THE DEVELOPMENT OF A 3D-NAVIER-STOKES CODE FOR THE SIMULATION OF AN AIRBAG INFLATION

Sinz Wolfgang[1], Steffan Hermann[1]

[1]Graz University of Technology, Vehicle Safety Institute,
Member of the Frank Stronach Institute, Inffeldgasse 11/II, 8010 Graz, Austria

wolfgang.sinz@tugraz.at (Sinz Wolfgang)

Abstract

In the field of vehicle development, numerical simulations are essential due to the complexity and the high costs of full scale crash tests. Hence, numerical methods are used for the development and optimization of restraint systems (belt pretensioner, airbags, etc.). The numerical simulation of an airbag inflation is very time consuming because the volume enclosed by the airbag is dynamic and such complex surfaces are generated. The aim of this work is to develop a robust, stable and fast algorithm for handling the fluid structure interaction problem in airbag inflation. The mechanical structure problem is solved with commercially available finite element software. A three dimensional Navier-Stokes-Code for compressible fluids is developed to solve the fluid mechanic problem. For this purpose, an explicit TVD upwind method by Roe is implemented. For reasons of stability, a fixed rectangular grid is used. The outer contour of this grid adjusts automatically to the surface of the airbag during the calculation. The contour adjustment is achieved by blocking off the outer cells, which can be switched on or off as the airbag surface passes by. To provide a conservative system during the contour adjustment, a special method for the handling of the boundary was developed. For the validation of the developed method, analytically solvable examples were used. These analytically solved solutions were compared with the numerically calculated solutions. To verify the suitability of the developed method, an airbag inflation was simulated.

Keywords: airbag, fluid-structure interaction, fixed rectangular grid, moving boundary, 3D Navier-Stokes-code.

Presenting Author's biography

Wolfgang Sinz was born on the 28th December of 1976 in Bregenz (Austria). He studied mechanical engineering at the Graz University of Technology, graduating in June 2003 with honours. In working toward his diploma thesis at the Vehicle Safety Institute, he became acquainted with fluid structure interaction. Continuing his research in the field of fluid structure interaction, Wolfgang Sinz finished his PhD on the numerical simulation of airbags in February 2007 with honours. Currently he is working as an assistant lecturer at the Graz University of Technology.

1 Introduction

In the field of vehicle development, numerical simulations are essential due to the complexity and the high costs of full scale crash tests. Hence, numerical methods are used for the development and optimization of restraint systems (belt pretensioner, airbags, etc.). The numerical simulation of an airbag inflation is difficult because the airbag inflation is a complex surface-coupled-multifield-problem of fluid structure interaction. The fields which must be described are the gas flow (gas generator, gas flow inside the airbag, etc.) and the airbag inflation (inflation, strength, etc.). The computer applications used in the field of vehicle safety were originally pure finite element (FE) programs. To realize a fluid structure interaction, all established numerical finite element programs which are normally used to improve the vehicle structure have a CFD solver implemented. In nearly all programs, the two fields (CFD, FE) are solved alternately and mechanical properties of the airbag surface are treated with common FEM. Programs differ with respect to the description of the gas flow. In the following section, the most commonly used methods for the numerical simulation of airbag inflations are briefly described.

1.1 Uniform Pressure and Temperature Model

This method is implemented in nearly all programs (Madymo, LS-Dyna, Pam Crash, etc.). The gas in the airbag chamber, which is composed of membrane elements, is treated as an ideal gas. Inside the chamber the state values pressure, density and temperature are constant. The airbag inflation itself is modelled as an expanding volume, where mass can exhaust or inflow within the volume. Many analytic models for gas generators are implemented in order to model the influence of the inflow jet. In every time step, the gas volume which is enclosed by the airbag membrane is determined. The energy equation is solved with the pressure of the previous time step (t_{n-1}) in order to determine the temperature. So the pressure of the actual time step (t_n) can be solved with the gas state equation. Due to the fact that the state values are constant inside the airbag, the uniform pressure method is unsuitable for the simulation of out of position (OOP) situations. The assumption that the values are constant has only a marginal influence on the overall shape of the airbag at the end of the inflation. Since the Navier-Stokes equations do not have to be solved, the uniform pressure model saves a good deal of CPU time. Therefore, this method is often used for the optimization of different parameters (e.g. fire time of the airbag).

1.2 Finite Pointset Method (FPM)

This method is implemented in Pam Crash starting from the version 2004 [1]. The FPM method was originally developed by Kuhnert [2] and is a grid free method which is based on the Lagrange approach. The values which describe the gas flow (velocity, density and temperature) are stored on freely positioned particles. These particles move with the local flow velocity. Due to the fact that the FPM needs no grid, the method has advantages in terms of grid generation and grid adaptation. Therefore the FPM method is suitable for applications in which the geometry of the flow area changes strongly over time. The particles are generated automatically and the gradients of the flow are determined by the interpolation of surrounding particles. The radius (smoothing length [3]) which defines the influence area is the most important parameter of the finite pointset method. The smaller the smoothing length, the more particles are required. Due to weighting the neighbours, which are close to the considered particle, are taken more into account. A lot of integration points (smoothing length of about 15mm [4]) are required for realistic numerical simulations, thus the calculation time is considerably greater.

1.3 Arbitrary Lagrangian Euler Method (ALE)

Another possibility to capture the gas dynamic effects in a more precise way (compared to the uniform pressure method) is the ALE method [5]. For example, LSTC implemented this method in their FE code (LS-Dyna). In this method, multi-field problems can be solved in a mathematically elegant manner. The method is based on a combination of the Euler and Lagrangian approach, and can be divided into two phases. In the first phase (Lagrange), the grid can be deformed if needed. The movement of the grid is based on the airbag motion. Therefore the ALE method is able to depict the surface of the airbag precisely. In the second phase, the flow field is frozen and a mesh smoothing is performed. Afterwards the solution of the deformed mesh is assigned to the smoothed grid. Hence the simulation procedure is a three step approach [6]:

a. Lagrangian step
b. Mesh smoothing
c. Remapping the solution to the smooth mesh – advection step

The ALE method is very CPU-intensive because the time step can’t be arbitrarily large. This is due the fact that the mesh isn’t deformed much between two time steps. Additionally, the cost (CPU time) of an advection step is typically two to five times the cost of the Lagrangian time step. [6].

1.4 Selection of the numerical method

Even today, the numerical simulation of an airbag-deployment is time-consuming so it is often difficult to optimize the airbag systems within the demanded time frame. Therefore the aim of the work was to prove that the calculation time can be reduced with the help of a specially developed boundary condition, which also should be as numerically stable as possible. The two-field problems are solved alternately. The results computed by the two independent programs are exchanged through an interface.

- **Structural mechanic problem**
  Commercial available FE-program LS-Dyna

- **Fluid mechanic problem**
  Developed CFD program “Bagjet”

The developed research CFD code is a three dimensional Euler-Code for compressible fluids. To solve the Euler equation an explicit TVD (Total Variation Diminishing) upwind method by Roe [7], which is a Godunov type method [8], is implemented. To get a fast and stable algorithm a fixed rectangular grid is used. The outer contour of the grid adjusts automatically to the surface of the airbag during the calculation. For the time integration two explicit methods are implemented. (First order explicit method, 4-Step-Runge-Kutta-Method [9]).

1.5 Advantages and Disadvantages

The differences between the currently used methods used to determine the fluid-structure interaction of airbag inflation are very great. It is not possible to give a generally valid conclusion regarding the pros and cons. Therefore, only the properties of the developed method are described in the table below.

Tab. 1 Advantages/Disadvantages

Advantages	Disadvantages
Cartesian grid: No transformation of the flux vector to a curvilinear grid required	The boundary faces are less accurate (“stepped”). The emerging error can be reduce with AMR
Unstructured grid: An automatic mesh refinement (AMR) can be easily implemented	Unstructured grid: Due to additional search algorithm (e.g. cells neighbouring), additional CPU time is required
Switching only on/off total cells: Stable algorithm (no grid deformation, no complicate cell cuts unlike by the Cut-Cell method [10])	

In the next chapters, the numerical basis and the principles for the development of the method are briefly described.

2 Description of flows

The flow is characterized by the motion of fluid particles. Primarily two methods are used to describe the fluid’s motion, the Euler approach and the Lagrangian approach. The methods differ with respect to the “kind” of observer. The ALE method is a combination of both methods. For a graphic interpretation of the three approaches refer to Fig. 1.

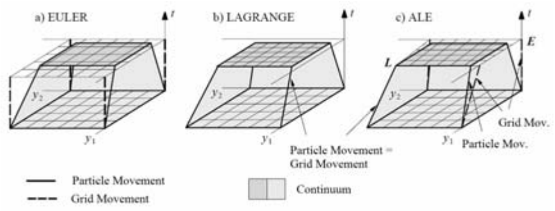

Fig. 1 Lagrangian, Euler and ALE approach [11]

The Lagrangian and the ALE approach aren’t specifically described in this chapter. The sub-chapter 2.1 focuses only on the Euler approach.

