MSc in High Performance Computing
Coursework for Threaded Programming Part 1

The object of this assessment is to parallelise a simple piece of code using OpenMP. This assessment in worth 10% of the total marks for the course.

You are provided with a piece of code which you should parallelise with OpenMP directives.

The code can be found on the course pages on Learn. You may choose to work with *either* the C (`C/adi.c`) *or* the Fortran (`Fortran90/adi.f90`) version.

You should use the Intel 19 compilers to build the code. On Cirrus, type

```
module load intel-compilers-19
```

before building the code. There is a Makefile supplied that shows how the marker will build your code for correctness testing.

Parallelisation

Add OpenMP directives to parallelise all the computation between the calls to the timer routine `omp_get_wtime()`. Your are *not* expected to otherwise optimise the code, or modify it except for the addition of OpenMP. Your solution will be marked for correctness, by running it on up to 36 threads. The `diff` value printed should be identical to that of the sequential code for all thread numbers. You will be given credit for keeping the changes made to the original C or Fortran code to a minimum. Your code will *not* be marked directly for performance, but you will be given additional credit for minimising the number of implicit barriers that are executed, and you will be penalised if the performance is very poor (i.e. worse than the sequential code).

Submission

You are required to submit your source code as a single file. Details of the submission process will be provided separately. The deadline for submission is 16:00 on Friday 16th October 2020. The standard University late penalties will be applied: 5% of the maximum available mark will be deducted for every calendar day or part thereof it is late, up to a maximum of 7 calendar days. After this time a mark of zero will be recorded.