# Iterative Eigen Solver and Large Rayleigh-Ritz

Haizhao Yang
Department of Mathematics, Duke University
Collaborators: Jianfeng Lu♯
♯ Department of Mathematics and Chemistry and Physics, Duke University

August 19, 2016

## Iterative eigensolver

- Linear and nonlinear;
- No solving equations;
- No Rayleigh-Ritz method with a large problem size;
- Or try to reduce the number of the procedures above.

# LOBPCG and OMM for standard eigven value problem

## A single eigen pair

LOBPCG: try to minimize the Rayleigh quotient

$$\lambda = \arg\min_x \mu(x) = \arg\min_x x^* H x,$$

with the constrain $x^* x = 1$, to find the smallest (or the largest) eigen pair.

## OMM

$$E = \min_{x \in \mathbb{C}^{n \times 1}} E_{\text{omm}}(x) = \min_{x \in \mathbb{C}^{n \times 1}} \text{tr}\big((2 - x^* x)(x^*(H - \eta I_n)x)\big),$$

where $\eta$ is a proper shift.

# LOBPCG

## Steepest descend (SD)

The gradient of the Rayleigh quotient is

$$r = Hx - \mu(x)x = Hx - (x^* Hx)x.$$

Let $T$ be the preconditioner, then the preconditioned direction is $w = Tr$.

$$x^{i+1} = x^i + t^i w^i,$$

where $t^i$ is the step size of the line search.

## Rayleigh-Ritz method

Three choices

- SD flavor: $x^{i+1} = \arg\min_{y \in span\{x^i, w^i\}} \mu(y)$;
- CG flavor: $x^{i+1} = \arg\min_{y \in span\{x^i, w^i, x^{i-1}\}} \mu(y)$;
- CG flavor: $x^{i+1} = \arg\min_{y \in span\{x^i, w^i\ p^i\}} \mu(y)$, where $p^i = x^{i-1} - x^i$;

Use Rayleigh-Ritz to solve the minimization problem.

# OMM

## Conjugate gradient (CG)

The gradient of the OMM is

$$r := 2Hx - x(x^*Hx) - Hx(x^*x). \tag{1}$$

Let $T$ be the preconditioner, then the preconditioned direction is $w = Tr$. Use the old searching direction to update

$$w^i = Tr^i + \beta^i w^{i-1},$$

and hence

$$x^{i+1} = x^i + t^i w^i,$$

where $t^i$ is the step size of the line search.

## Borrow the idea from LOBPCG

- Solve $x^{i+1} = \arg\min_{y \in span\{x^i, w^i\, p^i\}} E_{\text{omm}}(y)$, where $p^i = x^{i-1} - x^i$;
- Solve the small problem by alternative direction minimization.

