High-Order GPU-Based Compressible Fluid Flow Solver for Unstructured Grids

Patrice Castonguay
Aerospace Computing Lab, Stanford University
Supervisor: Antony Jameson

Affiliates Meeting
April 24th, 2012
SD++

• 2D/3D compressible viscous flow solver

• Mixed grids of quadrilaterals and triangles in 2D and hexahedra, prisms and tetrahedra in 3D

• Arbitrary order of accuracy

• Solver can run on multiple CPUs or GPUs (C++/Cuda/MPI)
Talk Overview

• Part 1: Unstructured High-Order Methods
  – Why are they useful?

• Part 2: Flux Reconstruction Method for the Navier-Stokes equations
  – Algorithm details
  – Why it’s a good fit for GPUs

• Part 3: GPU Implementation Details
  – Single-GPU: Efficient use of GPU memory hierarchy
  – Multi-GPU : How to obtain good scalability

• Part 4: Performance analysis and Applications
  – Performance on a single GPU
  – Strong and weak scaling study
  – How GPUs enable previously intractable fluid flow simulations
Unstructured High-Order Methods

• What does high-order mean?

• **Low-order** methods:
  – Order of accuracy is 1 or 2 (Error is of order $h$ or order $h^2$)
  – Robust and simple to implement
  – Dissipative

• **High-order** methods:
  – Order of accuracy is $> 2$
  – Not as mature as low-order methods
  – More work per DOF
  – Required for applications where accuracy requirement is high
Unstructured High-Order Methods

- Why do we need high-order methods?

![Graph showing error vs. cost for high-order and low-order methods.](image)
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)

$\text{t} = 1$

$\text{t} = 1$
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)  

4th order (25,600 DOFs)
Unstructured High-Order Methods

- Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)
Unstructured High-Order Methods

- Why is high-order useful?

2nd order (25,600 DOFs)  

4th order (25,600 DOFs)
Unstructured High-Order Methods

• Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)
Unstructured High-Order Methods

- Why is high-order useful?

2nd order (25,600 DOFs)

4th order (25,600 DOFs)

\[ t = 180 \]
Unstructured High-Order Methods

• Why are they useful:
  
  Complex geometry + High Accuracy

• In computational fluid dynamics, they enable:
  
  – Simulation of wave propagation over long distances in vicinity of complex geometries
  
  – Simulation of vortex motion over long distances in vicinity of complex geometries
  
  – Effective Large Eddy Simulations (LES) in vicinity of complex geometries
Unstructured High-Order Methods

- Airframe noise (turbulence + generation/propagation of sound waves + complex geometry)
Unstructured High-Order Methods

- Rotorcraft (turbulence + track vortices over long distances + complex geometry)
• Flapping wing flight (transitional Reynolds number + vortex dominated flow + complex geometry)
Unstructured High-Order Methods

- Computations are **demanding**:  
  - Millions of DOFS  
  - Hundreds of thousands of time steps

- Until recently, high-order simulations over complex 3D geometries were **intractable**, unless you had access to large clusters

- GPUs to the rescue!
Flux Reconstruction Method

• For a conservation law in strong form

\[
\frac{\partial Q}{\partial t} + \nabla \cdot \mathbf{F}(Q, \nabla Q) = 0
\]

• Ex: Euler equations

\[
Q = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
E
\end{bmatrix}, \quad F_x = \begin{bmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
\rho ve + p
\end{bmatrix}, \quad F_y = \begin{bmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
\rho ve + p
\end{bmatrix}
\]

• Solve differential form within each element, with boundary data from neighbouring elements

• Can recover Spectral Difference and Discontinuous Galerkin methods
Flux Reconstruction Method

- Solution in each element approximated by a multi-dimensional polynomial of order $N$

- Order of accuracy: $h^{N+1}$

- Multiple DOFs per element
Flux Reconstruction Method

• Method maps well to the GPUs:
  
  – High-level of parallelism (millions of DOFs)
  
  – More work per DOF compared to low-order methods (flops are “free” on a GPU)
  
  – Cell-local operations benefit from fast on-chip shared-memory
GPU Implementation

• Test case: Viscous flow over sphere, Re=100, Mach = 0.2

• 4th order RK time-stepping scheme

• Considered 3 grid types, each made up of one of the 3 element types

• Every effort was made to maximize performance of CPU code:
  – Intel Math Kernel Library (MKL) version 10.3 for dense MM
  – Optimized Sparse Kernel Interface (OSKI) for sparse MM
  – Cuthill-McKee renumbering of cells to maximize cache-hits

• All simulations use double precision math
GPU Implementation

Speedup of the single-GPU algorithm (C2050) relative to a parallel computation on a quad-core Intel i7 930 @ 2.80GHz
Multi-GPU Implementation

Speedup relative to 1 GPU versus the number of GPUs for a 6th order accurate simulation running on a mesh with 55947 tetrahedral elements
Multi-GPU Implementation

Weak Scalability of multi-GPU code: 27915 ± 1% Tets per GPU
Applications

- At Reynolds number in range $10^4$ to $10^5$, flow over wings often characterized by formation of a Laminar Separation Bubble.
- Important: birds and small UAVs fly in that regime.
- Complex flow physics:

![Diagram showing flow physics](image-url)
Applications

- Transitional flow over SD7003 airfoil, Re=60000, Mach=0.2, AOA=4°
- 4\textsuperscript{th} order accurate solution, 400000 RK iterations, 21.2 million DOFs
Applications
Applications

15 hours on 16 C2070s

202 hours (> one week)
on 16 Xeon x5670 CPUs
Conclusions

- Developed fast high-order CFD solver that can run on mixed unstructured grids on multiple GPUs
- GPUs enable simulation of previously intractable problems
- More than 100 Gigaflops on a workstation, few Teraflops on small GPU cluster
- Scaling demonstrated on up to 32 GPUs
- Next steps: LES models, more complex geometries
Acknowledgments

• Acknowledgments:
  – Peter Vincent, David Williams, Kui Ou, Yves Allaneau
  – NSF (Grants 0708071 and 0915006)
  – AFOSR (Grants FA9550-07-1-0195 and FA9550-10-1-0418)
  – Stanford Graduate Fellowship (SGF) Program
  – National Sciences and Engineering Council (NSERC) of Canada
  – Fonds Quebecois de la Recherche sur la Nature et les Technologies (FQRNT)
  – NSF Graduate Research Fellowship Program
  – NVIDIA

• Questions?

Patrice Castonguay  pcasto2@stanford.edu