Numerical Linear Algebra for Computational Science and Information Engineering

Lecture 10
Locally Optimal Block Preconditioned Conjugate Gradient

Nicolas Venkovic
nicolas.venkovic@tum.de

Chair of Computational Mathematics
School of Computation, Information and Technology
Technical University of Munich

Winter 2025-26

Outline I

Methods based on optimization

Extremal generalized eigenvalue problem

- **Generalized eigenvalue problem**:

  Find non-trivial (x, λ) such that $Ax = \lambda Bx$

  - A is symmetric (or Hermitian)
  - B is symmetric positive definite (SPD)
  - Applications: structural dynamics, quantum mechanics, data analysis, ...
- **Challenges**:
  - Computing a few extremal eigenvalues of large sparse matrices
  - Achieving fast convergence while limiting memory usage
  - Handling ill-conditioned problems effectively
- **Characterization of the extremal generalized eigenpair**:
  We are looking for an extremal (min or max) generalized eigenpair (λ, x).
  Since B is SPD, it admits a Cholesky decomposition $B = LL^T$.
  Let the generalized Rayleigh quotient of (A, B) be given by

$$\rho(x) = \frac{x^T A x}{x^T B x} \quad \text{for} \quad x^T B x > 0$$

Characterization of the extremal generalized eigenpair

Making the substitution $y := L^T x$, the generalized Rayleigh quotient becomes the standard Rayleigh quotient of $L^{-1}AL^{-T}$:

$$\rho = \frac{x^T A x}{x^T B x} = \frac{(L^{-T}y)^T A (L^{-T}y)}{(L^{-T}y)^T B (L^{-T}y)} = \frac{y^T L^{-1} A L^{-T} y}{y^T y}$$

Since $L^{-1}AL^{-T}$ is symmetric, the Courant-Fischer theorem implies that the extremum of ρ is the extremal eigenvalue λ of $L^{-1}AL^{-T}$.

Then, since we have

$$\begin{aligned} L^{-1}AL^{-T}y &= \lambda y \\ L^{-1}AL^{-T}L^T x &= \lambda L^T x \\ Ax &= \lambda L L^T x \\ Ax &= \lambda B x \end{aligned}$$

the extremum value λ of $\rho(x)$ is the extremal eigenvalue of the generalized eigenvalue problem $Ax = \lambda Bx$, with eigenvector x.

- Finding an extremal generalized eigen-pair of (A, B) is equivalent to an optimization problem of the generalized Rayleigh quotient $\frac{x^T A x}{x^T B x}$.

Steepest descent iteration

- Iterative methods based on the optimization of the quotient $\rho(x) = \frac{x^T Ax}{x^T Bx}$ generate sequences $x_1, x_2, \dots$ of approximate eigenvectors based on a given recurrence formula with an initial iterate x_0.
- In particular, the steepest descent (SD) iteration is of the form

  $$x_{i+1} = x_i - \alpha_i \nabla\rho(x_i)$$

  where α_i is a step size chosen to minimize $\rho(x_{i+1})$ in the direction of the gradient $\nabla\rho(x_i)$ given by:

  $$\nabla\rho(x_i) = -\frac{2}{x_i^T Bx_i}(Ax_i - \rho(x_i)Bx_i) = -\frac{2}{x_i^T Bx_i} r_i \propto r_i$$

  where $r_i := Ax_i - \rho(x_i)Bx_i$ is the generalized eigen-residual.
  Thus, the iterate x_{i+1} is searched in $x_i + \mathrm{span}\{r_i\}$.
- To accelerate convergence, a preconditioner T can be applied to r_i, in which case the new iterate x_{i+1} is searched in $x_i + \mathrm{span}\{Tr_i\}$.
- As 1-dimensional affine subspace approximations, SD iterations exhibit slow convergence, especially for ill-conditioned problems.

Conjugate gradient iteration

- Conjugate gradient (CG) iterates $x_0, x_1, \dots$ were introduced by Hestenes and Stiefel (1952) to approximate $A^{-1}b$ where A is SPD (see Lect. 13).

  From the lens of optimization, those iterates attempt to minimize a quadratic function of the form $x^TAx - 2x^Tb$.
- Later, Fletcher and Reeves (1964) adapted CG iterations to optimize more general forms of functions, leading to the so-called nonlinear CG.
  - For vector-valued functions $f : \mathbb{R}^n \to \mathbb{R}$, the nonlinear CG iterate

$$x_{i+1} \in x_i + \mathsf{span}\{p_i\}$$

    is set to optimize f along the search direction p_i, where the search directions $p_0, p_1, \dots$ are updated such that

$$p_i \in -\nabla f(x_i) + \mathsf{span}\{p_{i-1}\}$$

    for $i = 1, 2, \dots$, with $p_{-1} := 0$.
  - Different search direction update formulae lead to different variants.

Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards, 49(6), 409-436.

R. Fletcher & C.M. Reeves (1964). Function minimization by conjugate gradients, Computer Journal, 7, 149–154.

Preconditioned conjugate gradient iteration

- When applied to general Rayleigh quotients of symmetric matrix pencils (A, B) with SPD B, the exact implementation of nonlinear CG iterates is such that

  $$x_{i+1} := \underset{x \in x_i + \text{span}\{p_i\}}{\arg\min \text{ (or } \max)} \;\; \rho(x)$$

  where the search direction

  $$p_i \in z_i + \text{span}\{p_{i-1}\} \;\text{ with }\; p_{-1} := 0\,,\;\; z_i := Tr_i \;\text{ and }\; r_i := Ax_i - \lambda_i B x_i$$

  can be specified after different formulae, leading to different variants of the algorithm, see Feng and Owen (1996).
- To accelerate convergence, one can introduce a preconditioner $z \mapsto Tr$, leading to preconditioned CG (PCG) iterations.

Feng, Y. T., & Owen, D. R. J. (1996). Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems. International Journal for Numerical Methods in Engineering, 39(13), 2209-2229.

Locally optimal preconditioned conjugate gradient iteration

- To motivate the **locally optimal preconditioned conjugate gradient** (**LOPCG**) iteration, we note that while PCG iterates $x_1, x_2, \dots$ can be expressed as approximations over a three-dimensional subspace, i.e.,

$$x_{i+1} \in \mathsf{span}\{x_i, z_i, p_{i-1}\},$$

  they are generally not optimal over that space.
- On the other hand, the LOPCG iterates introduced by Knyazev (1991, 2001) are actually optimal over that space. That is,

$$x_{i+1} = \arg \min_{x \in \mathsf{span}\{x_i, z_i, p_{i-1}\}} (\text{or } \max)\ \rho(x).$$

- Now, to deploy LOPCG, we need to be able to find the approximation x_{i+1} that optimizes the Rayleigh quotient $\rho(x)$ over a search space range(V_{i+1}) with a given full-rank basis matrix V_{i+1}, e.g., $V_{i+1} := [x_i, z_i, p_{i-1}]$.

Knyazev, A. V. (1991). A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace. In Numerical Treatment of Eigenvalue Problems Vol. 5/Numerische Behandlung von Eigenwertaufgaben Band 5: Workshop in Oberwolfach, February 25–March 3, 1990/Tagung in Oberwolfach, 25. Februar–3. März 1990 (pp. 143-154). Birkhäuser Basel.

Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM journal on scientific computing, 23(2), 517-541.

Rayleigh-Ritz projection of generalized eigenvalue problems

- Rayleigh-Ritz approximations are an instance of orthogonal projection (see Lecture 01) for (generalized) eigenvalue problems.
  They are essential to the deployment of LOPCG.

Rayleigh-Ritz procedure with respect to range(V)

For some matrices $A \in \mathbb{F}^{n\times n}$ and $B \in \mathbb{F}^{n\times n}$ along with a basis matrix $V \in \mathbb{F}^{n\times m}$ (typically with $m \ll n$), the Rayleigh-Ritz procedure is given by:

$$\begin{aligned}\mathsf{RR} : (A, B, V, k) \; &\mapsto ([\hat{x}_1, \dots, \hat{x}_k], \mathsf{diag}(\lambda_1, \dots, \lambda_k)) \\ &\text{s.t. } r_\ell := AV\hat{x}_\ell - \lambda_\ell BV\hat{x}_\ell \perp \mathsf{range}(V) \text{ for } \ell = 1, \dots, k\end{aligned}$$

for any $1 \leq k \leq m$ where each $(\lambda_\ell, \hat{x}_\ell) \in \mathbb{C} \times \mathbb{C}^m \setminus \{0\}$ is a distinct eigenpair of the projected problem given by:

$$V^H AV\hat{x}_\ell = \lambda_\ell V^H BV\hat{x}_\ell.$$

Then, $(\lambda_\ell, V\hat{x}_\ell)$ is called a Rayleigh-Ritz pair of the matrix pencil (A, B).

Remark: Rayleigh-Ritz procedures may be applied to non-symmetric matrices and their use is not limited to the context of LOBPCG.

