A moving piston boundary condition including gap flow in OpenFOAM

CLEMENS FRIES
Johannes Kepler University
IMH
Altenbergerstrasse 69, 4040 Linz
AUSTRIA
clemens.fries@jku.at

BERNHARD MANHARTSGRUBER
Johannes Kepler University
IMH
Altenbergerstrasse 69, 4040 Linz
AUSTRIA
bernhard.manhartsgruber@jku.at

Abstract: Fuel injection as well as digital switching strategies in fluid power applications are not only famous representatives of a large field of technology but also a main reason for the increasing interest in wave propagation effects in research. While there is a huge number of works dealing with the pressure drop of different hydraulic components in the steady state, many issues still remain unresolved in the transient regime, even in the case of laminar fluid flow. A better understanding of these processes would be a great benefit as it would lead to a higher accuracy of predicted system responses. In order to reach a higher degree of precision, the highly sophisticated computational fluid dynamics (CFD) codes are a wide-spread tool. These codes solve the famous Navier-Stokes equations in all three dimensions of space and therefore result in the full resolution of the pressure field as well as of the velocity field. A very awkward topic of performing a CFD simulation is the choice of the boundary condition, which should correspond to a physical one. At the latest when measurements for validation are carried out, the boundary condition of the experimental setup should match the one used in the simulation. Especially the use of a volumetric flow rate boundary condition is fraught with problems. Using a moving piston, a definite volumetric flow rate could be forced on a boundary. In an experimental setup only the measurement of the position of the piston would be necessary to use it in the simulation. This measurement has no backlash on the system, which is therefore well separated. In this work a moving piston boundary condition including gap flow is implemented and used in OpenFOAM. For this reason moving walls have to be used and the mesh has to change during the simulation. Results of simulations done with this moving piston boundary condition are compared with simulations done with an ordinary volumetric boundary condition.

Key–Words: Fluid power, fluid mechanics, fluid flow, transmission lines, hydraulics, CFD, OpenFOAM

1 Introduction

While a lot of work has been done concerning the steady state pressure drop of countless hydraulic components, the transient effects have been treated like an orphan in the hydraulic community for a long time. With the increasing academic interests in digital switching strategies for fluid power applications - just to mention one famous representative - also the transient regime became focus of attention. In particular the well-known wave propagation effects caused for instance by fast switching valves moved in the center of interests. Contemporary processing power has grown steadily and so computational fluid dynamics (CFD) codes became a valuable and meanwhile widespread tool. These highly sophisticated codes solve the famous Navier-Stokes equations, which describe the motion of vicous fluids, and yield the velocity field as well as the pressure field in all three dimensions of space. One of the most challenging tasks executing a CFD simulation is the choice of the boundary conditions. Reason for this is, that, with the exception of walls, boundary conditions are often not known in a great detail from a physical point of view. Nevertheless it's quite common to use for instance a constant value velocity, or a constant pressure boundary condition. Thinking of a fluid power system with its widenings and narrowings, its bends and junctions, it's anything but easy to find an appropriate position for such an assumption. Knowing that simulation results can be rather sensitive to aberrations of the physical boundaries, this results in the strong desire of a well defined boundary condition. Regarding the fact, that walls as boundaries are the only ones, which can be described in a physical correct way, the idea comes up to use a moving piston as boundary input. A schematic of this idea is shown in Fig. 1. In order to build an experimental setup for measurements, Lukas Muttenthaler [1] developed a hydraulic

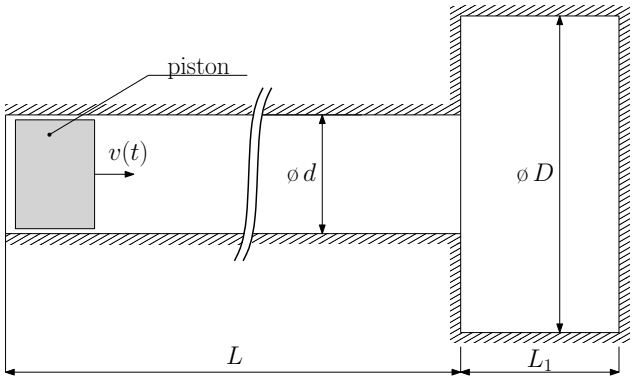

Figure 1: Principle of experimental setup

servo cylinder with an elastohydrostatic linear bearing in his masterthesis. Measuring the velocity (or the position) of the piston also the volumetric flow through the inlet would be known. The measurement of the velocity compared to the measurement of the flowrate itself has no backlash on the system, which would be a great benefit.

For this work a moving piston boundary condition including gap flow has been implemented in OpenFOAM, an open source CFD package. Details of this softwarepackage can be found on http://www.openfoam.com/ [2]. Simulations have been accomplished with this new boundary condition and are compared to a ordinary constant velocity boundary condition.