# Preliminary results

## Examples

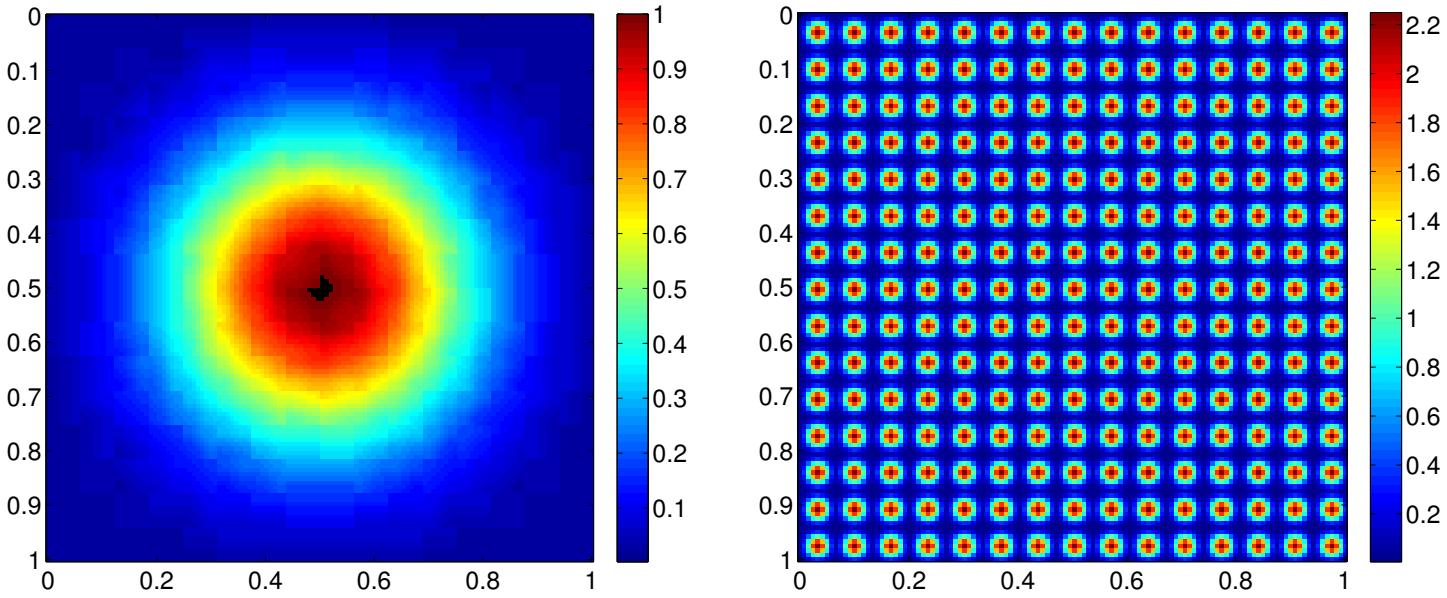



Figure: Left: a Gaussian well on the unit square $[0,1)^2$. Right: the potential energy operator on the unit square $[0,1)^2$. Plane wave discritization.

# Preliminary results

| | problem size | iter |
|---|---|---|
| LOBPCG | 576 | 107 |
| newPCGOMM | 576 | 328 |
| oldPCGOMM | 576 | >4000 |
| | | |
| LOBPCG | 1600 | 180 |
| newPCGOMM | 1600 | 773 |
| oldPCGOMM | 1600 | >4000 |
| | | |
| LOBPCG | 3136 | 276 |
| newPCGOMM | 3136 | 1308 |
| oldPCGOMM | 3136 | >4000 |

# Better solver for OMM?

## A single vector

- Solve $x^{i+1} = \arg\min_{y \in span\{x^i, w^i\ p^i\}} E_{\text{omm}}(y)$, where $p^i = x^{i-1} - x^i$;
- Solve the small problem by alternative direction minimization.

## Multi vectors

- Solve $X^{i+1} = \arg\min_{Y \in span\{X^i, W^i\ P^i\}} E_{\text{omm}}(Y)$, where $P^i = X^{i-1} - P^i$;
- Simpler idea: solve $X^{i+1} = \arg\min_{Y = X^i D_1 + W^i D_2 + P^i D_3} E_{\text{omm}}(Y)$;
- Simplest idea: solve $X^{i+1} = \arg\min_{Y = X^i \alpha_1 + W^i \alpha_2 + P^i \alpha_3} E_{\text{omm}}(Y)$;
- Question:
  - How to solve this problem efficiently?
  - How to solve it in a blocked way?

# PPCG and OMM

**Algorithm 2:** The projected preconditioned conjugate gradient (PPCG) algorithm.

**Input:** The matrix $A$, a preconditioner $T$, and a starting guess of the invariant subspace $X^{(0)} \in \mathbb{C}^{n\times k}$ associated with the $k$ smallest eigenvalues of $A$;

**Output:** An approximate invariant subspace $X \in \mathbb{C}^{n\times k}$ associated with the $k$ smallest eigenvalues of $A$;

1: $X \leftarrow \mathrm{orth}(X^{(0)})$; $P \leftarrow [\,]$;
2: **while** convergence not reached **do**
3: $W \leftarrow T(AX - X(X^*AX))$;
4: $W \leftarrow (I - XX^*)W$;
5: $P \leftarrow (I - XX^*)P$;
6: **for** $j = 1, \ldots, k$ **do**
7: $S \leftarrow [x_j, w_j, p_j]$;
8: Find the smallest eigenpair $(\theta_{\min}, c_{\min})$ of $S^*ASc = \theta S^*Sc$, where $c^*S^*Sc = 1$;
9: $\alpha_j \leftarrow c_{\min}(1)$, $\beta_j \leftarrow c_{\min}(2)$; and $\gamma_j \leftarrow c_{\min}(3)$ ($\gamma_j = 0$ at the initial step);
10: $p_j \leftarrow \beta_j w_j + \gamma_j p_j$;
11: $x_j \leftarrow \alpha_j x_j + p_j$;
12: **end for**
13: $X \leftarrow \mathrm{orth}(X)$;
14: If needed, perform the Rayleigh–Ritz procedure within span($X$);
15: **end while**

PPCG

Main ideas:

- Projected gradient: $W = (I - XX^*)W$ and $P = (I - XX^*)P$;
- Orth and Rayleigh-Ritz needed every a few (say 5) steps of iterations;

# PPCG and OMM

## OMM without preconditioner

Gradient: if $X^* = X$,

$$\begin{aligned} & 2HX - X(X^*HX) - HX(X^*X) \\ = \; & HX - X(X^*HX) + (I - X^*X)HX \\ = \; & HX - X(X^*HX) \end{aligned}$$

