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

(S4255)

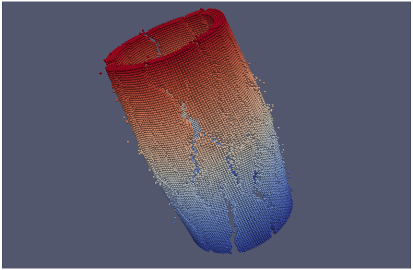
Patrick Diehl*, Marc Alexander Schweitzer

Institute for Numerical Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

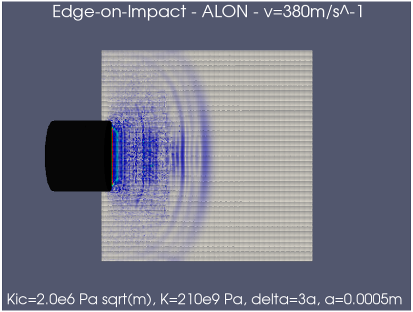
March 2014

Motivation

Crack patterns



Wave propagation



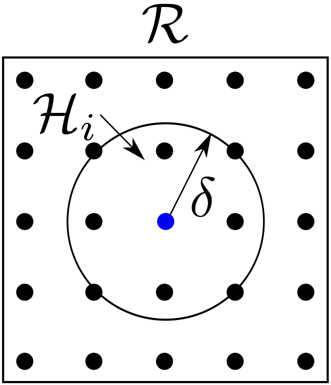
Outline

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

Schematic



What is it?

- A non local theory in continuum mechanic
- Admits discontinues solutions
- High computational load

Peridynamic equation of motion

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

More details in – A meshfree method based on the peridynamic model of solid mechanics [3]

Discretization

Semi-discrete Peridynamic equation of motion

$$\varrho(x_i)V_i\ddot{u}_i^t = \sum_{j\in\mathcal{F}_i} f(\xi,\eta)\tilde{V}_jV_i + b(x,t)$$

with $\mathcal{F}_i = \{j \mid \|x_j(0) - x_i(0)\| \leq \delta,\ j \neq i\}$

Usage of thrust data structures:

- `thrust::device_vector`
- `thrust::host_vector`

Neighbor Search:

$\rightarrow$ High computational load

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

Neighbor Search

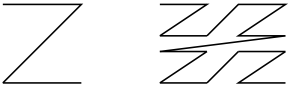
– Sketch of our in house CUDA library

Algorithm

Data: Morton order compare operator $<, >$ and Point cloud P
Result: k-nearest neighbors $\forall p_i \in P$
$P \leftarrow$ ParallelSort$(P, <)$;
for $p_i \in P$ **do**
  $A_i \leftarrow nn^k(p_i), \{P_{i-k}, \ldots, P_{i+k}\}$;
  if $p_i^{\lceil rad(A_i) \rceil} < p_{i+k}$ **then**
    $u \leftarrow i$
  else
    $I \leftarrow 1$; **while** $p_i^{\lceil rad(A_i) \rceil} < p_{i+2^i}$ **do** $++I$; $u \leftarrow \min(i + 2^I, n)$;
  end
  if $p_i^{-\lceil rad(A_i) \rceil} > p_{i-k}$ **then**
    $l \leftarrow i$;
  else
    $I \leftarrow 1$; **while** $p_i^{-\lceil rad(A_i) \rceil} > p_{i-2^i}$ **do** $++I$; $l \leftarrow \min(i - 2^I, n)$;
  end
  if $l \neq u$ **then**
    CSEARCH(p_i, l, u);
  end
end

More details in – Fast construction of k-Nearest Neighbor Graphs for Point Clouds [4]

Morton Order



y \ x	0 000	1 001	2 010	3 011
0 000	000000	000001	000100	001010
1 001	000010	000011	000110	000111
2 010	001000	001001	001100	001101
3 011	001010	001011	001110	001111

