LA-UR 93 - 0303

LA-UR--93-303

DE93 007366

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: **PCG: A SOFTWARE PACKAGE FOR THE ITERATIVE
SOLUTION OF LINEAR SYSTEMS ON SCALAR, VECTOR
AND PARALLEL COMPUTERS**

AUTHOR(S): Wayne Joubert and Graham F. Carey

SUBMITTED TO: **Proceedings of the Sixth SIAM Conference on
Parallel Processing for Scientific Computing
March 22-24, 1993
Norfolk, Virginia**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive royalty-free license to publish or reproduce the published form of this contribution or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

MASTER

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

PCG: A Software Package for the Iterative Solution of Linear Systems on Scalar, Vector and Parallel Computers *

Wayne Joubert †

Graham F. Carey ‡

Abstract

The expanded use of parallel computers heightens the need for numerical software which meets the joint requirements of ease of use, transportability across different architectures, state-of-the-art methods, and high megaflop rates. This study concerns the PCG package which is currently being developed to solve sparse systems of linear equations by preconditioned gradient iterative methods. Here we briefly describe the basic design and some of the features of the package. Issues involved in designing the software to be transportable across different architectures are discussed. Finally some preliminary timings for a model test problem are given to verify feasibility.

1 Introduction

A number of successful software packages have been developed for solving systems of linear equations of the form

$$Au = b \quad (1)$$

or, equivalently, preconditioned systems of the form $(Q_L A Q_R)(Q_R^{-1} u) = (Q_L b)$ (A , Q_L , Q_R square, nonsingular), by use of iterative methods, including for example ITPACK 2C [6], NSPCG [5], PCGPACK [7] and SLAP [8]. These packages are designed to solve linear equations on scalar, and in some cases vector, computer architectures.

The recent proliferation of shared memory parallel computers and distributed memory SIMD and MIMD machines has largely outstripped the ability of the software to keep pace. Unfortunately, scalar or vector codes ported to the new machines rarely give good performance without significant redesign of the software.

This difficulty is compounded by the lack of a universally accepted Fortran-based programming model for the new machines. Although most machines possess Fortran 77 or some generalization thereof, the compilers are typically based on either a shared memory, data parallel or message passing programming model. Even within these categories, significant differences exist. This is coupled with the failure of many current compilers to yield peak performance for all desired algorithms, necessitating the use of mathematical libraries

* This work is supported in part by the Department of Energy through grant W-7403-ENG-36, with Los Alamos National Laboratory. This work was also supported in part by DARPA through grant DABT63-92-C-0024, with the University of Texas at Austin.

† Los Alamos National Laboratory, Computer Research and Applications, Los Alamos, NM 87545.

‡ The University of Texas, Department of Aerospace Engineering and Engineering Mechanics, Austin, TX 78712.

or assembly language. These factors taken together make it difficult to develop codes which perform well on all machines. The present work is directed toward this problem.

It should be emphasized that this software is presently in its early developmental stages. Hence, it is anticipated that details regarding the final version of the software may vary somewhat from the description given here. Furthermore, the optimization of computational kernels thus far developed is currently in progress.

2 Basic Design

The main design objectives for the PCG package include: (1) transportability across different machines; (2) encapsulation of critical algorithm components into a simple set of kernel operations; (3) the inclusion of a useful set of sparse matrix storage formats; and, (4) accessibility to native high performance system routines and languages in order to attain high performance across different machines. Following is a discussion of how these objectives are addressed.

2.1 Transportability Across Machines

Transportability is achieved by use of various software tools which allow the coding of the algorithms to take place at a high level. In the present work, the Unix `m4` macro preprocessor is employed. By use of macros which expand differently for different machine versions, a substantial portion of the source code can be kept machine-independent. Other components which may require machine-specific code optimizations may be accommodated in this setting as well.

To enhance usability of the package across machines, the user-accessible subroutines are designed to have identical calling sequences, independent of the machine. Likewise, the LINPACK/LAPACK [1,3] naming conventions distinguish between single/double and real/complex versions of the same basic routine. On machines with a data parallel or shared memory implementation of Fortran, a subroutine argument for an array such as `U` is interpreted to refer to a global vector, whereas in message passing cases `U` is used to refer to the portion of the global vector resident on that processor, based on an assumed mapping of the vector to the processors.

2.2 PCG Iterative Kernels

The family of gradient-type iterative methods constituting this software system is particularly amenable to parallel transportable development, since the schemes can be decomposed to the repeated use of fundamental operations such as SAXPY operations, matrix-vector products, vector dot products and transpose matrix-vector products. We refer to these as the PCG "kernels" which comprise the main building blocks of the various gradient iterative schemes under development. Moreover, these are also the computation intensive and communication intensive operations. The Unix `m4` preprocessor generates calls to specific subroutines implementing the kernels for a given machine.

2.3 Sparse Matrix Storage Formats

The PCG package provides two levels of user interface. First, a standard top-level interface allows the user to select the iterative accelerator (e.g. conjugate gradient) as well as preconditioner (e.g. Jacobi), assuming a standard matrix storage format. Second, the package may be accessed directly from the accelerator level, the user providing alternative matrix

vector product and preconditioning routines. A third reverse communication interface is also planned.

The most difficult aspect related to portability is to provide matrix formats which are not only general and high performance but also as similar as possible across machines. Two approaches toward solving this problem are adopted: first, the provision of a small number of essential formats (e.g. one for structured problems and one for unstructured problems) which are nearly identical across machines yet give high performance; and second, the provision of additional machine-specific formats to exploit any machine-dependent strengths not accessible to the machine-independent formats.

