrices for the row-by-row frontal method. These include the RCMD and NMNC algorithms (Camarda, 1997). However, numerical experiments have shown that although the cost of reordering with the RCMD algorithm is negligible compared with the time required by the subsequent numerical factorization of the matrix, for many problems it performs poorly and frequently yields orderings that are worse than the original. The NMNC algorithm is more consistent, but is generally only able to achieve relatively modest improvements. This led us to look at designing and developing new row ordering algorithms, building on the expertise and experience we already have for ordering elements.

Our new algorithm is called the MSRO algorithm. It uses the *row graph* of the matrix A, which is defined to be the undirected graph of the symmetric matrix $B = A * A^T$, where $*$ denotes matrix multiplication without taking cancellations into account. The nodes of the row graph are the rows of A and two nodes i and j $(i \neq j)$ are adjacent if and only if there is at least one column k of A for which a_{ik} and a_{jk} are both nonzero. Row permutations of A correspond to relabelling the nodes of the row graph.

The MSRO algorithm comprises two distinct phases: the selection of a global ordering and then a local row reordering. The global ordering defines the global priority of each row. The row with the lowest global priority is chosen as the *start* row (that is, the row that is first in the global ordering is ordered first in the new ordering). In the second phase of the algorithm, the global ordering is used to guide the reordering. Rows with a high global priority are chosen towards the end of the ordering. The global ordering methods we have used are the pseudodiameter of the row graph, the spectral ordering of the row graph, and the NMNC ordering.

The local ordering is based on a *priority function*. The basic idea is to select the next row in the ordering by choosing, from a set of eligible rows, a row with minimum priority. Eligible rows are those that are adjacent to a row that has already been ordered together with their neighbours in the row graph. The priority of a row is the weighted average of its global priority and its local priority, which is based on the increases in the row and column frontsizes resulting from ordering the row next. Thus, the MSRO algorithm attempts to maintain a balance between having only a small number of rows and columns in the front and including rows that have a low global priority.

The MSRO algorithm has been tested and compared with the RCMD and NMNC algorithms on a range of practical problems. Some results are given in Tables 2.5 and 2.6. Our main conclusion is that the MSRO algorithms generally perform significantly better than existing algorithms and, when used with the frontal solver `MA42`, can achieve CPU savings of up to 90 per cent compared with the original ordering. Looking at using the different global orderings with the MSRO algorithm we found that, for most problems, the pseudodiameter gives better results than using the NMNC ordering, while in turn the spectral ordering is better than the pseudodiameter. The only problems for which the MSRO algorithm with the pseudodiameter or spectral ordering does not appear to work well are those which have a large average number of neighbours for each row and, compared

Table 2.5: The mean frontal matrix size for the different reordering algorithms. † denotes spectral ordering not available.

Identifier	Order	Original	RMCD	NMNC	MSRO+	MSRO+	MSRO+
					pseudo	spectral	NMNC
					diameter		
`4cols`	2218	11770	361	982	30	45	61
`10cols`	7091	29496	448	2422	39	†	87
`bayer04`	1911	20545	4162	1683	334	59	972
`bayer09`	249	3083	152	230	20	12	178
`ethylene-1`	1452	10673	11249	573	3910	2449	213
`extr1`	49	2837	486	34	4	3	18
`lhr34c`	1499	35152	48940	1499	283	172	472

Table 2.6: The factorization times (in seconds) for the frontal solver `MA42` before and after reordering with the MSRO algorithm (Sun Ultra).

Identifier	Before	After	Identifier	Before	After
`4cols`	17.8	2.8	`10cols`	84.4	7.6
`bayer04`	21.8	5.0	`bayer09`	0.9	0.4
`ethylene-1`	7.5	5.1	`extr1`	0.5	0.3
`lhr34c`	214	158			

with the order of the matrix, the pseudodiameter is small. Full results are given in Scott (1999a, 1999b).

A code, MC62, that implements the MSRO algorithm and optionally allows the user to supply a global ordering, has been designed and developed for the Harwell Subroutine Library. While performing numerical experiments, we observed that it can be advantageous to reverse a given row ordering. We investigated this and were able to prove a number of results on the invariance of the column frontsize (Reid and Scott, 1999).

References

K. V. Camarda. *Ordering strategies for sparse matrices in chemical process simulation.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.

J. K. Reid and J. A. Scott. Reversing the row order for the row-by-row frontal solver. Technical Report RAL-TR-1999-037, Rutherford Appleton Laboratory, 1999.

J. A. Scott. *A new row ordering strategy for frontal solvers. Numerical Linear Algebra with Applications*, **6**, 1–23, 1999*a*.

J. A. Scott. Row ordering for frontal solvers in chemical process engineering. Technical Report RAL-TR-1999-035, Rutherford Appleton Laboratory, 1999*b*.

2.5 Further developments for the frontal solver MA62 (I. S. Duff and J. A. Scott)

We have developed a new frontal solver MA62 (Duff and Scott, 1997, Duff and Scott, 1999) for the Harwell Subroutine Library. MA62 is designed specifically for the efficient solution of sparse positive-definite symmetric systems from finite-element applications with the coefficient matrix A in element form $A = \sum_{l=1}^{m} A^{(l)}$. The code has been available for the past two years and has been used under licence by W.S. Atkins, who have incorporated the code within their finite-element analysis package, ASAS, and have achieved some impressive results with it. In an extreme case recently, Paul Schofield from W.S. Atkins told us of an analysis requiring 6 hours with their original solver that took only 5 minutes using the new MA62-based solver. Partly in response to the findings of W.S. Atkins and partly from our own extensive testing of the code, a number of minor modifications have recently been made to MA62. The most significant of these reduces memory requirements by overwriting the right-hand side array with the solution vector.

A major deficiency of the frontal solution scheme is the lack of scope for parallelism other than that which can be obtained within the high-level BLAS. In a multiple front algorithm, the finite-element domain is partitioned into non-overlapping subdomains and a frontal decomposition is performed on each subdomain separately. To enable MA62 to be used for this, we have developed MA72. MA72 provides routines for generating lists of interface variables, for preserving the partial factorization of a matrix when the sequence of calls to the factorization routine MA62B is incomplete, and for performing forward elimination or backsubstitution on a subdomain. MA62 and MA72 will be included in a forthcoming release of the Harwell Subroutine Library.

References

I. S. Duff and J. A. Scott. MA62 – a new frontal code for sparse positive-definite symmetric systems from finite-element applications. Technical Report RAL-TR-97-012, Rutherford Appleton Laboratory, 1997.

I. S. Duff and J. A. Scott. A frontal code for the solution of sparse positive-definite symmetric systems arising from finite-element applications. *ACM Trans. Mathematical Software*, to appear, 1999.

2.6 The design and development of two parallel frontal solvers (J. A. Scott)

We have designed and developed two parallel frontal solvers: MP42 is for unsymmetric finite-element problems and is based upon MA42 and MA52, while MP62 is for symmetric positive definite problems and uses MA62 and MA72. A major difficulty in developing parallel software is enabling it to be used on a wide range of architectures. To overcome portability problems, the codes MP42 and MP62 are written in Fortran 90 and use MPI for message passing. Additionally, the codes do **not** assume that there is a single file system that can be accessed by all the process. This allows them to be used on distributed memory parallel computers as well as on shared memory machines.

A key design feature of MP42 and MP62 is a straightforward user interface, with the user required to specify only a few parameters. Essentially, the user need only provide the division of the finite-element domain into subdomains and to input the element data in a suitable format. All allocation of workspace, division of work between processors, and ordering of elements is done automatically. However, for flexibility, control parameters allow the user to select a number of different options. For example, the matrix factors may optionally be held in files, enabling large problems to be solved using relatively small amounts of main memory. Additionally, the user may choose how to partition the subdomains between processors and how to order the elements. The wide range of options makes the code suitable for use both by those with minimal knowledge of the multiple front method and by expert users with specific requirements.

Experiments have been performed on a model finite-element problem and on groundwater flow problems supplied by AEA Technology. The experiments were performed on an SGI Origin 2000 (a shared memory machine) and on a Cray T3E. Results for the groundwater flow problems on an Origin 2000 are presented in Table 2.7. The first problem is a 2 dimensional problem with 40000 square elements; problems 2 and 3 are 3 dimensional with 27000 and 125000 8-noded cubic elements, respectively. In the reported experiments, 4 subdomains were used. In all our numerical experiments, we achieved good speedups although, because the number of interface variables is proportionally higher for the 3 dimensional problems, the speedups for the factor times for these problems was not quite as good as for 2 dimensional problems.

Table 2.7: Wall clock timings (in seconds) for MP42 on 1, 2, and 4 for three groundwater flow problems.

Problem	Number of variables	Number of elements	No. of processes	Factor+ Solve	Speedup	Solve	Speedup
1	159999	40000	1	138	-	11.3	-
			2	77	1.8	6.2	1.8
			4	42	3.3	3.6	3.1
2	29785	27000	1	204	-	3.2	-
			2	125	1.6	2.0	1.6
			4	85	2.4	1.2	2.6
4	132651	125000	1	5823	-	48	-
			2	3560	1.6	28	1.7
			4	2050	2.8	15	3.2

A limitation of our new codes is that the interface problem is currently solved by a frontal scheme on a single process. As the number of subdomains increases so does the number of interface variables and the solution of the interface problem can become a serious bottleneck. In the future, we plan to look at solving the interface problem using other sparse direct solvers (such as the Harwell Subroutine Library multifrontal code MA41). An alternative approach is to assemble the local Schur complements and treat the resulting system as a dense system that can be solved using (for example) ScaLAPACK routines (see http://www.netlib.org/scalapack/). The design of MP42 and MP62 using library subroutines as building blocks should allow us to try different solvers for the interface problem within the existing code.

References

J. A. Scott. The design of a parallel frontal solver. Technical Report RAL-TR-99-075, Rutherford Appleton Laboratory, 1999.

2.7 The use of MA27 in a domain decomposition context (I. S. Duff, J. Koster and J. A. Scott)

In this study, we considered the use of the sparse direct HSL code MA27 in the solution of large positive definite systems that are subdivided into several 'subdomains'. This is an extension to our work on implementing MA42 in parallel (Duff and Scott, 1994). In this domain decomposition approach, the overall matrix can be ordered into bordered block diagonal form where the diagonal blocks (except the last) correspond to the internal variables of the local subproblems, the border columns to the boundary variables of the local subproblems and the last diagonal block to the reduced problem on these interface variables.

This work was carried out partly at the request of NEC in Stuttgart, who had bought a licence for MA27 but wanted to make a version available that could utilize the parallelism of their NEC SX-4 supercomputer. In particular, they had some applications in the automotive industry on which they wanted to test this approach.

The submatrices on the diagonal (except the last) corresponding to the local problems on the subdomains are factorized using MA27. Of course, these factorizations can be performed in parallel and require no communication. The factors are then used to update the contribution of this subdomain to the Schur complement matrix, which is the coefficient matrix for the interface problem. We also use MA27 to factor the block sparse matrix corresponding to the interface problem and to obtain the solution for the interface variables. These are then substituted into the boundary values of the local problems, and finally the solution to the overall problem is obtained by backsubstitution within the local subdomain problems. A test code was written using MPI for message passing.

A crucial part of this approach and one for which modifications to MA27 are needed is in the update of the Schur complement matrix. One approach would be to provide MA27 with a 2 by 2 block matrix with the off-diagonal block corresponding to border columns and the (2,2) block to the contributions to the Schur complement from this local problem. This would involve changing the interface to MA27 to prevent pivoting on the boundary variables, and we chose not to do this at this stage although it would preserve sparsity in the boder columns and the Schur complement. Another approach would be to pass only the diagonal block and develop a solution routine for a sparse right-hand side (in fact a sparse forward substitution) but this was not possible within the time scales demanded by our collaborator. Instead, we removed some of the worst inefficiencies and increased greatly the parallelism of the method by designing a code for MA27 to perform forward and back substitutions on several right-hand sides at once by parallelizing across the right-hand sides.

References

I. S. Duff and J. A. Scott. The use of multiple fronts in Gaussian elimination. *In* J. G. Lewis, ed., 'Proceedings of the Fifth SIAM Conference on Applied Linear Algebra', pp. 567–571, Philadelphia, 1994. SIAM Press.

2.8 A multifrontal approach for shared memory computers for sparse QR factorization. (P. R. Amestoy, I. S. Duff and C. Puglisi)

We have designed and implemented a parallel QR decomposition algorithm for a large sparse matrix A. The algorithm is based on the multifrontal approach and makes use of Householder transformations. The tasks are distributed among processors according to an assembly tree which is built from the symbolic factorization of the matrix A^TA (Amestoy, Duff and Puglisi, 1996).

We first addressed uniprocessor issues and then considered the multiprocessor implementation of the method. We considered the parallelization of both the factorization phase and the solve phase. We used relaxation of the sparsity structure of both the original matrix and the frontal matrices to improve the performance. We found that, in this case, the use of Level 3 BLAS led to very significant gains in performance.

