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

```
1 // ==
2 // 11
         CLASS: mpiInfo
3 // ||
4 // ||
5 // ==
7 class mpiInfo
8 {
  public:
9
10
    int myPE;
11
12
    int numPE;
    int nRealx, nRealy;
13
14
    int nx, ny;
    int nPEx, nPEy;
15
    int iPE , jPE;
16
    int iMin, iMax, jMin, jMax; // The global i-j numbers on this processor
17
    int nei_n, nei_s, nei_e, nei_w;
18
19
    int nei_ne, nei_nw, nei_se, nei_sw;
20
21
    int countx, county;
22
    MPI_Status status;
23
    int err;
24
    int
25
              tag;
    MPI_Request request;
26
27
    // -
28
    // |
29
    // |
           GridDecomposition: Set up PE numbering system in figure below and
30
    // |
                            establish communication arrays.
31
    // |
    // |
                            nPEx -- number of PEs in the x-direction
    // |
                            nPEy -- number of PEs in the y-direction
34
                            numPE = total number of PEs
    // |
35
    // |
36
    //
37
                               | |
                                                    1 1
    //
38
    //
40
    //
41
    //
    //
42
    11
43
    //
44
45
    // |
    // |
47
    // |
                             | nPEx | nPEx+1|
   // |
                             48
   // |
49
                                                    1 1
   // |
                               1 1
50
   // |
                               0 | 1 |
                                                    | nPEx -1|
51
    // |
                                           52
                                    - 1
   //
                              +----+
```

```
// |
54
     // |
     //
56
57
58
     void GridDecomposition(int _nPEx, int _nPEy, int nCellx , int nCelly)
59
60
61
       nRealx = nCellx + 1;
62
       nRealy = nCelly + 1;
63
64
65
       // Store and check incoming processor counts
66
       nPEx = _nPEx;
67
       nPEy = _nPEy;
68
69
       if (nPEx*nPEy != numPE)
70
71
        {
         if ( myPE == 0 ) cout << "Fatal Error: Number of PEs in x-y directions do
72
      not add up to numPE" << endl;
         MPI_Barrier(MPI_COMM_WORLD);
73
        MPI_Finalize();
74
         exit(0);
75
76
77
       // Get the i-j location of this processor, given its number. See figure above
78
79
       jPE = int(myPE/nPEx);
80
       iPE = myPE - jPE*nPEx;
81
82
       // Set neighbor values
83
84
       nei_n = nei_s = nei_e = nei_w = -1;
85
86
       if ( iPE > 0
87
88
           nei_w = myPE - 1
89
90
       if ( jPE > 0
91
        {
92
         nei_s = myPE - nPEx ;
93
94
       if ( iPE < nPEx-1 )
95
         {
96
         nei_e = myPE + 1
97
98
       if ( jPE < nPEy-1 )
99
100
101
         nei_n = myPE + nPEx ;
102
103
       nei_nw = nei_sw = nei_ne = nei_se = -1;
104
       106
```

```
if ( iPE < nPEx-1 && jPE > 0 ) nei_se = myPE - nPEx + 1 ;
107
                          && jPE < nPEy-1 ) nei_nw = myPE + nPEx - 1;
       if ( iPE > 0
108
       if ( iPE < nPEx-1 && jPE < nPEy-1 ) nei_ne = myPE + nPEx + 1 ;
109
110
       // Acquire memory for the communication between adjacent processors:
111
       countx = nRealx + 2;
112
       county = nRealy + 2;
113
114
       tag = 0;
115
116
117
118
119
     //
        ==
120
     //
         - 11
     //
        - 11
             ParticlesExchange
122
123
     //
         - 11
     //
        - 11
              Exchange particles between processors
     //
        - 11
125
     // 11
126
     // ==
127
128
     void ParticleExchange( VI &ptcl_send_list , VI &ptcl_send_PE , particles &PTCL)
129
130
       MPI_Status status;
132
       MPI_Request request;
133
       // (1) Get the max number particles to be sent by any particular processor,
134
       and make sure all processors know that number.
135
       int numToSend = ptcl_send_list.size();
                                                      int maxToSend;
136
137
       /* TO-DO Place MPI call here to complete (1) */
138
       MPI_Barrier(MPI_COMM_WORLD);
139
       MPI_Iallreduce(&numToSend, &maxToSend, 1, MPI_INT, MPI_MAX, MPI_COMM_WORLD, &
140
       request); MPI_Wait(&request,&status);
       MPI_Barrier(MPI_COMM_WORLD);
141
142
       // (2) Allocate contributions to the upcoming Gather operation. Here, "C" for
143
        "Contribution" to be Gathered
144
              *Cptcl_PE; Cptcl_PE = new int
                                                   [ maxToSend ]; // Particles'
       int
145
       destination PEs
       double *Cptcl_x ;
                           Cptcl_x = new double [ maxToSend ];
146
       double *Cptcl_y ; Cptcl_y = new double [ maxToSend ];
147
       double *Cptcl_vx; Cptcl_vx = new double [ maxToSend ];
148
       double *Cptcl_vy; Cptcl_vy = new double [ maxToSend ];
149
       // (3) Populate contributions on all processors for the upcoming Gather
151
       operation
       for ( int i = 0 ; i < maxToSend ; ++i ) { Cptcl_PE[i] = -1; Cptcl_x [i] = 0.;
153
       Cptcl_y [i] = 0.; Cptcl_vx[i] = 0.; Cptcl_vy[i] = 0.; }
155
```

```
// (4) Populate with all the particles on this PE. Note that some/most
156
      processors will have left-over space in the C* arrays.
157
       for ( int i = 0 ; i < ptcl_send_list.size() ; ++i )</pre>
158
159
        {
      int id
                = ptcl_send_list[ i ];
160
      Cptcl_PE[i] = ptcl_send_PE [ i ];
161
      Cptcl_x [i] = PTCL.x
                            [ id ];
162
      Cptcl_y [i] = PTCL.y
                                [ id ];
163
      Cptcl_vx[i] = PTCL.vx
                                [ id ];
164
      Cptcl_vy[i] = PTCL.vy
                                 [ id ];
165
166
        }
167
       // (5) Allocate and initialize the arrays for upcoming Gather operation to PEO
168
      . The sizeOfGather takes
       11
             into account the number of processors, like this figure:
169
       //
170
             |<---- sizeOfGather</pre>
       //
171
       ----->|
            //
173
             |<- maxToSend ->|<- maxToSend ->|<- maxToSend</pre>
                                                                  ->|<- maxToSend
174
       //
      +----+
                     PE0
       //
                                      PE1
                                                         PE2
                                                                           PE3
176
177
178
       int sizeOfGather = maxToSend * numPE;
179
180
             *Gptcl_PE; Gptcl_PE = new int
                                              [ sizeOfGather ];
181
       double *Gptcl_x ; Gptcl_x = new double [ sizeOfGather ];
182
       double *Gptcl_y ; Gptcl_y = new double [ sizeOfGather ];
183
       double *Gptcl_vx; Gptcl_vx = new double [ sizeOfGather ];
184
       double *Gptcl_vy; Gptcl_vy = new double [ sizeOfGather ];
186
       for ( int i = 0 ; i < sizeOfGather ; ++i ) { Gptcl_PE[i] = -1; Gptcl_x [i] = -1
187
      0.; Gptcl_y [i] = 0.; Gptcl_vx[i] = 0.; Gptcl_vy[i] = 0.; }
188
189
       // (6) Gather "Contributions" ("C" arrays) from all PEs onto all PEs into
190
      these bigger arrays so all PE will know what particles
              need to go where.
191
192
      MPI_Barrier(MPI_COMM_WORLD);
193
194
      MPI_Iallgather( Cptcl_PE, maxToSend, MPI_INT, Gptcl_PE, maxToSend, MPI_INT,
195
      MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
      MPI_Iallgather( Cptcl_x, maxToSend, MPI_DOUBLE, Gptcl_x, maxToSend, MPI_DOUBLE
196
       , MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
       MPI_Iallgather( Cptcl_y, maxToSend, MPI_DOUBLE, Gptcl_y, maxToSend, MPI_DOUBLE
197
      , MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
      MPI_Iallgather( Cptcl_vx, maxToSend, MPI_DOUBLE, Gptcl_vx, maxToSend,
198
```

```
MPI_DOUBLE, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
       MPI_Iallgather( Cptcl_vy, maxToSend, MPI_DOUBLE, Gptcl_vy, maxToSend,
199
       MPI_DOUBLE, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
200
201
       MPI_Barrier(MPI_COMM_WORLD);
202
203
       // (7) Put in vector form so they can be added to PTCL. These arrays are 1-
204
       based.
205
       int Np = 0; for ( int i = 0 ; i < sizeOfGather ; ++i ) if ( Gptcl_PE[i] ==</pre>
       myPE ) ++Np;
207
       ۷D
           std_add_x ; std_add_x.resize
                                             (Np+1);
208
       ۷D
                      ; std_add_y.resize
                                             ( Np+1 );
           std_add_y
209
       ۷D
           std_add_vx ; std_add_vx.resize ( Np+1 );
211
          std_add_vy ; std_add_vy.resize ( Np+1 );
212
       int count = 1;
213
       for ( int i = 0 ; i < sizeOfGather ; ++i )</pre>
214
         if ( Gptcl_PE[i] == myPE )
215
216
        std_add_x [ count ] = Gptcl_x[i];
217
        std_add_y [ count ] = Gptcl_y [i];
218
219
        std_add_vx[ count ] = Gptcl_vx[i];
        std_add_vy[ count ] = Gptcl_vy[i];
220
        ++ count;
221
222
223
       PTCL.add( std_add_x, std_add_y, std_add_vx, std_add_vy);
224
225
       // (8) Free up memory
226
227
       if (maxToSend
                        > 0 ) { delete[] Cptcl_PE; delete[] Cptcl_x ;
                                                                            delete[]
228
       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; }
230
231
232
233
     int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }
234
235
```

