

A New Simulation Method for Airbags With Multiple Compartments (Adaptive Euler Domains)

Authors:

R.S. Sadeghi and W.A. van der Veen*
MSC.Software Corporation, Santa Ana, California.
*MSC.Software BV, Gouda, The Netherlands.

Correspondence:

R.S. Sadeghi
MSC.Software Corporation
2 MacArthur Place
Santa Ana, CA U.S.A.

Tel: +011-(714)-444-5020
Fax: +011-(714)-784-4419
E-mail: reza.sadeghi@mscsoftware.com

ABSTRACT

The paper presents a new simulation method for airbags with multiple compartments.

In recent years many accidents happened with passengers that were out of position (OOP) when their airbag deployed. Therefore new safety regulations have come into effect in the USA and an airbag now has to meet several requirements that concern out of positioning. In meeting these requirements simulations of deploying airbags are very useful and are widely used.

In OOP the airbag hits the passenger when it is still deploying and the pressure field is far from uniform and so there is a complicated interaction between gas flow and the deploying membrane. This calls for an analysis in which the dynamics of the airbag membrane is coupled with a CFD (Computational Fluid Dynamics) solver.

To weaken the frontal blow, an airbag with multiple compartments is sometimes used. The idea is to divert the gas stream in the airbag away from the frontal direction into sideways directions. Since these airbags especially need to operate well in OOP setting the CFD approach is required. The paper presents a new method to simulate a multi-compartment airbag in which each compartment is modeled by the CFD approach and each with a separate Euler domain.

Before discussing multi-compartment simulations, we present background on simulations for one airbag using one Euler mesh. In these simulations the gas flow is described by the conservation laws of mass, momentum and energy. Also modeling of the interaction between CFD solver and airbag membrane solver is discussed.

The conservation laws are applied to the 3D objects that are formed by taking an Euler element and cutting away everything that is outside the airbag surface. The boundary of this 3D object is in general a multi-faceted surface that consists of:

- (1) Polygons that connect an Euler element to another Euler element and that are called faces.
- (2) Polygons that connect Euler elements to the airbag surface, and are called polpacks.

In the conservation laws several surface integrals occur. Many of these integrals have to do with transport and are computed by running over faces and polpacks. This gives rise to mass being transported across faces. In most cases there is no transport across polpacks since they are part of the airbag surface.

For multi-compartment simulations we have to consider multiple Euler domains. Each Euler domain models one airbag. Communication between the airbags proceeds via the holes that connect one airbag with the other. Holes are modeled as a piece of airbag surface that is fully porous.

Given that the largest possible timestep depends on the mesh size, each Euler domain could be advanced with a specific timestep. This would require subcycling. For the multi-compartment application mesh sizes are expected to be of comparable magnitude and all Euler meshes are advanced with the same timestep.

Applying conservation laws is straightforward except at a polpack that connects an Euler element to the surface that models the hole. This polpack refers to only one Euler element in one domain and cannot provide the communication between the domains. It has to be replaced by a polpack that can refer to one Euler element in the first domain and one Euler element in the second domain. These new polpacks will be called flow polpacks.

In the vicinity of each hole two Euler domains overlap. Each flow polpack is located inside this overlap and should be completely in an element in the first mesh and completely in an element in the second mesh. Clearly flow polpacks cannot be created by the polpack creation procedure mentioned that only considers one Euler domain.

A good solution to this problem is to create an auxiliary mesh, that we shall call the hole mesh. This mesh is obtained by putting the two meshes on top of each other. So each mesh is a subset of this hole mesh. To make flow polpacks the airbag surface and the hole mesh are passed to the polpack creation routine. This gives polpacks that refer to elements in the hole mesh. Since each hole mesh element lies in exactly one element of the first and in exactly one element of the second mesh we can easily form flow polpacks.

The flow polpacks are used to transport mass, energy, and momentum from one domain to the other. Also through, these flow polpacks, the Euler elements in one domain exert pressure forces on the boundary of Euler elements in another domain.

For this new method three test problems are presented. The first two are theoretical, but the last one is a real life airbag simulation. The results show that the method presented is a promising method for multi-compartment airbag simulations.

INTRODUCTION

This paper presents a new method for the simulation of multi-compartmented airbag deployment, using CFD.

The value of numerical airbag simulations is that they shorten the design process and reduce the number of experiments that are needed to get to a reliable and safe end product. In this paper we will describe the recent enhancements made to the algorithms in MSC.Dytran. This is a commercial software package for short-term transient analyses that involve structural parts and/or CFD parts. Important areas of applications are airbag deployment, tank fuel sloshing and impact processes. For further information refer to [3]. For background on CFD simulations with airbags see [1,2] and for CFD in general see [4,5].

For airbag simulation one has the choice between an accurate CFD analysis, and a potentially inaccurate uniform pressure description of the airbag.