2 Boundary condition

2.1 Moving Mesh

In order to use a moving piston boundary condition in OpenFOAM, the use of moving meshes is necessary. The mathematical background for instance can be found in Ferziger and Peric [3]. Jasak and Tukovic [4] describe the idea of automatic mesh motion for unstructerd meshes and Kassiotis [5] explains different stratgies for mesh motion in OpenFOAM. Basically there exist two different approaches for the movement of a boundary. On the one hand the original mesh can be stretched and sqeezed in the direction of the movement and on the other hand cells can be added or removed when a maximum or minimum of a desired cell size has been reached. These principles are shown in Fig. 2. The first method is directly implemented in OpenFOAM. All OpenFOAM solvers with the letters "DyM" (DynamicMesh) in the name are capable of handling these moving meshes. For the pupose of usage of one of these solvers the file $dynamicMeshDict$ in the $constant$ folder as well as the file $pointMotionUz$ in the 0 folder is necessary. The entries for these files can be found in the listings 1 and 2. For this work the ability of moving meshes has been added to the solver sonicLiquidFOAM, a transient solver for laminar flow of a compressible liquid. The movement of the cylinder has been defined in a way it is shown in Fig. 3. The upper function corresponds to the piston position, whereas the lower one is equal to the velocity. The function of the position can be written as

$$s\left(t\right) = s_0 \cdot \left(\frac{1}{2} - \frac{1}{2} \cdot \cos\left(\frac{t}{T} \cdot \pi\right)\right),$$

where s_0 corresponds to the upstroke and T to the length of time. For the case shown in Fig. 3 $s_0 = 20\,\text{mm}$ and $T = 10\,\text{ms}$. The function values for the velocity at equidistant timesteps have been written in a file (called tableFile in OpenFOAM), which can be directly used in OpenFOAM. The distance between two timesteps has been chosen to be equal as the timestep used in the CFD simulation. If the timestep of the CFD simulation lies in between two timesteps of the tablefile, this value is interpolated linearly automatically.

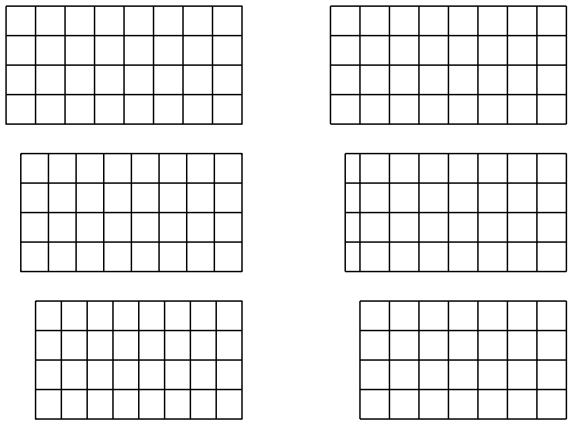

Figure 2: Different approaches of mesh motion

2.2 Gap Flow

The flow through the gap between piston and cylinder indeed could be simulated with CFD, the calculation effort however would be immense. For this gap, which is approximately $\frac{1}{1000}$ of the cylinder diameter, the spatial resolution would scale up, which would also lead to a tiny temporal resolution. Because of this reason the gap flow has been modeled with the Reynolds equation as it can be found in Hori [6].

```
motionSolverLibs
(
   "libfvMotionSolvers.so"
);

dynamicFvMesh   dynamicMotionSolverFvMesh;

solver          velocityComponentLaplacian;

velocityComponentLaplacianCoeffs
{
    component       z;
    diffusivity     uniform;
}
```

Listing 1: dynamicMeshDict

```
dimensions    [0 1 -1 0 0 0 0];

internalField uniform 0;

boundaryField
{
    inlet
    {
        type            uniformFixedValue;
        value           uniform 0;
        uniformValue    tableFile;
        tableFileCoeffs
        {
            fileName    "u1";
        }
    }

    inlet_gap
    {
        type            uniformFixedValue;
        value           uniform 0;
        uniformValue    tableFile;
        tableFileCoeffs
        {
            fileName    "u1";

        }
    }

    outerWalls
    {
        type   slip;
    }

    fixedWalls
    {
        type   fixedValue;
        value  uniform 0;
    }
}
```

Listing 2: pointMotionUz

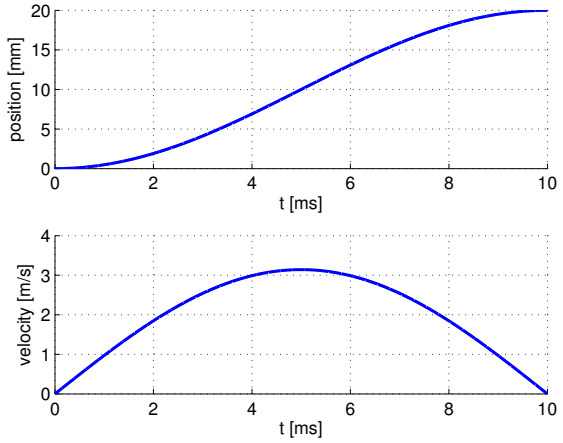

Figure 3: Positon and velocity of moving piston

The equation reads as follows

$$\frac{\partial}{\partial x}\left(h \cdot \frac{\partial p}{\partial x}\right) = 6 \cdot \mu \cdot \left((u_1 - u_2) \cdot \frac{\partial h}{\partial x} + 2 \cdot v_2\right).$$

In Fig. 4 all used variables in this formula can be found. Assuming

$$\frac{\partial h}{\partial x} = 0,$$

meaning that the height h remains constant, and further

$$v_2 = 0,$$

which is equivalent to the statement that the change of height h with respect to time t vanishes, this leads to

$$\frac{\partial}{\partial x}\left(h \cdot \frac{\partial p}{\partial x}\right) = 0.$$

