HPC for mathematicians - Assignment 1

Yasmin Hengster

February 2021

Exercise 2

The exercise was to calculate a matrix product $\mathbf{A} \times \mathbf{B}$ where $\mathbf{A}, \mathbf{B}$ are $N \times N$ matrices. The elements for the matrices are given by:

$$\mathbf{A}_{i,j} = (N - j + i + 1)i$$
$$\mathbf{B}_{i,j} = (j + i)(N - j + 1)$$

where i is the corresponding row and j the column. As `C++` starts counting at 0, the formulas in the script have been adjusted.

The functions `MPI_Init`, `MPI_rank` and `MPI_size` were used to initialise the parallel computations.

In the first step the root process (rank= 0) computed matrix $\mathbf{B}$, which was broadcasted to all other processes (`MPI_Bcast`). Before continuing a barrier waited until each process received matrix $\mathbf{B}$.

In the second step each process (named $0, \ldots, N-1$) calculated the corresponding row of $\mathbf{A}$ and calculated the product of the row with the matrix $\mathbf{B}$. The result is stored in a N dimensional vector $\mathbf{c}$. These vectors are send by `MPI_Send` to process 0. Before the analysis in process 0 a barrier stopped again until all processes are finished. Using `MPI_Revc` all results were merged to the matrix $\mathbf{D}$.

The script is based with on a GNU compiler, the following modules have to be loaded:

- module load gcc
- module load mpt

The `compile_now_yasmin` file loads the required modules and compiles the `C++`-code. The job `run_ex2_yasmin.slurm` with n `tasks_per_node` computes matrix $\mathbf{D}$ for a given size n.

The result for $n = 3$ is

$$D = \begin{pmatrix} 75 & 68 & 43 \\ 204 & 184 & 116 \\ 387 & 348 & 219 \end{pmatrix}$$

which is saved in `matrix.out`

Exercise 3

In the following we want to find a numerical solution of the integral:

$$I = \int_0^b \int_0^a x\sin(x^2) + y\sin(y^2)\,dx\,dy = 0.5(-b\cos(b^2) + a + b)$$

For $a = b = 100$ the integral is $I = 195.216$. The analytic result is used to estimate an error of the numerical methods. To solve the integral I used the trapezoidal rule:

$$\begin{aligned} I =& \frac{\Delta x \Delta y}{4} \left(f(x_0, y_0) + 2f(x_0, y_1) + \cdots + 2f(x_0, y_{n-1}) + f(x_0, y_n)\right. \\ &+ 2\left(f(x_1, y_0) + 2f(x_1, y_1) + \cdots + 2f(x_1, y_{n-1}) + f(x_1, y_n)\right) \\ &+ \ldots \\ &+ 2\left(f(x_{n-1}, y_0) + 2f(x_{n-1}, y_1) + \cdots + 2f(x_{n-1}, y_{n-1}) + f(x_{n-1}, y_n)\right) \\ &+ f(x_n, y_0) + 2f(x_n, y_1) + \cdots + 2f(x_n, y_{n-1}) + f(x_n, y_n)\left.\right) \end{aligned}$$

where Δx and Δy are the length of each trapezoidal ($\Delta x = \frac{a}{n}$) and n of trapezoids. The values x_i and y_i are given by:

$$\begin{aligned} x_i &= 0 + i\Delta x \\ y_i &+ 0 + i\Delta y \end{aligned}$$

Figure 1 shows the convergence of the numerical computed integral for a increasing number of trapezoids. The accuracy of three digits isn't achieved for the tested range of n.

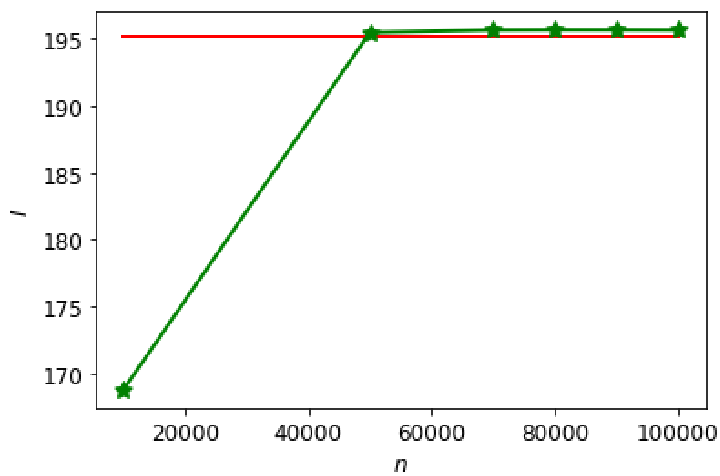

Figure 1: Convergence of the integral for an increasing number of trapezoids. The red line marks the real value, the green stars are the computed values.