A Nonnested Augmented Subspace Method for eigenvalue problems

Haikun Dang, Pierre Jolivet, HeHu Xie, Gang Zhao, Chenguang Zhou

Thanks Prof. Pierre J for all the discussion,suggestions and kind helps!

Email: danghaikun@lsec.cc.ac.cn

Institute of Computational Mathematics,
Academy of Mathematics and Systems Sciences,
Chinese Academy of Sciences

12th Virtual FreeFEM Days
Dec. 11, 2020

Outline

BackgroundI

Applications of eigenvalue problem

- Elasticity
- Bose-Einstein Condensation
- Electronic Structure Calculation

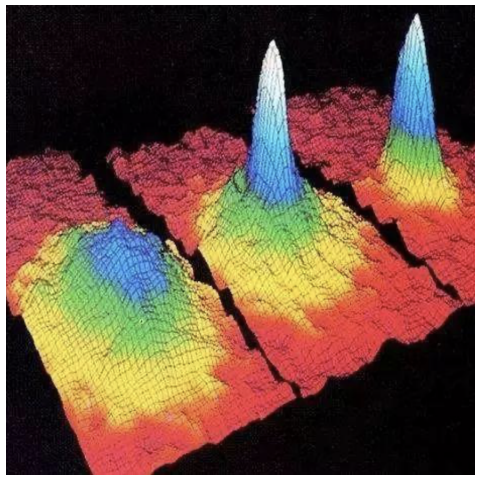
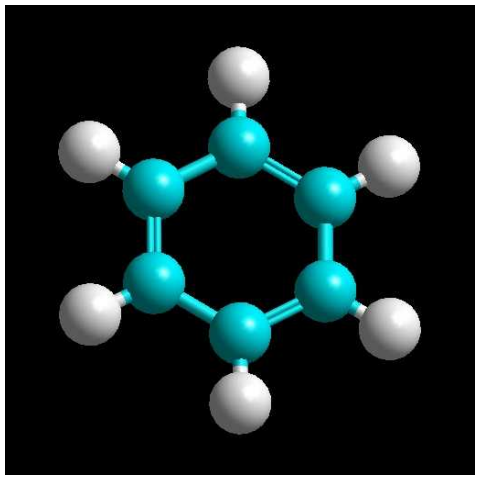




Figure: Applications

BackgroundII

The Development of Efficient Finite Element Numerical Algorithms for Solving Eigenvalue Problems

1. Efficient Finite Element Numerical Solution Method
   - Two-Grid Method[J. Xu, A. Zhou, 2001]
   - Multilevel Correction Method[Q. Lin, H. Xie, 2010]
   - Multilevel Correction Method+Tensor Assembly Technique[H. Xie, F. Xu, N. Zhang, 2019]
2. Solver for Solving Algebraic Eigenvalue Problems
   - Arnoldi, LOBPCG, Krylov-Schur...
   - ARPACK, SLEPc...
   - HYPRE BoomerAMG, PETSc-GAMG, MUMPS, Block Jacobi, ASM...

Classical Model

Second-order elliptic eigenvalue problem

Elliptic eigenvalue problem with curved boundary surface on a region $\Omega \subset \mathbb{R}^d$: Solving (λ, u), s.t.

$$\begin{cases} -\nabla \cdot (\mathcal{A}\nabla u) &= \lambda u, & \text{in } \Omega\ , \\ u &= 0, & \text{on } \partial\Omega\ , \end{cases}$$

where $\mathcal{A} = (a_{i,j})_{d\times d}$ is SPD matrix, and element $a_{i,j} \in W^{0,\infty}(\Omega)$, $(i, j = 1, 2, \cdots, d)$ is fragmentation constant, and their interface is curved(d=2 or 3).

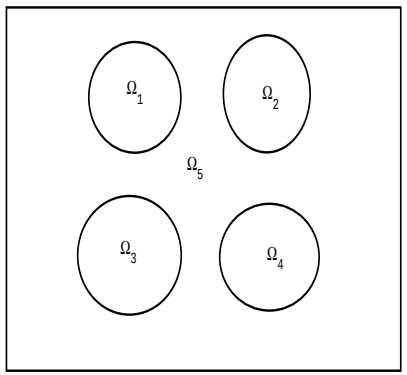


Figure: Computational area with curved edges

Second-order elliptic eigenvalue problem

Variational form

Solving $(\lambda, u) \in \mathbb{R} \times V$ s.t. $b(u,u) = 1$ and

$$a(u,v) = \lambda b(u,v), \quad \forall v \in V, \tag{2.1}$$

where $V := H_0^1(\Omega)$, $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are

$$a(u,v) = (\mathcal{A}\nabla u, \nabla v), \qquad b(u,v) = (u,v).$$

Define $\|\cdot\|_a$ and $\|\cdot\|_b$ following

$$\|v\|_a := \sqrt{a(v,v)}, \quad \|v\|_b := \sqrt{b(v,v)}, \quad \forall v \in V.$$

- Divide the calculation area Ω to get the grid $\mathcal{T}_H (T_{h_1} := T_H)$.
- Use P1 finite element to discrete u and get finite element spaces V_H and V_{h_1}.
- Perform adaptive refinement on T_{h_1} to get $\mathcal{T}_h$.

