QMRPACK AND APPLICATIONS

Roland W. Freund*
AT&T Bell Laboratories
Room 2C–420
600 Mountain Avenue
Murray Hill, NJ 07974–0636
freund@research.att.com

Noël M. Nachtigal†
Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Building 6012
Oak Ridge, TN 37831–6367
santa@msr.epm.ornl.gov

Abstract

QMRPACK is a package of quasi-minimal residual algorithms for the iterative solution of linear systems. In this paper, we describe the main features of QMRPACK, comment on the implementations found in the package, and discuss how they would be invoked in a user code. We briefly describe two applications of the codes in the area of simulation of material properties from first principles, and in circuit simulation. We conclude by presenting future directions for QMRPACK.

1 Introduction

Since the introduction of the quasi-minimal residual (QMR) algorithm for complex symmetric linear systems by Freund [3], much work has gone into developing more general and robust QMR methods. In [6], we extended the original algorithm to the case of general non-Hermitian matrices. The resulting algorithm uses the look-ahead variant [5] of the classical three-term Lanczos process [11] as the underlying method for building the Krylov basis. In [7, 8], we developed a coupled two-term recurrence variant of the look-ahead Lanczos process [7, 8], where the three-term recursions are replaced by coupled two-term recursions that retain look-ahead but in practice have better numerical properties. In [4], Freund introduced the transpose-free QMR method (TFQMR), which alleviates the need for the transpose, but did not include any look-ahead and thus is susceptible to breakdowns. Work is in progress on a look-ahead variant of TFQMR that will allow look-ahead steps of limited sizes.

While the underlying look-ahead algorithms are relatively simple to explain at a high level, they are non-trivial to implement robustly. As a result, in 1990, we started a project to produce portable FORTRAN-77 codes implementing the various QMR algorithms as black-box solvers. The result is QMRPACK, a software package containing several QMR methods, as well as an eigenvalue solver.

*Part of this research was performed while this author was in residence at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035, and it was supported in part by Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration and the Universities Space Research Association.

†The research of this author was supported in part by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy, under contract DE–AC05–840R21400 with the Martin Marietta Energy Systems, Inc.

2 Description of QMRPACK

QMRPACK is a collection of FORTRAN-77 library routines implementing several of the QMR algorithms and an eigenvalue solver, available from NETLIB. The package is distributed in the form of the compressed tar file "qmrpack.tar.Z" in the "linalg" subdirectory of NETLIB. It is only retrievable via xnetlib or anonymous ftp. In the latter case, ftp to "netlib.att.com" and then get the file "qmrpack.tar.Z" from the directory "netlib/linalg".

The QMRPACK package currently includes three main QMR algorithms for the solution of square linear systems:

- The original QMR algorithm [6], based on the look-ahead three-term Lanczos process described in [5];
- The QMR algorithm based on the look-ahead coupled two-term recurrence Lanczos process [7, 8];
- The TFQMR algorithm [4].

Even though the main emphasis of the package is on the linear systems solvers, also included is a routine for obtaining approximate eigenvalues from the look-ahead three-term Lanczos algorithm. In addition, the package includes simplified no-look-ahead versions of both the three-term and the coupled two-term QMR algorithms, as well as an implementation of the QMR-from-BCG algorithm proposed by Freund and Szeto [9]. The codes are available for general non-Hermitian matrices, in single and double precision, real and complex data types. We will not describe here any of the algorithms in the package; full details can be found in the corresponding references. Although QMRPACK contains no-look-ahead variants of the QMR codes, this was done in the interest of completeness and with the understanding that the look-ahead variants are the more robust versions and their use is recommended over their no-look-ahead counterparts.

The distribution for QMRPACK includes the solvers, library support routines, example drivers, and example data formats and preconditioners. The library support routines include the BLAS routines, routines from LINPACK, and, for the eigenvalue solver, routines from EISPACK. The example drivers are intended to demonstrate the use of the black-box solvers, and include an example data format (the compressed sparse row format) and several preconditioners for this data format: right, left, and two-sided variants of the ILUT preconditioner proposed by Saad, and right, left, and two-sided variants of the SSOR preconditioner. We note in passing that the two-sided SSOR preconditioner is implemented using the Eisenstat trick, and thus the application of the preconditioned matrix is only marginally more expensive than the application of the unpreconditioned matrix.