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:

$$\begin{array}{rr} & 000000 \\ \oplus & 000001 \\ \hline & 000001 \end{array}$$

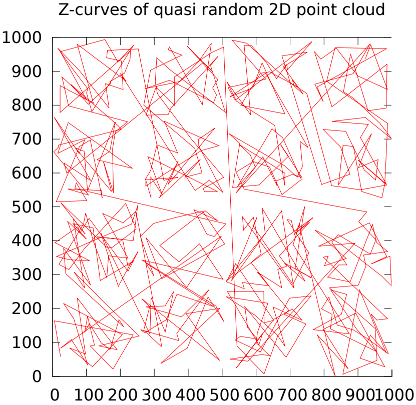
Most significant bit delivers the Morton order $\to$ 1

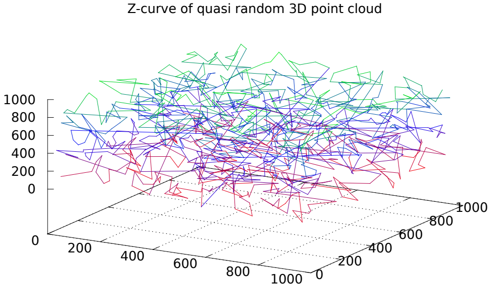
Morton order for floating points

Data: d-dimensional points p and q
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 $\Rightarrow$ Interesting for Uncertainty Quantification (UQ)

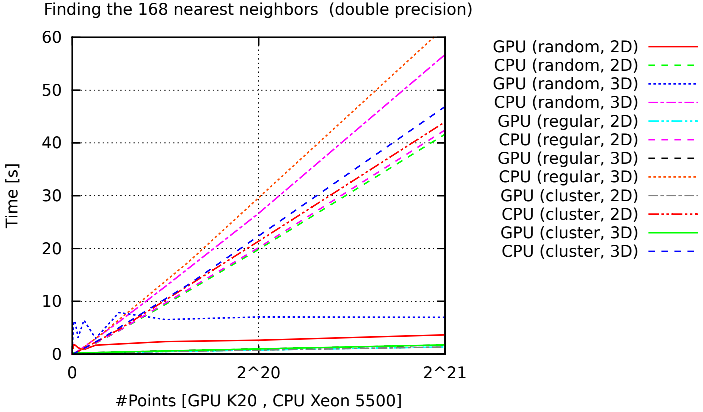
Sorting with the Morton order compare operator





$\text{ParallelSort}(P, <)$ with `thrust::sort`

Performance - Overview



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

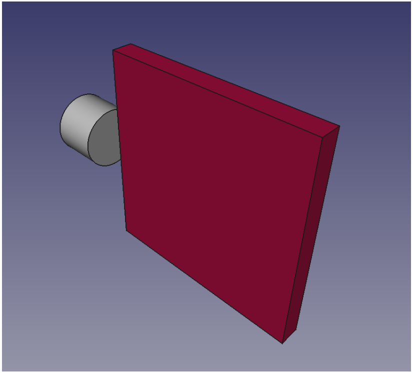


Fig.: Experimental setup

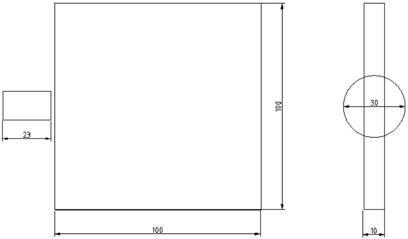


Fig.: Blueprint of the experiment

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

Results of the experiment

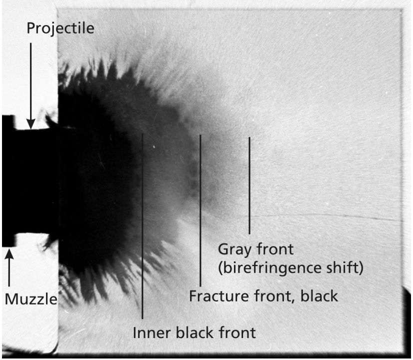


Fig.: Different fronts

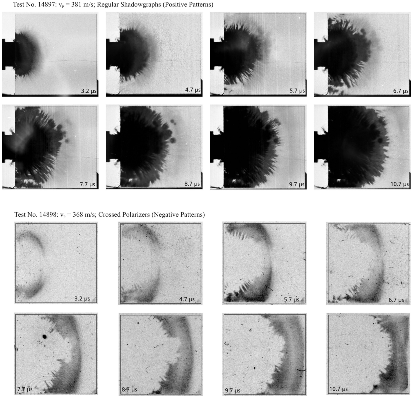


Fig.: Shadow graphs for ALON

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)

