Performance of Parallel Eigensolvers on Electronic Structure Calculations

Robert C. Ward[*§], Yihua Bai[*§] and Justin Pratt[*§]

Technical Report UT-CS-05-560[1]

University of Tennessee

February 2005

Abstract. Many models employed to solve problems in quantum mechanics, such as electronic structure calculations, result in nonlinear eigenproblems. The solution to these problems typically involves iterative schemes requiring the solution of a large symmetric linear eigenproblem during each iteration. This paper evaluates the performance of various popular and new parallel symmetric linear eigensolvers applied to the Self-Consistent Field procedure in electronic structure calculations on the distributed memory supercomputers at the Oak Ridge National Laboratory. Results using established routines from ScaLAPACK and vendor optimized packages, are presented, as well as from two recently developed parallel eigensolvers, the method of Multiple Relatively Robust Representations using PLAPACK support routines and the block divide-and-conquer algorithm.

1. Introduction. The problem of describing the motion of N electrons in the field of M fixed nuclear point charges is a central problem in quantum chemistry. It translates into the task of finding and describing approximate solutions of the electronic Schrödinger equation. The solutions to this equation involving the electronic Hamiltonian are the electronic wave functions, which describe the motion of electrons, and the electronic energy. The wave function for a single particle is called an orbital.

Using the Hartree-Fock approximation [25, 27] in the electronic Schrödinger equation leads to the nonlinear Hartree-Fock equation, a spatial integro-differential equation for the orthonormal Hartree-Fock orbitals and the corresponding orbital energies. The N orbitals with the lowest energies are called the occupied orbitals. In practice, the solutions of this equation are approximated by introducing a finite set of n basis functions, expanding the unknown molecular orbitals in terms of this basis, and converting the Hartree-Fock equation to a set of algebraic equations. The problem of computing the molecular orbitals then reduces to the problem of computing the matrix C of expansion coefficients, which can be formulated as the *Roothaan equations*

$$F(C)\,C = S\,C\,E \tag{1}$$

---

[*]Department of Computer Science, University of Tennessee, 203 Claxton Complex, 1122 Volunteer Blvd., Knoxville, TN 37996-3450.

[§]This work was partially supported by Oak Ridge National Laboratory's Computing and Computational Sciences Directorate and the University of Tennessee's Science Alliance Program.

[1]Available from: http://www.cs.utk.edu/~library/TechReports.html

with the Hermitian (usually real symmetric) n x n matrices S (*overlap matrix*) and F (*Fock matrix*). S is positive definite with unit entries along the diagonal, and its off-diagonal entries satisfy $|S_{ij}| < 1$ for $i \neq j$. E is a diagonal matrix that contains the orbital energies along the diagonal, and C contains the expansion coefficients columnwise.

1.1 The Self-Consistent Field Method. The Roothaan equations (1) establish a generalized eigenproblem with the unknowns E and C as the eigenvalues and eigenvectors, respectively. Since F depends on the coefficient matrix C, this is a nonlinear eigenproblem. Its solution is a very central and time-consuming task arising in many quantum chemistry applications. We have described the problem as it is normally derived in Hartree-Fock theory, however, very similar equations occur in density functional theory [23, 11] and in semi-empirical quantum chemistry [24, 8].

The standard approach in quantum chemistry for solving (1) is the *self-consistent field (SCF)* method, summarized in Algorithm 1.1. After reduction of the generalized nonlinear eigenvalue problem (1) to a standard nonlinear eigenproblem, for example, using a Cholesky factorization of the overlap matrix S, this nonlinear problem is transformed into a linear eigenproblem using an initial guess C_0 for the expansion coefficients. The linear eigenproblem is solved, the new expansion coefficients C_1 are computed from the eigenvectors of the linear eigenproblem, and a new linear eigenproblem is formulated. This procedure is iterated until *self-consistency* is achieved, i.e., until the new expansion coefficients C_k are close enough to the old expansion coefficients C_{k-1} or the new computed electronic energy is sufficiently close to the computed electronic energy in the previous iteration.

Input: *guess initial values* C_0
1. *factorize* $S = U\,U^T$
2. *transform (1) into a standard problem* $A(C)\,V = V\,E$
3. **repeat** $k = 1,2, \ldots$
   - *(i) compute* $A(C_{k-1})$
   - *(ii) solve* $A(C_{k-1})\,V = V\,E_k$
   - *(iii) compute* $C_k = U^{-T}\,V$

   until *converged*

Output: *electronic energy, orbitals*

Algorithm 1.1 Self-Consistent Field Method

The iterative Algorithm (1.1) does not always converge, for example, if the initial guess is poor. Various techniques have been suggested for ensuring or accelerating convergence [18]. As a convergence criterion, it is common to require that two successive values of the total energy do not differ by more than a tolerance δ. A value of $\delta = 10^{-6}$ is adequate for most purposes [27]. Alternatively, it is also possible to require that the change $\|C_k - C_{k-1}\|$ be bounded in some norm.

1.2 Synopsis. The purpose of this paper is to report on the performance evaluation of various parallel standard symmetric eigensolvers as they solve the linear problem given in Step 3(ii) of Algorithm 1.1. The tests will involve applying the SCF method to molecules from various types of materials using the distributed memory supercomputers located in the Center for Computational Sciences (CCS) [1] at the Oak Ridge National Laboratory (ORNL). Due to its architecture and maturity, our tests will concentrate on the IBM pSeries system.

The algorithms will be briefly described in Section 2, the test matrices in Section 3, and the supercomputers in Section 4. Section 5 will present the test results and provide summary comments.

2. Parallel Eigensolvers. For our performance tests, we include the two parallel eigensolvers that are readily available to the users of the distributed memory supercomputers located in ORNL's CCS. Both of these have been tuned for improved performance through the use of BLAS and PBLAS optimized by the computer vendor. In addition, we will include an initial parallel implementation of a recent algorithm, the method of Multiple Relatively Robust Representations, and an initial parallel version of the block divide-and-conquer algorithm. The salient features of each algorithm will be given below with algorithmic and parallel implementation details provided in the cited references.

2.1 PDSYEVD. The parallel eigensolver, PDSYEVD, used in our experiments is found in ScaLAPACK [6] and is based upon the divide-and-conquer algorithm [7]. It consists of the classical three steps: reduction to symmetric tridiagonal form, eigen-decomposition of the tridiagonal matrix, and back-transformation of the eigenvectors. The eigen-decomposition of the tridiagonal matrix is computed by the Tisseur-Dongarra parallel implementation [28] of the divide-and-conquer algorithm using the Gu-Eisenstat method [16] for stably computing the eigenvectors. The reduction to tridiagonal form does not incorporate some of the more recent methods [5, 12] designed to increase the number of BLAS3 operations. Algorithms implementing all three steps use a vendor-optimized version of BLAS on all three (IBM, SGI, and Cray) distributed-memory supercomputers at ORNL.

2.2 PDSYEVX. The parallel eigensolver, PDSYEVX, used in our experiments is found in PESSL [2], which is a library of parallel solvers for scientific problems supplied by IBM. PESSL incorporates optimizations for the intended computational platform. PDSYEVX also consists of the classical three steps described above but uses a different algorithm for the eigen-decomposition of the symmetric tridiagonal matrix. The bisection method [19] is used for computing its eigenvalues and inverse iteration (with re-orthogonalization for tightly clustered eigenvalues) [19] is used for computing its eigenvectors. The same algorithms as in PDSYEVD are used for the reduction and back-transformation steps.

2.3 PMR3. Significant progress has been made in computing the eigensystem of symmetric tridiagonal matrices. The method of Multiple Relatively Robust Representations (MRRR) [21, 9, 20, 10, 22] has greatly improved the efficiency of these solvers while maintaining accuracy and obtaining eigenvector orthogonality. The MRRR provides an $O(n^2)$ algorithm for computing the n eigenpairs to full accuracy and eigenvectors to numerical orthogonality. Since the solution to the full eigenvector problem generally requires the computation of n^2 values, this algorithm is considered optimal, hence its informal name of "Holy Grail."

The parallel MRRR algorithm used in our tests, PMR3, is described in [4]. It is the first parallel implementation of an MRRR algorithm and was obtained from the PLAPACK researchers at the University of Texas. The same basic algorithms as used in PDSYEVD and PDSYEVX, but with different implementations, are used to reduce the full matrix to tridiagonal form and back-transform the eigenvectors of the tridiagonal matrix to those of the original matrix [15].

2.4 PBDnC. The recently developed block divide-and-conquer algorithm [14, 13] has proven very attractive for computing eigensystems of symmetric, block tridiagonal matrices with reduced accuracy requirements, that is, accuracy less than machine precision. This algorithm has been used with good results in electronic structure calculations[13]. An initial parallel implementation of this algorithm [3] has recently been developed. Since parallel algorithms for reducing dense matrices to block tridiagonal form are still under development, PBDnC can only be tested on matrices that are already in nontrivial block tridiagonal form (more than 2 blocks) or can be placed in nontrivial block tridiagonal form without affecting the required accuracy, e.g., when the elements outside the block tridiagonal are small enough to be set to zero.

