Parallel Programming

A note on compiling

Before make all, run the following (this definitely works on Adroit, I haven't run on Nobel):
module load openmpi/intel-16.0 intel/16.0
I compiled with the Intel compiler because it has been recommended in class and in mini courses I have attended, but I couldn't get the modules to load correctly within my makefile so it needs to be done separately.

OpenMP

The speedup is close to the number of cores for 2 and 4 threads, but the improvement starts to tail off after 8 threads for smaller grids.

Number of threads	Time for 128 x 128	Time for 256 x 256	Time for 512 x 512
1	34.0 s	533.0 s	
2	17.1 s (1.99x speedup)	271.3 s (1.96x speedup)	4256.0 s
4	9.0 s (3.8x speedup)	164.2 s (3.2x speedup)	2146.0 s
8	5.0 s (6.8x speedup)	72.0 s (7.4x speedup)	1367.0 s
16	4.4 s (7.7x speedup)	38.8 s (13.7x speedup)	740.0 s

MPI

The speedup for MPI is comparable to that for open MP

Number of Processors	Time for 128 x 128	Time for 256 x 256	Time for 512 x 512
1	34.4 s	539.0 s	
2	17.6 s (1.95 x speedup)	273.3 s (1.97 x)	
4	9.2 s (3.7 x speedup)	140.3 s (3.8 x)	2153.2 s
8	5.6 s (6.1 x speedup)	77.1 s (7 x)	1151.9 s
16	3.0 s (11.5 x speedup)	39.9 s (13.5 x)	536.9 s

MPI IO

For my file IO in MPI I chose to write a separate file for each process, and to combine the datafiles in my Python plotting routine. The reason I did this was mainly for debugging purposes (so that I could check that my end columns had been passed correctly), and because I think it could be more easily adapted to writing to a single file from multiple processes using MPI IO. I chose not to send the different chunks of the domain to one process to write one file because that seemed to be a bad solution in terms of efficiency. If I had had more time, I would have written to a single file from

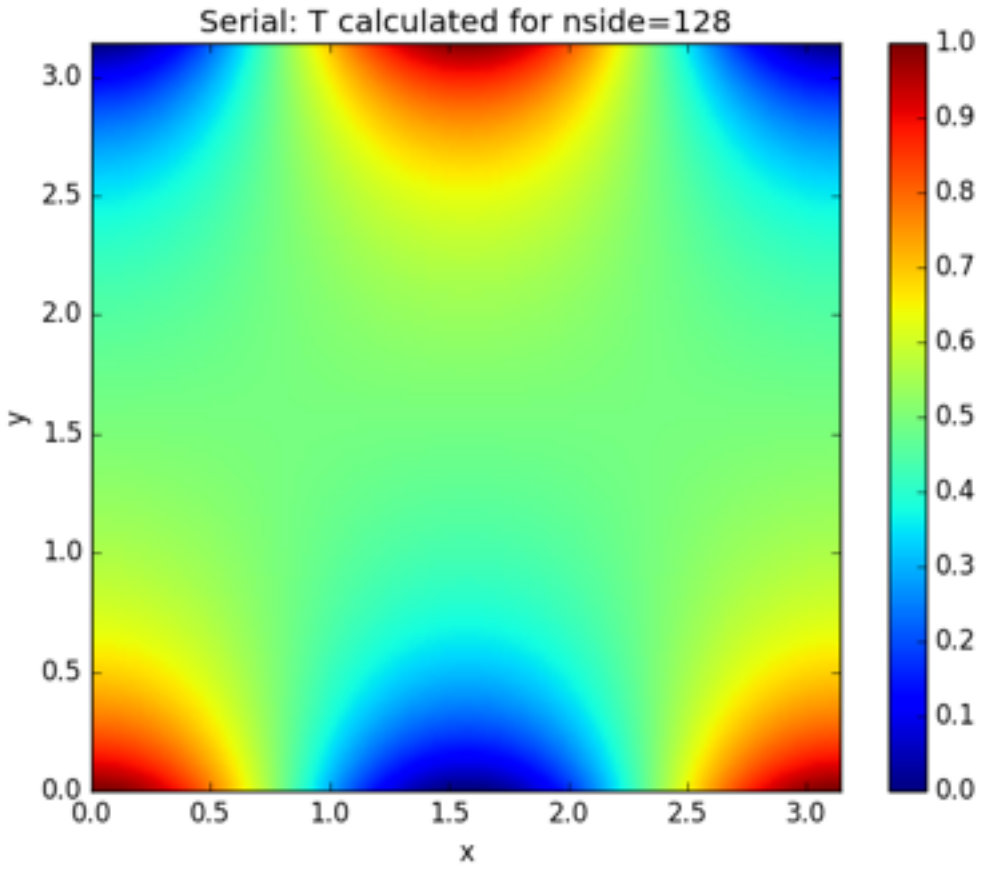
multiple processes using MPI IO, which seems like the best solution because each process has a fixed size chunk that should be written to a separate portion of the datafile.

