A preconditioning technique based on two-level domain decomposition methods

Duk-Soon Oh

Courant Institute of Mathematical Sciences
New York University

February 16, 2012
Introduction

- Conventional methods for large systems of algebraic equations arising from FDM and FEM
  - Direct methods - expensive to use
  - Iterative methods - may need many iteration counts due to a large condition number

- Domain decomposition methods
  - Provide good preconditioners
  - Combination of direct methods and iterative methods
  - Easy to parallelize
The Idea of Domain Decomposition Methods

- Decompose the domain $\Omega$ into overlapping or non-overlapping subdomains.
- Assign one or several subdomains to each processor of parallel machine.

In each iteration:

- In each subdomain, solve small local subproblems.
- In addition, solve one small global problem (two-level methods).
Motivation

One-level vs Two-level

- the number of subdomains may effect the efficiency for one-level methods.
- the performance of two-level methods only depends on the size of local subproblems.
- in some cases, e.g., solving nonlinear problems, we can reuse the coarse solver.
Motivation

Conventional two-level methods

- we usually need additional information, e.g., coarse coordinate information.
- we need quite regular meshes.
- it is hard to apply for irregular subdomains.
A preconditioning technique based on two-level domain decomposition methods

Alternative Approach

Generalized Dryja, Smith, Widlund (GDSW) coarse space technique

- this technique is based on energy minimizing discrete harmonic extensions.
- it has been applied to many applications
  - almost incompressible elasticity (Dohrmann, Widlund)
  - Reissner-Mindlin plates (Lee)
  - Raviart-Thomas vector fields (Oh)
Alternative Approach

Advantage

- the method can be implemented in an algebraic manner - we do not need any coarse discretization.
- it works well for irregular subdomains and unstructured meshes.
- it has well-established theoretical results, e.g., upper bounds of condition number.
A function $u^{(i)}$ defined on $\Omega_i$ is said to be discrete harmonic on $\Omega_i$ if

$$A^{(i)}_{II} u^{(i)} + A^{(i)}_{I\Gamma} u^{(i)} = 0.$$  

$u^{(i)}$ is completely defined by $u^{(i)}_{\Gamma}$. The discrete harmonic extension has the minimal energy property.

$$a(u, u) = \min_{v|_{\Gamma}=u_{\Gamma}} a(v, v)$$
Coarse Component

Interface($\Gamma$): Vertex + Edge
A preconditioning technique based on two-level domain decomposition methods

Coarse Component

- $R_0$: restriction to coarse space
  - We choose one coarse edge or vertex and give 1 to the nodes on the edge or vertex.
  - We assign 0 to other nodes on the interface.
  - We use the discrete harmonic extension for interior parts.

- $A_0 = R_0 A R_0^T$

We note that this coarse component can be implemented in an algebraic manner. We do not need any coarse discretizations.
Additive Schwarz Perconditioner

Additive Schwarz Method

\[ P^{-1} = R_0^T A_0^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i \]

- \( A_0 \): coarse matrix (restriction to the coarse space)
- \( A_i \): local matrix (restriction to extended subdomain \( \Omega_i' \))
- \( R_0 \): restriction to coarse space
- \( R_i \): restriction to extended subdomain \( \Omega_i' \)
A preconditioning technique based on two-level domain decomposition methods

**Restricted Additive Schwarz Perconditioner**

**Restricted Additive Schwarz Method**

\[ P^{-1} = R_0^T A_0^{-1} R_0 + \sum_{i=1}^{N} \tilde{R}_i^T A_i^{-1} R_i \]

- \( A_0 \): coarse matrix (restriction to the coarse space)
- \( A_i \): local matrix (restriction to extended subdomain \( \Omega'_i \))
- \( R_0 \): restriction to coarse space
- \( R_i \): restriction to extended subdomain \( \Omega'_i \)
- \( \tilde{R}_i \): restriction to subdomain \( \Omega_i \)
We implemented the algorithm with Trilinos Ifpack interface. Ifpack supports one-level (restricted) additive schwarz preconditioners. We only need an Epetra RowMatrix and an Epetra Map to construct the coarse component.
Conclusion

- We can construct an algebraic, parallel and scalable preconditioner with our new coarse space technique.
- We are applying this preconditioner to ice-sheet problems.
Numerical Experiments

1024 \times 1024 2D Laplace equation
1 subdomain per each processor, preconditioned GMRES
local solver : Amesos KLU

Table: total elapsed time in second

<table>
<thead>
<tr>
<th># of processors</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>one-level method</td>
<td>132.14</td>
<td>69.24</td>
<td>49.47</td>
<td>32.71</td>
<td>24.58</td>
</tr>
<tr>
<td>two-level method</td>
<td>175.71</td>
<td>85.39</td>
<td>44.38</td>
<td>23.17</td>
<td>14.39</td>
</tr>
</tbody>
</table>

Table: iteration counts

<table>
<thead>
<tr>
<th># of processors</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>one-level method</td>
<td>76</td>
<td>93</td>
<td>155</td>
<td>162</td>
<td>183</td>
</tr>
<tr>
<td>two-level method</td>
<td>48</td>
<td>62</td>
<td>76</td>
<td>64</td>
<td>67</td>
</tr>
</tbody>
</table>