3. Test Matrices. The test matrices used in this eigensolver performance study come from various molecule families and derivation schemes. A brief description of each is given in the subsections below along with a picture of one normalized Fock matrix (produced by Step 2 in algorithm 1.1) from each family. The picture uses a color to indicate the magnitude of the exponent of the element in that matrix position. The color-coded scale bar to the right of the picture shows the value of the exponent. Pictures of different normalized Fock matrices in the same family are very similar. The major difference in test matrices in a family comes from the number of molecules in the model and, except for the impure hydrogen family, result in matrices of different orders. Note that the orders of the matrices range from fairly small (i.e., 1934) to reasonably large (i.e., 16,000).

3.1 Alkane Family. The alkane family consists of acyclic hydrocarbons in which the molecule has the maximum number of hydrogen atoms – hence has no double bonds. The general formula for alkanes is C_nH_{2n+2}. The nonlinear eigenproblem for this family has been derived from the semi-empirical method CNDO, and as such, does not have an overlap matrix. Thus, there is no normalization (steps 1 and 2 of Algorithm 1.1) of the Fock matrix.

Two Fock matrices from this family were included in our performance study. One came from the $C_{322}H_{646}$ alkane with $n = 1934$ and the other from $C_{502}H_{1006}$ with $n = 3014$. Figure 3.1 shows a picture of the Fock matrix for $C_{322}H_{646}$ using 1934 functions in the basis set.

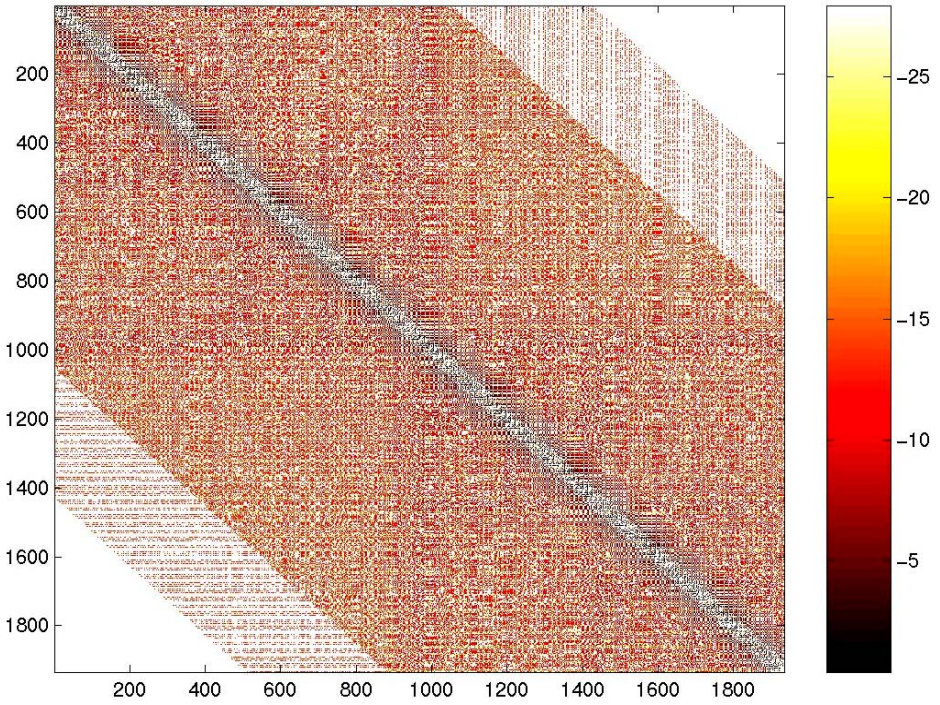

Figure 3.1 Element Magnitudes of the Initial Fock matrix for the $C_{322}H_{646}$ Molecule in the Alkane Family

3.2 Polyalanine Family. The matrices in this family were formed from modeling polypeptides made solely from alanine. Linear polyalanine chains of differing lengths were constructed, then a classical force field was used to randomize the chain conformations so that the molecules were no longer linear. The MINDO method was then used to construct the Fock matrices, all of which had a bandwidth of 79.

Our performance study includes two Fock matrices from this family. One comes from a polyalanine chain of length 125, which yields a matrix size of 3152, and the other from a chain of length 200, resulting in a matrix size of 5027. Figure 3.2 shows a picture of the magnitude of the elements in the Fock matrix produced by the chain of length 125.

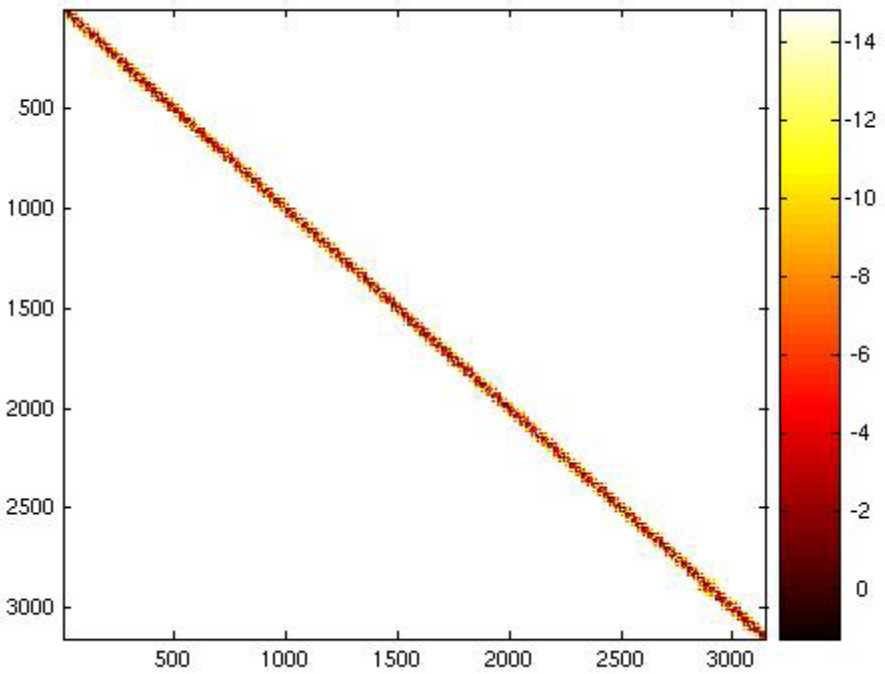

Figure 3.2 Element Magnitudes of the Fock matrix for the Polyalanine Chain of Length 125

3.3 Silicon Crystal Family. This family of test matrices has been generated using the PBE form of Density Functional Theory on Silicon crystals with differing numbers of atoms. We denote the different problems in the family by the number of unit cells in each of the x, y, and z directions. Thus, the 111 problem has one unit cell in each of the directions, and the 432 problem has 4 unit cells in the x direction, 3 in the y direction, and 2 in the z direction. Each unit cell has 8 atoms using 104 basis functions from the Double Zeta basis set. Thus, the 111 problem models 8 atoms with 104 basis function resulting in an eigenproblem of order 104, and the 443 problem models 384 atoms with 4992 basis functions resulting in an eigenproblem of 4992.

We include two Fock matrices from this family in our tests: the 443 and 544 Silicon crystals. The 443 Fock matrix is of order 4992, and the 544 is 8320. Figure 3.3 shows a picture of the magnitude of the elements in the initial normalized Fock matrix produced from the 443 lattice.

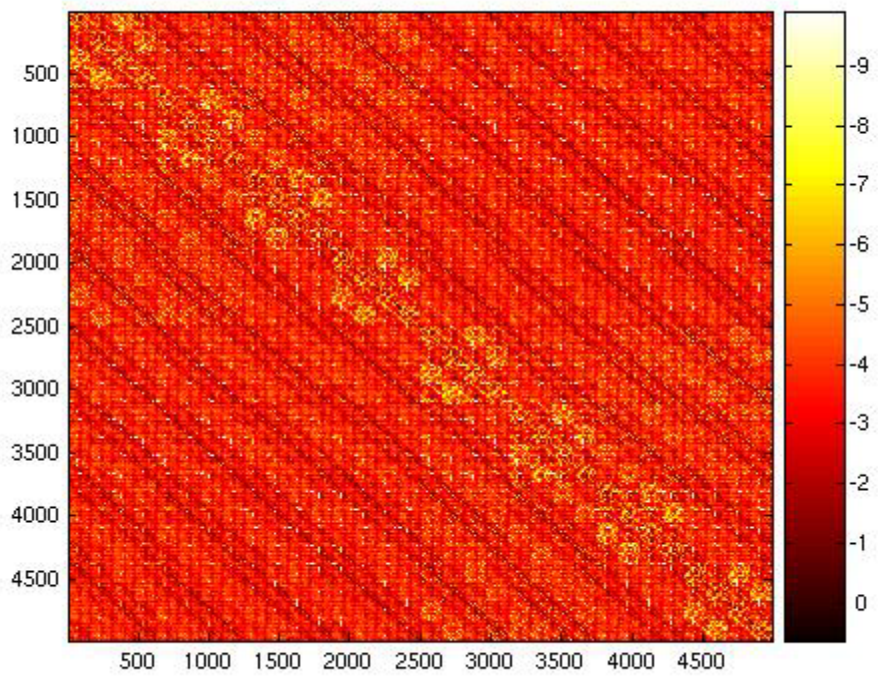

Figure 3.3 Element Magnitudes of the Initial Normalized Fock matrix for the 443 Lattice in the Silicon Crystals Family

3.4 Hydrogen Molecules with Impurities. These test matrices are derived from a finite-basis representation of the vibrational Hamiltonian of a randomly close, packed solid sample of hydrogen molecules H_2 in which 0.1% of the molecules have been replaced by impurities [17]. The solid is represented by a collection of 21 x 24 x 26 molecules on a lattice with periodic boundary conditions imposed in all three dimensions. The off-diagonal elements represent couplings between nearby H_2 molecules and decrease in magnitude the further they appear from the diagonal. Different matrices in this family are produced by including more distant neighbors in the model. All the matrices are of order 13,104.