Listing 1: Parallel Code (mpiInfo.h)

```
5 // ||
                     FREE PARTICLE
     -11
6 // ||
     -11
7 // 11
                     DEMONSTRATION CODE
     -11
8 // ||
     -11
9 // 11
      \Box
10 // ||
               Developed by: Scott R. Runnels, Ph.D.
      \Pi
11 // ||
                            University of Colorado Boulder
      \Pi
12 // ||
      \Pi
                       For: CU Boulder CSCI 4576/5576 and associated labs
13 // ||
     \Box
14 // ||
     \Box
15 // ||
                  Copyright 2020 Scott Runnels
     -11
16 // ||
     -11
17 // ||
                            Not for distribution or use outside of the
      \Pi
18 //
                           this course.
     - 11
     \Box
19 //
     - 11
     -11
20 //
21
#include "mpi.h"
#include "fp.h"
24 #include "particles.h"
  #include "mpiInfo.h"
27
28 // ==
29 // ||
           CLASS: MESH
30 // ||
31 // ||
         (21)----(22)----(23)----(24)-----(25) <--- nRealx+1 , nRealy+1
35 // ||
         - 1
              1 1
                              1
                                        36 // 11
         37 //
                 - 1
          (16) ----[17] ----[18] ----[19] -----(20)
38 // ||
         | 1,3| 2,3| 3,3|
39 // ||
40 // ||
         - 1
                 - 1
                         - 1
41 // ||
         42 // || (11) ----[12] ----[13] ----[14] -----(15)
```

```
43 // || | 1,2| 2,2| 3,2|
               44 // 11
                       45 // ||
                 - 1
                          46 // ||
        (6) ----[7] ----[8] ----[9] ----(10)
47 // ||
        | 1,1| 2,1| 3,1|
48 // ||
                        49 // ||
50 // || (1) ----(2) ----(3) ----(4) -----(5)
51 // || 0,0
52 // ||
53 // ==
55
  class Mesh
  {
56
57
58 public:
59
    double x0, x1, y0, y1;
60
61
    VD x,y;
    int nRealx , nRealy , nField, nReal;
62
    double lengthx, lengthy, dx, dy;
63
    int myPE;
64
    VD Qval;
65
66
    // ==
67
    // 11
68
    // ||
            Constructor: Initialize values
69
    // 11
70
    // ==
71
72
    Mesh(double _x0 , double _x1, double _y0, double _y1 , int ncell_x , int
73
     ncell_y, mpiInfo &myMPI )
74
75
76
      // Copy incoming values
77
78
79
      x0 = _x0;
                           x1 = _x1;
      y0 = _y0;
                           y1 = _y1;
80
      myPE = myMPI.myPE;
81
82
      // Compute number of real (physical) nodes, and their spacing
83
84
      nRealx
                  = ncell_x + 1;
85
      nRealy
                  = ncell_y + 1;
86
      nReal
                  = nRealx*nRealy;
87
                  = (x1-x0) / ncell_x;
88
                  = (y1-y0) / ncell_y;
      dу
89
90
      // Compute the size of the field variables, which also lie on ghost nodes
91
92
      nField
                  = (nRealx+2)*(nRealy+2);
93
94
      // Allocate memory -- Note that the node numbers and field variable numbers
95
      must
```

```
// match. So even though we only will be caring about real nodes, their node
96
       // numbers are naturally ordered, so must be of size nField.
97
98
       x.resize(nField+1); y.resize(nField+1); Qval.resize(nField+1);
99
100
       for ( int i = 1 ; i <= nRealx ; ++i )
101
          for ( int j = 1 ; j \le nRealy ; ++j )
103
         int p = pid(i,j);
104
        x[p] = x0 + (i-1)*dx;
105
        y[p] = y0 + (j-1)*dy;
106
107
108
     }
109
111
112
     //
         ==
113
     //
         -11
114
              ParticlesOnMesh
115
     //
         -11
     11
         \Box
116
     //
         \Box
              Quantify the particles' interaction with the mesh
117
         \Box
118
     //
119
     //
         -11
              References: [1] https://www.particleincell.com/2010/es-pic-method/
120
     //
         -11
     //
121
122
     void ParticlesOnMesh(particles &PTCL, mpiInfo &myMPI)
123
124
       double hx, hy;
125
       double w[5];
126
               p[5];
127
       int
        int
               iL, iR, jB, jT;
128
       int
               iPEnew, jPEnew;
                                             // These store the i-j indicies of the
129
       processor receiving a particle,
                                                   if that particle is leaving the mesh.
130
                                            // These collect information about particles
       VI ptcl_send_list, ptcl_send_PE;
        that
                                             //
                                                   have left this processor and are
132
       heading onto
                                             //
                                                   another processor.
134
       // -
135
136
       // | Determine which particles are still on this mesh and which have left
137
       // |
138
       // -
139
140
       for ( int k = 1 ; k <= PTCL.n ; ++k )
141
142
       // First, check to be sure the particle is still in the mesh. If it is not,
143
       // "active" flag to zero, and note the processor to which it is going for MPI
144
       exchange.
145
```

```
if ( PTCL.active[k] == 1 )
146
147
           iPEnew = myMPI.iPE;
148
           jPEnew = myMPI.jPE;
149
150
151
           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;
153
       }
           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;
156
157
        }
158
159
      // The particle is not in the mesh. Collect this particle into a holding array
160
        that will
      // be sent to the neighboring processor.
161
162
      if ( PTCL.active[k] == -1)
163
        {
164
           if ( iPEnew >= 0 && iPEnew < myMPI.nPEx )</pre>
165
           if ( jPEnew >= 0 && jPEnew < myMPI.nPEy )
166
167
          ptcl_send_list.push_back(k);
          ptcl_send_PE .push_back( iPEnew + jPEnew * myMPI.nPEx);
169
170
171
          PTCL.active[k] = 0; // Remove it from the list of active particles
172
         }
173
         }
174
175
       // -
176
        // |
177
        // | Give and receive particles to/with other processors
178
        // |
179
       // -
180
181
       myMPI.ParticleExchange( ptcl_send_list , ptcl_send_PE , PTCL);
182
183
       // -
184
       // |
185
       // | Accumulate particles to the nodes (to be completed in the esPIC code,
186
       next week)
       // |
187
       // -
188
189
       for ( int k = 1 ; k <= nField ; ++k ) Qval[k] = 0.;</pre>
190
191
        // -
192
        // |
193
       // | Compute forces on particles
194
```

```
// |
195
        // -
196
197
        for ( int k = 1 ; k \le PTCL.n ; ++k ) PTCL.xf[k] = PTCL.yf[k] = 0.;
198
199
        for ( int k = 1 ; k \le PTCL.n ; ++k )
200
201
       if ( PTCL.active[k] == 1 )
202
         {
203
           PTCL.xf[k] = 0.;
204
           PTCL.yf[ k ] =-.4;
205
206
207
208
209
210
211
   #include "mesh_plotter.h"
212
213
      int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }
214
215
   };
216
217
218
219
220
221
222 //
        ==
223 //
        \Box
        \Box
224 //
225 //
       Main Program
226 //
       -11
227 //
       -11
228 //
229 //
230
   int main(int argc, char *argv[])
231
232
233
       mpiInfo myMPI;
234
       MPI_Init
                     (&argc
                                      , &argv
235
       MPI_Comm_size(MPI_COMM_WORLD, &myMPI.numPE);
236
       MPI_Comm_rank(MPI_COMM_WORLD,&myMPI.myPE );
237
238
239
       int nPEx, nPEy, nCellx, nCelly;
       double tEnd, dt;
240
       double flux;
241
242
       // -
243
       // |
244
       // | Banner and Input
245
       // |
246
       // -
247
248
       if ( myMPI.myPE == 0 )
249
```

```
{
250
         cout << "\n";
251
         cout << "----\n";
252
         cout << "\n";
253
         cout << "FREE PARTICLE
                                                               \n";
254
         cout << " D E M O C O D E
                                                               \n";
255
         cout << "\n";
256
         cout << " Running on " << myMPI.numPE << " processors \n";</pre>
257
         cout << "\n";
258
          cout << "----\n";
259
          cout << "\n";
260
        }
261
262
      for (int count = 0; count < argc; ++count)</pre>
263
264
          if ( !strcmp(argv[count],"-nPEx" ) ) nPEx = atoi(argv[count+1]);
265
          if ( !strcmp(argv[count],"-nPEy" ) ) nPEy = atoi(argv[count+1]);
266
          if ( !strcmp(argv[count],"-nCellx") ) nCellx = atoi(argv[count+1]);
267
          if ( !strcmp(argv[count],"-nCelly") ) nCelly = atoi(argv[count+1]);
268
          if ( !strcmp(argv[count],"-flux" ) ) flux = atof(argv[count+1]);
269
          if ( !strcmp(argv[count],"-tEnd" ) ) tEnd = atof(argv[count+1]);
270
          if (!strcmp(argv[count],"-dt" ) ) dt
                                                    = atof(argv[count+1]);
271
272
273
      if ( myMPI.myPE == 0 )
274
275
         cout << endl;</pre>
276
          cout << "Input Summary: " << endl;</pre>
277
          cout << "----- " << endl;
278
         cout << "No. PE in x-direction: " << nPEx << endl;</pre>
279
         cout << " y-direction: " << nPEy << endl;</pre>
280
         cout << "No. Cells in x-direction: " << nCellx << endl;</pre>
281
         cout << " y-direction: " << nCelly << endl;</pre>
282
                                    : " << flux << endl;
         cout << "Flux density
283
                                         : " << tEnd << endl;
         cout << "End Time
284
         cout << "Time Step
                                          : " << dt
                                                        << endl;
285
          cout << endl;</pre>
286
        }
287
288
      // -
289
      // |
290
      // | MPI / Processor ID
291
      // |
292
      // -
293
294
      myMPI.GridDecomposition(nPEx,nPEy,nCellx,nCelly);
295
296
      // -
297
      // |
298
      // | Parallel Grid Generation
299
      // |
300
      // -
301
302
      double totalLength = 1.;
303
      double eachPElength_x = totalLength / nPEx;
304
```

```
double eachPElength_y = totalLength / nPEy;
305
306
       double x0 = eachPElength_x * myMPI.iPE;
                                                     double x1 = x0 + eachPElength_x;
307
       double y0 = eachPElength_y * myMPI.jPE;
                                                     double y1 = y0 + eachPElength_y;
308
309
       Mesh MESH( x0 , x1 , y0 , y1 , nCellx , nCelly , myMPI );
310
311
       // -
312
       // |
313
       // | Set up Particles
314
       // |
315
       // -
316
317
       particles PTCL(500);
318
319
                      = 0;
              count
       int
320
       double density = flux/(MESH.y1-MESH.y0);
321
       double vx_bdy = .5;
322
       double vy_bdy = .0;
323
324
       // -
325
       // |
326
       // | Time Marching Loop
327
       // |
328
       // -
330
       for ( double t = 0.; t \le tEnd; t += dt)
331
332
           cout << endl;</pre>
333
           cout << "myPE: " << myMPI.myPE << " Time = " << t << endl;</pre>
334
335
           // Inject particles
336
337
           if ( myMPI.iPE == 0 ) PTCL.addFlux( t , MESH.y0, MESH.y1, density, vx_bdy ,
338
        vy_bdy );
339
           // Move particles
340
341
           PTCL.move( dt );
342
343
           // Map between particles and the mesh
344
345
           MESH.ParticlesOnMesh(PTCL,myMPI);
346
347
           // Plot
348
349
           PTCL.plot( "ptcl" , count , myMPI.myPE );
350
           MESH.plot( "mesh" , count , myMPI );
351
352
353
           ++count;
         }
354
355
356
       // -
357
       // |
358
```

```
// | Wrap-Up
359
       // |
360
       // -
361
362
       if ( myMPI.myPE == 0 ) cout << "\n\n ** Successful Completion ** \n\n";
363
364
       MPI_Finalize();
365
366
       return 0;
367
368
369
   }
```

Listing 2: Parallel Code (fp.cpp)

```
#!/bin/bash
3
4 # |
5 # | This is a batch script for running a MPI parallel job on Summit
7 # | (o) To submit this job, enter: sbatch --export=CODE='/home/scru5660/HPSC/
      codes/fd_mpi/src' ex_01.bat
9 # | (o) To check the status of this job, enter: squeue -u <username>
10 # |
11 # -
12
13 # -
14 # |
# | Part 1: Directives
16 # |
17 # -
18
19 #SBATCH --nodes=1
20 #SBATCH --ntasks=4
21 #SBATCH --time=00:01:00
22 #SBATCH --partition=amilan
#SBATCH --output=ex01-%j.out
24
25 # -
26 # |
27 # | Part 2: Loading software
28 # |
29 # -
30
31 module purge
32 module load intel
33 module load impi
35 # -
36 # |
37 # | Part 3: User scripting
38 # |
39 # -
41 echo "=="
```