2.1 Euler Approach

In contrast to the Lagrangian approach, the observer is fixed in space. Properties of the fluid particles coming by are monitored by the observer. Metaphorically speaking, the observer stands on a bridge watching the particles flowing under the bridge. Due to the fixed nature of the observer, large particle movements can be treated unproblematically as compared to grid-based Lagrangian methods [12]. To describe time-dependent areas, an additional data structure is needed, which is described in chapter 4. A further disadvantage is the existence of the convective term in the balance equation, caused by the relative movement between the particle and the observer. The convective term is difficult to handle numerically. Nevertheless the Euler approach usually is the preferred method in fluid mechanics.

With the help of the different approaches it is possible to describe the flow mathematically. These equations consist of a system of coupled partial differential equations, which are often solved numerically. The used numerical method is described in the next chapter.

3 Used numerical algorithm

The base equations (continuity-, momentum- and energy equation) can be easily arranged in a flux vector form. The mass forces are neglected as they influence the airbag inflation only marginally.

$$\int_V \frac{\partial \mathbf{Q}}{\partial t} dV + \int_S \left(\mathbf{E} - \mathbf{E}_v\right) dS = 0 \qquad (1)$$

The flux vector **E** is called the Euler flux or convective flux and contains all the rates needed to describe an inviscid flow. The friction term and the heat flow are summarized in the diffusion flux $\mathbf{E}_v$.

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho \vec{w} \\ e \end{bmatrix} \qquad (2)$$

$$\mathbf{E} = \left(\vec{w} \cdot \vec{n}\right)\mathbf{Q} + \begin{bmatrix} 0 \\ p\vec{n} \\ p\left(\vec{w} \cdot \vec{n}\right) \end{bmatrix} \qquad (3)$$

$$\mathbf{E}_v = \begin{bmatrix} 0 \\ \vec{\tau}^{s} \\ \vec{\tau}^{s} \cdot \vec{w} + \dot{q}^{s} \end{bmatrix} \qquad (4)$$

Within this work only the Euler flux is considered. Due to the flux vector notation a friction term or a turbulence model can be implemented in a later stage - the required terms only have to be added in Eq. (1).

Eq. (1) is solved with a TVD upwind method by Roe. To get the initial values ($\mathbf{Q}^+$, $\mathbf{Q}^-$) at the cell border for the Riemann problem a MUSCL method (Monotonic Upstream Schemes for Conservations Laws) is used. A polynomial approach for upwind methods is used together with a MINMOD limiter function by Chakravarthy and Osher [13].

The decision for a time integration method is influenced by the fact that the fluid structure interaction problem is solved alternately. Therefore the time step of both solvers should be equivalent. A frequent value for a time step in FEM-vehicle crash calculations is 1.0E-06 seconds. Thus it is possible to calculate the Courant-Friedrich-Levy number (CFL) [14] for a typical grid (x_{Grid}=1.0E-02m) used in numerical airbag inflation calculations. Assuming the flow velocity of the gas amounting to 100m/s and the gas having the properties of air (R_{Gas}=287J/kgK, T=293K, κ=1.4) at ambient temperature the CFL number has approximately the value 0.5. Thus the stability criteria for explicit methods (CFL<1) is satisfied. Therefore two explicit one-step methods are embedded:

- Explicit Euler method
- Explicit Four-Step-Runge-Kutta method

4 Flows in time-dependent areas

It is obvious that during an airbag inflation, the area where the Navier-Stokes equations have to be solved changes greatly over time, so the numerical solution method is required to integrate time-dependent areas. How boundary conditions of a time-dependent area are handled depends largely on the description of the flow (refer to chapter 2). After a brief overview of existing procedures for the treatment of moving boundaries, the procedure used in this work is presented.

4.1 Lagrangian methods

The grid-free Lagrangian methods are very suitable for handling time-dependent areas – even if deformations are large. The boundary is very well approximated given that enough particles are at the edge. In addition, it is easy to apply the boundary conditions to these particles. These advantages of grid-free procedures bring with them a high cost of computation since a collision test with geometry is required for every single particle [15]. Grid-based Lagrangian methods are sensitive to large particle movement in the flow. In general, the movement of the particles at the boundary does not cause difficulty, rather the flow within the flow area (vortex, etc.). This is caused by the fact that grid-based Lagrange methods cannot handle large grid deformations. The only way out is to remesh the area. A remeshing of the area is not required, given that grid-free methods are used. If the particle density is too low in certain regions, additional particles are inserted. [16]

4.2 Arbitrary Lagrangian Euler methods

ALE methods combine the advantages of the Euler and Lagrangian method. At the boundary, the ALE method behaves like the Lagrange approach therefore the boundary can be approximated consistently and accurately. The grid points inside the area can also move, but this movement is not coupled with the local flow velocity. The nodes can move freely to avoid large grid deformations. A disadvantage of such approaches is the disproportionate increase in the numerical calculation effort when borders are heavily deformed. Additionally, areas growing together are difficult to handle using the ALE method [11].

4.3 Euler methods

Euler methods use fixed grids for treating time-dependent areas. The real calculation area is embedded in the base grid as is illustrated in the Fig. 2.

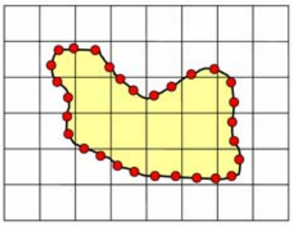

Fig. 2 Surface Marker Method

So an additional data structure is needed. The classic Euler method needs to be extended in order to solve problems with moving boundaries. The grid deformation is apparently no problem, but it is difficult to apply the boundary conditions accurately.

Listed below are the established Euler methods to handle time-dependent areas. In principle, the methods can be categorized in two groups [17]

a. Interface tracking methods

The motion of the boundary is traced directly

- Surface Marker Method
- Level Set Method

b. Interface capturing methods

The boundary is reconstructed

- Marker and Cell Method
- Volume of Fluid Method
- Cut Cell Method

4.4 Method Used

The decision as to which method is used to handle flexible geometry depends on the underlying problem. An essential aspect is the manner in which the boundaries are approximated by the different methods. Another point is the stability of the method, which is also influenced by the moving boundary. For this reason, the Euler method is chosen. The shape of the boundary is defined by an external FE program (LS-Dyna). In the FE calculation, the airbag membrane is defined by membrane elements.

Methods differ with respect to the accuracy of the intersecting cuts between the membrane elements and the grid of the CFD program:

a. The cuts are **calculated exactly** for example as shown by Aftosmis et al. [18] (e.g. Cut Cell Methods)
b. The exact course of the cut is **replaced by a simplified course** of the cut. (Simple Line Interface Calculation etc.)
c. Similar to the Marker and Cell Method **complete cells are switched on or off**.

For the decision as to which method is to be used, the following arguments have to be considered. The calculation of the cuts can be very difficult, because very complicated cuts can occur. Also it may be difficult to assess the membership of the areas (inner, outer) if complicated cuts appear. Another point is the higher calculation effort for the calculation of the required cuts, therefore a method is chosen which allows the switching on or off of complete cells

To realize this assertion, all cells exhibit, in addition to the base properties (density, velocity, etc.), additional properties (cell types: inflow cells, airbag obstacle cells, ghost cells, fluid cells, etc.) which are defined by the position of the airbag membrane. The current position of the airbag membrane is checked at every time step, hence the cell type can be changed during the calculation.

By definition, a cell exhibits the cell type “airbag obstacle cell” if the airbag membrane cuts a cell. Therefore it is not possible to suspend an airbag fold inside a cell, as is illustrated in Fig. 3

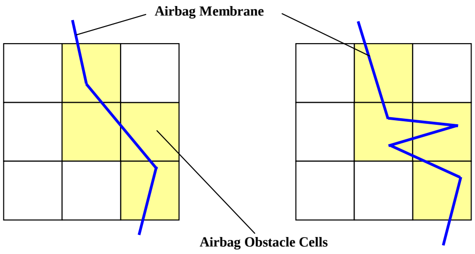

Fig. 3 Cell type assignment: airbag obstacle cell

The error emerging from this assumption can be reduced by an adaptive mesh refinement.

5 Boundary conditions

The boundary conditions are essential for the accurate solution of the differential equation. In this work the boundary conditions are applied on “ghost cells”. These ghost cells are attached to the calculation area and the values of the variables inside these cells are calculated in respect to the required boundary condition, as is shown in the Fig. 4

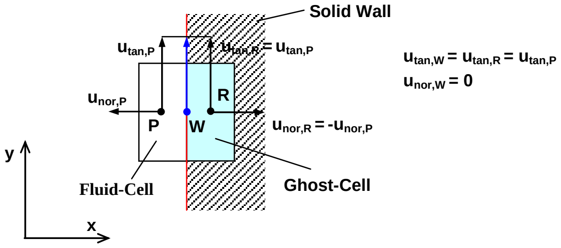

Fig. 4 Ghost Cell - wall boundary condition

In this work, an explicit method is used for the calculation of the ghost cell values. Furthermore, different extrapolation methods for the boundary condition can also be used. A linear extrapolation schema is implemented because the accuracy of the whole schema is not reduced if the schema of the boundary extrapolation is one order below the inner schema [8].