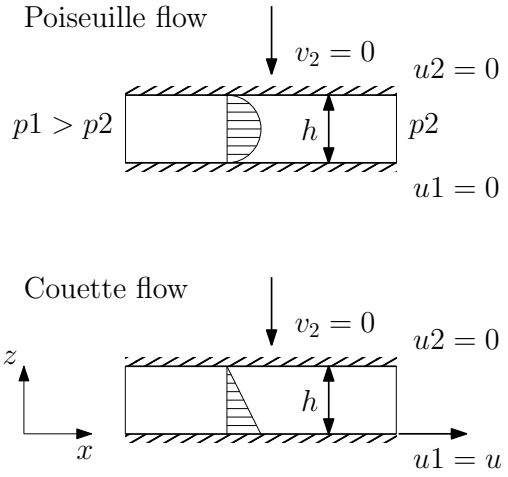

Figure 4: Theory of gap flow

After integration and use of the boundary conditions we get

$$p(x) = \frac{p_2 - p_1}{L_{gap}} \cdot x + p_1.$$

The relation between pressure p and velocity u can be written as

$$\frac{\partial p}{\partial x} = \frac{\partial}{\partial z}\left(\mu \cdot \frac{\partial u}{\partial z}\right)$$

which after integration and using the boundary values leads to

$$u(z) = \frac{1}{\mu} \cdot \frac{\partial p}{\partial x} \cdot \frac{z}{2} \cdot (z - h) + U \cdot \left(1 - \frac{z}{h}\right).$$

Out of this equation and out of Fig. 4 it can be seen, that the gap flow it this case is a combination of poiseuille flow (pressure flow) and couette flow (shear flow).

From this the mean value

$$\overline{u}(z) = \frac{1}{h} \cdot \int_0^h u(z)\, dz = -\frac{1}{\mu} \cdot \frac{\partial p}{\partial x} \cdot \frac{h}{12} + \frac{U}{2}$$

of the velocity of the gap flow follows.

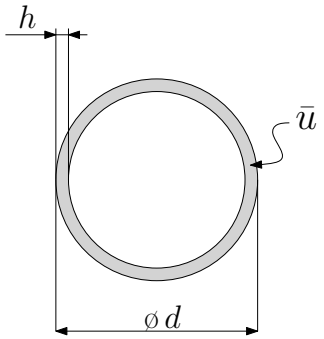

Figure 5: Cross section of piston and cylinder (gap)

This value has been taken for the velocity of the outer annulus of the cylinder, respectively the grey part in the sketch in Fig. 5. Simulations with the explained model with the data found in Tab. 1 and with no massflow through the gap have been accomplished. The difference is rather small and can be seen in Fig. 7. As the massflow through the gap is minor and as there is no measurement for the pressure p_1 on the left side, meanwhile the massflow through the gap has been chosen to be zero.

3 Simulation

Simulations have all been accomplished with $OpenFOAM$, an open source software package. The

density	ρ_0	860	$\frac{kg}{m}$
kinematic viscosity	ν	46	$cSt.$
dynamic viscosity	μ	$\rho \cdot \nu$	$Pa \cdot s$
pressure	p_1	50	bar
diameter cylinder	d	10	mm
outer diameter	D	60	mm
gap height	h	$\frac{d}{1000}$	mm
gap length	L_{gap}	10	mm
length	L	200	mm
length	L_1	30	mm

Table 1: Data

geometry as well as the mesh have been composed with $blockMesh$, an $OpenFOAM$ utility. Both can be seen in Fig. 6. As it is shown in Fig. 1 - with the exeption of the gapflow - all of the boundaries are $walls$. For this work the results of simulations with two differnt $inlet$ boundary conditions have been compared. The first one is equal to a constant velocity boundary condition for the whole crosssection, but the gap. The second one corresponds to a moving piston boundary condition as it is explained before. In this case also the inlet boundary is equivalent to a wall. The fluid has the same velocity as the piston and so the same velocity as in the first approach. According to the first method the diameter of the moving piston is two times the gap height smaller than the diameter of the cylinder.

