An implementation of level set based topology optimization using GPU

D. Herrero, J. Martínez and P. Martí

Department of Structures and Construction, Technical University of Cartagena, Campus Muralla del Mar, 30202 Cartagena, Spain, e-mail: {david.herrero, jesus.martinez, pascual.marti}@upct.es

1. Abstract
This work presents the implementation of a topology optimization approach based on level set method in massively parallel computer architectures, in particular on a Graphics Processing Unit (GPU). Such architectures are becoming so popular during last years for complex and tedious scientific computation. They are composed of dozens, hundreds, or even thousands of cores specially designed for parallel computing. The speedup process consists of using these graphic units to exploit data parallelism of expensive and parallelizable parts of the method, while non-parallelizable parts are calculated in standard processing units (CPUs). The paper analyzes the computational complexity of the different steps of the method. The parallelization of both the finite element method and the specific operations of the optimization approach are also analyzed. The implementation of the method is benchmarked with some tests. The massively parallel results are compared with the sequential version of the method. The results show the advantages and disadvantages of the implementation of this method using GPU.

2. Keywords: Topology optimization, Level Set Methods, GPU Computing.

3. Introduction
Last years the use of GPUs for high performance scientific computation has become very popular. It can be mentioned its application in a wide variety of fields, such as image processing and computer vision [1], encryption algorithms [2], financial analysis and modeling [3], and simulation [4]. These devices also have been used in the field of structural optimization; we can mention the implementation of an iterative solver on a GPU [5] to address the topology optimization problem using the SIMP method. This implementation is restricted to a uniform grid to be able to take advantage of the parallelization capabilities of GPU architectures in order to address the topology optimization problem in high scale problems.

In this work, the level set method [7], developed by Osher and Sethian, is adopted to address the topology optimization problem. Such a method also has been broadly used in many diverse disciplines, such as structural optimization [8, 9], computer vision [10, 11] and planning [12], to name but a few. In the field of structural optimization, the approaches based on this method consider the boundary of the material as design variable. The evolution of such a boundary is represented by the Hamilton-Jacobi equation for which Osher and Sethian proposed a solution. Contrary to conventional structural optimization methods, the representation of design variables as a boundary avoids well-know problems such check-board pattern or intermediate densities in the final topology.

From the early work of Osher and Sethian [13], diverse works have studied the topology optimization problem [9, 14, 15] using the level set method in its conventional form, i.e. relating the normal velocity of the Hamilton-Jacobi equation with the shape sensitivities of the model. This approach does not permit the generation of new holes, and thus they require the initialization with an initial topology. Such a topology evolves by the union and elimination of the existing holes to the optimum topology. The topology design obtained by this kind of approaches has a high dependency on the initial topology [14]. This problem has motivated several works that aim to incorporate topology sensitivities [16] during the topology optimization process using level set method. Some of these works propose the combination of level set method in its conventional form with other optimization approaches [14, 17, 18].

In this work, we analyze the parallelization of the topology optimization problem using level set. For this purpose, the computational complexity of each step of the optimization approach are discussed. On the one side, the parallelization of the finite element method is analyzed, and on the other side, the specific operations of the level set method, evolution and sensitivities, are analyzed and their parallelization is discussed. Finally, some experiments validating the implementation of the topology optimization method.
and the comparison of the computational cost of one iteration of the optimization method using CPU and CPU+GPU are presented.

The document is organized as follows. Section 2 presents the fundamentals of the approach adopted to address the topology optimization problem using the level set method. Section 3 is focused on the implementation of the method and the analysis of the possibilities of parallelization of the different parts of the algorithm. Section 4 presents the validation of the implementation using three benchmarks and the evaluation of the computational cost of one iteration of the optimization method using CPU and CPU+GPU. Finally, some conclusions and future works are presented in section 5.

4. Topology optimization using level set method

The topology optimization problem consists of the optimization of a linear elastic solid (Figure 1) with domain $\Omega = \Omega \cup \partial \Omega$ in $D \in \mathbb{R}^{2,3}$ with boundary $\partial \Omega$. Given $b : \Omega \to \mathbb{R}^{2,3}$, $\pi : \partial_\Omega \Omega \to \mathbb{R}^{2,3}$, $\bar{t} : \partial \Omega \to \mathbb{R}^{2,3}$, the displacement field in the domain $u : \Omega \to \mathbb{R}^{2,3}$ is the solution of the state equation:

$$\begin{align*}
\nabla \cdot \sigma(u) + b &= 0 \text{ en } \Omega, \\
\sigma(u) \cdot n &= \bar{t} \text{ en } \partial_N \Omega, \\
u &= \pi \text{ en } \partial_D \Omega,
\end{align*}$$

(1)

where $n$ is the exterior normal vector in any position on the boundary $\partial \Omega$, $b$ are the volume forces, $\bar{t}$ are the surface forces applied on the boundary $\partial \Omega$, and $\sigma$ is the Cauchy stress tensor.

