

Is there life after the Lanczos method?

What is LOBPCG?

Andrew V. Knyazev

Department of Mathematics

Center for Computational Biology

Center for Computational Mathematics

University of Colorado at Denver

Theory and Numerics of Matrix Eigenvalue Problems, BIRS workshop
Banff, Alberta, Canada, November 24, 2003

This material is based upon work supported by the NSF

Center for Computational Mathematics, University of Colorado at Denver

Is there life after the Lanczos method?

Andrew Knyazev, CU-Denver

Original image

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

Spectral segmentation 153.1405 sec

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

Abstract

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, CU-Denver

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. Traditional eigenvalue solvers: Lanczos methods
2. Preconditioned eigenvalue solvers
3. What is **LOBPCG**?
4. A bit of theory
5. Block iterations in **LOBPCG**
6. Comparison of Features with Lanczos and Block Lanczos
7. What is **LOBPCG**?
8. Numerical Examples
9. **LOBPCG** Applications
10. Availability of **LOBPCG** Software

Traditional eigenvalue solvers for large scale problems

The classical Lanczos method is an evident candidate for an eigenvalue solver of choice to compute a few eigenpairs of matrix A . Several software implementations of the Lanczos method are publicly available and are well maintained, e.g., ARPACK.

A trivial and traditional possibility is to apply the Lanczos method directly to A , using the standard Krylov subspace, based on A . The only, but significant, difficulty in this approach is that the smallest eigenvalues may not be well relatively separated; therefore, the convergence of the Lanczos method may be extremely slow. E.g., for the finite difference approximation of the Laplacian operator, we might expect that the separation gets worse with the increase in the problem size; thus, we do not likely achieve the linear complexity in this way.

Is there life after the Lanczos method?

Andrew Knyazev, CU-Denver

Another well-known possibility is to apply the Lanczos method to the so-called shift-and-invert matrix, $(A - \alpha I)^{-1}$, where α is a properly chosen shift. These matrices are not, of course, formed explicitly. Instead, each time the Lanczos method requires a multiplication of a vector f by matrix $(A - \alpha I)^{-1}$, a linear solver subroutine is called to solve the corresponding linear systems. If these linear systems are solved sufficiently accurately, the convergence of the Lanczos method is typically much faster compared to that when the matrix A is used in the Lanczos method. The difficulty now is that accurate numerical solution of linear systems, needed on each iteration of the Lanczos method, can be costly.

Preconditioned eigenvalue solvers

In the present talk, we discuss an alternative class of methods, namely, the preconditioned eigensolvers. With an appropriate selection of a preconditioning technique, such methods would ideally converge as fast as the Lanczos method, applied to the shift-and-inverse, at the computational costs per iteration comparable with that of the Lanczos method, applied to the matrix A , thus, allowing to find a path between Scylla of the slow convergence and Charybdis of expensive linear solves.

Is there life after the Lanczos method?

Andrew Knyazev, CU-Denver

One of the main simplifying ideas of preconditioned eigensolvers is to separate the choice of the eigensolver from the choice of the preconditioner, thus, treating the preconditioner as a “black box.” E.g., if a multilevel preconditioner is used, it is called as an external subroutine outside of the eigensolver, in contrast to earlier known multilevel spectral graph bisection methods, where the levels are tightly embedded into the eigensolver. At the costs of losing some efficiency, it allows for a great flexibility in choosing a preconditioner.

Is there life after the Lanczos method?

Andrew Knyazev, CU-Denver

We concentrate in this talk on one of the most promising preconditioned eigensolvers, namely, on the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method, which has earlier been suggested and analyzed by the speaker. In LOBPCG for computing a single eigenpair of matrix A , the new iterate is determined by the Rayleigh–Ritz method on a three-dimensional subspace, which includes the previous iterate in addition to the current iterate and the preconditioned residual of the two-dimensional trial subspace of the steepest descent method. To compute the second eigenpair of A , the iterations are performed in the orthogonal complement to the known null space. If we need more than one eigenpair, we can compute them simultaneously.

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*.