We have written a code MA49 implementing this algorithm. We have tested the code extensively on the CRAY J90 and the SGI Origin, including runs on large test problems from animal breeding. A version of the code has been developed that uses Open-MP, and early tests with this have indicated that we can have good portability and efficiency over a range of shared memory computers.

References

P. R. Amestoy, I. S. Duff, and C. Puglisi. Multifrontal QR factorization in a multiprocessor environment. *Numerical Linear Algebra with Applications*, **3**(4), 275–300, 1996.

3 Other numerical linear algebra

3.1 Development of kernels for dense and sparse numerical linear algebra (I. S. Duff)

Matrix-matrix multiplication (of dense matrices) can be performed at often close to peak rates on a wide range of computers although a highly tuned code is sometimes required. This is because the regular structures involved allow various forms of partitioning and blocking that can be used to exploit caches, vector registers, or parallelism to ensure that the normal bottleneck of high performance computing, namely access to memory, is avoided. Daydé and Duff have designed versions of the Level 3 BLAS kernel for matrix-matrix multiply (GEMM), using only Fortran, so that it performs well on RISC based computers. They have also designed the other Level 3 BLAS routines so that they can use either this GEMM kernel or one supplied by the computer vendor. Their paper on this work has recently been accepted for publication in ACM Trans Math Software (Daydé and Duff, 2000). Their intention is to develop these codes further when new architectures appear. The current codes are freely available from the anonymous ftp site ftp.enseeiht.fr in directory pub/numerique/BLAS/RISC.

Duff has also been involved in discussions on the design of basic Linear Algebra Subprograms for Level 2 and Level 3 kernels for sparse matrices. A paper on User Level codes has appeared in ACM Trans Math Software (Duff, Marrone, Radicati and Vittoli, 1997) and ideas from this paper have influenced the design of the kernels within the BLAS being developed by the BLAS Technical forum `http://www.netlib.org/cgi-bin/checkout/blast/blast.pl`. A key part of this design is the idea of matrix handles so that the user need not be concerned with the details of the storage schemes for the sparse matrix. It is envisaged that these kernels will be widely used in the solution of sparse equations by iterative methods. In collaboration with a summer student at CERFACS, Laurent Sutra, and a PhD student, Christof Voemel, Iain has developed a Fortran 95 instantiation of the sparse BLAS for the BLAS Technical forum project.

References

M. J. Daydé and I. S. Duff. A blocked implementation of Level 3 BLAS for RISC processors. *ACM Trans. Math. Softw.*, to appear, 2000.

I. S. Duff, M. Marrone, Giuseppe Radicati, and Carlo Vittoli. Level 3 Basic Linear Algebra Subprograms for sparse matrices: a user level interface. *ACM Trans. Math. Softw.*, **23**(3), 379–401, 1997.

3.2 Permuting large entries to the diagonal (I. S. Duff and J. Koster)

We have continued the development of several algorithms based on bipartite weighted matching algorithms for permuting a matrix so that the entries on the diagonal of the permuted matrix are large relative to the off-diagonal entries. We have also implemented a scaling with one of the options which gives a unit diagonal with no off-diagonal entries larger than one. This work is based on earlier work reported in Duff and Koster (1999*a*) and is described in detail in Duff and Koster (1999*b*). A highly efficient Fortran code `MC64` has been written to implement this algorithm.

We have experimented with using these algorithms to preorder matrices prior to using various factorization and solution schemes and have sometimes found a dramatic effect. For example, when used with the `MA41` multifrontal code, far fewer operations and considerably less storage are needed for the factorization of unsymmetric systems because the multifrontal factorization does not perturb the initial selection of pivots by so much. Other researchers, for example Li and Demmel (1998), have used our code so that they can predetermine a pivot sequence and avoid subsequent dynamic scheduling in a parallel implementation of their supernodal factorization.

We have also found that the robustness and performance of preconditioned iterative solvers can be greatly improved by using `MC64` as a preprocessing step. For example, the scaling option in the code has a profound influence on the convergence of the Block Cimmino iterative scheme. Some problems do not converge without the reordering and scaling but require only a few iterations if such preprocessing is performed. Benzi (Los Alamos National Laboratory), Haws (LANL) and Tuma (Czech Academy of Sciences) have experimented with our orderings as a preprocessing step in the solution of indefinite and nonsymmetric linear systems. They confirmed our findings that the convergence of Krylov subspace methods preconditioned with standard incomplete factorizations can be greatly improved. They observed similar gains for other preconditioning techniques such as factored sparse approximate inverses.

References

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 Appl.*, **20**(4), 889–901, 1999*a*.

I. S. Duff and J. Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. Technical Report RAL-TR-1999-030, Rutherford Appleton Laboratory, 1999*b*.

X. S. Li and J. W. Demmel. Making sparse Gaussian elimination scalable by static pivoting. *in* 'Proceedings of Supercomputing', Orlando, Florida, November 1998.

3.3 Linear systems and optimization (N. I. M. Gould, M. E. Hribar, C. Keller, J. Nocedal and A. J. Wathen)

The approximate minimization of a quadratic function $\frac{1}{2}\langle x, Hx\rangle - \langle c, x\rangle$ subject to a set of linear constraints $Ax = 0$ is a key computation in the numerical solution of general nonlinear programming problems. As a critical point for this problem must satisfy the Lagrange equations

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}$$

for some multipliers y, much of our recent work has been on the solution of symmetric linear systems whose coefficient matrix involves a diagonal zero block. In addition, since we are interested in solving large problems for which a direct matrix factorization may be too expensive, and since it is often not necessary to obtain a highly accurate solution to the system, we have concentrated on preconditioned iterative methods.

We have considered applying the conjugate gradient method to the solution of these structured problems. We have shown that, so long as one is prepared to precondition each iteration by solving an auxiliary linear system of the form

$$\begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix},$$

it is as if the conjugate gradient method had been applied in the null-space of the constraints $Ax = 0$, and thus we could appeal to existing convergence estimates to show that this is a viable approach. Traditionally, the conjugate gradient method is only appropriate if the coefficient matrix is definite, while here it is the use of an indefinite preconditioner which leads us to the same conclusions for our indefinite system. There is considerable flexibility in choices for M. At one extreme, picking M as the identity matrix leads to a preconditioning system of the form usual when solving least-squares problems (and for which there are often good direct solvers), while at the other, picking $M = H$ leads to convergence in a single iteration (since the preconditioning system is then identical to the original). In Keller, Gould and Wathen (1999), we examined the spectrum and form of the eigenvectors of the preconditioned matrix and its minimum polynomial, and indicated how the choice of M affects these values. However, in Gould, Hribar and Nocedal (1998), we observed that in practice the conjugate-gradient iterates sometimes wander away from the constraints $Ax = 0$ unless care is taken. To combat this possibility, we proposed iterative refinement techniques, as well as an adaptive reformulation of the quadratic problem, that can greatly reduce these errors without incurring a high computational overhead. These techniques have subsequently proved to be most effective in practice.

A similar approach has been considered by Gould (1999) for the related problem of minimizing the penalty function $\frac{1}{2}\langle x, Hx\rangle - \langle c, x\rangle + \frac{1}{2}\|Ax\|_2^2/\mu$, for some small scalar $\mu > 0$. In this case, a critical point must satisfy

$$\begin{pmatrix} H & A^T \\ A & -\mu I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix},$$

which may be regarded as a small perturbation of the system considered earlier, and conjugate gradient methods using preconditioners of the form

$$\begin{pmatrix} M & A^T \\ A & -\mu I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}$$

are both appropriate and have the beneficial effect of removing the dominant form of ill-conditioning associated with penalty-function methods. As before, these methods are made more stable using a restricted form of iterative refinement, and numerical results have indicated the effectiveness of such an approach.

Of course, neither of these approaches are appropriate if the objective functions are unbounded from below (on their constraint sets). In this case, it is now traditional to impose a normalization constraint of the form $\|x\| \leq \Delta$, for some appropriate $\Delta > 0$. In Gould, Lucidi, Roma and Toint (1999), we investigated conjugate gradient-like methods for solving these problems when the ℓ_2 norm is used for the normalization, but in the absence of the linear constraints $Ax = 0$. Fortunately, by combining these methods with the abovementioned preconditioners, it is now possible to handle the linearly constrained case as well, and this has important, positive repercussions for our new quadratic programming code HSL_VE12.

References

N. I. M. Gould. Iterative methods for ill-conditioned linear systems from optimization. *In* G. Di Pillo and F. Gianessi, eds, 'Nonlinear Optimization and Applications 2', pp. 123–142, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic problems arising in optimization. Technical Report RAL-TR-98-069, Rutherford Appleton Laboratory, 1998.

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.

C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. Technical Report RAL-TR-1999-016, Rutherford Appleton Laboratory, 1999. To appear in *SIAM J. Matrix Analysis and Applications.*

3.4 Incomplete QR factorizations (Z.-Z. Bai, I. S. Duff and A. J. Wathen)

We have studied a class of incomplete orthogonal factorization methods based upon Givens rotations for the solution of large sparse nonsingular and unsymmetric systems of linear equations. These methods include: Incomplete Givens Orthogonalization (IGO-method) and Generalized Incomplete Givens Orthogonalization (GIGO-method), which drop entries from the incomplete orthogonal and upper triangular factors by position; Threshold Incomplete Givens Orthogonalization (TIGO(τ)-method), which drops entries dynamically by their magnitudes; and Generalized Threshold Incomplete Givens Orthogonalization (GTIGO(τ, p)-method), which drops entries dynamically by both their magnitudes and positions. Theoretical analyses show that these methods can produce a nonsingular sparse incomplete upper triangular factor and either a complete orthogonal factor or a sparse nonsingular incomplete orthogonal factor for a general nonsingular matrix. Therefore, these methods can potentially generate efficient preconditioners for Krylov subspace iterations for solving large sparse systems of linear equations. Moreover, the upper triangular factor is an incomplete Cholesky factorization preconditioner for the normal equations from least-squares problems. We study incomplete Givens orthogonalization methods in detail for a tridiagonal matrix to illustrate more clearly the behaviour of our new methods.

Full details of this work are given in the technical report by Bai, Duff and Wathen (1999).

References

Z.-Z. Bai, I. S. Duff, and A. J. Wathen. A class of incomplete orthogonal factorization methods. I: Methods and theories. Technical Report RAL-TR-1999-045, Rutherford Appleton Laboratory, 1999. Also Report 99/13 from OUCL, Oxford University.

3.5 Locking and restarting quadratic eigenvalue solvers (K. Meerbergen)

The quadratic eigenvalue problem

$$Ku + i\omega Cu - \omega^2 Mu \;=\; 0$$

arises in the simulation of acoustic cavities with acoustic damping material. The real part of the eigenvalue ω is the eigenfrequency and the imaginary part represents the amount of damping of the corresponding mode u. This enables engineers to study the acoustic properties of cavities with damping material. The traditional approach is to apply Arnoldi's method to the linearized problem

$$\begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} \begin{pmatrix} u \\ \omega u \end{pmatrix} = \omega \begin{bmatrix} -iC & M \\ M & 0 \end{bmatrix} \begin{pmatrix} u \\ \omega u \end{pmatrix}$$

which we also denote as $Ax = \omega Bx$, but this doubles the size of the problem and does not use its special structure. Methods such as residual iteration (Neumaier, 1998), rational Krylov (Huitfeldt and Ruhe, 1990) and Jacobi-Davidson (Sleijpen, van der Vorst and van Gijzen, 1996) solve the quadratic eigenvalue problem without linearization. The major difficulty here is how to compute more than one eigenvalue, since the quadratic Schur form is not guaranteed to exist. In Meerbergen (1999), we propose building a subspace using quadratic residual iteration, and use this subspace to build a partial Schur factorization of the linearized problem. Quadratic residual iteration builds a basis $V_k = [v_1, \ldots, v_k]$ by repeatedly applying for $j = 1, \ldots, k-1$ the transform

$$y_j = (K + i\sigma_j C - \sigma_j^2 M)^{-1}(K + i\omega_j C - \omega_j^2 M)u_j$$

to an approximate eigenvector u_j, which we call a Ritz vector, with corresponding Ritz value ω_j. The result y_j is added to V_j to construct V_{j+1}. We use this subspace to form an approximate partial Schur form for the linearized problem, that is we compute $U_k \in$ Range(V_k) and an upper triangular matrix S_k with the eigenvalues on the main diagonal so that

$$\begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} \begin{pmatrix} U_k \\ U_k S_k \end{pmatrix} - \omega \begin{bmatrix} -iC & M \\ M & 0 \end{bmatrix} \begin{pmatrix} U_k \\ U_k S_k \end{pmatrix} S_k = \begin{pmatrix} F_k \\ 0 \end{pmatrix}$$

where $V_k^* F_k = 0$. We use this Schur form for locking, purging and restarting quadratic residual iteration.