The accuracy of a uniform pressure description of the airbag depends largely on the position of the passenger with respect to the deploying airbag. If the passenger is located close to the deploying airbag, the airbag will hit the passenger while it is still being deployed. The gas flow inside the airbag has a major influence on the loading the occupant will feel. The problem with the uniform pressure method is that the gas flow inside the airbag cannot be calculated.

The cases when the airbag will contact the passenger before full deployment are called Out of Position (OOP). Recently new safety regulations have come into effect in the USA. Many of them apply to OOP airbags.

New types of airbags, which are safer for OOP passengers have been developed. Most injury to the passenger occurs when high velocity gas flows out of the inflator into the airbag and hits the passenger. The idea is to divert this stream sideways, by employing multiple compartments, and thus weaken the blow and impact on the passenger. Another example of a multi-compartmented airbag is a side curtain.

Multi-compartmented airbag could only be analyzed by a simplified approach, where only one of the compartments could use CFD. The current enhancements make it possible to:

- Analyze multiple compartments with the CFD approach.
- Simulate flow from one CFD domain into another CFD domain.
- The individual CFD domains are dynamic and adaptive. The user does not need to mesh the CFD domains, nor does he have to worry about the size, since the CFD domains will automatically follow the deploying airbag compartments.
- Flow through both small and large holes is accurately calculated.

This paper describes these enhancements for multi-compartment airbags, and presents both theoretical and real life examples.

FORMULATION OF THE PHYSICAL PROBLEM

In the deployment of an airbag an inflator supplies high velocity gas into an airbag causing it to expand rapidly. The gas inside the airbag is assumed to be ideal, to be of constant entropy and to satisfy the equation of state

$$p = (\gamma - 1)\rho e$$

Here p , ρ and e are respectively the pressure, density, specific internal energy and γ is the ratio of the heat capacities of the gas. The gas flow is described by the conservation laws for mass, momentum and energy, that read:

$$\begin{aligned} \frac{d}{dt} \int_V \rho dV + \int_A \rho(u \cdot n) dA &= 0 \\ \frac{d}{dt} \int_V \rho u_i dV + \int_A \rho u_i (u \cdot n) dA &= - \int_A p n_i dA \\ \frac{d}{dt} \int_V \rho e dV + \int_A \rho e (u \cdot n) dA &= - \int_A u_i p n_i dA \end{aligned}$$

Here V is a volume, A is the boundary of this volume, n is the normal vector along the surface A and u denotes the velocity vector in the volume.

The gas exerts a pressure load on the airbag causing it to expand. This expansion puts the airbag under tensile stress lowering the expansion rate. On the other hand the airbag defines a moving boundary to the gas inside the Euler domain.

There may be multiple airbags that are connected to each other through holes in the airbags. Heat conduction and heat transfer is not taken into account.

INTRODUCTION: ANALYSES USING ONE EULER DOMAIN

Before discussing the methods for analyses with several Euler domains, we briefly discuss the solution procedure used for analyses with one Euler domain and one airbag. In this paper we limit ourselves to analyses that are first order in space and time.

In the simulation there will be an airbag surface and gas that fills up the inside of this surface. The airbag surface consists of flexible membrane elements that deform under tensile stresses and cannot carry compressive loads. An explicit finite element solver solves the airbag dynamics, and an explicit Euler solver models the gas inside the airbag. The interaction between these two solvers takes place in two ways

- The mass in the Euler elements exerts a pressure load on the airbag. These loads constitute the boundary conditions for the finite element solver, resulting in new gridpoint accelerations and velocities for the airbag. Finally the airbag gridpoints are moved with the new velocities
- The airbag grid points move and so the Euler mesh has a new boundary. Consequently, the volume of mass in each element may change. Since density is mass divided by the volume of the mass, densities will also change, and so will pressure.

THE EULER SOLVER

Within an Euler element the state variables are assumed to be constant, and are defined in the element center. The elements will remain stationary throughout the simulation.

The Euler mesh is updated by applying a finite volume method directly to the physical domain. So no coordinate transformation is needed. The finite volume method is applied to the 3D object that consists of that part of the Euler element that is inside the airbag. This is in general not a cube but a multi-faceted object. For the 2D case this is sketched in Figure 1.

