Introduction to Supercomputing

TMA4280 · Project II

1 Problem description

This project deals with the implementation of a parallel solver for the Helmoltz and Unsteady Heat equation using the Finite Element Method with linear Lagrange elements.

The first part is dedicated to implementing data structures for the mesh, the discrete space, and the linear system, then test them in one dimension on a simple solution to the Poisson problem.

In a second part the Helmoltz equation in two dimension of space will be considered. Consider the solution u to the two-dimensional Helmoltz problem supplemented with Dirichlet and Neumann boundary conditions,

$$u(\boldsymbol{x}) - \Delta u(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in \Omega = (0,1)^2 \tag{1}$$

$$u(\boldsymbol{x}) = 0 \qquad \boldsymbol{x} \in \Gamma_D = \big\{\boldsymbol{x} \in \partial\Omega : y \in \{0,1\}\big\} \tag{2}$$

$$\frac{\partial u}{\partial \boldsymbol{n}}(\boldsymbol{x}) = 0 \qquad \boldsymbol{x} \in \Gamma_N = \big\{\boldsymbol{x} \in \partial\Omega : x \in \{0,1\}\big\} \tag{3}$$

with $f(\boldsymbol{x}) = (2\pi^2+1)\cos(\pi x)\sin(\pi y)$ a given right-hand side and $\boldsymbol{n}$ the outward normal to the domain boundary $\partial\Omega$.

One can verify formally that $\tilde{u}(\boldsymbol{x}) = \cos(\pi x)\sin(\pi y)$ is solution to the Helmoltz problem. The goal is to implement a solver for the given problem in two dimensions of space using linear Lagrange elements $\mathbb{P}_1$, implement a CG solver, then verify the convergence order of the method using the analytical solution.

In a third part this solver will modified to extended to a time-dependent problem. Consider the solution u to the two-dimensional Unsteady Heat problem with homogeneous Dirichlet boundary conditions,

$$\partial_t u(\boldsymbol{x},t) - \nu\Delta u(\boldsymbol{x},t) = f(\boldsymbol{x},t) \qquad (\boldsymbol{x},t) \in Q = (0,1)^2 \times [0,T) \tag{4}$$

$$u(\boldsymbol{x},t) = \tilde{u}(\boldsymbol{x},t) \qquad \boldsymbol{x} \in \partial\Omega, t \in [0,T) \tag{5}$$

$$u(\boldsymbol{x},0) = \tilde{u}(\boldsymbol{x},0) \tag{6}$$

with $\nu > 0$, $f(\boldsymbol{x}) = 0$, and $\tilde{u}(\boldsymbol{x},t) = e^{-\nu^2 t}\sin\big(\sqrt{\nu}\cos(\theta)\ x + \sqrt{\nu}\sin(\theta)\ y\big)$ is solution to the problem. Implement a time-marching scheme using Backward–Euler, then assess the convergence in time and in space by comparing the computed solution u_h to the analytical solution $\tilde{u}$ at $t = 1$ using the L^2-norm (or at minima the L^∞-norm).

1.1 Steps

1.1.1 Mesh Generation and 1D case

The mesh in $\mathbb{R}^d$ consists of two components:

1. A geometry consisting of the coordinates of mesh vertices.
2. A topology consisting of the connectivities between cells and vertices.

Additionally for any non-owned shared vertex (ghost), there should only be one owner and a mapping to a global number.

Question 1 (Unit Interval). Generate a mesh in parallel for the unit interval discretized with N cells. The geometry and the topology should be distributed like any normal array. Any shared vertex is owned by the process with the lowest rank. A data structure should be created to map non-owned vertices to their owner.

Question 2 (Vector). Implement a distributed vector class that provide the range information, a mapping to ghost entries.

Question 3 (Matrix). Implement a row-wise distributed sparse matrix class that provide the range information, and is split into a diagonal and off-diagonal block.

Question 4 (Assembly of the mass matrix). Implement the assembly of the Poisson problem in 1D and parallelize the loop with OpenMP. Test the assembled matrix on an analytical solution.

1.1.2 Extension to 2D

Question 5 (Unit Square). Generate a mesh in parallel for the unit square discretized with $N \times N$ cells with $N = 2^k$ using the structured approach with right-crossed triangles. The geometry and the topology should be distributed using the implementation of the previous section. For the sake of simplicity the partitioning should work on square blocks of cells of size given by a power of two. Any shared vertex can be owned by the process with the lowest rank.

Question 6. Implement the assembly of the Helmotz equation using the suggest Gauss quadrature rule on a triangle. Test the assembly in parallel without and with OpenMP. Test the assembled matrix on the suggested analytical solution by assembling and computing the residual $A\boldsymbol{x} = \boldsymbol{x}$.

1.1.3 Extension to time-dependent problems

Question 7. Modify the assembly of the stiffness matrix and the right-hand side to extend the implementation to the Unsteady Heat equation with a Backward-Euler discretization in time.

Question 8. Implement the Conjugate Gradient in parallel using the previously developped matrix and vector classes.

Question 9. Solve the suggested homogeneous Unsteady Heat equation with the provided initial and boundary values and perform a small convergence study.

2 General comments

The program should be organized, easy to understand as well as well parallelized and load balanced.

A report describing the results of parallelization of a scientific problem typically contains

- a description of the problem;
- a discussion of possible solution strategies;
- a brief explanation of the finished program;
- a description of the computer on which the numerical results were obtained, which compiler (with version) and compiler options you used and other relevant information, such as libraries used and their versions;
- numerical results (preferably plots)
- analysis (theoretical and experimental) of the performance of the algorithm and its implementation (time usage, speedup and efficiency as functions of problem size and number of processes or threads);
- a discussion of bottlenecks and possible improvements; and
- possibly a listing of relevant parts of the source code.