Also it is important that number of boundary conditions fits the number of differential equations [19]. If there are too many boundary conditions, the problem may not be solvable. However, if there are not enough boundary conditions several solutions are possible in which some solutions are not physically realistic. Boundary conditions should only be specified if information is conveyed from the boundary into the calculation area – otherwise the boundary condition has no effect. The propagation of the information is described by the characteristics of the differential equations. Therefore, the number of predetermined boundary conditions should be equal to the characteristics which appear in the calculation area. The other state values have to be extrapolated from the flow field to the corresponding ghost cells.

5.1 Boundary conditions at time-dependent areas

In this chapter the boundary conditions for time-dependent areas and inviscid flows are considered. Generally, the boundary conditions can be treated similar to the boundary conditions at fixed walls. For viscid flows, the same considerations are valid. The following points have to be considered for all boundary conditions (BC). Therefore these scores are also important for the BC of time-dependent areas:

a. **Number of predefined boundary conditions**
   For inviscid BC, only the tangential condition is needed.

b. **Preservation of the conservative values**
   For the system under consideration, the mass, momentum and energy have to be preserved during the calculation; e.g., due to the tangential condition at a fixed wall, the system is conserved because the velocity standard to the wall is zero. Therefore there is no flux inherent to the wall.

The treatment of the boundary conditions of time-dependent areas are strongly depending on the possibilities for describing flows (see also chapter 2). If the flow is described with a grid free Lagrangian method, the description of the BC of time-dependent areas causes no difficulties, thus the flow dynamic field information of the particles, which are on the edge of the area, accurately represents the boundary conditions. Therefore the state variables can be changed easily so that the tangential condition is fulfilled.

The ALE method is also convenient for the description of BC of time-dependent areas, because the border cells are adapted exactly to the form of the boundary [20]. Due to this fact, the number of cells is not changing within a time step – this is an advantage for the preservation of the conservative variables as be shown later.

The Euler method doesn‘t have the advantage that the grid can be deformed respectively as no grid is necessary. To take advantages of fixed grids for the treatment of time-dependent areas, special methods are developed [21, 22, et al.].

As described in chapter 4.3, the “real” calculation area is embedded in the base grid. Therefore the boundary is “moving” through the fixed base grid and limits the calculation area. With the help of the methods which are listed in chapter 4.3, it is possible to determine the position of the boundary for every time step.

Three different cases can be distinguished as being caused by the moving airbag membrane within a Cartesian grid.

It is important that for the treatment of all three cases, the conservative values are maintained. In the following the three cases, the difficulties involved are explained.

a. Cell type does not change

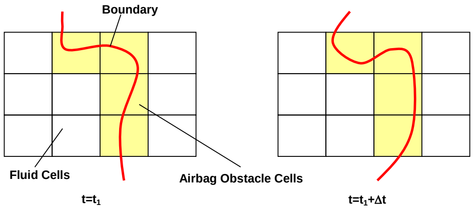

Fig. 5 Cell type does not change

During a time step, the fraction of the fluid volume at the cell volume and the surfaces which enclose the cell are changing. If the cells which are cut by the boundary are treated as border cells, the state values have to be changed so that the BC are satisfied and the calculation area is conservative.

b. Fluid Cell before – Airbag Obstacle Cell later

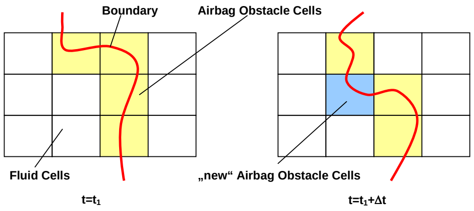

Fig. 6 Fluid cell before – airbag obstacle cell later

This is the case if the calculation area decreases. The new airbag obstacle cell is a fluid cell at time t_1 and therefore the cell contains mass, momentum and energy at this time. In the next time step ($t_{1+\Delta t}$), the cell changes its property from a fluid to an airbag obstacle cell. The conservative values which are contained in the new airbag obstacle cell have to be applied to the calculation area.

c. Airbag Obstacle Cell before – Fluid Cell later

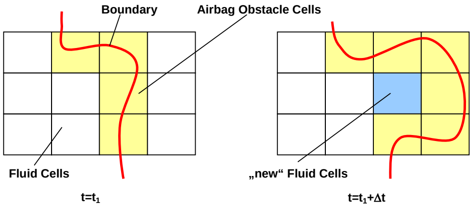

Fig. 7 Airbag obstacle cell before – fluid cell later

The third case is the inverse of the case two, wherein the calculation area enlarges. Therefore additional fluid cells have to be integrated in the system. A problem for the integration of this cell is that the new fluid cell has no time background and therefore no initial values. Therefore the cells have to be initialized – the initialization has to be done in a way that the system is conservative.

Methods for treatment of moving boundaries based on an Euler grid must be able to correctly handle these cases physically.

Two methods from the literature, which can handle moving boundaries in combination with an Euler grid, are described briefly below. There are a lot of variations respectively combinations of different methods to handle these situations.

Cell Merging [21, 23, et al.]
The principal idea behind the cell merging method is to combine cells at the beginning of a time step and separate these cells at the end of the time step.

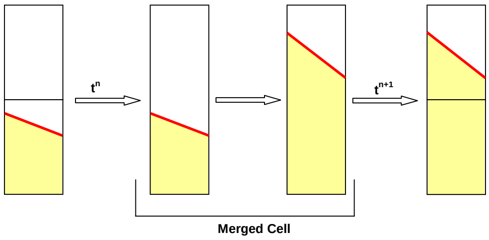

Fig. 8 Increasing area – cell merging method [24]

If the cell volume and the cell surface areas are known for every time step, it is possible to integrate cells in a conservative way. No initial values are needed due to the merged cells. If the area decreases, the system remains conservative because the mass in the border cells at the time step t_n is automatically in the border cell at the time step t_{n+1}.

Space Time [25]
In the space time method, the problem of the moving boundaries is not only considered in the space geometrically. Additionally, this approach considers the time dimension directly in combination with the geometrical space, thus in this method space-time areas are opened or closed during a time step.

To aid understanding, a one dimensional case is considered.

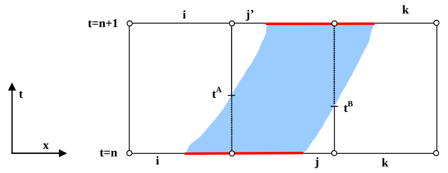

Fig. 9 Space time – one dimensional moving piston

The Fig. 9 shows a boundary movement (red line), were the cell j’ is newly opened and the cell j is closed. The flux through the space-time for the cell k consists of one part of the uncropped area at the right side (inflow into the cell k from time t^n until t^{n+1}) and two parts from the left side (inflow into the cell k from t^n until t^B and a wall boundary condition from time t^B until t^{n+1}). To calculate the balanced equation, the volumes of the cut cells have to be determined, and in addition the wall pressure and the space time geometry (blue area) have to be approximated. With the help of these approximations, moving boundaries within Euler grids can be handled. The space time methods are described more precisely by Murman et al. [25].

5.2 Developed boundary conditions for time depending areas within an Euler grid

In the chapter above, two methods which are capable of handling boundary conditions of time varying areas are described. In both procedures it is however necessary to determine the portion of the fluid volume of the cell volume and the flow-entering surfaces. The calculation effort to determine these values is high, especially for the three dimensional case. Another problem is the number of possible special cases.

For this reason, a simple procedure was developed. For the method developed, it is not necessary to determine the portion of the fluid volume of the cell volume and the flow-entering surfaces.

Only complete cells are switched on and off. For the boundary condition the tangential condition is used.

Predefining an airbag membrane velocity and turning cells respectively on and off results in a flux.

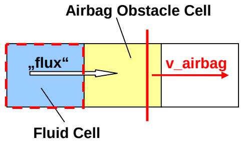

Fig. 10 Developed method – conservative

Consequently the calculation area (red dashed line) is not a conservative system. In order to prevent a non-conservative system, the flux across the boundary is determined and the conservative "deficit values" are stored in the corresponding border cells (airbag obstacle cells). These deficit values are applied to the system depending on the currently occurring cell conditions. Therefore the cell exhibits not only the state vector **Q**, but also additional values. These values are the conservative "deficit values": mass, momentum and energy. The dimension of the "deficit value vector" and the state vector are the same.

The "deficit values" are calculated with the help of the Euler flux.

$$\mathbf{E}_x = \begin{bmatrix} E_{x0} \\ E_{x1} \\ E_{x2} \\ E_{x3} \\ E_{x4} \end{bmatrix} = u \cdot \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} + \begin{bmatrix} 0 \\ p \\ 0 \\ 0 \\ up \end{bmatrix} \qquad (5)$$

$$e = \left[\frac{p}{\kappa - 1} + \rho \cdot \frac{u^2 + v^2 + w^2}{2} \right] \qquad (6)$$

At the end of the time step the Euler flux $\mathbf{E_x}$ is known. Eq. 5 is a linear system of equations and the velocity u can be calculated with Eq. 7.

$$u_{12} = -\frac{m}{2} \pm \sqrt{\left(\frac{m}{2}\right)^2 - q} \qquad (7)$$

With the coefficient:

$$m = -\frac{2 \cdot \kappa}{1 + \kappa} \cdot \frac{E_{x1}}{E_{x0}} \qquad (8)$$

$$q = -\frac{\kappa - 1}{\kappa + 1} \cdot \left(\frac{E_{x2}^2 + E_{x3}^2}{E_{x0}^2} - 2 \cdot \frac{E_{x4}}{E_{x0}} \right) \qquad (9)$$

For the calculation of the "deficit values" the calculated velocity (u_{12}) which diverges least from the airbag membrane velocity is taken.

