CAAM Rice University

CAAM 551: Final Project

Spectral Element Method and Additive Schwarz Preconditioner

Zheng Wang

1 Problem Description

In this final project, I want solve the 2D Poisson's equation

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{1}$$

where $\Omega = [-1, 1]^2$.

This equation has already been solved in many different ways. In this project, I have applied the spectral element method (SEM) and the additive Schwarz preconditioner to this problem.

The purpose is implementing an effective SEM solver and examining the additive Schwarz preconditioner.

2 Methods

This project focuses on two methods: one is the spectral element method for solving the PDE; the other is the additive Schwarz method for preconditioning. This section briefly introduces these two methods.

2.1 Spectral Element Method

The SEM is a formulation of the finite element method (FEM). The discretization of the SEM is similar to that of the FEM, and details of the SEM can be referred to [1].

To formulate the linear system of the SEM, we multiply a test function v on both sides of the PDE (1), integrate both sides over the domain Ω, and apply integration by parts. The following equality can be thereby obtained,

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v \tag{2}$$

Denoting u as the linear combination of basis functions

$$u = \sum_i \hat{u}_i \varphi_i \tag{3}$$

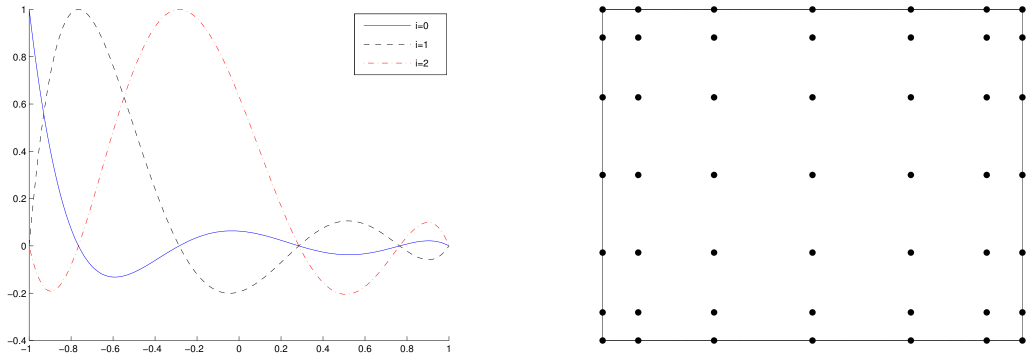
and letting v run over all the basis functions, we arrive at the following system,

$$A\hat{u} = b \tag{4}$$

where $A_{i,j} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$ is the stiffness matrix, $b_i = \int_\Omega f\varphi_i$ is the right-hand side and $\hat{u} = [\hat{u}_1, \hat{u}_2 \ldots]$ are the coefficients of basis functions in the solution.

Different from the FEM, the SEM uses high degree piecewise polynomials as basis functions. In order to make high order polynomials applicable, the SEM employs rectangular elements and Gauss-Lobatto-Legendre (GLL) nodes which enables the Gauss quadrature in the computation.

Since matrix A in equation (4) is symmetric positive definite (s.p.d.), the conjugate gradient (CG) method is suitable for solving the linear system (4).



(a) (b)

Figure 1: (a) 1D GLL Interpolation. (b) 2D GLL Nodes

However, the condition number of A grows quickly as the polynomial degree N increases or the mesh sizes h decreases. According to the literature, the condition number of A is in the scale of $O(N^4/h^2)$. Because of this fact, a preconditioned conjugate gradient (PCG) method is a better choice for the SEM solver.

2.2 Additive Schwarz Preconditioner

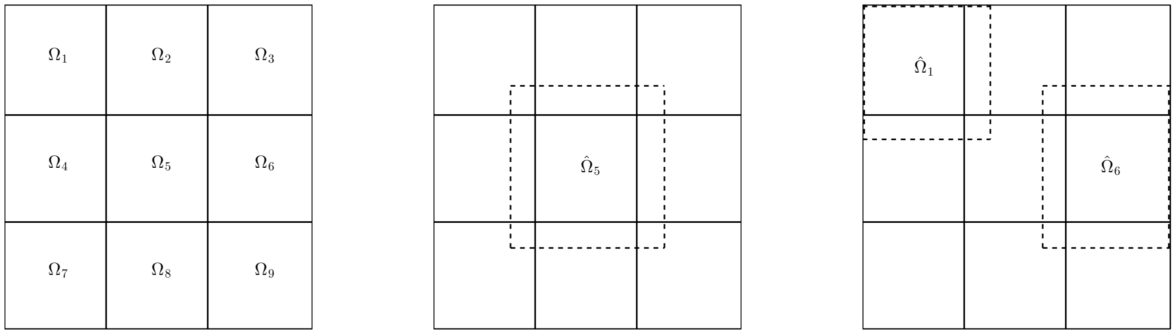
In the following context, the capital letter M is denoted as the preconditioner matrix and we want to minimize $\text{cond}(M^{-1}A)$ at a low cost. The preconditioner in this project is the additive Schwarz preconditioner. I read the first several sections of [2] which is a review of domain decomposition and implemented it to my SEM solver.

