GPU Accelerated Numerical Methods For Tsunami Modeling

R Gandham, D S Medina, and T Warburton
Department of Computational and Applied Mathematics, Rice University

Context

Goal: To develop portable high performance computational tools and numerical models to predict tsunami impacts in a timely manner.

Challenge: Tsunami waves travel at very high speeds after the occurrence of earthquakes in the ocean bed and reach coastal areas within few hours.

Numerical methods: Discontinuous Galerkin methods are high order accurate, flexible in handling unstructured coastal aligned meshes, and highly parallelizable on many core architectures making them attractive for simulating tsunami wave propagation.

Programming model: OCCA unified threading model provides portability across threading models OpenCL, CUDA, and OpenMP.

Current Target Model: Shallow Water Equations

\[
\begin{align*}
\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} &= 0, \\
\frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2 + \frac{1}{2}gh^2)}{\partial x} &= -\frac{\partial (ph)}{\partial x}, \\
\frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2 + \frac{1}{2}gh^2)}{\partial x} &= -\frac{\partial (ph)}{\partial x}.
\end{align*}
\]

$h$: fluid height, $u$: longitudinal velocity, $v$: lateral velocity, $B$: Bathymetry

Numerical Simulation

Region Selection

Coastal Aligned Triangulation

Simulation on GPUs

Specifications:
- GMSh for coastal aligned mesh generation.
- GSSHG data sets for coastal information.
- GEBCO for Bathymetry distribution.
- Nodal discontinuous Galerkin discretizations [Hesthaven et al., 2008].
- Well balanced Lax-Friedrich fluxes [Xing et al., 2010].
- Positivity preserving limiter [Xing et al., 2010].
- Multirate Adams-Bashforth local time-stepping [Gear et al., 1984].
- Total variational bounded (TVB) slope limiter [Gandham et al., 2014].
- Computation kernels are written in OCCA [Medina et al., 2014].
- MPI for computation on multiple GPUs.

Discontinuous Galerkin

General Conservation Law

\[ \frac{\partial \phi}{\partial t} + \nabla \cdot F = S \]

DG Variational Statement

\[ \frac{\partial \phi}{\partial t} = \left( F, \nabla \phi_{\sigma} + \nabla (S_{\sigma})_{\sigma} \right) \quad \forall \phi \in \mathcal{P}^i(D) \]

Update kernel
Volume kernel
Surface kernel

Discretization: Warp & Blend Nodal Elements

OCCA: Portable Multi-Threading Programming

OCCA Host API
- Abstracts different language APIs.
- Focuses on abstracting the device, memory and kernels.

OCCA Device API
- Relies on macros for masking the different supported languages.
- Uses the GPU programming model of work-groups and work-items.

OCCA Class
- Choose between using the GPU or available accelerators.
- OpenCL-mode pick from supported platforms and devices.
- CUDA-mode pick NVIDIA CUDA-enabled GPUs.

OCCA Memory
- Encapsulates function handles found in each language.
- Uses run-time compilation.

OCCA Kernel
- Abstracts the memory handles found in each language.

OCCA Based Simulation Performance

GPU (NVidia Titan)

CPU (Intel i7-2930K)

OCCA-CL
OCCA-OpenMP
OCCA-CUDA

Polynomial Order

Surface Kernel
Volume Kernel
Polynomial Order

OCCA: Portable Multi-Threading Programming

Supported in part by DOE, ANL, ORNL, Shell, Pan American Advanced Studies Institute, NVIDIA, and AMD