We now compare several methods on an application related to the acoustic simulation of a $0.4m \times 0.4m \times 0.06m$ sample made of a poro-elastic material. A full description of the problem can be found in Meerbergen (1999). The finite-element mesh has 324 nodes and 192 HEXA8 elements. The total number of degrees of freedom is 1944.

We performed 30 steps of the shift-invert Arnoldi method starting with a random initial vector and pole (or shift) $\sigma = 300$. After 18 iterations, Arnoldi's method computed six Ritz values with residual norm $\rho_j = \|Ku_j + i\omega_j C u_j - \omega_j^2 M u_j\|$ smaller than 10^{-4} and after 20 iterations, six Ritz values had ρ_j smaller than 10^{-7}. When we use quadratic residual iteration with $\sigma = 300$, we need only 14 iterations for computing six eigenvalues with residual norms smaller than 10^{-4}, which is faster than the shift-invert Arnoldi method, but we need 30 iterations to compute six eigenvalues with residual norm smaller than 10^{-7} which is slower than Arnoldi's method. So it is difficult to say which method is best. We noticed that the convergence of Ritz values in the quadratic residual iteration method is less smooth than in Arnoldi's method, which may lead to slower convergence when a number of eigenvalues with small residual norms are wanted. In addition, the method focuses convergence on one eigenvalue at a time. This may lead to some minor gain in convergence speed when one eigenvalue is of interest. The major interest of the method is the gain in storage of the iteration vectors.

The Jacobi-Davidson method for the quadratic eigenvalue problem can also be combined with the computation of the Schur form of the linearized problem. The major difference is that y_j is now computed from the correction equation

$$\left(I - \frac{(iCu_j - 2\omega_j M u_j)u_j^*}{u_j^*(iCu_j - 2\omega_j M u_j}\right)(K + i\omega_j C - \omega_j M)\left(I - \frac{u_j u_j^*}{u_j^* u_j}\right) y_j = Ku_j + i\omega_j Cu_j - \omega_j^2 M u_j \ .$$

For our example, we solved the correction equation with GMRES preconditioned with $K + i\sigma C - \sigma^2 M$ with $\sigma = 300$ in order to make a fair comparison with the shift-invert Arnoldi method and quadratic residual iteration. In this case, it took Jacobi-Davidson only 13 iterations to compute six eigenvalues with residual norms smaller than 10^{-7}, but the total number of linear solves with $K + i\sigma C - \sigma^2 M$ is 118, which is far higher than for the other methods.

References

J. Huitfeldt and A. Ruhe. A new algorithm for numerical path following applied to an example from hydrodynamical flow, *SIAM J. Sci. Statist. Comput.* **11**(6), 1181–1192, 1990.

K. Meerbergen. Locking and restarting quadratic eigenvalue solvers, Technical Report RAL-TR-1999-011, Rutherford Appleton Laboratory, 1999.

A. Neumaier. Residual inverse iteration for the nonlinear eigenvalue problem, *SIAM J. Numer. Anal.* **22**, 914–923, 1998.

G. Sleijpen, H. van der Vorst and M. van Gijzen. Quadratic eigenproblems are no problem, *SIAM News* **29**(7), 8–9, 1996.

3.6 Dangers in changing poles in the rational Lanczos method for the Hermitian eigenvalue problem (K. Meerbergen)

The spectral transformation Lanczos method (Ericsson and Ruhe, 1980) is very popular for solving the eigenvalue problem $Ax = \lambda Bx$ where A and B are $n \times n$ symmetric matrices and B is positive (semi) definite. This method builds a Krylov space

$$\mathcal{K}_k = \text{span}\{v_1, Sv_1, S^2v_1, \ldots, S^{k-1}v_1\}$$

for the spectral transformation $S = (A - \mu B)^{-1}B$. Typically, the eigenvalues near the pole $\mu \in \mathbf{R}$ converge very quickly. It may happen that some wanted eigenvalues are far away from μ and converge very slowly. Hence it may be advantageous to alter μ from time to time. However, this requires building a new subspace with a new spectral transformation $(A - \nu B)^{-1}B$ and the old Krylov subspace is entirely thrown away. When the rational Krylov method (Ruhe, 1998) is used, μ can change at any time without throwing away the Krylov subspace. Unfortunately, the rational Krylov method does not use the symmetry of the eigenvalue problem. Therefore, we propose the following method (Meerbergen, 1999).

1. Choose a pole $\mu \in \mathbf{R}$ and perform k iterations of the Lanczos method. This produces a tridiagonal matrix $T_k \in \mathbf{C}^{k\times k}$ and a set of k basis vectors $V_k = [v_1, \ldots, v_k] \in \mathbf{C}^{n\times k}$ from which the eigenvalues and eigenvectors are computed.

2. Choose a new pole $\nu \in \mathbf{R}$. Compute two $k \times k$ matrices Q and M so that $QM = I + (\mu - \nu)T_k$ where Q is unitary and $K_k = Q^*T_kM^{-1}$ is tridiagonal. Perform a change of basis $W_k = [w_1, \ldots, w_k] = V_kQ$. The new basis W_k and tridiagonal matrix K_k correspond to a Lanczos process for $(A - \nu B)^{-1}B$ with starting vector w_1.

3. Continue the Lanczos method with pole ν with k additional iterations to obtain W_{2k} and K_{2k}.

We call this algorithm the rational Lanczos method since it is a combination of the Lanczos method and the rational Krylov method. The only difference with the Lanczos method lies in the change of pole, so we can inherit all Lanczos technology including partial reorthogonalization (Grimes, Lewis and Simon, 1994), implicit restarting (Sorensen, 1992), locking and purging (Lehoucq and Sorensen, 1996).

An important caveat is in order. It is essential not to pick the pole very close to an eigenvalue. (In practice, we pick the pole in between clusters of computed eigenvalues.) The reasons are the following:

- The spectral transformation $(A - \mu B)^{-1}B$ is applied to a sequence of vectors v_j for $j = 1, \ldots, k$ and requires the solution of the linear system $(A - \mu B)y_j = Bv_j$ for y_j. Usually, $A - \mu B$ is factorized once and the linear systems are solved using backtransformations. The solution y_j becomes less accurate when μ is close to an eigenvalue since the condition number becomes extremely large and only the eigenvalues very close to μ will be computed.

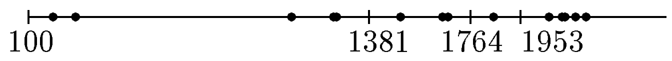


Figure 3.1: Eigenvalues computed via the rational Lanczos method. The vertical lines denote the positions of the poles.

- Changing the pole or locking converged eigenvectors may multiply the rounding errors in the Lanczos method by the factor $(\max_{\lambda \in \lambda(A,B)} |\lambda - \mu|/|\lambda - \nu|)^2$, which is large when ν is close to an eigenvalue λ.

We now illustrate the method for an application coming from acoustics. For a complete description of the problem we refer to Meerbergen (1999). This example is related to the acoustic simulation of a $0.4m \times 0.4m \times 0.06m$ sample made of a poro-elastic material. The finite-element mesh has 324 nodes and 192 HEXA8 elements. The total number of degrees of freedom is $n = 1944$.

We have experimented with a prototype of the new Lanczos routine `EA16`. We computed 25 eigenvalues to the right of 100. Figure 3.1 shows the computed eigenvalues and the different poles. The pole was changed every 25 iterations, except for the first change of pole, where 50 iterations were performed. The new poles are chosen so that $\max_{\lambda \in \lambda(T_k)} |\lambda - \mu|/|\lambda - \nu| \leq 50$. At each change of pole, the converged eigenvectors were locked and the Lanczos basis was reduced to dimension 25 by an implicit restart. All the computed Ritz pairs (θ, x) had residual norms
$\|(A - \nu B)^{-1} Bx - \theta x\|_B \leq 10^{-7}$.

References

T. Ericsson and A. Ruhe. The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems, *Math. Comp.* **35**, 1251–1268, 1980.

R. Grimes, J. Lewis and H. Simon. A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems, *SIAM J. Matrix Anal. Applic.* **15**, 228–272, 1994.

R. Lehoucq and D. Sorensen. Deflation techniques within an implicitly restarted Arnoldi iteration, *SIAM J. Matrix Anal. Applic.* **17**, 789–821, 1996.

K. Meerbergen. The rational Lanczos method for Hermitian eigenvalue problems, Technical Report RAL-TR-1999-025, Rutherford Appleton Laboratory, 1999.

A. Ruhe. Rational Krylov: a practical algorithm for large sparse nonsymmetric matrix pencils, *SIAM J. Sci. Comput.* **19**(5), 1535–1551, 1998.

D. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM J. Matrix Anal. Applic.* **13**, 357–385, 1992.

3.7 Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry. (K. Meerbergen)

This work was carried out in the framework of a consultancy for LMS International (Belgium). The purpose was to apply the FETI-H2LM domain decomposition method (Magoulès, Roux and de La Bourdonnaye, 1998) to industrial problems. The motivation was to enable the solution of large problems arising from the finite-element discretization of acoustic cavities, decoupled from their surrounding structure. The main unknown is the acoustic pressure field while boundary conditions are related to (Dirichlet) pressure constraints, (Neumann) normal pressure gradient constraints, and (Robin) normal admittance constraints.

One approach, called domain decomposition, relies on the decomposition of the entire domain into subdomains, so that the global problem is decomposed in a number of local problems, that can be solved independently. Because of this property, domain decomposition is well suited for parallel computing.

The general Helmholtz problem for the acoustic pressure u, in a bounded domain Ω with boundary conditions on $\partial\Omega$ can be written as follows: for $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$, find $u \in H^1(\Omega)$ such that

$$\begin{aligned} -\nabla^2 u - k^2 u &= f && \text{in } \Omega \\ \frac{\partial u}{\partial \nu} + \alpha u &= g && \text{on } \partial\Omega \end{aligned}$$

where k denotes the wave number, ν the unit outward normal on $\partial\Omega$, and α is a scalar.

The FETI-H2LM method can be defined for two subdomains as follows. Let the domain Ω be decomposed into two non-overlapping subdomains Ω_s, $s = 1, 2$; let Γ_I denote the interface $\Gamma_I = \partial\Omega_1 \cap \partial\Omega_2$. Then for $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$, find $u_s \in H^1(\Omega_s)$ so that

$$\begin{aligned} -\nabla^2 u_s - k^2 u_s &= f_s && \text{in } \Omega_s \\ \frac{\partial u_s}{\partial \nu_s} + \alpha u_s &= g_s && \text{on } \partial\Omega_s \cap \partial\Omega \\ \frac{\partial u_s}{\partial \nu_s} + iku_s &= \lambda_s && \text{on } \Gamma_I \end{aligned}$$

under the global constraints

$$\begin{aligned} \lambda_1 + \lambda_2 - 2iku_2 &= 0 \\ \lambda_1 + \lambda_2 - 2iku_1 &= 0 \end{aligned}$$

where f_s (resp. g_s) denotes the restriction of the function f (resp. g) in subdomain Ω_s (resp. $\partial\Omega_s \cap \partial\Omega$), and ν_s the external normal vector of Ω_s, for $s = 1, 2$. The λ's are called Lagrange multipliers and are also unknown. The elimination of the unknowns u_1 and u_2 leads to an interface system in λ_1 and λ_2, that is solved with an iterative method. The elimination of u_s requires the solution of a local Helmholtz problem on each subdomain, which is done using a direct sparse linear solver.

The example is related to a car compartment, generated by the simulation software SYSNOISE (LMS, Belgium). The main objective of this evaluation is the synthesis of the frequency response function at the driver's and passenger's ears, subject to velocity boundary conditions along the firewall. This example is representative of a wider class of problems where the acoustic response within a cavity, induced by vibrating panels, is evaluated. The evaluation of the acoustic response is performed using a three-dimensional finite-element model as presented in Figure 3.2. The discrete model involves 6448 hexahedral elements, 544 pentahedral elements, and 8417 nodes. See Magoulès, Meerbergen and Coyette (1999) for the details. The first mesh partitioning consists of two subdomains ($N_s = 2$) each with 4429 degrees of freedom (dofs) and the interface with 441 dofs. The second partition into four subdomains ($N_s = 4$) consists of two subdomains with 2443 dofs and two with 2229 dofs; the interface has 955 dofs. The order of the interface problem for the FETI-H2LM method is double the size of the interface, because of the use of two Lagrange multipliers. This leads to an interface problem of order 882 for the first partition and to one of order 1910 for the second.

