

Is there life after the Lanczos method?

What is LOBPCG?

Andrew V. Knyazev

*Department of Mathematics and
Center for Computational Mathematics
University of Colorado at Denver*

SIAM ALA Meeting, July 17, 2003

Supported by the NSF DMS

Abstract

$$B^{-1}(A - \text{?})u = 0$$

Let us consider the problem of computing the smallest eigenvalue λ and the corresponding eigenvector x of a real symmetric matrix A . We restrict ourselves to the class of polynomial iterative methods, i. e. methods that generate approximations to x within the Krylov subspaces, and we use the Rayleigh quotient to measure the approximation quality. Under these assumptions, the Lanczos method gives the best possible approximation as it provides the global minimization of the Rayleigh quotient on Krylov subspaces. Is there a possible competitor?

Is there life after the Lanczos method?

Andrew Knyazev

Let us consider the local minimization of the Rayleigh quotient on the subspace spanned by the current approximation, the current residual and the previous approximation. This method was suggested in Knyazev, 1986. The preconditioned version is in Knyazev, 1991, the block version in Knyazev, 1998, and a “practically stable” implementation in Knyazev, 2001. It is called the locally optimal block preconditioned conjugate gradient (LOBPCG) method. The costs per iteration and the memory use in LOBPCG are competitive with those of the Lanczos method. According to preliminary numerical tests, the LOBPCG is capable of capturing the same linear (but not super-linear) convergence speed rate as that of the Lanczos method. It computes an approximation to the eigenvector directly on every iteration and has no numerical stability issues similar to those of the Lanczos method.

OUTLINE:

1. What is LOBPCG?
2. A Bit of Theory
3. Block iterations in LOBPCG
4. Comparison of Features with Lanczos and Block Lanczos
5. LOBPCG Software
6. Numerical Examples
7. LOBPCG Applications
8. Conclusions

1. What is LOBPCG?

The method combines *robustness and simplicity* of the steepest descent method with a *three-term recurrence* formula:

$$x^{(i+1)} = w^{(i)} + \tau^{(i)} x^{(i)} + \gamma^{(i)} x^{(i-1)},$$

$$w^{(i)} = Ax^{(i)} - \lambda^{(i)} x^{(i)}, \quad \lambda^{(i)} = \lambda(x^{(i)}) = (x^{(i)}, Ax^{(i)}) / (x^{(i)}, x^{(i)})$$

with properly chosen scalar iteration parameters $\tau^{(i)}$ and $\gamma^{(i)}$. The easiest and most efficient choice of parameters is based on an idea of *local optimality* Knyazev 1986, namely, select $\tau^{(i)}$ and $\gamma^{(i)}$ that minimize the Rayleigh quotient $\lambda(x^{(i+1)})$ by using the *Rayleigh–Ritz method*.

$$\boxed{\text{Three-term recurrence}} + \boxed{\text{Rayleigh–Ritz method}} = \boxed{\text{Locally Optimal Conjugate Gradient Method}}$$

2. A Bit of Theory (Knyazev, Neymeyr 2001)

If $\lambda^{(i)} \in [\lambda_k, \lambda_{k+1}[$ then it holds for the Rayleigh quotient $\lambda^{(i+1)}$ that either $\lambda^{(i+1)} < \lambda_k$ (unless $k = 1$), or $\lambda^{(i+1)} \in [\lambda_k, \lambda^{(i)}[$. In the latter case,

$$\frac{\lambda^{(i+1)} - \lambda_k}{\lambda_{k+1} - \lambda^{(i+1)}} \leq q^2 \frac{\lambda^{(i)} - \lambda_k}{\lambda_{k+1} - \lambda^{(i)}}, \quad (1)$$

where

$$q = 1 - \frac{2}{\kappa(A) + 1} \left(1 - \frac{\lambda_k}{\lambda_{k+1}} \right). \quad (2)$$

3. Block iterations in LOBPCG

A well known idea of using *Simultaneous, or Block Iterations* provides an important improvement over single-vector methods, and permits us to compute an $m > 1$ dimensional invariant subspace, rather than one eigenvector at a time. It can also serve as an acceleration technique over a single-vector methods on parallel computers, as convergence for extreme eigenvalues usually increases with the size of the block, and every step can be naturally implemented on wide varieties of *multiprocessor computers* as well as to take advantage of high level *BLAS* libraries.

The LOBPCG Method, is a straightforward generalization of the single-vector version enhanced with the Rayleigh–Ritz procedure on a larger trial subspace. Compared to other alternatives, e.g., trace minimization methods, LOBPCG demonstrates optimal convergence numerically.