Listing 3: Slurm File (ex01.bat)

4 Appendix B: Output

4.1 Output of the Slurm

```
FREE PARTICLE
4
  DEMO CODE
5
   Running on 4 processors
  _____
9
10
11
12 Input Summary:
13
No. PE in x-direction: 2
   y-direction: 2
15
  No. Cells in x-direction: 5
17
     y-direction: 5
Flux density : 10
19 End Time
                     : 5
  Time Step
                     : 0.05
20
21
22
  myPE: O Time = O
23
24
  myPE: 1 Time = 0
25
26
  myPE: 2 Time = 0
27
28
  myPE: 3 Time = 0
30
myPE: 0 Time = 0.05
32
myPE: 3 Time = 0.05
myPE: 1 Time = 0.05
```

```
36
  myPE: 2 Time = 0.05
37
38
39 myPE: 0 Time = 0.1
40
  myPE: 3 Time = 0.1
41
42
  myPE: 1 Time = 0.1
43
44
   myPE: 2 Time = 0.1
45
   myPE: O Time = 0.15
47
48
  myPE: 2 Time = 0.15
49
50
  myPE: 3 Time = 0.15
51
53 myPE: 1 Time = 0.15
54
55 myPE: 0 Time = 0.2
56
  myPE: 1 Time = 0.2
57
58
  myPE: 3 Time = 0.2
60
  myPE: 2 Time = 0.2
61
62
  myPE: O Time = 0.25
63
64
  myPE: 3 Time = 0.25
65
66
  myPE: 1 Time = 0.25
67
68
  myPE: 2 Time = 0.25
69
70
  myPE: O Time = 0.3
71
72
   myPE: 2 Time = 0.3
73
74
   myPE: 3 Time = 0.3
75
76
  myPE: 1 Time = 0.3
77
78
  myPE: O Time = 0.35
79
80
  myPE: 3 Time = 0.35
81
82
myPE: 1 Time = 0.35
84
  myPE: 2 Time = 0.35
85
86
  myPE: 1 Time = 0.4
87
88
  myPE: O Time = 0.4
89
90
```

```
myPE: 3 Time = 0.4
91
92
93 myPE: 2 Time = 0.4
   myPE: 1 Time = 0.45
95
96
   myPE: 2 Time = 0.45
97
98
   myPE: O Time = 0.45
99
100
   myPE: 3 Time = 0.45
101
102
   myPE: 2 Time = 0.5
103
104
   myPE: 3 Time = 0.5
105
106
   myPE: O Time = 0.5
108
   myPE: 1 Time = 0.5
109
110
   myPE: 3 Time = 0.55
111
112
   myPE: 2 Time = 0.55
113
114
   myPE: 1 Time = 0.55
115
116
   myPE: O Time = 0.55
117
118
   myPE: 3 Time = 0.6
119
120
   myPE: 2 Time = 0.6
121
122
   myPE: 1 Time = 0.6
123
124
   myPE: O Time = 0.6
125
126
   myPE: 3 Time = 0.65
127
128
   myPE: 2 Time = 0.65
129
130
   myPE: O Time = 0.65
131
132
   myPE: 1 Time = 0.65
133
134
myPE: 2 Time = 0.7
136
   myPE: 3 Time = 0.7
137
138
   myPE: O Time = 0.7
139
140
   myPE: 1 Time = 0.7
141
142
   myPE: 3 Time = 0.75
143
144
myPE: 2 Time = 0.75
```

```
146
   myPE: O Time = 0.75
147
148
myPE: 1 Time = 0.75
150
   myPE: 3 Time = 0.8
151
152
   myPE: 2 Time = 0.8
153
154
   myPE: 1 Time = 0.8
155
   myPE: 0 Time = 0.8
157
158
   myPE: 3 Time = 0.85
159
160
   myPE: 2 Time = 0.85
161
163 myPE: 1 Time = 0.85
164
165 \text{ myPE}: 0 \text{ Time} = 0.85
166
   myPE: 2 Time = 0.9
167
168
   myPE: 3 Time = 0.9
169
170
   myPE: 1 Time = 0.9
171
172
   myPE: O Time = 0.9
173
174
   myPE: 3 Time = 0.95
175
176
   myPE: O Time = 0.95
177
178
   myPE: 2 Time = 0.95
179
180
   myPE: 1 Time = 0.95
181
   myPE: 3 Time = 1
183
184
   myPE: 1 Time = 1
185
186
   myPE: 0 Time = 1
187
188
   myPE: 2 Time = 1
189
190
   myPE: 3 Time = 1.05
191
192
   myPE: 1 Time = 1.05
193
194
   myPE: 0 Time = 1.05
195
196
   myPE: 2 Time = 1.05
197
198
   myPE: 3 Time = 1.1
199
200
```

```
myPE: 1 Time = 1.1
201
202
   myPE: O Time = 1.1
203
    myPE: 2 Time = 1.1
205
206
   myPE: 3 Time = 1.15
207
208
    myPE: 1 Time = 1.15
209
210
    myPE: 0 Time = 1.15
211
212
    myPE: 2 Time = 1.15
213
214
   myPE: 3 Time = 1.2
215
216
217 myPE: 2 Time = 1.2
218
219 myPE: 0 Time = 1.2
220
221 \text{ myPE: } 1 \text{ Time = } 1.2
222
   myPE: 3 Time = 1.25
223
224
    myPE: 2 Time = 1.25
225
226
   myPE: 0 Time = 1.25
227
228
   myPE: 1 Time = 1.25
229
230
   myPE: 3 Time = 1.3
231
232
   myPE: 2 Time = 1.3
233
234
   myPE: 0 Time = 1.3
235
236
    myPE: 1 Time = 1.3
237
    myPE: 3 Time = 1.35
239
240
   myPE: 1 Time = 1.35
241
242
243 myPE: 0 Time = 1.35
244
245 myPE: 2 Time = 1.35
246
247 \text{ myPE}: 1 \text{ Time} = 1.4
248
249 myPE: 2 Time = 1.4
250
   myPE: O Time = 1.4
251
253 \text{ myPE}: 3 \text{ Time} = 1.4
254
255 \text{ myPE}: 1 \text{ Time} = 1.45
```

```
256
   myPE: 3 Time = 1.45
257
258
259 \text{ myPE: } 0 \text{ Time } = 1.45
260
   myPE: 2 Time = 1.45
261
262
   myPE: 3 Time = 1.5
263
264
   myPE: 1 Time = 1.5
265
    myPE: 0 Time = 1.5
267
268
   myPE: 2 Time = 1.5
269
270
   myPE: 3 Time = 1.55
271
272
273 myPE: 2 Time = 1.55
274
275 \text{ myPE: } 0 \text{ Time } = 1.55
276
   myPE: 1 Time = 1.55
277
278
   myPE: 3 Time = 1.6
279
280
    myPE: 2 Time = 1.6
281
282
   myPE: O Time = 1.6
283
284
   myPE: 1 Time = 1.6
285
286
   myPE: 3 Time = 1.65
287
288
   myPE: 1 Time = 1.65
289
290
    myPE: O Time = 1.65
291
292
    myPE: 2 Time = 1.65
293
294
   myPE: 3 Time = 1.7
295
296
   myPE: 2 Time = 1.7
297
298
   myPE: O Time = 1.7
299
300
   myPE: 1 Time = 1.7
301
302
   myPE: 3 Time = 1.75
303
304
    myPE: 2 Time = 1.75
305
306
   myPE: 1 Time = 1.75
307
308
   myPE: O Time = 1.75
309
310
```

```
311 myPE: 2 Time = 1.8
312
313 myPE: 1 Time = 1.8
314
315 myPE: 0 Time = 1.8
316
317 myPE: 3 Time = 1.8
318
   myPE: 1 Time = 1.85
319
320
   myPE: 2 Time = 1.85
321
   myPE: 0 Time = 1.85
323
324
   myPE: 3 Time = 1.85
325
326
   myPE: 1 Time = 1.9
327
328
   myPE: 2 Time = 1.9
329
330
   myPE: 0 Time = 1.9
331
332
   myPE: 3 Time = 1.9
333
334
   myPE: 2 Time = 1.95
335
336
   myPE: 1 Time = 1.95
337
338
   myPE: 0 Time = 1.95
339
340
   myPE: 3 Time = 1.95
341
342
   myPE: 2 Time = 2
343
344
   myPE: 3 Time = 2
345
346
   myPE: 0 Time = 2
347
   myPE: 1 Time = 2
349
350
   myPE: 2 Time = 2.05
351
352
353 myPE: 1 Time = 2.05
354
355 \text{ myPE}: 3 \text{ Time} = 2.05
356
357 \text{ myPE}: 0 \text{ Time} = 2.05
358
   myPE: 3 Time = 2.1
359
360
   myPE: 2 Time = 2.1
361
362
   myPE: 1 Time = 2.1
363
364
365 myPE: 0 Time = 2.1
```

```
366
    myPE: 2 Time = 2.15
367
368
   myPE: 1 Time = 2.15
370
   myPE: 0 Time = 2.15
371
372
   myPE: 3 Time = 2.15
373
374
   myPE: 2 Time = 2.2
375
    myPE: 3 Time = 2.2
377
378
    myPE: 0 Time = 2.2
379
380
   myPE: 1 Time = 2.2
381
   myPE: 1 Time = 2.25
383
384
385 \text{ myPE}: 3 \text{ Time} = 2.25
386
   myPE: 2 Time = 2.25
387
388
   myPE: 0 Time = 2.25
389
390
    myPE: 3 Time = 2.3
391
392
    myPE: 2 Time = 2.3
393
394
   myPE: 0 Time = 2.3
395
   myPE: 1 Time = 2.3
397
398
   myPE: 1 Time = 2.35
399
400
   myPE: 3 Time = 2.35
401
402
    myPE: 2 Time = 2.35
403
404
    myPE: 0 Time = 2.35
405
406
   myPE: 2 Time = 2.4
407
408
   myPE: 1 Time = 2.4
409
410
   myPE: 3 Time = 2.4
411
412
_{413} myPE: 0 Time = 2.4
414
   myPE: 1 Time = 2.45
415
416
   myPE: 2 Time = 2.45
417
418
419 \text{ myPE}: 3 \text{ Time} = 2.45
420
```

```
