Asynchronous Distributed-Memory Task-Parallel Algorithm for Compressible Flows on 3D Unstructured Grids

J. Bakosi, M. Charest, **A. Pandare**, J. Waltz
Los Alamos National Laboratory, Los Alamos, NM, USA

October 20, 2020
LA-UR-20-28309

Project goals

- Large-scale Computational Fluid Dynamics (CFD) capability
- Simulation use cases
  - shocked flow over surrogate reentry bodies
  - blast loading on vehicles or other complex structures
  - weapons effects calculations in urban environments
- Distinguishing characteristics
  - external flows over complex 3D geometries
  - high-speed compressible flow
- Capability requirements compared to internal flow calculations
  - complex domain must be explicitly meshed (rather than modeled)
  - multiple orders of magnitude larger computational meshes
  - larger demand for HPC: $\mathcal{O}(10^9)$ cells, $\mathcal{O}(10^4)$ CPUs must be routine calculations

Quinoa::Inciter: Built on Charm++

- Compressible hydro (single or multiple materials)
- Unstructured 3D (tetrahedra only) grids
- Continuous and discontinuous Galerkin finite elements
- Adaptive: mesh refinement (WIP), polynomial-degree refinement
- Native Charm++ code interoperating with MPI libs
- Overdecomposition
- Parallel I/O
- SMP, non-SMP
- Automatic load balancing
- Open source: quinoacomputing.org

Quinoa::Inciter: ALECG hydro scheme, numerical method

- Edge-based finite element (or node-centered finite volume) method
- Compressible single-material (Euler, ideal gas) flow

$$\frac{\partial U}{\partial t} + \frac{\partial F_j}{\partial x_j} = 0, \qquad U = \begin{Bmatrix} \rho \\ \rho u_i \\ \rho E \end{Bmatrix}, \quad F_j = \begin{Bmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ u_j(\rho E + p) \end{Bmatrix}$$

- Galerkin lumped-mass, locally conservative formulation

$$\frac{\mathrm{d}U^v}{\mathrm{d}t} = -\frac{1}{V^v} \sum_j \left[\sum_{vw \in v} D_j^{vw} F_j^{vw} + \sum_{vw \in v} B_j^{vw} \left(F_j^v + F_j^w \right) + B_j^v F_j^v \right]$$

$$U(\vec{x}) = \sum_{v \in \Omega_h} N^v(\vec{x}) U^v, \qquad D_j^{vw} = \frac{1}{2} \sum_{\Omega_h \in vw} \int_{\Omega_h} \left(N^v \frac{\partial N^w}{\partial x_j} - N^w \frac{\partial N^v}{\partial x_j} \right) \mathrm{d}\Omega$$

$$B_j^{vw} = \frac{1}{2} \sum_{\Gamma_h \in vw} \int_{\Gamma_h} N^v N^w n_j \, \mathrm{d}\Gamma, \qquad B_j^v = \sum_{\Gamma_h \in v} \int_{\Gamma_h} N^v N^v n_j \, \mathrm{d}\Gamma$$

Quinoa::Inciter: ALECG hydro scheme, References I

- [1, 2, 3]

J. Waltz, N. Morgan, T.R. Canfield, M.R.J. Charest, L.D. Risinger, and J.G. Wohlbier.
A three-dimensional finite element arbitrary Lagrangian-Eulerian method for shock hydrodynamics on unstructured grids.
Computers & Fluids, 92:172–187, 2014.

J. Waltz, T.R. Canfield, N.R. Morgan, L.D. Risinger, and J.G. Wohlbier.
Verification of a three-dimensional unstructured finite element method using analytic and manufactured solutions.
Computers & Fluids, 81:57 – 67, 2013.

J. Waltz, T.R. Canfield, N.R. Morgan, L.D. Risinger, and J.G. Wohlbier.
Manufactured solutions for the three-dimensional Euler equations with relevance to Inertial Confinement Fusion.
J. Comp. Phys., 267:196 – 209, 2014.

