Title

A Survey of High-Quality Computational Libraries and their Impact in Science and Engineering Applications

Authors
Drummond, L.A.
Hernandez, V.
Marques, O.
et al.

Publication Date
2004-09-20

A Survey of High-Quality Computational Libraries and their Impact in Science and Engineering Applications

L. A. Drummond[1], V. Hernandez[2], O. Marques[1], J. E. Roman[2], and V. Vidal[2]

[1] Lawrence Berkeley National Laboratory
One Cyclotron Road, MS 50F-1650, Berkeley, California 94720, USA
ladrummond@lbl.gov, oamarques@lbl.gov
[2] Universidad Politécnica de Valencia
DSIC, Camino de Vera, s/n – 46022 Valencia, Spain
vhernand@dsic.upv.es, jroman@dsic.upv.es, vvidal@dsic.upv.es

Abstract. Recently, a number of important scientific and engineering problems have been successfully studied and solved by means of computational modeling and simulation. Many of these computational models and simulations benefited from the use of available software tools and libraries to achieve high performance and portability. In this article, we present a reference matrix of the performance of robust, reliable and widely used tools mapped to scientific and engineering applications that use them. We aim at regularly maintaining and disseminating this matrix to the computational science community. This matrix will contain information on state-of-the-art computational tools, their applications and their use.

1 Introduction

In recent years, a number of important scientific and engineering problems, ranging in scale from the atomic to the cosmic, have been successfully studied and solved by means of computational modeling and simulation. Many of these computational models and simulations benefited from the use of available software tools and libraries to achieve high performance. These tools have facilitated code portability across many computational platforms while guaranteeing the robustness and correctness of the algorithmic implementations.

Software development time includes the time required for individual tasks, such as coding and debugging, and the time needed for group coordination, such as reporting bugs, synchronizing source code, and preparing releases. The inefficiency of *ad hoc* approaches to coordinating group activities has already become a significant hidden cost in many large scientific computing projects. Software tools greatly reduce the software development and maintenance costs because other specialized software development teams have taken on the hardy task of guaranteeing portability, scalability and robustness. In addition, software tools offer a variety of solutions to a problem through interfaces. Nowadays, there are several initiatives and approaches for making tool interfaces interoperable. Interoperability aggregates benefits to the use of tools because interoperable tools facilitate code changes, that are due to an increase in the complexity of the problem being addressed or algorithmic formulation.

As a result of all the advancements in tool development and deployment, there is nowadays a growing need and interest among computational scientists for identifying software tools that have been successfully used in the solution of computational problems that are similar to theirs. In this article, we highlight some of the most successful and robust numerical tool development projects and example of their applications. For the centerpiece of this work, we propose the creation of a matrix of software tools, their most relevant applications in science and engineering, with comments on tool performance and functionality with reference to other available similar tools. Further, we also propose to regularly update this table with feedback and results from the international community of computational scientists, as well as with software developments from other teams. We based our first efforts on current work with the U.S. Department of Energy (DOE) Advanced CompuTational Software (ACTS) Collection project [1] and the multidisciplinary research carried out by members of the High Performance Networking and Computing group of the Valencia University of Technology in Spain. Our goal is to promote the reuse of robust software tools and at the same time provide guidance on their use. Additionally, this reference matrix will be instrumental in the dissemination of state of the art software technology, while providing feedback not only to users but also to tool developers, software research groups and even to computer vendors. In Section 2, we briefly introduce the functionality of the selected computational tools. Section 3 provides an introduction to the different applications used in the performance matrix presented in Section 4. Lastly, conclusions are presented in Section 5.

2 Software Tools

In the computational science community, the complexity of today's scientific and engineering applications has led to new computational challenges that stress the need for more flexible, portable, scalable, reliable and high performing code developments. In turn, these challenges along with changes in the computer technology suggest that we need to move to a higher-level programing style, where library tools, and perhaps to some extent the compilers, take care of the details of the implementations [2, 3]. Several examples of these state-of-the-art software developments will be studied in this article. First we take a look at some of the leading tool development projects. For the sake of space, we focus mostly on computational tools that provide implementation of numerical algorithms.