myPE: 0 Time = 2.45
421
422
   myPE: 3 Time = 2.5
423
424
   myPE: 2 Time = 2.5
425
426
   myPE: 1 Time = 2.5
427
428
   myPE: 0 Time = 2.5
429
430
   myPE: 3 Time = 2.55
431
   myPE: 2 Time = 2.55
433
434
   myPE: 0 Time = 2.55
435
436
   myPE: 1 Time = 2.55
437
438
   myPE: 2 Time = 2.6
439
440
   myPE: 0 Time = 2.6
441
442
   myPE: 3 Time = 2.6
443
444
   myPE: 1 Time = 2.6
445
446
   myPE: 2 Time = 2.65
447
448
   myPE: 3 Time = 2.65
449
450
   myPE: 0 Time = 2.65
451
452
   myPE: 1 Time = 2.65
453
454
   myPE: 3 Time = 2.7
455
456
   myPE: 2 Time = 2.7
457
   myPE: 0 Time = 2.7
459
460
   myPE: 1 Time = 2.7
461
462
   myPE: 1 Time = 2.75
463
464
   myPE: 3 Time = 2.75
465
466
   myPE: 0 Time = 2.75
467
468
   myPE: 2 Time = 2.75
469
470
   myPE: 1 Time = 2.8
471
472
   myPE: 3 Time = 2.8
473
474
_{475} myPE: 0 Time = 2.8
```

```
476
   myPE: 2 Time = 2.8
477
478
479 \text{ myPE}: 3 \text{ Time} = 2.85
480
   myPE: 1 Time = 2.85
481
482
   myPE: 0 Time = 2.85
483
484
    myPE: 2 Time = 2.85
485
    myPE: 1 Time = 2.9
487
488
   myPE: 0 Time = 2.9
489
490
   myPE: 3 Time = 2.9
491
492
493 \text{ myPE}: 2 \text{ Time} = 2.9
494
495 \text{ myPE}: 1 \text{ Time} = 2.95
496
   myPE: 3 Time = 2.95
497
498
   myPE: 0 Time = 2.95
499
500
    myPE: 2 Time = 2.95
501
502
   myPE: 0 Time = 3
503
504
   myPE: 3 Time = 3
505
507 myPE: 1 Time = 3
508
   myPE: 2 Time = 3
509
510
   myPE: 0 Time = 3.05
511
512
    myPE: 2 Time = 3.05
513
514
   myPE: 1 Time = 3.05
515
516
517 myPE: 3 Time = 3.05
518
519 myPE: 3 Time = 3.1
520
   myPE: 0 Time = 3.1
521
522
523 myPE: 1 Time = 3.1
524
   myPE: 2 Time = 3.1
525
526
   myPE: 0 Time = 3.15
527
528
529 myPE: 3 Time = 3.15
530
```

```
531 \text{ myPE}: 1 \text{ Time} = 3.15
532
533 myPE: 2 Time = 3.15
   myPE: 0 Time = 3.2
535
536
   myPE: 2 Time = 3.2
537
538
   myPE: 1 Time = 3.2
539
540
    myPE: 3 Time = 3.2
541
    myPE: 0 Time = 3.25
543
544
   myPE: 2 Time = 3.25
545
546
   myPE: 3 Time = 3.25
547
548
549 \text{ myPE}: 1 \text{ Time} = 3.25
550
   myPE: 2 Time = 3.3
551
552
   myPE: O Time = 3.3
553
554
    myPE: 3 Time = 3.3
555
556
   myPE: 1 Time = 3.3
557
558
   myPE: 3 Time = 3.35
559
560
   myPE: 2 Time = 3.35
561
562
   myPE: O Time = 3.35
563
564
   myPE: 1 Time = 3.35
565
566
    myPE: 2 Time = 3.4
567
    myPE: 3 Time = 3.4
569
570
   myPE: 0 Time = 3.4
571
572
573 myPE: 1 Time = 3.4
574
575 \text{ myPE: } 0 \text{ Time } = 3.45
576
577 myPE: 2 Time = 3.45
578
   myPE: 3 Time = 3.45
579
580
   myPE: 1 Time = 3.45
581
582
   myPE: 3 Time = 3.5
583
584
585 myPE: 2 Time = 3.5
```

```
586
   myPE: 0 Time = 3.5
587
588
589 \text{ myPE}: 1 \text{ Time} = 3.5
590
   myPE: 3 Time = 3.55
591
592
   myPE: 1 Time = 3.55
593
594
   myPE: O Time = 3.55
595
   myPE: 2 Time = 3.55
597
598
   myPE: 3 Time = 3.6
599
600
   myPE: 1 Time = 3.6
601
603 myPE: 0 Time = 3.6
604
605 myPE: 2 Time = 3.6
606
   myPE: 3 Time = 3.65
607
608
   myPE: 0 Time = 3.65
609
610
   myPE: 1 Time = 3.65
611
612
   myPE: 2 Time = 3.65
613
614
myPE: 3 Time = 3.7
616
617 myPE: 1 Time = 3.7
618
   myPE: 2 Time = 3.7
619
620
   myPE: O Time = 3.7
621
622
   myPE: 2 Time = 3.75
623
624
   myPE: 1 Time = 3.75
625
626
   myPE: 3 Time = 3.75
627
628
   myPE: 0 Time = 3.75
629
630
   myPE: 2 Time = 3.8
631
632
   myPE: 0 Time = 3.8
633
634
   myPE: 3 Time = 3.8
635
636
   myPE: 1 Time = 3.8
637
638
   myPE: 3 Time = 3.85
639
640
```

```
641 \text{ myPE}: 0 \text{ Time} = 3.85
642
   myPE: 1 Time = 3.85
643
   myPE: 2 Time = 3.85
645
646
   myPE: 3 Time = 3.9
647
648
   myPE: 0 Time = 3.9
649
650
   myPE: 1 Time = 3.9
651
   myPE: 2 Time = 3.9
653
654
   myPE: 3 Time = 3.95
655
656
   myPE: 2 Time = 3.95
657
658
   myPE: 0 Time = 3.95
659
660
   myPE: 1 Time = 3.95
661
662
   myPE: O Time = 4
663
664
   myPE: 2 Time = 4
665
666
   myPE: 3 Time = 4
667
668
   myPE: 1 Time = 4
669
670
   myPE: 2 Time = 4.05
671
672
   myPE: O Time = 4.05
673
674
   myPE: 1 Time = 4.05
675
676
   myPE: 3 Time = 4.05
677
   myPE: 1 Time = 4.1
679
680
   myPE: 2 Time = 4.1
681
682
   myPE: O Time = 4.1
683
684
685 \text{ myPE}: 3 \text{ Time} = 4.1
686
   myPE: 1 Time = 4.15
687
688
   myPE: 2 Time = 4.15
689
690
   myPE: O Time = 4.15
691
692
   myPE: 3 Time = 4.15
693
694
695 myPE: 1 Time = 4.2
```

```
696
    myPE: 2 Time = 4.2
697
698
699 myPE: 0 Time = 4.2
700
    myPE: 3 Time = 4.2
701
702
    myPE: 2 Time = 4.25
703
704
    myPE: O Time = 4.25
705
    myPE: 1 Time = 4.25
707
708
    myPE: 3 Time = 4.25
709
710
myPE: 0 Time = 4.3
712
713 \text{ myPE}: 1 \text{ Time} = 4.3
714
myPE: 2 Time = 4.3
716
717 \text{ myPE}: 3 \text{ Time} = 4.3
718
   myPE: O Time = 4.35
719
720
    myPE: 2 Time = 4.35
721
722
    myPE: 1 Time = 4.35
723
724
myPE: 3 Time = 4.35
726
727 \text{ myPE: } 0 \text{ Time } = 4.4
728
   myPE: 1 Time = 4.4
729
730
    myPE: 2 Time = 4.4
731
732
    myPE: 3 Time = 4.4
733
734
    myPE: O Time = 4.45
735
736
    myPE: 1 Time = 4.45
737
738
myPE: 2 Time = 4.45
740
741 \text{ myPE}: 3 \text{ Time} = 4.45
742
myPE: 2 Time = 4.5
744
    myPE: O Time = 4.5
745
746
747 \text{ myPE}: 1 \text{ Time} = 4.5
748
749 \text{ myPE}: 3 \text{ Time} = 4.5
750
```

```
751 \text{ myPE: } 2 \text{ Time = } 4.55
752
753 myPE: 1 Time = 4.55
   myPE: 3 Time = 4.55
755
756
   myPE: O Time = 4.55
757
758
   myPE: 2 Time = 4.6
759
760
   myPE: 3 Time = 4.6
761
762
   myPE: 1 Time = 4.6
763
764
   myPE: O Time = 4.6
765
766
   myPE: 2 Time = 4.65
768
   myPE: 3 Time = 4.65
769
770
   myPE: O Time = 4.65
771
772
   myPE: 1 Time = 4.65
773
774
   myPE: 2 Time = 4.7
775
776
   myPE: 3 Time = 4.7
777
778
   myPE: 0 Time = 4.7
779
780
   myPE: 1 Time = 4.7
781
782
   myPE: 2 Time = 4.75
783
784
   myPE: 1 Time = 4.75
785
786
   myPE: 3 Time = 4.75
787
   myPE: O Time = 4.75
789
790
   myPE: 2 Time = 4.8
791
792
   myPE: 1 Time = 4.8
793
794
   myPE: 3 Time = 4.8
795
796
   myPE: O Time = 4.8
797
798
   myPE: 2 Time = 4.85
799
800
   myPE: 0 Time = 4.85
801
802
   myPE: 1 Time = 4.85
803
804
805 myPE: 3 Time = 4.85
```

```
806
   myPE: 1 Time = 4.9
807
808
   myPE: 0 Time = 4.9
810
   myPE: 2 Time = 4.9
811
812
   myPE: 3 Time = 4.9
813
814
   myPE: 1 Time = 4.95
815
   myPE: 3 Time = 4.95
817
818
   myPE: O Time = 4.95
819
820
   myPE: 2 Time = 4.95
821
   myPE: 1 Time = 5
823
824
   myPE: 3 Time = 5
825
826
   myPE: 0 Time = 5
827
   myPE: 2 Time = 5
829
830
831
    ** Successful Completion **
832
```