Solution verification: Vortical flow

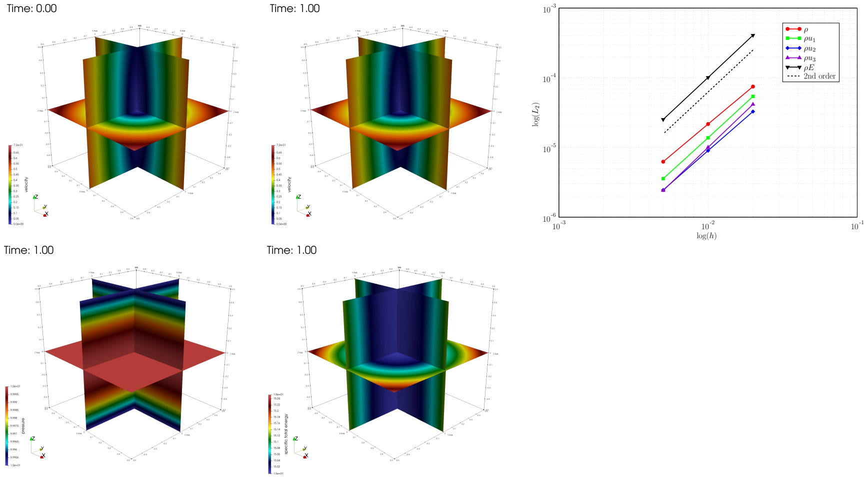
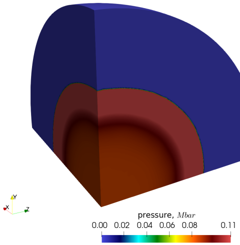


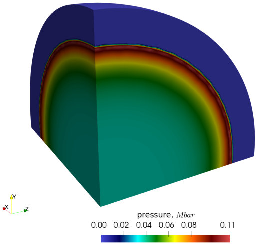
Figure: Left: initial (first column) and final (second column) velocity, pressure (third column), and total energy distributions (fourth column). Right: L_2 errors as a function of mesh resolution.

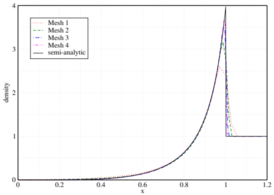
Solution verification: Sedov

Time = 0.49 μs

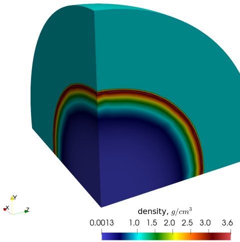


Time = 1.0 μs

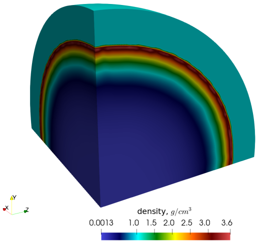


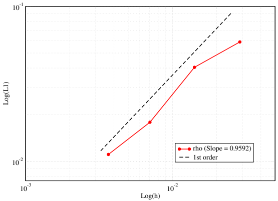


Time = 0.49 μs



Time = 1.0 μs





Solution validation: square cavity, domain and initial conditions

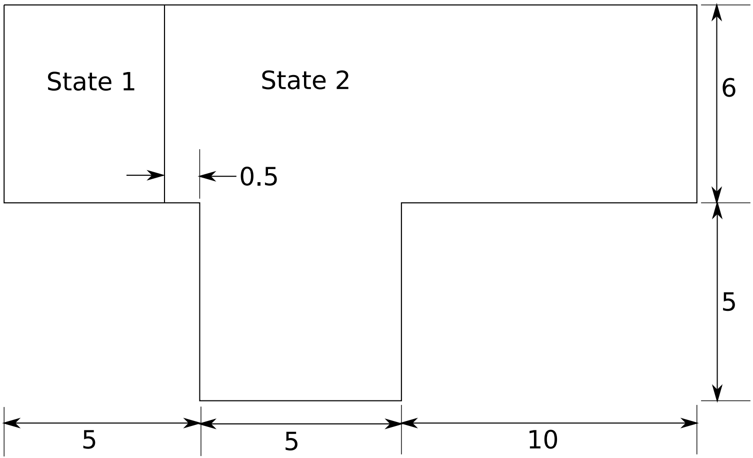
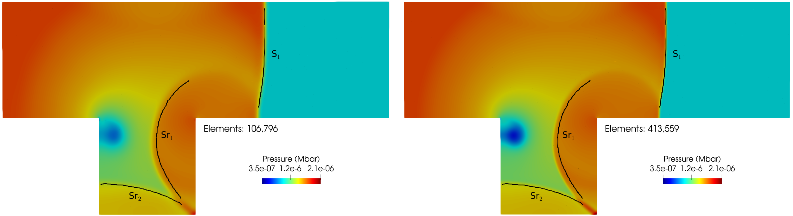


Figure: Domain and initial conditions for square cavity problem. Dimensions are in cm.

Solution validation: square cavity, solution with experimental data



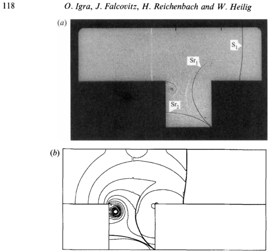


FIGURE 9. As in figure 3 but at $t = 200$ μs.

