



Delft university of technology
Faculty of electrical engineering, mathematics and computer science
Delft institute of applied mathematics

**Feasibility study of a tire hydroplaning simulation
in a finite element code using a coupled
Eulerian-Lagrangian method**

A thesis submitted to the
Delft institute of applied mathematics
in partial fulfillment of the requirements

for the degree

**MASTER OF SCIENCE
in
APPLIED MATHEMATICS**

by

ADRIAAN SILLEM

Delft, the Netherlands
October 2008



MSC THESIS APPLIED MATHEMATICS

”Feasibility study of a tire hydroplaning simulation in a finite element code using a coupled Eulerian-Lagrangian method”

Adriaan Sillem

Delft university of technology

Daily supervisors

Dr.ir. A. Böhmer
Dr.ir. V. Decouvreur

Other thesis committee members

Prof.dr.ir. D.J. Rixen
Dr. P. Wilders

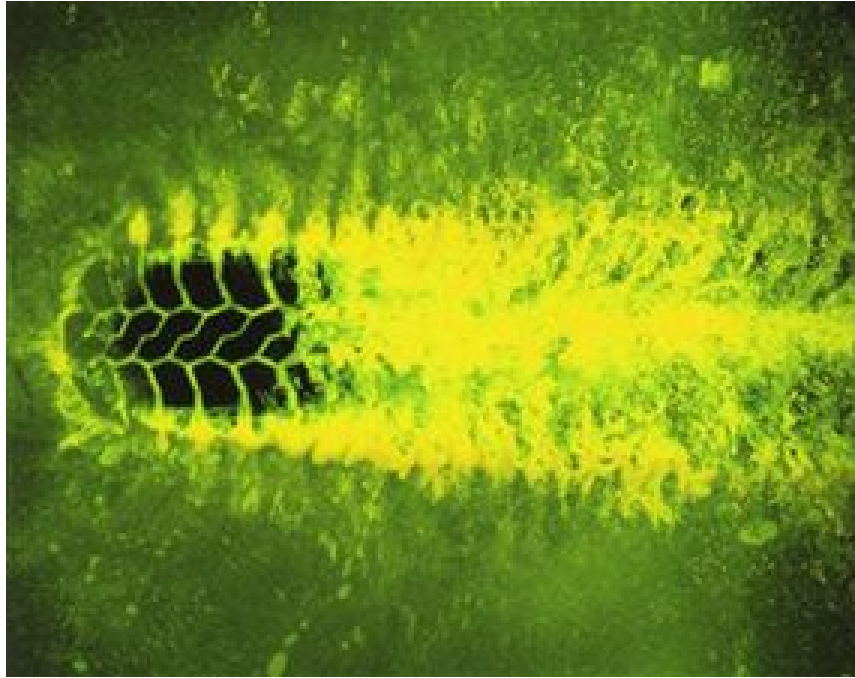
October 2008

Responsible professor

Prof.dr.ir. C. Vuik

Delft, the Netherlands

Feasibility study of a tire hydroplaning simulation in a finite element code using a coupled Eulerian-Lagrangian method
A. Sillem



Contents

List of figures	xii
List of tables	xiii
Preface	xv
Project	xv
Outline	xv
Acknowledgements	xv
I Theory	1
1 Hydroplaning	3
1.1 Description of hydroplaning	3
1.2 Interest	3
1.3 Definition of the problem	4
2 Discretisation mesh	7
2.1 Introduction	7
2.2 Eulerian formulation	7
2.3 Lagrangian formulation	7
2.4 Arbitrary Lagrangian-Eulerian formulation	8
2.5 Coupled Eulerian-Lagrangian formulation	8
3 Fluid dynamics	11
3.1 Introduction	11
3.2 Reference frame and scale	11
3.3 Conservation of mass	12
3.3.1 Continuum scale with fixed frame	12
3.3.2 Continuum scale with moving frame	12
3.3.3 Infinitesimal scale with fixed frame	13
3.3.4 Infinitesimal scale with moving frame	14
3.3.5 Equivalence of the equations	14
3.4 Conservation of momentum	16
3.5 Conservation of energy	17
3.5.1 Rate of work due to body and surface forces	17
3.5.2 Net flux of heat	18
3.5.3 Rate of change of energy	19
3.5.4 The energy equation	19
3.6 Additional relations	19
3.6.1 Shear stresses	19
3.6.2 Fourier's law	20
3.6.3 Equations of state	20
3.7 Navier-Stokes and Euler equations	20
3.7.1 Navier-Stokes equations	20
3.7.2 Euler equations	21
3.7.3 Initial and boundary conditions	21
3.7.4 Nondimensionalising the Navier-Stokes equations	21
3.8 Hydroplaning case	22
3.8.1 Incompressibility and isotropy	23
3.8.2 Newtonian viscous flow	23
3.8.3 Body force	24
3.8.4 Decoupled energy equation	24
3.8.5 Initial and boundary conditions	25
3.8.6 Nondimensionalising the incompressible Navier-Stokes equations	25

4	The tire	27
4.1	Introduction	27
4.2	Functions and components	27
4.2.1	Tread	27
4.2.2	Carcass	28
4.2.3	Belt package	29
4.3	Materials	29
4.4	Structure equations	30
4.4.1	Conservation of momentum	31
4.4.2	Initial and boundary conditions	32
5	Fluid-structure interaction	33
5.1	Introduction	33
5.2	Monolithic or partitioned	33
5.3	Coupling nonmatching meshes	34
5.4	Interface tracking	36
6	Finite element method	39
6.1	Introduction	39
6.2	Procedure	39
7	Abaqus' CEL method	41
7.1	Introduction	41
7.2	Mesh treatment	41
7.3	Fluid dynamics	41
7.3.1	Compressible or incompressible fluid?	41
7.3.2	Mie-Grüneisen with Hugoniot fit	42
7.3.3	Galerkin's weak formulation	43
7.4	Structure mechanics	44
7.4.1	Incompressibility for solids	44
7.4.2	Mooney-Rivlin	44
7.4.3	Galerkin's weak formulation	46
7.5	Fluid-structure interaction	46
7.6	Solving the coupled fluid and structure equations	47
7.7	Summary of Abaqus' CEL method	48
II	Test models	51
8	Couette flow	53
8.1	Introduction	53
8.2	Model	53
8.3	Analytical solution	54
8.3.1	Stationary case	54
8.3.2	Instationary case	55
8.4	Matlab solution	56
8.4.1	Handling assumptions	56
8.4.2	Spatial discretisation	56
8.4.3	Continuity equation	57
8.4.4	Implications of the initial and boundary conditions	57
8.4.5	Momentum equations	58
8.4.6	Final system	59
8.4.7	Solution	60
8.5	Abaqus solution	61
8.5.1	Formulation	62
8.5.2	Solution	63
8.6	Comparison	66

9	Channel flow interacting with elastic body	67
9.1	Introduction	67
9.2	Model	67
9.2.1	Fluid properties	67
9.2.2	Structure properties	68
9.2.3	Initial and boundary conditions	68
9.3	Matlab solution	68
9.3.1	Implications of the initial and boundary conditions	68
9.3.2	Solution channel flow	69
9.3.3	Solution channel flow with cube	73
9.4	Abaqus solution	77
9.4.1	Formulation channel flow	77
9.4.2	Solution channel flow	77
9.4.3	Solution channel flow with bump	84
9.4.4	Formulation channel flow with cube	84
9.4.5	Solution channel flow with cube	85
9.4.6	Formulation channel flow with cylinder	86
9.4.7	Solution channel flow with cylinder	86
9.4.8	Formulation channel flow with elastic body	93
9.4.9	Solution channel flow with elastic body	93
9.5	Comparison	96
10	Grosch wheel	97
10.1	Introduction	97
10.2	Formulation	97
10.3	Solution	98
11	Conclusions and recommendations	105
11.1	Introduction	105
11.2	Conclusions	105
11.3	Recommendations	106
III	Appendices	107
A	Abaqus	109
A.1	Introduction	109
A.2	Abaqus/CAE	109
A.3	Abaqus/Viewer	110
IV	Bibliography	113
	Bibliography	116

List of Figures

1.1	hydroplaning.	3
1.2	cross-section of a car tire.	3
1.3	motorcycle tire [4].	3
1.4	sprinkling the track for a hydroplaning test.	4
1.5	sideview hydroplaning [5].	5
2.1	plastic strain field of an impact of a rod on a wall, ALE meshed [33].	7
2.2	Eulerian mesh, plastic strain [33].	7
2.3	Lagrangian mesh, plastic strain [33].	8
2.4	ALE mesh, plastic strain [33].	8
2.5	CEL mesh in Abaqus, Lagrangian structure and Eulerian fluid [33].	9
3.1	differences in frame and scale [18].	11
3.2	model of an infinitesimal small element fixed in space [18].	13
3.3	different forms of the continuity equation [18].	15
3.4	forces in x direction [18].	16
3.5	energy fluxes in x direction [18].	18
4.1	tire components [1].	27
4.2	tire cross-section [1].	28
4.3	tire components [35].	28
4.4	Maxwell model [5].	30
4.5	Kelvin-Voigt model [5].	30
4.6	Standard Linear Solid model [5].	30
4.7	stress-strain cycle for rubber [17]. The ellipse represents a set of empirical measures.	30
4.8	forces in x direction.	31
4.9	tire-fluid boundary indicated in red [31].	32
5.1	fluid-structure interaction [33].	33
5.2	monolithic or partitioned [7].	34
5.3	a strong partition coupling scheme [9].	34
5.4	nonmatching meshes in 2D [9].	34
5.5	nonmatching meshes [9].	35
5.6	information transfer by means of projection [9].	36
5.7	interface reconstruction [19].	37
5.8	classes of interface reconstruction methods.	37
6.1	unstructured mesh of carbody [3].	39
7.1	inflation of an airbag [33].	41
7.2	relation between pressure p and density ρ	43
7.3	uniform hydrostatic pressure [32].	44
7.4	multiple surface normals per element.	47
7.5	leakage.	47
7.6	subsequent mesh and information treatment of the CEL method.	49
8.1	Couette flow [5].	53
8.2	instationary Couette flow for various times [18].	55
8.3	staggered grid [6].	56
8.4	eigenvalues system matrix Couette flow in complex plane with logarithmic scale.	61
8.5	$u, v[\frac{m}{s}]$ at different times.	62
8.6	Eulerian part for Couette flow, one element thick making it a two dimensional problem.	62
8.7	$V_1[\frac{m}{s}]$ at different times.	64
9.1	channel flow interacting with elastic body [36].	67
9.2	channel flow, a parabolic velocity profile.	71
9.3	analysis time and critical time step size channel flow.	72
9.4	channel flow with cube.	73
9.5	realistic velocity profile.	73

9.6	Kármán vortex street off the coast of Rishiri Island in Japan.	73
9.7	channel flow with cube.	75
9.8	analysis time and critical time step size, for both with and without cube.	76
9.9	fluid domain with BC symbols.	77
9.10	$V_1[\frac{m}{s}]$, $c_0 = 1483[\frac{m}{s}]$ at different times.	79
9.11	density $[\frac{kg}{m^3}]$, $c_0 = 1483[\frac{m}{s}]$ at different times.	80
9.12	$V_1[\frac{m}{s}]$, $c_0 = 1000[\frac{m}{s}]$ at different times.	81
9.13	Womersley velocity profile due to no slip conditions and compression effects.	82
9.14	density $[\frac{kg}{m^3}]$, $c_0 = 1000[\frac{m}{s}]$ at different times.	83
9.15	mesh distortion.	84
9.16	channel flow with bump.	85
9.17	critical element distorted mesh.	85
9.21	cylinder modeled by a Lagrangian body, mesh of domain.	86
9.18	cube modeled by a BC.	87
9.19	channel flow with Lagrangian cube, $V_1[\frac{m}{s}]$ at different times.	88
9.20	cylinder modeled as a no slip BC.	89
9.22	channel flow with cylinder BC, $V[\frac{m}{s}]$ at different times.	90
9.23	channel flow with Lagrangian cylinder, separation allowed, $V[\frac{m}{s}]$ at different times.	91
9.24	channel flow with Lagrangian cylinder, separation not allowed, $V[\frac{m}{s}]$ at different times.	92
9.25	mesh for the benchmark.	93
9.26	displacement of flag in FSI test 1.	93
9.27	Results from FSI2 and FSI3 [36].	94
9.28	displacement of right end of flag.	95
10.1	different rubber compounds assigned to different sections.	97
10.2	an example of parallel edges seeded equally.	97
10.4	measure points in red.	98
10.3	mesh Grosch wheel test model.	100
10.5	contact normal force for different loads and inflow velocities, at measure points as indicated in Figure 10.4.	101
10.6	contact shear force for different loads and inflow velocities, at measure points as indicated in Figure 10.4.	102
10.7	contact normal and shear force for the measurement points.	103
10.8	leakage as a result $EVF < 0.5$ at interface nodes.	103
A.1	Abaqus/CAE	109
A.2	Abaqus/Viewer	110

List of Tables

8.1	results Couette flow model in Matlab.	61
8.2	results Couette flow model in Abaqus/Explicit.	65
9.1	results channel flow model in Matlab.	70
9.2	results channel flow with cube model in Matlab.	74
9.3	results channel flow model.	78
9.4	results channel flow with bump model.	84
9.5	data of the channel flow with cylinder model.	87
9.6	required parameters for Abaqus.	93
9.7	displacement of left end of flag according to [36].	94
9.8	orders of displacement of left end of flag, for both Abaqus and [36].	94
9.9	data of the channel flow interacting with elastic body model.	94
10.1	contact normal and shear forces.	98
10.2	data of the Grosch wheel model.	99

Preface

Project

This is the master thesis for the degree of master of science in applied mathematics, faculty of electrical engineering, mathematics and computer science of the Delft university of technology. The graduation is under supervision of division numerical analysis and has a duration of nine months, rewarded with 42 European credits (EC) when completed successfully.

The graduation is carried out at the Goodyear technical centre Luxembourg (GTC*L) in Colmar-Berg. GTC*L is a part of Goodyear where computer simulations are done. Goodyear is prominent in the worldwide production of tires for cars, trucks, airplanes and earthmovers. The purpose is to make a feasibility study of a tire hydroplaning simulation in a finite element analysis (FEA) code using a coupled Eulerian-Lagrangian method. The concerning FEA code Abaqus is provided by Simulia.

The committee for examination of this master thesis consists of

prof.dr.ir. C. Vuik	(Delft university of technology),
prof.dr.ir. D.J. Rixen	(Delft university of technology),
dr. P. Wilders	(Delft university of technology),
dr.ir. V. Decouvreur	(Goodyear technical centre Luxembourg).

Outline

The purpose of this master thesis is to make a study of the feasibility of Abaqus for a tire hydroplaning simulation. Prerequisites for this study are knowledge about the mechanics of tires and fluids. This scope of disciplines each has its own research field. The mechanics of tires involve e.g. theory of hyperelasticity and rolling contact. The mechanics of fluids involve e.g. surface reconstruction and turbulence. A combination of both fields belongs to the fluid-structure interaction field, a research field gaining popularity the last decade. The mathematics to solve these problems, whereof the majority exists of numerical methods, is also of importance. These prerequisites are treated in part one, the theory part of the thesis. This part is divided into chapters explaining hydroplaning, discretisation, fluid and tire mechanics, fluid-structure interaction and Abaqus' CEL method.

When the prerequisites are mastered, one can continue to part two, which treats well known academic test problems like Couette flow and channel flow for the sake of benchmarking Abaqus. The last test model chapter of this part treats a simplified hydroplaning model. Moreover, a zip package containing all the *.cae and *.inp files to run the models is deposited at [2]. In this way it is possible for everyone in the possession of Abaqus 6.8-MNT or a higher version to run, or even improve, the models themselves for testing the feasibility of Abaqus' CEL method. The part is concluded with conclusions and recommendations.

The appendices part, part three, consists of explanations about the programs used of Simulia, namely Abaqus/CAE and Abaqus/Viewer. Part four merely contains the bibliography.

I have tried to write the thesis in a didactical manner, on the level of a master student applied mathematics. In line with the practical purposes of a commercial company, in this case GTC*L, the thesis does inevitably contain issues of mechanical engineering. Nevertheless, any person participating in a technical master study or more advanced, should be able to understand the thesis.

*Due to a secrecy agreement between the student of TU Delft and GTC*L, it was prohibited to publish certain details. Nevertheless, the censorship is minimised for the sake of completeness and readability of this published master thesis.*

Acknowledgements

I wish to thank all of my colleagues of the Goodyear technical centre Luxembourg stationed in Colmar-Berg. They have made sure that I had an educational and interesting environment. Both in engineering and cultural sense. The gentlemen of divisions simulation technology (ST), tire and vehicle mechanics (TVM) and the helpdesk: Pawan Handa, Tom Ebbott, Dany Assaker, Joel Schmitt, Olivier Rosset, Jean Dheur, Duc

Bao Doan, Patrick Hock, Edwin Deerenberg, Paul Joubert, Daniel Hu-Yongkang, Jean-Bernard Boudeux, Philippe Manne, Didier Quoirin. In particular I would like to thank the following. Ronald van Zelst for the many kilometers we jogged together, even then discussing engineering issues. My supervisor within GTC*L, namely Vincent Decouvreur. And Alexander Böhmer, my former supervisor within GTC*L, who in the meanwhile has left GTC*L.

I also want to thank Vincent Bouwman and Gertjan Kloosterman from Simulia Benelux, for helping me with the Abaqus errors encountered throughout the exploration of their product. In particular "my colleague in mathematics" Gertjan Kloosterman for the mathematical discussions on some of the fundamentals of Abaqus' algorithms.

From TU Delft I want to thank my two enthusiastic supervisors Kees Vuik and Daniel Rixen, and Sander van Zuijlen, Guus Segal and Fred Vermolen.

Last but not least, I would like to thank my parents for their financial support and in particular my father for the endless amount of engineering issues we discussed throughout my life. For sure, these discussions have improved my analytical thinking and motivated me for the fields of applied sciences, in particular applied mathematics and mechanical engineering.

Adriaan Sillem
October 2008, Colmar-Berg



Goodyear Technical Center Luxembourg in Colmar-Berg [1].

Part I
Theory

Chapter 1

Hydroplaning

1.1 Description of hydroplaning

Hydroplaning or aquaplaning by a road vehicle occurs when a layer of fluid builds up between the rubber tires of the vehicle and the road surface, leading to the loss of traction and thus preventing the vehicle from responding to control inputs such as steering, braking or accelerating. It becomes, in effect, an unpowered and unsteered sled.

Every vehicle function that changes the acceleration results in a load on the tires. Control of this load relies on the friction between the tire contact surface -also called the footprint- and the road surface.

The tread or grooves of a tire are designed to remove fluid from beneath the tire, providing friction with the road surface even in wet conditions. Hydroplaning occurs when a tire encounters more fluid than it can dissipate. Fluid pressure in front of the wheel forces a wedge of fluid under the leading edge of the tire, causing it to lift from the road. The tire then skates on a sheet of fluid with little, if any, direct road contact, resulting in loss of control.

Likelihood of hydroplaning increases with

- velocity: in less time the same amount of fluid has to be dissipated
- depth of fluid: more depth increases fluid volume to be dissipated
- viscosity and mass of fluid: results in inertia in the dissipation of the fluid
- tread wear: the grooves are less deep and thus the volume available for fluid storage/transportation is less
- tire-wideness: narrower tires are less vulnerable to hydroplaning because the vehicle weight is distributed over a smaller footprint, resulting in a greater ability for the tires to push fluid out of the way. Also, the volume of the fluid to be dissipated is smaller for narrower tires.

The last item is the reason that bicycles, motorcycles and similar vehicles with a semi-elliptic cross-section are less likely to hydroplane. But while a slide in a four-wheeled vehicle is correctable with practice, the same slide on a motorcycle will generally cause the rider to fall, with severe consequences. Thus, despite the relative lack of hydroplaning danger, motorcycle riders must be even more cautious because overall traction is reduced by wet roads.

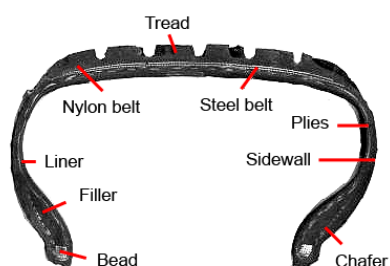


Figure 1.2: cross-section of a car tire.



Figure 1.3: motorcycle tire [4].



Figure 1.1: hydroplaning.

1.2 Interest

Predicting hydroplaning has been a challenge for the tire industry. The velocity at which a tire starts to hydroplane is a safety criterion. With retrospect, improvements concerning hydroplaning were obtained experimentally. This is an expensive process. A tread profile has to be designed. A mould design or laser

cutting should realise this profile. Afterwards it needs to be tested experimentally. Cost- and timewise, making improvements by simulation instead of empirism is more attractive.

Studies made on simulating hydroplaning are very rare. The results of the available studies are not useful. Because of this, it is only since the previous year that FEA program developers have made implementations that make the simulating of hydroplaning possible. Abaqus, Ansys and LS-Dyna are examples of these FEA programs. Still, these implementations are brand new and therefor need to be reviewed intensively before assuming an adequate solution is produced by them. On the issue of hydroplaning, GTC*L is one of the first to do such a review.

The program concerned by GTC*L is Simulia's Abaqus. Besides the regular in house code, Abaqus is also used for tire modelling issues like footprint size, deflection and rolling, involving both rim-tire and tire-road contact. Abaqus has proven to be efficient and reliable when it comes to structure calculations within GTC*L.



Figure 1.4: sprinkling the track for a hydroplaning test.

1.3 Definition of the problem

To judge which factors are of importance in the modelling of hydroplaning, we have to define the problem. Say we consider rubber rotating tire with a tread profile that enters a pond, cf. Figure 1.5. This sentence implies a lot in modelling language, explained roughly in the following.

Rubber is an elastomer and in general modeled as a viscoelastic and hyperelastic material, cf. 4.3. It is bonded to a carcass and belt package existing of anisotropic composites, cf. 4.3. The rotation of the tire implies moving boundaries between the tire, tarmac and the fluids water and air. The latter is negligible. The tire has a tread profile which needs to be meshed for the finite element method (FEM) used for discretisation, cf. chapter 6. The mesh can, depending on the reference frame, move with the material or be fixed in space, cf. chapter 2 and 3.2. This also holds for the fluid and the no slip tarmac. Fluid can also be discretised by the finite volume method (FVM) instead of the FEM. The gravitational load of the vehicle is transferred to the fluid by means of contact and to the internal energy of the rubber. Even simplified rolling contact has been a headcase for engineers [20]. Variation in the internal energy of rubber is accompanied with hysteresis effects, cf. 4.3. The load transfer to the fluid results in an increase of the pressure in the fluid. This transfer takes place over a moving boundary interface with on both sides different meshes. Depending on the tread profile, viscosity and density of the fluid is dissipated with a certain speed and direction.

So a seemingly simple sentence gives, in modelling language, raise to a complex problem.



Figure 1.5: sideview hydroplaning [5].

Summarising, the modelling of hydroplaning can be split up in correlated issues:

- viscoelastic and hyperelastic rubber,
- anisotropic composites,
- rolling contact,
- moving boundary/interface,
- fluid dynamics,
- discretisation with FEM,
- reference frame,
- mesh of tire,
- mesh of fluid,
- information exchange over nonmatching meshes,

where the first three belong to tire mechanics, the two thereafter are specific to hydroplaning and the latter five are issues of numerical theory. All items are treated more detailed in the subsequent chapters.

This characterisation of hydroplaning implies it being an element of the class of fluid-structure interaction (FSI) problems. In this class of problems, tracking of the interface and transfer of information over the interface are focal points. FSI and its link to hydroplaning is explained extensively in chapter 5.

Chapter 2

Discretisation mesh

2.1 Introduction

Simulating hydroplaning requires attention in the mesh generation process of both the fluid and the tire. Hydroplaning is a dynamic process involving large gradients of the variables. There are several reference frames or coordinate systems to be considered for mesh forming. Extent of deformation and movement are factors to be taken into account. Each frame has its own merits.

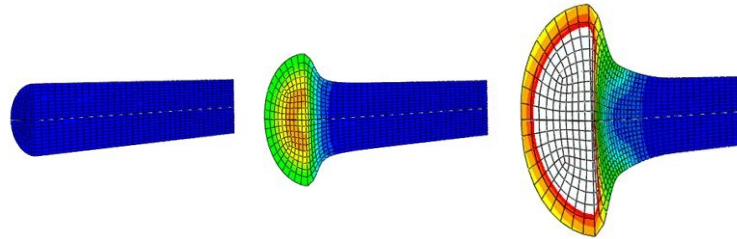


Figure 2.1: plastic strain field of an impact of a rod on a wall, ALE meshed [33].

2.2 Eulerian formulation

An Eulerian frame is fixed in space. The considered domain is divided into elements. With deformation and movement, materials flow through the elements. Where high gradients are expected, a fine mesh is needed, see Figure 2.2. Advantages of the Eulerian approach are that the mesh does not change per time step and that large deformations of the rod do not increase the needed computation power. For this reason, the Eulerian frame is used widely in fluid dynamics. A disadvantage is that an interface is not tracked accurately, which is of importance with FSI problems.

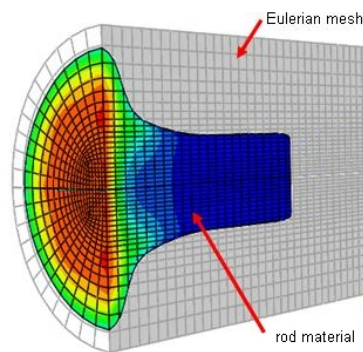


Figure 2.2: Eulerian mesh, plastic strain [33].

2.3 Lagrangian formulation

A Lagrangian frame moves with the material. In this way the interface of the material is tracked precisely, see Figure 2.3. Large deformations however, lead to mesh tangling implying a less precise solution. Remeshing is needed in this case, costing computational power. When considering materials with high elastic or Young's modulus, deformations are relatively smaller and the Lagrangian frame is attractive. For this reason, the Lagrangian frame is used widely in structure mechanics.

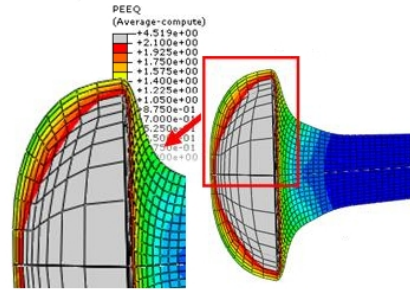


Figure 2.3: Lagrangian mesh, plastic strain [33].

2.4 Arbitrary Lagrangian-Eulerian formulation

Because of the shortcomings of purely Lagrangian and purely Eulerian descriptions, a technique has been developed that succeeds, to a certain extent, in combining the best features of both the Lagrangian and the Eulerian approaches. Such a technique is known as the arbitrary Lagrangian-Eulerian (ALE) description. In the ALE description, the nodes of the computational mesh may be moved with the continuum in normal Lagrangian fashion, or be held fixed in Eulerian manner, or be moved in some arbitrary specified way to give a continuous rezoning capability, see Figures 2.1 and 2.4. Because of this freedom in moving the computational mesh offered by the ALE description, greater distortions of the continuum can be handled than would be allowed by a purely Lagrangian method, with more resolution than that offered by a purely Eulerian approach. The mesh follows the boundary. However, this freedom in mesh movement has its limits. A treatment of mesh quality metrics and its limitations can be found in [9] and an extensive description of ALE methods can be found in [16].

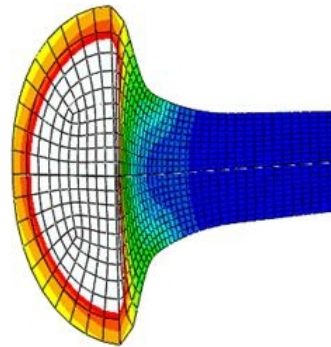


Figure 2.4: ALE mesh, plastic strain [33].

2.5 Coupled Eulerian-Lagrangian formulation

The coupled Eulerian-Lagrangian (CEL) method also attempts to capture the strengths of the Lagrangian and Eulerian methods. In general, a Lagrangian frame is used to discretise the moving structure while an Eulerian frame is used to discretise the fluid domain. The boundary of the Lagrangian domain is taken to represent the interface between the different domains. Interface models use the velocity of the Lagrangian boundary as a kinematic constraint in the Eulerian calculation and the stress from the Eulerian cell to calculate the resulting surface stress on the Lagrangian domain [12]. Different CEL algorithms may be characterized by the details of how this interface condition is treated [21].

As the name clearly states, Abaqus' CEL method uses a CEL approach, cf. Figure 2.5. This method is used for the class of FSI problems, which involves large deformations. At this point it is improper to explain the theory behind Abaqus' CEL because it is inherent with fluid dynamics and fluid-structure interaction, to be treated in chapters 3 and 5. Therefore we will revisit the theory of Abaqus' CEL method at the end of the theory part in chapter 7.

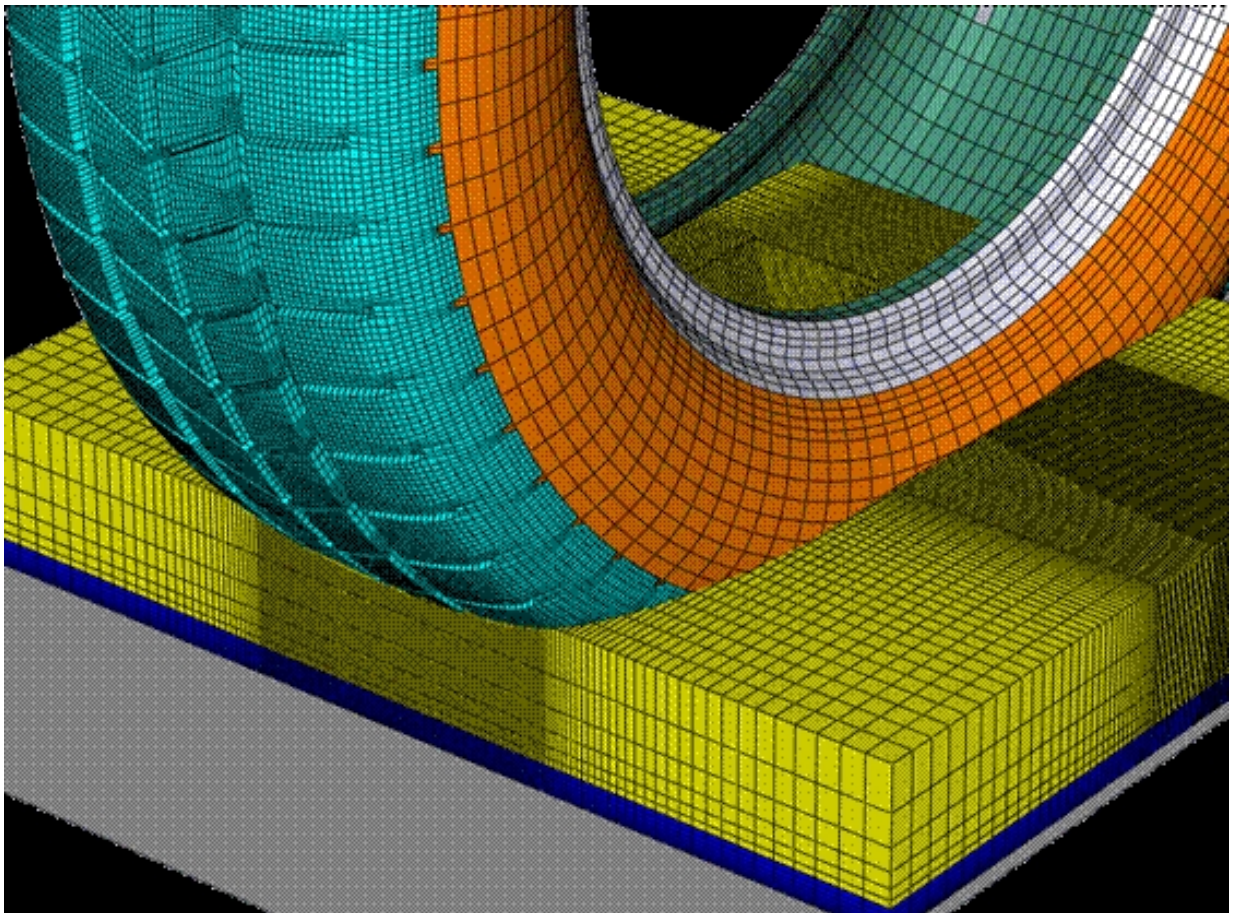


Figure 2.5: CEL mesh in Abaqus, Lagrangian structure and Eulerian fluid [33].

Chapter 3

Fluid dynamics

3.1 Introduction

Fluid dynamics is governed by three fundamental equations: the continuity, momentum and energy equation. They are the mathematical statements of three fundamental principles upon which all of fluid dynamics is based.

- Continuity from mass conservation.
- Momentum from Newton's second law $\mathbf{F} = m\mathbf{a}$.
- Energy equation from conservation of energy.

The deduction of these equations differs per reference frame and scale, which is shown in 3.3 for the conservation of mass. Although the reference frame and scale might differ, equivalence still holds, as shown in 3.3.5.

Considering hydroplaning, assumptions for the flow are made, changing the governing equations. Initial and boundary conditions are explained.

3.2 Reference frame and scale

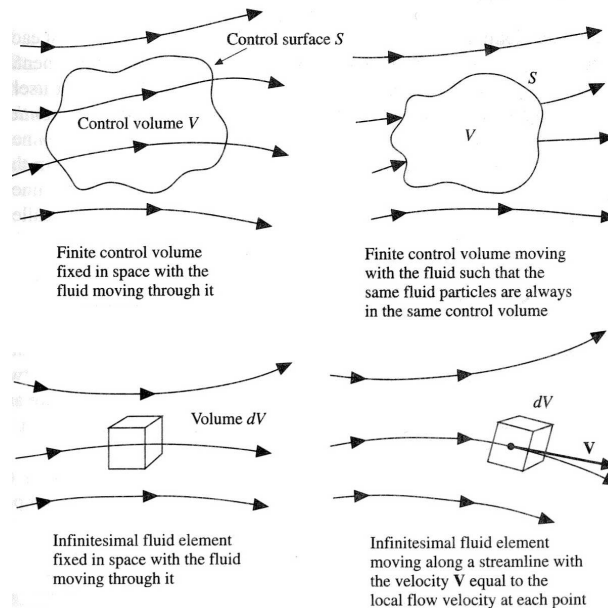


Figure 3.1: differences in frame and scale [18].

Problems in fluid dynamics can be considered on different scales and with different reference frames. The scale can be either continuum or infinitesimal. The reference frame can be either fixed in space or moving with the fluid (Figure 3.1), called the Eulerian and Lagrangian perspective respectively. The corresponding deduced equations are to be called in conservation and nonconservation form respectively. The latter denomination merely comes from the tradition of stating conservation laws relative to a fixed or Eulerian frame. In case of a Lagrangian domain a term is comprised in the substantial derivative (see 3.3.2), which makes it nonconservative. The fluid-flow equations that we directly obtain by applying the fundamental physical principles to a finite control volume are in integral form. These integral forms of the governing equations can be manipulated to obtain partial differential equations. This is shown in 3.3.5.

3.3 Conservation of mass

3.3.1 Continuum scale with fixed frame

Consider a control volume of arbitrary shape and of finite size fixed in space (Figure 3.1 upperleft). The fluid moves through the volume, across the surface. Mass conservation can in this case be described by

$$\begin{array}{lcl} \text{net mass flow out} & & \text{time rate of} \\ \text{of control volume} & = & \text{decrease of mass} \\ \text{through surface} & & \text{inside control volume} \end{array} .$$

The mass flow of a moving fluid across any fixed surface is equal to (density) \times (area of surface) \times (component of velocity perpendicular to the surface). Notice that the density is a function of space and time, so $\rho = \rho(x, y, z, t)$ in $[\frac{kg}{m^3}]$. In formula we have for the net mass flow out of the control volume through surface $\partial V = S \subseteq \mathbb{R}^3$ in $[m^2]$

$$\iint_S \rho \mathbf{v} \cdot \mathbf{n} dS$$

with $\mathbf{v}, \mathbf{n} \in \mathbb{R}^3$ the velocity in $[\frac{m}{s}]$ and normal to the closed surface S respectively.

The time rate of decrease of mass inside volume $V \subseteq \mathbb{R}^3$ in $[m^3]$ is

$$-\frac{\partial}{\partial t} \iiint_V \rho dV.$$

It follows that

$$\iint_S \rho \mathbf{v} \cdot \mathbf{n} dS = -\frac{\partial}{\partial t} \iiint_V \rho dV$$

or

$$\iint_S \rho \mathbf{v} \cdot \mathbf{n} dS + \frac{\partial}{\partial t} \iiint_V \rho dV = 0. \quad (3.1)$$

Equation (3.1) is the integral form of mass conservation. The finite aspect of the control volume is the cause of the integral form.

3.3.2 Continuum scale with moving frame

Consider the model in Figure 3.1 in the upperright corner. Because the element is moving with the fluid, mass conservation is in this case

$$\frac{D}{Dt} \iiint_V \rho dV = 0. \quad (3.2)$$

One might wonder why the derivative is suddenly written as $\frac{D}{Dt}$ instead of the regular $\frac{\partial}{\partial t}$? This is done to emphasise the difference of the space depending on time in this model, namely $\rho = \rho(x(t), y(t), z(t), t)$. This was not the case before. It is the differentiation operator in time considering that during the time interval Dt the particles considered remain the same, even if they are moving as in the current model. With $x_1 = x$, $x_2 = y$ and $x_3 = z$ in $[m]$ as the principal directions, denote

$$\begin{aligned} \frac{D}{Dt} &= \left. \frac{\partial}{\partial t} \right|_{\text{given particle}} \\ &= \left. \frac{\partial}{\partial t} \right|_{\text{given space position}} + \sum_{i=1}^3 \left. \frac{\partial x_i}{\partial t} \right|_{\text{given particle}} \frac{\partial}{\partial x_i} \end{aligned} \quad (3.3)$$

called the material or substantial derivative. For the density we would get

$$\begin{aligned} \frac{D}{Dt} \rho(x(t), y(t), z(t), t) &= \left. \frac{\partial}{\partial t} \rho(x(t), y(t), z(t), t) \right|_{\text{given particle}} \\ &= \frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \rho}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial \rho}{\partial z} \frac{\partial z}{\partial t} \\ &= \frac{\partial \rho}{\partial t} + \nabla \rho \cdot \mathbf{v}. \end{aligned}$$

Note that this can be seen as the chain rule.

3.3.3 Infinitesimal scale with fixed frame

Consider the model in Figure 3.1 in the lowerleft corner. Analogous to the model with a finite size and fixed frame, recall that

$$\begin{array}{lcl} \text{net mass flow out} & & \text{time rate of} \\ \text{of control volume} & = & \text{decrease of mass} \\ \text{through surface } S & & \text{inside control volume} \end{array} .$$

The net mass flow out is equal to (density) \times (area of surface) \times (component of velocity perpendicular to the surface). Because the infinitesimal element is a cube, considering directions is simplified compared to the continuum scale. Normal \mathbf{n} is superfluous, considering the x , y and z direction separately is sufficient. Net outflow in x direction is

$$(\rho u)|_{x+dx} dy dz - (\rho u)|_x dy dz$$

Taylor expansion of $(\rho u)|_{x+dx}$ around x gives

$$(\rho u)|_{x+dx} = \sum_{n=0}^{\infty} \frac{(dx)^n}{n!} \frac{\partial^n (\rho u)}{\partial x^n} \Big|_x .$$

Because dx is infinitesimal, we can neglect the higher order terms to get $(\rho u)|_x + dx \frac{\partial(\rho u)}{\partial x} \Big|_x$. Herewith, the net outflow in x direction becomes (cf. Figure 3.2)

$$\begin{aligned} (\rho u)|_{x+dx} dy dz - (\rho u)|_x dy dz &= \left[(\rho u)|_x + dx \frac{\partial(\rho u)}{\partial x} \Big|_x \right] dy dz - (\rho u)|_x dy dz \\ &= \frac{\partial(\rho u)}{\partial x} \Big|_x dx dy dz . \end{aligned}$$

Analogous we have in the y and z direction

$$\begin{aligned} (\rho v)|_{y+dy} dx dz - (\rho v)|_y dx dz &= \frac{\partial(\rho v)}{\partial y} \Big|_y dx dy dz \\ (\rho w)|_{z+dz} dx dy - (\rho w)|_z dx dy &= \frac{\partial(\rho w)}{\partial z} \Big|_z dx dy dz . \end{aligned}$$

Furthermore

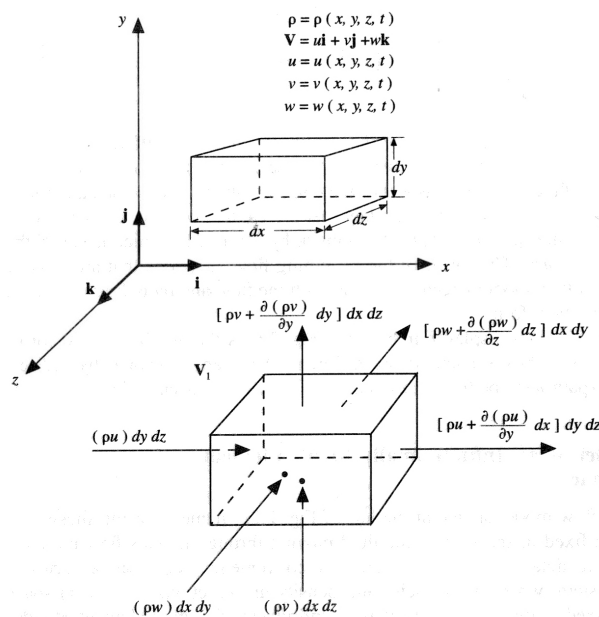


Figure 3.2: model of an infinitesimal small element fixed in space [18].

$$\text{time rate of mass decrease} = -\frac{\partial \rho}{\partial t} dx dy dz .$$

Notice that the density ρ can be assumed constant in space as the element is infinitesimal.

Hence, we get

$$\begin{aligned} \left(\frac{\partial(\rho u)}{\partial x} \Big|_x + \frac{\partial(\rho v)}{\partial y} \Big|_y + \frac{\partial(\rho w)}{\partial z} \Big|_z \right) dx dy dz &= -\frac{\partial \rho}{\partial t} dx dy dz \\ \left(\frac{\partial(\rho u)}{\partial x} \Big|_x + \frac{\partial(\rho v)}{\partial y} \Big|_y + \frac{\partial(\rho w)}{\partial z} \Big|_z \right) &= -\frac{\partial \rho}{\partial t} \end{aligned}$$

leading to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0. \quad (3.4)$$

3.3.4 Infinitesimal scale with moving frame

Consider the model in Figure 3.1 in the lower-right corner. An infinitesimal fluid element moving with the flow. This element has a fixed mass, but in general its shape and volume will change as it moves downstream. Denote the fixed mass and variable volume of this element by δm and δV respectively:

$$\delta m = \rho \delta V.$$

Analogous with the continuum scale case, the rate of change of the mass in time is equal to zero. The element is moving with the fluid so we again use the notation of the material derivative:

$$\begin{aligned} \frac{D(\delta m)}{Dt} &= 0 \\ \frac{D(\rho \delta V)}{Dt} &= 0 \\ \delta V \frac{D\rho}{Dt} + \rho \frac{D(\delta V)}{Dt} &= 0 \\ \frac{D\rho}{Dt} + \frac{\rho}{\delta V} \frac{D(\delta V)}{Dt} &= 0 \end{aligned}$$

From the physical meaning of the divergence we have $\frac{1}{\delta V} \frac{D(\delta V)}{Dt} = \nabla \cdot \mathbf{v}$ [18]. Thus

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0. \quad (3.5)$$

3.3.5 Equivalence of the equations

We deduced equations (3.1), (3.2), (3.4) and (3.5); these are four equations either in integral or partial differential form, for either the conservative or nonconservative case (cf. Figure 3.3). But in a way, they are all equivalent. An important difference is that the integral forms allow the presence of discontinuities inside the fixed control volume, whereas the partial differential forms assume first order differentiability, implying continuity. The integral form is therefore considered to be more fundamental than partial differential form.

We will deduce (3.5) from (3.4), (3.4) from (3.1) and (3.1) from (3.2) to prove that every equation can be rewritten in each of the remaining three equations. So we start with

$$\frac{D}{Dt} \iiint_V \rho dV = 0.$$

Since the material derivative itself represents a time rate of change associated with a moving element and the limits on the volume integral in (3.2) are determined by these same moving elements, the material

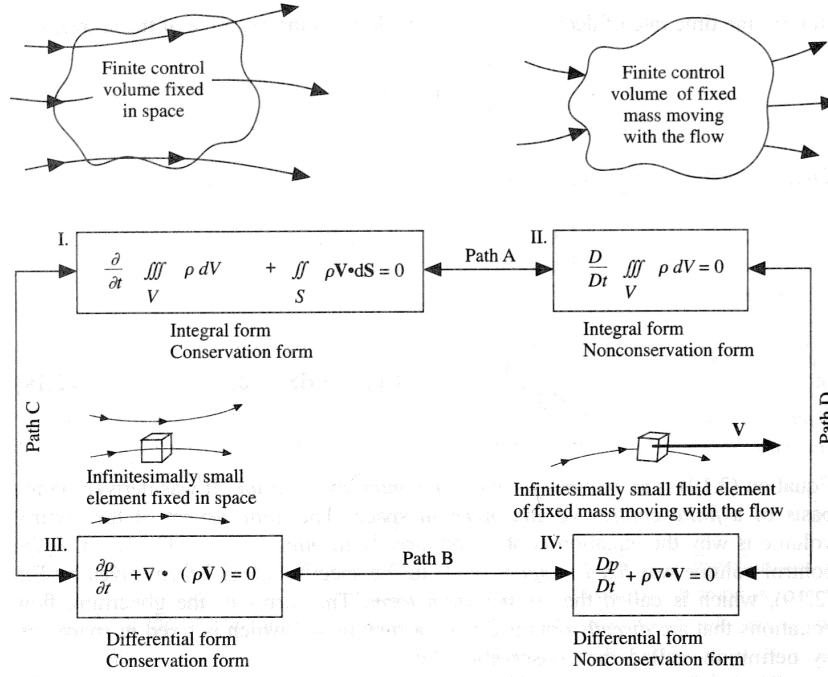


Figure 3.3: different forms of the continuity equation [18].

derivative can be taken inside the integral.

$$\begin{aligned}
 \iiint_V \frac{D(\rho dV)}{Dt} &= 0 && \text{product-rule} \\
 \iiint_V \frac{D\rho}{Dt} dV + \iiint_V \rho \frac{D(dV)}{Dt} &= 0 && \text{multiply by } \frac{dV}{dV} = 1 \\
 \iiint_V \frac{D\rho}{Dt} dV + \iiint_V \rho \left[\frac{1}{dV} \frac{D(dV)}{Dt} \right] dV &= 0 && \text{equivalence } \frac{1}{\delta V} \frac{D(\delta V)}{Dt} = \nabla \cdot \mathbf{v} \\
 \iiint_V \frac{D\rho}{Dt} dV + \iiint_V \rho \nabla \cdot \mathbf{v} dV &= 0 && (3.3) \\
 \iiint_V \left[\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho \right] dV + \iiint_V \rho \nabla \cdot \mathbf{v} dV &= 0 \\
 \iiint_V \left[\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} \right] dV &= 0 && \text{product-rule} \\
 \iiint_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) \right] dV &= 0 && \text{divergence theorem} \\
 \iiint_V \frac{\partial \rho}{\partial t} dV + \iint_S \rho \mathbf{v} \cdot \mathbf{n} dS &= 0 && \mathbf{v} \text{ independent of } t \\
 \frac{\partial}{\partial t} \iiint_V \rho dV + \iint_S \rho \mathbf{v} \cdot \mathbf{n} dS &= 0
 \end{aligned}$$

The last equation corresponds with (3.1).

