Recent Developments in the High Order Finite Element Hydrodynamics Code BLAST

ROBERT ANDERSON[1], VESELIN DOBREV[1], TZANIO KOLEV[1] AND ROBERT RIEBEN[2]

[1] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

[2]Weapons and Complex Integration, Lawrence Livermore National Laboratory

Abstract: BLAST is a hydrodynamics research code which implements the high-order finite element formulations of [1,2,3] and is based on the open source finite element software library, MFEM [4]. We consider the ALE extension of a hypo-elastic constitutive model and its use in 2D axisymmetric and full 3D calculations. We also discuss the high performance computing advantages that high-order methods provide for the case of parallel strong scaling for large problems of a fixed size and present our latest work in using GPUs to accelerate the compute intensive low level kernels of the Lagrangian algorithm.

The BLAST ALE Algorithm

BLAST solves the Euler equations using a high-order finite element ALE formulation based on three phases:

- **Lagrangian phase**: solve on moving curvilinear mesh
- **Mesh optimization phase**: harmonic or inverse-harmonic smoothing
- **Remap phase**: conservative and monotonic DG advection based remap

On a semi-discrete level our method can be written as

	Lagrangian Phase	Remap Phase
Mass:	$\rho\lvert\mathbf{J}\rvert = \rho_0\lvert\mathbf{J}_0\rvert$	$\mathbf{M}_\rho \frac{\partial \boldsymbol{\rho}}{\partial\tau} = \mathbf{K}_\rho \boldsymbol{\rho}$
Momentum:	$\mathbf{M}_\mathbf{v}\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\mathbf{F}\cdot\mathbf{1}$	$\mathbf{M}_\mathbf{v}\frac{\partial \mathbf{v}}{\partial\tau} = \mathbf{K}_\mathbf{v}\mathbf{v}$
Energy:	$\mathbf{M}_\mathbf{e}\frac{\mathrm{d}\mathbf{e}}{\mathrm{d}t} = \mathbf{F}^\mathbf{T}\cdot\mathbf{v}$	$\mathbf{M}_\mathbf{e}\frac{\partial \mathbf{e}}{\partial\tau} = \mathbf{K}_\mathbf{e}\mathbf{e}$
Stress Deviator:	$\mathbf{M}_\mathbf{e}\frac{\mathrm{d}\mathbf{s}_{ij}}{\mathrm{d}t} = \mathbf{g}$	$\mathbf{M}_\mathbf{e}\frac{\partial \mathbf{s}_{ij}}{\partial\tau} = \mathbf{K}_\mathbf{e}\mathbf{s}_{ij}$

where $\mathbf{F}$ is the rectangular force matrix, $\boldsymbol{\rho}$ is the density with discontinuous basis ψ, $\mathbf{v}$ is the velocity with continuous vector basis w, and $\mathbf{e}$ and $\mathbf{s}_{ij}$ are the energy and stress deviators, each with discontinuous basis ϕ.
The mass and advection matrices are defined as:

$$(\mathbf{M}_{\boldsymbol{\rho}})_{ij} = \int_\Omega \psi_j\psi_i \qquad (\mathbf{K}_{\boldsymbol{\rho}})_{ij} = \sum_z \int_z u\cdot\nabla\psi_j\psi_i - \sum_f\int_f (u\cdot n)[\![\psi_j]\!](\psi_i)_d$$

$$(\mathbf{M}_\mathbf{v})_{ij} = \int_\Omega \rho w_j w_i \qquad (\mathbf{K}_\mathbf{v})_{ij} = \sum_z \int_z \rho u\cdot\nabla w_j w_i$$