LOPCG iteration

- The main LOPCG iterate $x_{i+1} = V_{i+1}\hat{x}_{i+1}$ is recast as follows

$$x_{i+1} = \begin{bmatrix} x_i & z_i & p_{i-1} \end{bmatrix} \begin{bmatrix} \hat{x}_{i+1|x_i} \\ \hat{x}_{i+1|z_i} \\ \hat{x}_{i+1|p_{i-1}} \end{bmatrix} = x_i\hat{x}_{i+1|x_i} + \underbrace{z_i\hat{x}_{i+1|z_i} + p_{i-1}\hat{x}_{i+1|p_{i-1}}}_{p_i}$$

  for $i = 1, 2, \ldots$, which leads to the following algorithm:

  LOPCG(A, B, x_0, T):

```
$r_0 := Ax_0 - Bx_0\lambda_0$
for $i = 0, 1, \ldots$ do
    $z_i := Tr_i$
    if $i = 0$ then $V_{i+1} := [x_i, z_i]$ else $V_{i+1} := [x_i, z_i, p_{i-1}]$
    $(\hat{x}_{i+1}, \lambda_{i+1}) \mapsfrom \mathsf{RR}(A, B, V_{i+1}, 1)$
    $x_{i+1} := V_{i+1}\hat{x}_{i+1}$
    $r_{i+1} := Ax_{i+1} - Bx_{i+1}\lambda_{i+1}$
    if $i = 0$ then $p_i := z_i\hat{x}_{i+1|z_i}$ else $p_i := z_i\hat{x}_{i+1|z_i} + p_{i-1}\hat{x}_{i+1|p_{i-1}}$
```

Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM journal on scientific computing, 23(2), 517-541.

LOBPCG iterations

Locally optimal block preconditioned conjugate gradient

- We now search for $X \in \mathbb{R}^{n\times k}$ and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$ for $k \leq n$ s.t.

  $$AX = BX\Lambda$$

  where $X^TBX = I_k$ and $\lambda_1, \ldots, \lambda_k$ are extremal generalized eigenvalues of the matrix pencil (A, B).

  X can then be defined as the minimizer of trace(X^TAX) subjected to the constraint $X^TBX = I_k$.
- Knyazev (2001) introduced **LOBPCG** as a **block version of LOPCG**, which simultaneously produces iterates for the approximation of multiple dominant eigen-pairs.
- Given an initial iterate X_0, LOBPCG generates a sequence of iterates $X_1, X_2, \ldots$ which, in their earliest form, are obtained by Rayleigh-Ritz projection in locally optimal subspaces range$([X_i, Z_i, P_{i-1}])$, where $Z_i := TR_i$ in which the columns of R_i are eigen-residuals, and P_{i-1} is block of search directions.

Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM journal on scientific computing, 23(2), 517-541.

LOBPCG iterations

- As a means to accelerate convergence, one can a number m of vector iterates, i.e., $X_i \in \mathbb{R}^{n \times m}$, while monitoring the convergence of only $k < m$ dominant eigenpairs:

LOBPCG(A, B, X_0, T, k):

```
$(\hat{X}_0, \Lambda_0) \mapsfrom \text{RR}(A, B, X_0, m)$ ; $X_0 := X_0\hat{X}_0$ ; $R_0 := AX_0 - BX_0\Lambda_0$
for $i = 0, 1, \dots$ do
    $Z_i := TR_i$
    if $i = 0$ then $V_{i+1} := [X_i, Z_i]$ else $V_{i+1} := [X_i, Z_i, P_{i-1}]$
    $(\hat{X}_{i+1}, \Lambda_{i+1}) \mapsfrom \text{RR}(A, B, V_{i+1}, m)$
    $X_{i+1} := V_{i+1}\hat{X}_{i+1}$; $R_{i+1} := AX_{i+1} - BX_{i+1}\Lambda_{i+1}$
    if $i = 0$ then $P_i := Z_i\hat{X}_{i+1|Z_i}$ else $P_i := Z_i\hat{X}_{i+1|Z_i} + P_{i-1}\hat{X}_{i+1|P_{i-1}}$
```

where use is made of the following notation:

$$\hat{X}_{i+1} = [\hat{X}_{i+1|X_i}^T, \hat{X}_{i+1|Z_i}^T, \hat{X}_{i+1|P_{i-1}}^T]^T$$

to refer to the different blocks of the reduced block of Raleigh-Ritz vectors.

Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM journal on scientific computing, 23(2), 517-541.