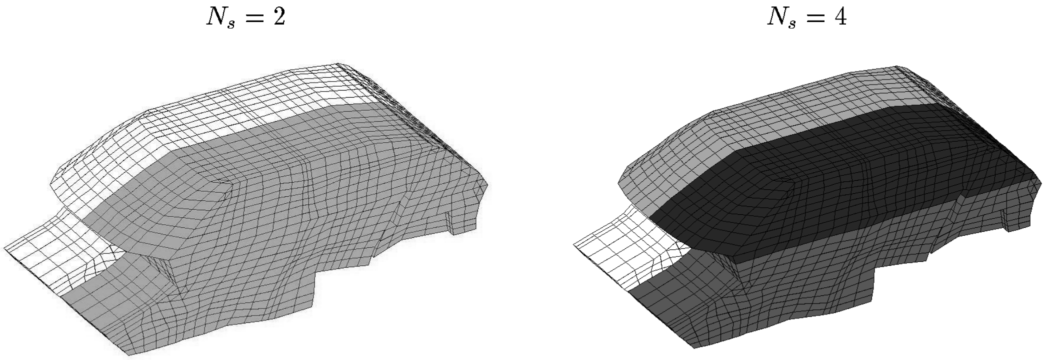
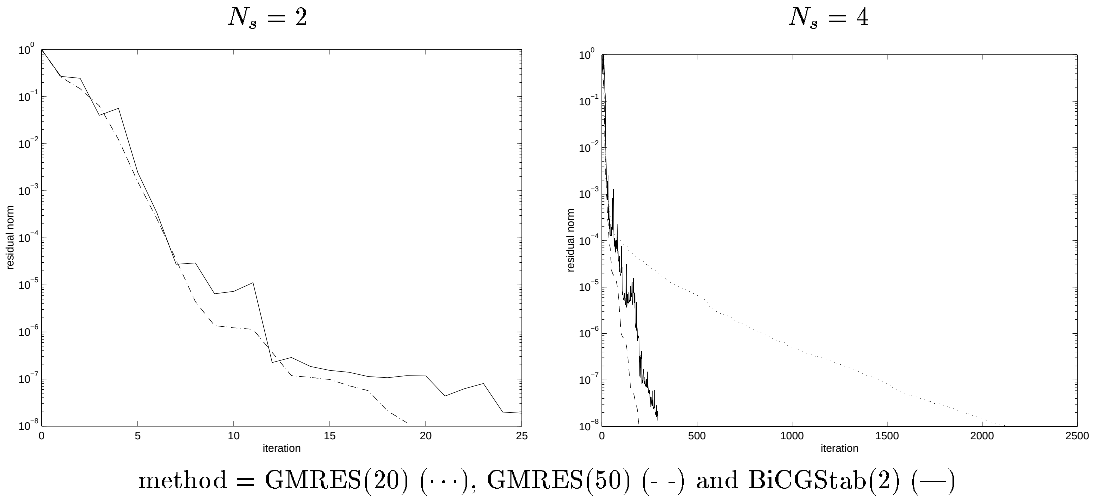


Figure 3.2: Car compartment — domain decompositions

The convergence curves obtained for the two partitions for 100 Hz are shown in Figure 3.3. The curves show the scaled interface residual norm versus the iteration number. For all cases, GMRES(50) is the fastest in terms of iterations. Note that the figures do not represent the cost per iteration. Indeed, GMRES(m) requires more (parallel) inner products than BiCGStab(2), which can be a significant overhead. However, GMRES(m) requires only one matrix-vector product per iteration while BiCGStab(2) requires two. Another difference between GMRES(m) and BiCGStab(2) is the convergence behaviour: GMRES(m) converges smoothly, while BiCGStab(2) has a somewhat irregular behaviour. It can be seen that the decomposition into four domains leads to a more difficult problem, since each of the methods requires more iterations. This is very pronounced for GMRES(20), which requires more than 2000 iterations.



$N_s = 2$ $N_s = 4$

method = GMRES(20) ($\cdots$), GMRES(50) (- -) and BiCGStab(2) (—)

Figure 3.3: Car compartment — convergence history for 100 Hz

References

F. Magoulès, K. Meerbergen and J.-P. Coyette. Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry, Technical Report RAL-TR-1999-039, Rutherford Appleton Laboratory, 1999.

F. Magoulès, F. Roux, and A. de La Bourdonnaye. Méthode de décomposition de domaines pour des problèmes hyperboliques, *Calculateurs Parallèles Réseaux et Systèmes répartis* **10**(4), 353–361, 1998.

3.8 The design of a block Lanczos and rational Lanczos code for the symmetric eigenvalue problem (K. Meerbergen and J.A. Scott)

We are developing a Fortran code for the block Lanczos method and the rational Lanczos method for the solution of large-scale real symmetric eigenvalue problems. The code is designed for the computation of a selected number of eigenvalues and the corresponding eigenvectors for the following applications :

- the standard eigenvalue problem $Ax = \lambda x$, with A symmetric ;
- the generalized eigenvalue problem $Ax = \lambda Mx$, with A and M symmetric and M positive (semi) definite ;
- the generalized eigenvalue problem $Ax = \lambda Mx$, with A and M symmetric and A positive semi-definite, referred to as the buckling problem.

We allow the computation of a number of eigenvalues nearest a point, furthest from a point, inside an interval, to the right or the left of a point, the left-most or right-most eigenvalues, or the eigenvalues on both ends of the spectrum. The new code is called EA16.

The code can be used in regular, shift-invert, or buckling modes. In regular mode, a Krylov subspace with A (or $M^{-1}A$) is built; in shift-invert mode, the code builds a Krylov subspace with $(A-\sigma I)^{-1}$ (or $(A-\sigma M)^{-1}M$), where σ is the pole; and in buckling mode, it builds a Krylov subspace with $(A-\sigma M)^{-1}A$. The major computations are matrix-vector operations with A, $(A-\sigma I)^{-1}$, $M^{-1}A$, $(A-\sigma M)^{-1}M$, or $(A-\sigma M)^{-1}A$. These are performed by the user through a reverse communication interface. The pole may be chosen either by the code or by the user. For greater flexibility, the user can start the computation using the regular mode then, once a few approximate eigenvalues have been found that enable a suitable σ to be chosen, the user may switch to shift-invert mode to speed up the convergence.

We have spent some effort to understand partial reorthogonalization (Grimes et al. 1994) in combination with an implicit restart and change of pole (rational Lanczos).

When there is no storage left to expand the Lanczos basis, the code uses an implicit restart to reduce the subspace dimension. We have adopted a number of different shift choices proposed in the literature, including exact shifts (Sorensen, 1992), Leja shifts (Baglama, Calvetti and Reichel, 1998), Chebyshev shifts and purging (Lehoucq and Sorensen, 1996).

The major differences between EA16 and other symmetric eigenvalue codes are summarised in the following table.

code	cheap orthogonalization	implicit restart	blocking	rational Krylov	automatic pole selection
EA15	×				
ARPACK		×			
BLZPACK	×		×		×
Boeing	×		×		×
EA16	×	×	×	×	×

Hence we are providing a state-of-the-art code by employing the latest research on the Lanczos method and the rational Krylov method.

In addition, we have included a number of options to improve the reliability of the code when used for the solution of the Stokes and buckling problems. These include starting the Lanczos method with a pre-filtered initial vector, the computation of purified Ritz vectors (Meerbergen and Spence, 1997), and the implicit filtering of unwanted subspaces (Meerbergen and Spence, 1997).

References

J. Baglama, D. Calvetti and L. Reichel. Fast Leja points, *ETNA* **7**, 124–140, 1998.

R. Grimes, J. Lewis and H. Simon A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems, *SIAM J. Matrix Anal. Applic.* **15**, 228–272, 1994.

R. Lehoucq and D. Sorensen. Deflation techniques within an implicitly restarted Arnoldi iteration, *SIAM J. Matrix Anal. Applic.* **17**, 789–821, 1996.

K. Meerbergen. The rational Lanczos method for Hermitian eigenvalue problems, Technical Report RAL-TR-1999-025, Rutherford Appleton Laboratory, 1999.

K. Meerbergen and A. Spence Implicitly restarted Arnoldi and purification for the shift-invert transformation, *Math. Comp.*, **66**, 667–689, 1997.

D. Sorensen Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM J. Matrix Anal. Applic.* **13**, 357–385, 1992.

3.9 The computation of the right-most eigenvalues of the linearized and discretized Navier-Stokes equations (K. Meerbergen and J.A. Scott)

The determination of the stability of steady state solutions and the detection of Hopf bifurcations of the Navier-Stokes equations is a challenging problem. In a classical analysis, time integration is used: a random perturbation of the steady-state solution is used as the initial value and the solution after a large number of time steps is compared with the steady state. If this solution is close to the steady state, the latter is stable, otherwise it is unstable. Because the Navier-Stokes equations form a system of differential-algebraic equations (an index 2 DAE), special techniques are required to ensure stable integration. An alternative approach is to compute the right-most eigenvalues of $Ax = \lambda Bx$, where A is the Jacobian matrix of the discretized Navier-Stokes equations and B is the mass matrix. When an eigenvalue crosses the imaginary axis, the solution becomes unstable. If the right-most eigenvalues form a complex conjugate imaginary pair, then we have a Hopf bifurcation. It is thus sufficient to monitor the sign of the right-most eigenvalue of $Ax = \lambda Bx$ for a branch of steady-state solutions. Lehoucq (Sandia Laboratories, USA), Meerbergen, and Scott have considerable experience of computing right-most eigenvalues (Meerbergen and Roose, 1996, Lehoucq and Scott, 1997) while Wathen (Oxford University) has experience of preconditioning the linear systems that arise from the discretization of the Navier-Stokes equations (Golub and Wathen, 1998, Keller, Gould and Wathen, 1999). We have embarked on a collaborative project, partially funded by the EPSRC Grant GR/M74542. At this stage, we have made an inventory of existing techniques and exchanged experience. We plan to investigate two topics.

- The costs (in terms of CPU time and memory requirements) of direct linear solvers and iterative methods for solving the linear systems that arise in the eigenvalue computations will be analysed and compared for a model problem. This will enable us to give guidelines on which method to choose for the Navier-Stokes problem in 2 and 3 dimensions.

- The computation of the right-most eigenvalue of the Navier-Stokes equations is related to time-integration of DAE's. We can show that the eigenvalue solvers developed in Cliffe, Garratt and Spence (1994) and Meerbergen and Spence (1997) correspond to the solution of an index 1 DAE. We will consider other DAE formulations and investigate whether these offer good alternatives.

Finally, Lehoucq has access to real large-scale applications. We plan to apply the conclusions of this work to these applications.

References

K. Cliffe, T. Garratt, and A. Spence. Eigenvalues of block matrices arising from problems in fluid mechanics, *SIAM J. Matrix Anal. Applic.* **15**, 1310–1318, 1994.

G. H. Golub and A. J. Wathen. An iteration for indefinite systems and its application to the Navier-Stokes equations, *SIAM J. Sci. Comput.* **19**(2), 530–539, 1998.

C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems, Technical Report RAL-TR-1999-016, Rutherford Appleton Laboratory, 1999.

R. B. Lehoucq and J. A. Scott. Implicitly restarted Arnoldi methods and eigenvalues of the discretized Navier-Stokes equations, Technical Report RAL-TR-97-058, Rutherford Appleton Laboratory, 1997.

K. Meerbergen and D. Roose. Matrix transformations for computing rightmost eigenvalues of real nonsymmetric matrices. *IMA J. Numerical Analysis*, **16**, 297–346, 1996.

K. Meerbergen and A. Spence. Implicitly restarted Arnoldi and purification for the shift-invert transformation, *Math. Comp.*, **66**, 667–689, 1997.

3.10 Techniques for the solution of sparse equations on high performance computers (I. S. Duff)

The energy of computer architects and the strength of the marketplace for high performance computers ensure that the goal posts for designing and implementing algorithms to exploit these computers keep moving.

It was thus the case that our book (Dongarra, Duff, Sorensen and van der Vorst, 1991) was becoming quite out of date, addressing high performance of a bygone era. Although many of the principles described at that time, for example the use of tuned high level Basic Linear Algebra Subprograms, still apply, it was felt that a completely new book rather than a revision was required.

Thus in Dongarra, Duff, Sorensen and van der Vorst (1998), which was published by SIAM Press in November 1998, we have not only rewritten the chapters concerned with dense and sparse equation solution but have also included new chapters on eigensolutions and on preconditioning. We now include the design of algorithms for computers with distributed memory. The book is already a best seller. A draft copy was used in a tutorial by the authors at the SIAM Annual Meeting in Stanford (July 1998) and the launch was timed to provide copies of the book to attendees at a similar tutorial in the SC'98 meeting at Orlando in November 1998. Draft versions of two chapters of the book have appeared as RAL reports (Duff, 1998, Duff and van der Vorst, 1998).

Since then, Duff and van der Vorst have developed some of the discussion on preconditioning, including many more references and historical detail, and have written a report intended for a more numerically sophisticated audience (Duff and van der Vorst, 1999). This will appear in a special millennium issue of the Journal Parallel Computing.

References

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers.* SIAM Press, Philadelphia, 1991.