Motivation

Remark

The nesting relationship between V_H and V_h is not satisfied, i.e., $V_H \subsetneq V_h$. Mainly based on the following four reasons:

- In order to ensure the accuracy equivalent to V_h at the curved side interface;
- Even if $V_H + \mathrm{Span}\{\widetilde{\mathrm{u}}_\mathrm{h}\} = \mathrm{V}_{\mathrm{H,h}} \subsetneq \mathrm{V}_\mathrm{h}$, but $V_{H,h} \subset V$, the eigenvalue problem on $V_{H,h}$ can still be regarded as a subspace approximation of an original variational problem.
- Can be combined with movemesh or adaptive technology to further improve the discrete efficiency;
- For a sufficiently smooth interface and a suitable mesh, the linear finite element has a rich error order[J. Li, J. Zou et al., 2010];

A Nonnested Augmented Subspace Algorithm

Assuming that the eigenpair $(\lambda_h^{(\ell)}, u_h^{(\ell)})$ of the ℓth correction cycle has been obtained, the augmented subspace algorithm is described as follows:

Algorithm 1 $(\lambda_h^{(\ell+1)}, u_h^{(\ell+1)}) = \texttt{AugSubspace}(\lambda_h^{(\ell)}, u_h^{(\ell)}, V_H, V_h)$.

1: Define the linear boundary value problem as following: Solving $\widehat{u}_h^{(\ell+1)} \in V_h$ satisfy

$$a(\widehat{u}_h^{(\ell+1)}, v_h) = \lambda_h^{(\ell)} b(u_h^{(\ell)}, v_h), \quad \forall v_h \in V_h.$$

Using the multigrid method to solve the eigenfunction approximation $\widetilde{u}_h^{(\ell+1)}$ and satisfy

$$\|\widehat{u}_h^{(\ell+1)} - \widetilde{u}_h^{(\ell+1)}\|_a \leq \theta \|\widehat{u}_h^{(\ell+1)} - u_{h_k}^{(\ell)}\|_a,$$

where $\theta < 1$ and is independent of the mesh size h and iterations ℓ .

2: Define augmented subspace $V_{H,h} = V_H + \text{span}\{\widetilde{u}_h^{(\ell+1)}\}$, and solve following eigenvalue problem: Obtaining $(\lambda_h^{(\ell+1)}, u_h^{(\ell+1)}) \in \mathbb{R} \times V_{H,h}$ s.t. $b(u_h^{(\ell+1)}, u_h^{(\ell+1)}) = 1$ and satisfy

$$a(u_h^{(\ell+1)}, v_{H,h}) = \lambda_h^{(\ell+1)} b(u_h^{(\ell+1)}, v_{H,h}), \qquad \forall v_{H,h} \in V_{H,h}.$$

Error Estimate

Theorem

Assume there exists an exact eigenpair (λ, u) *such that the eigenpair approximation* $(\lambda_h^{(\ell)}, u_h^{(\ell)})$ *satisfies*

$$\|\lambda u - \lambda_h^{(\ell)} u_h^{(\ell)}\|_b \quad \leq \quad \bar{C}_\lambda \eta_a(V_H) \|u - u_h^{(\ell)}\|_a ,$$

Then, the eigenpair approximation $(\lambda_h^{(\ell+1)}, u_h^{(\ell+1)}) \in \mathbb{R} \times V_h$ *obtained by Algorithm 1 satisfies*

$$\|u - u_h^{(\ell+1)}\|_a \quad \leq \quad \gamma \, \|u - u_h^{(\ell)}\|_a + \zeta \, \|u - \mathcal{P}_h u\|_a ,$$

$$\|\lambda u - \lambda_h^{(\ell+1)} u_h^{(\ell+1)}\|_b \quad \leq \quad \bar{C}_\lambda \eta_a(V_H) \|u - u_h^{(\ell+1)}\|_a .$$

Corollary

Under the conditions of Theorem above, after executing L *augmented subspace iteration step defined by Algorithm 1, the resultant eigenpair approximation* $(\lambda_h^{(L)}, u_h^{(L)}) \in \mathbb{R} \times V_h$ *has following error estimates*

$$\|u - u_h^{(L)}\|_a \quad \leq \quad \gamma^L \|u - u_h^{(0)}\|_a + \frac{1 - \gamma^L}{1 - \gamma} \zeta \|u - \mathcal{P}_h u\|_a ,$$

$$\|\lambda u - \lambda_h^{(L)} u_h^{(L)}\|_b \quad \leq \quad \bar{C}_\lambda \eta_a(V_H) \|u - u_h^{(L)}\|_a .$$

Theorem (Computation Work Estimate)

Assume solving the linear eigenvalue problem on $V_{H,h}$ *needs work* $\mathcal{O}(M_H)$*, and the work for solving linear boundary value problem on* V_h *is* $\mathcal{O}(N_h)$*, Then the computational work included in Algorithm 1 is*:

$$\text{Work} = \mathcal{O}\left(\mathbf{N_h} + \mathbf{M_H}\right) \sim \mathcal{O}\left(\mathbf{N_h}\right).$$

Implement with FreeFEM

- $V_H \not\subset V_h$, we use $P = I_H^h : V_H \to V_h$ from FreeFEM function "interpolate" or "transferMat".
- Solve eigenvalue problem on space $V_{H,h} = V_H + \mathrm{span}\{\widetilde{u}_h\}$.

$$\begin{pmatrix} A_H & a_h \\ a_h^T & \alpha \end{pmatrix} \begin{pmatrix} u_H \\ \xi \end{pmatrix} = \lambda_h \begin{pmatrix} B_H & b_h \\ b_h^T & \beta \end{pmatrix} \begin{pmatrix} u_H \\ \xi \end{pmatrix}. \tag{3.1}$$

$$(A_H)_{i,j} = \sum_{K \in \mathcal{T}_h} \int_K \nabla \psi_{i,H} \cdot \mathcal{A} \nabla \psi_{j,H} \, dK, \quad 1 \le i, j \le N_H, \tag{3.2}$$

$$(a_h)_i = \sum_{K \in \mathcal{T}_h} \int_K \nabla \psi_{i,H} \cdot \mathcal{A} \nabla \widetilde{u}_h \, dK, \quad 1 \le i \le N_H, \tag{3.3}$$

$$\alpha = \int_\Omega \nabla \widetilde{u}_h \cdot \mathcal{A} \nabla \widetilde{u}_h \, d\Omega = \sum_{K \in \mathcal{T}_h} \int_K \nabla \widetilde{u}_h \cdot \mathcal{A} \nabla \widetilde{u}_h \, dK. \tag{3.4}$$

- Now, we just can only assemble A_H by $A_H = P^T A_h P$, $a_h = P^T A_h \widetilde{u}_h$, $\alpha = (\widetilde{u}_h)^T A_h \widetilde{u}_h$. But if it is a problem of nonlinear eigenvalues, how to efficiently assemble it?

