Efficient Particle-Based Simulation of Dynamic Cracks and Fractures in Ceramic Material (S4255)

Patrick Diehl*, Marc Alexander Schweitzer

Institute for Numerical Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

March 2014
Motivation

Crack patterns

Wave propagation

Edge-on-Impact - ALON - v=380m/s^-1

Kic=2.0e6 Pa sqrt(m), K=210e9 Pa, delta=3a, a=0.0005m
Outline

1. Peridynamics
   – Schematic of Peridynamics theory

2. Neighbor Search
   – Sketch of our in house CUDA library

3. Evaluation of our code
   – Experiment and simulation setup

4. Results
   – Numerical results and benchmarks

5. Conclusion and Outlook
Peridynamics

– Schematic of Peridynamics theory
What is it?
- A non local theory in continuum mechanic
- Admits discontinues solutions
- High computational load

Peridynamic equation of motion

$$\rho(x) \ddot{u}(x, t) = \int_{\mathcal{H}_i} f(\xi, \eta) dV_{x'} + b(x, t)$$

More details in – A meshfree method based on the peridynamic model of solid mechanics [3]
Semi-discrete Peridynamic equation of motion

\[ \rho(x_i) V_i \ddot{u}_i = \sum_{j \in F_i} f(\xi, \eta) \tilde{V}_j V_i + b(x, t) \]

with \( F_i = \{ j \mid \| x_j(0) - x_i(0) \| \leq \delta, j \neq i \} \)

Usage of thrust data structures:

- thrust::device_vector
- thrust::host_vector

Neighbor Search:

→ High computational load

- No library with fits to our tasks available!
- Implement an efficient neighbor search on GPU.
Neighbor Search

– Sketch of our in house CUDA library
Algorithm

Data: Morton order compare operator $<, >$ and Point cloud $P$
Result: $k$-nearest neighbors $\forall p_i \in P$

\begin{align*}
P &\leftarrow \text{ParallelSort}(P, <); \\
\text{for } p_i \in P \text{ do} & \quad A_i \leftarrow \text{nn}_k(p_i), \{P_{i-k}, \ldots, P_{i+k}\}; \\
& \quad \text{if } p_i^{[\text{rad}(A_i)]} < p_{i+k} \text{ then} \\
& \quad \quad \quad u \leftarrow i \\
& \quad \text{else} \\
& \quad \quad \quad I \leftarrow 1; \text{while } p_i^{[\text{rad}(A_i)]} < p_{i+2^I} \text{ do} \quad ++I; \quad u \leftarrow \min(i + 2^I, n); \\
& \quad \text{end} \\
& \quad \text{if } p_i^{[-\text{rad}(A_i)]} > p_{i-k} \text{ then} \\
& \quad \quad \quad l \leftarrow i; \\
& \quad \text{else} \\
& \quad \quad \quad I \leftarrow 1; \text{while } p_i^{[-\text{rad}(A_i)]} > p_{i-2^I} \text{ do} \quad ++I; \quad l \leftarrow \min(i - 2^I, n); \\
& \quad \text{end} \\
& \quad \text{if } l \neq u \text{ then} \\
& \quad \quad \quad \text{CSEARCH}(p_i, l, u); \\
& \quad \text{end} \\
& \text{end}
\end{align*}

Morton Order

Morton order of $P_1$ and $P_2$

$P_1 = (0, 0)_{10} = (000, 000)_2$  
$P_2 = (1, 0)_{10} = (001, 000)_2$

Reduce $d = 2$ to $d = 1$:

$P_1 = 000000$  
$P_2 = 000001$

X-Oring the reduced points:

$000000 \oplus 000001 = 000001$

Most significant bit delivers the Morton order $\rightarrow 1$
Morton order for floating points

**Data:** d-dimensional points p and q

```plaintext
COMPARE(point p, point q)

x = 0; dim = 0;

for j=0 to d do
    y = XORmsb(p_j, q_j);
    if x < y then
        x = y; dim = j;
    end
end

return p_dim < q_dim [4]
```

Conclusion: Algorithm works with high dimensional points ⇒
Interesting for Uncertainty Quantification (UQ)
Sorting with the Morton order compare operator

ParallelSort($P, <$) with thrust::sort
Remarks:
- Computation time between single and double precision differs marginal.
- STANN 0.74 [https://sites.google.com/a/compgeom.com/stann/Home]
Evaluation of our code

– Experiment and simulation setup
Edge On Impact (EOI) Experiment

Fig.: Experimental setup

Fig.: Blueprint of the experiment

More details in – Visualization of Impact Damage in Ceramics Using the Edge-On Impact Technique [1].
Results of the experiment

More details in – HIGH-SPEED PHOTOGRAPHIC STUDY OF WAVE PROPAGATION AND IMPACT DAMAGE IN FUSED SILICA and ALON USING THE EDGE-ON IMPACT (EOI) METHOD [2]
Results

– Numerical results and benchmarks
Results for Aluminium oxynitride (ALON)

Fig.: Shadow graph at $t=8.7\mu s$

Fig.: Velocity at $t=8.7\mu s$

Material costs: US$ 20,000/m$^2$
Results for Fused silica

Fig.: Shadow graph at $t=9.7\mu s$

Fig.: Velocity at $t=9.7\mu s$
Impact on disk [Intel Xeon X7460 2.66GHz vs. NVIDIA GeForce GTX 560 Ti]

Measurements for the CPU with LAMMPS (22 Mar 2013) [http://lammps.sandia.gov].
Conclusion and Outlook
Conclusion and Outlook

Summary

- Efficient simulation of dynamic crack and fractures
- Comparison of the simulation with experiments

Outlook

- Integrate the GPU accelerated neighbor search.
- Use HPC Nvidea GPU instead of consumer cards

Thank you for your attention.
Conclusion and Outlook

Summary

- Efficient simulation of dynamic crack and fractures
- Comparison of the simulation with experiments

Outlook

- Integrate the GPU accelerated neighbor search.
- Use HPC Nvida GPU instead of consumer cards

Thank you for your attention.

@diehlpk #GTC14

[2] E. Strassburger and et al., HIGH-SPEED PHOTOGRAPHIC STUDY OF WAVE PROPAGATION AND IMPACT DAMAGE IN FUSED SILICA and ALON USING THE EDGE-ON IMPACT (EOI) METHOD.