4. Comparison of Features with Lanczos and Block Lanczos

Property	LOBPCG	Lanczos	B. Lanczos
Optimal convergence	Almost	Yes	Yes
Superlinear convergence	No	Yes	Yes
Missing eigenpairs	No	May be	No
Using block vectors BLAS	Yes	No	Yes
Algorithm simplicity	Yes	No	No
Stability/Extra costs	Low	High	Higher
Extra costs for eigenvectors	None	High	Higher
Inexact inverse/preconditioning	Yes	No	No
Recursive use	Yes	No	Partially

5. LOBPCG Software

LOBPCG is publicly available as:

MATLAB code posted at <http://www.mathworks.com/>

MPI C code for Hypre posted at <http://www.llnl.gov/CASC/hypre/>

Both are also at the Web page maintained by the speaker,

<http://math.cudenver.edu/~aknyazev/>

6. Numerical Examples

7-Point 3-D Laplacian with $1/h = 100, 126, 159, 200, 252$ on IBM ASCI BLUE (4 processors per node) at LLNL. The smallest eigenpair with the tolerance of 10^{-6} using Hypre Schwarz-PCG preconditioner is computed.

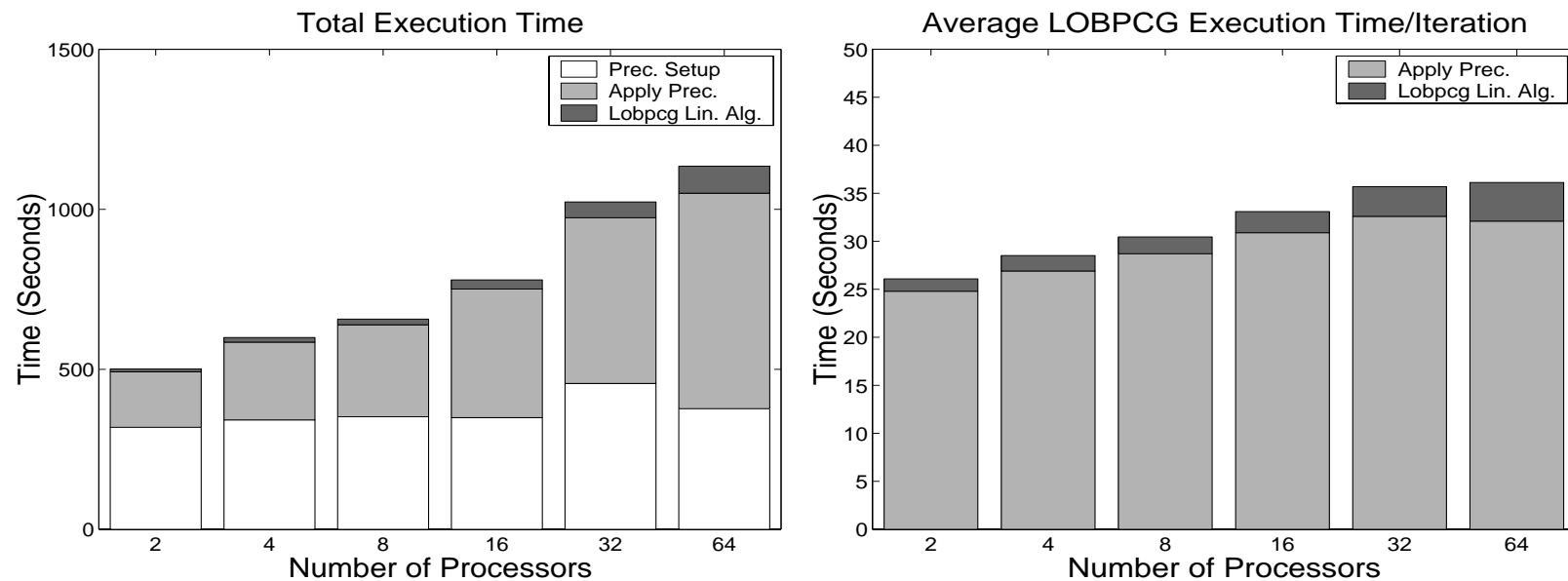


FIGURE 1: *Execution Time to Converge as the Problem Size Doubles*

7. LOBPCG Applications

1. Electronic structure calculations (joint with John Pask, LLNL, work in progress)
2. Spectral biclustering for microarrays DNA data (joint with Marina Kniazeva, work in progress)
3. Structural dynamics in ANSYS (work in progress)
4. Electromagnetic resonators (Rich Lehoucq and Peter Arbenz)
5. Image segmentation (private communication)

8. Conclusion

Conditions, favorable for LOBPCG vs. Lanczos:

1. Good preconditioner is available, especially if the preconditioning function can accept block vectors.
2. Computation of eigenvectors is necessary and approximations to eigenvectors of interest are known.
3. No superlinear convergence of Lanczos is expected.