Flexible Computational Science Infrastructure (FleCSI)

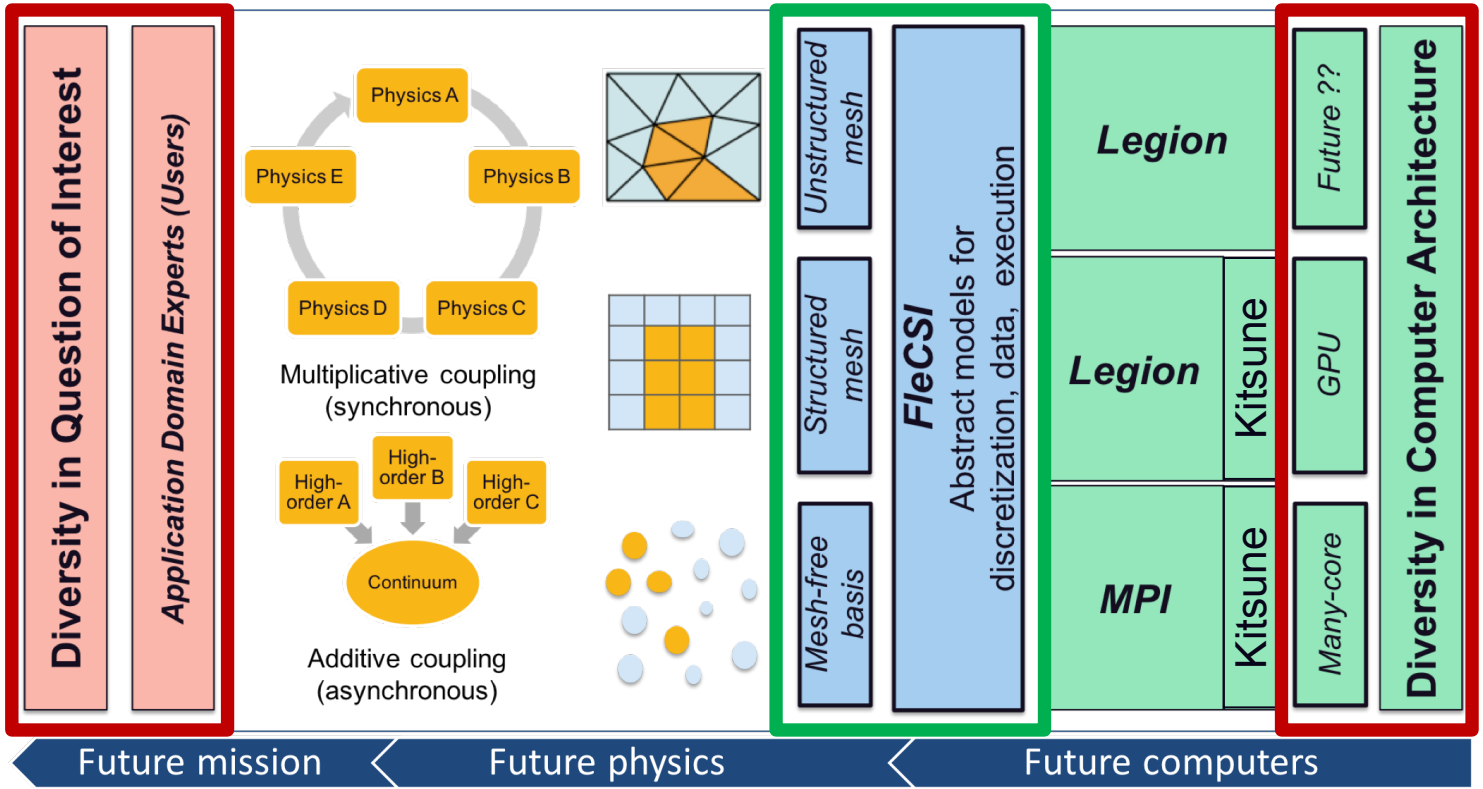
17th Annual Workshop on Charm++ and Its Applications