Implicit product updates

- From the fact that $P_1 = Z_0\hat{X}_{1|Z_0}$ we can compute the products AP_1 and BP_1 from AZ_0 and BZ_0 as follows:

$$AP_1 := AZ_0\hat{X}_{1|Z_0} \text{ and } BP_1 := BZ_0\hat{X}_{1|Z_0}.$$

- From the fact that $X_1 = X_0\hat{X}_{1|X_0} + Z_0\hat{X}_{1|Z_0}$, the products AX_1 and BX_1 can be formed from AX_0, AP_1, BX_0 and BP_1 as follows:

$$AX_1 := AX_0\hat{X}_{1|X_0} + AP_1 \text{ and } BX_1 := BX_0\hat{X}_{1|X_0} + BP_1.$$

- For $i > 0$, from the fact that $P_i = Z_i\hat{X}_{i+1|Z_i} + P_{i-1}\hat{X}_{i+1|P_{i-1}}$, the AP_i and BP_i can be calculated as follows from AZ_i, AP_{i-1}, BZ_i and BP_{i-1}:

$$\begin{aligned} AP_i &:= AZ_i\hat{X}_{i+1|Z_i} + AP_{i-1}\hat{X}_{i+1|P_{i-1}} \\ BP_i &:= BZ_i\hat{X}_{i+1|Z_i} + BP_{i-1}\hat{X}_{i+1|P_{i-1}}. \end{aligned}$$

- For $i > 0$, from the fact that $X_{i+1} = P_i + X_i\hat{X}_{i+1|X_i}$, AX_{i+1} and BX_{i+1} can be calculated as follows from AP_i, AX_i, BP_i and BX_{i+1}:

$$AX_{i+1} := AP_i + AX_i\hat{X}_{i+1|X_i} \text{ and } BX_{i+1} := BP_i + BX_i\hat{X}_{i+1|X_i}.$$

Sources of instability in LOBPCG iterations

The instability of LOBPCG was showcased and related to the ill-conditioning of $V_{i+1}^T BV_{i+1}$. This was explained as follows in the works of Hetmaniuk and Lehoucq (2006), Knyazev et al. (2007) and Duersch (2015):

- RR(A, B, V_{i+1}, m) needs to solve for $(\hat{X}, \Lambda)$ in the reduced equation

  $$V_{i+1}^T AV_{i+1}\hat{X} = V_{i+1}^T BV_{i+1}\hat{X}\Lambda.$$

  This is done by computing the Cholesky decomposition $LL^T = V_{i+1}^T BV_{i+1}$ before solving for $(\hat{Y}, \Lambda)$ in the standard symmetric eigenvalue problem

  $$L^{-1}V_{i+1}^T AV_{i+1}L^{-T}\hat{Y} = \hat{Y}\Lambda$$

  and letting $\hat{X} := L^{-T}\hat{Y}$.

  When $V_{i+1}^T BV_{i+1}$ is ill-conditioned, the Cholesky factorization may fail or lead to significant round-off error upon factor deployment.

Hetmaniuk, U., & Lehoucq, R. (2006). Basis selection in LOBPCG. Journal of Computational Physics, 218(1), 324-332.

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Duersch, J. A. (2015). High Efficiency Spectral Analysis and BLAS-3 Randomized QRCP with Low-Rank Approximations. University of California, Berkeley.

Sources of instability in LOBPCG iterations, cont'd

- Additionally, the following observations were made:
  - The projected matrix $V_{i+1}^T B V_{i+1}$ can become ill-conditioned irrespective of the conditioning of B.
  - When the number k of approximated eigenvectors is large, $V_{i+1}^T B V_{i+1}$ can become ill-conditioned before any eigenvector is accurately approximated.

Hetmaniuk, U., & Lehoucq, R. (2006). Basis selection in LOBPCG. Journal of Computational Physics, 218(1), 324-332.

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Duersch, J. A. (2015). High Efficiency Spectral Analysis and BLAS-3 Randomized QRCP with Low-Rank Approximations. University of California, Berkeley.

More stable variants of LOBPCG iterations

- "**Ortho_LOBPCG**": Hetmaniuk and Lehoucq (2006) perform a full B-orthonormalization of the basis matrix $V_{i+1} := [Z_i, X_i, P_{i-1}]$.
  We will see that the B-orthonormalization of V_{i+1} can be decomposed into two parts: an expensive procedure, and an "economic" one.
- **BLOPEX**: Knyazev et al. (2007) propose to B-orthonormalize the Z_i and P_i blocks independently. In exact arithmetic, this yields:

  $$X_i^T B X_i = I_m \text{ (by RR construction)}, \; Z_i^T B Z_i = I_m \text{ and } P_{i-1}^T B P_{i-1} = I_m$$

  but, in general, $Z_i^T B X_i \neq I_m$, $Z_i^T B P_{i-1} \neq I_m$ and $P_{i-1}^T B X_i \neq I_m$.
- "**Skip_ortho_LOPBCG**": Duersch et al. (2018) propose to skip the expensive part of the B-orthonormalization of V_{i+1} when it is not needed.
- **Mixed precision LOBPCG**: Kressner et al. (2023) introduce mixed-precision orthonormalization for standard problems, i.e., $B := I_n$.