The first machine-independent format being implemented is the Regular Stencil format, designed for structured problems such as those arising from structured finite difference or finite element grids. This format is expected to give high performance for all target architectures. The Regular Stencil format generalizes the sparse diagonal format of [5] by assuming a rectangular or toroidal regular grid of arbitrary physical dimension. Each grid point allows a fixed number of degrees-of-freedom, and a fixed (finite difference or finite element) stencil is defined at each grid point to relate it to neighbors. For message passing machines, the grid is divided regularly into subgrids of equal size, and a subgrid-to-processor mapping is assumed which preserves neighbor relations.

Other formats are under development, including, for example, element-based formats for unstructured finite element and finite volume grids.

2.4 Low-Level High Performance Routines

The flexibility provided by a macro preprocessor such as `m4` allows the incorporation of machine-specific languages and libraries for the computation- and communication-intensive sections of the code. These can be interfaced to the machine-independent sections of the package to provide high performance. The present software for example utilizes such utilities as Mathlib routines (Ncube 2), BLAS routines or `as860` assembler (Intel iPSC860), the Paris library (CM-2 fieldwise), or the CM Runtime System library (CM-2 slicewise, CM-5).

3 Usage

The top-level calling sequence for the package is given by

```
CALL _PCG (<precon>, <accel>, IA, A, U, UBAR, B, IWK, FWK, IPARM, FPARM, IER)
```

where “_” is either S, D, C or Z depending on whether single/double precision or real/complex arithmetic is desired; `<precon>` and `<accel>` denote externally-defined preconditioner and accelerator names; `IA` and `A` store the matrix A ; `U`, `UBAR` and `B` store the approximate solution, (optional) true solution and right hand side vectors; `IWK` and `FWK` denote integer and floating point workspace; `IPARM` and `FPARM` are parameter arrays; and `IER` is the error code.

The interface to a particular iterative accelerator such as conjugate gradient is given by

```
CALL <accel> (<suba>, IA, A, <subq>, IQ, Q, U, UBAR, B, IWK, FWK, IPARM, FPARM, IER)
```

where `<accel>` is the desired accelerator, `Q` and `IQ` are used to store the preconditioner, `<suba>` and `<subq>` are user-defined matrix-vector product and preconditioning routines, and the other arguments are defined as before. The `<suba>` and `<subq>` routines must have calling sequences of the form

```
CALL SUBA (IJOB, IA, A, VI, VO)
CALL SUBQ (IJOB, IQ, Q, VI, VO)
```

where `VI` and `VO` are respectively the input and output vectors and `IJOB` specifies the task requested, e.g. whether $A \cdot v$ or $A^* \cdot v$ is requested.

4 Preliminary Numerical Results

Computational kernels for the PCG package are currently operational on several different architectures, including the Cray YMP, Intel iPSC860, Ncube 2, CM-2 and CM-5 (CMF and f77/MIMD interface) computers. Representative timings for a basic CG solve on the Ncube 2 computer are given below. The model problem solved is Laplace's equation $-u_{xx} - u_{yy} = f$ on the unit square, discretized with five-point central differences, with uniform mesh size h and Dirichlet boundary data. The unpreconditioned conjugate gradient method is iterated to a convergence criterion of $\|r^{(n)}\|/\|b\| < \zeta = 10^{-3}$. Single precision real arithmetic is employed.

$h^{-1} - 1$	no. processors	iterations	Time, secs.	Mflop rate
4096	256(= 16×16)	1611	1338.550	403.8

5 Conclusion

This study describes a design for a transportable gradient iterative solution package. The package is to be transportable not only across different machine architectures but also across Fortran-based programming models. The software is currently in prototype form and will be expanded and optimized for specific representative architectures.

6 Acknowledgements

The use of the CM-2 and CM-5 computers of Los Alamos National Laboratory, the Intel iPSC-860 computer of the University of Texas, and the Ncube 2 computer of Sandia National Laboratory is greatly appreciated. The assistance of R. McLay, H. Kohli, A. Lorber and Y. Shen is gratefully acknowledged.

References

- [1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *LAPACK User's Guide*, Philadelphia: SIAM, 1992.
- [2] Steven F. Ashby and Mark K. Seager, *A Proposed Standard for Iterative Linear Solvers, Version 1.0*, Lawrence Livermore National Laboratory, Report UCRL-102860, January 1990.
- [3] J. J. Dongarra, J. R. Bunch, C. B. Moler, G. W. Stewart, *LINPACK Users' Guide*. Philadelphia: SIAM, 1979.
- [4] Wayne Joubert, Peter Highnam and Graham Carey, *PCG/CM: A Package for the Iterative Solution of Large Sparse Linear Systems on the Connection Machine*, in Proceedings of Fifth SIAM Conference on Parallel Processing for Scientific Computing, Jack J. Dongarra, et. al., eds. Philadelphia: SIAM, 1992, pp. 631-636.
- [5] D. R. Kincaid, T. C. Oppe and W. D. Joubert, *An Overview of NSPCG: A Nonsymmetric Preconditioned Conjugate Gradient Package*, Computer Physics Communications, 53, 1989.
- [6] D. R. Kincaid, J. R. Respass, D. M. Young, and R. G. Grimes, *ITPACK 2C: A FORTRAN Package for Solving Large Sparse Linear Systems by Adaptive Accelerated Iterative Methods*, ACM Transactions on Mathematical Software, Vol. 8, No. 3, September 1982, pp. 302-322.
- [7] Scientific Computing Associates, New Haven, CT, *PCGPACK User's Guide*, 1984.
- [8] Mark K. Seager, *A SLAP for the Masses*, in G. F. Carey, editor, *Parallel Supercomputing: Methods, Algorithms and Applications*, pp. 135-155, John Wiley and Sons Ltd, New York, 1989.
- [9] David M. Young, *Iterative Solution of Large Linear Systems*. New York: Academic Press, 1971.