As QMRPACK is intended to be a portable library of codes, the low-level solvers are written in strict ANSI FORTRAN-77. Only the example drivers are not strictly ANSI FORTRAN-77; they use the "$" format specifier to indicate to the output processor that it should not advance to the next line after displaying a string. This format is supported by most FORTRAN compilers today (in fact, by all the compilers encountered by the authors), but strictly speaking, it is not ANSI FORTRAN-77.

Some of the issues that come up in the design and use of iterative methods solvers are the choice of a convergence criterion, the integration of the preconditioner in the algorithm, and the implementation of the matrix-vector operations. In QMRPACK, the convergence criterion used is the relative residual norm, $\|r_n\|_2/\|r_0\|_2$. This choice is hard-coded in the algorithms, and there is no facility for the user to change it without additional coding. We are aware that this is not the only choice available for convergence criteria. The preconditioner is not explicitly incorporated into the linear system solvers, though of course this could be done for all the algorithms in the package. The application of the preconditioner rests with the user, inside the matrix-vector routines, and the codes solve the already-preconditioned system (in fact, the codes further assume that the starting guess is always $x_0 = 0$). This means that all the quantities generated by the codes belong to the preconditioned system, and not the original system; in particular, the convergence criterion will use the preconditioned residual (unless right preconditioned is used) to determine convergence. For the matrix-vector routines, QMRPACK uses reverse communication. This mechanism consists of having the solver routines return to the caller, with a flag set, when a matrix-vector multiplication is needed, then having the caller perform the operation and call back the solver. The advantage of reverse communication is that it allows complete flexibility over the data structure for the matrix and the implementation of the matrix-vector operations, since all the matrix information is external to the solvers. The example drivers demonstrate how to set up the calls to the iterative solvers, how to recover the iterates of the original system from those returned by the routines, and how to handle the reverse communication protocol.

Finally, the routines in QMRPACK have several control parameters, allowing the user complete control over such aspects of the algorithm as the generation of the auxiliary starting vectors, the choice between the computation of the true residual norm at every step or the use of the residual norm upper bounds that are available in the QMR algorithms, output history, and so forth. We should also mention that all the vector operations in QMRPACK are implemented via calls to BLAS routines, which means that the package will benefit from optimized versions of these routines where available. It also means that porting the package to parallel architectures is fairly straightforward; indeed, codes from one of the earlier versions of QMRPACK have been ported to the Thinking Machines CM-5 and incorporated into the CMSSL scientific library.

3 Examples of Application of QMRPACK

Nachtigal and Shelton (Oak Ridge National Laboratory) have recently incorporated several QMR algorithms from QMRPACK into codes for the computation of the electronic structure of materials using an approach based on first principles. At the heart of these computations is the explicit inversion of a matrix. The operation is repeated at many points on a grid, and many times in a self-consistent loop, leading to thousands or possibly millions of matrix inversions; this is the major time-consuming part of the computation. Previous versions of the codes used a direct approach to compute the inverses (either Gauss-Jordan or an LU decomposition). The new version replaces the direct solver with an iterative solver from QMRPACK. Preliminary results indicate that the new approach is significantly faster than the old one and that it scales much better as well. Figure 1 shows average times of a computation for Molybdenum, where TFQMR and an LU solver were used. Note that TFQMR and LU perform comparably if the cluster size is 9 atoms, but the TFQMR solver becomes significantly faster as the cluster size is increased and reaches a factor of 11 speedup in CPU time for a cluster size of 113 atoms. The difference is even more pronounced in user time, where the TFQMR solver is more than 40 times faster, as the direct solver codes are swapped out more often. Equally importantly, the tests have shown that the solution time increases linearly with the cluster size when the TFQMR solver is used, as opposed to cubic growth for LU. Thus, the new approach is shown to have significant potential both to speed up the computation and to extend it to larger cluster sizes than those previously attainable.

