

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 track 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 (x_0, x_1, y_0, y_1) and divided into cells in both the x and y directions (`ncellx` and `ncelly`).
- 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  // ||
3  // ||   C L A S S:   m p i I n f o
4  // ||
5  // ==
6
7  class mpiInfo
8  {
9  public:
10
11     int myPE;
12     int numPE;
13     int nRealx, nRealy;
14     int nx, ny;
15     int nPEx, nPEy;
16     int iPE , jPE;
17     int iMin, iMax, jMin, jMax ; // The global i-j numbers on this processor
18     int nei_n, nei_s, nei_e, nei_w;
19     int nei_ne, nei_nw, nei_se, nei_sw;
20
21     int countx, county;
22
23     MPI_Status status;
24     int      err;
25     int      tag;
26     MPI_Request request;
27
28     // -
29     // |
30     // |   GridDecomposition: Set up PE numbering system in figure below and
31     // |   establish communication arrays.
32     // |
33     // |   nPEx -- number of PEs in the x-direction
34     // |   nPEy -- number of PEs in the y-direction
35     // |   numPE = total number of PEs
36     // |
37     // |
38     // |
39     // |
40     // |
41     // |
42     // |
43     // |
44     // |
45     // |
46     // |
47     // |
48     // |
49     // |
50     // |
51     // |
52     // |
53     // |

```

+-----+-----+ . . . +-----+	
	numPE
+-----+-----+ . . . +-----+	
.	.
.	.
.	.
+-----+-----+ . . . +-----+	
nPEx	nPEx+1
+-----+-----+ . . . +-----+	
0	1
	nPEx-1
+-----+-----+ . . . +-----+	

```

54 // |
55 // |
56 // -
57
58
59 void GridDecomposition(int _nPEx, int _nPey, int nCellx , int nCelly)
60 {
61
62     nRealx = nCellx + 1;
63     nRealy = nCelly + 1;
64
65     // Store and check incoming processor counts
66
67     nPEx = _nPEx;
68     nPEy = _nPey;
69
70     if (nPEx*nPEy != numPE)
71     {
72         if ( myPE == 0 ) cout << "Fatal Error:  Number of PEs in x-y directions do
not add up to numPE" << endl;
73         MPI_Barrier(MPI_COMM_WORLD);
74         MPI_Finalize();
75         exit(0);
76     }
77
78     // Get the i-j location of this processor, given its number.  See figure above
:
79
80     jPE = int(myPE/nPEx);
81     iPE = myPE - jPE*nPEx;
82
83     // Set neighbor values
84
85     nei_n = nei_s = nei_e = nei_w = -1;
86
87     if ( iPE > 0      )
88     {
89         nei_w = myPE - 1      ;
90     }
91     if ( jPE > 0      )
92     {
93         nei_s = myPE - nPEx ;
94     }
95     if ( iPE < nPEx-1 )
96     {
97         nei_e = myPE + 1      ;
98     }
99     if ( jPE < nPEy-1 )
100    {
101        nei_n = myPE + nPEx ;
102    }
103
104    nei_nw = nei_sw = nei_ne = nei_se = -1;
105
106    if ( iPE > 0      && jPE > 0      ) nei_sw = myPE - nPEx - 1 ;

```

```

107     if ( iPE < nPEx-1 && jPE > 0      )   nei_se = myPE - nPEx + 1 ;
108     if ( iPE > 0      && jPE < nPEy-1 )   nei_nw = myPE + nPEx - 1 ;
109     if ( iPE < nPEx-1 && jPE < nPEy-1 )   nei_ne = myPE + nPEx + 1 ;
110
111     // Acquire memory for the communication between adjacent processors:
112     countx = nRealx + 2;
113     county = nRealy + 2;
114
115     tag = 0;
116 }
117
118
119
120 // ==
121 // ||
122 // ||   ParticlesExchange
123 // ||
124 // ||   Exchange particles between processors
125 // ||
126 // ||
127 // ==
128
129 void ParticleExchange( VI &ptcl_send_list , VI &ptcl_send_PE , particles &PTCL)
130 {
131     MPI_Status  status;
132     MPI_Request request;
133
134     // (1) Get the max number particles to be sent by any particular processor,
135     // and make sure all processors know that number.
136
137     int numToSend = ptcl_send_list.size();      int maxToSend;
138
139     /* TO-DO Place MPI call here to complete (1) */
140     MPI_Barrier(MPI_COMM_WORLD);
141     MPI_Iallreduce(&numToSend, &maxToSend, 1, MPI_INT, MPI_MAX, MPI_COMM_WORLD, &
142     request); MPI_Wait(&request,&status);
143     MPI_Barrier(MPI_COMM_WORLD);
144
145     // (2) Allocate contributions to the upcoming Gather operation. Here, "C" for
146     // "Contribution" to be Gathered
147
148     int      *Cptcl_PE; Cptcl_PE = new int      [ maxToSend ]; // Particles'
149     destination PEs
150     double *Cptcl_x ; Cptcl_x  = new double [ maxToSend ];
151     double *Cptcl_y ; Cptcl_y  = new double [ maxToSend ];
152     double *Cptcl_vx; Cptcl_vx = new double [ maxToSend ];
153     double *Cptcl_vy; Cptcl_vy = new double [ maxToSend ];
154
155     // (3) Populate contributions on all processors for the upcoming Gather
156     operation
157
158     for ( int i = 0 ; i < maxToSend ; ++i ) { Cptcl_PE[i] = -1; Cptcl_x [i] = 0.;
159     Cptcl_y [i] = 0.; Cptcl_vx[i] = 0.; Cptcl_vy[i] = 0.; }

```

```

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
159 for ( int i = 0 ; i < ptcl_send_list.size() ; ++i )
160 {
161     int id      = ptcl_send_list[ i ];
162     Cptcl_PE[i] = ptcl_send_PE  [ i ];
163     Cptcl_x [i] = PTCL.x        [ id ];
164     Cptcl_y [i] = PTCL.y        [ id ];
165     Cptcl_vx[i] = PTCL.vx       [ id ];
166     Cptcl_vy[i] = PTCL.vy       [ id ];
167 }
168
169 // (5) Allocate and initialize the arrays for upcoming Gather operation to PEO
170 // The sizeofGather takes
171 // into account the number of processors, like this figure:
172 //
173 // |<----- sizeofGather
174 // ----->|
175 //
176 // |
177 // |
178 // |
179 // |<- maxToSend ->|<- maxToSend ->|<- maxToSend ->|<- maxToSend
180 // ->|
181 //
182 +-----+-----+-----+-----+
183 //          PEO          PE1          PE2          PE3
184
185 int sizeofGather = maxToSend * numPE;
186
187 int *Gptcl_PE; Gptcl_PE = new int [ sizeofGather ];
188 double *Gptcl_x ; Gptcl_x = new double [ sizeofGather ];
189 double *Gptcl_y ; Gptcl_y = new double [ sizeofGather ];
190 double *Gptcl_vx; Gptcl_vx = new double [ sizeofGather ];
191 double *Gptcl_vy; Gptcl_vy = new double [ sizeofGather ];
192
193 for ( int i = 0 ; i < sizeofGather ; ++i ) { Gptcl_PE[i] = -1; Gptcl_x [i] =
194 0.; Gptcl_y [i] = 0.; Gptcl_vx[i] = 0.; Gptcl_vy[i] = 0.; }
195
196 // (6) Gather "Contributions" ("C" arrays) from all PEs onto all PEs into
197 // these bigger arrays so all PE will know what particles
198 // need to go where.
199
200 MPI_Barrier(MPI_COMM_WORLD);
201
202 MPI_Iallgather( Cptcl_PE, maxToSend, MPI_INT, Gptcl_PE, maxToSend, MPI_INT,
203 MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
204 MPI_Iallgather( Cptcl_x, maxToSend, MPI_DOUBLE, Gptcl_x, maxToSend, MPI_DOUBLE
205 , MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
206 MPI_Iallgather( Cptcl_y, maxToSend, MPI_DOUBLE, Gptcl_y, maxToSend, MPI_DOUBLE
207 , MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
208 MPI_Iallgather( Cptcl_vx, maxToSend, MPI_DOUBLE, Gptcl_vx, maxToSend,

```

```

199 MPI_DOUBLE, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
    MPI_Iallgather( Cptcl_vy, maxToSend, MPI_DOUBLE, Gptcl_vy, maxToSend,
200 MPI_DOUBLE, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
201
202 MPI_Barrier(MPI_COMM_WORLD);
203
204 // (7) Put in vector form so they can be added to PTCL. These arrays are 1-
    based.
205
206 int Np = 0; for ( int i = 0 ; i < sizeofGather ; ++i ) if ( Gptcl_PE[i] ==
    myPE ) ++Np;
207
208 VD std_add_x ; std_add_x.resize ( Np+1 );
209 VD std_add_y ; std_add_y.resize ( Np+1 );
210 VD std_add_vx ; std_add_vx.resize ( Np+1 );
211 VD std_add_vy ; std_add_vy.resize ( Np+1 );
212
213 int count = 1;
214 for ( int i = 0 ; i < sizeofGather ; ++i )
215     if ( Gptcl_PE[i] == myPE )
216     {
217         std_add_x [ count ] = Gptcl_x[i];
218         std_add_y [ count ] = Gptcl_y [i];
219         std_add_vx[ count ] = Gptcl_vx[i];
220         std_add_vy[ count ] = Gptcl_vy[i];
221         ++ count;
222     }
223
224 PTCL.add( std_add_x, std_add_y, std_add_vx, std_add_vy );
225
226 // (8) Free up memory
227
228 if (maxToSend > 0 ) { delete[] Cptcl_PE; delete[] Cptcl_x ; delete[]
    Cptcl_y ; delete[] Cptcl_vx ; delete[] Cptcl_vy; }
229 if (sizeofGather > 0 ) { delete[] Gptcl_PE; delete[] Gptcl_x ; delete[]
    Gptcl_y ; delete[] Gptcl_vx ; delete[] Gptcl_vy; }
230
231 }
232
233
234 int pid(int i,int j) { return (i+1) + (j)*(nRealx+2); }
235
236 };

```

Listing 1: Parallel Code (mpiInfo.h)

```

1 // =====
2 // ||
3 // || fp
4 // || -----
    ||

```



```

5 // || F R E E   P A R T I C L E
6 // ||
7 // || D E M O N S T R A T I O N   C O D E
8 // || -----
9 // ||
10 // || Developed by: Scott R. Runnels, Ph.D.
11 // || University of Colorado Boulder
12 // ||
13 // || For: CU Boulder CSCI 4576/5576 and associated labs
14 // ||
15 // || Copyright 2020 Scott Runnels
16 // ||
17 // || Not for distribution or use outside of the
18 // || this course.
19 // ||
20 // ||
=====

21
22 #include "mpi.h"
23 #include "fp.h"
24 #include "particles.h"
25 #include "mpiInfo.h"
26
27
28 // ==
29 // ||
30 // || C L A S S:   M E S H
31 // ||
32 // ||
33 // ||
34 // || (21) ----(22) ----(23) ----(24) ----(25) <--- nRealx+1 , nRealy+1
35 // || |         |         |         |         |
36 // || |         |         |         |         |
37 // || |         |         |         |         |
38 // || (16) ----[17] ----[18] ----[19] ----(20)
39 // || |   1,3|   2,3|   3,3|         |
40 // || |         |         |         |         |
41 // || |         |         |         |         |
42 // || (11) ----[12] ----[13] ----[14] ----(15)

```