We include two matrices from this family in our tests. The first includes multiple layers of molecular couplings producing a matrix with 0.42% of its elements nonzero. The matrix contains 716,273 nonzero elements that are all between -10 and -0.015. Its zero-nonzero structure is given in Figure 3.4. The second is the most dense matrix in the family and includes all molecular pairs with a coupling greater than or equal to -10^{-5}. The matrix contains 24,231,553 nonzero elements, i.e., about 14%, with values between -10 and -10^{-5}.

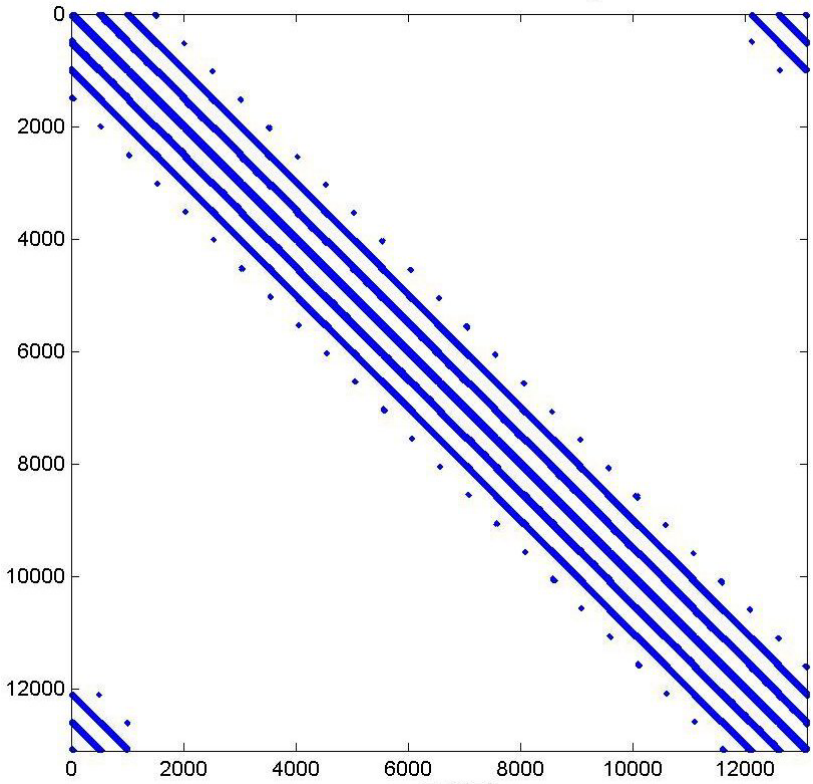

Figure 3.4 Zero-Nonzero Structure of Hamiltonian from Impure Hydrogen Lattice with 0.42% sparsity (Blue for nonzero, White for zero)

3.5 trans-Polyacetylene Family. Polyacetylene (PA) has been the subject of intensive investigation for many years. It shows a very fast response time upon laser excitation, making it a candidate for fast-optical switching and optical computation. Our test matrices come from the trans-form of PA with C-H chains sufficiently long that end effects may be ignored. The SSH Hamiltonian[26], which is a tight-binding approximation and includes only nearest neighbor atoms, is used within the Hartree-Fock approximation framework. As a tight-binding approximation, each cell unit has only one basis function; therefore, the size of the matrix is the number of atoms being modeled.

We include two matrices from this family in our tests: one with 12,000 atoms and one with 16,000. Figure 3.5 shows a picture of the magnitude of the elements in the initial Fock matrix produced from the 12,000-atom problem.

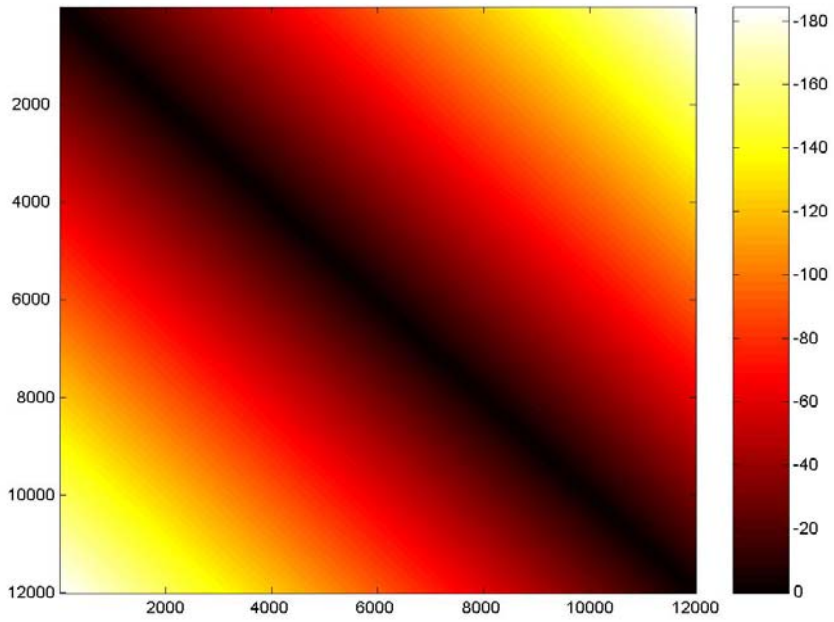

Figure 3.5 Element Magnitudes of the Fock matrix for a trans-Polyacetylene with 12,000 atoms

4. Supercomputers. The test cases were run on the three distributed memory supercomputers in the Center for Computational Sciences at the Oak Ridge National Laboratory. Brief descriptions of these systems are given below.

4.1 IBM 4.5 Teraflops pSeries System. The system, installed in 2002 and nicknamed Cheetah, consists of twenty-seven p690 nodes of thirty-two 1.3 GHz IBM Power4 processors each for a total of 864 processors. There are 2 processors per chip. All nodes are connected via IBM's Federation interconnect. With each processor having a peak rating of 5.2 GFlops, the peak computational power of the system is 4.5 TFlops. The system has a peak LINPACK rating of 2.3 TFlops.

Twenty of the nodes have 32 GB of SMP memory each, five nodes have 64 GB of SMP memory, and two have 128 GB; thus, the total available main memory on the system is 1.2 TB. Access to data residing off-node is via the interconnect at a slower speed, resulting in a non-uniform memory access (NUMA) system. In addition, 14 nodes have 160 GB of local temporary disk space.

The Power4 memory hierarchy consists of 3 levels of cache. The first and second levels are on-chip. The split L1 instruction and data caches are 128 KB and 64 KB respectively or 64 KB and 32 KB respectively per processor. The unified L2 cache is 1.5 MB and is shared between the 2 processors. The L3 cache shared by the 2 processors is 32 MB and is located on a separate chip.

4.2 SGI 1.5 Teraflops Altix. The system, installed in 2003 and nicknamed Ram, consists of 256 Intel Itanium2 processors running at 1.5 GHz. With each processor having a peak rating of 6 GFlops, the peak computational power of the system is 1.5 TFlops. The system has a peak LINPACK rating of 1.3 TFlops. Currently, only a maximum of 128 processors, which form a cluster, can be used together in parallel.

Ram is a cache-coherent, shared-memory system with 8 GB of main memory per processor for a total of 2 TB of system memory. Memory is physically distributed with different access times for local and remote memory, resulting in a NUMA system. The Itanium2 cache hierarchy consists of 32 KB of on-die equally split L1 instruction and data caches, 256 KB of on-die unified L2 cache, and 6 MB of on-die L3 cache.

4.3 Cray 6.4 Teraflops X1. The system, installed/upgraded in 2003/2004 and nicknamed Phoenix, consists of 512 multi-streaming processors (MSP) with 4 MSPs forming a node. Each MSP has four single-steaming processors (SSP), each with 2 floating-point vector units and one super-scalar unit. Each SSP is capable of 3.2 GFlops on 64-bit operations, yielding a system peak performance of 6.4 TFlops. The system has a peak LINPACK rating of 5.9 TFlops.