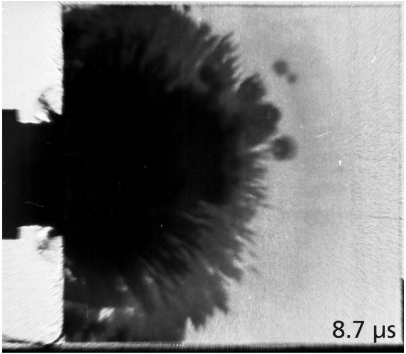


Fig.: Shadow graph at t=8.7μs

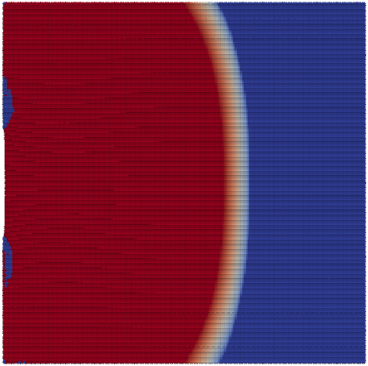


Fig.: Velocity at t=8.7μs

Material costs: US$ 20,000/m^2

Results for Fused silica

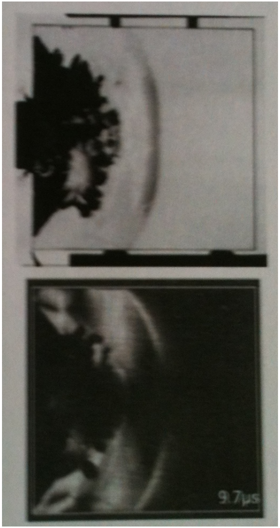


Fig.: Shadow graph at t=9.7μs

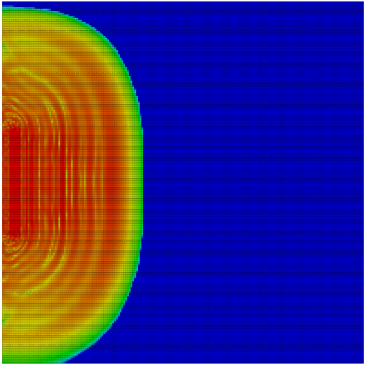
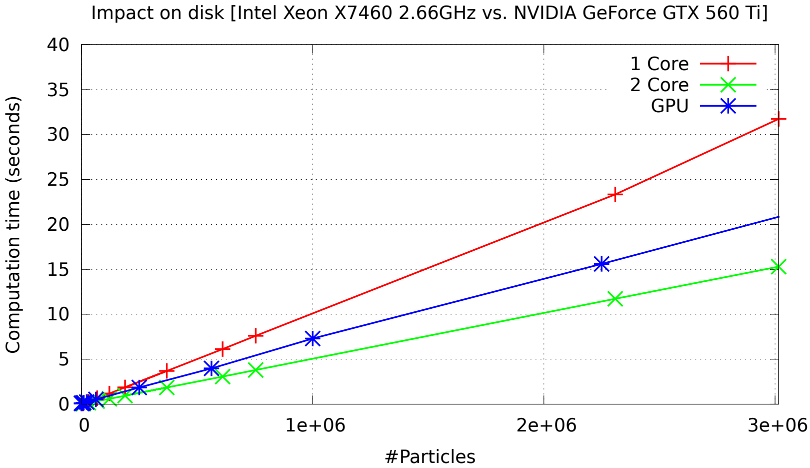


Fig.: Velocity at t=9.7μs

Benchmark



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

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.

@diehlpk #GTC14

Literature

[1] E. Strassburger, Visualization of Impact Damage in Ceramics Using the Edge-On Impact Technique, International Journal of Applied Ceramic Technology, 1:235-242, 2004.

[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.

[3] S. A. Silling and E. Askari, A meshfree method based on the peridynamic model of solid mechanics, Computer & Structures, 83:1526-1535, 2005.

[4] Connor, M. & Kumar, P., Fast construction of k-Nearest Neighbor Graphs for Point Clouds, IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, 2009.