



Swansea University
Prifysgol Abertawe

**A particle-based fluid modelling
and its application to lubricating grease**

Kazuya YAMAMOTO


Zienkiewicz Institute for Modelling, Data and AI
Faculty of Science and Engineering
Swansea University

Submitted to Swansea University in fulfilment of
the requirements for the Degree of *PhD*

September 2024


Declarations

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed..... 


Date..... **23 September 2024**

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed..... 


Date..... **23 September 2024**

I hereby give consent for my thesis, if accepted, to be available for electronic sharing

Signed..... 

Date..... **23 September 2024**

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed..... 

Date..... **23 September 2024**

Abstract

Lubricating grease is used in a wide range of applications, such as gearboxes and ball bearings due to its unique advantages over oil, such as low maintenance and leak resistance. This study is sponsored by NSK, the author's employer and a leading bearing manufacturer in Japan. In their product development process of the most important product, ball bearings, a significant amount of trial and error is required in the experiment. Fast and accurate grease simulation could alleviate the situation, but in the relevant academic field, there have only been a limited number of works on grease simulation. Moreover, those works hardly discuss the difference between greases with different material properties most likely because of the limited reproducibility of grease in simulation while there have been numerous experiment-based studies about the complex material properties of grease and the ball bearing performance. The simulation process is also reported to be time-consuming

Setting the ultimate goal as the utilisation of the simulation in the speedy product development of grease-lubricated ball bearings, this study expands upon the material properties of grease by adding elastoplasticity to previously used strain-rate dependent viscosity, and examines the effectiveness of the widely used cone penetration test and a newly designed plate separation test, approximately representing the responses to compression and tension respectively. The model was found to improve the result in the cone penetration test, but some discrepancy was found in the plate separation test. This study was performed with an improved smoothed particle hydrodynamics called the reproducing kernel particle method, which was implemented from scratch.

Considering the need for efficiency in product development, GPU parallel computing was deployed to accelerate simulation. While there was a significant speed-up compared to a serial CPU equivalent, the overall performance was limited by the time step constraint, an issue that exists regardless of whether a CPU or GPU is used.

Contents

1	Introduction	1
1.1	Material properties of grease	3
1.2	Experimental study of grease in the application	9
1.3	Numerical simulation of grease	11
1.3.1	Elastohydrodynamic Lubrication (EHL) simulation	11
1.3.2	Micro-scale grease simulation	12
1.3.3	Macro-scale grease simulation	13
1.4	Simulation techniques potentially applicable to grease	15
1.5	The goal of this study	18
2	Simulation method and its implementation in this work	21
2.1	Spatial discretisation of particle methods	21
2.1.1	SPH discretisation	25
2.2	Elastoplasticity	27
2.2.1	Benchmark	30
2.3	Fluid-rigid body interaction	37
2.3.1	Benchmark	37
2.4	Reproducing Kernel Particle Method	41
2.4.1	Inconsistency problem in SPH	41
2.4.2	Reproducing Kernel Particle Method(RKPM)	42
2.4.3	Other corrected methods	50
2.5	Ill-distributed particles	52
2.6	Surface cohesion	57
2.7	GPU parallelisation	63
2.7.1	GPU computing overview	63
2.7.2	SPH on the GPU	64
2.7.3	GPU implementation in this study	66
2.7.4	GPU parallelisation benchmark	69
2.8	Overall algorithm	76
2.9	Verification of the Method through Benchmark; lid-driven cavity flow	78
2.10	Conclusions	90
3	Numerical simulation of grease; cone penetration and plate separation tests	92
3.1	Modelling of grease	92
3.2	Cone penetration	100
3.2.1	Introduction	100
3.2.2	Results of simulation	101

3.3	Plate separation	123
3.3.1	Introduction	123
3.3.2	Results of simulation	124
3.4	Conclusions	133
4	Conclusions and future work	135
4.1	Conclusions	135
4.2	Future work	136
	References	137