Four MSPs and 16 GB of flat, shared memory form a Cray X1 node. The interconnect system within a node provides 200 GB/s bandwidth with the off-node interconnect system providing 50 GB/s. The Cray X1 memory system is globally addressable and distributed across nodes with nonuniform access time. The four SSPs in an MSP share 2 MB of cache.

5. Test Results. Given the difference in architectural features of the three supercomputers, the degree of algorithmic optimization present in the available algorithms, and the software available, we will report the performance results from the IBM Cheetah separately since these tests were more comprehensive.

5.1 Tests On IBM Cheetah.

Our tests on Cheetah invoked the Fortran versions of PDSYEVD, PSYEVX and PBDnC through version 8.1 of IBM's `xlf` compiler and C versions of PMR3 and PLAPACK through version 6 of IBM's `xlc` compiler. Both compilers were set to the default 32-bit compile mode and linked to the 32-bit PESSL library. The compiler options for PDSYEVD, PDSYEVX and PBDnC were:

```
-O4 -qarch=auto -qcache=auto -bmaxdata:0x70000000
```

and for PMR3 and PLAPACK:

```
-O4 -qtune=pwr4 -qarch=pwr4 -bmaxdata:0x70000000
```

.

The wall-clock times (in seconds) for the different algorithms using consecutive powers of 2 as the number of processors are given in tabular form for each test matrix followed by graphical representations of their performance. For the smaller matrices, we start with 4 processors and go up to 512; we start with more processors for the larger matrices.

For an estimate of the accuracy of the computed eigensystem, we will use the following two scaled values:

$$\mathcal{R} = \max_{i,j=1,2,\cdots,n} \left[\left| A - VEV^T \right| \right]_{ij} \Big/ \max_i |E_{ii}|$$

$$\mathcal{O} = \max_{i,j=1,2,\cdots,n} \left[\left| V^T V - I \right| \right]_{ij} \Big/ n \ ,$$

where A is the Fock matrix, V is the eigenvector matrix, and E is the diagonal matrix of eigenvalues. We use $\mathcal{R}$ and $\mathcal{O}$ as accuracy indicators rather than including the eigenvalue gaps [19] since these measures are frequently used in applications and are sufficient in many cases. For each family of test matrices, we provide one table giving the maximum $\mathcal{R}$ and $\mathcal{O}$ over all the computed eigensystems in that family.

Since the PBDnC algorithm requires the matrix to be in block tridiagonal form, only the larger polyalanine Fock matrix and the two polyacetylene Fock matrices were run using this algorithm. These matrices can be easily represented in nontrivial block tridiagonal form without affecting the accuracy of the eigensystem to machine precision, although we only asked PBDnC to compute the eigensystem to an accuracy of 10^{-6}. For the polyalanine (n = 5027), we used a block size of 104 for the first and last diagonal blocks and 79 for all other diagonal blocks. All the blocks in both polyacetylene matrices were of size 500.

PMR3 ran into some difficulty on some of the test cases and did not complete the computations, ending with an error. A reasonable amount of effort was spent trying to solve the problems without success. The errors appear to occur in the reduction and back-transformation routines rather than the PMR3 tridiagonal eigensolver and appear to be a portability problem since some of these tests ran successfully on a smaller cluster. These tests are shown with a DNF (did not finish) tag for the wall clock time. We anticipate these problems will disappear when PLAPACK is formally ported to these architectures.

5.1.1. Alkane Family.

# procs	PDSYEVD	PDSYEVX	PMR3
4	19.8	29.9	7.7
8	5.3	8.5	7.2
16	3.2	4.8	7.1
32	2.2	6.3	3.7
64	2.0	5.8	4.1
128	2.2	11.0	5.2
256	2.7	13.1	6.9
512	5.2	21.7	9.4

Table 5.1 Wall Clock Time in Seconds for $C_{322}H_{646}$

# procs	PDSYEVD	PDSYEVX	PMR3
4	20.3	68.2	DNF
8	16.8	26.2	19.8
16	10.4	12.4	DNF
32	5.6	12.2	DNF
64	4.6	10.3	8.2
128	5.4	17.1	10.1
256	5.7	19.4	11.7
512	11.8	33.4	15.2

Table 5.2 Wall Clock Time in Seconds for $C_{502}H_{1006}$

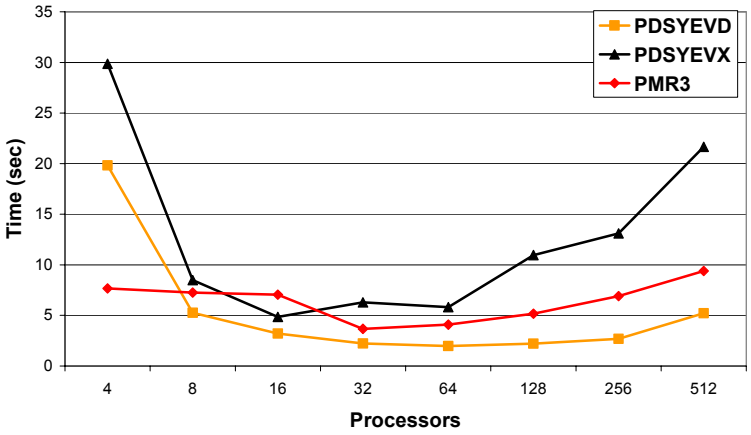

Figure 5.1 Plots of Wall Clock Time for $C_{322}H_{646}$

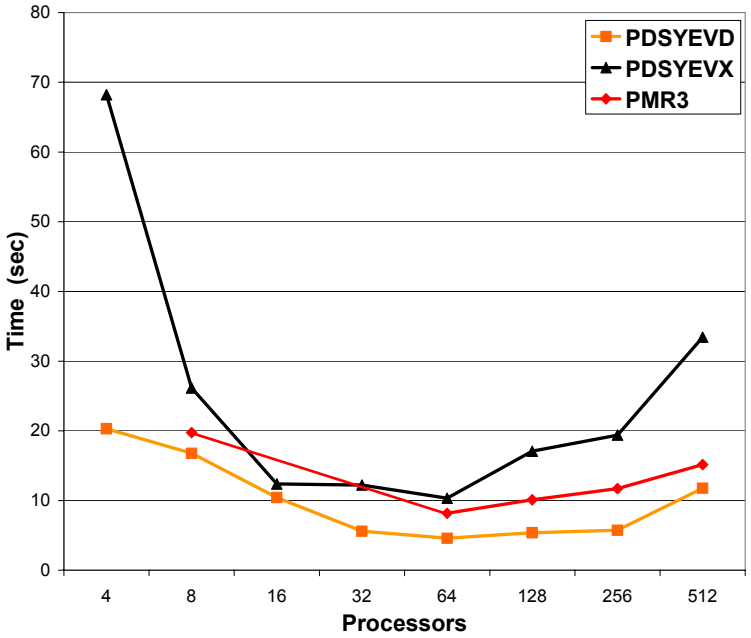

Figure 5.2 Plots of Wall Clock Time for $C_{502}H_{1006}$

	$\mathcal{R}$	$\mathcal{O}$
PDSYEVD	2.53E-15	1.84E-18
PDSYEVX	8.56E-15	1.28E-17
PMR3	1.46E-13	2.29E-15

Table 5.3 Maximum Residual and Orthogonality Results for Alkanes

5.1.2 Polyalanine Family.

# procs	PDSYEVD	PDSYEVX	PMR3
4	28.1	45.9	32.0
8	17.9	23.4	DNF
16	10.3	14.0	12.3
32	6.2	14.0	DNF
64	5.7	12.0	9.2
128	4.5	19.9	11.5
256	4.7	19.4	12.9
512	7.0	37.5	19.0

Table 5.4 Wall Clock Time in Seconds for Polyalanine Chain of Length 125

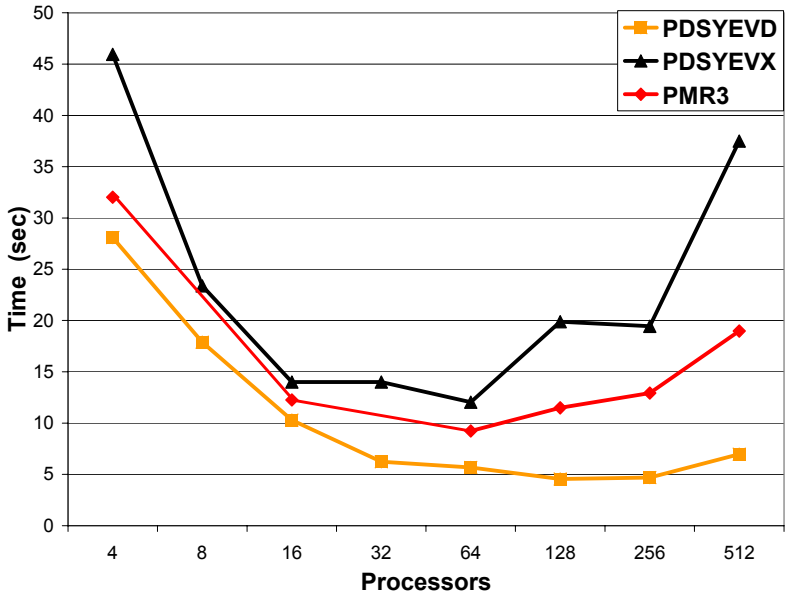

Figure 5.3 Plots of Wall Clock Time for Polyalanine Chain of Length 125

# procs	PDSYEVD	PDSYEVX	PMR3	PBDnC
8	56.6	84.9	55.6	23.0
16	33.8	39.8	DNF	16.2
32	17.9	33.8	24.0	8.7
64	14.2	25.6	DNF	6.3
128	10.5	35.7	21.7	5.1
256	10.2	33.3	26.1	3.5
512	12.4	62.8	30.8	4.8