```

43 // || | 1,2| 2,2| 3,2| |
44 // || | | | | | |
45 // || | | | | | |
46 // || (6)-----[7]-----[8]-----[9]------(10)
47 // || | 1,1| 2,1| 3,1| |
48 // || | | | | | |
49 // || | | | | | |
50 // || (1)------(2)------(3)------(4)------(5)
51 // || 0,0 | | | | | 4,0
52 // ||
53 // ==
54
55 class Mesh
56 {
57
58 public:
59
60     double x0, x1, y0, y1;
61     VD x,y;
62     int nRealx , nRealy , nField, nReal;
63     double lengthx, lengthy, dx, dy;
64     int myPE;
65     VD Qval;
66
67     // ==
68     // ||
69     // || Constructor: Initialize values
70     // ||
71     // ==
72
73     Mesh(double _x0 , double _x1, double _y0, double _y1 , int ncell_x , int
        ncell_y, mpiInfo &myMPI )
74     {
75
76
77         // Copy incoming values
78
79         x0 = _x0;          x1 = _x1;
80         y0 = _y0;          y1 = _y1;
81         myPE = myMPI.myPE;
82
83         // Compute number of real (physical) nodes, and their spacing
84
85         nRealx = ncell_x + 1;
86         nRealy = ncell_y + 1;
87         nReal = nRealx*nRealy;
88         dx = (x1-x0) / ncell_x;
89         dy = (y1-y0) / ncell_y;
90
91         // Compute the size of the field variables, which also lie on ghost nodes
92
93         nField = (nRealx+2)*(nRealy+2);
94
95         // Allocate memory -- Note that the node numbers and field variable numbers
        must

```

```

96 // match. So even though we only will be caring about real nodes, their node
97 // numbers are naturally ordered, so must be of size nField.
98
99 x.resize(nField+1); y.resize(nField+1); Qval.resize(nField+1);
100
101 for ( int i = 1 ; i <= nRealx ; ++i )
102     for ( int j = 1 ; j <= nRealy ; ++j )
103     {
104         int p = pid(i,j);
105         x[p] = x0 + (i-1)*dx;
106         y[p] = y0 + (j-1)*dy;
107     }
108
109 }
110
111
112
113 // ==
114 // ||
115 // || ParticlesOnMesh
116 // ||
117 // || Quantify the particles' interaction with the mesh
118 // ||
119 // || References: [1] https://www.particleincell.com/2010/es-pic-method/
120 // ||
121 // ==
122
123 void ParticlesOnMesh(particles &PTCL, mpiInfo &myMPI)
124 {
125     double hx, hy;
126     double w[5];
127     int p[5];
128     int iL, iR, jB, jT;
129     int iPEnew, jPEnew; // These store the i-j indicies of the
processor receiving a particle,
130 // if that particle is leaving the mesh.
131 VI ptcl_send_list, ptcl_send_PE; // These collect information about particles
that
132 // have left this processor and are
heading onto
133 // another processor.
134
135 // -
136 // |
137 // | Determine which particles are still on this mesh and which have left
138 // |
139 // -
140
141 for ( int k = 1 ; k <= PTCL.n ; ++k )
142     {
143 // First, check to be sure the particle is still in the mesh. If it is not,
set its
144 // "active" flag to zero, and note the processor to which it is going for MPI
exchange.
145

```

```

146 if ( PTCL.active[k] == 1 )
147 {
148     iPEnew = myMPI.iPE;
149     jPEnew = myMPI.jPE;
150
151
152     if ( PTCL.x[k] < x0      ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE - 1 ;
153 }
154     if ( PTCL.x[k] > x1      ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE + 1 ;
155 }
156     if ( PTCL.y[k] < y0      ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE - 1 ;
157 }
158     if ( PTCL.y[k] > y1      ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE + 1 ;
159 }
160
161 }
162
163 // The particle is not in the mesh.  Collect this particle into a holding array
164 // that will
165 // be sent to the neighboring processor.
166
167 if ( PTCL.active[k] == -1)
168 {
169     if ( iPEnew >= 0 && iPEnew < myMPI.nPEx )
170     if ( jPEnew >= 0 && jPEnew < myMPI.nPEy )
171     {
172         ptcl_send_list.push_back(k);
173         ptcl_send_PE .push_back( iPEnew + jPEnew * myMPI.nPEx);
174     }
175
176     PTCL.active[k] = 0;  // Remove it from the list of active particles
177 }
178
179 // -
180 // |
181 // | Give and receive particles to/with other processors
182 // |
183 // -
184
185 myMPI.ParticleExchange( ptcl_send_list , ptcl_send_PE , PTCL);
186
187 // -
188 // |
189 // | Accumulate particles to the nodes (to be completed in the esPIC code,
190 // | next week)
191 // |
192 // -
193
194 for ( int k = 1 ; k <= nField ; ++k ) Qval[k] = 0.;
195
196 // -
197 // |
198 // | Compute forces on particles

```

```

195 // |
196 // -
197
198 for ( int k = 1 ; k <= 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 // ||
225 // || Main Program
226 // ||
227 // ||
228 // ==
229 //
230
231 int main(int argc, char *argv[])
232 {
233
234     mpiInfo myMPI;
235     MPI_Init      (&argc      , &argv      );
236     MPI_Comm_size(MPI_COMM_WORLD, &myMPI.numPE);
237     MPI_Comm_rank(MPI_COMM_WORLD,&myMPI.myPE );
238
239     int nPEx, nPEy, nCellx, nCelly;
240     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 << " F R E E   P A R T I C L E           \n";
255     cout << " D E M O   C O D E                   \n";
256     cout << "\n";
257     cout << " Running on " << myMPI.numPE << " processors \n";
258     cout << "\n";
259     cout << "-----\n";
260     cout << "\n";
261 }
262
263 for (int count = 0 ; count < argc; ++count)
264 {
265     if ( !strcmp(argv[count], "-nPEx" ) ) nPEx = atoi(argv[count+1]);
266     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]);
271     if ( !strcmp(argv[count], "-dt" ) ) dt = atof(argv[count+1]);
272 }
273
274 if ( myMPI.myPE == 0 )
275 {
276     cout << endl;
277     cout << "Input Summary: " << endl;
278     cout << "----- " << endl;
279     cout << "No. PE in x-direction: " << nPEx << endl;
280     cout << "                y-direction: " << nPEy << endl;
281     cout << "No. Cells in x-direction: " << nCellx << endl;
282     cout << "                y-direction: " << nCelly << endl;
283     cout << "Flux density          : " << flux << endl;
284     cout << "End Time              : " << tEnd << endl;
285     cout << "Time Step             : " << dt << endl;
286     cout << endl;
287 }
288
289 // -
290 // |
291 // | 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 int      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 <= tEnd ; t += dt )
332 {
333     cout << endl;
334     cout << "myPE: " << myMPI.myPE << " Time = " << t << endl;
335
336     // Inject particles
337
338     if ( myMPI.iPE == 0 ) PTCL.addFlux( t , MESH.y0, MESH.y1, density, vx_bdy ,
339     vy_bdy );
340
341     // Move particles
342
343     PTCL.move( dt );
344
345     // Map between particles and the mesh
346
347     MESH.ParticlesOnMesh(PTCL,myMPI);
348
349     // Plot
350
351     PTCL.plot( "ptcl" , count , myMPI.myPE );
352     MESH.plot( "mesh" , count , myMPI );
353
354     ++count;
355 }
356
357 // -
358 // |

```

```

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

```

Listing 2: Parallel Code (fp.cpp)

```

1 #!/bin/bash
2
3 # -
4 # |
5 # | This is a batch script for running a MPI parallel job on Summit
6 # |
7 # | (o) To submit this job, enter: sbatch --export=CODE='/home/scru5660/HPSC/
   codes/fd_mpi/src' ex_01.bat
8 # |
9 # | (o) To check the status of this job, enter: squeue -u <username>
10 # |
11 # -
12
13 # -
14 # |
15 # | Part 1: Directives
16 # |
17 # -
18
19 #SBATCH --nodes=1
20 #SBATCH --ntasks=4
21 #SBATCH --time=00:01:00
22 #SBATCH --partition=amilan
23 #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
34
35 # -
36 # |
37 # | Part 3: User scripting
38 # |
39 # -
40
41 echo "=="

```



```

42 echo "||"
43 echo "|| Begin Execution of fp in slurm batch script."
44 echo "||"
45 echo "=="
46
47 srun -n 4 $CODE/fp -nPEx 2 -nPEy 2 -nCellx 5 -nCelly 5 -flux 10 -tEnd 5 -dt 0.05 >
    tty.out
48
49 echo "=="
50 echo "||"
51 echo "|| Execution of fp in slurm batch script complete."
52 echo "||"
53 echo "=="

```

Listing 3: Slurm File (ex01.bat)

4 Appendix B: Output

4.1 Output of the Slurm

```

1 -----
2
3
4 F R E E   P A R T I C L E
5 D E M O   C O D E
6
7 Running on 4 processors
8
9 -----
10
11
12 Input Summary:
13 -----
14 No. PE    in  x-direction: 2
15                y-direction: 2
16 No. Cells in x-direction: 5
17                y-direction: 5
18 Flux density          : 10
19 End Time              : 5
20 Time Step            : 0.05
21
22
23 myPE: 0 Time = 0
24
25 myPE: 1 Time = 0
26
27 myPE: 2 Time = 0
28
29 myPE: 3 Time = 0
30
31 myPE: 0 Time = 0.05
32
33 myPE: 3 Time = 0.05
34
35 myPE: 1 Time = 0.05

```

```
36
37 myPE: 2 Time = 0.05
38
39 myPE: 0 Time = 0.1
40
41 myPE: 3 Time = 0.1
42
43 myPE: 1 Time = 0.1
44
45 myPE: 2 Time = 0.1
46
47 myPE: 0 Time = 0.15
48
49 myPE: 2 Time = 0.15
50
51 myPE: 3 Time = 0.15
52
53 myPE: 1 Time = 0.15
54
55 myPE: 0 Time = 0.2
56
57 myPE: 1 Time = 0.2
58
59 myPE: 3 Time = 0.2
60
61 myPE: 2 Time = 0.2
62
63 myPE: 0 Time = 0.25
64
65 myPE: 3 Time = 0.25
66
67 myPE: 1 Time = 0.25
68
69 myPE: 2 Time = 0.25
70
71 myPE: 0 Time = 0.3
72
73 myPE: 2 Time = 0.3
74
75 myPE: 3 Time = 0.3
76
77 myPE: 1 Time = 0.3
78
79 myPE: 0 Time = 0.35
80
81 myPE: 3 Time = 0.35
82
83 myPE: 1 Time = 0.35
84
85 myPE: 2 Time = 0.35
86
87 myPE: 1 Time = 0.4
88
89 myPE: 0 Time = 0.4
90
```

