

**Final Report: EIA-0103642 NGS: Cache Efficient and Parallel Householder Bidiagonalization— \$103,750**

## 1 Summary

The contract was awarded to Gary Howell and Charles Fulton (co-PI) at Florida Tech in the summer of 2001 with consultants Sven Hammarling and Jim Demmel. In Jan. 2002, Dr. Howell went to work for HP (consulting in parallel computing at the MSRC ERDC computing center) and the contract renewal was with Prof. Fulton as PI, with Howell, Hammarling, and Demmel as consultants. Howell and Hammarling have continued to contribute time to the project both on their own and in trips to Melbourne, Florida. Howell came for long working weekends every few months and Hammarling for week and ten day trips twice a year.

The main endeavor of the group has been to construct Fortran 77 code `dgebr3.f` for Householder bidiagonalization that scheduled to replace and speed the `dgebrd.f` code currently used in the LAPACK package. Bidiagonalization is (due to recent algorithmic advances in computing singular vectors) the most computationally intensive part of computing the singular value decomposition. The work of producing a more efficient and LAPACK compatible version of the LAPACK bidiagonal reduction routine DGEWRD has been completed. We have also completed work on the unsymmetric eigenvalue problem which resulted in an efficient and published algorithm and code.

Remaining work is in providing automated tuning routines to ensure that the code runs at full speed on various cache-based architectures, in exploring the trade-offs with competing band reduction algorithms, in extending the work to the parallel case, and in considering how the computation should be performed on novel architectures.

## 2 Background

Currently, computational speed on most computer architectures is limited by the bandwidth from RAM to CPU. In work accomplished under this proposal, we developed algorithms and implemented portable code to significantly speed some fundamental computations by reducing the volume of data communication required. In particular, we produced a faster algorithm to calculate eigenvalues of unsymmetric matrices, with accompanying Fortran 77 code recently published as an ACM Transactions on Mathematical Software algorithm.

Techniques learned in the unsymmetric eigenvalue computation were applied to Householder bidiagonalization, the predominant expense in calculating least squares solutions via singular value decomposition.

In particular, we used two new BLAS 2.5 operators which had been put in the new BLAS standard mainly for this purpose of speeding bidiagonalization. These are the GEMVER and GEMVT operators. Explicitly, GEMVT accomplishes

$$v \leftarrow \alpha A^T x + \beta y$$

followed by

$$w \leftarrow \gamma Av$$

GEMVER accomplishes also a preliminary rank 2 update.

$$\begin{aligned} A &\leftarrow A + x_1 y_1^T + x_2 y_2^T \\ v &\leftarrow \alpha A^T x + \beta y \\ w &\leftarrow \gamma Av \end{aligned}$$

Our first BLAS 2.5 bidiagonalization was purely BLAS 2.5 using calls to GEMVER (two matrix vector multiplies combined with a rank two update). The purely BLAS 2.5 algorithm is not quite competitive with the LAPACK DGEHRD bidiagonalization, for which half the computation and half BLAS 3. Taking suggestions from Jim Demmel, Jim Stanley, and Sven Hammarling, we revised the bidiagonalization so that half the flops are GEMVT and half are BLAS 3 DGEMM (matrix matrix multiplication) calls.

Using GEMVT as opposed to two calls to GEMV halves the reads of data. On some architectures, bidiagonalization time is almost halved compared to LAPACK DGEHRD. On the architectures we have tested (for the case that a matrix is too large to fit in cache memory), the new bidiagonalization algorithm offers significant speedup.

Working with Sven Hammarling of NAG, we have produced LAPACK compatible Fortran 77 subroutines which pass the LAPACK testing routines and which we propose to integrate into LAPACK. A paper detailing our algorithm has been submitted to TOMS and is available on the web site.