Additive Schwarz algorithm is an overlapping subdomain algorithm. To initiate this algorithm, we need an overlapping decomposition of domain Ω into p subregions $\hat{\Omega}_1, \ldots, \hat{\Omega}_p$. To construct this, let $\Omega_1, \ldots, \Omega_p$ be a nonoverlapping decomposition, where Ω_i are chosen from a mesh τ^h of size h. Next, we extend Ω_i to $\hat{\Omega}_i$ by select all the points within a distance of βh from Ω_i, thus

$$\hat{\Omega}_i = \{x \in \Omega : d(x, \Omega_i) < \beta h\} \tag{5}$$

See Figure 2 for illustration of a 2D region.

Once all the $\hat{\Omega}_i$ are well defined, we begin to define restriction maps R_i, extension maps R_i^T



(a) (b) (c)

Figure 2: (a) Nonoverlapping partition (b) Overlapping subdomain: $\hat{\Omega}_5$ (c) Overlapping subdaomains: $\hat{\Omega}_1$ and $\hat{\Omega}_6$

and local matrices A_i corresponding to each subdomain $\hat{\Omega}_i$. Let A be $n \times n$, $\{\hat{I}_1, \ldots, \hat{I}_p\}$ denote the indices of the nodes in $\hat{\Omega}_i$, and the cardinality $|\hat{I}_i| = \hat{n}_i$. For each region $\hat{\Omega}_i$, R_i denotes the $\hat{n}_i \times n$ restriction matrix. R_i maps a vector x of length n to a vector $R_i x$ of length $\hat{n}_i$. If $\hat{I}_i(k)$ is the kth element of $\hat{I}_i$, we can write

$$(R_i x)_k = (x)_{\hat{I}_i(k)} \tag{6}$$

R_i^T is the transpose of R_i and it extends a vector x_i of length $\hat{n}_i$ to a vector $R_i^T x_i$ of length n by filling the rest entries which do not have indices in $\hat{I}_i$ with 0s. We have

$$(R_i^T x_i)_k = \begin{cases} (x_i)_k & \text{for} \quad k \in \hat{I}_i \\ 0 & \text{for} \quad k \notin \hat{I}_i \end{cases} \tag{7}$$

Finally we let $A_i = R_i A R_i^T$ which is a principal submatrix of A corresponding to subdomain $\hat{\Omega}_i$.

A one-level preconditioner can be then generalized as the following,

$$M_1^{-1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i \tag{8}$$

The overlapping enables one subdomain to have a communication with its halos. If we pay a small cost to build a global communication of information for the subdomains, we can do a better job.

To this end, we need an assumption that the fine grid τ^h is a refinement of a coarse mesh τ^H. Suppose there are n_c coarse gird vertices and let R_0^T denote a $n \times n_c$ matrix which extends coarse gird vertices to the fine grid. Similarly, R_0 is the transpose of R_0^T and $A_0 = R_0 A R_0^T$. the two-level preconditioner M_2 is

$$M_2^{-1} = \sum_{i=0}^{p} R_i^T A_i^{-1} R_i \tag{9}$$

3 Implementation Details

This section discusses the implementation details in the project. The source code can be found at `https://github.com/zw14/CAAM551`.

3.1 SEM Mesh

In this project, the mesh grid is a structured grid. It is equally spaced in both dimensions. We set

$$dx = dy = h \tag{10}$$

since our domain Ω is $[-1, 1]^2$, each subdomain Ω_i is in the form of

$$\Omega_i = [-1 + (m-1) \cdot dx, -1 + m \cdot dx] \times [-1 + (n-1) \cdot dy, -1 + n \cdot dy] \tag{11}$$

3.2 Overlapping Region

In the implementation, the overlapping domain $\hat{\Omega}_i$ extends Ω_i by one layer of its halo nodes. Hence, the higher the polynomial degree, the smaller the overlapping region.