2.1 Tools from the ACTS Collection

The ACTS Collection comprises a set of computational tools developed primarily at DOE laboratories, sometimes in collaboration with universities and other funding agencies (NSF, DARPA), aimed at simplifying the solution of common and important computational problems. A number of important scientific problems have been successfully studied and solved by means of computer simulations built on top of ACTS Tools [4, 5, 6]. Here, we will focus on a subset of these tools. Namely, we will review some high performing applications that use ScaLAPACK, SuperLU, PETSc, SUNDIALS, and Global Arrays. We begin with brief descriptions of these tools. The reader is referred to the ACTS Information Center [7] for more details on these and other tools available in the Collection.

ScaLAPACK [8] is a library of high-performance linear algebra routines for distributed-memory message-passing multiple instruction multiple data computers and networks of workstations. It is complementary to the LAPACK library which provides analogous software for workstations, vector supercomputers, and shared-memory parallel computers. The ScaLAPACK library contains routines for solving systems of linear equations, least squares, eigenvalue problems and singular value problems. It also contains routines that handle many computations related to those, such as matrix factorizations or estimation of condition numbers.

SuperLU [9] is a general purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations. The library is written in C and is callable from either C or Fortran. SuperLU performs an LU decomposition with numerical pivoting and triangular system solves through forward and back substitution. The LU factorization routines can handle non-square matrices but the triangular solves are performed only for square matrices. Before the factorization step, the matrix columns may be preordered through library or user provided routines. Working precision iterative refinement subroutines are provided for improved backward stability. Routines are also provided to equilibrate the system, estimate the condition number, calculate the relative backward error, and estimate error bounds for the refined solutions.

PETSc (**P**ortable, **E**xtensible **T**oolkit for **S**cientific **c**omputation) [10] provides sets of tools for the parallel, as well as serial, numerical solution of PDEs that require solving large-scale, sparse linear and nonlinear systems of equations. PETSc includes nonlinear and linear equation solvers that employ a variety of Newton techniques and Krylov subspace methods. PETSc provides several parallel sparse matrix formats, including compressed row, block compressed row, and block diagonal storage. PETSc has a growing number of application users and development projects that re-use some of its functionality (see Section on SLEPc). The PETSc team has long worked on the development of interoperable interfaces with other ACTS tools, as well as other widely used tools.

SUNDIALS (**SU**ite of **N**onlinear and **DI**fferential/**AL**gebraic equation **S**olvers) [11] refers to a family of four closely related solvers: CVODE, for systems of ordinary differential equations; CVODES, which is variant of CVODE for sensitivity analysis; KINSOL, for systems of nonlinear algebraic equations; and IDA, for systems of differential-algebraic equations. These solvers have some code modules in common, primarily a module of vector kernels and generic linear system solvers, including one based on a scaled preconditioned GMRES method. All of the solvers are suitable for either serial or parallel environments.

Global Arrays (GA) [12] facilitates the writing of parallel programs that use large arrays distributed across different processing elements. GA offers a shared-memory view of distributed arrays without destroying their non-uniform memory access characteristics. Originally developed to support arrays as vectors and matrices (one or two dimensions), currently it supports up to seven dimensions in Fortran and even more in C. GA also comes with a visualizer that uses trace files to animate array access patterns. Its main purpose is to analyze the impact of distribution on performance. Additionally, GA offers some interfaces to numerical libraries like ScaLAPACK, TAO and the eigensolver library Peigs [13]. GA was originally developed for use in the NWChem computational chemistry package but has also been used in other chemistry packages such as GAMESS-UK and Columbus, as well as in other scientific fields.

2.2 Other Software Tools

The following tools have also made relevant contributions to computational sciences and are also included in the reference matrix.

SLEPc (**S**calable **L**ibrary for **E**igenvalue **P**roblem **c**omputations) [14], is a software library for computing a few eigenvalues and associated eigenvectors of a large sparse matrix or matrix pencil. It has been developed on top of PETSc and enforces the same programming paradigm. The emphasis of the software is on methods and techniques appropriate for problems in which the associated matrices are sparse, for example, those arising after the discretization of partial differential equations. SLEPc provides basic methods as well as more sophisticated algorithms and interfaces to some external software packages such as PARPACK [15]. It also provides built-in support for spectral transformations such as the shift-and-invert technique. SLEPc is a general library in the sense that it covers standard and generalized eigenvalue problems, both Hermitian and non-Hermitian, using either real or complex arithmetic.