```
91 myPE: 3 Time = 0.4
92
93 myPE: 2 Time = 0.4
94
95 myPE: 1 Time = 0.45
96
97 myPE: 2 Time = 0.45
98
99 myPE: 0 Time = 0.45
100
101 myPE: 3 Time = 0.45
102
103 myPE: 2 Time = 0.5
104
105 myPE: 3 Time = 0.5
106
107 myPE: 0 Time = 0.5
108
109 myPE: 1 Time = 0.5
110
111 myPE: 3 Time = 0.55
112
113 myPE: 2 Time = 0.55
114
115 myPE: 1 Time = 0.55
116
117 myPE: 0 Time = 0.55
118
119 myPE: 3 Time = 0.6
120
121 myPE: 2 Time = 0.6
122
123 myPE: 1 Time = 0.6
124
125 myPE: 0 Time = 0.6
126
127 myPE: 3 Time = 0.65
128
129 myPE: 2 Time = 0.65
130
131 myPE: 0 Time = 0.65
132
133 myPE: 1 Time = 0.65
134
135 myPE: 2 Time = 0.7
136
137 myPE: 3 Time = 0.7
138
139 myPE: 0 Time = 0.7
140
141 myPE: 1 Time = 0.7
142
143 myPE: 3 Time = 0.75
144
145 myPE: 2 Time = 0.75
```

```
146
147 myPE: 0 Time = 0.75
148
149 myPE: 1 Time = 0.75
150
151 myPE: 3 Time = 0.8
152
153 myPE: 2 Time = 0.8
154
155 myPE: 1 Time = 0.8
156
157 myPE: 0 Time = 0.8
158
159 myPE: 3 Time = 0.85
160
161 myPE: 2 Time = 0.85
162
163 myPE: 1 Time = 0.85
164
165 myPE: 0 Time = 0.85
166
167 myPE: 2 Time = 0.9
168
169 myPE: 3 Time = 0.9
170
171 myPE: 1 Time = 0.9
172
173 myPE: 0 Time = 0.9
174
175 myPE: 3 Time = 0.95
176
177 myPE: 0 Time = 0.95
178
179 myPE: 2 Time = 0.95
180
181 myPE: 1 Time = 0.95
182
183 myPE: 3 Time = 1
184
185 myPE: 1 Time = 1
186
187 myPE: 0 Time = 1
188
189 myPE: 2 Time = 1
190
191 myPE: 3 Time = 1.05
192
193 myPE: 1 Time = 1.05
194
195 myPE: 0 Time = 1.05
196
197 myPE: 2 Time = 1.05
198
199 myPE: 3 Time = 1.1
200
```

```
201 myPE: 1 Time = 1.1
202
203 myPE: 0 Time = 1.1
204
205 myPE: 2 Time = 1.1
206
207 myPE: 3 Time = 1.15
208
209 myPE: 1 Time = 1.15
210
211 myPE: 0 Time = 1.15
212
213 myPE: 2 Time = 1.15
214
215 myPE: 3 Time = 1.2
216
217 myPE: 2 Time = 1.2
218
219 myPE: 0 Time = 1.2
220
221 myPE: 1 Time = 1.2
222
223 myPE: 3 Time = 1.25
224
225 myPE: 2 Time = 1.25
226
227 myPE: 0 Time = 1.25
228
229 myPE: 1 Time = 1.25
230
231 myPE: 3 Time = 1.3
232
233 myPE: 2 Time = 1.3
234
235 myPE: 0 Time = 1.3
236
237 myPE: 1 Time = 1.3
238
239 myPE: 3 Time = 1.35
240
241 myPE: 1 Time = 1.35
242
243 myPE: 0 Time = 1.35
244
245 myPE: 2 Time = 1.35
246
247 myPE: 1 Time = 1.4
248
249 myPE: 2 Time = 1.4
250
251 myPE: 0 Time = 1.4
252
253 myPE: 3 Time = 1.4
254
255 myPE: 1 Time = 1.45
```

```
256
257 myPE: 3 Time = 1.45
258
259 myPE: 0 Time = 1.45
260
261 myPE: 2 Time = 1.45
262
263 myPE: 3 Time = 1.5
264
265 myPE: 1 Time = 1.5
266
267 myPE: 0 Time = 1.5
268
269 myPE: 2 Time = 1.5
270
271 myPE: 3 Time = 1.55
272
273 myPE: 2 Time = 1.55
274
275 myPE: 0 Time = 1.55
276
277 myPE: 1 Time = 1.55
278
279 myPE: 3 Time = 1.6
280
281 myPE: 2 Time = 1.6
282
283 myPE: 0 Time = 1.6
284
285 myPE: 1 Time = 1.6
286
287 myPE: 3 Time = 1.65
288
289 myPE: 1 Time = 1.65
290
291 myPE: 0 Time = 1.65
292
293 myPE: 2 Time = 1.65
294
295 myPE: 3 Time = 1.7
296
297 myPE: 2 Time = 1.7
298
299 myPE: 0 Time = 1.7
300
301 myPE: 1 Time = 1.7
302
303 myPE: 3 Time = 1.75
304
305 myPE: 2 Time = 1.75
306
307 myPE: 1 Time = 1.75
308
309 myPE: 0 Time = 1.75
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
319 myPE: 1 Time = 1.85
320
321 myPE: 2 Time = 1.85
322
323 myPE: 0 Time = 1.85
324
325 myPE: 3 Time = 1.85
326
327 myPE: 1 Time = 1.9
328
329 myPE: 2 Time = 1.9
330
331 myPE: 0 Time = 1.9
332
333 myPE: 3 Time = 1.9
334
335 myPE: 2 Time = 1.95
336
337 myPE: 1 Time = 1.95
338
339 myPE: 0 Time = 1.95
340
341 myPE: 3 Time = 1.95
342
343 myPE: 2 Time = 2
344
345 myPE: 3 Time = 2
346
347 myPE: 0 Time = 2
348
349 myPE: 1 Time = 2
350
351 myPE: 2 Time = 2.05
352
353 myPE: 1 Time = 2.05
354
355 myPE: 3 Time = 2.05
356
357 myPE: 0 Time = 2.05
358
359 myPE: 3 Time = 2.1
360
361 myPE: 2 Time = 2.1
362
363 myPE: 1 Time = 2.1
364
365 myPE: 0 Time = 2.1
```

```
366
367 myPE: 2 Time = 2.15
368
369 myPE: 1 Time = 2.15
370
371 myPE: 0 Time = 2.15
372
373 myPE: 3 Time = 2.15
374
375 myPE: 2 Time = 2.2
376
377 myPE: 3 Time = 2.2
378
379 myPE: 0 Time = 2.2
380
381 myPE: 1 Time = 2.2
382
383 myPE: 1 Time = 2.25
384
385 myPE: 3 Time = 2.25
386
387 myPE: 2 Time = 2.25
388
389 myPE: 0 Time = 2.25
390
391 myPE: 3 Time = 2.3
392
393 myPE: 2 Time = 2.3
394
395 myPE: 0 Time = 2.3
396
397 myPE: 1 Time = 2.3
398
399 myPE: 1 Time = 2.35
400
401 myPE: 3 Time = 2.35
402
403 myPE: 2 Time = 2.35
404
405 myPE: 0 Time = 2.35
406
407 myPE: 2 Time = 2.4
408
409 myPE: 1 Time = 2.4
410
411 myPE: 3 Time = 2.4
412
413 myPE: 0 Time = 2.4
414
415 myPE: 1 Time = 2.45
416
417 myPE: 2 Time = 2.45
418
419 myPE: 3 Time = 2.45
420
```



```
421 myPE: 0 Time = 2.45
422
423 myPE: 3 Time = 2.5
424
425 myPE: 2 Time = 2.5
426
427 myPE: 1 Time = 2.5
428
429 myPE: 0 Time = 2.5
430
431 myPE: 3 Time = 2.55
432
433 myPE: 2 Time = 2.55
434
435 myPE: 0 Time = 2.55
436
437 myPE: 1 Time = 2.55
438
439 myPE: 2 Time = 2.6
440
441 myPE: 0 Time = 2.6
442
443 myPE: 3 Time = 2.6
444
445 myPE: 1 Time = 2.6
446
447 myPE: 2 Time = 2.65
448
449 myPE: 3 Time = 2.65
450
451 myPE: 0 Time = 2.65
452
453 myPE: 1 Time = 2.65
454
455 myPE: 3 Time = 2.7
456
457 myPE: 2 Time = 2.7
458
459 myPE: 0 Time = 2.7
460
461 myPE: 1 Time = 2.7
462
463 myPE: 1 Time = 2.75
464
465 myPE: 3 Time = 2.75
466
467 myPE: 0 Time = 2.75
468
469 myPE: 2 Time = 2.75
470
471 myPE: 1 Time = 2.8
472
473 myPE: 3 Time = 2.8
474
475 myPE: 0 Time = 2.8
```

```
476
477 myPE: 2 Time = 2.8
478
479 myPE: 3 Time = 2.85
480
481 myPE: 1 Time = 2.85
482
483 myPE: 0 Time = 2.85
484
485 myPE: 2 Time = 2.85
486
487 myPE: 1 Time = 2.9
488
489 myPE: 0 Time = 2.9
490
491 myPE: 3 Time = 2.9
492
493 myPE: 2 Time = 2.9
494
495 myPE: 1 Time = 2.95
496
497 myPE: 3 Time = 2.95
498
499 myPE: 0 Time = 2.95
500
501 myPE: 2 Time = 2.95
502
503 myPE: 0 Time = 3
504
505 myPE: 3 Time = 3
506
507 myPE: 1 Time = 3
508
509 myPE: 2 Time = 3
510
511 myPE: 0 Time = 3.05
512
513 myPE: 2 Time = 3.05
514
515 myPE: 1 Time = 3.05
516
517 myPE: 3 Time = 3.05
518
519 myPE: 3 Time = 3.1
520
521 myPE: 0 Time = 3.1
522
523 myPE: 1 Time = 3.1
524
525 myPE: 2 Time = 3.1
526
527 myPE: 0 Time = 3.15
528
529 myPE: 3 Time = 3.15
530
```

```
531 myPE: 1 Time = 3.15
532
533 myPE: 2 Time = 3.15
534
535 myPE: 0 Time = 3.2
536
537 myPE: 2 Time = 3.2
538
539 myPE: 1 Time = 3.2
540
541 myPE: 3 Time = 3.2
542
543 myPE: 0 Time = 3.25
544
545 myPE: 2 Time = 3.25
546
547 myPE: 3 Time = 3.25
548
549 myPE: 1 Time = 3.25
550
551 myPE: 2 Time = 3.3
552
553 myPE: 0 Time = 3.3
554
555 myPE: 3 Time = 3.3
556
557 myPE: 1 Time = 3.3
558
559 myPE: 3 Time = 3.35
560
561 myPE: 2 Time = 3.35
562
563 myPE: 0 Time = 3.35
564
565 myPE: 1 Time = 3.35
566
567 myPE: 2 Time = 3.4
568
569 myPE: 3 Time = 3.4
570
571 myPE: 0 Time = 3.4
572
573 myPE: 1 Time = 3.4
574
575 myPE: 0 Time = 3.45
576
577 myPE: 2 Time = 3.45
578
579 myPE: 3 Time = 3.45
580
581 myPE: 1 Time = 3.45
582
583 myPE: 3 Time = 3.5
584
585 myPE: 2 Time = 3.5
```

