HPSC Lab3

Luna McBride, Mehmood Ali

September 21 2023

1 Parallel Design

Now that we have a grid, it is time to get some random particles moving across it. Real life is not predictable, so trusting everything to go along the mesh perfectly is simply a fever dream. That is why it is time to add checks for diagonal movements between nodes and track where the particles decide to fly.

1.1 Purpose of the Lab and Requirement

The purpose of lab3 is to simulate the behavior of particles in a two-dimensional space over a specified time interval. It models the motion of particles under the influence of a given flux density and boundary conditions. The simulation uses a grid-based approach to interact with particles, allowing for the calculation of forces on particles and the accumulation of particles on grid nodes.

- **Parallel Computing:** We have to fill the to-dos in code and it is designed to run on multiple processors in a parallel computing environment using MPI.
- **Grid Decomposition:** The domain is divided into a grid of processors, and each processor is responsible for a portion of the grid.
- **Particle Tracking:** We have to tracks the motion of particles over time, including their positions and velocities. Particles can move within the domain, and the code accounts for particles leaving one processor's domain and entering another.
- **Interaction with Mesh:** Particles interact with a grid mesh, and the code quantifies this interaction. It computes forces on particles based on their positions within the mesh.
- **Particles Movement Visualization:** At the end we have to visualize the particle moving from one processor to its neighboring processor.

Below is the description of how parallel code works to make particles move:

1.2 MPI Initialization and Setup (mpiInfo Class)

The code starts by initializing MPI (Message Passing Interface), which is a library for parallel computing. It determines the number of processors (`numPE`) and assigns a unique identifier to each processor (`myPE`).

1.3 Grid Decomposition

The grid decomposition is a fundamental aspect of the parallel design. The domain is divided into smaller subdomains, with each subdomain assigned to a different processor.

- The domain is divided into a grid of processors, and the number of processors in the x and y directions is specified by `nPEx` and `nPEy`.
- The `GridDecomposition` method of the `mpiInfo` class ensures that the total number of processors (`numPE`) matches the product of `nPEx` and `nPEy`. If not, it raises an error and terminates the program.
- Each processor is assigned an (iPE, jPE) index, representing its position in the processor grid. For example, (iPE, jPE) $(0, 0)$ corresponds to the processor in the bottom-left corner of the grid.
- Neighboring processor information is computed for each processor. This information is used for inter-processor communication.

1.4 Mesh Initialization (`Mesh` Class)

The code initializes a mesh that represents the grid over which particles will move. The mesh is divided into cells, and its properties are defined as follows:

- The mesh is created within the specified domain boundaries $(x0, x1, y0, y1)$ and divided into cells in both the x and y directions (`ncell_x` and `ncell_y`).
- The mesh properties, such as the number of real nodes (`nRealx`, `nRealy`), and the number of field variables (`nField`), are computed based on the mesh size.
- The mesh coordinates (`x` and `y`) are stored in arrays for all nodes, including ghost nodes.

1.5 Particle Initialization (`Particles` Class)

Particles are initialized and injected into the simulation. Key particle-related parameters include:

- `PTCL` is an instance of the `Particles` class that manages particle properties and behaviors.
- Particles are injected into the simulation domain, and their positions, velocities, and active states are tracked.
- The number of particles, density, and boundary velocities are specified to initialize the particles.

1.6 Time Marching Loop

The simulation progresses through time with a time-stepping loop. At each time step, several key operations are performed:

- Particle Movement: Particles move according to their velocities and forces. This is done in the `move` method of the `Particles` class.
- Particle-Mesh Interaction: Particles interact with the mesh in the `ParticlesOnMesh` method of the `Mesh` class. Forces on particles are computed based on their positions within the mesh.
- Visualization: The code generates visualizations of particle trajectories and mesh properties at specified time intervals. To visualize the simulation's progress.

1.7 Inter-Processor Communication

Particles that leave one processor's domain and enter another are exchanged among processors using MPI. The `ParticleExchange` method in the `mpiInfo` class manages this communication.

2 Self Evaluation

2.1 Luna

I honestly tried very hard to get it to work, but could not. My time was very limited between work and school. I also could not do anything over the weekend because trying to login to ondemand on Friday while that original site was down completely locked down my CU account, not just for ondemand or research computing. I could not even access Canvas or Buff Portal over the weekend.

This being considered, however, a lot of my time went toward debugging what I could. I would tweak something here, see how it went. I added print statements to each major part of mpiInfo to see where it was stopping. I also tried to learn Valgrind, as looking at the errors it was giving, it would stop around time 0.55 with dt = 0.01 with a different memory/malloc error each time, even if I had not even changed the code. I could not fit office hours in with my work schedule and the Valgrind document did not point to any lines of our code, so that is where I was stuck.

2.2 Mehmood

In the beginning, I encountered some significant challenges when starting my project. I encountered various errors, including issues like "bad signal termination," "core dump," and problems related to MPI (Message Passing Interface). To overcome these hurdles, I employed a debugging strategy, primarily using multiple print statements to identify and address the issues.

One notable problem I faced was incorrectly assigning neighbor values. Initially, when I managed to run the project, I encountered a frustrating situation where only a single line of particles was visible, and they appeared to be immobile. This problem, as it turned out, was caused by an error in how I assigned neighbor values. Also I was not getting any output because I forgot to pass the **tEnd** and **dt** values in slurm file. After dedicating a substantial amount of time to debugging and resolving these issues, I was ultimately successful in getting the project to work as intended.

3 Appendix A: Parallel Code

```
//  ==
//  ||
//  ||   C L A S S:    m p i I n f o
//  ||
//  ==

class mpiInfo
{
 public:

  int myPE;
  int numPE;
  int nRealx,  nRealy;
  int nx,  ny;
  int nPEx, nPEy;
  int iPE , jPE;
  int iMin, iMax, jMin, jMax ; // The global i-j numbers on this processor
  int nei_n, nei_s, nei_e, nei_w;
  int nei_ne, nei_nw, nei_se, nei_sw;

  int countx, county;

  MPI_Status status;
  int        err;
  int         tag;
  MPI_Request request;

  //  -
  //  |
  //  |   GridDecomposition: Set up PE numbering system in figure below and
  //  |                      establish communication arrays.
  //  |
  //  |                      nPEx -- number of PEs in the x-direction
  //  |                      nPEy -- number of PEs in the y-direction
  //  |                      numPE = total number of PEs
  //  |
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       |       |       |         |       |
  //  |                       |       |       |         | numPE |
  //  |                       |       |       |         |       |
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       .       .       .         .       .
  //  |                       .       .       .         .       .
  //  |                       .       .       .         .       .
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       |       |       |         |       |
  //  |                       | nPEx  | nPEx+1|         |       |
  //  |                       |       |       |         |       |
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       |       |       |         |       |
  //  |                       |   0   |   1   |         | nPEx-1|
  //  |                       |       |       |         |       |
  //  |                       +-------+-------+ . . .   +-------+
  //  |
  //  |
  //  -


  void GridDecomposition(int _nPEx, int _nPEy, int nCellx , int nCelly)
  {

    nRealx = nCellx + 1;
    nRealy = nCelly + 1;

    // Store and check incoming processor counts

    nPEx = _nPEx;
    nPEy = _nPEy;

    if (nPEx*nPEy != numPE)
      {
      if ( myPE == 0 ) cout << "Fatal Error:  Number of PEs in x-y directions do not add up to numPE" << endl;
      MPI_Barrier(MPI_COMM_WORLD);
      MPI_Finalize();
      exit(0);
      }

    // Get the i-j location of this processor, given its number.  See figure above:

    jPE = int(myPE/nPEx);
    iPE = myPE - jPE*nPEx;

    // Set neighbor values

    nei_n = nei_s = nei_e = nei_w = -1;

    if ( iPE > 0      )
      {
        nei_w = myPE - 1    ;
      }
    if ( jPE > 0      )
      {
       nei_s = myPE - nPEx ;
      }
    if ( iPE < nPEx-1 )
      {
       nei_e = myPE + 1    ;
      }
    if ( jPE < nPEy-1 )
      {
       nei_n = myPE + nPEx ;
      }

    nei_nw = nei_sw = nei_ne = nei_se = -1;

    if ( iPE > 0      && jPE > 0      )  nei_sw = myPE - nPEx - 1 ;
    if ( iPE < nPEx-1 && jPE > 0      )  nei_se = myPE - nPEx + 1 ;
    if ( iPE > 0      && jPE < nPEy-1 )  nei_nw = myPE + nPEx - 1 ;
    if ( iPE < nPEx-1 && jPE < nPEy-1 )  nei_ne = myPE + nPEx + 1 ;

    // Acquire memory for the communication between adjacent processors:
    countx = nRealx + 2;
    county = nRealy + 2;

    tag = 0;
  }



  //  ==
  //  ||
  //  ||  ParticlesExchange
  //  ||
  //  ||  Exchange particles between processors
  //  ||
  //  ||
  //  ==

  void ParticleExchange( VI &ptcl_send_list , VI &ptcl_send_PE , particles &PTCL)
  {
    MPI_Status  status;
    MPI_Request request;

    // (1) Get the max number particles to be sent by any particular processor, and make sure all processors  know that number.

    int numToSend = ptcl_send_list.size();      int maxToSend;

    /* TO-DO Place MPI call here to complete (1) */
    MPI_Barrier(MPI_COMM_WORLD);
    MPI_Iallreduce(&numToSend, &maxToSend, 1, MPI_INT, MPI_MAX, MPI_COMM_WORLD, &request); MPI_Wait(&request,&status);
    MPI_Barrier(MPI_COMM_WORLD);

    // (2) Allocate contributions to the upcoming Gather operation.  Here, "C" for "Contribution" to be Gathered

    int    *Cptcl_PE;  Cptcl_PE = new int    [ maxToSend ];  // Particles' destination PEs
    double *Cptcl_x ;  Cptcl_x  = new double [ maxToSend ];
    double *Cptcl_y ;  Cptcl_y  = new double [ maxToSend ];
    double *Cptcl_vx;  Cptcl_vx = new double [ maxToSend ];
    double *Cptcl_vy;  Cptcl_vy = new double [ maxToSend ];

    // (3) Populate contributions on all processors for the upcoming Gather operation

    for ( int i = 0 ; i < maxToSend ; ++i ) { Cptcl_PE[i] = -1; Cptcl_x [i] = 0.; Cptcl_y [i] = 0.; Cptcl_vx[i] = 0.; Cptcl_vy[i] = 0.; }
```

```
    // (4) Populate with all the particles on this PE.  Note that some/most
    processors will have left-over space in the C* arrays.

    for ( int i = 0 ; i < ptcl_send_list.size() ; ++i )
      {
   int id      = ptcl_send_list[ i ];
   Cptcl_PE[i] = ptcl_send_PE  [ i ];
   Cptcl_x [i] = PTCL.x        [ id ];
   Cptcl_y [i] = PTCL.y        [ id ];
   Cptcl_vx[i] = PTCL.vx       [ id ];
   Cptcl_vy[i] = PTCL.vy       [ id ];
      }

    // (5) Allocate and initialize the arrays for upcoming Gather operation to PE0
    .  The sizeOfGather takes
    //     into account the number of processors, like this figure:
    //
    //     |<----------------------------- sizeOfGather
    ------------------------------>|
    //     |
         |
    //     |
         |
    //     |<- maxToSend    ->|<- maxToSend    ->|<- maxToSend    ->|<- maxToSend
      ->|
    //
    +------------------+------------------+------------------+------------------+
    //            PE0               PE1                PE2               PE3


    int sizeOfGather = maxToSend * numPE;

    int    *Gptcl_PE;  Gptcl_PE = new int    [  sizeOfGather  ];
    double *Gptcl_x ;  Gptcl_x  = new double [  sizeOfGather  ];
    double *Gptcl_y ;  Gptcl_y  = new double [  sizeOfGather  ];
    double *Gptcl_vx;  Gptcl_vx = new double [  sizeOfGather  ];
    double *Gptcl_vy;  Gptcl_vy = new double [  sizeOfGather  ];

    for ( int i = 0 ; i < sizeOfGather ; ++i ) { Gptcl_PE[i] = -1; Gptcl_x [i] =
    0.; Gptcl_y [i] = 0.; Gptcl_vx[i] = 0.; Gptcl_vy[i] = 0.;  }


    // (6)  Gather "Contributions" ("C" arrays) from all PEs onto all PEs into
    these bigger arrays so all PE will know what particles
    //      need to go where.

    MPI_Barrier(MPI_COMM_WORLD);

    MPI_Iallgather( Cptcl_PE, maxToSend, MPI_INT, Gptcl_PE, maxToSend, MPI_INT,
    MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_x, maxToSend, MPI_DOUBLE, Gptcl_x, maxToSend, MPI_DOUBLE
    , MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_y, maxToSend, MPI_DOUBLE, Gptcl_y, maxToSend, MPI_DOUBLE
    , MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_vx, maxToSend, MPI_DOUBLE, Gptcl_vx, maxToSend,
    MPI_DOUBLE, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_vy, maxToSend, MPI_DOUBLE, Gptcl_vy, maxToSend, MPI_DOUBLE, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);


    MPI_Barrier(MPI_COMM_WORLD);

    // (7) Put in vector form so they can be added to PTCL.  These arrays are 1-based.

    int Np = 0;  for ( int i = 0 ; i < sizeOfGather ; ++i ) if ( Gptcl_PE[i] == myPE ) ++Np;

    VD  std_add_x  ;  std_add_x.resize  ( Np+1 );
    VD  std_add_y  ;  std_add_y.resize  ( Np+1 );
    VD  std_add_vx ;  std_add_vx.resize ( Np+1 );
    VD  std_add_vy ;  std_add_vy.resize ( Np+1 );

    int count = 1;
    for ( int i = 0 ; i < sizeOfGather ; ++i )
      if ( Gptcl_PE[i] == myPE )
   {
     std_add_x [ count ] = Gptcl_x[i];
     std_add_y [ count ] = Gptcl_y [i];
     std_add_vx[ count ] = Gptcl_vx[i];
     std_add_vy[ count ] = Gptcl_vy[i];
     ++ count;
   }

    PTCL.add( std_add_x, std_add_y, std_add_vx, std_add_vy );

    // (8) Free up memory

    if (maxToSend    > 0 ) { delete[] Cptcl_PE;  delete[] Cptcl_x ;  delete[] Cptcl_y ; delete[] Cptcl_vx ; delete[] Cptcl_vy;  }
    if (sizeOfGather > 0 ) { delete[] Gptcl_PE;  delete[] Gptcl_x ;  delete[] Gptcl_y ; delete[] Gptcl_vx ; delete[] Gptcl_vy;  }

  }


  int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }

};
```

