Parallel Solution of Large Sparse Linear Systems, SS 2015 — Exercise sheet 4
Prof. Dr. Peter Bastian, Marian Piatkowski — Deadline 15th June 2015
IWR, Universität Heidelberg

EXERCISE 9 MULTIPLICATIVE SCHWARZ AS GAUSS-SEIDEL ITERATION

In the lecture you have learned the multiplicative Schwarz method applied to an overlapping decomposition of the domain Ω with p subdomains (c.f. Algorithm 3.2 in the script: `http://conan.iwr.uni-heidelberg.de/teaching/parsolve_ss2015/chapter03.pdf`).

In this exercise we are going to combine the multiplicative Schwarz method together with one dimensional subspace correction methods.

Let $V_h = \operatorname{span}\{\varphi_i^h, i \in I_h\}$ be the Finite Element space of the triangulation $\mathcal{T}_h$ of the domain Ω. For the subspaces, set $V_i = \operatorname{span}\{\varphi_i^h\}$ for $i \in I_h$. This decomposition has the property

$$V_h = \bigoplus_{i=1}^{N_h} V_i,$$

thus the index sets $I_i = \{i\}$ belonging to the V_i are now non-overlapping and the decomposition

$$u_h = \sum_{i=1}^{N_h} u_i \quad , \; u_i \in V_i$$

is unique.

1. Show that one iteration of the multiplicative Schwarz method in this case is equivalent to the Gauss-Seidel iteration

$$x^{(k+1)} = x^{(k)} + W^{-1}(b - Ax^{(k)})$$

where $W = D + L$.

2. Generalize the previously shown multiplicative formula of the Gauss-Seidel iteration to show that the multiplicative Schwarz method for an overlapping decomposition with p subdomains is equivalent to one block Gauss-Seidel step with overlapping index sets.

7 Points

EXERCISE 10 OPTIMAL UPPER BOUND FOR THE ADDITIVE SCHWARZ METHOD

Purpose of this excercise is to show a better bound for the largest eigenvalue for the additive overlapping Schwarz method than is given by Lemma 4.5 and Remark 4.6 in the lecture notes.

It takes the form

$$\langle \sum_{i=0}^{p} P_i x, x\rangle_A \leqslant (1 + n_c)\langle x, x\rangle_A \tag{1}$$

where n_c is the number of colors in the parallelized multiplicative Schwarz method introduced in Observation 3.4. In order to prove this result proceed as follows:

1. With the notation of Observation 3.4 define

$$\mathbb{P}_n = \sum_{i \in J_n} P_i, \qquad \text{i.e.} \qquad \sum_{i=1}^{p} P_i = \sum_{n=1}^{n_c} \sum_{i \in J_n} P_i = \sum_{n=1}^{n_c} \mathbb{P}_n.$$

Note that the coarse grid (index $i = 0$) is excluded.

2. Now show that the $\mathbb{P}_n$ are A-orthogonal projections, i.e. they satisfy the same properties as the P_i in Lemma 4.4.

3. Then estimate $\langle \sum_{n=1}^{n_c} \mathbb{P}_n x, x\rangle_A$ using the properties of $\mathbb{P}_n$ and the Cauchy-Schwarz inequality.

8 Points

Exercise 11 Parallel overlapping Schwarz methods

In this exercise we are going to experiment for the first time with an implementation of a parallel solver. We will implement and compare three variants of the overlapping Schwarz methods in total which work both in two and three dimensions.