List of Figures

1.1	Thickener structures of (a) lithium complex soap, (b) polyurea, (c) calcium sulfonate complex soap, (d) bentonite [197]	4
1.2	Quarter-size penetrometer at the beginning(left) and at the end(right) of the test, taken by author at NSK, the ball bearing manufacturer and the author's sponsor	5
1.3	Shear-thinning, where viscosity η decreases as strain rate $\dot{\gamma}$ increases [197]	6
1.4	Grease threads between the railway track and the wheel[61]	8
1.5	The centrifugal force-driven grease bleeding test rig [197]	9
1.6	Different wear patterns by grease[33]	10
1.7	Film shape and pressure distribution at the contact in EHL viewed from the side of the primary flow direction [41]	11
1.8	A tapered-roller bearing [107]	12
1.9	Grease distributions in the gearbox [111]	14
1.10	Grease distribution after settling. Top: CFD, bottom: X-ray CT [130]	14
1.11	Grease distrubution in the double-groove ball bearing with the newly designed cage [81]	15
1.12	Viscoelastic fluids draining from a tank [52]	16
1.13	Filaments of a viscoelastic fluid [198]	17
1.14	Disentanglement of molecular chains [91]	17
1.15	Grease in ball bearings of different sizes before the shield installation (courtesy to the author's colleague)	19
1.16	Grease stretching when scraped with spatula(author's own)	20
1.17	Modes of grease motions in the ball bearing courtesy of the lead author of [130]	20
2.1	Effect of elasticity in 2D	31

2.2	Effect of elasticity in 3D	32
2.3	Particle files used in the elasticity benchmarks	33
2.4	Different elastic moduli in 3D	35
2.5	Particle file used in the elastic moduli benchmark	36
2.6	Dropping heavier and lighter balls into a fluid	38
2.7	Ball displacements over time	39
2.8	Particle file used in the fluid rigid body interaction benchmark	40
2.9	Particle distribution and SPH consistency [97]	42
2.10	TaylorGreen vortices. Left: Particle clustering with Divergence-Free, right: uniform distribution with Divergence-Free and particle shifting [183]	52
2.11	Particle clustering under tension. Left: Density-Invariant, right: Divergence-Free [80]	53
2.12	Separation of fluid. (a) without re-sampling, (b) with re-sampling [59]	55
2.13	Effect of particle shifting in a chronological order	56
2.14	Particle file used in shifting benchmark	57
2.15	Effect of surface cohesion, $E = 0$ in a chronological order . . .	60
2.16	Effect of surface tension, $E = 100$ in a chronological order . .	61
2.17	Particle file used in Figs. 2.15, 2.16	62
2.18	Evolution of single GPU performance in throughput [38] . . .	64
2.19	Speed-up with the GPU from the CPU equivalent	70
2.20	CPU(left) vs GPU(right) for the 2D dam break	71
2.21	CPU(left) vs GPU(right) for the 3D small dam break	72
2.22	3D large dam break with the GPU	73
2.23	Particle file used in the GPU benchmarks	74
2.24	Particle file with velocity condition used in Figs. 2.15, 2.16 . .	79
2.25	Normalised velocities in lid-driven cavity flow, $Re = 1$	80
2.26	Normalised velocities in lid-driven cavity flow, $Re = 10$	81
2.27	Normalised velocities in lid-driven cavity flow, $Re = 50$	82
2.28	Velocities in lid-driven cavity flow, $Re = 100$	83
2.29	Streamline patterns in lid-driven cavity flow, $Re = 1$	84
2.30	Streamline patterns in lid-driven cavity flow, $Re = 10$	85
2.31	Streamline patterns in lid-driven cavity flow, $Re = 50$	86
2.32	Streamline patterns in lid-driven cavity flow, $Re = 100$	87
2.33	Flow curves	89
2.34	Normalised velocities in non-Newtonian lid-driven cavity flow, $Re = 10$	90
3.1	Greases used in this study, stored in the container(author's own)	93
3.2	Stress-strain rate curves with viscosity only. Symbols for data from experiment and lines for fitted curve.	94