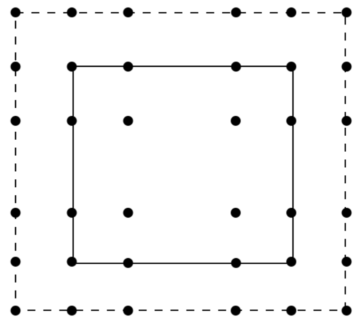


Figure 3: Overlapping Region: the solid line square is the non-overlapping element Ω_i, the dash line square is the overlapping element $\hat{\Omega}_i$

3.3 Matrix Free

In the PCG process, matrix-vector multiplications are conducted. Instead of storing a matrix, we regard the matrix as an operator and implement the matrix-vector multiplication without storing the matrix.

This technic is applied to the stiffness matrix A, restriction matrix R_i and extension matrix R_i^T.

For A, we compute the contribution of each element, and scatter the contribution to the vector. For R_i, we collect the entries of a given vector by a predefined index set. For R_i^T, we scatter the values in a small vector to a large vector.

However, the matrix free technic is not applied to the overlapping subdomain matrix A_i, because A_i^{-1} is required in the PCG process.

3.4 BLAS & LAPACK

To obtain A_i^{-1}, BLAS & LAPACK is used to compute the LU factorization of A_i. The factorized A_i is computed one time and stored in the preprocessing part, and hence can be used in the PCG iterations.

3.5 OpenMP

Both SEM and additive Schwarz preconditioner are highly parallelizable. The implementation uses OpenMP to accelerate the iterations.

4 Results

In the test, we choose different mesh sizes h and polynomial order N, and compare the number of iterations and time cost among naive CG, one-level additive Schwarz and two-level additive Schwarz.

Some basic configurations are:

- In the two-level additive Schwarz preconditioner, the coarse mesh are made up of the center nodes in the subdomains in the fine mesh.
- The terminating criteria for CG/PCG iterations is that the residue r is less than a tolerance of $1e-3$.
- The exact solution used in the test case is $u = e^{(x^2-1)(y^2-1)} - 1$. The L^2 errors between the exact solutions and the numerical solutions are computed.
- The hardware for test is Intel(R) Core(TM) i7-4820K CPU @ 3.70GHz.
- Ten threads are used in the OpenMP parallelization.

Two test cases with different problem sizes are presented as follows.

4.1 $h = 0.2, N = 5$, degree of freedom $= 2401$

Methods	No. of Iterations	Time Cost	Residue Norm	L^2 Error
Naive CG	74	3.2e-2s	9.6e-4	1.0-14
One-level Schwarz	17	1.9e-2s	8.9e-4	4.9e-15
Two-level Schwarz	18	1.9e-2s	7.7e-4	2.8e-15

Table 1: Test Case 1

4.2 $h = 0.1, N = 10$, degree of freedom $= 39601$

Methods	No. of Iterations	Time Cost	Residue Norm	L^2 Error
Naive CG	319	2.44s	9.0e-4	6.1e-19
One-level Schwarz	52	0.94s	9.0e-4	1.7e-19
Two-level Schwarz	52	0.94s	9.7e-4	2.1e-19

Table 2: Test Case 2

4.3 $h = 0.08, N = 15$, degree of freedom $= 139876$

Methods	No. of Iterations	Time Cost	Residue Norm	L^2 Error
Naive CG	662	22.26s	9.9e-4	7.9e-22
One-level Schwarz	94	7.93s	8.9e-4	7.9e-23
Two-level Schwarz	96	8.07s	8.6e-4	4.5e-23

Table 3: Test Case 3

5 Conclusion

It can be seen from the preceding section that the additive Schwarz preconditioner reduces the number of iterations and lowers the computational time cost.

However, we easily notice that the two-level Schwarz preconditioner is not better than the one-level Schwarz preconditioner. I haven't figured out the reason of this result in the project. My guesses are: (1) The coarse grid is not properly chosen; (2) The overlapping region is too small; (3) The Poisson equation is so simple that the global communication does not bring any benefits. (4) Bugs in the implementation.

The one-level Schwarz preconditioner can be used in my future research. Meanwhile, I would like to read more materials about the additive Schwarz preconditioner, especially some implementation details, to figure out how the two-level preconditioner works.

References

[1] Deville, Michel O., Paul F. Fischer, and Ernest H. Mund, eds. High-order methods for incompressible fluid flow. Vol. 9. Cambridge University Press, 2002.

[2] Chan, Tony F., and Tarek P. Mathew. "Domain decomposition algorithms." Acta numerica 3 (1994): 61-143.