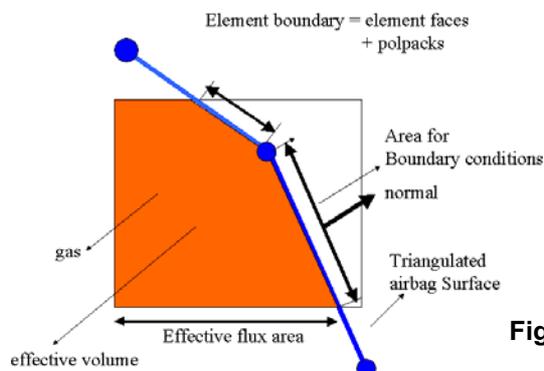


Figure 1: The boundary of an element

In this figure the square is the Euler element. It is intersected by the airbag surface. The area bounded by the square and the airbag surface is the effective volume of the gas in the element. The boundary of this portion consists of two types of surfaces:

- (1) Euler element boundaries that connect two neighboring elements and will be called 'Euler faces'.
- (2) Parts of the airbag surface that are within the Euler element. They will be called 'polpacks', which is short for "polyhedron packets".

A polpack is the intersection of an airbag element with an Euler element and is completely inside an Euler element and completely inside an airbag element. An algorithm is available that can compute these polpacks for any given, closed 3D faceted surface, and any 3D Euler domain.

Faces refer to two Euler elements, whereas polpacks refer to only one Euler element. For both faces and polpacks areas and normals are computed. So, for each Euler element the boundary consists of faces and polpacks.

In case of MSC.Dytran the finite volume method amounts to applying the conservation laws for mass, momentum, and energy to these 3D objects. In applying these laws, we encounter volume and surface integrals. The volume Integrals are straightforward to compute and the surface integrals are computed by running over the faces and polpacks. In the three conservation laws the surface integral on the left-hand side signifies transport of either, mass, momentum or energy through polpacks and faces. A contribution of a polpack or a face to these integrals is called a flux.

In the mass conservation law, the mass flux across a face gives

$$\begin{aligned}\Delta M_{DONOR} &= -\rho_{DONOR} |\vec{V} \cdot \vec{A}| \Delta t \\ \Delta M_{ACCEPTOR} &= +\rho_{DONOR} |\vec{V} \cdot \vec{A}| \Delta t\end{aligned}$$

Here M is the mass in the Euler element, V is the velocity vector, A denotes the area vector of the face, Δt is the timestep, DONOR denotes the element supplying mass and ACCEPTOR denotes the element receiving mass. In most cases the airbag surface is not permeable, and there will be no transport across the polpacks. However, in case the airbag surface is permeable, or has a hole, the flux equations will take that into account.

The momentum in an element can increase by either transport of momentum, or by pressure forces working on the polpacks and faces. The force contribution to the momentum increase is the integral $-\int_A p n_i dA \Delta t$. The force contribution of a face to the momentum increase of the element left to the face and right to the face reads:

$$\Delta \vec{P}_{Left} = -p^{Face} \vec{A} \Delta t$$

$$\Delta \vec{P}_{Right} = p^{Face} \vec{A} \Delta t$$

Here P is the momentum of an element, A is the area vector pointing from the left element to the right element and p^{Face} is a weighted average of the pressure in the two elements that are on the left and right of the face. For polpacks the contribution is likewise, but now the pressure at the polpack is given by the pressure in the Euler element that contains the polpack. To conserve momentum the negative of this momentum contribution is put as a force on the airbag element that hosts the polpack. This is the way boundary conditions are imposed on the airbag surface.

In a similar way the energy equation is applied.

The procedure for advancing the Euler domain with one timestep is as follows:

1. Move the gridpoints of the airbag using the velocities on the gridpoints. Determine contact between airbag, dummy and other structural parts.
2. Using the new position of the airbag compute new polpacks. Using polpacks and faces compute for all Euler elements the volume of the portion that is inside the airbag.
3. Transport mass, momentum, and energy across all faces and permeable polpacks using the mass conservation law. The flux velocity is the average of the left and right Euler element velocity. In case no right Euler element is available, like for permeable polpacks, the flux velocity is determined from an inflow condition and in some cases the velocity in the Euler element.
4. For each Euler element compute density from the new mass and compute pressure from the equation of state using the new density.
5. Compute the force contribution to the momentum increase for all faces and polpacks. The transport contribution to the momentum increase has already been computed in step 3.
6. Advance the airbag with one timestep using the internal membrane forces and external forces from the Euler domain and compute new velocities on the grid points.
7. Compute a new stable timestep based on the meshsize, speed of sound and velocity. The stability criterion used is the CFL condition.

Recently, a dynamic Euler mesher has been introduced to the gas solver of MSC.Dytran. The user no longer needs to make Euler elements. Supplying Euler element sizes, and the airbag surface that will encapsulate the Euler mesh, suffices. Given the initial shape of the airbag and mesh sizes, the dynamic Euler mesher creates an Euler mesh that covers the inside and the direct vicinity of the initial airbag. After this the Euler mesh is checked every cycle whether it is still large enough to contain the whole airbag. If not, new Euler elements are created, and Euler elements that are no longer needed are removed.