Listing 4: tty.out

4.2 Output from Alpine

```
1 ==
2 ||
3 || Begin Execution of fp in slurm batch script.
4 ||
5 ==
6 ==
7 ||
8 || Execution of fp in slurm batch script complete.
9 ||
10 ==
```

Listing 5: ex01-3062889.out

4.3 Plot Images, Showing Particles Moving

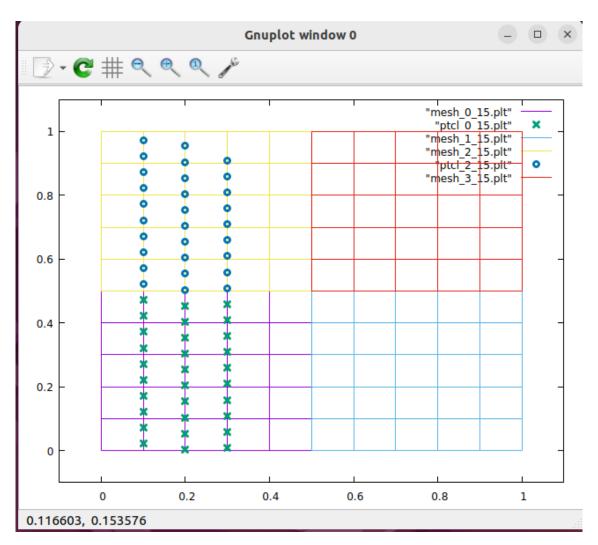


Figure 1: Particle at Start

4.4 Plot Images, Showing Particles Moving

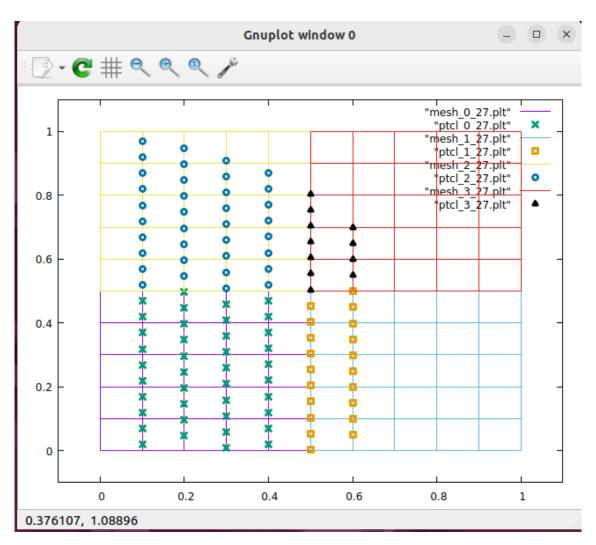


Figure 2: Particles Start Moving to other Processors

4.5 Plot Images, Showing Particles Moving

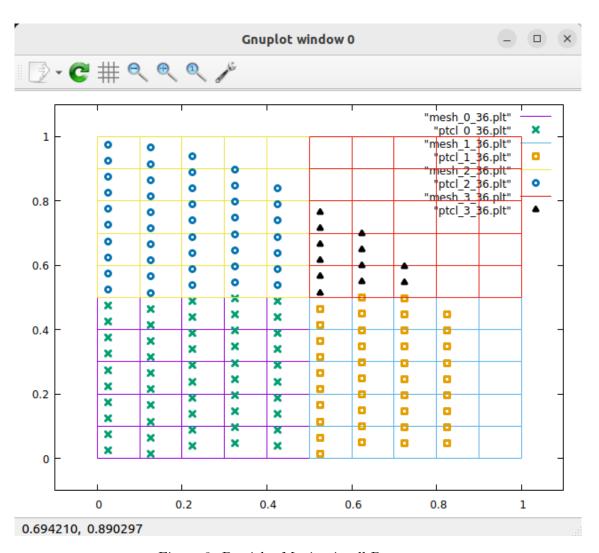


Figure 3: Particles Moving in all Processors

4.6 Debugging Code

```
#!/bin/bash

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.

# ''.
```

```
12
13 # -
14 # |
# | Part 1: Directives
16 # |
17 # -
18
19 #SBATCH --nodes=1
20 #SBATCH --ntasks=4
21 #SBATCH --time=00:02:00
#SBATCH --partition=atesting
#SBATCH --output=ex01-%j.out
24
25 # -
26 # |
27 # | Part 2: Loading software
29 # -
30
31 module purge
32 module update gcc
33 module load gcc
34 module load openmpi
35 module load impi
37 # -
38 # |
39 # | Part 3: User scripting
40 # |
41 # -
42
43 echo "=="
44 echo "||"
45 echo "|| Begin Execution of fp in slurm batch script."
46 echo "||"
  echo "=="
47
48
   mpirun -n 4 $CODE/fp -nPEx 2 -nPEy 2 -nCellx 5 -nCelly 5 -flux 10 -tEnd 3 -dt 0.01
       > tty.out
50
51 echo "=="
52 echo "||"
echo "|| Execution of fp in slurm batch script complete."
54 echo "||"
55 echo "=="
```

Listing 6: Testing Slurm File

```
4 // 11
     -11
5 // ||
                     FREE PARTICLE
     -11
6 // 11
     -11
7 // 11
                      DEMONSTRATION CODE
     \Box
8 // ||
     \Box
9 // 11
10 // ||
               Developed by: Scott R. Runnels, Ph.D.
      \Pi
11 // ||
                            University of Colorado Boulder
     \Box
12 // ||
13 // ||
                        For: CU Boulder CSCI 4576/5576 and associated labs
     -11
14 // ||
     -11
15 // ||
                   Copyright 2020 Scott Runnels
     -11
16 // ||
      \perp
17 // ||
                            Not for distribution or use outside of the
     \Box
18 // ||
                            this course.
     -11
19 // ||
     -11
20 //
#include "mpi.h"
23 #include "fp.h"
24 #include "particles.h"
25 #include "mpiInfo.h"
26
27
28 // ==
29 // 11
           CLASS: MESH
         (21) ----(22) ----(23) ----(24) -----(25) <--- nRealx+1 , nRealy+1
         1 1
35 // ||
                       38 // ||
        (16) ----[17] ----[18] ----[19] -----(20)
39 // ||
        | 1,3| 2,3| 3,3|
40 // ||
```

```
41 // ||
        (11) ----[12] ----[13] ----[14] -----(15)
42 // ||
43 // ||
        | 1,2| 2,2| 3,2|
                                      1
                      44 // ||
        - 1
45 // ||
46 // ||
        (6) ----[7] ----[8] ----[9] ----(10)
47 // 11
        | 1,1| 2,1| 3,1|
                                        48 // ||
        - 1
                       - 1
49 // []
        50 // || (1) ----(2) ----(3) ----(4) -----(5)
51 // || 0,0
52 // ||
53 // ==
54
55 class Mesh
56
58 public:
59
    double x0, x1, y0, y1;
60
    VD x,y;
61
    int nRealx , nRealy , nField, nReal;
62
    double lengthx, lengthy, dx, dy;
63
64
    int myPE;
65
    VD Qval;
66
    // ==
67
    // 11
68
           Constructor: Initialize values
    // 11
69
    // 11
70
    // ==
71
72
    Mesh(double _x0 , double _x1, double _y0, double _y1 , int ncell_x , int
73
     ncell_y, mpiInfo &myMPI )
74
75
76
      // Copy incoming values
77
78
      хO
          = x0;
                          x1 = _x1;
79
      y0 = _y0;
                          y1 = _y1;
80
      myPE = myMPI.myPE;
81
82
      // Compute number of real (physical) nodes, and their spacing
83
84
                 = ncell_x + 1;
85
      nRealx
      nRealy
                 = ncell_y + 1;
86
                 = nRealx*nRealy;
      nReal
87
                 = (x1-x0) / ncell_x;
      dx
88
                 = (y1-y0) / ncell_y;
89
      dу
90
      // Compute the size of the field variables, which also lie on ghost nodes
91
92
      nField
                 = (nRealx+2)*(nRealy+2);
93
94
```

```
// Allocate memory -- Note that the node numbers and field variable numbers
95
       must
       // match. So even though we only will be caring about real nodes, their node
96
97
        // numbers are naturally ordered, so must be of size nField.
98
       x.resize(nField+1); y.resize(nField+1); Qval.resize(nField+1);
99
100
       for ( int i = 1 ; i \le nRealx ; ++i )
101
          for ( int j = 1 ; j \le nRealy ; ++j )
102
103
         int p = pid(i,j);
104
105
         x[p] = x0 + (i-1)*dx;
         y[p] = y0 + (j-1)*dy;
106
107
108
     }
109
110
111
112
     //
         ==
113
     //
         -11
114
     //
         \Box
              ParticlesOnMesh
115
         11
116
     //
117
     //
         -11
              Quantify the particles' interaction with the mesh
118
     //
         \Box
         -11
              References: [1] https://www.particleincell.com/2010/es-pic-method/
119
     //
         -11
     11
120
     //
121
122
     void ParticlesOnMesh(particles &PTCL, mpiInfo &myMPI)
123
124
       double hx, hy;
125
       double w[5];
126
        int
               p[5];
127
               iL, iR, jB, jT;
        int
128
               iPEnew, jPEnew;
                                             // These store the i-j indicies of the
129
        int
       processor receiving a particle,
130
                                                   if that particle is leaving the mesh.
        VI ptcl_send_list, ptcl_send_PE;
                                            // These collect information about particles
131
        that
                                             //
                                                   have left this processor and are
       heading onto
                                             //
                                                   another processor.
133
        // -
135
136
        // | Determine which particles are still on this mesh and which have left
137
        // |
138
        // -
139
140
        for ( int k = 1 ; k \leftarrow PTCL.n ; ++k )
141
142
       // First, check to be sure the particle is still in the mesh. If it is not,
143
       set its
       // "active" flag to zero, and note the processor to which it is going for MPI
144
```

```
exchange.
145
      if ( PTCL.active[k] == 1 )
146
147
           iPEnew = myMPI.iPE;
148
           jPEnew = myMPI.jPE;
149
151
           if ( PTCL.x[k] < x0
                                     ) { PTCL.active[k] = -1; iPEnew = myMPI.iPE - 1;
152
       }
                                     ) { PTCL.active[k] = -1; iPEnew = myMPI.iPE + 1;
           if (PTCL.x[k] > x1
           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;
156
157
        }
158
159
      // The particle is not in the mesh. Collect this particle into a holding array
160
        that will
      // be sent to the neighboring processor.
161
      if ( PTCL.active[k] == -1)
163
        {
164
           if ( iPEnew >= 0 && iPEnew < myMPI.nPEx )</pre>
165
           if ( jPEnew >= 0 && jPEnew < myMPI.nPEy )</pre>
166
167
          ptcl_send_list.push_back(k);
168
          ptcl_send_PE .push_back(iPEnew + jPEnew*nRealx);
169
170
171
          PTCL.active[k] = 0; // Remove it from the list of active particles
172
         }
173
         }
174
175
        // -
176
        // |
177
        // | Give and receive particles to/with other processors
178
        // |
179
       // -
180
181
       myMPI.ParticleExchange( ptcl_send_list , ptcl_send_PE , PTCL);
182
       cout << "After Exchange" << endl;</pre>
183
184
       // -
185
       // |
186
       //\ | Accumulate particles to the nodes (to be completed in the esPIC code,
187
       next week)
       // |
188
       // -
189
190
       for ( int k = 1 ; k \le nField ; ++k ) Qval[k] = 0.;
191
192
```

```
// -
193
        // |
194
        // | Compute forces on particles
195
        // |
196
        // -
197
198
        for ( int k = 1 ; k \le PTCL.n ; ++k ) PTCL.xf[k] = PTCL.yf[k] = 0.;
199
200
        for ( int k = 1 ; k <= PTCL.n ; ++k )
201
202
       if ( PTCL.active[k] == 1 )
203
204
           PTCL.xf[k] = 0.;
205
           PTCL.yf[ k ] =-.4;
206
            }
207
208
209
210
211
212
   #include "mesh_plotter.h"
213
214
      int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }
215
216
217
   };
218
219
220
221
222
223 //
224 //
        \Pi
225 //
        \Pi
226 //
       -11
            Main Program
227 //
       -11
   11
        \Box
228
   11
229
        ==
   //
230
231
   int main(int argc, char *argv[])
232
   {
233
234
       mpiInfo myMPI;
235
                                      , &argv
236
       MPI_Init
                   (&argc
       MPI_Comm_size(MPI_COMM_WORLD, &myMPI.numPE);
237
       MPI_Comm_rank(MPI_COMM_WORLD,&myMPI.myPE );
238
239
240
       int nPEx, nPEy, nCellx, nCelly;
       double tEnd, dt;
241
       double flux;
242
243
       // -
244
       // |
245
       //\ | Banner and Input
246
247
       // |
```

```
// -
248
249
      if ( myMPI.myPE == 0 )
250
251
        {
          cout << "\n";
252
         cout << "----\n";
253
         cout << "\n";
254
         cout << "FREE PARTICLE
                                                                 \n";
255
         cout << " D E M O C O D E
                                                                 \n";
256
          cout << "\n";
257
          cout << " Running on " << myMPI.numPE << " processors \n";</pre>
          cout << "\n";
259
          cout << "----\n";
260
          cout << "\n";
261
262
263
      for (int count = 0; count < argc; ++count)</pre>
264
265
266
          if ( !strcmp(argv[count],"-nPEx" ) ) nPEx = atoi(argv[count+1]);
          if ( !strcmp(argv[count],"-nPEy" ) ) nPEy = atoi(argv[count+1]);
267
          if ( !strcmp(argv[count],"-nCellx") ) nCellx = atoi(argv[count+1]);
268
          if ( !strcmp(argv[count],"-nCelly") ) nCelly = atoi(argv[count+1]);
269
          if ( !strcmp(argv[count],"-flux" ) ) flux = atof(argv[count+1]);
270
          if (!strcmp(argv[count],"-tEnd" ) ) tEnd = atof(argv[count+1]);
if (!strcmp(argv[count],"-dt" ) ) dt = atof(argv[count+1]);
271
                                                      = atof(argv[count+1]);
272
273
274
      if ( myMPI.myPE == 0 )
275
       {
276
277
         cout << endl;</pre>
         cout << "Input Summary: " << endl;</pre>
278
         cout << "----- " << endl;
279
         cout << "No. PE in x-direction: " << nPEx << endl;</pre>
280
         cout << " y-direction: " << nPEy
                                                        << endl;
281
         cout << "No. Cells in x-direction: " << nCellx << endl;</pre>
282
          cout << "
                       y-direction: " << nCelly << endl;
283
          cout << "Flux density : " << flux << endl;</pre>
284
                                           : " << tEnd
          cout << "End Time
285
                                                        << endl;
                                           : " << dt
          cout << "Time Step
                                                         << endl;
286
          cout << endl;</pre>
287
288
289
      // -
290
      // |
      // | MPI / Processor ID
292
      // |
293
      // -
294
295
      myMPI.GridDecomposition(nPEx,nPEy,nCellx,nCelly);
296
297
298
299
      // | Parallel Grid Generation
300
      // |
301
      // -
302
```

```
303
       double totalLength = 1.;
304
       double eachPElength_x = totalLength / nPEx;
305
       double eachPElength_y = totalLength / nPEy;
306
307
       double x0 = eachPElength_x * myMPI.iPE;
                                                     double x1 = x0 + eachPElength_x;
308
       double y0 = eachPElength_y * myMPI.jPE;
                                                     double y1 = y0 + eachPElength_y;
309
310
       Mesh MESH( x0 , x1 , y0 , y1 , nCellx , nCelly , myMPI );
311
312
       // -
313
       // |
314
       // | Set up Particles
315
       // |
316
       // -
317
318
       particles PTCL(500);
319
320
              count
                      = 0;
321
       double density = flux/(MESH.y1-MESH.y0);
322
       double vx_bdy = .5;
323
       double vy_bdy = .0;
324
325
       // -
326
327
       // | Time Marching Loop
328
       // |
329
       // -
330
331
       for ( double t = 0.; t \le tEnd; t += dt)
332
         {
333
           cout << endl;</pre>
334
           cout << "myPE: " << myMPI.myPE << " Time = " << t << endl;</pre>
335
336
           // Inject particles
337
338
           if ( myMPI.iPE == 0 ) PTCL.addFlux( t , MESH.y0, MESH.y1, density, vx_bdy ,
339
        vy_bdy );
340
           // Move particles
341
342
           PTCL.move( dt );
343
344
           // Map between particles and the mesh
345
346
           MESH.ParticlesOnMesh(PTCL,myMPI);
347
348
           // Plot
349
350
           PTCL.plot( "ptcl" , count , myMPI.myPE );
351
           MESH.plot( "mesh" , count , myMPI );
353
           ++count;
354
355
356
```

```
357
       // -
358
       // |
359
       // | Wrap-Up
360
       // |
361
       // -
362
363
       if ( myMPI.myPE == 0 ) cout << "\n\n ** Successful Completion ** \n\n";</pre>
364
365
       MPI_Finalize();
366
367
368
       return 0;
369
370 }
```