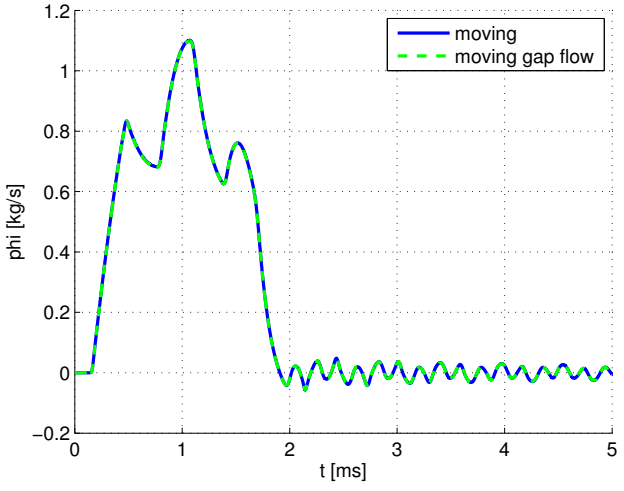

Figure 7: Comparison gapflow and zero gapflow

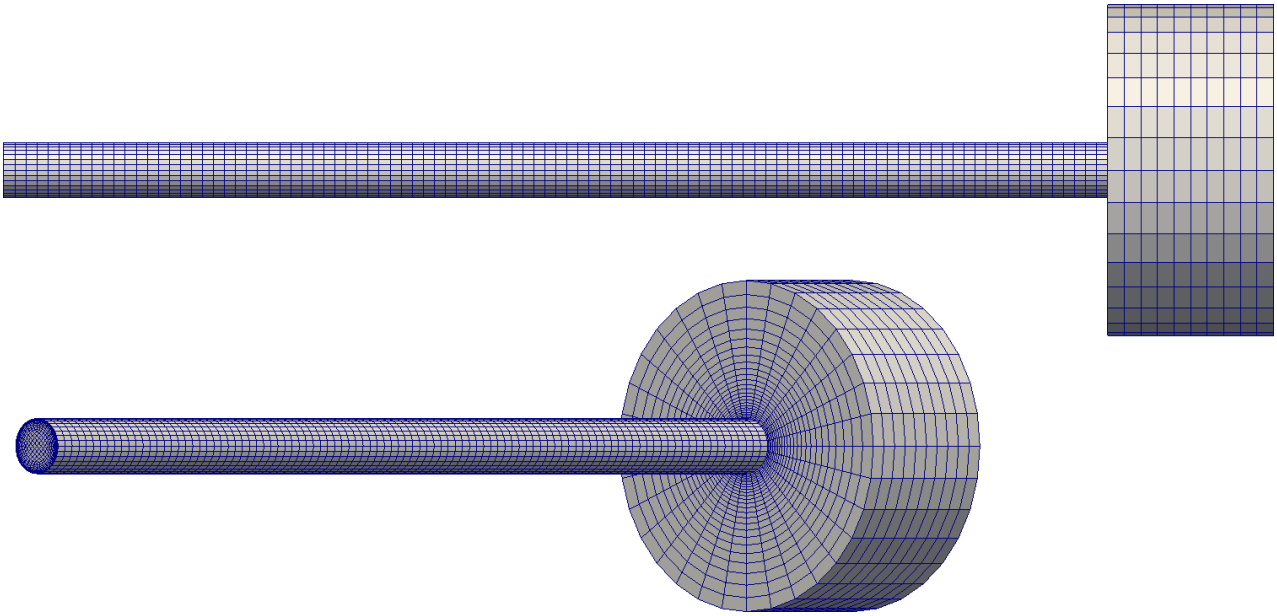

Figure 6: Geometry for CFD simulation

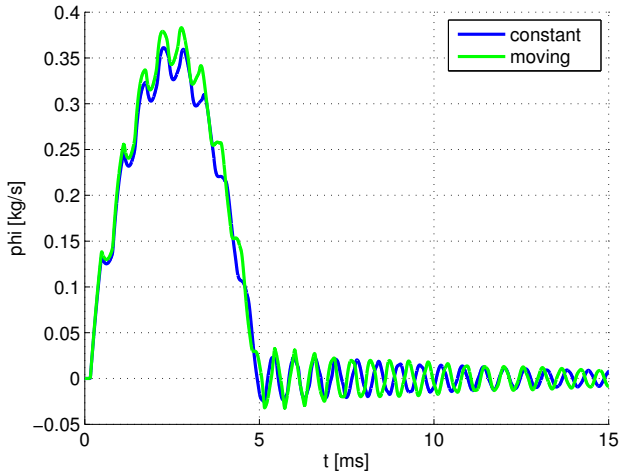

Figure 8: Simulation result constant velocity input vs. moving mesh

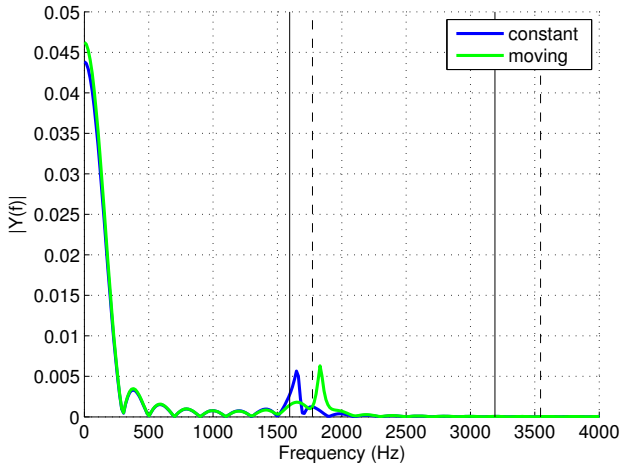

Figure 9: Simulation result constant velocity input vs. moving mesh

The mass flow through the face at $L = 200\,\text{mm}$, the average pressure of the cells right after the end of the pipe (all cells in a cylinder with the diameter $\varnothing\, d$ and a length of a cell layer after $L = 200\,\text{mm}$), as well as the pressure immediately after the piston, which is proportional to the force on the piston, have been recorded.

For the simulation an input velocity, meaning s_0 and T have been defined. Further simulations have been done multiplying the velocity with $\sqrt{2}$. This can be interpreted as (approximately) doubling the energy in the system. In order to keep the mass inflow constant, the duration T has been divided by the same amount.

4 Results

For the first simulations a stroke of $s_0 = 20$ mm has been chosen, while $T = 10$ ms. The time the waves need to travel through a pipe of the length $L = 200$ mm back and forth can be expressed as follows. The velocity of a wave can be written as

$$c_0 = \sqrt{\frac{K}{\rho_0}}.$$

Using a bulk modulus of $K = 14000$ bar and a density $\rho_0 = 860\,\frac{\mathrm{kg}}{\mathrm{m}^3}$ the travelling speed results in

$$c_0 = 1275.9\,\frac{\mathrm{m}}{\mathrm{s}}.$$

This means that a wave needs

$$\Delta t = \frac{2 \cdot L}{c_0} = 0.31\,\mathrm{ms}$$

to run through the whole pipe.