A proof of the divergence theorem $\iiint_V \nabla \cdot \mathbf{v} dV = \iint_S \mathbf{v} \cdot \mathbf{n} dS$, $\mathbf{v} \in \mathbb{R}^3$ can be found in [27].

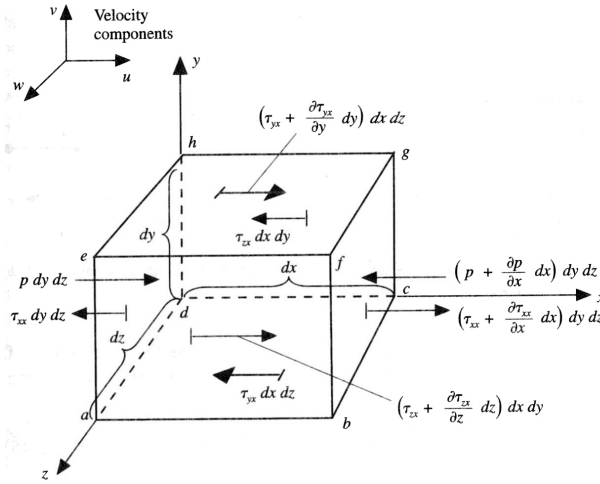
Along the deduction we encountered $\iiint_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) \right] dV = 0$. Since the volume is arbitrarily drawn in space, the only way for the integral to be equal to zero is if

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

which corresponds with (3.4).

The equivalence of (3.4) and (3.5) follows from application of the product and chain rules:

$$\begin{aligned}
 \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) &= 0 \\
 \frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} &= 0 \\
 \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} &= 0.
 \end{aligned}$$

Figure 3.4: forces in x direction [18].

3.4 Conservation of momentum

Newton's second law states $\mathbf{F} = m\mathbf{a}$. When considering a moving infinitesimal fluid element, it can experience body and/or surface forces.

- Body forces, which act directly on the volumetric mass of the fluid element. These forces "act at a distance"; examples can be gravitational and electromagnetic forces.
- Surface forces, which act directly on the surface of the fluid element. They are only due to two sources.
 - The pressure distribution acting on the surface, imposed by the outside fluid surrounding the fluid element.
 - The shear and normal stress distributions acting on the surface, also imposed by the outside fluid by means of friction.

Let \mathbf{f} denote the body force per unit mass in $[\frac{N}{kg}]$, then

$$\text{body force on element} = \rho \mathbf{f} dx dy dz.$$

From Figure 3.4 it follows that

$$\begin{aligned} \text{net surface force in } x \text{ direction} &= \left[p - \left(p + \frac{\partial p}{\partial x} dx \right) \right] dy dz + \left[\left(\tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} dx \right) - \tau_{xx} \right] dy dz \\ &+ \left[\left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy \right) - \tau_{yx} \right] dx dz + \left[\left(\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} dz \right) - \tau_{zx} \right] dx dy \\ &= \left[-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right] dx dy dz \end{aligned}$$

Note that the first index of stress τ in $[Pa]$ denotes the orientation of the surface normal and the second the direction of the stress. The pressure p is in $[Pa]$. The $\star + \frac{\partial \star}{\partial \{x,y,z\}} d\{x,y,z\}$ terms are again from Taylor expansions, analogous with the deduction in 3.3.3. In fact, this method is used throughout all deductions on infinitesimal scale.

An important notification is the vanishing of the original term p , leaving only the gradient term of p . As a result, the pressure variation is solved by the equation, not the absolute pressure. It is, in other words, referenceless. This problem will arise later on in II. Nevertheless, one speaks, in short, of the pressure and not of the pressure variation.

With analogy in y and z direction, the total force becomes

$$\begin{aligned}
\mathbf{F} &= \left[\rho \mathbf{f} - \nabla p + \begin{pmatrix} \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} \\ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \end{pmatrix} \right] dx dy dz \\
&= \left[\rho \mathbf{f} - \nabla p + \left\{ \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{pmatrix} \underbrace{\begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{yz} & \tau_{zz} \end{pmatrix}}_{=\boldsymbol{\tau}} \right\}^T \right] dx dy dz \\
&= [\rho \mathbf{f} - \nabla p + (\nabla^T \boldsymbol{\tau})^T] dx dy dz \\
&= [\rho \mathbf{f} - \nabla p + (\nabla \cdot \boldsymbol{\tau})^T] dx dy dz
\end{aligned}$$

Irrespective to matrix dimensions, the last term $(\nabla \cdot \boldsymbol{\tau})^T$ is often written as $\nabla \cdot \boldsymbol{\tau}$, the divergence of a matrix. Matrix $\boldsymbol{\tau}$ is also often denoted as \mathbf{S} or $\boldsymbol{\sigma}_{\text{dev}}$. Throughout this report, the notation of $\boldsymbol{\sigma}_{\text{dev}}$ shall be maintained.

With $\mathbf{F} = m\mathbf{a}$ and noting that $m = \rho dx dy dz$, $\mathbf{a} = \frac{D\mathbf{v}}{Dt}$ and for the stress tensor $\boldsymbol{\sigma}_{\text{dev}} = \boldsymbol{\sigma}_{\text{dev}}^T$ we get

$$\begin{aligned}
[\rho \mathbf{f} - \nabla p + \nabla \cdot \boldsymbol{\sigma}_{\text{dev}}] dx dy dz &= \rho \frac{D\mathbf{v}}{Dt} dx dy dz \\
\rho \mathbf{f} - \nabla p + \nabla \cdot \boldsymbol{\sigma}_{\text{dev}} &= \rho \frac{D\mathbf{v}}{Dt}
\end{aligned} \tag{3.6}$$

which are the Navier-Stokes equations in nonconservation form. Furthermore, in $\mathbf{a} = \frac{D\mathbf{v}}{Dt}$ the material derivative is used because the velocity \mathbf{v} can be relative to a fixed or moving frame.

Writing out the material derivative gives the conservation form

$$\rho \mathbf{f} - \nabla p + \nabla \cdot \boldsymbol{\sigma}_{\text{dev}} = \rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) \tag{3.7}$$

where the notation $\mathbf{v} \cdot \nabla = u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z}$ is a scalar operator called the advective operator. This is with slight abuse of notation because an inner-product, and thus also the standard inner-product \cdot , is commutative by definition [26]. By definition $\mathbf{v} \cdot \nabla = \nabla \cdot \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \neq u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z} = \mathbf{v} \cdot \nabla$. This abuse is a result of the fact that ∇ is a vector operator - a vector whose elements are operators - and not a vector. Note that the term $(\mathbf{v} \cdot \nabla) \mathbf{v}$ is nonlinear, which makes it more difficult to solve.

The terms $-\nabla p + \nabla \cdot \boldsymbol{\sigma}_{\text{dev}}$ are referred to as the divergence of the Cauchy stress tensor $\boldsymbol{\sigma}$, also written as

$$\begin{aligned}
-\nabla p + \nabla \cdot \boldsymbol{\sigma}_{\text{dev}} &= -\nabla \cdot p\mathbf{I} + \nabla \cdot \boldsymbol{\sigma}_{\text{dev}} \\
&= \nabla \cdot \left(\underbrace{-p\mathbf{I}}_{\text{volumetric}} + \underbrace{\boldsymbol{\sigma}_{\text{dev}}}_{\text{deviatoric}} \right)
\end{aligned}$$

where $-p\mathbf{I}$ and $\boldsymbol{\sigma}_{\text{dev}}$ are the volumetric and deviatoric parts respectively of the Cauchy stress tensor, which we will denote by $\boldsymbol{\sigma}$. The deviatoric part is dependent of the material in question.

3.5 Conservation of energy

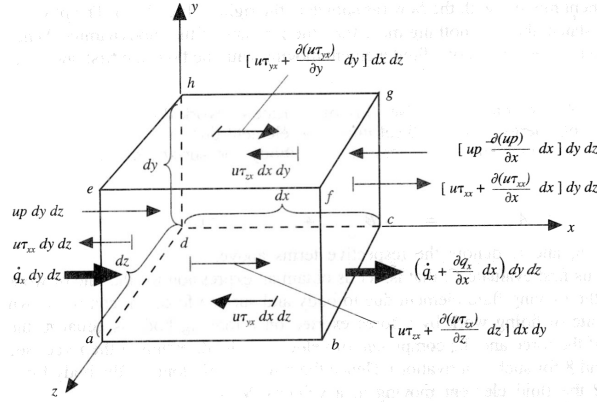
The physical principle stated in this section is merely the first law of thermodynamics, namely

$$\begin{array}{lcl}
\text{rate of change} & \text{net flux of} & \text{rate of work done on} \\
\text{of energy inside} & \text{heat into} & \text{element due to} \\
\text{fluid element} & \text{element} & \text{body and surface forces}
\end{array} = \tag{3.8}$$

3.5.1 Rate of work due to body and surface forces

We first evaluate the rate of work due to forces. Again, we make difference between body and surface forces. For body forces we have

$$\text{rate of work done by body forces} = \rho \mathbf{f} \cdot \mathbf{v} dx dy dz. \tag{3.9}$$

Figure 3.5: energy fluxes in x direction [18].

Analogous with the deduction in 3.3.3, 3.4 and with the help of Figure 3.5 we obtain the surface forces in x , y and z direction respectively:

$$\begin{aligned}
 \text{rate of work done by surface forces} &= \left[\underbrace{-\frac{\partial(up)}{\partial x} + \frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z}}_{x \text{ direction}} + \underbrace{-\frac{\partial(vp)}{\partial y} + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y} + \frac{\partial(v\tau_{zy})}{\partial z}}_{y \text{ direction}} \right. \\
 &\quad \left. + \underbrace{-\frac{\partial(wp)}{\partial z} + \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z}}_{z \text{ direction}} \right] dx dy dz \\
 &= \left[-\nabla \cdot p\mathbf{v} + \nabla \cdot u \begin{pmatrix} \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \end{pmatrix} + \nabla \cdot v \begin{pmatrix} \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \end{pmatrix} + \nabla \cdot w \begin{pmatrix} \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \end{pmatrix} \right] dx dy dz \\
 &= \left[-\nabla \cdot p\mathbf{v} + \nabla \cdot \left\{ u \begin{pmatrix} \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \end{pmatrix} + v \begin{pmatrix} \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \end{pmatrix} + w \begin{pmatrix} \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \end{pmatrix} \right\} \right] dx dy dz \\
 &= \left[-\nabla \cdot p\mathbf{v} + \nabla \cdot \left\{ (u \ v \ w) \begin{pmatrix} \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{pmatrix}^T \right\} \right] dx dy dz \\
 &= \left[-\nabla \cdot p\mathbf{v} + \nabla \cdot (\mathbf{v}^T \boldsymbol{\tau}^T)^T \right] dx dy dz \\
 &= \left[-\nabla \cdot p\mathbf{v} + \nabla \cdot (\boldsymbol{\tau}\mathbf{v}) \right] dx dy dz & (3.10) \\
 &= \nabla \cdot \boldsymbol{\sigma}\mathbf{v} dx dy dz & (3.11)
 \end{aligned}$$

3.5.2 Net flux of heat

The net flux of heat into the element is due to

- volumetric heating such as absorption or emission of radiation
- heat transfer across the surface due to temperature gradients, i.e. conduction.

Phenomena that result in volumetric heating are beyond the scope of this thesis and will therefore not be treated.

We obtain from Figure 3.5

$$\begin{aligned}
 \text{heating of element by conduction} &= -\left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right) dx dy dz \\
 &= -\nabla \cdot \mathbf{q} dx dy dz & (3.12)
 \end{aligned}$$

with $\mathbf{q} = (q_x \ q_y \ q_z)^T$ the flux of energy in $[\frac{J}{m^2 \cdot s}]$.

3.5.3 Rate of change of energy

The fluid element has two contributions to its total energy E in $[\frac{J}{kg}] = [\frac{m^2}{s^2}]$:

- the internal energy due to random molecular motion, e in $[\frac{J}{kg}]$
- the kinetic energy due to translational motion, $\frac{v^2}{2}$ in $[\frac{J}{kg}]$.

The mass of the element is $\rho dx dy dz$ and we get the time rate of change by the operator $\frac{D}{Dt}$. Herewith

$$\text{time rate of change of energy inside element} = \rho \frac{D}{Dt} \left(e + \frac{v^2}{2} \right) dx dy dz. \quad (3.13)$$

3.5.4 The energy equation

With (3.9) to (3.13), (3.8) becomes

$$\begin{aligned} \rho \frac{D}{Dt} \left(\underbrace{e + \frac{v^2}{2}}_{\text{total energy}} \right) dx dy dz &= \left[\underbrace{-\nabla \cdot \mathbf{q}}_{\text{conduction}} + \underbrace{\rho \mathbf{f} \cdot \mathbf{v}}_{\text{body forces}} \underbrace{-\nabla \cdot \boldsymbol{\sigma} \mathbf{v}}_{\text{surface forces}} \right] dx dy dz \\ \rho \frac{D}{Dt} \left(e + \frac{v^2}{2} \right) &= \rho \mathbf{f} \cdot \mathbf{v} + \nabla \cdot (\boldsymbol{\sigma} \mathbf{v} - \mathbf{q}) \end{aligned} \quad (3.14)$$

which is the nonconservation form. Again, writing out the material derivative gives the conservation form

$$\rho \left[\frac{\partial}{\partial t} \left(e + \frac{v^2}{2} \right) + \mathbf{v} \cdot \nabla \left(e + \frac{v^2}{2} \right) \right] = \rho \mathbf{f} \cdot \mathbf{v} + \nabla \cdot (\boldsymbol{\sigma} \mathbf{v} - \mathbf{q}) \quad (3.15)$$

3.6 Additional relations

The conservation of mass, momentum and energy form a set of five relations to determine ρ , \mathbf{v} and e . However, there are still other unknowns such as p , $\boldsymbol{\sigma}_{\text{dev}}$ and \mathbf{q} . The following equations have to be taken into account to solve the system.

3.6.1 Shear stresses

In many applications, it is assumed that the fluid is Newtonian which means that

$$\sigma_{xx} = \sigma_x = -p + \lambda \nabla \cdot \mathbf{v} + 2\mu \frac{\partial u}{\partial x} \quad (3.16)$$

$$\sigma_{yy} = \sigma_y = -p + \lambda \nabla \cdot \mathbf{v} + 2\mu \frac{\partial v}{\partial y} \quad (3.17)$$

$$\sigma_{zz} = \sigma_z = -p + \lambda \nabla \cdot \mathbf{v} + 2\mu \frac{\partial w}{\partial z} \quad (3.18)$$

$$\sigma_{xy} = \sigma_{yx} = \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \quad (3.19)$$

$$\sigma_{xz} = \sigma_{zx} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \quad (3.20)$$

$$\sigma_{yz} = \sigma_{zy} = \mu \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \quad (3.21)$$

where μ is the molecular viscosity coefficient and λ is the second viscosity coefficient, both in $[Pa \cdot s]$. Stokes made the hypothesis that

$$\lambda = -\frac{2}{3}\mu \quad (3.22)$$

which is frequently used but to the present day not confirmed.

With the assumption of Newtonian fluid, the stress tensor $\boldsymbol{\tau}$ can be written as

$$\begin{aligned}
\boldsymbol{\sigma} &= -p\mathbf{I} + \begin{pmatrix} -\frac{2}{3}\mu\nabla\cdot\mathbf{v} + 2\mu\frac{\partial u}{\partial x} & \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & -\frac{2}{3}\mu\nabla\cdot\mathbf{v} + 2\mu\frac{\partial v}{\partial y} & \mu\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) \\ \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \mu\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) & -\frac{2}{3}\mu\nabla\cdot\mathbf{v} + 2\mu\frac{\partial w}{\partial z} \end{pmatrix} \\
&= -p\mathbf{I} + 2\mu \begin{pmatrix} \frac{-\nabla\cdot\mathbf{v}}{3} + \frac{\partial u}{\partial x} & \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & \frac{-\nabla\cdot\mathbf{v}}{3} + \frac{\partial v}{\partial y} & \frac{1}{2}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) \\ \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) & \frac{-\nabla\cdot\mathbf{v}}{3} + \frac{\partial w}{\partial z} \end{pmatrix} \\
&= -p\mathbf{I} + 2\mu \left(\frac{-\nabla\cdot\mathbf{v}}{3}\mathbf{I} + \frac{1}{2}[\nabla\mathbf{v}^T + (\nabla\mathbf{v}^T)^T] \right) \\
&:= -p\mathbf{I} + 2\mu\mathbf{D}
\end{aligned} \tag{3.23}$$

which will be of importance in 3.7.4 with nondimensionalising. The dimension of \mathbf{D} is $[\frac{1}{s}]$.

3.6.2 Fourier's law

The heat flux \mathbf{q} is related to the space variation of the temperature T in $[K]$. Fourier's law states that the heat flux is given by

$$\mathbf{q} = -k\nabla T \tag{3.24}$$

where k in $[\frac{W}{m\cdot K}]$ is the coefficient of thermal conductivity.

3.6.3 Equations of state

There are still two relations needed to fully determine the fluid behaviour: the equations of state (EOS). Choosing ρ and e as independent variables, the EOS must provide

$$\begin{aligned}
p &= p(e, \rho) \\
T &= T(e, \rho)
\end{aligned} \tag{3.25}$$

Some usual EOS:

- incompressible fluid

$$\begin{aligned}
\rho &= \text{constant} \\
e &= c_p T = c_v T
\end{aligned}$$

- perfect gas and constant specific heat

$$\begin{aligned}
p &= \frac{nRT}{V} \\
e &= c_v T
\end{aligned}$$

with n the quantity of material in $[mol]$, R the gas constant in $[\frac{J}{mol\cdot K}]$ and c_p, c_v the specific heat in $[\frac{J}{kg\cdot K}]$ for a constant pressure and a constant volume respectively.

3.7 Navier-Stokes and Euler equations

Summarising the aforementioned, a system of equations is provided in the well-known Navier-Stokes or Euler form.

3.7.1 Navier-Stokes equations

The system of fluid equations in nonconservation partial differential form, namely (3.5), (3.6) and (3.14):

$$\begin{aligned}
\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{v} &= 0 \\
\rho\frac{D\mathbf{v}}{Dt} - \rho\mathbf{f} - \nabla\cdot\boldsymbol{\sigma} &= 0 \\
\rho\frac{D}{Dt}\left(e + \frac{v^2}{2}\right) - \rho\mathbf{f}\cdot\mathbf{v} - \nabla\cdot(\boldsymbol{\sigma}\mathbf{v} - \mathbf{q}) &= 0.
\end{aligned} \tag{3.26}$$

Together with the Newtonian shear stress equations, Fourier's law and the EOS the dependent variables ρ, \mathbf{v}, e, p and T can be solved with the independent $x, y, z \in V \wedge t \in [0, \infty)$ with boundary $\partial V = S$ (or Ω with boundary $\partial\Omega = \Gamma$). The system is solvable and thus fully determines the fluid behaviour.

3.7.2 Euler equations

In many cases, the inertia terms due to convective contribution are much bigger than the viscous forces, e.g. in gas dynamics. The viscosity can be neglected and the Navier-Stokes equations become the so-called Euler equations. This is often also referred to as the difference between viscous and inviscid flow. With viscous flow the transport phenomena of friction, mass diffusion and thermal conduction are taken into account. In the inviscid case they are excluded and therefore $\boldsymbol{\sigma}_{\text{dev}} = \mathbf{0}$. Herewith, the Euler equations are

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} &= 0 \\ \rho \frac{D\mathbf{v}}{Dt} - \rho \mathbf{f} + \nabla p &= 0 \\ \rho \frac{D}{Dt} \left(e + \frac{v^2}{2} \right) - \rho \mathbf{f} \cdot \mathbf{v} + \nabla \cdot (\mathbf{q} + p\mathbf{v}) &= 0. \end{aligned} \quad (3.27)$$

3.7.3 Initial and boundary conditions

To solve (3.26) or (3.27) uniquely, you need initial and boundary conditions (IC and BC). This is determined by looking at the highest order of derivatives in time and space for each dependent variable. The dependent variables are ρ, \mathbf{v}, e, p and \mathbf{q} .

The highest order of derivatives of ρ are included in $\nabla \cdot (\rho\mathbf{v}) = \rho \cdot \nabla\mathbf{v} + \nabla\rho \cdot \mathbf{v}$. For the case $\nabla\star$ it is known that one BC at the inflow boundary suffices [14]. Thus, one BC at the inflow boundary for ρ is needed. An IC is needed due to the term $\frac{\partial\rho}{\partial t}$.

For \mathbf{v} the terms $(\mathbf{v} \cdot \nabla)\mathbf{v}$, $\boldsymbol{\sigma}\mathbf{v}$ and $\nabla \cdot (\boldsymbol{\tau}\mathbf{v})$ determine what kind of BC is needed. In particular $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{v})$. In the Newtonian case as discussed in 3.6.1, $\boldsymbol{\sigma} = \boldsymbol{\sigma}(p, \nabla \cdot \mathbf{v})$. Herewith, the term $\nabla \cdot (\boldsymbol{\sigma}\mathbf{v})$ results in second order derivatives that need a BC for the whole boundary [14]. Again, one IC is needed due to $\frac{\partial\mathbf{v}}{\partial t}$.

The internal energy e is included in $\frac{\partial e}{\partial t}$, ∇e and $\nabla \cdot \mathbf{q}$. With (3.24), $\mathbf{q} = \mathbf{q}(T(\rho, e))$ with first order derivatives ∇ . Term $\nabla \cdot \mathbf{q}$ thus results in the need for a BC for e over the whole boundary. And $\frac{\partial e}{\partial t}$ for one IC.

Variables p and \mathbf{q} need not be considered because they are functions of ρ and e and thus the necessary conditions are already provided.

To summarise, for the Navier-Stokes equations in combination with its additional relations the need for ICs and BCs consists of

- ICs for ρ, \mathbf{v} and e
- inflow BC for ρ
- BCs on the whole boundary for \mathbf{v} and e .

For the Euler equations, less conditions are needed. The terms $\boldsymbol{\sigma}_{\text{dev}}\mathbf{v}$ and $\nabla \cdot (\boldsymbol{\sigma}_{\text{dev}}\mathbf{v})$ are no longer of concern because $\boldsymbol{\sigma}_{\text{dev}} = \mathbf{0}$. Hence, only $\nabla \cdot \mathbf{v}$ is left, so only a BC at the inflow boundary is needed.

The specific case of ICs and BCs for hydroplaning is treated in 3.8.

3.7.4 Nondimensionalising the Navier-Stokes equations

Nondimensionalising a set of equations yields a dimensionless form of the set. Dimension analysis by using the theorem of Buckingham [37] gives insight in the scaling relations of the system without using knowledge about the solutions. A corollary is that the total number of variables and parameters is minimal, making the equations easier in consideration. This minimisation is also the purpose of scaling. These techniques show that it is not useful to vary all the variables and parameters separately to gain knowledge about the

system, but that only certain combinations of them matter. The scaling used to simplify the Navier-Stokes equations is

$$\bar{\mathbf{x}} = \frac{\mathbf{x}}{L}, \quad \bar{\nabla} = \frac{\nabla}{\frac{1}{L}}, \quad \bar{t} = \frac{t}{\frac{L}{U}}, \quad \bar{\mathbf{v}} = \frac{\mathbf{v}}{U}, \quad \bar{\mathbf{D}} = \frac{\mathbf{D}}{\frac{U}{L}}, \quad \bar{\mathbf{f}} = \frac{\mathbf{f}}{\frac{U^2}{L}}, \quad \bar{\rho} = \frac{\rho}{\rho_0}, \quad \bar{p} = \frac{p}{\rho_0 U^2} \quad (3.28)$$

with L , U and ρ_0 a characteristic length, speed and density for the system. Note that these characteristics are not determined physically but merely chosen to suit the considered system. Written in this way, it is evident that the dimensions of the nominator and denominator of the scaled parameters are equal and thus the parameters itself are dimensionless.

The scaling (3.28) yields for (3.5)

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \bar{\nabla} \cdot \bar{\mathbf{v}} = 0 &\Leftrightarrow \frac{D(\rho_0 \bar{\rho})}{D(\frac{L}{U} \bar{t})} + \rho_0 \bar{\rho} \frac{\bar{\nabla}}{L} \cdot (U \bar{\mathbf{v}}) = 0 \\ &\Leftrightarrow \frac{\rho_0 U}{L} \frac{D\bar{\rho}}{D\bar{t}} + \frac{\rho_0 U}{L} \bar{\rho} \bar{\nabla} \cdot \bar{\mathbf{v}} = 0 \\ &\Leftrightarrow \frac{D\bar{\rho}}{D\bar{t}} + \bar{\rho} \bar{\nabla} \cdot \bar{\mathbf{v}} = 0 \end{aligned}$$

and for (3.6)

$$\begin{aligned} \rho \frac{D\mathbf{v}}{Dt} = \rho \mathbf{f} + \nabla \cdot \boldsymbol{\sigma} &\Leftrightarrow \rho \frac{D\mathbf{v}}{Dt} = \rho \mathbf{f} - \nabla \cdot (p\mathbf{I} + 2\mu\mathbf{D}) \\ &\Leftrightarrow \rho_0 \bar{\rho} \frac{D(U\bar{\mathbf{v}})}{D(\frac{L}{U}\bar{t})} = \rho_0 \bar{\rho} \frac{U^2}{L} \bar{\mathbf{f}} + \frac{\bar{\nabla}}{L} (-\rho_0 U^2 \bar{p}\mathbf{I} + 2\mu \frac{U}{L} \bar{\mathbf{D}}) \\ &\Leftrightarrow \frac{\rho_0 U^2}{L} \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \frac{\rho_0 U^2}{L} \bar{\rho} \bar{\mathbf{f}} - \frac{\rho_0 U^2}{L} \bar{\nabla} \bar{p} + \frac{2\mu U}{L^2} \bar{\nabla} \cdot \bar{\mathbf{D}} \\ &\Leftrightarrow \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \bar{\rho} \bar{\mathbf{f}} - \bar{\nabla} \cdot \bar{p} + 2 \frac{\mu}{\rho_0 U L} \bar{\nabla} \cdot \bar{\mathbf{D}} \\ &\Leftrightarrow \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \bar{\rho} \bar{\mathbf{f}} - \bar{\nabla} \bar{p} + \frac{2}{Re} \bar{\nabla} \cdot \bar{\mathbf{D}} \end{aligned}$$

where Re is the Reynolds number. The Reynolds number represents the ratio between inertial forces ρU and viscous forces $\frac{\mu}{L}$, both in $[\frac{N}{m^3 s}]$, force per volume per second. Consequently, it quantifies the relative importance of these two types of forces. It is the most important dimensionless number in fluid dynamics and is used to provide a criterion for determining dynamic resemblance. When two geometrically similar flow patterns, in perhaps different fluids with possibly different flowrates, have the same values for the relevant dimensionless numbers, they are said to be dynamically similar, and will have similar flow geometry.

The Reynolds number is also used to identify and predict different flow regimes, such as laminar or turbulent flow. Laminar flow occurs at low Reynolds numbers, where viscous forces are dominant, and is characterized by smooth, constant fluid motion. Turbulent flow, on the other hand, occurs at high Reynolds numbers and is dominated by inertial forces, which tend to produce vortices and other flow fluctuations.

$$\begin{aligned} Re &= \frac{\text{dynamic pressure}}{\text{shearing stress}} = \frac{\frac{\rho_0 U^2}{L}}{\frac{\mu U}{L^2}} \\ &= \frac{\text{inertia forces}}{\text{viscous forces}} = \frac{\rho_0 U}{\frac{\mu}{L}} \\ &= \frac{\rho_0 U L}{\mu} \end{aligned}$$

We skip the scaling of (3.14) because it is not of interest in the case of hydroplaning. Argumentation for this can be found in 3.8.4.

Nondimensionalising the ICs and BCs goes in a similar way.

3.8 Hydroplaning case

Concerning the case of hydroplaning, some assumptions can be made for the fluid equations. The fluid is considered to be incompressible, isotropic and Newtonian viscous with a constant viscosity μ . Furthermore, the only body force that is present is gravitation.

3.8.1 Incompressibility and isotropy

In an incompressible flow, the volume is unchanged during deformation and the density ρ remains constant. A motion of an incompressible material is called isochoric. Thus, in formula, incompressibility means

$$J = \det(\mathbf{F}) = 1, \rho = \text{constant and } e = cT \quad (3.29)$$

where J is called the Jacobian and $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ the deformation gradient with \mathbf{X} the material particle points and \mathbf{x} the spatial particle points, not to be confused with independent variables x, y, z . In other words, particles of a body described by coordinates \mathbf{X} are strained to new body coordinates \mathbf{x} [11]. Furthermore [8], $K = \infty$ with K the bulk modulus. These quantities will also be used later on in 7.4.2. It is often useful, particularly for incompressible materials, to write the stress and strain rate measures as the sum of deviatoric and hydrostatic or volumetric parts. The latter is also called the spherical part of a tensor. Conditions (3.29) imply that $\frac{D\rho}{Dt} = 0$. As a consequence, (3.5) becomes

$$\rho \nabla \cdot \mathbf{v} = 0.$$

Note that because $\rho = \text{constant} \neq 0$

$$\nabla \cdot \mathbf{v} = 0$$

holds. We say that the velocity is solenoidal or divergence free.

Isotropy of the fluid implies that the coefficient of thermal conductivity k is independent of the spatial coordinates and thus a constant.

The solenoidal velocity and isotropy in combination with (3.24) simplifies the energy equation (3.14) to

$$\rho \frac{D}{Dt} \left(cT + \frac{v^2}{2} \right) = \rho \mathbf{f} \cdot \mathbf{v} + (\nabla \cdot \boldsymbol{\sigma}) \mathbf{v} + \nabla \cdot k \nabla T.$$

3.8.2 Newtonian viscous flow

From the assumption that the flow is Newtonian viscous it follows that the shear stresses suffice to the relations described in 3.6.1. Together with the incompressibility condition $\nabla \cdot \mathbf{v} = 0$ the shear stresses become

$$\sigma_{xx} = \sigma_x = -p + 2\mu \frac{\partial u}{\partial x} \quad (3.30)$$

$$\sigma_{yy} = \sigma_y = -p + 2\mu \frac{\partial v}{\partial y} \quad (3.31)$$

$$\sigma_{zz} = \sigma_z = -p + 2\mu \frac{\partial w}{\partial z} \quad (3.32)$$

$$\sigma_{xy} = \sigma_{yx} = \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \quad (3.33)$$

$$\sigma_{xz} = \sigma_{zx} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \quad (3.34)$$

$$\sigma_{yz} = \sigma_{zy} = \mu \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right). \quad (3.35)$$

In this way, we get

$$(\nabla \cdot \boldsymbol{\sigma}_{\text{dev}})^T = \mu \begin{pmatrix} 2\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 w}{\partial x \partial z} \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial y \partial z} + \frac{\partial^2 v}{\partial z^2} \\ \frac{\partial^2 u}{\partial x \partial z} + \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 v}{\partial y \partial z} + 2\frac{\partial^2 w}{\partial z^2} \end{pmatrix}$$

Furthermore, from $\nabla \cdot \mathbf{v} = 0$ it follows that

$$\begin{aligned} \nabla \cdot \mathbf{v} = 0 &\Leftrightarrow \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \\ &\Leftrightarrow \frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \\ &\Leftrightarrow \frac{\partial^2 u}{\partial x^2} = -\frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 w}{\partial x \partial z} \\ &\Leftrightarrow 2\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 w}{\partial x \partial z} \\ &\Leftrightarrow 2\mu \frac{\partial^2 u}{\partial x^2} = \mu \left(\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 w}{\partial x \partial z} \right) \end{aligned}$$

so that the first row-element of $(\nabla \cdot \boldsymbol{\sigma}_{\text{dev}})^T$ becomes

$$\begin{aligned} \mu \left(2 \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 w}{\partial x \partial z} \right) &= 2\mu \frac{\partial^2 u}{\partial x^2} + \mu \left(\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 w}{\partial x \partial z} \right) \\ &= \mu \left(\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 w}{\partial x \partial z} \right) + \mu \left(\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 w}{\partial x \partial z} \right) \\ &= \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \\ &= \mu \nabla \cdot \nabla u \end{aligned}$$

or in short $\nabla^2 u$. In a similar way

$$\begin{aligned} \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 u}{\partial x \partial y} + 2 \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial y \partial z} + \frac{\partial^2 v}{\partial z^2} &= \mu \nabla^2 v \\ \frac{\partial^2 u}{\partial x \partial z} + \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 v}{\partial y \partial z} + 2 \frac{\partial^2 w}{\partial z^2} &= \mu \nabla^2 w \end{aligned}$$

and thus

$$(\nabla \cdot \boldsymbol{\sigma}_{\text{dev}})^T = \mu \nabla^2 \mathbf{v}$$

so the momentum equation (3.6) simplifies to

$$\rho \frac{D\mathbf{v}}{Dt} = \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{v}. \quad (3.36)$$

Note that sometimes the scalar operator $\nabla \cdot \nabla = \nabla^2$ is denoted by Δ .

3.8.3 Body force

The only body force present when considering hydroplaning is gravitation which works in the z -direction. Herewith

$$\mathbf{f} = \begin{pmatrix} 0 \\ 0 \\ f_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (3.37)$$

with $g = 9.81 \left[\frac{N}{kg} \right]$ the gravitation-constant.

3.8.4 Decoupled energy equation

The energy equation simplified by the incompressibility condition

$$\rho \frac{D}{Dt} \left(cT + \frac{v^2}{2} \right) = \rho \mathbf{f} \cdot \mathbf{v} + (\nabla \cdot \boldsymbol{\sigma}) \mathbf{v} + \nabla \cdot k \nabla T.$$

With the assumptions for the consideration of hydroplaning, the system of equations is now

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0 \\ \rho \frac{D\mathbf{v}}{Dt} &= \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{v} \\ \rho \frac{D}{Dt} \left(cT + \frac{v^2}{2} \right) &= \rho \mathbf{f} \cdot \mathbf{v} + (\nabla \cdot \boldsymbol{\sigma}) \mathbf{v} + \nabla \cdot k \nabla T. \end{aligned} \quad (3.38)$$

Note that the energy equation has been decoupled from the continuity and momentum equations. The last are four equations with four unknowns and can thus be solved on their own. If the problem were to involve heat transfer and hence temperature gradients in the flow, the temperature field is obtained from the energy equation by simply filling in the p and \mathbf{v} after.

In the case of hydroplaning we are not interested in the temperature of the fluid and its influence on the tire. Consequently, a review of the energy equation to solve T and the hereby needed EOS are redundant.

3.8.5 Initial and boundary conditions

To solve

$$\begin{aligned}\nabla \cdot \mathbf{v} &= 0 \\ \rho \frac{D\mathbf{v}}{Dt} &= \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{v}\end{aligned}\quad (3.39)$$

uniquely you need initial and boundary conditions. This is determined by looking at the highest order of derivatives in time and space for each dependent variable. Where in the compressible case $p = p(e, \rho)$, in the incompressible case we have $\rho = \text{constant}$ and a decoupled energy equation cf. 3.8.4 whereby p becomes a focal variable itself. So the dependent variables are p and \mathbf{v} .

The highest order of derivatives of p are included in $\nabla p = \left(\frac{\partial p}{\partial x} \quad \frac{\partial p}{\partial y} \quad \frac{\partial p}{\partial z} \right)^T$. For the case $\nabla \star$ it is known that one BC at the inflow boundary suffices [14]. However, a condition is already 'concealed' in the constraint $\nabla \cdot \mathbf{v} = 0$ in (3.39). At first sight this is hard to see. Applying $\nabla \cdot$ to (3.39) however, yields

$$\begin{aligned}\nabla \cdot \left(\rho \frac{D\mathbf{v}}{Dt} \right) &= \nabla \cdot (\rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{v}) \\ \nabla \cdot \rho \frac{D\mathbf{v}}{Dt} &= \nabla \cdot \rho \mathbf{f} - \nabla \cdot \nabla p + \nabla \cdot \mu \nabla^2 \mathbf{v} \\ \rho \frac{D(\nabla \cdot \mathbf{v})}{Dt} &= \rho \nabla \cdot \mathbf{f} - \nabla^2 p + \mu \nabla^2 (\nabla \cdot \mathbf{v}) \\ 0 &= \rho \nabla \cdot \mathbf{f} - \nabla^2 p \\ \nabla^2 p &= \rho \nabla \cdot \mathbf{f},\end{aligned}\quad (3.40)$$

a Poisson equation for p . When there are no body forces, i.e. $\mathbf{f} = \mathbf{0}$, the right hand side is equal to zero and thus (3.40) is called a Laplace equation. Herewith, a condition is already provided. There is no time derivative of the pressure present so no IC is needed.

The highest order of derivatives in space and time of \mathbf{v} are included in $\nabla^2 \mathbf{v}$ and $\frac{D\mathbf{v}}{Dt}$ respectively. Accordingly, a BC at the whole boundary and one IC are needed [14].

To summarise, for the incompressible, Newtonian viscous Navier-Stokes equations the need for ICs and BCs consists of

- IC for \mathbf{v}
- BC for the whole boundary for \mathbf{v} .

The IC for the velocity is $\mathbf{v} = \mathbf{0}$ in the Eulerian frame, because the puddle is assumed nonmoving on the tarmac. The fluid is in contact with either the air, the tarmac or the tire. Everywhere along the boundary the velocity \mathbf{v} should be prescribed. Where the fluid is in contact with air, the velocity follows from $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{0}$ where \mathbf{n} is the normal to the fluid surface. This is because the stresses coming from the air are negligible. The tarmac and tire have velocities $\mathbf{v} = \mathbf{0}$ and $\mathbf{v} = \boldsymbol{\omega} \times \mathbf{r}$ in the Eulerian frame respectively with $\boldsymbol{\omega}$ and \mathbf{r} the rotational speed in [$\frac{rad}{s}$] and radius of the tire in [m]. As a result of adhesion the velocity of the fluid equals the velocity of the tarmac and tire at the designated boundaries. These are also called no slip conditions.

3.8.6 Nondimensionalising the incompressible Navier-Stokes equations

The nondimensionalising of the incompressible Navier-Stokes equations is similar to the compressible case.

$$\bar{\mathbf{x}} = \frac{\mathbf{x}}{L}, \quad \bar{\nabla} = \frac{\nabla}{L}, \quad \bar{t} = \frac{t}{L/U}, \quad \bar{\mathbf{v}} = \frac{\mathbf{v}}{U}, \quad \bar{\mathbf{f}} = \frac{\mathbf{f}}{U^2}, \quad \bar{\rho} = \frac{\rho}{\rho_0}, \quad \bar{p} = \frac{p}{\rho_0 U^2}\quad (3.41)$$

with L , U and ρ_0 a characteristic length, speed and density for the system. As characteristic length we choose the width of the tire $L = 0.220[m]$. For characteristic speed the speed where hydroplaning starts occurring would be appropriate, so $U \in [80, 100][\frac{km}{h}]$. We assumed incompressibility so $\rho = \text{constant} = \rho_0$. Choose the density of water, namely $\rho_0 = 1000[\frac{kg}{m^3}]$.

The scaling (3.41) yields for (3.38)

$$\begin{aligned}\nabla \cdot \mathbf{v} = 0 &\Leftrightarrow \frac{\bar{\nabla}}{L} \cdot (U\bar{\mathbf{v}}) = 0 \\ &\Leftrightarrow \frac{U}{L} \bar{\nabla} \cdot \bar{\mathbf{v}} = 0 \\ &\Leftrightarrow \bar{\nabla} \cdot \bar{\mathbf{v}} = 0\end{aligned}$$

and

$$\begin{aligned}\rho \frac{D\mathbf{v}}{Dt} = \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{v} &\Leftrightarrow \rho_0 \bar{\rho} \frac{D(U\bar{\mathbf{v}})}{D(\frac{L}{U}\bar{t})} = \rho_0 \bar{\rho} \frac{U^2}{L} \bar{\mathbf{f}} - \frac{\bar{\nabla}}{L} (\rho_0 U^2 \bar{p}) + \mu \left(\frac{\bar{\nabla}}{L}\right)^2 U \bar{\mathbf{v}} \\ &\Leftrightarrow \frac{\rho_0 U^2}{L} \bar{\rho} \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \frac{\rho_0 U^2}{L} \bar{\rho} \bar{\mathbf{f}} - \frac{\rho_0 U^2}{L} \bar{\nabla} \bar{p} + \frac{\mu U}{L^2} \bar{\nabla}^2 \bar{\mathbf{v}} \\ &\Leftrightarrow \bar{\rho} \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \bar{\rho} \bar{\mathbf{f}} - \bar{\nabla} \bar{p} + \frac{\mu}{\rho_0 U L} \bar{\nabla}^2 \bar{\mathbf{v}} \\ &\Leftrightarrow \bar{\rho} \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \bar{\rho} \bar{\mathbf{f}} - \bar{\nabla} \bar{p} + \frac{1}{Re} \bar{\nabla}^2 \bar{\mathbf{v}} \\ &\Leftrightarrow \frac{D\bar{\mathbf{v}}}{D\bar{t}} = \bar{\mathbf{f}} - \bar{\nabla} \bar{p} + \frac{1}{Re} \bar{\nabla}^2 \bar{\mathbf{v}}.\end{aligned}\tag{3.42}$$

The last vector equation is obtained by noting that due to incompressibility $\rho = \rho_0 \Leftrightarrow \bar{\rho} = 1$.

Noting that $\mu_{\text{water}} = 8.90 \cdot 10^{-4} [\text{Pa} \cdot \text{s}]$, the Reynolds number is $Re = \frac{\rho_0 U L}{\mu} = \frac{1000 \frac{\text{U}}{3.6} 0.220}{8.90 \cdot 10^{-4}} \in [5.49 \cdot 10^6, 6.87 \cdot 10^6] = O(10^6)$. This confirms that the process is in the regimes of turbulence.

Nondimensionalising the ICs and BCs goes in a similar way.

Chapter 4

The tire

4.1 Introduction

In this chapter the tire is discussed. With the use of illustrations, the components and its functions are explained. Material properties specific for tire rubber are treated. The structure equations to model the deformation of the tire are derived. Although a tire seems to be a simple torus of rubber, it is more than that. A tire is composed of different materials which are assembled in order to suffice to certain demands. It is the only interface between the vehicle/driver and the road.

A tire can be considered in different views.

- Geometrically: a torus, a simple form.
- Mechanically: a flexible membrane pressure container.
- Structurally: a high performance composite.
- Chemically: composition of long-chain macromolecules.

This chapter aims to describe the components of a tire and their functions.

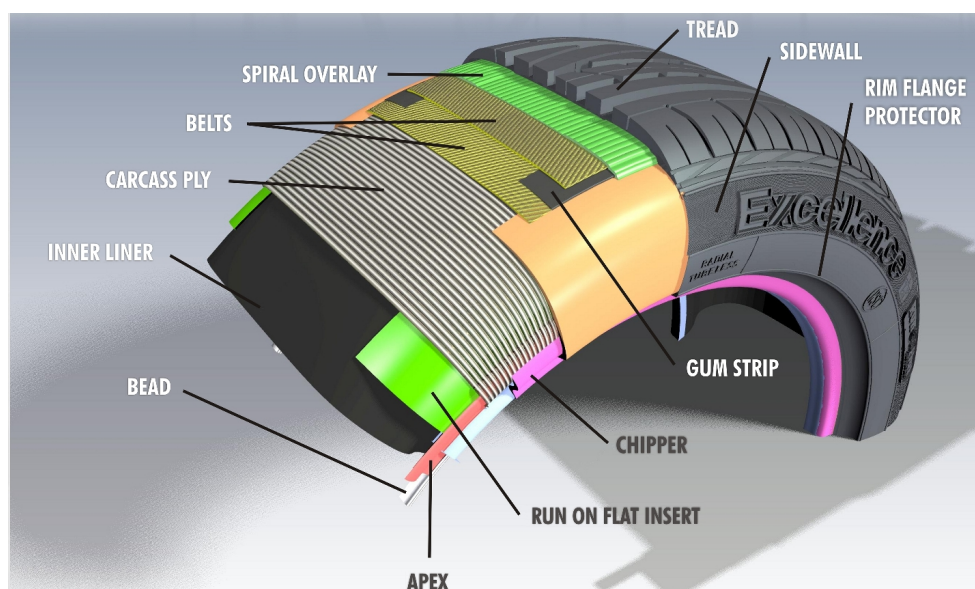


Figure 4.1: tire components [1].