Hetmaniuk, U., & Lehoucq, R. (2006). Basis selection in LOBPCG. Journal of Computational Physics, 218(1), 324-332.

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Duersch, J. A., Shao, M., Yang, C., & Gu, M. (2018). A robust and efficient implementation of LOBPCG. SIAM Journal on Scientific Computing, 40(5), C655-C676.

Kressner, D., Ma, Y., & Shao, M. (2023). A mixed precision LOBPCG algorithm. Numerical Algorithms, 94(4), 1653-1671.

BLOPEX iterations

BLOPEX iterations

- Any given basis matrix W can be B-orthonormalized by computing $W \mapsfrom W\text{chol}(W^TBW)^{-T}$ where $\text{chol}(W^TBW)$ denotes the Cholesky factor of W^TBW.
  B-orthonormalizing the Z_i and P_i iterates of LOBPCG independently before each Rayleigh-Ritz procedure yields the BLOPEX algorithm:

BLOPEX(A, B, X_0, T, k):

```
$B$-orthogonalize $X_0$: $L \mapsfrom \text{chol}(X_0^TBX_0)$ ; $X_0 := X_0L^{-T}$
for $i = 0, 1, \ldots$ do
    $R_i := AX_i - BX_i\Lambda_i$; $Z_i := TR_i$
    $B$-orthogonalize $Z_i$: $L \mapsfrom \text{chol}(Z_i^TBZ_i)$ ; $Z_i := Z_iL^{-T}$
    if $i = 0$ then $V_{i+1} := [X_i, Z_i]$
    else
        $B$-orthogonalize $P_{i-1}$: $L \mapsfrom \text{chol}(P_{i-1}^TBP_{i-1})$ ; $P_{i-1} := P_{i-1}L^{-T}$
        $V_{i+1} := [X_i, Z_i, P_{i-1}]$
    $(\hat{X}_{i+1}, \Lambda_{i+1}) \mapsfrom \text{RR}(A, B, V_{i+1}, m)$
    if $i = 0$ then $P_i := Z_i\hat{X}_{i+1|Z_i}$ else $P_i := Z_i\hat{X}_{i+1|Z_i} + P_{i-1}\hat{X}_{i+1|P_{i-1}}$
    $X_{i+1} := X_i\hat{X}_{i+1|X_i} + P_i$
```

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned

LOBPCG iterations with full B-orthonormalization

LOBPCG iterations with full B-orthonormalization

- LOBPCG is made more robust by making V_{i+1} B-orthonormal, i.e., by making sure that $V_1^T B V_1 = I_{2m}$ and $V_{i+1}^T B V_{i+1} = I_{3m}$ for $i = 1, 2, \ldots$
- To do so, Hetmaniuk and Lehoucq (2006) rely on a generic procedure that B-orthonormalizes a basis matrix $Z \in \mathbb{R}^{n\times p}$ against another basis matrix $W \in \mathbb{R}^{n\times q}$:

$$\text{Ortho}_B : (Z, W) \mapsto V \in \mathbb{R}^{n\times p} \text{ s.t. } \begin{cases} V^T B V = I_p \\ V^T B W = 0_{p\times q} \\ \text{range}(Z) \subseteq \text{range}(V) \end{cases} .$$

- Then, the fully orthogonalized variant of LOBPCG consists of performing

$$V_{i+1} \hookleftarrow \text{Ortho}_B([Z_i, X_i, P_{i-1}], [Z_i, X_i, P_{i-1}])$$

before each Rayleigh-Ritz projection with respect to range(V_{i+1}).

- Assuming X_i is B-orthonormal by construction, we can equivalently perform the following sequence of B-orthogonalizations:

$$Z_i \hookleftarrow \text{Ortho}_B(Z_i, [X_i, P_{i-1}])$$
$$P_i \hookleftarrow \text{Ortho}_B(P_i, X_{i+1})$$

Hetmaniuk, U., & Lehoucq, R. (2006). Basis selection in LOBPCG. Journal of Computational Physics, 218(1), 324-332.