Listing 1: Parallel Code (mpiInfo.h)

```
// ==============================================================================
//  ||                                                                          ||
//  ||              fp                                                          ||
//  ||              -------------------------------------------                 ||
//  ||              F R E E   P A R T I C L E                                ||
//  ||                                                                       ||
//  ||              D E M O N S T R A T I O N   C O D E                      ||
//  ||              -------------------------------------------              ||
//  ||                                                                       ||
//  ||       Developed by: Scott R. Runnels, Ph.D.                           ||
//  ||                     University of Colorado Boulder                    ||
//  ||                                                                       ||
//  ||                For: CU Boulder CSCI 4576/5576 and associated labs     ||
//  ||                                                                       ||
//  ||           Copyright 2020 Scott Runnels                                ||
//  ||                                                                       ||
//  ||                     Not for distribution or use outside of the        ||
//  ||                     this course.                                      ||
//  ||                                                                       ||
//  ================================================================================

#include "mpi.h"
#include "fp.h"
#include "particles.h"
#include "mpiInfo.h"


// ==
// ||
// ||      C L A S S:   M E S H
// ||
// ||
// ||
// ||  (21)----(22)----(23)----(24)-----(25)  <--- nRealx+1 , nRealy+1
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (16)----[17]----[18]----[19]-----(20)
// ||   |    1,3|    2,3|    3,3|        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (11)----[12]----[13]----[14]-----(15)
// ||   |    1,2|    2,2|    3,2|        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (6)-----[7]-----[8]-----[9]------(10)
// ||   |    1,1|    2,1|    3,1|        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (1)-----(2)-----(3)-----(4)------(5)
// || 0,0                              4,0
// ||
// ==

class Mesh
{

public:

  double x0, x1, y0, y1;
  VD x,y;
  int nRealx    , nRealy   , nField, nReal;
  double lengthx, lengthy, dx, dy;
  int myPE;
  VD Qval;

  //  ==
  //  ||
  //  ||  Constructor:  Initialize values
  //  ||
  //  ==

  Mesh(double _x0  , double _x1,  double _y0, double _y1 , int ncell_x , int ncell_y, mpiInfo &myMPI )
  {


    // Copy incoming values

    x0   = _x0;          x1 = _x1;
    y0   = _y0;          y1 = _y1;
    myPE = myMPI.myPE;

    // Compute number of real (physical) nodes, and their spacing

    nRealx      = ncell_x + 1;
    nRealy      = ncell_y + 1;
    nReal       = nRealx*nRealy;
    dx          = (x1-x0) / ncell_x;
    dy          = (y1-y0) / ncell_y;

    // Compute the size of the field variables, which also lie on ghost nodes

    nField      = (nRealx+2)*(nRealy+2);

    // Allocate memory -- Note that the node numbers and field variable numbers must
    // match.  So even though we only will be caring about real nodes, their node
    // numbers are naturally ordered, so must be of size nField.

    x.resize(nField+1); y.resize(nField+1); Qval.resize(nField+1);

    for ( int i = 1 ; i <= nRealx ; ++i )
      for ( int j = 1 ; j <= nRealy ; ++j )
   {
     int p = pid(i,j);
     x[p] = x0 + (i-1)*dx;
     y[p] = y0 + (j-1)*dy;
   }

  }



  //  ==
  //  ||
  //  ||  ParticlesOnMesh
  //  ||
  //  ||  Quantify the particles' interaction with the mesh
  //  ||
  //  ||  References:  [1] https://www.particleincell.com/2010/es-pic-method/
  //  ||
  //  ==

  void ParticlesOnMesh(particles &PTCL, mpiInfo &myMPI)
  {
    double hx, hy;
    double w[5];
    int    p[5];
    int    iL, iR, jB, jT;
    int    iPEnew, jPEnew;            // These store the i-j indicies of the processor receiving a particle,
                                       //    if that particle is leaving the mesh.
    VI ptcl_send_list, ptcl_send_PE;  // These collect information about particles that
                                       //    have left this processor and are heading onto
                                       //    another processor.

    // -
    // |
    // | Determine which particles are still on this mesh and which have left
    // |
    // -

    for ( int k = 1 ; k <= PTCL.n ; ++k )
      {
   // First, check to be sure the particle is still in the mesh.  If it is not, set its
   // "active" flag to zero, and note the processor to which it is going for MPI exchange.

```

```
    if ( PTCL.active[k] == 1 )
      {
        iPEnew = myMPI.iPE;
        jPEnew = myMPI.jPE;


        if ( PTCL.x[k] < x0     ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE - 1 ; }
        if ( PTCL.x[k] > x1     ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE + 1 ; }
        if ( PTCL.y[k] < y0     ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE - 1 ; }
        if ( PTCL.y[k] > y1     ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE + 1 ; }


      }

    // The particle is not in the mesh.  Collect this particle into a holding array that will
    // be sent to the neighboring processor.

    if ( PTCL.active[k] == -1)
      {
        if ( iPEnew >= 0 && iPEnew < myMPI.nPEx )
        if ( jPEnew >= 0 && jPEnew < myMPI.nPEy )
          {
       ptcl_send_list.push_back(k);
       ptcl_send_PE  .push_back( iPEnew + jPEnew * myMPI.nPEx);
          }

        PTCL.active[k] = 0;  // Remove it from the list of active particles
      }
       }

     // -
     // |
     // | Give and receive particles to/with other processors
     // |
     // -

     myMPI.ParticleExchange( ptcl_send_list , ptcl_send_PE , PTCL);

     // -
     // |
     // | Accumulate particles to the nodes (to be completed in the esPIC code, next week)
     // |
     // -

     for ( int k = 1 ; k <= nField ; ++k ) Qval[k] = 0.;

     // -
     // |
     // | Compute forces on particles
     // |
     // -

     for ( int k = 1 ; k <= PTCL.n ; ++k ) PTCL.xf[k] = PTCL.yf[k] = 0.;

     for ( int k = 1 ; k <= PTCL.n ; ++k )
       {
    if ( PTCL.active[k] == 1 )
      {
        PTCL.xf[ k ] = 0.;
        PTCL.yf[ k ] =-.4;
         }
        }


   }

 #include "mesh_plotter.h"

   int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }

 };





 //  ==
 //  ||
 //  ||
 //  ||  Main Program
 //  ||
 //  ||
 //  ==
 //

 int main(int argc, char *argv[])
 {

    mpiInfo myMPI;
    MPI_Init     (&argc         , &argv       );
    MPI_Comm_size(MPI_COMM_WORLD, &myMPI.numPE);
    MPI_Comm_rank(MPI_COMM_WORLD,&myMPI.myPE  );

    int nPEx, nPEy, nCellx, nCelly;
    double tEnd, dt;
    double flux;

    // -
    // |
    // | Banner and Input
    // |
    // -

    if ( myMPI.myPE == 0 )
```

```
    {
      cout << "\n";
      cout << "------------------------------------------\n";
      cout << "\n";
      cout << " F R E E  P A R T I C L E                 \n";
      cout << " D E M O  C O D E                         \n";
      cout << "\n";
      cout << " Running on " << myMPI.numPE << " processors \n";
      cout << "\n";
      cout << "------------------------------------------\n";
      cout << "\n";
    }

  for (int count =  0 ; count < argc; ++count)
    {
      if ( !strcmp(argv[count],"-nPEx"  ) ) nPEx   = atoi(argv[count+1]);
      if ( !strcmp(argv[count],"-nPEy"  ) ) nPEy   = atoi(argv[count+1]);
      if ( !strcmp(argv[count],"-nCellx") ) nCellx = atoi(argv[count+1]);
      if ( !strcmp(argv[count],"-nCelly") ) nCelly = atoi(argv[count+1]);
      if ( !strcmp(argv[count],"-flux"  ) ) flux   = atof(argv[count+1]);
      if ( !strcmp(argv[count],"-tEnd"  ) ) tEnd   = atof(argv[count+1]);
      if ( !strcmp(argv[count],"-dt"    ) ) dt     = atof(argv[count+1]);
    }

  if ( myMPI.myPE == 0 )
    {
      cout << endl;
      cout << "Input Summary: " << endl;
      cout << "-------------------------- " << endl;
      cout << "No. PE   in  x-direction: " << nPEx   << endl;
      cout << "             y-direction: " << nPEy   << endl;
      cout << "No. Cells in x-direction: " << nCellx << endl;
      cout << "             y-direction: " << nCelly << endl;
      cout << "Flux density           : " << flux << endl;
      cout << "End Time               : " << tEnd   << endl;
      cout << "Time Step              : " << dt     << endl;
      cout << endl;
    }

  // -
  // |
  // | MPI / Processor ID
  // |
  // -

  myMPI.GridDecomposition(nPEx,nPEy,nCellx,nCelly);

  // -
  // |
  // | Parallel Grid Generation
  // |
  // -

  double totalLength = 1.;
  double eachPElength_x = totalLength / nPEx;
  double eachPElength_y = totalLength / nPEy;

  double x0 = eachPElength_x * myMPI.iPE;   double x1 = x0 + eachPElength_x;
  double y0 = eachPElength_y * myMPI.jPE;   double y1 = y0 + eachPElength_y;

  Mesh MESH( x0  , x1 , y0 , y1 , nCellx , nCelly , myMPI );

  // -
  // |
  // | Set up Particles
  // |
  // -

  particles PTCL(500);

  int    count   = 0;
  double density = flux/(MESH.y1-MESH.y0);
  double vx_bdy  = .5;
  double vy_bdy  = .0;

  // -
  // |
  // | Time Marching Loop
  // |
  // -

  for ( double t = 0. ; t <= tEnd ; t += dt )
   {
    cout << endl;
    cout << "myPE: " << myMPI.myPE << " Time = " << t << endl;

    // Inject particles

    if ( myMPI.iPE == 0 ) PTCL.addFlux( t , MESH.y0, MESH.y1, density, vx_bdy ,
   vy_bdy );

    // Move particles

    PTCL.move( dt );

    // Map between particles and the mesh

    MESH.ParticlesOnMesh(PTCL,myMPI);

    // Plot

    PTCL.plot( "ptcl" , count , myMPI.myPE );
    MESH.plot( "mesh" , count , myMPI );

    ++count;
   }


  // -
  // |
```

```cpp
   // | Wrap-Up
   // |
   // -

   if ( myMPI.myPE == 0 ) cout << "\n\n ** Successful Completion ** \n\n";

   MPI_Finalize();

   return 0;

}
```

