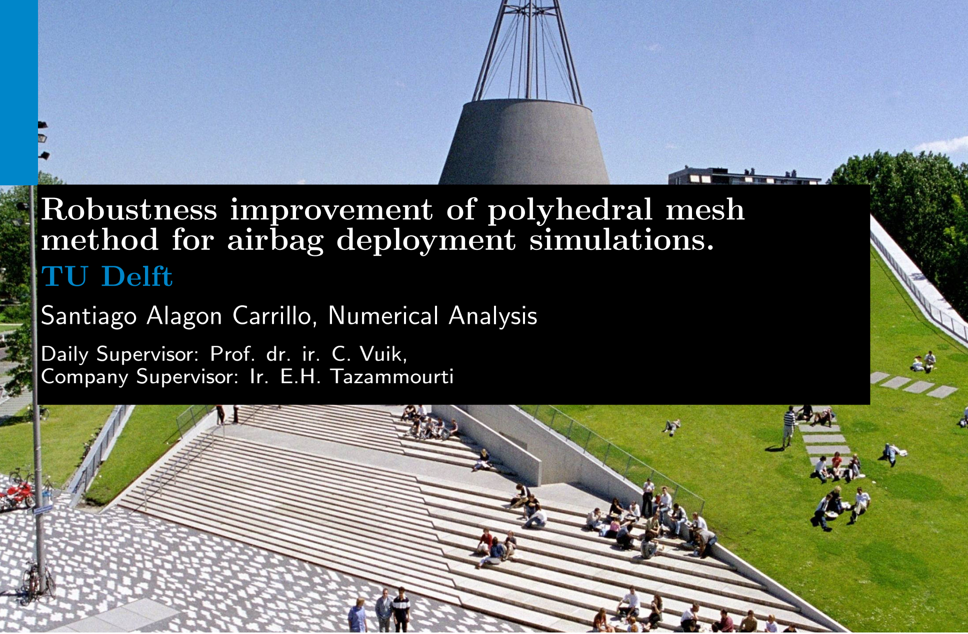


Robustness improvement of polyhedral mesh method for airbag deployment simulations.

TU Delft

Santiago Alagon Carrillo, Numerical Analysis

Daily Supervisor: Prof. dr. ir. C. Vuik,
Company Supervisor: Ir. E.H. Tazammourti

Outline

MADYMO and Gasflow2.

Cartesian Mesh and Cut-cells.

Discretization, Numerical Scheme and Coupling.

Exact Geometry.

Point-in-Polyhedron problem.

Search algorithm.

Topological tests.

Exact Geometry and Robustness issues.

Research questions.

Test Problems.

Looking for solutions and First results.

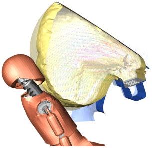
Future work.

Madymo

MADYMO is a software package developed to analyze safety systems in transport vehicles.

Gasflow

Gasflow is the module in charge of the numerical computations used in MADYMO to simulate the deployment of airbags.



The solution is approximated using *Euler's equations*

$$\frac{d}{dt}\int_{\Omega} \mathbf{q}\, d\Omega + \int_{S} \Phi(\mathbf{q}, \hat{n})\, dS = \mathbf{S}(\mathbf{q}) \qquad (1)$$

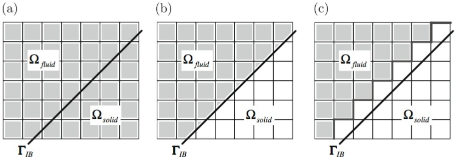
where

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix}, \text{ and } \Phi = \begin{pmatrix} \rho v_n \\ \rho v_n u + p \cdot n_x \\ \rho v_n v + p \cdot n_y \\ \rho v_n w + p \cdot n_z \\ \rho v_n H \end{pmatrix} \qquad (2)$$

Cartesian Mesh and Cut-cells 1.