4.1 Topology optimization: Fundamentals

The topology optimization problem is formulated as:

$$\min_{\Omega} J(u, \Omega) = \int_{\Omega} F(u) d\Omega,$$

(2)

s.t. $\int_{\Omega} d\Omega \leq V$,

where $F(u) = \varepsilon(u)^T D \varepsilon(u)$ is the compliance, $D$ is the elasticity matrix, and $\varepsilon(u)$ is the strain tensor. The design variable is the domain of the structure, $\Omega$, defined explicitly on the limits of the integral and implicitly in $u$. The volume constraint (2) limits the volume of the structure. This constraint is incorporated to the objective function using the augmented lagrangian method [19] to obtain the augmented objective function:

$$\bar{J}(u, \Omega) = J(u, \Omega) + \lambda \left( \int_{\Omega} d\Omega - V \right) + \frac{r}{2} \left( \int_{\Omega} d\Omega - V \right)^2,$$

(3)
where $\lambda > 0$ is the Lagrange multiplier and $r > 1$ is the penalty parameter.

### 4.2 Boundary evolution using level set method

The level set method represents the boundary of the structure $\partial \Omega$ using a scalar level set function defined at the domain $D \in \mathbb{R}^{2,3}$ that contains the possible domains $\Omega$. The level set function is defined as:

$$\phi(x, t) = \begin{cases} > 0 & x \in \Omega, \\ = 0 & x \in \partial \Omega, \\ < 0 & x \in (D \setminus \Omega) \end{cases}$$ \hspace{1cm} (4)

The evolution of the level set function (4) during the optimization process is governed by the Hamilton-Jacobi [7] equation:

$$\frac{\partial \phi(x, t)}{\partial t} + V_n(x, t) |\nabla \phi(x, t)| = 0 \quad \text{in } D,$$

where $V_n(x, t)$ is the normal velocity of the boundary, and $t$ is a fictitious parameter $t \in \mathbb{R}^t$ that indicates the way of the evolution of the domain $\Omega(t)$ during the optimization process (Figure 2). The Hamilton-Jacobi equation (5) is solved by an explicit second order upwind scheme of finite differences on a Cartesian grid [12]. In order to avoid negative values of the normal velocity on the limits of $D$, the $\partial_n \phi = 0$ Neumann boundary condition is imposed. To ensure the convergence of the differential equation (5) using finite differences, the step time should satisfy the Courant-Friedrichs-Lewy (CFL) [21] condition:

$$\Delta t \leq \frac{h}{\max |V_n|}$$ \hspace{1cm} (6)

where $h$ is the minimum distance between points of the grid and $\max |V_n|$ is the maximum value of the points of the grid [12, 10]. The normal velocity $V_n(x, t)$ is the link between the structural optimization and the level set method, and it relates the descent gradient direction to the shape sensitivity as shown below.

### 4.3 Shape sensitivity

The shape sensitivity of $\bar{J}$ is related to the normal velocity $V_n$ by the equation [8, 9]:

$$d_n \bar{J} = \int_{\partial \Omega} (b \cdot u - \varepsilon(u)^T D \varepsilon u + \lambda) V_n d(\partial_\Omega).$$ \hspace{1cm} (7)

Considering the hypothesis that the Dirichlet boundary cannot move in its normal direction, and that the Neumann boundaries are fixed, $\lambda$ can be obtained by:
\[
\lambda = \max \left\{ 0, \lambda + \tau \left( \int_{\Omega} d\Omega - V \right) \right\}.
\]  

(8)

In order to ensure the minimization of the augmented objective function (3), the boundary should move with a velocity \( V_n \) satisfying \( J < 0 \). The following expression [8]:

\[
V_n = - (b \cdot u - \varepsilon(u)^T D \varepsilon(u) + \lambda)
\]  

(9)
relates the velocity \( V_n \) to the shape sensitivities.