We have also done work in porting the code to a parallel version suitable for inclusion in SCALAPACK. On parallel machines with large caches and fast interconnects, we find that GEMVT calls are faster than paired calls to GEMV, with the first scalable results coming on Compaq SC cluster at ERDC, using approximately square local matrices. Two Florida Tech master's theses addressed this problem. Sumit Malhotra saw good speedups for a column blocking algorithm, but limited scalability. Briefly, if the number of processors  $p$  is increased while the number of matrix entries per processor remains constant, then the volume of parallel communication goes like  $O(\log p \sqrt{p})$ . More recently, Madhan Premkumar, using a PBLAS compatible algorithm using SCALAPACK style block cyclic data layout, showed scalable speedups on the Alpha SC cluster at the NSF Alpha SC cluster at Pittsburgh. In this case (as for parallel GEMV with a block cyclic layout) increasing the number of processors  $p$  while keeping the number of matrix entries per process constant results in parallel communication volume  $O(\log p)$ . Whether GEMVT is faster than two calls to GEMV is architecture dependent. If a collective communication call on a "column communicator" can be completed faster than the data cache can be reloaded from RAM, then GEMVT has an advantage over GEMV. Else (if data caches are small and the processor interconnect is slow) the calls to GEMVT has no advantage over calling GEMV and if blocking parameters are badly chosen, may slow the parallel computation.

### 3 Serial Code for LAPACK

We have serial code Fortran 77 code that replaces the LAPACK bidiagonalization code DGEBRD. Installing it in LAPACK requires some changes also to the ILAENV code, allowing choice of blocking parameters for DGEMVT and DGEMVER. Additionally, incorporation into LAPACK requires introducing DGEMVT and DGEMVER as auxiliary routines.

Currently the code exists in double precision and double precision complex versions, with NAG able to use software to automatically generate single precision floating and complex routines.

Incorporation of the routines into LAPACK mainly requires some design decisions. We get significant speedups over the current DGEBRD when the matrix  $A$  to be reduced to upper bidiagonal form has a number of rows greater than or equal to the number of columns. For the case of more columns than rows, we get good speedups (on most architectures) only by doing in-place matrix transpositions. In some instances, such transpositions may be acceptable. For example, when a matrix is square and the desired reduction is to lower bidiagonal (as when the initial matrix has many more columns than rows, and the initial reduction is to a square matrix). In other instances, in-place transposition can require extra storage and is therefore not feasible.

For a more complete explanation of the techniques we use, see some of the papers referenced below, in particular, `bidiag9.ps` available from the web site `/newline www.ncsu.edu/itd/hpc/Publications/gary_howell/contents.html` Some figures here show the speed-ups on current Xeon and Opteron processors.

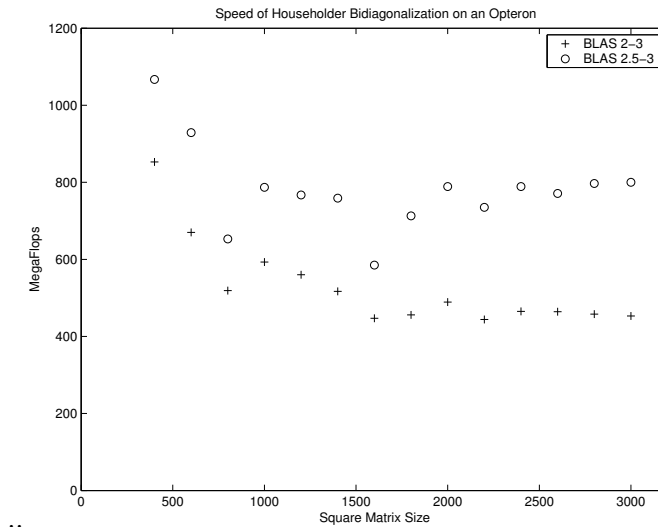


Figure 1: Comparison of speed of a BLAS 2-3 algorithm (LAPACK) and BLAS 2.5-3 algorithm, using Intel compiler, ATLAS BLAS, on a 2 GHz Opteron with a 2 Mbyte cache.