4.2 Functions and components

The principal functions of a tire are load carrying capacity, geometric stability, damping vibrations (mechanical and acoustic), transmission of acceleration forces in longitudinal and lateral directions, steering response, good mileage, low fuel consumption, and safety. Some of these functions can be realised by just the application of rubber, but to improve the performance, reinforcing agents must be added. The design is a compromise of the many factors that come into play. In the majority of applications, the tire is to operate on a variety of road surfaces in different weather conditions with a performance as uniform as possible.

In this section the components a tire consists of are explained (Figures 4.1, 4.2 and 4.3). The components can be sorted into three groups: tread, carcass and belt package.

4.2.1 Tread

The tread is the component in contact with the road. It ensures that all driving forces are properly transmitted via a friction mechanism under a broad range of environmental conditions. It has to be

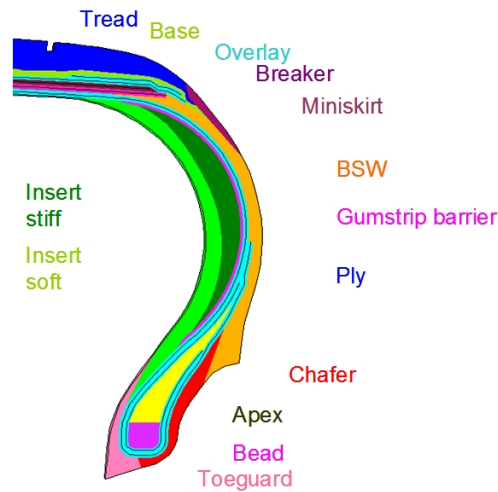


Figure 4.2: tire cross-section [1].

designed for wear-out resistance, traction, silent running and low heat build-up. The composition of the rubber, the cross-sectional shape of the tread, the number of ribs and grooves are important to determine wear, traction and running temperature performance.

- **Ribs** are the circumferential rows of tread rubber in direct contact with the road surface.
- **Grooves** are the channels between the tread ribs and are essential for traction, resistance to aquaplaning, directional control and cool-running properties. Treadwear indicators are molded into the bottom of the tread grooves to indicate when the tire should be replaced.
- **Tread base** is placed between the tread cap and the belt package in some tire constructions to improve energy efficiency or to reduce the operating temperature of the crown area.

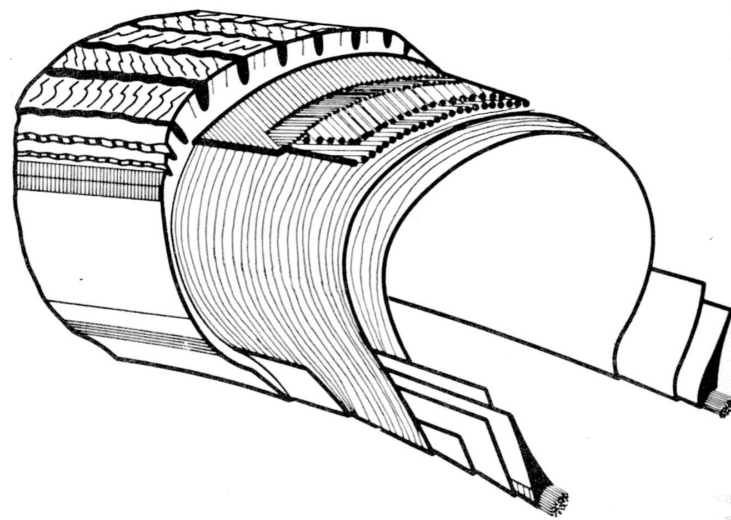


Figure 4.3: tire components [35].

4.2.2 Carcass

The carcass consists of high elastic modulus cords (e.g. steel, nylon, rayon, Kevlar) and low elastic modulus elastomeric compounds.

- **Ply** is made of multiple flexible high elastic modulus cords embedded and bonded to a matrix of low elastic modulus elastomeric material. The number of the layers depends on the tire type, the tire size, the inflation pressure and the speed rating. Its function is to give basic shape to the tire.
- **Liner** is the first tire component that faces the inflation pressure. It stops or reduces air diffusion and keeps the inflation pressure constant during service.

- **Bead** is a ring-like structural component which carries the load generated in the ply cords due to inflation. It is the component that holds the tire locked against the rim under different loading conditions.
- **Apex** is a polymeric component placed in the bead area, i.e. in the lower sidewall. It has three primary functions: to act as filler, to provide the desirable gauge between the ply and the ply turn-up, to increase the stiffness of the lower sidewall and to reduce the lateral tire displacement and to define the ply-line in the lower sidewall region for a given mold shape.
- **Toeguard** is a polymeric component which reduces the possibility of tire damage during mounting and dismounting.
- **Chipper** increases the stiffness of the lower sidewall and increases lateral stiffness.
- **Black side wall (BSW)** has several functions. It defines ply-line for given mold shape, defines stiffness distribution along ply, protects the ply by providing scuff resistance and influences damping characteristics of the tire.
- **Chafer** has the following functions: to provide proper interface between tire and rim for good fitment which reduces tire-rim slip and maintains inflation pressure, to reduce/eliminate bead area chafing during service.
- **Soft and stiff inserts** are only present in 'run-on-flat' tires. When a tire is punctured and deflates because of this, the soft and stiff inserts are the backup for driving to the nearest garage.

4.2.3 Belt package

The belt package consists of usually two belts of reinforcement cords lying between the carcass and tread. Its function is to control tire growth during inflation, provide rigidity to the tread to prevent distortions in the lateral direction during manoeuvres, reduce tread wear, give lateral and torsional stiffness to improve handling, supply puncture resistance and protect carcass from failure. It consists of the following components.

- **Belts** are made up of flexible high elasticity modulus cords bonded to a matrix of low elasticity modulus material.
- **Overlay** is a rubber-coated fabric cord layer. Its aim is to reduce tire deformations at high speed. The cords are usually in nylon and are circumferentially oriented.
- **Shoulder wedge** is the upper portion of the sidewall just below the tread edge. Shoulder design affects temperature development in the belt edges and cornering characteristics.

4.3 Materials

The materials to be used in tires must deal with a lot of contradicting demands. A tire needs to have grip. Grip increases with the temperature, but so does indurability. A winter-tire should have grip at low temperatures, but not be frictionless at high temperatures. Periodicity of the treadprofile makes the adhesion uniform and thus comfortable, but also comes with resonance resulting in sound. These contradictions make the task to find a proper material harder. Besides that, governments prescribe regulations for tires. Certain solvents used in the past for producing tires have been forbidden due to the health of personal in tire plants. Tires have to suffice to criteria like durability, maximum speed and maximum load. Listing these criteria on the tire itself is obligatory. Rubber can be produced in a way that all of the beforementioned is taken into account. Fillers and vulcanization chemicals help realise this.

Fillers are added to the compound to improve hardness, stiffness and abrasion resistance. Hereby the size, amount and dispersion of the particles of the fillers is of importance. The traditional filler is carbon black. Vulcanization chemicals include the initiators, accelerators and sulphur. During vulcanization, rubber molecules are cross-linked by atoms of sulphur. This makes the material harder, more durable and more resistant to chemical attacks. The surface of the material gets smoother and prevents it from sticking to metal or plastic chemical catalysts.

Rubber is a hyper- and viscoelastic material. A hyperelastic or Green elastic material is a material for which a linear elastic model does not suffice. In contrast with the latter, the stress-strain relation is derived from a strain energy density function. An example of a hyperelastic material model is treated in

7.4.2. Viscoelasticity is the property of materials that exhibit both viscous and elastic characteristics when undergoing deformation. A viscoelastic material has the following properties:

- hysteresis is seen in the stress-strain curve: history dependent behaviour between stress and strain
- stress relaxation occurs: constant strain causes decreasing stress
- creep occurs: step constant stress causes increasing strain.

Viscoelastic behavior is comprised of elastic and viscous components modeled as linear combinations of springs and dashpots, respectively. Each model differs in the arrangement of these elements. The elastic components, as previously mentioned, can be modeled as springs of elastic constant E , given the formula:

$$\sigma = E\epsilon$$

where σ is the normal stress in $[Pa]$, E is the elastic modulus of the material in $[Pa]$ and ϵ is the strain that occurs under the given stress, conform Hooke's Law [28]. The viscous components can be modeled as dashpots such that the stress-strain rate relationship can be given as

$$\sigma = \mu \frac{d\epsilon}{dt}$$

where μ is viscosity in $[Pa \cdot s]$ and t the time in $[s]$. Famous models that arrange these elements are the Maxwell, Kelvin-Voigt and Standard Linear Solid models (Figures 4.4, 4.5 and 4.6 respectively).

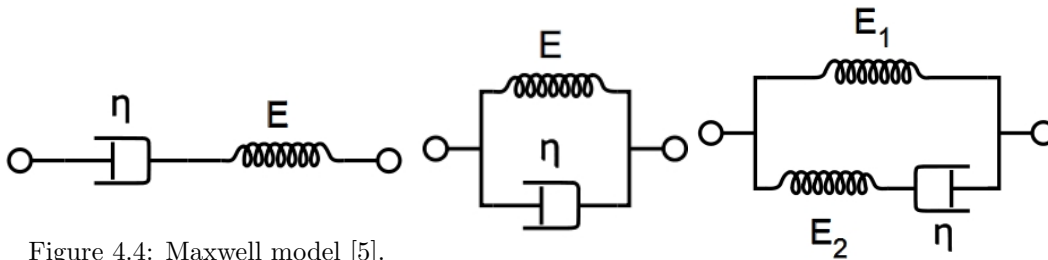


Figure 4.4: Maxwell model [5].

Figure 4.5: Kelvin-Voigt model [5].

Figure 4.6: Standard Linear Solid model [5].

The springs and dashpots in these models determine the loss modulus G'' and storage modulus G' respectively, cf. Figure 4.7. An explanation hereof belongs in the discipline material science and is thus beyond the scope of this report. These moduli are related to the material model Mooney-Rivlin, explained in 7.4.2.

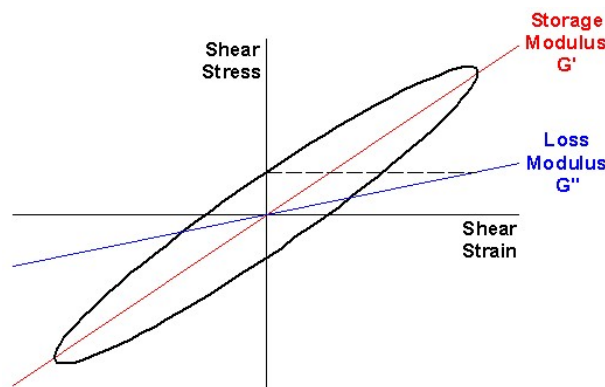
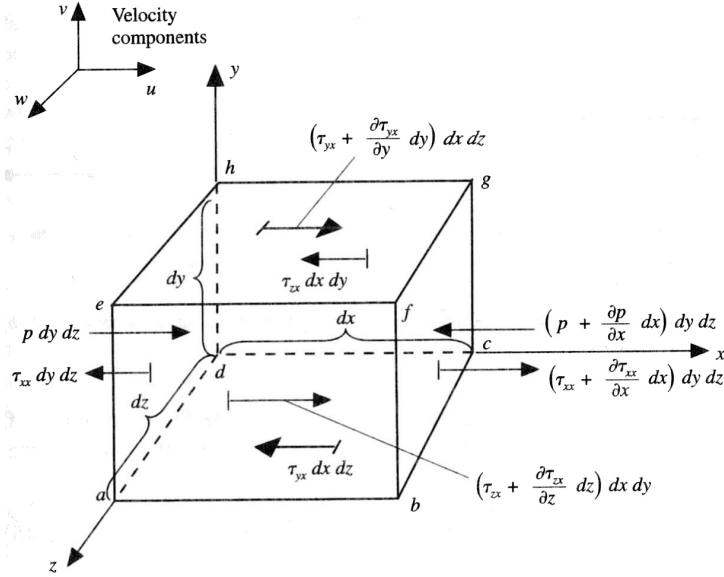


Figure 4.7: stress-strain cycle for rubber [17]. The ellipse represents a set of empirical measures.

4.4 Structure equations

The equations of the tire, the structure, come from the conservation of momentum and material models for hyper- and viscoelasticity.

Figure 4.8: forces in x direction.

4.4.1 Conservation of momentum

With Newton's second law we have $\mathbf{F} = m\mathbf{a} = m \frac{D^2 \mathbf{d}}{Dt^2}$ where $\mathbf{d} = \mathbf{x} - \mathbf{X}$ contains the displacements of the rubber. When considering a moving infinitesimal tire element, it can experience body and/or surface forces.

- Body forces, which act directly on the volumetric mass of the tire element. These forces "act at a distance"; examples can be gravitational and electromagnetic forces. In case of hydroplaning only gravitation is taken into account.
- Surface forces, which act directly on the surface of the tire element. They are only due to two sources.
 - The pressure distribution acting on the surface, imposed by the outside fluid surrounding the tire element.
 - The shear and normal stress distributions acting on the surface, also imposed by the outside fluid by means of friction.

Let \mathbf{f} denote the body force per unit mass in $[\frac{N}{kg}]$, then

$$\text{body force on element} = \rho \mathbf{f} dx dy dz.$$

From Figure 4.8 it follows that

$$\begin{aligned} \text{net surface force in } x \text{ direction} &= \left[p - \left(p + \frac{\partial p}{\partial x} dx \right) \right] dy dz + \left[\left(\tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} dx \right) - \tau_{xx} \right] dy dz \\ &+ \left[\left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy \right) - \tau_{yx} \right] dx dz + \left[\left(\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} dz \right) - \tau_{zx} \right] dx dy \\ &= \left[-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right] dx dy dz \end{aligned}$$

The $\star + \frac{\partial \star}{\partial \{x,y,z\}} d\{x,y,z\}$ terms are again from Taylor expansions, analogous with the deduction in 3.3.3.

With analogy in y and z direction and 3.4, the total force becomes

$$\mathbf{F} = [\rho \mathbf{f} + \nabla \cdot \boldsymbol{\sigma}] dx dy dz \quad (4.1)$$

With $\mathbf{F} = m\mathbf{a}$ and noting that $m = \rho dx dy dz$, $\mathbf{a} = \frac{D^2 \mathbf{d}}{Dt^2}$ and $\mathbf{f} = \mathbf{g}$ we get

$$\begin{aligned} \rho \frac{D^2 \mathbf{d}}{Dt^2} dx dy dz &= [\rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}] dx dy dz \\ \rho \frac{D^2 \mathbf{d}}{Dt^2} &= \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma} \end{aligned} \quad (4.2)$$

where the deviatoric stress $\boldsymbol{\sigma}_{\text{dev}}$ from $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\sigma}_{\text{dev}}$ is related to the material model used for rubber. This is discussed in 7.4.

4.4.2 Initial and boundary conditions

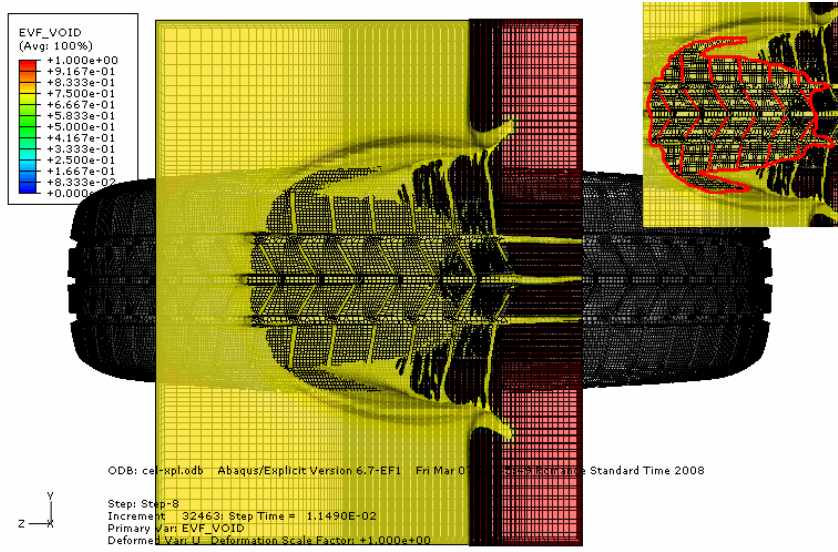


Figure 4.9: tire-fluid boundary indicated in red [31].

Considering (4.2)

$$\rho \frac{D^2 \mathbf{d}}{Dt^2} = \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}$$

three equations are provided. The dependent variables are \mathbf{d} , ρ , p and $\boldsymbol{\tau}$. Displacement $\mathbf{d} = \mathbf{x} - \mathbf{X}$ is a function of \mathbf{x} , since the material coordinates \mathbf{X} of the tire are known. Density of rubber ρ is determined by the volume change $J \equiv \det(\mathbf{F}) = \det\left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}}\right)$ thus $\rho = \rho(\mathbf{x})$. Pressure p stems from the fluid equations and in 7.4.2 it was determined that $\boldsymbol{\tau}$ is also a function of \mathbf{x} .

From (4.2) it follows that two ICs and one BC over the whole boundary are needed for a unique solution. Where the tire is in contact with air, the deformation and velocity can be assumed $\mathbf{d} = \mathbf{0} \wedge \frac{\partial \mathbf{d}}{\partial t} = \mathbf{0}$ at initiation. For $t > 0$, it follows from the stress free condition $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{0}$. For the footprint, cf. Figure 4.9, the displacement and velocity follow from the tire being in a steady state rolling (SSR) situation. A quantitative approach of SSR is beyond the scope of this report. For contact with the fluid, the displacement follows from stress of the fluid. Summarised in formula

$$\mathbf{d}|_{t=t_0} \wedge \frac{\partial \mathbf{d}}{\partial t}|_{t=t_0} \ni \begin{cases} \mathbf{d} = \mathbf{0}, \frac{\partial \mathbf{d}}{\partial t} = 0 & \Gamma_{\text{air}} \\ \text{SSR} & \Gamma_{\text{tarmac}} \end{cases} \quad (4.3)$$

$$\mathbf{d} \ni \begin{cases} \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{0} & \Gamma_{\text{air}} \\ \text{SSR} & \Gamma_{\text{tarmac}} \\ \text{fluid stress} & \Gamma_{\text{interface}} \end{cases} \quad (4.4)$$

The BC for the pressure p was treated in 3.7.3.

*In the construction of a three dimensional model of a tire, a segment is repeated along the arc. The length of this repeating segment along the circumference of the tire is called the pitch. The smaller this pitch, the easier the tire model. Obtaining a SSR situation is thus easier for tire models with a small pitch and is therefor done in Abaqus/Standard by implicit calculations with regard to computation time. When the segment repeats itself six or fewer times per circumference, as in the complex tire models GTC*L uses, it is more efficient to obtain a SSR situation with Abaqus/Explicit with explicit calculations. A more detailed description of this stabilisation is treated in chapter 10.*

Chapter 5

Fluid-structure interaction

5.1 Introduction

A problem involving fluid flow interacting with structures is a fluid-structure interaction problem, FSI in short. Flow for instance around or alongside a structure, influencing or interacting with the physical quantities of the structure and vice versa. From the fluid's point of view the pressure, velocity and temperature are influenced. From the structure's point of view the displacement, temperature and electrical properties (see Figure 5.1). These are fluid-structure interaction problems. The fluid flow usually undergoes large deformations but also the structure does not need to be fixed and rigid but can be moving and deformable. Examples are sloshing of tanks in launchers, limit cycle oscillations of wings, instabilities in the wind interaction with cable stayed bridges, the inflation of an airbag and of course hydroplaning.

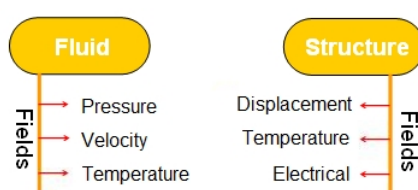


Figure 5.1: fluid-structure interaction [33].

These problems can be handled nowadays thanks to the increase of computing power and advances in numerical methods in the last decade. Where classically, FSI problems have been analyzed using cumbersome analytical methods, computational fluid dynamic (CFD) and FEM methods have taken over [9].

FSI problems can be divided into the following parts:

- monolithic or partitioned solver,
- fluid mesh,
- structure mesh,
- coupling of nonmatching meshes,
- interface tracking,
- space discretisation,
- time discretisation.

5.2 Monolithic or partitioned

In order to come to an approach for the development of an efficient FSI solver, the first choice that has to be made is whether to develop a monolithic or a partitioned solver. A monolithic solver aims at putting all the necessary components (the physical modelling, discretisations and solution algorithms) into a single computational model and solver, see Figure 5.2. It solves the set of fluid and structure equations simultaneously. Although some papers have appeared on monolithic solvers, the general opinion seems to be that monolithic solvers are not practical: not only the implementation of different physical models within a single solution environment can be difficult, also the continuous effort of keeping the solver up-to-date with the latest developments in each research field may prove to be a daunting task. Therefore the popular approach is to develop a partitioned solver [7]. A partitioned solver treats each physical domain separately, see Figure 5.2. Hereby it can use existing solvers that can be developed and maintained independently.

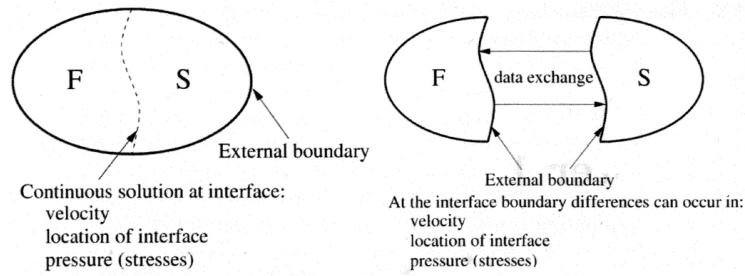


Figure 5.2: monolithic or partitioned [7].

A partitioned FSI solver consists of a flow solver, a structure solver and a coupling algorithm that couples the solvers at the fluid-structure interface both in space and time. The coupling algorithm contains an interpolation method to transfer data from one system to the other and an iteration scheme to obtain a coupled solution that is within the desired accuracy. Hereby, it matters if the considered problem is loose partition coupled or strong partition coupled. In case of strong partition coupling, the structure influences the fluid and vice versa in such a strong way, that iterations per time step are needed before continuing to the next time step, for the sake of accuracy. An example of a coupling scheme for strong partition coupling is depicted in Figure 5.3. Time progresses from left to right. The connections between the fluid and structure sides represent information exchange. As indicated by the arrows, these information exchanges also go back in time, in contrast with loose partition coupling where no the coupling is done without iterations.

Abaqus' FSI solver is partitioned and uses a loose partition coupling algorithm. Thus the fluid and structure equations are solved subsequently.

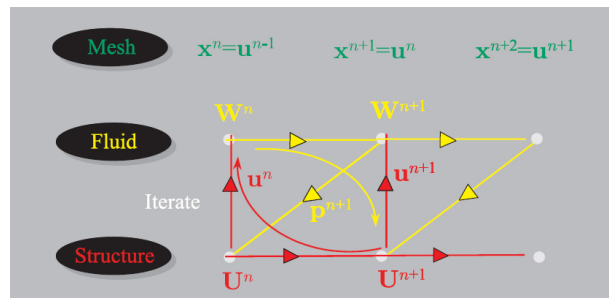


Figure 5.3: a strong partition coupling scheme [9].

5.3 Coupling nonmatching meshes

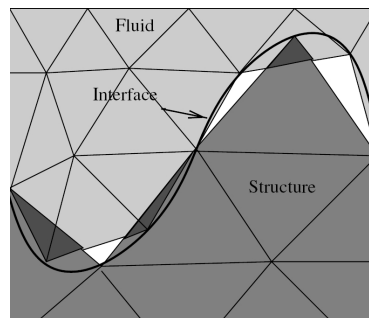


Figure 5.4: nonmatching meshes in 2D [9].

Domain decomposition is a common way to speed up complex computations. The discrete meshes used in the different domains do not have to match at their common interface. This is also the case when different physical fields are involved such as in fluid-structure interaction computations. Exchange of information over this interface is therefore no longer trivial [10].

Computers and numerical methods have significantly advanced over the past decade such that the simulation of complex physical problems and even multi-field problems has become feasible. For an efficient computation of these problems parallel computing is a must. A common way to perform these parallel computations is by domain decomposition. The total domain is subdivided into smaller domains on which the problem is computed in parallel and interaction effects between the domains are treated as boundary conditions at their common interfaces. Especially when different physical aspects are involved the meshes of the different domains do not have to match at their common interface. This means that the discrete interface between the domains may not only be nonconforming, but there may also be gaps and/or overlaps between the meshes. The exchange of data over the discrete interface becomes then far from trivial. In Figure 5.4 a 2D example of a nonmatching discrete interface between a flow and structure domain is shown. Another example is Figure 5.5.

For information transfer FSI computations require that pressure loads are transmitted from the fluid side of the fluid-structure interface to the structural nodes on that interface. Also, once the motion of the structure has been determined, the motion of the fluid mesh points on the interface has to be imposed. In FSI simulations generating matching meshes at the fluidstructure interface is usually not desirable, because the flow generally requires a much finer mesh than the structure and, due to the modularity of the partitioned coupling technique, different teams may take care of the different solvers. When meshes are nonmatching, an interpolation/projection step has to be carried out to enable transfer of information between the two domains.

There are several criteria which such a data exchange or coupling method ideally should satisfy. The most important are

- global conservation of energy over the interface,
- global conservation of loads over the interface,
- accuracy,
- conservation of the order of the coupled solvers,
- efficiency, which is defined as a ratio between accuracy and computational costs.

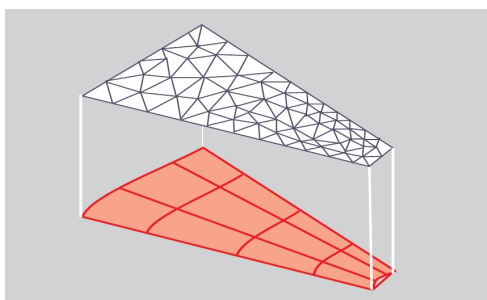


Figure 5.5: nonmatching meshes [9].

The simplest and fastest way to perform the information transfer is to obtain the information from the closest point in the other mesh, the so-called nearest neighbour interpolation. However, this only provides satisfactory results if the two grids are almost matching. A more accurate way of handling the data transfer is by projection, cf. Figure 5.6. To obtain information from the other mesh, a point can be orthogonally projected on that mesh and the information in that projection point can be used in the original point. Similarly, a whole element can be orthogonally projected on the other mesh and the size of the area of intersection can then be used to define to what degree the values of that element have to be taken into account. Note that the perspective in Figure 5.6 does not imply the surfaces being separated (no contact). The separation is only for comprehension of the projection. The third way to exchange data is using spline based methods. These are often applied in interpolation schemes in finite element methods. An example is radial basis functions interpolation.

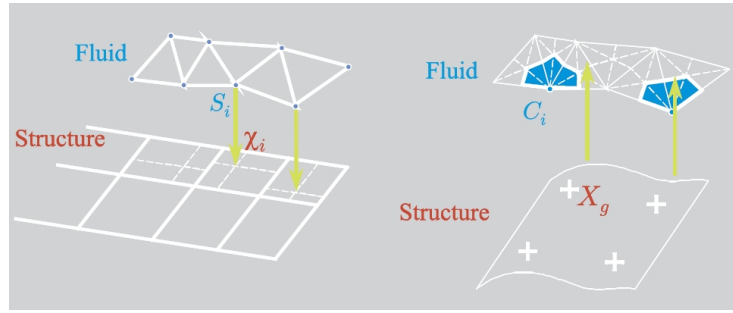


Figure 5.6: information transfer by means of projection [9].

5.4 Interface tracking

Abaqus' CEL method for treating FSI problems contains an interface tracking method like every FSI solver. In this case the volume of fluid (VOF) method is used [15]. VOF is a method based on the concept of a fractional volume of fluid per mesh cell.

Free boundaries are considered to be surfaces on which discontinuities exist in one or more variables. Examples are free surfaces, material interfaces, shock waves and, like in our case, interfaces between fluid and deformable structures. Three types of problems arise in the numerical treatment of free boundaries:

1. discrete representation,
2. evolution in time,
3. imposing boundary conditions on them.

Considering a cell of a mesh it is customary to use only one value for each dependent variable defining the fluid state. The use of several points in a cell to define the region occupied by the fluid, therefore, seems unnecessarily excessive. Suppose that we define a function $F \in [0, 1]$ whose value is unity at any point occupied by fluid and zero otherwise. The average value of F in a cell would then represent the fractional volume of the cell occupied by fluid. In particular, a unit value of F would correspond to a cell full of fluid, while a zero value would indicate that the cell contained no fluid. Accordingly, $\forall t \in [t_0, \infty)$

$$(FV)_{i,j,k}(t) = \int_{V_{i,j,k}} F(x, y, z, t) dV \Leftrightarrow F_{i,j,k} = \frac{\int_{V_{i,j,k}} F(x, y, z, t) dV}{V_{i,j,k}} \quad (5.1)$$

with F the volume fraction and V the volume in $[m^3]$. Cells with a volume fraction F that is element of the open interval $(0, 1)$ must contain a free surface. In this way, VOF only uses one value per cell which is consistent with the requirements of all other dependent variables.

From conservation of mass, the conservation of volume follows. Conservation of volume implies conservation of F . Accordingly, it is expressed in a similar way. Recall that for conservation of a material property F

$$\begin{array}{l} \text{net flow out } F \\ \text{of control volume} \\ \text{through surface} \end{array} = \begin{array}{l} \text{time rate of} \\ \text{decrease of } F \\ \text{inside control volume} \end{array} .$$

This yields

$$\frac{\partial}{\partial t} \iiint_V F dV = - \iint_S F \mathbf{v} \cdot \mathbf{n} dS.$$

For VOF methods an Eulerian frame is used and thus V is independent of the time t . Herewith and the divergence theorem applied on the right hand side it follows that

$$\iiint_V \left[\frac{\partial F}{\partial t} + \nabla \cdot F \mathbf{v} \right] = 0 \quad (5.2)$$

called the advection equation in integral form [29]. The velocity \mathbf{v} stems from the fluid equations. Typically in interface tracking methods, an important issue is the prevention of loss of fluid. The used discretisation

scheme plays a role here. An explanation hereof can be found in 7.5.

Notice that in case of incompressibility, (3.29) yields $\nabla \cdot F\mathbf{v} = \nabla F \cdot \mathbf{v} + F\nabla \cdot \mathbf{v} = \nabla F \cdot \mathbf{v}$ and thus

$$\iiint_V \left[\frac{\partial F}{\partial t} + \nabla F \cdot \mathbf{v} \right] = 0.$$

Apart from the evolution of F , it is still unknown where the fluid is in each cell. This information is needed for the reconstruction of the interface.

There are several interface reconstruction methods, both first and second order accurate in space, cf. Figure 5.8. In [19], a design criteria is proposed, namely that when lines in two dimensions or planes in three dimensions are reproduced exactly, the method is second order accurate. The authors introduce two algorithms based on least squares, that are second order accurate. Furthermore, methods like simple line and piecewise linear interface calculation (SLIC and PLIC) are recalled, cf. Figure 5.7. Abaqus' CEL method makes use of a PLIC [41]. Although treating this PLIC thoroughly is in line with the thesis, Simulia's lack in giving details unfortunately restricts us in doing so. Information that is available, can be found in 7.5.

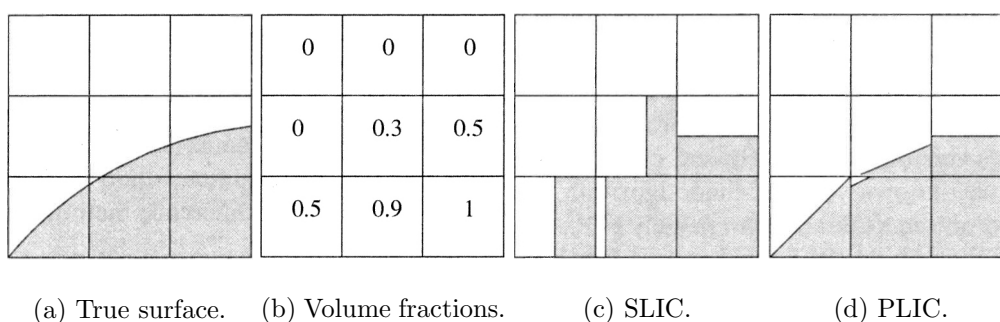


Figure 5.7: interface reconstruction [19].

1. Capturing methods
2. Tracking methods
 - (a) Front tracking
 - i. Hybrid front tracking front capturing
 - ii. Immersed boundary
 - iii. Cut cell
 - (b) Volume tracking
 - i. Marker and cell
 - ii. Volume of fluid
 - A. Hirt and Nichols
 - B. Simple linear interface construction
 - C. Piecewise linear interface construction
 - D. Flux corrected transport
 - E. Constrained interpolation profile
 - iii. Level set
 - iv. Coupled level set volume of fluid

Figure 5.8: classes of interface reconstruction methods.

Chapter 6

Finite element method

6.1 Introduction

As a reminder, FEM is treated shortly in this chapter. For a more detailed description we refer to [38]. The FEM is a discretisation method widely used by both industries and academics. FEM is used in fluid and structure software packages, in particular in the latter. FEM is well suited for unstructured problems cf. Figure 6.1, and has a strict local character [38]. All information in one element is used, without considering neighbours. This makes the method attractive for computer implementation. An important advantage of the FEM is that the treatment of BCs is almost always natural and therefore simpler than in classical difference methods.

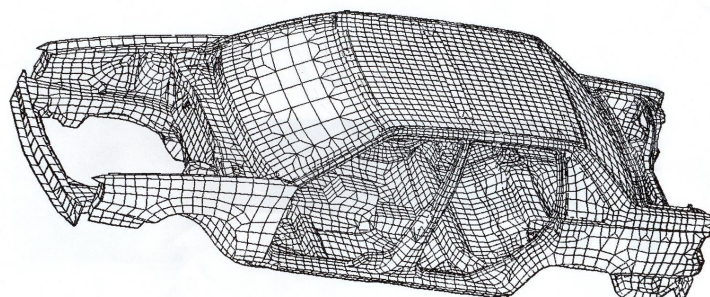


Figure 6.1: unstructured mesh of carbody [3].

6.2 Procedure

One typically starts with a partial differential equation (PDE) or minimisation problem. Under the conditions of linearity, symmetry and positivity, a PDE corresponds with a minimisation problem. The Euler-Lagrange equations are a well known example hereof. Subsequently, the method of Ritz is used to obtain an linear system from the minimisation problem. It approximates the variables in question in a linear combination of a finite fixed set of basis functions φ_j . For example

$$u^N(x, t) = \sum_{j=1}^N c_j(t) \varphi_j(x). \quad (6.1)$$

A clever choice of basis functions is made, such that they satisfy the Kronecker delta property at certain locations of each element, depending on the chosen order of geometric interpolation. The latter is often linear or quadratic. The unknowns left are the $c_j(t)$, $j = 1, 2, \dots, N$ in $\min(J(c_1, c_2, \dots, c_N))$. Because we are dealing with a minimisation problem, and thus looking for the extremes, we are in essence looking for the solution of

$$\frac{\partial J}{\partial c_i} = 0, i = 1, 2, \dots, N, \quad (6.2)$$

which forms a set of N equations with N unknowns. The integrals in these equations are typically approximated with the midpoint, trapezium or Simpson rules or Gauss quadrature. The choice of integration rule depends on the maintained order of error in space. When the coefficients c_i are solved with the N equations, the expansion (6.1) is known and thus the approximation of the variable is known. Note that minimisation problems usually admit a larger solution class than a PDE formulation because extremes can be local as well as absolute. The solution of a minimisation problem is therefore referred to as the generalised solution of the PDE.

When the conditions of linearity, symmetry and positivity are not satisfied, Galerkin's weak formulation is used alternatively. Galerkin is a generalisation of the Ritz method. The PDE in question is multiplied with a test function η . This η is typically an element of the same space as the solution is. When it is not, we speak of the Petrov-Galerkin method. Subsequently one integrates over the domain and replaces the variables with a linear combination of a finite fixed set of basis functions φ_j as in (6.1). Again, a set of N equations with N unknowns is obtained wherein the integrals are approximated as mentioned before.

Chapter 7

Abaqus' CEL method

7.1 Introduction

With discretisation meshes, fluid dynamics, tire mechanics and FSI treated, Abaqus' CEL method can be revisited. The method was developed initially for the simulation of the inflation of an airbag, cf. Figure 7.1. Having simulated one FSI problem, problems like tank sloshing due to impact loads, collision between a bird and aircraft and hydroplaning followed. The method consists of mesh treatment, surface reconstruction and solving the coupled fluid and structure equations.

The information stems from [34, 32, 33].

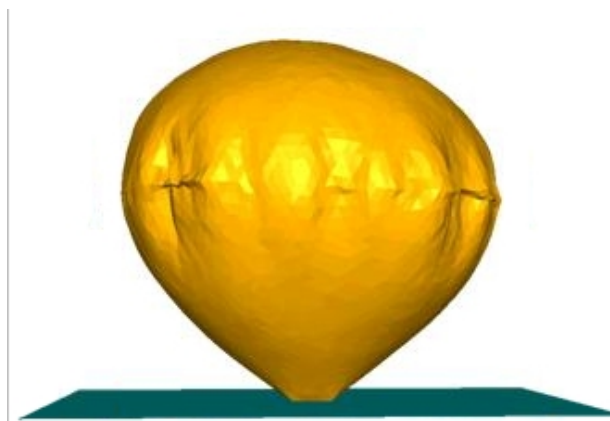


Figure 7.1: inflation of an airbag [33].

7.2 Mesh treatment

If a discretisation element is fully filled by material, 'regular' finite element theory can be used to determine the deformation. Even if it is filled with fluid. The 'drawback' of fluid is that it has to keep deforming to carry a load. Simulia claims to have developed an element that is able to do this. Herewith, FEM will give the correct deformation for an applied load for a volume of material. The strategy here is to determine the deformation of a prescribed Lagrangian volume of fluid, not a volume in space. Notice that apparently CEL solves the displacement, not the velocity, as in regular CFD codes. In order to limit the deformation, the method remeshes in Lagrangian sense to improve the shape of the elements. Subsequently, the mesh is remapped onto the original Eulerian mesh by interpolation. The interpolation contains an error. The remap can be seen as a convection of material from the old onto the new mesh. The claim is that the interpolation error is no greater than the error made in a convection step in 'regular' CFD codes. The remapping is done every time step.

What is not accounted for is the behaviour of the fluid at length scales smaller than the smallest element length, for which turbulence models like $k - \epsilon$ and Reynolds averaged NavierStokes equations are typically added to a CFD code. Herewith, it can be concluded that CEL does not take turbulence into account other than in the sense of direct numerical simulation (DNS). With DNS the mesh is taken extremely fine to take turbulence effects into account. A detailed description of turbulence models is beyond the scope of this report.

7.3 Fluid dynamics

7.3.1 Compressible or incompressible fluid?

In contrast with 3.8, Abaqus uses the compressible form of the fluid equations, cf. (3.26). In combination with a high bulk modulus K , the incompressible case is approximated. The infamous problems arising with the incompressible form, e.g. singularities in the system of equations, are however not circumvented in this way. The higher the bulk modulus, the more these problems show up. Methods like the penalty

function method and alternative geometric locations of the variables within the discretisation elements are used to solve these issues in solving the incompressible form [30]. It is unknown how Abaqus solves these problems. No information on a stabilisation method is known. The small time step needed for the explicit solving has maybe relaxed the beforementioned problems.

The energy equation is also part of (3.26). The need to solve it depends on the EOS. If the pressure p is merely a function of mass density ρ but not of energy density e , so $p = p(\rho)$ and not $p = p(e, \rho)$, then the energy equation is decoupled from the continuity and momentum equations. Abaqus uses the Mie-Grüneisen EOS described in 7.3.2, which states a relation between p and ρ . Therefore it is not mandatory to solve the energy equation.

7.3.2 Mie-Grüneisen with Hugoniot fit

The EOS used by Abaqus is Mie-Grüneisen with a Hugoniot fit [32].

The most common form of Mie-Grüneisen is

$$p - p_H = \Gamma \rho (e - E_H), \quad (7.1)$$

where p_H and E_H are the Hugoniot pressure and specific energy per unit mass and are functions of density only. The Hugoniot pressure p_H is, in general, defined from experimental data (curve-fitting). The Γ is the Grüneisen ratio which describes the alteration in a crystal lattice's vibration's frequency, also called a phonon, based on the lattice's increase or decrease in volume as a result of temperature change. It is used in quantummechanics and is defined as

$$\Gamma = \Gamma_0 \frac{\rho_0}{\rho} \quad (7.2)$$

where Γ_0 is the initial Grüneisen ratio and ρ_0 is the reference density. The Hugoniot energy E_H is related to p_H by

$$E_H = \frac{p_H \eta}{2\rho_0} \quad (7.3)$$

where $\eta = 1 - \frac{\rho_0}{\rho}$ is the nominal volumetric compressive strain. Elimination of Γ and E_H from (7.1) yields

$$p = p_H \left(1 - \frac{\Gamma_0 \eta}{2} \right) + \Gamma_0 \rho_0 e. \quad (7.4)$$

The EOS and the energy equation represent coupled equations for pressure and internal energy. Abaqus/Explicit, the solver of Abaqus using explicit methods only, solves these equations simultaneously at each material point.

A common fit to the Hugoniot data is given by

$$p_H = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2} \quad (7.5)$$

where c_0 and s define the linear relationship between the shock velocity U_s and the particle velocity U_p , both in $[\frac{m}{s}]$, as follows:

$$U_s = c_0 + sU_p.$$

With these assumptions (7.4) becomes

$$p = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2} \left(1 - \frac{\Gamma_0 \eta}{2} \right) + \Gamma_0 \rho_0 e \quad (7.6)$$

where $\rho_0 c_0^2 > 0$ is equivalent to the elastic bulk modulus at small nominal strains.

In the case of modelling water, Abaqus prescribes $s = 0 \wedge \Gamma_0 = 0$. Filling this in leads to a simplification of the EOS:

$$p|_{s=0 \wedge \Gamma_0=0} = \left[\frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2} \left(1 - \frac{\Gamma_0 \eta}{2} \right) + \Gamma_0 \rho_0 e \right]_{s=0 \wedge \Gamma_0=0} = \rho_0 c_0^2 \eta.$$

Recalling that $\eta = 1 - \frac{\rho_0}{\rho}$ we get

$$p = \rho_0 c_0^2 \left(1 - \frac{\rho_0}{\rho} \right).$$

Note that by definition of the physical quantities pressure and density it holds that $0 < p, \rho < \infty$. From $p > 0$ it follows that

$$\rho_0 c_0^2 \left(1 - \frac{\rho_0}{\rho}\right) > 0 \quad \Rightarrow_{\rho_0 c_0^2 > 0} \quad 1 - \frac{\rho_0}{\rho} > 0 \quad \Leftrightarrow \quad 1 > \frac{\rho_0}{\rho} \quad \Leftrightarrow_{\rho > 0} \quad \rho > \rho_0$$

so $0 < p < \infty \wedge 0 < \rho_0 \leq \rho < \infty$. Herewith

$$p(\rho_0) = 0 \wedge \lim_{\rho \rightarrow \infty} p = \rho_0 c_0^2.$$

Note that

$$\frac{dp}{d\rho} = \frac{d}{d\rho} \left(\rho_0 c_0^2 \left(1 - \frac{\rho_0}{\rho}\right) \right) = \frac{(\rho_0^2 c_0^2)}{\rho^2}$$

from where $\frac{dp}{d\rho}(\rho_0) = c_0^2$ follows. This equation represents isentropy in the linear acoustic regime [25] and in addition when $c_0 \rightarrow \infty$, $\frac{dp}{d\rho}(\rho_0) = \infty$ which represents incompressibility.

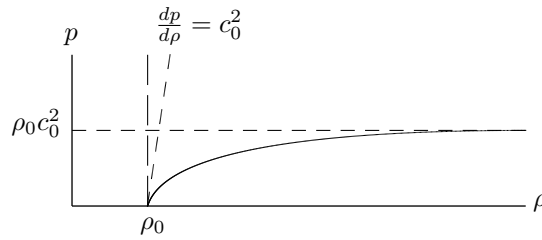


Figure 7.2: relation between pressure p and density ρ .

The Mie-Grüneisen EOS are normally used when considering processes with high pressures and high sound-velocities, e.g. SONAR or explosive applications. Therefore, at first hand the choice seems unlogical. Filling in $s = 0 \wedge \Gamma_0 = 0$ simplifies the relation between pressure p and density ρ to a inversely proportional equation. Claim is that this simplification suffices for the modelling of water. The reasoning is that water is nearly incompressible. Incompressibility implies $\frac{dp}{d\rho} = \infty$. If c_0 is taken sufficiently large, then $\frac{dp}{d\rho} = c_0^2 \rightarrow \infty$, approximating incompressibility. In physical terms, when the speed of sound of the medium is large, the change in pressure due to a change in density is large. Nevertheless, Simulia can not explain why their Mie-Grüneisen model differs from references, e.g. [13, 23].

7.3.3 Galerkin's weak formulation

CEL solves the continuity and momentum equations. We do not simply say the Navier-Stokes equations because this would imply solving the velocity which is not the case. It is claimed that the equations contain the displacements instead. This was done because elastic continua generates shear stresses not by velocity gradients, but by displacement gradients. This approach we have never heard of. Neither have colleagues at the universities. The form maintained is

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \nabla \cdot \frac{D\mathbf{d}}{Dt} &= 0 \\ \rho \frac{D^2 \mathbf{d}}{Dt^2} - \rho \mathbf{g} - \nabla \cdot \boldsymbol{\sigma} &= 0 \end{aligned} \quad (7.7)$$

where the volumetric and deviatoric parts of Cauchy stress tensor $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\sigma}_{\text{dev}}$ are dependent of the EOS as described in 7.3.2 and $\mathbf{v} = \frac{D\mathbf{d}}{Dt}$ as defined by Newtonian shear in 3.6.1 respectively. Note that because the Lagrangian reference frame is used and thus the frame moves with the material, the continuity equation is satisfied by $\rho = \rho_0$. Only the momentum equation will be considered.