As a second example of an application of QMRPACK, we mention the recent work of Feldmann and Freund [2] who use the look-ahead Lanczos algorithm from QMRPACK to compute Padé approximations to the impulse-response functions of large linear electronic circuits. The resulting algorithm is called PVL (**P**adé approximation **V**ia the look-ahead **L**anczos process). We illustrate the behavior of the PVL algorithm with one typical example taken from [2]. In Figure 2, we show the exact impulse response (solid line) and the approximation obtained after 60 Lanczos steps (dash-dotted line) for the PEEC circuit, which consists of 2100 capacitors, 172 inductors, and 6990 inductive couplings. Notice the excellent agreement of the two curves over a wide frequency range.

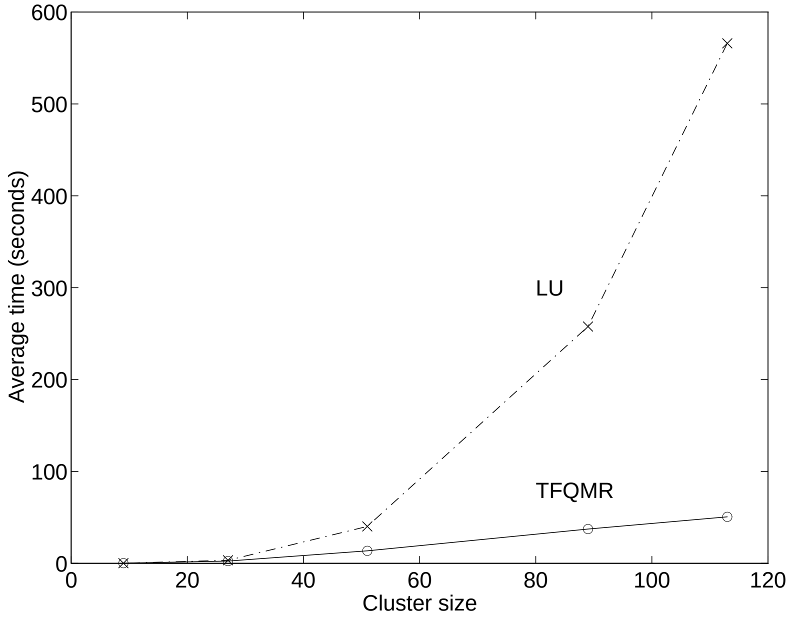

Figure 1: *Average times for* LU *and* TFQMR

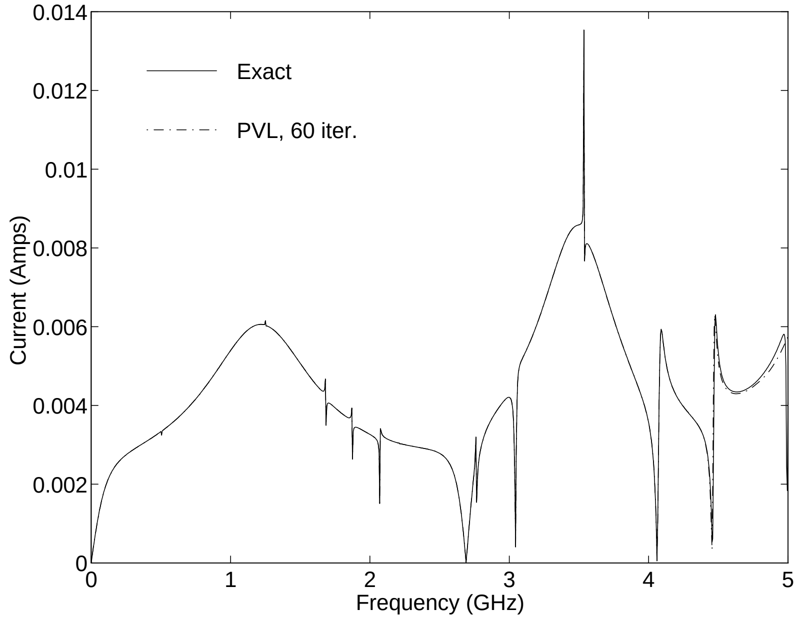

Figure 2: *Simulation of the* PEEC *circuit,* 60 PVL *steps*

4 Future Directions

The next addition to QMRPACK will be codes for symmetric matrices, for which the underlying Lanczos algorithms simplify considerably, reducing work and storage requirements by a factor of two; see [3, 10]. Of particular interest in this class are real symmetric indefinite matrices, and complex symmetric matrices. For real symmetric indefinite matrices, the QMR algorithms have the advantage of allowing the use of a symmetric indefinite preconditioner, while some of the other methods for symmetric indefinite systems, such as MINRES and SYMMLQ, require a symmetric positive definite preconditioner. For complex symmetric matrices, QMR has already been used quite successfully in industry for several years (see e.g. [1]), and the inclusion of codes for this class of matrices in QMRPACK will make them widely available.

As already indicated, work is in progress on a variant of TFQMR that would allow limited look-ahead. Preliminary results indicate that the new method is more robust than the original TFQMR algorithm, and an implementation of this method will be included in QMRPACK once the development is complete.

