Recent Developments in *DEFINE_PRESSURE_TUBE for Simulating Pressure Tube Sensors in Pedestrian Crash

Jesper Karlsson
DYNAmore Nordic

Abstract

*This paper presents recent developments of the new keyword *DEFINE_PRESSURE_TUBE, designed to efficiently simulate pressure waves in thin air-filled tubes. The primary application is a new crash detection system for pedestrian impact, where a thin air-filled tube is embedded in the front bumper and fitted with pressure sensors at the ends. On impact, the tube is compressed and a pressure wave travels to the sensors, enabling localization and extent of the impact. In recent years, such systems have gained popularity in the automotive industry, posing a challenging task in efficient and accurate simulations.*

Introduction

The *DEFINE_PRESSURE_TUBE keyword defines a gas filled tube using beam elements. Gas pressure propagation over the tube length is governed by a 1D acoustic model where pressure waves are created from changes in the tube cross-section area. Originally, the cross-section area was calculated from contact penetration of the beam elements, which is almost entirely governed by contact stiffness. The lack of a physical model for radial compression of beam elements thus makes it difficult to find contact stiffness parameters that accurately model the tube compression. A new addition to the keyword is the possibility to automatically generate a shell element tube which gives a more accurate physical tube response. The cross-section area is then given by the actual tube deformation and not by contact penetration.

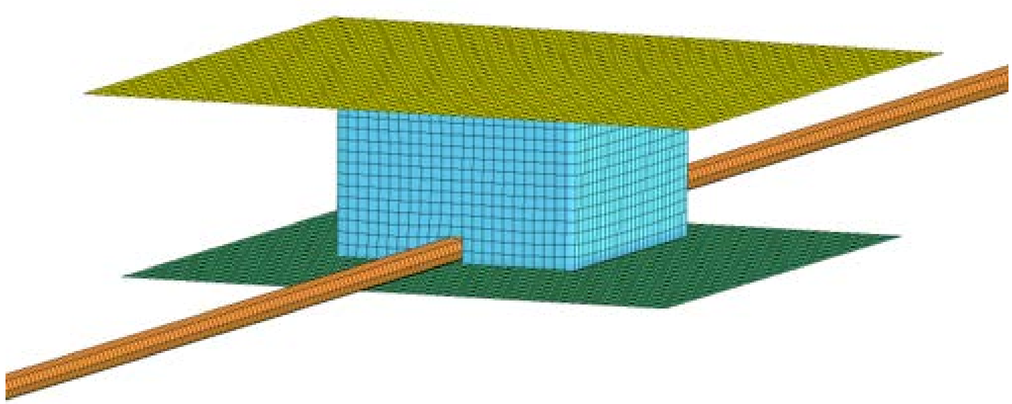

Figure 1: Partially foam-encapsulated tube.

Model

Pressure propagation is governed by a 1D acoustic model based on the compressible Euler equations, resulting in a very efficient method compared to 3D CFD or particle methods. See [1] for a detailed derivation.

From the tube cross-section area $A(x,t)$, where $(x,t) \in \mathbb{R} \times \mathbb{R}_+$, the gas pressure $p(x,t)$ and gas velocity $u(x,t)$ are given by the acoustic equations

$$\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A}\frac{\partial y}{\partial x} = 0,$$
$$\frac{\partial y}{\partial t} + A\frac{c^2}{p_0}\frac{\partial p}{\partial x} = 0.$$

where $y = Au$, c is the speed of sound, and p_0 is the initial pressure. This is a generalized version of the classic acoustic wave equation

$$\frac{\partial^2 p}{\partial t^2} - c^2\frac{\partial^2 p}{\partial x^2} = 0,$$

which we get from the above equations if $A = 1$.

The acoustic equations are solved using the standard Galerkin finite element method, using piecewise linear basis functions and artificial diffusion/viscosity. The computational mesh for the pressure and velocity consists of the initial locations of the beam nodes, i.e. beam deformation will not affect the acoustic solution. The only coupling between the mechanical and the acoustic solver is that the former will supply the cross-section area to the latter.