Listing 2: Parallel Code (fp.cpp)

```bash
#!/bin/bash

# -
# |
# | This is a batch script for running a MPI parallel job on Summit
# |
# | (o) To submit this job, enter:  sbatch --export=CODE=’/home/scru5660/HPSC/codes/fd_mpi/src’ ex_01.bat
# |
# | (o) To check the status of this job, enter: squeue -u <username>
# |
# -

# -
# |
# | Part 1: Directives
# |
# -

#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --time=00:01:00
#SBATCH --partition=amilan
#SBATCH --output=ex01-%j.out

# -
# |
# | Part 2: Loading software
# |
# -

module purge
module load intel
module load impi

# -
# |
# | Part 3: User scripting
# |
# -

echo "=="
echo "||"
echo "|| Begin Execution of fp in slurm batch script."
echo "||"
echo "=="

srun -n 4 $CODE/fp -nPEx 2 -nPEy 2 -nCellx 5 -nCelly 5 -flux 10 -tEnd 5 -dt 0.05 > tty.out

echo "=="
echo "||"
echo "|| Execution of fp in slurm batch script complete."
echo "||"
echo "=="
```

Listing 3: Slurm File (ex01.bat)

4 Appendix B: Output

4.1 Output of the Slurm

```

---------------------------------------------

 F R E E   P A R T I C L E
 D E M O   C O D E

 Running on 4 processors

---------------------------------------------


Input Summary:
---------------------------
No. PE   in  x-direction: 2
             y-direction: 2
No. Cells in x-direction: 5
             y-direction: 5
Flux density            : 10
End Time                : 5
Time Step               : 0.05


myPE: 0 Time = 0

myPE: 1 Time = 0

myPE: 2 Time = 0

myPE: 3 Time = 0

myPE: 0 Time = 0.05

myPE: 3 Time = 0.05

myPE: 1 Time = 0.05
```

```

myPE: 2 Time = 0.05

myPE: 0 Time = 0.1

myPE: 3 Time = 0.1

myPE: 1 Time = 0.1

myPE: 2 Time = 0.1

myPE: 0 Time = 0.15

myPE: 2 Time = 0.15

myPE: 3 Time = 0.15

myPE: 1 Time = 0.15

myPE: 0 Time = 0.2

myPE: 1 Time = 0.2

myPE: 3 Time = 0.2

myPE: 2 Time = 0.2

myPE: 0 Time = 0.25

myPE: 3 Time = 0.25

myPE: 1 Time = 0.25

myPE: 2 Time = 0.25

myPE: 0 Time = 0.3

myPE: 2 Time = 0.3

myPE: 3 Time = 0.3

myPE: 1 Time = 0.3

myPE: 0 Time = 0.35

myPE: 3 Time = 0.35

myPE: 1 Time = 0.35

myPE: 2 Time = 0.35

myPE: 1 Time = 0.4

myPE: 0 Time = 0.4

```

```
myPE: 3 Time = 0.4

myPE: 2 Time = 0.4

myPE: 1 Time = 0.45

myPE: 2 Time = 0.45

myPE: 0 Time = 0.45

myPE: 3 Time = 0.45

myPE: 2 Time = 0.5

myPE: 3 Time = 0.5

myPE: 0 Time = 0.5

myPE: 1 Time = 0.5

myPE: 3 Time = 0.55

myPE: 2 Time = 0.55

myPE: 1 Time = 0.55

myPE: 0 Time = 0.55

myPE: 3 Time = 0.6

myPE: 2 Time = 0.6

myPE: 1 Time = 0.6

myPE: 0 Time = 0.6

myPE: 3 Time = 0.65

myPE: 2 Time = 0.65

myPE: 0 Time = 0.65

myPE: 1 Time = 0.65

myPE: 2 Time = 0.7

myPE: 3 Time = 0.7

myPE: 0 Time = 0.7

myPE: 1 Time = 0.7

myPE: 3 Time = 0.75

myPE: 2 Time = 0.75
```

```

myPE: 0 Time = 0.75

myPE: 1 Time = 0.75

myPE: 3 Time = 0.8

myPE: 2 Time = 0.8

myPE: 1 Time = 0.8

myPE: 0 Time = 0.8

myPE: 3 Time = 0.85

myPE: 2 Time = 0.85

myPE: 1 Time = 0.85

myPE: 0 Time = 0.85

myPE: 2 Time = 0.9

myPE: 3 Time = 0.9

myPE: 1 Time = 0.9

myPE: 0 Time = 0.9

myPE: 3 Time = 0.95

myPE: 0 Time = 0.95

myPE: 2 Time = 0.95

myPE: 1 Time = 0.95

myPE: 3 Time = 1

myPE: 1 Time = 1

myPE: 0 Time = 1

myPE: 2 Time = 1

myPE: 3 Time = 1.05

myPE: 1 Time = 1.05

myPE: 0 Time = 1.05

myPE: 2 Time = 1.05

myPE: 3 Time = 1.1

```

```
myPE: 1 Time = 1.1

myPE: 0 Time = 1.1

myPE: 2 Time = 1.1

myPE: 3 Time = 1.15

myPE: 1 Time = 1.15

myPE: 0 Time = 1.15

myPE: 2 Time = 1.15

myPE: 3 Time = 1.2

myPE: 2 Time = 1.2

myPE: 0 Time = 1.2

myPE: 1 Time = 1.2

myPE: 3 Time = 1.25

myPE: 2 Time = 1.25

myPE: 0 Time = 1.25

myPE: 1 Time = 1.25

myPE: 3 Time = 1.3

myPE: 2 Time = 1.3

myPE: 0 Time = 1.3

myPE: 1 Time = 1.3

myPE: 3 Time = 1.35

myPE: 1 Time = 1.35

myPE: 0 Time = 1.35

myPE: 2 Time = 1.35

myPE: 1 Time = 1.4

myPE: 2 Time = 1.4

myPE: 0 Time = 1.4

myPE: 3 Time = 1.4

myPE: 1 Time = 1.45
```

```

myPE: 3 Time = 1.45

myPE: 0 Time = 1.45

myPE: 2 Time = 1.45

myPE: 3 Time = 1.5

myPE: 1 Time = 1.5

myPE: 0 Time = 1.5

myPE: 2 Time = 1.5

myPE: 3 Time = 1.55

myPE: 2 Time = 1.55

myPE: 0 Time = 1.55

myPE: 1 Time = 1.55

myPE: 3 Time = 1.6

myPE: 2 Time = 1.6

myPE: 0 Time = 1.6

myPE: 1 Time = 1.6

myPE: 3 Time = 1.65

myPE: 1 Time = 1.65

myPE: 0 Time = 1.65

myPE: 2 Time = 1.65

myPE: 3 Time = 1.7

myPE: 2 Time = 1.7

myPE: 0 Time = 1.7

myPE: 1 Time = 1.7

myPE: 3 Time = 1.75

myPE: 2 Time = 1.75

myPE: 1 Time = 1.75

myPE: 0 Time = 1.75

```

```
myPE: 2 Time = 1.8

myPE: 1 Time = 1.8

myPE: 0 Time = 1.8

myPE: 3 Time = 1.8

myPE: 1 Time = 1.85

myPE: 2 Time = 1.85

myPE: 0 Time = 1.85

myPE: 3 Time = 1.85

myPE: 1 Time = 1.9

myPE: 2 Time = 1.9

myPE: 0 Time = 1.9

myPE: 3 Time = 1.9

myPE: 2 Time = 1.95

myPE: 1 Time = 1.95

myPE: 0 Time = 1.95

myPE: 3 Time = 1.95

myPE: 2 Time = 2

myPE: 3 Time = 2

myPE: 0 Time = 2

myPE: 1 Time = 2

myPE: 2 Time = 2.05

myPE: 1 Time = 2.05

myPE: 3 Time = 2.05

myPE: 0 Time = 2.05

myPE: 3 Time = 2.1

myPE: 2 Time = 2.1

myPE: 1 Time = 2.1

myPE: 0 Time = 2.1
```

```

myPE: 2 Time = 2.15

myPE: 1 Time = 2.15

myPE: 0 Time = 2.15

myPE: 3 Time = 2.15

myPE: 2 Time = 2.2

myPE: 3 Time = 2.2

myPE: 0 Time = 2.2

myPE: 1 Time = 2.2

myPE: 1 Time = 2.25

myPE: 3 Time = 2.25

myPE: 2 Time = 2.25

myPE: 0 Time = 2.25

myPE: 3 Time = 2.3

myPE: 2 Time = 2.3

myPE: 0 Time = 2.3

myPE: 1 Time = 2.3

myPE: 1 Time = 2.35

myPE: 3 Time = 2.35

myPE: 2 Time = 2.35

myPE: 0 Time = 2.35

myPE: 2 Time = 2.4

myPE: 1 Time = 2.4

myPE: 3 Time = 2.4

myPE: 0 Time = 2.4

myPE: 1 Time = 2.45

myPE: 2 Time = 2.45

myPE: 3 Time = 2.45

```

```
myPE: 0 Time = 2.45

myPE: 3 Time = 2.5

myPE: 2 Time = 2.5

myPE: 1 Time = 2.5

myPE: 0 Time = 2.5

myPE: 3 Time = 2.55

myPE: 2 Time = 2.55

myPE: 0 Time = 2.55

myPE: 1 Time = 2.55

myPE: 2 Time = 2.6

myPE: 0 Time = 2.6

myPE: 3 Time = 2.6

myPE: 1 Time = 2.6

myPE: 2 Time = 2.65

myPE: 3 Time = 2.65

myPE: 0 Time = 2.65

myPE: 1 Time = 2.65

myPE: 3 Time = 2.7

myPE: 2 Time = 2.7

myPE: 0 Time = 2.7

myPE: 1 Time = 2.7

myPE: 1 Time = 2.75

myPE: 3 Time = 2.75

myPE: 0 Time = 2.75

myPE: 2 Time = 2.75

myPE: 1 Time = 2.8

myPE: 3 Time = 2.8

myPE: 0 Time = 2.8
```

```

myPE: 2 Time = 2.8

myPE: 3 Time = 2.85

myPE: 1 Time = 2.85

myPE: 0 Time = 2.85

myPE: 2 Time = 2.85

myPE: 1 Time = 2.9

myPE: 0 Time = 2.9

myPE: 3 Time = 2.9

myPE: 2 Time = 2.9

myPE: 1 Time = 2.95

myPE: 3 Time = 2.95

myPE: 0 Time = 2.95

myPE: 2 Time = 2.95

myPE: 0 Time = 3

myPE: 3 Time = 3

myPE: 1 Time = 3

myPE: 2 Time = 3

myPE: 0 Time = 3.05

myPE: 2 Time = 3.05

myPE: 1 Time = 3.05

myPE: 3 Time = 3.05

myPE: 3 Time = 3.1

myPE: 0 Time = 3.1

myPE: 1 Time = 3.1

myPE: 2 Time = 3.1

myPE: 0 Time = 3.15

myPE: 3 Time = 3.15

```

```
myPE: 1 Time = 3.15

myPE: 2 Time = 3.15

myPE: 0 Time = 3.2

myPE: 2 Time = 3.2

myPE: 1 Time = 3.2

myPE: 3 Time = 3.2

myPE: 0 Time = 3.25

myPE: 2 Time = 3.25

myPE: 3 Time = 3.25

myPE: 1 Time = 3.25

myPE: 2 Time = 3.3

myPE: 0 Time = 3.3

myPE: 3 Time = 3.3

myPE: 1 Time = 3.3

myPE: 3 Time = 3.35

myPE: 2 Time = 3.35

myPE: 0 Time = 3.35

myPE: 1 Time = 3.35

myPE: 2 Time = 3.4

myPE: 3 Time = 3.4

myPE: 0 Time = 3.4

myPE: 1 Time = 3.4

myPE: 0 Time = 3.45

myPE: 2 Time = 3.45

myPE: 3 Time = 3.45

myPE: 1 Time = 3.45

myPE: 3 Time = 3.5

myPE: 2 Time = 3.5
```

```

myPE: 0 Time = 3.5

myPE: 1 Time = 3.5

myPE: 3 Time = 3.55

myPE: 1 Time = 3.55

myPE: 0 Time = 3.55

myPE: 2 Time = 3.55

myPE: 3 Time = 3.6

myPE: 1 Time = 3.6

myPE: 0 Time = 3.6

myPE: 2 Time = 3.6

myPE: 3 Time = 3.65

myPE: 0 Time = 3.65

myPE: 1 Time = 3.65

myPE: 2 Time = 3.65

myPE: 3 Time = 3.7

myPE: 1 Time = 3.7

myPE: 2 Time = 3.7

myPE: 0 Time = 3.7

myPE: 2 Time = 3.75

myPE: 1 Time = 3.75

myPE: 3 Time = 3.75

myPE: 0 Time = 3.75

myPE: 2 Time = 3.8

myPE: 0 Time = 3.8

myPE: 3 Time = 3.8

myPE: 1 Time = 3.8

myPE: 3 Time = 3.85

