An Overlapping Balancing Domain Decomposition Method for Elliptic PDEs

Jung-Han Kimn and Blaise Bourdin

Department of Mathematics
and
The Center for Computation and Technology
Louisiana State University
kimn@math.lsu.edu
Balancing Domain Decomposition Method
- Substructuring Algorithms on Nonoverlapping Subdomains
- Local Problems: Neumann Boundary Conditions
- Compatibility Conditions
- Using $D_i, \sum \bar{R}_i \bar{R}_i^T D_i = I$ (Inverse of the counting functions)

Balancing Domain Decomposition Method extended to overlapping subdomains.

Overlapping Balancing Domain Decomposition
- Extended Substructures: Overlapping Subdomains
- $\sum_i \bar{R}_i^\delta (\bar{R}_i^\delta)^T D_i^\delta = I$ ($D_i^\delta$ is related to the regular PU.)
- Compatibility Conditions: Satisfied through Coarse Problem based on PU
Hybrid type preconditioner

For each CG iteration, the main cost is:
- one Neumann solver on each extended region
- one coarse problem.

Coarse problem is easy and cheap to construct

The coarse matrix is sparse

Less sensitive to the roughness of boundary.

A convergence theory for the Poisson problem (Sarkis, Kimn).

Application for the Helmholtz equation (Sarkis, Kimn)
Model problem & PU

\[ -\Delta u = f \quad \text{in} \quad \Omega, \]
\[ u = g \quad \text{on} \quad \partial \Omega \]  

Implementation of PU

- Overlapping subdomain
  - Automatically generated from nonoverlapping subdomain.
  - Inspired by fast marching algorithms (Unstructured Meshes)

- Partition of Unity Functions
  - One function per subdomain
  - Constructed simultaneously to Overlapping subdomain.
  - Communication between Coarse and Fine meshes
  - Restriction and Interpolation operators
  - A value for each processor (Poisson Equation)
Figure 1: My Elems (Light-Blue)
Figure 2: My Nodes (Red dots)
Figure 3: Ghost Elems (Light-Green)
Figure 4: Ghost Nodes (Yellow-Green dots)
The First PU: $\theta_i^\delta$

- No Dirichlet Treatment: **Not** Include the Dirichlet Boundary PU
- $\vartheta_i^\delta: \vartheta_i^\delta(x) = 1$ for $\Omega_i$ and $\vartheta_i^\delta(x) = 0$ for $\Omega \setminus \Omega_i$, $i = 1, \ldots, N$
- $(k)$th layer: $\vartheta_i^\delta(x) = (\delta - k)/\delta$

$$\theta_i^\delta = I_h\left( \frac{\vartheta_i^\delta}{\sum_{j=1}^N \vartheta_j^\delta} \right), \quad i = 1, \ldots, N$$

$$\sum_{i=1}^N \theta_i^\delta(x) = 1, \quad 0 \leq \theta_i^\delta(x) \leq 1, \text{ and } |\nabla \theta_i^\delta(x)| \leq C/\delta h$$

Used as Balancing Weights ($D_i^\delta$)
The Second PU: $\hat{\theta}_i^\delta$

- Dirichlet Treatment: Control Decaying to zero near $\partial\Omega_D$
  
  \[ \hat{\theta}_D^\delta(x) = 1 \text{ and } \hat{\theta}_D^\delta(x) = 0 \text{ for nodes } x \text{ on } \partial\Omega_D \text{ and } \Omega \setminus \Omega_D^\delta \]

- $\hat{\theta}_i^\delta(x) = 1$ for $\Omega_i$ and $\hat{\theta}_i^\delta(x) = 0$ for $\Omega \setminus \Omega_i^\delta$, $i = 1, \ldots, N$

- $(k)st$ layer: $\hat{\theta}_D^\delta(x) = (\delta - k)/\delta$

\[
\hat{\theta}_i^\delta = I_h \left( \frac{\hat{\theta}_i^\delta}{\sum_{D,j=1}^N \hat{\theta}_j^\delta} \right)
\]

- $0 \leq \hat{\theta}_i^\delta(x) \leq 1$, $|\nabla \hat{\theta}_i^\delta(x)| \leq C/(\delta h)$, $\forall x \in \Omega$, and
  \[
  \sum_{D,j=1}^N \hat{\theta}_j^\delta = 1
  \]

- Used $\hat{\theta}_i^\delta(x)$ as Coarse Spaces

- $\partial\Omega_i^\delta \cap \Omega_D^\delta = \emptyset$ then $\hat{\theta}_i^\delta = \theta_i^\delta$
Figure 5: An illustration of a function $\theta_i^\delta$

Figure 6: An illustration of a function $\hat{\theta}_i^\delta$
\( \hat{V}_0^\delta \): linear combination of \( \hat{\theta}_i^\delta, i = 1, \cdots, N. \)

\textbf{Not} include the \( \hat{\theta}_D^\delta \)

Restriction matrix \( R_0^\delta : V \to V_0^\delta \) \( \hat{P}_0 : V \to \hat{V}_0^\delta \) by: for any \( u \in V \)

\[ a(\hat{P}_0^\delta u, v) = a(u, v), \quad \forall v \in \hat{V}_0^\delta. \]