An optional linear damping term can be added to model energy losses from friction between the gas and the tube walls. Adding linear damping and artificial viscosity, the final system can be written as

$$\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A}\frac{\partial y}{\partial x} = \epsilon\frac{\partial^2 p}{\partial x^2} - DAMP \cdot (p - p_0)$$
$$\frac{\partial y}{\partial t} + A\frac{c^2}{p_0}\frac{\partial p}{\partial x} = \epsilon\frac{\partial^2 y}{\partial x^2},$$

where the artificial viscosity ϵ is proportional to the maximum initial beam element length, i.e.

$$\epsilon = VISC \cdot c \max_i \Delta x_i,$$

The artificial viscosity is a numerical stabilization technique that prevents numerical errors to accumulate over time and destroy the solution.

Time integration is done with Heun's method, a second order Runge-Kutta method, and is independent of the mechanical solver. It uses a step size less than or equal to the global time step, satisfying the CFL condition

$$\Delta t < \min_i \frac{CFL \cdot \Delta x_i}{\Delta x_i \left|\frac{\partial \ln A}{\partial t}\right| + 3c}.$$

This is a necessary condition for convergence and makes sure that the time it takes for a pressure wave to pass a computational domain of size Δx is longer than the computational time step Δt. Thus, the computational solution has a reasonable chance to keep up with the wave.

Keywords

Given a part ID consisting of beam elements, a tube is defined by *DEFINE_PRESSURE_TUBE with the input given as below:

Card 1	1	2	3	4	5	6	7	8
Variable	PID	WS	PR	MTD	ATYPE			
Type	I	F	F	I	I			
Default	0	0.0	0.0	0	0			

Optional card if MTD=0:

Card 2	1	2	3	4	5	6	7	8
Variable	VISC	CFL	DAMP					
Type	F	F	F					
Default	1.0	0.9	0.0					

Optional card if ATYPE=1 (requires card 1):

Card 3	1	2	3	4	5	6	7	8
Variable	NSHL	ELFORM	NIP	SHRF	BPID			
Type	F	F	F	F	F			
Default	12.0	16.0	3.0	1.0				

Only the first three parameters on the first card are compulsory:

- **PID:** Part ID of tube. The tube consists of all the beam elements in the part. Only tubular beam elements are supported. Initial tube cross sectional area is calculated using the beam inner diameter, or if no inner diameter is given, the outer diameter.
- **WS:** Speed of sound in gas.
- **PR**: Initial gas pressure.

Parameter 4 and 5 concerns solver type and tube area calculation:

- **MTD:** Solution method. Only Standard Galerkin (MTD=0) is currently supported.
- **ATYPE:** Type of cross-section area calculation of the tube:
    0. The tube is entirely simulated with beam elements. Cross-section area is given from contact penetration of the beam elements. The mechanical response in radial direction of the beam elements is governed by contact stiffness. Only mortar contacts are supported.
    1. The tube is simulated by automatic generation of shell elements, which are assigned the beam part ID and the beam material model. A new part ID is given to the beam elements, and those are no longer part of the mechanical solution, but still contain the acoustic variables. Contacts and other properties associated with the old beam part ID will now apply to the new shell part. Cross-section area is given from the shell element nodes and the mechanical response is governed entirely by the shells. Supports all contact definitions.

For MTD=0 an optional card with solver parameters may be given:

- **VISC:** Artificial viscosity factor.
- **CFL:** Time step factor for tube sub-stepping.
- **DAMP:** Linear damping factor to emulate pressure losses.

For ATYPE=1, a second optional card with the shell tube parameters may be given:

- **NSHL:** Number of automatically generated shells on circumference of tube.
- **ELFORM:** ELFORM for automatically generated shells, see *SECTION_SHELL.
- **NIP:** NIP for automatically generated shells, see *SECTION_SHELL.
- **SHRF:** SHRF for automatically generated shells, see *SECTION_SHELL.
- **BPID**: New optional PID given to beam elements when automatically generating shells.