4.4 Topology sensitivity

The conventional expression of Hamilton-Jacobi allows to represent modifications of the topology by the nucleation of holes in the initial topology [9]. The main disadvantage of this approach is that it can converge to local minimum. It is not possible to reduce the objective function from such a local minimum by only evolving the boundary. The topology should be modified generating new holes with \( \phi(x, t) < 0 \) [8]. This problem is addressed by incorporating a new term in the Hamilton-Jacobi equation [17] which depends on the topology. This term penalizes the negative values of the level set function; increasing them when it is suitable the generation of new holes and viceversa. This approach provides a first order Hamilton-Jacobi equation as follows:

\[
\frac{\partial \phi(x, t)}{\partial t} + V_n(x, t) |\nabla \phi(x, t)| + w \cdot G = 0 \quad \text{in } D,
\]  

(10)

where \( w \in \mathbb{R}^t \) represents the influence of the additional term \( G \) that depends of the topology sensitivities. The following equation [17] can be used to calculate it:

\[
G(x, t) = - sgn \phi(x, t) \cdot g(x, t)
\]  

(11)

where \( g(x, t) = d_T J(\Omega(t), x) \).

Considering that it is not possible the nucleation of void areas, the equation (11) is null in areas where \( \phi \geq 0 \). The following equation [14] can be used to calculate the topology sensitivity in a two-dimensional domain:

\[
d_T J = \frac{\pi}{2\mu(\lambda + \mu)} \left\{ 4\mu \varepsilon(u)^T D \varepsilon(u) + (\lambda - \mu) \text{tr}(\varepsilon(u)) \text{tr}(D \varepsilon(u)) \right\},
\]  

(12)

where it is assumed that the hole boundary is traction free (\( b = 0 \)). \( \lambda \) and \( \mu \) are the Lamé parameters for a solid material. The topology sensitivity of the volume when the hole is a unit circle is \( d_T V = -\pi \).

5. Level set based topology optimization implementation using GPU

In order to implement the level set method, the state equation (1) is discretized using the finite element method leading to \( KU = F \), where \( F \) and \( U \) are the global displacement and force vectors respectively, and \( K \) is the global stiffness matrix. The optimization problem in discretized form is then formulated as follows [20] :

\[
\min_x \ c(x) = U^T K U = \sum_{e=1}^{N} u_e^T k_e u_e = \sum_{e=1}^{N} x_e u_e^T k_e u_e
\]  

(13)

s.t.: \[
\begin{align*}
V(x) &= V\text{req} \\
KU &= F \\
x_e &= 0 \text{ or } x_e = 1 \forall e = \{1, \ldots, N\}
\end{align*}
\]
where $x = (x_1, \ldots, x_N)$ are the element “densities”. The $x_e$ values of $x$ are 0 or 1 for void and solid elements respectively, and thus there is not intermediate densities. $u_e$ and $k_e$ are the displacement and local stiffness, respectively, of elements $e$. $k_1$ is the stiffness of matrix after consider if the element is void or solid. $c(x)$ is the compliance. $V(x)$ is the total number of solid elements and $V_{req}$ is the number of required solid elements.

The equation used to calculate the level set function (4) is the discretized equation (10):

$$\frac{\partial \phi(x, t)}{\partial t} = -v_n(x, t) \nabla \phi(x, t) - w \cdot g.$$  \hspace{1cm} (14)

The flowchart of discretized topology optimization method based on level set is depicted in Figure 3. The stages are described in detailed below.

**Initialization:**

The geometry of the problem domain is defined and discretized using a regular Cartesian grid. This grid is used to discretize the level set function along its central point. Considering that $x_e$ represents the central position of the element $e$, the discretized level set function $\phi$ should satisfy:

$$\phi(x_e, t) = \begin{cases} > 0 & x_e = 1 \\ < 0 & x_e = 0 \end{cases}.$$  \hspace{1cm} (15)

This function $\phi$ can be updated, to represent a different structure, solving (14) numerically. The initialization of this function $\phi$ is performed by a signed distance function, and the shape and topology sensitivities are initialize to zero.

**Stage 1: Finite element method**

For each step of the optimization approach, the discretized elasticity problem $KU = F$ should be solved using the finite element method. Despite the Dirichlet and Neumann boundary conditions remain fixed during the topology optimization, the structure is modified by the design variable $x_e$ at each step. The local stiffness matrices of void elements ($x_e = 0$) are assigned to a minimum value $k_e > 0$ to avoid singularities when solving $KU = F$. This can significantly increase the computational cost of the solver, especially when iterative solvers are used. According to our experience, this stage has the higher computational cost of the optimization algorithm.
Stage 2: Calculation of objective function and sensitivities

The shape sensitivity of the objective function is obtained using the displacements as follows [8]:

\[
\frac{\partial c(x, t)}{\partial \Omega} |_{c} = -u_k^T k_{e} u_e. \tag{16}
\]

The topology sensitivity of the objective function is calculated using (12). All the calculations performed on the regular Cartesian grid are similar to the processing performed in image processing. These kind of operations are parallelizable and they only depend on the displacements obtained in the previous stage.

