


DMSWARM

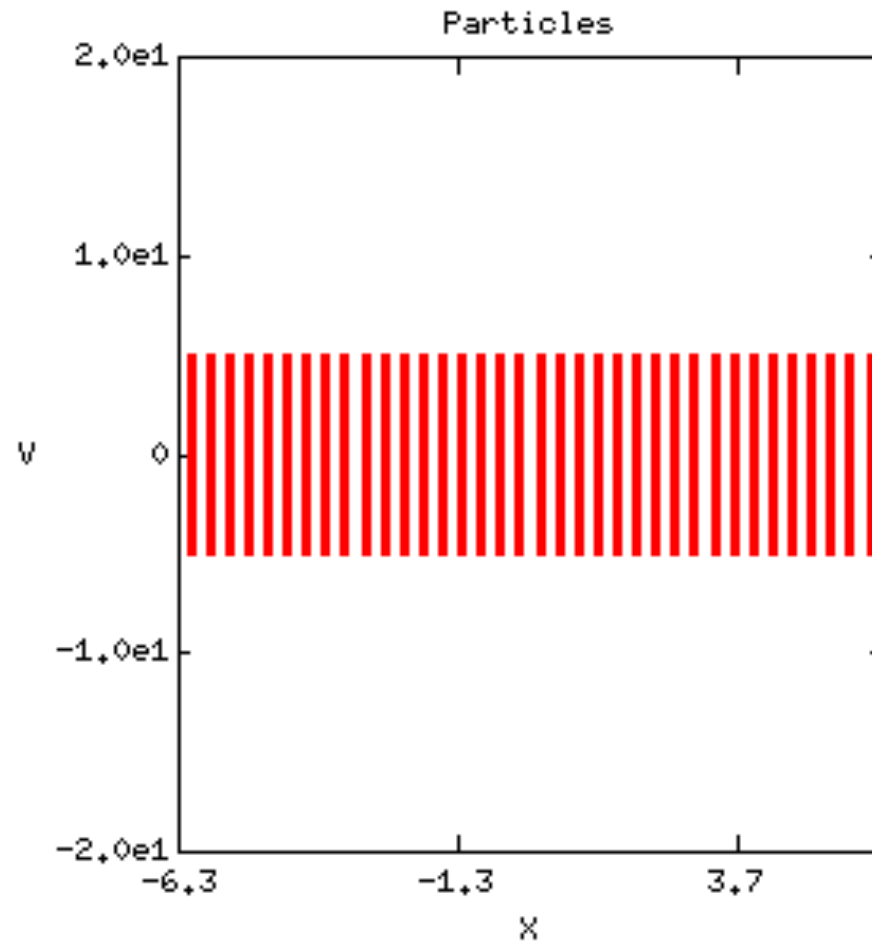
PETSc 2023 Minitutorial

Joseph Puzstay

 University at Buffalo
Department of Computer Science
and Engineering
School of Engineering and Applied Sciences

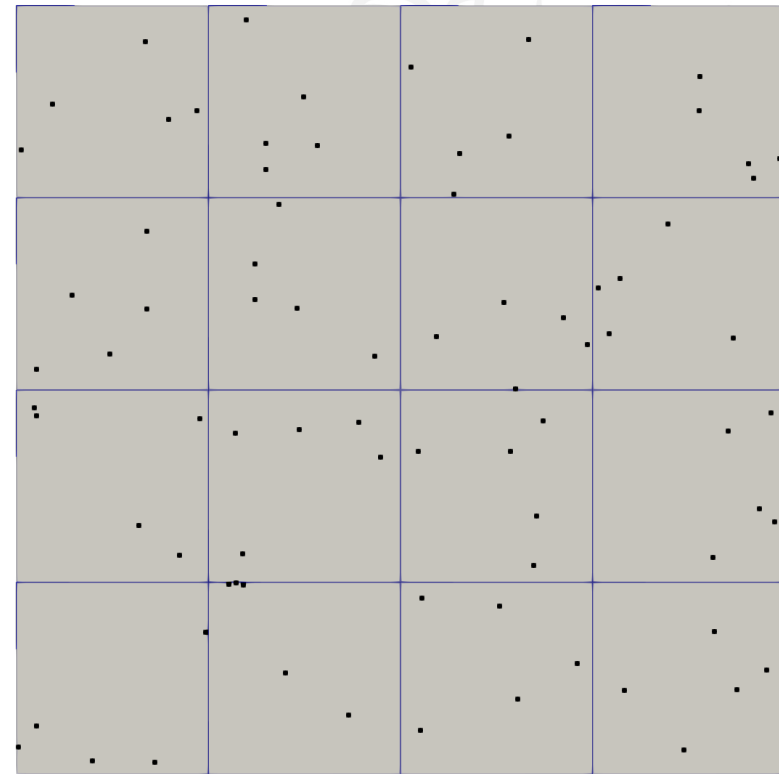


Parallel Particle Management



Parallel Particle Management

- Data structures and algorithms for particle management
- Particle Discretizations
- PIC discretizations
- Distributed particle communication
 - Migration among MPI ranks
- Structured/Unstructured grids (DMDA/DMPlex)
- Point Cell Location via Plex and DA bounding box
- Works alongside TS for particle push