Li-Ta Lo for the FleCSI Team

May 1, 2019

NNSA National Nuclear Security Administration
Managed by Triad National Security for the U.S. Department of Energy's NNSA

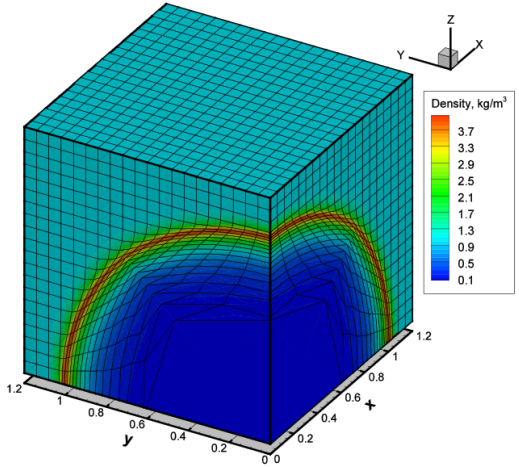
Ristra Big Picture
Advanced Technology Development & Mitigation (ATDM)



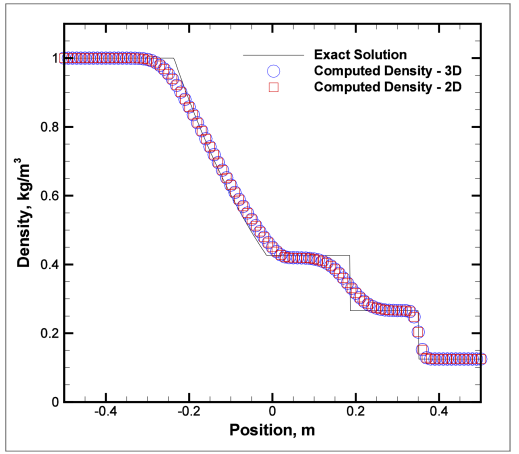
What is FleCSI?

FleCSI is a C++ programming system for developing multi-physics simulation codes

- **Runtime abstraction layer**
  - High-level user interface, mid-level static specialization, low-level building blocks, tasking and fine-grained threading back-ends
- **Programming model**
  - Data, execution, and control models
- **Useful data structure support**
  - Mesh, N-Tree (N=3 → Octree), and Set topologies