\( \hat{P}_0^\delta = (R_0^\delta)^T (A_0^\delta) + R_0^\delta \), where \( A_0^\delta = R_0^\delta A (R_0^\delta)^T \)
\( \hat{R}_i^\delta : V \rightarrow \bar{V}_i^\delta, \quad \bar{V}_i^\delta = V(\Omega_i^\delta) \cap H^1_{\partial \Omega_D \cap \partial \Omega_i^\delta}(\Omega_i^\delta) \)

\[ R_i^\delta = \hat{R}_i^\delta \hat{\theta}_i^\delta \]

\[ V = (R_1^\delta)^T \hat{V}_1^\delta + (R_2^\delta)^T \hat{V}_2^\delta + \cdots + (R_N^\delta)^T \hat{V}_N^\delta \]

\( \hat{V} \subset \bar{V} \) such that \( \hat{V} \perp kernel(\bar{V}) \)

\[ V = \hat{V}_0^\delta + (R_1^\delta)^T \hat{V}_1^\delta + (R_2^\delta)^T \hat{V}_2^\delta + \cdots + (R_N^\delta)^T \hat{V}_N^\delta \]

\( \hat{T}_i^\delta : V \rightarrow \bar{V}_i^\delta \)

\[ a_{\Omega_i^\delta}(\hat{T}_i^\delta u, v) = a(u, (R_i^\delta)^T v), \quad \forall v \in \bar{V}_i^\delta, \quad i = 1, \ldots, N \]

\[ T_i^\delta = (R_i^\delta)^T \hat{T}_i^\delta \]

\[ T_{hyb}^\delta = \hat{P}_0^\delta + (I - \hat{P}_0^\delta)(I - \sum_{i=1}^N T_i^\delta)(I - \hat{P}_0^\delta) \]

Floating Subdomain: Compatibility by Solving a Coarse Problem
Table 1: Two-level Hybrid/CG using PU: $\hat{\theta}_i^\delta$ (coarse) and $\theta_i^\delta$ (local), Neumann local solver considering original conditions for solving the Poisson’s equation with overlapping size $\delta$.

<table>
<thead>
<tr>
<th>$n \times n$</th>
<th>$n_{sub} \times n_{sub}$</th>
<th>$\delta$</th>
<th>$\text{iter}$</th>
<th>$\text{cond}$</th>
<th>$\text{max}$</th>
<th>$\text{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>301 $\times$ 301</td>
<td>30 $\times$ 30</td>
<td>1</td>
<td>28</td>
<td>15.3317</td>
<td>6.0097</td>
<td>0.3920</td>
</tr>
<tr>
<td>301 $\times$ 301</td>
<td>30 $\times$ 30</td>
<td>2</td>
<td>24</td>
<td>11.0037</td>
<td>3.0247</td>
<td>0.2749</td>
</tr>
<tr>
<td>301 $\times$ 301</td>
<td>30 $\times$ 30</td>
<td>3</td>
<td>21</td>
<td>7.1452</td>
<td>1.9993</td>
<td>0.2798</td>
</tr>
<tr>
<td>201 $\times$ 201</td>
<td>20 $\times$ 20</td>
<td>1</td>
<td>28</td>
<td>15.3496</td>
<td>6.0152</td>
<td>0.3919</td>
</tr>
<tr>
<td>201 $\times$ 201</td>
<td>20 $\times$ 20</td>
<td>2</td>
<td>24</td>
<td>11.0148</td>
<td>3.0270</td>
<td>0.2748</td>
</tr>
<tr>
<td>201 $\times$ 201</td>
<td>20 $\times$ 20</td>
<td>3</td>
<td>21</td>
<td>7.3194</td>
<td>1.9994</td>
<td>0.2732</td>
</tr>
</tbody>
</table>
Unstructured finite elements libraries

- Object-Oriented, Fortran90
- 2D, 3D, scalar, vector valued

Interfaced with

- PETSc (Argonne Nat’l Lab) / MPI
  Parallel communication / matrix storage / solvers
- EXODUS (Sandia Nat’l Lab)
  I/O libraries, model representation
- Cubit (Sandia Nat’l Lab)
  2D/3D structured / unstructured mesh generator
- Ensight (CEI Int’l)
  Post-treatment, visualization
- METIS (George Karypis (UNM))
  Multilevel partitioning
Hybrid Algorithm

\[ \Omega : \text{Fine Space} \]

\[ \Omega_c : \text{Coarse Space} \]

Step 1: \( r^n \) → \( r^n \) → \( r^{n+1/3} \)

Step 2: \( r^{n+1/3} \) \( \rightarrow \) \( r^{n+1/3} \) \( \rightarrow \) \( r^{n+2/3} \)

Step 3: Solve Local Problems

Step 4: \( r^{n+2/3} \) → \( r^{n+2/3} \) → \( r^{n+1} \)

Step 5: \( r^{n+1} \) → \( r^{n+1} \)
Implementation Issues

- Restriction / Interpolation Operator
  - A PU function on each subdomain
  - Parallel Construction

- Coarse Problem: $A_c x_c = b_c$
  - $A_c = (PU)^T A (PU)$, $b_c = (PU)^T b$.
  - $A_c$ is Sparse Matrix
  - Easy to generate and to use PETSc Library
  - Assemble by neighbour using MPI
  - Each processor generates a row of the coarse matrix
Example (Coarse Solution)
Figure 7: 2,000,000 elements and 1,002,001 nodes
Figure 8: 2,000,000 elements and 1,002,001 nodes
Balancing Domain Decomposition Method can be extended to overlapping subdomains.

The coarse problem is easy and cheap to construct, and with a sparse stencil.