Andreas Stathopoulos

PRIMME: PReconditioned Iterative MultiMethod Eigensolver A robust, efficient and flexible Hermitian eigenvalue software

Department of Computer Science
Box 8795
College of William and Mary
Williamsburg
VA 23187-8795
andreas@cs.wm.edu
James McCombs

The numerical solution of large, sparse, hermitian eigenvalue problems continues to be one of the most computationally intensive tasks. Iterative eigensolvers are routinely called to solve for a large number of eigenvalues of matrices with dimension of excess of a million. As with linear systems of equations, large dimensions almost always necessitate the use of preconditioning. In addition, eigensolvers must also deal with the issue of storing and orthogonalizing an increasingly large set of eigenvectors.

In recent years, many preconditioned eigensolvers have emerged that perform well in many, but not all applications. Notable examples are the many variants of the Jacobi-Davidson method (JDQR, JDCG, JDQMR), the Generalized-Davidson+1 method, and the LOBPCG method, while variations of the traditional RQI and Inverse Iteration methods are still in use. Until recently, however, general purpose software that implements these methods both efficiently and robustly has been very scarce. As a result, application developers could not compare the various algorithms to find the most suitable for their application and computing environment. Often, they would develop in-house implementations of algorithms specifically tuned for their application. Such in-house approaches, however, cannot benefit from current advances in eigenvalue research. Such flexibility in choosing from a variety of methods is missing in todays software.

In some cases of existing software, robustness has taken a secondary role behind efficiency. Ideally, an eigensolver should find all the required eigenvalues, in the shortest possible time, and produce an orthonormal basis for their invariant space. It is well known, that iterative eigensolvers cannot guarantee that eigenvalues are not missed. However, certain algorithmic techniques can increase the confidence in the computed results, albeit at a higher computational cost. This increased confidence is sometimes needed in applications.

Although some of the above eigenvalue methods have been shown to provide

nearly optimal convergence for one eigenvalue in terms of matrix-vector operations, the question of efficiency is a much more complex issue. Actual execution times depend on the computing platform (hardware, compiler, libraries), on the number of eigenvalues required, on the quality of the preconditioner, and often on the problem solved. Block methods have become a prerequisite for good cache performance, yet maintaining good convergence with a large block size is not straightforward. In addition, there is a multitude of techniques for restarting, locking, stopping inner-outer iterations, that when properly implemented can significantly improve the efficiency of the eigensolvers.

In view of the above, our group has developed a robust, efficient multi-method software called PRIMME. PRIMME is based on a Davidson-type main iteration, but it implements various techniques such block, locking, various projections for preconditioning (e.g., Olsen's, Jacobi-Davidson, etc), CG-type restarting (giving rise to JD+1, and LOBPCG-type methods), and adaptive inner-outer iterations (allowing for JDQMR/JDCG or inexact Inverse Iteration type methods). The implementation of all these techniques on top of a common platform, allows PRIMME to transform to any of the above state-of-the-art eigenvalue methods, as well as to hybrids representing arbitrary combinations of techniques.

More than thirty features are controllable and tunable by the user. However, this flexibility is not at the expense of usability. A complete set of defaults is provided, and the user can simply select from a list of twelve predefined methods. Alternatively, a method selection can be further tuned by resetting particular features. Such a multi-layer transparency addresses the different levels of expertise of potential users, from end-users to eigenvalue experts.

Robustness decisions are prevalent in various implementation details of PRIMME, such as multiple levels of convergence checking, validation processes, out of order convergence of required eigenvalues, especially interior eigenvalues, and consideration of numerical error. In cases where a choice between robustness and efficiency has to be made, PRIMME favors robustness by default, but it still implements the efficient approach as a user-defined alternative.

Finally, the software runs both on parallel and sequential machines and can use the optimized BLAS and LAPACK libraries of the target machines. Additionally, the structure of PRIMME allows for a runtime capability of sensing both the computing environment and the problem solved, and adapting the choice of parameters accordingly. The potential of such a fully dynamic multi-method is a particular focus of our current research.

In this talk, we present an overview of PRIMME and its interface, and show some sample numerical results that demonstrate its robustness and efficiency. In particular, we show that JDQMR and GD+1, two of the supported methods, provide minimal execution time and number of iterations respectively to a number of applications. The hope is that the choice between the two can be

fully automated within PRIMME. Finally, we show a surprising result that for cases where a preconditioner is not available, block JDQMR can be substantially better than ARPACK, even for large numbers of eigenvalues.

The software is freely available under the lesser GPL license at:

`http://www.cs.wm.edu/~andreas/software/`