```
586
587 myPE: 0 Time = 3.5
588
589 myPE: 1 Time = 3.5
590
591 myPE: 3 Time = 3.55
592
593 myPE: 1 Time = 3.55
594
595 myPE: 0 Time = 3.55
596
597 myPE: 2 Time = 3.55
598
599 myPE: 3 Time = 3.6
600
601 myPE: 1 Time = 3.6
602
603 myPE: 0 Time = 3.6
604
605 myPE: 2 Time = 3.6
606
607 myPE: 3 Time = 3.65
608
609 myPE: 0 Time = 3.65
610
611 myPE: 1 Time = 3.65
612
613 myPE: 2 Time = 3.65
614
615 myPE: 3 Time = 3.7
616
617 myPE: 1 Time = 3.7
618
619 myPE: 2 Time = 3.7
620
621 myPE: 0 Time = 3.7
622
623 myPE: 2 Time = 3.75
624
625 myPE: 1 Time = 3.75
626
627 myPE: 3 Time = 3.75
628
629 myPE: 0 Time = 3.75
630
631 myPE: 2 Time = 3.8
632
633 myPE: 0 Time = 3.8
634
635 myPE: 3 Time = 3.8
636
637 myPE: 1 Time = 3.8
638
639 myPE: 3 Time = 3.85
640
```

```
641 myPE: 0 Time = 3.85
642
643 myPE: 1 Time = 3.85
644
645 myPE: 2 Time = 3.85
646
647 myPE: 3 Time = 3.9
648
649 myPE: 0 Time = 3.9
650
651 myPE: 1 Time = 3.9
652
653 myPE: 2 Time = 3.9
654
655 myPE: 3 Time = 3.95
656
657 myPE: 2 Time = 3.95
658
659 myPE: 0 Time = 3.95
660
661 myPE: 1 Time = 3.95
662
663 myPE: 0 Time = 4
664
665 myPE: 2 Time = 4
666
667 myPE: 3 Time = 4
668
669 myPE: 1 Time = 4
670
671 myPE: 2 Time = 4.05
672
673 myPE: 0 Time = 4.05
674
675 myPE: 1 Time = 4.05
676
677 myPE: 3 Time = 4.05
678
679 myPE: 1 Time = 4.1
680
681 myPE: 2 Time = 4.1
682
683 myPE: 0 Time = 4.1
684
685 myPE: 3 Time = 4.1
686
687 myPE: 1 Time = 4.15
688
689 myPE: 2 Time = 4.15
690
691 myPE: 0 Time = 4.15
692
693 myPE: 3 Time = 4.15
694
695 myPE: 1 Time = 4.2
```

```
696
697 myPE: 2 Time = 4.2
698
699 myPE: 0 Time = 4.2
700
701 myPE: 3 Time = 4.2
702
703 myPE: 2 Time = 4.25
704
705 myPE: 0 Time = 4.25
706
707 myPE: 1 Time = 4.25
708
709 myPE: 3 Time = 4.25
710
711 myPE: 0 Time = 4.3
712
713 myPE: 1 Time = 4.3
714
715 myPE: 2 Time = 4.3
716
717 myPE: 3 Time = 4.3
718
719 myPE: 0 Time = 4.35
720
721 myPE: 2 Time = 4.35
722
723 myPE: 1 Time = 4.35
724
725 myPE: 3 Time = 4.35
726
727 myPE: 0 Time = 4.4
728
729 myPE: 1 Time = 4.4
730
731 myPE: 2 Time = 4.4
732
733 myPE: 3 Time = 4.4
734
735 myPE: 0 Time = 4.45
736
737 myPE: 1 Time = 4.45
738
739 myPE: 2 Time = 4.45
740
741 myPE: 3 Time = 4.45
742
743 myPE: 2 Time = 4.5
744
745 myPE: 0 Time = 4.5
746
747 myPE: 1 Time = 4.5
748
749 myPE: 3 Time = 4.5
750
```

```
751 myPE: 2 Time = 4.55
752
753 myPE: 1 Time = 4.55
754
755 myPE: 3 Time = 4.55
756
757 myPE: 0 Time = 4.55
758
759 myPE: 2 Time = 4.6
760
761 myPE: 3 Time = 4.6
762
763 myPE: 1 Time = 4.6
764
765 myPE: 0 Time = 4.6
766
767 myPE: 2 Time = 4.65
768
769 myPE: 3 Time = 4.65
770
771 myPE: 0 Time = 4.65
772
773 myPE: 1 Time = 4.65
774
775 myPE: 2 Time = 4.7
776
777 myPE: 3 Time = 4.7
778
779 myPE: 0 Time = 4.7
780
781 myPE: 1 Time = 4.7
782
783 myPE: 2 Time = 4.75
784
785 myPE: 1 Time = 4.75
786
787 myPE: 3 Time = 4.75
788
789 myPE: 0 Time = 4.75
790
791 myPE: 2 Time = 4.8
792
793 myPE: 1 Time = 4.8
794
795 myPE: 3 Time = 4.8
796
797 myPE: 0 Time = 4.8
798
799 myPE: 2 Time = 4.85
800
801 myPE: 0 Time = 4.85
802
803 myPE: 1 Time = 4.85
804
805 myPE: 3 Time = 4.85
```