Multilevel Correction Algorithm

A series of grid sizes h_k that satisfy the following geometric relationship

$$h_1 < H, \quad h_k = \frac{1}{\beta} h_{k-1}, \quad k = 2, \cdots, n.$$

are used to generate non-nested meshes and the corresponding finite element space V_{h_k}.

Algorithm 2 Multilevel Correction Algorithm

1: Solve eigenvalue problem on V_{h_1} : Find $(\lambda_{h_1}, u_{h_1}) \in \mathbb{R} \times V_{h_1}$, such that

$$a(u_{h_1}, v_{h_1}) = \lambda_{h_1}(u_{h_1}, v_{h_1}), \quad \forall v_{h_1} \in V_{h_1}.$$

2: For $k = 2, \cdots, n-1$, do:

1. Set $u_{h_k}^{(0)} = \mathcal{I}_{h_{k-1}}^{h_k} u_{h_{k-1}}$.
2. For $\ell = 0, \cdots, L-1$, do

$$(\lambda_{h_k}^{(\ell+1)}, u_{h_k}^{(\ell+1)}) = \texttt{AugSubspace}(V_H, \lambda_{h_k}^{(\ell)}, u_{h_k}^{(\ell)}, V_{h_k}).$$

3. Set $u_{h_k} = u_{h_k}^{(L)}$.

Error Estimate

Theorem

After executing Algorithm 2, the resultant eigenpair approximation $(\lambda_{h_n}, u_{h_n}) \in \mathbb{R} \times V_{h_n}$ *has following error estimates*

$$\|u - u_{h_n}\|_a \leq \frac{1 - (\beta\gamma^L)^n}{1 - \beta\gamma^L} \mu \delta_{h_n}(u),$$

where

$$\mu := \frac{1 - \gamma^L}{1 - \gamma} \zeta.$$

Theorem (Computation Work Estimate)

Assuming that the degrees of freedom of each layer of finite element space satisfy the relationship $N_{h_{k+1}} = \beta^d N_{h_k}, \quad k = 1, \cdots, n-1$. *Then the total work of Algorithm 2 is*

$$W_{\text{total}} = \mathcal{O}\left(LN_{h_n} + LM_H \ln N_{h_n} + M_{h_1}\right).$$

Numerical Test by FreeFEM

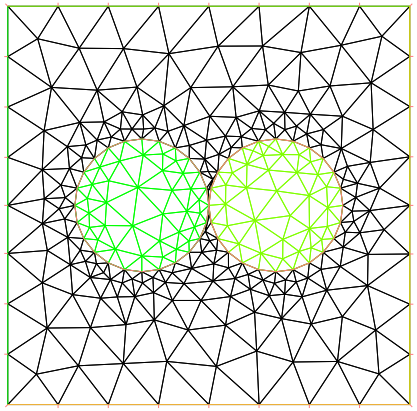
Example I（With two curved edges）

Find (λ, u) such that

$$\begin{cases} -\nabla \cdot (\mathcal{A}\nabla u) = \lambda u, & \text{in } \Omega, \\ [u] = 0, \quad [\mathbf{n}_\Gamma A \nabla u] = 0, & \text{on } \Gamma, \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = (0,2) \times (0,2)$ and $\mathcal{A}$ is defined as

$$\mathcal{A} = \begin{cases} 10, & \text{in } \Omega_1 = \{(x,y) \in \mathbb{R}^2 | (x-2/3)^2 + (y-1)^2 \leq 1/9\}, \\ 10, & \text{in } \Omega_2 = \{(x,y) \in \mathbb{R}^2 | (x-4/3)^2 + (y-1)^2 \leq 1/9\}, \\ 1, & \text{in } \Omega_3 = \Omega/(\bar{\Omega}_1 \cup \bar{\Omega}_2). \end{cases}$$



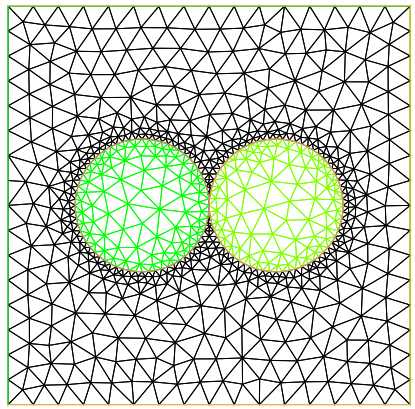


Figure: Two coarse meshes $\mathcal{T}_H$ for Example I: The left has 550 elements and the right one 1456 elements.Generated by the mesh function (border+adaptmesh) that comes with FreeFEM

