A GPU-accelerated Boundary Element Method and Vortex Particle Method
Mark J. Stock and Adrin Gharakhani
Applied Scientific Research, Inc., Santa Ana, CA

Technical Challenge

Our goal is to simulate high-Reynolds-number turbulent flows around multiple moving geometries, accounting for the often-complex behavior of the wake. Predicting the flow and forces will help wind generator manufacturers increase efficiency and reliability, and help rotorcraft designers reduce noise and identify dangerous flight regimes.

Vortex Particle Method

All vorticity is discretized onto isotropic, desingularized, Lagrangian vortex blobs of appropriate size. Computational elements are only needed where there is vorticity. Vortex particles connect using their local velocity, which arises as a solution to the integral equation below:

\[ \mathbf{v}_\omega(\mathbf{x}) = \mathbf{u}_\infty + \sum_{\mathbf{x}_i} K_\omega(\mathbf{x}, \mathbf{x}_i) \times \mathbf{f}(\mathbf{x}_i) \]

\[ K_\omega(\mathbf{x}, \mathbf{x}_i) = K(\mathbf{x}, \mathbf{x}_i) \int_0^{1/4\pi} g(r) r^2 dr \]

The system evolves according to the following equations.

\[ \frac{d\mathbf{v}_\omega}{dt} = \mathbf{u}_\infty + \nabla \times \mathbf{f} \]

\[ \frac{d\mathbf{f}}{dt} = \mathbf{v}_\omega - \mathbf{f} \frac{\nabla \cdot \mathbf{f}}{|\mathbf{f}|} \]

VPM - Implementation

The kinematic velocity inversion uses a modified treecode (not FMM) and takes most of the time in the all-CPU version, thus it was the focus of our GPU efforts. MPI, OpenMP, and CUDA are all used to parallelize the treecode.

For each velocity solve, the CPU performs the following tasks:
- Order the centers into a VAMSplit k-D Tree (bucket size is usually 64), multipole moments (usually of order 6-9) are made for each box using an upward pass, and interaction lists are generated from a tree traversal for each leaf box. The GPU then computes the multipole multiplications (far-field) and the direct summations (near-field). Each GPU kernel computes either the near- or far-field influences for all particles in one leaf box in the target tree.

VPM - Performance

All comparisons use the fastest methods and parameters for the hardware, and solve to \( \approx 10^{-6} \) RMS velocity error.

VPM - Performance, cont’d

The implementation’s weak scaling, on Lincoln, using 8-core nodes, is shown below. The left plot uses no GPUs.

Massive Moving Bodies Result in More Complex Flow Fields

The GPU parallel efficiency on Lincoln, MPI+OpenMP, and CUDA are shown below. The left plot uses no GPUs.

No-slip/no-flux boundary conditions are imposed by placing vortex sheets at the wall, with their surface-tangent strengths \( \gamma \) obtained by the solution of a Poisson boundary integral equation of the second kind.

\[ \frac{1}{2} \gamma(\mathbf{x}) \times \mathbf{h}(\mathbf{x}) + \mathbf{n}(\mathbf{x}) \times \nabla \times \mathbf{f}(\mathbf{x}) = -\mathbf{U}(\mathbf{x}) \times \nabla \times \mathbf{f}(\mathbf{x}) \]

The solution of this equation is obtained in the collocation (or near-field) formulation by discretizing the surface into \( N_s \) contiguous triangular panels with piecewise-continuous vortex sheet (vector) strength. The fluid volume needs no grid.

BEM - Implementation

The near-field portion of the resulting dense influence matrix is calculated and stored, and the far-field part is approximated with multiple expansions. One matrix-vector multiply consists of the near-field (direct) part done on all CPUs, and the far-field part computed on all GPUs.

BEM - Performance

We solved for the unknown vortex sheet strengths of a simplified artificial heart leaf geometry at a variety of resolutions. An 4-core Phenom II with and without two NVIDIA 275 GTX GPUs was used for this calculation. The iterations were CPU-limited—the second GPU added little.

VPM+BEM Results

These 3D uniform-resolution viscous results show particle size and discretized domain for runs on one dual-GPU node. The Re=400 run used 328k panels and peaked at 26.5M particles.

Figure: Centerline velocity behind sphere. Left: Grid solution with fixed spatial adaptivity. Right: Hybrid grid-particle solution with fixed resolution and solution adaptivity in the particle wake.

Hybrid Grid-Particle Results

We created a hybrid method that uses grid-based CFD (OVERFLOW 2.1) in the near-body region, and our vortex particle method for the wake and applied it to the problems of a Re=100 sphere and a Re=860k free 4-bladed advancing rotor.

Figure: Top: Centerline velocity behind sphere. Left: Grid solution with fixed spatial adaptivity. Right: Hybrid grid-particle solution with fixed resolution and solution adaptivity in the particle wake.

About ASR

Founded in 1998, Applied Scientific Research is a consulting firm, providing engineering, Information Technology (IT), and advanced computational and algorithm development support for the high performance computing needs of the U.S. government and industrial clients. The above research was sponsored by AMRDEC, NSF, and NCRR.

Research Goals

Question: Can we use a mesh-free solution method to reduce the effort of gridding and eliminate the numerical diffusion?

Question: Will that mesh-free method also benefit from the massive parallelism now afforded by clusters of GPUs?

Question: Can we use those same GPU routines to accelerate the Boundary Element Method component of the mesh-free solution?