The elements used for the fluid in the finite element formulation of the problem are EC3D8R, which stands for Eulerian (E), continuum (C) and three-dimensional (3D) elements with the unknowns placed at the nodes. The elements for the structure can be chosen according the preferences of the user.

Where typically the Gaussian quadrature, midpoint, trapezium and Simpson rules are used to approximate the integrals in the elements of FEM system matrix, Abaqus often uses a reduced amount of integration

points. This is indicated with the R in the element name. This reduction in amount of integration points is not a problem in case of an Eulerian grid, because the grid is fixed. Spurious hourglass modes as in the Lagrangian case are therefor not possible, see 7.4.

The basis functions for the fluid elements are fixed in Abaqus on trilinear. They are of the form

$$\begin{aligned}\varphi_i(x, y, z) &= (a_x x + b_x)(a_y y + b_y)(a_z z + b_z) \\ &= b_x b_y b_z + a_x b_y b_z x + b_x a_y b_z y + b_x b_y a_z z + a_x a_y b_z x y + a_x b_y a_z x z + b_x a_y a_z y z + a_x a_y a_z x y z \\ &:= a_0^i + a_1^i x + a_2^i y + a_3^i z + a_4^i x y + a_5^i x z + a_6^i y z + a_7^i x y z.\end{aligned}\quad (7.8)$$

The eight coefficients are determined by the demand that it must suffice to the Kronecker delta property in each of the eight nodes, which gives eight equations with eight unknowns.

The weak formulations of the fluid and structure equations are identical. This is the result of the deliberate choice of Simulia using FEM and the Lagrangian frame for both the fluid and the structure. As this choice is traditionally proper for the structure equations, it will accordingly be treated in 7.4.3.

7.4 Structure mechanics

7.4.1 Incompressibility for solids

Many problems involve the prediction of material response of almost incompressible materials. In particular, this is true at large strains, since most solid materials show relatively incompressible behavior under large deformations. When the material response is incompressible, the solution to a problem cannot be obtained in terms of the displacement history only, since a purely hydrostatic pressure can be added without changing the displacements. Consider for example Figure 7.3. An element is subjected to an uniform hydrostatic pressure. The nodes will not displace, so a change in pressure can not be calculated from the displacements. The pressure, however, does result in an internal stress.

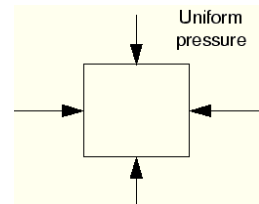


Figure 7.3: uniform hydrostatic pressure [32].

The nearly incompressible case, in Abaqus when the bulk modulus is much larger than the shear modulus or when $\nu > 0.4999999$, exhibits behavior approaching this limit, in that a very small change in displacement produces extremely large changes in pressure, so that a purely displacement-based solution is too sensitive to be useful numerically. For instance, roundoff errors on the computer may cause the method to fail. In Abaqus/Standard, this singular behavior is removed from the system by treating the pressure stress as an independently interpolated basic solution variable, coupled to the displacement solution through the constitutive theory and the compatibility condition. Herewith, the coupling is implemented by a Lagrange multiplier¹ [32]. This independent interpolation of pressure stress is the basis of 'hybrid' elements. More precisely, they are "mixed formulation" elements, using a mixture of displacement and stress variables with an augmented variational principle to approximate the equilibrium equations and compatibility conditions.

As of now, hybrid elements are not available in Abaqus/Explicit. A fully incompressible constraint can not be imposed in Abaqus/Explicit. Instead, the initial bulk modulus K_0 is 'relaxed' to solve the problem of nearly incompressibility. It has by default a value of $\frac{K_0}{G_0} = 20$, corresponding with $\nu = 0.475$. Since typical unfilled elastomers have ratios $\frac{K_0}{G_0} \in [10^3, 10^4]$ and filled elastomers $\frac{K_0}{G_0} \in [50, 200]$, the modelling is only accurate when the material is relatively unconfined. In case an elastomer is highly confined, like with stiff contact with tarmac, results may not be feasible. The common C3D8R element is used. Details on the element usage will be discussed in 7.4.3.

7.4.2 Mooney-Rivlin

As the Goodyear material model for rubber is classified, alternatively the simpler Mooney-Rivlin material model is discussed. This material model is an additional relation specifying the relation between displacement \mathbf{d} and Cauchy stress tensor $\boldsymbol{\sigma}$. Herewith, (4.2) can be solved.

Elastic materials for which the work is independent of the load path are said to be hyperelastic or Green

¹Lagrange multiplier is explained in [27].

elastic materials. Hyperelastic materials are characterised by the existence of a strain (or stored) energy function that is a potential for stress

$$\boldsymbol{\sigma}_{\text{dev}} = 2 \frac{\partial}{\partial \mathbf{C}} \psi(\mathbf{C}) \quad (7.9)$$

where ψ is the strain energy potential and $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ the right Cauchy-Green deformation tensor. A consequence of the existence of a strain energy function is that the work done on a hyperelastic material is independent of the deformation path. This is confirmed by

$$\frac{1}{2} \int_{\mathbf{C}_1}^{\mathbf{C}_2} \boldsymbol{\sigma}_{\text{dev}} : d\mathbf{C} = \psi(\mathbf{C}_2) - \psi(\mathbf{C}_1)$$

This behaviour is observed approximately in many rubber-like materials [8].

Given isotropy and additive decomposition of the deviatoric and volumetric strain energy contributions in the presence of incompressible or almost incompressible behaviour, the strain energy potential can be written as

$$\exists f, g \quad \ni \quad \psi = \underbrace{f(\bar{I}_1 - 3, \bar{I}_2 - 3)}_{\text{deviatoric}} + \underbrace{g(J_{\text{element}} - 1)}_{\text{volumetric}}$$

with

$$\begin{aligned} \bar{I}_1 &\equiv \text{trace}(\mathbf{B}) \\ \bar{I}_2 &\equiv \frac{1}{2}(\bar{I}_1^2 - \text{trace}(\mathbf{B}^2)) \\ \mathbf{B} &\equiv \mathbf{F}\mathbf{F}^T \\ J_{\text{element}} &= \frac{J}{J_{\text{thermal}}} \\ J_{\text{thermal}} &= (1 + \epsilon_{\text{thermal}})^3 \end{aligned}$$

where linear thermal expansion $\epsilon_{\text{thermal}}$ follows from the temperature and the isotropic thermal expansion coefficient defined by the user. The \bar{I}_1 and \bar{I}_2 are called the first and second invariants of the left Cauchy-Green deformation tensor \mathbf{B} respectively. The J , J_{element} and J_{thermal} are the total, elastic and thermal volume strain.

Setting [32]

$$g = \sum_{i=1}^N \frac{1}{D_i} (J_{\text{element}} - 1)^{2i}$$

and expanding f in a Taylor series, it follows that

$$\psi = \sum_{i+j=1}^N C_{ij} (\bar{I}_1 - 3)^i (\bar{I}_2 - 3)^j + \sum_{i=1}^N \frac{1}{D_i} (J_{\text{element}} - 1)^{2i} \quad , N \in \mathbb{N}.$$

The N is usually not taken higher than two when both the first and second invariants are taken into account. The coefficients C_{ij} and D_i are functions of the temperature defined by the user. The D_i determine the compressibility of the material: if all D_i are zero, the material is fully incompressible. Mooney-Rivlin prescribes that if $D_1 = 0$, then $D_i = 0 \quad \forall i$. Regardless of the value of N , the initial shear and bulk modulus G and K depend only on the polynomial coefficients of order $N = 1$:

$$G = 2(C_{10} + C_{01}) \quad K = \frac{2}{D_1}.$$

If $N = 1$ the Mooney-Rivlin form is recovered

$$\psi = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) + \frac{1}{D_1} (J_{\text{element}} - 1)^2 \quad (7.10)$$

which is an expansion of the Neo-Hookean form

$$\psi = C_{10}(\bar{I}_1 - 3) + \frac{1}{D_1} (J_{\text{element}} - 1)^2. \quad (7.11)$$

This form is the simplest hyperelastic model. A nice detail is that, surprisingly, its representation resembles the model for an ideal gas: the Neo-Hookean represents the Helmholtz free energy of a molecular network with Gaussian chain-length distribution [32].

Note that $\psi = \psi(\bar{I}_1(\mathbf{B}(\mathbf{F}(\mathbf{x}))), \bar{I}_2(\mathbf{B}(\mathbf{F}(\mathbf{x}))))$ and is thus a function of spatial coordinates \mathbf{x} . Together with the relation $p = \frac{\partial \psi}{\partial J}$, we have sufficient information to solve (4.2).

7.4.3 Galerkin's weak formulation

The element used for discretisation is C3D8R. This stands for continuum element (C) in three dimensions (3D) with eight nodes (8) using reduced integration (R). This reduction in amount of integration points can lead to zero energy modes. In case of a structure, this is circumvented by hourglass stiffness control².

The basis functions are, just as in the fluid element case (7.8), trilinear and thus of the form

$$\varphi_i(x, y, z) = a_0^i + a_1^i x + a_2^i y + a_3^i z + a_4^i xy + a_5^i xz + a_6^i yz + a_7^i xyz.$$

The eight coefficients are determined by the demand that it must suffice to the Kronecker delta property in each of the eight nodes, which gives eight equations with eight unknowns.

The weak formulation is obtained by expanding the variable \mathbf{d} in N terms as

$$\mathbf{d}^*(x, y, z, t) = \sum_{j=1}^N \begin{pmatrix} c_j^{d_x}(t) & 0 & 0 \\ 0 & c_j^{d_y}(t) & 0 \\ 0 & 0 & c_j^{d_z}(t) \end{pmatrix} \begin{pmatrix} \varphi_j^{d_x}(x, y, z) \\ \varphi_j^{d_y}(x, y, z) \\ \varphi_j^{d_z}(x, y, z) \end{pmatrix} = \sum_{j=1}^N \mathbf{C}_j(t) \boldsymbol{\varphi}_j(\mathbf{x}), \quad (7.12)$$

multiplying the equations with test function matrix

$$\mathbf{H} = \begin{pmatrix} \eta^{d_x}(t) & 0 & 0 \\ 0 & \eta^{d_y}(t) & 0 \\ 0 & 0 & \eta^{d_z}(t) \end{pmatrix} \in \Sigma_{\mathbf{d}}, \quad (7.13)$$

and integrating over the domain Ω . Herein, $\Sigma_{\mathbf{d}}$ is a Sobolev³ space sufficing to the BCs and ICs for \mathbf{d} . The momentum equations become

$$\begin{aligned} \rho \frac{D^2 \mathbf{d}}{Dt^2} &= \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}(\mathbf{d}) \\ \int_{\Omega} \mathbf{H} \rho_0 \frac{D^2 \mathbf{d}^*}{Dt^2} d\Omega &= \int_{\Omega} \mathbf{H} [\rho_0 \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}(\mathbf{d}^*)] d\Omega \\ \int_{\Omega} \mathbf{H} \rho \frac{D^2}{Dt^2} \left[\sum_{j=1}^N \mathbf{C}_j(t) \boldsymbol{\varphi}_j(\mathbf{x}) \right] d\Omega &= \int_{\Omega} \mathbf{H} \left[\rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma} \left(\sum_{j=1}^N \mathbf{C}_j(t) \boldsymbol{\varphi}_j(\mathbf{x}) \right) \right] d\Omega \\ \rho_0 \int_{\Omega} \mathbf{H} \frac{D^2}{Dt^2} \left[\sum_{j=1}^N \mathbf{C}_j(t) \boldsymbol{\varphi}_j(\mathbf{x}) \right] d\Omega &= \rho_0 \mathbf{g} \int_{\Omega} \mathbf{H} d\Omega + \int_{\Omega} \nabla \cdot \boldsymbol{\sigma} \left(\sum_{j=1}^N \mathbf{C}_j(t) \boldsymbol{\varphi}_j(\mathbf{x}) \right) d\Omega \\ \rho_0 \sum_{j=1}^N \int_{\Omega} \mathbf{H} \frac{D^2 \mathbf{C}_j(t)}{Dt^2} \boldsymbol{\varphi}_j(\mathbf{x}) d\Omega &= \rho_0 \mathbf{g} \int_{\Omega} \mathbf{H} d\Omega + \sum_{j=1}^N \int_{\Omega} \nabla \cdot \boldsymbol{\sigma}(\mathbf{C}_j(t) \boldsymbol{\varphi}_j(\mathbf{x})) d\Omega. \end{aligned} \quad (7.14)$$

The test function matrix is often chosen somewhat similar to the expansion (7.12) in combination with the introduction of a new index $\in \{1, 2, \dots, N\}$. Herewith, and under the condition that the relation between the Cauchy stress tensor $\boldsymbol{\sigma}$ and \mathbf{d}^* is linear, (7.14) can be written in the form of a linear system $\mathbf{S}\mathbf{c} = \mathbf{f}$ [38]. Note that the fluid and structure momentum equations only differ in $\boldsymbol{\sigma}(\mathbf{d})$.

7.5 Fluid-structure interaction

The coupling used by Abaqus' CEL method is a loose partition coupling. Details hereof can be found in 5.2.

As mentioned in 5.4, no details about the method used for surface reconstruction has been provided by Simulia, restricting us in treating it properly in line with this thesis. Because it is only available in Abaqus/Explicit, the reconstruction is probably also done by an explicit method. This was not confirmed. It was expected that some details could be revealed by testing the surface reconstruction with a coarse mesh and small test problems. However, these attempts were in vain. The reasons for this are that a coarse

²Treating energy modes and hourglass control is beyond the scope of this report. A thorough explanation can be found in [8]. Although widely used in industries for the sake of decreasing computation times, hourglass control is not popular in academic environments and is therefore often unknown in these circuits.

³A Sobolev space is a vector space of functions equipped with a norm that is a combination of L_p norms of the function itself as well as its derivatives up to a given order. The derivatives are understood in a suitable weak sense to make the space complete and thus a Banach space. More details to be found in [26].

mesh results in inadequate behaviour of Abaqus and that Abaqus/Viewer, the postprocessor, modifies the visual representation of the surface making evaluation of it useless, cf. Figure 7.4. It shows multiple surface normals per element, which is impossible from the perspective of the surface reconstruction algorithm. It is merely a result of visual enhancement.

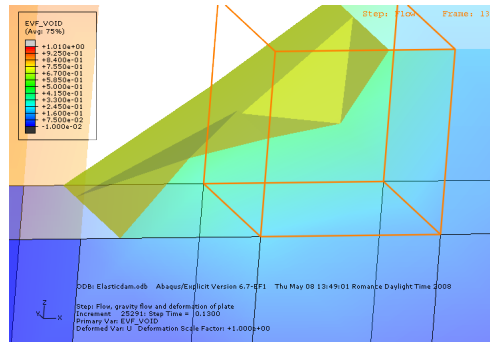


Figure 7.4: multiple surface normals per element.

The general facts known about the surface reconstruction come from the theory of VOF methods, discussed shortly in 5.4. The Eulerian volume fraction (EVF) is treated as a variable per cell. Volume fraction information on the nodes is determined by the arithmetic mean of the EVF values of the cells surrounding the considered node. When $EVF < 0.5$ at a node, contact is not enforced on the fluid located at the node. The fluid can thus 'leak' into a Lagrangian body and so conservation of mass in the flow does not hold anymore. Taking a finer mesh decreases the amount of fluid that is lost due to leakage because the cells are smaller. Furthermore, the leaking problem is self recovering, cf. Figure 7.5. Say $EVF < 0.5$ at a node. Contact is not enforced and the fluid leaks into the cells surrounding the node with $EVF < 0.5$. Herewith the surrounding cells get an higher EVF value, increasing the arithmetic mean to 0.5. Contact is then enforced again.

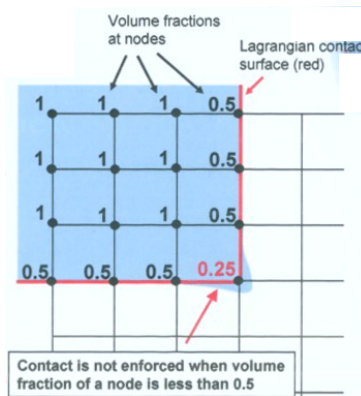


Figure 7.5: leakage.

Concerning surface reconstruction, the threshold of $EVF = 0.5$ also has its influence. When $EVF < 0.5$ for a cell, surface reconstruction is skipped and as a result, the location of the fluid is unknown. Merely the EVF value is known. The fluid is "smeared out" over the cell. When the fluid in these cells have nonzero velocity, they will, because of this smearing, enter the next cell earlier than in the case where there is no smearing. In this next cell, $EVF < 0.5$ again holds and the fluid again enters the next cell too fast. This results in unrealistic acceleration.

7.6 Solving the coupled fluid and structure equations

Equations (7.14) are solved explicitly and subsequently. The explicitness solves the problem of nonlinearity of the convective terms, because they are taken at the previous, known time. This introduces an $O(h_t)$ error. This is however not an issue because explicit calculations are typically done for time steps dominated by solving stability instead of accuracy issues.

The time integration rules used come from the Newmark scheme:

$$\begin{aligned}\dot{\mathbf{c}}^{t_{n+\frac{1}{2}}} &= \dot{\mathbf{c}}^{t_{n-\frac{1}{2}}} + \frac{\Delta t_{n+1} + \Delta t_n}{2} \ddot{\mathbf{c}}^{t_n} \\ \mathbf{c}^{t_{n+1}} &= \mathbf{c}^{t_n} + \Delta t_{n+1} \dot{\mathbf{c}}^{t_{n+\frac{1}{2}}}\end{aligned}\tag{7.15}$$

where $\ddot{\mathbf{c}}^{t_n} = \mathbf{M}^{-1}(\mathbf{F}^{t_n} - \mathbf{I}^{t_n})$ with \mathbf{M} the mass matrix, \mathbf{F} the applied load vector and \mathbf{I} the internal force vector.

The coupling between the structure and fluid equations is subsequent without reiterations per time step, also called loose partition coupling cf. 5.2. Again, an $O(h_t)$ is introduced. In hydroplaning the interaction between structure and fluid is strong, demanding strong coupling. Considering it with a loose partition coupling algorithm demands a considerably smaller time step size.

7.7 Summary of Abaqus' CEL method

Abaqus' CEL method can be summarised in how it treats the fluid, structure and interaction. Focal points for the fluid are

- compressible form of the Navier-Stokes equations,
- displacement is solved instead of velocity, an unique choice,
- Eulerian \rightarrow Lagrangian \rightarrow Eulerian steps,
- Mie-Grüneisen EOS,
- no need to solve the energy equation because of the EOS,
- equations solved with FEM,
- trilinear basis functions,
- reduced integration.

For the structure

- incompressible rubber,
- Goodyear hyperelastic and viscoelastic EOS,
- equations solved with FEM,
- trilinear basis functions,
- reduced integration.

The interaction has as focal points

- surface reconstruction by a PLIC VOF algorithm,
- surface reconstruction algorithm probably explicit, but this is unconfirmed,
- contact *EVF* threshold on node fixed at 0.5,
- leakage possible, but self recovering up to a certain extent,
- surface reconstruction only in elements with $EVF \geq 0.5$ cell value,
- Newmark time integration,
- loose partition coupling, cf. Figure 7.6.

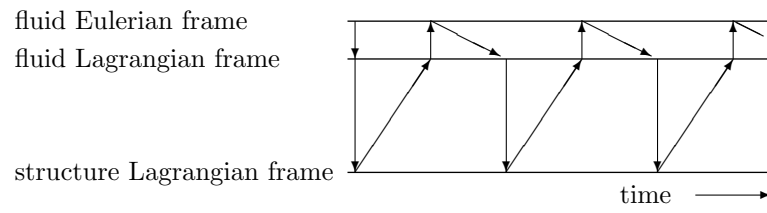


Figure 7.6: subsequent mesh and information treatment of the CEL method.

Part II

Test models

Chapter 8

Couette flow

8.1 Introduction

The CEL method of Abaqus needs to be tested on accuracy, computational speed and plausibility of the produced solution. This kind of testing is also referred to as benchmarking. Herefor, we need to treat an example problem. Preferably a problem whereof the solution is already known. The ideal case being that an analytical solution is known.

Couette flow is a simple model to simulate viscous flow [18]. Although there is interaction between a fluid and a structure, it is not a typical FSI problem because the structure is not subjected to changes throughout space or time as a result of the fluid. However, it gives a good introduction in fluid dynamics and can be solved analytically. It is programmable in both Matlab and Abaqus and thus ideal for benchmarking the fluid part of the CEL method.

The Matlab analysis were run with AMD Opteron dual core 2.8 GHz CPUs and the Abaqus analysis were divided over the clusters of AMD Opteron single core 2.4 GHz, dual core 2.2 GHz and dual core 2.8 GHz. Of the latter the dual core 2.2 GHz is in the majority.

Additional output is available in movie form, to be found at [2].

8.2 Model

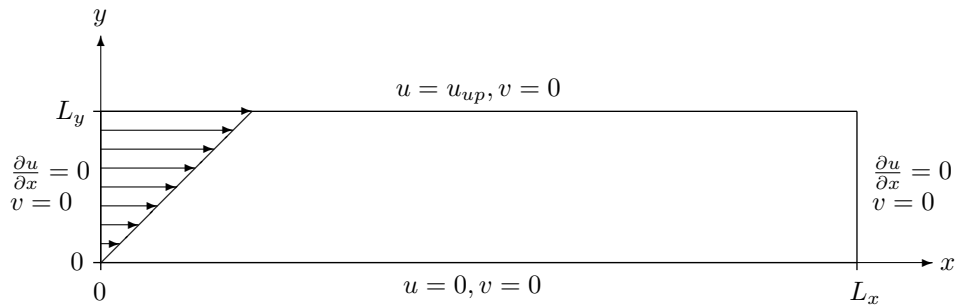


Figure 8.1: Couette flow [5].

Consider a flow between two rigid parallel plates vertically separated by a distance D , cf. Figure 8.1. The upper plate is moving with velocity $u = u_{up}$ and the lower plate with $u = 0$, providing two BCs. The flow field between the two plates is driven exclusively by the shear stress exerted on the fluid by the moving upper plate. The initial velocity is zero on the whole domain.

As for the in- and outflow, the BCs have to be chosen carefully [6, 30, 24]. It is known that for incompressible flow no explicit BCs are given for the pressure [30]. Usually BCs for the pressure are given implicitly by prescribing the normal stress. The following types of BCs are commonly used for the two-dimensional incompressible Navier-Stokes equations:

- \mathbf{v} given,
- v_n and σ_{nt} given,
- v_t and σ_{nn} given,
- σ_{nn} and σ_{nt} given.

Here, $v_n = \mathbf{v} \cdot \mathbf{n}$ and $v_t = \mathbf{v} \cdot \mathbf{t}$ denote the normal and tangential components of the velocity on the boundary. The $\sigma_{nn} = \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{n}$ and $\sigma_{nt} = \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{t}$ denote the normal and tangential components of the Cauchy stress tensor $\boldsymbol{\sigma}$.

At the in- and outflow one may prescribe the velocity. However, for convection dominated flows, such a BC may lead to oscillations due to inaccuracies of the BCs. Less restrictive BCs are for example $v_t = 0 \wedge \sigma_{nn} = 0$ or $\sigma_{nn} = 0 \wedge \sigma_{nt} = 0$. The former prescribes a parallel flow with zero normal stress. From (3.23) in combination with the incompressibility condition $\nabla \cdot \mathbf{v} = 0$ it follows for the dimensionless form that

$$\sigma_{nn} = -p + \frac{2}{Re} \frac{\partial v_n}{\partial n} \quad (8.1)$$

$$\sigma_{nt} = \frac{1}{Re} \left(\frac{\partial v_n}{\partial t} + \frac{\partial v_t}{\partial n} \right). \quad (8.2)$$

As a consequence, for high Reynolds numbers, $\sigma_{nn} \approx -p$, so $p \approx 0$.

For both in- and outflow we use the BCs $v_t = 0 \wedge \sigma_{nn} = 0 \Leftrightarrow v = 0 \wedge \sigma_{xx} = 0$. In combination with the incompressibility condition it follows that $\frac{\partial u}{\partial x} = 0$.

8.3 Analytical solution

Both the stationary and instationary case of Couette flow can be solved analytically. We will treat them shortly.

8.3.1 Stationary case

The problem is solved by considering the two dimensional version of the Navier-Stokes equations (3.26)

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} &= 0 \\ \rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) &= \rho \mathbf{f} - \nabla p + \tau \nabla. \end{aligned} \quad (8.3)$$

With Couette flow and additional assumptions, (8.3) is simplified:

- incompressible flow $\Leftrightarrow \rho = \text{constant}$
- Newtonian shear cf. 3.6.1
- infinitely long plates \Leftrightarrow (8.3) independent of x
- steady state \Leftrightarrow (8.3) independent of t
- no body forces $\Leftrightarrow \mathbf{f} = \mathbf{0}$.

Herewith (8.3) becomes

$$\begin{aligned} \frac{dv}{dy} &= 0 \\ \rho \begin{pmatrix} v \frac{du}{dy} \\ 0 \end{pmatrix} &= \begin{pmatrix} 0 \\ -\frac{dp}{dy} \end{pmatrix} + \mu \frac{d^2 \mathbf{v}}{dy^2} \end{aligned} \quad (8.4)$$

with u, v and p functions of y only. Considering the continuity equation it follows that

$$\frac{dv}{dy} = 0 \Leftrightarrow v = \text{constant}.$$

With one of the BCs $v(y)|_{y \in \{0, D\}} = 0$ it follows that $v \equiv 0$. As a consequence

$$\begin{aligned} \rho v \frac{du}{dy} = 0 &\Rightarrow \mu \frac{d^2 u}{dy^2} = 0 \\ &\Rightarrow \frac{d^2 u}{dy^2} = 0 \\ &\Rightarrow u = c_1 y + c_2 \end{aligned}$$

with c_1, c_2 constants. Remember that $u(0) = 0, u(D) = u_{up}$ which determines c_1 and c_2 . It follows

$$u(y) = \frac{u_{up}}{D} y.$$

Recalling that $v \equiv 0 \Rightarrow \frac{d^2 v}{dy^2} = 0$, we have for the last equation to be solved

$$-\frac{dp}{dy} = 0 \Leftrightarrow p = \text{constant}$$

where the constant follows from a prescribement.

8.3.2 Instationary case

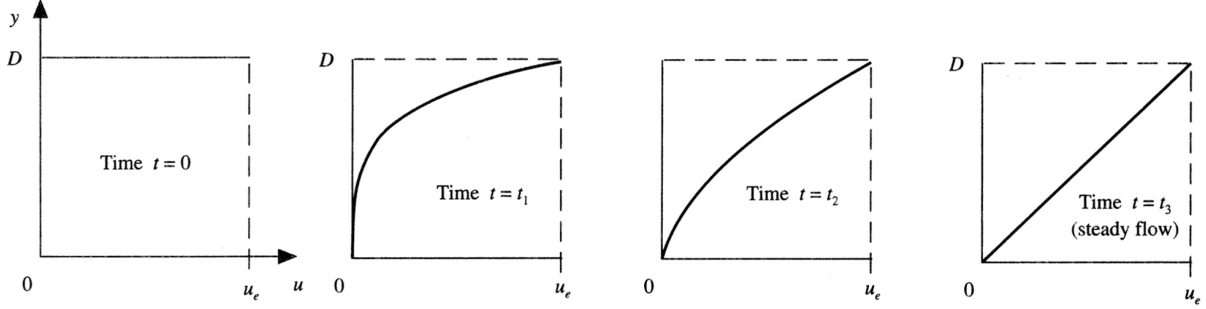


Figure 8.2: instationary Couette flow for various times [18].

The instationary case is more difficult. However, we already know from 8.3.1 that $\lim_{t \rightarrow \infty} u(y, t) = \frac{u_{up}}{D}y$. The IC for u is

$$u(y, 0) = \begin{cases} u_{up} & y = D \\ 0 & \text{otherwise} \end{cases} \quad (8.5)$$

cf. Figure 8.2. As a result of shear stresses $\tau_{yx} = \tau_{xy}$ and the no slip condition at the top and bottom plate, we can expect a progress in time as depicted in Figure 8.2. The infinitesimal film of fluid at the top plate with a velocity drags the 'next' layer with it. Subsequently, the next layer is dragged, and so on until a linear profile is achieved. The drag effect is linearly proportional to the kinematic viscosity and velocity gradient.

Cancelling the stationarity assumption from 8.3.1, (8.4) becomes

$$\begin{aligned} \frac{\partial v}{\partial y} &= 0 \\ \rho \left(\frac{\partial \mathbf{v}}{\partial t} + \begin{pmatrix} v \frac{\partial u}{\partial y} \\ 0 \end{pmatrix} \right) &= \begin{pmatrix} 0 \\ -\frac{\partial p}{\partial y} \end{pmatrix} + \mu \frac{\partial^2 \mathbf{v}}{\partial y^2} \end{aligned} \quad (8.6)$$

with u, v and p functions of y and t . From the continuity equation

$$\frac{\partial v}{\partial y} = 0 \Rightarrow \frac{\partial^2 v}{\partial y^2} = 0 \wedge \exists f : [t_0, \infty) \rightarrow \mathbb{R} \ni v = f(t).$$

Recall that $\mathbf{v}(y, 0) = \mathbf{0}$, which implies $v \equiv 0$. Herewith, we get from the momentum equation in y direction

$$0 = -\frac{\partial p}{\partial y} \Rightarrow p = p(t)$$

Choosing $p = p_0$ at of the boundaries we get $p \equiv p_0$.

The momentum equation in x direction simplifies to

$$\rho \frac{\partial u}{\partial t} = \mu \frac{\partial^2 u}{\partial y^2} \quad (8.7)$$

which is in heat equation without energy source form. Together with (8.5), $u(0, t) = 0$ and $u(D, t) = u_{up}$ equation (8.7) can be solved by the method of separation of variables. The solution is [14]

$$u(y, t) = \frac{u_{up}}{D}y + \sum_{n=1}^{\infty} \frac{D^2}{n\pi} (-1)^{n+1} e^{-\frac{\mu}{\rho} \left(\frac{n\pi}{D}\right)^2 t} \sin\left(\frac{n\pi}{D}y\right) \quad (8.8)$$

which indeed represents the form depicted in Figure 8.2. It is the result of superposition of an infinite amount of solutions, having $\frac{u_{up}}{D}y$, the first term, as stationary solution.

8.4 Matlab solution

Discretising the Navier-Stokes equations has been known to have its share of problems, in particular the incompressible case [6, 24]. For comparison reasons, the instationary two dimensional case will be treated here.

8.4.1 Handling assumptions

In 8.3.1 and 8.3.2 certain assumptions were made to simplify (3.26). In case of considering the construction of a discretisation, two assumptions protrude, namely the assumptions of infinite plates and stationarity. We emphasise these two assumptions because they are not as trivial in programming as they are in calculus. For programming in Matlab, the infinite plates assumption demands special attention. In Abaqus/CAE, both assumptions demand special attention, cf. 8.5.

Instationarity is for programming in Matlab as trivial as it is with calculus: leave out the time derivative term. The infinite plates assumption, however, is less trivial. The cause for this is that ∞ does not exist in programming language. Assigning the plates a length as high as floating numbers admit would imply unattractively much computational time. Taking the inflow of time t_{n+1} equal to the outflow of t_n , $u_{in}^{t_{n+1}} = u_{out}^{t_n}$, is the solution. In this way, the flow never 'ends' anywhere. Thus we have infinitely long plates and the flow will be independent of x . The domain to be considered can now be chosen of an arbitrary length, which emphasises the problem being independent of x .

Verifying the CEL method in Abaqus/Explicit involves evaluating space and time dependent behaviour. The Matlab models serve for comparison and therefor we will only treat models that are worth comparison. For this reason, we will treat instationary two dimensional case.

8.4.2 Spatial discretisation

The first problem that arises is the choice of the spatial discretisation. This holds for finite difference, volume and element methods (FDM, FVM and FEM) [6]. When a collocated grid is used, meaning that every variable is placed on the nodes, the infamous checkerboard pattern can appear. This is due to the simplicity of the discrete operator of the continuity equation when central difference is used. As a result, the operator has a nullspace $\{\mathbf{y} \mid \mathbf{A}_c \mathbf{y} = \mathbf{0}\}$ whereof the elements can, multiplied by a constant, be added to solution \mathbf{x} , making it non unique. Purpose is to find a geometric ordering of the variables such that the nullspace of the discrete operator is empty.

A choice often made, is the staggered grid as depicted in Figure 8.3: the velocity unknowns u and v are placed at the midpoints of the vertical and horizontal faces of the cells respectively, and the pressure unknowns at the cell centres. Consequently, equations (8.3) are also considered in different locations. The continuity equation is considered in the cell centres, the momentum equations in x and y direction in the midpoints of the vertical and horizontal faces of the cells respectively.

Say the rectangular domain Ω with dimensions $L_x \times L_y$ is divided into n_x and n_y elements in x and y direction respectively. In this way the cell centres are represented by the coordinates

$$(x, y) \in \left\{ \begin{aligned} &\left(\frac{h_x}{2}, \frac{h_y}{2}\right), \left(\frac{3h_x}{2}, \frac{h_y}{2}\right), \dots, \left(L_x - \frac{h_x}{2}, \frac{h_y}{2}\right), \\ &\left(\frac{h_x}{2}, \frac{3h_y}{2}\right), \left(\frac{3h_x}{2}, \frac{3h_y}{2}\right), \dots, \left(L_x - \frac{h_x}{2}, \frac{3h_y}{2}\right), \\ &\vdots \\ &\left(\frac{h_x}{2}, L_y - \frac{h_y}{2}\right), \left(\frac{3h_x}{2}, L_y - \frac{3h_y}{2}\right), \dots, \left(L_x - \frac{h_x}{2}, L_y - \frac{3h_y}{2}\right) \end{aligned} \right\} \\ := \Omega_{cc},$$

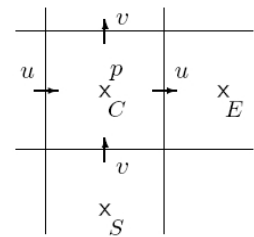


Figure 8.3: staggered grid [6].

the midpoints of the vertical cell faces by

$$(x, y) \in \left\{ \begin{aligned} &\left(h_x, \frac{h_y}{2}\right), \left(2h_x, \frac{h_y}{2}\right), \dots, \left(L_x - h_x, \frac{h_y}{2}\right), \\ &\left(h_x, \frac{3h_y}{2}\right), \left(2h_x, \frac{3h_y}{2}\right), \dots, \left(L_x - h_x, \frac{3h_y}{2}\right), \\ &\vdots \\ &\left(h_x, L_y - \frac{h_y}{2}\right), \left(2h_x, L_y - \frac{3h_y}{2}\right), \dots, \left(L_x - h_x, L_y - \frac{3h_y}{2}\right) \end{aligned} \right\} \\ := \Omega_{vm}$$

and the midpoints of the horizontal faces by

$$(x, y) \in \left\{ \begin{aligned} &\left(\frac{h_x}{2}, h_y\right), \left(\frac{3h_x}{2}, h_y\right), \dots, \left(L_x - \frac{h_x}{2}, h_y\right), \\ &\left(\frac{h_x}{2}, 2h_y\right), \left(\frac{3h_x}{2}, 2h_y\right), \dots, \left(L_x - \frac{h_x}{2}, 2h_y\right), \\ &\vdots \\ &\left(\frac{h_x}{2}, L_y - h_y\right), \left(\frac{3h_x}{2}, L_y - h_y\right), \dots, \left(L_x - \frac{h_x}{2}, L_y - h_y\right) \end{aligned} \right\} \\ := \Omega_{hm}.$$

The number of elements in Ω_{cc} , Ω_{vc} and Ω_{hc} is $n_x n_y$, $n_y(n_x - 1)$ and $n_x(n_y - 1)$ respectively. The total number of equations or unknowns is therefor $n_x n_y + n_y(n_x - 1) + n_x(n_y - 1) = 3n_x n_y - n_x - n_y$

For simplicity, (i, j) coordinates are introduced by dividing the (x, y) coordinates by $h_x = \frac{L_x}{n_x}$ and $h_y = \frac{L_y}{n_y}$ respectively.

8.4.3 Continuity equation

The continuity equation is considered in the $n_x n_y$ cell centres Ω_{cc} . Using central difference we obtain

$$\begin{aligned} \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{h_x} + O(h_x^2) + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{h_y} + O(h_y^2) &= 0 \\ \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{h_x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{h_y} + O(\max\{h_x, h_y\}^2) &= 0 \end{aligned} \quad (8.9)$$

which is used to construct the system $\mathbf{A}_c \mathbf{x}_c = \mathbf{b}_c$, with \mathbf{A}_c a sparse four diagonal matrix and \mathbf{b}_c such that the BC $u = u_{up}$ is taken into account. Matrix \mathbf{A}_c is however not symmetric because of the $O(h_x^3)$ one sided difference for $x \in \{0, L\}$. Divided by h_x this gives the maintained second order $O(h_x^2)$.

8.4.4 Implications of the initial and boundary conditions

Before continuing with the momentum equations, we consider the ICs and BCs mentioned in 8.2 to get information about pressure p at the boundaries. Depending on the strategy to solve the equations, this information is required or not for solving the momentum equations. If it is not, this information is processed implicitly by the BCs for the velocity \mathbf{v} .

To start simple, consider the homogeneous Dirichlet no slip BC $\{u = v = 0 | x \in [0, L], y = 0, t \in [t_0, \infty)\}$. Because x and t are not fixed, the BCs are independent of x and t . Consequently, the continuity equation implies

$$\begin{aligned} [\nabla \cdot \mathbf{v}]_{y=0} &= 0 \\ \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right]_{y=0} &= 0 \\ \frac{\partial v}{\partial y} \Big|_{y=0} &= 0 \end{aligned}$$

and the momentum equations

$$\begin{aligned} \left[\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) \right]_{y=0} &= \left[-\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right]_{y=0} \\ 0 &= -\frac{\partial p}{\partial x} \Big|_{y=0} + \mu \frac{\partial^2 u}{\partial y^2} \Big|_{y=0} \end{aligned}$$

and

$$\begin{aligned} \left[\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) \right]_{y=0} &= \left[-\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \right]_{y=0} \\ 0 &= -\frac{\partial p}{\partial y} \Big|_{y=0} + \mu \frac{\partial^2 v}{\partial y^2} \Big|_{y=0}. \end{aligned}$$

For $y = D$ we have $u = u_{up}$, $v = 0$, which would change the terms $u \frac{\partial u}{\partial x}$ and $u \frac{\partial v}{\partial x}$. However, $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0$ so the conditions for p remain the same for $y = D$.

The ICs and BCs at $x = 0$ and $x = L$ also have implications for the pressure p . Consider $\{u, v = 0 | x \in \{0, L\}, y \in [0, H], t \in [t_0, \infty), \sigma_{nn} = 0\}$. With $v = 0$ the continuity equation gives $\frac{\partial u}{\partial x} = 0$. Recall that when Newtonian shear cf. 3.6.1 and incompressibility is assumed, (8.1) holds:

$$\sigma_{nn} = -p + \frac{2}{Re} \frac{\partial v_n}{\partial n}.$$

So in our case

$$\sigma_{xx} = -p + \frac{2}{Re} \frac{\partial u}{\partial x}.$$

With $\sigma_{xx} = 0$ and recalling that $\frac{\partial u}{\partial x} = 0$ it follows that

$$p = \frac{2}{Re} \frac{\partial u}{\partial x} = 0$$

which implies $\frac{\partial p}{\partial y} = 0$. From the momentum equation in x direction we get for $\frac{\partial p}{\partial x}$

$$\begin{aligned} \left[\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) \right]_{x=L} &= \left[-\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right]_{x=L} \\ \rho \frac{\partial u}{\partial t} \Big|_{x=L} &= -\frac{\partial p}{\partial x} \Big|_{x=L} + \mu \left(\frac{\partial^2 u}{\partial x^2} \Big|_{x=L} + \frac{\partial^2 u}{\partial y^2} \Big|_{x=L} \right). \end{aligned}$$

Summarised we have for the pressure

- $y \in \{0, H\} : \nabla p = \mu \frac{\partial^2 v}{\partial y^2}$
- $x \in \{0, L\} : p = 0, \frac{\partial p}{\partial x} = \mu \nabla^2 u - \rho \frac{\partial u}{\partial t}, \frac{\partial p}{\partial y} = 0.$

8.4.5 Momentum equations

The momentum equations in x and y direction are considered in the midpoints of the vertical and horizontal faces of the cells respectively. We first consider the momentum equation in x direction on Ω_{vm} :

$$\begin{aligned} \rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) &= -\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \frac{\partial u}{\partial t} &= \frac{\mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x}}{\rho} - u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y}. \end{aligned} \quad (8.10)$$

Discretising (8.10) gives

$$\begin{aligned} \frac{\partial u}{\partial t} \Big|_{(i,j) \in \Omega_{vm}} &= \frac{\mu \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} \right) - \frac{P_{i+\frac{1}{2},j} - P_{i-\frac{1}{2},j}}{h_x}}{\rho} \\ &\quad - u_{i,j} \frac{u_{i+1,j} - u_{i-1,j}}{2h_x} \\ &\quad - \left(\frac{v_{i-\frac{1}{2},j-\frac{1}{2}} + v_{i+\frac{1}{2},j-\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2}}}{4} \right) \frac{u_{i,j+1} - u_{i,j-1}}{2h_y} + O(\max\{h_x, h_y\}^2). \end{aligned} \quad (8.11)$$

Note that, excluding the order term, the second and third term are nonlinear, so they can not be denoted in matrix times vector form \mathbf{Ax} .

The consideration of the momentum equation in y direction goes in a similar way:

$$\begin{aligned} \rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) &= -\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \\ \frac{\partial v}{\partial t} &= \frac{\mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial y}}{\rho} - u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y}. \end{aligned} \quad (8.12)$$

Discretising (8.12) gives

$$\begin{aligned} \frac{\partial v}{\partial t} \Big|_{(i,j) \in \Omega_{hm}} &= \frac{\mu \left(\frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{h_x^2} + \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{h_y^2} \right) - \frac{p_{i,j+\frac{1}{2}} - p_{i,j-\frac{1}{2}}}{h_y}}{\rho} \\ &\quad - \left(\frac{u_{i-\frac{1}{2},j-\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}} + u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}}}{4} \right) \frac{v_{i+1,j} - v_{i-1,j}}{2h_x} \\ &\quad - v_{i,j} \frac{v_{i,j+1} - v_{i,j-1}}{2h_y} + O(\max\{h_x, h_y\}^2). \end{aligned} \quad (8.13)$$

As with the momentum equation for the x direction, the second and third term are nonlinear so they can not be denoted in matrix times vector form \mathbf{Ax} . Note that both (8.11) and (8.13) have a local truncation error of $O(\max\{h_x, h_y\}^2)$ as a result of the use of central discretisation in both x and y direction. The BCs are implemented in a way to maintain this order. Ghost points and up to fourth order one sided difference is used to accomplish this.

The system obtained from (8.11) and (8.13) is in the form

$$\frac{\partial \mathbf{x}_m}{\partial t} = \mathbf{A}_m \mathbf{x} + \mathbf{b}_m \quad (8.14)$$

where the subscript m denotes momentum. Vector \mathbf{x} contains all the unknowns. Vector $\mathbf{x}_m = \mathbf{x}_c$ does not contain the pressure unknowns. Sparse seven diagonal matrix \mathbf{A}_m contains the coefficients and \mathbf{b}_m the nonlinear part.

8.4.6 Final system

A system of $3n_x n_y - n_x - n_y$ equations needs to be constructed from (8.9) and (8.14). They are, however, not of the same form. Equation (8.14) needs to be treated by a time integration method. There is a set of time integration methods to choose from, the simplest being the explicit Euler forward and implicit Euler backward. Although the choice of Euler forward might seem easier, it is in this case not, to be shown hereafter. The final choice will be the implicit Euler backward time integration.

Say we apply the $O(h_t^2)$ Euler forward time integration on (8.14)

$$\frac{\partial \mathbf{x}_m}{\partial t} = \mathbf{A}_m \mathbf{x} + \mathbf{b}_m.$$

The obtained system is

$$\mathbf{x}_m^{t_{n+1}} = \mathbf{x}_m^{t_n} + h_t (\mathbf{A}_m \mathbf{x}_m^{t_n} + \mathbf{b}_m^{t_n}). \quad (8.15)$$

Although the pressure is involved in the calculation, only the velocity unknowns are determined. The pressure unknowns at the new time increment would need to follow from the continuity equation in combination with the now known velocities. The continuity equation however, does not contain any pressure unknowns. This problem is solved by introducing the pressure correction method [18, 40]. This is more cumbersome than applying Euler backward, therefore we continue with the latter.