Listing 7: Testing fp

```
1 //
2 //
       \Pi
3 //
           CLASS: mpiInfo
      - 11
4 //
      -11
5 // ==
7 class mpiInfo
8 {
   public:
9
10
    int myPE;
11
12
    int numPE;
13
    int nRealx,
                nRealy;
14
    int nx, ny;
    int nPEx, nPEy;
15
    int iPE , jPE;
16
    int iMin, iMax, jMin, jMax; // The global i-j numbers on this processor
17
18
    int nei_n, nei_s, nei_e, nei_w;
19
     int nei_ne, nei_nw, nei_se, nei_sw;
20
    int countx, county;
21
22
    MPI_Status status;
23
    int
24
           err;
25
                tag;
26
    MPI_Request request;
27
    // -
28
    // |
29
     //
             GridDecomposition: Set up PE numbering system in figure below and
30
                                 establish communication arrays.
31
     //
32
    // |
33
    //
                                 nPEx -- number of PEs in the x-direction
    //
                                 nPEy -- number of PEs in the y-direction
34
                                 numPE = total number of PEs
    //
35
    //
36
    //
37
38
    //
39
    //
                                                             | numPE |
```

```
// |
                               40
                               +----+ . . .
    //
41
    //
42
43
    //
    //
44
    //
45
                               1 1 1
    //
46
    //
                               | nPEx | nPEx+1|
47
    //
                                 1 1
48
                               +----+ . . .
    //
49
                                 1 1
50
    11
51
    //
                                                        | nPEx -1|
    //
                                      52
                               +----+ . . .
    //
53
    //
54
    //
55
    // -
56
57
58
    void GridDecomposition(int _nPEx, int _nPEy, int nCellx , int nCelly)
59
60
61
      nRealx = nCellx + 1;
62
63
      nRealy = nCelly + 1;
64
      // Store and check incoming processor counts
65
66
      nPEx = _nPEx;
67
      nPEy = _nPEy;
68
69
      if (nPEx*nPEy != numPE)
70
71
        if ( myPE == 0 ) cout << "Fatal Error: Number of PEs in x-y directions do
72
     not add up to numPE" << endl;</pre>
       MPI_Barrier(MPI_COMM_WORLD);
73
        MPI_Finalize();
74
75
        exit(0);
76
77
      // Get the i-j location of this processor, given its number. See figure above
78
79
      jPE = int(myPE/nPEx);
80
      iPE = myPE - jPE*nPEx;
81
82
      // Set neighbor values
83
84
      nei_n = nei_s = nei_e = nei_w = -1;
85
86
      if ( iPE > 0 )
87
88
         nei_w = myPE - 1;
89
90
      if ( jPE > 0 )
91
92
```

```
nei_s = myPE - nPEx ;
93
94
       if ( iPE < nPEx-1 )
95
         {
96
          nei_e = myPE + 1
97
98
        if ( jPE < nPEy-1 )
99
         {
100
          nei_n = myPE + nPEx ;
101
102
103
104
       nei_nw = nei_sw = nei_ne = nei_se = -1;
       if (iPE > 0
                          && jPE > 0
                                           ) nei_sw = myPE - nPEx - 1 ;
106
       if ( iPE < nPEx-1 && jPE > 0
                                           ) nei_se = myPE - nPEx + 1 ;
                          && jPE < nPEy-1 ) nei_nw = myPE + nPEx - 1 ;
       if ( iPE > 0
108
       if ( iPE < nPEx-1 && jPE < nPEy-1 ) nei_ne = myPE + nPEx + 1 ;
109
110
       // Acquire memory for the communication between adjacent processors:
111
       countx = nRealx + 2;
112
       county = nRealy + 2;
113
114
115
       tag = 0;
     }
116
117
118
119
     // ==
120
     // ||
121
     // 11
             ParticlesExchange
122
     // 11
     // 11
              Exchange particles between processors
124
     // 11
125
     // 11
126
     // ==
127
128
     void ParticleExchange( VI &ptcl_send_list , VI &ptcl_send_PE , particles &PTCL)
129
130
       MPI_Status status;
131
       MPI_Request request;
132
133
       // (1) Get the max number particles to be sent by any particular processor,
134
       and make sure all processors know that number.
135
       int numToSend = ptcl_send_list.size();
                                                      int maxToSend;
136
       cout << "Before Reduce" << endl;</pre>
137
       /* TO-DO Place MPI call here to complete (1) */
138
       MPI_Barrier(MPI_COMM_WORLD);
139
       MPI_Iallreduce(&numToSend, &maxToSend, 1, MPI_INT, MPI_MAX, MPI_COMM_WORLD, &
140
       request); MPI_Wait(&request,&status);
141
       MPI_Barrier(MPI_COMM_WORLD);
       cout << "After Reduce, Before C Particles" << endl;</pre>
142
143
        // (2) Allocate contributions to the upcoming Gather operation. Here, "C" for
144
        "Contribution" to be Gathered
```

```
145
             *Cptcl_PE; Cptcl_PE = new int
                                               [ maxToSend ]; // Particles'
       int
146
      destination PEs
       double *Cptcl_x ; Cptcl_x = new double [ maxToSend ];
147
       double *Cptcl_y ; Cptcl_y = new double [ maxToSend ];
148
       double *Cptcl_vx; Cptcl_vx = new double [ maxToSend ];
149
       double *Cptcl_vy; Cptcl_vy = new double [ maxToSend ];
150
151
       // (3) Populate contributions on all processors for the upcoming Gather
152
      operation
153
       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.; }
155
156
       // (4) Populate with all the particles on this PE. Note that some/most
157
      processors will have left-over space in the C* arrays.
158
       for ( int i = 0 ; i < ptcl_send_list.size() ; ++i )</pre>
159
        {
160
      int id
                = ptcl_send_list[ i ];
161
      Cptcl_PE[i] = ptcl_send_PE [ i ];
162
      Cptcl_x [i] = PTCL.x
                                 [ id ];
163
      Cptcl_y [i] = PTCL.y
                                 [ id ];
164
      Cptcl_vx[i] = PTCL.vx
                                 [ id ];
165
      Cptcl_vy[i] = PTCL.vy
166
                                 [ id ];
167
       cout << "After C Particles" << endl;</pre>
169
       // (5) Allocate and initialize the arrays for upcoming Gather operation to PEO
170
       . The sizeOfGather takes
       //
             into account the number of processors, like this figure:
171
       //
172
       //
            |<---- sizeOfGather</pre>
173
               ---->|
       //
174
       //
             |<- maxToSend ->|<- maxToSend ->|<- maxToSend</pre>
                                                                   ->|<- maxToSend
176
         ->|
       //
177
      +----+
                     PE0
       //
                                       PE1
                                                        PE2
                                                                           PE3
178
       cout << "Before Gather Particles" << endl;</pre>
179
180
       int sizeOfGather = maxToSend * numPE;
181
182
       int
             *Gptcl_PE; Gptcl_PE = new int
                                              [ sizeOfGather ];
183
       double *Gptcl_x ; Gptcl_x = new double [ sizeOfGather ];
184
       double *Gptcl_y ; Gptcl_y = new double [ sizeOfGather ];
                                                 sizeOfGather ];
       double *Gptcl_vx; Gptcl_vx = new double [
186
       double *Gptcl_vy; Gptcl_vy = new double [ sizeOfGather ];
187
188
       for ( int i = 0 ; i < sizeOfGather ; ++i ) { Gptcl_PE[i] = -1; Gptcl_x[i] = -1
189
```

```
0.; Gptcl_y [i] = 0.; Gptcl_vx[i] = 0.; Gptcl_vy[i] = 0.; }
       cout << "After Gather Particles" << endl;</pre>
190
191
       // (6) Gather "Contributions" ("C" arrays) from all PEs onto all PEs into
192
       these bigger arrays so all PE will know what particles
                need to go where.
193
194
       cout << "Before Gather" << endl;</pre>
195
       MPI_Barrier(MPI_COMM_WORLD);
196
       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,
199
       MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
       MPI_Iallgather( Cptcl_y, sizeOfGather + 1, MPI_INT, Gptcl_y, maxToSend + 1,
200
       MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
       MPI_Iallgather( Cptcl_vx, sizeOfGather + 1, MPI_INT, Gptcl_vx, maxToSend + 1,
201
       MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
       MPI_Iallgather( Cptcl_vy, sizeOfGather + 1, MPI_INT, Gptcl_vy, maxToSend + 1,
202
       MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
203
204
        cout << "After Gather" << endl;</pre>
205
       MPI_Barrier(MPI_COMM_WORLD);
       // (7) Put in vector form so they can be added to PTCL. These arrays are 1-
208
       based.
209
       cout << "Before Adds" << endl;</pre>
210
211
       int Np = 0; for ( int i = 0 ; i < sizeOfGather ; ++i ) if ( Gptcl_PE[i] ==
212
       myPE ) ++Np;
213
       VD std_add_x ; std_add_x.resize ( Np+1 );
214
       VD std_add_y ; std_add_y.resize ( Np+1 );
215
           std_add_vx ; std_add_vx.resize ( Np+1 );
       VD
216
           std_add_vy ; std_add_vy.resize ( Np+1 );
       VD
217
218
        cout << "After adds, Before Assignment" << endl;</pre>
219
       int count = 1;
220
       for ( int i = 0 ; i < sizeOfGather ; ++i )</pre>
221
         if ( Gptcl_PE[i] == myPE )
222
223
        std_add_x [ count ] = Gptcl_x[i];
224
        std_add_y [ count ] = Gptcl_y [i];
225
        std_add_vx[ count ] = Gptcl_vx[i];
226
        std_add_vy[ count ] = Gptcl_vy[i];
227
        ++ count;
229
230
231
        cout << "Before Particle Add" << endl;</pre>
232
       PTCL.add( std_add_x, std_add_y, std_add_vx, std_add_vy);
233
       cout << "After Particle Add" << endl;</pre>
235
```

```
236
        // (8) Free up memory
237
238
                         > 0 ) { delete[] Cptcl_PE;
                                                      delete[] Cptcl_x ;
       if (maxToSend
       Cptcl_y ; delete[] Cptcl_vx ; delete[] Cptcl_vy; }
       if (sizeOfGather > 0 ) { delete[] Gptcl_PE; delete[] Gptcl_x ;
240
       Gptcl_y ; delete[] Gptcl_vx ; delete[] Gptcl_vy; }
241
     }
242
243
245
     int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }
246
247
   }:
```