Economic B-orthonormalization of search directions

- The cost of deploying the B-orthonormalization procedures is significant, so that it is particularly relevant if those can be simplified.
  As it turns out, the LOBPCG iterate given by

$$X_{i+1} = V_{i+1}\hat{X}_{i+1} = \begin{cases} X_i\hat{X}_{i+1|X_i} + Z_i\hat{X}_{i+1|Z_i} & \text{for } i=0 \\ X_i\hat{X}_{i+1|X_i} + Z_i\hat{X}_{i+1|Z_i} + P_{i-1}\hat{X}_{i+1|P_{i-1}} & \text{for } i=1,2,\ldots \end{cases}$$

  along with search direction given by

$$\mathsf{Ortho}_B(P_i, X_{i+1}) = \begin{cases} V_{i+1}\,\mathsf{Ortho}([0_{m\times m}, \hat{X}^T_{i+1|Z_i}]^T, \hat{X}_{i+1}) & \text{for } i=0 \\ V_{i+1}\,\mathsf{Ortho}([0_{m\times m}, \hat{X}^T_{i+1|Z_i}, \hat{X}^T_{i+1|P_{i-1}}]^T, \hat{X}_{i+1}) & \text{for } i=1,2,\ldots \end{cases} \quad (1)$$

  where $\mathsf{Ortho}(Z,W) := \mathsf{Ortho}_I(Z,W)$.
  Using the latter economic B-orthonormalizion of the search directions instead of the former helps mitigate this cost.
- Eq. (1) can be shown to hold in exact arithmetic by induction (homework problem).

Hetmaniuk, U., & Lehoucq, R. (2006). Basis selection in LOBPCG. Journal of Computational Physics, 218(1), 324-332.

Choice of B-orthonormalization procedure

- SVD-based B-orthonornomalization (SVQB, Stathopoulos & Wu (2002)) $\implies$ cache-efficient, highly stable, with low synchronization cost.

Ortho$_B(U, V)$:

```
do
    $U := U - V(V^T BU)$
    do
        $U :=$ SVQB$(B, U)$
    while $\frac{\|U^T BU - I_p\|}{\|BU\|\|U\|} < \tau_{ortho}$
while $\frac{\|V^T BU\|}{\|BV\|\|U\|} < \tau_{ortho}$
return $U$
```

SVQB(U, B):

```
$D := (\text{diag}(U^T BU))^{-1/2}$
Solve for eigen-pairs $Z, \Theta$ of $DU^T BUD$
such that $DU^T BUDZ = Z\Theta$
$\theta_{max} := \max_i |\Theta_{ii}|$
for $i = 1, \ldots, p$ do
    if $\Theta_{ii} < \tau\,\theta_{max}$ then $\Theta_{ii} := \tau\,\theta_{max}$
return $UDZ\Theta^{-1/2}$
```

where τ_{ortho} and τ are set to modest multiples of the machine precision.

- Householder QR $\implies$ highly stable, but difficult to implement for $B \neq I_n$.
- Gram-Schmidt procedures $\implies$ less efficient than SVQB.
- Cholesky QR procedures.

Stathopoulos, A., & Wu, K. (2002). A block orthogonalization procedure with constant synchronization requirements. SIAM Journal on Scientific Computing, 23(6), 2165-2182.

LOBPCG iterations with skipped B-orthonormalization

LOBPCG iterations with skipped B-orthonormalization

- B-orthonormalizing Z_i against $[X_i, P_{i-1}]$ is not always necessary for a stable implementation of LOBPCG.

  This fact is leveraged by Duersch et al. (2018) who propose to skip the most costly part of the B-orthonormalization when possible:

  1. Start with Ortho_LOBPCG iterations without the B-orthonormalization of Z_i against $[X_i, P_{i-1}]$:
     - The low-cost part of the B-orthonormalization in Ortho_LOBPCG iterations, i.e.,

       $$P_i \hookleftarrow V_{i+1}\text{Ortho}([0_{m\times m}, \hat{X}_{i+1|Z_i}^T, \hat{X}_{i+1|P_{i-1}}^T]^T, \hat{X}_{i+1})$$

       is equivalently carried out at each iteration.
     - Due to the fact that Z_i is not B-orthonormal with respect to $[X_i, P_{i-1}]$, the economic B-orthonormalization of P_i against X_{i+1} becomes

       $$P_i \hookleftarrow V_{i+1}\text{Ortho}_{V_{i+1}^T B V_{i+1}}([0_{m\times m}, \hat{X}_{i+1|Z_i}^T, \hat{X}_{i+1|P_{i-1}}^T]^T, \hat{X}_{i+1}).$$

Duersch, J. A., Shao, M., Yang, C., & Gu, M. (2018). A robust and efficient implementation of LOBPCG. SIAM Journal on Scientific Computing, 40(5), C655-C676.

LOBPCG iterations with skipped B-orthonormalization, cont'd$_1$

2. As long as V_{i+1} is not B-orthonormal, deploying the Rayleigh-Ritz procedure requires to factorize the projected matrix $V_{i+1}^T B V_{i+1}$ (see slide 12) whose ill-conditioning is addressed as follows:

   1. $D := \text{diag}(V_{i+1}^T B V_{i+1})^{-1/2}$
   2. Compute Cholesky decomposition $LL^T = DV_{i+1}^T B V_{i+1} D$
   3. Solve for reduced eigenpairs $(\Lambda, \hat{X})$ such that
   $$L^{-1} D V_{i+1}^T B V_{i+1} D L^{-T} \hat{X} = \hat{X}\Lambda$$
   4. Form Rayleigh-Ritz vectors as $X := V_{i+1} D L^{-T} \hat{X}$

   The conditioning of the Cholesky factor L is monitored to decide when to trigger the B-orthonormalization of Z_i against $[X_i, P_{i-1}]$.