Table 5.5 Wall Clock Time in Seconds for Polyalanine Chain of Length 200

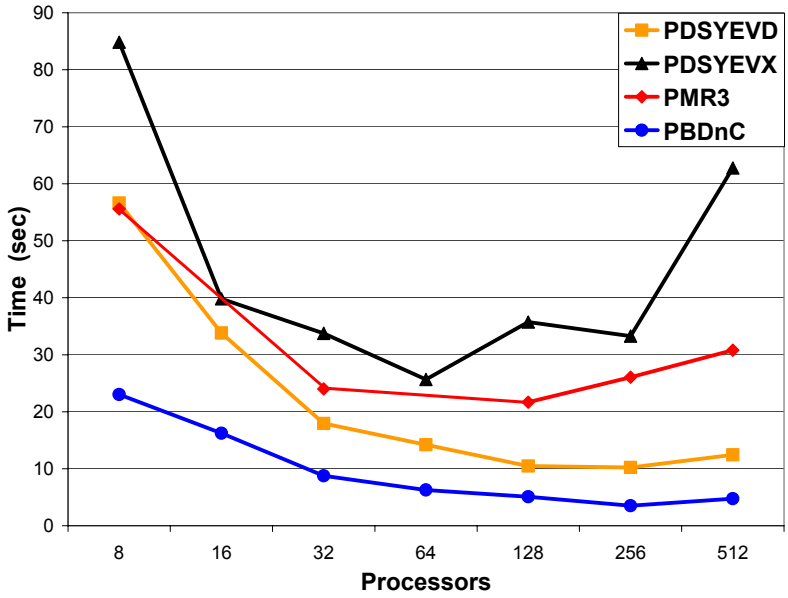

Figure 5.4 Plots of Wall Clock Time for Polyalanine Chain of Length 200

	$\mathcal{R}$	$\mathcal{O}$
PDSYEVD	1.14E-14	3.66E-18
PDSYEVX	3.62E-11	2.42E-14
PMR3	6.53E-12	1.30E-14
PBDnC	4.72E-06	9.51E-18

Table 5.6 Maximum Residual and Orthogonality Results for Polyalanines (Note that PBDnC was given an accuracy request of 10^{-6}.)

5.1.3 Silicon Crystal Family.

# procs	PDSYEVD	PDSYEVX	PMR3
8	70.0	63.6	51.1
16	35.1	41.3	DNF
32	18.1	32.9	22.8
64	14.1	24.6	DNF
128	10.6	36.1	22.1
256	13.8	33.3	25.9
512	12.5	62.5	30.9

Table 5.7 Wall Clock Time in Seconds for 443 Lattice

# procs	PDSYEVD	PDSYEVX	PMR3
16	147.2	179.0	DNF
32	66.9	105.0	81.1
64	45.8	63.1	DNF
128	31.0	72.0	55.8
256	25.0	63.8	59.1
512	26.2	109.4	66.5

Table 5.8 Wall Clock Time in Seconds for 544 Lattice

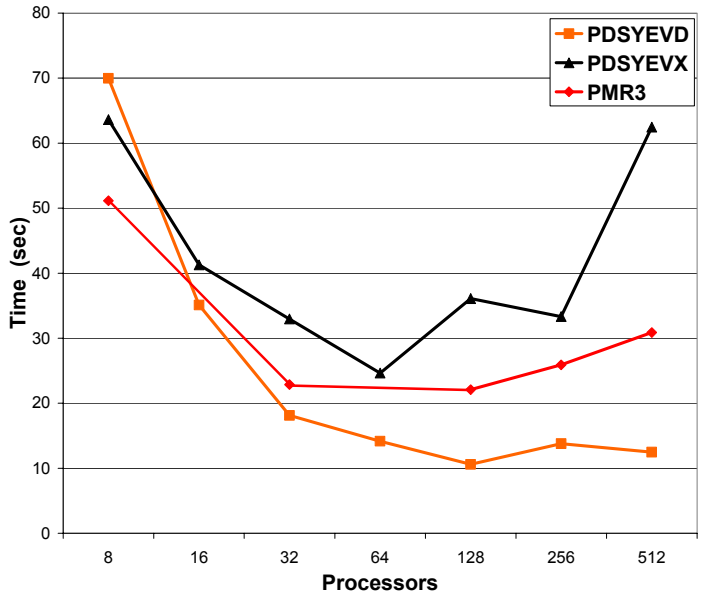

Figure 5.5 Plots of Wall Clock Time for 443 Lattice

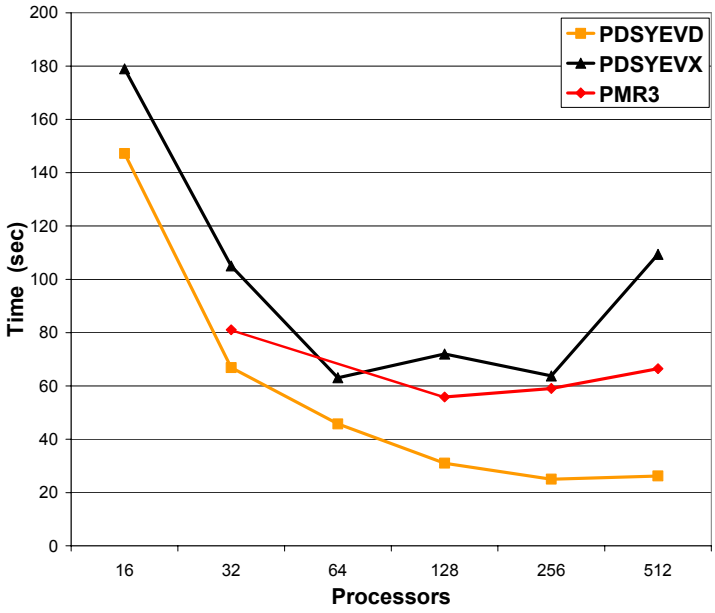

Figure 5.6 Plots of Wall Clock Time for 544 Lattice

	$\mathcal{R}$	$\mathcal{O}$
PDSYEVD	3.16E-16	1.02E-18
PDSYEVX	9.73E-11	7.68E-14
PMR3	1.75E-09	2.70E-10

Table 5.9 Maximum Residual and Orthogonality Results for Silicon Crystals

5.1.4 Hydrogen Molecules with Impurities.

# procs	PDSYEVD	PDSYEVX	PMR3
16	477.3	660.1	408.2
32	264.7	290.5	253.3
64	157.7	136.7	DNF
128	82.3	81.2	132.1
256	63.9	53.9	128.0
512	51.8	49.8	139.4

Table 5.10 Wall Clock Time in Seconds for Impure Hydrogen with 0.42% Sparsity

# procs	PDSYEVD	PDSYEVX	PMR3
16	476.3	609.2	506.7
32	266.1	282.9	255.5
64	152.4	129.7	DNF
128	81.5	78.9	127.8
256	62.1	52.5	119.7
512	49.6	44.8	124.3