With the help of the three cases described in sub-chapter 5.1, the developed method can be explained more precisely. In the following examples only the one dimensional case is summarized, but the same considerations apply to the three-dimensional case. Likewise, the principle of the developed method is only explained for the mass conservation. The same considerations are valid for the other conservative values (momentum and energy).

In the following figures the values m_{cell}, ρu_{cell}, ρv_{cell} , ρw_{cell} and e_{cell} are the conservative "deficit values". The airbag obstacle cells are marked in yellow and the fluid cells are marked in blue.

a. Cell type do not change

There are two sub-cases:

Increasing area

For the cell j the "outflow boundary condition" is applied if the area is increasing. Due to the direction of the characteristics only one value has to be predefined.

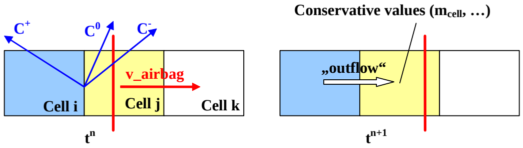

Fig. 11 BC: Case 1 – increasing area

With the help of Eq. 5 the flux of the conservative values across the cell i and the cell j are calculated and stored in the cell j as additional value.

$$m_{cell,j}^{t^{n+1}} = m_{cell,j}^{t^n} + \rho_{outflow,i} \cdot u_{outflow.i} \cdot A_x \cdot dt \qquad (10)$$

Decreasing area

In contrast to the increasing area the "inflow boundary condition" is applied to the cell j. Due to the direction of the characteristics two values have to be predefined. These are the airbag membrane velocity and the density of the neighbour cell. It is assumed that the density alteration is moderate in relation to the neighbour density.

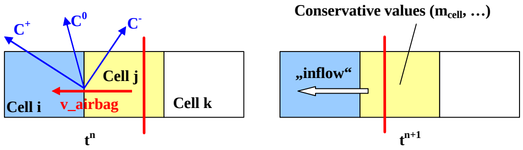

Fig. 12 BC: Case 1 – decreasing area

The flux is calculated across the cell boundary i-j and is subtracted from the conservative cell variables of the cell j.

b. Fluid Cell before – Airbag Obstacle Cell later

This case is treated as an "inflow boundary condition". The airbag membrane velocity and the density are predefined (compare with case 1) for the cell j. Again it is assumed that the density alteration is only moderate in relation to the neighbour density.

To get a conservative system, the remaining values of the cell k are added to the conservative value of the cell j.

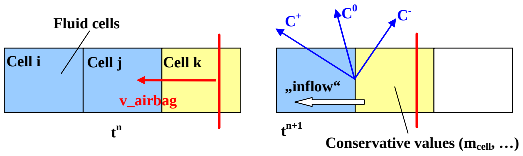

Fig. 13 BC: Case 2 – fluid before, airbag later

The values inside the cell j at the time step t^n are also added to the “deficit values”.

$$m^{t^{n+1}}_{cell,j} = m^{t^n}_{cell,j} + m^{t^n}_{cell,k} - \rho_{\inf low,j} \cdot u_{\inf low,j} \cdot A_x \cdot dt \qquad (11)$$

c. Airbag Obstacle Cell before – Fluid Cell later

In this case, additional fluid cells have to be integrated in the calculation area, and the “outflow boundary condition” has to be applied. Therefore only one value (airbag membrane velocity) has to be predetermined.

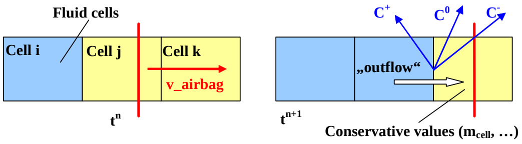

Fig. 14 BC: Case 3 – airbag before, fluid later

The stored deficit values are the initial values for the cell j. Therefore the following operational sequence results:

a. Time step t^n:
Conservative “deficit values” are stored in the cell j.
b. Cell j is now a fluid cell
Initial values for the density of the cell j at the time step t^{n+1}

$$\rho^{t^{n+1}}_{cell,j} = \frac{m^{t^n}_{cell,j}}{dx} \qquad (12)$$

c. Extrapolation of the density and predefine the airbag membrane velocity for the cell k.
d. Solve the time step t^{n+1}
e. The flux for the cell k is calculated and the deficit values are stored in the cell k

$$m^{t^{n+1}}_{cell,k} = \rho_{outflow,j} \cdot u_{outflow,j} \cdot A_x \cdot dt \qquad (13)$$

Special cases

It is easy to extend the basic one dimensional idea to the three dimensional case. For the multidimensional problems, a special case can occur. This special case is explained on the basis of a two dimensional calculation area. The increasing and decreasing areas are treated differently.

a. Increasing area

The green cell in the Fig. 15 has no initial values, because the green cell has no fluid cell as its direct neighbours. Therefore no conservative “deficit values” can be stored in such “isolated” cells.

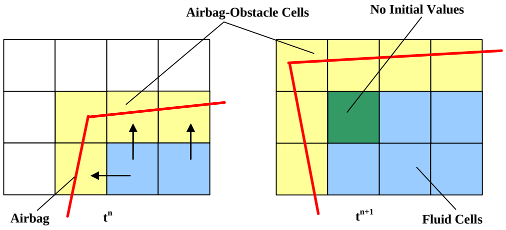

Fig. 15 Special case: no initial value

This problem can be solved by an intermediate time step (time step bisection). The green cell is still treated as a border cell. In the next “half” time step, the green cell gets the properties of a fluid cell.

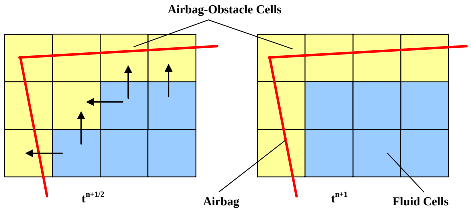

Fig. 16 Special case: bisection of the time step

Two intermediate time steps are required for the three dimensional case.

b. Decreasing area

For the cell configuration illustrated in Fig. 17, the green cell has “deficit values”, which cannot be applied back to the system because this cell also has no fluid cells as its direct neighbours.

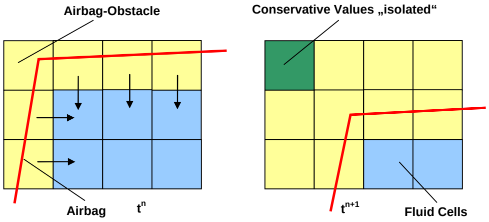

Fig. 17 Special case: isolated cell

For this special case, no intermediate time step has to be inserted. The deficit values of the green cell are distributed on the corresponding cells (airbag obstacle cells), so the direction of the airbag membrane velocity of the green cell is used as criterion for the distribution.

This procedure is shown in Fig. 18.

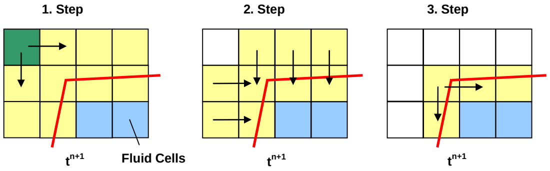

Fig. 18 Special case: division of deficit values

This procedure is very stable, because there are almost no additional special cases. In addition, it can handle the three cases which are described in chapter 5.1.

6 Fluid Structure interaction

To solve the fluid structure interaction of an airbag inflation, a weakly coupled and partitioned method is chosen. As was mentioned in chapter 1.4, both one field problems are solved alternately and the solutions of each programs is exchanged through a defined interface.

a. **Structural mechanic problem**
   FE-Program LS-Dyna

b. **Fluid mechanic problem**
   Developed CFD program "Bagjet"

The simulation procedure is shown in the Fig. 19. The CFD program controls the simulation and starts the FE-calculation at the required time.

Unfortunately the required interface in the FE-program LS-Dyna does not exist. For this reason it is necessary to restart LS-Dyna at each time step.

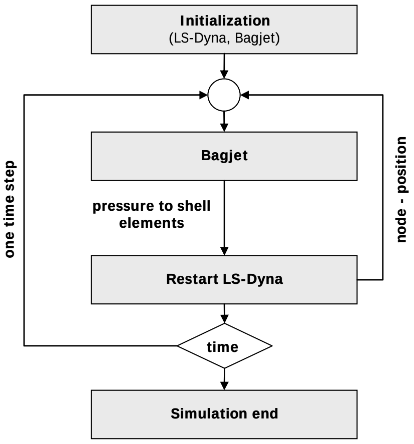

Fig. 19 Simulation procedure

The simulation procedure used needs two interfaces which are briefly described in the following sections.

6.1 Interface: FE-Program

For this work the airbag membrane is approximated by 4-point membrane elements. For each element, a corresponding pressure is determined by "Bagjet".

In a first step, the cell which contains the centre of gravity of a membrane element is determined. The pressure of the localized cell is distributed on the membrane element.