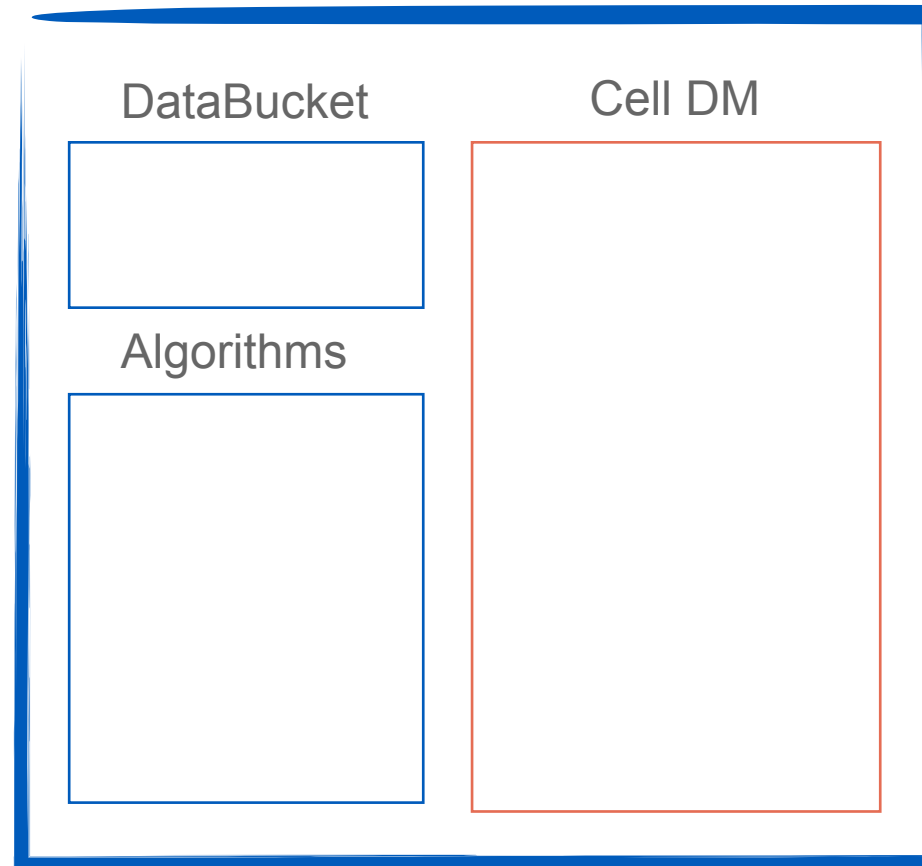
High Level Internals

- DataBucket (core)
 - Basic fields -> DMSwarmPICField_coor, DMSwarmPICField_cellid, ..rank
 - User defined fields -> velocity, weight, pressure etc.
- DMSwarmMigrate
 - DataEx (communication data packing)
 - DMLocatPoints
- Cell DM (DMSWARM_PIC)



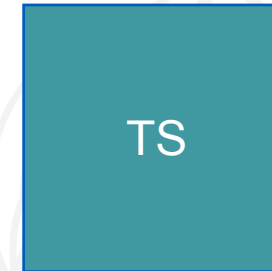
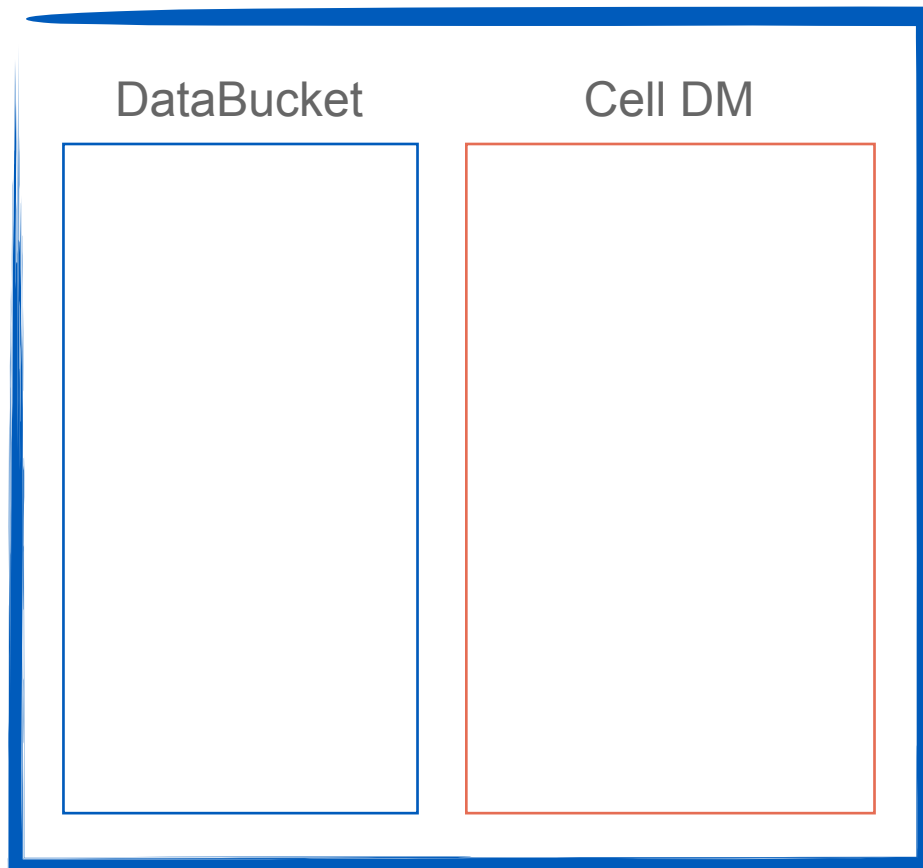
High Level Internals