## PPCG without preconditioner

The projected gradient is, if $X^* = X$,

$$\begin{aligned} & (I - XX^*)(HX - X(X^*HX)) \\ = \; & HX - X(X^*HX) - XX^*HX + XX^*X(X^*HX) \\ = \; & HX - X(X^*HX) - (I - XX^*)X(X^*HX) \\ = \; & HX - X(X^*HX) \end{aligned}$$

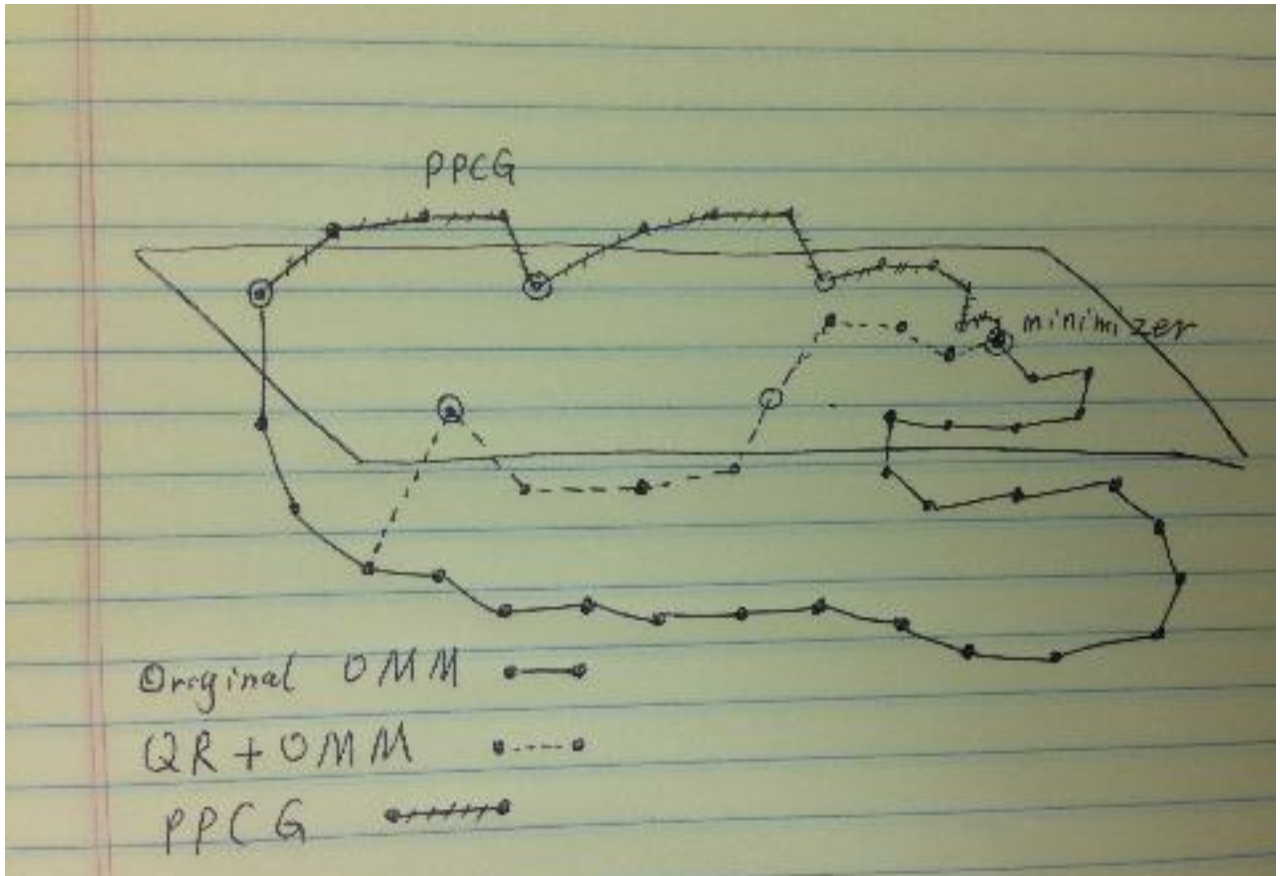
PPCG and OMM have the same gradient direction projected on to the constrain mainifold $X^*X = I$.

# PPCG and OMM

**Observation** PPCG and OMM have the same gradient direction projected on to the constrain mainifold $X^*X = I$.

**Question**

- Can we explain the convergence of PPCG in this point of view?
- Can we borrow the ideas in PPCG to solve OMM efficiently? For example, pulling $X$ approximately back to the mainifold using SCDM-k.