SLICOT provides implementations of numerical algorithms for computations in systems and control theory [16]. The basic ideas behind SLICOT are usefulness of algorithms, robustness, numerical stability and accuracy, performance with respect to speed and memory requirements, portability and reusability, standardization and benchmarking. The current version of SLICOT consists of about 400 user-callable and computational routines in various domains of systems and control. Almost all of these routines have associated on-line documentation. About

200 routines have associated example programs, data and results. Reusability of the software is obtained by using standard Fortran 77, so as SLICOT can serve as the core for various existing and future computer aided control system design platforms and production quality software. SLICOT routines can be linked to MATLAB through a gateway compiler. P-SLICOT is based on ScaLAPACK and provides parallel versions of some SLICOT's routines.

3 Applications

Cosmic Microwave Background Data Analysis. A Microwave Anisotropy Dataset Computational Analysis Package (MADCAP) [17] has been developed to determine the two-point angular correlation functions (or power spectra) of the Cosmic Microwave Background (CMB) temperature and polarization modes, which are both a complete characterization of the CMB if the fluctuations are Gaussian, and are statistics which can be predicted by candidate cosmological models. MADCAP recasts the extraction of a CMB power spectrum from a map of the sky into a problem in dense linear algebra, and employs ScaLAPACK for its efficient parallel solution. The first implementation of MADCAP on a 512-processor Cray T3E used Cholesky decomposition of a symmetric positive definite matrix, which corresponds to the correlation matrix, followed by repeated triangular solves. It typically achieved 70-80% of theoretical peak performance. Porting to a 6086 processor IBM SP3 was trivial, but the initial performance was disappointing: 30% of peak. To reduce the communication overhead on this less tightly-coupled system, the Cholesky decomposition and triangular solves were replaced by explicit inversions and matrix-matrix multiplications. The cohesion of the ScaLAPACK library made this a trivial programming task, leading to a performance increase to around 50-60% of peak. MADCAP has recently been rewritten to exploit the parallelism inherent in the need to perform matrix-matrix multiplications, using ScaLAPACK's block cyclic distribution. The result has been near-perfect scaling, with 50 ± 2% of peak being achieved analyzing the same data on 1024, 2048, 3072 and 4096 processors.

Collisional Breakup in a Quantum System of Three Charged Particles. The complete solution of electron-impact ionization of hydrogen is the simplest nontrivial example of atomic collision theory. This problem remained open and challenged scientists for decades until it was successfully solved by the exterior complex scaling technique described in [6]. This technique employs a two-dimensional radial grid finite difference approximations for derivatives, and requires the solution of large, complex nonsymmetric linear systems. In this context, SuperLU was used to construct preconditioners for a conjugate gradient squared algorithm [18]. This approach allowed for the solution of order up to 736,000 on 64 processors of a Cray T3E, and up to 8.4 million on 64 processors of an IBM SP.

NWChem [19], developed at the Pacific Northwest National Laboratory, is a computational chemistry package that aims to be scalable both in its ability to treat large problems efficiently, and in its usage of available parallel computing resources. NWChem provides many methods to compute the properties of molecular and periodic systems using standard quantum mechanical descriptions of the electronic wavefunction or density. In addition, it has the capability to perform classical molecular dynamics and free energy simulations. These approaches may be combined to perform mixed quantum-mechanics and molecular-mechanics simulations. NWChem uses GA for the manipulation of data and computation in a distributed computing environment. The use of GA gives NWChem the necessary flexibility to scale the size of the problem and the number of processors used in the computation without any direct changes to the code. NWChem has been inside many computational chemistry applications and it has exhibited very good scaling (at least 70% efficiency [20]).

FUN3D is a code developed at NASA-Langley for investigating algorithmic issues related to the calculation of turbulent flows on unstructured grids. It is a tetrahedral vertex-centered unstructured grid code for compressible and incompressible Euler and Navier-Stokes equations. FUN3D is a valuable legacy code that has been promptly parallelized with the help of the PETSc library [21]. The developers of the FUN3D-PETSc code won the Gordon Bell Prize, 1999 in "Special Category" for achieving 0.23 Tflop/s on 3072 nodes of ASCI Red for unstructured mesh calculation.

