

MPI: Practical 5 Solutions

1 Mandelbrot set

1.1 Mandelbrot in parallel

See `mandelbrot_load.cpp`.

Note that, on rank 0, we need an array that is large enough to fit the colour data for the whole of the domain (i.e. `res * res` cells). The issue is that when resolution is large, there may not be enough space on the stack to fit this array, so it should be dynamically allocated:

```
int* array = new unsigned char* lines[3 * res * res];
```

This space should eventually be freed using `delete[] lines;` to avoid memory leaks.

The gathering of results from each process can be done using a single collective operation, `MPI_Gatherv`, if offsets and counts are set appropriately

1.2 Load balancing

See `mandelbrot_load.cpp`.

This can be done by assigning rows to processes in alternating order. But we can't have a single gather operation now to collect the results; we need multiple `MPI_Gather()`s.

1.3 Master-slave parallelisation

See `mandelbrot_ms.cpp`.

This can be done by having the master sending messages to each slave indicating which row number to work on. When a slave finishes a row, it will send back the resulting data, and await further instruction. A row number of -1 will instruct the slave to exit. When sending results back to the master, the slave uses the tag value to indicate which row it has calculated.

The C++ solution uses `tellp` and `seekp` to write to the correct location of the output file as the simulation is run, but it is also possible to store results into a bigger array in rank 0, and output everything at the end (as we did before).

The timings for the master-slave approach are better than those for the approach that divides up the domain equally between the processors. This follows because the computational effort required to determine whether a point is in the Mandelbrot set varies enormously over the domain. Better load balancing (and better performance) results if work is divided more-or-less equally between processors. This happens in the master-slave approach because each process asks for more work when it finishes the previous batch.

1.4 Area calculation

See `mandelbrot_area.cpp`.