Numerical Test by FreeFEM

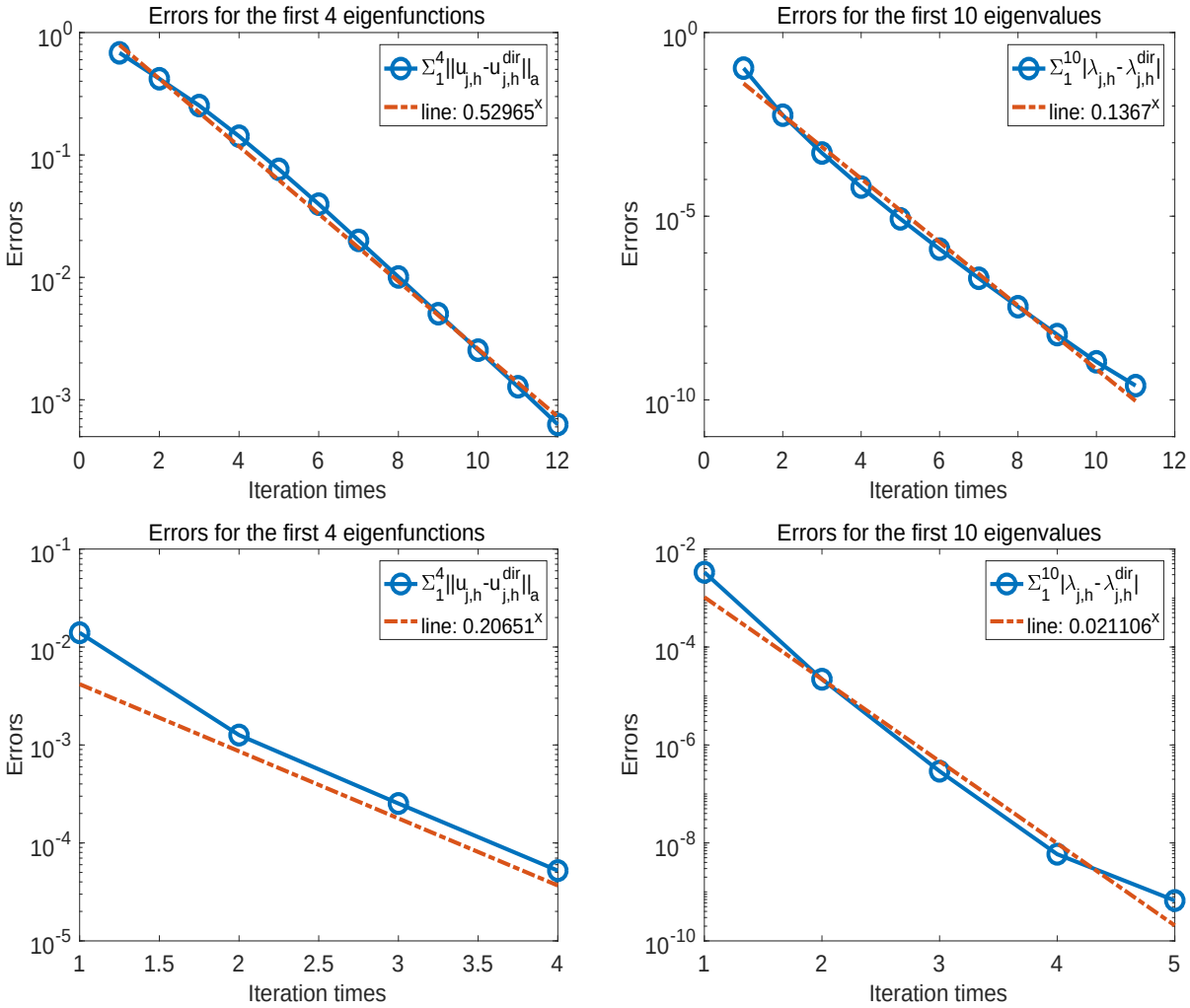


Figure: Error estimates for the first 4 and 10 eigenpair approximations by Algorithm 2. Here the coarse mesh of the top two figure has 550 elements, and bottom has 1456 elements.

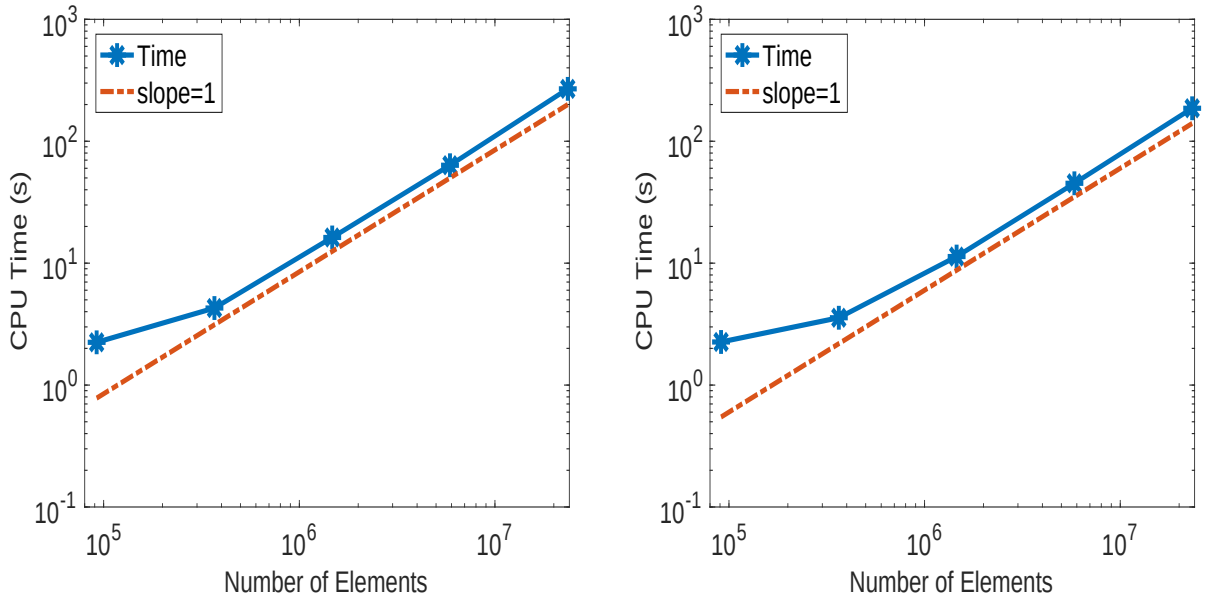


Figure: CPU time for Algorithm 2 with 32 processors, the coarse mesh of the left subfigure has 550 elements and the right subfigure has 1456 elements.

