UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Scalable Library for Eigenvalue Problem Computations SLEPc

SLEPc Technical Report **STR-3**
Available at http://slepc.upv.es

Subspace Iteration in SLEPc

V. Hernández
J. E. Román
A. Tomás
V. Vidal

Last update: June, 2005 (SLEPc 2.3.0)
Previous updates: –

About SLEPc Technical Reports: These reports are part of the documentation of SLEPc, the *Scalable Library for Eigenvalue Problem Computations*. They are intended to complement the Users Guide by providing technical details that normal users typically do not need to know but may be of interest for more advanced users.

Contents

1 Introduction

Subspace Iteration (or Simultaneous Iteration) is a simple method for approximating eigenvalues and eigenvectors of matrices. It can be seen as a generalization of the Power Method (see SLEPc Technical Report STR-2, "Single Vector Iteration Methods in SLEPc"), in the sense that it iterates simultaneously on m initial vectors, instead of just one. Orthogonality of the vectors is explicitly enforced in order to avoid linear dependence as the iteration proceeds.

Subspace Iteration can be combined with a Rayleigh-Ritz projection procedure in order to improve convergence. In this case, the algorithm computes approximations of the eigenvalues and eigenvectors by projecting the problem onto the subspace spanned by the columns of $A^k X_0$, where X_0 is the initial set of vectors and k is the current iteration number. In spite of this enhancement, Subspace Iteration is generally inferior to Krylov methods such as Arnoldi, except in some cases in which it is still competitive, such as when the relative gap between the desired eigenvalues and the rest is large.

2 Description of the Method

The implementation currently available in SLEPc is based on the SRRIT subroutine [Bai and Stewart, 1997], which performs a Rayleigh-Ritz projection and handles converged eigenpairs by locking. This report provides just an overview of the algorithm, and the reader is referred to [Bai and Stewart, 1997] for additional details.

2.1 Basic Subspace Iteration

The method implemented in SLEPc is sketched in Algorithm 1. The algorithm computes a set of dominant Schur vectors, from which the eigenvectors are computed afterwards.

Even with projection, the subspace iteration method usually converges very slowly, thus requiring many iterations. In order to pursue efficiency, not all the operations have to be carried out in all the iterations. Since the orthogonalization is a costly operation, it is avoided unless it is strictly necessary. The multiplication $V \leftarrow AV$ is the operation that is computed most often (in the innermost loop), while the orthogonalization and projection steps are only performed occasionally.

Algorithm 1 (Subspace Iteration)

Input: Matrix A
Output: m dominant Schur vectors V and corresponding eigenvalues
  Generate a set of initial orthonormal vectors $V \in \mathbb{C}^{n\times m}$
  For $k = 1, 2, \ldots$
    Perform a Rayleigh-Ritz Projection step (algorithm 2)
    Check convergence of eigenvalues and lock the converged ones
    Orthogonalization loop
      Repeatedly compute $V \leftarrow AV$ and normalize columns of V
      Orthonormalize columns of V
    end
  end

Algorithm 2 (Rayleigh-Ritz Projection)

Input: Matrix A and an orthonormal set of vectors V
Output: Schur vectors V and quasi-triangular matrix T
  Compute the Rayleigh quotient $T = V^*AV$
  Reduce to Hessenberg form: $T \leftarrow U_1^*TU_1$
  Reduce to quasi-triangular form: $T \leftarrow U_2^*TU_2$
  Sort the 1×1 or 2×2 diagonal blocks: $T \leftarrow U_3^*TU_3$
  $U = U_1U_2U_3$
  $V \leftarrow VU$

In Algorithm 2, the quasi-triangular matrix T is sorted in descending order of magnitude, so that eigenvalues with largest modulus are always on top.

2.2 Available Implementations

The first implementation of Subspace Iteration was developed by Rutishauser in the late 1960's [Rutishauser, 1969, 1970]. His Algol procedure RITZIT is also described in [Parlett, 1980, ch. 14] and was intended only for symmetric matrices. A Fortran translation of this procedure can be found in the SVDPACK software for singular value computations [Berry, 1992a,b].

The Fortran subroutine LOPSI [Stewart and Jennings, 1981] implements the Subspace Iteration method for non-symmetric matrices. This version computes eigenvectors directly, which can lead to numerical difficulties in some cases.

SRRIT [Bai and Stewart, 1997] is another implementation that can deal with non-symmetric matrices, but it is more robust since it is based on the Schur decomposition. It combines Subspace Iteration with a Rayleigh-Ritz projection. Deflation is handled by locking converged eigenvectors.

Finally, a proprietary implementation of subspace iteration is subroutine EB12 in the HSL library, which incorporates Chebyshev acceleration [Duff and Scott, 1993].

3 The SLEPc Implementation

As stated above, the SLEPc implementation of the Subspace Iteration method mimics that of the SRRIT subroutine. The corresponding solver is `EPSSUBSPACE` (or `-eps_type subspace` from the command-line).

Currently, no specific options for this solver are available. All the parameters such as the tolerance for detecting a group of eigenvalues are fixed.

Supported problem types	All
Allowed portion of the spectrum	Largest $\|\lambda\|$
Support for complex numbers	Yes
Support for left eigenvectors	No

References

Bai, Z. and G. W. Stewart (1997). Algorithm 776: SRRIT: A FORTRAN Subroutine to Calculate the Dominant Invariant Subspace of a Nonsymmetric Matrix. *ACM Trans. Math. Softw.*, 23(4):494–513.

Berry, M. W. (1992a). Large Scale Sparse Singular Value Computations. *Int. J. Supercomp. Appl.*, 6:13–49.

Berry, M. W. (1992b). SVDPACK: A Fortran-77 Software Library for the Sparse Singular Value Decomposition. Technical Report UT-CS-92-159, Department of Computer Science, University of Tennessee.

Duff, I. S. and J. A. Scott (1993). Computing Selected Eigenvalues of Sparse Unsymmetric Matrices Using Subspace Iteration. *ACM Trans. Math. Softw.*, 19(2):137–159.

Parlett, B. N. (1980). *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, NJ. Reissued with revisions by SIAM, Philadelphia, 1998.

Rutishauser, H. (1969). Computational Aspects of F. L. Bauer's Simultaneous Iteration Method. *Numer. Math.*, 13:4–13.

Rutishauser, H. (1970). Simultaneous Iteration Method for Symmetric Matrices. *Numer. Math.*, 16:205–223.

Stewart, W. J. and A. Jennings (1981). Algorithm 570: LOPSI: A Simultaneous Iteration Method for Real Matrices [F2]. *ACM Trans. Math. Softw.*, 7(2):230–232.