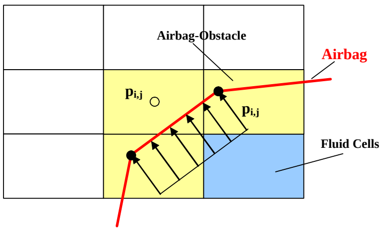

Fig. 20 Interface: FE-program

In a next step the decision is made as to which side of the membrane element the pressure acts. The direction of the element standard (initially pointing outwards) is compared with the position of the fluid neighbour cells.

6.2 Interface: CFD program

The developed CFD program requires the position of the airbag membrane. The positions of the membrane element nodes are read at every time step. Also it is necessary to determine a corresponding shell element velocity for the tangential boundary condition of the cells, which are intersected by the airbag membrane.

Two cases are distinguished for the determination of the corresponding membrane velocity.

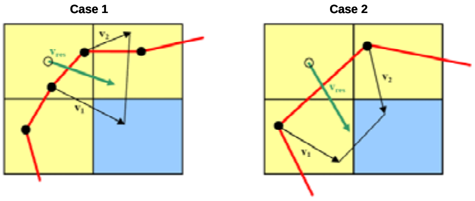

Fig. 21 Interface: CFD program

If nodes from the shell elements are inside a cell, the node velocities are added as vectors in order to determine a corresponding velocity. It could also be, as is shown in case 2, that no node is inside a cell. In this case, the velocity of all four shell element nodes which divide the cell are used to determine a corresponding velocity

7 Numerical results

The aim of this chapter is to validate the developed method with the help of different examples with known analytic results, so the solution can be compared easily. Additionally the conservativity of the developed method is proven. To show the performance of the method, an airbag inflation is also calculated.

In this paper, only three examples are chosen although more examples (Shock tube, Moving piston with free expansion…) were used for the validation of the new method.

7.1 Maintenance of the conservative values

In this chapter, the maintenance of conservative values is verified. The verification had been done using a defined movement of the boundary to get a time-dependent area. One and three dimensional examples are simulated.

Only one three dimensional case is mapped in the following. In this example, the special cases which are described in chapter 5.2 are simulated. The conservative laws of all values (mass, momentum and energy) are checked. As an example, the maintenance of the mass is shown.

For this example, a box-shaped calculation area is used where all sides have a defined velocity, so in respect to the sign of the velocity the calculation area can either increase or decrease.

As can be seen in the figure below, the box-shaped calculation area (blue cube) is embedded in the Euler grid of the CFD program.

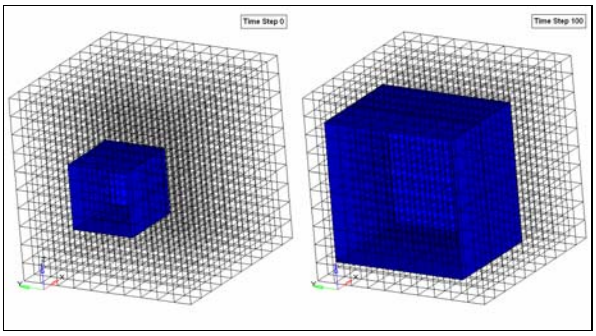

Fig. 22 3D moving boundary

Area extension (Expansion)
As was explained in section 5.2, two intermediate time steps are needed to get initial values for the "isolated" cells in the special cell configuration.

Tab. 2 shows the number of fluid cells versus time step respective of time.

Tab. 2 Number of fluid cells – expansion

Timestep	Time [s]	Δt [s]	Number of fluid cells
10	9.000E-05	1.000E-05	12
11	1.000E-04	1.000E-05	12
12	**1.100E-04**	**3.333E-06**	**12**
13	**1.133E-04**	**3.333E-06**	**40**
14	**1.167E-04**	**3.333E-06**	**60**
15	**1.200E-04**	1.000E-05	**64**
16	1.300E-04	1.000E-05	64

It can be seen that the number of the fluid cell does not increase "directly" from 12 to 64, as it should be a result of the special defined boundary motion. The system used two intermediate time steps where certain cells are treated still as "border cells".

In the Fig. 23, the mass balance is mapped. In this figure the principles of the developed method are obvious. The total mass is constant during the calculation.

The mass of the fluid cells decreases due to the flux "out" of the calculation area. The mass of the airbag obstacle cell increases in the same manner as the mass of the fluid cell decreases. If the motion of the boundary is large enough for new fluid cells to be integrated into the calculation area, the mass of the airbag obstacle cell is used as the initial mass.

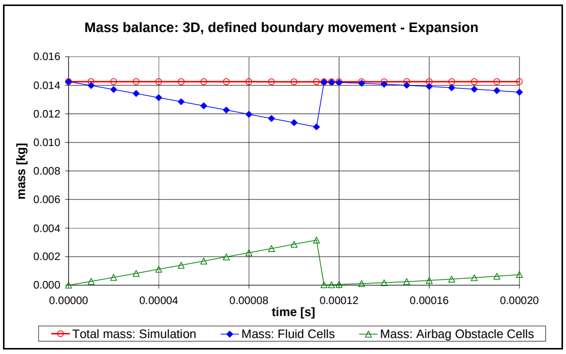

Fig. 23 Mass balance - expansion

Area reduction (compression)
No intermediate time-step is required for the compression (compare with sub-chapter 5.2). The conservative deficit values of the "isolated" cells (no fluid cell as neighbour) are distributed on the corresponding neighbour cells, taking the boundary velocity into consideration.

Tab. 3 Number of fluid cells – compression

Timestep	Time [s]	Δt [s]	Number of fluid cells
10	9.0E-05	1.0E-05	216
11	1.0E-04	1.0E-05	216
12	**1.1E-04**	**1.0E-05**	**216**
13	**1.2E-04**	**1.0E-05**	**80**
14	1.3E-04	1.0E-05	80
15	1.4E-04	1.0E-05	80
16	1.5E-04	1.0E-05	80

Due to the fact that no intermediate time step is required, the number of fluid cell decreases "directly" from 216 to 80 (see Tab.3).

In this case the total mass remains constant (see Fig 24).

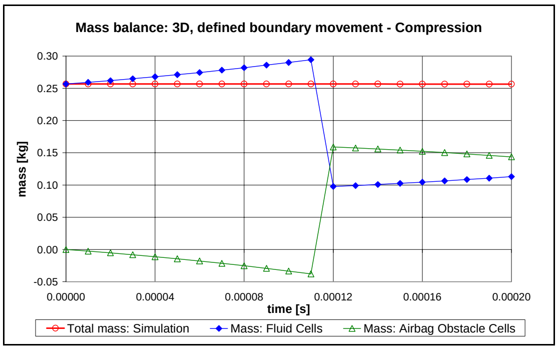

Fig. 24 Mass balance – compression

7.2 One dimensional spring mass system

A one dimensional problem is depicted in the Fig. 25. A frictionless piston moves inside an infinite tube. The piston may displace in the x-direction and is supported by a spring.

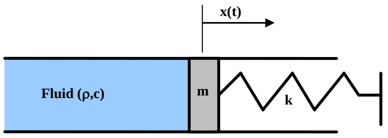

Fig. 25 1D spring mass system

The one dimensional spring-mass problem can be solved analytically [26].

The equation of motion for the piston is divided by its area to get an area independent consideration.

$$m \cdot \ddot{x} = p(t) - k \cdot (x - x_0) \qquad (14)$$

In this equation x(t) is the position of the piston, x_0 the equilibrium position, k the stiffness of the spring per piston area [N/m³], p_t the pressure and m the mass of the piston per piston area [kg/m²]. The current pressure can be solved with the simple unsteady wave equation [27]

$$p(t) = p_0 \cdot \left(1 + \frac{\kappa - 1}{2} \cdot \frac{\dot{x}}{c_0}\right)^{\frac{2\cdot\kappa}{\kappa-1}} \qquad (15)$$

Therefore the equation of motion can be written as:

$$\ddot{x} + \omega_0^{\,2} \cdot (x - x_0) = \frac{p_0}{m} \cdot \left(1 - \frac{\kappa - 1}{2} \cdot \frac{\dot{x}}{c_0}\right)^{\frac{2\cdot\kappa}{\kappa-1}} \qquad (16)$$

with the resonance frequency:

$$\omega_0^{\,2} = \frac{k}{m} \qquad (17)$$

One recognizes from the motion equation that the equation Eq. 16 represent a nonlinear oscillator with the following equilibrium position:

$$x = x_0 + \frac{p_0}{k} \qquad (18)$$

To make the equation dimensionless a characteristic length $L=p_0/k$ is introduced.

$$X'' + X' = (1 - \alpha \cdot X')^{\frac{2\cdot\kappa}{\kappa-1}} \qquad (19)$$

The coefficients of this equation are:

$$\alpha = \frac{\kappa - 1}{2} \cdot \frac{\omega_0 \cdot L}{c_0} \text{ and } X = \frac{x - x_0}{L} \qquad (20)$$

The derivative of Eq. 19 is taken with respect to $\tau = t\ \omega_0$ and describes the displacement of a free damped oscillator. Assuming $\alpha \ll 1$ and the starting conditions

$$x(0) = 0 \qquad \dot{x}(0) = 0$$

the following solution is obtained:

$$\frac{x}{L} = 1 - e^{-\frac{\kappa}{\kappa-1}\alpha\cdot\omega_0\cdot t} \cdot \left(\cos(\beta \cdot \omega_0 \cdot t) + \frac{\kappa}{\kappa - 1} \cdot \frac{\alpha}{\beta} \cdot \sin(\beta \cdot \omega_0 \cdot t)\right) \qquad (21)$$

$$\beta = \sqrt{1 - \left(\frac{\kappa}{\kappa - 1} \cdot \alpha\right)^2} \qquad (22)$$

This analytical solution can now be compared with the numerical solution.

