7
Experimental Results

To test the performance and the correctness of our parallel code, we have run a set of computational experiments. The experiments were split in two parts: inserting cohesive elements decoupled from mechanics analysis and running the fracture and fragmentation simulation. The GPU simulation results are compared to CPU counterparts running on a Intel Core i7 CPU @ 2.80GHz with 12GB of RAM on a 64-bit Windows 7 operating system. The GPU used device is a NVIDIA GeForce GTX 480 with 15 multiprocessors, each with 32 cores and a total of 480 CUDA cores, with a clock rate of 1.40 GHz and using compute capability 2.0 since we use double precision in the simulation. Figure 7.1 shows the finite element models used in the test cases.

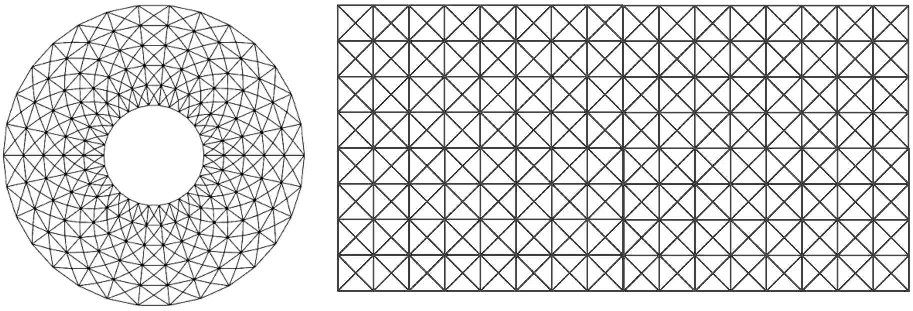

Figure 7.1: T6 disc mesh used to test insertion of cohesive element decoupled from analysis code (left). T6 bar mesh used to test the fracture and fragmentation simulation (right).

7.1
Insertion of cohesive elements

To check the correctness of the algorithm to insert cohesive elements in parallel, we have run a computational test decoupled from any mechanics simulation (setting up experiments similar to the ones described by Pandolfi and Ortiz (19) and by Paulino et al. (21)). Cohesive elements were inserted, in a random order, at all the facets of the underlying meshes. The random order in which the cohesive elements are inserted results in arbitrarily complex crack pattern during the experiment. In the end, each node of the mesh is used by only one bulk element. We then have checked if the final obtained number of topological entities were the expected ones. In the experiments ran by Pandolfi and Ortiz (19) and Paulino et al. (21), the cohesive elements were inserted in a serial order. In our experiment, the cohesive elements are inserted in parallel. To better mimic insertion of cohesive elements in actual simulations, the facets were grouped in 20 sets, inserting 5% of cohesive elements concurrently within each group of facets, using the color model. To color the mesh, we used the Welsh Powell algorithm (26). The number of color achieved was 10.

We have employed a T6 disk mesh 7.1 with different discretizations, varying the number of bulk elements from 240,000 to 3,840,000. The results are shown in Table 7.1. Figure 7.2 depicts that the time to insert all cohesive elements varies linearly with the total number of inserted elements. 7.3 shows the speedup of the GPU implementation when compared to the CPU counterpart. As can be noted, the gain in performance delivered by the GPU implementation is quite significant, even though we are more interested in validating the GPU results.

Bulk elements	Initial nodes	Final nodes	Cohes. elem.	CPU Time (s)	GPU Time (s)	Speedup
240,000	481,200	1,440,000	359,400	9.29	0.0407	228.3
960,000	1,922,400	5,760,000	1,438,800	36.946	0.1016	363.6
2,160,000	4,323,600	12,960,000	3,238,200	84.94	0.1935	439.0
3,840,000	7,684,800	23,040,000	5,757,600	150.04	0.3101	483.8

Table 7.1: Results for insertion of cohesive elements decoupled from analysis code.

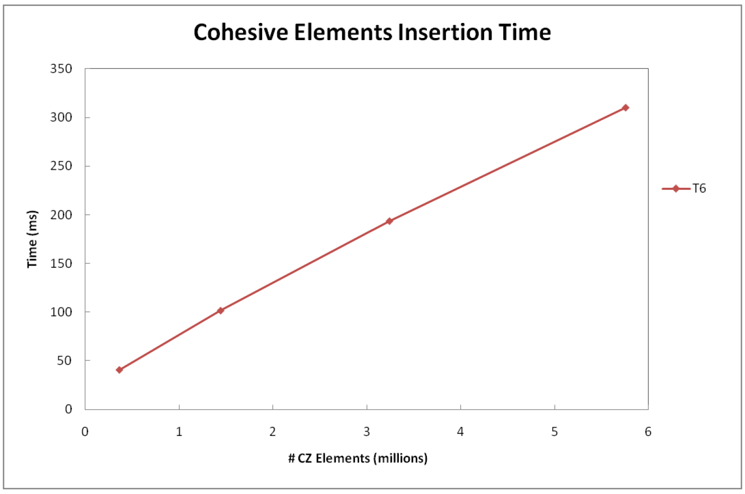

Figure 7.2: Time for cohesive elements insertion of a T6 mesh.