DMSwarm



Interactions

DMSwarm



- Explicit/Implicit Time steppers can leverage the Swarm for information on the problem
 - VectorDefineField
 - Define Initial conditions vector for the TS from Swarm

Configuring a Swarm

```
DMCreate(PetscObjectComm((PetscObject)dm), sw);  
DMSetType(*sw, DMSWARM);  
DMSetDimension(*sw, dim);  
DMSwarmSetType(*sw, DMSWARM_PIC);  
DMSwarmSetCellDM(*sw, dm);
```

Defining user fields for particles

```
DMSwarmRegisterPetscDatatypeField(*sw, "w_q", 1, PETSC_SCALAR);  
DMSwarmRegisterPetscDatatypeField(*sw, "velocity", dim, PETSC_REAL);  
DMSwarmRegisterPetscDatatypeField(*sw, "species", 1, PETSC_INT);  
DMSwarmFinalizeFieldRegister(*sw);
```


Configuring Data

```
DMSwarmGetField(*sw, DMSwarmPICField_coor, NULL, NULL, (void **)&coords);  
for (c = 0; c < Ncell; ++c) {  
    DMPlexComputeCellGeometryFVM(dm, c, NULL, centroid, NULL);  
    cellid[c] = c;  
    for (d = 0; d < dim; ++d) coords[c * dim + d] = centroid[d];  
}  
DMSwarmRestoreField(*sw, DMSwarmPICField_coor, NULL, NULL, (void **)&coords);
```

Allocating Particles option 1

```
DMSwarmComputeLocalSizeFromOptions(*sw);
DMSwarmInitializeCoordinates(*sw);
DMSwarmInitializeVelocitiesFromOptions(*sw, v0);
DMSetFromOptions(*sw);
-dm_swarm_num_particles 375 -dm_swarm_coordinate_density {{constant gaussian}}
// -dm_swarm_num_species 2 -dm_swarm_velocity_function constant
```

Allocating Particles option 2

```
DMSwarmFinalizeFieldRegister(*sw);  
DMPlexGetHeightStratum(dm, 0, NULL, &Ncell);  
DMSwarmSetLocalSizes(*sw, Ncell * Np, 0);  
DMSetFromOptions(*sw);
```

Working with Swarm

```
// Interact via Vec Object
DMSwarmCreateGlobalVectorFromField(sw, "w_q", &f);
DMSwarmDestroyGlobalVectorFromField(sw, "w_q", &f);
// Communicate particles to other ranks
DMSwarmMigrate(*sw, PETSC_FALSE);
// protected contexts for swarm information.
DMSwarmSortGetAccess(sw);
DMSwarmSortGetPointsPerCell(sw, cell, &Np, &pidx);
// Do something with the particle indices, ie. loop through cells
// then loop through the particles in each cell.
DMSwarmSortRestoreAccess(sw);
```

Examples/Tests

- `src/dm/impls/swarm/tests/ex1.c` -> Particle Migration
- `src/dm/impls/swarm/tests/ex2.c` -> Particle/Field Projection
- `src/dm/impls/swarm/tests/ex3.c` -> SubSwarm
- `src/dm/impls/swarm/tests/ex4.c` -> Harmonic Oscillator w/ TS
- `src/dm/impls/swarm/tests/ex5.c` -> Vlasov/Central Orbit
- `src/dm/impls/swarm/tests/ex6.c` -> Vlasov-Poisson Central Orbit
- `src/dm/impls/swarm/tests/ex7.c` -> Projection w/ openMP in cylindrical domain
- `src/dm/impls/swarm/tests/ex8.c` -> Initialization w/ KS test (Requires `—download-ks`)
- `src/dm/impls/swarm/tests/ex9.c` -> Linear Landau Damping (being merged into TS tutorial `hamiltonian/ex2.c`)
- `src/ts/tutorials/hamiltonian/ex2.c` -> Two-Stream instability electrostatics problem w/ Symplectic Integrators
 - Undergoing some updating

Tutorials

- `src/dm/tutorials/swarm_ex1.c` -> 4 process particle collection tests
- `src/dm/tutorials/swarm_ex2.c` -> Checks variable block sizes
- `src/dm/tutorials/swarm_ex3.c` -> uses with DMShell and user defined operations