Parallel Full-potential Linearized Augmented Plane-Wave. Electronic structure calculations have become increasingly important for describing and predicting properties of materials. Developments in theory, modeling and simulation, resulted in new opportunities for the study of quantum dots, nanotubes, nanoparticles, nano-interfaces, etc. The study of such systems, however, usually require hundreds of atoms and can only be realized by using high-end computers and efficient algorithmic implementations. P-FLAPW [4] is a parallel implementation of the FLAPW (full-potential linearized augmented plane-wave) method for electronic structure calculations. This is one of the most accurate and heavily used method in materials science for determining structural, electronic and magnetic properties of new materials. The method is based on a density functional approach to approximating and solving Schrödinger's equation. The main part of the calculation involves the construction of the Hamiltonian matrix and then its diagonalization using ScaLAPACK to determine the lowest 5-10% of the eigenvalues and eigenfunctions which corresponds to the wavefunctions and energies of the electrons in the system being studied. The construction of the Hamiltonian and its diagonalization take approximately equal times. The use of ScaLAPACK has allowed the simulation of systems containing up to 700 atoms [4] on machines such as the IBM SP and the CRAY T3E. This corresponds to matrix sizes of about 35,000, which run efficiently on up to 512 processors on the SP and T3E. ScaLAPACK is also used for eigenvalue calculations in other LAPW codes such as the widely used WEIN2K code.

NIMROD. The NIMROD Project [22] is developing a modern computer code suitable for the study of long-wavelength, low-frequency, nonlinear phenomena in fusion reactor plasmas. These phenomena involve large-scale changes in the shape and motion of the plasma and severely constrain the operation of fusion experiments. NIMROD uses a high-order finite element representation of the poloidal plane and a finite Fourier series representation of the toroidal direction. The time advance is semi-implicit. SuperLU was incorporated in NIMROD as an alternative for a PCG solver. As a result, for each time-step of the linearized problem, SuperLU is more than 100 times and 64 times faster than PCG on 1 and 9 processors, respectively, of an IBM SP. The improvement in solve time for nonlinear problems is 10-fold. In this case, SuperLU is used to precondition matrix-free CG solves, in addition to solving other linear systems, and the overall improvement in code performance is 5-fold. In summary, the performance improvements are dramatic and equivalent to 3-5 years progress in hardware.

M3D [23] is multi-institutional code that implements a Multilevel, 3D, parallel, plasma simulation code. It is suitable for performing linear and non-linear calculations of plasmas in toroidal topologies including tokamaks and stellarators. Recent tokamak studies conducted with M3D have addressed such topics as high-beta disruptions, two-fluid effects on internal modes, Toroidicity-induced Alfven Eigenmodes (TAE modes) using a hybrid model, pellet injection, and the formation of current holes in discharges with off-axis current drive. M3D uses PETSc for the manipulation of unstructured meshed problems and solution of linear systems of equations. Non-linear solvers from PETSc are currently being studied for integration into the M3D code. The M3D developers reported that PETSc not only reduced the development time of M3D but also allowed them to easily test with different already parallelized preconditioners present in PETSc and also in another ACTS tool called Hypre. The analysis of steady state electromagnetic waves in cavities has many applications such as the design of industrial microwave ovens, particle accelerators or semi-anechoic chambers for electromagnetic compatibility studies.

Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities. This code calculates cavity mode frequencies and field vectors by solving a generalized eigenvalue problem $Kx = \lambda Mx$ arising from a finite element discretization of Maxwell's equations. In this problem, K is the symmetric positive semi-definite stiffness matrix and M is the positive definite mass matrix. Both matrices are very sparse. The eigenvalues of interest are in the interior of the spectrum and usually tightly clustered. The original inexact shift-invert Lanczos algorithm [24] implemented in Omega3P showed poor convergence in applications where the matrices are ill-conditioned and the eigenvalues are not well separated, as in the case of long accelerator structures with many cells. A parallel exact shift-invert eigensolver was then devised, where SuperLU was incorporated with PARPACK [15], which implements the implicit restarted Arnoldi algorithm. This strategy allowed for the solution of a problem of order 7.5 million with 304 million nonzeros.

CAVIRES is another code for electromagnetic analysis of cavities. The underlying mathematical model is the eigenvalue problem solving the Maxwell equations in a bounded volume. The source-free wave equation is discretized with finite elements, which provide enough flexibility to handle strong variations in the solution as in the case of inhomogeneous cavities with several materials. Edge elements are used in order to avoid so-called spurious modes. All the element contributions yields a generalized algebraic eigenvalue problem in which the matrices can be complex and non-Hermitian. The dimension of the problem can be in the order of millions in research applications. In [25], the solution of the associated eigenvalue problem is carried out with SLEPc. The code computes the eigenvectors corresponding to a few of the smallest positive nonzero eigenvalues, avoiding the unwanted high-dimensional null-space. The efficiency obtained on 32 processors of an SGI Altix was 57 % for a problem of size 150,000.