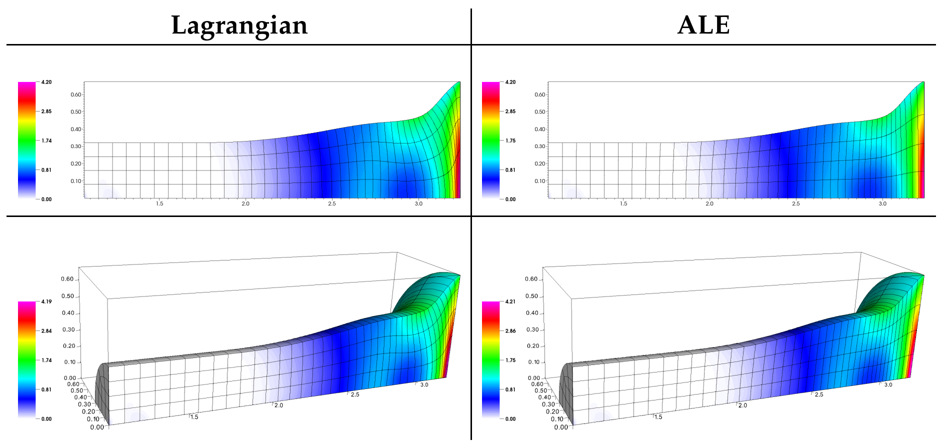
$$(\mathbf{M}_\mathbf{e})_{ij} = \int_\Omega \rho\phi_j\phi_i \qquad (\mathbf{K}_\mathbf{e})_{ij} = \sum_z \int_z \rho u\cdot\nabla\phi_j\phi_i - \sum_f\int_f \rho_u(u\cdot n)[\![\phi_j]\!](\phi_i)_d$$

The axisymmetric extension of the remap involves simple radial scaling, e.g.

$$(\mathbf{M}^{\mathbf{rz}}_{\boldsymbol{\rho}})_{ij} = \int_\Omega r\psi_j\psi_i \qquad (\mathbf{K}^{\mathbf{rz}}_{\boldsymbol{\rho}})_{ij} = \sum_z \int_z ru\cdot\nabla\psi_j\psi_i - \sum_f\int_f r(u\cdot n)[\![\psi_j]\!](\psi_i)_d$$

2Drz and 3D Lagrangian vs. ALE results for Taylor Impact

We consider the Taylor high-velocity impact problem which consists of a cylindrical copper rod impacting a rigid wall. We compare Lagrangian vs. ALE results for a 2Drz calculation on 32 processors and a full 3D calculation on 128 processors. Each calculation uses Q_4-Q_3 finite elements with 25 dof/zone in 2D and 125 dof/zone in 3D.



Method	Length	Max. Rad.	Max. EPS	Cycles	Run Time
2Drz ALE	2.1738	0.6744	4.20	23,890	2.15 min
2Drz Lag.	2.1738	0.6742	4.20	496,347	26.32 min
3D ALE	2.1739	0.6743	4.21	29,348	120.30 min
3D Lag.	2.1739	0.6741	4.19	509,468	869.27 min

The ALE calculations provide a **12X speedup in 2D** and a **7X speedup in 3D**, giving effectively identical answers relative to Lagrangian calculations.

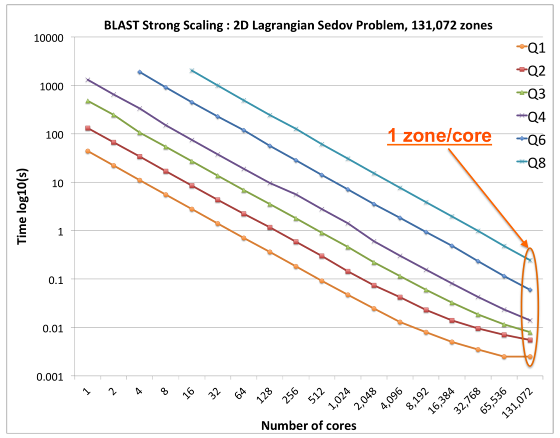
Strong Parallel Scaling on LLNL's Vulcan Supercomputer *

High-order methods excel at strong parallel scaling:

For a fixed mesh resolution, 512 × 256 zones, we vary the number of cores, down to **one zone per core**.

This is a p-refinement study – increasing the order leads to increased resolution in terms of total number of unknowns.

The higher-order methods, $Q4$, $Q6$, and $Q8$, exhibit nearly perfect strong scaling.



* Results courtesy of Michael Kumbera, LLNL.