From a given beam PID, the solver will get the initial tube dimensions from the length and thickness of each beam element on that part. After initialization, the tube solver only uses either beam contact penetration (ATYPE=0), or deformation of the automatically generated shell elements (ATYPE=1), to calculate cross-section area over time. The tube solver is independent of the length-wise beam/shell element deformation, i.e. it is assumed that the length of the tube is not too distorted over time. It also uses a separate time integration routine, with time step size less than or equal to the global time step.

Pressure, density, velocity and tube area are output through the keyword *DATABASE_PRTUBE and can be visualized in LS-PrePost®.

Examples

In the first example a mass is dropped onto a 1.7 m long silicone tube with inner diameter 4 mm and outer diameter 8 mm. Pressure is measured at the end. Three different cases were tested, see Figure 2-5:

1. **CPM:** The Corpuscular Particle Method may be the most physically accurate method of the three since it models the interaction between "gas particles" and the structure, without any special assumptions on the tube geometry and tube-gas interaction. However, CPM generally gives noisy results and is quite expensive. In our examples the tube is modeled by shell elements and is filled with two million particles, requiring 170 hours of total CPU time.
2. **Shell tube with embedded pressure tube:** Here tubular beam elements (ATYPE=0) with inner diameter 0 are used, embedded inside a shell tube. The beam elements model air only, thus density and stiffness of the associated material are set to reasonably low values. Structural response is governed by the shell elements and the beam-to-shell contact stiffness corresponds to tube air pressure response. This case requires about four hours total CPU time.
3. **Automatically generated shell tube:** Here a shell tube is generated from beam elements using ATYPE=1. This case requires about one hour total CPU time.

The second example is the partially foam-encapsulated tube in Figure 1. In Figure 6, the automatically generated shell tube embedded in a block of foam is compared with experimental data.