Tips

- We propose an augmented subspace algorithm for solving eigenvalue problems on complex regions with curved boundary surfaces, even when there is no nesting property.
- Futhermore, for nonlinear eigenvalue problem (even nonlinear boundary value problem), we can use this algorithm combined with moving mesh or adaptive technology to further improve the discrete efficiency;

Other Application I

Gross-Pitaevskii Equation[Bao,Tang,2003]

Find the smallest $(\lambda, u) \in \mathbb{R} \times V$ such that

$$\begin{cases} -\Delta u + V_{\text{ext}}(x)u + \zeta|u|^2 u &= \lambda u, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega, \\ \int_\Omega |u|^2 \mathrm{d}\Omega &= 1. \end{cases}$$

Where $V_{\text{ext}}(x) \in L^\infty(\Omega)$ and ζ is nonlinear coefficient. $V := \{\phi(x) \in H_0^1(\Omega) \mid E(\phi) < \infty\}$, energy $E(\phi)$ is defined as

$$E(\phi) = \int_{\mathbb{R}^d} \left[\frac{1}{2}|\nabla\phi|^2 + V_{\text{ext}}(x)|\phi|^2 + \frac{\zeta}{2}|\phi|^4\right] \mathrm{d}x.$$

Variational form

Find $(\lambda, u) \in \mathbb{R} \times V$ such that

$$\begin{cases} \widehat{a}(u, v) &= \lambda b(u, v), \quad v \in V, \\ b(u, u) &= 1, \end{cases}$$

where

$$\widehat{a}(u, v) := a(u, v) + \int_\Omega \zeta|u|^2 uv \, \mathrm{d}x, \quad b(u, v) := \int_\Omega uv \, \mathrm{d}x,$$

and

$$a(u, v) := \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x + \int_\Omega V_{\text{ext}}(x)uv \, \mathrm{d}x.$$

Multilevel Correction Algorithm with Moving Mesh

Algorithm 3 Moving mesh adaptive GPE solver

1: Set $k = 0,\ m = 0$, initial energy E_{h_0}, Perform the following Picard iteration:

(a) Find eigenpair $(u_{h_{k+1}}^{(m+1)}, \lambda_{h_{k+1}}^{(m+1)})$ such that

$$\begin{cases} a(u_{h_{k+1}}^{(m+1)}, v) + (\zeta |u_{h_{k+1}}^{(m)}|^2 u_{h_{k+1}}^{(m+1)}, v) = \lambda_{h_{k+1}}^{(m+1)} b(u_{h_{k+1}}^{(m+1)}, v), \quad \forall v \in V_{h_{k+1}}, \\ b(u_{h_{k+1}}^{(m+1)}, u_{h_{k+1}}^{(m+1)}) = 1. \end{cases}$$

(b) If $\|\delta_{h_{k+1}}\|_{\ell^2} := \|u_{h_{k+1}}^{(m+1)} - u_{h_{k+1}}^{(m)}\|_{\ell^2} < \mathrm{tol}$, then compute $E_{h_{k+1}}$; Elseif, set $m := m + 1$, goback to (a).

2: Calculate the absolute value of the energy difference ΔE on the two adjacent grid layers, if $\Delta E < \mathrm{tol}$ Endif; Elseif, the eigenfunction $u_{h_{k+1}}$ set as the adaptive function, and move mesh,set $k := k + 1$, assign the updated eigenfunction and eigenvalue to $(\lambda_{h_{k+1}}^{(0)}, u_{h_{k+1}}^{(0)})$.

3: Set $(\lambda_{h_{k+1}}^{(0)}, u_{h_{k+1}}^{(0)})$as initial input, for $\ell = 0,\ 1,\ \cdots,\ L$ do

$$(\lambda_{h_{k+1}}^{(\ell+1)}, u_{h_{k+1}}^{(\ell+1)}) = \texttt{AugSubspace}(\lambda_{h_{k+1}}^{(\ell)}, u_{h_{k+1}}^{(\ell)}, V_H, V_{h_{k+1}}).$$

4: Compute $E_{h_{k+1}}$, goback to step 2 .

Implement with FreeFEM

- Use r-adaptive for algorithm design: Here we use the three-dimensional anisotropy adaptive software Mshmet and Mmg3d (the three-dimensional module in Mmg) based on the principle of $\mathcal{M}$-unit side length in FreeFEM. The metric used is the given adaptive function u in Correction of Hessian matrix $H_u(z)$ at grid node z
- Assemble the stiffness matrix in (2.6) still uses the algebraic assembly form of Algorithm 1.

$$\begin{pmatrix} A_H & a_h \\ a_h^T & \alpha \end{pmatrix} \begin{pmatrix} u_H \\ \xi \end{pmatrix} = \lambda_h \begin{pmatrix} B_H & b_h \\ b_h^T & \beta \end{pmatrix} \begin{pmatrix} u_H \\ \xi \end{pmatrix}. \tag{5.1}$$

- Disscussion: Can it be implemented efficiently ? [Xie, Xu, Zhang, 2019] Define Tensor T_H

$$\begin{aligned}
(A_H)_{i,j} &= (A_{H,1})_{i,j} + (A_{H,2})_{i,j}, \\
(A_{H,1})_{i,j} &= \int_\Omega \nabla\phi_{i,H}\nabla\phi_{j,H}\,d\Omega + \int_\Omega w\phi_{i,H}\phi_{j,H}\,d\Omega, \\
(A_{H,2})_{i,j} &= \int_\Omega \zeta\,(u_H + \alpha\tilde{u}_h)^2\,\phi_{i,H}\phi_{j,H}\,d\Omega \\
&= \int_\Omega \zeta\left((u_H)^2 + 2\alpha u_H\tilde{u}_h + \alpha^2(\tilde{u}_h)^2\right)\phi_{i,H}\phi_{j,H}\,d\Omega \\
&= \underbrace{\int_\Omega \zeta\,(u_H)^2\,\phi_{i,H}\phi_{j,H}\,d\Omega}_{\sim} + 2\alpha\underbrace{\int_\Omega \zeta\tilde{u}_h u_H\phi_{i,H}\phi_{j,H}\,d\Omega}_{\cdots} \\
&\quad + \alpha^2\underline{\int_\Omega \zeta\,(\tilde{u}_h)^2\,\phi_{i,H}\phi_{j,H}\,d\Omega} \\
&:= \underbrace{(A_{H,2,1})_{i,j}}_{\sim} + 2\alpha\underbrace{(A_{H,2,2})_{i,j}}_{\cdots} + \alpha^2\underline{(A_{H,2,3})_{i,j}}.
\end{aligned}$$