GPU Acceleration of Force Matrix Calculation *

Force matrix calculation uses local dense linear algebra operations (see [1]):

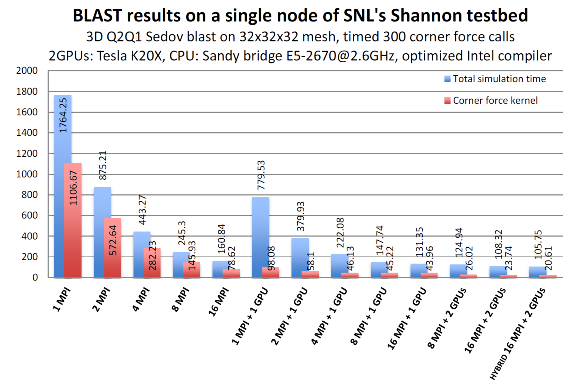
$$\mathbf{F}_z = \mathbf{A}_z\mathbf{B}_z^T, \quad (\mathbf{A}_z)_{ik} = \alpha_k\left[\hat\sigma(\hat q_k)\,\mathrm{adj}\,(\mathbf{J}_z(\hat q_k))^T\right] : \hat\nabla\hat w_i(\hat q_k), \quad (\mathbf{B}_z)_{jk} = \hat\phi_j(\hat q_k),$$

$$\mathbf{J}_z(\hat q_k) = \sum_i \mathbf{x}_{z,i}\hat\nabla\hat w_i(\hat q_k), \quad \hat\rho(\hat q_k) = (\hat\rho_0\lvert\mathbf{J}_{0,z}\rvert)(\hat q_k)/\lvert\mathbf{J}_z(\hat q_k)\rvert, \quad \hat e(\hat q_k) = (\mathbf{B}_z^T\mathbf{e}_z)_k.$$

- Computation is split into six CUDA kernels (reduce register pressure).
- CPU-GPU memory copies used for vectors only ($\mathbf{F}_z$ are stored on GPU).
- Utilize Hyper-Q: multiple MPI tasks simultaneously use the same GPU.

GPU acceleration leads to **4X speedup** in the force matrix calculation.

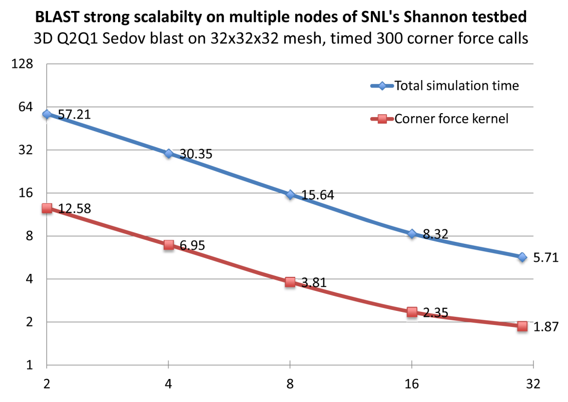
Additional GPU acceleration in other parts of BLAST is possible, leading to: (a) less CPU-GPU memory transfers, (b) better GPU utilization, and (c) better overall speedup.



Strong scaling study up to 30 compute nodes, a total of 480 CPUs and 60 GPUs.

Currently, MPI communications go through the main system (CPU) memory.

Direct GPU-GPU communications are technologically possible (GPUDirect).



* This work is a collaboration with Tingxing Dong, UT Knoxville.

References

[1] V. Dobrev, Tz. Kolev, and R. Rieben, High-order curvilinear finite element methods for Lagrangian hydrodynamics, *SIAM Journal on Scientific Computing*, **34 (5)**, 2012, pp. B606-B641.
[2] V. Dobrev, T. Ellis, Tz. Kolev, and R. Rieben, High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics, *Computers & Fluids*, **83**, 2013, pp. 58-69.
[3] V. Dobrev, Tz. Kolev, and R. Rieben, High order curvilinear finite elements for elastic-plastic Lagrangian dynamics, *Journal of Computational Physics*, 2013, (in press).
[4] MFEM library, `http://mfem.googlecode.com`

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-POST-643018