Say we apply the unconditionally stable $O(h_t^2)$ Euler backward time integration on (8.14)

$$\frac{\partial \mathbf{x}_m}{\partial t} = \mathbf{A}_m \mathbf{x} + \mathbf{b}_m.$$

The obtained system is

$$\mathbf{x}_m^{t_{n+1}} = \mathbf{x}_m^{t_n} + h_t (\mathbf{A}_m \mathbf{x}_m^{t_{n+1}} + \mathbf{b}_m^{t_{n+1}}). \quad (8.16)$$

But (8.16) is still not in an usable form because it contains both $\mathbf{x}_m^{t_{n+1}}$ and $\mathbf{x}^{t_{n+1}}$. This can be resolved by noting that

$$\underbrace{\begin{pmatrix} 1 & & & 0 & 0 & \dots & 0 \\ & 1 & & 0 & 0 & \dots & 0 \\ & & \ddots & \vdots & \vdots & \ddots & 0 \\ & & & 1 & 0 & \dots & 0 \\ & & & & 0 & 0 & \dots & 0 \end{pmatrix}}_{2n_x n_y - n_x - n_y \times 3n_x n_y - n_x - n_y} \underbrace{\mathbf{x}}_{3n_x n_y - n_x - n_y \times 1} = \underbrace{\mathbf{x}_m}_{2n_x n_y - n_x - n_y \times 1},$$

which we will denote with $\mathbf{I}_m \mathbf{x} = \mathbf{x}_m$. Furthermore, the term $h_t \mathbf{b}_m^{t_{n+1}}$ introduces a nonlinearity. Nonlinearity can be solved by the means of for instance the Newton-Raphson or Picard method [38]. Both methods require iterations which makes them computationally expensive. As an alternative, we take vector $\mathbf{b}_m^{t_n}$ instead of $\mathbf{b}_m^{t_{n+1}}$ to resolve the problem of nonlinearity, since $\mathbf{b}_m^{t_n}$ is known. From Taylor expansion, it follows that this approximation introduces an error of $O(h_t)$. The low order hereof results in the requirement of a small time step size for stability. Note that with the latter approximation, the time integration is not the genuine Euler backward and in effect not unconditionally stable. When using the Newton-Raphson or Picard method, the latter property would be conserved allowing a larger time step size. Equation (8.16) becomes

$$\begin{aligned} \mathbf{I}_m \mathbf{x}^{t_{n+1}} &= \mathbf{I}_m \mathbf{x}^{t_n} + h_t (\mathbf{A}_m \mathbf{x}^{t_{n+1}} + \mathbf{b}_m^{t_n}) \\ (\mathbf{I}_m - h_t \mathbf{A}_m) \mathbf{x}^{t_{n+1}} &= \mathbf{I}_m \mathbf{x}^{t_n} + h_t \mathbf{b}_m^{t_n}, \end{aligned} \quad (8.17)$$

where $\mathbf{b}_m^{t_n}$ is a function of \mathbf{x}^{t_n} . Since the initial velocity field is $\{u = v = 0 | u, v \in \Omega\}$ which suffices to the incompressibility condition $\nabla \cdot \mathbf{v} = 0$, it is self-evident to take the continuity equation at time t_{n+1} : $\mathbf{A}_c \mathbf{x}_c^{t_{n+1}} = \mathbf{b}_c$. Together with (8.17) and adding $n_x n_y$ zero columns, this completes the system:

$$\begin{pmatrix} \mathbf{I}_m - h_t \mathbf{A}_m \\ \mathbf{A}_c \vdots \emptyset \end{pmatrix} \mathbf{x}^{t_{n+1}} = \begin{pmatrix} \mathbf{I}_m \mathbf{x}^{t_n} + h_t \mathbf{b}_m^{t_n} \\ \mathbf{b}_c^{t_n} \end{pmatrix}, \mathbf{x}^{t_0} = \mathbf{0}, \quad (8.18)$$

or $\mathbf{A} \mathbf{x} = \mathbf{b}$ in short. The hierarchy momentum in x direction, momentum in y direction, continuity in the system is conform literature. In this way, the equations have the same order as the variables in the vector with unknowns. The momentum in x direction mainly solves the u unknowns, the momentum in y direction mainly solves the v unknowns and the continuity equation mainly solves the p unknowns. The unknowns are ordered similarly. This results in a matrix with with nonzero elements around the diagonal, also called a banded structure. The majority of solving algorithms benefit from this structure.

The condition number is of $O(10^{17})$, which is large. This is the result of the presence of two eigenvalues with a real part of $O(10^{-16})$, troubling calculations in the sense of computer precision. This is a result of the referencelessness of p , as mentioned in 3.4. Moreover, it is confirmed by the existence of a nullspace of \mathbf{A} . Say we choose a vector \mathbf{y} whereof the u and v unknowns are taken zero. Operator \mathbf{A}_c does not contain nonzero columns for the pressure unknowns so $\mathbf{A}_c \mathbf{y} = \mathbf{0}$. Neither does \mathbf{I}_m , thus only the symmetric pressure part of \mathbf{A}_m remains. The symmetry implies that $p = \text{constant}$, so that $(00 \dots 0 p_0 p_0 \dots p_0)^T \in \text{nullspace}(\mathbf{A})$ with $p_0 \in \mathbb{R}$ arbitrary. To resolve this referencelessness, one can correct the pressure calculations by setting $\min(p) = 0$. An alternative is replacing one continuity equation with $p = \text{constant}$ which sets a reference value for the pressure p . For reasons unknown, the latter strategy does not work with the program in question. Aside from the two near singularities, the eigenvalues are distributed well, cf. Figure 8.4.

Gaussian elimination is used to solve this initial value problem (IVP).

8.4.7 Solution

The considered domain is $(x, y, t) \in [0, 0.4] \times [0, 0.4] \times [0, 3)$. The density is constant on $998 \frac{\text{kg}}{\text{m}^3}$. In case of Couette flow, the Reynolds number is characterised by $Re = \frac{\rho_0 u_{up} H}{\mu}$.

The first thing noticed when running the Couette flow model in Matlab is the size of the critical time step, found by trial and error. It needs to be very small for stability. The parameters u_{up} and μ play an important role here. They strongly influence the size of the time step, cf. Table 8.1. A larger u_{up} implies larger changes in speed, which means a smaller time step is needed to 'catch' these changes in time. A

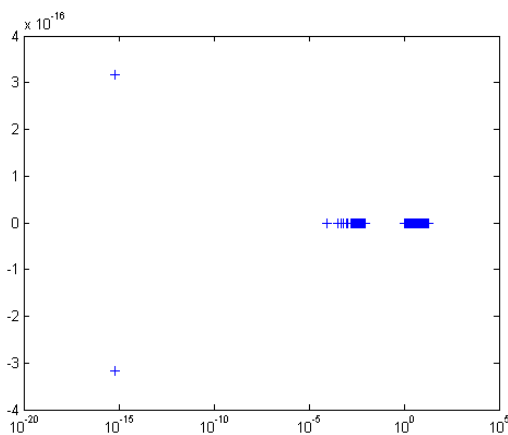


Figure 8.4: eigenvalues system matrix Couette flow in complex plane with logarithmic scale.

n_x	n_y	#DOF	$h_t[s]$	$u_{up}[\frac{m}{s}]$	$\mu[Pa \cdot s]$	Re	$t_{CPU}[s]$
50	50	7400	0.01	1	1	100	508
60	60	10680	0.01	1	1	100	799
40	40	4720	0.00125	5	1	500	3410
50	50	7400	0.00125	5	0.1	5000	5634
60	60	10680	0.0005	10	1	1000	16231
70	70	14560	0.0005	10	1	1000	22672

Table 8.1: results Couette flow model in Matlab.

smaller viscosity μ implies a weaker ability to transfer local energy changes to its surroundings by shear stress. Thus, local changes stay more local compared to the case with a larger μ and again a smaller time step is needed to 'catch' these changes.

The solving of the dimensional form of the Navier-Stokes equations instead of the dimensionless form reveals some unexpected results. Namely, that even if the Reynolds number is kept constant, simulation time still depends on the combination of the parameters used in the Reynolds number. This is reconfirmed in chapter 9. If the dimensionless form were to be solved, the only parameter present to vary is the Reynolds number, cf. (3.42). Varying the parameters in the Reynolds number, but keeping it constant would not make a difference in this case. This notion shows that although the complexity of the flow in the physical sense is mainly determined by the Reynolds number, this is not the case in mathematical sense when the dimensional form is considered. The condition number is the most important in the mathematical considerations. The results obtained in chapter 9 are somewhat more convincing with regard to this issue.

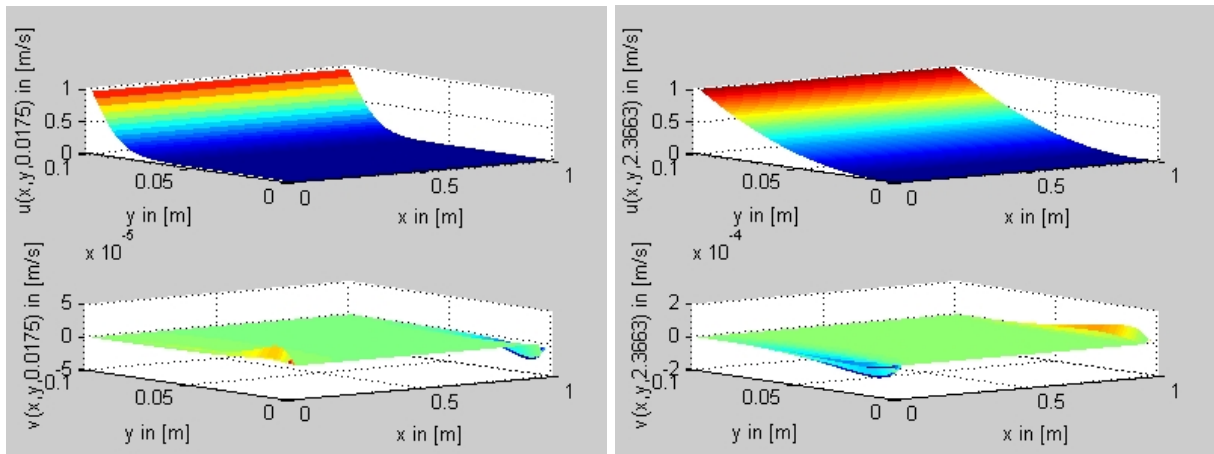
The condition number of the system matrix plays a role in the time step size. As mentioned in 8.4.6, it is on the high side which results in inaccurate calculations. To prevent these inaccuracies becoming too large, a smaller time step is needed.

When the time step is chosen sufficiently small, the simulation runs properly. The shear stresses result in a linear profile for the tangential velocity and the normal velocity which is of $O(10^{-4})$ can be neglected, cf. Figure 8.5.

8.5 Abaqus solution

In this section the modelling of Couette flow with Abaqus is treated. The usage of Abaqus/CAE for CAE model construction is shortly explained. Conform the scope of this report, the emphasise is typically on the solution that Abaqus/Explicit produces and the interpretation of it.

For a short introduction to the usage Abaqus, in particular the modelling of problems involving both Eulerian and Lagrangian grids, we refer to appendices A.2 and A.3. Herein, the used terminology is also explained.

(a) $\mu = 4[Pa \cdot s], t = 0.0175[s]$.(b) $\mu = 4[Pa \cdot s], t = 2.3663[s]$.Figure 8.5: $u, v[\frac{m}{s}]$ at different times.

8.5.1 Formulation

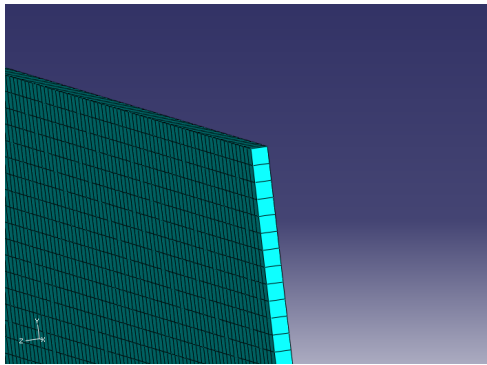


Figure 8.6: Eulerian part for Couette flow, one element thick making it a two dimensional problem.

An Eulerian part is defined for the rectangular fluid domain. Its edges in x and y direction are meshed with multiple elements. The z is meshed with only one element, cf. Figure 8.6. In this way, usage of a three-dimensional Eulerian part does not prohibit us from defining a two-dimensional domain without redundant calculations in the z direction. Via section assignment and predefined field, the material filling the Eulerian part is prescribed. The characteristics hereof are defined by the three parameters density, speed of sound and kinematic viscosity. To prevent dissipation of fluid through the xy and xz faces, normal velocities BCs of zero are defined on these faces.

As always, the BCs at the inflow and outflow should be chosen carefully [6, 30]. When omitted, Abaqus/Explicit assumes the stress free Neumann BC $\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0$. BCs for either u, v or p can be prescribed. Their influence is discussed in 8.5.2.

Finishing the construction of the Couette flow model in Abaqus/CAE can be done in two different ways. The first way is by defining additional BCs. Namely no slip on the xy faces and a velocity in x direction on the upper xy face. This simulates the upper plate moving with a velocity. Modellingwise this is the easiest. The second way is defining zero velocity in all three DOFs for the lower xy face, as in the first way, but modelling the upper plate by defining a Lagrangian body entering the Eulerian domain with a certain constant velocity. Rough contact is defined for no slip between the plate and the fluid.

The latter constructed model was not run due to a shortage of CPUs and Abaqus licenses at the time. It is, however, contained in the models package, which can be found at [2].

8.5.2 Solution

We start with considering the case where the upper plate is modeled by means of BCs. For the upper plate a velocity of $10[\frac{m}{s}]$ is taken. The fluid properties are taken for water, so $\rho = 998[\frac{kg}{m^3}]$, $c = 1483[\frac{m}{s}]$, $\mu = 0.001[Pa \cdot s]$. For convenience, we start with the unrealistic dynamic viscosity of honey $\mu = 4[Pa \cdot s]$ to speed up the process. When we choose the velocity of the upper plate and the width of the channel as characteristic velocity and length respectively, the Reynolds numbers are $9.98 \cdot 10^5$ and $2.495 \cdot 10^2$. Recall that turbulent flow typically has a Reynolds number of $O(10^3)$ or $O(10^4)$.

The results are represented by Figure 8.7 and Table 8.2. As mentioned, we start with the unrealistic $\mu = 4[Pa \cdot s]$ instead of the realistic $\mu = 0.001[Pa \cdot s]$. Herewith, the shear stresses are a factor four thousand bigger, thus increasing the ability of the fluid to transfer energy to its surroundings. The increase results in stabilisation in a shorter time, namely $1.5[s]$, cf. Figure 8.7(c). For $\mu = 1[Pa \cdot s]$ this time is $3[s]$, cf. Figure 8.7(b). The needed simulation time for $\mu = 0.001[Pa \cdot s]$ is excessive, cf. Figure 8.7(a). Therefor we will skip the consideration of this case.

The solution is as expected. The upper plate drags a horizontal film of fluid with its shear stress. The 'next' layer is dragged along by the accelerating fluid, and so on until a linear stable profile is reached, as described in 8.3.2. During $t \in [0, 0.06]$, errors of $O(10^{-4})$ are present, represented by the black part. These errors are negligible.

With approximating the incompressible case with marginal compressibility, as mentioned before in 7.3, some issues occur. The calculations for the CEL method are thus far done in Abaqus/Explicit. Explicit calculations imply the notion of stable time step size $h_{t,max}$. In this time, a wave has to be 'caught' in say $n \geq 1$ parts by the used elements. This notion is often referred to as the Courant Friedrichs Lewy (CFL) criterion:

$$h_{t,max} = \frac{L_{element}}{c} = \frac{L_{element}}{nc}. \quad (8.19)$$

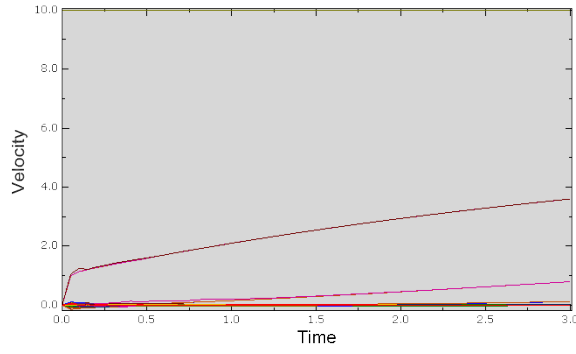
It proves that the stable time step size is linearly proportional to the element size. Noting that for fluids

$$c = \sqrt{\frac{K}{\rho}} \quad (8.20)$$

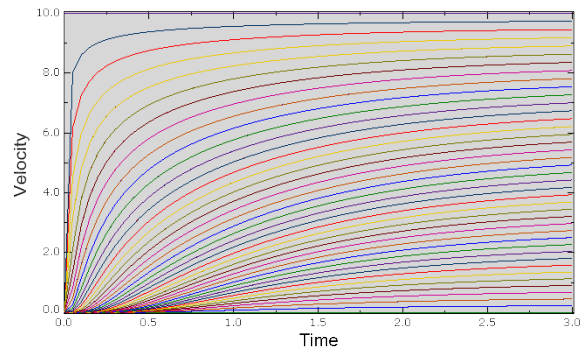
holds, when the speed of sound c is increased for a higher bulk modulus K and thus a better approximation of incompressibility, the stable time step size decreases, making the analysis take more time. Therefor, a proper balance between c and calculation time should be chosen. Calculation time must be within acceptable bounds, but not in a way that it influences the compressibility and thus the solution too much, making it unrealistic.

The calculations times of Abaqus are disappointing. Running on no less than eight processors with each eight gigabytes of memory to their disposal, the analysis still takes a few hours. For comparison with the Matlab model, the run time is taken for one processor instead of eight. Although the compressible equations are solved, which implies a larger possible time step size compared to the incompressible case, the time step size is still $O(10^{-6})$. That h_t is independent of the dynamic viscosity μ is striking. This can be explained by the fact that explicit calculations are always dominated by a choice of time step size based on stability and not on accuracy.

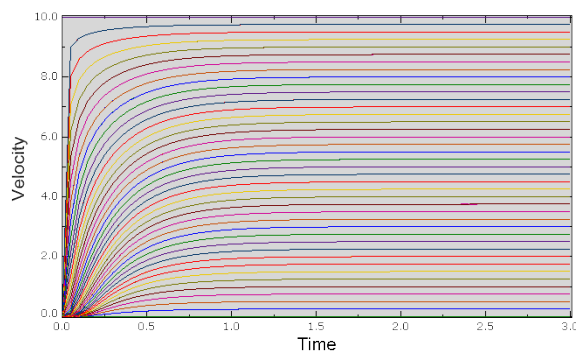
*The calculation times of Abaqus stem from the log output file of the job concerned. In these files, the end part contains output requested by the GTC*L script, not by Simulia. This includes the calculation time. The latter often appears to be faulty, at one time even more than the other. Perhaps parameters like amount of memory and variation of the CPU frequency were not taken into account properly. We emphasise that the GTC*L script should be considered when drawing conclusions with regard to the calculation times. In the last stage of the project it was brought to our attention that Abaqus/Explicit can give calculation times when requested. It is stated that herein, the amount of memory and variation of the CPU frequency are taken into account. This option is an alternative for people outside of GTC*L, who do not have the mentioned script at their disposal.*



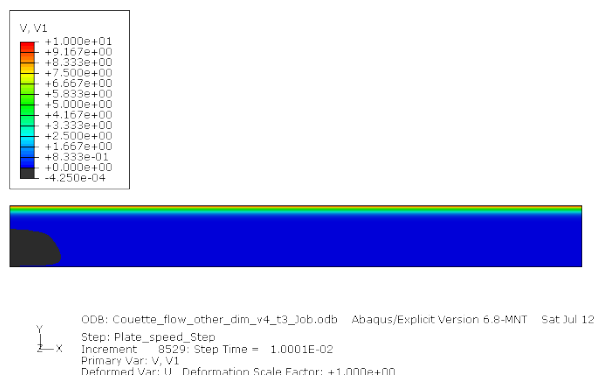
(a) $\mu = 0.001[Pa \cdot s], t \in [0, 3][s]$. The different lines represent the V_1 field for nodes along the y axis.



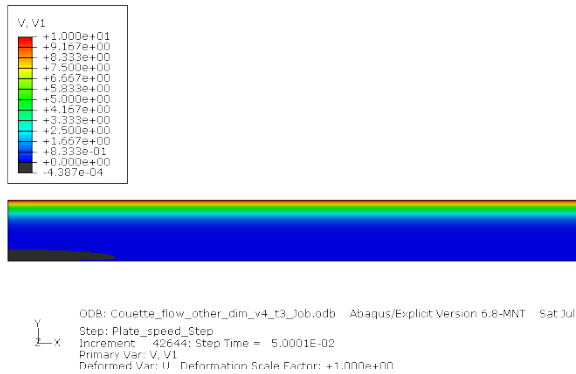
(b) $\mu = 1[Pa \cdot s], t \in [0, 10][s]$. The different lines represent the V_1 field for nodes along the y axis.



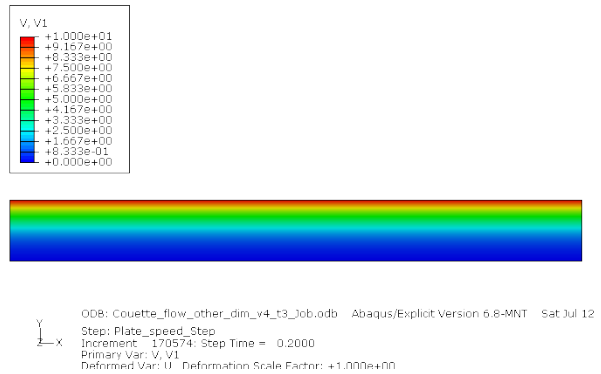
(c) $\mu = 4[Pa \cdot s], t \in [0, 10][s]$. The different lines represent the V_1 field for nodes along the y axis.



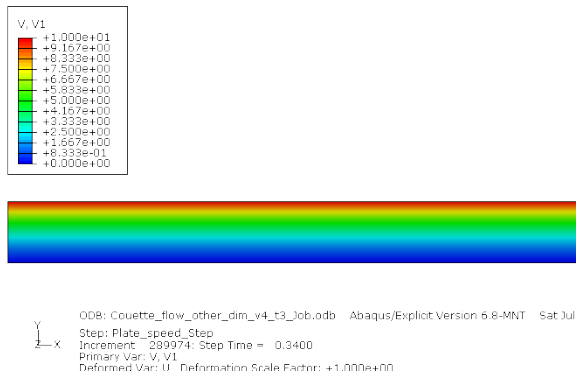
(d) $\mu = 4[Pa \cdot s], t = 1.0001 \cdot 10^{-2}[s]$.



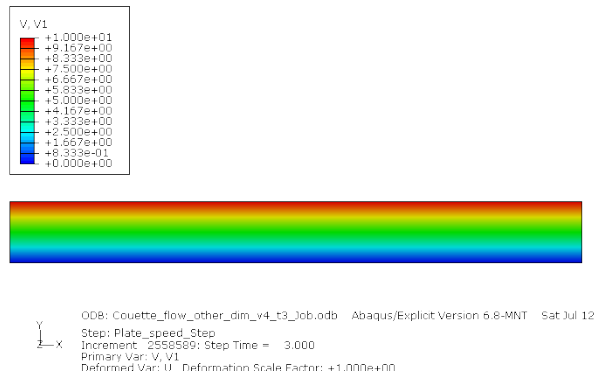
(e) $\mu = 4[Pa \cdot s], t = 5.0001 \cdot 10^{-2}[s]$.



(f) $\mu = 4[Pa \cdot s], t = 0.2[s]$.



(g) $\mu = 4[Pa \cdot s], t = 0.34[s]$.



(h) $\mu = 4[Pa \cdot s], t = 3[s]$.

Figure 8.7: $V_1[\frac{m}{s}]$ at different times.

#DOF	$h_t[s]$	$u_{up}[\frac{m}{s}]$	$\mu[Pa \cdot s]$	$t_{total}[s]$	$t_{CPU}[s]$
24846	$1.172 \cdot 10^{-6}$	10	4	3	154776
24846	$1.173 \cdot 10^{-6}$	10	1	3	183049
24846	$1.170 \cdot 10^{-6}$	10	0.001	3	82312

Table 8.2: results Couette flow model in Abaqus/Explicit.

8.6 Comparison

The Couette flow simulation runs properly with both Matlab and Abaqus. The Matlab model solves the incompressible Navier-Stokes equations implicitly with the finite difference method (FDM) and Abaqus solves similar, but compressible equations explicitly by the use of FEM. Recalling the relation between the bulk modulus and the critical time step size from 8.5.2, the latter should imply a shorter critical time step size. This is not the case. Abaqus and Matlab use $h_{t,\max} = 1.17 \cdot 10^{-6}[s]$ and $h_{t,\max} = 5 \cdot 10^{-4}[s]$ respectively. This is partially caused by the difference between semiimplicit and explicit calculations and partially by relation (8.19).

For comparison of the solutions, one must consider Figures 8.5b and 8.7h. In the latter, the velocity is only indicated by colour, instead of by colour and height as in the former. Nevertheless, it is clear that they correspond in their linear velocity profile.

To compare the analysis times of Abaqus and Matlab would be inadequate because the analysis are run on different amounts of CPUs with different cores. These CPUs differ in memory size and clock frequency. Also Abaqus is subjected to domain decomposition to run on clusters of CPUs whereas Matlab is run on a CPU with two cores. Equalising the computation environment for both programs is impossible at this time.

Chapter 9

Channel flow interacting with elastic body

9.1 Introduction

In the article [36], a FSI benchmark is proposed. The considered problem is two dimensional. Programming in Matlab gets considerably more difficult. Therefore, we start with a simplified case. Afterwards, more elements are added to the problem in both Matlab and Abaqus, increasing the difficulty, ending up at the full benchmark problem. In Matlab, we treat the channel flow without an object and with a cube. Adding more structural elements leads to models whereof manual programming is cumbersome. In Abaqus, this is not an issue.

The Matlab analysis were run with an AMD Athlon single core 2.2 GHz CPU and the Abaqus analysis were divided over the clusters of AMD Opteron single core 2.4 GHz, dual core 2.2 GHz and dual core 2.8 GHz. Of the latter the dual core 2.2 GHz is in the majority.

Additional output is available in movie form, to be found at [2].

9.2 Model

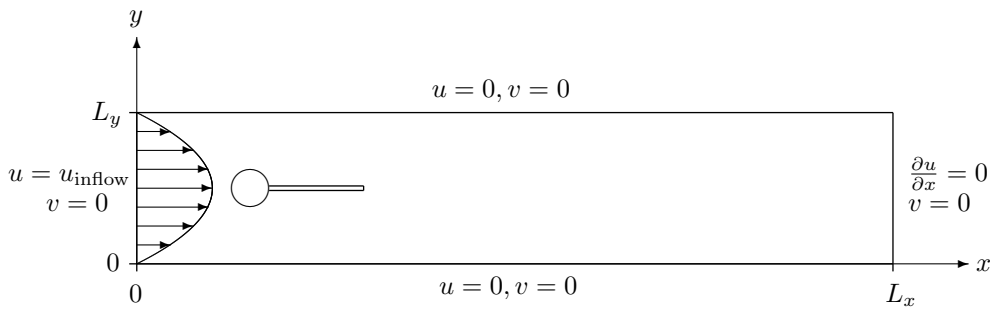


Figure 9.1: channel flow interacting with elastic body [36].

The benchmark proposed involves incompressible, Newtonian viscous channel flow around an elastic body on a rectangular domain, cf. Figure 9.1. Thus the fluid equations resemble the case of Couette flow, treated in chapter 8. The IC and BCs are different however, making the solution drastically different.

The fluid and structure are interacting. Because the body is elastic, the influence of the fluid flow on the structure is not negligible making this a typical FSI problem. However, because of the difficulty level of this problem, we start with the consideration of the model without the body.

9.2.1 Fluid properties

The problem is solved by considering the two dimensional version of the Navier-Stokes equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} &= 0 \\ \rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) &= \rho \mathbf{f} - \nabla p + \nabla \sigma_{\text{dev}}. \end{aligned} \quad (9.1)$$

Summarising the assumptions

- incompressible flow $\Leftrightarrow \rho = \text{constant}$
- Newtonian shear cf. 3.6.1
- no body forces $\Leftrightarrow \mathbf{f} = \mathbf{0}$,

by which (9.1) is simplified:

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0 \\ \rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) &= -\nabla p + \mu \nabla^2 \mathbf{v}. \end{aligned} \quad (9.2)$$

The independent variables are x, y and t . The dependent variables of interest are pressure $p = p(x, y, t)$ and velocity $\mathbf{v} = \mathbf{v}(x, y, t)$.

9.2.2 Structure properties

The structure is assumed to be elastic and compressible. Its configuration is described by the displacement \mathbf{d} , with velocity $\mathbf{v}_s = \frac{\partial \mathbf{d}}{\partial t}$. The balance equations are

$$\rho_s \left(\frac{\partial \mathbf{v}_s}{\partial t} + (\mathbf{v}_s \cdot \nabla) \mathbf{v}_s \right) = \nabla \cdot \boldsymbol{\sigma}_s.$$

Written in the more common for structures Lagrangian way we have

$$\rho_s \frac{\partial^2 \mathbf{d}}{\partial t^2} = \nabla \cdot (J \boldsymbol{\sigma}_s \mathbf{F}^{-T}) \quad (9.3)$$

where $\mathbf{F} = \mathbf{I} + \nabla \mathbf{d}$ is the deformation gradient tensor. The material is specified by the constitutive Cauchy stress tensor.

9.2.3 Initial and boundary conditions

At the walls of the domain we have no slip conditions, so $\{u = v = 0 | y \in \{0, H\}\}$. At the inflow we define a parabolic velocity profile:

$$u(0, y, t) = \begin{cases} u_{\text{inflow}} \frac{1 - \cos(\frac{\pi}{2}t)}{2} & t < 2 \\ u_{\text{inflow}} & \text{otherwise} \end{cases} \wedge u_{\text{inflow}} = 1.5\bar{u} \frac{y(H-y)}{(\frac{H}{2})^2} \quad (9.4)$$

which clearly has mean and maximum inflow \bar{u} and $1.5\bar{u}$ respectively, roots at $y \in \{0, H\}$ and the maximum at $y = \frac{H}{2}$. The inflow condition (9.4) represents, besides a BC, also a part of the IC. The IC is completed by taking the union of the latter and $\{u = v = 0 | x \neq 0, t = 0\}$.

We use $v_t = 0 \wedge \sigma_{nn} = 0$ for the outflow. Since $\mathbf{n} = (1 \ 0)^T \wedge \mathbf{t} = (0 \ 1)^T$ at the outflow we get $v = 0 \wedge \sigma_{xx} = 0$. Notice that $v(L, y, t) = 0 \Rightarrow \frac{\partial v}{\partial y}(L, y, t) = 0$ which with $\nabla \cdot \mathbf{v} = 0$ gives $\frac{\partial u}{\partial x}(L, y, t) = 0$. Herewith, BCs for both u and v are known at the outflow.

9.3 Matlab solution

In order to obtain a solution of (9.1) in combination with the conditions mentioned in 9.2.3 with Matlab, the equations have to be discretised. The discretisation goes analogous to that of Couette flow, treated in 8.4. The only difference is in the BCs. The implications for the pressure therefor differ and are treated.

9.3.1 Implications of the initial and boundary conditions

Before continuing with the momentum equations, we consider the ICs and BCs mentioned in 9.2.3 to get information about pressure p at the boundaries. Depending on the strategy to solve (9.2), this information is required or not for solving the momentum equations. If it is not, this information is processed implicitly by the BCs for the velocity \mathbf{v} .

To start simple, consider the homogeneous Dirichlet no slip BCs $\{u = v = 0 | x \in [0, L], y \in \{0, H\}, t \in [t_0, \infty)\}$. Because x and t are not fixed, the BCs are independent of x and t . Consequently, the continuity equation implies

$$\begin{aligned} [\nabla \cdot \mathbf{v}]_{y \in \{0, H\}} &= 0 \\ \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right]_{y \in \{0, H\}} &= 0 \\ \frac{\partial v}{\partial y} \Big|_{y \in \{0, H\}} &= 0 \end{aligned}$$

and the momentum equations

$$\begin{aligned} \left[\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) \right]_{y \in \{0, H\}} &= \left[-\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right]_{y \in \{0, H\}} \\ 0 &= -\frac{\partial p}{\partial x} \Big|_{y \in \{0, H\}} + \mu \frac{\partial^2 u}{\partial y^2} \Big|_{y \in \{0, H\}} \end{aligned}$$

and

$$\begin{aligned} \left[\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) \right]_{y \in \{0, H\}} &= \left[-\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \right]_{y \in \{0, H\}} \\ 0 &= -\frac{\partial p}{\partial y} \Big|_{y \in \{0, H\}} + \mu \frac{\partial^2 v}{\partial y^2} \Big|_{y \in \{0, H\}}. \end{aligned}$$

The ICs and BCs at $x = 0$ and $x = L$ also have implications for the pressure p . Consider $\{u = u(0, y, t), v = 0 | x = 0, y \in [0, H], t \in [t_0, \infty)\}$, with $u(0, y, t)$ from (9.4).

$$\begin{aligned} 0 &= -\frac{\partial p}{\partial x} \Big|_{x=0} + \mu \frac{\partial^2 u}{\partial y^2} \Big|_{x=0} \\ 0 &= -\frac{\partial p}{\partial x} \Big|_{x=0} - \frac{12\mu\bar{u}}{H^2} \\ &\wedge \\ \rho u \Big|_{x=0} \frac{\partial v}{\partial x} \Big|_{x=0} &= -\frac{\partial p}{\partial y} \Big|_{x=0} + \mu \frac{\partial^2 v}{\partial x^2} \Big|_{x=0} \\ \frac{6\rho\bar{u}}{H^2} y(H-y) \frac{\partial v}{\partial x} \Big|_{x=0} &= -\frac{\partial p}{\partial y} \Big|_{x=0} + \mu \frac{\partial^2 v}{\partial x^2} \Big|_{x=0} \end{aligned}$$

Consider $\{u, v = 0 | \sigma_{nn} = 0, x = L, y \in [0, H], t \in [t_0, \infty)\}$. Recall that when Newtonian shear cf. 3.6.1 and incompressibility is assumed, (8.1) holds:

$$\sigma_{nn} = -p + \frac{2}{Re} \frac{\partial v_n}{\partial n}.$$

So in our case

$$\sigma_{xx} = -p + \frac{2}{Re} \frac{\partial u}{\partial x}.$$

With $\sigma_{xx} = 0$ and recalling from 9.2.3 that $\frac{\partial u}{\partial x} = 0$ it follows for the dimensionless form that

$$p = \frac{2}{Re} \frac{\partial u}{\partial x} = 0$$

which implies $\frac{\partial p}{\partial y} = 0$. From the momentum equation in x direction we get for $\frac{\partial p}{\partial x}$

$$\begin{aligned} \left[\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) \right]_{x=L} &= \left[-\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right]_{x=L} \\ \rho \frac{\partial u}{\partial t} \Big|_{x=L} &= -\frac{\partial p}{\partial x} \Big|_{x=L} + \mu \left(\frac{\partial^2 u}{\partial x^2} \Big|_{x=L} + \frac{\partial^2 u}{\partial y^2} \Big|_{x=L} \right). \end{aligned}$$

Summarised we have for the pressure

- $y \in \{0, H\} : \frac{\partial v}{\partial y} = 0, \nabla p = \mu \frac{\partial^2 v}{\partial y^2}$
- $x = 0 : \frac{\partial p}{\partial x} = \frac{12\mu\bar{u}}{H^2}, \frac{\partial p}{\partial y} = \mu \frac{\partial^2 v}{\partial x^2} - \frac{6\rho\bar{u}}{H^2} y(H-y) \frac{\partial v}{\partial x}$
- $x = L : p = 0, \frac{\partial p}{\partial x} = \mu \nabla^2 u - \rho \frac{\partial u}{\partial t}, \frac{\partial p}{\partial y} = 0.$

9.3.2 Solution channel flow

The system is solved on the domain $(x, y, t) \in [0, 0.4] \times [0, 0.4] \times [0, 2.5]$ where both the x and y direction are partitioned in 52 parts of the same length. The resulting number of DOF is 8008. The density is kept constant on $998 \frac{kg}{m^3}$.

The solution is depicted in Figure 9.2. In the unadjusted case, the pressure has unrealistic values. This

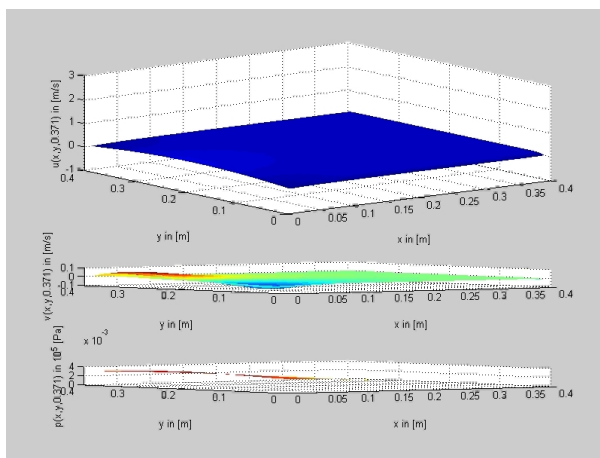
is a result of the referencelessness of p , as mentioned in 3.4. Moreover, it is confirmed by the existence of a nullspace of \mathbf{A} . Say we choose a vector \mathbf{y} whereof the u and v unknowns are taken zero. Operator \mathbf{A}_c does not contain nonzero columns for the pressure unknowns so $\mathbf{A}_c \mathbf{y} = \mathbf{0}$. Neither does \mathbf{I}_m , thus only the symmetric pressure part of \mathbf{A}_m remains. The symmetry implies that $p = \text{constant}$, so that $(0 \ 0 \dots 0 \ p_0 \ p_0 \dots p_0)^T \in \text{nullspace}(\mathbf{A})$ with $p_0 \in \mathbb{R}$ arbitrary. To resolve this referencelessness, we set $\min(p) = 0$. Herewith, the solution is as expected. The pressure variation is linear along the channel. The velocity inflow profile builds up while $t \in [0, 2)$ and afterwards remains constant. There are no compression/expansion waves present, since we treated the incompressible Navier-Stokes equations. A parabolic velocity profile is obtained as expected.

When \bar{u} is increased, $\text{image}(\frac{\partial u}{\partial t}(0, y, t))$ increases, which implies a smaller time step size h_t . Quantitative values of this can be found in Table 9.1. The effect of decreasing dynamic viscosity μ from the unrealistic $1[Pa \cdot s]$ to the realistic $0.001[Pa \cdot s]$ for water also has implications for the time step size. Recalling that we assumed Newtonian shear stress, μ can be seen as a measure for how much a change in local energy of the fluid can be transferred to its surroundings. After all, the shear stresses are proportional to μ . If the viscosity is low, this ability is weak. Thus, local changes in time and space stay more local than with a higher viscosity. The need for a finer grid and smaller time step is the result. Quantitative values hereof follow from Table 9.1.

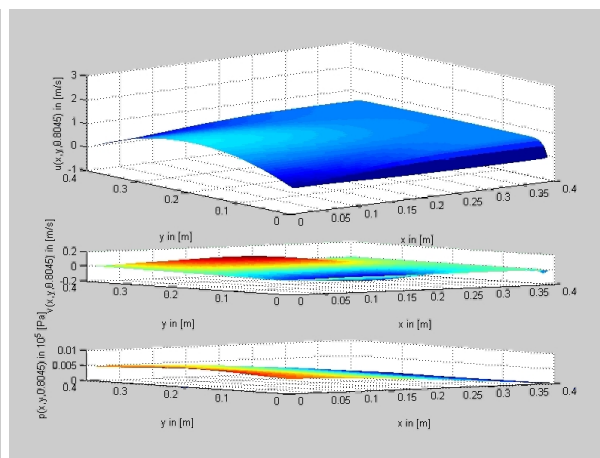
The relation between the analysis time and Reynolds number has a linear tendency. The relation between the critical time step size and the Reynolds number has a hyperbolic tendency. When the Reynolds number increases, the influence of the nonlinear convection terms increases. Since these terms are comprised in the explicit part of (8.18) and thus at the previous known time, they affect the critical time step size. In effect, when the Reynolds number increases, the critical time step size decreases.

Re	$h_t[s]$	$\bar{u}[\frac{m}{s}]$	$\mu[Pa \cdot s]$	$t_{CPU}[s]$
10	0.1	2	19.960	71
20	0.00625	2	9.980	1259
25	0.1	0.25	1.000	86
40	0.003125	2	4.990	2379
50	0.000195	10	19.960	22070
50	0.001563	2	3.992	2764
50	0.013	0.5	1.000	646
80	0.001	2	2.495	4366
200	0.000781	2	0.998	5542
500	0.00025	2	0.399	25244
750	0.0002	2	0.266	21508
1000	0.000025	10	0.998	75600
1000	0.0001	2	0.200	42803

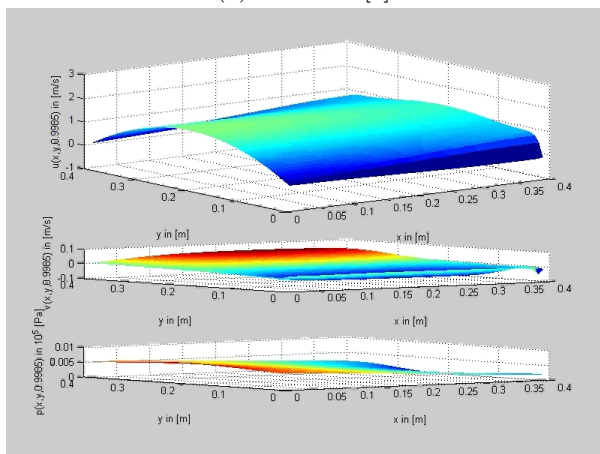
Table 9.1: results channel flow model in Matlab.



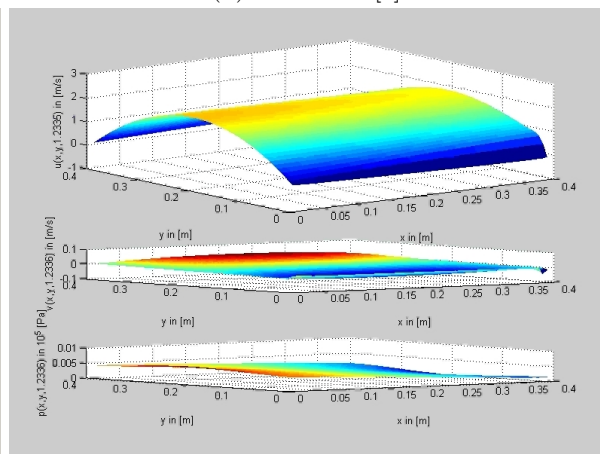
(a) $t = 0.371$ [s].



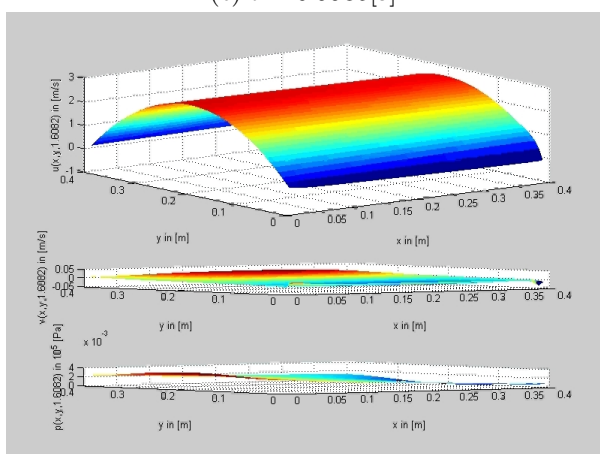
(b) $t = 0.8045$ [s].



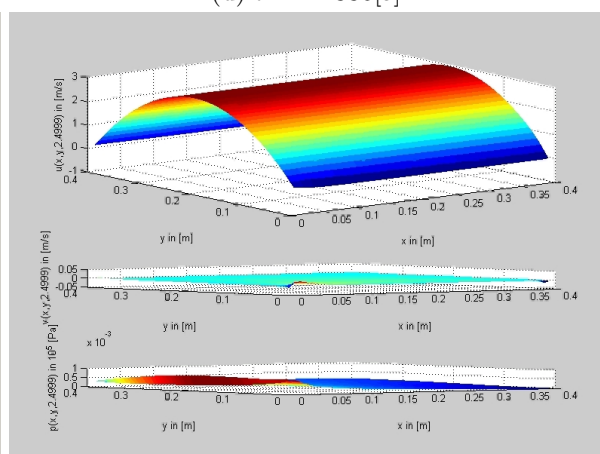
(c) $t = 0.9985$ [s].



(d) $t = 1.2336$ [s].



(e) $t = 1.6082$ [s].



(f) $t = 2.5$ [s].

Figure 9.2: channel flow, a parabolic velocity profile.

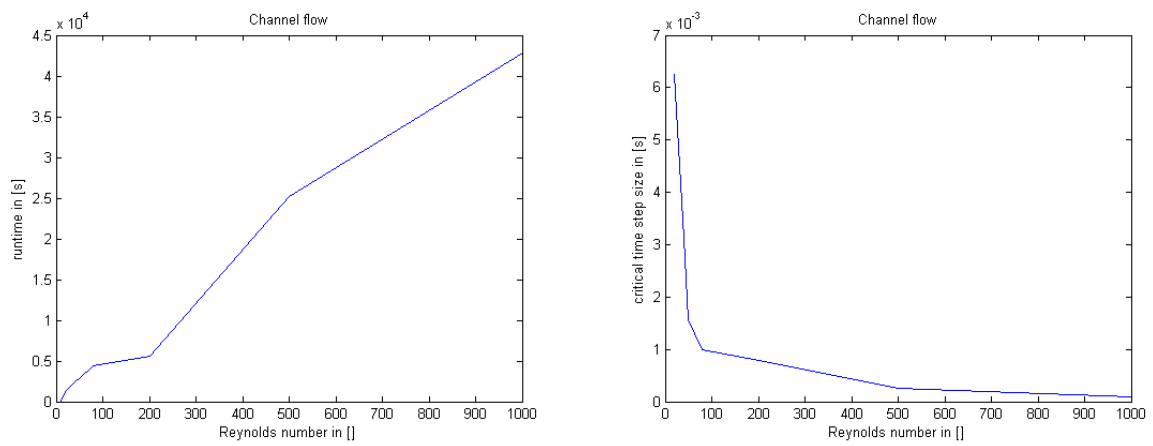


Figure 9.3: analysis time and critical time step size channel flow.

9.3.3 Solution channel flow with cube

The first element that is added to the channel flow model to increase the difficulty is a cube. It will be placed in the middle of the channel, forcing the fluid to go around it.