Stage 3: Update design

At this stage the evolution of the level set function \( \phi \) is implemented using a upwind scheme of finite differences. The time step should satisfy the Courant-Friedrichs-Lewy (CFL) stability condition (6). The operations performed in this step are also on the regular Cartesian grid: in particular displacements and convolutions on the grid to calculate the gradients of the level set function. All these operations are also typically used in image processing, where GPU devices have shown their impressive advantages. In fact, some educational works make use of image processing tools, e.g. the Image Processing Toolbox of MATLAB in [20], to implement the optimization method based on level set method.

Stage 4: Reinitialization of level set function

The level set function \( \phi \) is reinitialized using a signed distance function to ensure the precision of the numerical solution of \( \phi \) (14). However, this reinitialization should not be applied too often [17] in order to allow the hole nucleation in (14). Therefore, a trade-off between the precision of the numerical solution of (14) and the generation of new holes should be found by the update frequency of the level set function.

The operations performed at this stage consist of: (1) padding operations on the Cartesian grid to represent void elements at the border of the structure and (2) distance operations between the central points of the grid cells. As previously mentioned, they are fully parallelizable operations.

Check of convergence

The converge checking evaluates if the constraints of the topology optimization method are satisfied: the volume fraction \( V(x) \) should be close to the required volume fraction \( V_{req} \), and the objective function should not significantly change during last iterations.

Discussion

All the operations in the optimization method are performed on the Cartesian grid, which facilitates the parallelization of the implementation. However, we have observed that the bottleneck in the processing is the resolution of the finite element equations. According to our experience, the timing ratio between FEM solver and optimization increases with the size of the regular grid.

6. Numerical examples

This section presents some numerical examples to validate the implementation of the topology optimization method based on level set method. The tests consist of three benchmarks: the cantilever problem, the half-wheel problem, and the MBB beam problem. The default parameters of these tests are as follows:

- the fixed design domain is discretized using a structural mesh and four-node quadrilateral plane stress elements,
- the isotropic linear elastic material has Young’s modulus = 1 and Poisson’s ratio = 0.31,
- the residual for the preconditioned conjugate gradient is set to 1e-8.
6.1 Cantilever 2D

The first problem is the cantilevered beam shown in Figure 4(a), which is a benchmark problem in topology optimization. As shown in Figure 4(a), the length of the domain is \( L = 64 \) mm and the height is \( H = 40 \) mm. The cantilever is subjected to a concentrated load \( P = 80 \) N at the middle point of the free end. The volume constraint is 40% of the total domain volume. The domain is discretized by 160x80 elements. The domain is initialized with a configuration having no holes. The level-set function is reinitialized every 3 iterations of the optimization algorithm. The parameter \( w \) which determines the influence of the topology sensitivity is set to 4. The number of time steps for which the Hamilton-Jacobi equation is solved is set to 8. The final topology is obtained after 46 iterations of the optimization algorithm. Figure 4 (b-d) shows the final topology along with the intermediate topologies. The evolution of the compliance and the volume fraction are shown in Figure 7(a). The compliance increases over the first 25 iterations as the volume fraction decreases to meet the constraint. At this point the compliance decreases to a value of 77.25 at iteration 46.

6.2 Half-wheel problem

The second problem is the half-wheel problem shown in Figure 5 (a). The length of the domain is \( L = 60 \) and the height is \( H = 30 \). The problem is subjected to a concentrated load \( P = 1 \) at the bottom half of the vane. The domain is discretized by 60x30 elements. The volume fraction is constrained to 35% of the initial volume. The domain is initialized with a configuration having no holes. The level-set function is reinitialized every 4 iterations of the optimization algorithm. The parameter \( w \) related to the topology sensitivity is set to 4. The number of time steps for which the Hamilton-Jacobi equation is solved is set to 6. The final topology is obtained after 42 iterations of the optimization algorithm. Figure 5 (b-d) shows the final topology along with the topologies at intermediate iterations. The evolution of the compliance and the volume fraction are shown in Figure 7(b). The value of the compliance at the optimal design is 18.2561.

6.3 MBB Beam
The third problem is the MBB beam shown in Figure 6 (a). This problem is a widely used benchmark in topology optimization. The length of the domain is $L = 288$ and the height is $H = 48$. The problem is subjected to a concentrated load $P = 2$ at the upper half of the vane. The domain is discretized by 288x48 elements. Symmetry boundary conditions are not considered. The volume fraction is constrained to 40% of the initial volume. The domain is initialized with a configuration having no holes. The level-set function is reinitialized every 8 iterations of the optimization algorithm. The parameter $w$ related to the topology sensitivity is set to 4. The number of time steps for which the Hamilton-Jacobi equation is solved is set to 3. The final topology is obtained after 63 iterations of the optimization algorithm. Figure 6 (b-d) shows the final topology along with the topologies at intermediate iterations. The evolution of the compliance and the volume fraction are shown in Figure 7(c). The value of the compliance at the optimal design is 454.7.