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst. *Numerical Linear Algebra for High-Performance Computers.* SIAM Press, Philadelphia, 1998.

I. S. Duff. Direct methods. Technical Report RAL-TR-1998-054, Rutherford Appleton Laboratory, 1998.

I. S. Duff and H. A. van der Vorst. Preconditioning and parallel preconditioning. Technical Report RAL-TR-1998-052, Rutherford Appleton Laboratory, 1998.

I. S. Duff and H. A. van der Vorst. Developments and trends in the parallel solution of linear systems. Technical Report RAL-TR-1999-027, Rutherford Appleton Laboratory, 1999. To appear in *Parallel Computing*.

3.11 The dissemination of good practices in sparse equation solution (I. S. Duff and J. A. Scott)

We feel it is important not only to do high quality research in the solution of large sparse linear systems but also that we should disseminate our research and that of our colleagues to the community at large. Indeed we feel that much of the value of research in numerical analysis lies in its application to problems in science and engineering.

In the last year, we have given talks at meetings of computational physicists. Iain gave an invited talk at the meeting “Supercomputing, Collision Processes and Applications” held in The Queens University Belfast to commemorate the retirement of Professor Phil Burke. He spoke on a wide range of methods for the solution of linear equations and eigenvalues for both sparse and dense matrices and has produced a paper for the Proceedings (Duff, 1999*a*). Jennifer had a similar brief in her invited talk at the “Workshop on the TB-LMTO Method” in Strasbourg and her paper will be included in the conference proceedings (Scott, 2000).

Iain had quite a different audience in Helsinki where he was a keynote speaker at a meeting on “Computational Cattle Breeding”. The scientists, principally from agricultural research establishments, were concerned with determining breeding values in cattle with the intention of improving milk or beef yield. Essentially they require the solution of very large sparse least-squares problems with their current method of choice being the solution of weighted normal equations (the mixed model equations). The dimension of the MME can be in the tens of millions although use can be made of a hierarchical structure. Iain discussed methods of solving the MME including discussing preconditioning techniques which have proved useful in other application areas, notably computational fluid dynamics. He also suggested other approaches to solving their least-squares problems. He wrote a short annotated bibliography (Duff 1999*b*) which is available on the Web site for the conference proceedings `http://www.csc.fi/ttn/ccb99/program.htm`.

References

I. S. Duff. Matrix methods. *In* K. L. Bell, K. A. Berrington, D. S. F. Crothers, A. Hibbert and K. T. Taylor, eds, 'Supercomputing, Collision Processes and Applications', pp. 119–136, Kluwer Academic, New York, 1999*a*.

I. S. Duff. A brief bibliography of recent research and software for the parallel solution of large sparse linear equations. *In* 'Proceedings of the Computational Cattle Breeding '99 Workshop, Tuusula, Finland, March 18-20, 1999', Bulletin No 20, pp. 43–46, International Bull Evaluation Service, Uppsala, Sweden, 1999*b*.

J. A. Scott. Sparse direct methods: an introduction. *Springer Verlag Lecture Notes in Physics*, to appear, 2000.

4 Optimization

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

We have long recognised that our trusty, well known, large-scale optimization package LANCELOT (Conn, Gould and Toint, 1992) is showing its age. Our intention is to produce a successor, GALAHAD, within the next couple of years, but to date we have been primarily concerned with the theoretical issues which will inevitably underpin such an enterprise, and with the development of new, core quadratic programming procedures (see Section 4.3).

LANCELOT is based on the sequential minimization of an augmented Lagrangian function. This has some (particularly linear algebraic) advantages for large problems, but augmented Lagrangian methods are widely believed to be inferior to sequential quadratic programming (SQP) methods at least for small problems. For large problems, good methods for the solution of (nonconvex) quadratic programming problems are not widely available which has presumably been the reason for the limited experience with large-scale SQP methods. However, things have changed over the past few years following Karmarkar's method and the "interior-point" revolution, which has resulted in dramatic improvements in methods for both linear and convex quadratic programming. The ability to handle nonconvex problems, which we believe is essential for a general purpose optimization package, is not without difficulties, but both we (Conn, Gould and Toint, 1996, and Conn, Gould, Orban and Toint, 1999) and others have now proposed interior-point methods for nonconvex quadratic programming. Thus we believe that the main obstacle which has for so long stood in the way of large-scale SQP algorithms has been removed, and intend to experiment with such methods in the near future.

SQP methods themselves come in a variety of flavours. There are three current leading candidates for GALAHAD. Many of the theoretical issues that lie behind these are either well understood, or have recently been explored in our book (Conn, Gould and Toint, 2000) on trust-region methods, although there are still some outstanding difficulties that we hope to address. Of particular significance is the first proof of convergence of an SQP filter method for such problems (see Fletcher, Gould, Leyffer and Toint, 1999). This is particularly important because these new methods have recently been shown to be very effective in practice, and are thus front-runners for GALAHAD.

References

A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for minimizing a non-convex function subject to bound and linear equality constraints. Technical Report RAL-TR-1999-054, Rutherford Appleton Laboratory, 1999.

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.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A primal-dual algorithm for minimizing a non-convex function subject to bound and linear equality constraints. Technical Report RAL-TR-96-096, Rutherford Appleton Laboratory, 1996.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-region methods.* 966+xvii pages, to appear. SIAM, Philadelphia, USA, 2000.

R. Fletcher, N. I. M. Gould, S. Leyffer, and Ph. L. Toint. Global convergence of trust-region SQP-filter algorithms for nonlinear programming. Technical Report RAL-TR-1999-041, Rutherford Appleton Laboratory, 1999.

4.2 Trust-region methods (A. R. Conn, N. I. M. Gould and Ph. L. Toint)

Trust-region methods are one of the most popular techniques for solving nonlinear optimization problems. For the past two years, we have been writing a book, whose main aims are to describe and explore trust-region methods in all of their manifestations, to unify many of the hundreds of existing algorithms in this area, and, as a result, to develop a comprehensive theory which is capable of reproducing existing, and of providing exciting new, results. On March 31st 1999, and 966 pages later, our endeavours finally came to an end, as we passed our manuscript (Conn et al., 2000) to our publishers, SIAM Press in Philadelphia. The resulting book is set to appear in time for the Mathematical Programming Society's triennial symposium in August, 2000.

A typical method for nonlinear optimization aims to replace the presumably hard original problem by an easier model of this problem—by easier, we mean a model that is possible to (approximately) solve at reasonable cost. For example, if we wish to find an unconstrained minimizer of a nonlinear function, we might replace it by a locally-accurate quadratic approximation based on a current estimate of the required minimizer. Next we might minimize the quadratic model, and use the minimizer of this as an improved estimate of the required minimizer of the original problem. However, two difficulties might arise. Firstly, the model problem may not have a minimizer. Secondly, even if it does, the minimizer of the model may provide a worse estimate of the solution to the problem than the current estimate.

To avoid these difficulties, a *trust-region* method adds an extra requirement, the trust-region bound, on the model. The trust-region bound is simply that the distance between the current estimate and the minimizer of the resulting model problem should be bounded by a prescribed positive value, known as the trust-region radius. Since the solution to the model is now required to lie within the trust-region, the model minimizer must exist. Furthermore, the trust-region radius provides a mechanism for discarding a poor predicted estimate of the solution to the original problem. If the predicted value is poor, the radius should simply be reduced and the model problem resolved. On the other hand, if there is good agreement between the model and the original problem, there may be some justification for increasing the radius for the next iteration.

This simple idea is the basis for all of the methods considered in our book. Considerable efficiencies are possible because the conditions actually required of an approximate model minimizer are very weak. Moreover, such conditions are satisfied by most practical methods for approximately minimizing the model. Trust-region methods are particularly appealing since they mix powerful convergence properties in theory with excellent numerical performance in practice. The book is arranged as follows:

Part I: Preliminaries.

Chapter 1: Introduction, Chapter 2: Basic Concepts, Chapter 3: Basic Analysis and Optimality Conditions, Chapter 4: Basic Linear Algebra, and Chapter 5: Krylov Subspace Methods.

Part II: Trust-region Methods for Unconstrained Optimization.

Chapter 6: Global Convergence of the Basic Algorithm, Chapter 7: The Trust-Region Subproblem, Chapter 8: Further Convergence Theory Issues, Chapter 9: Conditional Models, Chapter 10: Algorithmic Extensions, and Chapter 11: Non-Smooth Problems.

Part: III Trust-region Methods for Constrained Optimization with Convex Constraints.

Chapter: 12: Projection Methods for Convex Constraints, and Chapter: 13: Barrier Methods for Inequality Constraints.

Part: IV: Trust-region Methods for General Constrained Optimization and Systems of Nonlinear Equations.

Chapter: 14: Penalty-Function Methods, Chapter: 15: Sequential Quadratic Programming Methods, and Chapter: 16: Nonlinear Equations and Nonlinear Fitting.

Part V: Final Considerations.

Chapter: 17: Practicalities, Chapter: 18: Afterword, and a comprehensive, annotated bibliography.

References

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-region methods.* 966+xvii pages, to appear. SIAM, Philadelphia, USA, 2000.

4.3 Quadratic Programming (N. I. M. Gould and Ph. L. Toint)

We have recently been developing two new quadratic programming methods that lie at the heart of our forthcoming nonlinear programming solver GALAHAD. Both algorithms are designed to handle large, sparse, nonconvex problems, the ultimate aim being to solve problems in hundreds of thousands of unknowns.

Our first method is based on a feasible-interior-point trust-region approach. At each outer iteration, an appropriate model of the logarithmic barrier function is minimized within the intersection of linear equality constraints and an appropriately-shaped ellipsoidal trust-region. The solution to this subproblem is (approximately) solved using a preconditioned Lanczos/conjugate gradient-based approach (see Gould et al., 1998 and Gould et al., 1999), in which some attempt is made to move around the boundary of the trust region in an attempt to find a better boundary solution, if such a solution occurs. The preconditioner used aims both to ensure that linear equality constraints remain satisfied, and to mimic any ill-conditioning resulting from the barrier model. This has consequences for both the shape of the trust-region, and for the underlying convergence theory. The theoretical justification of the overall scheme, for problems with general objectives and inequality constraints, is given by Conn et al. (1999). The method has been implemented as a Fortran 90 module HSL_VE12 in the Harwell Subroutine Library, and we have presented numerical results that suggest that it is indeed able to solve some problems of the size we had been aiming for. Most recently, we have been investigating the ultimate rate of convergence of such schemes, and have shown that, under fairly general conditions, a superlinear rate is achievable both for quadratic and general nonlinear programs (see Gould, Orban, Sartenaer and Toint, 2000).

Our second method (Gould and Toint, 2000) is of the active-set variety, and, although general in scope, is intended within GALAHAD to deal with the case where a good estimate of the optimal active set has been determined (and thus that relatively few iterations will be required). The method is iterative at two levels, one level relating to the selection of the current active set, and the second due to the method used to solve the equality-constrained problem for this active set. A preconditioned conjugate gradient method is used for this inner iteration, once again with the preconditioner chosen especially to ensure feasibility of the iterates. The preconditioner is updated at the conclusion of each outer iteration to ensure that this feasibility requirement persists. The well known equivalence between the conjugate-gradient and Lanczos methods is exploited when finding directions of negative curvature. This work is still ongoing, with the resulting software package HSL_VE19 planned for 2000.

References

A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for minimizing a non-convex function subject to bound and linear equality constraints. Technical Report RAL-TR-1999-054, Rutherford Appleton Laboratory, 1999.

N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic problems arising in optimization. Technical Report RAL-TR-98-069, Rutherford Appleton Laboratory, 1998.

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, A. Sartenaer, and Ph. L. Toint. On the local convergence of a trust-region method primal-dual interior point algorithm for constrained nonlinear programming. Technical Report (in preparation), Rutherford Appleton Laboratory, 2000.

N. I. M. Gould and Ph. L. Toint. An iterative active-set method for large-scale quadratic programming. Technical Report (in preparation), Rutherford Appleton Laboratory, 2000.

4.4 Steepest-edge simplex code LA04 for linear programming (J. K. Reid)

Work has been completed on the steepest-edge (Goldfarb and Reid, 1977) simplex code LA04 for linear programming. There has been a re-awakening of interest in simplex codes recently with the realization that for really large problems it is worthwhile to exploit sparsity in all the vectors as well as the matrix factors. Also, they are useful in conjunction with interior-point methods for finding final solutions and for following small changes to the data.

LA04 aims for robustness by using the principle that it seeks a solution that is exact for a nearby problem. It has run successfully on all but one of the test examples in the Netlib test set. This an extremely large and extremely degenerate problem and LA04 'stalls', that is, does a huge number of iterations without progress. The problem is recognized as too difficult for most codes.

References

D. Goldfarb and J. K. Reid A practical steepest-edge simplex algorithm. *Mathematical Programming*, **12**, 361-371, 1977.