ANALYSES WITH MULTIPLE EULER DOMAINS

Until recently analyses were limited to only one Euler domain. With the method described here it has become possible to analyze several airbags each with a separate Euler mesh. Furthermore, flow and other communication from one Euler mesh to the other takes place through a hole, which is common to both airbags. These holes are modeled as parts of the airbag surface. They can be fully porous or they can be partially porous. In this paper we only consider the fully porous case. The method presented does not put any restriction on the number of holes or the number of airbags. By connecting multiple airbags, a compartmented bag can be created.

In the Euler solver the stable timestep depends on the meshsize. Multiple Euler domains may have different meshsizes and therefore different stable timesteps. It is possible to update each Euler domain with a timestep that is specific for the domain, but this would require subcycling. For the analyses under consideration it is expected that the meshsizes of the various Euler domains are of the same order of magnitude. Therefore, it seems appropriate to evolve all Euler domains with the same timestep. When second-order accuracy is required one should consider employing subcycling.

To advance the Euler domains with one timestep the conservation laws are applied to all elements in the Euler domains using faces and polpacks. This is straightforward, except where the boundary of an element is connected to a surface modeling a hole. Although there are already polpacks for this hole surface, they are useless. They still refer to only one Euler element in one domain, and simply cannot provide the communication between two Euler domains. For enabling the communication between two Euler domains through a hole, polpacks are needed that refer to two Euler elements that are each in a different Euler domain. How this new type of polpack is constructed, will be described below. For now we assume that it can be done and we will call them flow polpacks. At these flow polpacks the flow between the Euler domains takes place.

The procedure to advance the Euler domains with one timestep consists of the following steps:

1. Move the gridpoints of the airbags using the velocity on the grid points.
2. For all Euler domains check whether the Euler mesh is still sufficiently large to contain the airbag. If not, extend the Euler mesh. If possible, delete empty Euler elements that have become too far away from the airbag surface.
3. Using the new position of the airbag surface, compute polpacks for all domains. Using polpacks and faces, compute for all elements the volume of that part of the element that is inside the airbag.
4. For all Euler Domains do transport across all faces and porous polpacks using the mass conservation law. The flux velocity is the average of the left and right Euler element velocity. Here, an Euler mesh is processed independently of the other meshes. Communication between the domains is postponed until step 5.
5. For all flow polpacks do transport. Here mass is going from an element in one domain to an element in the other domain.
6. For all Euler elements compute density from the new mass and compute pressure from the equation of state using the new density.

7. Compute the force contribution from the regular polpacks and faces to the momentum increase in the elements. The pressure at the face is a weighted average of the left and right Euler element. Pressures in the Euler elements that have polpacks give rise to forces on the airbag.
8. For all flow polpacks calculate the force contribution to the momentum increase. Here contributions from the flow polpacks are going into two Euler domains. The procedure is exactly the same as for faces. At regular polpacks the mass in the Euler element exerts a force on the airbag element, but at fully porous flow polpacks no forces are exerted on the airbag.
9. Advance the airbag with one timestep using internal membrane forces and external forces from the Euler domain. Compute new velocities for the grid points.
10. Based on the CFL condition, compute a new timestep using the meshsize, speed of sound and velocity. Every element in every domain yields a largest stable timestep. The timestep that will be used to advance all the domains to the next step is the smallest of these timesteps.

Now we describe the construction of flow polpacks and step 5 in more detail.

CONSTRUCTION OF FLOW POLPACKS

The current procedure for creating polpacks in MSC.Dytran takes as input a closed surface and an Euler mesh. The surface is defined by means of triangular and quadrilateral facets. The output consists of polpacks that refer to the Euler mesh that was passed as input. For every polpack we know the surface element from which it was derived and we know the Euler element it is located in.

In the vicinity of the hole there are two meshes that have a partial overlap and polpacks should refer to both one element in the first mesh as well as to one element in the second mesh. Furthermore, each flow polpack should fit completely in an element of the first mesh and completely in an element of the second mesh.

Here the problem arises that the polpack creation procedure can only handle one Euler domain. A good way to solve this problem is to create a third auxiliary mesh that is called the hole mesh. First we determine the largest box that is inside both Euler domains. This box represents the overlap of the two meshes. Putting the two meshes on top of each other, and ignoring all elements that are outside the overlap box, gives the hole mesh. This procedure is illustrated in Figure 2. By construction, every element in the hole mesh is completely in an element of the first mesh, and completely in a element of the second mesh. Also the surface of the hole is completely in this hole mesh, because the surface of the hole is in both Euler domains, and therefore in their overlap.