```

```
myPE: 0 Time = 3.85

myPE: 1 Time = 3.85

myPE: 2 Time = 3.85

myPE: 3 Time = 3.9

myPE: 0 Time = 3.9

myPE: 1 Time = 3.9

myPE: 2 Time = 3.9

myPE: 3 Time = 3.95

myPE: 2 Time = 3.95

myPE: 0 Time = 3.95

myPE: 1 Time = 3.95

myPE: 0 Time = 4

myPE: 2 Time = 4

myPE: 3 Time = 4

myPE: 1 Time = 4

myPE: 2 Time = 4.05

myPE: 0 Time = 4.05

myPE: 1 Time = 4.05

myPE: 3 Time = 4.05

myPE: 1 Time = 4.1

myPE: 2 Time = 4.1

myPE: 0 Time = 4.1

myPE: 3 Time = 4.1

myPE: 1 Time = 4.15

myPE: 2 Time = 4.15

myPE: 0 Time = 4.15

myPE: 3 Time = 4.15

myPE: 1 Time = 4.2
```

```

myPE: 2 Time = 4.2

myPE: 0 Time = 4.2

myPE: 3 Time = 4.2

myPE: 2 Time = 4.25

myPE: 0 Time = 4.25

myPE: 1 Time = 4.25

myPE: 3 Time = 4.25

myPE: 0 Time = 4.3

myPE: 1 Time = 4.3

myPE: 2 Time = 4.3

myPE: 3 Time = 4.3

myPE: 0 Time = 4.35

myPE: 2 Time = 4.35

myPE: 1 Time = 4.35

myPE: 3 Time = 4.35

myPE: 0 Time = 4.4

myPE: 1 Time = 4.4

myPE: 2 Time = 4.4

myPE: 3 Time = 4.4

myPE: 0 Time = 4.45

myPE: 1 Time = 4.45

myPE: 2 Time = 4.45

myPE: 3 Time = 4.45

myPE: 2 Time = 4.5

myPE: 0 Time = 4.5

myPE: 1 Time = 4.5

myPE: 3 Time = 4.5

```

```
myPE: 2 Time = 4.55

myPE: 1 Time = 4.55

myPE: 3 Time = 4.55

myPE: 0 Time = 4.55

myPE: 2 Time = 4.6

myPE: 3 Time = 4.6

myPE: 1 Time = 4.6

myPE: 0 Time = 4.6

myPE: 2 Time = 4.65

myPE: 3 Time = 4.65

myPE: 0 Time = 4.65

myPE: 1 Time = 4.65

myPE: 2 Time = 4.7

myPE: 3 Time = 4.7

myPE: 0 Time = 4.7

myPE: 1 Time = 4.7

myPE: 2 Time = 4.75

myPE: 1 Time = 4.75

myPE: 3 Time = 4.75

myPE: 0 Time = 4.75

myPE: 2 Time = 4.8

myPE: 1 Time = 4.8

myPE: 3 Time = 4.8

myPE: 0 Time = 4.8

myPE: 2 Time = 4.85

myPE: 0 Time = 4.85

myPE: 1 Time = 4.85

myPE: 3 Time = 4.85
```

```

myPE: 1 Time = 4.9

myPE: 0 Time = 4.9

myPE: 2 Time = 4.9

myPE: 3 Time = 4.9

myPE: 1 Time = 4.95

myPE: 3 Time = 4.95

myPE: 0 Time = 4.95

myPE: 2 Time = 4.95

myPE: 1 Time = 5

myPE: 3 Time = 5

myPE: 0 Time = 5

myPE: 2 Time = 5


 ** Successful Completion **
```

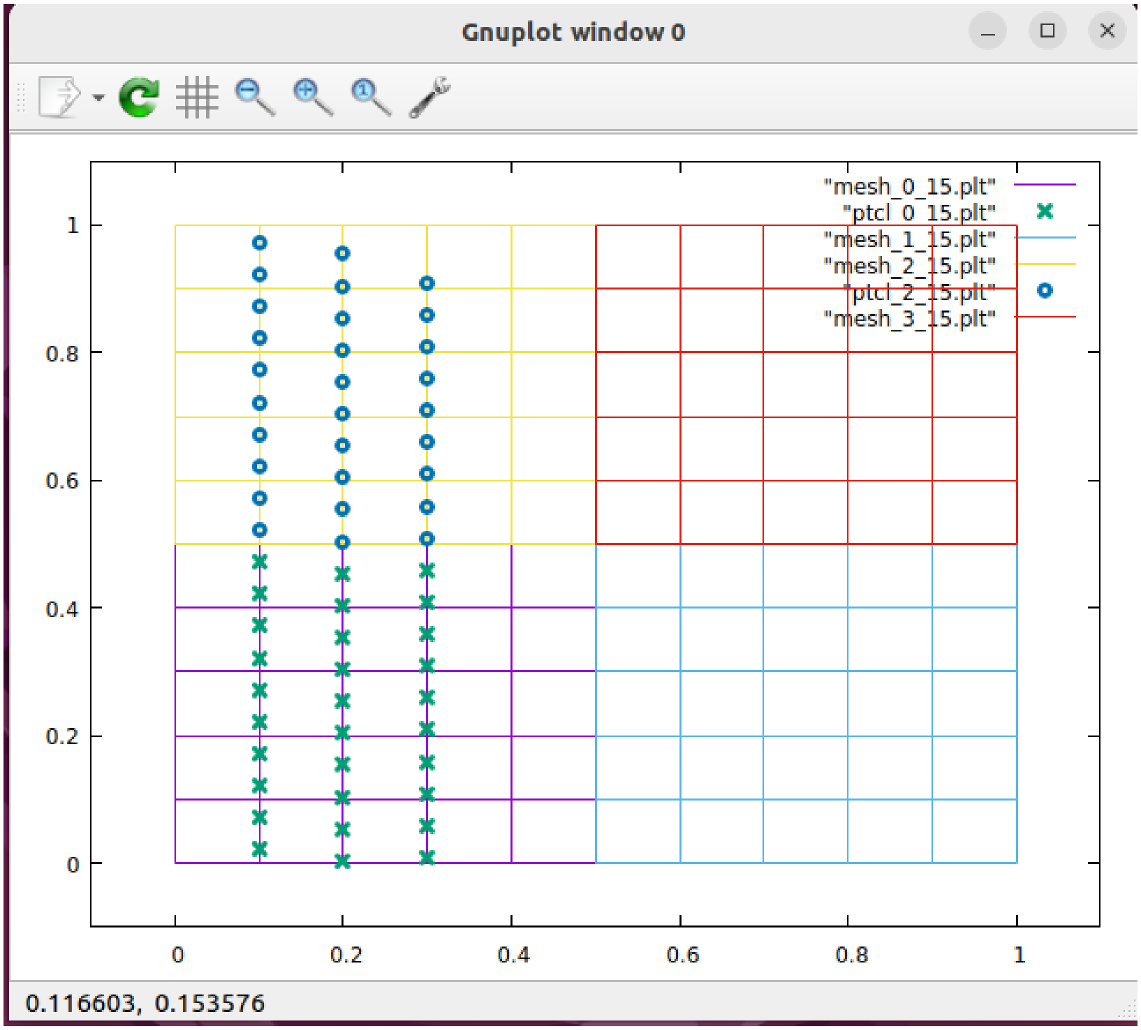
Listing 4: tty.out

4.2 Output from Alpine

```
==
||
|| Begin Execution of fp in slurm batch script.
||
==
==
||
|| Execution of fp in slurm batch script complete.
||
==
```