5 Fortran

5.1 Co-Array Fortran, a simple parallel extension to Fortran 90 (R. W. Numrich and J. K. Reid)

John Reid collaborated with Robert Numrich of SGI (formerly Cray Research) in the detailed design (Numrich and Reid, 1998) of an extension to Fortran 90 for parallel programming formerly called F^{--} and now renamed Co-Array Fortran.

A Co-Array program is interpreted as if it were replicated a number of times and that all copies were executed asynchronously. Each copy has its own set of data objects and is termed an 'image'. A data object is accessible only within its own image unless it is specified with additional dimensions in square brackets. Such an object has the same shape and address on all images and may be accessed from another image with the help of trailing subscripts enclosed in square brackets. Such a 'co-array' may be used in expressions and assignments as if it were an ordinary Fortran array.

References without square brackets are to the local array (or scalar), so that code that can run independently is uncluttered. Only where there are square brackets or there is a procedure call is communication between images involved. The use of array notation to address data on other images provides a very flexible and clear mechanism for parallel programming.

Array pointer components of co-arrays provide a mechanism for cases that require arrays to have different sizes on different images. We would have liked to use allocatable components, but unfortunately these are not part of Fortran 95, though an ISO Technical Report has now been adopted that makes their presence in Fortran 2000 certain. Co-Array Fortran limits the use of pointer components of co-arrays to that of allocatable components.

Very careful consideration has been given to making the extension easy for compiler writers to implement and to ensuring that optimizations that are available to ordinary Fortran programs are still available. For I/O, we have added an intrinsic called sync_file to allow each processor to perform its I/O through its own buffer; calling the intrinsic has the effect of making buffer data available to other images, and will probably involve flushing the buffer before the call and reloading it afterwards. Similar considerations apply to the use of local memory such as cache and we have added an intrinsic called sync_memory for this purpose.

Cray Research has a subset implementation on the T3E and other vendors are considering implementations.

References

R. W. Numrich and J. K. Reid. Co-Array Fortran for parallel programming. ACM Fortran Forum, **17**, 2 (Special Report), 1998.

6 Miscellaneous Activities

6.1 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. A major activity at CERFACS, and indeed in Toulouse, was the hosting of EuroPar'99 by CERFACS and ENSEEIHT-IRIT with the support of many other Laboratories in the Toulouse region. Iain was general chair for this meeting that was held at the beginning of September 1999. There were over four hundred attendees with nearly 200 talks split into 20 topics, making it the largest EuroPar ever. The social events reflected the "other half" of Toulouse and were well appreciated by natives and visitors alike. Details of the meeting can be found on the conference Web page `http://www.enseeiht.fr/europar99` and the Proceedings were published by Springer (Amestoy, Berger, Daydé, Duff, Fraysse, Giraud and Ruiz, 1999).

The main areas of research in the Parallel Algorithms Group 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 eigensystem calculations, optimization, and the reliability of computations. Other activities of the Group include advanced training by both courses and research. Amongst the research closest to the work of the Group at RAL is the use of domain decomposition techniques in solving large systems from partial differential equations and work on preconditioning, both using the techniques described in Section 3.2 and using sparse approximations to precondition dense matrices from electromagnetic applications. In collaboration with a summer student, Laurent Sutra, and later a new PhD student, Christof Voemel from Germany, we have developed a set of codes to implement the sparse BLAS kernels of the BLAS Technical forum (Section 3.1).

The Parallel Algorithms Team was also involved in many European Projects including the PARASOL Project with RAL and others (Section 6.2), PINEAPL (led by NAg Ltd), and ODESIM (on parallel optimization).

During the reporting period, three students completed their PhDs at CERFACS. Iain was a jury member for the thesis defence of one of them, Serge Gratton. He was also on the jury for two habilitation theses by Annick Sartenaer, a senior at CERFACS, and Patrick Amestoy from ENSEEIHT-IRIT, who had been a PhD student of Iain's at CERFACS many years ago.

Nick visited CERFACS to collaborate with Annick and a PhD student, Dominique Orban, and to work on his book with Philippe Toint, visiting CERFACS from Belgium.

The home page for CERFACS is `http://www.cerfacs.fr` and current information on the Parallel Algorithms Group can be found on page `http://www.cerfacs.fr/algor/`. Full details on the activities of the Parallel Algorithms Team can be found in the report CERFACS (1999).

References

P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé, L. Giraud, and D. Ruiz, editors. *EuroPar'99 Parallel Processing*, Lecture Notes in Computer Science, No. 1685, Springer-Verlag, Berlin, Heidelberg, New York, 1999.

CERFACS. Activity report of the parallel algorithms project at CERFACS. January 1998 - December 1998. Technical Report TR/PA/99/16, CERFACS, Toulouse, France, 1999.

6.2 European project PARASOL, an integrated programming environment for PARAllel sparse matrix SOLvers (I. S. Duff)

PARASOL was a long term research (LTR) ESPRIT IV Project for "An Integrated Environment for Parallel Sparse Matrix Solvers". This Project started on January 1st, 1996 and finished on June 30th, 1999. Its main aim was to develop a parallel scalable library of sparse matrix solvers using Fortran 90 and MPI. The codes from this project are now available in the public domain from the Web site `http://www.pallas.de/parasol`.

The PARASOL Consortium was managed by PALLAS in Germany and consisted of

- leading European research organizations with a well known experience and track-record in the development of parallel solvers (CERFACS, GMD-SCAI, ONERA, RAL, Univ. of Bergen);
- industrial code developers who defined the requirements for PARASOL and used its results (Apex Technologies, Det Norske Veritas (DNV), INPRO, MSC, Polyflow);
- two leading European HPC software companies who managed the project, disseminated information, and provided programming development tools (GENIAS, PALLAS).

The codes in the PARASOL Library include direct methods, domain decomposition techniques, and multigrid approaches. Within this project, RAL was involved in the development of direct solvers and worked in this context in close collaboration with CERFACS and with ENSEEIHT (Toulouse, France) who formally joined the Project as a subcontractor to CERFACS. The ENSEEIHT involvement was managed by Patrick Amestoy. More details on the MUMPS software can be found in Section 2.1.

There have been several researchers who have worked on the PARASOL Project at RAL during the period of this report. Jean-Yves L'Excellent joined the Project at CERFACS in October 1996 and worked on the MUMPS code until July 1999. He worked at RAL for the month of May 1999. Petr Plecháč started at RAL at the beginning of November 1997 and worked on combining graph partitioning with sparse matrix orderings and on the interface between MUMPS and PARASOL. He left the project at the end of August 1998. Jacko Koster joined the PARASOL Team at RAL in February 1998 and stayed until the end of the Project.

There have been a number of meetings of the PARASOL Project over the past two years. At a Project Review meeting at GMD in Bonn in February 1998, the European Commission gave permission for the Project to be extended from December 1998 to June 1999, although with no extra funding. As a condition of this extension, we were subjected to an extra "Health Check" Review at ONERA in Paris in November 1998. The extension was necessary largely because of the slow start to the Project, and it enabled the partners both to spend the EU money and to meet the objectives of the Project. The final Project review took place in Toulouse in August 1999, and we obtained a very good report from the reviewers. Between these meetings, we had three internal meetings of the Project. One in Bonn in December 1998 to finalize the Library interface, one in Munich in March 1999 to agree on test problems and a consistent way of reporting on them, and a third in Bergen in June 1999 to collate our experiences and prepare for the final review meeting.

Information on PARASOL can be obtained from the GENIAS or EU PROSOMA Web pages `http://www.genias.de/projects/parasol/details.html` or `http://www.prosoma.lu`, respectively.

7 Computing and mathematical software

7.1 The computing environment within the Group

Our policy of upgrading the Group's workstations has continued over the past two years. The main change has been that the Group's venerable "high-performance" machine, our IBM Risc Systems/6000 3BT, has been replaced by a considerably faster Compaq Alpha DS20 dual EV6-processor server, with 3.5Gbytes of memory. One of the Group's SUN workstations has also been replaced by an improved performance SUN Ultra Sparc 5, but we are now debating whether to move in future to cheaper, and more powerful, Intel-like machines. In addition, we now have three Dell CPi laptop machines of varying configurations, and have chosen to allow them to dual-boot for both Microsoft (NT) and (RedHat) Linux operating systems. Our productivity has most definitely improved with our ability to compute "on the move". Our previous portable machine, an IBM Thinkpad 701 has been retired.

The responsibility for SUN software support continues to be delegated to other parts of the Laboratory, although group members have still found it more convenient to get their hands dirty for simple tasks—in particular, we maintain our own Linux systems on the laptops. The Group continues to support a series of WWW pages describing its activities, and now use our own, slightly outdated, IBM RS/6000 as a WWW server.

We still benefit from other public RAL machines, in particular the Compaq multiprocessor systems. The Group's files continue to reside on a central UNIX data store, which is backed up daily by the Information Technology Department. We now have access to a number of Fortran 95 compilers, some on our own machines, and are now writing Fortran 95 codes. We have also made use of MPI, and associated parallel language support systems, on both our own machines and on those provided by ITD.

In combination with our Grant application, we obtained computing time on national facilities: namely on Columbus at RAL, and the CSAR machines (SGI Origin 2000 and CRAY T3E) at Manchester.

7.2 Software packages

The following new packages were developed during the reporting period.

LA04 Sparse linear programming: steepest-edge simplex method (J. K. Reid)

This package uses the simplex method to solve the linear programming problem

minimize $c^T x = \sum_{j=1}^{n} c_j x_j$

subject to the constraints

$Ax = b$
$l_j \leq x_j \leq u_j, 1 \leq j \leq k,$
$x_j \geq 0, l \leq j \leq n.$
The variables x_j, $k+1 \leq j \leq l-1$, if any, are free (have no bounds). Full advantage is taken of any zero coefficients a_{ij}. The inequalities $0 \leq k \leq l \leq n+1$ must hold. Special values $l_i = -\sigma$ and $u_i = \sigma$ may be used to remove one or both bounds.

To accommodate roundoff, all variables are permitted to lie slightly outside their bounds.

MA49 Sparse over-determined: least squares by QR (I. S. Duff)

This subroutine computes an orthogonal factorization of a sparse overdetermined matrix A and optionally solves the least squares problem $\min||b - Ax||_2$. Given a sparse matrix A of order $m \times n$, $m \geq n$, of full column rank, this subroutine computes the QR factorization $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$ where Q is an $m \times n$ orthogonal matrix and R is an $n \times n$ upper triangular matrix.

Given an n-vector b, this subroutine may also compute the minimum 2-norm solution of the linear system $A^T x = b$, by solving $\begin{bmatrix} R^T & 0 \end{bmatrix} z = b$ and performing the multiplication $x = Qz$, or, if the Q factor is not stored, by solving $R^T R z = b$ and performing the multiplication $x = Az$.

The subroutine can also solve systems with the coefficient matrix $\begin{bmatrix} R \\ 0 \end{bmatrix}$ or $\begin{bmatrix} R^T & 0 \end{bmatrix}$, or will compute the product of Q or Q^T with a vector.

The method used is based on the multifrontal approach and makes use of Householder transformations. Because an ordering for the columns is chosen using the pattern of the matrix $A^T A$, this code is not designed for matrices with full rows.

Versions exist for globally addressable parallel computers. The parallel versions are machine dependent but only require simple features like starting parallel tasks and locks. In principle, a code can be supplied for any shared memory parallel machine, but the only platforms on which the shared memory code has been tested are the SGI Origin 2000 and the CRAY C98.

HSL_MA55 Band symmetric positive-definite linear system (J. K. Reid)

This module solves a system of linear equations whose matrix is banded, symmetric and positive definite. It uses block Cholesky factorization, taking advantage of any variation in bandwidth. It has an option for storing the matrix itself as well as its factorization. A secondary entry provides for further systems with the same matrix but different right-hand sides to be solved economically. The secondary entry can also be used to calculate residuals, needed for iterative refinement, provided the matrix itself has been stored. For very large systems or if restart facilities are desired, HSL_MA55 uses a direct-access file for the factors and a sequential file for the matrix.

The user must first specify all the row lengths. The matrix itself is supplied by blocks of rows using reverse communication. The blocking for input is chosen by the user and is independent of the blocking used for solution.

MC60 may be used to get a good ordering.

MA72 Sparse symmetric finite-element equations: out-of-core multiple front method (J. A. Scott)

This collection of subroutines, when used in conjunction with the MC62 package, solves symmetric positive-definite finite-element equations using a multiple front algorithm. It is assumed that the underlying finite-element mesh has been partitioned into (non-overlapping) subdomains. In the multiple front algorithm, a frontal method is applied to each subdomain separately.

The application of the frontal method to each subdomain may be done in parallel. Using multiple fronts can also have the advantage of requiring less work than applying the frontal method to the whole domain.