```

806
807 myPE: 1 Time = 4.9
808
809 myPE: 0 Time = 4.9
810
811 myPE: 2 Time = 4.9
812
813 myPE: 3 Time = 4.9
814
815 myPE: 1 Time = 4.95
816
817 myPE: 3 Time = 4.95
818
819 myPE: 0 Time = 4.95
820
821 myPE: 2 Time = 4.95
822
823 myPE: 1 Time = 5
824
825 myPE: 3 Time = 5
826
827 myPE: 0 Time = 5
828
829 myPE: 2 Time = 5
830
831
832 ** Successful Completion **

```

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

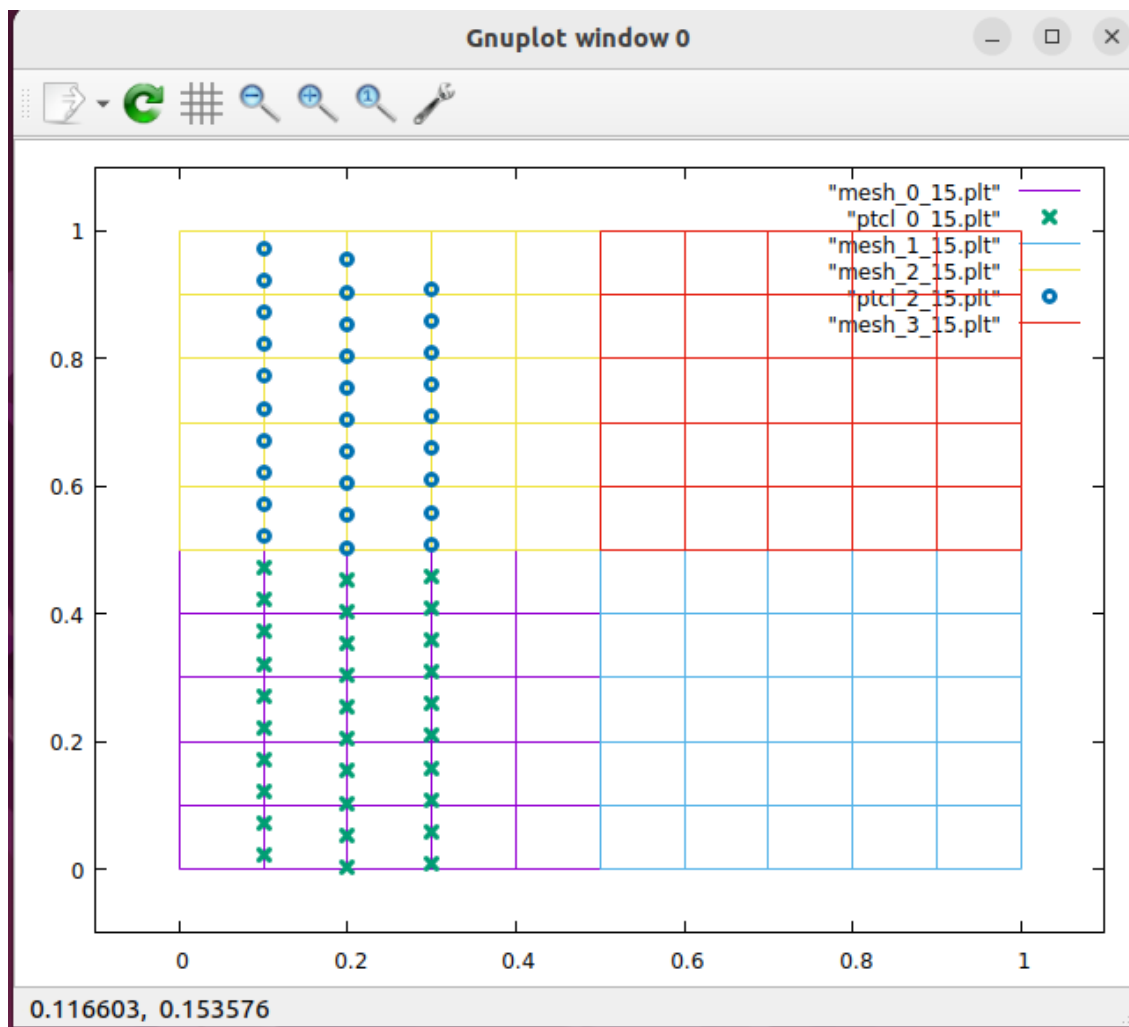


Figure 1: Particle at Start

4.4 Plot Images, Showing Particles Moving

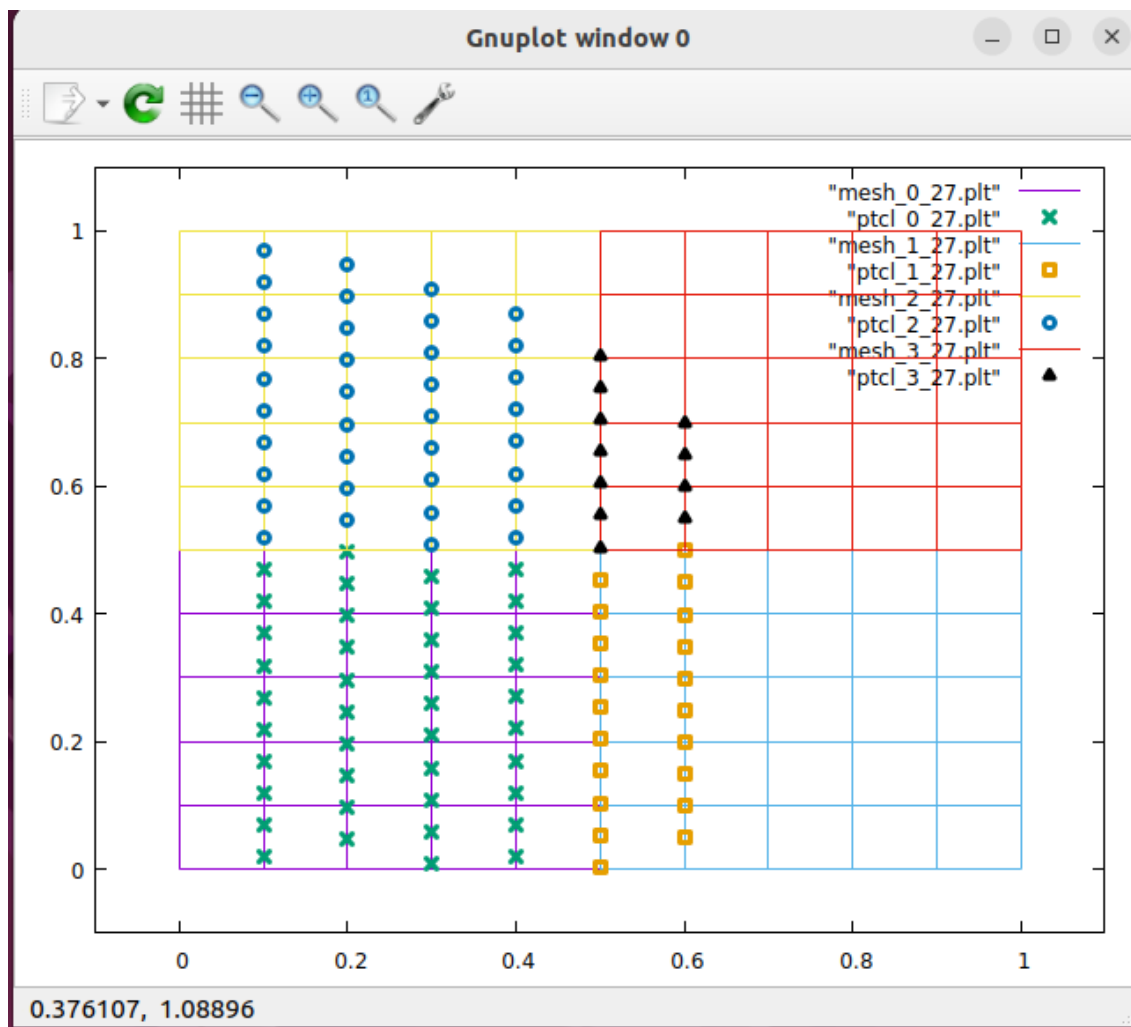


Figure 2: Particles Start Moving to other Processors

4.5 Plot Images, Showing Particles Moving

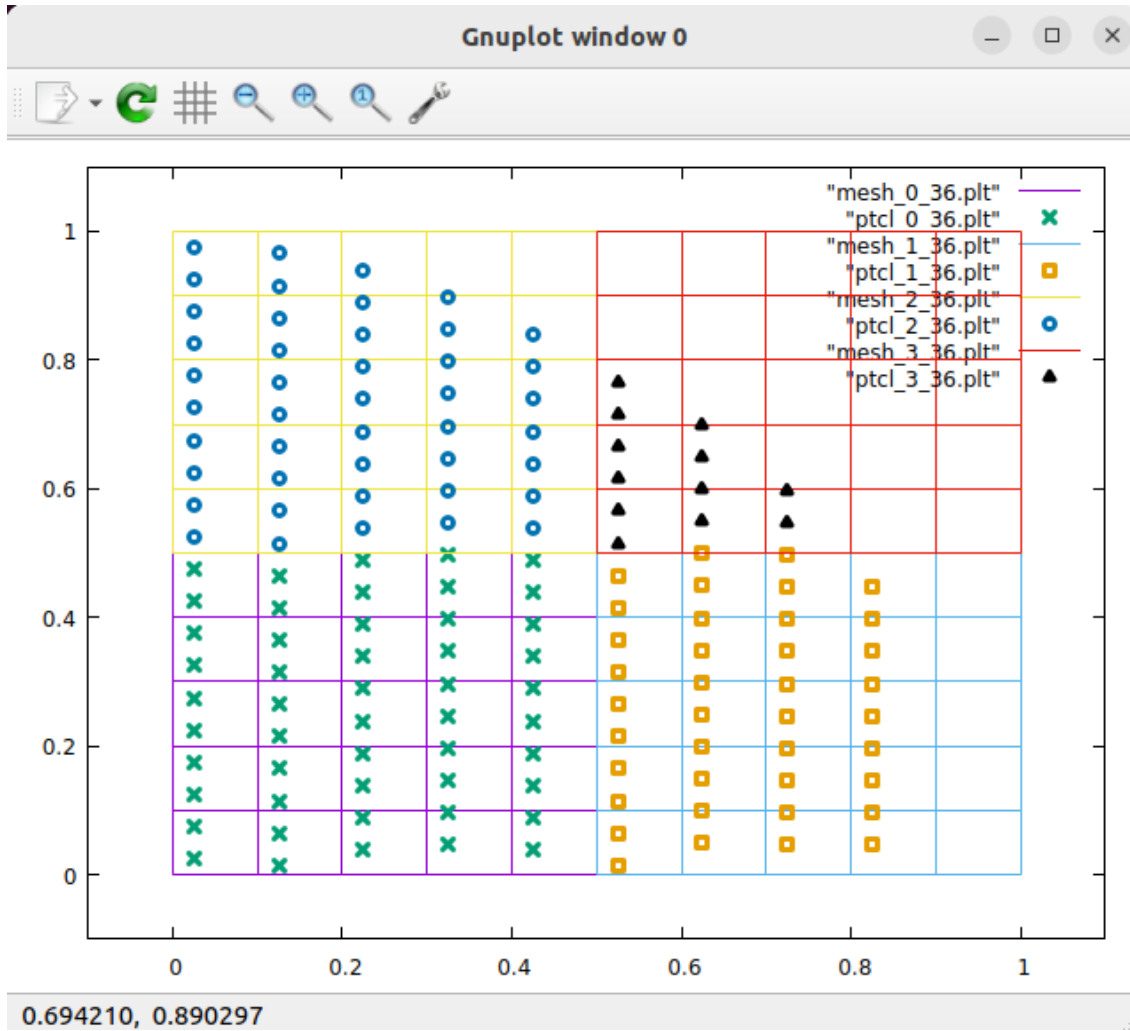


Figure 3: Particles Moving in all Processors

4.6 Debugging Code

```

1 #!/bin/bash
2
3 # -
4 # |
5 # | This is a batch script for running a MPI parallel job on Summit
6 # |
7 # | (o) To submit this job, enter: sbatch --export=CODE='/home/scru5660/HPSC/
   codes/fd_mpi/src' ex_01.bat
8 # |
9 # | (o) To check the status of this job, enter: squeue -u <username>
10 # |
11 # -

```

```

12
13 # -
14 # |
15 # | Part 1: Directives
16 # |
17 # -
18
19 #SBATCH --nodes=1
20 #SBATCH --ntasks=4
21 #SBATCH --time=00:02:00
22 #SBATCH --partition=atesting
23 #SBATCH --output=ex01-%j.out
24
25 # -
26 # |
27 # | Part 2: Loading software
28 # |
29 # -
30
31 module purge
32 module update gcc
33 module load gcc
34 module load openmpi
35 module load impi
36
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 "||"
47 echo "=="
48
49 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 "||"
53 echo "|| Execution of fp in slurm batch script complete."
54 echo "||"
55 echo "=="

```

Listing 6: Testing Slurm File

```

1 //
    =====
2 //  ||
    ||
3 //  ||          fp
    ||

```

```

4 // || -----
5 // || F R E E   P A R T I C L E
6 // ||
7 // || D E M O N S T R A T I O N   C O D E
8 // || -----
9 // ||
10 // || Developed by: Scott R. Runnels, Ph.D.
11 // || University of Colorado Boulder
12 // ||
13 // || For: CU Boulder CSCI 4576/5576 and associated labs
14 // ||
15 // || Copyright 2020 Scott Runnels
16 // ||
17 // || Not for distribution or use outside of the
18 // || this course.
19 // ||
20 // ||
21 =====
22 #include "mpi.h"
23 #include "fp.h"
24 #include "particles.h"
25 #include "mpiInfo.h"
26
27
28 // ==
29 // ||
30 // || C L A S S :   M E S H
31 // ||
32 // ||
33 // ||
34 // || (21) ---- (22) ---- (23) ---- (24) ---- (25)  <--- nRealx+1 , nRealy+1
35 // || |         |         |         |         |
36 // || |         |         |         |         |
37 // || |         |         |         |         |
38 // || (16) ---- [17] ---- [18] ---- [19] ---- (20)
39 // || |         1,3|         2,3|         3,3|         |
40 // || |         |         |         |         |

```

```

41 // || | | | |
42 // || (11)----[12]----[13]----[14]----(15)
43 // || | 1,2| 2,2| 3,2| |
44 // || | | | | |
45 // || | | | | |
46 // || (6)----[7]----[8]----[9]----(10)
47 // || | 1,1| 2,1| 3,1| |
48 // || | | | | |
49 // || | | | | |
50 // || (1)----(2)----(3)----(4)----(5)
51 // || 0,0 | | | | 4,0
52 // ||
53 // ==
54
55 class Mesh
56 {
57
58 public:
59
60 double x0, x1, y0, y1;
61 VD x,y;
62 int nRealx , nRealy , nField, nReal;
63 double lengthx, lengthy, dx, dy;
64 int myPE;
65 VD Qval;
66
67 // ==
68 // ||
69 // || Constructor: Initialize values
70 // ||
71 // ==
72
73 Mesh(double _x0 , double _x1, double _y0, double _y1 , int ncell_x , int
ncell_y, mpiInfo &myMPI )
74 {
75
76
77 // Copy incoming values
78
79 x0 = _x0; x1 = _x1;
80 y0 = _y0; y1 = _y1;
81 myPE = myMPI.myPE;
82
83 // Compute number of real (physical) nodes, and their spacing
84
85 nRealx = ncell_x + 1;
86 nRealy = ncell_y + 1;
87 nReal = nRealx*nRealy;
88 dx = (x1-x0) / ncell_x;
89 dy = (y1-y0) / ncell_y;
90
91 // Compute the size of the field variables, which also lie on ghost nodes
92
93 nField = (nRealx+2)*(nRealy+2);
94

```

```

95 // Allocate memory -- Note that the node numbers and field variable numbers
must
96 // match. So even though we only will be caring about real nodes, their node
97 // numbers are naturally ordered, so must be of size nField.
98
99 x.resize(nField+1); y.resize(nField+1); Qval.resize(nField+1);
100
101 for ( int i = 1 ; i <= nRealx ; ++i )
102     for ( int j = 1 ; j <= nRealy ; ++j )
103     {
104         int p = pid(i,j);
105         x[p] = x0 + (i-1)*dx;
106         y[p] = y0 + (j-1)*dy;
107     }
108
109 }
110
111
112
113 // ==
114 // ||
115 // || ParticlesOnMesh
116 // ||
117 // || Quantify the particles' interaction with the mesh
118 // ||
119 // || References: [1] https://www.particleincell.com/2010/es-pic-method/
120 // ||
121 // ==
122
123 void ParticlesOnMesh(particles &PTCL, mpiInfo &myMPI)
124 {
125     double hx, hy;
126     double w[5];
127     int p[5];
128     int iL, iR, jB, jT;
129     int iPEnew, jPEnew; // These store the i-j indicies of the
processor receiving a particle,
130 // if that particle is leaving the mesh.
131 VI ptcl_send_list, ptcl_send_PE; // These collect information about particles
that
132 // have left this processor and are
heading onto
133 // another processor.
134
135 // -
136 // |
137 // | Determine which particles are still on this mesh and which have left
138 // |
139 // -
140
141 for ( int k = 1 ; k <= PTCL.n ; ++k )
142     {
143 // First, check to be sure the particle is still in the mesh. If it is not,
set its
144 // "active" flag to zero, and note the processor to which it is going for MPI

```