Listing 5: ex01-3062889.out

4.3 Plot Images, Showing Particles Moving



0.116603, 0.153576

Figure 1: Particle at Start

4.4 Plot Images, Showing Particles Moving

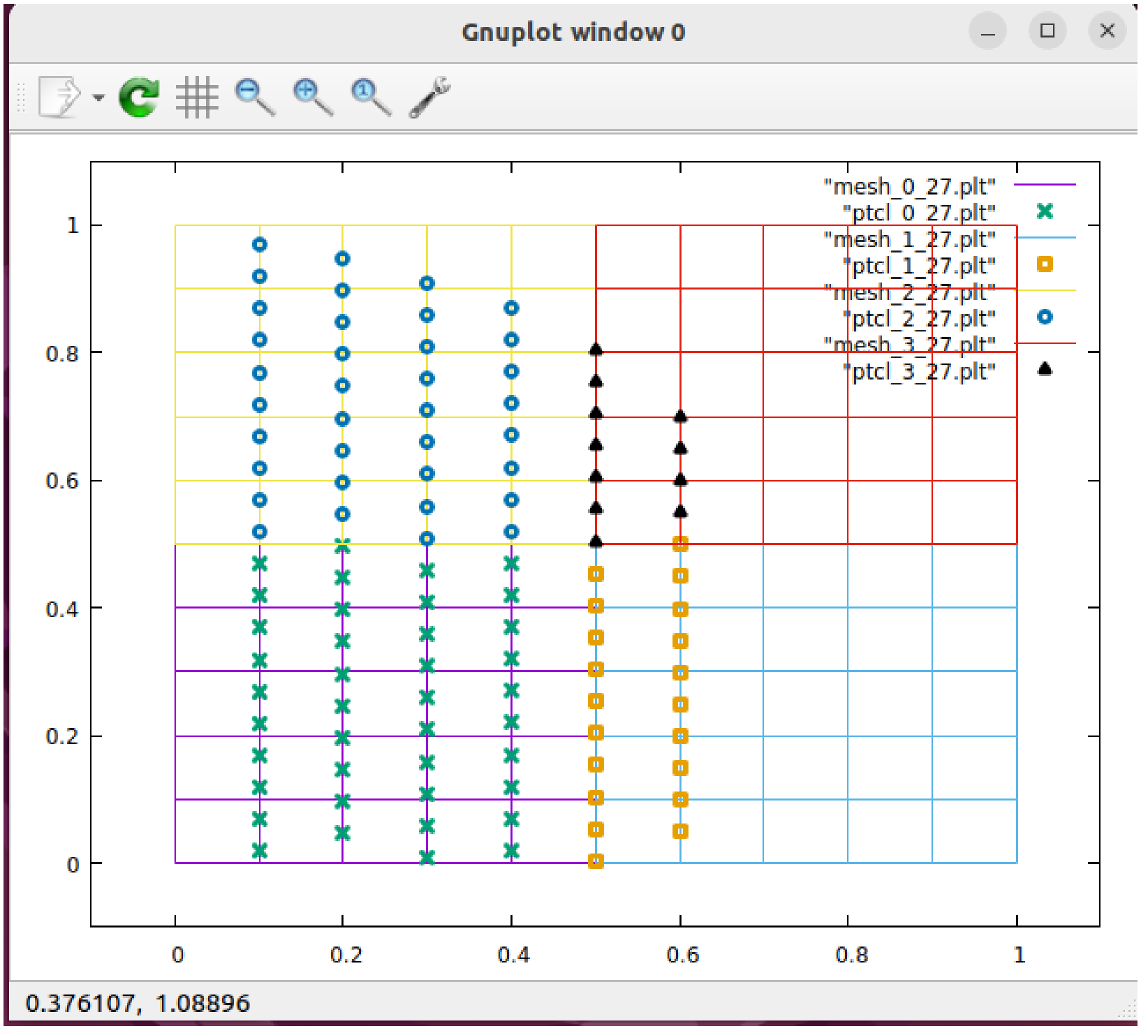


Figure 2: Particles Start Moving to other Processors

4.5 Plot Images, Showing Particles Moving

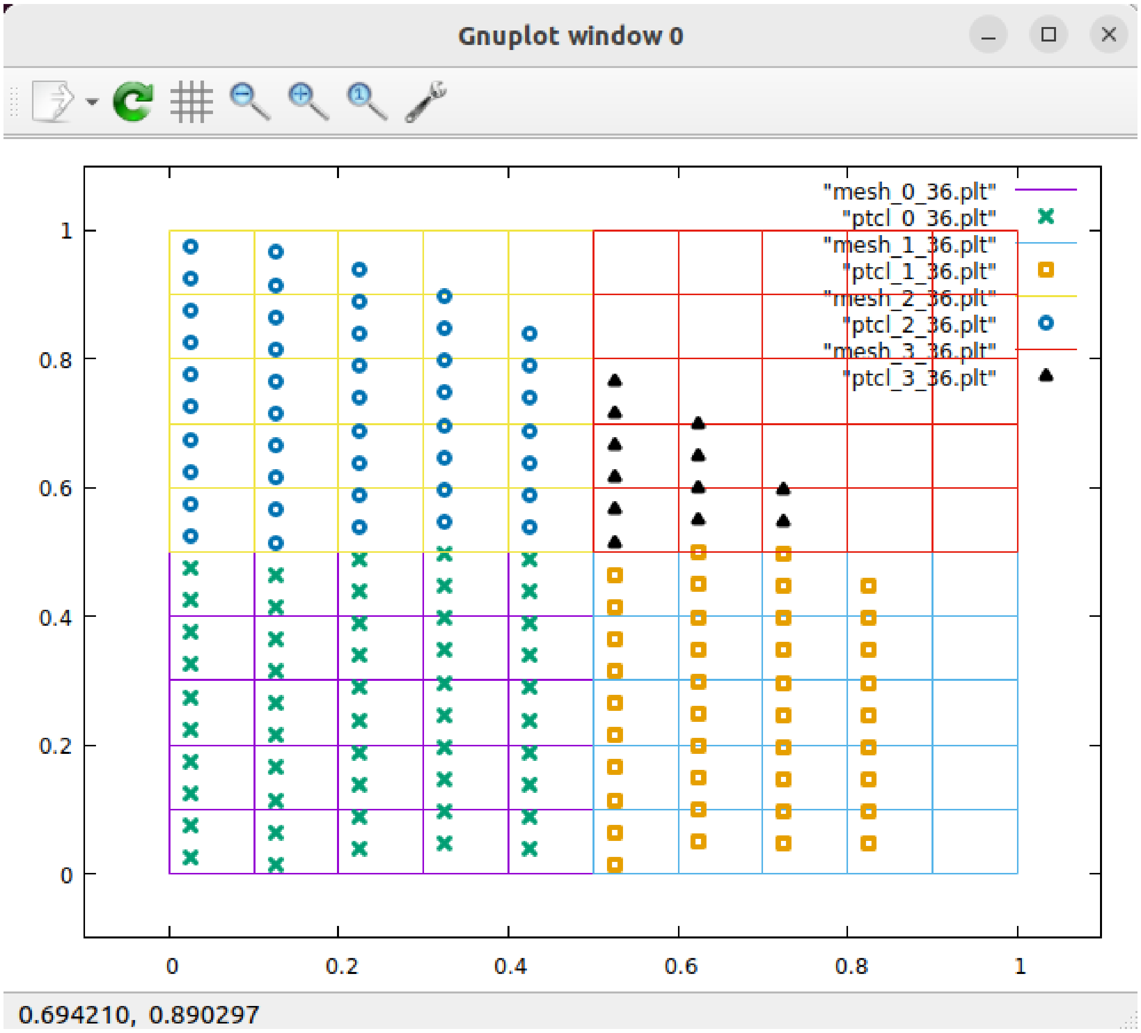


Figure 3: Particles Moving in all Processors

4.6 Debugging Code

```
#!/bin/bash

# -
# |
# | This is a batch script for running a MPI parallel job on Summit
# |
# | (o) To submit this job, enter:  sbatch --export=CODE='/home/scru5660/HPSC/
    codes/fd_mpi/src' ex_01.bat
# |
# | (o) To check the status of this job, enter: squeue -u <username>
# |
# -

# -
# |
# | Part 1: Directives
# |
# -

#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --time=00:02:00
#SBATCH --partition=atesting
#SBATCH --output=ex01-%j.out

# -
# |
# | Part 2: Loading software
# |
# -

module purge
module update gcc
module load gcc
module load openmpi
module load impi

# -
# |
# | Part 3: User scripting
# |
# -

echo "=="
echo "||"
echo "|| Begin Execution of fp in slurm batch script."
echo "||"
echo "=="

mpirun -n 4 $CODE/fp -nPEx 2 -nPEy 2 -nCellx 5 -nCelly 5 -flux 10 -tEnd 3 -dt 0.01 > tty.out

echo "=="
echo "||"
echo "|| Execution of fp in slurm batch script complete."
echo "||"
echo "=="
```

Listing 6: Testing Slurm File

```
// ===============================================================================
//  ||  ||
//  ||              fp  ||
//  ||              -------------------------------------------               ||
//  ||              F R E E   P A R T I C L E                                 ||
//  ||                                                                        ||
//  ||              D E M O N S T R A T I O N   C O D E                       ||
//  ||              -------------------------------------------               ||
//  ||                                                                        ||
//  ||       Developed by: Scott R. Runnels, Ph.D.                            ||
//  ||                     University of Colorado Boulder                     ||
//  ||                                                                        ||
//  ||                For: CU Boulder CSCI 4576/5576 and associated labs      ||
//  ||                                                                        ||
//  ||           Copyright 2020 Scott Runnels                                 ||
//  ||                                                                        ||
//  ||                     Not for distribution or use outside of the         ||
//  ||                     this course.                                       ||
//  ||                                                                        ||
// ==============================================================================

#include "mpi.h"
#include "fp.h"
#include "particles.h"
#include "mpiInfo.h"


// ==
// ||
// ||      C L A S S:   M E S H
// ||
// ||
// ||
// ||  (21)----(22)----(23)----(24)-----(25)  <--- nRealx+1 , nRealy+1
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (16)----[17]----[18]----[19]-----(20)
// ||   |    1,3|    2,3|    3,3|        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (11)----[12]----[13]----[14]-----(15)
// ||   |    1,2|    2,2|    3,2|        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (6)-----[7]-----[8]-----[9]------(10)
// ||   |    1,1|    2,1|    3,1|        |
// ||   |       |       |       |        |
// ||   |       |       |       |        |
// ||  (1)-----(2)-----(3)-----(4)------(5)
// || 0,0                              4,0
// ||
// ==

class Mesh
{

public:

  double x0, x1, y0, y1;
  VD x,y;
  int nRealx    , nRealy   , nField, nReal;
  double lengthx, lengthy, dx, dy;
  int myPE;
  VD Qval;

  //  ==
  //  ||
  //  ||  Constructor:  Initialize values
  //  ||
  //  ==

  Mesh(double _x0  , double _x1,  double _y0, double _y1 , int ncell_x , int ncell_y, mpiInfo &myMPI )
  {


    // Copy incoming values

    x0   = _x0;          x1 = _x1;
    y0   = _y0;          y1 = _y1;
    myPE = myMPI.myPE;

    // Compute number of real (physical) nodes, and their spacing

    nRealx      = ncell_x + 1;
    nRealy      = ncell_y + 1;
    nReal       = nRealx*nRealy;
    dx          = (x1-x0) / ncell_x;
    dy          = (y1-y0) / ncell_y;

    // Compute the size of the field variables, which also lie on ghost nodes

    nField      = (nRealx+2)*(nRealy+2);

    // Allocate memory -- Note that the node numbers and field variable numbers must
    // match.  So even though we only will be caring about real nodes, their node
    // numbers are naturally ordered, so must be of size nField.

    x.resize(nField+1); y.resize(nField+1); Qval.resize(nField+1);

    for ( int i = 1 ; i <= nRealx ; ++i )
      for ( int j = 1 ; j <= nRealy ; ++j )
   {
     int p = pid(i,j);
     x[p] = x0 + (i-1)*dx;
     y[p] = y0 + (j-1)*dy;
   }

  }



  //  ==
  //  ||
  //  ||  ParticlesOnMesh
  //  ||
  //  ||  Quantify the particles' interaction with the mesh
  //  ||
  //  ||  References:  [1] https://www.particleincell.com/2010/es-pic-method/
  //  ||
  //  ==

  void ParticlesOnMesh(particles &PTCL, mpiInfo &myMPI)
  {
    double hx, hy;
    double w[5];
    int    p[5];
    int    iL, iR, jB, jT;
    int    iPEnew, jPEnew;            // These store the i-j indicies of the processor receiving a particle,
                                       //    if that particle is leaving the mesh.
    VI ptcl_send_list, ptcl_send_PE;  // These collect information about particles that
                                       //    have left this processor and are heading onto
                                       //    another processor.

    // -
    // |
    // | Determine which particles are still on this mesh and which have left
    // |
    // -

    for ( int k = 1 ; k <= PTCL.n ; ++k )
      {
   // First, check to be sure the particle is still in the mesh.  If it is not, set its
   // "active" flag to zero, and note the processor to which it is going for MPI
```

```
     exchange.

    if ( PTCL.active[k] == 1 )
      {
        iPEnew = myMPI.iPE;
        jPEnew = myMPI.jPE;


        if ( PTCL.x[k] < x0     ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE - 1 ; }
        if ( PTCL.x[k] > x1     ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE + 1 ; }
        if ( PTCL.y[k] < y0     ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE - 1 ; }
        if ( PTCL.y[k] > y1     ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE + 1 ; }


      }

    // The particle is not in the mesh.  Collect this particle into a holding array that will
    // be sent to the neighboring processor.

    if ( PTCL.active[k] == -1)
      {
        if ( iPEnew >= 0 && iPEnew < myMPI.nPEx )
        if ( jPEnew >= 0 && jPEnew < myMPI.nPEy )
          {
       ptcl_send_list.push_back(k);
       ptcl_send_PE  .push_back(iPEnew + jPEnew*nRealx);
          }

        PTCL.active[k] = 0;  // Remove it from the list of active particles
      }
       }

     // -
     // |
     // | Give and receive particles to/with other processors
     // |
     // -

     myMPI.ParticleExchange( ptcl_send_list , ptcl_send_PE , PTCL);
     cout << "After Exchange" << endl;

     // -
     // |
     // | Accumulate particles to the nodes (to be completed in the esPIC code, next week)
     // |
     // -

     for ( int k = 1 ; k <= nField ; ++k ) Qval[k] = 0.;

     // -
     // |
     // | Compute forces on particles
     // |
     // -

     for ( int k = 1 ; k <= PTCL.n ; ++k ) PTCL.xf[k] = PTCL.yf[k] = 0.;

     for ( int k = 1 ; k <= PTCL.n ; ++k )
       {
    if ( PTCL.active[k] == 1 )
      {
        PTCL.xf[ k ] = 0.;
        PTCL.yf[ k ] =-.4;
         }
        }


   }

 #include "mesh_plotter.h"

   int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }

 };




 //  ==
 //  ||
 //  ||
 //  ||  Main Program
 //  ||
 //  ||
 //  ==
 //

 int main(int argc, char *argv[])
 {

    mpiInfo myMPI;
    MPI_Init     (&argc         , &argv       );
    MPI_Comm_size(MPI_COMM_WORLD, &myMPI.numPE);
    MPI_Comm_rank(MPI_COMM_WORLD,&myMPI.myPE  );

    int nPEx, nPEy, nCellx, nCelly;
    double tEnd, dt;
    double flux;

    // -
    // |
    // | Banner and Input
    // |
    // -

    if ( myMPI.myPE == 0 )
      {
        cout << "\n";
        cout << "---------------------------------------------\n";
        cout << "\n";
        cout << " F R E E   P A R T I C L E                   \n";
        cout << " D E M O   C O D E                           \n";
        cout << "\n";
        cout << " Running on " << myMPI.numPE << " processors \n";
        cout << "\n";
        cout << "---------------------------------------------\n";
        cout << "\n";
      }

    for (int count =  0 ; count < argc; ++count)
      {
        if ( !strcmp(argv[count],"-nPEx"  ) ) nPEx   = atoi(argv[count+1]);
        if ( !strcmp(argv[count],"-nPEy"  ) ) nPEy   = atoi(argv[count+1]);
        if ( !strcmp(argv[count],"-nCellx") ) nCellx = atoi(argv[count+1]);
        if ( !strcmp(argv[count],"-nCelly") ) nCelly = atoi(argv[count+1]);
        if ( !strcmp(argv[count],"-flux"  ) ) flux   = atof(argv[count+1]);
        if ( !strcmp(argv[count],"-tEnd"  ) ) tEnd   = atof(argv[count+1]);
        if ( !strcmp(argv[count],"-dt"    ) ) dt     = atof(argv[count+1]);
      }

    if ( myMPI.myPE == 0 )
      {
        cout << endl;
        cout << "Input Summary: " << endl;
        cout << "--------------------------- " << endl;
        cout << "No. PE   in  x-direction: " << nPEx   << endl;
        cout << "             y-direction: " << nPEy   << endl;
        cout << "No. Cells in x-direction: " << nCellx << endl;
        cout << "             y-direction: " << nCelly << endl;
        cout << "Flux density            : " << flux << endl;
        cout << "End Time                : " << tEnd   << endl;
        cout << "Time Step               : " << dt     << endl;
        cout << endl;
      }

    // -
    // |
    // | MPI / Processor ID
    // |
    // -

    myMPI.GridDecomposition(nPEx,nPEy,nCellx,nCelly);

    // -
    // |
    // | Parallel Grid Generation
    // |
    // -

    double totalLength = 1.;
    double eachPElength_x = totalLength / nPEx;
    double eachPElength_y = totalLength / nPEy;

    double x0 = eachPElength_x * myMPI.iPE;   double x1 = x0 + eachPElength_x;
    double y0 = eachPElength_y * myMPI.jPE;   double y1 = y0 + eachPElength_y;

    Mesh MESH( x0  , x1 , y0 , y1 , nCellx , nCelly , myMPI );

    // -
    // |
    // | Set up Particles
    // |
    // -

    particles PTCL(500);

    int    count   = 0;
    double density = flux/(MESH.y1-MESH.y0);
    double vx_bdy  = .5;
    double vy_bdy  = .0;

    // -
    // |
    // | Time Marching Loop
    // |
    // -

    for ( double t = 0. ; t <= tEnd ; t += dt )
      {
        cout << endl;
        cout << "myPE: " << myMPI.myPE << " Time = " << t << endl;

        // Inject particles

        if ( myMPI.iPE == 0 ) PTCL.addFlux( t , MESH.y0, MESH.y1, density, vx_bdy , vy_bdy );

        // Move particles

        PTCL.move( dt );

        // Map between particles and the mesh

        MESH.ParticlesOnMesh(PTCL,myMPI);

        // Plot

        PTCL.plot( "ptcl" , count , myMPI.myPE );
        MESH.plot( "mesh" , count , myMPI );

        ++count;
      }

