Homework 7
ME 5773 High Performance Computing Due March 31, 2024 by 11:59pm

MPI assignment:

This is a team assignment. The teams are assigned by the instructors. The team members are posted on Canvas. The deliverable for this assignment is an email to the instructors notifying the completion of the assignment. The activities of this assignment should be stored in the Team's github repository, within a folder named "assignment7". This assignment will be evaluated using the most updated version of the repository **by the due date**.

Gauss-Legendre integration

Gauss-Legendre integration is a numerical method for computing integrals using Gauss-Legendre quadrature. The quadrature defines locations, x_i, and weights, w_i, that can be used to approximate the integral, e.g.,

$$I = \int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

For example, using 3-point quadrature

$$I = \int_{-1}^{1} f(x)dx \approx (5/9)f(-\sqrt{3/5}) + (8/9)f(0) + (5/9)f(+\sqrt{3/5})$$

For $f(x) = xe^x$, the result is $I \approx 0.735362144160854$, compared to the exact result of $\frac{2}{e} = 0.735758882342885$. An improved result can be obtained using a higher order quadrature, e.g., more integration points. In addition, a transformation can be used to alter the integration limits to arbitrary real values a and b instead of -1 and 1.

Assignment:

Write an MPI code with Python that computes the integral of $f(x) = xe^x$ using quadrature values of $n = 1, 2, \ldots, 20$. That is, 20 evaluations of the integral will be obtained, each with a different quadrature. Each integral is to be computed by a different worker process.

The **Master** computes the quadrature (x and w values), passes the quadrature to a worker process, collects the results from each process, then constructs and prints a table with the results of the form:

Quadrature no.	Integration Result	Percent error	Run time (s)
1			
2			
3			
…			
20			

The Python function `gauleg.py` can be called to obtain the quadrature for any number of points and for arbitrary integration limits. This code can be found in `/work/ME5773/sec6/Homework`.

The **worker** processes obtain the quadrature, evaluate the integral, and send the results back to the master process.

Part 1: write the file that is able to execute the MPI program, and run the program with 21 tasks (1 master, and 20 workers). Save the outputs of run in a text file named '`part1.txt`'

Part 2 (15% extra credit): On a different python file, modify the program for the case where the number of tasks is < 20 but still completes the table for up to 20 integration points. Write the outputs of this run in '`part2.txt`'.