FleCSI: Pure 3D Lagrangian Sedov

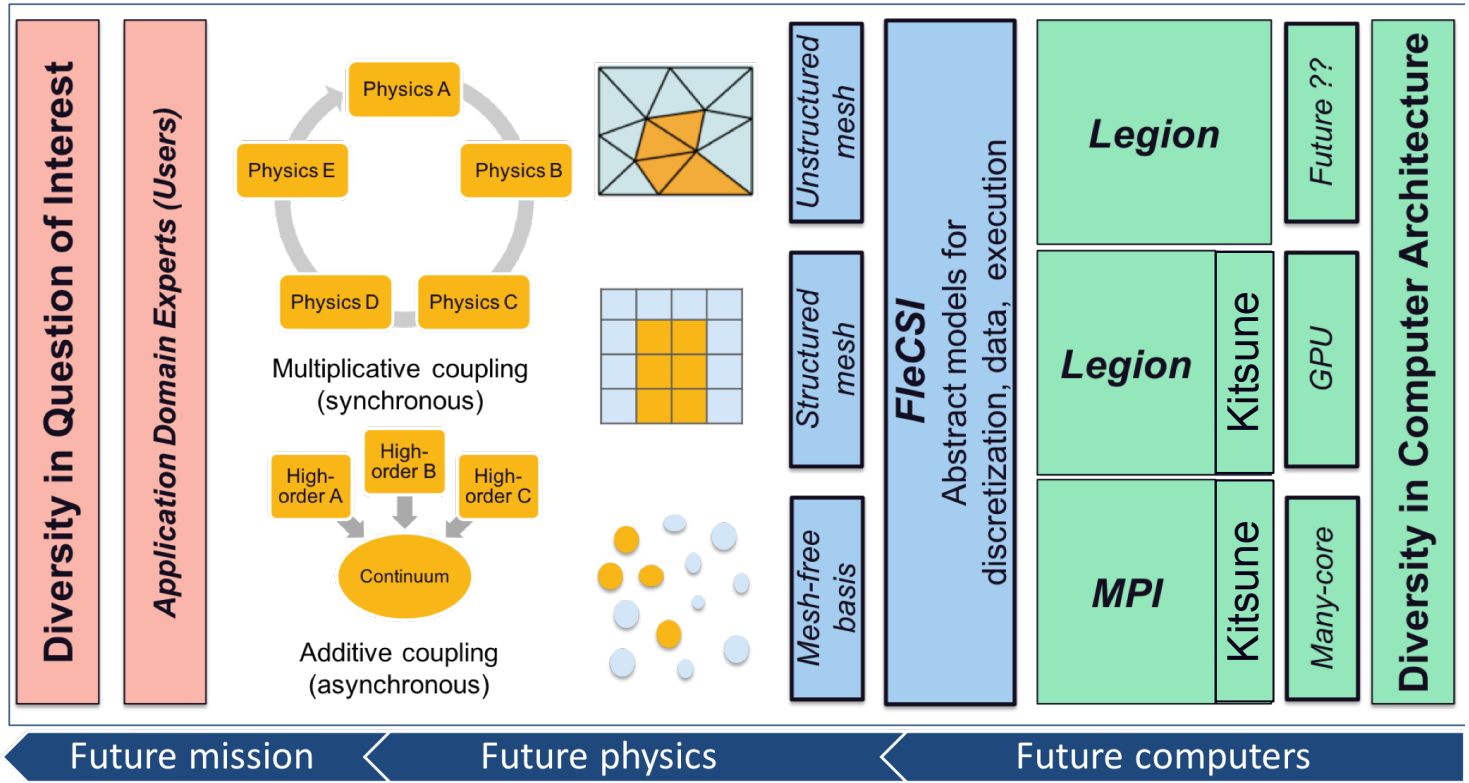


FleCSI: 2D/3D Eulerian Sod

The FleCSI programming structure is designed to encourage separation of concerns…