```

```

   // -
   // |
   // | Wrap-Up
   // |
   // -

   if ( myMPI.myPE == 0 ) cout << "\n\n ** Successful Completion ** \n\n";

   MPI_Finalize();

   return 0;

}
```

Listing 7: Testing fp

```
//  ==
//  ||
//  ||   C L A S S:    m p i I n f o
//  ||
//  ==

class mpiInfo
{
 public:

  int myPE;
  int numPE;
  int nRealx,  nRealy;
  int nx,  ny;
  int nPEx, nPEy;
  int iPE , jPE;
  int iMin, iMax, jMin, jMax ; // The global i-j numbers on this processor
  int nei_n, nei_s, nei_e, nei_w;
  int nei_ne, nei_nw, nei_se, nei_sw;

  int countx, county;

  MPI_Status status;
  int        err;
  int         tag;
  MPI_Request request;

  //  -
  //  |
  //  |   GridDecomposition: Set up PE numbering system in figure below and
  //  |                      establish communication arrays.
  //  |
  //  |                      nPEx -- number of PEs in the x-direction
  //  |                      nPEy -- number of PEs in the y-direction
  //  |                      numPE = total number of PEs
  //  |
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       |       |       |         |       |
  //  |                       |       |       |         | numPE |
  //  |                       |       |       |         |       |
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       .       .       .         .       .
  //  |                       .       .       .         .       .
  //  |                       .       .       .         .       .
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       |       |       |         |       |
  //  |                       | nPEx  | nPEx+1|         |       |
  //  |                       |       |       |         |       |
  //  |                       +-------+-------+ . . .   +-------+
  //  |                       |       |       |         |       |
  //  |                       |   0   |   1   |         | nPEx-1|
  //  |                       |       |       |         |       |
  //  |                       +-------+-------+ . . .   +-------+
  //  |
  //  |
  //  -


  void GridDecomposition(int _nPEx, int _nPEy, int nCellx , int nCelly)
  {

    nRealx = nCellx + 1;
    nRealy = nCelly + 1;

    // Store and check incoming processor counts

    nPEx = _nPEx;
    nPEy = _nPEy;

    if (nPEx*nPEy != numPE)
      {
      if ( myPE == 0 ) cout << "Fatal Error:  Number of PEs in x-y directions do not add up to numPE" << endl;
      MPI_Barrier(MPI_COMM_WORLD);
      MPI_Finalize();
      exit(0);
      }

    // Get the i-j location of this processor, given its number.  See figure above:

    jPE = int(myPE/nPEx);
    iPE = myPE - jPE*nPEx;

    // Set neighbor values

    nei_n = nei_s = nei_e = nei_w = -1;

    if ( iPE > 0      )
      {
        nei_w = myPE - 1    ;
      }
    if ( jPE > 0      )
      {
       nei_s = myPE - nPEx ;
      }
    if ( iPE < nPEx-1 )
      {
       nei_e = myPE + 1    ;
      }
    if ( jPE < nPEy-1 )
      {
       nei_n = myPE + nPEx ;
      }

    nei_nw = nei_sw = nei_ne = nei_se = -1;

    if ( iPE > 0      && jPE > 0      )  nei_sw = myPE - nPEx - 1 ;
    if ( iPE < nPEx-1 && jPE > 0      )  nei_se = myPE - nPEx + 1 ;
    if ( iPE > 0      && jPE < nPEy-1 )  nei_nw = myPE + nPEx - 1 ;
    if ( iPE < nPEx-1 && jPE < nPEy-1 )  nei_ne = myPE + nPEx + 1 ;

    // Acquire memory for the communication between adjacent processors:
    countx = nRealx + 2;
    county = nRealy + 2;

    tag = 0;
  }



  //  ==
  //  ||
  //  ||  ParticlesExchange
  //  ||
  //  ||  Exchange particles between processors
  //  ||
  //  ||
  //  ==

  void ParticleExchange( VI &ptcl_send_list , VI &ptcl_send_PE , particles &PTCL)
  {
    MPI_Status  status;
    MPI_Request request;

    // (1) Get the max number particles to be sent by any particular processor,
    and make sure all processors  know that number.

    int numToSend = ptcl_send_list.size();      int maxToSend;
    cout << "Before Reduce" << endl;
    /* TO-DO Place MPI call here to complete (1) */
    MPI_Barrier(MPI_COMM_WORLD);
    MPI_Iallreduce(&numToSend, &maxToSend, 1, MPI_INT, MPI_MAX, MPI_COMM_WORLD, &
    request); MPI_Wait(&request,&status);
    MPI_Barrier(MPI_COMM_WORLD);
    cout << "After Reduce, Before C Particles" << endl;

    // (2) Allocate contributions to the upcoming Gather operation.  Here, "C" for
     "Contribution" to be Gathered
```

```

    int    *Cptcl_PE;  Cptcl_PE = new int    [ maxToSend ];  // Particles' destination PEs
    double *Cptcl_x ;  Cptcl_x  = new double [ maxToSend ];
    double *Cptcl_y ;  Cptcl_y  = new double [ maxToSend ];
    double *Cptcl_vx;  Cptcl_vx = new double [ maxToSend ];
    double *Cptcl_vy;  Cptcl_vy = new double [ maxToSend ];

    // (3) Populate contributions on all processors for the upcoming Gather operation

    for ( int i = 0 ; i < maxToSend ; ++i ) { Cptcl_PE[i] = -1; Cptcl_x [i] = 0.; Cptcl_y [i] = 0.; Cptcl_vx[i] = 0.; Cptcl_vy[i] = 0.; }


    // (4) Populate with all the particles on this PE.  Note that some/most processors will have left-over space in the C* arrays.

    for ( int i = 0 ; i < ptcl_send_list.size() ; ++i )
      {
   int id      = ptcl_send_list[ i ];
   Cptcl_PE[i] = ptcl_send_PE  [ i ];
   Cptcl_x [i] = PTCL.x        [ id ];
   Cptcl_y [i] = PTCL.y        [ id ];
   Cptcl_vx[i] = PTCL.vx       [ id ];
   Cptcl_vy[i] = PTCL.vy       [ id ];
      }
    cout << "After C Particles" << endl;

    // (5) Allocate and initialize the arrays for upcoming Gather operation to PE0.  The sizeOfGather takes
    //     into account the number of processors, like this figure:
    //
    //     |<----------------------------- sizeOfGather ------------------------------>|
    //     |                                                                            |
    //     |                                                                            |
    //     |<- maxToSend    ->|<- maxToSend    ->|<- maxToSend    ->|<- maxToSend    ->|
    //     +------------------+------------------+------------------+------------------+
    //             PE0                PE1                 PE2               PE3
    cout << "Before Gather Particles" << endl;

    int sizeOfGather = maxToSend * numPE;

    int    *Gptcl_PE;  Gptcl_PE = new int    [  sizeOfGather ];
    double *Gptcl_x ;  Gptcl_x  = new double [  sizeOfGather ];
    double *Gptcl_y ;  Gptcl_y  = new double [  sizeOfGather ];
    double *Gptcl_vx;  Gptcl_vx = new double [  sizeOfGather ];
    double *Gptcl_vy;  Gptcl_vy = new double [  sizeOfGather ];

    for ( int i = 0 ; i < sizeOfGather ; ++i ) { Gptcl_PE[i] = -1; Gptcl_x [i] =
   0.; Gptcl_y [i] = 0.; Gptcl_vx[i] = 0.; Gptcl_vy[i] = 0.;  }
    cout << "After Gather Particles" << endl;

    // (6)  Gather "Contributions" ("C" arrays) from all PEs onto all PEs into these bigger arrays so all PE will know what particles
    //      need to go where.

    cout << "Before Gather" << endl;
    MPI_Barrier(MPI_COMM_WORLD);

    MPI_Iallgather( Cptcl_PE, sizeOfGather + 1, MPI_INT, Gptcl_PE, maxToSend + 1, MPI_INT, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_x, sizeOfGather + 1, MPI_INT, Gptcl_x, maxToSend + 1, MPI_INT, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_y, sizeOfGather + 1, MPI_INT, Gptcl_y, maxToSend + 1, MPI_INT, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_vx, sizeOfGather + 1, MPI_INT, Gptcl_vx, maxToSend + 1, MPI_INT, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_vy, sizeOfGather + 1, MPI_INT, Gptcl_vy, maxToSend + 1, MPI_INT, MPI_COMM_WORLD, &request );  MPI_Wait(&request,&status);


    cout << "After Gather" << endl;
    MPI_Barrier(MPI_COMM_WORLD);

    // (7) Put in vector form so they can be added to PTCL.  These arrays are 1-based.

    cout << "Before Adds" << endl;

    int Np = 0;  for ( int i = 0 ; i < sizeOfGather ; ++i ) if ( Gptcl_PE[i] == myPE ) ++Np;

    VD  std_add_x  ;  std_add_x.resize  ( Np+1 );
    VD  std_add_y  ;  std_add_y.resize  ( Np+1 );
    VD  std_add_vx ;  std_add_vx.resize ( Np+1 );
    VD  std_add_vy ;  std_add_vy.resize ( Np+1 );

    cout << "After adds, Before Assignment" << endl;
    int count = 1;
    for ( int i = 0 ; i < sizeOfGather ; ++i )
      if ( Gptcl_PE[i] == myPE )
   {
     std_add_x [ count ] = Gptcl_x[i];
     std_add_y [ count ] = Gptcl_y [i];
     std_add_vx[ count ] = Gptcl_vx[i];
     std_add_vy[ count ] = Gptcl_vy[i];
     ++ count;
   }

    cout << "Before Particle Add" << endl;

    PTCL.add( std_add_x, std_add_y, std_add_vx, std_add_vy );

    cout << "After Particle Add" << endl;
```

```

    // (8) Free up memory

    if (maxToSend    > 0 ) { delete[] Cptcl_PE;  delete[] Cptcl_x ;  delete[] Cptcl_y ; delete[] Cptcl_vx ; delete[] Cptcl_vy;  }
    if (sizeOfGather > 0 ) { delete[] Gptcl_PE;  delete[] Gptcl_x ;  delete[] Gptcl_y ; delete[] Gptcl_vx ; delete[] Gptcl_vy;  }

  }


  int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }

};
```

Listing 8: Debugging mpiInfo

In my attempts to debug, I had inserted multiple print statements to see where it would stop. It would most often either stop after the function or during the gather stage. Tweaks in sizes (sizeToGather, maxToSend, adding values here and there) would all throw malloc errors, just different ones each time. Midway through I started just running it locally with oversubscribing to have it more clear in front of me, as the use of slurm would output the error section in an ex01-(number).out executable while putting my cout print statements into tty.out, thus separating them and messing with my head.