Lambda Modes. Lambda modes can be used to study the steady state neutron flux distribution inside the reactor core (to detect sub-critical modes responsible for regional instabilities) as well as for transient analysis. The lambda modes equation is a differential eigenvalue problem derived from the neutron diffusion equation. The discretization of the problem through a collocation method leads to a generalized algebraic eigensystem whose dominant eigenmodes provide useful information about the neutron flux distribution and the safety of the reactor [26]. The matrices that are obtained have a block structure and the number of blocks depends on how many levels are considered when discretizing the energy. If upscattering effects are not considered the problem can be reduced to a standard eigenvalue problem in which the inverses of the diagonal blocks make it necessary to solve several linear systems in each step of the iterative eigensolver. The problem has been tackled with SLEPc for solving the eigenvalue problem combined with the linear system solvers provided by PETSc. Several benchmark reactors have been used for validation, with typical problem sizes ranging from 20,000 to 500,000. When computing only the dominant mode, A 55% efficiency was achieved with a problem size of 125,000, on 16 procs of an SGI cluster.

Time-dependent Neutron Diffusion Equation. Complex space and time phenomena related with the neutron flux oscillations have been observed in several boiling water reactors. Furthermore, rapid transients of reactor power caused by a reactivity insertion due to a postulated drop or abnormal withdrawal of control rods from the core have strong space dependent feedback associated with them. Therefore, a fast transient analysis code is needed for simulating these phenomena. The solution is based on discretizing the spatial domain, obtaining a large system of linear stiff ordinary differential equations (ODEs). FCVODE, which is part of the SUNDIALS suite (see Section 2.1), has been used to solve this problem [27]. The test case used is a simplified two-dimensional reactor known as Seed Blanket or TWIGL. For a finite difference discretization with 133*133 cells the dimension of the ODE system is 35,378 if two energy groups are considered. When running the code on a cluster of 10 PC's linked with Gigabit Ethernet a superlinear speedup is obtained because in this particular application increasing the number of processors enhances the convergence of the process.

Information Retrieval. Many information retrieval techniques are based on vector space models, thus involving algebraic computations [28]. A matrix is built containing the frequencies (or weights) at which terms appear in the documents. A preprocessing step reduces the size of this terms-by-documents matrix by running a stemming algorithm and also eliminating stop words, i.e. terms with low semantic relevance such as articles, conjunctions, etc. Then, a query can be evaluated by computing the cosine of its angle with respect to each document. In techniques such as the Latent Semantic Indexing, the large terms-by-documents matrix are usually approximated by low-rank matrices, obtained with a singular value decomposition, so that a query evaluation is sped up while maintaining a similar precision-recall ratio. A code based on SLEPc has been developed for computing a small percentage of the largest singular values and vectors. The validation of the code has been carried out with the collection DOE-TREC [29]. It contains DOE's abstracts, over 225,000 documents. When computing the largest 400 singular values, the code achieves 54% efficiency on a 16-processor cluster.

Model Reduction in Control Engineering. High order systems appear frequently in control problems, providing better precision of results required in a control or simulation process. However, a larger system involves a higher computational cost. For this reason, reducing the order of the model is often a necessity. Reduction of a linear control system model consists in obtaining a lower order model with similar behavior. The reduced model decreases the computational requirements in the control or simulation processes but obtaining such a model is an expensive operation. Parallel subroutines for model reduction methods based on balancing techniques (appropriate for stable systems) have been developed [30]. These routines are based on ScaLAPACK and are included in the parallel version of SLICOT. They scale up to a moderate number of processors and allow, for instance, the reduction of a model of order 2000 in 400 seconds with 5 PC's connected with Gigabit Ethernet.

Cardiac Simulation. Cardiac arrhythmias are among the leading causes of mortality in developed countries. The triggering mechanisms, development and termination of these arrhythmias are not clearly understood yet. Mathematical models of the propagation of cardiac electrical activity are increasingly being considered as a helpful tool for better understanding of the mechanisms involved in the development of ventricular arrhythmias. By combining the mathematical formulation of membrane ion kinetics and the structural complexity of the heart, it is possible to simulate the electrical propagation in cardiac tissues. To study the complex dynamics underlying the formation and development of ventricular tachycardias and ventricular fibrillation, a 3D model of the tissue is required. In addition, the analysis of the evolution of ischemic conditions require the simulation of tissues over a period of minutes. These requirements make this kind of simulations a computational challenge. A simulator based on PETSc has been developed [31], allowing the simulation of a 100x100x100 cells cardiac tissue for a period of 250 ms in about 2.5 hours on a 32-processors cluster.