$$(T_H)_{i,j,k} = \int_\Omega \zeta\tilde{u}_h\phi_{k,H}\phi_{i,H}\phi_{j,H}\,d\Omega$$

$$A_{H,2,2} = T_H \cdot \mathbf{u}_H.$$

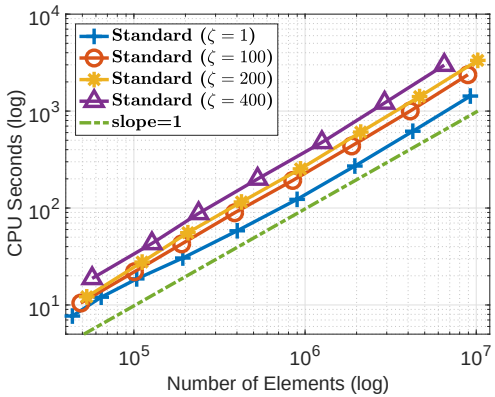
Numerical Test by FreeFEM

Example II（cubic GPE with 72 processors）

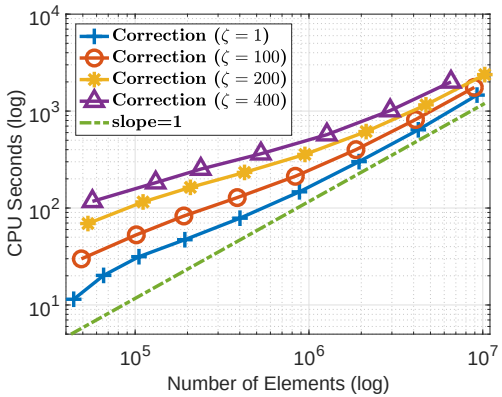
GPE on 3d cubic domain $\Omega := [-1,1]^3$

$$\begin{cases} -\Delta u + V_{\text{ext}}(x)u + \zeta|u|^2 u = \lambda u, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \\ \int_\Omega |u|^2 \mathrm{d}\Omega = 1, \end{cases}$$

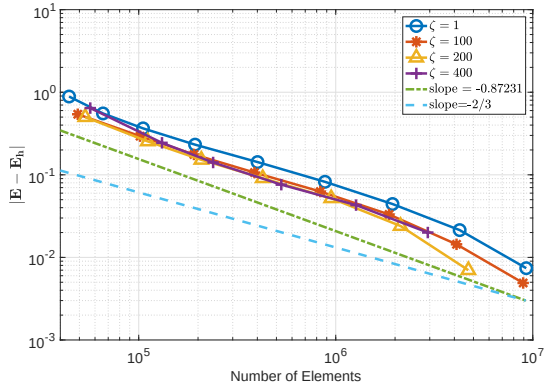
where $V_{\text{ext}}(x) = x^2+y^2+z^2+50\big(\sin^2(2\pi x)+\sin^2(2\pi y)+\sin^2(2\pi z)\big)$. And $\zeta = 1, 100, 200, 400$.



(a) AFEM



(b) Adaptive MLC



(c) Energy accurary of MLC

Elements(DIR)	2560	48955	100811	190179	387669	847087	1871655	4118185	8936327
CPUTIME(DIR)	2.505	5.787	10.529	18.613	36.782	**76.925**	**170.334**	**401.698**	**987.744**
Elements(MLC)	2560	48955	100983	192709	387390	844623	1878544	4120728	8935509
CPUTIME(MLC)	3.840	18.337	27.075	35.533	43.684	**53.840**	**71.621**	**100.524**	**155.412**

Table: CPU time (second):$\zeta = 100$

Elements(DIR)	2560	53045	111692	206744	424964	940339	2114604	4666878	10212840
CPUTIME(DIR)	2.813	7.132	14.654	27.906	54.462	**118.392**	**293.482**	**691.973**	**1675.090**
Elements(MLC)	2560	53045	111695	205869	424072	944343	2133012	4697122	10238513
CPUTIME(MLC)	4.144	32.448	47.356	59.100	72.264	**89.014**	**115.231**	**162.212**	**258.096**

Table: CPU time (second):$\zeta = 200$

Elements(DIR)	2560	57019	127421	238696	528077	1253909	2913851	6504404
CPUTIME(DIR)	5.440	13.958	27.966	53.371	**117.315**	**273.471**	**718.304**	**1788.264**
Elements(MLC)	2560	57019	127267	236254	532572	1273177	2956857	6553167
CPUTIME(MLC)	4.088	49.907	69.037	85.878	**104.274**	**128.115**	**169.981**	**250.949**

Table: CPU time (second):$\zeta = 400$

- The CPU time does not include Mmg part.