For the numerical solution the piston (m=3 kg, A=1m²) is modelled using four rigid shell elements. The vertices of the piston are supported by a spring (k_n=k/4). The initial values of the CFD program are: p_0=6.0E+05Pa, ρ_0=4kg/m³, κ=1.4, T_{init}=293.15K.

In all simulated examples, the value of α was assumed to be much smaller than 1 (α<< 1.)

For the comparison between the numerical solution and the analytic solution, following parameters vary:

The **cell length** of the CFD simulation

Tab. 4 Variation of the grid width

	cell length	timestep	spring stiffness
	[m]	[s]	[N/m]
K1	**0.0025**	2.00E-06	9.00E+07
K2	**0.0030**	2.00E-06	9.00E+07
K3	**0.0040**	2.00E-06	9.00E+07

The **time step** of the CFD simulation

Tab. 5 Variation of the time step

	cell length	timestep	spring stiffness
	[m]	[s]	[N/m]
K4	0.0025	**2.00E-06**	9.00E+07
K5	0.0025	**3.00E-06**	9.00E+07
K6	0.0025	**4.00E-06**	9.00E+07

The **spring stiffness** of the CFD simulation

Tab. 6 Variation of the spring stiffness k

	cell length	timestep	spring stiffness
	[m]	[s]	[N/m]
K7	0.0025	2.00E-06	**8.00E+07**
K8	0.0025	2.00E-06	**9.00E+07**
K9	0.0025	2.00E-06	**1.00E+08**

In these tables the respective variation parameters are highlighted.

Results:

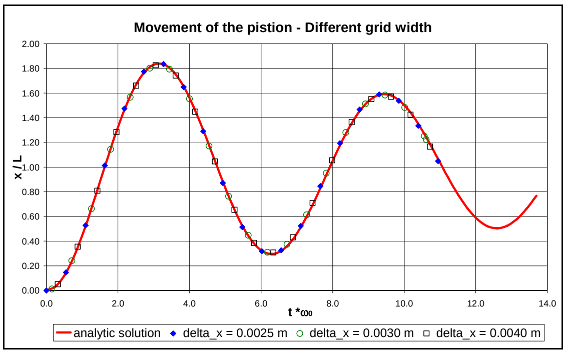

Fig. 26 1D piston movement – grid width

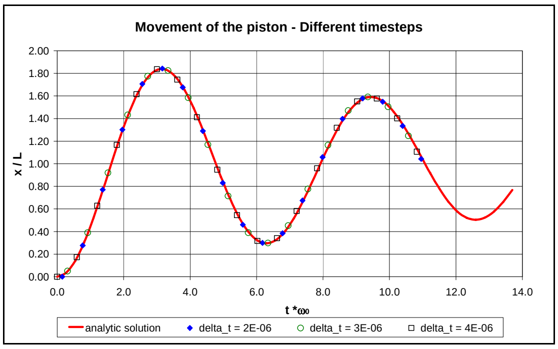

Fig. 27 1D piston movement – time step

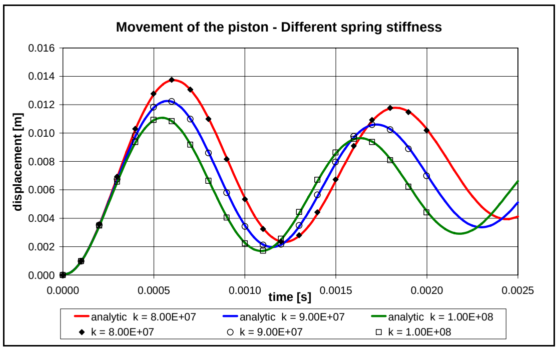

Fig. 28 1D piston movement – spring stiffness

For these configurations, the time step and the grid width has no influence on the displacement. Obviously the calculated resonance frequency and the amplitude correspond with the analytic solution.

7.3 Simulation of an airbag inflation

On the basis of this example, the principal capability of the developed method to simulate airbag inflation is checked.

At the beginning, the simulated airbag lies flat on a table. This is a common test set-up for the validation of airbag simulation models. In this simulation all respective geometric input parameters are assumed virtually for the following reasons:

a. number of Restarts

The required interface in the FE-program LS-Dyna does not exist. For this reason it is necessary to restart LS-Dyna at each time step. If for example the average airbag inflation time is about 30 ms and a common time step is used (1.0E-06 s), the program LS-Dyna has to be restarted 30,000 times. The initialization effort for the restarts is very high. This is explained later.

b. missing tank test date

In order to characterise the gas generator, a tank test has to be simulated. No appropriate tank test data were found.

Description of the simulation model

In the simulation model, as depicted in the Fig. 29, an airbag (diameter: 500 mm) is laid out flat on a table. A recess for the gas generator and the shooting channel is located at the table's centre.

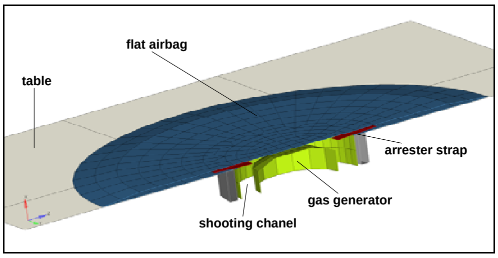

Fig. 29 Configuration of the simulation

A simple cylinder is used as a firing channel. The gas flow, which escapes radially from the gas generator, turns in a way that the gas flows axially into the airbag.

The airbag is held back during the inflation by means of four arrester straps, which are symmetrically arranged. Additionally it is assumed that the arrester straps do not represent an obstacle for the gas flow. Furthermore the gas generator, the shooting channel and the table are assumed to be rigid.

The gas flows radially over eight symmetrical holes (Area: 1,96cm²) horizontal from the gas generator in the shooting channel (diameter: 150 mm).

For the subsonic inflow boundary condition, it is sufficient that the mass flow, the inlet temperature, the inflow area and the inflow angle is given. If a supersonic inflow arises during the simulation, the inflow velocity is set equal to the speed of sound.

The airbag membrane is illustrated by 948 four point membrane elements (thickness: 0.1mm). The smallest arising edge length amounts to 8 mm. Usual values for an airbag membrane material amount to ρ=8.8E-07kg/mm³, E=300N/mm², ν=0.2. For this example, however, the following material properties are selected, in order to increase the minimum time step of the FE computation: ρ=9.0E-06kg/mm³, E=900N/mm², ν=0.3. The cell width of the CFD grid amounts to 17 mm in x-direction respectively, for 14mm in y- and z- direction (coordinate direction see Fig. 29). Altogether the flow grid consists of 28,577 cells. The FE (LS-Dyna) and CFD (Bagjet) time step are equivalent and amount for 6.00E-06 seconds.

Results

In the following figure the airbag inflation is depicted at the time t=0 ms, t= 2 ms t=4 ms and t=6ms.

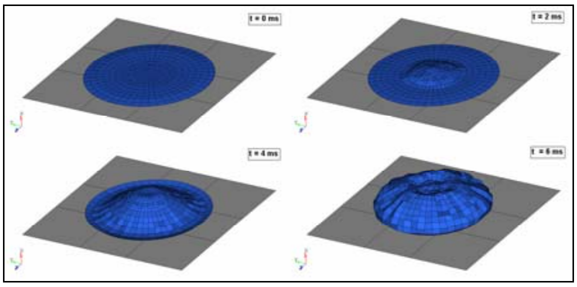

Fig. 30 Airbag inflation – FE mesh

In principal, the path of the motion of the simulation corresponds to a real airbag inflation. It is evident that the result is almost rotationally symmetrical despite use of a regular grid. Due to the fact that only 6ms are simulated, the airbag is not yet fully inflated as can be seen in Fig. 30.

In the Fig. 31, the allocation of the cell properties is represented during the simulation. The dark-blue cells represent the fluid cells. The Euler equations are solved only for these cells. The light blue cells however have the characteristics of airbag obstacle cells. The red cells are inflow, and the green cells are gas generator cells.

In this illustration, one can also easily recognize the functional aspect of the arrester straps (yellow "lines").

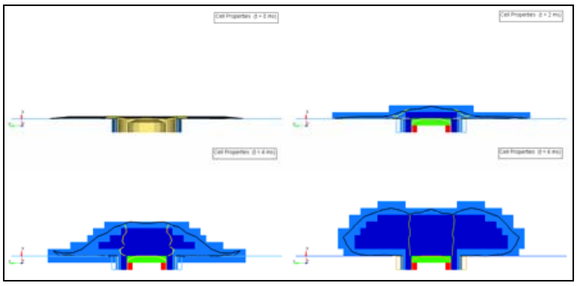

Fig. 31 Airbag inflation – cell properties

At the start, the arrester straps are folded within the airbag and do not limit the inflation. After they are completely unfolded (within approx. 5ms) they restrain the airbag axially. Thus the gas flow is turned around and the airbag spreads radially.