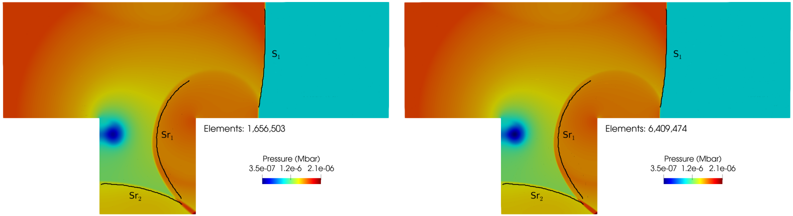


Figure: Solutions with increasingly finer meshes for the square cavity problem. Lines S_1, Sr_1, and Sr_2 denote experimental shock positions.

Solution validation: Onera M6 wing, mesh and numerical solution

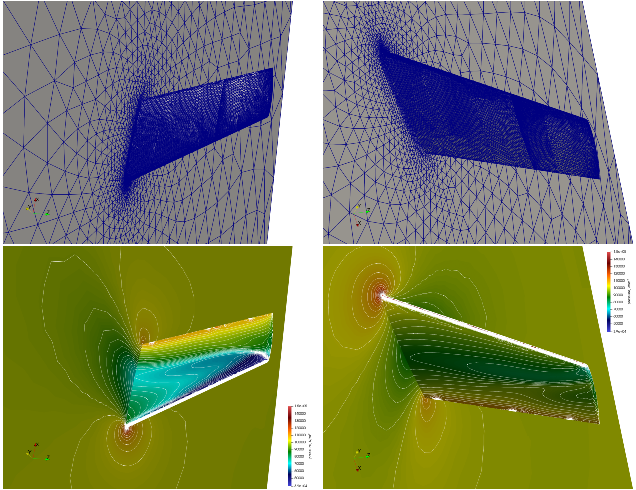


Figure: Top – upper and lower surface mesh used for the ONERA M6 wing configuration. Bottom – computed pressure contours on the upper and lower surface.

Solution validation: Onera M6 wing, simulation & experiments

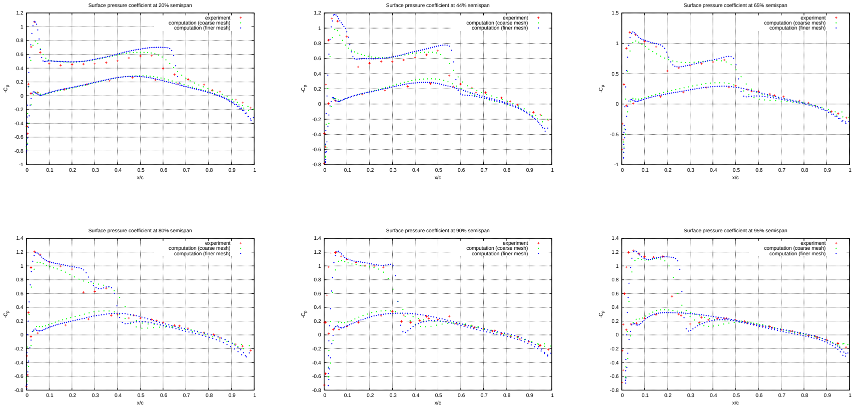


Figure: Comparison beween the computed and experimental surface pressure coefficient for the ONERA wing section at 20%, 44%, 65%, 80%, 90%, and 95% semispans.

Quinoa::Inciter: ALECG, on-node performance

Time step profile:

	μs	%
rhs	8482724	91
bgrad	34333	0.4
diag	48549	0.5
solve	40355	0.4
total	27830000	100

RHS profile:

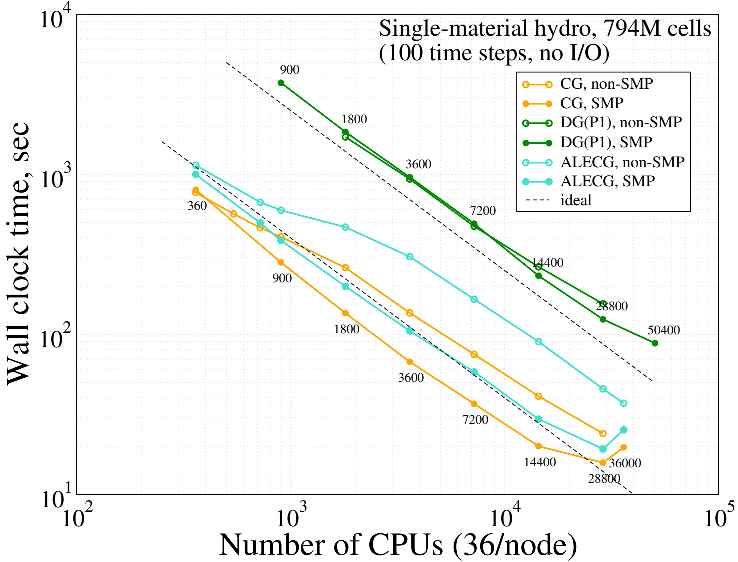
	μs	%
grad	1109746	51
domain	677741	30
bnd	2565	
src	413999	19
total	2183459	100