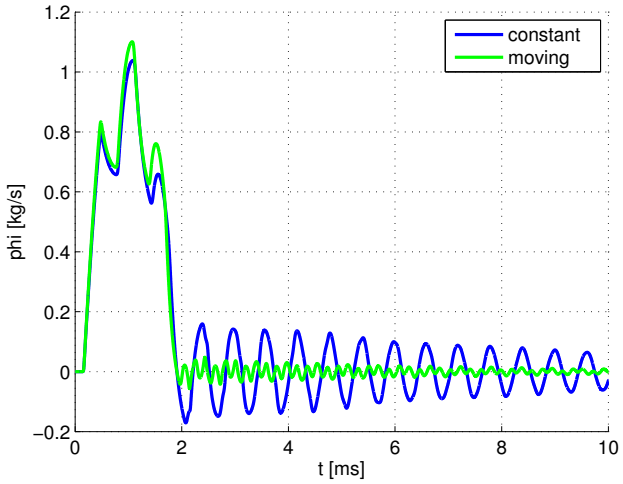

Figure 10: Simulation result constant velocity input vs. moving mesh

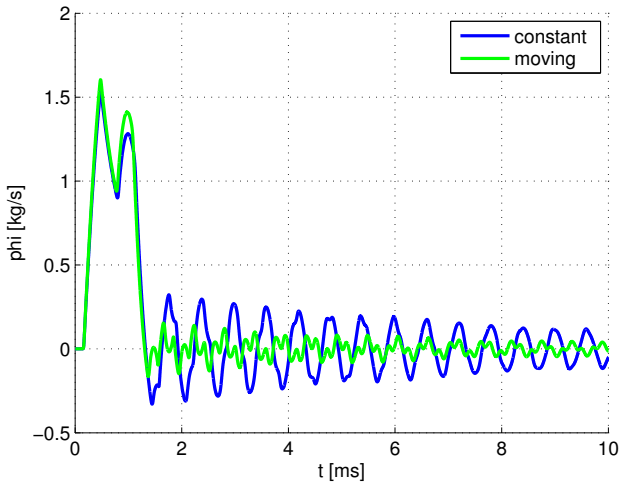

Figure 11: Simulation result constant velocity input vs. moving mesh

Thinking of $T = 10\,\mathrm{ms}$ this results in a number of wave reflections during the movement of the piston. A simulation result corresponding to this fact is shown in Fig. 8 and its fft in Fig. 9.

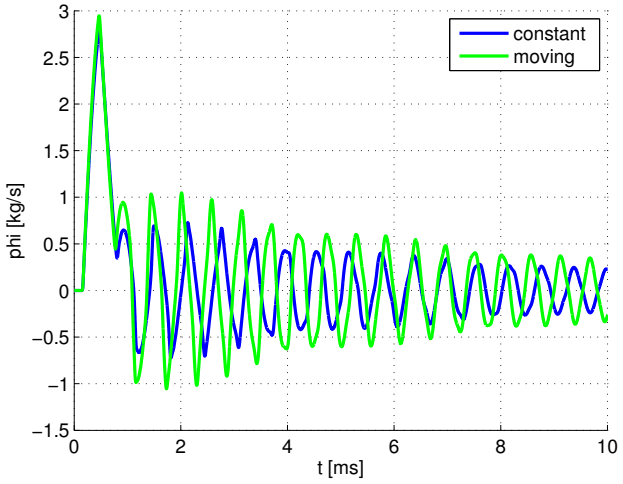

Figure 12: Simulation result constant velocity input vs. moving mesh

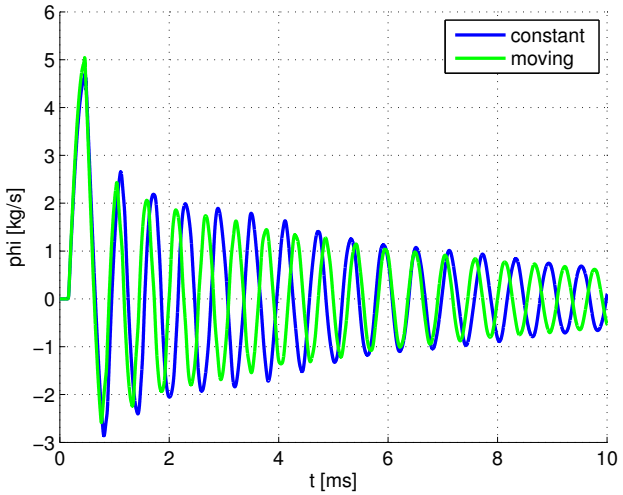

Figure 13: Simulation result constant velocity input vs. moving mesh

The blue lines represent the constant inflow and the green ones the moving piston boundaries. The only difference is the different travelling time of the waves due to the reduced length of the pipe. The vertical solid black lines in Fig. 9 represent the eigenfrequencies for a pipe with length L and the dashed line for a pipe with the length $L - s_0$. The eigenfrequencies can be calculated with