The following three illustrations represent the density distribution and the velocity distribution.

The distribution of velocity is symmetrical for approximately 4ms, although at the end of the simulation the distribution of the velocity is no longer symmetrical.

The principal reasons for this are too large cell choice, the numeric and the used algorithm for "switching" on or off of the cells, as was previously described in sub-chapter 4.4.

This has the consequence that asymmetrical cell characteristic distributions (switching cells on/off) can occur. This asymmetrical cell distribution causes an asymmetrical distribution of the velocity.

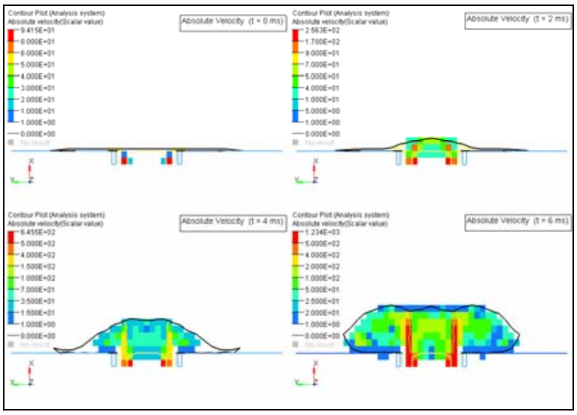

Fig. 32 Airbag inflation – absolute velocity

The strongly aligned gas flow, which is produced by the shooting channel, can be easily recognized. If this arranged gas flow hits the airbag membrane, the gas flow is deflected and an eddy develops. The developing large eddies are represented in Fig. 33.

This detour of the gas flow only takes place if the airbag is held back by means of arrester straps.

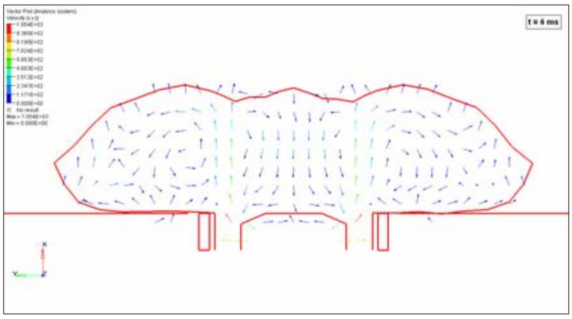

Fig. 33 Airbag inflation –velocity

The density decreases in the airbag in the course of the simulation, as the airbag volume more strongly increases than the mass inflow increases. This can also be seen by the mass accumulations at the sides and at the centre of the airbag.

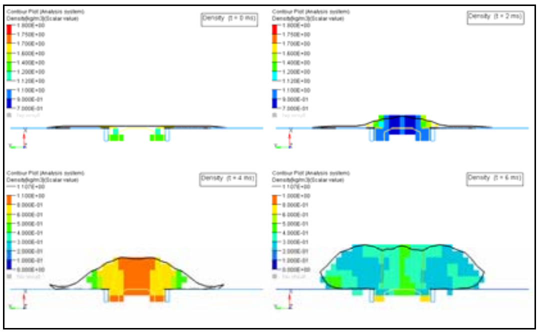

Fig. 34 Airbag inflation – density

Mass balance and energy balance

In the following two illustrations, the theoretical mass and energy are compared with mass and energy in the simulation.

The theoretical mass is derived from the initial mass at time zero and the entire mass flowing in. The represented simulation mass contains both - the mass of the fluids cells and that of the airbag obstacle cells.

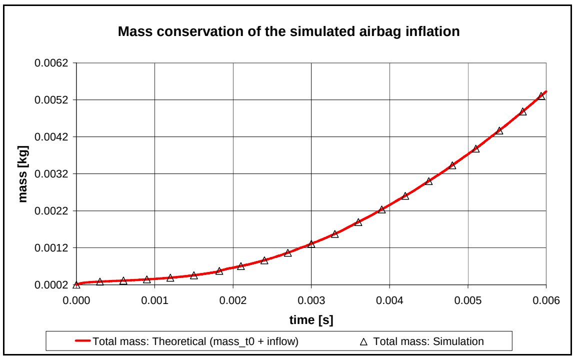

Fig. 35 Airbag inflation – mass conservation

In Fig. 35 it can be clearly seen that the system is conservative.

The same applies to the conservation of energy as it shown in the Fig. 36.

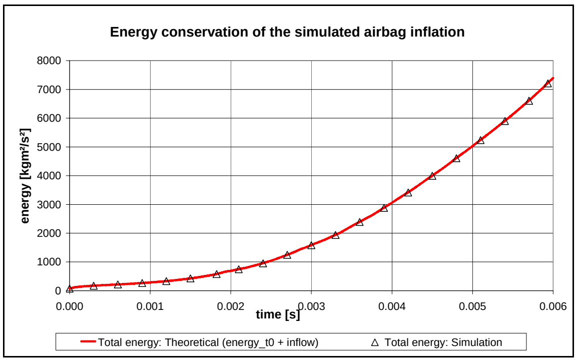

Fig. 36 Airbag inflation – energy conservation

Computational time

The necessary CPU time (1 AMD Opteron 246, 1 GB RAM) is divided into four sub processes:

a. The process "**LS Dyna**" covers the entire FE computation and the "waiting period" before calling and after terminating the FE program (altogether three seconds for each time step). This waiting period ensures that each program is terminated "definitely".

b. The process time "**Bagjet**" contains the entire time necessary for the flow simulation

c. In the process time "**Read node position**" the flow program reads the node positions written by the FE computation. For this, the LS Dyna output file, which contains the node positions, is read in.

d. The process time "**Refresh cell properties**" covers the CPU time, which is required, in order to update the cell properties (Airbag obstacle cell, fluid cell, etc.).

The required CPU time is depicted by process in Fig. 37. Obviously the sub process "Read node position" needs the most CPU time.

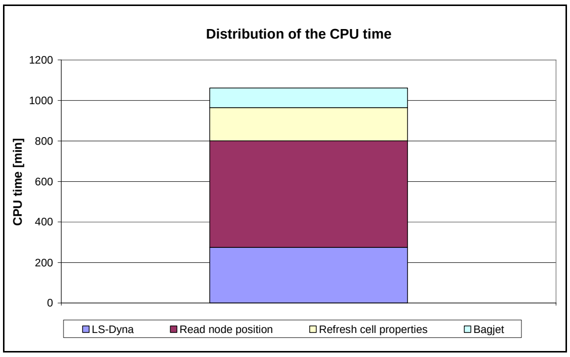

Fig. 37 Distribution of the CPU time – overview

The reason for the large CPU time demand for the "Read node position" process is caused by the interface used (file with the node position) for data exchange between the FE program and the flow program.

The output file used cannot be "deleted" after a time step and the current node positions are attached to the previous output file. Thus as the size of the output file increases, the more membrane elements for the modelling of the airbag are used.

In Fig. 38, the necessary CPU time is represented for each time step.

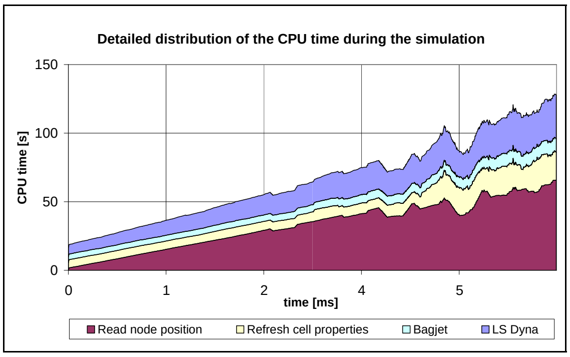

Fig. 38 Distribution of the CPU time - detailed

The program Bagjet needs more time as the number of fluids cells increases during the simulation. The increase of the CPU time for the process LS-Dyna is a result of the longer initialization time. Also the strong increase of the CPU time for reading the node position is obvious in Fig. 38.

8 Potential improvements

In the course of this work only the fitness in principle of this new method was examined for the simulation of an airbag inflation. In this chapter covering individual program sections, improvements are suggested and their implementation is described briefly. The fact that an improvement in one section may be counter-effective to another program-section must always be considered. Thus, an improvement in the accuracy of the flow computation for example causes a rise in the computation time. These possible side effects are not described in this chapter.

8.1 Flow model

Diffusion term

As was already described, only the Euler flux, which contains all portions for the description of frictionless flows, was implemented. The diffusion flux **Ev**, which contains the friction terms and heat flow, is left out. On average, it takes approximately 20 to 30ms to inflate the airbag. Due to this short time period, the heat transfer between the gas and the airbag membrane is negligible. During the inflation, the gas flow may show large velocity gradients. These gradients play a crucial role for the computation of the shear stresses. The influence of the shear stresses on the simulation accuracy has to be analysed.

Implicit method

The CFL condition indicates the maximum time step of the numeric procedure. With explicit procedures, the CFL number must be smaller than 1 – as for implicit procedures the CFL number can amount to 100 or more. Implicit procedures show temporally a very dissipative behaviour and are maximally second order exact, while for example a four step Runge Kutta method is fourth order exact.