3.3	Stress-strain rate curves with elasticity. Symbols for data from experiment and lines for fitted curve.	98
3.4	Industry standard grease worker [157]	101
3.5	Effect of decay rate α on consistency for grease Grease 1 . . .	102
3.6	Effect of decay rate α in half-size penetrometer for grease SHC460 in half-size penetrometer. Top: initial state, bottom: after 5[s]	103
3.7	Grease consistencies without elasticity	103
3.8	Grease consistencies with elasticity	104
3.9	Cone displacements over time without elasticity	104
3.10	Cone displacements over time	105
3.11	Cone speeds over time without elasticity	105
3.12	Cone speeds over time with elasticity	106
3.13	Cone speeds breakdown	107
3.14	Cone speeds breakdown	108
3.15	Grease 4 in half-size penetrometer in a chronological order without elasticity	109
3.16	Grease 2 in half-size penetrometer in a chronological order without elasticity	110
3.17	Grease 1 in half-size penetrometer in a chronological order without elasticity	111
3.18	Grease 3 in half-size penetrometer in a chronological order without elasticity	112
3.19	Grease 4 in half-size penetrometer in a chronological order with elasticity	113
3.20	Grease 1 in half-size penetrometer in a chronological order with elasticity	114
3.21	Grease 1 in half-size penetrometer in a chronological order with elasticity	115
3.22	Grease 3 in half-size penetrometer in a chronological order with elasticity	116
3.23	Elastic component of force exerting on cone	117
3.24	Static pressure component of force exerting on cone	118
3.25	Viscous component of force exerting on cone	119
3.26	Runtime breakdown in cone penetration	120
3.27	Section of particle file used in half-size penetrometer	121
3.28	Break-off lengths in plate separation test	125
3.29	Grease Grease 4 in plate separation in plate separation, beginning, during stretch, break-off moment, after break-off . . .	126
3.30	Grease Grease 2 in plate separation in plate separation, beginning, during stretch, break-off moment, after break-off . . .	127

3.31	Grease Grease 3 in plate separation in plate separation, beginning, during stretch, break-off moment, after break-off . . .	128
3.32	Grease Grease 1 in plate separation, beginning, during stretch, break-off moment, after break-off	129
3.33	Sections of greases coloured with downward/upward acceleration due to elasticity at 0.73[s]. Left: grease1, right: grease 4	131
3.34	Runtime breakdown in plate separation	132
3.35	Particle file used in plate separation	133

List of Tables

2.1	Parameters used in the benchmark of elasticity	34
2.2	Parameters used in the elastic moduli benchmark	36
2.3	Parameters used in the fluid rigid body interaction benchmark	40
2.4	Commonly-used sets of linear functions $\mathbf{P}(\mathbf{x})$	49
2.5	Parameters used in the particle shifting benchmark	57
2.6	Parameters used in the surface cohesion benchmark	59
2.7	Parameters used in the dam break GPU benchmark	75
2.8	Graphics card specs	75
2.9	Parameters used in the lid-driven cavity flow benchmark . . .	79
2.10	Fluid parameters used in non-Newtonian lid-driven cavity flow	89
3.1	Grease parameters with only viscosity	93
3.2	Grease parameters with elasticity	97
3.3	Cone penetration with grease 1	98
3.4	Cut-off length in <i>mm</i> in plate separation with grease 1	99
3.5	Maximum <i>CFL</i> in cone penetration	120
3.6	Runtime and breakdown in cone penetration	121
3.7	Parameters used in in half-size penetrometer. Physics parameters are shown in Table 3.2	122
3.8	Maximum <i>CFL</i> in plate separation	132
3.9	Runtime breakdown in plate separation	132
3.10	Parameters used in in plate separation. Physics parameters are shown in table 3.2	133

Nomenclature

- **English Letters:**

- a : Scale factor in the shape function
- B : Equation of state coefficient
- \mathbf{b} : Product of \mathbf{P} and \mathbf{M}^{-1}
- c_0, c_1 : Constants in a linear function
- C : Spline function
- C_i : Particle concentration
- ∇C : Concentration gradient
- \mathcal{D} : Diffusion coefficients
- dx : Particle size
- Δt : Time step size
- E : Elastic modulus
- f : An arbitrary field function
- f^h : Interpolated value of f
- \mathbf{g} : Gravitational acceleration
- h : Smoothing length or kernel radius of SPH
- i : Particle index
- \mathbf{I} : Identity matrix
- j : Neighbour particle index
- J : Flux on Fick's law
- m : Particle mass
- \mathbf{M} : Matrix involved in the gradient RK shape function
- M_{rigid} : Rigid body mass
- N : Number of neighbour particles
- $N_j(\mathbf{x})$: RKPM shape function
- p : Pressure
- \mathbf{P} : Vector of linearly independent functions
- r : Distance between particles
- $\delta r_{ii'}$: Particle shifting