$$f_i = \frac{c_0}{\lambda_i}$$

where

$$\lambda_i = \frac{4 \cdot L}{i}$$

with $i = 1, 2, 3, ...$

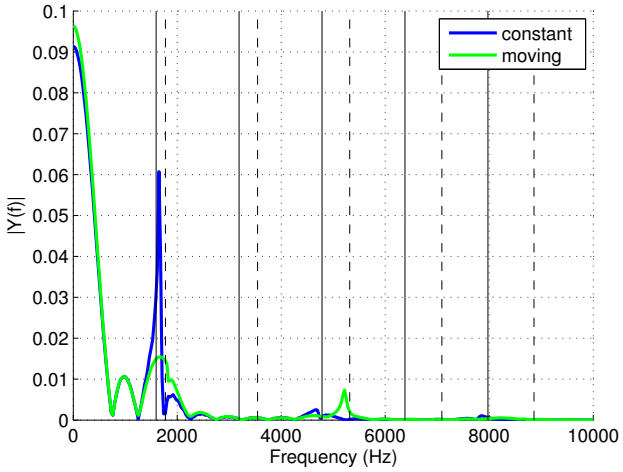

Figure 14: Simulation result constant velocity input vs. moving mesh

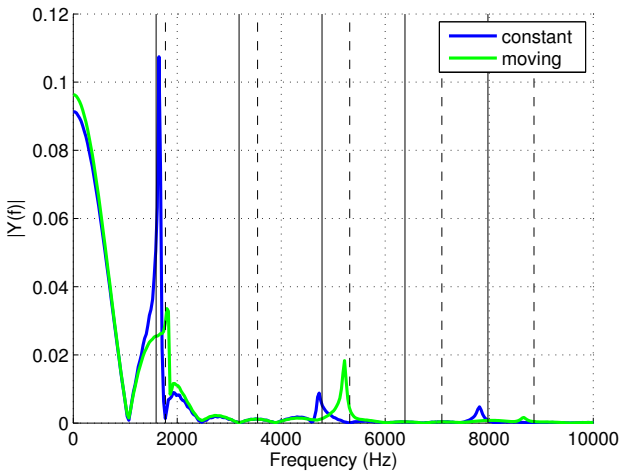

Figure 15: Simulation result constant velocity input vs. moving mesh

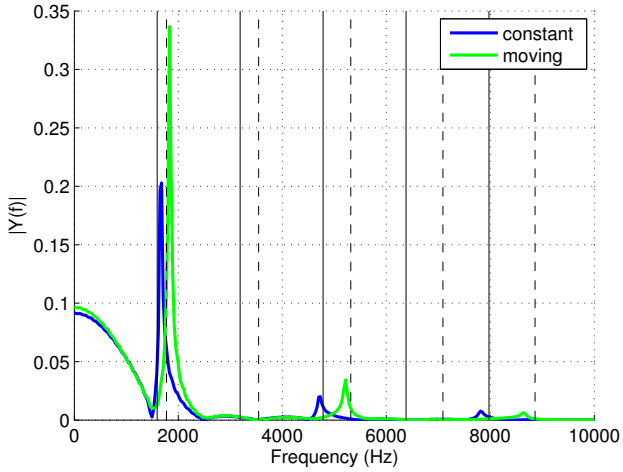

Figure 16: Simulation result constant velocity input vs. moving mesh

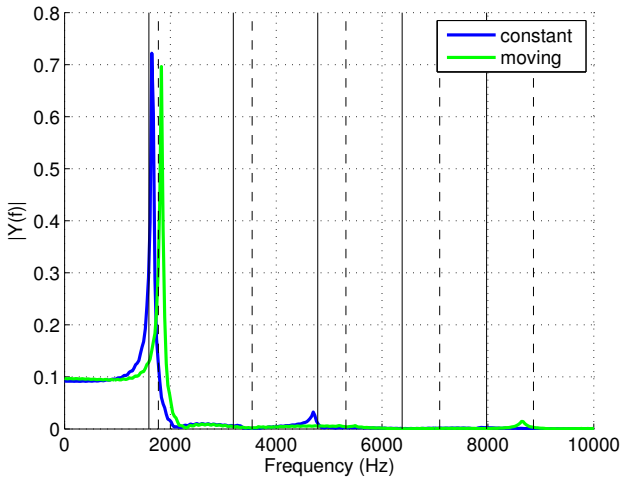

Figure 17: Simulation result constant velocity input vs. moving mesh

Fig. 10 shows the mass flow of a simulation result for $T = 2\,\text{ms}$. For the simulations corresponding to Fig. 11, 12 and 13 the value of T of the previous simulation has been devided by the factor of $\sqrt{2}$. The ffts in Fig. 14, 15, 16 and 17 match to the simulation results of the massflow. Reducing the process time is equivalent to increasing the velocity. Using $T = 2\,\text{ms}$ leads to a velocity of approximately $v = 15\,\frac{\text{m}}{\text{s}}$. These numbers result in a Reynolds number of

$$Re = \frac{u \cdot D}{\nu} \approx 2170.$$

Increasing the velocity further the fluid flow certainly reaches the turbulent region. For all these simulations turbulence models have been neglected. Reason for this is the short simulated timeframe. In this case turbulence has hardly chance to develop. For the purpose of confirming this statement, the results of a simulation carried out with a LES model compared to a simulation done without any turbulence model is shown in Fig. 26 and 27. The maximum of the velocity in this simulation reached $v = 30\,\frac{\text{m}}{\text{s}}$.

As it can be seen in Fig. 10 and 14, respectively in Fig. 11 and 15, there is much more damping for the first eigenfrequency in the case of the moving piston boundary as it is in the case for the uniform velocity input. For these simulations $T = 2\,ms$ and $T = \frac{2}{\sqrt{2}}\,\text{ms}$. Contrary, the amplitude for the third eigenfrequency is higher in the case of the moving boundary, but disappears for the case when $T = \frac{2}{\sqrt{8}}\,ms$, which can be seen in Fig. 17. In the third simulation (Fig. 16) the amplitude for the moving piston of the first eigenfrequency is higher. The more the dura-

tion of the impulse equals the wave propagation time the more the basic shape of the input function vanishes and the first eigenfrequency gets more and more prominent. In Fig. 12 and 13 as well as in all the ffts a shift in the eigenfrequencies due to the change of pipe length is visible.

In Fig. 18, 19, 20 and 21 the pressures at the end of the pipe is shown. As explained before the pressure is the average pressure of cells in between a cylinder with a diameter of d and the length of a cell layer. This cylinder starts at L. In the first two simulations with the moving piston boundary the pressure stood nearly on a constant level, while in the case of the uniform velocity already oscillations in the pressure can be seen.