Simulation of Automotive Silencers. One of the design requirements of automotive exhaust silencers is the acoustic attenuation performance at the usual frequencies. Analytical approaches such as the plane-wave propagation model can be considered for studying the acoustic behavior of silencers with rectangular and circular geometries, as well as with elliptical cross-section. Then, the designer can study the effects of different design parameters such as length of the chamber, eccentricity of the ellipse, or location of the inlet and outlet ports. However, numerical simulation provides many advantages with respect to the analytic study. The presence of sudden area changes between pipes of different cross-section is one of the most remarkable features in exhaust silencers. In the neighborhood of sudden area changes, wave propagation is three-dimensional even in the low-frequency range, due to the generation of evanescent higher order modes at the discontinuity. A numerical simulator based on the finite element method has been developed [32], which uses PETSc to solve a linear system of equations per excitation frequency. The code obtains the results for a problem with 880,000 unknowns in the 20-3000 Hz frequency range in about 3 hours using a 32-processors cluster.

4 A Matrix of Applications, Tools and their Performance

The software tools and computational applications introduced in the previous sections are only a small sample of the increasing number of scientific applications that now benefit from the use of advanced computational software tools. We propose the creation and update of a reference matrix to be available through the World Wide Web. In this section, we present this information in two tables. Table 1 lists applications and the software tools that they use while Table 2 lists applications, computational problems and performance. Our goal is to present these data in a single table with hyperlinks to more information on the applications, their scientific or engineering relevance, general performance information, list of software tools used and relevant information of the implementation.

Table 1. Summary of computational applications and the software tools that they use.

	ScaLAPACK	SuperLU	PETSc	SUNDIALS	GA	SLEPc	SLICOT
MADCAP	X						
3-Charged Particles		X					
NWChem					X		
FUN3D		X					
P-FLAPW	X						
NIMROD		X					
M3D			X				
Omega3P		X					
CAVIRES			X			X	
Lambda Modes			X			X	
Neutron Diffusion				X			
Info Retrieval						X	
Model Reduction	X						X
Cardiac Simulation			X				
Auto Silencers			X				

5 Conclusions

The realization of large complex, scalable interdisciplinary applications is ubiquitous in the agenda of many fields in computational sciences. Examples of these are all the emerging work on computational frameworks, problem solving environments, open source software development and cyber-collaborations. Many instances of such applications have led to remarkable discoveries in their domain fields. However, one of the great challenges we face today is how to best integrate and carry our achievements in science and engineering, as well as their corresponding software technology, to a solid base that will support and facilitate scientific discovery in the future. Furthermore, the ever evolving nature of the hardware and increase in the number of computational resources to support high end computing pose great penalties to software developments that are either monolithic, not scalable, contain hard-coded platform dependent tuning, or combinations of these. Quality general purpose tools can greatly benefit large and small software developments projects by providing core services that make applications less dependent on the hardware and programming trends. On high end computing, the use of parallel software tools offers more benefits for both legacy code and new code developments than compilers and language directives because the expertise and feedback from the use of tools can be embedded into the tools, thus providing further services to the computational science community. By contrast, the use of automatic parallelization through compilers and programming languages only gives, at best, parallel expression to algorithms known to the applications programmer.

Another essential aspect of this technology is user education. In building an infrastructure for today and future computational discoveries, we must first identify robust, scalable and portable software tools that have helped meet today's

Table 2. List of applications and relevant computational problems and performance.