GEMVT speeds the serial computation by combining to GEMV calls into one

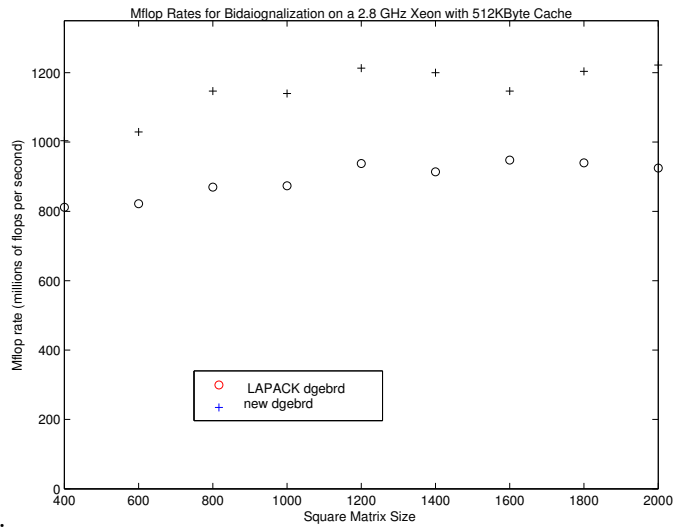


Figure 2: Comparison of speed of a BLAS 2-3 algorithm (LAPACK) and BLAS 2.5-3 algorithm, using g77 compiler, ATLAS BLAS, on a 2.8 GHz Opteron with a 512 KByte cache.

GEMVT call. Explicitly, we perform

$$v \leftarrow \alpha A^T x + \beta y$$

followed by

$$w \leftarrow \gamma A v$$

in one call by blocking the matrix  $A$  into column blocks.

$$A = [A_1 | A_2 \dots | A_k]$$

and similarly  $v^T$  as

$$v^T = [v_1^T | v_2^T \dots | v_k^T]$$

and performing two matrix vector multiplications on each column. In GEMVT call can then be performed by the loop:

```

for  $j = 1 : k$ 
   $v_j^T = \alpha * x^T A_j + \beta * y_j$ ;
   $w = w + \gamma * A_j v_j$ ;
end

```

If each column block  $A_j$  is small enough to fit in cache, then the matrix  $A$  need only be read from RAM to cache once in accomplishing both the matrix vector multiplications.

## 4 Parallel

We don't yet have a parallel bidiagonalization. But we have implemented GEMVT in parallel to see if use of GEMVT can improve the efficiency of parallel bidiagonalization.

For scalability, it's important that the volume of communication increase only as the log of the number of processors. This can be accomplished with two calls to GEMV, so long as each processor only gets the parts of a vector that it needs and so long as the size of local vectors do not grow with the number of processors.

Thus the column blocking scheme used in the serial case scales well to many processors only if the column blocks are spread across processors. Then a communication (e.g., an MPI\_Allreduce) occurs between the computations of  $v$  and  $w$  in the above snippet of pseudo code. Whether the parallel GEMVT call is preferable to two calls to GEMV depends on the machine architecture. (Is it faster to do the communication or is it faster to transfer more data from RAM to cache). On fast communication, large cache architectures, GEMVT turns out to make sense as the accompanying figure illustrates.

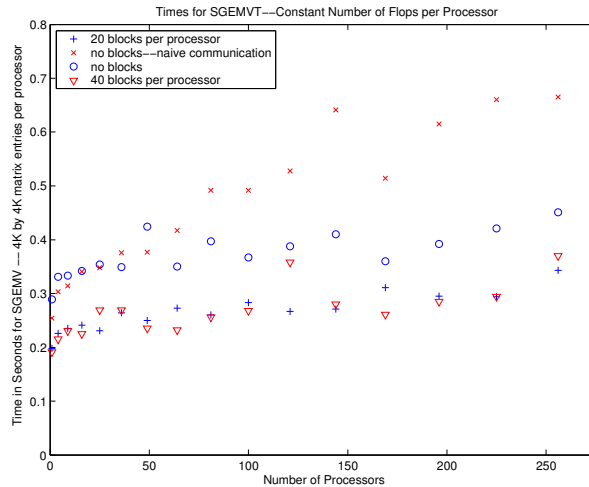


Figure 3: Blocking for Parallel cache-efficiency with MPI code on a 1 GHz Alpha SC. In this case, we are holding the problem size per processor constant as the problem size increase. The y-axis is time, so in this case, lower values are better. The cases of 20 and 40 blocks to a processor correspond to parallel GEMVT and are much faster than the single block algorithm. In the naive communication pattern, the communication time goes like  $\sqrt{p}$ . For the block cyclic layout, the communication times go like  $\log p$