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).
5
6
  11
     Begin Execution of fp in slurm batch script.
9
10
  \mathbf{I}
11
  free(): invalid next size (fast)
12
  [c3cpu-a5-u34-3:1351127] *** Process received signal ***
13
14 [c3cpu-a5-u34-3:1351127] Signal: Aborted (6)
  [c3cpu-a5-u34-3:1351127] Signal code:
                                          (-6)
16 [c3cpu-a5-u34-3:1351127]
                           [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x15303eb32ce0]
17 [c3cpu-a5-u34-3:1351127]
                            [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x15303e7a9a9f]
18 [c3cpu-a5-u34-3:1351127]
                            [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x15303e77ce05]
19 [c3cpu-a5-u34-3:1351127]
                            [ 3] /usr/lib64/libc.so.6(+0x91037)[0x15303e7ec037]
20 [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[0
      x404acf]
```

```
[c3cpu-a5-u34-3:1351127] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4051eb]
  [c3cpu-a5-u34-3:1351127] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
24
      x402c78]
  [c3cpu-a5-u34-3:1351127] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
      x15303e795cf3]
  [c3cpu-a5-u34-3:1351127] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
26
      x4023be]
  [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)
30
  [c3cpu-a5-u34-3:1351124] Signal code: (-6)
31
32 [c3cpu-a5-u34-3:1351124] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14731d7c1ce0]
33 [c3cpu-a5-u34-3:1351124] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14731d438a9f]
34 [c3cpu-a5-u34-3:1351124] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14731d40be05]
35 [c3cpu-a5-u34-3:1351124] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x14731d47b037]
36 [c3cpu-a5-u34-3:1351124] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x14731d48219c]
37 [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[0
38
      x404b08]
  [c3cpu-a5-u34-3:1351124] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4051eb]
  [c3cpu-a5-u34-3:1351124] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x402c78]
  [c3cpu-a5-u34-3:1351124] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
41
      x14731d424cf3]
  [c3cpu-a5-u34-3:1351124] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
42
      x4023be]
43 [c3cpu-a5-u34-3:1351124] *** End of error message ***
44 free(): invalid next size (fast)
45 [c3cpu-a5-u34-3:1351125] *** Process received signal ***
46 [c3cpu-a5-u34-3:1351125] Signal: Aborted (6)
47 [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]
49
  [c3cpu-a5-u34-3:1351125] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x150718d81e05]
51
  [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]
53
  [c3cpu-a5-u34-3:1351125] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
54
      x404acfl
  [c3cpu-a5-u34-3:1351125] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4051eb]
  [c3cpu-a5-u34-3:1351125] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x402c78]
  [c3cpu-a5-u34-3:1351125] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
      x150718d9acf3]
  [c3cpu-a5-u34-3:1351125] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4023be]
  [c3cpu-a5-u34-3:1351125] *** End of error message ***
  free(): invalid next size (fast)
  [c3cpu-a5-u34-3:1351126] *** Process received signal ***
61
62 [c3cpu-a5-u34-3:1351126] Signal: Aborted (6)
63 [c3cpu-a5-u34-3:1351126] Signal code: (-6)
```

```
64 [c3cpu-a5-u34-3:1351126] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14ebdfb6ace0]
65 [c3cpu-a5-u34-3:1351126] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14ebdf7e1a9f]
66 [c3cpu-a5-u34-3:1351126] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14ebdf7b4e05]
67 [c3cpu-a5-u34-3:1351126] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x14ebdf824037]
68 [c3cpu-a5-u34-3:1351126] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x14ebdf82b19c]
69 [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[0
      x404acf]
  [c3cpu-a5-u34-3:1351126] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4051eb]
  [c3cpu-a5-u34-3:1351126] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x402c78]
  [c3cpu-a5-u34-3:1351126] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
73
      x14ebdf7cdcf31
  [c3cpu-a5-u34-3:1351126] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4023be]
  [c3cpu-a5-u34-3:1351126] *** End of error message ***
  ______
77 Primary job terminated normally, but 1 process returned
78 a non-zero exit code. Per user-direction, the job has been aborted.
79
80
  mpirun noticed that process rank 2 with PID 1351126 on node c3cpu-a5-u34-3 exited
      on signal 6 (Aborted).
83
84
85 || Execution of fp in slurm batch script complete.
86
```

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).
4
5
6
  11
  || Begin Execution of fp in slurm batch script.
9
10
  11
11
malloc(): mismatching next->prev_size (unsorted)
13 [c3cpu-a5-u34-3:1336616] *** Process received signal ***
14 [c3cpu-a5-u34-3:1336616] Signal: Aborted (6)
15 [c3cpu-a5-u34-3:1336616] Signal code: (-6)
16 [c3cpu-a5-u34-3:1336616] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14c3178a2ce0]
17 [c3cpu-a5-u34-3:1336616] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14c317519a9f]
18 [c3cpu-a5-u34-3:1336616] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x14c3174ece05]
19 [c3cpu-a5-u34-3:1336616] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x14c31755c037]
20 [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]
22 [c3cpu-a5-u34-3:1336616] [ 6] /usr/lib64/libc.so.6(__libc_calloc+0x86)[0
     x14c317568486]
```

```
[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/
2.4
      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/
26
      lib/libmpi.so.40(PMPI_Iallgather+0x105)[0x14c3184f9585]
  [c3cpu-a5-u34-3:1336616] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x40466e]
  [c3cpu-a5-u34-3:1336616] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4051f9]
  [c3cpu-a5-u34-3:1336616] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
29
      x402c781
  [c3cpu-a5-u34-3:1336616] [14] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
30
      x14c317505cf3]
  [c3cpu-a5-u34-3:1336616] [15] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4023be]
  [c3cpu-a5-u34-3:1336616] *** End of error message ***
32
  malloc(): unsorted double linked list corrupted
34 [c3cpu-a5-u34-3:1336615] *** Process received signal ***
35 [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]
39 [c3cpu-a5-u34-3:1336615] [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x1514df793e05]
40 [c3cpu-a5-u34-3:1336615] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x1514df803037]
41 [c3cpu-a5-u34-3:1336615] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x1514df80a19c]
42 [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)[0
      x1514df80f486]
  [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[0
48
      x40466e]
  [c3cpu-a5-u34-3:1336615] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
49
      x4051f9]
  [c3cpu-a5-u34-3:1336615] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
  [c3cpu-a5-u34-3:1336615] [14] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
      x1514df7accf31
  [c3cpu-a5-u34-3:1336615] [15] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4023be]
  [c3cpu-a5-u34-3:1336615] *** End of error message ***
55
  Primary job terminated normally, but 1 process returned
  a non-zero exit code. Per user-direction, the job has been aborted.
56
```

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).
5
6
  11
  || Begin Execution of fp in slurm batch script.
10
  ==
11
12 realloc(): invalid pointer
13 [c3cpu-a5-u34-3:1350630] *** Process received signal ***
14 [c3cpu-a5-u34-3:1350630] Signal: Aborted (6)
15 [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]
18 [c3cpu-a5-u34-3:1350630]
                           [ 2] /usr/lib64/libc.so.6(abort+0x127)[0x145b3cce6e05]
19 [c3cpu-a5-u34-3:1350630] [ 3] /usr/lib64/libc.so.6(+0x91037)[0x145b3cd56037]
20 [c3cpu-a5-u34-3:1350630] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x145b3cd5d19c]
 21 \quad \texttt{[c3cpu-a5-u34-3:1350630]} \quad \texttt{[5]} \quad \texttt{/usr/lib64/libc.so.6(realloc+0x23a)} \\ \texttt{[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[0
  [c3cpu-a5-u34-3:1350630] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4051eb]
  [c3cpu-a5-u34-3:1350630] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
28
      x402c781
  [c3cpu-a5-u34-3:1350630] [13] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
      x145b3ccffcf3]
  [c3cpu-a5-u34-3:1350630] [14] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4023be]
  [c3cpu-a5-u34-3:1350630] *** End of error message ***
31
  Primary job terminated normally, but 1 process returned
  a non-zero exit code. Per user-direction, the job has been aborted.
   ______
```

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).
5
6
  11
9
  || Begin Execution of fp in slurm batch script.
10
  ==
11
  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.
13 [c3cpu-a5-u34-3:1362180] *** Process received signal ***
  [c3cpu-a5-u34-3:1362180] Signal: Aborted (6)
15 [c3cpu-a5-u34-3:1362180] Signal code: (-6)
16 [c3cpu-a5-u34-3:1362180] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x14ea34afcce0]
17 [c3cpu-a5-u34-3:1362180] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x14ea34773a9f]
18 \quad \texttt{[c3cpu-a5-u34-3:1362180]} \quad \texttt{[2]/usr/lib64/libc.so.6(abort+0x127)[0x14ea34746e05]}
19 [c3cpu-a5-u34-3:1362180] [ 3] /usr/lib64/libc.so.6(+0x9817a)[0x14ea347bd17a]
20 [c3cpu-a5-u34-3:1362180] [ 4] /usr/lib64/libc.so.6(+0x9a8e8)[0x14ea347bf8e8]
21 [c3cpu-a5-u34-3:1362180] [ 5] /usr/lib64/libc.so.6(+0x9b659)[0x14ea347c0659]
22 [c3cpu-a5-u34-3:1362180] [ 6] /usr/lib64/libc.so.6(+0x9bf5f)[0x14ea347c0f5f]
23 [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[0
28
      x404713]
   [c3cpu-a5-u34-3:1362180] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x405319]
   [c3cpu-a5-u34-3:1362180] [14] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x402c78]
  [c3cpu-a5-u34-3:1362180] [15] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
31
      x14ea3475fcf31
   [c3cpu-a5-u34-3:1362180] [16] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
      x4023be]
   [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).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).

mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited on signal 6 (Aborted).
```