As shown in 8.4, the boundary elements need to be distinguished from the rest of the elements. Adding a cube in the channel, even more distinguishments need to be made which would result in an even more complex program. To circumvent this, we 'add' the cube such that its right edge is on the inflow boundary, cf. Figure 9.4. In this way, only the velocity inflow profile has to be adjusted in the program. Note that this adjustment is, however, somewhat simplifying the model. In practice, the velocity profile does not make a sudden jump along the sides of the cube. It would be more bended due to no slip conditions on the cube, cf. Figure 9.5.

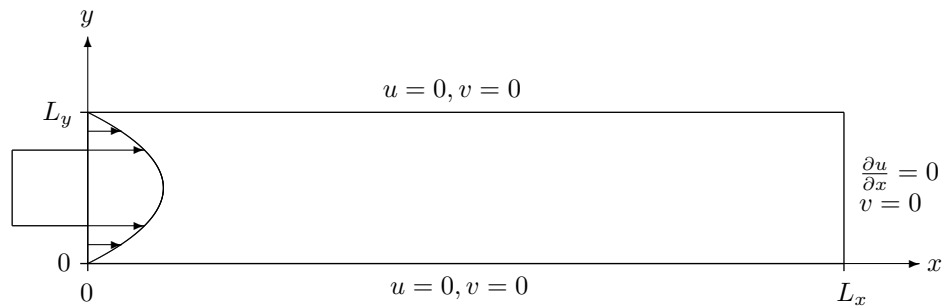


Figure 9.4: channel flow with cube.

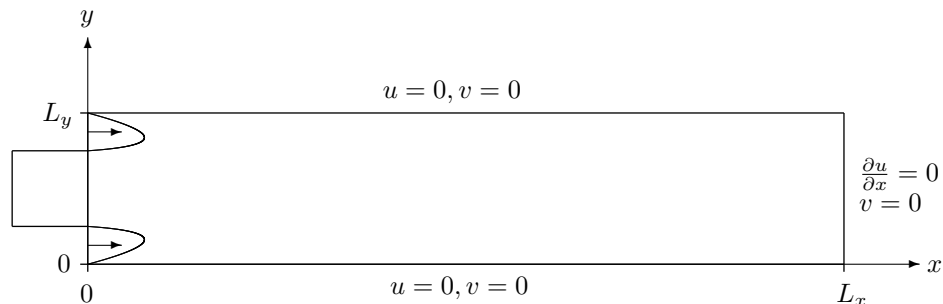


Figure 9.5: realistic velocity profile.

The solution is depicted in Figure 9.7. It is as expected. For certain Reynolds numbers it contains a Kármán vortex street, named after the engineer and fluid dynamicist Theodore von Kármán. It is a repeating pattern of swirling vortices caused by the unsteady separation of flow over bluff bodies, cf. Figure 9.6. A Kármán vortex street appears for $47 < Re < 10^7$, where the inflow velocity and diameter of the cube or cylinder are chosen as the characteristic velocity and distance respectively. There is an empirical formula known for the frequency f of it:

$$\frac{fd}{\bar{u}} = 0.198 \left(1 - \frac{19.7}{Re} \right) \quad (9.5)$$

where d is the diameter of the object in the flow. Equation (9.5) is valid for $250 < Re < 10^5$ [39]. For $d = 0.1[m]$, $\bar{u} = 2[\frac{m}{s}]$, $Re = 750$, (9.5) gives $f = 3.9[\frac{1}{s}]$. When compared to the frequency of the corresponding output of Matlab [2], the frequency does not correspond. The frequency in the Matlab output is about two times higher. Perhaps this is due to (9.5) being valid for a cylinder instead of a cube. There is a lack in further arguments for this difference.

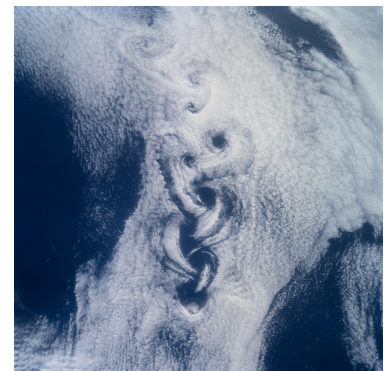


Figure 9.6: Kármán vortex street off the coast of Rishiri Island in Japan.

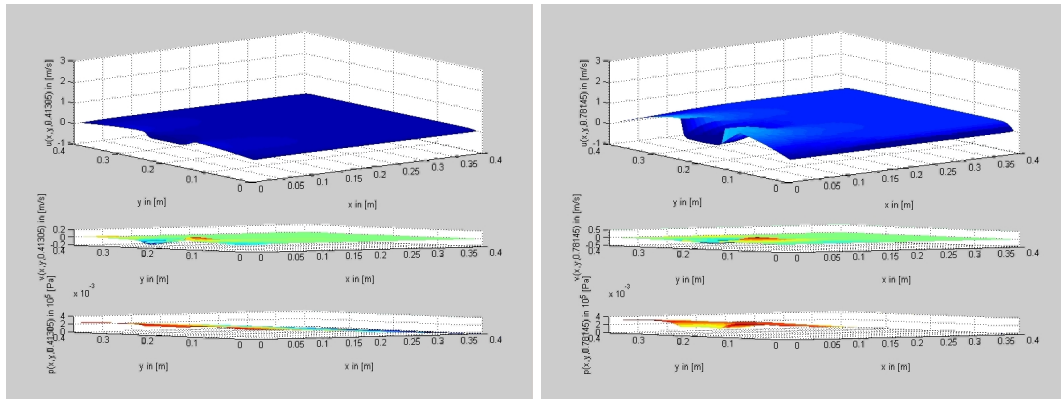
The horizontal velocity is under such influence of the vortex street, that it even becomes negative just 'behind' the cube. This is also the case for the vertical velocity, which is in this case certainly not negligible.

In contrast with the channel flow without the cube, the vertical velocity is of the same order of magnitude as the horizontal velocity. Also the pressure is not linear anymore.

As can be seen in Figure 9.8, the analysis time for channel flow with the cube is longer than without the cube. This is only due to the smaller critical time step size. The time per calculation step is still the same, namely 1.72[s]. Again, higher Reynolds numbers increase the influence of the explicitly taken nonlinear convection terms, decreasing the critical time step size. Measurements are denoted in Table 9.2.

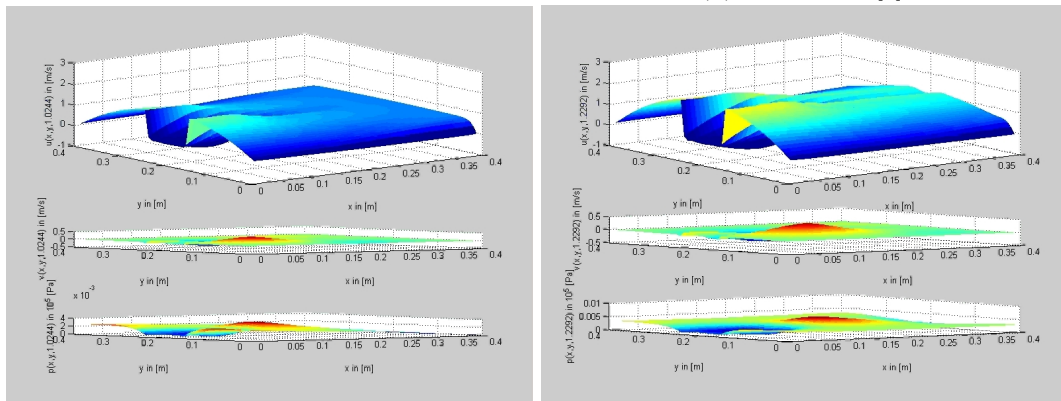
$h_t[s]$	$\bar{u}[\frac{m}{s}]$	$\mu[Pa \cdot s]$	Re	$t_{CPU}[s]$
0.010000	2	19.960	10	471
0.005000	2	3.992	50	925
0.000313	2	1.996	100	13932
0.000200	2	0.798	250	21770
0.000100	2	0.399	500	43768
0.000050	2	0.266	750	87193

Table 9.2: results channel flow with cube model in Matlab.



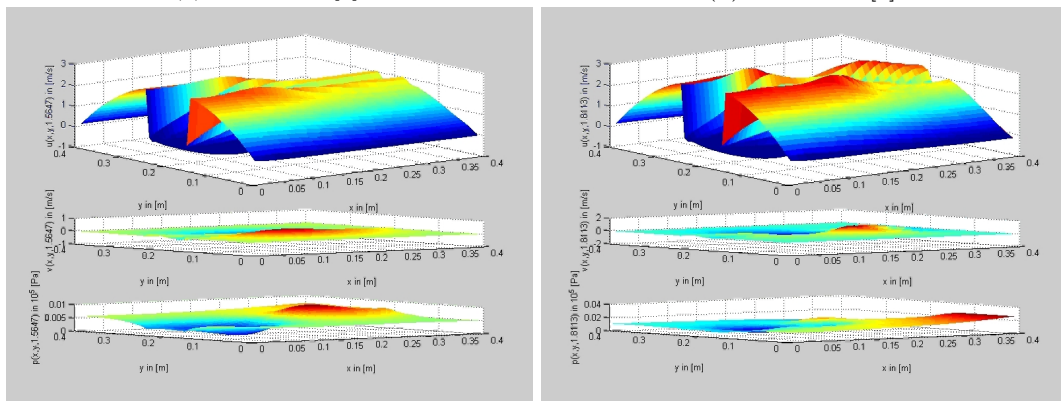
(a) $t = 0.41305[s]$.

(b) $t = 0.78145[s]$.



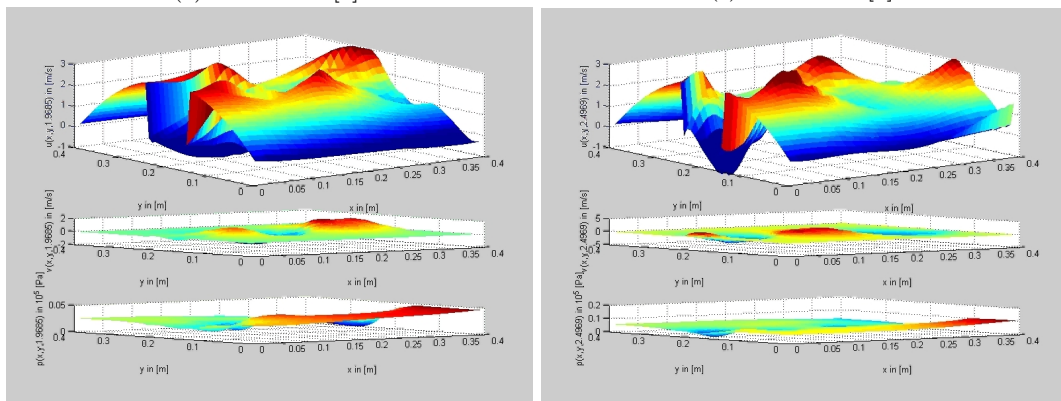
(c) $t = 1.0244[s]$.

(d) $t = 1.2292[s]$.



(e) $t = 1.5647[s]$.

(f) $t = 1.8113[s]$.



(g) $t = 1.9685[s]$.

(h) $t = 2.5[s]$.

Figure 9.7: channel flow with cube.

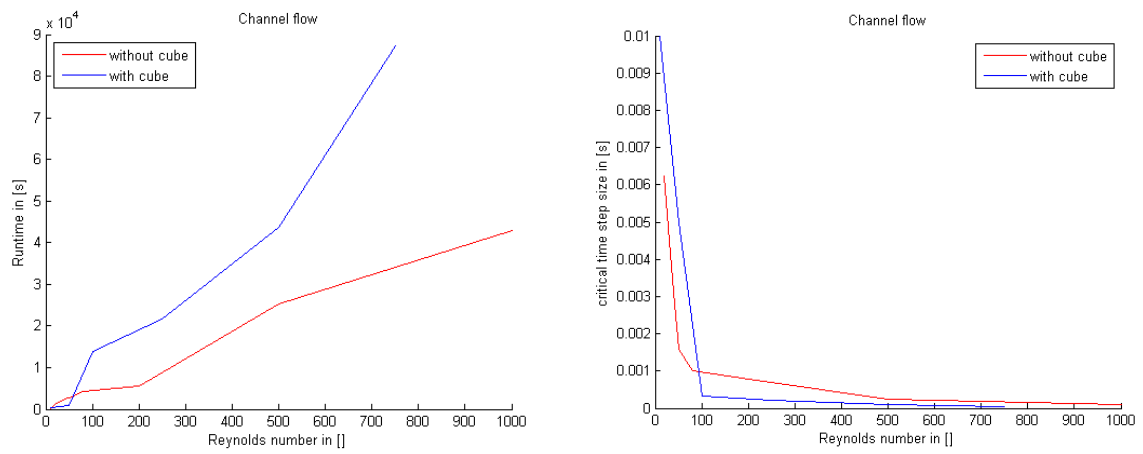


Figure 9.8: analysis time and critical time step size, for both with and without cube.

9.4 Abaqus solution

In this section the modelling of the channel flow with Abaqus is treated. The usage of Abaqus/CAE for CAE model construction is shortly explained. Conform the scope of this report, the emphasise is typically on the solution that Abaqus/Explicit produces and the interpretation of it.

We start with the simplified case, channel flow, and afterwards add the cylinder and finally the flag, which completes the benchmark as described in [36]. The results of Abaqus are compared with those given in [36].

For a short introduction to the usage Abaqus, in particular the modelling of problems involving both Eulerian and Lagrangian grids, we refer to appendices A.2 and A.3. Herein, terminology is also explained.

9.4.1 Formulation channel flow

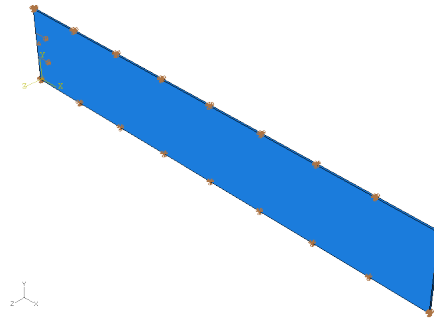


Figure 9.9: fluid domain with BC symbols.

An Eulerian part is defined for the fluid domain. As fluid material, water is defined by the three parameters density, speed of sound and kinematic viscosity. Via section assignment the material is assigned to the Eulerian part. The Eulerian part is instanced in the assembly.

Next, a time step is created where we can add BCs and predefined fields for the fluid domain. The BCs are in accordance with 9.2.3. The inflow profile is realised by creating an analytical field resembling (9.4). At the outflow the $\sigma_{nn} = \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{n} = 0$ BC is not prescribed since Abaqus assumes the Neumann BC $\boldsymbol{\sigma} \cdot \mathbf{n} = 0 \Rightarrow \sigma_{nn} = 0$ automatically when no components for the n direction are prescribed [34]. With the feature predefined field the domain is fully filled with fluid.

The output we are interested in are the velocities in x and y directions and the density. Although the latter is a known constant in the Matlab model and thus not interesting for output, it is in the case of Abaqus because it approximates the incompressible case with a high, but not infinite, bulk modulus K . This high bulk modulus K is realised by taking the speed of sound c sufficiently large. Because

$$c = \sqrt{\frac{K}{\rho}},$$

it follows $c \rightarrow \infty \Leftrightarrow K \rightarrow \infty$.

9.4.2 Solution channel flow

The results of Abaqus are depicted in Figures 9.10, 9.11, 9.12 and 9.14. Data on the computation time and critical time step size is denoted in Table 9.3. In 9.10a the velocities of several nodes are plotted against time.

At initiation, the water is in rest. The inflow velocity increases, in accordance with (9.4). The fluid is 'pushed' to the right, increasing in velocity. Oscillations occur. When an incompressible medium is considered, these oscillations can not stem from compression effects. Abaqus, however, allows slight compressibility, which is the source of these oscillations. The oscillations are not the result of 'noise' of explicit calculations, because they are too pure for noise as a result of explicit calculations. What is striking of the physical oscillations is that the frequencies are the same for $c_0 \in \{1000, 1483\}[\frac{m}{s}]$. Only the amplitudes differ. Perhaps this is a result of the Mie-Grüneisen EOS. Herefor, we lack further arguments at this time.

With the increase of velocity the fluid is longitudinally compressed as a result of mass inertia. The compression builds up until it reaches a certain maximum density. During this time the fluid is still being pushed to the right by the inflow source. After the compression has reached its maximum, the fluid will start expanding, giving the fluid an acceleration and thus an additional velocity. This explains the grey part in 9.10e which represents the fluid with velocity surpassing the maximum inflow velocity $1.5\bar{u} = 1.5 \cdot 10 = 15[\frac{m}{s}]$. The density oscillates $\in [997, 1000][\frac{kg}{m^3}]$. The expansion does not last forever and also has its maximum. Subsequently a deceleration occurs, causing the fluid velocity to go below $1.5\bar{u}[\frac{m}{s}]$. The no slip conditions oppose these oscillations. They delay velocity changes. As a result, a velocity profile as depicted in Figure 9.13 is obtained. The damped oscillation repeats itself several times until it reaches the density corresponding to the ruling pressure. The amount of covered fuses depends on bulk modulus K and the inflow velocity \bar{u} . As Figures 9.10a and 9.11a show, for $c_0 = 1483[\frac{m}{s}]$, $\bar{u} = 10[\frac{m}{s}]$ it takes approximately six seconds to stabilise. After stabilisation, $\rho \in [997.6, 999.5][\frac{kg}{m^3}]$. This slight difference in density is the result of a difference in pressure throughout the fluid. The difference in pressure is at its turn the result of a difference in velocity.

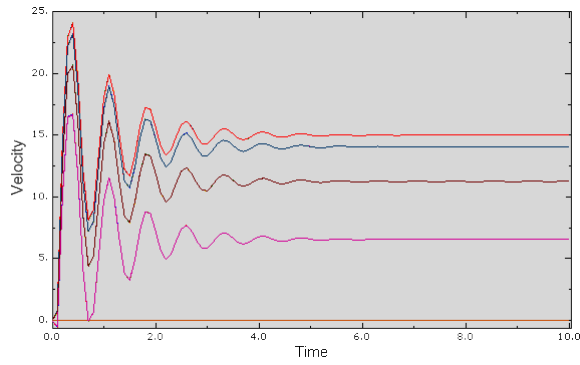
With approximating the incompressible case with marginal compressibility, some issues occur. The calculations for the CEL method are thus far done in Abaqus/Explicit. Explicit calculations imply the notion of stable time step size $h_{t,max}$. In this time, a wave has to be 'caught' in say $n \geq 1$ parts by the used elements. Thus, when speed of sound c_0 is increased for a higher bulk modulus K and thus a better approximation of incompressibility, the stable time step size decreases, making the analysis take more time. Therefore, a proper balance between c_0 and calculation time should be chosen. Calculation time must be within acceptable bounds, but not in a way that it influences the compressibility and thus the solution too much, making it unrealistic. In accordance, we compare the $c_0 = 1483[\frac{m}{s}]$ case with the artificial $c_0 = 1000[\frac{m}{s}]$. The results are depicted in 9.12 and 9.14.

Taking $c_0 = 1000[\frac{m}{s}]$ instead of $c_0 = 1483[\frac{m}{s}]$ clearly influences the solution too much, rendering it unrealistic. At initiation, the expansion of the fluid results in absolute negative speeds, cf. the black parts in Figures 9.12b, c and f. Also the surpassing of $1.5\bar{u}$ is considerably greater, cf. the grey part in 9.12e. The time to stabilise is approximately thirty seconds. The density oscillates $\in [987, 1009][\frac{kg}{m^3}]$.

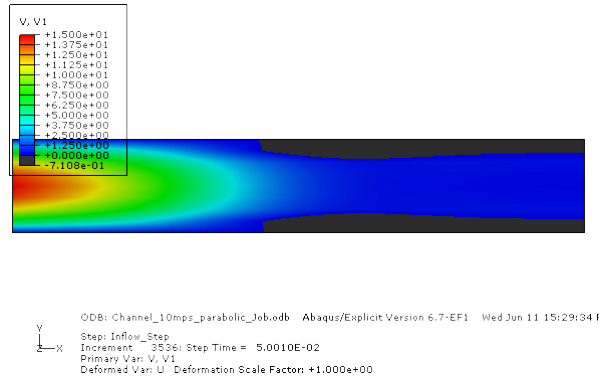
The difference in computation times, cf. Table 9.3, is the result of the differences in h_t , \bar{u} , c_0 and $\#DOF$. The h_t differs a factor 2.2 and \bar{u} a factor 2. The computation time differs a factor 8.7. The influence of \bar{u} and $\#DOF$ is big, because even though the speed of sound is relaxed to $1000[\frac{m}{s}]$, the computation time is 8.7 times larger.

$\#DOF$	$\bar{u}[\frac{m}{s}]$	$c_0[\frac{m}{s}]$	$\mu[Pa \cdot s]$	Re	$h_t[s]$	$t_{CPU}[s]/[h]$
5454	5	1483	0.001	$5 \cdot 10^5$	$1.417 \cdot 10^{-5}$	20399.97/5.7
25452	10	1000	0.001	10^6	$6.407 \cdot 10^{-6}$	176784.02/49.1

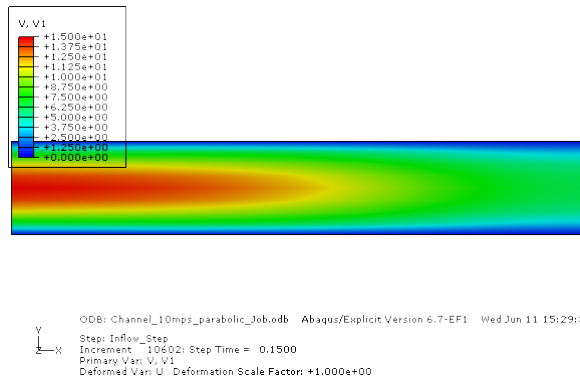
Table 9.3: results channel flow model.



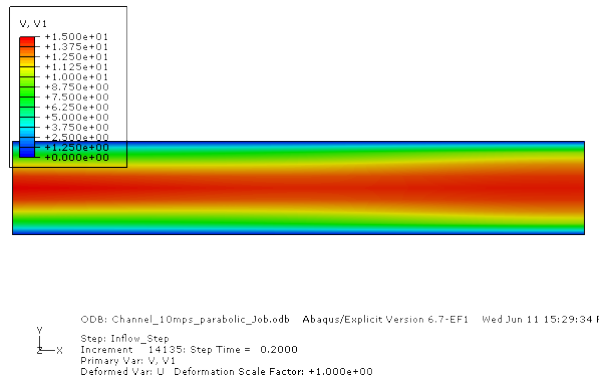
(a) $t \in [0, 10][s]$. The different lines represent the V_1 field for nodes along the y axis.



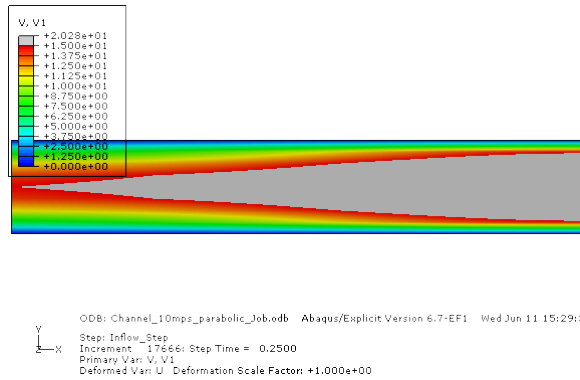
(b) $t = 5.001 \cdot 10^{-2}[s]$.



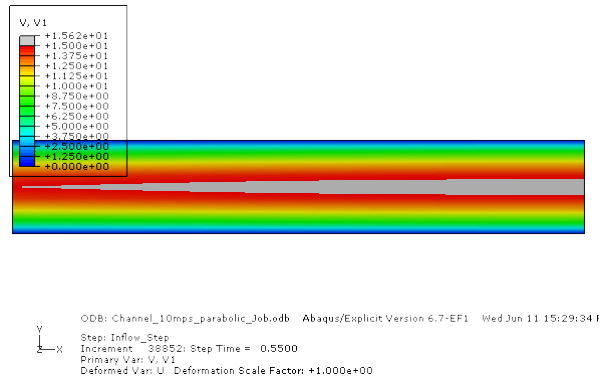
(c) $t = 0.15[s]$.



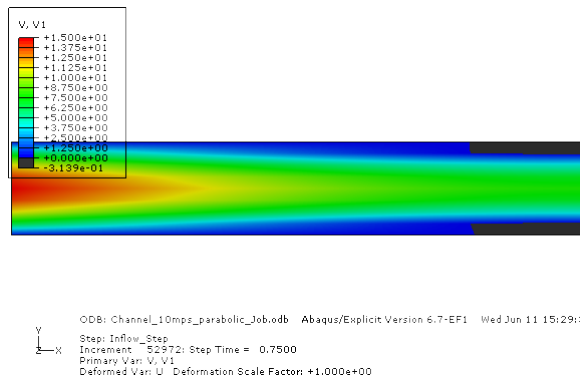
(d) $t = 0.2[s]$.



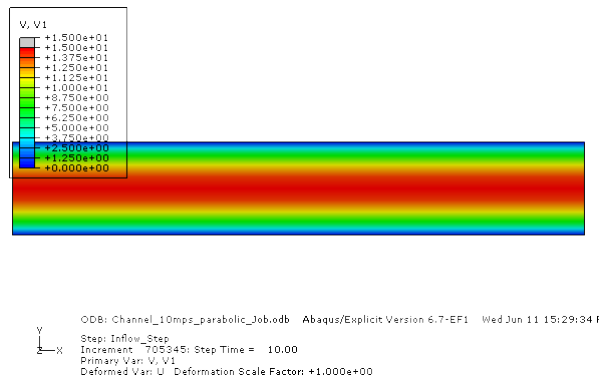
(e) $t = 0.25[s]$.



(f) $t = 0.55[s]$.

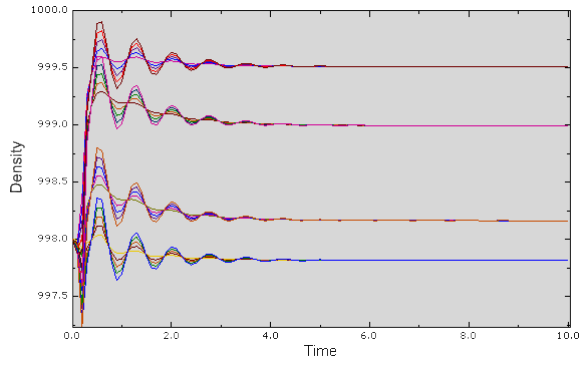


(g) $t = 0.75[s]$.

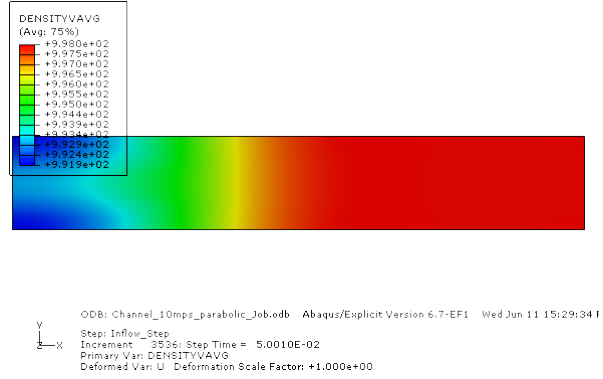


(h) $t = 10[s]$.

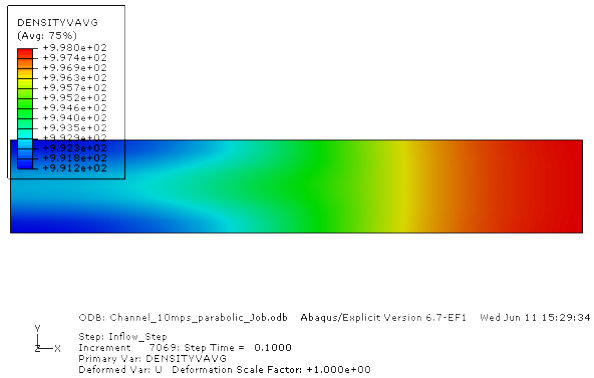
Figure 9.10: $V_1[\frac{m}{s}], c_0 = 1483[\frac{m}{s}]$ at different times.



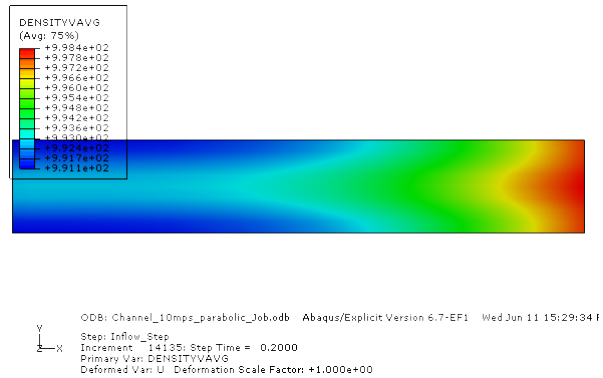
(a) $t \in [0, 10][s]$. The different lines represent the V_1 field for nodes along the y axis.



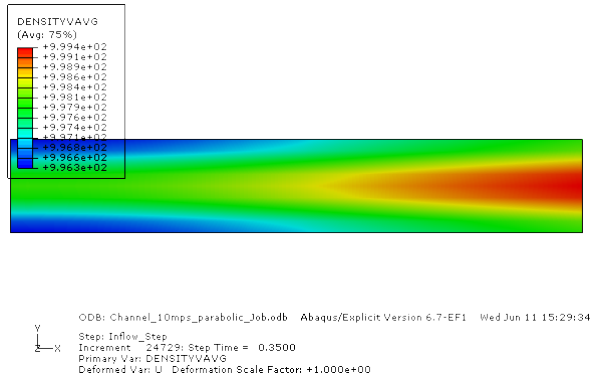
(b) $t = 5.001 \cdot 10^{-2}[s]$.



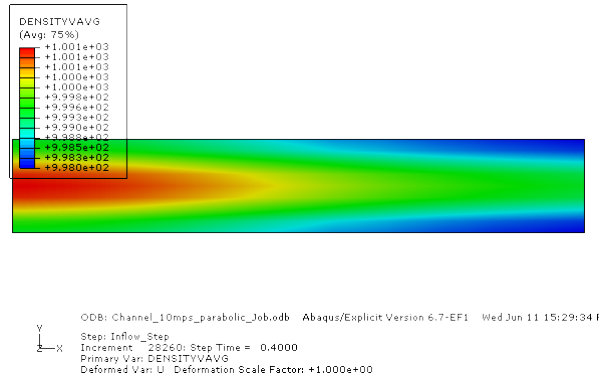
(c) $t = 0.1[s]$.



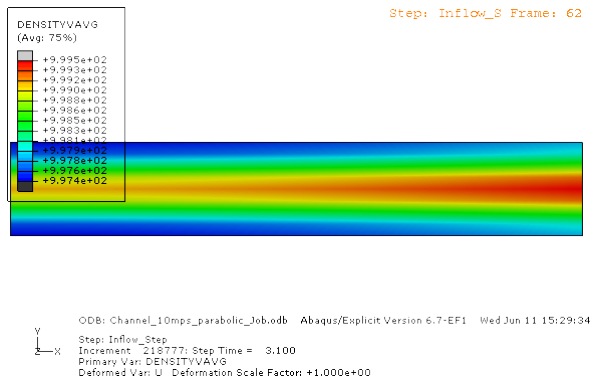
(d) $t = 0.2[s]$.



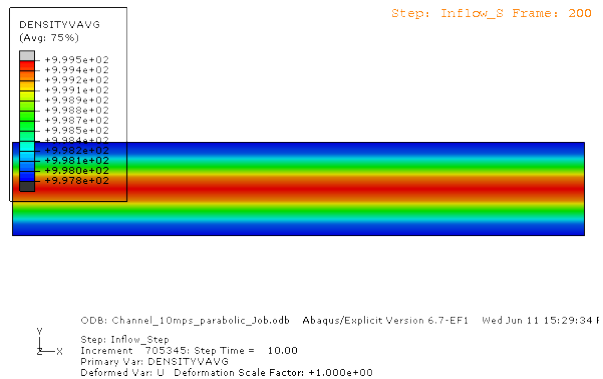
(e) $t = 0.35[s]$.



(f) $t = 0.4[s]$.

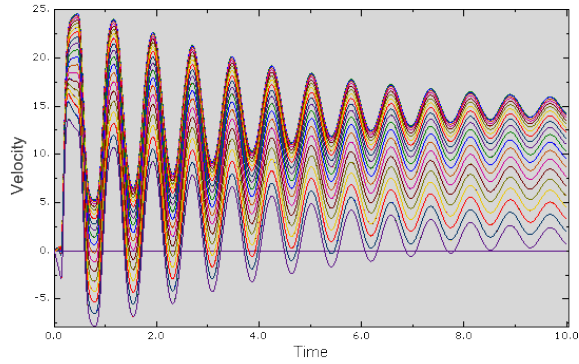


(g) $t = 3.1[s]$.



(h) $t = 10[s]$.

Figure 9.11: density $[\frac{kg}{m^3}]$, $c_0 = 1483[\frac{m}{s}]$ at different times.

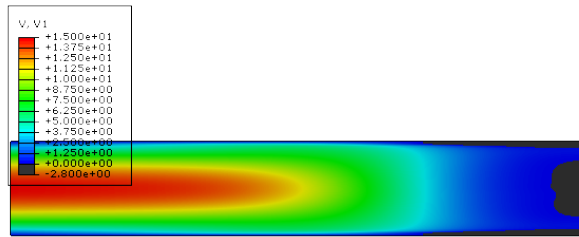


(a) $t \in [0, 10][s]$. The different lines represent the V_1 field for nodes along the y axis.



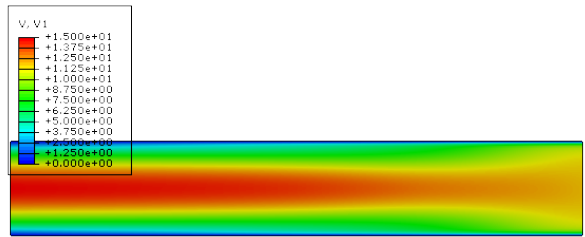
ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 1570; Step Time = 1.0005E-02
 Primary Var: V, V1
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(b) $t = 1.0005 \cdot 10^{-2}[s]$.



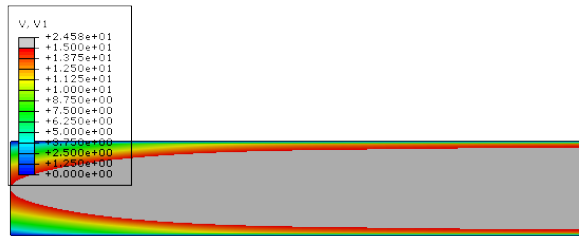
ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 20393; Step Time = 0.1300
 Primary Var: V, V1
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(c) $t = 0.13[s]$.



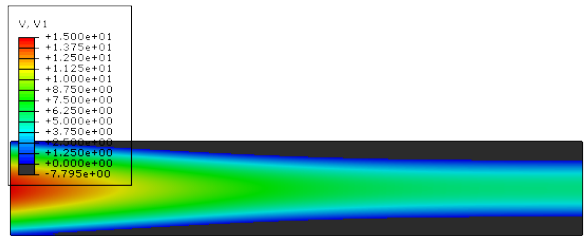
ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 31371; Step Time = 0.2000
 Primary Var: V, V1
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(d) $t = 0.2[s]$.



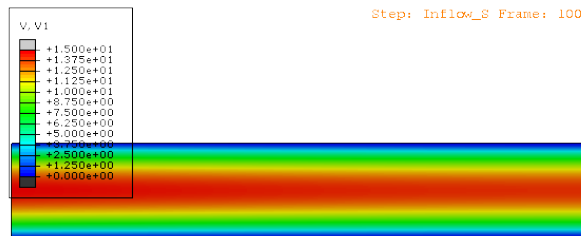
ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 59593; Step Time = 0.3800
 Primary Var: V, V1
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(e) $t = 0.38[s]$.



ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 125396; Step Time = 0.8000
 Primary Var: V, V1
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(f) $t = 0.8[s]$.



ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 1563218; Step Time = 10.00
 Primary Var: V, V1
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(g) $t = 10[s]$.

Figure 9.12: $V_1[\frac{m}{s}], c_0 = 1000[\frac{m}{s}]$ at different times.

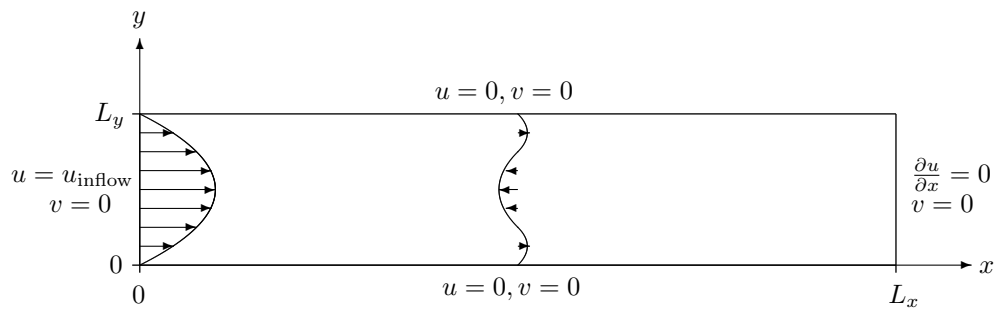
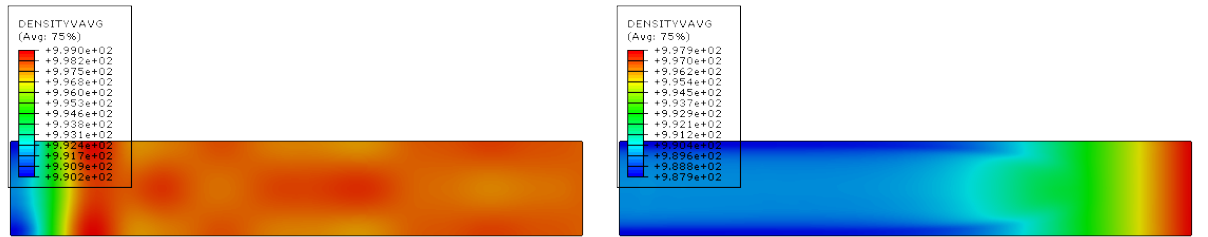


Figure 9.13: Womersley velocity profile due to no slip conditions and compression effects.

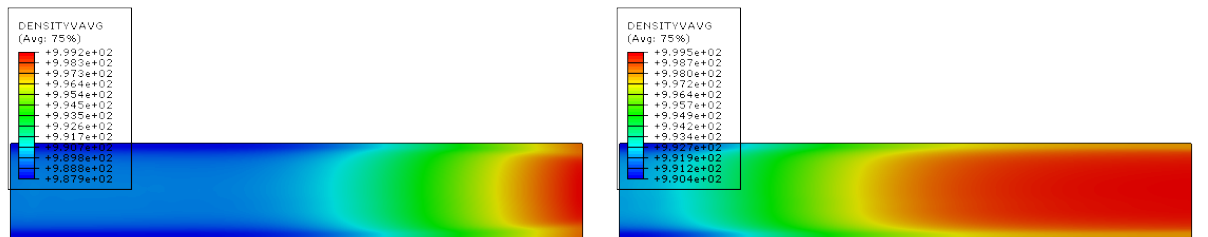


ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 1570; Step Time = 1.0005E-02
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 25098; Step Time = 0.1600
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(a) $t = 1.0005 \cdot 10^{-2} [s]$.

(b) $t = 0.16 [s]$.

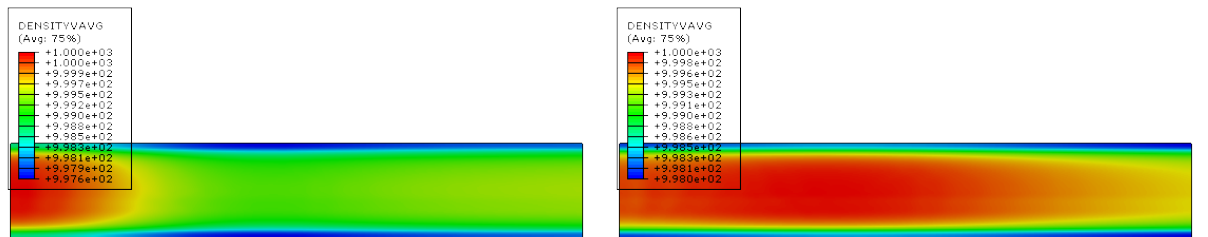


ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 37644; Step Time = 0.2400
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 53322; Step Time = 0.3400
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(c) $t = 0.24 [s]$.

(d) $t = 0.34 [s]$.

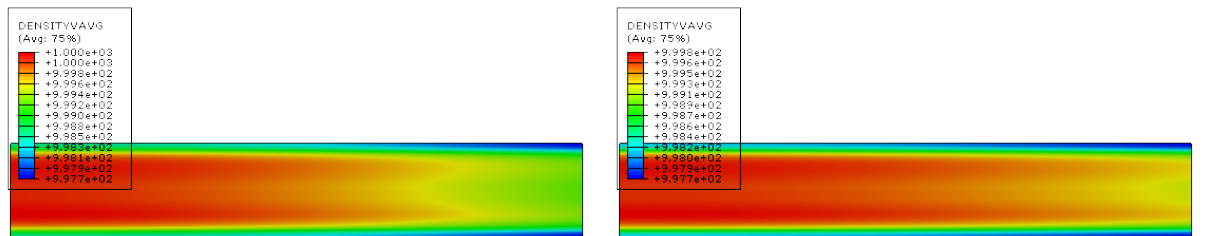


ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 61160; Step Time = 0.3900
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 10697; Step Time = 0.7700
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(e) $t = 0.39 [s]$.

(f) $t = 0.77 [s]$.



ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 1539797; Step Time = 9.850
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

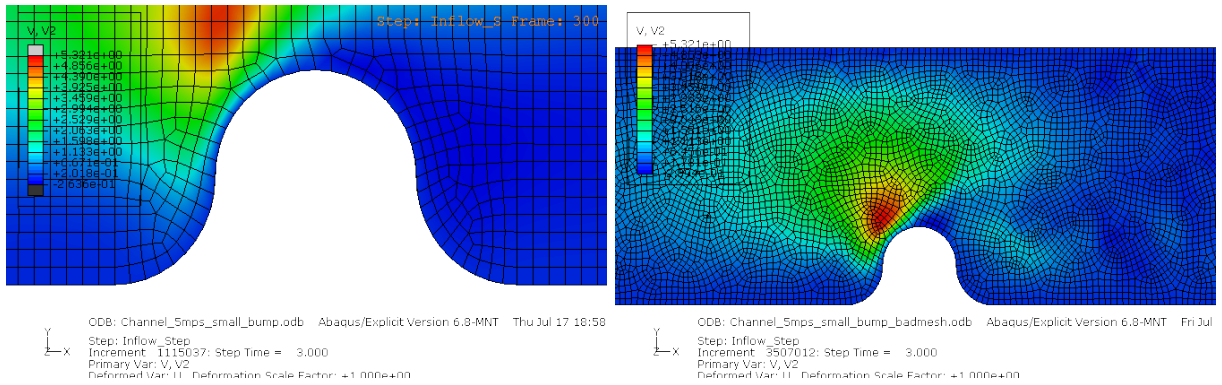
ODB: Channel_10mps_parabolic_c0=1000_Job.odb Abaqus/Explicit Version 6.8-MNT Wed Jul
 Step: Inflow_Step
 Increment: 1563218; Step Time = 10.00
 Primary Var: DENSITYVAVG
 Deformed Var: U Deformation Scale Factor: +1.000e+00

(g) $t = 9.85 [s]$.

(h) $t = 10 [s]$.

Figure 9.14: density $[\frac{kg}{m^3}]$, $c_0 = 1000 [\frac{m}{s}]$ at different times.

9.4.3 Solution channel flow with bump



(a) V_2 for partitioned domain, locally unstructured mesh. (b) V_2 for unpartitioned domain, completely unstructured mesh.

Figure 9.15: mesh distortion.

For considering the influence of the mesh on the solution we make a small bump in the domain of the channel, cf. Figures 9.16 and 9.15. We consider the cases where the domain is partitioned around the bump and unpartitioned. In the partitioned case, the mesh is only locally unstructured, cf. Figure 9.15a. In the unpartitioned case, the mesh over the whole domain is unstructured, cf. Figure 9.15b.

There is a difference in the solutions of the two cases. There is an absolute difference in velocity of $O(10^{-1})$ to the right of the bump and at the left end of the red area, cf. Figure 9.16. The computation time t_{CPU} and stable time step size h_t differ a lot, cf. Table 9.4. The stable time step size differs a factor 3.2 and the CPU run time a factor 4.2. Although the edges of the domain were seeded equally, the unstructuredness of the mesh as a result of the bump has increased the number of elements in the unpartitioned case compared to the partitioned case. This increase in DOF increases the computation time. The number of DOF differ a factor 1.2, whereas the computation times differ a factor of 4.2. In contrast with FDM and FVM, FEM calculates all quantities per element. Thus, unstructuredness of a grid does not introduce the need of additional interpolation in case of 'regular' FEM. Unstructuredness does, however, make the element matrices less sparse which increases computation time. Furthermore, the critical element is worse off cf. Figure 9.17, resulting in a smaller critical time step size. The combination of the smaller critical time step size and less sparse element matrices justifies the difference in computation time. The locally partitioned approach is clearly superior.

	#DOF	$h_t[s]$	$t_{CPU}[s]$
locally unstructured mesh	62670	$2.687 \cdot 10^{-6}$	121155
completely unstructured mesh	73752	$8.537 \cdot 10^{-7}$	509653

Table 9.4: results channel flow with bump model.

9.4.4 Formulation channel flow with cube

Adding the cube to the channel flow can be done in two ways. It can be modeled by prescribing a no slip BC or by adding a Lagrangian body. For the no slip BC a square hole is made in the Eulerian domain, cf. Figure 9.18. The DOFs $V_1 = V_2 = V_3 = 0$ in the BC, represent zero normal/tangential velocity and dimension reduction respectively. In case of the Lagrangian cube body general contact has to be prescribed for rough contact and the penalty method to avoid normal penetration and separation. The meshes are taken the same for the sake of comparison.