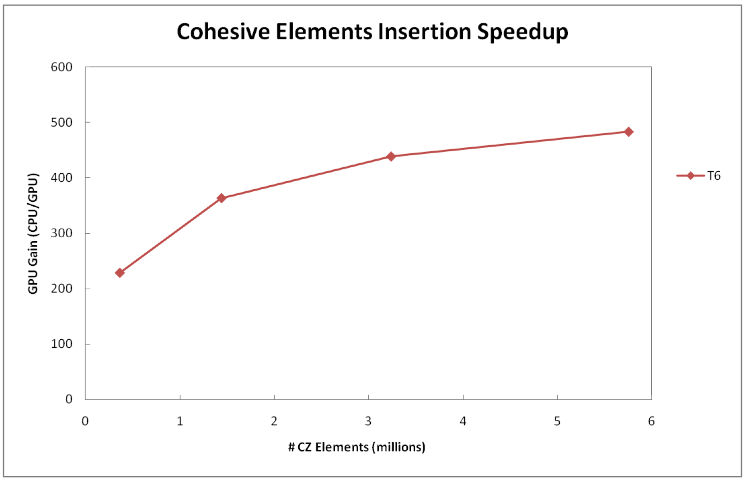

Figure 7.3: Speedup for cohesive elements insertion of a T6 mesh.

When duplicating thousands to millions of nodes concurrently, as in this experiment, atomic operations can be quite slow. In order to optimize node duplications in such scenarios, we present a new algorithm that greatly speed up the kernel. The algorithm is discussed in Appendix A. However, during actual fragmentation simulations, few nodes are duplicated concurrently in a timestep, making the new strategys performance gain negligible.

7.2 Fragmentation simulation

Fragmentation simulation was tested on a 2D model representing a rectangular specimen with an initial notch, as illustrated in Figure 7.4. The model is discretized into T6 (quadratic triangle) elements (Figure 7.1). Fracture propagation is based on mixed-mode fracture and extrinsic cohesive zone model (20, 18, 10). Initial analysis parameters are as follows: initial strain = 0.015, elastic modulus = 3.24 GPa, Poisson coefficient = 0.35, specific mass = 1190 kg/m3, fracture energy (GI) = 352 N/m, cohesive strength (smax) = 324 MPa, and shape parameters (a) = 2.

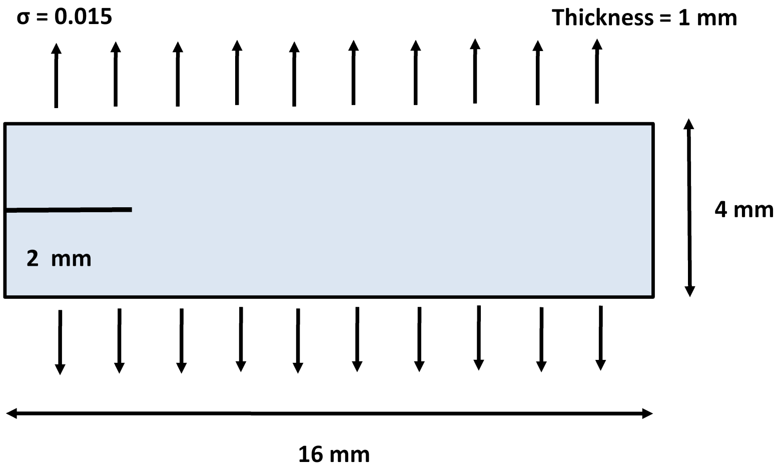

Figure 7.4: 2D model of a rectangular specimen with initial notch of 2 mm. Initial strain is 0.015, with node thickness of 1 mm. Model dimensions are 16mm per 4mm.