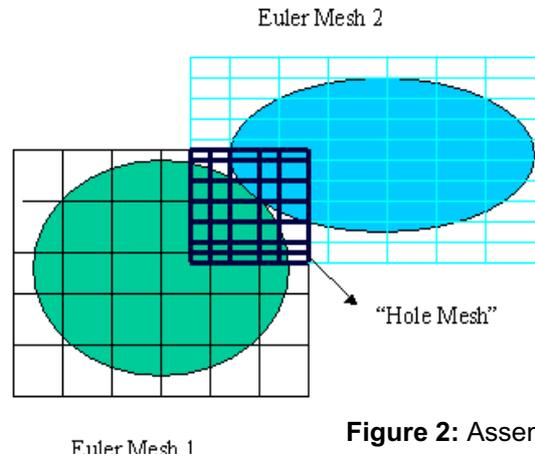


Figure 2: Assembling the hole mesh

To create the flow polpacks we take the following steps:

1. Create all polpacks for the individual Euler domains as explained in the section "Analyses using only one Euler domain".
2. Assemble the hole mesh from the Euler meshes. For every Euler element in the hole mesh we compute in which element of the first domain and in which element of second domain it is located.
3. Put the hole mesh plus the airbag surface into the polpacks creation procedure. Only that part of the airbag that is inside the hole mesh is processed. The resulting polpacks are called hole polpacks. They are not flow polpacks yet, because they still refer to one mesh, that is the hole mesh.
4. This new set of polpacks also contains polpacks that are not part of a hole. Therefore it is determined which ones are part of a hole surface.
5. In the list of polpacks created in step 1 for the individual Euler domains, delete all polpacks that are part of the hole and that still refer to one Euler domain.
6. We have obtained hole polpacks that refer to elements in the hole mesh. Now we can make flow polpacks that refer to two elements in the two domains using the information obtained in step 2.

Note: It is not needed to make a new hole mesh every cycle. Only when grid-points move outside the hole mesh, a new hole mesh is created. Also a new mesh is created when the dynamic Euler mesher changes one of the Euler domains.

FLUXING ACROSS THE HOLE POLPACKS

For Fluxing across flow polpacks two methods are available. The first method, the most general, is called the velocity method and is effective for holes of all sizes. The other method, the pressure method applies a theoretical 1D solution and applies to small holes only.

To obtain a second-order solution, a Riemann problem has to be solved at each flow polpack. This second-order solution will be implemented in the near future for the Roe solver of MSC.Dytran. It will not be described further in this paper.

THE VELOCITY METHOD

Given a flow polpack the transport of mass across the hole is given by

$$\vec{V}_{TRANS} = \frac{1}{2}(\vec{V}_{ELT1} + \vec{V}_{ELT2}) - \vec{V}_{AIRBAG}$$

$$\Delta M = \rho_{DONOR} \vec{V}_{TRANS} \cdot \vec{n}_{pp} Area_{pp} \Delta t$$

Here, V_{ELT1} is the velocity vector in the element in the first Euler domain, V_{ELT2} is the velocity vector in the second Euler domain, pp stands for polpack and ρ_{DONOR} is the density of the Euler element that is supplying mass. In the same way momentum and energy is transported

THE PRESSURE METHOD

Mass flux is computed using a theoretical tube calculation that models the flow situation in the vicinity of the hole. The diameter of this tube is minimal at the middle and diverges away from the middle. The middle section of the tube models the hole and the left and right diverging exits model the vicinity of the hole in the left and right airbag. The mass flux only depends on the states at the entrance and exit of the tube. Applying Bernoulli's equation gives

$$\frac{1}{2}u^2 + \frac{\gamma}{\gamma-1} \frac{p}{\rho} = \frac{1}{2}u_{ex}^2 + \frac{\gamma}{\gamma-1} \frac{p_{ex}}{\rho_{ex}}$$

Here the subscript ex denotes quantities at the throat of the tube and u, p and ρ denote the quantities inside that part of the tube that is supplying mass. This gives

$$u_{ex}^2 = \frac{2\gamma}{\gamma-1} \left(\frac{p}{\rho} - \frac{p_{ex}}{\rho_{ex}} \right)$$

The rate of the mass flow and the transported mass are respectively given by

$$\dot{m}_{ex} = \rho_{ex} A u_{ex} \text{ and } \Delta m \text{ is } \Delta t * \dot{m}_{ex}.$$

CHANGING MESH SIZES DURING THE ANALYSIS

In the first stages of Airbag deployment and expansion, the flow is highly non-uniform, and rather small Euler Elements are required. As the expansion goes on, the pressure becomes more uniform, and the small Euler elements are no longer needed. It is then efficient to coarsen the Euler mesh during the simulation. In MSC.Dytran a method has been implemented for doing this. Furthermore the larger elements leads to larger timesteps.

At the beginning of any timestep it is possible to map an Euler domain to a coarse or a fine domain. This mapping conserves mass, momentum and internal energy. The old mesh is deleted and replaced by the newly mapped mesh. For determining when a mesh should be coarsened, no automatic procedures are available yet, and the user has to specify the times, at which the mesh is coarsened or refined.