### 6.4 Speedup using GPU

The implementation presented above has been validated through the optimization of three benchmarks widely studied in the field of topology optimization. According to the timing of the different stages of the algorithm, we have observed that the bottleneck is the resolution of the system of equations derived from the finite element formulation. In order to demonstrate the speedup of the GPU implementation the MBB beam problem is solved modifying the degree of discretization. We examine several cases whose degree of discretization is subjected to the following mesh parameters: $n^*288 \times n^*48$. Table 1 shows the results for different tessellation sizes of the MBB problem. The comparison is done using the timing of one iteration of the optimization algorithm. One can observe that for problems with a small number of degrees
of freedom (DoF) the performance of the GPU implementation decreases in comparison with the CPU implementation. However, as the number of DoF increases, the performance of GPU improves leading to speedups around 3. It indicates that the GPU implementation shows predominance in optimizing large scale problems due to its high degree of parallelism, of which the CPU is lacking.

### Table 1: Speedup for one iteration of the optimization algorithm in the MBB beam problem.

<table>
<thead>
<tr>
<th>n</th>
<th>DoF</th>
<th>CPU (seg)</th>
<th>CPU (iterations)</th>
<th>GPU (seg)</th>
<th>GPU (iterations)</th>
<th>Speedup ( \frac{CPU}{GPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28322</td>
<td>1.81</td>
<td>1472</td>
<td>3.42</td>
<td>1512</td>
<td>0.529</td>
</tr>
<tr>
<td>2</td>
<td>111938</td>
<td>16.07</td>
<td>2947</td>
<td>15.31</td>
<td>2996</td>
<td>1.049</td>
</tr>
<tr>
<td>3</td>
<td>250850</td>
<td>54.16</td>
<td>4423</td>
<td>49.17</td>
<td>4477</td>
<td>1.101</td>
</tr>
<tr>
<td>4</td>
<td>445058</td>
<td>132.83</td>
<td>5900</td>
<td>87.12</td>
<td>5963</td>
<td>1.525</td>
</tr>
<tr>
<td>5</td>
<td>694562</td>
<td>265.66</td>
<td>7366</td>
<td>143.112</td>
<td>7445</td>
<td>1.856</td>
</tr>
<tr>
<td>6</td>
<td>999362</td>
<td>462.18</td>
<td>8827</td>
<td>237.77</td>
<td>8928</td>
<td>1.944</td>
</tr>
<tr>
<td>9</td>
<td>2245538</td>
<td>1640.86</td>
<td>13244</td>
<td>694.33</td>
<td>13376</td>
<td>2.363</td>
</tr>
<tr>
<td>12</td>
<td>3989378</td>
<td>4160.08</td>
<td>17655</td>
<td>1574.71</td>
<td>17823</td>
<td>2.641</td>
</tr>
<tr>
<td>15</td>
<td>6230882</td>
<td>10340.1</td>
<td>22051</td>
<td>3027.59</td>
<td>22268</td>
<td>3.415</td>
</tr>
<tr>
<td>20</td>
<td>11072642</td>
<td>24608.3</td>
<td>29394</td>
<td>7444.41</td>
<td>29679</td>
<td>3.305</td>
</tr>
</tbody>
</table>

### 6. Conclusion

An implementation of level set based topology optimization using GPU has been presented in this work. According to our experience, we have observed the following conclusions: (1) working on a regular grid allows to exploit massive parallelization of the optimization algorithm, (2) the bottleneck of the problem lies on the finite element analysis, which is a common problem in many applications, (3) the operations needed by the optimization algorithm are similar to the used in the the field of image processing, where the graphical processing units have shown numerous benefits. This gives us some interesting information for future work in order to improve performance. In the future, we’d like to investigate different ways to alleviate data transfer delays between GPU and CPU memories.

### 5. Acknowledgements

The work presented in this paper was carried out with the support of the Spanish Ministry of Economy and Competitiveness under DPI2011-26394 research project. Its support is greatly appreciated.

### 6. References


R. Courant, K. Friedrichs, and H. Levy, Über die partiellen Differenzengleichungen der mathematischen Physik, Mathematische Annalen (in German), 100(1), 32–74, 1928.