The accuracy and the stability are not only influenced by the time step. Additionally the time step should be selected in such a way that the moved edge does not "skip" complete cells (geometrical restriction).

Therefore the time step cannot be selected arbitrarily. Due to the possible higher temporal accuracy and the "geometrical" restriction during an area enlargement, explicit methods are preferred over implicit methods for the developed procedure.

Adaptive mesh refinement

For the developed procedure, the convention was made that as soon as the airbag membrane divides a cell, this cell gets the characteristics of a boundary cell so a "level" edge develops. For this reason it would be wise to implement an adaptive net refinement in the peripheral areas. This would help in making the procedure more exact and minimizing the algorithm-conditioned errors as they are depicted in the figure below.

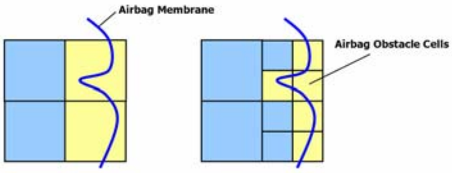

Fig. 39: Adaptive mesh refinement

In order to facilitate an implementation of an adaptive mesh refinement, an unstructured grid is used.

8.2 Internal program improvements

Search algorithm

An additional data structure is needed when using an unstructured grid. In order to be able to address an adjoining cell, for example, it is not sufficient to simply change the cell index. It is necessary to search in the additional data structure for the appropriate cell.

In the context of this work all computation cells are stored, including the characteristics (density, etc.), in a vector. A unique number is assigned to each computation cell. However, this "computation cell vector" contains no information about the grid structure.

To reconstruct the grid structure a matrix is used which contains only cell numbers. Each line in this matrix is assigned to a computation cell. In the first column the cell numbers of the computation cells are located. In the additional columns the neighbour cell numbers are stored. Therefore it is possible to reconstruct the structure of grid with this matrix.

If for example the velocity of a cell is searched, the "computation cell vector" is scanned downwards line by line until the appropriate cell is found. It would be wise to change the vector structure in such a way that faster search algorithms can be implemented. By an appropriate tree structure [28] the search algorithm is accelerated and therefore the numeric computation time is reduced.

Parallelization of the CFD program
Using computational clusters, it is wise to split the computation into sub-problems meaning that the computation time can be reduced significantly. Parallelization can be applied well in explicit methods.

8.3 Interface: FE – CFD program

Ideally the developed program is to implement in a FE program. The direct access to the data will lead to a noticeable reduction in the computation time.

8.4 Validation examples

It would be reasonable to validate the developed procedure using real airbag inflations.

Eventually a comparison of the computing time of the developed procedure with the computing times of other procedures (ALE, etc.) should be carried out. To do so, a fast interface is required.

9 Summary

The goal of this work was to develop as stable a procedure as possible for the surface-coupled multi-field problem of an airbag inflation which could be expected to yield shorter computing times. As a solution procedure for this fluid structure interaction, a weakly partitioned procedure was used. For the solution of the structure-mechanical problem, the commercially available finite elements program LS Dyna was chosen.

For the solution of the flow mechanical problem a 3D-Euler code was developed. A TVD Upwind procedure by Roe and for the temporal discretization an explicit Euler procedure and a four step Runge Kutta procedure were implemented.

The area of an airbag inflation, where a solution of the Navier Stokes equations is searched, is subject to a temporal change. This change is caused by the deformation of the edge of area, so the procedure must be able to include time-dependent areas into the solution process. The outer contour of the grid adjusts automatically to the surface of the airbag during the calculation. The contour adjustment is achieved by blocking off the outer cells, which can be switched on and off during the calculation process.

By means of the characteristic theory, the boundary conditions were applied. Additional it is important that the procedure remained conservative. For this reason, a special conservative procedure for the treatment of the boundary condition was developed which permits the use of a regular grid. This method allows the switching on or off of complete cells.

For the examination of the developed flow program, examples were used in which the analytic solutions are known, meaning that the numeric and analytic solutions could be compared. Furthermore, on the basis of examples it was shown that the procedure is conservative. As a further example, an airbag inflation was simulated in order to be able to examine the fitness of the developed procedure.

Finally, possible improvement suggestions and extensions were described. Particularly the missing suitable interface between LS Dyna and the developed CFD program was described.

Finally it can be said that the procedure used is very stable, including the developed boundary condition for the simulation of an airbag inflation and suggests short computing times.

10 References

[1] Pam Crash. Pam Talk – News from the Virtual Try-Out Space. ESI Group, Rungis Cedex, 2004.

[2] J. Kuhnert. General Smoothed Particle Hydrodynamics: Ph.D. Thesis, Kaiserslautern University. 1999.

[3] Pam Crash. Pam Safe 2005 – Solver Notes Manual. ESI Group, Rungis Cedex, 2005.

[4] B. Beesten, R. Reilink, A. Hirth, R Remensperger, D. Rieger, G. Seer. OOP-Simulation – A Tool to Design Airbags? Current Capabilities in Numerical Simulation. Proceedings of the 7th Int. Symposium and Exhibition on Sophisticated Car Occupant Safety Systems, Airbag 2004. Karlsruhe. 2004.

[5] L. Olovsson. On the Arbitrary Lagrangian-Eulerian Finite Element Method. Ph.D. Thesis, Linköping University, 2000.

[6] J. Hallquist. LS-Dyna – Theoretical Manual 2005. Livermore Software Technology Corporation, Livermore, 2005.

[7] P.L.Roe. Approximate Riemann Solvers - Parameter Vectors and Differencing Schemas. Journal of Computational Physics, Vol. 43. 1981.

[8] W Sanz. CFD in Turbomaschinen und Energieanlagen. Graz University of Technology. Graz. 2003.

[9] N. Bronstein, K. Semendjajew, G. Musiol, H. Mühlig. Taschenbuch der Mathematik. 5. Auflage. Frankfurt. 2000

[10] C.W.Hirt, B.D.Nichols. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. Journal of Computational Physics, Vol. 39. 1981.

[11] W. Wall. Fluid Struktur Interaktion mit stabilisierten Finiten Elementen. Universität Stuttgart. Stuttgart. 1999.

[12] C. Hilchenbach. Simulation der Wechselwirkungen von Wind und umströmten Brückenquerschnitten. Univeristät Stuttgart. Stuttgart. 2004.

[13] S.R.Chakravarthy. High Resolution Upwind Formulations for the Navier-Stokes-Equations. VKI Lecture Series on Computational Fluid Dynamics. 1988.

[14] M. Neumann. Computational Physics I: Grundlagen. Univeristät Wien.Wien. 2006

[15] M Baas, M. Fautz, A. Schmitt. Fotorealistische Wasseranimation bei Brunnenanlagen. Technische Universität Karlsruhe. Karlsruhe. 2000

[16] T. Pyttel, A. Floss, C. Thibaud, C. Goertz. Realitätsnahe Simulationsmodelle für Airbag und Mensch - Neue Möglichkeiten und Grenzen der FE Simulation. ESI Group, Rungis Cedex. 2005

[17] J. Sauer. Instationär kavitierende Strömungen - Ein neues Modell, basierend auf Front Capturing (VoF) und Blasendynamik. Universität Karlsruhe. Karlsruhe. 2000.

[18] M.Aftosmis, M. Berger, M. Melton. Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry. AIAA Journal, Vol. 36. 1998.

[19] E. Laurien. Randbedingungen für kompressible Strömungen. Universität Stuttgart. Stuttgart. 2006.

[20] Li, Hesse, Ziegler, Woods. An arbitrary Lagrangian Eulerian Method for Moving-Boundary Problems and its Application to Jumping Over Water. Journal of Computational Physics , Vol. 208. 2005.

[21] H.S. Udaykumar, R.Mittal, P. Rampunggoon, A. Khanna. A Sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries. Journal of Computational Physics, Vol. 174. 2001.

[22] R. LeVeque, R. Russell, S. Ruuth. Computational Techniques for Moving Interfaces. Banff International Research Station for Mathematical Innovation and Discovery. 2003.

[23] S.A Bayyuk, K.G. Powell, B. van Leer. A Simulation Technique for 2-D Unsteady Inviscid Flows Around Aribitrarily Moving and Deforming Bodies of Arbitrary Geometry. AIAA Technical Paper 93-3391. Washington DC. 1993.

[24] J. D. Hunt, K. G. Powell. An Adaptive 3D Cartesian Approach for the Parallel Computation of Inviscid Flow About Static and Dynamic Configurations. The University of Michigan, Department of Aerospace Engineering. 2003

[25] S. M. Murman, M. J. Aftosmis, M. J. Berger. Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method. 41st AIAA Aerospace Sciences Meeting January 6-9. Reno. 2003.

[26] M. Arienti, P. Hung, E. Morano, J. E. Shepherd. A Level Set Approach to Eulerian-Lagrangian Coupling. Caltech ASCI Technical Report 136. 2003.

[27] Ashley, Holt, Garabed, Zartarian. Piston Theory- A New Aerodynamic Tool for the Aeroelastician. Presented at the Twenty-Forth Annual Aeroelasticity Meeting. New York. 1956

[28] M.J. Aftosmis. Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries. Karman Institute for Fluid Dynamics, Lecture Notes for 28th Computational Fluid Dynamics Lecture Series. 1997

ISBN 978-3-901608-32-2