To discretize the domain a cut-cell Cartesian Mesh is used. The discretization is performed as follows:

- The whole computational domain is divided into cartesian, hexahedral, cells.
- The cells are flagged as **active cells** if they belong entirely to the flow region, **inactive** if they belong entirely outside the flow region and **cut-cells** if they intersect the boundary of the object immersed in the flow, the airbag.
- The geometry of the airbag is preserved by removing the inactive region of the cut-cells.

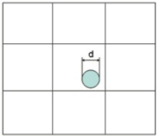


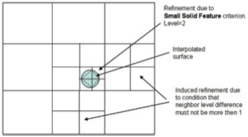
Cartesian Mesh and Cut-cells 2.

- Only the *active cells* and the active regions of the *cut-cells* are used to compute the solution

Cartesian Mesh discretization is more suitable than body fitted discretization for this problem because

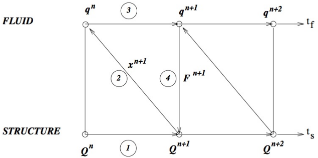
- The geometry changes at every time step. The cartesian meshing process can be implemented as an automatic procedure.
- Preserves the geometry and results in conservative schemes.
- Cut-cells are de-coupled from the surface description.
- Refinement is possible to capture more detail of the geometry.





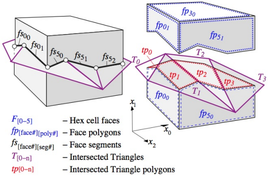
Discretization, Numerical Scheme and Coupling.

- The boundary of the airbag is approximated by a Finite Element triangulation.
- Euler equation's are discretized using a Finite Volume method over hexahedral cells that cover the whole flow domain.
- The Finite Volume scheme is based on a Roe scheme using flow-differencing splitting the approximate the flux derivatives.
- Boundary conditions are enforced through the specification of the fluxes at the boundary.
- At each time step one evaluation of FE and flow solver are performed, *loose coupling* procedure.



Exact Geometry

The **exact geometry** of the cut-cells and a **robust method** to determine active and inactive cells as well as active and inactive regions of each cut-cell are important to preserve accuracy and performance of the finite volume solver.



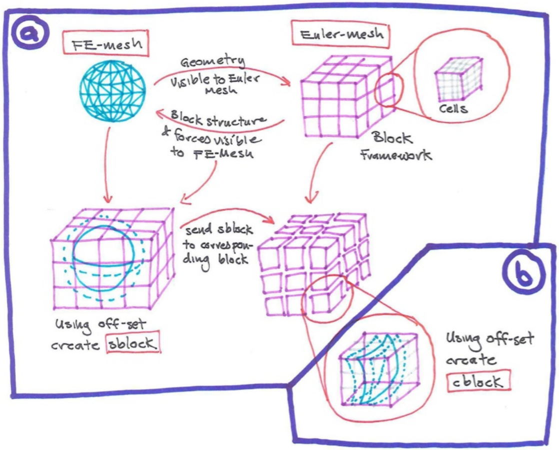
Search algorithm Computes the exact geometry of the cut-cells and classifies the cells and cut-cells regions into active and inactive.

Search Algorithm 1

Search algorithm
The whole domain is decomposed into *Blocks*, which are sets of cells, each *Block* containing the same number of cells.
Using the *Block* structure the search algorithm is performed into three stages.

- **Global search level**. This search determines a list of candidate triangular segments, form the FE triangulation, that may intersect each bock.
- **Euler search**. This search determines a set of triangular segments that may intersect every cell.
- **Exact Geometry**. This search determines all intersections between Euler cells and FE-segments using the intersection candidate list. After that the face polygons, boundary faces and polyhedrons of each cut cell are obtained.