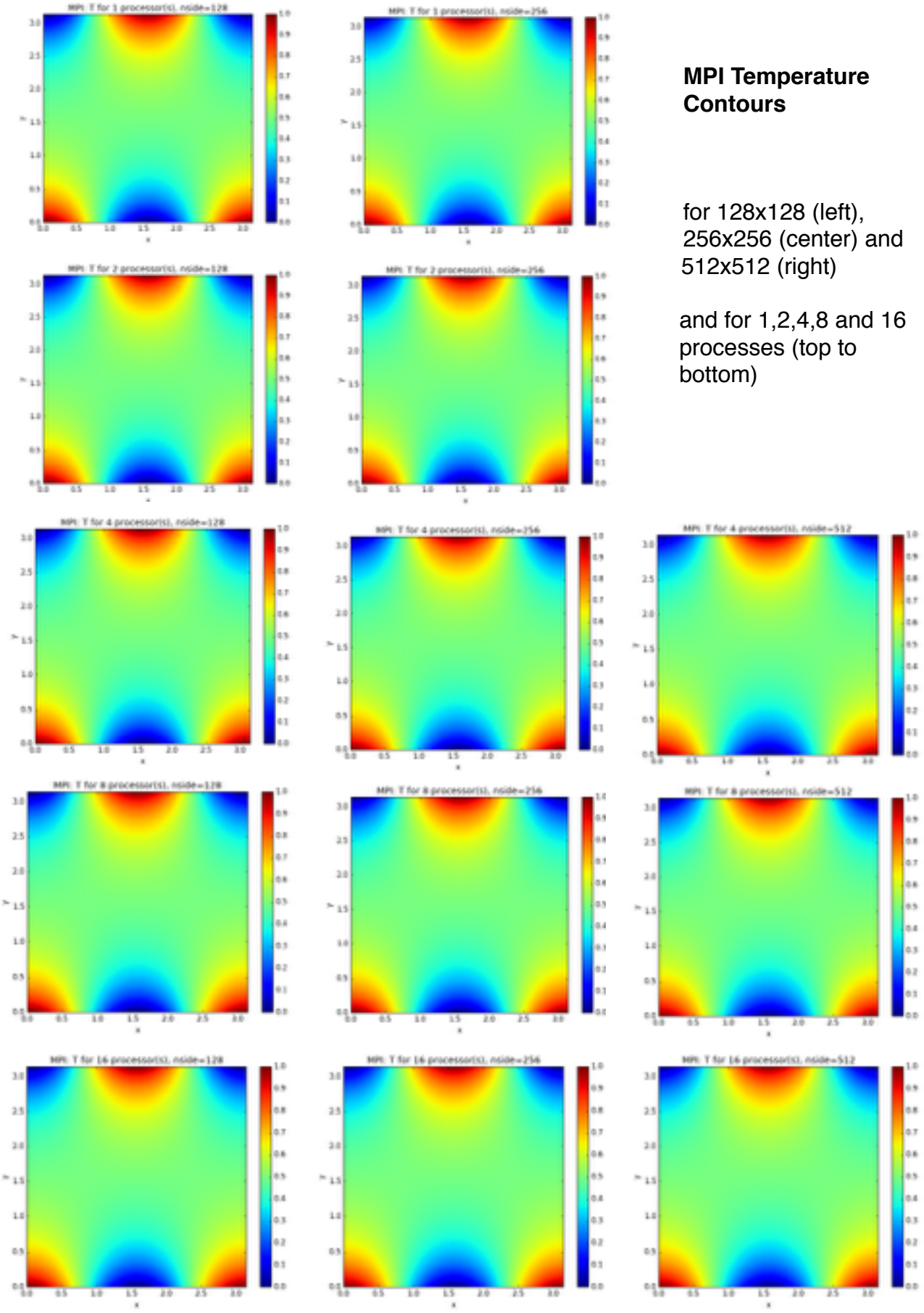
OpenMP vs MPI
Parallelizing with OpenMP was much easier than with MPI because there was no need for communication between processes and IO was easier. However MPI can scale to large numbers of processes, while OpenMP is limited by the number of threads a single processor can handle.