4.7 Debugging Output

4.7.1 The Wonderful World of Malloc Errors

When I say there were a lot of different malloc errors, I mean I rarely got the same error twice, both when changing sizes and not changing anything. I deleted so many, but here are some I have to show this (just because I forgot to delete them).

```
Lmod has detected the following error: These module(s) or extension(s) exist
but cannot be loaded as requested: "impi"
   Try: "module spider impi" to see how to load the module(s).




==
||
|| Begin Execution of fp in slurm batch script.
||
==
free(): invalid next size (fast)
[c3cpu-a5-u34-3:1351127] *** Process received signal ***
[c3cpu-a5-u34-3:1351127] Signal: Aborted (6)
[c3cpu-a5-u34-3:1351127] Signal code:  (-6)
[c3cpu-a5-u34-3:1351127] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x15303eb32ce0]
[c3cpu-a5-u34-3:1351127] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x15303e7a9a9f]
[c3cpu-a5-u34-3:1351127] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x15303e77ce05]
[c3cpu-a5-u34-3:1351127] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x15303e7ec037]
[c3cpu-a5-u34-3:1351127] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x15303e7f319c]
[c3cpu-a5-u34-3:1351127] [ 5] /usr/lib64/libc.so.6(+0x99b38)[0x15303e7f4b38]
[c3cpu-a5-u34-3:1351127] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x404acf]
```

```
[c3cpu-a5-u34-3:1351127] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051eb]
[c3cpu-a5-u34-3:1351127] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1351127] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x15303e795cf3]
[c3cpu-a5-u34-3:1351127] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1351127] *** End of error message ***
free(): invalid next size (fast)
[c3cpu-a5-u34-3:1351124] *** Process received signal ***
[c3cpu-a5-u34-3:1351124] Signal: Aborted (6)
[c3cpu-a5-u34-3:1351124] Signal code:  (-6)
[c3cpu-a5-u34-3:1351124] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14731d7c1ce0]
[c3cpu-a5-u34-3:1351124] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14731d438a9f]
[c3cpu-a5-u34-3:1351124] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14731d40be05]
[c3cpu-a5-u34-3:1351124] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x14731d47b037]
[c3cpu-a5-u34-3:1351124] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x14731d48219c]
[c3cpu-a5-u34-3:1351124] [ 5] /usr/lib64/libc.so.6(+0x99b38)[0x14731d483b38]
[c3cpu-a5-u34-3:1351124] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x404b08]
[c3cpu-a5-u34-3:1351124] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051eb]
[c3cpu-a5-u34-3:1351124] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1351124] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x14731d424cf3]
[c3cpu-a5-u34-3:1351124] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1351124] *** End of error message ***
free(): invalid next size (fast)
[c3cpu-a5-u34-3:1351125] *** Process received signal ***
[c3cpu-a5-u34-3:1351125] Signal: Aborted (6)
[c3cpu-a5-u34-3:1351125] Signal code:  (-6)
[c3cpu-a5-u34-3:1351125] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x150719137ce0]
[c3cpu-a5-u34-3:1351125] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x150718daea9f]
[c3cpu-a5-u34-3:1351125] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x150718d81e05]
[c3cpu-a5-u34-3:1351125] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x150718df1037]
[c3cpu-a5-u34-3:1351125] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x150718df819c]
[c3cpu-a5-u34-3:1351125] [ 5] /usr/lib64/libc.so.6(+0x99b38)[0x150718df9b38]
[c3cpu-a5-u34-3:1351125] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x404acf]
[c3cpu-a5-u34-3:1351125] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051eb]
[c3cpu-a5-u34-3:1351125] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1351125] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x150718d9acf3]
[c3cpu-a5-u34-3:1351125] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1351125] *** End of error message ***
free(): invalid next size (fast)
[c3cpu-a5-u34-3:1351126] *** Process received signal ***
[c3cpu-a5-u34-3:1351126] Signal: Aborted (6)
[c3cpu-a5-u34-3:1351126] Signal code:  (-6)
```

```
[c3cpu-a5-u34-3:1351126] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14ebdfb6ace0]
[c3cpu-a5-u34-3:1351126] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14ebdf7e1a9f]
[c3cpu-a5-u34-3:1351126] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14ebdf7b4e05]
[c3cpu-a5-u34-3:1351126] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x14ebdf824037]
[c3cpu-a5-u34-3:1351126] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x14ebdf82b19c]
[c3cpu-a5-u34-3:1351126] [ 5] /usr/lib64/libc.so.6(+0x99b38)[0x14ebdf82cb38]
[c3cpu-a5-u34-3:1351126] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x404acf]
[c3cpu-a5-u34-3:1351126] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051eb]
[c3cpu-a5-u34-3:1351126] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1351126] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x14ebdf7cdcf3]
[c3cpu-a5-u34-3:1351126] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1351126] *** End of error message ***
--------------------------------------------------------------------------
Primary job  terminated normally, but 1 process returned
a non-zero exit code. Per user-direction, the job has been aborted.
--------------------------------------------------------------------------
--------------------------------------------------------------------------
mpirun noticed that process rank 2 with PID 1351126 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).
--------------------------------------------------------------------------
==
||
|| Execution of fp in slurm batch script complete.
||
==
```

Listing 9: Next Size Error

```
Lmod has detected the following error: These module(s) or extension(s) exist
but cannot be loaded as requested: "impi"
   Try: "module spider impi" to see how to load the module(s).



==
||
|| Begin Execution of fp in slurm batch script.
||
==
malloc(): mismatching next->prev_size (unsorted)
[c3cpu-a5-u34-3:1336616] *** Process received signal ***
[c3cpu-a5-u34-3:1336616] Signal: Aborted (6)
[c3cpu-a5-u34-3:1336616] Signal code:  (-6)
[c3cpu-a5-u34-3:1336616] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14c3178a2ce0]
[c3cpu-a5-u34-3:1336616] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14c317519a9f]
[c3cpu-a5-u34-3:1336616] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14c3174ece05]
[c3cpu-a5-u34-3:1336616] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x14c31755c037]
[c3cpu-a5-u34-3:1336616] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x14c31756319c]
[c3cpu-a5-u34-3:1336616] [ 5] /usr/lib64/libc.so.6(+0x9b3a4)[0x14c3175663a4]
[c3cpu-a5-u34-3:1336616] [ 6] /usr/lib64/libc.so.6(__libc_calloc+0x86)[0x14c317568486]
```

```
[c3cpu-a5-u34-3:1336616] [ 7] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(+0x1043b0)[0x14c3185683b0]
[c3cpu-a5-u34-3:1336616] [ 8] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(+0x108de9)[0x14c31856cde9]
[c3cpu-a5-u34-3:1336616] [ 9] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(ompi_coll_libnbc_iallgather+0x1a)[0x14c31856d35a]
[c3cpu-a5-u34-3:1336616] [10] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(PMPI_Iallgather+0x105)[0x14c3184f9585]
[c3cpu-a5-u34-3:1336616] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x40466e]
[c3cpu-a5-u34-3:1336616] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051f9]
[c3cpu-a5-u34-3:1336616] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1336616] [14] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x14c317505cf3]
[c3cpu-a5-u34-3:1336616] [15] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1336616] *** End of error message ***
malloc(): unsorted double linked list corrupted
[c3cpu-a5-u34-3:1336615] *** Process received signal ***
[c3cpu-a5-u34-3:1336615] Signal: Aborted (6)
[c3cpu-a5-u34-3:1336615] Signal code:  (-6)
[c3cpu-a5-u34-3:1336615] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x1514dfb49ce0]
[c3cpu-a5-u34-3:1336615] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x1514df7c0a9f]
[c3cpu-a5-u34-3:1336615] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x1514df793e05]
[c3cpu-a5-u34-3:1336615] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x1514df803037]
[c3cpu-a5-u34-3:1336615] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x1514df80a19c]
[c3cpu-a5-u34-3:1336615] [ 5] /usr/lib64/libc.so.6(+0x9b1cc)[0x1514df80d1cc]
[c3cpu-a5-u34-3:1336615] [ 6] /usr/lib64/libc.so.6(__libc_calloc+0x86)[0x1514df80f486]
[c3cpu-a5-u34-3:1336615] [ 7] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(+0x1043b0)[0x1514e080f3b0]
[c3cpu-a5-u34-3:1336615] [ 8] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(+0x108de9)[0x1514e0813de9]
[c3cpu-a5-u34-3:1336615] [ 9] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(ompi_coll_libnbc_iallgather+0x1a)[0x1514e081435a]
[c3cpu-a5-u34-3:1336615] [10] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(PMPI_Iallgather+0x105)[0x1514e07a0585]
[c3cpu-a5-u34-3:1336615] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x40466e]
[c3cpu-a5-u34-3:1336615] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051f9]
[c3cpu-a5-u34-3:1336615] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1336615] [14] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x1514df7accf3]
[c3cpu-a5-u34-3:1336615] [15] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1336615] *** End of error message ***
--------------------------------------------------------------------------
Primary job  terminated normally, but 1 process returned
a non-zero exit code. Per user-direction, the job has been aborted.
--------------------------------------------------------------------------
--------------------------------------------------------------------------
```

```
mpirun noticed that process rank 3 with PID 1336616 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).
--------------------------------------------------------------------------
==
||
|| Execution of fp in slurm batch script complete.
||
==
```

Listing 10: Next to Previous Size Error

```
Lmod has detected the following error: These module(s) or extension(s) exist
but cannot be loaded as requested: "impi"
   Try: "module spider impi" to see how to load the module(s).



==
||
|| Begin Execution of fp in slurm batch script.
||
==
realloc(): invalid pointer
[c3cpu-a5-u34-3:1350630] *** Process received signal ***
[c3cpu-a5-u34-3:1350630] Signal: Aborted (6)
[c3cpu-a5-u34-3:1350630] Signal code:  (-6)
[c3cpu-a5-u34-3:1350630] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x145b3d09cce0]
[c3cpu-a5-u34-3:1350630] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x145b3cd13a9f]
[c3cpu-a5-u34-3:1350630] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x145b3cce6e05]
[c3cpu-a5-u34-3:1350630] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x145b3cd56037]
[c3cpu-a5-u34-3:1350630] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x145b3cd5d19c]
[c3cpu-a5-u34-3:1350630] [ 5] /usr/lib64/libc.so.6(realloc+0x23a)[0x145b3cd6212a]
[c3cpu-a5-u34-3:1350630] [ 6] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(NBC_Sched_recv+0xfb)[0x145b3dd6279b]
[c3cpu-a5-u34-3:1350630] [ 7] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(+0x109053)[0x145b3dd67053]
[c3cpu-a5-u34-3:1350630] [ 8] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(ompi_coll_libnbc_iallgather+0x1a)[0x145b3dd6735a]
[c3cpu-a5-u34-3:1350630] [ 9] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(PMPI_Iallgather+0x105)[0x145b3dcf3585]
[c3cpu-a5-u34-3:1350630] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x40466d]
[c3cpu-a5-u34-3:1350630] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4051eb]
[c3cpu-a5-u34-3:1350630] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1350630] [13] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x145b3ccffcf3]
[c3cpu-a5-u34-3:1350630] [14] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1350630] *** End of error message ***
--------------------------------------------------------------------------
Primary job  terminated normally, but 1 process returned
a non-zero exit code. Per user-direction, the job has been aborted.
--------------------------------------------------------------------------
--------------------------------------------------------------------------
```

```
mpirun noticed that process rank 2 with PID 1350630 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).
--------------------------------------------------------------------------
==
||
|| Execution of fp in slurm batch script complete.
||
==
```

Listing 11: Invalid Pointer

```
Lmod has detected the following error: These module(s) or extension(s) exist
but cannot be loaded as requested: "impi"
   Try: "module spider impi" to see how to load the module(s).



==
||
|| Begin Execution of fp in slurm batch script.
||
==
fp: malloc.c:2396: sysmalloc: Assertion `(old_top == initial_top (av) && old_size == 0) || ((unsigned long) (old_size) >= MINSIZE && prev_inuse (old_top) && ((unsigned long) old_end & (pagesize - 1)) == 0)' failed.
[c3cpu-a5-u34-3:1362180] *** Process received signal ***
[c3cpu-a5-u34-3:1362180] Signal: Aborted (6)
[c3cpu-a5-u34-3:1362180] Signal code:  (-6)
[c3cpu-a5-u34-3:1362180] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14ea34afcce0]
[c3cpu-a5-u34-3:1362180] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14ea34773a9f]
[c3cpu-a5-u34-3:1362180] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14ea34746e05]
[c3cpu-a5-u34-3:1362180] [ 3] /usr/lib64/libc.so.6(+0x9817a)[0x14ea347bd17a]
[c3cpu-a5-u34-3:1362180] [ 4] /usr/lib64/libc.so.6(+0x9a8e8)[0x14ea347bf8e8]
[c3cpu-a5-u34-3:1362180] [ 5] /usr/lib64/libc.so.6(+0x9b659)[0x14ea347c0659]
[c3cpu-a5-u34-3:1362180] [ 6] /usr/lib64/libc.so.6(+0x9bf5f)[0x14ea347c0f5f]
[c3cpu-a5-u34-3:1362180] [ 7] /usr/lib64/libc.so.6(realloc+0x33b)[0x14ea347c222b]
[c3cpu-a5-u34-3:1362180] [ 8] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(NBC_Sched_recv+0xfb)[0x14ea357c279b]
[c3cpu-a5-u34-3:1362180] [ 9] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(+0x109053)[0x14ea357c7053]
[c3cpu-a5-u34-3:1362180] [10] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(ompi_coll_libnbc_iallgather+0x1a)[0x14ea357c735a]
[c3cpu-a5-u34-3:1362180] [11] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libmpi.so.40(PMPI_Iallgather+0x105)[0x14ea35753585]
[c3cpu-a5-u34-3:1362180] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x404713]
[c3cpu-a5-u34-3:1362180] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x405319]
[c3cpu-a5-u34-3:1362180] [14] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x402c78]
[c3cpu-a5-u34-3:1362180] [15] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0x14ea3475fcf3]
[c3cpu-a5-u34-3:1362180] [16] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0x4023be]
[c3cpu-a5-u34-3:1362180] *** End of error message ***
--------------------------------------------------------------------------
```

```
Primary job  terminated normally, but 1 process returned
a non-zero exit code. Per user-direction, the job has been aborted.
--------------------------------------------------------------------------
--------------------------------------------------------------------------
mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).
--------------------------------------------------------------------------
==
||
|| Execution of fp in slurm batch script complete.
||
==
```