```

exchange.
145
146 if ( PTCL.active[k] == 1 )
147 {
148     iPEnew = myMPI.iPE;
149     jPEnew = myMPI.jPE;
150
151
152     if ( PTCL.x[k] < x0      ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE - 1 ;
153 }
154     if ( PTCL.x[k] > x1      ) { PTCL.active[k] = -1;  iPEnew = myMPI.iPE + 1 ;
155 }
156     if ( PTCL.y[k] < y0      ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE - 1 ;
157 }
158     if ( PTCL.y[k] > y1      ) { PTCL.active[k] = -1;  jPEnew = myMPI.jPE + 1 ;
159 }
160
161 // The particle is not in the mesh.  Collect this particle into a holding array
162 // that will
163 // be sent to the neighboring processor.
164
165 if ( PTCL.active[k] == -1)
166 {
167     if ( iPEnew >= 0 && iPEnew < myMPI.nPEx )
168     if ( jPEnew >= 0 && jPEnew < myMPI.nPEy )
169     {
170         ptcl_send_list.push_back(k);
171         ptcl_send_PE .push_back(iPEnew + jPEnew*nRealx);
172     }
173
174     PTCL.active[k] = 0;  // Remove it from the list of active particles
175 }
176
177 // -
178 // |
179 // | Give and receive particles to/with other processors
180 // |
181 // -
182
183 myMPI.ParticleExchange( ptcl_send_list , ptcl_send_PE , PTCL);
184 cout << "After Exchange" << endl;
185
186 // -
187 // |
188 // | Accumulate particles to the nodes (to be completed in the esPIC code,
189 // | next week)
190 // |
191 // -
192
193 for ( int k = 1 ; k <= nField ; ++k ) Qval[k] = 0.;

```



```

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

```

```

248 // -
249
250 if ( myMPI.myPE == 0 )
251 {
252     cout << "\n";
253     cout << "-----\n";
254     cout << "\n";
255     cout << " F R E E   P A R T I C L E           \n";
256     cout << " D E M O   C O D E           \n";
257     cout << "\n";
258     cout << " Running on " << myMPI.numPE << " processors \n";
259     cout << "\n";
260     cout << "-----\n";
261     cout << "\n";
262 }
263
264 for (int count = 0 ; count < argc; ++count)
265 {
266     if ( !strcmp(argv[count], "-nPEx" ) ) nPEx = atoi(argv[count+1]);
267     if ( !strcmp(argv[count], "-nPey" ) ) nPEy = atoi(argv[count+1]);
268     if ( !strcmp(argv[count], "-nCellx" ) ) nCellx = atoi(argv[count+1]);
269     if ( !strcmp(argv[count], "-nCelly" ) ) nCelly = atoi(argv[count+1]);
270     if ( !strcmp(argv[count], "-flux" ) ) flux = atof(argv[count+1]);
271     if ( !strcmp(argv[count], "-tEnd" ) ) tEnd = atof(argv[count+1]);
272     if ( !strcmp(argv[count], "-dt" ) ) dt = atof(argv[count+1]);
273 }
274
275 if ( myMPI.myPE == 0 )
276 {
277     cout << endl;
278     cout << "Input Summary: " << endl;
279     cout << "----- " << endl;
280     cout << "No. PE in x-direction: " << nPEx << endl;
281     cout << " y-direction: " << nPEy << endl;
282     cout << "No. Cells in x-direction: " << nCellx << endl;
283     cout << " y-direction: " << nCelly << endl;
284     cout << "Flux density : " << flux << endl;
285     cout << "End Time : " << tEnd << endl;
286     cout << "Time Step : " << dt << endl;
287     cout << endl;
288 }
289
290 // -
291 // |
292 // | MPI / Processor ID
293 // |
294 // -
295
296 myMPI.GridDecomposition(nPEx, nPEy, nCellx, nCelly);
297
298 // -
299 // |
300 // | Parallel Grid Generation
301 // |
302 // -

```

```

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

```

```

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

```

Listing 7: Testing fp

```

1 // ==
2 // ||
3 // || C L A S S :   m p i I n f o
4 // ||
5 // ==
6
7 class mpiInfo
8 {
9     public:
10
11     int myPE;
12     int numPE;
13     int nRealx, nRealy;
14     int nx, ny;
15     int nPEEx, nPEy;
16     int iPE , jPE;
17     int iMin, iMax, jMin, jMax ; // The global i-j numbers on this processor
18     int nei_n, nei_s, nei_e, nei_w;
19     int nei_ne, nei_nw, nei_se, nei_sw;
20
21     int countx, county;
22
23     MPI_Status status;
24     int err;
25     int tag;
26     MPI_Request request;
27
28 // -
29 // |
30 // | GridDecomposition: Set up PE numbering system in figure below and
31 // | establish communication arrays.
32 // |
33 // | nPEEx -- number of PEs in the x-direction
34 // | nPEy -- number of PEs in the y-direction
35 // | numPE = total number of PEs
36 // |
37 // |
38 // |
39 // |

```

+-----+	+-----+	. . .	+-----+
			numPE


```

93     nei_s = myPE - nPEx ;
94 }
95 if ( iPE < nPEx-1 )
96 {
97     nei_e = myPE + 1 ;
98 }
99 if ( jPE < nPEy-1 )
100 {
101     nei_n = myPE + nPEx ;
102 }
103
104 nei_nw = nei_sw = nei_ne = nei_se = -1;
105
106 if ( iPE > 0      && jPE > 0      ) nei_sw = myPE - nPEx - 1 ;
107 if ( iPE < nPEx-1 && jPE > 0      ) nei_se = myPE - nPEx + 1 ;
108 if ( iPE > 0      && jPE < nPEy-1 ) nei_nw = myPE + nPEx - 1 ;
109 if ( iPE < nPEx-1 && jPE < nPEy-1 ) nei_ne = myPE + nPEx + 1 ;
110
111 // Acquire memory for the communication between adjacent processors:
112 countx = nRealx + 2;
113 county = nRealy + 2;
114
115 tag = 0;
116 }
117
118
119
120 // ==
121 // ||
122 // ||  ParticlesExchange
123 // ||
124 // ||  Exchange particles between processors
125 // ||
126 // ||
127 // ==
128
129 void ParticleExchange( VI &ptcl_send_list , VI &ptcl_send_PE , particles &PTCL)
130 {
131     MPI_Status  status;
132     MPI_Request request;
133
134     // (1) Get the max number particles to be sent by any particular processor,
135     // and make sure all processors know that number.
136
137     int numToSend = ptcl_send_list.size();      int maxToSend;
138     cout << "Before Reduce" << endl;
139     /* TO-DO Place MPI call here to complete (1) */
140     MPI_Barrier(MPI_COMM_WORLD);
141     MPI_Iallreduce(&numToSend, &maxToSend, 1, MPI_INT, MPI_MAX, MPI_COMM_WORLD, &
142     request); MPI_Wait(&request,&status);
143     MPI_Barrier(MPI_COMM_WORLD);
144     cout << "After Reduce, Before C Particles" << endl;
145
146     // (2) Allocate contributions to the upcoming Gather operation. Here, "C" for
147     // "Contribution" to be Gathered

```

```

145
146     int      *Cptcl_PE;  Cptcl_PE = new int      [ maxToSend ];  // Particles'
destination PEs
147     double *Cptcl_x ;  Cptcl_x  = new double [ maxToSend ];
148     double *Cptcl_y ;  Cptcl_y  = new double [ maxToSend ];
149     double *Cptcl_vx;  Cptcl_vx = new double [ maxToSend ];
150     double *Cptcl_vy;  Cptcl_vy = new double [ maxToSend ];
151
152     // (3) Populate contributions on all processors for the upcoming Gather
operation
153
154     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
157     // (4) Populate with all the particles on this PE.  Note that some/most
processors will have left-over space in the C* arrays.
158
159     for ( int i = 0 ; i < ptcl_send_list.size() ; ++i )
160     {
161     int id      = ptcl_send_list[ i ];
162     Cptcl_PE[i] = ptcl_send_PE  [ i ];
163     Cptcl_x [i] = PTCL.x      [ id ];
164     Cptcl_y [i] = PTCL.y      [ id ];
165     Cptcl_vx[i] = PTCL.vx     [ id ];
166     Cptcl_vy[i] = PTCL.vy     [ id ];
167     }
168     cout << "After C Particles" << endl;
169
170     // (5) Allocate and initialize the arrays for upcoming Gather operation to PE0
.  The sizeofGather takes
171     //      into account the number of processors, like this figure:
172     //
173     //      |<----- sizeOfGather
----->|
174     //      |
175     //      |
176     //      |<- maxToSend    ->|<- maxToSend    ->|<- maxToSend    ->|<- maxToSend
->|
177     //
+-----+-----+-----+-----+
178     //      PE0              PE1              PE2              PE3
179     cout << "Before Gather Particles" << endl;
180
181     int sizeofGather = maxToSend * numPE;
182
183     int      *Gptcl_PE;  Gptcl_PE = new int      [ sizeofGather ];
184     double *Gptcl_x ;  Gptcl_x  = new double [ sizeofGather ];
185     double *Gptcl_y ;  Gptcl_y  = new double [ sizeofGather ];
186     double *Gptcl_vx;  Gptcl_vx = new double [ sizeofGather ];
187     double *Gptcl_vy;  Gptcl_vy = new double [ sizeofGather ];
188
189     for ( int i = 0 ; i < sizeofGather ; ++i ) { Gptcl_PE[i] = -1; Gptcl_x [i] =

```

