Prof. Dr. Peter Bastian, René Heß

IWR, Universität Heidelberg

Submission date: 02 December 2020

Exercise 1 *The parallel Richardson iteration*

We want to solve the linear system $Ax = b$ with the Richardson iteration

$$x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)}).$$

Let A be the stiffness matrix coming from the discretization of the Poisson equation on the unitsquare with P_1 Finite Elements. We use a structured simplicial mesh with $N = n^2$ degrees of freedom.

To accelerate the computational time we want to do the iteration in parallel with p processors. Therefore we subdivide the unitsquare into p smaller squares and the degrees of freedom are distributed accordingly. With this partitioning we assume that p is a square number such that every processor has $(n/\sqrt{p})^2$ degrees of freedom. The index set of the degrees of freedom is denoted by I, the index set of the i-th processor by I_i. Every processor stores the entries of $x^{(k)}$ corresponding to its degrees of freedom and relevant rows of A.

One iteration of the parallel Richardson iteration consists of the following steps:

- Communication of required entries of $x^{(k)}$ from the neighbouring processors.
- Calculation of $x^{(k+1)}$.

1. Describe the index sets I_i and specify which entries of $x^{(k)}$ the processor i has to communicate with which processor.
2. The computation time for an arbitrary arithmetic operation (addition, subtraction or multiplication) is denoted by t_{op}, the time needed for sending one byte to another processor by t_{byte} and the time needed to set up a message to another processor is denoted by t_{msg}.

   Derive a formula for the total computational time of one iteration with p processors. The entries of $x^{(k)}$ are stored in double precision such that every entry requires 8 byte of memory.

   The formula has to be only asymptotically correct. The matrix rows of the nodes next to the boundary which have less entries can be considered as interiour nodes.
3. The speedup of a parallel method is defined as the runtime of its sequential equivalent divided by the runtime of its parallel version. Present the speedup of the parallel iteration in a table using the following parameters:

$$t_{\text{op}} = 2 \text{ ns}$$
$$t_{\text{byte}} = 20 \text{ ns}$$
$$t_{\text{msg}} = 5000 \text{ ns}$$
$$n \in \{1024, 4096\}$$
$$p \in \{1, 4, 16, 256, 4096\}$$

(12 Points)

Exercise 2 *Domain decomposition*

In the lecture you have learned the following theorem:
Let $\Omega \subset \mathbb{R}^d$ be Lipschitz domain (open, bounded and connected) and let $f \in L^2(\Omega)$. Then the Poisson problem

$$\begin{aligned} -\Delta u(x) &= f(x) \qquad \forall x \in \Omega \\ u(x) &= 0 \qquad \forall x \in \partial\Omega \end{aligned} \tag{1}$$

is equivalent to the solution of the two subproblems

$$\begin{aligned} -\Delta u_1(x) &= f(x) && \forall x \in \Omega_1 \\ u_1(x) &= 0 && \forall x \in \partial\Omega_1 \setminus \Gamma \\ u_1(x) &= u_2(x) && \forall x \in \Gamma \\ \frac{\partial u_1(x)}{\partial n_1} &= -\frac{\partial u_2(x)}{\partial n_2} && \forall x \in \Gamma \\ -\Delta u_2(x) &= f(x) && \forall x \in \Omega_2 \\ u_2(x) &= 0 && \forall x \in \partial\Omega_2 \setminus \Gamma \end{aligned} \tag{2}$$

with the non-overlapping decomposition

$$\overline{\Omega} = \overline{\Omega_1 \cup \Omega_2}, \quad \Omega_1 \cap \Omega_2 = \varnothing, \quad \Gamma = \partial\Omega_1 \cap \partial\Omega_2, \quad \mu(\partial\Omega_i) > 0,$$

such that the $\partial\Omega_i$ are Lipschitz continuous.

The theory for the Poisson equation can be formulated for right-hand sides $f \in H^{-1}(\Omega)$ as well. But for the general assumption $f \in H^{-1}(\Omega)$, the equivalence (1) $\Leftrightarrow$ (2) does not hold. To see this we are going to consider the following counter example in one dimension for $\Omega = (-1, 1)$:

$$\begin{aligned} -\Delta u(x) &= -2\delta(x) \quad \text{in } \Omega \\ u(-1) &= u(1) = 0 \end{aligned}$$

where $\delta(x)$ denotes the *Dirac delta function*.

1. Split Ω at the origin. What are the transmission conditions of $u(x)$ on Γ? Compare them to the transmissions conditions given in (2).
2. Find the unique weak solution $u \in H^1(\Omega)$.
3. What changes in the derivation for the transmission conditions if we take again a right-hand side $f \in L^2(\Omega)$? **Hint:** Cauchy-Schwarz inequality.

(8 Points)

Exercise 3 *Parallel Computation of L^2-Norm using DUNE*

The code in the program `uebung03/uebung03.cc` computes the solution to a PDE and computes its L^2-Norm by numerical integration. Your task is to extend it to work correctly in parallel.

Parallelism in DUNE is achieved using the MPI communication framework. This provides a way to launch multiple instances of a program as well as providing means for communication of arbitrary data between the instances. The same abstract MPI interface works on your own computer as well as modern supercomputers.

You can run the program in parallel by calling

```
mpirun -np 4 ./uebung03
```

where you can specify an arbitrary number of processes after `-np`.

Have a look at the vtk output. How are subdomains arranged for different numbers of processes?

The two main points that require work are to make sure that each process only integrates over its own subdomain, excluding overlap, as well as making sure the results of all processes are summed up correctly.

Summing up a value over multiple processe requires communication. This can be done directly using MPI; it is however easier to make use of existing DUNE infrastructure which wraps MPI functionality in a way convenient for our purposes.

You can test your modifications by trying increasing overlap or increasing numbers of subdomains; this should expose any errors in your code. Conversely, setting overlap 0 and running only one process should always give you the correct value.

(4 Points)