- \mathbf{u} : Particle velocity
- V : Particle volume
- \mathbf{x} : Particle position
- \mathbf{W} : Anti-symmetric spin tensor

- **Greek Letters:**

- α : Decay rate of plastic deformation
- $\boldsymbol{\epsilon}$: Strain tensor
- $\boldsymbol{\epsilon}'$: Strain deviation tensor
- ϵ_Y : Yield strain
- γ : Yield point
- γ : Surface tension coefficient
- $\dot{\gamma}$: Strain rate
- μ : Viscosity
- Ω : support domain
- $\boldsymbol{\sigma}$: Stress tensor
- σ_Y : Yield stress
- ρ : Density
- ζ : Volume viscosity

Acknowledgements

I would like to thank my supervisor Chenfeng Li who has greatly helped me throughout my PhD. At the initial stage of the study, he instructed me on skill development for my later research. Over the years, he directed my research through regular meetings.

I appreciate my PhD colleagues who were eager to have technical discussions whenever I needed. Ashutosh Bhokare and Sizeng Yu in particular, taught me and answered all my questions on programming languages and fluid dynamics. My colleagues were helpful not just academically, but also psychologically. All of them were understanding towards me, sharing the same worries and stress as me as a PhD student. They made the lonely and individual study so much more tolerable.

I would like to show gratitude towards Xiaosong Cheng, a masters student at Tsinghua University as my fellow researcher. Being impressibly knowledgeable in both physics and computing, he was willing to have detailed discussions on my research topic. My deep understanding in this field and research progress would not have been possible without him.

I thank Swansea University, who offered such a great environment for study with the freedom to participate in whatever courses. The location of Bay Campus, combined with the well-equipped office, helped me focus on my study whilst occasionally relaxing in the nature.

I am extremely grateful to NSK Ltd., my employer who has offered me the full financial support on the course and expenses for the entirety of my life in the UK, also allowing my own control over everything. Their support made possible not only to pursue my PhD study, but also to immerse myself in a completely new environment that would have been otherwise impossible. Moreover, a few of my colleagues kindly did experiment remotely from Japan, which was absolutely necessary for my research.

Last but not least, I cannot thank enough my dear friends in the UK. Despite meeting only a few years ago, they truly care about me, and made my life in the UK rich and meaningful. I occasionally had difficult times due to the PhD stress, COVID19 and so on, when they listened to me, accompanied me, and offered me help. They have, and will continue to make a positive difference to my life.

1 Introduction

Lubricating grease is defined as a solid to semi-fluid product or dispersion of a thickening agent in a liquid lubricant. Other ingredients imparting special properties may also be included [103].

Grease has a wide range of applications, one of which is lubricant in die casting for better quality of casting [143]. Grease is also used in open gear of mining equipment [145], rolling linear guide [171], and spindle [90], [181]. In addition, grease lubrication is used in many components of the car, such as door hinges, constant-velocity joints, turbochargers to cite a few [108]. Electric contacts can be lubricated with grease as protection from factors that could increase the electric resistance [32]. Grease is also used in various types of bearings, such as journal bearings [74], needle roller bearings [98], spherical roller bearings [101], ball bearings [130], and so on.

Lubricating grease is an important research topic in terms of sustainability considering its wide usage. [66] estimates that 23% of the total energy consumption worldwide comes from tribological contacts, 87% of which to overcome friction, the rest for replacing parts due to wear and wear-related failures. Furthermore, when it comes to roller bearings, 80 to 90 per cent of them are lubricated with grease [104]. Hence it is vital to understand the grease dynamics and design bearings effectively.

The wide use of grease as a lubricant is due to several advantages over oil, another commonly used lubricant. In the case of bearings, grease is easier to use as it does not easily leak, and it can also work as a corrosion inhibitor [103]. Another upside of grease is its better protection against contaminants and lower friction when used between the seal, and the shaft or bearing ring [11].

Various studies of grease applications can be found for different interests. Those works include the life of railway tracks and wheels [61], power loss and life of the gearbox [111], [112]. As for rolling bearings, there are studies on noise [83], lower torque [33], [34], [37], [79], [98], and life [23], [84], [104], [170]. Condition monitoring of ball bearings is also studied [162]. Moreover, certain applications have specific requirements, which include space bearings [21], [22] and bearings in electric vehicles [148].