The mapping procedure for mass works as follows. Consider an element of the old mesh. This element will have a number of intersections with elements of the new mesh. All mass of this old element is distributed evenly among the new elements that intersect the old element. It is easy to do this in a conservative fashion by comparing intersection volumes with the volume of the old element.

Denote the old element by *old* and denote an intersecting new element by *new*. Then the mass assigned to the element *new* that comes from *old* is

$$Mass(old) \frac{Vol(old \cap new)}{Vol(old)}$$

Where *Vol()* denotes the volume of an element. This assignment is conservative if the total mass on the new mesh that originates from element *old* equals the mass of element *old*. The total mass coming from element *old* is given by

$$Mass(old) \sum \frac{Vol(old \cap new)}{Vol(old)}$$

The summation runs over all new elements that intersect the old element. The intersection volumes $Vol(old \cap new)$ exactly make up the volume of the old element and therefore the summation equals the mass of the element *old*. This proves that the assignment is conservative.

The quantities momentum, internal energy are mapped in the same way. Velocities for the new elements are computed from momentum and mass.

INTRODUCTION: TESTS AND APPLICATIONS

We shall present results of two test problems and one application. In the first two problems the analyses can very well be done with only one domain, but to test the multi domain code we shall also analyze the first two problems by dividing the problem into two parts. These parts then communicate through a porous hole. For validation we shall compare the one domain analysis with the multiple domain analysis. The last problem really needs the multi-domain approach and can not be run with a one domain analysis. The three problems are

1. 1D Shock tube problem
2. Two containers are linked with a hose. Initially one container is at a higher pressure than the other
3. A real life airbag simulation

1. 1D SHOCK TUBE PROBLEM

A tube is filled with gas from $x=0$ to $x=1$. The tube is initialized as shown in Table 1. The states in the left and the right part of the tube are constant but there is a jump at $x=0.5$. This jump will cause shock waves, expansion waves and a contact discontinuity. The application of MSC.Dytran for one domain has already been validated and we shall not repeat it here, since we only have to validate the multi-domain approach by comparing it to a one-domain analysis.

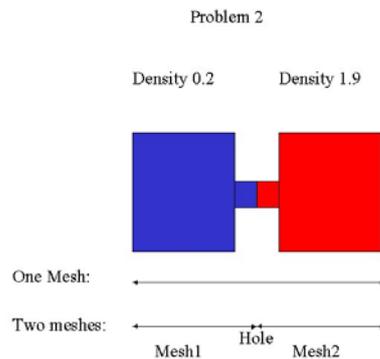


Figure 4: The two container problem

	x=0 to x=0.5	x=0.5 to x=1.0
Pressure	1	0.125
Velocity	0	0
Specific In- ternal Energy	2.5	2.0
Density	1	0.1

Table 1

This problem has been run with one domain and with two domains. In the run with one domain the Euler domain extends from $x=0$ to $x=1$. In the two domain run, the first domain covers $x=0$ to $x=0.5$ and the last domain covers $x=0.5$ to $x=1$. Both domains have been encapsulated by an airbag surface. To get an impression of how this is done see Figure 5 that illustrated meshing for problem 2. The hole that connects these airbags is located at $x=0.50001$. Figure 3 shows these results.

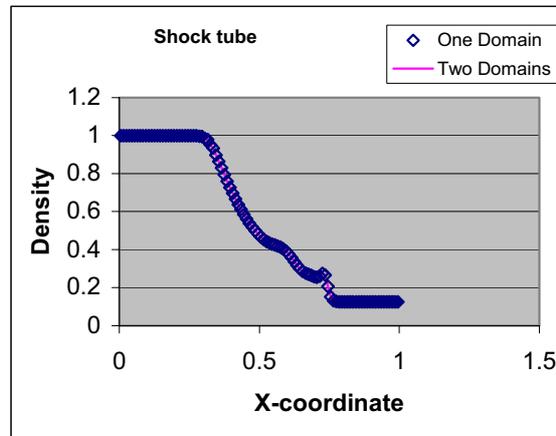


Figure 3: The shock tube

2. TWO CONTAINERS CONNECTED WITH A HOSE

The problem is described in Figure 4. Both containers have dimensions of 0.1 m by 0.1 m by 0.1 m. The hose has dimension 0.02 m by 0.02 m by 0.04 m. The left container is initialized with density 0.2 kg/m^3 and specific energy 400000 J/kg and the container on the right is initialized with a density of 1.9 kg/m^3 and with the same specific energy as the left one.