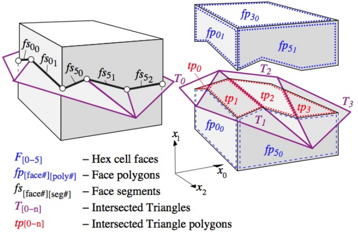
Search Algorithm 2



Figuur: (a) global search level, (b) Euler search level

Search Algorithm 2

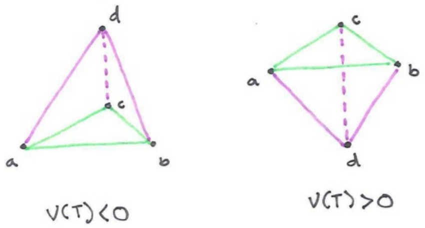
- The *Global search* and *Euler search* are based on **topological** tests to determine the intersection of the cells and FE segments.
- The *Exact Geometry* is obtained by **constructing the intersection** points of the FE triangulation edges and the cell faces.



Topological tests 1

Signed volume of a simplex

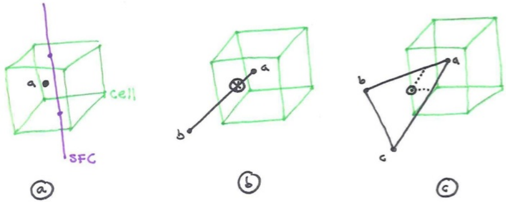
$$V(T_{v_0 v_1 \dots v_d}) = \frac{1}{3!} \begin{pmatrix} v_{a_0} & v_{a_1} & v_{a_2} & 1 \\ v_{b_0} & v_{b_1} & v_{b_2} & 1 \\ v_{c_0} & v_{c_1} & v_{c_2} & 1 \\ v_{d_0} & v_{d_1} & v_{d_2} & 1 \end{pmatrix} \quad (3)$$



Figuur: Signed Volume Property

Topological tests 2

- *Point test*. For each FE node find in which cell it resides, see Figure(3a).
- *Edge test*. For each FE edge test wether it intersected by a certain cell's face, see Figure(3b).
- *Slice test*. For each FE segment test wether it is intersected by a certain cell's edge, see Figure(3c).



Figuur: Exact Tests

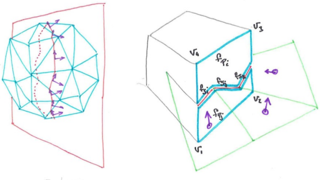
Exact Geometry and Robustness issues 1

State of the art in MADYMO:

- The implementation in MADYMO is capable of determining the exact geometry of the cut-cells.
- The implementation in MADYMO is capable of determining the inner and outer regions of the cut-cells in general cases, but there are particular cases where it fails.

Implementation

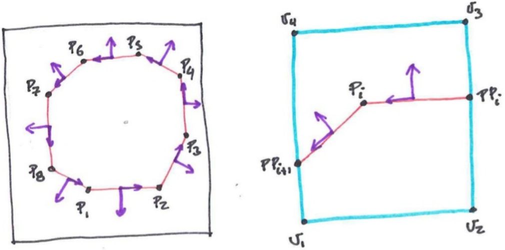
The airbag is a **simple polyhedron**. Considering a transversal cut the problem can be equivalently stated in two-dimensions, this is the case for each cell-face.



Figuur: Equivalence to two-dimensional problem

Exact Geometry and Robustness issues 2

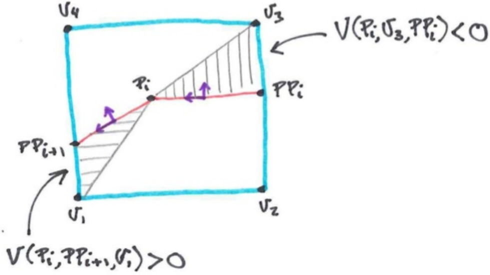
The outward normal to the FE triangulation is known, this determines a consistent order for the points in the two-dimensional equivalent problem.



Figuur: Consistent ordering

Exact Geometry and Robustness issues 2

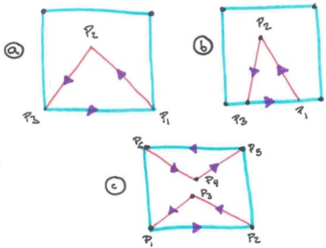
Determination of the inner and outer regions of the cut-cell's faces using the consistent ordering defined by the normal, the *signed volume of a simples* and a vertex as a test point,



Figuur: Consistent ordering

Exact Geometry and Robustness issues 3

Robustness problems arise when no vertex can be chosen as test point.