```

190 0.; Gptcl_y [i] = 0.; Gptcl_vx[i] = 0.; Gptcl_vy[i] = 0.; }
191 cout << "After Gather Particles" << endl;
192
193 // (6) Gather "Contributions" ("C" arrays) from all PEs onto all PEs into
194 // these bigger arrays so all PE will know what particles
195 // need to go where.
196
197 cout << "Before Gather" << endl;
198 MPI_Barrier(MPI_COMM_WORLD);
199
200 MPI_Iallgather( Cptcl_PE, sizeofGather + 1, MPI_INT, Gptcl_PE, maxToSend + 1,
201 MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
202 MPI_Iallgather( Cptcl_x, sizeofGather + 1, MPI_INT, Gptcl_x, maxToSend + 1,
203 MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
204 MPI_Iallgather( Cptcl_y, sizeofGather + 1, MPI_INT, Gptcl_y, maxToSend + 1,
205 MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
206 MPI_Iallgather( Cptcl_vx, sizeofGather + 1, MPI_INT, Gptcl_vx, maxToSend + 1,
207 MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
208 MPI_Iallgather( Cptcl_vy, sizeofGather + 1, MPI_INT, Gptcl_vy, maxToSend + 1,
209 MPI_INT, MPI_COMM_WORLD, &request ); MPI_Wait(&request,&status);
210
211 cout << "After Gather" << endl;
212 MPI_Barrier(MPI_COMM_WORLD);
213
214 // (7) Put in vector form so they can be added to PTCL. These arrays are 1-
215 // based.
216
217 cout << "Before Adds" << endl;
218
219 int Np = 0; for ( int i = 0 ; i < sizeofGather ; ++i ) if ( Gptcl_PE[i] ==
220 myPE ) ++Np;
221
222 VD std_add_x ; std_add_x.resize ( Np+1 );
223 VD std_add_y ; std_add_y.resize ( Np+1 );
224 VD std_add_vx ; std_add_vx.resize ( Np+1 );
225 VD std_add_vy ; std_add_vy.resize ( Np+1 );
226
227 cout << "After adds, Before Assignment" << endl;
228 int count = 1;
229 for ( int i = 0 ; i < sizeofGather ; ++i )
230 if ( Gptcl_PE[i] == myPE )
231 {
232     std_add_x [ count ] = Gptcl_x[i];
233     std_add_y [ count ] = Gptcl_y [i];
234     std_add_vx[ count ] = Gptcl_vx[i];
235     std_add_vy[ count ] = Gptcl_vy[i];
236     ++ count;
237 }
238
239 cout << "Before Particle Add" << endl;
240
241 PTCL.add( std_add_x, std_add_y, std_add_vx, std_add_vy );
242
243 cout << "After Particle Add" << endl;

```



```

236
237 // (8) Free up memory
238
239 if (maxToSend > 0 ) { delete[] Cptcl_PE; delete[] Cptcl_x ; delete[]
Cptcl_y ; delete[] Cptcl_vx ; delete[] Cptcl_vy; }
240 if (sizeofGather > 0 ) { delete[] Gptcl_PE; delete[] Gptcl_x ; delete[]
Gptcl_y ; delete[] Gptcl_vx ; delete[] Gptcl_vy; }
241
242 }
243
244
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).

```

1 Lmod has detected the following error: These module(s) or extension(s) exist
2 but cannot be loaded as requested: "impi"
3 Try: "module spider impi" to see how to load the module(s).
4
5
6
7 ==
8 ||
9 || Begin Execution of fp in slurm batch script.
10 ||
11 ==
12 free(): invalid next size (fast)
13 [c3cpu-a5-u34-3:1351127] *** Process received signal ***
14 [c3cpu-a5-u34-3:1351127] Signal: Aborted (6)
15 [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]
21 [c3cpu-a5-u34-3:1351127] [ 5] /usr/lib64/libc.so.6(+0x99b38)[0x15303e7f4b38]
22 [c3cpu-a5-u34-3:1351127] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x404acf]

```

```

23 [c3cpu-a5-u34-3:1351127] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4051eb]
24 [c3cpu-a5-u34-3:1351127] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x402c78]
25 [c3cpu-a5-u34-3:1351127] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
    x15303e795cf3]
26 [c3cpu-a5-u34-3:1351127] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4023be]
27 [c3cpu-a5-u34-3:1351127] *** End of error message ***
28 free(): invalid next size (fast)
29 [c3cpu-a5-u34-3:1351124] *** Process received signal ***
30 [c3cpu-a5-u34-3:1351124] Signal: Aborted (6)
31 [c3cpu-a5-u34-3:1351124] Signal code: (-6)
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]
38 [c3cpu-a5-u34-3:1351124] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x404b08]
39 [c3cpu-a5-u34-3:1351124] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4051eb]
40 [c3cpu-a5-u34-3:1351124] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x402c78]
41 [c3cpu-a5-u34-3:1351124] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
    x14731d424cf3]
42 [c3cpu-a5-u34-3:1351124] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    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)
48 [c3cpu-a5-u34-3:1351125] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x150719137ce0]
49 [c3cpu-a5-u34-3:1351125] [ 1] /usr/lib64/libc.so.6(gsignal+0x10f)[0x150718daea9f]
50 [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]
52 [c3cpu-a5-u34-3:1351125] [ 4] /usr/lib64/libc.so.6(+0x9819c)[0x150718df819c]
53 [c3cpu-a5-u34-3:1351125] [ 5] /usr/lib64/libc.so.6(+0x99b38)[0x150718df9b38]
54 [c3cpu-a5-u34-3:1351125] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x404acf]
55 [c3cpu-a5-u34-3:1351125] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4051eb]
56 [c3cpu-a5-u34-3:1351125] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x402c78]
57 [c3cpu-a5-u34-3:1351125] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
    x150718d9acf3]
58 [c3cpu-a5-u34-3:1351125] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4023be]
59 [c3cpu-a5-u34-3:1351125] *** End of error message ***
60 free(): invalid next size (fast)
61 [c3cpu-a5-u34-3:1351126] *** Process received signal ***
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]
70 [c3cpu-a5-u34-3:1351126] [ 6] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x404acf]
71 [c3cpu-a5-u34-3:1351126] [ 7] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4051eb]
72 [c3cpu-a5-u34-3:1351126] [ 8] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x402c78]
73 [c3cpu-a5-u34-3:1351126] [ 9] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
    x14ebdf7cdcf3]
74 [c3cpu-a5-u34-3:1351126] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
    x4023be]
75 [c3cpu-a5-u34-3:1351126] *** End of error message ***
76 -----
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 -----
81 mpirun noticed that process rank 2 with PID 1351126 on node c3cpu-a5-u34-3 exited
    on signal 6 (Aborted).
82 -----
83 ==
84 ||
85 || Execution of fp in slurm batch script complete.
86 ||
87 ==

```

Listing 9: Next Size Error

```

1 Lmod has detected the following error: These module(s) or extension(s) exist
2 but cannot be loaded as requested: "impi"
3 Try: "module spider impi" to see how to load the module(s).
4
5
6
7 ==
8 ||
9 || Begin Execution of fp in slurm batch script.
10 ||
11 ==
12 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]
21 [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]

```

```

23 [c3cpu-a5-u34-3:1336616] [ 7] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
lib/libmpi.so.40(+0x1043b0)[0x14c3185683b0]
24 [c3cpu-a5-u34-3:1336616] [ 8] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
lib/libmpi.so.40(+0x108de9)[0x14c31856cde9]
25 [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]
26 [c3cpu-a5-u34-3:1336616] [10] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
lib/libmpi.so.40(PMPI_Iallgather+0x105)[0x14c3184f9585]
27 [c3cpu-a5-u34-3:1336616] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x40466e]
28 [c3cpu-a5-u34-3:1336616] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x4051f9]
29 [c3cpu-a5-u34-3:1336616] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x402c78]
30 [c3cpu-a5-u34-3:1336616] [14] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
x14c317505cf3]
31 [c3cpu-a5-u34-3:1336616] [15] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x4023be]
32 [c3cpu-a5-u34-3:1336616] *** End of error message ***
33 malloc(): unsorted double linked list corrupted
34 [c3cpu-a5-u34-3:1336615] *** Process received signal ***
35 [c3cpu-a5-u34-3:1336615] Signal: Aborted (6)
36 [c3cpu-a5-u34-3:1336615] Signal code: (-6)
37 [c3cpu-a5-u34-3:1336615] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x1514dfb49ce0]
38 [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]
43 [c3cpu-a5-u34-3:1336615] [ 6] /usr/lib64/libc.so.6(__libc_callocc+0x86)[0
x1514df80f486]
44 [c3cpu-a5-u34-3:1336615] [ 7] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
lib/libmpi.so.40(+0x1043b0)[0x1514e080f3b0]
45 [c3cpu-a5-u34-3:1336615] [ 8] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
lib/libmpi.so.40(+0x108de9)[0x1514e0813de9]
46 [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]
47 [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]
48 [c3cpu-a5-u34-3:1336615] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x40466e]
49 [c3cpu-a5-u34-3:1336615] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x4051f9]
50 [c3cpu-a5-u34-3:1336615] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x402c78]
51 [c3cpu-a5-u34-3:1336615] [14] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
x1514df7accf3]
52 [c3cpu-a5-u34-3:1336615] [15] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
x4023be]
53 [c3cpu-a5-u34-3:1336615] *** End of error message ***
54 -----
55 Primary job terminated normally, but 1 process returned
56 a non-zero exit code. Per user-direction, the job has been aborted.
57 -----
58 -----

```

```

59 mpirun noticed that process rank 3 with PID 1336616 on node c3cpu-a5-u34-3 exited
   on signal 6 (Aborted).
60 -----
61 ==
62 ||
63 || Execution of fp in slurm batch script complete.
64 ||
65 ==

```

Listing 10: Next to Previous Size Error

```

1  Lmod has detected the following error: These module(s) or extension(s) exist
2  but cannot be loaded as requested: "impi"
3      Try: "module spider impi" to see how to load the module(s).
4
5
6
7  ==
8  ||
9  || 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)
16 [c3cpu-a5-u34-3:1350630] [ 0] /usr/lib64/libpthread.so.0(+0x12ce0)[0x145b3d09cce0]
17 [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 [c3cpu-a5-u34-3:1350630] [ 5] /usr/lib64/libc.so.6(realloc+0x23a)[0x145b3cd6212a]
22 [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]
23 [c3cpu-a5-u34-3:1350630] [ 7] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
   lib/libmpi.so.40(+0x109053)[0x145b3dd67053]
24 [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]
25 [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]
26 [c3cpu-a5-u34-3:1350630] [10] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x40466d]
27 [c3cpu-a5-u34-3:1350630] [11] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x4051eb]
28 [c3cpu-a5-u34-3:1350630] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x402c78]
29 [c3cpu-a5-u34-3:1350630] [13] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
   x145b3ccffcf3]
30 [c3cpu-a5-u34-3:1350630] [14] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x4023be]
31 [c3cpu-a5-u34-3:1350630] *** End of error message ***
32 -----
33 Primary job  terminated normally, but 1 process returned
34 a non-zero exit code. Per user-direction, the job has been aborted.
35 -----
36 -----

```

```

37 mpirun noticed that process rank 2 with PID 1350630 on node c3cpu-a5-u34-3 exited
   on signal 6 (Aborted).
38 -----
39 ==
40 ||
41 || Execution of fp in slurm batch script complete.
42 ||
43 ==

```

Listing 11: Invalid Pointer

```

1 Lmod has detected the following error: These module(s) or extension(s) exist
2 but cannot be loaded as requested: "impi"
3 Try: "module spider impi" to see how to load the module(s).
4
5
6
7 ==
8 ||
9 || Begin Execution of fp in slurm batch script.
10 ||
11 ==
12 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 ***
14 [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 [c3cpu-a5-u34-3:1362180] [ 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]
24 [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+0xfcb)[0x14ea357c279b]
25 [c3cpu-a5-u34-3:1362180] [ 9] /curc/sw/install/openmpi/4.1.1/gcc/11.2.0_slurmpmi/
   lib/libmpi.so.40(+0x109053)[0x14ea357c7053]
26 [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]
27 [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]
28 [c3cpu-a5-u34-3:1362180] [12] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x404713]
29 [c3cpu-a5-u34-3:1362180] [13] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x405319]
30 [c3cpu-a5-u34-3:1362180] [14] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x402c78]
31 [c3cpu-a5-u34-3:1362180] [15] /usr/lib64/libc.so.6(__libc_start_main+0xf3)[0
   x14ea3475fcf3]
32 [c3cpu-a5-u34-3:1362180] [16] /home/trmc7708/Lab3/HPSC_Lab3/Lab3/fp_mpi/src/fp[0
   x4023be]
33 [c3cpu-a5-u34-3:1362180] *** End of error message ***
34 -----