In the recent thesis of Madhan Premkumar, this idea is extended to work with PBLAS, and in the block cyclic case. This is an essential step in extending the GEMVT based bidiagonalization algorithm to SCALAPACK.

## 5 Ongoing Work

Jack Dongarra and Jim Demmel have an ongoing NSF project (ST-HEC: Reliable and Scalable Software for Linear Algebra Computations on High End Computers – <http://www.cs.berkeley.edu/~demmel/Sca-LAPACK-Proposal.pdf>) to update the LAPACK package, partly to incorporate this work.

PIs Gary Howell, Charles Fulton, Xiaobai Sun and Nikos Pitsianis (the last two from Duke University), with consultants Jim Demmel, Sven Hammarling, and Bruno Lang have submitted an NSF proposal to ensure that the LAPACK version is automatically self-tuning, choosing the best bidiagonalization (speed depends on blocking parameters for the GEMVT) or band reduction scheme. That proposal will also extend efficient bidiagonalization techniques to the sparse and parallel cases. Additionally, we will explore how best to use high bandwidth graphics architectures in performing numerical linear algebra, implementing a prototype BLAS package. Our collaboration with the rest of the LAPACK team will enable implementations adapted across architectures. The proposal is MSPA-MCS Bidiagonalization and PCA: Algorithms and Architectures, online at [http://www.ncsu.edu/itd/hpc/Publications/gary\\_howell/finalpr](http://www.ncsu.edu/itd/hpc/Publications/gary_howell/finalpr)

## 6 Publications, Theses and Presentations

Much of the work described here is available from the web page [http://my.fit.edu/beowulf/research\\_publications/research\\_pub.html](http://my.fit.edu/beowulf/research_publications/research_pub.html) (referred to below as “blue marlin web site”) and the web page [http://www.ncsu.edu/itd/hpc/Publications/gary\\_howell/contents.html](http://www.ncsu.edu/itd/hpc/Publications/gary_howell/contents.html) (referred to below as “NCSU HPC web site”).

- P. Angeli, O. Basset, C. Fulton, G. Howell, R. Hsu, D. Richardson, A. Sawetprawhichkal, M. Schuster, H. Thompson, and S. Wilberscheid “Some Issues in Efficient Implementation of a Vector Based Model for Document Retrieval”, presented by Don Richardson at the 2001 International Systems and Engineering (ISE’2001) June 25-28, 2001. Monte Carlo Resort, Las Vegas, Nevada, USA. Blue-Marlin web site, NCSU HPC web site as `sparmul.ps` (pdf).
- P. Escallon, “Reduced Index Sparse Representation in a Parallel Environment”, CS Master’s Thesis, Florida Institute of Technology, Dec. 2004, blue marlin web site.
- G. Howell, C. Fulton, J. Parker, and S. Malhotra, “Cache Efficient and Parallel Householder Bidiagonalization”, SIAM Conference on Applied Linear Algebra, July 15-19, 2003, Williamsburg, VA – blue marlin, NCSU HPC web sites as `siam2003_3.ps` (pdf)
- G. Howell, C. Fulton, and M. Premkumar – “Parallel GEMVT-Cache Efficiency in Combined Left and Right Matrix Vector Multiplications” Draft document with code, results, and analysis from Dec. 2003, an initial draft of paper, NCSU HPC web site as `pargemvt.pdf`
- G. Howell, J. Demmel, C. Fulton, S. Hammarling, K. Marmol, Cache Efficient Bidiagonalization Using BLAS 2.5 Operators, being submitted to ACM