Three-term recurrence + Rayleigh–Ritz method =
Locally Optimal Conjugate Gradient Method

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

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

if $A > 0$.

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.

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	Partially
Recursive use	Yes	No	Partially
Inexact inverse/preconditioning	Yes	No	No

What is LOBPCG?

The previous version with no preconditioning:

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

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

The new version with the B^{-1} preconditioner:

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

$$w^{(i)} = B^{-1}(Ax^{(i)} - \lambda^{(i)}x^{(i)}), \quad \lambda^{(i)} = \lambda(x^{(i)}) = (x^{(i)}, Ax^{(i)}) / (x^{(i)}, x^{(i)})$$

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.

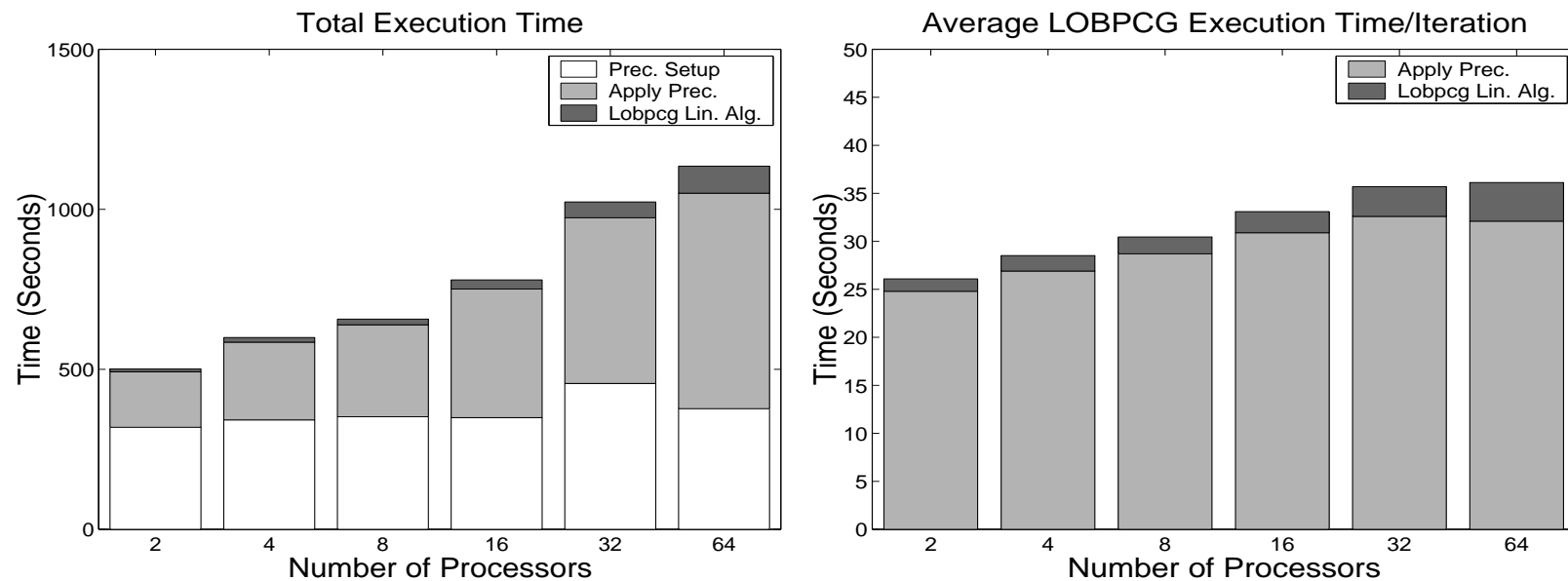
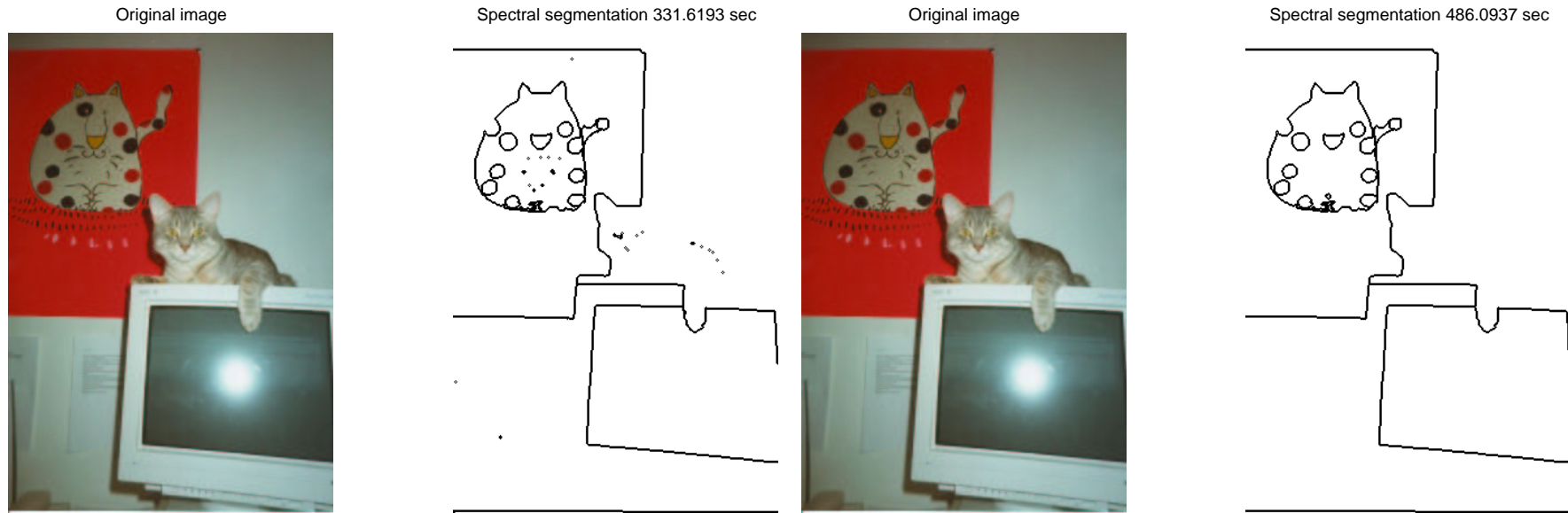