An alternative, interesting case to formulate and compare can be a frictionless cube in the channel flow. In the BC case, the normal velocity should be prescribed zero. The tangential velocity is, contradicting the no slip cube, not prescribed. In the Lagrangian body case, the contact can be prescribed frictionless under interaction properties. This case is however not covered in this report.

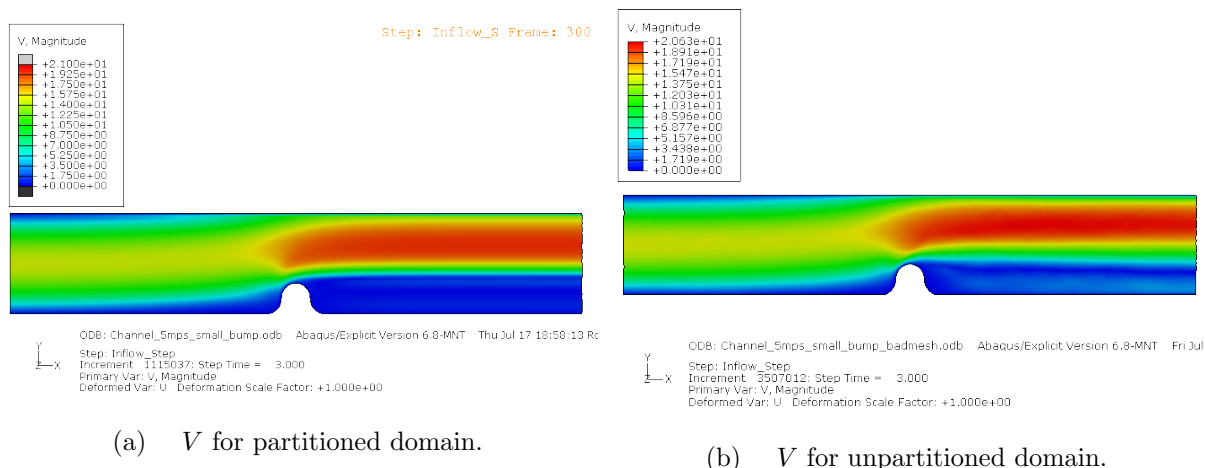


Figure 9.16: channel flow with bump.

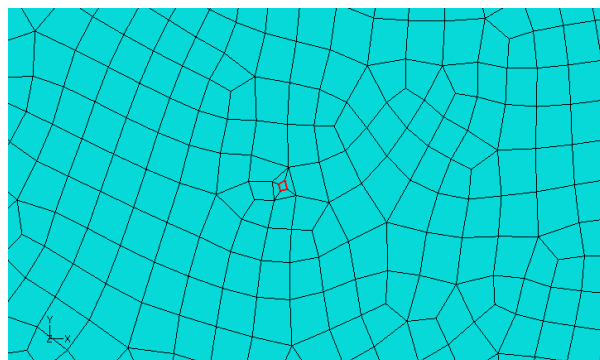


Figure 9.17: critical element distorted mesh.

9.4.5 Solution channel flow with cube

The results obtained from Abaqus are disappointing. A time frame of four seconds was completed in the case with the cube modeled by a Lagrangian body. Herefor Abaqus version 6.8.1 was used. The BC case encounters problems. When run with 6.8.1, the critical time step size decreased to an $O(10^{-18})$ at 3.022 seconds. Letting the simulation finish would take forever. Cancelling the analysis unfortunately resulted in a unrecoverable output database. When run with 6.8-MNT, the critical time step size decreased to an $O(10^{-14})$ at 1.726 seconds. Again, aborting the job corrupted the output database. Nevertheless, the fact that the BC case runs unstable and the Lagrangian case stable on the time frame of four seconds, confirms that analysis differ up to an unacceptable extent. There is even an enormous difference in the solutions from 6.8.1 and 6.8-MNT. Although this might be explained by the memory error in 6.8.1, the 6.8-MNT run should still be running stable for the same time the Lagrangian case does. Details on the versions can be found in chapter 11 and appendix A.

Concerning the solution obtained for the Lagrangian case, it is not adequate. Strange phenomena like the blimp in Figure 9.19a appear. This is an error of $O(10^0)$. Although there are some Von Kármán vortex street effects cf. 9.4.7, within 1.5 seconds these are distorted cf. Figures 9.19d-h. As of $t = 3.5[s]$, the solution is nonsense. Although the mesh is completely uniform cf. Figure 9.18, there is a constant change in leakage in the Lagrangian body, recalling section 7.5. This is depicted in Figures 9.19d-h.

Perhaps these inadequate results can be explained by the fact that $Re = \frac{UL}{\nu} = \frac{1.5 \cdot 10 \cdot 0.1}{\frac{0.001}{998}} = 7.5 \cdot 10^5$, which is in the low regimes of turbulence. Nevertheless, the errors mentioned already appear when the inflow velocity profile is "building up", when the flow is still in the regimes of laminar flow. The fine mesh has resulted in a computation time of $7293508.50[s] = 2025.97[h]$ which is disastrous. Abaqus has completely failed to suffice in this test case.

9.4.6 Formulation channel flow with cylinder

Adding the cylinder to the channel flow can be done in two ways. It can be modeled by prescribing a no slip BC or by adding a Lagrangian body. For the no slip BC cylinder a circular hole is made in the Eulerian domain, cf. Figure 9.20. The DOFs $V_1 = V_2 = V_3 = 0$ in the BC, represent zero normal/tangential velocity and dimension reduction respectively. In case of the Lagrangian cylinder body, cf. Figure 9.21, general contact has to be prescribed for rough contact and the penalty method to avoid normal penetration and separation [32].

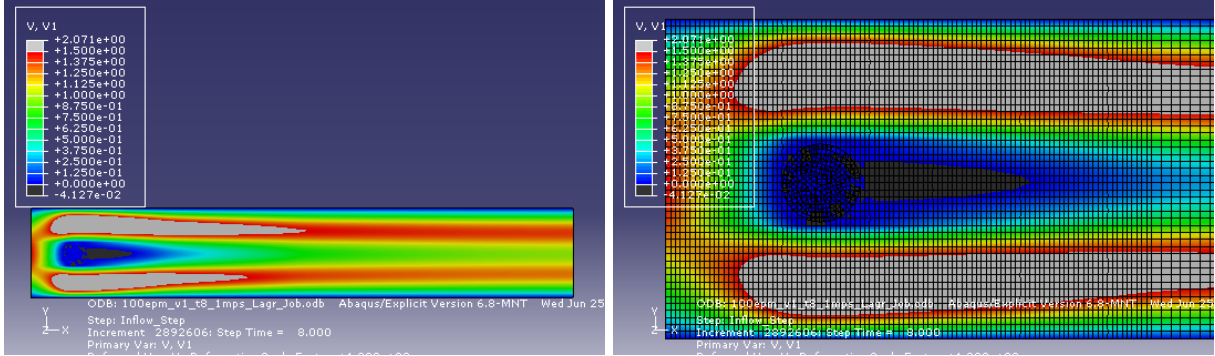


Figure 9.21: cylinder modeled by a Lagrangian body, mesh of domain.

An alternative, interesting case to formulate and compare can be a frictionless cylinder in the channel flow. In the BC case, the normal velocity should be prescribed zero. The tangential velocity is, contradicting the no slip cylinder, not prescribed. It is replaced by the BC $\boldsymbol{\tau} = \mathbf{0}$. In the Lagrangian body case, the contact can be prescribed frictionless under interaction properties. This case is however not covered in this report.

9.4.7 Solution channel flow with cylinder

The results obtained with Abaqus are surprising. Either if the cylinder is modeled by a BC or a Lagrangian body, the results should be the same. This is not the case. The solution of the model where the cylinder is modeled by a BC contains a Kármán vortex street. In our case $Re = \frac{UL}{\nu} = \frac{1.5 \cdot 10^{-0.1}}{\frac{0.001}{998}} = 7.5 \cdot 10^5$ which is out of the range for the formula but still of $O(10^5)$. The frequency of the Kármán vortex street of the Abaqus solution is completely off. According to (9.5), $f = 29.7[\frac{1}{s}]$. This does not correspond with the $f = 4.25[\frac{1}{s}]$ in Figure 9.22(a). This can be explained by the fact that (9.5) is an empirical law, which are often only accurate around the middle of the prescribed valid domain.

As a part of the contact definitions in Abaqus there is the option of fluid separating from surfaces. This gives a somewhat strange first impression of the code because it should be able to detect this itself. In this case the flow should not be able to separate from the cylinder after passing it. After all, an under pressure will exert a force on the fluid pulling it back, with $F \rightarrow \infty$ when $p \downarrow 0$. However, the option is available for model considerations and simplifications. If for instance a car crashing into a wall would be modeled with the CEL method, the wall would be the Lagrangian body and the car would be the fluid. During the crash, particles of the latter should be able to separate from the carbody. The results of allowing separation is depicted in Figure 9.23. The flow is separated by the cylinder and does not reattach. Besides that, there are errors in the solution. Despite the mesh containing one hundred elements per meter, the flow has an irregular pattern. It is also clear that some fluid has leaked into the Lagrangian body, cf. Figure 9.23h. This is the result of the fact that contact is not enforced on the fluid when $EVF < 0.5$. The result for the case where no separation is allowed, are depicted in Figure 9.24. In the first 1.6 seconds, the solution seems plausible. A Kármán vortex street originates as in the BC case. At $t = 1.8[s]$ however, the vortex street vanishes and the solution becomes more chaotic. Mesh refinement does not prevent this issue. This is an obscurity that can not be overcome without knowing the details of the implementation of Abaqus.

From Table 9.5 it follows that the no separation option has an impact on the computation time. Although it seems trivial in fluid calculations, it brings about factor 1.8 difference while the critical time step size is of the same order. This can, unfortunately, again not be explained because the code of Abaqus is not available for us.

#DOF	separation	$\bar{u}[\frac{m}{s}]$	Re	$t_{total}[s]$	$h_t[s]$	$t_{CPU}[s]/[h]$
57423	yes	10	10^6	8	$2.778 \cdot 10^{-6}$	530964.19/147.5
57423	no	10	10^6	8	$1.747 \cdot 10^{-6}$	961023.50/267.0

Table 9.5: data of the channel flow with cylinder model.

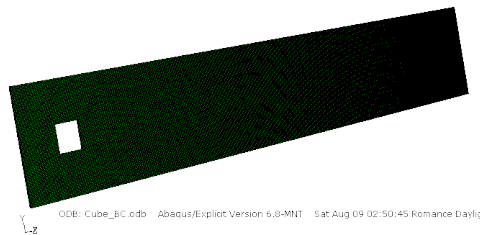
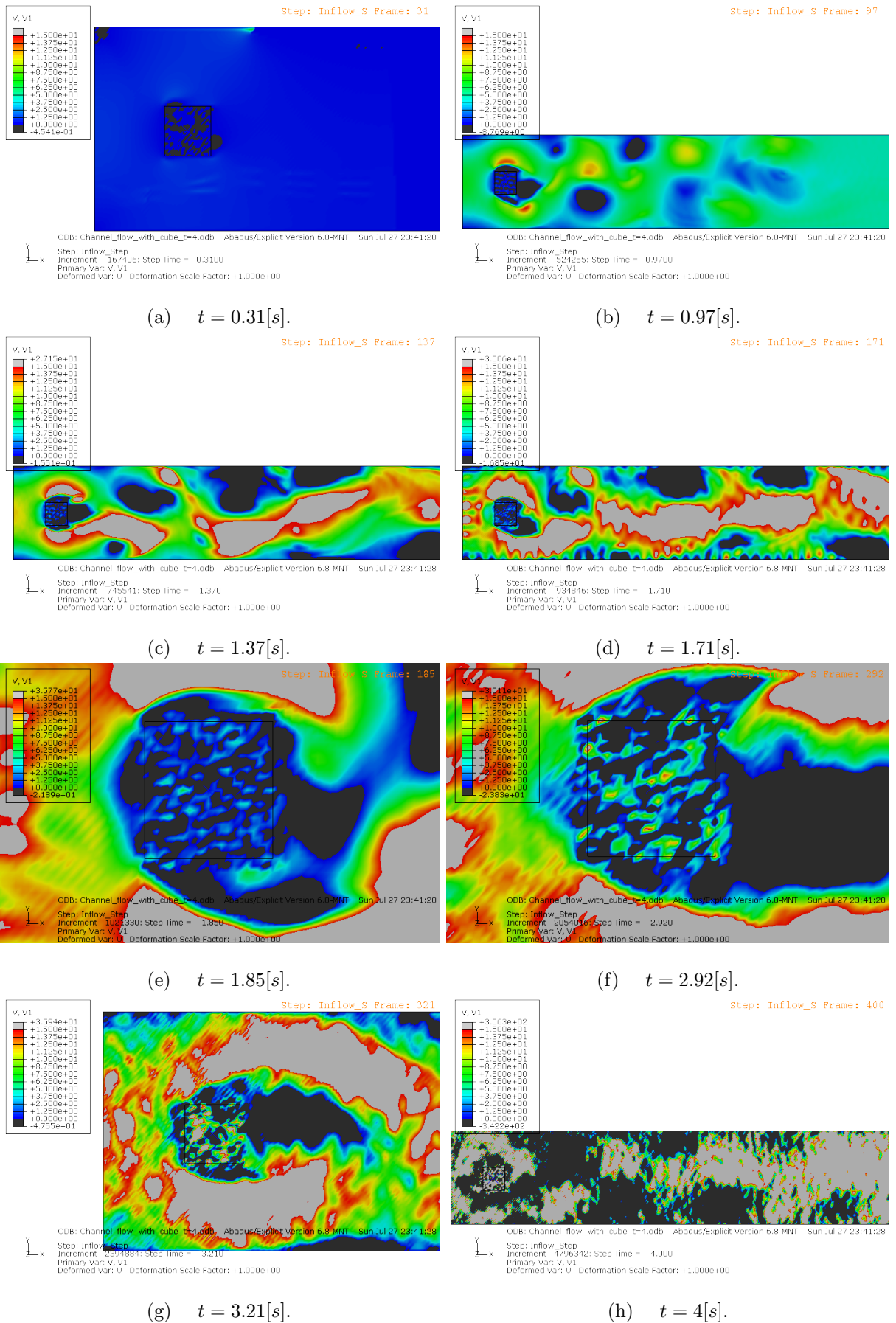


Figure 9.18: cube modeled by a BC.

Figure 9.19: channel flow with Lagrangian cube, $V_1 [\frac{m}{s}]$ at different times.

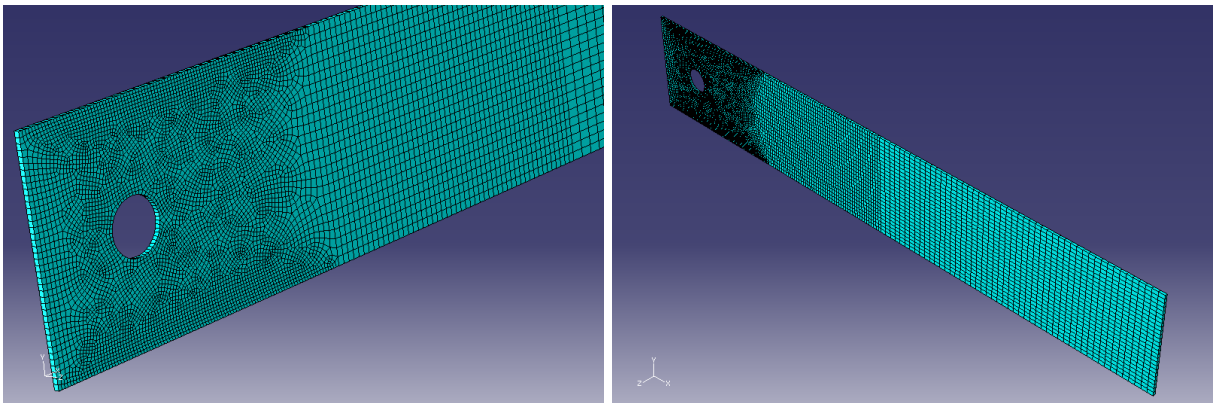
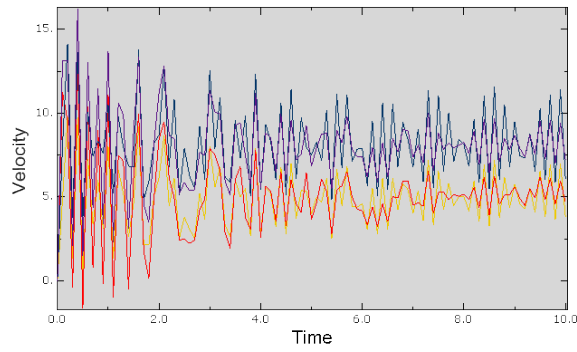
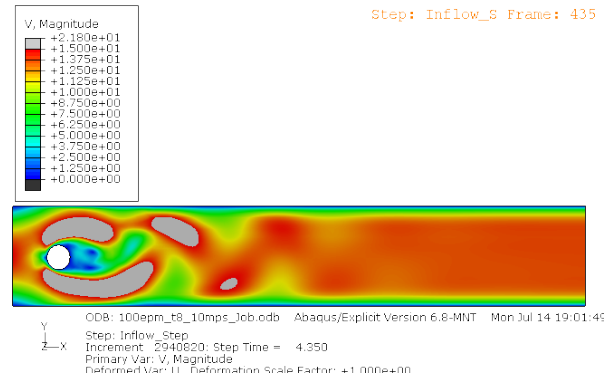


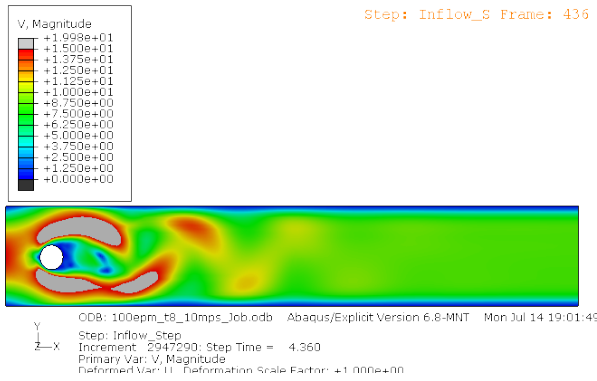
Figure 9.20: cylinder modeled as a no slip BC.



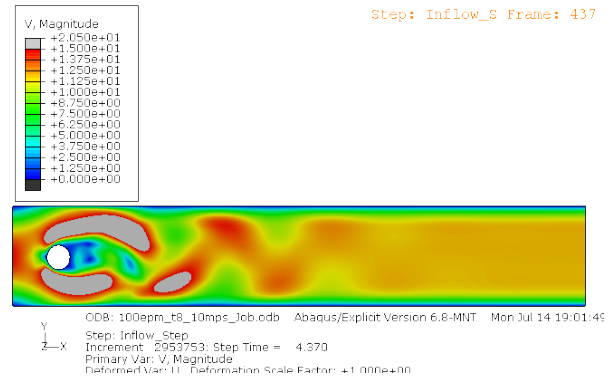
(a) $t \in [0, 10][s]$. The different lines represent the V field for nodes along the y axis.



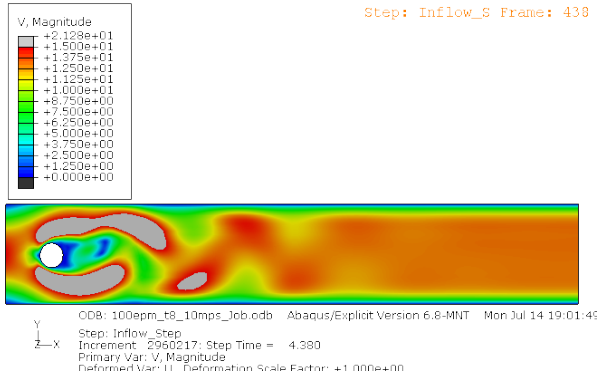
(b) $t = 4.35[s]$.



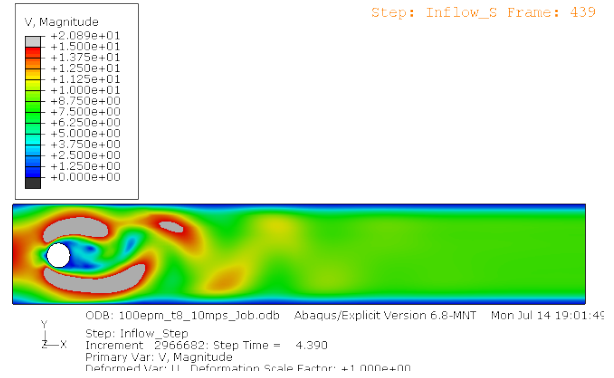
(c) $t = 4.36[s]$.



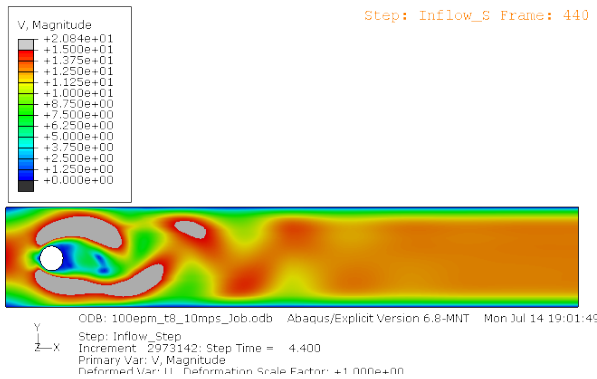
(d) $t = 4.37[s]$.



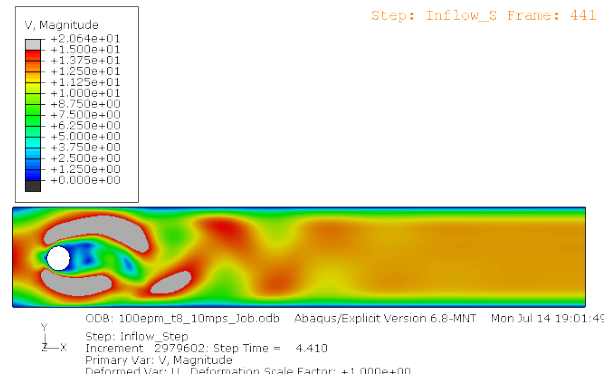
(e) $t = 4.38[s]$.



(f) $t = 4.39[s]$.

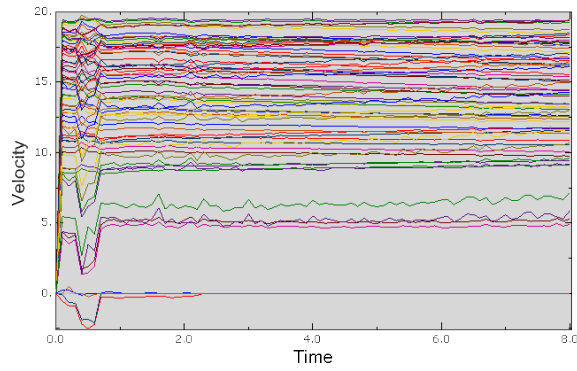


(g) $t = 4.4[s]$.

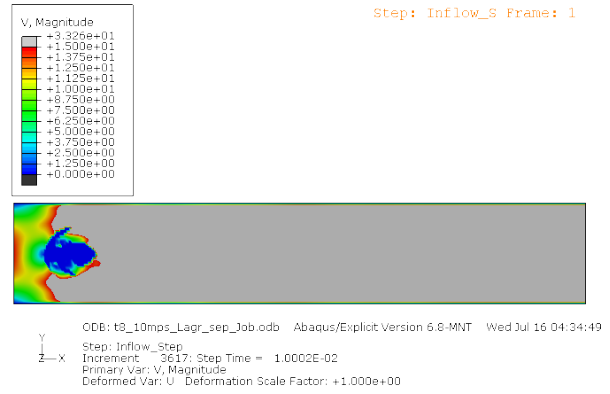


(h) $t = 4.41[s]$.

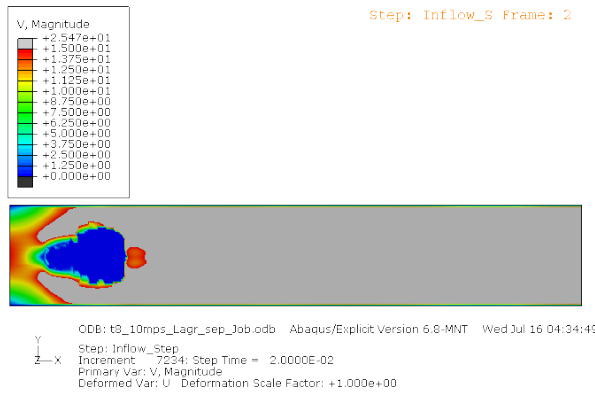
Figure 9.22: channel flow with cylinder BC, $V[\frac{m}{s}]$ at different times.



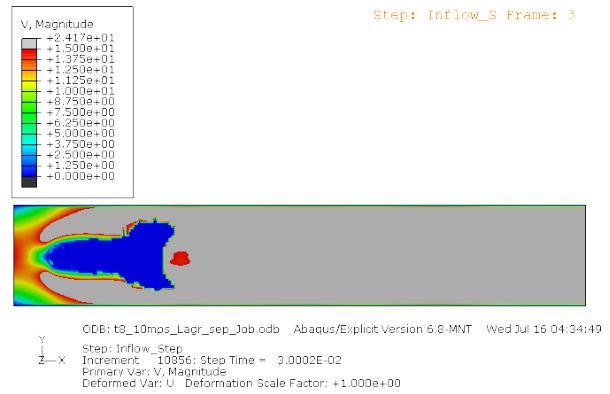
(a) $t \in [0, 10][s]$. Different lines represent the V filed for nodes along the y axis.



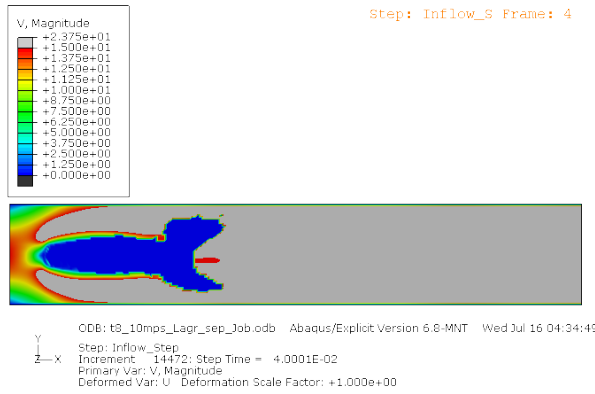
(b) $t = 1.0002 \cdot 10^{-2}[s]$.



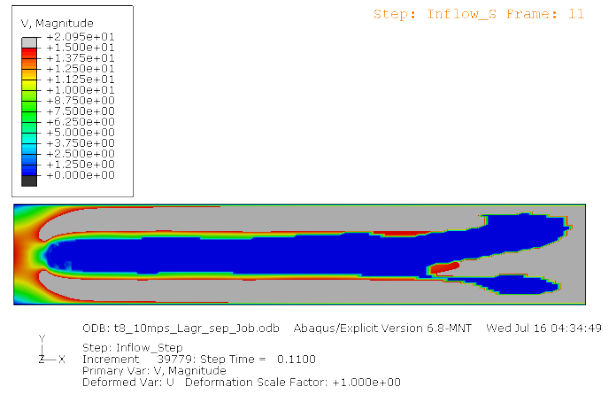
(c) $t = 2 \cdot 10^{-2}[s]$.



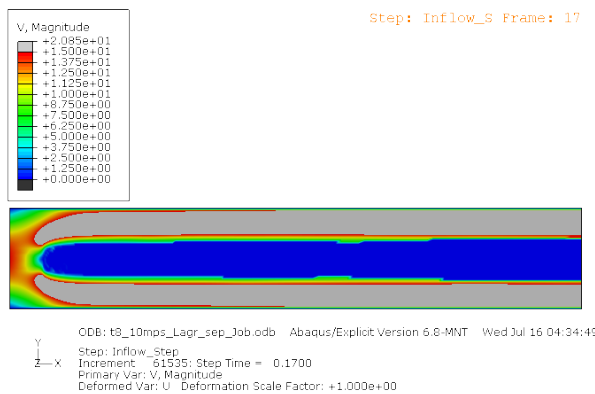
(d) $t = 3.0002 \cdot 10^{-2}[s]$.



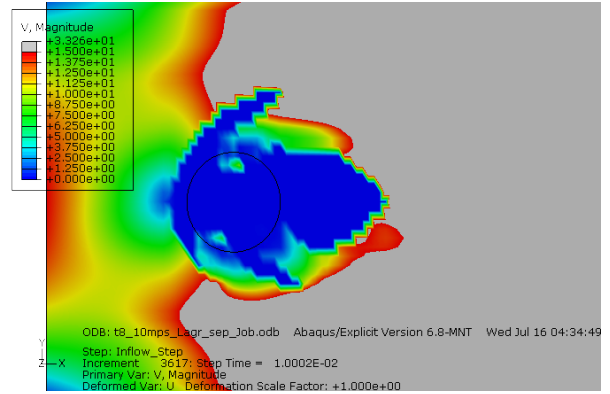
(e) $t = 4.0001 \cdot 10^{-2}[s]$.



(f) $t = 0.11[s]$.



(g) $t = 0.17[s]$.



(h) $t = 1.0002 \cdot 10^{-2}[s]$.

Figure 9.23: channel flow with Lagrangian cylinder, separation allowed, $V[\frac{m}{s}]$ at different times.

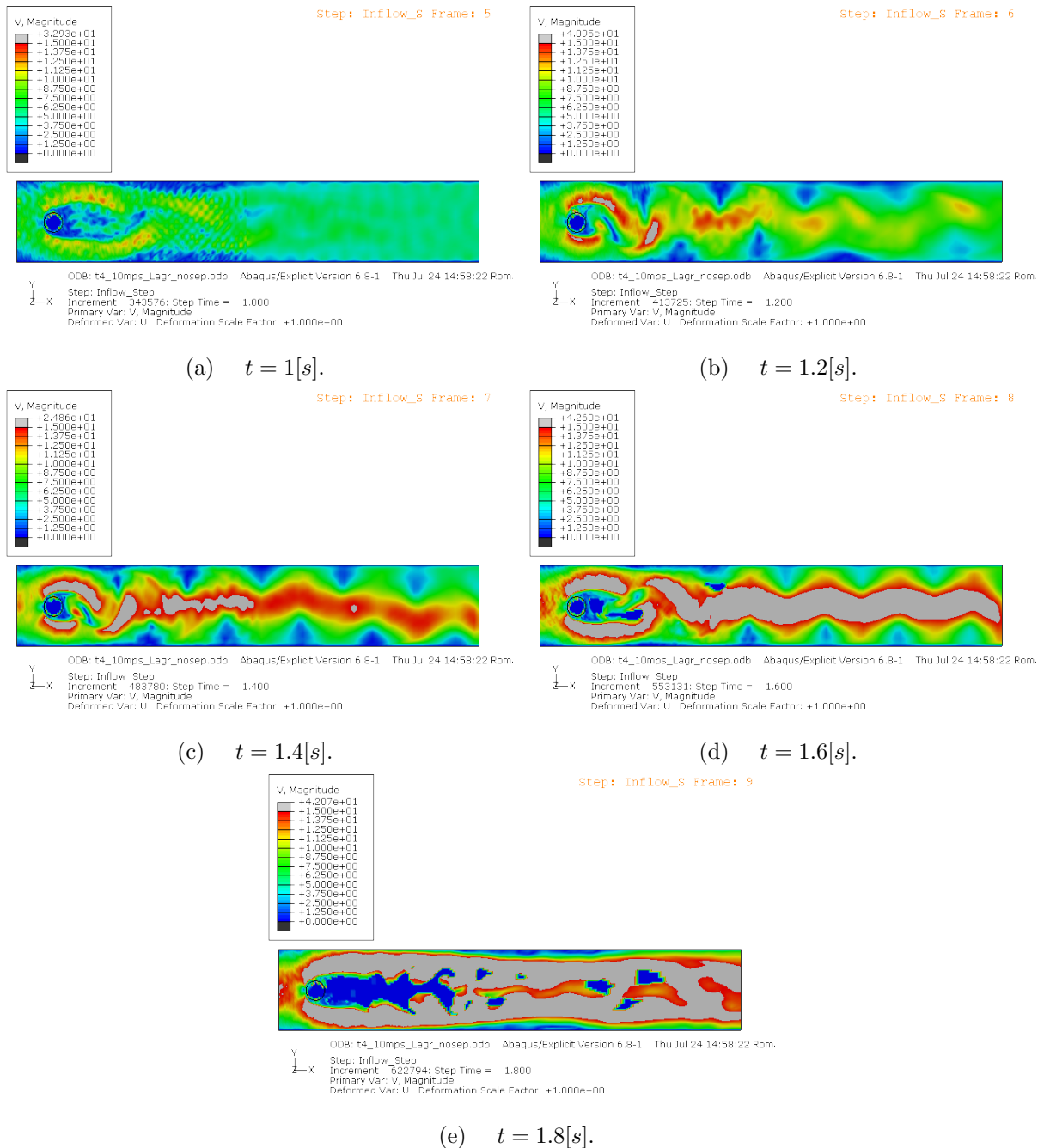


Figure 9.24: channel flow with Lagrangian cylinder, separation not allowed, $V[\frac{m}{s}]$ at different times.

9.4.8 Formulation channel flow with elastic body

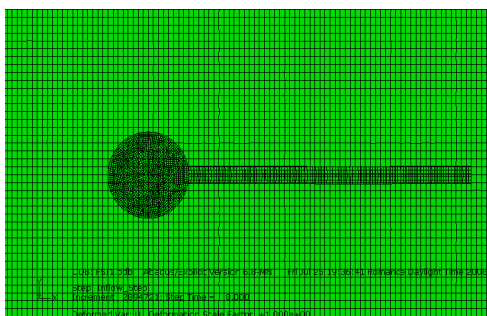


Figure 9.25: mesh for the benchmark.

Starting off with a simplified model and increasing the complexity of the model we have arrived at the full model as described in [36]. In addition to the channel flow with cylinder, an elastic flag is tied on the cylinder cf. Figures 9.1 and 9.25. The structure properties are now also of importance.

The elastic body is subjected to drag and lift forces due to the fluid flowing through the channel. As a result of these forces, the body will deform. Thinking of a flag on a pole in the wind, we, more specifically, expect the flag to oscillate.

Our interest lies in the interaction between the elastic body and the fluid. Herewith, quantities of interest are displacement and frequency of the expected oscillation.

The flag is merely constructed in Abaqus by defining a part with elastic material properties. Restrict the left side of the flag from movement and rotation to 'bind' it to the cylinder.

The material properties specified in [36] are different from those required by Abaqus. They are summarised in Table 9.6. Note that the speed of sound $c_0 = 1483[\frac{m}{s}]$ is taken. It is just for the definition of the bulk modulus K of the fluid, which herewith resembles the bulk modulus of water. Density and viscosity are taken $10^3[\frac{kg}{m^3}]$ and $1[Pa \cdot s]$ respectively.

	FSI test 1	FSI test 2	FSI test 3
$\rho_s[\frac{kg}{m^3}]$	10^3	10^4	10^3
$E[Pa]$	$1.4 \cdot 10^6$	$1.4 \cdot 10^6$	$5.6 \cdot 10^6$
$\nu[\frac{m}{m}]$	0.4	0.4	0.4
$\bar{u}[\frac{m}{s}]$	0.2	0.4	0.4

Table 9.6: required parameters for Abaqus.

9.4.9 Solution channel flow with elastic body

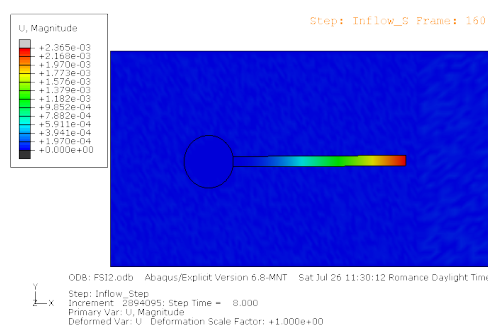


Figure 9.26: displacement of flag in FSI test 1.

Article [36] states that FSI test 1 results in a steady state solution. Tests 2 and 3 result in periodic solutions. From comparison of Figures 9.28, 9.27 and Table 9.7 it clearly follows that Abaqus does not

conform this. Test 1 gives an alternating solution cf. Figure 9.26 and tests 2 and 3 give steady state solutions. The horizontal and vertical displacements of the right end of the flag differ a lot from the results given in [36]. In the majority of the cases, the order is not even the same, cf. Table 9.8. These differences can not be explained without knowledge about the implementation of Abaqus. The models are constructed conform Simulia's recommendations, which should be sufficient for obtaining a plausible solution.

	FSI test 1	FSI test 2	FSI test 3
$u_x [\cdot 10^{-3} m]$	0.0227	-14.58 ± 12.44	-2.69 ± 2.53
$u_y [\cdot 10^{-3} m]$	0.8209	1.23 ± 80.6	1.48 ± 34.38

Table 9.7: displacement of left end of flag according to [36].

	FSI test 1	FSI test 2	FSI test 3
Abaqus $O(10 \log u_x)$	-5	-4	-3
Abaqus $O(10 \log u_y)$	-5	-3	-4
[36] $O(10 \log u_x)$	-5	-2	-3
[36] $O(10 \log u_y)$	-4	-2	-2

Table 9.8: orders of displacement of left end of flag, for both Abaqus and [36].

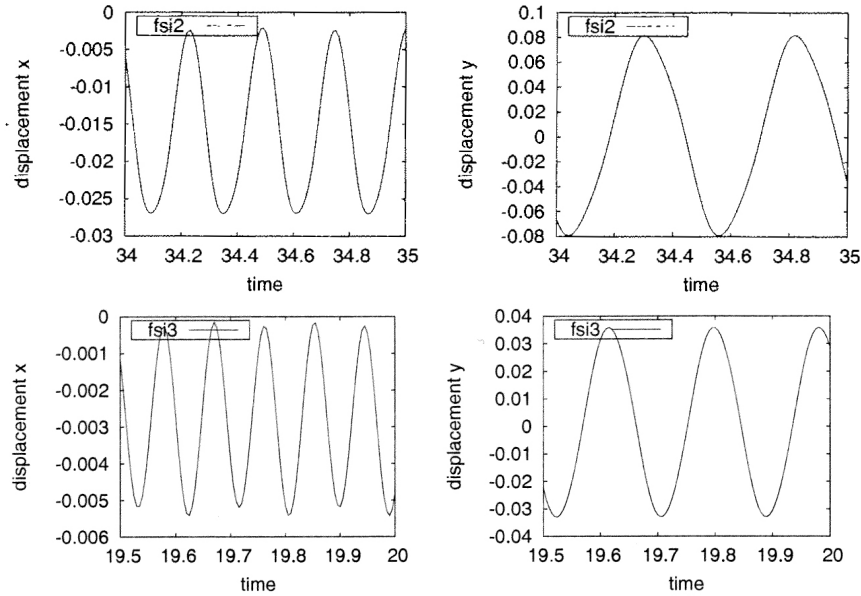
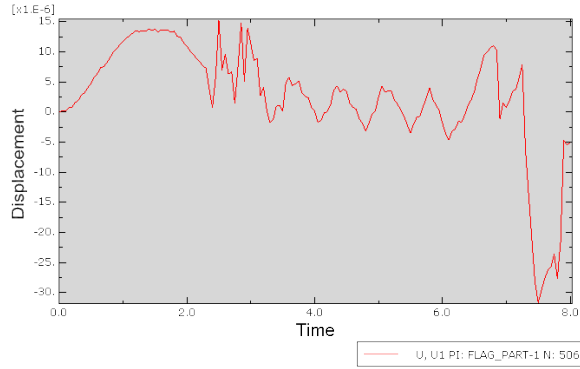


Figure 9.27: Results from FSI2 and FSI3 [36].

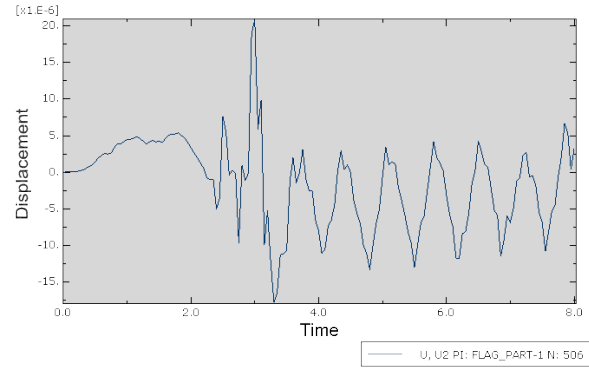
Although considering the simulation times and critical time step sizes seems redundant because the solutions of Abaqus and article [36] differ a lot, still a short survey is made for the sake of completeness. The parameters varied for the three tests are the inflow velocity, stiffness and density of the elastic body, cf. Table 9.6. It is hard to say something about their influence because they are varied separately. From the comparison of tests FSI2 and FSI3 it seems that the density and stiffness of the solid do not have a lot of influence on the computation time, since they do not differ a lot. Comparison of tests FSI1 and FSI2 has the tendency that the inflow velocity does influence the computation time a lot. They differ a factor 1.7.

test	#DOF	Re	$h_t [s]$	$t_{CPU} [s] / [h]$
FSI1	69702	10^6	$2.765 \cdot 10^{-6}$	284694.84/79.1
FSI2	69702	10^6	$2.766 \cdot 10^{-6}$	474324.09/131.8
FSI3	69702	10^6	$2.756 \cdot 10^{-6}$	565472.94/157.1

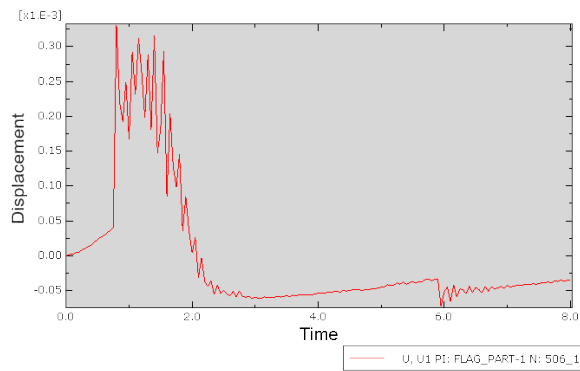
Table 9.9: data of the channel flow interacting with elastic body model.



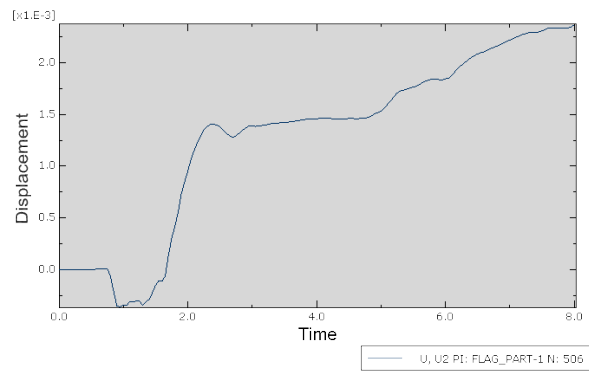
(a) FSI test 1, horizontal displacement.



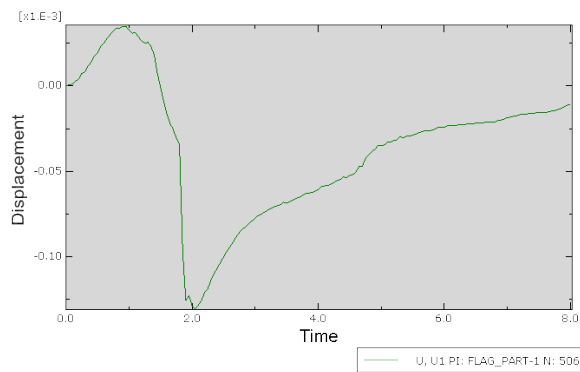
(b) FSI test 1, vertical displacement.



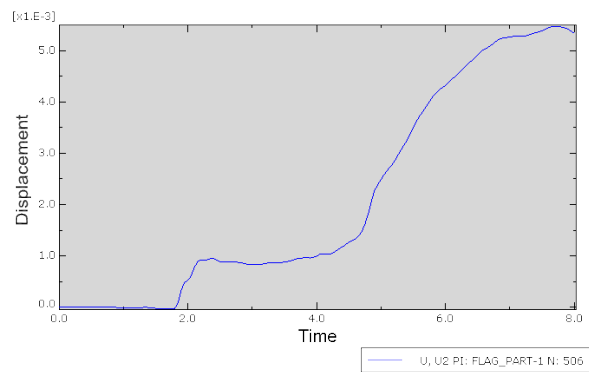
(c) FSI test 2, horizontal displacement.



(d) FSI test 2, vertical displacement.



(e) FSI test 3, horizontal displacement.



(f) FSI test 3, vertical displacement.

Figure 9.28: displacement of right end of flag.

9.5 Comparison

Comparing the Matlab and Abaqus simulations, it can be concluded that the Matlab programs run better. With the channel flow Abaqus needs a stabilisation time. In addition the computation time is long. The Matlab simulation runs flawlessly. The computation time is considerably shorter but can not be adequately compared because the number of DOF is a lot lower.

When an object is added into the channel flow, be it a cube or cylinder, Abaqus fails to produce a correct solution. In particular in the case where the cylinder or cube is modeled by a Lagrangian body in combination with contact definitions. When the cylinder is modeled by a BC, the solution is good. It reproduces a vortex street correctly. Computation time is also considerably smaller compared to the Lagrangian body case. Matlab runs fine with the cube in channel flow. It reproduces a vortex street correctly.

Unstructuredness of a mesh drastically increases computation time. The channel flow interacting with an elastic body is a complete failure.

Chapter 10

Grosch wheel

10.1 Introduction

With the academic test problems handled, the treatment of a problem of practical importance for GTC*L would be proper. A simplified hydroplaning model containing the so called Grosch wheel, a downscaled wheel with a chamfer, will be modeled. Quantities of interest are lift and drag force and footprint size. Their dependence on the velocity and compressibility of the fluid and the mesh density is to be investigated.