Due to those various applications, grease is expected to satisfy certain requirements that are unique to each application. One is the protection of mechanical parts against contaminant [11]. In some conditions, grease causes undesirable noise and vibration in the bearing, and they need to be minimised [104], [192]. In machinery for food production, the contact of grease and food is occasionally inevitable, and therefore the grease is required to be food-grade [117], [136]. During the operation of certain machinery, a large

amount of grease can spill into the environment, so there is a demand for biodegradable grease to minimise the ecological impact [47].

Because of the wide range of applications and requirements, there are various kinds of greases with different compositions. The main components of grease are base oil and thickener. Base oil can be either mineral oil or synthetic oil. Amongst mineral oils, naphthenic oils have better pumpability at low temperatures, whereas paraffinic oils enable higher soap contents, which yield better mechanical stability [144]. Despite the limited use due to the high prices [40], synthetic oils possess certain advantages over mineral oils. For example, Polyalphaolefin (PAO) and ester oils are said to be suitable for low-temperature conditions thanks to the lower start-up torques [37]. Perfluoropolyether (PFPE) oils are suitable for high-temperature usages as they exhibit little chemical reaction [83].

As was the case with base oils, there is more than one type of thickeners, such as soap, polyurea, and Polytetrafluoroethylene (PTFE). Calcium sulfonate thickeners can be used at temperatures up to 120°C, and are often used in the car industry [158], [43]. Lithium soap and polyurea exhibit good mechanical stability and softening resistance at high temperatures [43]. PTFE can be used at even higher temperatures up to 250 to 270°. PTFE-thickened PFPE greases can work in the presence of aggressive chemicals, such as oxygen or chlorine [104], [108].

In addition to base oil and thickener, additives are normally added to grease to enhance the lubricating performance. Additives can improve wear resistance and lower friction [24]. It also works as a corrosion inhibitor by forming protective films on the metal surface [67]. Some additives are used as antioxidants to prolong grease life [46].

To illuminate the grease lubrication mechanism, there have been experimental studies on the material properties of grease and its applications as will be discussed later. At the same time, simulation is becoming increasingly relevant thanks to the observability of internal mechanical parts [124], [130], time- and cost-reductions in industry [112], and the limitations of experimental conditions [91], along with the progress in computing efficacy combined with the reduced costs of Computational Fluid Dynamics (CFD) software packages [131].

A state-of-the-art review of the studies on grease follows to grasp the methods and findings made so far along with the room for further development in this field.

1.1 Material properties of grease

Tribological properties of grease, i.e. all the behaviours of grease under contacts or friction, such as the mechanical response and chemical degradation, are attributed to the thickener dispersed in the base oil that forms a microstructure [192], which gives grease its unique properties as explained later in this chapter. Studies are found about the observation of such structures. [197] used Atomic Force Microscopy (AFM), and found different microstructures of thickener according to the thickener types (Fig 1.1).

Whilst AFM allows observing thickeners with the presence of base oil, Scanning Electron Microscopy (SEM) is also used to complement AFM. SEM offers a better resolution with detailed thickener skeletons. [39] used SEM to observe the thickener structure of lithium grease, and found the difference in the structure according to the stage of the manufacturing process even with the same composition.