Table 5.11 Wall Clock Time in Seconds for Impure Hydrogen with 14% Sparsity

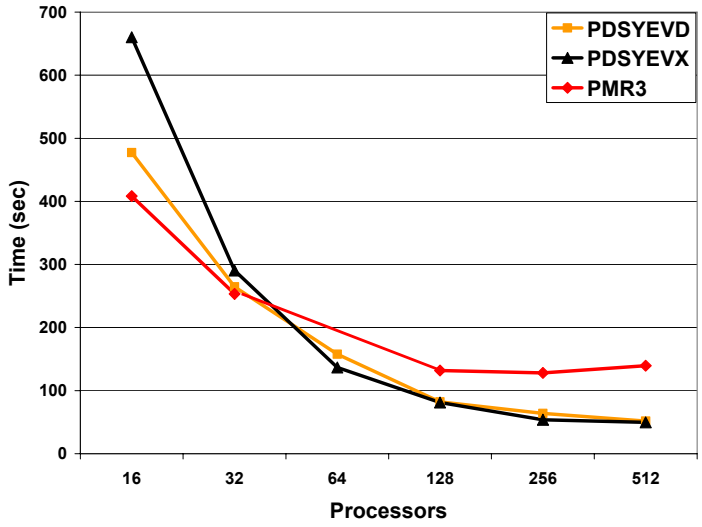

Figure 5.7 Plots of Wall Clock Time for Impure Hydrogen with 0.42% Sparsity

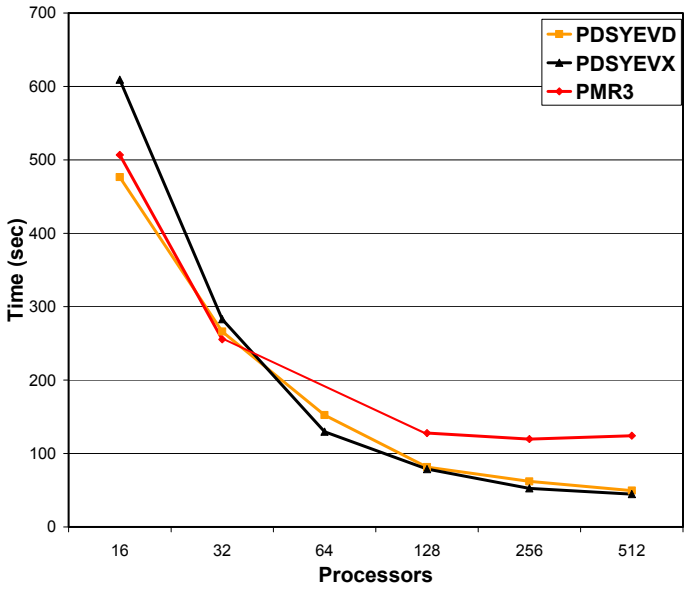

Figure 5.8 Plots of Wall Clock Time for Impure Hydrogen with 14% Sparsity

	$\mathcal{R}$	$\mathcal{O}$
PDSYEVD	1.79E-14	6.27E-19
PDSYEVX	2.98E-14	9.32E-19
PMR3	1.44E-10	9.43E-15

Table 5.12 Maximum Residual and Orthogonality Results for Impure Hydrogens

5.1.5 trans-Polyacetylene Family.

# procs	PDSYEVD	PDSYEVX	PMR3	PBDnC
16	402.6	529.2	434.5	33.0
32	229.5	298.4	DNF	14.1
64	119.3	142.2	DNF	15.7
128	66.2	135.2	283.4	6.4
256	50.8	108.8	108.9	4.5
512	45.1	163.6	128.0	4.6

Table 5.13 Wall Clock Time in Seconds for 12K-Atom trans-PA

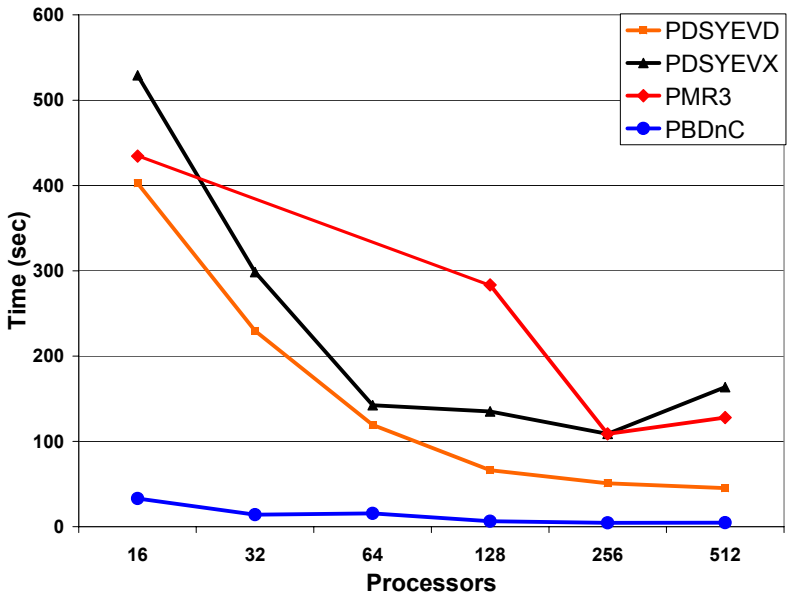

Figure 5.9 Plots of Wall Clock Time for 12K-Atom trans-PA

# procs	PDSYEVD	PDSYEVX	PMR3	PBDnC
32	499.2	656.4	497.4	18.8
64	322.1	328.0	DNF	13.0
128	152.7	251.6	221.4	8.6
256	96.8	174.0	202.6	6.4
512	76.4	251.9	211.4	7.2

Table 5.14 Wall Clock Time in Seconds for16K-Atom trans-PA

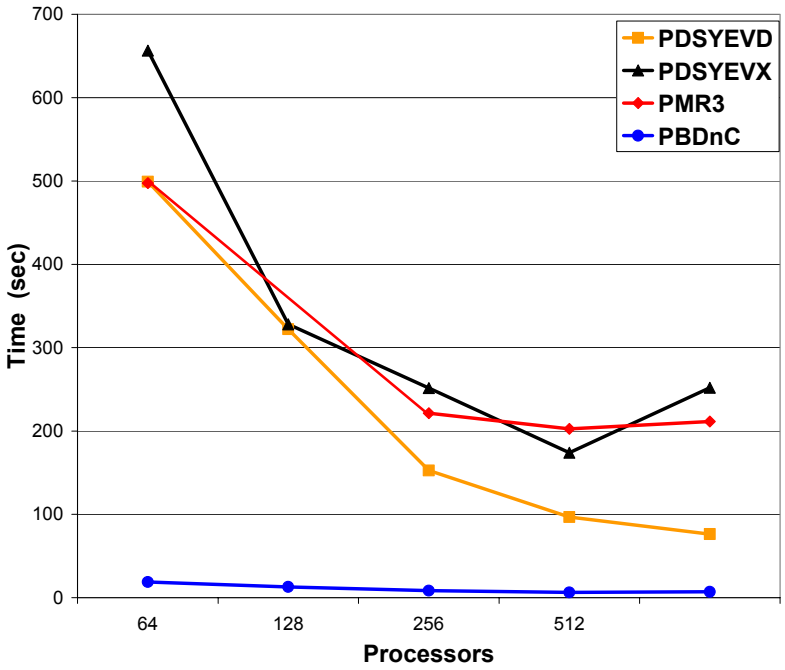

Figure 5.10 Plots of Wall Clock Time for 16K-Atom trans-PA

	$\mathcal{R}$	$\mathcal{O}$
PDSYEVD	5.43E-16	3.52E-19
PDSYEVX	6.07E-16	7.33E-18
PMR3	1.91E-15	2.64E-16
PBDnC	2.90E-06	1.59E-18

Table 5.15 Maximum Residual and Orthogonality Results for trans-Polyacetylenes (Note that PBDnC was given an accuracy request of 10^{-6}.)

5.1.6 Performance of the Classical 3-Steps in Eigensolvers. As mentioned earlier, many eigensolvers, including PDSYEVD, PDSYEVX and PMR3, involve 3 major steps: (1) reduction to symmetric tridiagonal form, (2) eigen-decomposition of the tridiagonal matrix, and (3) back-transformation of the eigenvectors. During our tests, the time required for each step was recorded for PDSYEVD and PMR3. We were unable to include PDSYEVX in this test since the source code was not available for the insertion of proper timing statements.

The below area charts present the average for each algorithm over all test matrices of the percentage of time spent in each step as a function of the number of processors. As is obvious, the reduction step continues to dominate eigensolvers. The below charts also illustrate the fact that PMR3's reduction and back-transformation steps did not perform as well as PDSYEVD's on most test cases, which is why PMR3 did not usually perform as well as PDSYEVD.

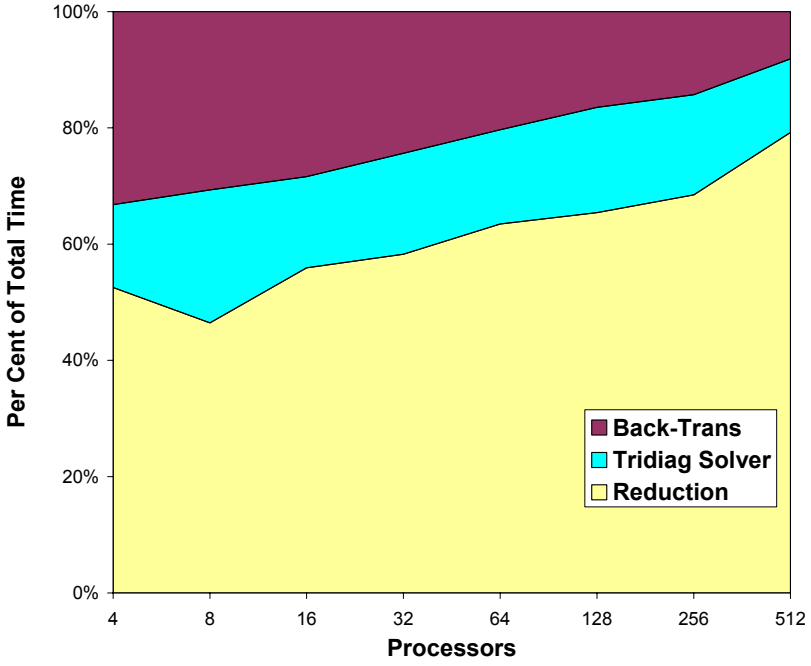

Figure 5.11 Mean Percentage of Solver Time Spent in the Three Steps of PDSYEVD as a Function of the Number of Processors