Formulation of the model and results of the analysis will be discussed. *Additional output is available in movie form, to be found at [2].*

10.2 Formulation

The Grosch wheel is imported from an input deck used previously for friction simulations. In addition, an Eulerian part for the fluid and an analytical rigid Lagrangian part for the road are defined. Different rubber compounds are defined and assigned to sections of the Grosch wheel, cf. Figure 10.1. Water is used as fluid.

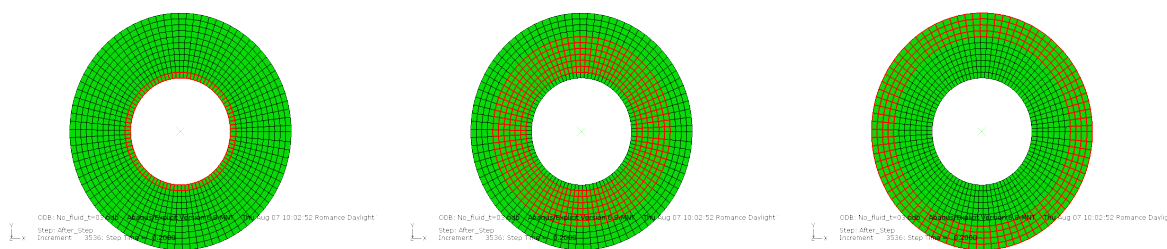


Figure 10.1: different rubber compounds assigned to different sections.

When the water comes into contact with the wheel, the wheel must already be in a steady state. To this end, the wheel is given an initial rotational velocity and a downward force induced by the cars weight, to initiate contact with the rigid nonporous road. Subsequently, the fluid gets a translational velocity corresponding to the rotational velocity of the wheel.

No penetration and no slip BCs and contact definitions are defined self evidently. The wheel is limited to displacement in vertical direction and rotation about its axis. The mesh is constructed as in Figure 10.3. The mesh is coarse for the sake of computation time. When refining, the area enclosing the footprint should be taken the finest since here the fluid is expected to be enclosed by the road and tire surface. The Eulerian elements enclosing the area of the footprint are about one fourth of the size of the Lagrangian elements. The rest of the Eulerian elements about one half and thus coarser. If the domain is rectangular, as in this case, parallel edges should contain an equal amount of mesh element vertices to conserve a structured mesh for the sake of computation time, cf. Figure 10.2. In Abaqus the amount of mesh element vertices is also called the amount of seeds. If parallel edges are not seeded equally, an unstructured mesh is constructed. Mortar elements could resolve this, but they are not available in Abaqus.

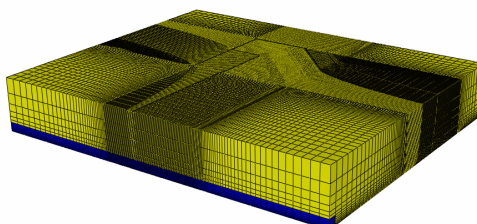


Figure 10.2: an example of parallel edges seeded equally.

10.3 Solution

The lift force is determined by the amount of water under the tire and the values ρ , v and p have there. In case of compressible 'water', another factor comes into play in causing lift force, namely compression. Or, to say it in a perhaps clearer way, expansion, being negative compression. Expansion increases lift force and therefor decreases the footprint size. Unfortunately, with hydroplaning the water will never arrive under the tire in an expanded state. To think in water terms, the water 'encounters' the tire which merely forms a narrowing, causing compression. Being in compressed state, there will be an additional expansion or lift force extra compared to the physically correct incompressible case.

The coarse mesh has the consequence that around the tire, nodes will more often than with a fine mesh have $EVF < 0.5$ resulting in more leakage through the surface of the tire. Leakage decreases the (potential) lift force. In particular this happens often in the narrow passage between the tarmac and the tire. For the water to still get under the tire, several nodes must in one time step cause a lift force, equivalent with lift that allows nodes with $EVF > 0.5$. If this is not managed, fluid will partially leak into the Lagrangian body due to $EVF < 0.5$ cells causing an unrealistic decrease in lift force. Unfortunately, varying the mesh density was not done because an even coarser grid instantly resulted in disastrous solutions and a finer mesh drastically increased the computational time.

The measure points are cf. Figure 10.4. The maximum contact normal force (CNF) as a result of fluid pressure is given for each of the five measure points, the maximum contact shear force (CSF) only for the point with the highest value. This was done because the measurements are noisy. The results are as expected. Contact normal and shear force decreases when inflow velocity increases and the tire load decreases. The wheel is subjected to the highest contact normal forces at the point next to the chamfer. The essence is that an increase in inflow velocity and decrease in tire load bring about an increase in lift and drag force, decreasing the contact normal and shear force. This is denoted in Table 10.1 and depicted in Figures 10.5, 10.6 and 10.7. The limited amount of data suggests that the relation between the contact normal force and tire load has a logarithmic or root tendency, cf. Figure 10.7. The relation between the contact shear force and the tire load is linear.

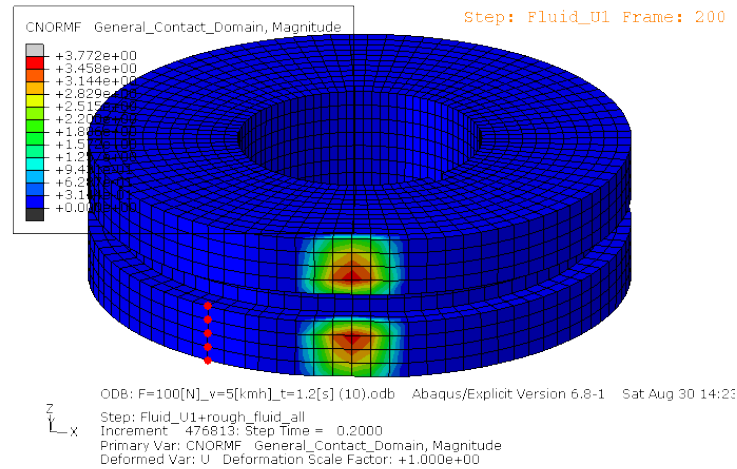


Figure 10.4: measure points in red.

	maximum contact normal force [N]	maximum contact shear force [N]
25[N], 5[$\frac{km}{h}$]	1.7/1.6/1.38/0.75/0.7	0.25
50[N], 2.5[$\frac{km}{h}$]	3.4/3/2.7/2.25/1.7	1.18
50[N], 5[$\frac{km}{h}$]	2.6/2.4/2.1/1.25/1.2	0.52
100[N], 5[$\frac{km}{h}$]	4/3.4/2.9/1.9/1.8	1.15
100[N], 10[$\frac{km}{h}$]	3/2.6/2.3/2/1.5	1
200[N], 10[$\frac{km}{h}$]	5.2/4.5/4/3.6/3	1.82

Table 10.1: contact normal and shear forces.

The coarse mesh results in interface nodes with EVF values below 0.5. A certain extent of leakage into the Lagrangian Grosch wheel body is the consequence, cf. Figure 10.8. As mentioned, this leakage also decreases potential lift force. As the EVF threshold is fixed at 0.5, a way to decrease the leakage is to

take a finer mesh. Another, more cumbersome way, is to place the Eulerian grid slightly below the contact road surface with the lowest layer of cells already filled with fluid. This somewhat manipulates the EVF values at the lower nodes to obtain less $EVF < 0.5$ nodes and thus less leakage.

Although the results are as expected in the qualitative sense, there is not much to say in the quantitative sense. Previous test models in chapters 8 and 9 have shown that Lagrangian bodies in combination with contact definitions lead to faulty results. In addition, for a downscaled wheel like the Grosch wheel, there is not much comparison material except for the alternative tool that GTC*L has in place. This comparison can reveal differences between the codes. The outcome will give an indication of the performance of the CEL method.

Last but not least the analysis time denoted in Table 10.2. Clearly something went wrong in the generation of the output because the computation times differ extremely, even though the critical time step sizes are almost the same. The output is generated by a script from Goodyear. The short computation times are with the jobs that were ran on 28 CPUs. It seems that the Goodyear script contains an error in this particular case. See also the log files, to be found on [2]. Herewith, the computation times will not be considered any further.

#DOF	$F[N]$	$v[\frac{m}{s}]$	Re	$t_{total}[s]$	$h_t[s]$	$t_{CPU}[s]/[h]$
124752	25	5	10^5	1.2	$4.171 \cdot 10^{-7}$	577759.06/160.5
124752	50	5	10^5	1.2	$4.175 \cdot 10^{-7}$	627164.81/174.2
124752	100	2.5	$5 \cdot 10^4$	1.2	$4.171 \cdot 10^{-7}$	24766.84/6.9
124752	100	5	10^5	1.2	$4.175 \cdot 10^{-7}$	43168.88/12.0
124752	100	10	$2 \cdot 10^5$	1.2	$4.17 \cdot 10^{-7}$	391904.66/108.9

Table 10.2: data of the Grosch wheel model.

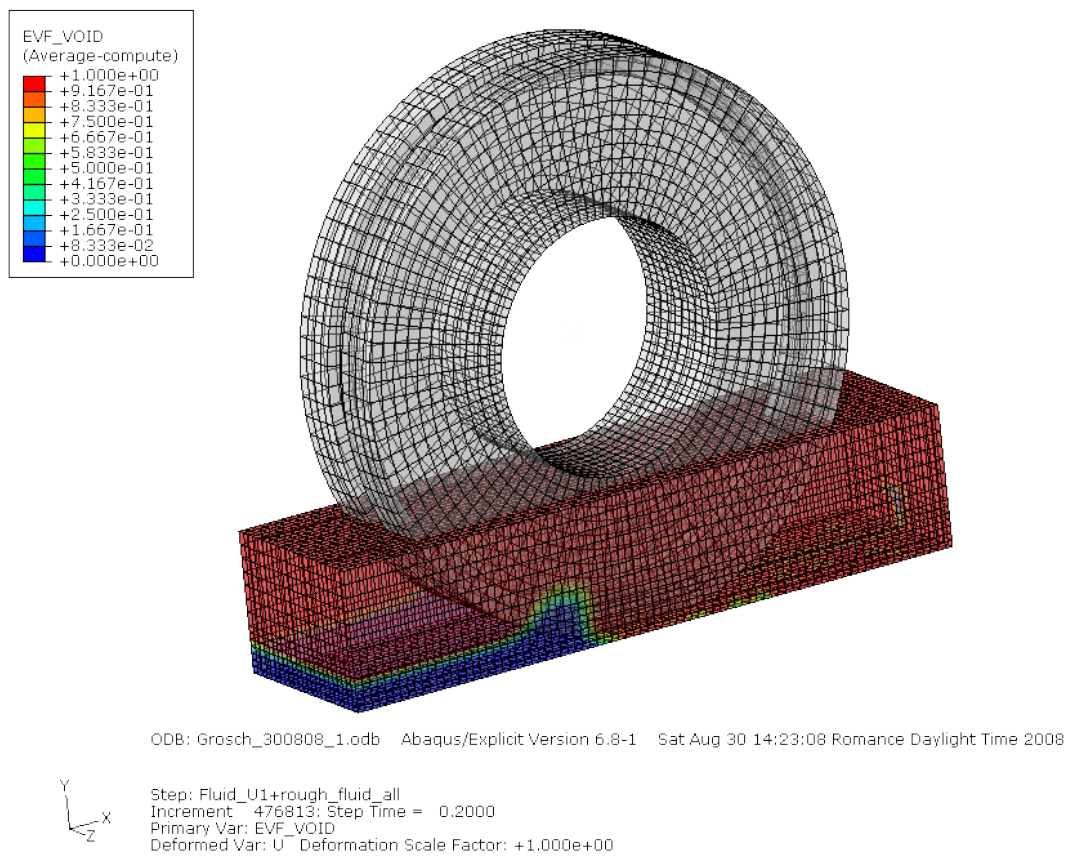


Figure 10.3: mesh Grosch wheel test model.

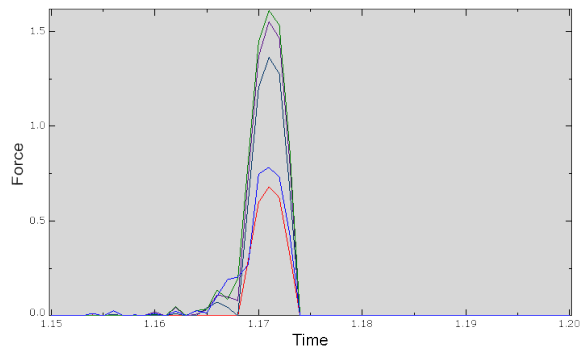
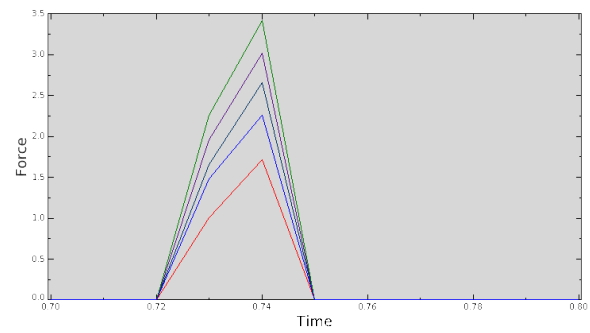
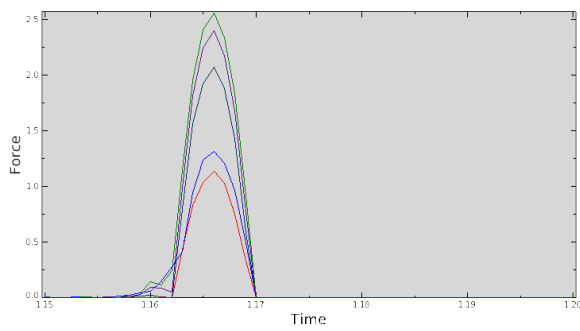
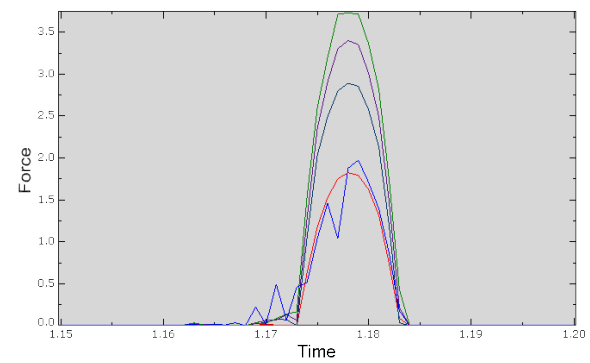
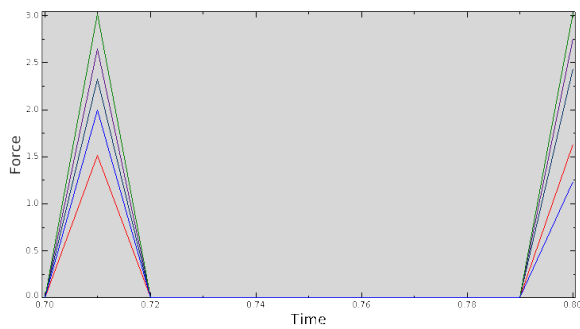
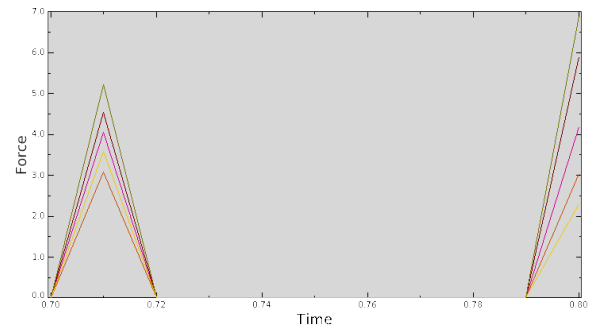
(a) $F = 25[N], v = 5[\frac{km}{h}]$.(b) $F = 50[N], v = 2.5[\frac{km}{h}]$.(c) $F = 50[N], v = 5[\frac{km}{h}]$.(d) $F = 100[N], v = 5[\frac{km}{h}]$.(e) $F = 100[N], v = 10[\frac{km}{h}]$.(f) $F = 200[N], v = 10[\frac{km}{h}]$.

Figure 10.5: contact normal force for different loads and inflow velocities, at measure points as indicated in Figure 10.4.

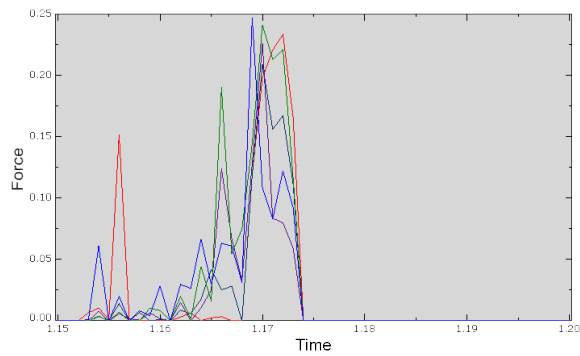
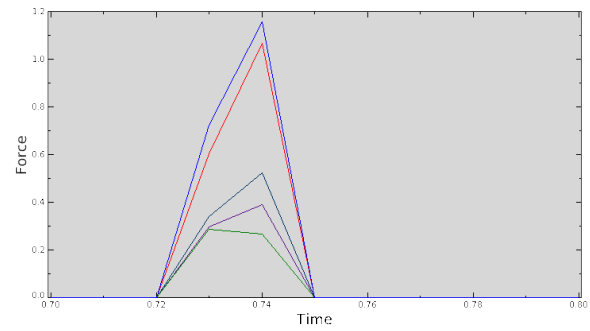
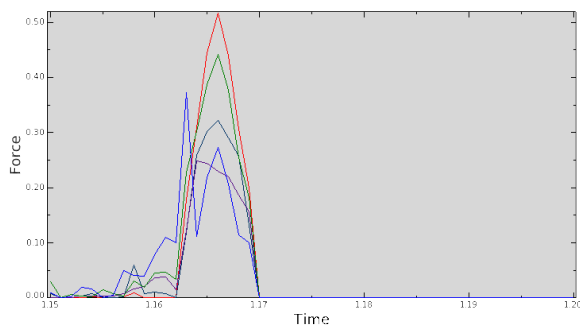
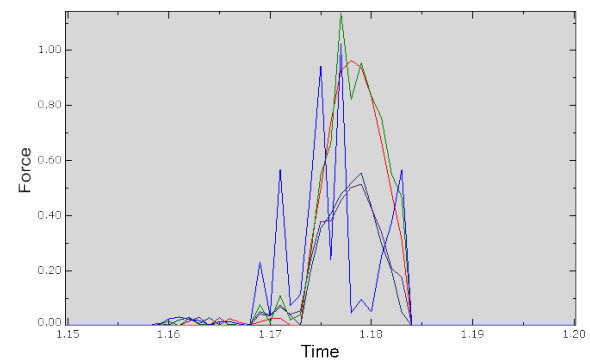
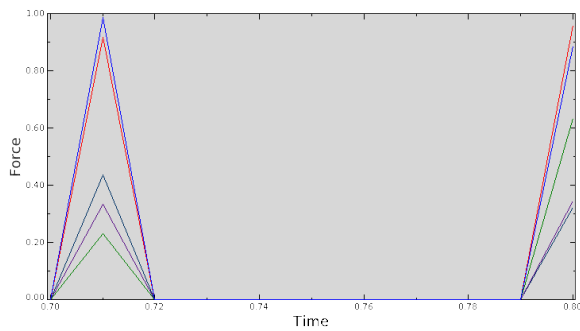
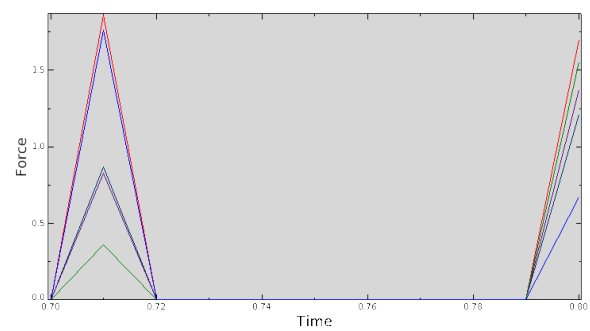
(a) $F = 25[N], v = 5[\frac{km}{h}]$.(b) $F = 50[N], v = 2.5[\frac{km}{h}]$.(c) $F = 50[N], v = 5[\frac{km}{h}]$.(d) $F = 100[N], v = 5[\frac{km}{h}]$.(e) $F = 100[N], v = 10[\frac{km}{h}]$.(f) $F = 200[N], v = 10[\frac{km}{h}]$.

Figure 10.6: contact shear force for different loads and inflow velocities, at measure points as indicated in Figure 10.4.

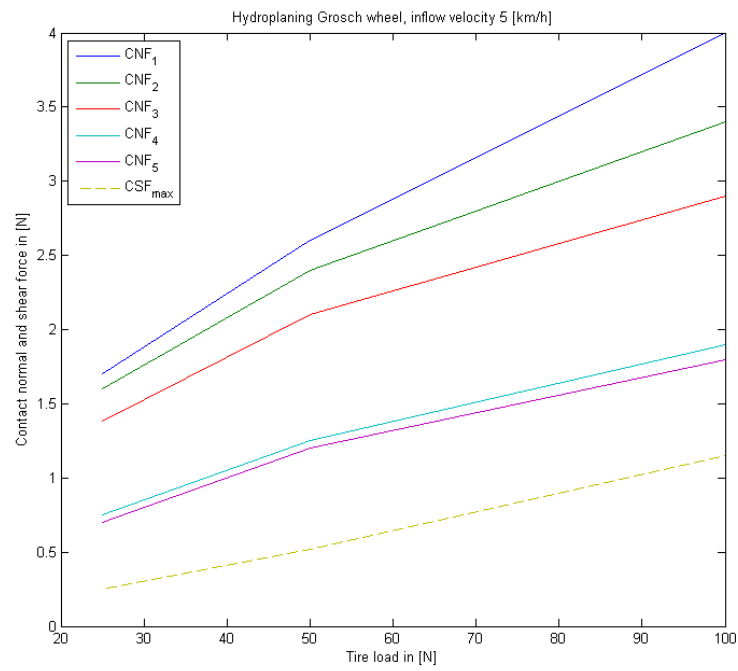


Figure 10.7: contact normal and shear force for the measurement points.

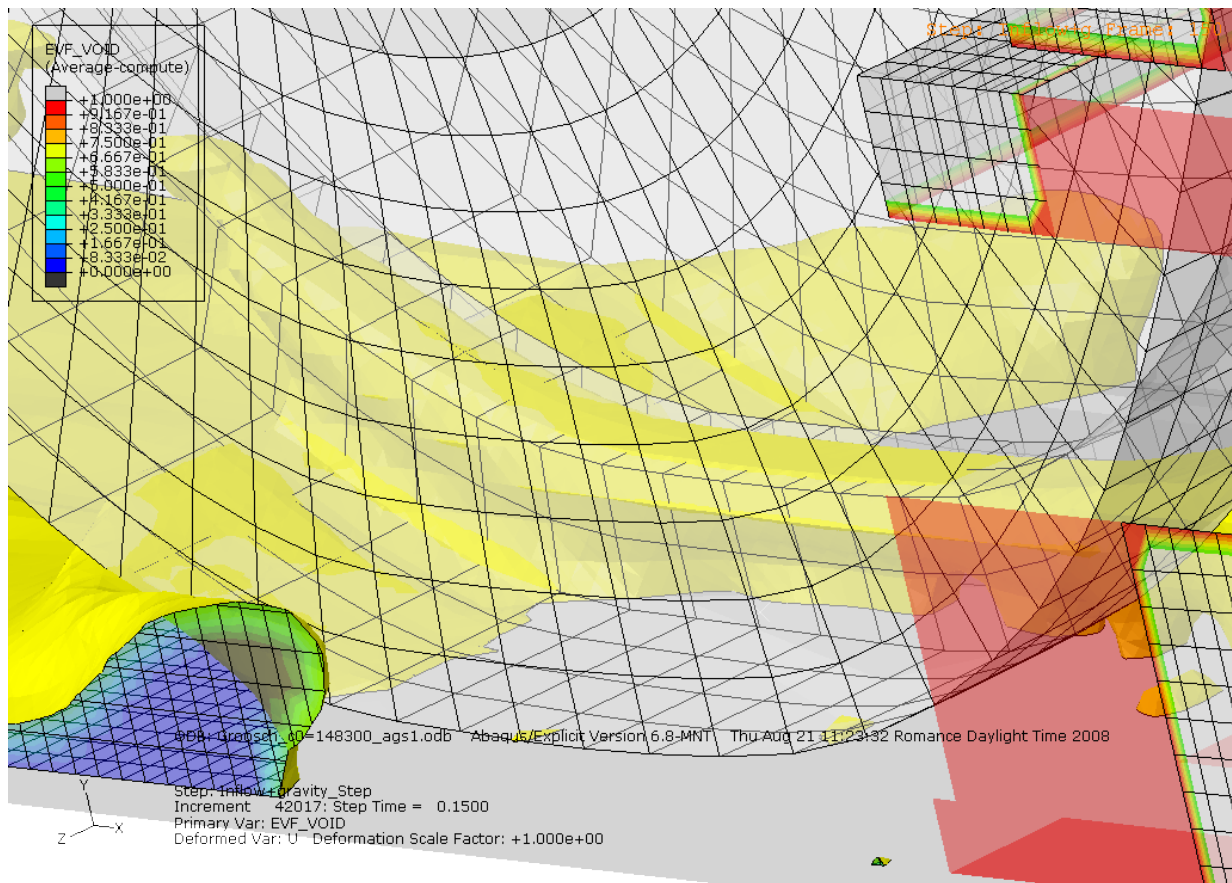


Figure 10.8: leakage as a result $EVF < 0.5$ at interface nodes.

Chapter 11

Conclusions and recommendations

11.1 Introduction

Based on the evaluations made throughout the project, conclusions can be drawn and recommendations can be made concerning Abaqus' CEL method. Both in theoretical and practical sense.

11.2 Conclusions

From the gathered information about the theory behind Abaqus' CEL method, it can be concluded that it is still in a nonmature stage. The mesh treatment is certainly unique, but time consuming, having to remesh from the Lagrangian mesh back to the original Eulerian mesh every time step. This partially explains the long simulation times. The displacement for the fluid is solved instead of the velocity. Also very unique, but disputable. There are no turbulence modelling possibilities. The possibility of choice between the compressible or incompressible form of the fluid equations is missing. The choice of the Mie-Grüneisen EOS is inappropriate for fluids of the incompressible class. The loose partition coupling is a fixed option. Strong partition coupling schemes are not yet implemented. The choice of a PLIC VOF method is a proper choice. Then again, surprisingly the contact threshold is fixed at 0.5, rendering control over leakage in this sense impossible. Merely the choice of a finer mesh offers a solution. Recall that many aspects of Abaqus' CEL method can/could not be reviewed due to company secrecy of Simulia.

From the construction and analysis of the test models, conclusions in practical sense can be drawn. Abaqus/CAE is helpful with the modelling. Compared to input file construction by programming, Abaqus/CAE simplifies constructing a model and gives clear visual representations of the parts, assembly and the forces and BCs the assembly is subjected to. Couette flow and channel flow worked perfectly and the solutions on a structured and unstructured mesh resemble sufficiently. However, when a Lagrangian body is introduced into the Eulerian domain, in combination with contact, the CEL method seems to fail. The solution of the cylinder in a channel flow model is sufficient when the cylinder is modeled by a BC. When it is modeled with a Lagrangian body, the solution is noisy and does not resemble the BC case. Regardless of how the cylinder is modeled, the results should be the same. The same holds for the cube in the channel flow. Comparing the results from [36] and Abaqus, the confidence in the handling of the interaction between fluid and structures decreases further. Obviously, contact is not treated properly because there is no resemblance in the solutions at all. Herewith, the results of the Grosch wheel model are already mistrusted, although they are correct in the qualitative sense. Comparison with the alternative tool should reveal more in the quantitative sense.

Besides conclusions about the fundamentals and the test models, there are some additional issues. The costs for using the multiphysics module of Abaqus are unacceptable. They are typically the double of the costs of a 'regular' analysis. There is even a request for licenses when a data check is requested, while a data check only takes thirty to sixty seconds. In combination with a limited amount of licenses, these license costs are unnecessary and annoyingly increasing work time. The initial versions used turned out to contain a memory problem. The unofficial unstable version 6.8-MNT provided to us to fix this problem, coped with other problems. Simulation analysis would suddenly stop, wasting time. The volume fraction tool would fail every now and then, resulting in strange Eulerian domains filled with fluid. Furthermore, while there are periodic BCs available for Lagrangian domains, they are not for Eulerian domains. For the sake of decreasing calculation times, mesh density variation features should be available. There are plenty of options for this, but mortar elements are missing. An implementation of this feature would be helpful.

Overall, Abaqus' CEL method needs attention before it can be used for a treaded tire hydroplaning simulation. Although the structure part of Abaqus is mature because of years of development and experience, its new fluid part is not. Hydroplaning is a demanding problem with strong interaction between the structure and fluid. Perhaps it is too demanding and specific to be handled by a 'common problem program'. Herewith, we emphasise that every problem is unique and that there is no such thing as a best method for each problem. Implementing as few as possible and handling as much problems as possible, in other words the purpose of commercial tools, is somewhat in contrast with the treatment of demanding problems like hydroplaning.

11.3 Recommendations

Self evidently, it is recommended to wait for the first stable version, namely 6.8.2, avoiding unnecessary problems. To make sure the models are not dependent of the version that was used for the analysis, the models can be rerun with 6.8.2.

If GTC*L is still interested in Abaqus' CEL method, this published report should be brought to the attention of Simulia. Their response is of importance for possible corrections in the models used or in the CEL method. Either way, GTC*L will benefit from this. Besides this investment, a consideration with regard to further cooperation with Simulia in improving the CEL method should be made.

There are several possibilities for further research to improve the CEL method. The benchmarks as described in [22] can be used to test Abaqus. The emphasis of this article is on incompressible fluid flow with structural interactions. The extent of the faulty fluid acceleration as mentioned in 7.5 should be investigated. Its dependence on mesh density and velocity is of interest. The possibility of loss of fluid and to what extent this loss is present are interesting issues. A surface reconstruction test with a coarse mesh can be revealing. An example of such a test is the dropping of a Lagrangian cube of one cell into a fluid tank of $5 \times 5 \times 5$ or slightly more cells. Initial results of this test witness the bouncing of the cube on the fluid surface. In the Grosch wheel test model there are some parameters whereof the influence is yet to be determined. Examples are mesh density and fluid inflow velocity. Also, the placement of the Eulerian grid slightly below the Lagrangian road contact surface to avoid $EVF < 0.5$ nodes is of interest.

Depending on Simulia's response on this report, it is perhaps worth to make another investment. On the other hand, this can also be left to other companies.

If GTC*L is not interested anymore, it is recommended to develop a hydroplaning prediction tool in cooperation with universities. With the development of this tool, resemblances with the discipline tribology should be taken into account. Modelling hydroplaning using the Reynolds equation for textured surfaces, which is used in tribology, is perhaps promising. The tool could be integrable in Goodyear's FEM program. Herewith, all the needed theory will be at hand and not in secrecy. Implementations can therefor be corrected and improved efficiently.

Part III

Appendices

Appendix A

Abaqus

A.1 Introduction

The FEA code in question, Abaqus, is provided by Simulia. The Abaqus software package exists of the preprocessor Abaqus/CAE, implicit and explicit solvers Abaqus/Standard and Abaqus/Explicit respectively and postprocessor Abaqus/Viewer.

For model construction versions 6.7-EF1 and 6.8-MNT were used. Version 6.7-EF1 contained a memory error in the CEL solving module. Details of this error are unknown. An unofficial release 6.8-MNT was provided to GTC*L to fix this issue. Accordingly, the solving was done mainly with Abaqus/Explicit 6.8-MNT, using domain decomposition for multiple processor usage. When due to errors in version 6.8-MNT jobs would fail, 6.8.1 was used alternatively.

A zip package containing all the *.cae and *.inp files to run the models is deposited at [2]. In this way it is possible for everyone in the possession of Abaqus 6.8-MNT or a higher version to run, or even improve, the models themselves for testing the feasibility of Abaqus' CEL method.

A.2 Abaqus/CAE

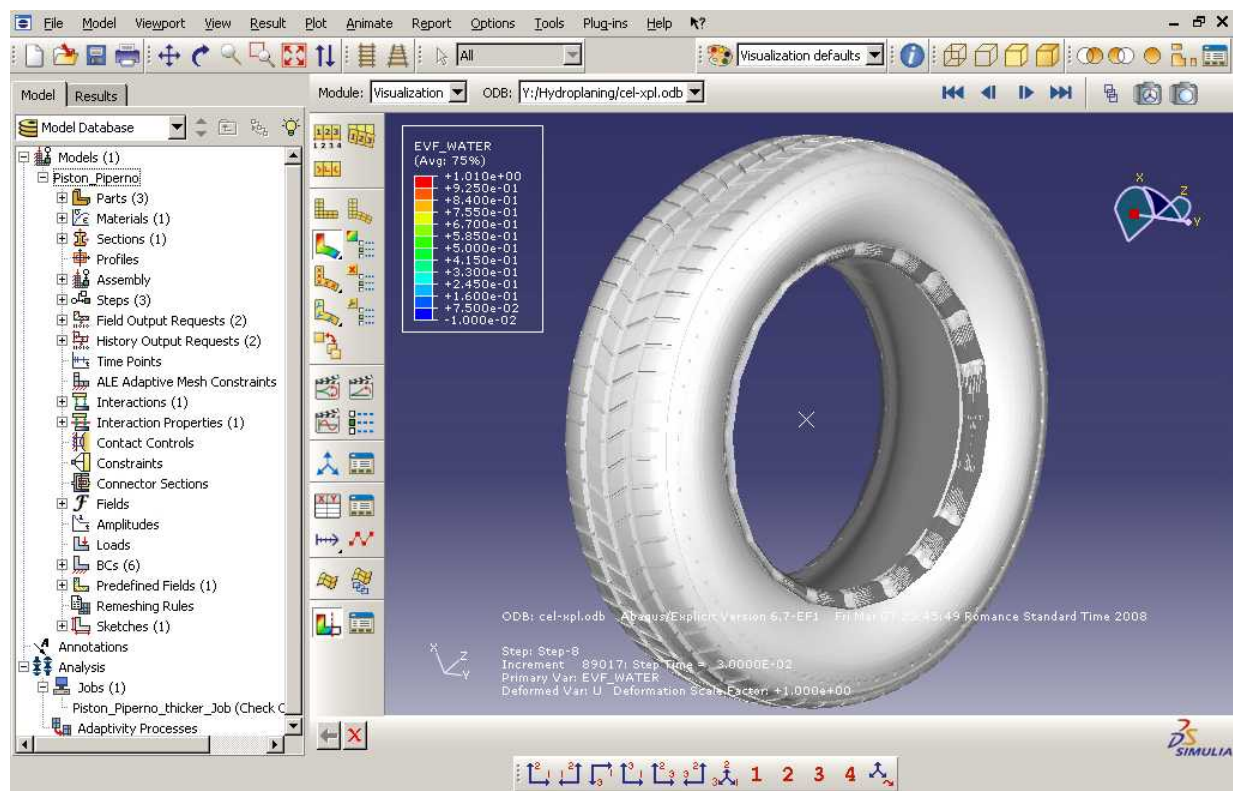


Figure A.1: Abaqus/CAE

CAE stands for computer aided engineering. Abaqus/CAE is the part of the Abaqus software package that, by means of a GUI, simplifies the modelling. In Abaqus/CAE the whole model is defined by a set of features, ordered in a viewport, cf. Figure A.1.

- **Parts.** Choose what kind of part is to be designed. Options are deformable, rigid or Eulerian part. Baseline can be solid, shell, plane, wire of point. After the part is sketched, it can be meshed. If the part is rigid, mass and inertia properties have to be assigned to the reference point (RP) of the part. Sets of elements or nodes can be created to simplify selection later on. Mesh tools are available to seed the part to demands and subsequently mesh it.

- **Materials.** Material properties need to be assigned to the part. Examples are elastic modulus, conductivity, density, viscosity and shear modulus.
- **Sections.** Parts can exist of different materials. By creating multiple sections, multiple section assignments to the part can be realised.
- **Profiles.** Profiles can be assigned to shells, e.g. an I-profile to a beam.
- **Assembly.** In assembly all the parts are instanced to complete the model in sense of space. Constraints of movements can be introduced.
- **Steps.** To create time steps for the analysis. Per time step BCs, field output, history output and predefined fields can be created or modified.
- **Field and history output request.** The quantities of interest can be requested here per set of parts, elements or nodes.
- **Interactions.** Types of interactions can be chosen here. Assign per surface pair of for the whole model.
- **Interaction property.** Tangential and normal contact can be described.
- **Fields.** To make discrete fields to define where the fluid is initially placed and to make analytical functions to define for example a velocity inflow profile.
- **Amplitudes.** Amplitudes for functions.
- **Loads.** Define pressures and stresses.
- **BCs.** Displacement, velocity or acceleration BCs.
- **Predefined field.** To define ICs.
- **Jobs.** Choose amount of processors and single or double precision.

A.3 Abaqus/Viewer

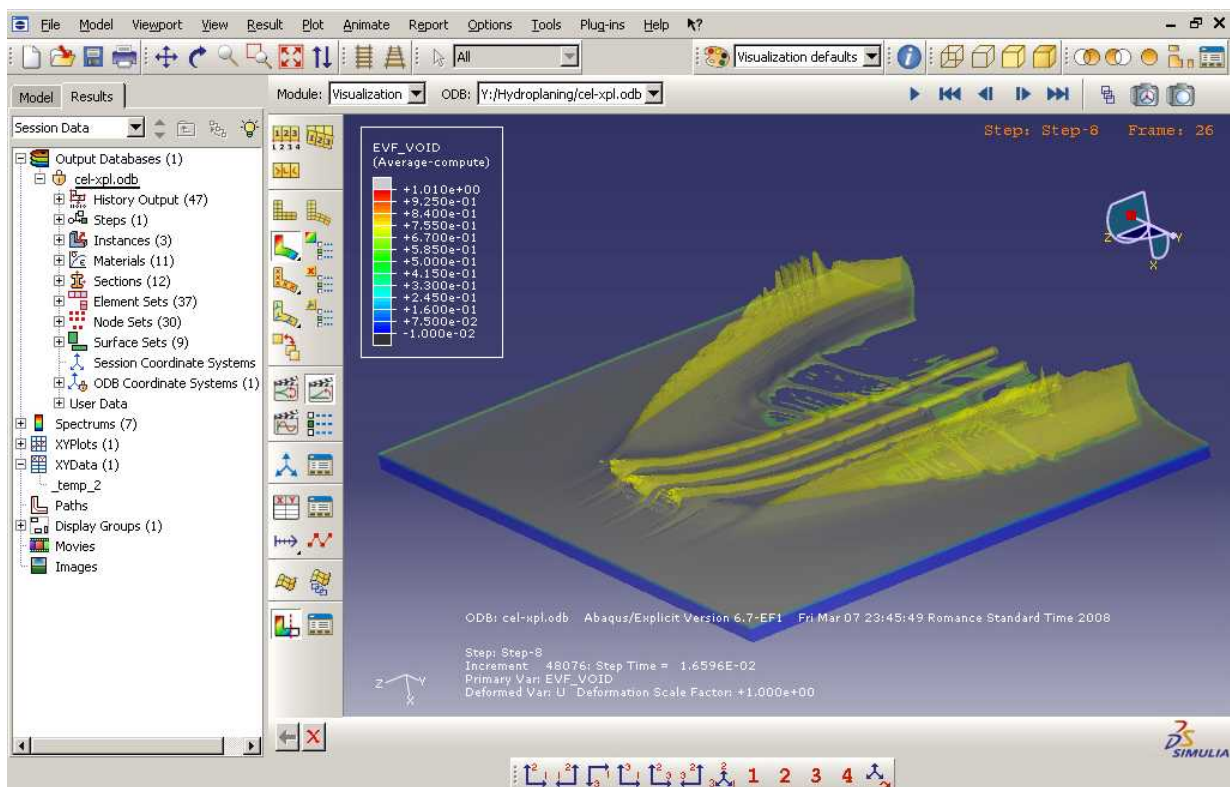


Figure A.2: Abaqus/Viewer

Abaqus/Viewer is the part of the Abaqus software package to view the simulation results created in a database by the analysis modules Abaqus/Standard or Abaqus/Explicit. Field and history output can be visualised with time animations and plots.

Part IV

Bibliography

Bibliography

- [1] *GTC*L printshop*.
- [2] http://ta.twi.tudelft.nl/nw/users/numanal/sillem_eng.html.
- [3] <http://www.compmech.se>.
- [4] <http://www.dunloptires.com>.
- [5] <http://www.wikipedia.com>.
- [6] A.E.P. VELDMAN, *Computational fluid dynamics*. University of Groningen, Groningen, 2006.
- [7] A.H. VAN ZUIJLEN, *Fluid-structure interaction simulations, efficient high order time integration of partitioned systems*, PhD thesis, TU Delft, Delft, 2006.
- [8] T. BELYTSCHKO, W.K. LIU, AND B. MORAN, *Nonlinear finite elements for continua and structures*, Wiley, West Sussex, 2007.
- [9] H. BIJL, A.H. VAN ZUIJLEN, A. DE BOER, AND D.J. RIXEN, *Fluid-structure interaction*. Delft institutes of aerospace and mechanical engineering, Delft, 2007.
- [10] A. DE BOER, A.H. VAN ZUIJLEN, AND H. BIJL, *Review of coupling methods for non-matching meshes*, *Computer Methods in Applied Mechanics and Engineering*, 196 (2006), pp. 1515–1525.
- [11] D.F. PARKER, *Fields, flows and waves*, Springer, London, 2003.
- [12] D.J. BENSON, *Computational methods in Lagrangian and Eulerian hydrocodes*, *Comput. Methods Applied Mechanical Engineering*, 99 (1992), pp. 235–394.
- [13] D.S. STEWART, S. YOO, AND B.L. WESCOTT, *High-order numerical simulation and modelling of the interaction of energetic and inert materials*, *Combustion theory and modelling*, 11 (2007), pp. 305–332.
- [14] R. HABERMAN, *Applied partial differential equations with Fourier series and boundary value problems*, Pearson Prentice Hall, New Jersey, fourth ed., 2004.
- [15] C. HIRT AND B. NICHOLS, *Volume of fluid (VOF) method for the dynamics of free boundaries*, *Journal of Computational Physics*, 39 (1981), pp. 201–225.
- [16] J. DONEA, A. HUERTA, J.P. PONTHOT, AND A. RODRÍGUEZ-FERRAN, *Arbitrary Lagrangian-Eulerian methods*, *Encyclopedia of computational mechanics*, Volume 1: fundamentals (2004).
- [17] J.B. BODEUX, *Thermo-mechanical modeling for temperature and rolling resistance predictions*. GTC*L, Colmar-Berg, 2007.
- [18] J.D. ANDERSON JR., *Computational fluid dynamics, the basics with applications*, McGraw-Hill Inc., Singapore, 1995.
- [19] J.E. PILLIOD JR. AND E.G. PUCKETT, *Second-order accurate volume-of-fluid algorithms for tracking material interfaces*, *Journal of Computational Physics*, 199 (2004), pp. 465–502.
- [20] J.J. KALKER, *Three-dimensional elastic bodies in rolling contact*, Kluwer Academic Publishers, Dordrecht, 1990.
- [21] K.H. BROWN, S.P. BURNS, AND M.A. CHRISTON, *Coupled Eulerian-Lagrangian Methods for Earth Penetrating Weapon applications*, U.S. Department of Commerce, (2002).
- [22] K.J. BATHE AND G.A. LEDEZMA, *Benchmark problems for incompressible fluid flows with structural interactions*, *Computers and structures*, 85 (2007), pp. 628–644.
- [23] M.A. MEYERS, *Dynamic behaviour of materials*, Wiley, New York, 1994.
- [24] M.I. GERRITSMAN, *Computational fluid dynamics incompressible flows*. Delft University Press, Delft, 2002.
- [25] S. PIPERNO, *Simulation numérique de phénomènes d'interaction fluide-structure*, PhD thesis, École nationale des ponts et chaussées, Champs-sur-Marne, 1995.

- [26] G. POST, *Applied functional analysis*, Delft, 2000.
- [27] R.A. ADAMS, *Calculus: a complete course*, Pearson Addison Wesley, Toronto, sixth ed., 2006.
- [28] R.C. HIBBELER, *Mechanics of materials*, Pearson Prentice Hall, Singapore, sixth ed., 2005.
- [29] R.J. LEVEQUE, *Finite volume methods for hyperbolic problems*, Cambridge University Press, Cambridge, 2002.
- [30] A. SEGAL, *Finite element methods for the incompressible navier-stokes equations*. Delft University Press, Delft, 2006.
- [31] SIMULIA, *Hydroplaning model*.
- [32] SIMULIA, *Abaqus version 6.7-EF manual*, USA, 2007.
- [33] ———, *Coupled Eulerian-Lagrangian analysis with Abaqus/Explicit*. Simulia, USA, 2008.
- [34] SIMULIA BENELUX SUPPORT STAFF MEMBER, *G. Kloosterman*.
- [35] S.K. CLARK, *Mechanics of pneumatic tires*, U.S. Government Printing Office, Washington, revised ed., 1975.
- [36] S. TUREK AND J. HRON, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*, Fluid-Structure Interaction: modelling, simulation, optimisation, 53 (2006), pp. 371–385.
- [37] E. VAN GROESEN AND J. MOLENAAR, *Advanced modelling in science*. UT Twente, Enschede, 2007.
- [38] J. VAN KAN, A. SEGAL, AND F. VERMOLEN, *Numerical methods in scientific computing*, VSSD, Delft, 2005.
- [39] T. VON KÁRMÁN, *Aerodynamics*, McGraw-Hill, New York, 1963.
- [40] P. WESSELING, *Elements of computational fluid dynamics*. Delft University Press, Delft, 2002.
- [41] W.J. RIDER AND D.B. KOTHE, *Reconstructing volume tracking*, Journal of Computational Physics, 141 (1998), pp. 112–152.



Straight hydroplaning test [1].