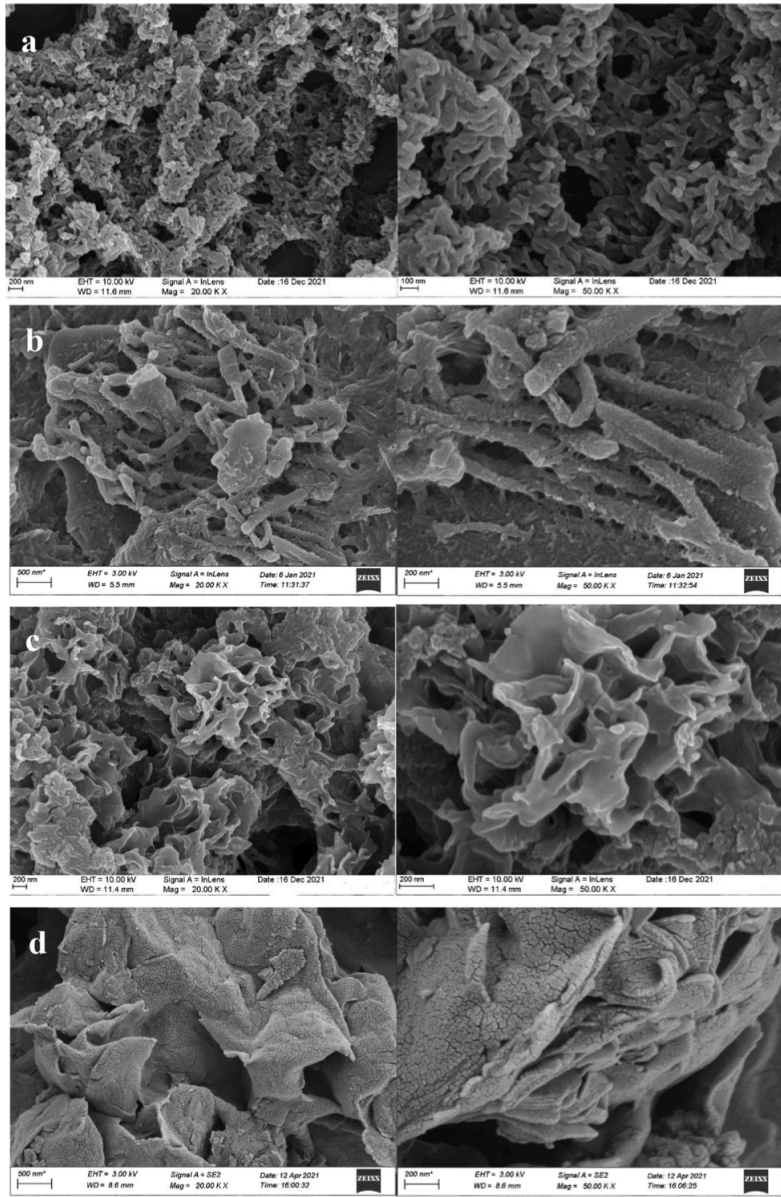


Figure 1.1: Thickener structures of (a) lithium complex soap, (b) polyurea, (c) calcium sulfonate complex soap, (d) bentonite [197]

[192] studied the difference in the thickener network structure by base oil using the Field-Emission Scanning Electron Microscope (FE-SEM), and also analysed its role in thixotropy. [197] investigated the effect of thickener structure on the tribological properties, finding that calcium sulfonate complex soap has a dense porous structure, which produces a strong film

against wear. The microstructure of the thickener gives grease its unique macroscopic material properties as detailed below.

One of the material properties of grease is consistency. Consistency refers to firmness, and it is used as an indicator of the suitable usage of a grease [58]. Consistency is measured in a cone penetration test that is standardised by The National Lubricating Grease Institute (NLGI) [8]. Each grease is given an NLGI consistency number according to the penetration depth, which is widely used in grease specifications. Half-size and quarter-size equipment is also used for easier operation [157] (Fig. 1.2).

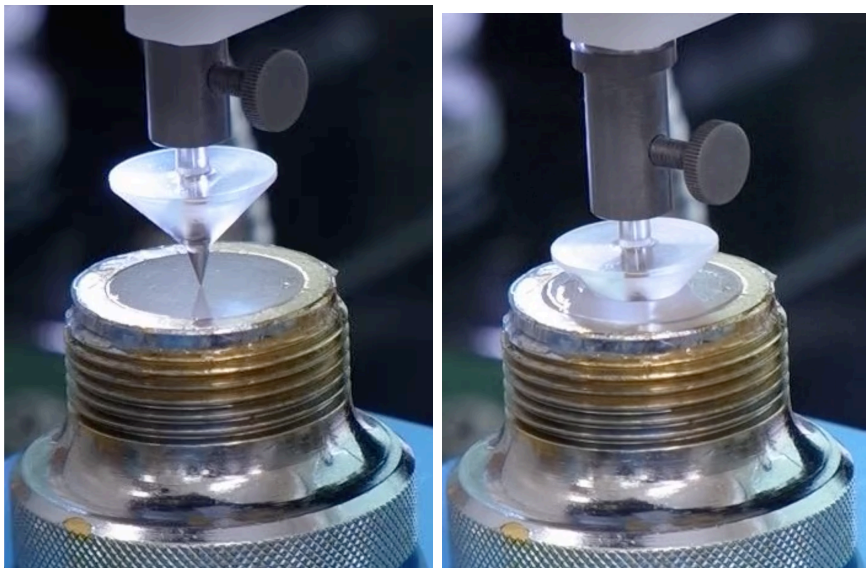


Figure 1.2: Quarter-size penetrometer at the beginning(left) and at the end(right) of the test, taken by author at NSK, the ball bearing manufacturer and the author’s sponsor

