

# Recent implementations, applications, and extensions of the Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)

Knyazev, Andrew

TR2017-078 June 18, 2017

## Abstract

Since introduction [A. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method, SISC (2001) doi:10.1137/S1064827500366124] and efficient parallel implementation [A. Knyazev et al., Block locally optimal preconditioned eigenvalue solvers (BLOPEX) in HYPRE and PETSc, SISC (2007) doi:10.1137/060661624], LOBPCG has been used in a wide range of applications in mechanics, material sciences, and data sciences. We review its recent implementations and applications, as well as extensions of the local optimality idea beyond standard eigenvalue problems.

*Householder Symposium on Numerical Linear Algebra*

© 2017 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# Recent implementations, applications, and extensions of the Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)

*Andrew Knyazev (Mitsubishi Electric Research Laboratories)*

## Abstract

Since introduction [A. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method, SISC (2001) doi:[10.1137/S1064827500366124](https://doi.org/10.1137/S1064827500366124)] and efficient parallel implementation [A. Knyazev et al., Block locally optimal preconditioned eigenvalue solvers (BLOPEX) in HYPRE and PETSc, SISC (2007) doi:[10.1137/060661624](https://doi.org/10.1137/060661624)], LOBPCG has been used in a wide range of applications in mechanics, material sciences, and data sciences. We review its recent implementations and applications, as well as extensions of the local optimality idea beyond standard eigenvalue problems.

## 1 Background

Kantorovich in 1948 has proposed calculating the smallest eigenvalue  $\lambda_1$  of a symmetric matrix  $A$  by steepest descent using a direction  $r = Ax - \lambda(x)$  of a scaled gradient of a Rayleigh quotient  $\lambda(x) = (x, Ax)/(x, x)$  in a scalar product  $(x, y) = x'y$ , where the step size is computed by minimizing the Rayleigh quotient in the span of the vectors  $x$  and  $w$ , i.e. in a locally optimal manner. Samokish in 1958 has proposed applying a preconditioner  $T$  to the vector  $r$  to generate the preconditioned direction  $w = Tr$  and derived asymptotic, as  $x$  approaches the eigenvector, convergence rate bounds. Block locally optimal multi-step steepest descent is described in Cullum and Willoughby in 1985.

Local minimization of the Rayleigh quotient on the subspace spanned by the current approximation, the current residual and the previous approximation, as well as its block version, appear in AK PhD thesis; see 1986. The preconditioned version is analyzed in 1991 and 1998, and a “practically stable” implementation [1]. LOBPCG general technology can also be viewed as a particular case of generalized block Davidson diagonalization methods with thick restart, or accelerated block gradient descent with plane-search. The main points on LOBPCG are as follows:

- The costs per iteration and the memory use in LOBPCG are competitive with those of the Lanczos method, computing a single extreme eigenpair.
- Linear convergence is theoretically guaranteed and practically observed, since local optimality implies that LOBPCG converges at least as fast as the gradient descent. In numerical tests, LOBPCG typically shows no super-linear convergence.
- LOBPCG blocking allows utilizing highly efficient matrix-matrix operations, e.g., BLAS 3.
- LOBPCG can directly take advantage of preconditioning, in contrast to the Lanczos method. LOBPCG allows variable and non-symmetric or positive definite preconditioning.
- LOBPCG allows warm starts and computes an approximation to the eigenvector on every iteration. It has no numerical stability issues similar to those of the Lanczos method.
- LOBPCG is reasonably easy to implement, so many implementations have appeared.
- Very large block sizes in LOBPCG become expensive to deal with due to orthogonalizations and the use of the Rayleigh-Ritz method on every iteration.

## 2 General purpose open source public software implementations

- MATLAB/Octave version by AK of the reference algorithm from 2001, publicly available from MATLAB/Octave repositories.
- BLOPEX, developed by AK and his team [5]. An abstract implementation of the reference algorithm from 2001. Originally at Google Code, currently hosted at bitbucket.org by Jose E. Roman.
- BLOPEX included into SLEPc/PETSc by Jose E. Roman.
- BLOPEX incorporated into hypre by AK and his team with hypre developers since 2007.
- LOBPCG with orthogonalizations, Anasazi/Trilinos by U. Hetmaniuk and R. Lehoucq since 2006.
- LOBPCG in Java at Google Code Archive by M. E. Argentati.
- LOBPCG in Python: SciPy, scikit-learn and megaman for manifold learning.
- LOBPCG GPU implementation in NVIDIA AmgX and spectral graph partitioning.
- LOBPCG heterogeneous CPU-GPU implementation by H. Anzt, S. Tomov, and J. Dongarra using blocked sparse matrix vector products in MAGMA (2015).

## 3 LOBPCG in applications

As a generic eigenvalue solver for large-scale symmetric eigenproblems, LOBPCG is being used in a wide range of applications. Our own small contribution is to data clustering and image segmentation (2003) [4] and (2015) [2], as well as for low-pass graph-based signal filtering (2015) [3]. Below we list some application-oriented open source software implementations of LOBPCG in various domains.

Material Sciences:

- ABINIT (including CUDA version) by AK with ABINIT developers, since 2008. Implements density functional theory, using a plane wave basis set and pseudopotentials.
- Octopus TDDFT, employs pseudopotentials and real-space numerical grids.
- Pescal, nonselfconsistent calculations of electron/hole states in a plane wave basis set.
- Gordon Bell Prize finalist at ACM/IEEE Conferences on Supercomputing in 2005, [7].
- Gordon Bell Prize finalist at ACM/IEEE Conferences on Supercomputing in 2006, [6].
- KSSOLV A MATLAB Toolbox for Solving the Kohn Sham Equations, Yang (2009).
- TTPY, spectra of molecules using tensor train decomposition, Rakhuba, Oseledets (2016).
- PlatypusQM, PLATform for dYnamic protein unified simulation, Takano et al. (2016).
- MFDn, Nuclear Configuration Interaction, Shao et al. (2016)

Data mining:

- sklearn: Machine Learning in Python, spectral clustering.
- Megaman: Scalable Manifold Learning in Python, McQueen et al. (2016).