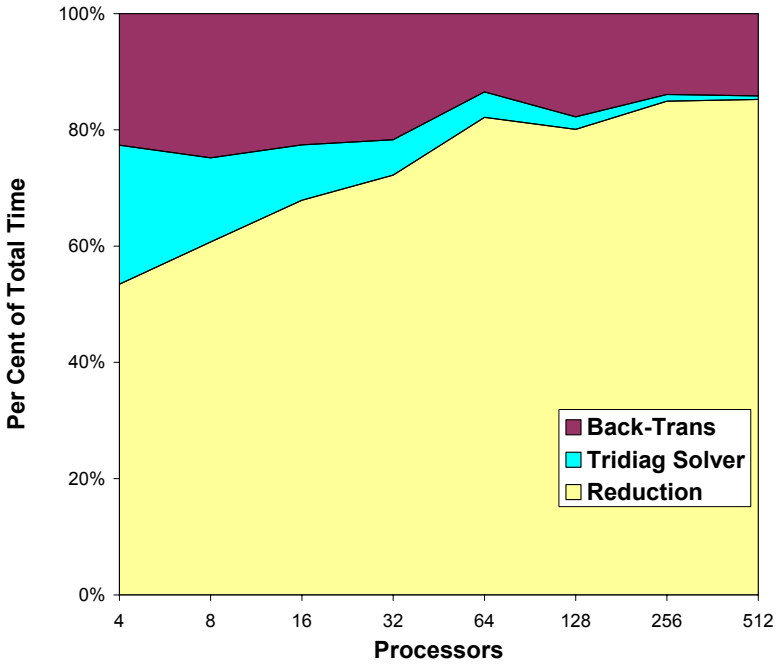

Figure 5.12 Mean Percentage of Solver Time Spent in the Three Steps of PMR3 as a Function of the Number of Processors

Area charts of percentage time in each step as a function of the matrix order are given below for the problem with the fastest time over all problems with the same matrix order regardless of number of processors and test family.

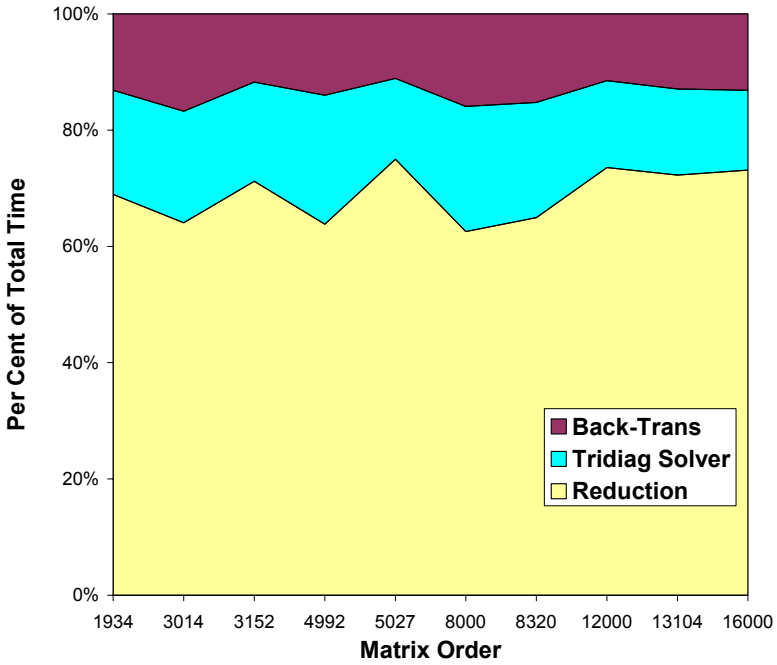

Figure 5.13 Percentage of Solver Time Spent in the Three Steps of PDSYEVD as a Function of the Matrix Size

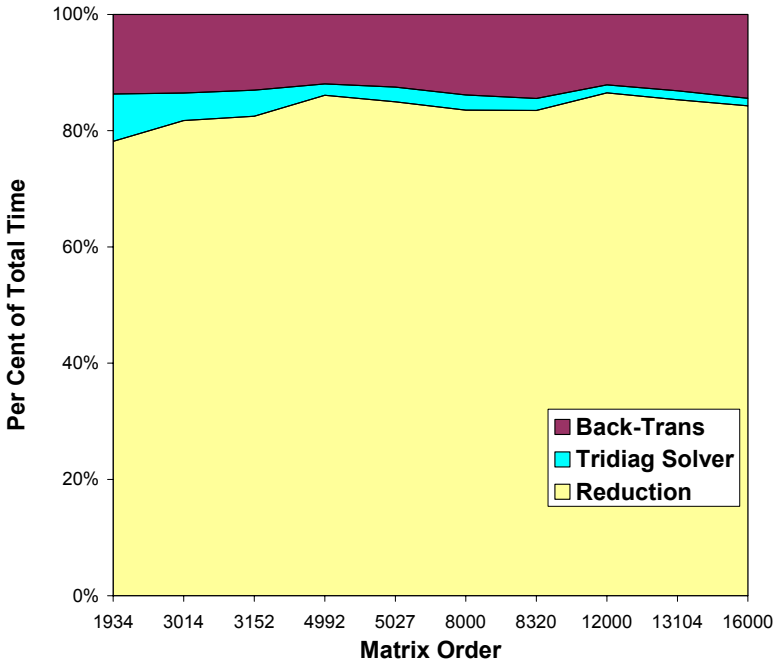

Figure 5.14 Percentage of Solver Time Spent in the Three Steps of PMR3 as a Function of the Matrix Size

5.2 PDSYEVD Tests on the SGI Ram and Cray Phoenix.

The only eigensolver available on these two supercomputers was PDSYEVD as provided by the vendors. SGI included the ScaLAPACK solver in their Scientific Computing Software Library, a collection of high-performance, optimized routines in support of scientific computing. Cray included the solver in their CrayLibSci software library of mathematical routines, but many of the routines available in CrayLibSci, including PDSYEVD, have not yet incorporated algorithmic optimizations (except for BLAS) for the Cray X1's unique architecture. PLAPACK was not successfully ported to either of these 64-bit computers within the limited time available for the tests.

The Fortran version of PDSYEVD was invoked on Ram through the Intel Fortran compiler `efc` version 7 with option `-O3` and on Phoenix with the Cray FORTRAN compiler `ftn` version 5.2 with options `-O3 stream2`.

Performance data for PDSYEVD on the SGI Ram are given in Table 5.16a & 5.16b below, and data for Cray Phoenix are given in Table 5.17 below.

# procs	$C_{322}H_{646}$	$C_{502}H_{1006}$	Polyala 125	Polyala 200	443 Si Cryst
4	4.82	18.04	20.34		
8	2.66	9.02	10.19	37.60	37.73
16	2.14	6.20	5.91	22.27	22.45
32	1.88	4.05	4.45	12.14	12.70
64	1.56	3.33	3.50	8.95	7.85
128	1.46	2.86	3.03	6.45	6.53

Table 5.16a Wall Clock Time for PDSYEVD on the SGI Ram

# procs	544 Si Cryst	Impure H 0.42%	Impure H 14%	12K trans-PA	16K trans-PA
16	93.34			231.31	
32	52.88	173.10	173.16	130.03	283.18
64	32.62	103.28	103.29	80.68	171.57
128	17.34	54.24	54.40	40.55	96.58

Table 5.16b Wall Clock Time for PDSYEVD on the SGI Ram

# procs	$C_{322}H_{646}$	$C_{502}H_{1006}$	443 Si Cryst
4	11.46	23.91	55.03
8	8.16	16.63	25.40
16	5.64	10.43	26.97
32	6.34	12.03	18.61
64	6.28	10.73	21.34
128	6.11	10.96	55.03

Table 5.17 Wall Clock Time for PDSYEVD on the Cray Phoenix

Although **comparisons at this time are very unfair** due to the differences in their basic architecture, ported software and levels of optimization, a few graphs are given below comparing the performance of PDSYEVD on the three supercomputers. These performance graphs are expected to change over the next several months as appropriate optimizations are made to mathematical software libraries, especially those for the Cray X1.