FIGURE 1: *Execution Time to Converge as the Problem Size Doubles*

Is there life after the Lanczos method?

Andrew Knyazev, CU-Denver



Segmentations of a natural 291×433 image using LOBPCG and EIGS, both applied to matrix $(A - \alpha I)^{-1}$ with $\alpha = 10^{-7}$ with no preconditioning. The MATLAB's built-in direct sparse linear solver is employed and the number of iterations is capped by three by both codes. Memory use topped slightly above 2GB. LOBPCG is faster with no quality loss.

Using Graph Analysis Toolbox by Leo Grady.

Center for Computational Mathematics, University of Colorado at Denver

LOBPCG Applications

1. Electronic structure calculations (joint with John Pask, LLNL, work in progress)
2. Structural dynamics in ANSYS (work in progress)
3. Electromagnetic resonators (Rich Lehoucq and Peter Arbenz)
4. Image segmentation

LOBPCG Software

The LOBPCG is publicly available in MATLAB, see <http://math.cudenver.edu/~aknyazev/software/CG/> and in C using MPI and HYPRE libraries for massively parallel computers, see <http://www.llnl.gov/CASC/hypre/>

Other people wrote their own LOBPCG implementations, in MATLAB and Fortran, not yet publicly available.

Conclusion

LOBPCG may be a valuable alternative to the classical Lanczos method.
Conditions, favorable for LOBPCG vs. ARPACK:

1. Computation of eigenvectors is necessary and approximations to eigenvectors of interest are known.
2. Eigenvalues of interest are multiple or clustered.
3. No superlinear convergence of Lanczos is expected.

Conditions, favorable for LOBPCG vs. ARPACK AND Block Lanczos:

- Good preconditioner is available, especially if the preconditioning function can accept block vectors.