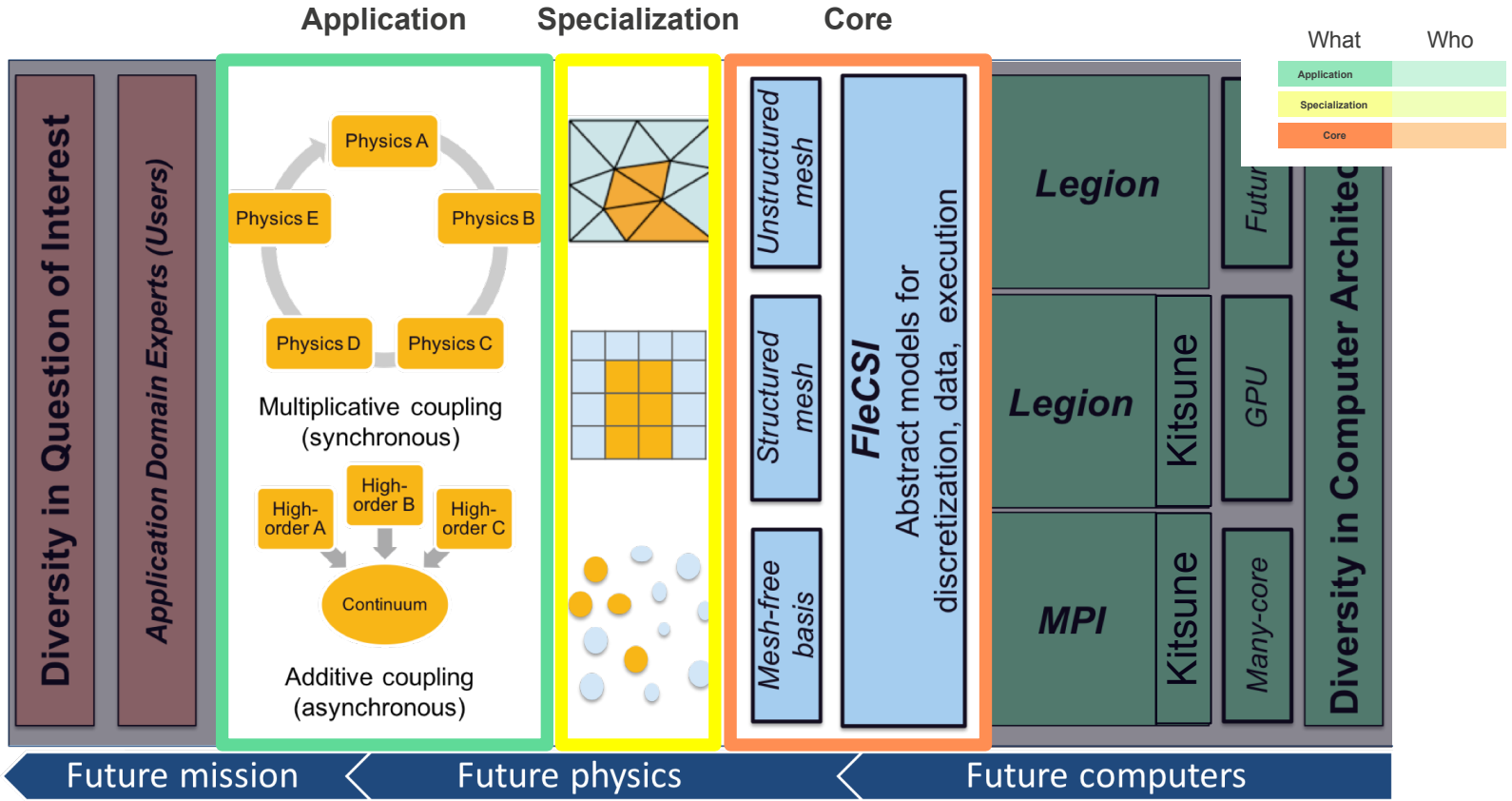
What	Who
Application Development	Physicists & Applied Mathematicians
FleCSI Specialization Development	Computational Scientists
FleCSI Core	Computer Scientists

Ristra Big Picture
Advanced Technology Development & Mitigation (ATDM)



Ristra Big Picture

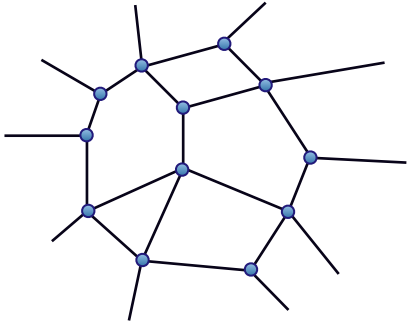
Advanced Technology Development & Mitigation (ATDM)



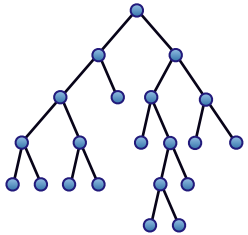
Data Model

FleCSI Topology Data Structures

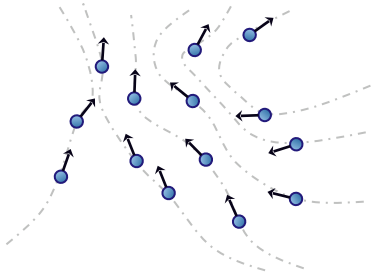
- **flecsi::topology::mesh_topology__**
  - Support for unstructured meshes with user-defined mesh entity types, and user-defined adjacency storage



- **flecsi::topology::tree_topology__**
  - Support for hashed trees with user-defined node types, and user-defined relational functions, e.g., “who are my neighbors?”