MA72 provides routines for generating lists of interface variables, for preserving the partial factorization of a matrix when the sequence of calls to the frontal solver factorization routine MA62B/BD is incomplete, and for performing forward elimination or backsubstitution on a subdomain.

MA72 uses reverse communication.

MC57 Assemble a set of finite-element matrices (J. A. Scott)

This subroutine assembles a set of element matrices, that is, it forms the summation $A = \sum_l A^{[l]}$, where each element matrix $A^{[l]}$ has entries only in the principal submatrix corresponding to the variables in element l. Each $A^{[l]}$ must be held in packed form as a small full square matrix, together with a list of the variables associated with element l. The assembled matrix A has a symmetric sparsity pattern but may be unsymmetric. An option exists for assembling only the sparsity pattern of A. If the variables are not indexed contiguously, absent rows and columns may optionally be removed.

MC62 Generate a row ordering for a row-by-row frontal solver (J. A. Scott)

Given an $n \times n$ matrix A with an unsymmetric sparsity pattern, this subroutine generates a row ordering for a row-by-row frontal solver (for example, the HSL packages MA42 and MA43.

MC62 generates a row ordering that is designed to reduce the maximum and mean row and column frontsizes, the maximum and mean frontal matrix size, and the sum of the lifetimes, which in turn reduce storage requirements and operation counts for the frontal solver. Only the pattern of the matrix is used. MC62 is not recommended if A has one or more rows that are full or have a large number of nonzeros.

MC62 offers the option of generating the row graph of an $m \times n$ matrix A. The nodes of the row graph are the rows of A and two rows i and j $(i \neq j)$ are defined to be adjacent if and only if there is at least one column k of A for which $a_{ik}.a_{jk} \neq 0$.

MC63 Generate an element assembly ordering for a frontal solver (J. A. Scott)

This subroutine uses a variant of Sloan's algorithm to generate an element assembly ordering that is efficient when subsequently used with a frontal solver (for example, the packages MA42 and MC62). The number of floating-point operations and the storage required by a frontal solver for an unassembled finite-element matrix are dependent upon the order in which the elements are assembled; the variation in the performance of different element orderings can be significant. The assembly ordering obtained by MC63 is designed to reduce the maximum and root-mean-square (r.m.s.) wavefronts and the profile, which in turn reduce storage requirements and computation times for the frontal solver. Only the pattern of the finite elements is used.

MC64 Find permutation that places large entries on the diagonal of a sparse matrix (I. S. Duff and J. Koster)

Given a sparse $n \times n$ matrix A, this subroutine attempts to find a column permutation vector that makes the permuted matrix have n entries on its diagonal. If the matrix is structurally nonsingular, the subroutine optionally returns a column permutation that maximizes the smallest element on the diagonal, maximizes the sum of the diagonal entries, or maximizes the product of the diagonal entries of the permuted matrix. For the latter option, the subroutine also finds scaling factors that may be used to scale the original matrix so that the nonzero diagonal entries of the permuted and scaled matrix are one in absolute value and all the off-diagonal entries are less than or equal to one in absolute value. The natural logarithms of the scaling factors u_i, $i = 1, \ldots, n$, for the rows and v_j, $j = 1, \ldots, n$, for the columns are returned so that the scaled matrix B has entries $b_{ij} = a_{ij}e^{(u_i+v_j)}$.

ME62 Sparse Hermitian or complex symmetric linear system: multifrontal method (J. A. Scott)

This package solves one or more sets of sparse Hermitian or complex symmetric linear unassembled finite-element equations, $AX = B$, by the frontal method, optionally holding the matrix factor out-of-core in direct access files. The package is primarily designed for positive-definite matrices since numerical pivoting is not performed. Use is made of high-level BLAS kernels. The coefficient matrix A must of the form $A = \sum_{k=1}^{m} A^{(k)}$, with $A^{(k)}$ nonzero only in those rows and columns that correspond to variables in the k-th element.

The frontal method is a variant of Gaussian elimination and involves the factorization $A = PLD(PL)^T$, where P is a permutation matrix, D is a diagonal matrix, and L is a unit lower triangular matrix. MC62 stores the reals of the factors and their indices separately. A principal feature of MC62 is that, by holding the factors out-of-core, large problems can be solved using a predetermined and relatively small amount of in-core memory. At an intermediate stage of the solution, l say, the 'front' contains those variables associated with one or more of $A^{(k)}$, $k = 1, 2, \ldots, l$, which are also present in one or more of $A^{(k)}$, $k = 1, 2, \ldots, m$. For efficiency, the user should order the $A^{(k)}$ so that the number of variables in the front (the 'front size') is small. For example, a very rectangular grid should be ordered pagewise parallel to the short side of the rectangle. The elements may be preordered using the Harwell Subroutine Library routine MC63.

HSL_MP42 Unsymmetric finite-element equations: multiple-front method (J. A. Scott)

The module HSL_MP42 uses the multiple front method to solve sets of finite-element equations $AX = B$ that have been divided into non-overlapping subdomains. The routines MA42 and MA52 are used with MPI for message passing.

In the multiple front method, a frontal decomposition is performed on each subdomain separately. Thus, on each subdomain, L and U factors are computed. Once all possible eliminations have performed within a subdomain, there remain the interface variables, which are shared by more than one subdomain together with any variables that are not eliminated because of stability or efficiency considerations. If F_i is the remaining frontal matrix for subdomain i, and C_i is the corresponding right-hand side matrix, then the remaining problem is $FY = C$, where $= \sum_i F_i$ and $C = \sum_i C_i$. By treating each F_i as an element matrix, the interface problem (3) is also solved by the frontal method. Once (3) has been solved, backsubstitution on the subdomains completes the solution.

The element data and/or the matrix factors are optionally held in direct-access files.

HSL_MP62 Symmetric finite-element equations: multiple-front method (J. A. Scott)

The module HSL_MP62 uses the multiple front method to solve sets of symmetric positive-definite finite-element equations $AX = B$ that have been divided into non-overlapping subdomains. The Harwell Subroutine Library routines MA62 and MA72 are used with MPI for message passing.

In the multiple front method, a frontal decomposition is performed on each subdomain separately. Thus, on each subdomain, L and U factors are computed. Once all possible eliminations have performed within a subdomain, there remain the interface variables, which are shared by more than one subdomain. If F_i is the remaining frontal matrix for subdomain i, and C_i is the corresponding right-hand side matrix, then the remaining problem is $FY = C$, where $= \sum_i F_i$ and $C = \sum_i C_i$. By treating each F_i as an element matrix, the interface problem (3) is also solved by the frontal method. Once (3) has been solved, backsubstitution on the subdomains completes the solution.

The element data and/or the matrix factors are optionally held in direct-access files.

HSL_VE12 Quadratic programming problem: interior-point trust-region method (N. I. M. Gould)

This package uses a primal-dual interior-point trust-region method to solve the quadratic programming problem

minimize $\frac{1}{2}x^T Hx + g^T x$

subject to the general linear constraints

$c_i^l \leq a_i^T x \leq c_i^u$, $i = 1, 2, \ldots, m$,

and the simple bound constraints

$x_j^l \leq x_j \leq x_j^u$, $j = 1, 2, \ldots, n$,

where the $n \times n$ symmetric matrix H and the vectors g, a_i, x^l, x^u, c^l, and c^u are given. Full advantage is taken of any zero coefficients in the matrix H or the vectors a_i. Any of the constraint bounds x_j^l, x_j^u, c_i^l and c_i^u may be infinite.

If the matrix H is positive semi-definite, a global solution is found. However, if H is indefinite, the procedure may find a local solution which is not the global solution to the problem.

HSL_VE13 Constrained least distance problem:interior-point trust-region method (N. I. M. Gould)

This package uses an primal-dual interior-point trust-region method to solve the constrained least distance problem

minimize $\sqrt{\sum_{i=1}^n w_i^2 (x_i - x_i^0)^2}$

subject to the general linear constraints

$c_i^l \leq a_i^T x \leq c_i^u$, $i = 1, 2, \ldots, m$

and the simple bound constraints

$x_j^l \leq x_j \leq x_j^u,\ j = 1, 2, \ldots, n,$
where the vectors w, x^0, a_i, x^l, x^u, c^l, and c^u are given.

Full advantage is taken of any zero coefficients in the vectors a_i. Any of the constraint bounds x_j^l, x_j^u, c_i^l and c_i^u may be infinite. In the special case where $w = 0$, the so-called analytic centre of the feasible set will be found.

HSL_VE15 Quadratic programming: reorder the problem (N. I. M. Gould)

This package reorders to a standard form the variables and constraints for the quadratic programming problem

minimize $\frac{1}{2}x^T Hx + g^T x$
subject to the general linear constraints

$c_i^l \leq a_i^T x \leq c_i^u,\ i = 1, 2, \ldots, m,$
and the simple bound constraints

$x_j^l \leq x_j \leq x_j^u,\ j = 1, 2, \ldots, n,$
where the $n \times n$ symmetric matrix H and the vectors g, a_i, x^l, x^u, c^l, and c^u are given. Full advantage is taken of any zero coefficients in the matrix H and the vectors a_i. Any of the constraint bounds x_j^l, x_j^u, c_i^l and c_i^u may be infinite.

The variables are reordered so that any free variables (i.e., those without bounds) occur first, followed respectively by non-negativities (i.e., those for which the only bounds are that $x_j \geq 0$), lower-bounded variables (i.e., those for which the only bounds are that $x_j \geq x_j^l \neq 0$), range-bounded variables (i.e., those for which the bounds satisfy $-\infty < x_j^l < x_j^u < \infty$) upper-bounded variables (i.e., those for which the only bounds are that $x_j \leq x_j^u \neq 0$), and finally non-positivities (i.e., those for which the only bounds are that $x_j \leq 0$). Fixed variables will be removed. Within each of the above categories, the variables are further ordered so that those with non-zero diagonal Hessian entries occur before the remainder.

The constraints are reordered so that equality constraints (i.e., those for which $c_i^l = c_i^u$) occur first, followed respectively by those which are lower-bounded (i.e., those for which the only bounds are that $a_i^T x \geq c_i^l$), those which have ranges (i.e., those for which the bounds satisfy $-\infty < c_j^l < c_j^u < \infty$), and finally those which are upper-bounded (i.e., those for which the only bounds are that $a_i^T x \leq c_i^u$). Free constraints, that is those for which $c_i^l = -\infty$ and $c_i^u = \infty$), are removed.

Procedures are provided to determine the required ordering, to reorder the problem to standard form, and to recover the problem, or perhaps just the values of the original variables, once it has been converted to standard form.

It is anticipated that this module will principally be used as a pre- and post-processing tool for other HSL packages.

HSL_VF06 Global minimization of sparse quadratic function with norm constraint (N. I. M. Gould)

Given a real $n \times n$ symmetric matrix H, a real n vector c and a positive radius Δ, this package finds a global minimizer of the quadratic objective function $\frac{1}{2}x^T Hx + c^T x$, where the vector x is required to satisfy the constraint $||x||_M \leq \Delta$, and where the M-norm of x is $||x||_M = \sqrt{x^T Mx}$. The symmetric positive-definite matrix M is constructed from an appropriate factorization of H, and is chosen so that the problem is easy to solve. Such problems commonly occurs as a trust-region subproblems in nonlinear optimization calculations, and it is envisaged that this will be the primary use for this package.

8 Seminars

19 February 1998 Professor N. Nichols (Reading) Uncertain systems and data assimilation: the problem of weather prediction.

5 March 1998 Dr R. Fletcher (Dundee) Recent progress with filter methods for nonlinear programming.

17 March 1998 Dr M. van Gijzen (Utrecht, The Netherlands) GMRES-like methods on distributed memory computers.

7 May 1998 Dr K.D. Andersen (Dash Associates Ltd) The parallelization of the XPRESS interior point optimizer for a shared-memory multiprocessor using an OpenMP (like) programming environment.

15 October 1998 Dr A.R. Krommer (NAg Ltd) Parallel sparse matrix computations in the PINEAPL Library.

26 November 1998 Dr P. Knight (Strathclyde) k-SAT and Markov chains.

4 February 1999 Dr A. Trefethen (NAg Ltd) Developing numerical software for today's computing environments.

19 February 1999 Mr F. Magoulès (ONERA, France) Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic problems.

23 February 1999 Dr Y. Hu (Daresbury Laboratory) Problems and algorithms for dynamic load balancing and unsymmetric matrix ordering in parallel computation.