A first version of the mesh was composed by 74,257 nodes and 36,864 bulk elements. Due to the regular mesh pattern, we employed a simple procedure to subdivide the elements into 8 color groups, which is the optimal. Total simulated time is 2 μs, in 10,000 steps of 0.2 ns. Figure 8.1 shows an extruded 2D model after the simulation and fracture propagation and Figure 8.2 shows the fracture evolution on the same 2D model. The refined version of the same model with 295,969 nodes and 147,456 bulk elements was also tested using timesteps of 0.5 ns, performing 40,000 simulation steps. In both experiments, fractured facets are checked at every 10 simulation steps, and cohesive elements inserted as necessary. The principal stress was plotted with the fracture propagation and is shown in Figure 6.3. Tables 7.2 and 7.3 present results for both the mesh and its refined version. Figures 7.5 and 7.6 show the final plotted image of the T6 mesh and its refined version at the end of the simulation. The fractured evolved in a straight path in consequence of the intial noth of the model and the transverse strain applied on the model. Figure 7.7 shows the nodal stress wave propagation with the fracture and simulation evolution.

No. of bulk elements	No. of nodes	No. of new nodes	No. of Cohes. elem.	No. of Colors
36,864	74,257	1,901	979	8
147,456	295,969	5,842	2,976	8

Table 7.2: Simulation and mesh parameters for a T6 mesh and its refined version.

No. of bulk elements	Timestep	CPU time	GPU time	Speedup
36,864	2.0e-9	410.181 s	11.788 s	34.8
147,456	0.5e-9	6,537.839 s	153.809 s	42.5

Table 7.3: Simulation and mesh parameters and results (GPU speedup and GPU and CPU time) for a T6 mesh and its refined version.

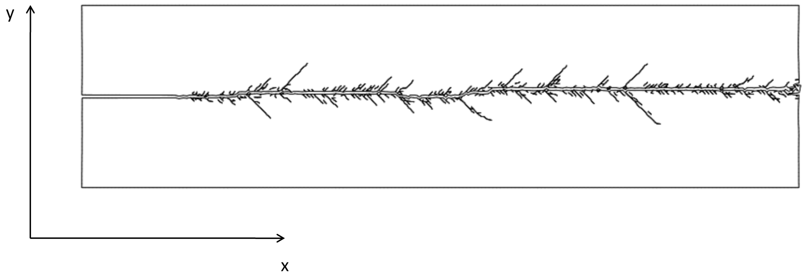

Figure 7.5: T6 FEM mesh with 36,864 bulk elements at the end of the fragmentation simulation.

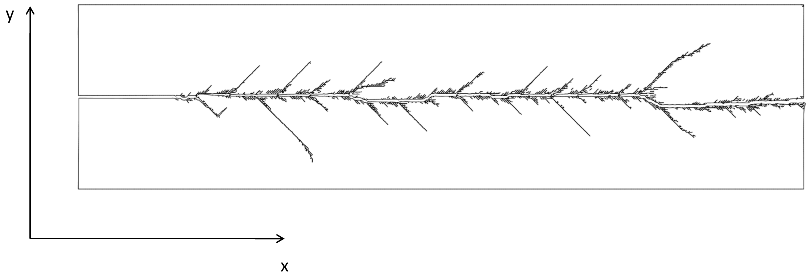

Figure 7.6: Refined T6 FEM mesh with 147,456 bulk elements at the end of the fragmentation simulation.

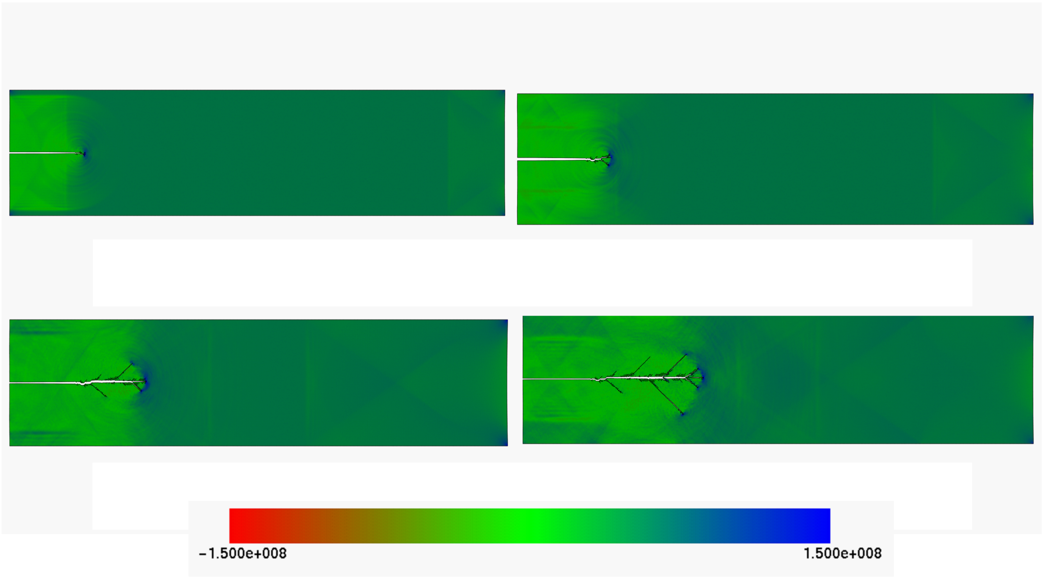

Figure 7.7: Principal stress evolution with crack propagation.