Quinoa::Inciter: ALECG, on-node performance improvements

1. Remove unnecessary code for generating unused derived data structures: **1.6x**.
2. Replace a tree-based data structure with a flat one, enabling a streaming-style (contiguous) access to normals associated to edges: **1.3x**.
3. Re-write domain-integral from a nested loop (over mesh points and over edges connected to a point) as a single loop over unique edges: **1.3x**.
4. Optimize data access in the source term: **1.4x**.
5. Re-write the loop computing primitive-variable gradients from a gather-scatter loop over elements to a nested loop over mesh points with an inner loop over edges connected to a point: **1.5x**.

Altogether: 6.2x speedup

Quinoa::Inciter: 3 hydro schemes, strong scaling



Quinoa::Inciter: Parallel load imbalance triggered by physics

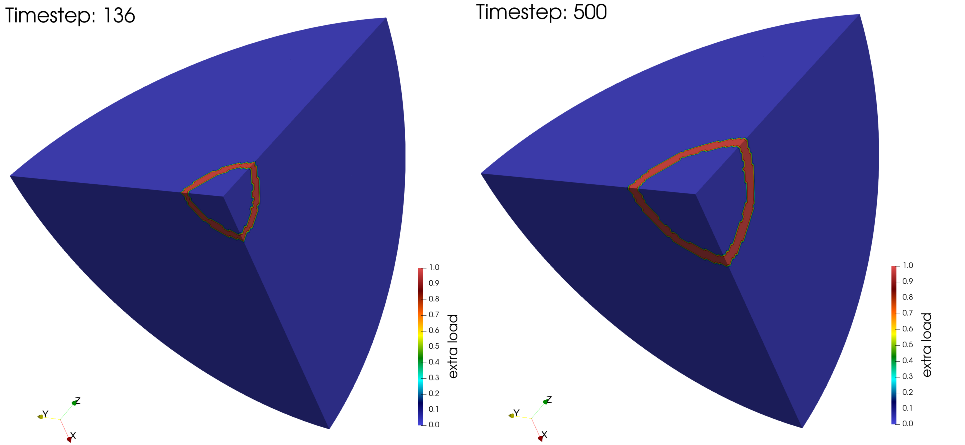


Figure: Spatial distributions of extra load in each cell whose fluid density exceeds the value of 1.5, during time evolution of the Sedov problem: (left) shortly after the onset of load imbalance, (right) at a later time of the simulation.

Quinoa::Inciter: Automatic load balancing yields 10x speedup

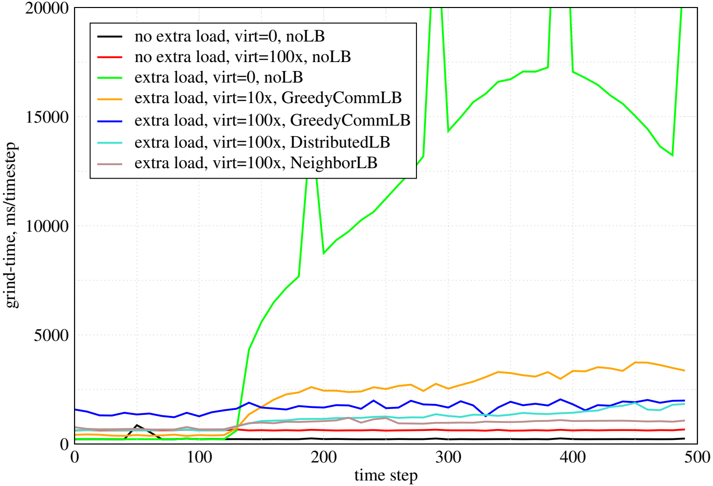


Figure: Grind-time during time stepping computing a Sedov problem with load imbalance, using various built-in load balancers in Charm++. Run on 10 compute nodes with 36CPUs/node.

Current and future work

1. Multi-material FV/DG at large scales
2. P-adaptation
3. Productization (SBIR, PI:Charmworks)
4. 3D mesh-to-mesh solution transfer toward large-scale fluid-structure interaction (see next talk by Eric Mikida)