Application	Computational Problem	Highlights
MADCAP	Matrix factorization and triangular solves.	Currently, 50% peak performance on an IBM SP, and nearly perfect scalability on 1024, 2048, 3072 and 4096 processors.
3-Charged Particles	Solution of large, complex nonsymmetric linear systems.	Solves systems of equations of order 8.4 million on 64 processors in 1 hour of wall clock time; 30 Gflops.
NWChem	Distribute large data arrays, collective operations.	Very good scaling for large problems.
FUN3D	Parallelization of legacy code, unstructured grids, compressible and incompressible Euler and Navier-Stokes equations.	Gordon Bell price, 0.23 Tflops on 3072 processors of ASCI Red.
P-FLAPW	Eigenvalue problems.	Study of systems up to 700 atoms corresponding to matrix sizes of about 35,000. Enables new scientific studies such impurities and disordered systems.
NIMROD	Quad and triangular high order finite elements, semi-implicit time integration, conjugate gradient based sparse matrix solvers.	Use of SuperLU in precondition matrix free CG results in code improvement of 5 fold, equivalent to 3-5 years progress in computing hardware.
M3D	Triangular linear finite elements, linear system solvers.	Currently, scales to 256 IBM SP processors. Many intrinsically 3D phenomena in laboratory plasmas have been modeled and understood in isolation.
Omega3P	Generalized eigenvalue problem.	Solution of a problem of order 7.5 million with 304 million nonzeros. Finding 10 eigenvalues required about 2.5 hours on 24 processors of an IBM SP.
CAVIRES	General eigenvalue problems, complex and nonsymmetric sparse matrices.	57% efficiency with 32 processors for a problem of size 150,000. Scalable to millions of unknowns.
Lambda Modes	Eigenvalue problems, implicit matrix.	55% efficiency on 16 processors for a problem of size 125,000.
Neutron Diffusion	System of linear stiff ODEs, finite differences, implicit time integration.	Very good scaling for large problems. Superlinear speedup. Tested with problem size 35,378.
Information Retrieval	Singular Value Decomposition of sparse rectangular matrices.	54% efficiency, solving problems of size 225,000.
Model Reduction	Dense matrices, solves Sylvester and Lyapunov equations.	48% efficiency solving problems of size 2,000.
Cardiac Simulation	Finite differences, sparse symmetric linear systems.	Very good scaling, tested with 100x100x100 cells tissues with 32 processors.
Automotive Silencers	3D finite elements, multiple sparse linear systems.	Good efficiency with 32 processors. Scalable to millions of unknowns.

computational challenges. Then, we need to look for the missing functionality to tackle emerging problems. These efforts demand the creation of references to documentation on the tools and their applications.
The reference matrix presented in this article, and its more comprehensive version available on-line, is one of our mechanisms for disseminating information on state-of-the-art tool development projects and their functionality. The applications are used as a reference to new users on the areas where these tools have proved beneficial, and sometimes essential. The computational problems are listed to guide potential users in the tool selection process. We are aiming at collecting information on the best general purpose software tools available which will help us bring todays discoveries to support tomorrow research and development through high end computing.

References

[1] Drummond, L.A., Marques, O.: The Advanced Computational Testing and Simulation Toolkit (ACTS): *What can ACTS do for you?* Technical Report LBNL-50414, Lawrence Berkeley National Laboratory (2002) Available at http://acts.nersc.gov/documents.

[2] Press, W.H., Teukolsky, S.A.: Numerical Recipes: Does this paradigm have a future? http://www.nr.com/CiP97.pdf (1997)

[3] Henderson, M.E., Anderson, C.R., Lyons, S.L., eds.: Object Oriented Methods for Interoperable Scientific and Engineering Computing. SIAM, Philadelphia (1999)

[4] Canning, A., Mannstadt, W., Freeman, A.J.: Parallelization of the FLAPW method. Computer Physics Communications **130** (2000) 233–243

[5] Mitnik, D.M., Griffin, D.C., Badnel, N.R.: Electron-impact excitation of Ne5+. J. Phys. B: At. Mol. Opt. Phys. **34** (2001) 4455–4473

[6] Rescigno, T., Baertschy, M., Isaacs, W., McCurdy, W.: Collisional breakup in a quantum system of three charged particles. Science **286** (1999) 2474–2479

[7] Marques, O., Drummond, L.A.: The DOE ACTS Information Center. http://-acts.nersc.gov (2001)

[8] Blackford, L.S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J.W., Dhillon, I., Dongarra, J.J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., Whaley, R.C.: ScaLAPACK User's Guide. SIAM, Philadelphia, PA (1997)

[9] Demmel, J.W., Gilbert, J.R., Li, X.: SuperLU Users' Guide. (2003)

[10] Balay, S., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Technical Report ANL-95/11 - Revision 2.2.0, Argonne National Laboratory (2004)