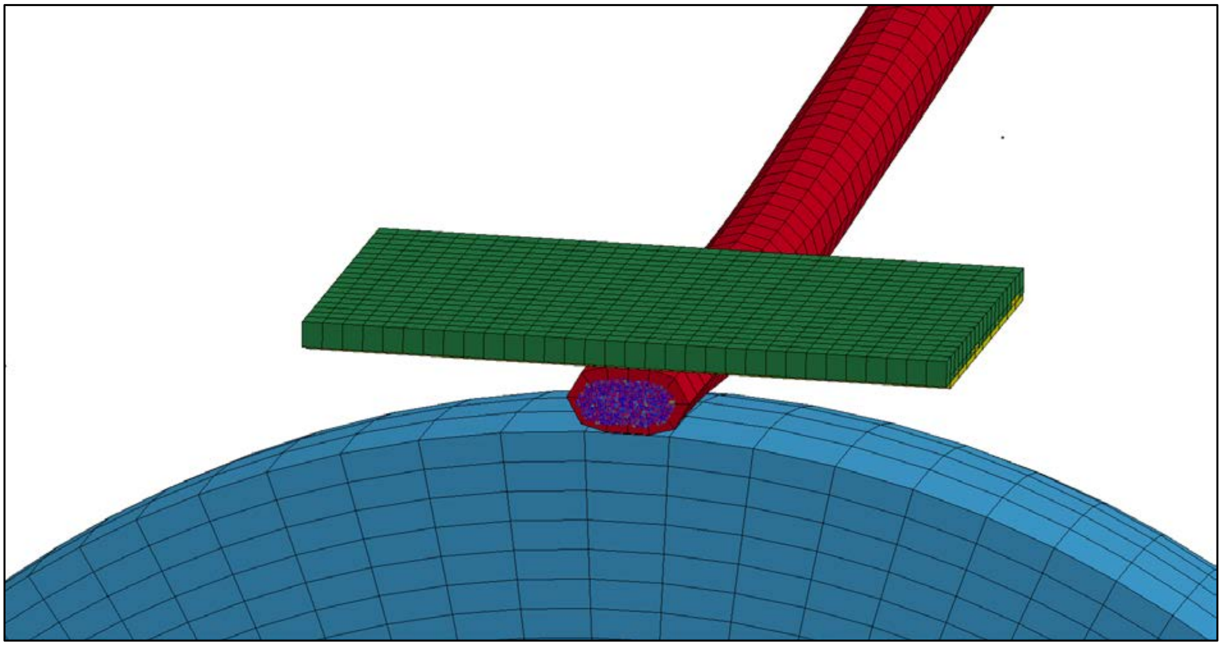

Figure 2: CPM

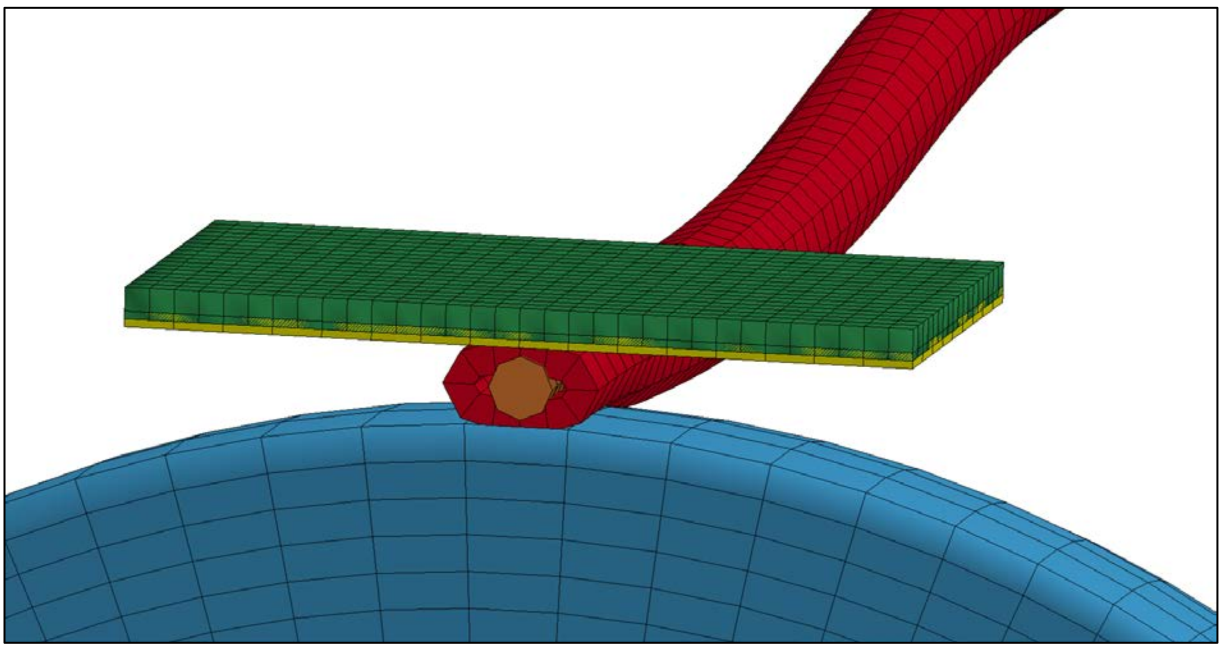

Figure 3: Beam elements embedded in a shell tube.
Cross-section area is calculated from shell-to-beam contact penetration.