Finally, Freund has recently proposed a block Lanczos algorithm that would be suitable for use in the solution of linear systems with multiple right-hand sides. Once again, when the development is complete, we expect to incorporate codes implementing QMR algorithms based on the block Lanczos algorithm in QMRPACK. As opposed to the rest of the codes already in QMRPACK and those mentioned above, we expect that the block variants of QMR will be coded in C, rather than FORTRAN. We also expect to investigate a distributed-computing version of this block code, based on PVM, and if successful, to include it in the package.

References

[1] W.E. Boyse and A.A. Seidl, "A sparse matrix iterative method for efficiently computing multiple simultaneous solutions", Manuscript, March 1994.

[2] P. Feldmann and R.W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process", Numerical Analysis Manuscript 94–01, AT&T Bell Laboratories, Murray Hill, NJ, 1994.

[3] R.W. Freund, "Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices", SIAM J. Sci. Statist. Comput. 13, pp. 425–448, 1992.

[4] R.W. Freund, "A transpose-free quasi-minimal residual method for non-Hermitian linear systems", SIAM J. Sci. Comput. 14, pp. 425–448, 1993.

[5] R.W. Freund, M.H. Gutknecht, and N.M. Nachtigal, "An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices", SIAM J. Sci. Comput. 14, pp. 137–158, 1993.

[6] R.W. Freund and N.M. Nachtigal, "QMR: a quasi-minimal residual method for non-Hermitian linear systems", Numer. Math. 60, pp. 315–339, 1991.

[7] R.W. Freund and N.M. Nachtigal, "Implementation details of the coupled QMR algorithm", Numerical Linear Algebra (L. Reichel, A. Ruttan, and R.S. Varga, eds.), pp. 123–140, W. de Gruyter, Berlin, 1993.

[8] R.W. Freund and N.M. Nachtigal, "An implementation of the QMR method based on coupled two-term recurrences", SIAM J. Sci. Comput. 15, pp. 313–337, 1994.

[9] R.W. Freund and T. Szeto, "A transpose-free quasi-minimal residual squared algorithm for non-Hermitian linear systems", Advances in Computer Methods for Partial Differential Equations – VII (R. Vichnevetsky, D. Knight, and G. Richter, eds.), pp. 258–264, IMACS, 1992.

[10] R.W. Freund and H. Zha, "Simplifications of the nonsymmetric Lanczos process and a new algorithm for Hermitian indefinite linear systems", AT&T Numerical Analysis Manuscript, Bell Laboratories, Murray Hill, NJ, 1994.

[11] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators", J. Res. Nat. Bur. Standards 45, pp. 255–282, 1950.