Prof. Dr. Kai Diethelm
Faculty of Applied Natural Sciences and Humanities
University of Applied Sciences Würzburg-Schweinfurt

Summer Semester 2024

Parallel Programming
Assignment 3

Deadline: May 07, 2024, 23:59 CEST

General Notes:

- For the programming assignments, you may use either of the programming languages C, C++, or Fortran. On the FANG HPC system, the corresponding compilers can be invoked by the commands `gcc`, `g++` and `gfortran`, respectively.
- In order to submit your solutions, follow the instructions given in the course webpage in our Elearning system.

Mathematical Background: The program `jacobi.c` which is available in the Elearning system provides a sequential algorithm for approximately solving the inhomogeneous Helmholtz equation

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = \alpha u(x,y) + f(x,y)$$

in two space dimensions with

$$f(x,y) = -\alpha(1-x^2)(1-y^2) - 2(1-x^2) - 2(1-y^2),$$

subject to the homogeneous boundary conditions

$$u(x,y) = 0 \qquad \text{for } |x| = 1 \text{ or } |y| = 1$$

on the square $[-1,1] \times [-1,1]$ and for computing the error of the approximation. To this end, it applies a finite difference method with n subintervals in x direction and m subintervals in y direction to discretize the differential equation. This leads to a linear equation system that is then approximately solved by a Jacobi SOR (successive over-relaxation) method with parameter ω and user specified values for the error tolerance and the maximum number of iterations as termination criteria.

(The differential equation has important applications, e.g., for the description of electromagnetic waves; the exact solution under the assumptions given here is $u(x,y) = (1-x^2)(1-y^2)$.)

Question:

Parallelize the given program `jacobi.c` with OpenMP. Provide detailed comments explaining your choice of the OpenMP constructs, including the handling of variables (shared vs. private). Provide a document that lists the run times and the errors output by (a) the sequential program, and (b) the parallelized program when running with 2, 4, 8, and 16 threads.

The timing data should be given for program runs with any combination of the following parameter values:

- $\alpha = 0.1, 1, 10$;
- $\omega = 1, 1.5, 1.9$;
- $m = n$ with $m = 10, 100, 1000, 10000$;
- error tolerance $= 10^{-6}, 10^{-9}$;
- maximum number of iterations $= 20, 200, 2000$.

(This can be created in an automatic manner by a carefully designed Slurm batch file.)

(350 Points)