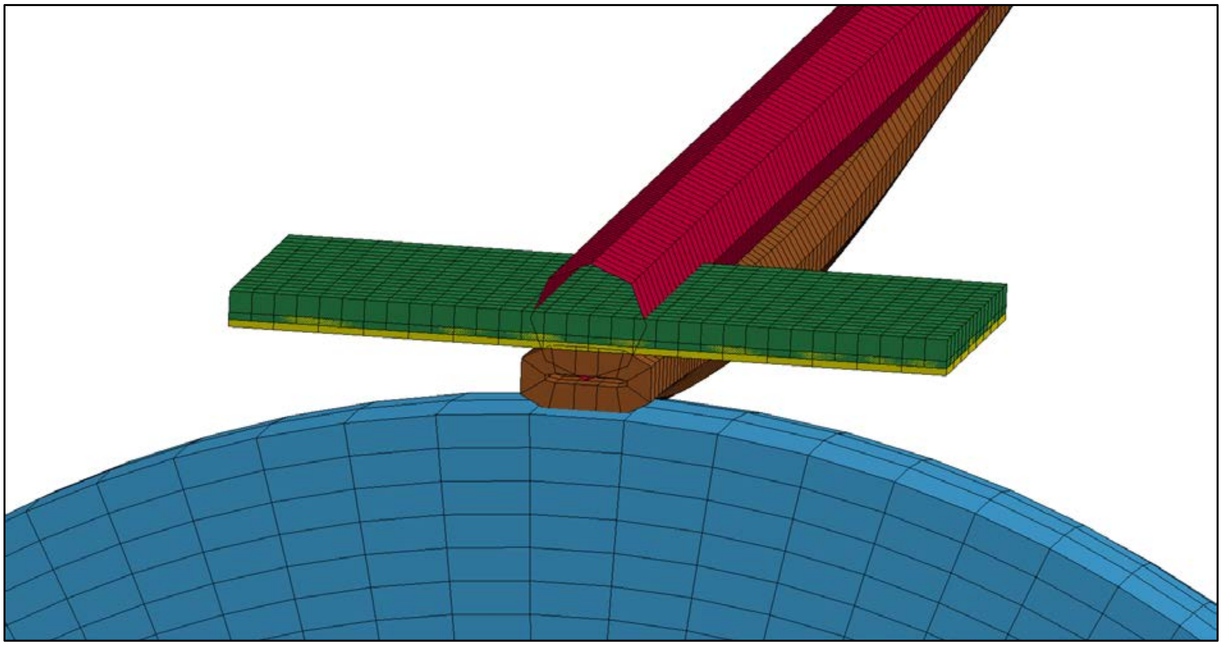

Figure 4: Automatically generated shell elements using ATYPE=1. The beam tube is still present in the model (visualized using beam prisms) but is given a new PID and is not part of the simulation. The cross-section area is calculated from displacement of the shell tube nodes.