Multi-physics:

- NGSolve (Joachim Schoberl), a general purpose adaptive finite element library with multigrid.

Electromagnetic calculations:

- NGSolve by Joachim Schoberl Maxwell solvers with Python interface.
- MFEM by Tzanio Kolev, highly scalable parallel high-order adaptive finite elements interfaced to hypre LOBPCG and multigrid preconditioning.
- Python based PYFEMax by R. Geus, P. Arbenz, L. Stingelin. Nedelec finite element discretisation of Maxwells equations.

## 4 Methods, motivated by LOBPCG

Suppose that the number  $n_b$  of vectors in the block must be small, e.g., due to memory limitations. Let us consider the extreme case  $n_b = 1$  of single-vector iterations only. At the same time, assume that we need to compute  $n_v > n_b$  eigenpairs. It is well-known that one can compute  $n_b$  eigenpairs at the time, lock them together with all the previously computed eigenvectors, and sequentially compute the next  $n_b$  eigenpairs in the orthogonal complement to all already locked approximate eigenvectors. E.g., if  $n_b = 1$ , the eigenpairs are computed one-by-one sequentially. In contrast, LOBPCG II, see [1] utilizes  $n_v$  parallel execution of single vector ( $n_b = 1$ ) 3-term locally optimal recurrences combined with the Rayleigh-Ritz procedure on the  $n_v$ -dimensional subspace spanned by the current approximations to all  $n_v$  eigenvectors. E. Vecharynski, C. Yang, and J. Pask (2015) extend the LOBPCG II approach to  $n_v/n_b$  parallel execution of  $n_b > 1$  block-vector 3-term recurrences with additional orthogonalizations, and demonstrate that Rayleigh-Ritz can be omitted on many iterations, in Quantum Espresso plane-wave DFT electronic structure software.

There are various other recent developments, related to LOBPCG:

- Indefinite variant of LOBPCG for definite matrix pencils, D. Kressner (2013) is based on an indefinite inner product, and generalizes LOBP4DCG method by Bai and Li (2012 and 2013) for solving product eigenvalue problems
- Folded spectrum to LOBPCG (FS-LOBPCG), V.S. Sunderam (2005) adds three more block vectors that store the matrix-vector products of the blocks  $X$ ,  $R$ , and  $P$ .
- Szyld and Xue (2016) extend LOBPCG to nonlinear Hermitian eigenproblems with variational characterizations.

Low-rank tensor related variants of LOBPCG:

- Hierarchical Tucker decomposition in low-rank LOBPCG, D. Kressner et al., 2011, 2016.
- Tensor-Train (TT) manifold-preconditioned LOBPCG for many-body Schrödinger equation, Oseledets et al., 2014, 2016

## 5 Conclusions

Still nobody can pronounce it, but LOBPCG:

- has now a life on its own, with 7K hits in google search,
- accepted as one of standard methods in DFT electronic structure calculations,
- appears in most major HPC open source libraries: PETSc, hypre and Anasazi/Trilinos, e.g., pre-packaged in OpenHPC cluster management software
- implemented for GPUs for ABINIT and in MAGMA and AmgX

More efforts are needed to decrease the use of orthogonalizations and Rayleigh-Ritz for large block sizes, while still preserving fast convergence, maintaining reasonable stability, and keeping the algorithm simple to implement.

## References

- [1] A. Knyazev. Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. *SIAM J. Scientific Computing*, 23(2):517–541, 2001. doi:[10.1561/0600000020](https://doi.org/10.1561/0600000020).
- [2] A. Knyazev. Method for Kernel Correlation-Based Spectral Data Processing. *US patent application* 20150363361 <http://www.freepatentsonline.com/y2015/0363361.html>.
- [3] A. Knyazev and A. Malyshev. Accelerated graph-based spectral polynomial filters. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, page 16. IEEE Xplore, 2015. doi:[10.1109/MLSP.2015.7324315](https://doi.org/10.1109/MLSP.2015.7324315).
- [4] A. V. Knyazev. Modern preconditioned eigensolvers for spectral image segmentation and graph bisection. In Boley, Dhillon, Ghosh, and Kogan, editors, *Proceedings of the workshop Clustering Large Data Sets; Third IEEE International Conference on Data Mining (ICDM 2003)*, pages 59–62, Melbourne, Florida, 2003. IEEE Computer Society. <http://math.ucdenver.edu/~aknyazev/research/conf/ICDM03.pdf>.
- [5] A. V. Knyazev, M. E. Argentati, I. Lashuk, and E. E. Ovtchinnikov. Block locally optimal preconditioned eigenvalue solvers (blopex) in hypre and petsc. *SIAM J. Scientific Computing*, 29(5):2224–2239, 2007. doi:[10.1137/060661624](https://doi.org/10.1137/060661624).
- [6] S. Yamada, T. Imamura, T. Kano, and M. Machida. High-performance computing for exact numerical approaches to quantum many-body problems on the earth simulator. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, SC '06*, New York, NY, USA, 2006. ACM. doi:[10.1145/1188455.1188504](https://doi.org/10.1145/1188455.1188504).
- [7] S. Yamada, T. Imamura, and M. Machida. 16.447 tflops and 159-billion-dimensional exact-diagonalization for trapped fermion-hubbard model on the earth simulator. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 44–44, Nov 2005. doi:[10.1109/SC.2005.1](https://doi.org/10.1109/SC.2005.1).