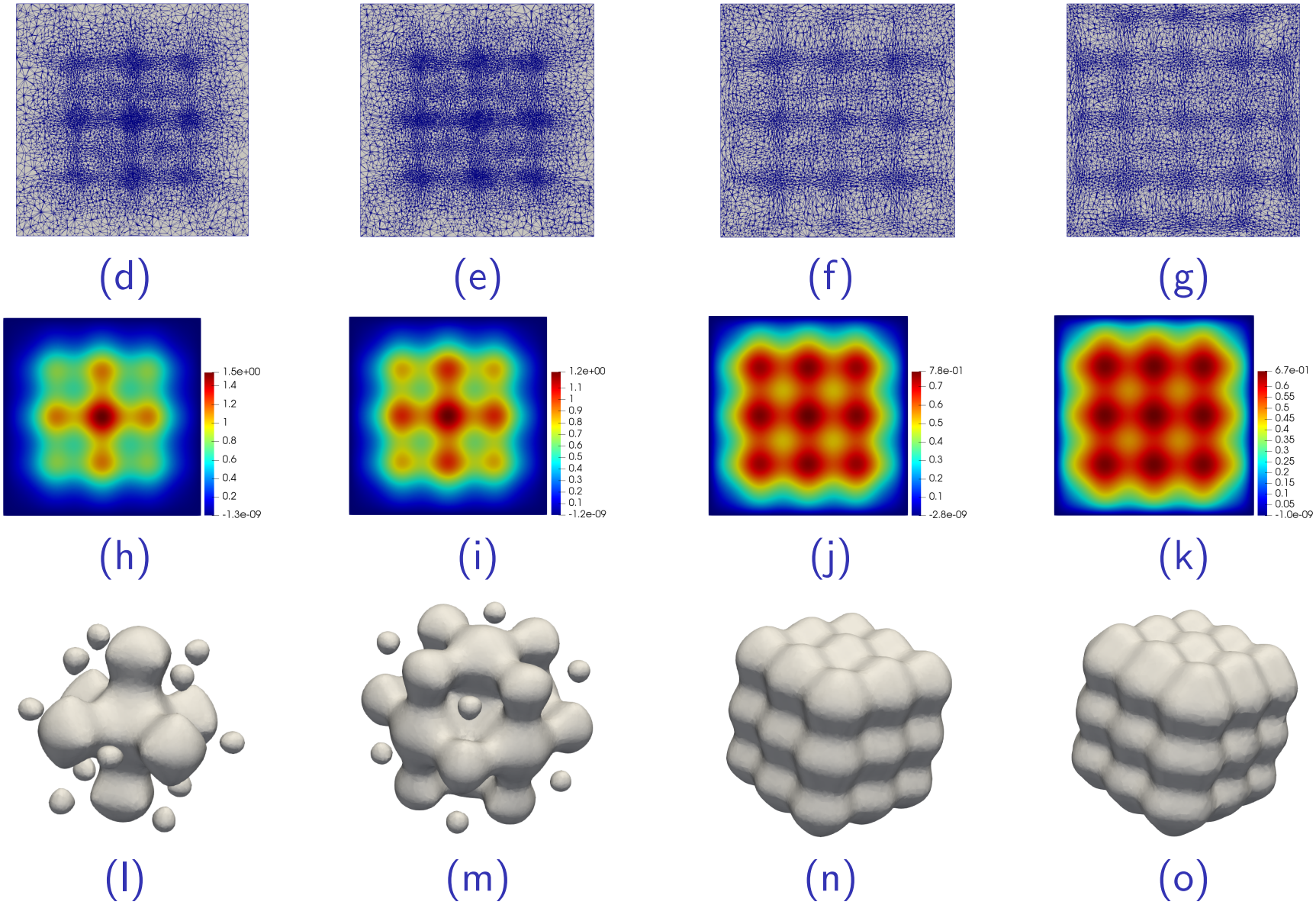


Figure: The ground state wave function: (d)-(g) The cross section of adaptive mesh ($\zeta = 1$, 10, 100, 200), (h)-(k) The cross section of the ground state wave function ($\zeta = 1$, 10, 100, 200), (l)-(o) The isosurface of the ground state wave function ($\zeta = 1$, 10, 100, 200).

Other Application II

Kohn-Sham Equation

1. Find $(\Lambda, \Psi) \in \mathbb{R}^N \times V$ such that

$$\begin{cases} H_\Psi \psi_i &= \lambda_i \psi_i \ \text{ in } \Omega, \quad i = 1, \cdots, N, \\ \int_\Omega \psi_i \psi_j dx &= \delta_{ij}, \end{cases}$$

where $V := (H_0^1(\Omega))^N$. Define symbol $\Lambda = (\lambda_1, \cdots, \lambda_N)$, $\lambda_i = (H_\Psi \psi_i, \psi_i)$ and Hamiltonian H_Ψ

$$H_\Psi = -\frac{1}{2}\Delta + V_{\text{ext}} + \int_\Omega \frac{\rho_\Psi(y)}{|\cdot - y|} dy + V_{\text{xc}}(\rho_\Psi).$$

2. Variational form: Find $(\Lambda, \Psi) \in \mathbb{R}^N \times \mathcal{H}$ such that

$$\begin{cases} (H_\Psi \psi_i, v) &= \lambda_i (\psi_i, v), \quad \forall v \in H_0^1(\Omega), \quad i = 1, 2, \cdots, N, \\ \int_\Omega \psi_i \psi_j dx &= \delta_{ij}, \quad i, j = 1, 2, \cdots, N \end{cases}$$