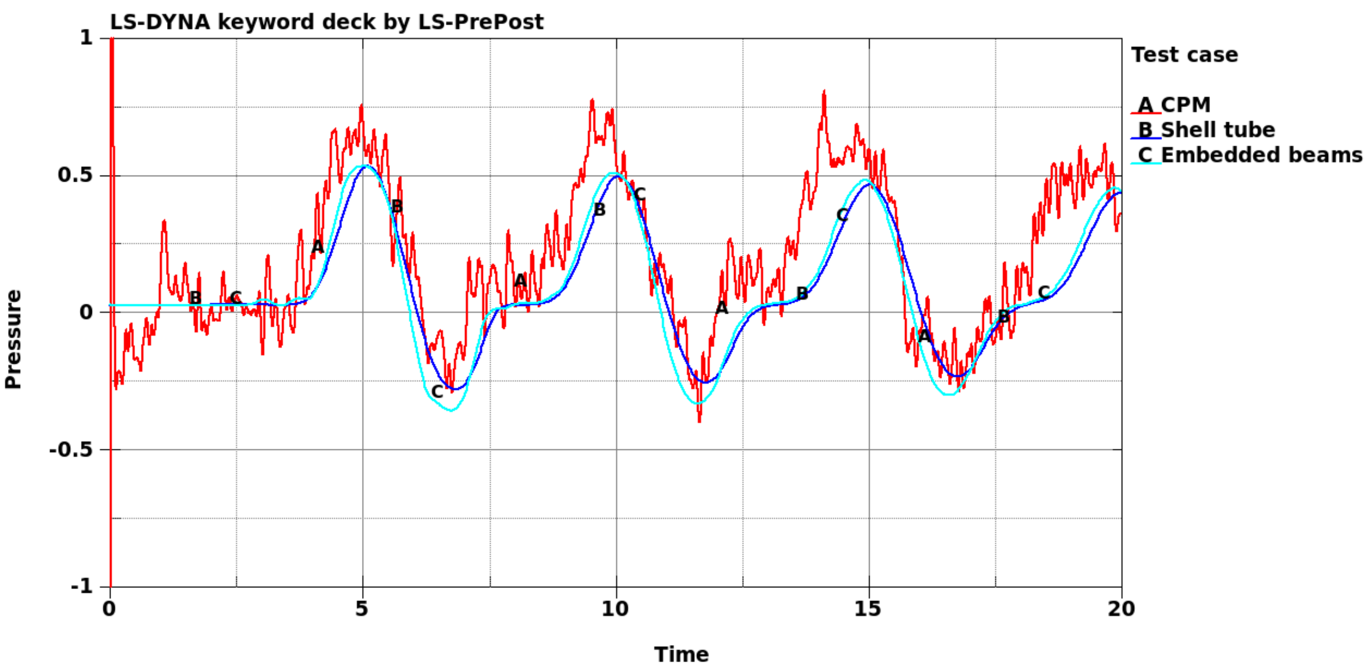

Figure 5: Pressure at end of tube. As expected, the results for the automatically generated shell tube and the beams embedded in a shell tube agree quite well.

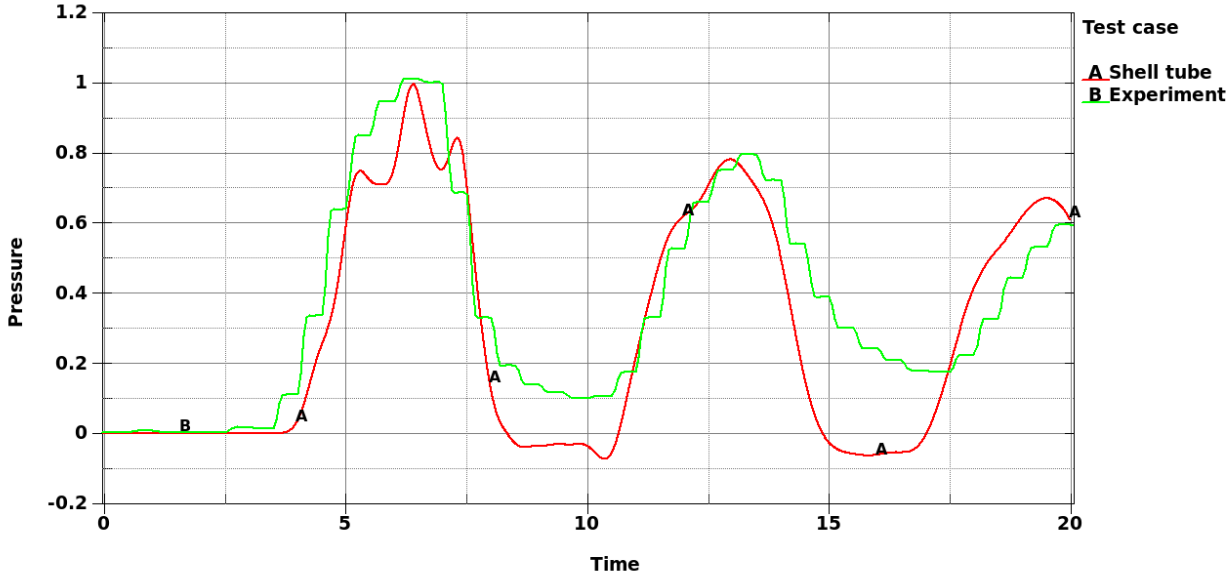

Figure 6: Comparison with experimental data for the partially foam-encapsulated shell tube (generated with ATYPE=1) in Figure 1. Experimental data and model courtesy of Volvo Car Corporation.

References

[1] J, Karlsson, DEFINE_PRESSURE_TUBE: Simulating Pressure Tube Sensors in Pedestrian Crash, *11th European LS-DYNA® conference*, 2017.