```

```

35 Primary job terminated normally, but 1 process returned
36 a non-zero exit code. Per user-direction, the job has been aborted.
37 -----
38 -----
39 mpirun noticed that process rank 1 with PID 1362180 on node c3cpu-a5-u34-3 exited
   on signal 6 (Aborted).
40 -----
41 ==
42 ||
43 || Execution of fp in slurm batch script complete.
44 ||
45 ==

```

Listing 12: Old Top

```

1 Lmod has detected the following error: These module(s) or extension(s) exist
2 but cannot be loaded as requested: "impi"
3   Try: "module spider impi" to see how to load the module(s).
4
5
6
7 ==
8 ||
9 || Begin Execution of fp in slurm batch script.
10 ||
11 ==
12 [c3cpu-a5-u34-3:1352023:0:1352023] Caught signal 11 (Segmentation fault: address
   not mapped to object at address 0x100)
13 BFD: Dwarf Error: Can't find .debug_ranges section.
14 BFD: Dwarf Error: Can't find .debug_ranges section.
15 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) ====
22 0 0x00000000000012ce0 __funlockfile() :0
23 1 0x00000000000203ec6 mca_pml_ucx_isend() ???:0
24 2 0x000000000001055f1 NBC_Start_round() nbc.c:0
25 3 0x00000000000105ab7 NBC_Start() ???:0
26 4 0x0000000000010936a ompi_coll_libnbc_iallgather() ???:0
27 5 0x00000000000095585 PMPI_Iallgather() ???:0
28 6 0x000000000004046be mpiInfo::ParticleExchange() ???:0
29 7 0x000000000004051eb Mesh::ParticlesOnMesh() ???:0
30 8 0x00000000000402c78 main() ???:0
31 9 0x0000000000003acf3 __libc_start_main() ???:0
32 10 0x000000000004023be _start() ???:0
33 =====
34 -----
35 Primary job terminated normally, but 1 process returned
36 a non-zero exit code. Per user-direction, the job has been aborted.
37 -----
38 -----
39 mpirun noticed that process rank 2 with PID 1352023 on node c3cpu-a5-u34-3 exited
   on signal 11 (Segmentation fault).

```

```

40 -----
41 ==
42 ||
43 || Execution of fp in slurm batch script complete.
44 ||
45 ==

```

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

```

1
2 myPE: 3 Time = 0
3 Before Reduce
4
5 myPE: 1 Time = 0
6 Before Reduce
7
8 myPE: 2 Time = 0
9 Before Reduce
10
11 -----
12
13 F R E E   P A R T I C L E
14 D E M O   C O D E
15
16 Running on 4 processors
17
18 -----
19
20
21 Input Summary:
22 -----
23 No. PE    in  x-direction: 2
24                y-direction: 2
25 No. Cells in x-direction: 5
26                y-direction: 5
27 Flux density          : 10
28 End Time              : 3
29 Time Step             : 0.01
30
31
32 myPE: 0 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

```

...

```

1 myPE: 3 Time = 0.55
2 Before Reduce
3
4 myPE: 1 Time = 0.55
5 Before Reduce
6
7 myPE: 0 Time = 0.55
8 Before Reduce
9 After Reduce, Before C Particles
10 After C Particles
11 Before Gather Particles
12 After Gather Particles
13 Before Gather
14 After Reduce, Before C Particles
15 After C Particles
16 Before Gather Particles
17 After Gather Particles

```

```

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

```

Listing 14: tty.out

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

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

```

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
4 ==12158== Command: mpirun --oversubscribe -np 4 ./fp -nPEx 2 -nPEy 2 -nCellx 5 -
    nCelly 5 -flux 10 -tEnd 3 -dt 0.01
5 ==12158== Parent PID: 2865
6 ==12158==
7 ==12158== Warning: noted but unhandled ioctl 0x5441 with no size/direction hints.
8 ==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()
14 ==12162==     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()
18 ==12163== Warning: invalid file descriptor 1048565 in syscall close()
19 ==12163== Warning: invalid file descriptor 1048566 in syscall close()
20 ==12163== Warning: invalid file descriptor 1048567 in syscall close()
21 ==12163==     Use --log-fd=<number> to select an alternative log fd.
22 ==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()
38 ==12158==
39 ==12158== HEAP SUMMARY:
40 ==12158== in use at exit: 20,396 bytes in 128 blocks
41 ==12158== total heap usage: 27,478 allocs, 27,350 frees, 12,255,873 bytes
    allocated
42 ==12158==
43 ==12158== 32 bytes in 1 blocks are definitely lost in loss record 24 of 73
44 ==12158== at 0x4C37135: malloc (vg_replace_malloc.c:381)
45 ==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)
46 ==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)
47 ==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)
48 ==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)
49 ==12158== by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
50 ==12158== by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
51 ==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)
56 ==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)
57 ==12158== by 0x4EA1F77: orte_pmix_server_register_nspace (in /curc/sw/install/
    openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
58 ==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)
59 ==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)
60 ==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)
61 ==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)
62 ==12158== by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
63 ==12158== by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
64 ==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)
66 ==12158==
67 ==12158== 321 (136 direct, 185 indirect) bytes in 1 blocks are definitely lost in
    loss record 53 of 73
68 ==12158== at 0x4C37135: malloc (vg_replace_malloc.c:381)
69 ==12158== by 0x4E817B2: show_help (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
    _slurmpmi/lib/libopen-rte.so.40.30.1)
70 ==12158== by 0x4E81AE6: orte_show_help_recv (in /curc/sw/install/openmpi/4.1.1/

```

```

gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
71 ==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)
72 ==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
73 ==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
74 ==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
    _slurmpmi/bin/orterun)
75 ==12158==    by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
76 ==12158==
77 ==12158== 569 (64 direct, 505 indirect) bytes in 1 blocks are definitely lost in
    loss record 62 of 73
78 ==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
79 ==12158==    by 0x4F204CF: assign (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
    _slurmpmi/lib/libopen-rte.so.40.30.1)
80 ==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)
81 ==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)
82 ==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)
83 ==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)
84 ==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)
85 ==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
86 ==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
87 ==12158==    by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
    _slurmpmi/bin/orterun)
88 ==12158==    by 0x8674CF2: (below main) (in /usr/lib64/libc-2.28.so)
89 ==12158==
90 ==12158== 962 (96 direct, 866 indirect) bytes in 2 blocks are definitely lost in
    loss record 67 of 73
91 ==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)
92 ==12158==    by 0x532DCD0: new_tracker.constprop.0 (in /curc/sw/install/openmpi
    /4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
93 ==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)
94 ==12158==    by 0x532417C: server_switchyard (in /curc/sw/install/openmpi/4.1.1/
    gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
95 ==12158==    by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/
    openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
96 ==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)
97 ==12158==    by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
98 ==12158==    by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
99 ==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)
100 ==12158==    by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
101 ==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
104 ==12158==    at 0x4C37135: malloc (vg_replace_malloc.c:381)

```

```

105 ==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)
106 ==12158==      by 0x51DAAD0: opal_dss_pack_int32 (in /curc/sw/install/openmpi/4.1.1/
    gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
107 ==12158==      by 0x51DAB45: opal_dss_pack (in /curc/sw/install/openmpi/4.1.1/gcc
    /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
108 ==12158==      by 0x4EA2BBB: orte_pmix_server_register_nspace (in /curc/sw/install/
    openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-rte.so.40.30.1)
109 ==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)
110 ==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)
111 ==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)
112 ==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)
113 ==12158==      by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
114 ==12158==      by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
115 ==12158==      by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
    _slurmpmi/bin/orterun)
116 ==12158==
117 ==12158== 2,048 bytes in 1 blocks are definitely lost in loss record 71 of 73
118 ==12158==      at 0x4C37135: malloc (vg_replace_malloc.c:381)
119 ==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)
120 ==12158==      by 0x51DAAD0: opal_dss_pack_int32 (in /curc/sw/install/openmpi/4.1.1/
    gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
121 ==12158==      by 0x51DAB45: opal_dss_pack (in /curc/sw/install/openmpi/4.1.1/gcc
    /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
122 ==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)
123 ==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)
124 ==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)
125 ==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)
126 ==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)
127 ==12158==      by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
128 ==12158==      by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
129 ==12158==      by 0x400FF9: orterun (in /curc/sw/install/openmpi/4.1.1/gcc/11.2.0
    _slurmpmi/bin/orterun)
130 ==12158==
131 ==12158== 2,436 (1,656 direct, 780 indirect) bytes in 3 blocks are definitely lost
    in loss record 72 of 73
132 ==12158==      at 0x4C3BE4B: calloc (vg_replace_malloc.c:1328)
133 ==12158==      by 0x533C45C: pmix_server_log (in /curc/sw/install/openmpi/4.1.1/gcc
    /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
134 ==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)
135 ==12158==      by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/
    openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)

```

```

136 ==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)
137 ==12158==      by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
138 ==12158==      by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
139 ==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)
140 ==12158==      by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
141 ==12158==      by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
142 ==12158==
143 ==12158== 3,090 (1,656 direct, 1,434 indirect) bytes in 3 blocks are definitely
    lost in loss record 73 of 73
144 ==12158==      at 0x4C3BE4B: calloc (vg_replace_malloc.c:1328)
145 ==12158==      by 0x533C0D8: pmix_server_log (in /curc/sw/install/openmpi/4.1.1/gcc
    /11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
146 ==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)
147 ==12158==      by 0x53255B1: pmix_server_message_handler (in /curc/sw/install/
    openmpi/4.1.1/gcc/11.2.0_slurmpmi/lib/libopen-pal.so.40.30.1)
148 ==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)
149 ==12158==      by 0x821D9B4: ??? (in /usr/lib64/libevent_core-2.1.so.6.0.2)
150 ==12158==      by 0x821E3B6: event_base_loop (in /usr/lib64/libevent_core-2.1.so
    .6.0.2)
151 ==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)
152 ==12158==      by 0x64EB1CE: start_thread (in /usr/lib64/libpthread-2.28.so)
153 ==12158==      by 0x8673DD2: clone (in /usr/lib64/libc-2.28.so)
154 ==12158==
155 ==12158== LEAK SUMMARY:
156 ==12158==      definitely lost: 7,872 bytes in 14 blocks
157 ==12158==      indirectly lost: 3,770 bytes in 28 blocks
158 ==12158==      possibly lost: 0 bytes in 0 blocks
159 ==12158==      still reachable: 8,754 bytes in 86 blocks
160 ==12158==      suppressed: 0 bytes in 0 blocks
161 ==12158== Reachable blocks (those to which a pointer was found) are not shown.
162 ==12158== To see them, rerun with: --leak-check=full --show-leak-kinds=all
163 ==12158==
164 ==12158== For lists of detected and suppressed errors, rerun with: -s
165 ==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.