Numerical Test by FreeFEM

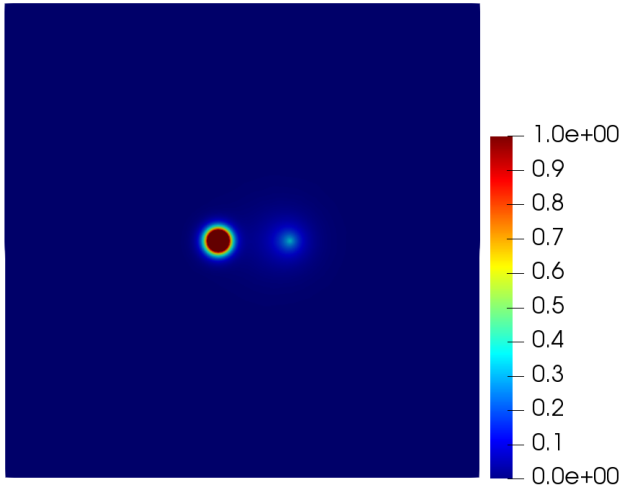
Example III: （LiH with 72 processors）

$$\begin{cases} \left(-\frac{1}{2}\Delta - \frac{3}{|x-R_1|} - \frac{1}{|x-R_2|} + \int_\Omega \frac{\rho_\Psi(y)}{|x-y|}\mathrm{d}y + V_{\mathrm{xc}}(\rho)\right)\psi_i = \lambda\psi_i, \quad \text{in } \Omega, \\ \psi_i = 0, \quad \text{on } \partial\Omega, \\ \int_\Omega \psi_i\psi_j \mathrm{d}x = \delta_{ij}, \end{cases}$$

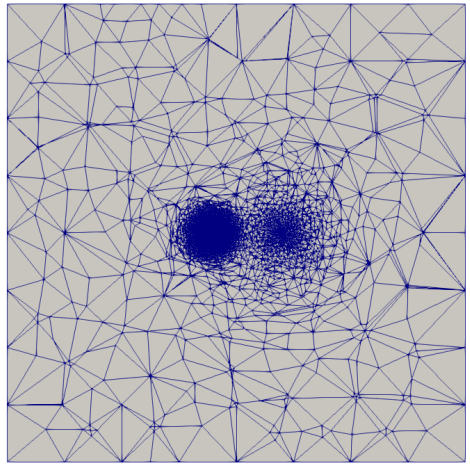
where $\Omega = [-10, 10]^3$, Li and H Nuclear position R_i, $i = 1, 2$ are $R_1 = (-1.0075, 0, 0)$ and $R_2 = (2.0075, 0, 0)$, and Nuclear charge are 3 and 1. The experimental value of the non-relativistic ground state energy of LiH molecule is -8.070 hartrees[Cencek W,Rychlewski J,2000].

Elements(DIR)	9008	11463	15943	29015	57078	113729	240435	518786
CPU TIME(DIR)	7.024	10.231	14.188	18.587	**23.694**	**32.339**	**46.215**	**74.595**
Elements(MLC)	8819	11240	17235	31297	55690	113168	236780	510949
CPU TIME(MLC)	8.642	11.176	15.370	19.260	**23.340**	**30.560**	**38.086**	**49.016**

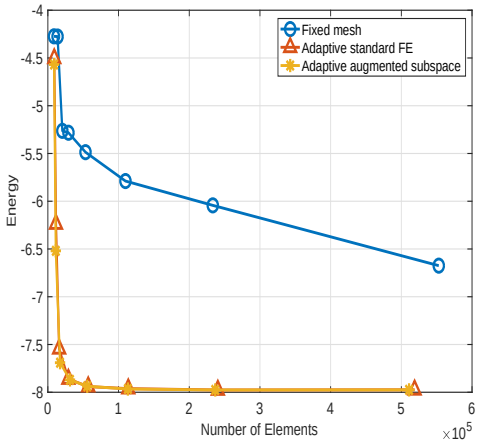
Table: CPU TIME(s): Adaptive standard finite element method (direct method) and adaptive augmented subspace method (correction method).



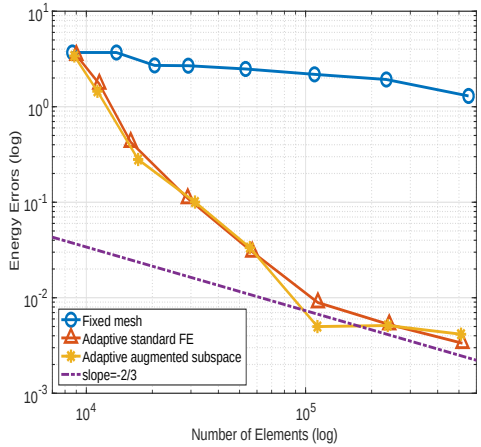
(a) Electron density distribution



(b) Adaptive mesh of electron density



(c) Ground state energy



(d) Ground state energy error

Computing Bottleneck

- So far, the coarse grid $\mathcal{T}_H$ in above-mentioned numerical tests are global. When the coarse grid is also parallel, the parallel scalability of "transferMat" is a challenge.
- CPU time of serial moving mesh operations(Serial Mmg) are very large, but fortunately, Prof.Pierre will introduce parallel grid adaptive operations(ParMmg in FreeFEM) at this conference, hoping that this operation will reduce the overall calculation time.
- Since we need to construct the corresponding stiffness matrix on augmented space $V_{H,h}$, and in order to ensure the approximate accuracy of the fine grid, we need to assemble the stiffness matrix on the current fine grid $\mathcal{T}_h$. And if it is possible to do following in a parallel environment, then the cost of assembling matrix A_H will reduce.

$$\text{varf}(u, v) = \text{intN}(Th)(.....) \text{ and Mat } A_H = \text{varf}(V_H, V_H, \text{tgv} = -2)$$

Summary

Summary

- We propose an augmented subspace method based on non-nested and adaptive grid to solve nonlinear problems, including eigenvalue problem, nonlinear eigenvalue problem, actually nonlinear boundary value problem also can use this frame.
- This method essentially solves the linear boundary value problem on the fine layer, and solves the nonlinear problem on the constructed special augmented subspace. The amount of calculation is greatly reduced while the accuracy is guaranteed.
- If the tensor data structure and the product operation of the tensor and the vector can be realized in FreeFEM, then the computation work can be the same as the linear equation for the nonlinear term equation with polynomial form.
- In our future work, we will apply this method to solve Poisson-Boltzmann equation and Poisson-Nernst-Planck equation in biological field.

Thanks Prof. Pierre Jolivet for all help and kind discussions!