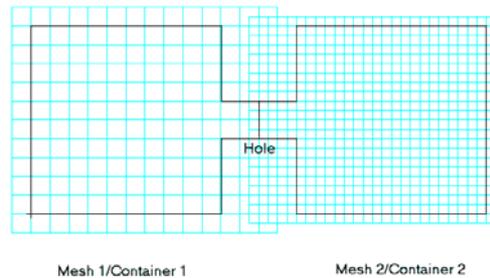


Figure 5: Meshing in problem 2

In the analysis gas will flow across the hose. Three analyses have been carried out for this problem

1. The whole structure is modeled as one rigid airbag and one mesh is used with mesh size 0.01 m
2. The structure is modeled as two rigid airbags that are connected via a hole as shown in Figure 5. Two Euler meshes are used with the same Euler mesh size 0.01 m.
3. The same as analysis 2, but now the meshsize of mesh 2 is 0.005 m and the meshsize of mesh 1 is unchanged at 0.01 m.

In Figure 6 results are shown for the three analyses. Here the average density in the left part of hose is shown as a function of time. In analysis 2 and 3 mass will flow across the hole. As shown in Figure 6, analyses 1 and 2 give results that are nearly identical. For analysis 3 we see some differences that are expected but moderate.

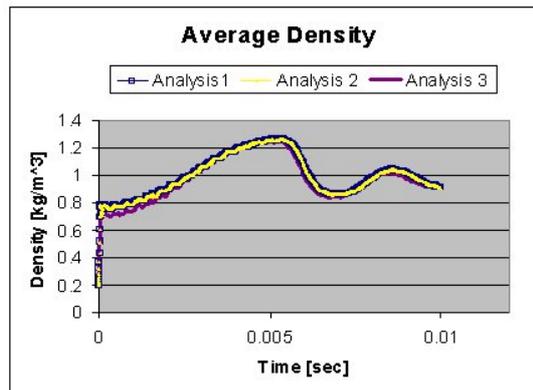


Figure 6: Results for the two-container problem

3. THE AIRBAG SIMULATION

The airbag is generic and consists of two compartments. The inner compartment is connected to the inflator and has a hole in each of the four corners. These holes connect the inner compartment with the outer compartment. The simulation time is 30 ms. Figures 7 and 8 show results for this deploying airbag halfway and at the end of the analysis, using the multi-domain approach. Here the velocity method is used for fluxing.

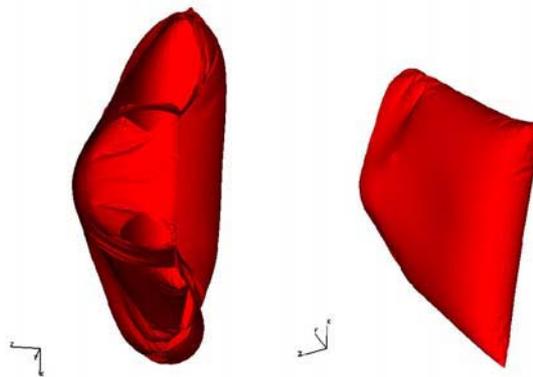


Figure 7: Airbag deployment at t=30ms

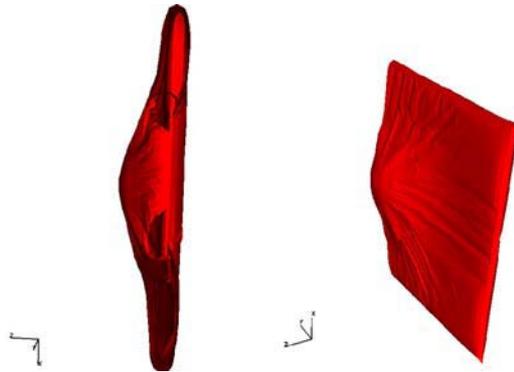


Figure 8: Airbag deployment at t=15ms

In figure 9 a comparison is shown at 20 ms between a simulation with one CFD airbag and a uniform gas bag on the left and the multi-domain CFD approach as shown on the right. The jet is clearly more pronounced in the multi-domain approach.

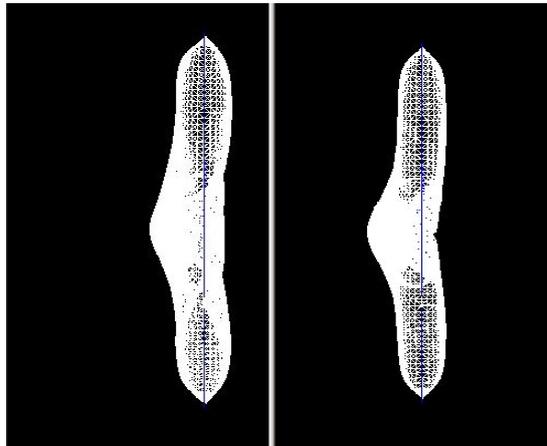


Figure 9: Comparison between two types of airbag simulations

Test runs reveal that growing mesh sizes have a limited impact on the simulation time for this problem. Performance increases were less than 5 %. For this problem the airbag determines the timestep. Also the CPU time needed for the airbag finite element solver and contact is significant compared to the Euler solver.