Transactions on Mathematical Software – blue marlin and NCSU HPC web sites as bidiag9.ps (pdf) – latest revision is found at <http://my.fit.edu/~cfulton/gr0130624.html>

- G. Howell, "Sparse Householder Bidiagonalization", presented at CERFACS Sparse Day, Toulouse, FRANCE, June 15, 2001, NCSU HPC web site as cerfacs01.ps (pdf)
- G. Howell and N. Diaa, "Algorithm 841: Gaussian Reduction to a Similar Hessenberg form", ACM Transactions on Mathematical Software March 2005, Vol. 31, no. 1, an HPC NCSU web site draft is bhess.ps (pdf).
- Gary Howell, communications with the BLAST committee, functionality.ps, functionality2.ps, march99blas.tex, from the anonymous ftp site.
- Sumit Malhotra, "Parallelization of BLAS 2.5 Operator GEMVT", CS Master's Thesis, Florida Institute of Technology, July 2003 - blue marlin web site.
- Madhan Premkumar, "A Parallel and Cache-Efficient BLAS 2.5 Operator GEMVT, CS Master's Thesis, Florida Institute of Technology, May 2005 – blue marlin web site.

## 7 Coded Algorithms

- Gary Howell and Nadia Diaa. bhess.tar.gz from HPC NCSU.
- Gary Howell. Matlab "pseudo code" for cache efficient Householder bidiagonalization – pseudo011905.tar from HPC NCSU.
- Gary Howell, Charles Fulton, Sven Hammarling, and Karen Marmol. 030905.tar, from HPC NCSU. These are LAPACK compatible fortran 77 files to do bidiagonalization. (The files on the web page are the ones for the case of more rows than column, double precision). Others include several versions of the more columns than rows case and also the double precision case).
- Madhan Premkumar, PBLAS versions of DGEMVT and DGEMVER, for blue marlin.

## 8 Supported Students

- Jim Parker (American citizen) worked both on this project and as a graduate student in oceanographic engineering, graduating with a master's degree in oceanographic engineering. He helped us with automated tuning in interfacing with the ATLAS BLAS and gave us the perspective of building his own Beowulf.
- Sofia Wilberscheid (American citizen) helped build automated testing routines. The work enabled her to get enough computer science experience that on graduating with a master's in math, she went to work teaching computer science at Indian River Community College.

- Madhan Premkumar has accepted a position with Omegassi Co. in Olney, Maryland. His work on this project helped him secure a summer 2004 internship at Lawrence Livermore Laboratory. He will be a co-author.
- Sumit Malhotra wrote a master's thesis, graduating in 2004, and going to work as a computer professional in Orlando.
- Don Richardson (American citizen and disabled veteran) is a PhD student at Florida Tech.

## 9 Some Associated Students

- Ron Hsu (American citizen, then a high school student) was one of the authors of a paper supported by this grant. Under our direction, he extended the work to win the 5K Yang Science prize, and went on to participate in the international science fair. He is currently an undergraduate at Harvard.
- Brittany Owens (American citizen, minority undergraduate at Hampton Roads, Va.) while working as an intern for Gary Howell at ERDC in 2003, developed matlab code extending the cache efficient approach of our algorithm to apply to the reduced computation bidiagonalization (one side Householder, the other side modified Gram-Schmidt) algorithm Jesse Barlow showed to be backward stable. The algorithm rearrangement makes it likely that the reduced computation algorithm is competitive in speed.
- Herbert Thompson (American citizen), one of the paper co-authors has gone on to publish a Dr. Dobbs article on computer security and is working in a startup.
- A. Sawetprawichkal got a PhD in mechanical engineering from Florida Tech.
- Mark Schuster (American citizen) went on to do a master's with C. Fulton in math and now works for Harris Corp. as a system administrator.
- Karen Marmol (American citizen) completed a master's in math at Florida Tech and now works at Harris Corporation developing algorithms and codes for analysis of satellite imagery.
- Olivier Basset was named the outstanding undergraduate student at Florida Tech at the time of the paper and went on to graduate school in computer science.