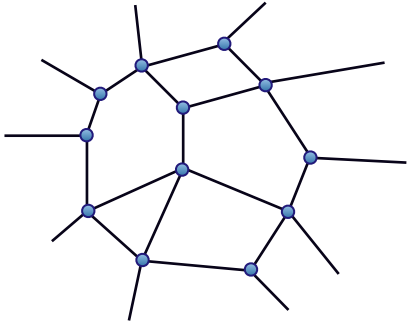


- **flecsi::topology::set_topology__**
  - Support for sets of user-defined entities, e.g., non-interacting particles, and user-defined rules for entity migration, coloring, and binning

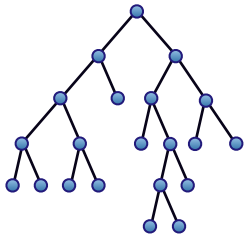


FleCSI Topology Data Structures

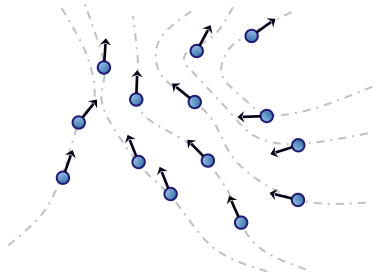
- **flecsi::topology::mesh_topology__**
  - Hydrodynamics (Eulerian, Lagrangian, ALE, Re-ALE, DG), Radiation/Heat Conductivity



- **flecsi::topology::tree_topology__**
  - N-Body, Smoothed-Particle Hydrodynamics



- **flecsi::topology::set_topology__**
  - Particle-in-Cell (PIC), Material-Point Method (MPM), Charged/Neutral Particle Transport



What does Topology do for you?

- **FleCSI automatically generates iterators for each entity type, connectivity, and binding, *or* node**

```
foreach(auto c: mesh.cells()) {
  foreach(auto v: mesh.vertices(c)) {
  } // for
} // for
```

What does Topology do for you?

- **Topological entities define index spaces where data can be attached to the mesh**

```
flecsi_register_data(mesh, hydro, temperature , double, dense, cells);
flecsi_register_data(mesh, hydro, avg_temperature , double, dense, cells);

foreach(auto c: mesh.cells()) {
   foreach(auto v: mesh.vertices(c)) {
       avg_temp(c) += temp(v);
   } // for
   avg_temp(c) /= mesh.vertices(c).size();
} // for
```

Execution Model

What does Execution do for you?

- **Launch task via backends**

```
for ( size_t num_steps = 0; num_steps < max_steps; ++num_steps ) {
   flecsi_execute_task( evaluate_fluxes, apps::hydro, index, mesh,
                                d, v, e, p, T, a, F );
} // for
```

What does Execution do for you?

- **Maintain the illusion of single address space**

```
for ( size_t num_steps = 0; num_steps < max_steps; ++num_steps ) {
    // ghost copy within execute_task, happens before/after calling
    // evaluate_fluxes()
    flecsi_execute_task( evaluate_fluxes, apps::hydro, index, mesh,
                                 d, v, e, p, T, a, F );
} // for
```

FleCSALE Application

Fully unstructured 2D and 3D mesh specializations developed on top of FleCSI

Mesh is templated on dimension:

2D: `burton_mesh_t<2> mesh;`

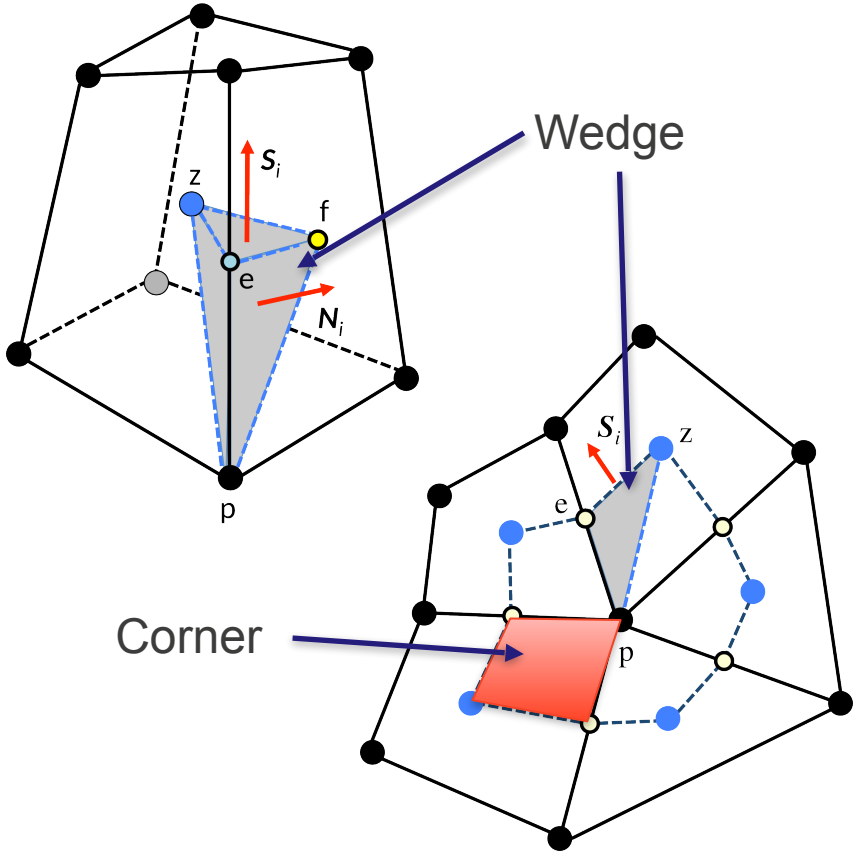
3D: `burton_mesh_t<3> mesh;`

Application code doesn't change (code works in 2D and 3D):

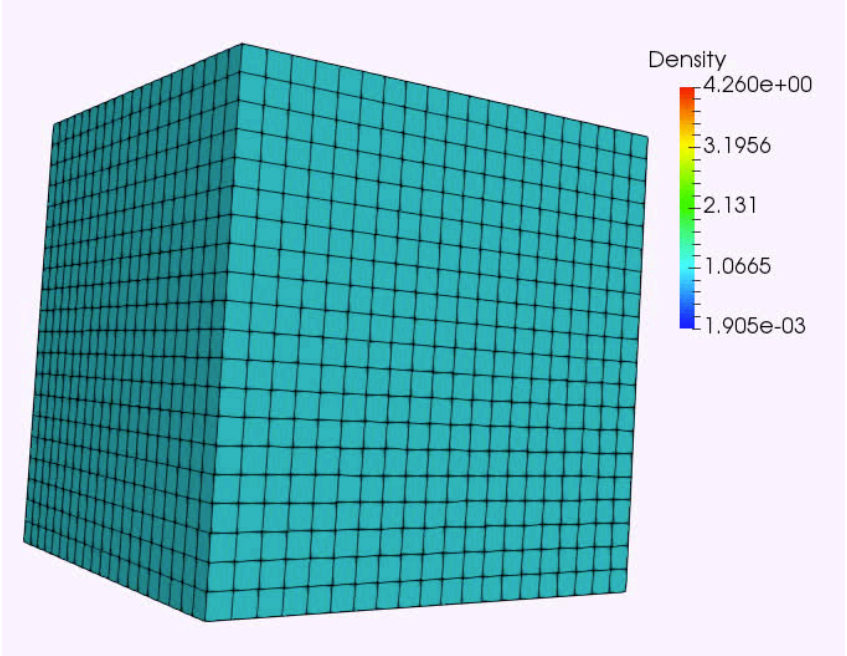
```
for ( auto f : mesh.faces() )
  auto n = f->normal();
  // do some work
```

Mesh has wedges and corner data structures in addition to vertex, edge, face, and cell primitives:

```
for ( auto cn : mesh.corners() )
  for ( auto wg : mesh.wedges(cn) )
    auto n = wg->facet_normal();
    // do some other work
```