Graphs 7.8 and 7.9 present results for the portion and average simulation execution times for each kernel for the first T6 mesh. Graph 7.8 shows that the kernel responsible for computing the stresses is by far the most expensive one. However, the kernel responsible computing the internal forces dominates the simulation time with almost twice the time of the stress kernel due to the fact that the internal forces are computed at each time step, while stresses are computed at each ten steps. Another kernel that greatly occupies the simulation time is computing the cohesive forces due to its many numeric computations, although kernel splitting helped increase performance. The node duplication kernel does not occupy a large portion of the simulation because the number of cohesive elements is relatively small. Filtering elements, updating velocities, accelerations, displacements, and nodal masses, and applying boundary conditions are small job kernels with few global memory accesses and coalesce readings that do not have high execution time.

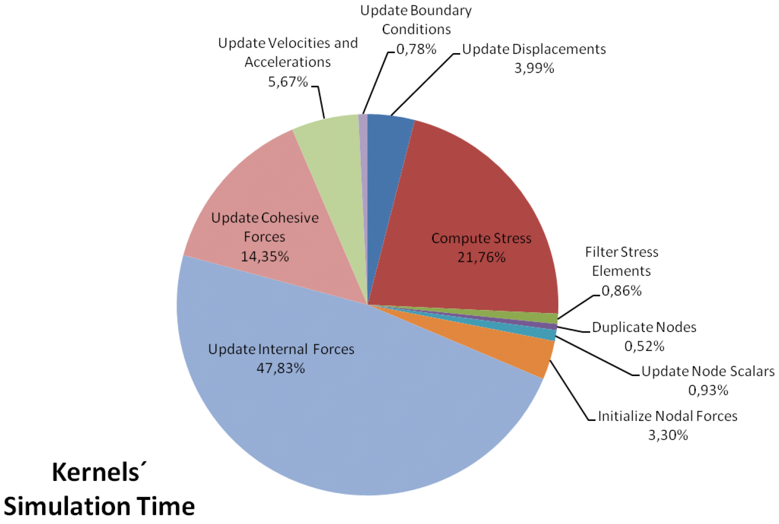

Figure 7.8: Execution time for each kernel relative to the entire simulation time for a T6 mesh with 36,864 bulk elements.

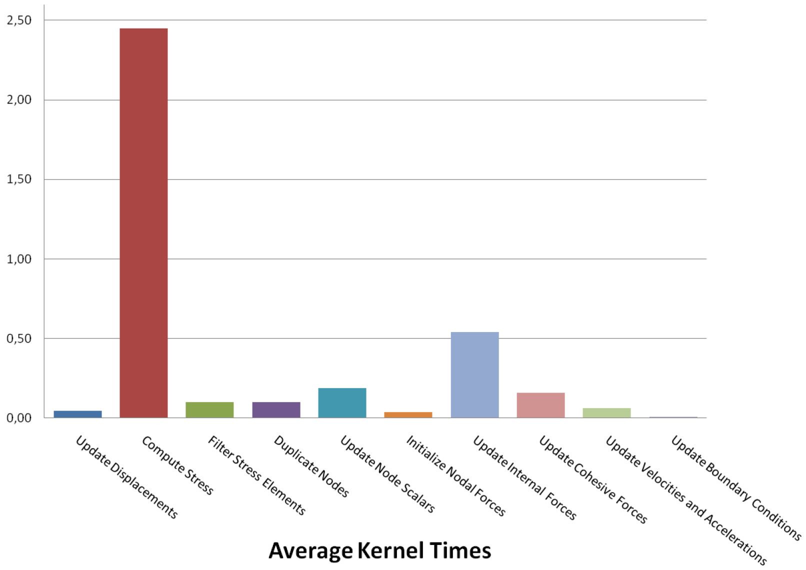

Figure 7.9: Kernels' average time for the simulation for a T6 mesh with 36,864 bulk elements.