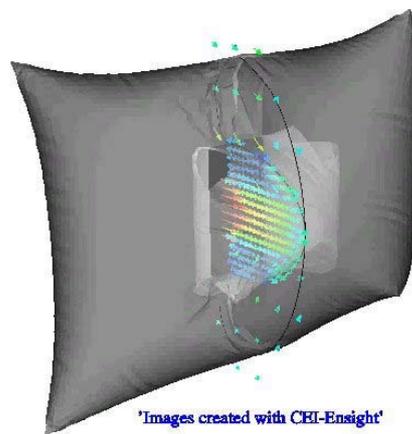
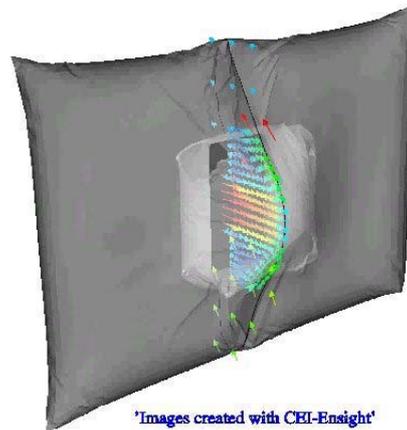
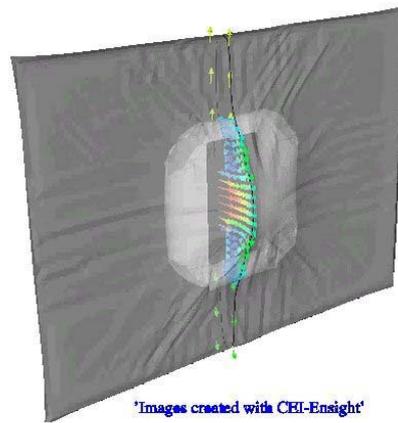
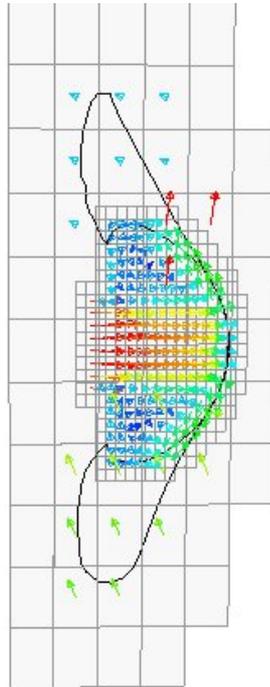
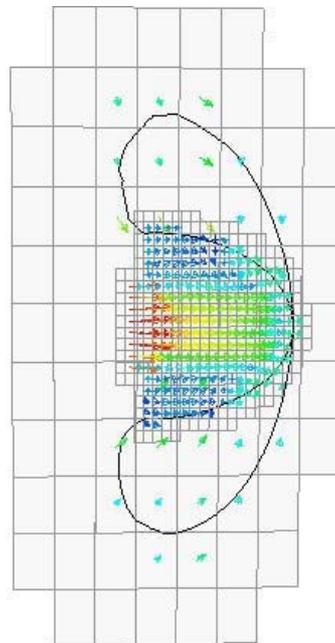


Figure 10: Airbag deployment at t=10ms, t=22ms and t=30ms



'Images created with CEI-Enight'



'Images treated with CEI-Enight'

Figure 11: The Euler mesh in airbag deployment for t=22ms and t=30ms

CONCLUSIONS AND FUTURE WORK

The test problems in this paper show that the method presented in this paper is a promising method for simulating multi-compartment airbags. Further testing with real life airbags and comparison with experiments is required. This will be carried out in the near future.

Also the implementation of the method in the second-order Roe solver will be considered. If so, subcycling will be implemented to retain second-order accuracy in time.

ACKNOWLEDGEMENTS

The author would like to thank C.J.L. Florie for his helpful suggestions and comments.

Figures 7-8 and 10-11 are made with courtesy of CEI-Ensignt.

REFERENCES

1. Koichi Kamiji, Nobuyuki Kawamaru, *Study of Airbag Interference with Out of Position Occupant by the computer simulation*, Honda R&D Co., LTD., Japan, Paper Number 374
2. A.J. Buijk, *Significance of gasdynamics in airbag simulations*, 26th ISATA, Aachen, Germany, September, 1993
3. A link to the website of MSC.Dytran can be found at the following URL: <http://www.mscsoftware.com>
4. C.Hirsch, *Numerical Computation of Internal and External Flows*, Vol 1 and 2, John Wiley&Sons, 1990
5. R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002