27 May 1999 Dr J. Gondzio (Edinburgh) Exploiting structure in the linear algebra of interior point methods.

3 November 1999 Dr D. Ryan (Auckland, New Zealand) Real operations research in practice.

25 November 1999 Dr J. Hall (Edinburgh) Exploiting hypersparsity in the revised simplex method.

9 Reports issued in 1998-1999

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-98-003	A linesearch algorithm with memory for unconstrained optimization. N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint.
RAL-TR-98-005	Subspace-by-Subspace preconditioners for structured linear systems. M. J. Daydé, J. Décamps, and N. I. M. Gould.
RAL-TR-98-016	Ordering symmetric sparse matrices for small profile and wavefront. J. K. Reid and J. A. Scott.
RAL-TR-98-027	Implicit scaling of linear least squares problems. J. K. Reid.
RAL-TR-98-028	Numerical Analysis Group Progress Report. January 1996 - December 1997. I. S. Duff (Editor).
RAL-TR-98-031	On ordering elements for a frontal solver. J. A. Scott.
RAL-TR-98-039	PARASOL An Integrated Programming Environment for Parallel Sparse Matrix Solvers. P. Amestoy, I. Duff, J.-Y. L'Excellent, and P. Plecháč.
RAL-TR-1998-051	Multifrontal Parallel Distributed Symmetric and Unsymmetric Solvers. P. Amestoy, I. Duff, and J.-Y. L'Excellent.
RAL-TR-1998-052	Preconditioning and Parallel Preconditioning. I. S. Duff and H. A. van der Vorst.
RAL-TR-1998-054	Direct Methods. I. S. Duff.
RAL-TR-1998-056	A new row ordering strategy for frontal solvers. J. A. Scott.
RAL-TR-1998-057	AD01, A Fortran 90 code for automatic differentiation. J. D. Pryce and J. K. Reid.
RAL-TR-1998-060	Co-Array Fortran for parallel programming. R. W. Numrich and J. K. Reid.

RAL-TR-1998-069 On the solution of equality constrained quadratic programming problems arising in optimization. N. I. M. Gould, M. E. Hribar, and J. Nocedal.

RAL-TR-1998-076 Matrix methods. I. S. Duff.

RAL-TR-1999-011 Locking and restarting quadratic eigenvalue solvers. K. Meerbergen.

RAL-TR-1999-016 Constraint preconditioning for indefinite linear systems. C. Keller, N. I. M. Gould and A. J. Wathen.

RAL-TR-1999-025 The rational Lanczos method for the Hermitian eigenvalue problem. K. Meerbergen.

RAL-TR-1999-027 Developments and trends in the parallel solution of linear systems. I. S. Duff and H. A. van der Vorst.

RAL-TR-1999-030 On algorithms for permuting large entries to the diagonal of a sparse matrix. I. S. Duff and J. Koster.

RAL-TR-1999-035 Row ordering for frontal solvers in chemical process engineering. J. A. Scott.

RAL-TR-1999-037 Reversing the row order for the row-by-row frontal method. J. K. Reid and J. A. Scott.

RAL-TR-1999-039 Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry. F. Magoulès, K. Meerbergen and J.-P. Coyette.

RAL-TR-1999-041 Global convergence of trust-region SQP-filter algorithms for general nonlinear programming. R. Fletcher, N. I. M. Gould, S. Leyffer and Ph. L. Toint.

RAL-TR-1999-045 A class of incomplete orthogonal factorization methods I: methods and theories. Z.-Z. Bai, I. S. Duff, and A. J. Wathen.

RAL-TR-1999-054 A primal-dual trust-region algorithm for minimizing a non-convex function subject to general inequality and linear equality constraints. A. R. Conn, N. I. M. Gould, D. Orban and Ph. L. Toint.

RAL-TR-1999-055 SQP methods for large-scale nonlinear programming. N. I. M. Gould and Ph. L. Toint.

RAL-TR-1999-059 A fully asynchronous multifrontal solver using distributed dynamic scheduling. P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster.

RAL-TR-1999-072 The impact of high performance computing in the solution of linear systems: trends and problems. I. S. Duff.

RAL-TR-1999-075 The design of a parallel frontal solver. J. A. Scott.

CERFACS Reports

TR/PA/98/02 MUMPS MUltifrontal Massively Parallel Solver. Version 2.0. P. R. Amestoy, I. S. Duff, and J-Y L'Excellent.

TR/PA/99/12 A brief bibliography of recent research and software for the parallel solution of large sparse linear equations. I. S. Duff.

10 External Publications in 1998-1999

References

P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé, L. Giraud, and D. Ruiz, editors. *EuroPar'99 Parallel Processing*, Lecture Notes in Computer Science, No. 1685, Springer-Verlag, Berlin, Heidelberg, New York, 1999.

P. R. Amestoy, I. S. Duff, and J. -Y. L'Excellent. Multifrontal solvers within the PARASOL environment. *In* B. Kågström, J. Dongarra, E. Elmroth and J. Waśniewski, eds, 'Applied Parallel Computing, PARA'98', Lecture Notes in Computer Science, No. 1541, Springer-Verlag, Berlin, pp. 7–11, 1998.

P. R. Amestoy, I. S. Duff, and J. -Y. L'Excellent. Parallélisation de la factorisation lu de matrices creuses non-symmétriques pour des architectures à mémoire distribuée. *Calculateurs Parallèles Réseaux et sytèmes répartis*, **10(5)**, 509–520, 1998.

P. R. Amestoy, I. S. Duff, and J. -Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods in Appl. Mech. Eng.*, (to appear), 2000.

P. R. Amestoy, I. S. Duff, J. -Y. L'Excellent, and P. Plecháč. PARASOL. An integrated programming environment for parallel sparse matrix solvers. *In* R. J. Allan, M. F. Guest, A. D. Simpson, D. S. Henty and D. A. Nicole, eds, 'High-Performance Computing', Kluwer Academic/Plenum Publishers, New York, pp. 79–90, 1999.

J. Cardenal, I. S. Duff, and J. M. Jiménez. Solution of sparse quasi-square rectangular systems by Gaussian elimination. *IMA J. Numerical Analysis*, **18**(2), 165–177, 1998.

C. Cartensen and P. Plecháč. Adaptive algorithms for scalar non-convex variational problems. *Appl. Numer. Math.*, **26**(1–2), 203–216, 1998.

K. A. Cliffe, I. S. Duff, and J. A. Scott. Performance issues for frontal schemes on a cache-based high performance computer. *Int. J. Numerical Methods in Engineering*, **42**, 127–143, 1998.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A primal-dual algorithm for minimizing a nonconvex function subject to bound and linear equality constraints. *In* G. Di Pillo and F. Gianessi, eds, 'Nonlinear Optimization and Applications 2', pp. 15–50, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-region methods.* SIAM, Philadelphia, USA, (to appear), 966+xvii pages, 2000.

A. C. Damhaug, A. Bergseth, and J. K. Reid. The impact of an efficient linear solver on finite-element analyses. *Computer and Structures* **72**, 595–604, 1999.

M. J. Daydé, J. P. Décamps, and N. I. M. Gould. On the use of block stretching for solving unassembled linear systems. *Calculateurs Paralléles, Réseaux et Systémes Répartis*, **10**(4), 391–399, 1999.

M. J. Daydé, J. P. Décamps, and N. I. M. Gould. Subspace-by-subspace preconditioners for structured linear systems. *Numerical Linear Algebra*, **6**, 213–234, 1999.

T. A. Davis and I. S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Trans. Math. Softw.*, **25**(1), 1–20, 1999.

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst. *Numerical Linear Algebra for High-Performance Computers.* SIAM Press, Philadelphia, 1998.

I. S. Duff. A brief bibliography of recent research and software for the parallel solution of large sparse linear equations. *In* 'Proceedings of the Computational Cattle Breeding '99 Workshop, Tuusula, Finland, March 18-20, 1999', Bulletin No 20, International Bull Evaluation Service, Uppsala, Sweden, pp. 43–46, 1999.

I. S. Duff. Matrix methods. *In* K. L. Bell, K. A. Berrington, D. S. F. Crothers, A. Hibbert and K. T. Taylor, eds, 'Supercomputing, Collision Processes and Applications', Kluwer Academic, New York, pp. 119–136, 1999.

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. A. Scott. A comparison of frontal software with other Harwell Subroutine Library sparse direct solvers. *In* P. Arbenz, M. Paprzycki, A. Sameh and V. Sarin, eds, 'High Performance Algorithms for Structured Matrix Problems', NOVA Science Publishers, Inc., Commack, NY, pp. 1–25, 1998.

I. S. Duff and J. A. Scott. A frontal code for the solution of sparse positive-definite symmetric systems arising from finite-element applications. *ACM Trans. Math. Softw.*, (to appear), 2000.

I. S. Duff and J. A. Scott. MA62 – a frontal code for sparse positive-definite symmetric systems from finite element applications. *In* M. Papadrakakis and B. Topping, eds, 'Innovative Computational Methods for Structural Mechanics', Saxe-Coburg Publications, Edinburgh, pp. 1–25, 1999.

N. I. M. Gould. Iterative methods for ill-conditioned linear systems from optimization. *In* G. Di Pillo and F. Gianessi, eds, 'Nonlinear Optimization and Applications 2', Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 123–142, 1999.

N. I. M. Gould. On modified factorizations for large-scale, linearly-constrained optimization. *SIAM Journal on Optimization*, **9**(4), 1041–1063, 1999.

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, S. Lucidi, M. Roma, and Ph. L. Toint. A linesearch algorithm with memory for unconstrained optimization. *In* R. D. Leone, A. Murli, P. M. Pardalos and G. Toraldo, eds, 'High Performance Algorithms and Software in Nonlinear Optimization', Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 207–223, 1998.

N. I. M. Gould and J. Nocedal. The modified absolute-value factorization norm for trust-region minimization. *In* R. D. Leone, A. Murli, P. M. Pardalos and G. Toraldo, eds, 'High Performance Algorithms and Software in Nonlinear Optimization', Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 225–241, 1998.

N. I. M. Gould and J. A. Scott. Sparse approximate-inverse preconditioners using norm-minimization techniques. *SIAM Journal on Scientific Computing*, **19**(2), 605–625, 1998.

N. I. M. Gould and Ph. L. Toint. A note on the second-order convergence of optimization algorithms using barrier functions. *Mathematical Programming*, **85**(2), 433–438, 1999.

N. I. M. Gould and Ph. L. Toint. SQP methods for large-scale nonlinear programming. *In* M. J. D. Powell and S. Scholtes, eds, 'Proceedings of the IFIP TC7 Conference on System Modelling and Optimization, Cambridge, 1999', Kluwer Academic Publishers, Dordrecht, The Netherlands, (to appear), 2000.

D. A. Johnston and P. Plecháč. Equivalence of ferromagnetic spin models on trees and random graphs. *J. Phys. A-Math. Gen.*, **31**(2), 475–482, 1998.

C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Analysis and Applications*, (to appear), 2000.

R. Lehoucq and K. Meerbergen. Using generalized Cayley transformations within an inexact rational Krylov sequence method. *SIAM J. Matrix Anal. Applic.* **20**(1), 131–148, 1998.

K. Meerbergen. A theoretical comparison between inner products in the shift-invert Arnoldi method and the spectral transformation Lanczos method. *ETNA* **7**, 90–103, 1998.

K. Meerbergen and M. Sadkane. Using Krylov approximations to the matrix exponential operator in Davidson's method. *Applied Numerical Mathematics* **31**, 331–351, 1999.

J.-L. Migeot, K. Meerbergen, and Ch. Lecomte. Implementation and accuracy issues. *In* O. von Estorff, ed., 'Boundary Element Methods in Acoustics'. Computational Mechanics Ltd, (to appear), 2000.

R. Morgan and K. Meerbergen. §11.2. Inexact methods, *In* Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst, eds, 'Templates for the solution of algebraic eigenvalue problems: a practical guide', SIAM, Philadelphia, USA, (to appear), 2000.

R. W. Numrich, K. Kim, and J. K. Reid. Writing a multigrid solver using Co-array Fortran. *In* B. Kågström, J. Dongarra, E. Elmroth and J. Waśniewski, eds, 'Applied Parallel Computing, PARA'98', Lecture Notes in Computer Science, No. 1541, Springer-Verlag, Berlin, pp. 390–399, 1998.

J. K. Reid. Implicit scaling of linear least squares problems. *BIT*, **40**(1), 146–157, 2000.

J. K. Reid and J. A. Scott. Ordering symmetric sparse matrices for small profile and wavefront. *Inter. Journal on Numerical Methods in Engineering*, **45**, 1737–1755, 1999.

J. K. Reid, A. Supalov, and C. -A. Thole. PARASOL interface to new parallel solvers for industrial applications. *In* E. H. D'Hollander, G. R. Joubert, F. J. Peters, and U. Trottenberg, eds, 'Parallel Computing: Fundamentals, Applications and New Directions', Advances in Parallel Computing 12, Elsevier, The Netherlands, pp. 525-532, 1998.

J. A. Scott. On ordering elements for a frontal solver. *Communications in Numerical Methods in Engineering*, **15**, 309–323, 1999.

J. A. Scott. A new row ordering strategy for frontal solvers. *Numerical Linear Algebra with Applications*, **6**, 1–23, 1999.

J. A. Scott. Sparse direct methods: an introduction. *Springer Verlag Lecture Notes in Physics*, (to appear), 2000.