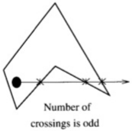
Figuur: No test point can be chosen systematically

.

Research Question

How to determine which region in a cut-cell belongs to the flow and which out-side in a consistent way?

In literature there are standard techniques to solve this so called *Point in polygon problem*, such as the *line crossing test*

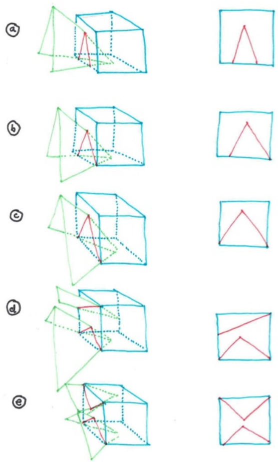


Figuur: Line crossing test

but this techniques have other problems such as

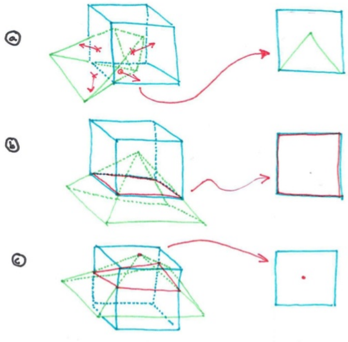
- Election of test point, round off error, use of geometric constructors, ignores information provided by normal vector.

Test Problems1



Figuur: Test cases with no test point.

Test Problems2

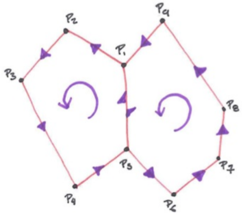


Figuur: Test cases with coplanar FE segments and cell faces.

.

Looking for solutions and First results, 1.

Consistency in traversing direction 1.
Two simple polygons sharing an edge traversed in a counter-clock wise direction result in the shared edge being traverses in opposite directions for each polygon.



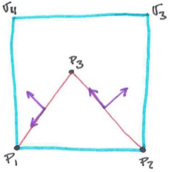
Figuur: Simple polygons with coinciding edge.

.

Looking for solutions and First results, 2.

Consistency in traversing direction 2.

- For each cut-cell face, traverse each of the face polygons in counter-clock wise direction.
- For each of the polygons, if the counter-clock wise direction is not consistent with the direction dictated by the normal of at least one of the face segments then the face polygon is determined to be outside.



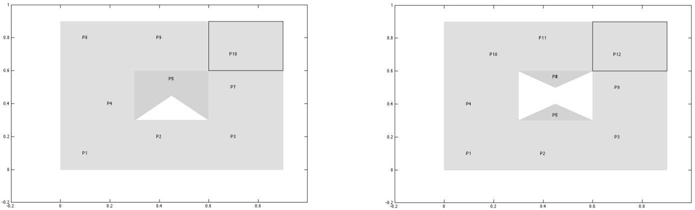
Figuur: Traversing direction consistency.

- CCW Polygons: P_1, P_2, P_3, P_1 and $P_1, P_3, P_2, V_3, V_4, P_1$
- Face segments in normal induced direction: P_2, P_3 and P_3, P_1

$\Rightarrow$ $P_1, P_3, P_2, V_3, V_4, P_1$ is outside and P_1, P_2, P_3, P_1 is inside

Looking for solutions and First results, 2.

This method has been implemented in Matlab for two of the test cases. A correct Inside/Outside determination was obtained for both cases.



Figuur: First results

.

Future work.

- Adapt and implement the method for test cases with co-planar triangular segments.
- Adapt and implement the method for non simple polygons.
- Explore other possible method based on a consistent election of test points.
- Explore the use of ideas from the *Level set Method*