Final Project due date: December 7

Due date: Wednesday, December 7 at midnight. You may work with up to one other person on the entirety of the assignment and turn in a joint program if you wish; in this case, each of you gets the same grade.

Introduction:

In this homework, you will parallelize a serial C, C++ or a Fortran code using MPI. The provided serial code solves the fully nonlinear shallow water equations including earth rotation. The code simulates the nonlinear interaction of a twin tropical cyclones placed in the northern and the southern hemispheres. After parallelizing the code, you will execute it on 10 processes to compute the numerical solutions of the elevation h and the wind vector field in the shallow equations at a specified time (see below) and use the provided "ncl" script (see below) to generate a 2D vector-contour plot with your numerical solutions. The files containing the procedures and the graphic script that you will use as a starting point are combined in "tar" files cyclones_for_ser.tar (Fortran), cyclones_c_ser.tar (C) and cyclones_cc_ser.tar (C++). The initial conditions required for the program are provided in the tar file input.tar. You might practice with Fortran, C or C++ programming languages if you wish, but please select only one for the submission of your assignment.

Serial codes that solve the linear shallow water equations and the corresponding MPI parallel codes are provided in the tar files lin_for_ser.tar (lin_for_mpi.tar) for fortran, lin_c_ser.tar (lin_c_mpi.tar) for C, and lin_cc_ser.tar (lin_cc_mpi.tar) for C++. You can compare these codes and use them as a guide to parallelize the fully nonlinear shallow water equations code.

If you are using a computer with unix or linux shell, you can transfer these tar files to Agave by using the following commands:

```
sftp username@agave.asu.edu
sftp> cd APM512
(or the name of the working directory you created for the class)
sftp> put cyclones_for_ser.tar  (or cyclones_c_ser.tar or cyclones_cc_ser.tar)
sftp> put input.tar
sftp> put lin_for_ser.tar  (or lin_c_ser.tar or lin_cc_ser.tar)
sftp> put lin_for_mpi.tar (or lin_c_mpi.tar or lin_cc_mpi.tar)
sftp> quit
```

Login to Agave and extract the files, using the commands:

```
ssh username@agave.asu.edu
$ cd APM512 (or your working directory)
$ tar -xvf cyclones_for_ser.tar  (or cyclones_c_ser.tar or cyclones_cc_ser.tar)
$ tar -xvf input.tar      (or input.tar)
$ tar -xvf lin_for_ser.tar  (or lin_c_ser.tar or lin_cc_ser.tar)
$ tar -xvf lin_for_mpi.tar  (or lin_c_mpi.tar or lin_cc_mpi.tar)
```

The following folder should be created:
CYCLONES_FOR_SER: contains module_precision.f, module_input_output.f, module_init.f, module_adv_diff_cor.f, module_flux.f, module_tendency.f, main.f, and plot_huv.ncl

CYCLONES_CC_SER: contains input_output.cc (.h), init.cc (.h), flux.cc (.h), tendency.cc (.h), adv_diff_cor.cc (.h), main.cc, and plot_huv.ncl

CYCLONES_C_SER: contains input_output.c (.h), init.c (.h), flux.c (.h), tendency.c (.h), adv_diff_cor.c (.h), main.c, and plot_huv.ncl

INPUT: contains shallow_init_h.bin, shallow_init_u.bin and shallow_init_v.bin

LIN_FOR_SER: contains module_precision.f, module_input_output.f, module_flux.f, module_tendency.f, main.f and plot_h.ncl

LIN_FOR_MPI: contains module_precision.f, module_mpi.f, module_input_output.f, module_flux.f, module_tendency.f, main.f and plot_h.ncl

LIN_CC_SER: contains input_output.cc (.h), flux.cc (.h), tendency.cc (.h) main.cc and plot_h.ncl

LIN_CC_MPI: contains mpi_fun.cc (.h), input_output.cc (.h), flux.cc (.h), tendency.cc (.h) main.cc and plot_h.ncl

LIN_C_SER: contains input_output.c (.h), flux.c (.h), tendency.c (.h) main.c and plot_h.ncl

LIN_C_MPI: contains mpi_fun.c (.h), input_output.c (.h), flux.c (.h), tendency.c (.h) main.c and plot_h.ncl

To compile the code, request 10 compute nodes:

```
$ interactive -n 10
```

- **Compilation of the Linear code:**

  $ module load intel-mpi/2018x

  $ ifort -O3 -FR -o main module_precision.f module_input_output.f module_flux.f module_tendency.f main.f (Fortran serial)

  $ mpiifort -O3 -FR -o main module_precision.f module_mpi.f module_input_output.f module_flux.f module_tendency.f main.f (Fortran MPI)

  $ icpc -O3 -o main input_output.cc flux.cc tendency.cc main.cc (C++ serial)

  $ mpiicpc -O3 -o main mpi_fun.cc input_output.cc flux.cc tendency.cc main.cc (C++ MPI)

  $ icc -O3 -o main input_output.c flux.c tendency.c main.c (C serial)

  $ mpiicc -O3 -o main mpi_fun.c input_output.c flux.c tendency.c main.c (C MPI)

- **Compilation of the nonlinear code:**

  $ module load intel-mpi/2018x

  $ ifort -O3 -FR -o main module_precision.f module_input_output.f module_init.f module_adv_diff_cor.f module_flux.f module_tendency.f main.f (Fortran serial)

  $ icc -O3 -o main input_output.c init.c flux.c tendency.c adv_diff_cor.c main.c (C serial)

  $ icpc -O3 -o main input_output.cc init.cc flux.cc tendency.cc adv_diff_cor.cc main.cc (C++ serial)

Note: For the nonlinear case, the files shallow_init_h.bin, shallow_init_u.bin and shallow_init_v.bin must be placed in the folder where you are runing the code. The code as it stands WILL compile and run serially.
For parallel compilation and execution, you will need to modify the program files following a similar strategy as in the provided codes for the linear case.