# Randomized blocking for PPCG?

Blocked PPCG

8: $W \leftarrow TW$;
9: $W \leftarrow (I - XX^*)W$ and $W \leftarrow (I - X_{\text{lock}}X_{\text{lock}}^*)W$;
10: $P \leftarrow (I - XX^*)P$ and $P \leftarrow (I - X_{\text{lock}}X_{\text{lock}}^*)P$;
11: **for** $j = 1, \ldots, s$ **do**
12: $S \leftarrow [X_j, W_j, P_j]$ ($P_j = 0$ if $P = [\,]$);
13: Find eigenvectors $C = [C_X, C_W, C_P]^T$ ($C_P = 0$ if $P = [\,]$) associated with the $k$ smallest eigenvalues $\Omega$ of (3)
14: $P_j \leftarrow W_j C_W + P_j C_P$;
15: $X_j \leftarrow X_j C_X + P_j$;
16: **end for**

Original idea:

- Do blocking uniformly, i.e., each block with the same size and starts from the first column of $X$;
- Perform Rayleigh-Ritz using $[X_j, W_j, P_j]$ to update $X_j$;
- $X_j^* X_j = I$ but $X_j^* X_i \neq 0$ for $i \neq j$.

Randomized idea:

- Do blocking randomly, e.g., randomly shift the uniform blocks;
- $X_j^* X_j = I$ and $X_j^* X_i$ is closer to 0 for $i \neq j$.
- Probably the randomized idea can avoid Rayleigh-Ritz with a large problem size.

# Preliminary results

| | problem size | iter |
|---|---|---|
| PPCG | 576 | 34 |
| RPPCG | 576 | 33 |
| | | |
| PPCG | 576 | 175 |
| RPPCG | 576 | 118 |
| | | |
| PPCG | 576 | 439 |
| RPPCG | 576 | 238 |

# CheFSI idea for sparse OMM

1 Initialize: density and Hamiltonian
2 Use a Lanczos method to solve a standard eigenvalue problem with a few steps of iterations. This estimates the range of desired spectrum and initial eigenpairs $(X, \Lambda)$.
3 **while** *not converged* **do**
4 Apply Chebshev filter and update $X = \rho(X)$;
5 Apply $k$ steps of OMM iterations to update $X$;
6 Normalize the eigen subspace $X$ individually;
7 Apply Rayleigh-Ritz procedure to get new eigenpairs $(X, \Lambda)$;
8 Update density and Hamiltonian

**Algorithm 1:** ChebOMM for nonlinear eigenvalue problem.

# Preliminary results

## Examples in PARSEC

| | $iter_c$ | $iter_{a1}$ | $iter_{a2}$ | $time_c$ | $time_{a1}$ | $time_{a2}$ |
|---|---|---|---|---|---|---|
| C2H6 | 17 | 16 | 16 | 9.215e+01 | 1.328e+02 | 8.879e+01 |
| CO | 20 | 17 | 19 | 1.678e+02 | 2.211e+02 | 1.717e+02 |
| H2O | 22 | 19 | 22 | 2.475e+02 | 3.341e+02 | 2.609e+02 |
| H4O2P1 | 59 | 52 | 60 | 4.352e+03 | 7.760e+03 | 5.409e+03 |
| Mg3C4O3H5 | 45 | 38 | 40 | 3.349e+03 | 8.692e+03 | 4.558e+03 |
| Mg2O3 | 45 | 42 | 51 | 3.131e+03 | 5.832e+03 | 4.148e+03 |
| Si2H4 | 16 | 15 | 16 | 4.948e+01 | 8.345e+01 | 5.743e+01 |
| SiH4 | 17 | 18 | 16 | 3.317e+01 | 6.100e+01 | 3.878e+01 |
| SiO2 | 35 | 101 | 101 | 4.379e+02 | 2.172e+03 | 1.346e+03 |
| Si28H36 | 35 | 34 | 36 | 3.983e+03 | 1.133e+04 | 6.729e+03 |
| C12H26 | 42 | 46 | 51 | 5.706e+03 | 1.802e+04 | 1.303e+04 |
| silicon dicarbide | 24 | 20 | 23 | 9.490e+01 | 1.373e+02 | 1.034e+02 |
| Al | 25 | 34 | 26 | 9.948e+01 | 2.786e+02 | 1.460e+02 |
| MgO2 | 101 | 101 | 101 | 3.021e+03 | 5.726e+03 | 3.674e+03 |
| benzene | 21 | 20 | 21 | 8.596e+02 | 1.755e+03 | 1.030e+03 |
| benzonitrile | 26 | 29 | 27 | 2.921e+03 | 6.965e+03 | 3.668e+03 |
| C60 | 22 | 23 | 23 | 3.675e+03 | 1.040e+04 | 4.860e+03 |
| Carbon Suboxide | 40 | 61 | 34 | 5.444e+02 | 1.477e+03 | 5.091e+02 |
| CO benzene | 33 | 27 | 30 | 4.722e+03 | 8.541e+03 | 5.575e+03 |
| Hydrochloric acid | 13 | 15 | 14 | 3.382e+01 | 6.544e+01 | 3.980e+01 |

Figure: c: CheFSI; a1: 2 steps of OMM iterations; a2: 10 steps of OMM iterations.