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).
3
4
5
6
  ==
7
9 || Begin Execution of fp in slurm batch script.
10 H
11
  [c3cpu-a5-u34-3:1352023:0:1352023] Caught signal 11 (Segmentation fault: address
12
     not mapped to object at address 0x100)
BFD: Dwarf Error: Can't find .debug_ranges section.
14 BFD: Dwarf Error: Can't find .debug_ranges section.
BFD: Dwarf Error: Can't find .debug_ranges section.
16 BFD: Dwarf Error: Can't find .debug_ranges section.
17 BFD: Dwarf Error: Can't find .debug_ranges section.
18 BFD: Dwarf Error: Can't find .debug_ranges section.
19 BFD: Dwarf Error: Can't find .debug_ranges section.
20 BFD: Dwarf Error: Can't find .debug_ranges section.
21 ==== backtrace (tid:1352023) ====
  0 0x000000000012ce0 __funlockfile() :0
   1 0x0000000000203ec6 mca_pml_ucx_isend() ???:0
   2 0x0000000001055f1 NBC_Start_round() nbc.c:0
   3 0x000000000105ab7 NBC_Start() ???:0
   4 0x00000000010936a ompi_coll_libnbc_iallgather()
27
   5 0x0000000000095585 PMPI_Iallgather() ???:0
   6 0x0000000004046be mpiInfo::ParticleExchange()
                                                  ???:0
28
   7 0x0000000004051eb Mesh::ParticlesOnMesh() ???:0
29
  8 0x0000000000402c78 main() ???:0
30
  9 0x00000000003acf3 __libc_start_main() ???:0
31
32 10 0x00000000004023be _start() ???:0
  _____
35 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).
```

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
5 \text{ myPE}: 1 \text{ Time} = 0
6 Before Reduce
  myPE: 2 Time = 0
  Before Reduce
10
11
12
   FREE PARTICLE
13
   D E M O
            C O D E
14
15
   Running on 4 processors
16
17
18
19
20
  Input Summary:
21
  -----
23
  No. PE in x-direction: 2
              y-direction: 2
24
  No. Cells in x-direction: 5
25
                y-direction: 5
26
                    : 10
27
  Flux density
  End Time
                           : 3
29
  Time Step
                           : 0.01
30
31
32 \text{ myPE: } 0 \text{ Time = } 0
33 Before Reduce
34 After Reduce, Before C Particles
35 After C Particles
36 Before Gather Particles
37 After Gather Particles
38 Before Gather
39 After Gather
40 Before Adds
41 After adds, Before Assignment
```

```
42 Before Particle Add
43 After Particle Add
44 After Exchange
45 After Reduce, Before C Particles
46 After C Particles
47 Before Gather Particles
48 After Gather Particles
49 Before Gather
50 After Gather
51 Before Adds
52 After adds, Before Assignment
53 Before Particle Add
54 After Particle Add
55 After Exchange
56 After Reduce, Before C Particles
57 After C Particles
58 Before Gather Particles
59 After Gather Particles
60 Before Gather
61 After Gather
62 Before Adds
63 After adds, Before Assignment
64 Before Particle Add
65 After Particle Add
66 After Exchange
67 After Reduce, Before C Particles
68 After C Particles
69 Before Gather Particles
70 After Gather Particles
71 Before Gather
72 After Gather
73 Before Adds
74 After adds, Before Assignment
75 Before Particle Add
76 After Particle Add
77 After Exchange
myPE: 3 \text{ Time} = 0.55
2 Before Reduce
```

```
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

Before Gather Particles

After Reduce, Before C Particles

After Gather Particles

Before Gather

After Gather Particles

Before Gather

After Reduce, Before C Particles

After C Particles

Before Gather

After Reduce, Before C Particles

After C Particles

Before 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).

```
1 ==12158== Memcheck, a memory error detector
2 ==12158== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
3 ==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.
9 ==12158==
                See README_MISSING_SYSCALL_OR_IOCTL for guidance on writing a proper
      wrapper.
10 ==12162== Warning: invalid file descriptor 1048564 in syscall close()
11 ==12162== Warning: invalid file descriptor 1048565 in syscall close()
12 ==12162== Warning: invalid file descriptor 1048566 in syscall close()
13 ==12162== Warning: invalid file descriptor 1048567 in syscall close()
               Use --\log-fd=<number> to select an alternative log fd.
15 ==12162== Warning: invalid file descriptor 1048568 in syscall close()
16 ==12162== Warning: invalid file descriptor 1048569 in syscall close()
17 ==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.
21
==12163== Warning: invalid file descriptor 1048568 in syscall close()
_{23} ==12163== Warning: invalid file descriptor 1048569 in syscall close()
24 ==12164== Warning: invalid file descriptor 1048564 in syscall close()
25 ==12164== Warning: invalid file descriptor 1048565 in syscall close()
26 ==12164== Warning: invalid file descriptor 1048566 in syscall close()
27 ==12164== Warning: invalid file descriptor 1048567 in syscall close()
28 ==12164==
               Use --log-fd=<number> to select an alternative log fd.
29 ==12164== Warning: invalid file descriptor 1048568 in syscall close()
30 ==12164== Warning: invalid file descriptor 1048569 in syscall close()
31 ==12165== Warning: invalid file descriptor 1048564 in syscall close()
32 ==12165== Warning: invalid file descriptor 1048565 in syscall close()
```

```
33 ==12165== Warning: invalid file descriptor 1048566 in syscall close()
_{34} ==12165== Warning: invalid file descriptor 1048567 in syscall close()
35 ==12165==
               Use --log-fd=<number> to select an alternative log fd.
36 ==12165== Warning: invalid file descriptor 1048568 in syscall close()
37 ==12165== Warning: invalid file descriptor 1048569 in syscall close()
39 ==12158== HEAP SUMMARY:
                 in use at exit: 20,396 bytes in 128 blocks
40 ==12158==
               total heap usage: 27,478 allocs, 27,350 frees, 12,255,873 bytes
41 ==12158==
      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)
44
  ==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)
                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
  ==12158==
      .6.0.2)
  ==12158==
                by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
      _slurmpmi/bin/orterun)
52 ==12158==
               by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
53 ==12158==
54 ==12158== 136 bytes in 1 blocks are definitely lost in loss record 45 of 73
55 ==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)
                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)
                by Ox4ED5AA9: orte_odls_base_default_construct_child_list (in /curc/
  ==12158==
      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)
                by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc
  ==12158==
      /11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
                by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi
  ==12158==
      /4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
               by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
  ==12158==
  ==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)
65 ==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
                at 0x4C37135: malloc (vg_replace_malloc.c:381)
  ==12158==
  ==12158==
               by 0x4E817B2: show_help (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
69
      _slurmpmi/lib/libopen-rte.so.40.30.1)
              by Ox4E81AE6: orte_show_help_recv (in /curc/sw/install/openmpi/4.1.1/
  ==12158==
```

```
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)
                by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
   ==12158==
   ==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)
                by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
   ==12158==
   ==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)
78
   ==12158==
                by 0x4F204CF: assign (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
79
       _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)
                by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc
   ==12158==
       /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)
85
   ==12158==
                by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
86
       .6.0.2)
                by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
   ==12158==
       _slurmpmi/bin/orterun)
   ==12158==
                by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
   ==12158==
89
   ==12158== 962 (96 direct, 866 indirect) bytes in 2 blocks are definitely lost in
      loss record 67 of 73
                at 0x4C37135: malloc (vg_replace_malloc.c:381)
   ==12158==
                by 0x532DCD0: new_tracker.constprop.0 (in /curc/sw/install/openmpi
   ==12158==
       /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/
94
       gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
                by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/
   ==12158==
95
       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
98
       .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)
102 ==12158==
_{103} ==12158== 2,048 bytes in 1 blocks are definitely lost in loss record 70 of 73
                at 0x4C37135: malloc (vg_replace_malloc.c:381)
104 ==12158==
```

```
==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/
106
      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
107
      /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
   ==12158==
                by 0x4EA2BBB: orte_pmix_server_register_nspace (in /curc/sw/install/
108
      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)
                by 0x4E9088C: orte_daemon_recv (in /curc/sw/install/openmpi/4.1.1/gcc
   ==12158==
111
       /11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
   ==12158==
                by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi
112
       /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)
                by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
   ==12158==
115
       _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/
120
      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
121
      /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)
                by 0x4F18D50: orte_rml_base_process_msg (in /curc/sw/install/openmpi
126
   ==12158==
       /4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
                by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
   ==12158==
   ==12158==
                by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
128
       .6.0.2)
   ==12158==
                by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
129
       _slurmpmi/bin/orterun)
   ==12158==
130
   ==12158== 2,436 (1,656 direct, 780 indirect) bytes in 3 blocks are definitely lost
131
        in loss record 72 of 73
   ==12158==
                at 0x4C3BE4B: calloc (vg_replace_malloc.c:1328)
132
                by 0x533C45C: pmix_server_log (in /curc/sw/install/openmpi/4.1.1/gcc
   ==12158==
       /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)
                by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/
   ==12158==
      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
136
       /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)
137
                by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
   ==12158==
       .6.0.2)
                by 0x530BD8D: progress_engine (in /curc/sw/install/openmpi/4.1.1/gcc
   ==12158==
139
      /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
   ==12158==
                by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
140
                by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
   ==12158==
141
   ==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)
144
   ==12158==
                by 0x533C0D8: pmix_server_log (in /curc/sw/install/openmpi/4.1.1/gcc
145
       /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
                by 0x53250B8: server_switchyard (in /curc/sw/install/openmpi/4.1.1/
   ==12158==
146
      gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
   ==12158==
                by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/
147
      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
148
      /4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
                by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
   ==12158==
149
   ==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)
152
   ==12158==
                by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
153
   ==12158==
154
   ==12158== LEAK SUMMARY:
   ==12158==
                definitely lost: 7,872 bytes in 14 blocks
                indirectly lost: 3,770 bytes in 28 blocks
157 ==12158==
                  possibly lost: O bytes in O blocks
158 ==12158==
159 ==12158==
                still reachable: 8,754 bytes in 86 blocks
                      suppressed: 0 bytes in 0 blocks
160 ==12158==
   ==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.