   Since 3 triangular solves need be applied to form the Rayleigh-Ritz vectors, Duersch et al. (2018) check if $\text{cond}(L)^{-3}$ is greater than a modest multiple of machine precision to decide when to trigger the B-orthonormalization of Z_i against $[X_i, P_{i-1}]$.

Duersch, J. A., Shao, M., Yang, C., & Gu, M. (2018). A robust and efficient implementation of LOBPCG. SIAM Journal on Scientific Computing, 40(5), C655-C676.

LOBPCG iterations with skipped B-orthonormalization, cont'd$_2$

- We refer to these iterations as Skip_ortho_LOBPCG defined as follows:

  Skip_ortho_LOBPCG(A, B, X_0, T, k, τ_{skip}):

```
$(\hat{X}_0, \Lambda_0) \hookleftarrow$ RR$(A, B, X_0, m)$
$X_0 := X_0\hat{X}_0$; $R_0 := AX_0 - BX_0\Lambda_0$; $Z_0 := TR_0$
for $i = 0, 1, \dots$ do
    if skipOrtho
        if $i = 0$ then $Z_i \hookleftarrow$ Ortho$_B(Z_i, X_i)$ else $Z_i \hookleftarrow$ Ortho$_B(Z_i, [X_i, P_{i-1}])$
    if $i = 0$ then $V_{i+1} := [X_i, Z_i]$ else $V_{i+1} := [X_i, Z_i, P_{i-1}]$
    $(\hat{X}_{i+1}, \Lambda_{i+1}) \hookleftarrow$ RR$(A, B, V_{i+1}, m)$ ▷ $L$ is a by-product s.t. $LL^T = DV_{i+1}^T BV_{i+1}D$
    if skipOrtho
        if cond$(L)^{-3} < \tau_{skip}$ then skipOrtho := False ; restart $i$-th iteration
    $X_{i+1} := V_{i+1}\hat{X}_{i+1}$; $R_{i+1} := AX_{i+1} - BX_{i+1}\Lambda_{i+1}$; $Z_{i+1} := TR_{i+1}$
    if $i = 0$ then
        $\hat{Y}_{i+1} \hookleftarrow$ Ortho$_{V_{i+1}^T BV_{i+1}}([0_{m\times m}, \hat{X}_{i+1|Z_i}^T]^T, \hat{X}_{i+1})$
    else
        $\hat{Y}_{i+1} \hookleftarrow$ Ortho$_{V_{i+1}^T BV_{i+1}}([0_{m\times m}, \hat{X}_{i+1|Z_i}^T, \hat{X}_{i+1|P_{i-1}}^T]^T, \hat{X}_{i+1})$
    $P_i := V_{i+1}\hat{Y}_{i+1}$
```

Duersch, J. A., Shao, M., Yang, C., & Gu, M. (2018). A robust and efficient implementation of LOBPCG. SIAM Journal on Scientific Computing, 40(5), C655-C676.

Monitoring and handling convergence

Handling convergence

- Different eigenvectors may converge at different stages of the iteration. Maintaining converged eigenvectors to perform subsequent iterations
  1. requires unnecessary computational work,
  2. can lead to instabilities.

$\Longrightarrow$ A robust and efficient implementation of LOBPCG needs to detect, and properly handle converged eigenvectors.

- Two approaches possible, see Knyazev (2004) and Knyazev et al. (2007):
  - **Hard locking**: converged eigenvectors are set aside, kept unchanged, and B-orthogonalized against by the non-converged, still iterated eigenvectors.
    - As the number of hard locked vectors increases, the attainable accuracy of the iterated eigenvectors may decrease, possibly making convergence unachievable.
  - **Soft locking**: the residuals and search directions of converged eigenvectors are set aside, and kept unchanged, but the corresponding locked eigenvectors still participate to subsequent Rayleigh-Ritz procedures.
    - The locked eigenpairs keep getting more accurate over subsequent iterations, and the B-orthogonality is maintained implicitly through the Rayleigh-Ritz procedures.

Knyazev, A. V. (2004). Hard and soft locking in iterative methods for symmetric eigenvalue problems. In Presentation at the eighth copper mountain conference on iterative methods.