[11] Hindmarsh, A.C., Brown, P., Grant, K., Lee, S., Serban, R., Shumaker, D., Woodward, C.: SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation-Solvers. ACM TOMS (2004) (to appear)

[12] Nieplocha, J., Harrison, J., Littlefield, R.J.: Global Arrays: A portable 'shared-memory' programming model for distributed memory computers. In: Supercomputing'94. (1994) 340–349

[13] PNNL US: PeIGS, A Parallel Eigensolver Library. http://www.emsl.pnl.gov-/docs/global/peigs.html (2000)

[14] Hernández, V., Román, J.E., Vidal, V.: SLEPc: Scalable Library for Eigenvalue Problem Computations. Lecture Notes in Computer Science **2565** (2003) 377–391

[15] Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK Users' Guide, Solution of Large-Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods. SIAM, Philadelphia, PA (1998)

[16] Benner, P., Mehrmann, V., Sima, V., Huffel, S.V., Varga, A.: SLICOT: A subroutine library in systems and control theory. In Datta, B.N., ed.: Applied and Computational Control, Signal and Circuits. Volume 1., Birkhauser (1999)

[17] Borrill, J.: MADCAP: the microwave anisotropy dataset computational analysis package. Available at ArXiv: http://xxx.lanl.gov/ps/astro-ph/9911389 (1999)

[18] Baertschy, M., Rescigno, T., Isaacs, W., Li, X., McCurdy, W.: Electron-impact ionization of atomic hydrogen. Physical Review A **63** (2001)

[19] High Performance Computational Chemistry Group: NWChem, a computational chemistry package for parallel computers, version 4.5. Pacific Northwest National Laboratory. Richland, Washington 99352, USA (2003)

[20] High Performance Computational Chemistry Group: NWChem Benchmarks. http://www.emsl.pnl.gov/docs/nwchem/index.html (2003)

[21] Anderson, W.L., Gropp, W.D., Kaushik, D.K., Keyes, D.E., Smith, B.F.: Achieving high sustained performance in an unstructure mesh CFD application. In: Proceedings from SC'99. (1999) also ANL/MCS-P776-0899.

[22] Sovinec, C.R., Barnes, D.C., Gianakon, T.A., Glasser, A.H., Nebel, R.A., Kruger, S.E., Schnack, D.D., Plimpton, S.J., Tarditi, A., Chu, M.S., the NIMROD Team: Nonlinear magnetohydrodynamics with high-order finite elements. Journal of Computational Physics (2003) submitted.

[23] Park, W., Belova, E.V., Fu, G.Y., Tang, X., Strauss, H.R., Sugiyama, L.E.: Plasma simulation studies using multilevel physics models. Physics Plasmas **6** (1999)

[24] Sun, Y.: The Filter Algorithm for Solving large-Scale Eigenproblems from Accelerator Simulations. PhD thesis, Stanford University (2003)

[25] Roman, J.E.: Software Portable, Escalable y Extensible para la Resolución de Problemas de Valores Propios Dispersos de Gran Dimensión. PhD thesis, Universidad Politécnica de Valencia (2002)

[26] Hernandez, V., Roman, J.E., Vidal, V., Verdu, G., Ginestar, D.: Resolution of the neutron diffusion equation with SLEPc, the Scalable Library for Eigenvalue Problem Computations. In: Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew, Gatlinburg, TN, ANS (2003)

[27] García, V.M., Vidal, V., Verdu, G., Garayoa, J., Miro, R.: Parallel resolution of the two-group time-dependent neutron difussion equation with public domain ODE codes. 6th VECPAR Conference (2004)

[28] Berry, M.W., Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM, Philadelphia, PA (1999)

[29] NIST: TREC: Text retrieval conference. http://trec.nist.gov (2003)

[30] Guerrero, D., Hernandez, V., Roman, J.E.: Parallel SLICOT model reduction routines: The Cholesky factor of Grammians. In: 15th Triennal IFAC World Congress, Barcelona, Spain (2002)

[31] Alonso, J.M., Hernandez, V., Molto, G., Monserrat, M., Saiz, J.: Three-dimensional cardiac electrical activity simulation on cluster and grid platforms. 6th VECPAR Conference (2004)

[32] Alonso, J.M., Garcia, G., Hernandez, V., Denia, F.D., Fuenmayor, F.J.: A parallel computing approach for solving the Helmholtz equation in 3D domains: Application to the study of acoustic behaviour of silencers. For ECCOMAS (2004)