The pressure on the moving piston, which is proportional to the force on the piston, can be seen in Fig. 22, 23, 24 and 25. Here a strong damping (of the first eigenfrequency) caused by the moving piston boundary also can be seen in the first two simulations. The pressure amplitude gets rather high, so that the pressure for the last two simulations even reaches negative values. These negative pressures are of course unphysical. In a physical system these low pressures would lead to cavitation.

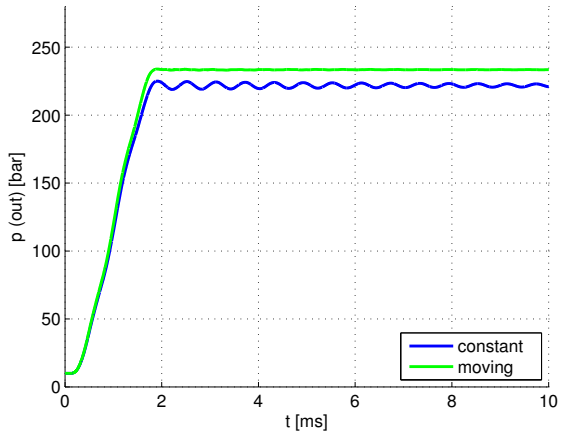

Figure 18: Simulation result constant velocity input vs. moving mesh

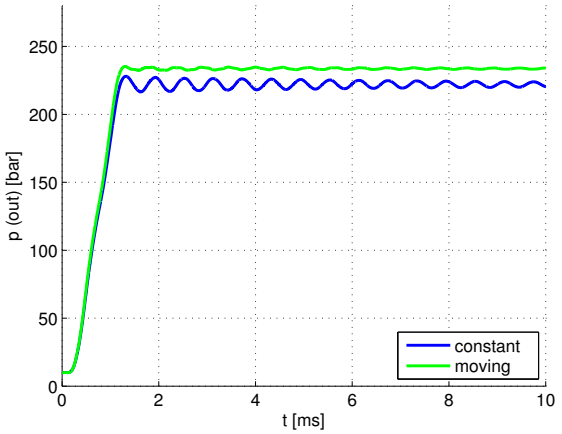

Figure 19: Simulation result constant velocity input vs. moving mesh

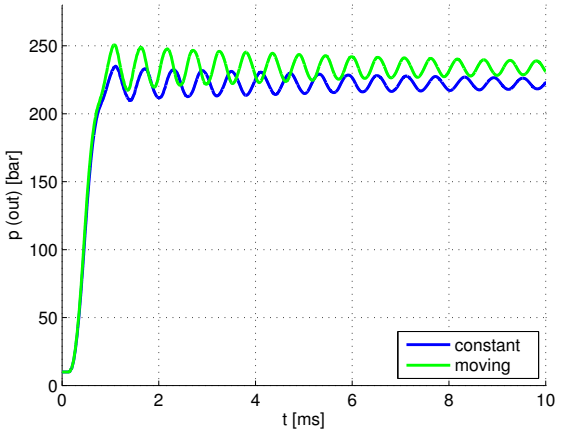

Figure 20: Simulation result constant velocity input vs. moving mesh

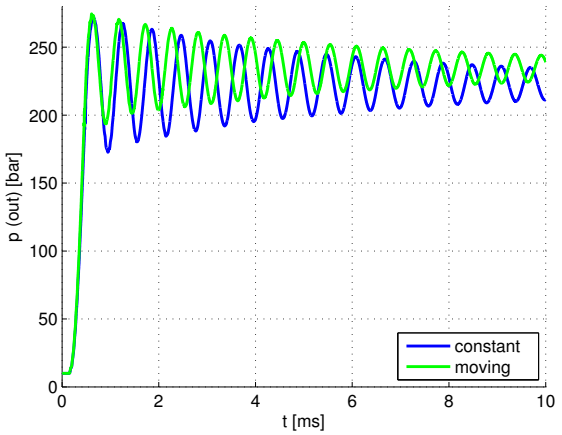

Figure 21: Simulation result constant velocity input vs. moving mesh

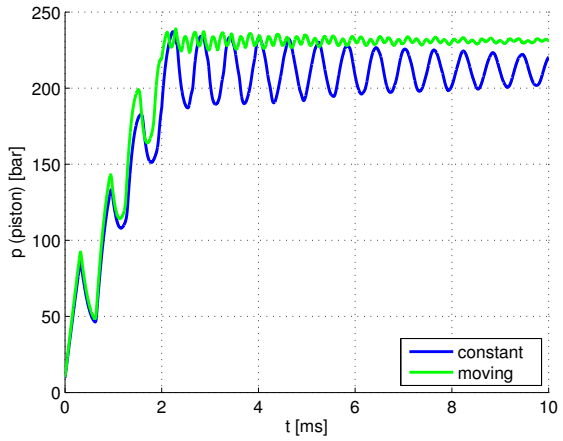

Figure 22: Simulation result constant velocity input vs. moving mesh

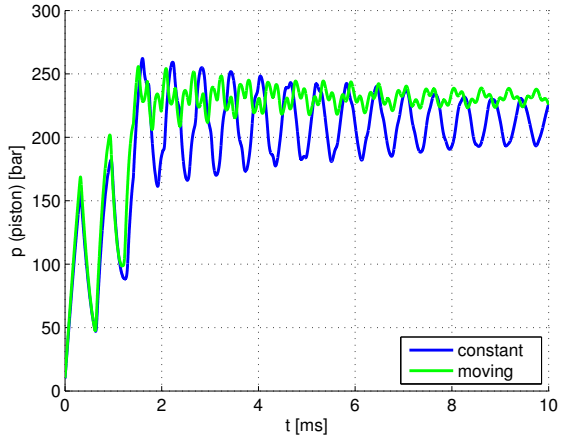

Figure 23: Simulation result constant velocity input vs. moving mesh

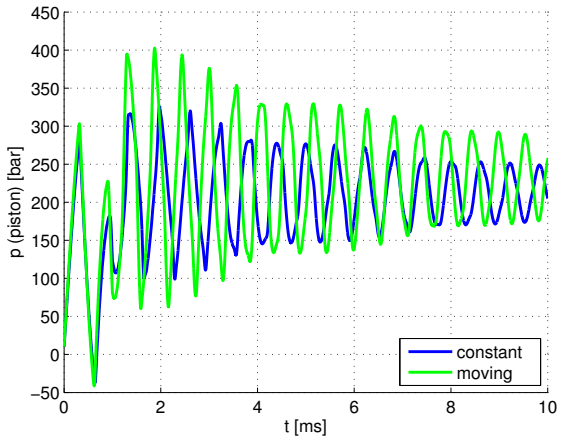

Figure 24: Simulation result constant velocity input vs. moving mesh

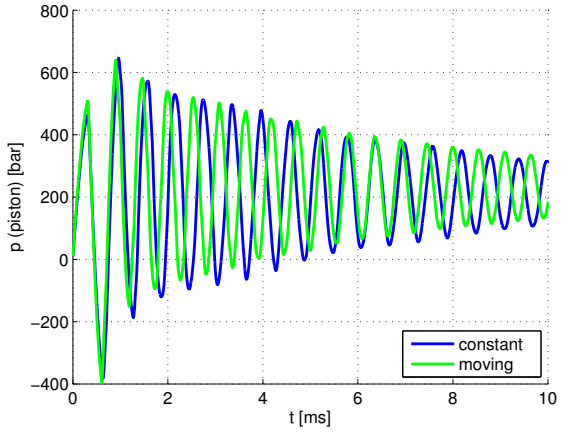

Figure 25: Simulation result constant velocity input vs. moving mesh

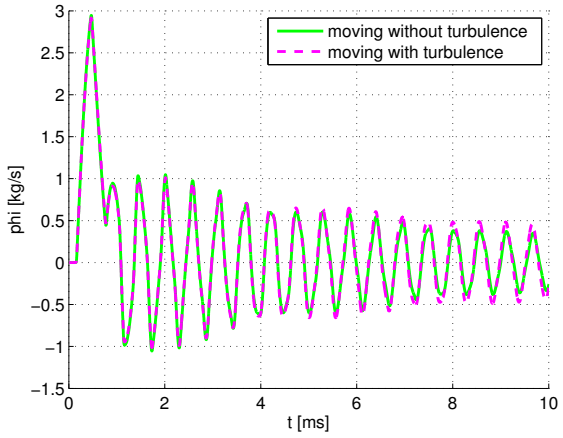

Figure 26: Simulation result with vs. without turbulence model

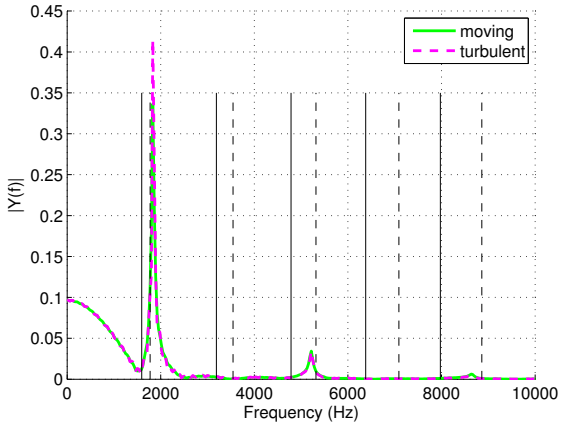

Figure 27: Simulation result with vs. without turbulence model

5 Conclusion