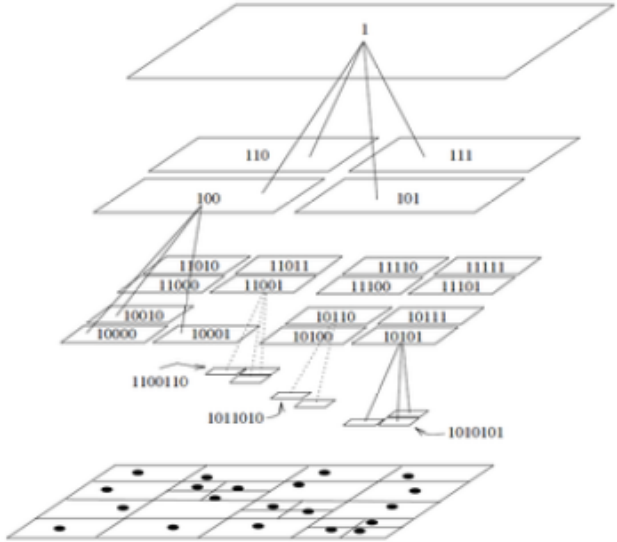
Sedov blast wave predictions computed with the 3D cell-centered Lagrange method

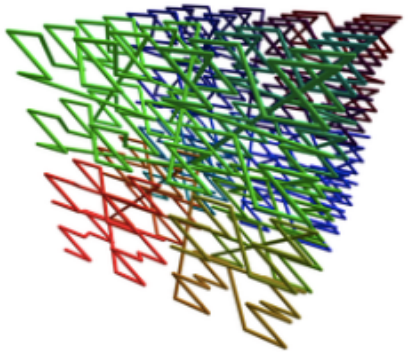


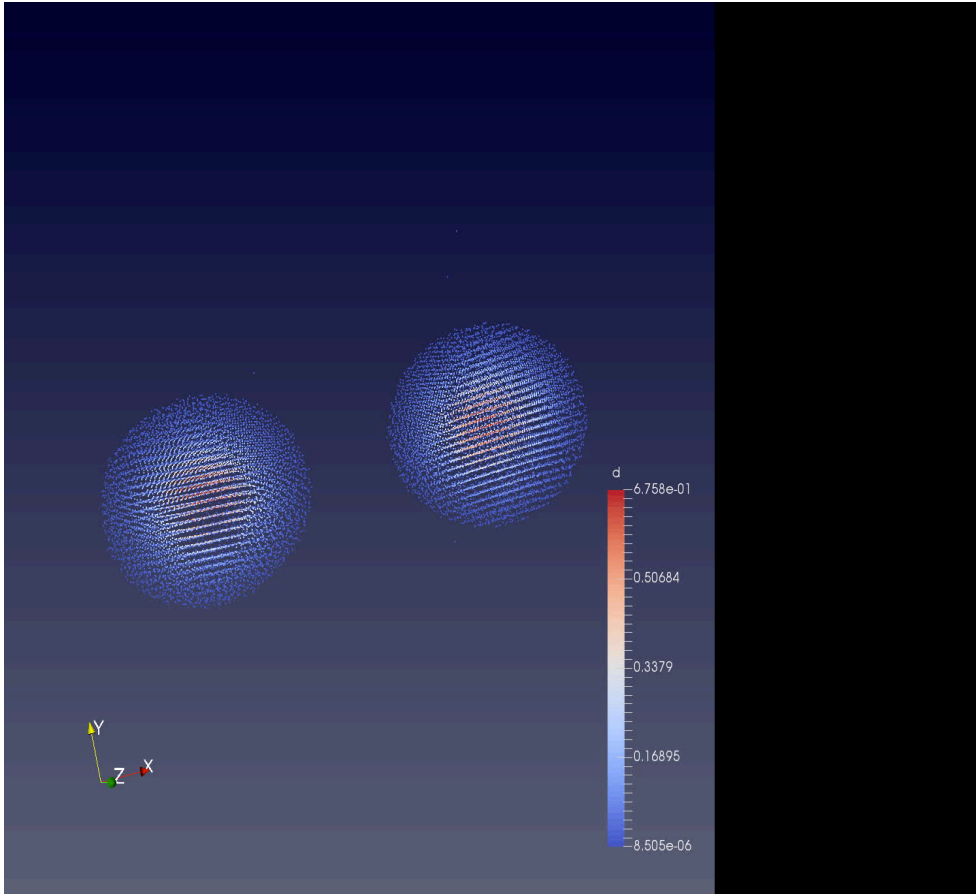
FleCSPH Application

FleCSI : Tree Data Structures

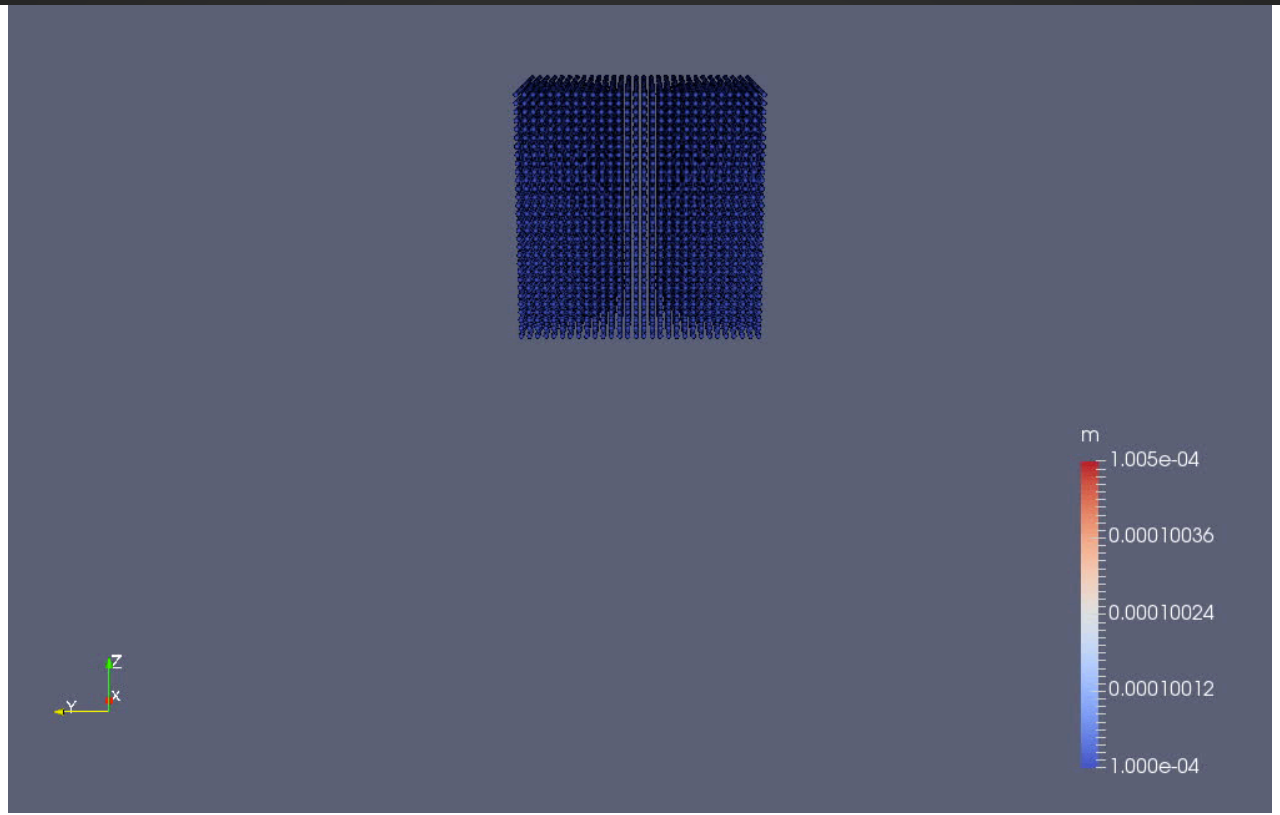
- **Tree topology**
  - Support n-tree (also hashed n-tree)
  - Constant-time neighbor look-up
  - Morton ordering
  - Refinement and coarsening
  - Applications: SPH, N-body, AMR, Complex Flows, Monte Carlo, Molecular Dynamics







Head on Collision of two neutron stars



3D water cube drop

Thanks for your attention!