As in the previous exercises, go to your `dune-parsolve` directory and type

```
$ git stash
$ git pull
$ git stash pop
```

to obtain the latest software version. The problem for the Schwarz methods is

$$\begin{aligned} -\Delta u(x) &= 0 && \text{in } \Omega = (0,1)^d, \\ u(x) &= \exp(-\|x\|_2^2) && \text{on } \partial\Omega. \end{aligned}$$

The file `additive_schwarz.cc` in the directory `uebungen/uebung04` provides a working implementation of the additive Schwarz method (ASM). Have a look at the functionality of the program, in particular

- `Dune::PDELab::OverlappingOperator`
- `Dune::PDELab::istl::ParallelHelper`
- `Dune::PDELab::OverlappingScalarProduct`
- `Dune::PDELab::SuperLUSubdomainSolver`

and do some test calculations. The sequential version of the program can be run with the command

```
./additive_schwarz
```

as you are used to. In order to use `<p>` processors of the computer, you simply have to add `mpirun -np <p>` before the run-command, i.e.

```
mpirun -np <p> ./additive_schwarz
```

In both cases the program creates an output file which can be visualized with ParaView. For the sequential case the file has the name `additive_schwarz.vtu`, for the parallel case the name has the structure `s00XX-additive_schwarz.pvtu` where `XX` is a placeholder for the number of processors used in the computation.

In the lecture you have also learned the multiplicative Schwarz method (MSM). In order to parallelize this algorithm, one has to color the subdomains. Figure 1 shows a collection of subdomains whose corrections can be calculated simultaneously. This method is called `coloring`. It is possible to extend this idea also to three dimensions.

This exercises only uses `Yasp-Grid` which numbers the subdomains in the parallel computation lexicographically. This ordering of the subdomains can be exploited in the coloring.

Task 1 The implementation of the class *MultiplicativeSchwarzPreconditioner* can be found in the file `multiplicative_schwarz_preconditioner.hh`. Currently, the sequential version of the preconditioner involving `p` processors is implemented. Your task is to supplement the coloring in the preconditioner.

Task 2 It is possible to modify this preconditioner to obtain a symmetric preconditioner. This symmetrization is referred to as the Symmetric Multiplicative Schwarz Method (SMSM).
Implement the symmetric preconditioner in the class *SymmetricMultiplicativeSchwarzPreconditioner*.

The main task of this exercise consists of comparing the different variants. Test the program for various mesh sizes, with overlap 1 or 2, in two and three dimensions with the number of processors $\in \{1, 4, 16, 64\}$.

Task 3 Present the number of iterations for various combinations in form of a table.

Example for two dimensions, overlap 1, number of processors equal to 1:

	ASM	Seq. MSM	Col. MSM	SMSM (col.)
mesh size	IT	IT	IT	IT
1/32				
1/64				
1/128				
1/256				
1/512				

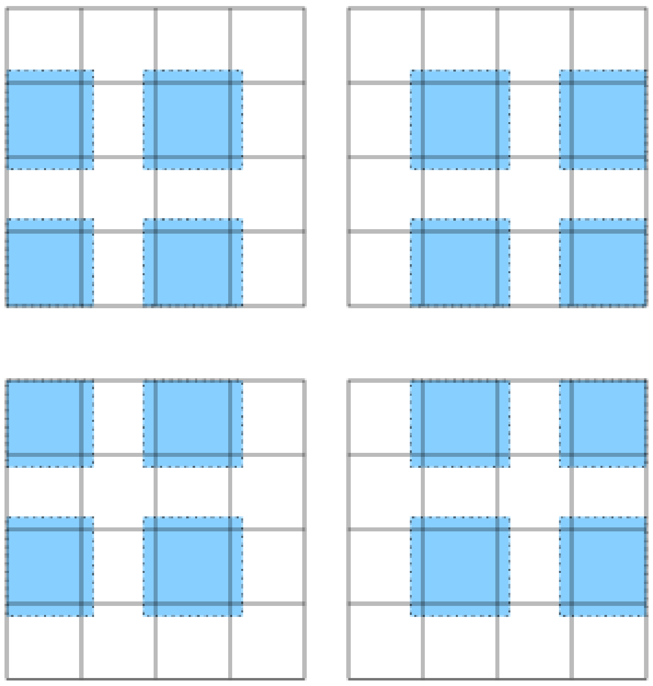

Abbildung 1: Simultaneously treated corrections in multiplicative Schwarz

15 Points