Listing 12: Old Top

```
Lmod has detected the following error: These module(s) or extension(s) exist
but cannot be loaded as requested: "impi"
   Try: "module spider impi" to see how to load the module(s).



==
||
|| Begin Execution of fp in slurm batch script.
||
==
[c3cpu-a5-u34-3:1352023:0:1352023] Caught signal 11 (Segmentation fault: address not mapped to object at address 0x100)
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
==== backtrace (tid:1352023) ====
 0 0x0000000000012ce0 __funlockfile()  :0
 1 0x0000000000203ec6 mca_pml_ucx_isend()  ???:0
 2 0x00000000001055f1 NBC_Start_round()  nbc.c:0
 3 0x0000000000105ab7 NBC_Start()  ???:0
 4 0x000000000010936a ompi_coll_libnbc_iallgather()  ???:0
 5 0x0000000000095585 PMPI_Iallgather()  ???:0
 6 0x00000000004046be mpiInfo::ParticleExchange()  ???:0
 7 0x00000000004051eb Mesh::ParticlesOnMesh()  ???:0
 8 0x0000000000402c78 main()  ???:0
 9 0x000000000003acf3 __libc_start_main()  ???:0
10 0x00000000004023be _start()  ???:0
=================================
--------------------------------------------------------------------------
Primary job  terminated normally, but 1 process returned
a non-zero exit code. Per user-direction, the job has been aborted.
--------------------------------------------------------------------------
--------------------------------------------------------------------------
mpirun noticed that process rank 2 with PID 1352023 on node c3cpu-a5-u34-3 exited on signal 11 (Segmentation fault).
```

```
--------------------------------------------------------------------------
==
||
|| Execution of fp in slurm batch script complete.
||
==
```

Listing 13: dwarf error

These are just some of the wide array of errors I got, all of which showing issues with memory allocation.

4.7.2 Trying to Find the Where and The Why

```

myPE: 3 Time = 0
Before Reduce

myPE: 1 Time = 0
Before Reduce

myPE: 2 Time = 0
Before Reduce

---------------------------------------------

 F R E E   P A R T I C L E
 D E M O   C O D E

 Running on 4 processors

---------------------------------------------


Input Summary:
---------------------------
No. PE   in  x-direction: 2
             y-direction: 2
No. Cells in x-direction: 5
             y-direction: 5
Flux density            : 10
End Time                : 3
Time Step               : 0.01


myPE: 0 Time = 0
Before Reduce
After Reduce, Before C Particles
After C Particles
Before Gather Particles
After Gather Particles
Before Gather
After Gather
Before Adds
After adds, Before Assignment
```

```
Before Particle Add
After Particle Add
After Exchange
After Reduce, Before C Particles
After C Particles
Before Gather Particles
After Gather Particles
Before Gather
After Gather
Before Adds
After adds, Before Assignment
Before Particle Add
After Particle Add
After Exchange
After Reduce, Before C Particles
After C Particles
Before Gather Particles
After Gather Particles
Before Gather
After Gather
Before Adds
After adds, Before Assignment
Before Particle Add
After Particle Add
After Exchange
After Reduce, Before C Particles
After C Particles
Before Gather Particles
After Gather Particles
Before Gather
After Gather
Before Adds
After adds, Before Assignment
Before Particle Add
After Particle Add
After Exchange
```

...

```
myPE: 3 Time = 0.55
Before Reduce

myPE: 1 Time = 0.55
Before Reduce

myPE: 0 Time = 0.55
Before Reduce
After Reduce, Before C Particles
After C Particles
Before Gather Particles
After Gather Particles
Before Gather
After Reduce, Before C Particles
After C Particles
Before Gather Particles
After Gather Particles
```

```
18 Before Gather
19
20 myPE: 2 Time = 0.55
21 Before Reduce
22 After Reduce, Before C Particles
23 After C Particles
24 Before Gather Particles
25 After Gather Particles
26 Before Gather
27 After Reduce, Before C Particles
28 After C Particles
29 Before Gather Particles
30 After Gather Particles
31 Before Gather
```

Listing 14: tty.out

I removed lines 78-3102 there, otherwise this would make the document more than 80 pages. I feel this is enough to get the point across anyway.

This definitely looks better as a single blob in the command line, but I could not make that look good here. This would lead into one of the errors above (and more).

```
==12158== Memcheck, a memory error detector
==12158== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==12158== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==12158== Command: mpirun --oversubscribe -np 4 ./fp -nPEx 2 -nPEy 2 -nCellx 5 -nCelly 5 -flux 10 -tEnd 3 -dt 0.01
==12158== Parent PID: 2865
==12158==
==12158== Warning: noted but unhandled ioctl 0x5441 with no size/direction hints.
==12158==    This could cause spurious value errors to appear.
==12158==    See README_MISSING_SYSCALL_OR_IOCTL for guidance on writing a proper wrapper.
==12162== Warning: invalid file descriptor 1048564 in syscall close()
==12162== Warning: invalid file descriptor 1048565 in syscall close()
==12162== Warning: invalid file descriptor 1048566 in syscall close()
==12162== Warning: invalid file descriptor 1048567 in syscall close()
==12162==    Use --log-fd=<number> to select an alternative log fd.
==12162== Warning: invalid file descriptor 1048568 in syscall close()
==12162== Warning: invalid file descriptor 1048569 in syscall close()
==12163== Warning: invalid file descriptor 1048564 in syscall close()
==12163== Warning: invalid file descriptor 1048565 in syscall close()
==12163== Warning: invalid file descriptor 1048566 in syscall close()
==12163== Warning: invalid file descriptor 1048567 in syscall close()
==12163==    Use --log-fd=<number> to select an alternative log fd.
==12163== Warning: invalid file descriptor 1048568 in syscall close()
==12163== Warning: invalid file descriptor 1048569 in syscall close()
==12164== Warning: invalid file descriptor 1048564 in syscall close()
==12164== Warning: invalid file descriptor 1048565 in syscall close()
==12164== Warning: invalid file descriptor 1048566 in syscall close()
==12164== Warning: invalid file descriptor 1048567 in syscall close()
==12164==    Use --log-fd=<number> to select an alternative log fd.
==12164== Warning: invalid file descriptor 1048568 in syscall close()
==12164== Warning: invalid file descriptor 1048569 in syscall close()
==12165== Warning: invalid file descriptor 1048564 in syscall close()
==12165== Warning: invalid file descriptor 1048565 in syscall close()
```

```
==12165== Warning: invalid file descriptor 1048566 in syscall close()
==12165== Warning: invalid file descriptor 1048567 in syscall close()
==12165==    Use --log-fd=<number> to select an alternative log fd.
==12165== Warning: invalid file descriptor 1048568 in syscall close()
==12165== Warning: invalid file descriptor 1048569 in syscall close()
==12158==
==12158== HEAP SUMMARY:
==12158==     in use at exit: 20,396 bytes in 128 blocks
==12158==   total heap usage: 27,478 allocs, 27,350 frees, 12,255,873 bytes allocated
==12158==
==12158== 32 bytes in 1 blocks are definitely lost in loss record 24 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
==12158==    by 0x524FE46: opal_hwloc_base_get_npus (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x4F0E29A: orte_rmaps_rr_byobj (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F0D031: orte_rmaps_rr_map (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F04559: orte_rmaps_base_map_job (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/bin/orterun)
==12158==    by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
==12158==
==12158== 136 bytes in 1 blocks are definitely lost in loss record 45 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
==12158==    by 0x4F09449: orte_rmaps_base_print_mapping (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4EA1F77: orte_pmix_server_register_nspace (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED5AA9: orte_odls_base_default_construct_child_list (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED9D01: orte_odls_default_launch_local_procs (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/bin/orterun)
==12158==    by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
==12158==
==12158== 321 (136 direct, 185 indirect) bytes in 1 blocks are definitely lost in loss record 53 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
==12158==    by 0x4E817B2: show_help (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4E81AE6: orte_show_help_recv (in /curc/sw/install/openmpi/4.1.1/
gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/bin/orterun)
==12158==    by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
==12158==
==12158== 569 (64 direct, 505 indirect) bytes in 1 blocks are definitely lost in loss record 62 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
==12158==    by 0x4F204CF: assign (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F1EB91: orte_rtc_base_assign (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED5E2B: orte_odls_base_default_construct_child_list (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED9D01: orte_odls_default_launch_local_procs (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/bin/orterun)
==12158==    by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
==12158==
==12158== 962 (96 direct, 866 indirect) bytes in 2 blocks are definitely lost in loss record 67 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
==12158==    by 0x532DCD0: new_tracker.constprop.0 (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x53340F3: pmix_server_fence (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x532417C: server_switchyard (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x5359892: pmix_ptl_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x530BD8D: progress_engine (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
==12158==    by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
==12158==
==12158== 2,048 bytes in 1 blocks are definitely lost in loss record 70 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
```

```
==12158==    by 0x51D888A: opal_dss_buffer_extend (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x51DAAD0: opal_dss_pack_int32 (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x51DAB45: opal_dss_pack (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x4EA2BBB: orte_pmix_server_register_nspace (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED5AA9: orte_odls_base_default_construct_child_list (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED9D01: orte_odls_default_launch_local_procs (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/bin/orterun)
==12158==
==12158== 2,048 bytes in 1 blocks are definitely lost in loss record 71 of 73
==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
==12158==    by 0x51D888A: opal_dss_buffer_extend (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x51DAAD0: opal_dss_pack_int32 (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x51DAB45: opal_dss_pack (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x4EA2CB8: orte_pmix_server_register_nspace (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED5AA9: orte_odls_base_default_construct_child_list (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4ED9D01: orte_odls_default_launch_local_procs (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/bin/orterun)
==12158==
==12158== 2,436 (1,656 direct, 780 indirect) bytes in 3 blocks are definitely lost in loss record 72 of 73
==12158==    at 0x4C3BE4B: calloc (vg_replace_malloc.c:1328)
==12158==    by 0x533C45C: pmix_server_log (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x53250B8: server_switchyard (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
```

```
==12158==    by 0x5359892: pmix_ptl_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x530BD8D: progress_engine (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
==12158==    by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
==12158==
==12158== 3,090 (1,656 direct, 1,434 indirect) bytes in 3 blocks are definitely lost in loss record 73 of 73
==12158==    at 0x4C3BE4B: calloc (vg_replace_malloc.c:1328)
==12158==    by 0x533C0D8: pmix_server_log (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x53250B8: server_switchyard (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x5359892: pmix_ptl_base_process_msg (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so.6.0.2)
==12158==    by 0x530BD8D: progress_engine (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
==12158==    by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
==12158==    by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
==12158==
==12158== LEAK SUMMARY:
==12158==    definitely lost: 7,872 bytes in 14 blocks
==12158==    indirectly lost: 3,770 bytes in 28 blocks
==12158==      possibly lost: 0 bytes in 0 blocks
==12158==    still reachable: 8,754 bytes in 86 blocks
==12158==         suppressed: 0 bytes in 0 blocks
==12158== Reachable blocks (those to which a pointer was found) are not shown.
==12158== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==12158==
==12158== For lists of detected and suppressed errors, rerun with: -s
==12158== ERROR SUMMARY: 9 errors from 9 contexts (suppressed: 0 from 0)
```

Listing 15: Valgrind

I would later get the idea to look into programs to see the memory like Valgrind, as the errors I was getting were almost all malloc errors (besides the occasional ”forgot a semicolon” or ”you did not define this right” errors). I tried multiple configurations of this, but it never pointed to anything directly from our code. It is very possible I missed an argument I could have passed in or it does say something that I am just not understanding, but could not glean anything about our code from the Valgrind output. I feel like I could go deeper down this rabbit hole, but I just do not have any more time to explore this before the submission date. Heck, I am attending class online right now instead of in person so I can at least get this done before work (as well as pain stuff). It would be rude to just type away at this while he is trying to teach.