In this work two different boundary conditions have been compared. In the first case a usual constant velocity boundary condition has been taken for the inlet, whereas in the second case the velocity has been produced by the motion of a piston. Simulations with different inflow velocities, but equal mass inflow have been accomplished. The first simulations with rather low velocities compared to the wave propagation time showed - as expected - a shift in the excited eigenfrequencies caused by the change of length of the pipe. In the case of an excitation time in the order of the travelling time of a wave, the first eigenfrequency of the pipe is the most prominent one. A very interesting phenomenon occurs, if the massflow starts to rise again, while the input velocity converges to zero. Obviously, the final reduction of the velocity of the inflow compared with the rising massflow interact in that way, that the amplitude of the first eigenfrequency of the system is strongly diminished. Again there is this shift in the eigenfrequencies caused by the change of length of the pipe.

Acknowledgements: This work has been carried out at LCM GmbH as part of a K2 project. K2 projects are financed using funding from the Austrian COMET-K2 programme. The COMET K2 projects at LCM are supported by the Austrian federal government, the federal state of Upper Austria, the Johannes Kepler University and all of the scientific partners which form part of the K2-COMET Consortium.

References:

[1] B.Sc. Lukas Muttenthaler. Development of a hydraulic servo cylinder with an elastohydrostatic linear bearing. Master's thesis, JKU, 2015.

[2] http://www.openfoam.com/, 03 2014. URL `http://www.openfoam.com/`.

[3] J. H Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 2002.

[4] Hrvoje Jasak and Zeljko Tukovic. Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30(2):1–20, 2006.

[5] Christophe Kassiotis. Which strategy to move the mesh in the computational fluid dynamic code openfoam. Technical report, Laboratoire de Mcanique et Technologies, 2008.

[6] Yukio Hori. *Hydrodynamic Lubrication*. Springer, 2006.