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Handling convergence, cont'd

- Soft locking is more computationally demanding than hard locking, but it enables more robust convergence behaviors when more accurate solutions are needed.
- In practice, convergence may be detected in unordered fashions, i.e., the inner eigenpairs converge before the smallest eigenpairs.
- We denote two distinct approaches to deal with this situation:
  - **out-of-order locking**: if locking is implemented out of order, one needs to re-order the stored iterates so as to seamlessly rely on standard BLAS libraries, which operate most efficiently on contiguous data.
  - **in-order locking**: more commonly in practice, locking is implemented in order, disregarding the fact that some inner eigenpairs may converge before the sought least dominant eigenpairs.
    - Maintaining such unlocked but converged eigenvectors in the iterations can lead to unstable behaviors of LOBPCG.

Knyazev, A. V. (2004). Hard and soft locking in iterative methods for symmetric eigenvalue problems. In Presentation at the eighth copper mountain conference on iterative methods.

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Landscape of existing software

Existing implementations of LOBPCG

Different implementations of LOBPCG have been developed over the years. In particular, we know of implementations and bindings in the following libraries:

- BLOPEX: C implementation with MPI support after Knyazev et al. (2007). On GitHub at `lobpcg/blopex`.
  - BLOPEX also available in/through Matlab, SLEPc and Hypre.
- MAGMA: C++ implementation based on BLOPEX for $B := I_n$ and $T := I_n$ with GPU support. On GitHub at

  `CEED/MAGMA/sparse/src/zlobpcg.cpp`
- SciPy: Python implementation based on BLOPEX. On GitHub at

  `scipy/sparse/linalg/eigen/lobpcg/lobpcg.py`
- IterativeSolvers.jl: Julia implementation based on BLOPEX with multithreaded BLAS support. On GitHub at

  `JuliaLinearAlgebra/IterativeSolvers.jl/src/lobpcg.jl`

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Existing implementations of LOBPCG, cont'd

- BLOPEX (Knyazev et al., 2007) has become the most widely used implementations of LOBPCG.

  BLOPEX became the standard with adoption through Hypre, SLEPc, ...
- At the moment, there seems to be no widely used implementations of
  - Ortho_LOBPCG (Hetmaniuk and Lehoucq, 2006)
  - Skip_ortho_LOBPCG (Duersch et al., 2018)
  - Mixed precision LOBPCG (Kressner et al., 2023)

Knyazev, A. V., Argentati, M. E., Lashuk, I., & Ovtchinnikov, E. E. (2007). Block locally optimal preconditioned eigenvalue Xolvers (BLOPEX) in Hypre and PETSc. SIAM Journal on Scientific Computing, 29(5), 2224-2239.

Hetmaniuk, U., & Lehoucq, R. (2006). Basis selection in LOBPCG. Journal of Computational Physics, 218(1), 324-332.

Duersch, J. A., Shao, M., Yang, C., & Gu, M. (2018). A robust and efficient implementation of LOBPCG. SIAM Journal on Scientific Computing, 40(5), C655-C676.

Kressner, D., Ma, Y., & Shao, M. (2023). A mixed precision LOBPCG algorithm. Numerical Algorithms, 94(4), 1653-1671.

Homework problems

Homework problem

Turn in **your own** solution to **Pb. 21**:

Pb. 21 Show that the gradient of the generalized Rayleigh quotient of a symmetric pencil (A, B) with symmetric A and SPD B given by

$$\rho(x) = \frac{x^T Ax}{x^T Bx} \text{ for } x \neq 0$$

is $\nabla \rho(x) = -\frac{2}{x^T Bx}(Ax - \rho(x)Bx)$.

Pb. 22 Let A be symmetric and B be SPD. Then show that the matrix $X = [x_1, \ldots, x_k]$ of the smallest general eigenvectors $x_1, \ldots, x_k$ of (A, B) subjected to $X^T BX = I_k$ is such that trace$(X^T AX)$ is minimized.

Pb. 23 Prove the equivalence stated in Eq. (1) of slide 17 between the full B-orthonormalization of P_i against X_{i+1} and the economic procedure used in Ortho_LOBPCG.

Practice session

Practice session

1. Implement LOBPCG.
2. Implement BLOPEX.
3. Implement Ortho_LOBPCG.
4. Implement Skip_ortho_LOBPCG.
5. Use the matrices `bcsstk12` and `bcsstm12` from the SuiteSparse collection for A and B, respectively. Then, using $m = 200$, benchmark each algorithm by monitoring the convergence of the 10 smallest generalized eigenvalues, first with $\tau = 10^{-5}$, then with $\tau = 10^{-6}$. Use a block Jacobi preconditioner with 10 diagonal blocks based of A. What do you observe?