Mean Temperature and Temperature Contours
The mean temperature has a slight dependence on the grid size due to numerical differences, but doesn't change significantly.

Pixels	Average temperature
128 x 128	0.4971081155
256 x 256	0.4970966523
512 x 512	0.4970909392

Serial temperature contours

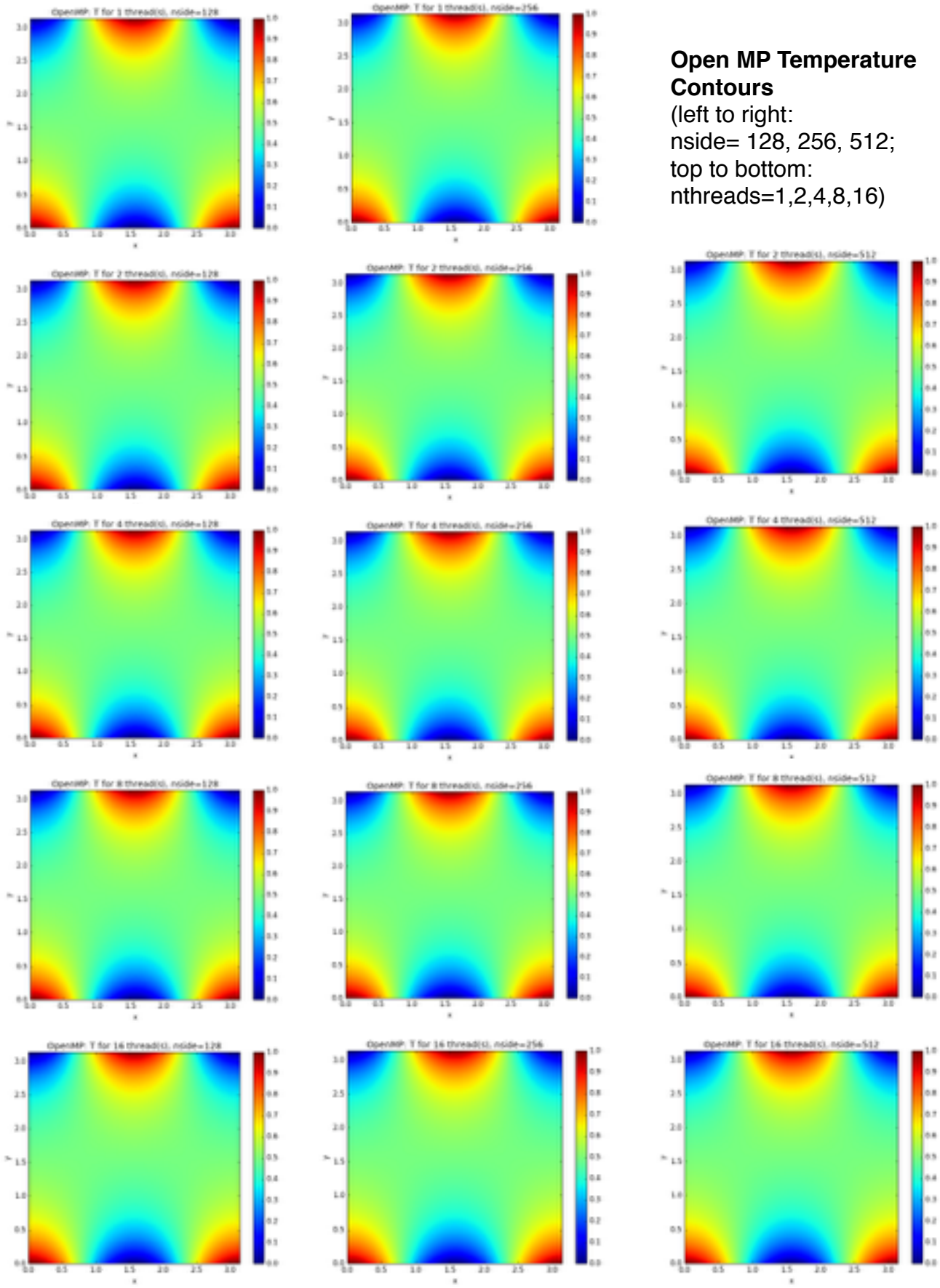




MPI Temperature Contours

for 128x128 (left), 256x256 (center) and 512x512 (right)

and for 1,2,4,8 and 16 processes (top to bottom)



Open MP Temperature Contours
(left to right:
nside= 128, 256, 512;
top to bottom:
nthreads=1,2,4,8,16)