After implementing the modifications, an executable file called **"main"** should be created. For parallel execution, type the following commands:

$ srun - -mpi=pmi2 -n 10 ./main (for 10 processes)

Assignment:

The nonlinear shallow water equations with rotation are described in detail in **lecture_18.pdf**

The provided code integrates the shallow water equations in the domain defined by

$$-\frac{L}{2} \le x < \frac{L}{2}; \quad -\frac{L}{2} \le y < \frac{L}{2}$$

where $L = 75,000km$.

The domain uses a staggered grid with combined absorbing and periodic boundary conditions

$$\psi(x+L, y, t) = \psi(x, t) \;\; \text{and} \;\; \psi(x, y+L, t) = \psi(x, y, t)$$

where ψ represents h, u or v. The code uses a total number of $N_x \times N_y$ grid points,

where $N_x = N_y = M = 600$, with a grid spacing of $\Delta x = \Delta y = 25km$

The time integration combines both the third-order Runge-Kutta (RK3) and a sem-implicit time stepping schemes; and the space derivatives are approximated by fourth-order finite differences.

The numerical solver integrates the equations from $t = 0$ to $t = n_t \Delta t$, where Δt is time step and $n_t = 5001$ is the total number of steps. The time step is determined from $\Delta t = r\Delta x/(c\sqrt{2})$, where $r = 0.4$ is the Courant number.

Algorithm for the solver:

- Begin the time step loop for $it = 1, 2, \cdots, n_t$

  1. Begin the RK3 substep loop for $ir = 1, 2, 3$
     (a) RK3 substep1: $ir = 1 \implies \Delta\tau = \Delta t/3, \; s = n, \quad s+1 = *$
     (b) RK3 substep2: $ir = 2 \implies \Delta\tau = \Delta t/2, \; s = *, \quad s+1 = **$
     (c) RK3 substep3: $ir = 3 \implies \Delta\tau = \Delta t, \; s = **, \quad s+1 = n+1$
     (d) Compute the tendency terms $F^s_{i,q}$, $G^s_{p,j}$ and $R^s_{p,q}$
     (e) Advance u, v and h for the substep $s+1$:
     $$u^{s+1}_{i,q} = \left[(1-\sigma\Delta\tau/2)u^n_{i,q} - \Delta\tau F^s_{i,q}\right]/(1+\sigma\Delta\tau/2)$$
     $$v^{s+1}_{p,j} = \left[(1-\sigma\Delta\tau/2)v^n_{p,j} - \Delta\tau G^s_{p,j}\right]/(1+\sigma\Delta\tau/2)$$
     $$h^{s+1}_{p,q} = \left[(1-\sigma\Delta\tau/2)h^n_{p,q} - \Delta\tau G^s_{p,q}\right]/(1+\sigma\Delta\tau/2)$$
     (f) Repeat steps a, b, c, d and e for the next substep $ir = ir + 1$

2. End the RK3 substep loop
3. Overwrite u^n, v^n and v^n:
$$u^n_{i,q} = u^{n+1}_{i,q}$$
$$v^n_{p,j} = v^{n+1}_{p,j}$$
$$h^n_{p,q} = h^{n+1}_{p,q}$$
4. Repeat steps 1-5 for the next time step $it = it + 1$

- End the time step loop

1) Parallelize the serial code provided (cyclones_for_ser.tar, cyclones_c_ser.tar or cyclones_cc_ser.tar). This code integrates the nonlinear shallow water equations described in lecture_18.pdf. It simulates the nonlinear beta-drift of a twin tropical cyclones straddling the equator. All the program files need modifications. Use the serial and parallel codes provided for the linear case as a guide to parallelize your procedures. Solve the equations using the parameters $(c, H, g, L, \cdots)$ described in lecture_18.pdf for a total number of time steps $n_t = 5001$. Binary files named "shallow_h.bin", "shallow_u.bin" and "shallow_v.bin", should be produced after successful execution of ./main. These files archive two dimensional fields of the simulated elevation, the eastward and the northward velocities for each 200 time steps. Your plot should include the archived field at $n_t = 5001$.

2) Use the NCAR Command Language (NCL) (https://www.ncl.ucar.edu) graphic software, which is available in AGAVE, to generate a contour-vector plot for the two dimensional field of elevation and the wind vector simulated at $n_t = 5001$. Use the following ranges for the axes: x-axis [-25km, 25km] and y-axis [-25km, 25km].
A graphic script called plot_huv.ncl is provided for you. To run the script use the following command lines:

- Generating the plot using NCL:

  ```
  $ module load ncl/6.3.0
  ```

  ```
  $ ncl plot.ncl
  ```

  A pdf file called cyclones.pdf should be generated

Grading rubric:

- Upload the following files to Canvas for grading:
  - A tar file finalproject.tar containing your program files.

  - A pdf plot showing the elevation and the wind fields fields simulated with 10 processes at $n_t = 5001$ in a pdf format. Please don't upload the binary files to canvas.

- This assignment is worth 20 points, as follows:
  - (2 points) The program structure contains the files and the functions mentioned above with the appropriate naming
  - (5 points) Your program compiles without warnings or errors with the flag options specified above.
  - (10 points) When executed, your program produces accurate approximation for h, u and v; and speeds up by a factor of at least 8 on 10 processes.
  - (3 points) The program is adequately (but not excessively) commented.