TDDC78 Lab 3

Anton Sterner
antst719@student.liu.se

Leo Nordling
leono184@student.liu.se

Linköping University

May 24, 2018

Chapter 1

Stationary Heat Conduction Using OpenMP

The code written in this lab is based on the example code written in Fortran. We chose to complete the lab in Fortran for simplicity.

1.1 Description of program

A matrix sized NxN is initialized with boundary values as described in the lab compendium. The matrix is divided veritcally into as many partitions as there are processes. To be able to efficiently parallelize the computations, two rows are added as padding in each iteration to each partition. One extra row for below the bottom row and one row for above the top. Each processor is then able to make calculations on its own partition without any chance of multiple processor trying to access the same memory. In order to parallelize the error computation, the program use a reduction function after every iteration that retrieves the highest error from all processes. The program stops when the error exceeds a certain threshold.

1.2 Execution times

We had trouble using the *Intel ifort* compiler, so the following measurements are made using the *gcc gfortran* compiler. Online sources suggest the Intel compiler is 4-5 times faster than using *gfortran*, but the scaling factors when increasing the number of processes should remain the same regardless of compiler choice. The program scales very well with increased parallelization, as can be seen in figure 1.1, with execution times cut almost in half each time the number of processes are doubled.

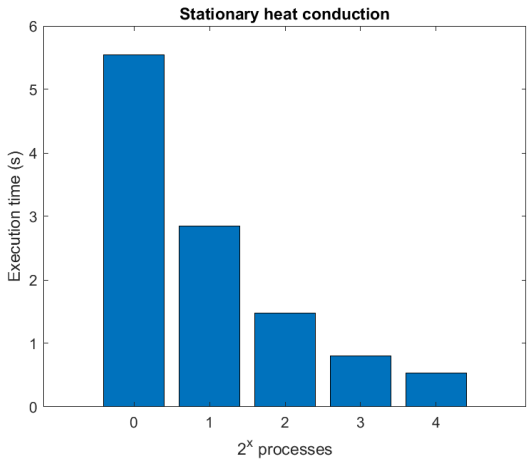

Figure 1.1: Execution times, 1-16 processes.