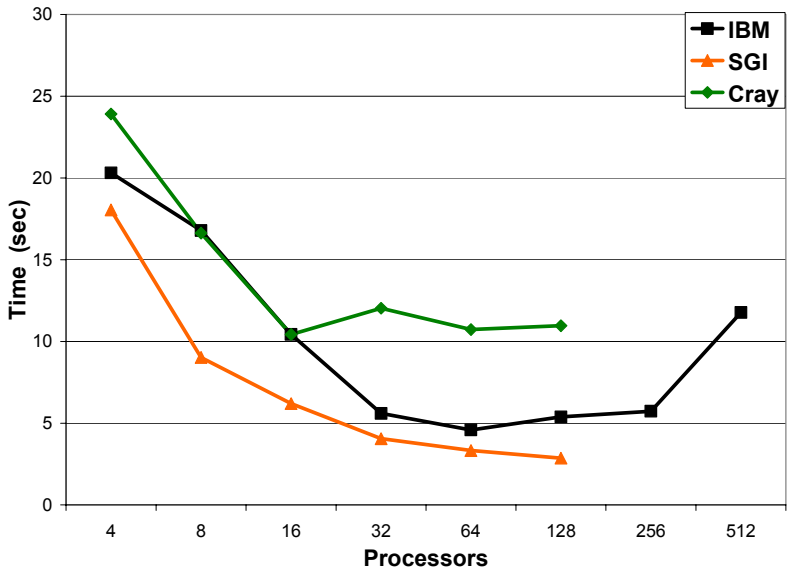

Figure 5.15 Wall Clock Time for PDSYEVD Solving $C_{502}H_{1006}$

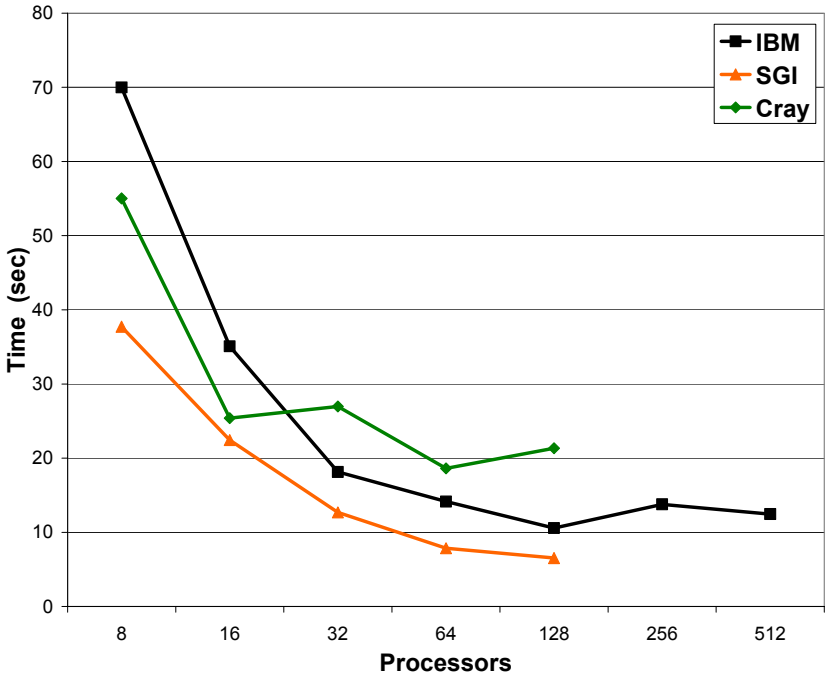

Figure 5.16 Wall Clock Time for PDSYEVD Solving 443 Silicon Crystal

Acknowledgments. The authors thank Dr. R. P. Muller of Sandia Laboratories for providing the alkane, polyalanine and silicon crystals families of test matrices, Dr. R. J. Hinde of the University of Tennessee for providing the hydrogen with impurities family,

and Dr. Guoping Zhang of Indiana State University for providing the polyacetylene family. We also thank A. S. Bland and Drs. Mark R. Fahey and Thomas H. Dunigan of the Oak Ridge National Laboratory for their invaluable assistance and detailed knowledge of the computing environment and systems at the ORNL Center for Computational Sciences and Peter Nagel and Paolo Bientinesi of the University of Texas for their assistance with the installation and porting of PLAPACK and PMR3.

References.

[1] http://nccs.gov/nccs-ccs.html.

[2] *Parallel Engineering and Scientific Subroutine Library for AIX, Version 3 Release 1, and Linux on pSeries, Version 3 Release 1, Guide and Reference*, Publication Number SA22-7906-01, IBM Corporation, Poughkeepsie, NY, 2003.

[3] Y. Bai, *Adaptive Parallel Eigensolver for Real Symmetric Matrices*, Ph.D. Thesis, Department of Computer Science, University of Tennessee at Knoxville, (in preparation).

[4] P. Bientinesi, I. S. Dhillon and R. A. v. d. Geijn, *A Parallel Eigensolver for Dense Symmetric Matrices based on Multiple Relatively Robust Representations*, Technical Report TR-03-26, Department of Computer Sciences, University of Texas at Austin, Austin, TX, 2003.

[5] C. H. Bischof, B. Lang and X. Sun, *A Framework for Symmetric Band Reduction*, Technical Report ANL/MCS-P586-0496, Argonne National Laboratory, Argonne, IL, 1996.

[6] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. W. Demmel, I. Dhillon, J. J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker and R. C. Whaley, *ScaLAPACK Users' Guide*, SIAM Press, Philadelphia, PA, 1997.

[7] J. J. M. Cuppen, *A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem*, Numer. Math., 36 (1981), pp. 177-195.

[8] M. J. S. Dewar and W. Thiel, *MNDO*, Journal of the American Chemical Society, 99 (1977), pp. 4899.

[9] I. S. Dhillon and B. N. Parlett, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra and Appl., 387 (2004), pp. 1-28.

[10] I. S. Dhillon, B. N. Parlett and C. Vömel, *LAPACK Working Note 162: The Design and Implementation of the MRRR Algorithm*, Technical Report UCB//CSD-04-1346, Computer Science Division, University of California at Berkeley, Berkeley, CA, 2004.

[11] R. M. Dreizler and E. K. U. Gross, *Density Functional Theory*, Springer-Verlag, Berlin/Heidelberg/New York/Tokyo, 1993.

[12] W. N. Gansterer, D. F. Kvasnicka and C. W. Ueberhuber, *Multi-Sweep Algorithms for the Symmetric Eigenproblem*, in VECPAR'98 -- Third International Conference for Vector and Parallel Processing, J. M. L. M. Palma, J. J. Dongarra and V. Hernandez, ed.^eds., Springer-Verlag, Berlin/Heidelberg/New York/Tokyo,1998, pp. 20--28.

[13] W. N. Gansterer, R. C. Ward, Y. Bai and R. M. Day, *A Framework for Approximating Eigenpairs in Electronic Structure Computations*, IEEE Computing in Science & Engineering, 6 (2004), pp. 50--59.
[14] W. N. Gansterer, R. C. Ward, R. P. Muller and W. A. G. III, *Computing Approximate Eigenpairs of Symmetric Block Tridiagonal Matrices*, SIAM J Sci. Comput., 25 (2003), pp. 65-85.
[15] R. A. v. d. Geijn, *Using `PLAPACK`: Parallel Linear Algebra Package*, The MIT Press, 1997.
[16] M. Gu and S. C. Eisenstat, *A Divide-and-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 172-191.
[17] R. J. Hinde, *Infrared-active vibron bands associated with substitutional impurities in solid parahydrogen*, J. Chem Phys, 119 (2003), pp. 6-9.
[18] R. P. Muller, J.-M. Langlois, M. N. Ringnalda, R. A. Friesner and W. A. G. III, *A Generalized Direct Inversion in the Iterative Subspace Approach for Generalized Valence Bond Wave Functions*, J. Chem. Phys., 100 (1994), pp. 1226.
[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM Press (reprinted version of Pentice Hall), Philadelphia, PA, 1997.
[20] B. N. Parlett and I. S. Dhillon, *Orthognal eigenvectors and relative gaps*, SIAM J Matrix Anal. Appl., 25 (2004), pp. 858-899.
[21] B. N. Parlett and I. S. Dhillon, *Relatively robust representations of symmetric tridiagonals*, Linear Algebra and Appl., 309 (2000), pp. 121-151.
[22] B. N. Parlett and C. Vömel, *Tight clusters of glued matrices and the shortcomings of computing orthogonal eigenvectors by multiple relatively robust representations*, University of California at Berkeley, Berkeley, CA, 2004.
[23] R. G. Parr and W. Yang, *Density Functional Theory of Atoms and Molecules*, Oxford University Press, New York, 1989.
[24] J. A. Pople and D. L. Beveridge, *Approximate Molecular Orbital Theory*, McGraw-Hill, New York,1st, 1970.
[25] C. C. J. Roothaan, *New Developments in Molecular Orbital Theory*, Reviews of Modern Physics, 23 (1951), pp. 69.
[26] W. P. Su, J. R. Schrieffer and A. J. Heeger, *Soliton Excitations in Polyacetylene*, Phys Rev B, 22 (1980), pp. 2099-2111.
[27] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry*, McGraw-Hill, Berkeley, CA, 1989.
[28] F. Tisseur and J. Dongarra, *A Parallel Divide and Conquer Algorithm for the Symmetric Eigenvalue Problem on Distributed Memory Architectures*, SIAM J. Sci. Comput., 20 (1999), pp. 2223-2236.