Shear-thinning is another material property of grease, which means the decreasing viscosity with increasing shear rates (Fig. 1.3). Shear-thinning is caused by the breakdown of the thickener structure under high shear rates [197]. At low shear rates, grease behaves like a solid [103], [130]. This solid-like behaviour prevents grease from leaking outside the bearing, and helps maintain the lubricating function [148].

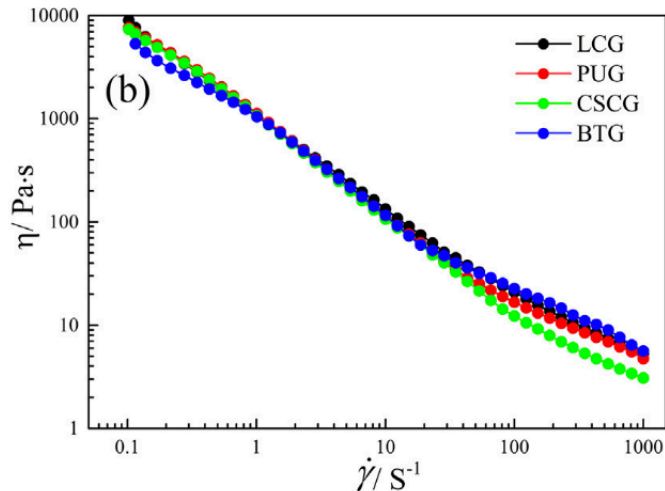


Figure 1.3: Shear-thinning,
where viscosity η decreases as strain rate $\dot{\gamma}$ increases [197]

Shear-thinning has been modelled in studies although it is mostly empirical relationships between the strain rate $\dot{\gamma}$ and viscosity μ . One such option that has been used for grease is the Carreau-Yasuda model[177] expressed as

$$\mu = \mu_{\infty} \frac{\mu_0 - \mu_{\infty}}{[1 + (K\dot{\gamma})^2]^{\frac{n-1}{2}}} \quad (1.1)$$

where μ_{∞} is the viscosity at high-strain rate, μ_0 the viscosity at low-strain rate, K and n constants. The viscosity in this model asymptotically approaches the high- and low-strain plateaus μ_{∞} and μ_0 , which agrees with the measured viscosities in some experiments. The modified Herchel-Berkley model, which avoids the zero division at zero strain unlike the original, is another option for shear-thinning, which reads

$$\mu = \frac{\tau_0}{\dot{\gamma}} (1 - e^{-m\dot{\gamma}}) + K\dot{\gamma}^{n-1} \quad (1.2)$$

where τ_0 is the yield stress, and m a constant. This model accounts for the yield stress with a numerical smoothing around the zero strain.

Grease also exhibits thixotropy [104], [187], which is defined as the reduction of viscosity due to the destruction of the inner structure under shear over time, and the recovery to the original state at rest. The recovery is also referred to as antithixotropy [153]. Thixotropy makes it virtually impossible for rheological measurements to reach a steady state, and therefore could be an issue in measurement protocols [104]. One way to evaluate thixotropy is

the use of a hysteresis loop of the stress-shear rate curve [192]. Another is the measurement of viscosity under constant shear rates [153].

There are models for thixotropy based on the thickener network integrity. It decreases under shear, and recovers under rest, and subsequently determines the viscosity. One of the simplest is [159], where firstly the network integrity λ is transported as

$$\frac{\partial \lambda}{\partial t} + u_i \frac{\partial \lambda}{\partial x_i} = -k_1 \dot{\gamma} \lambda + k_2 (1 - \lambda) \quad (1.3)$$

where k_1 and k_2 are the network destruction and recovery rates, respectively. $\lambda = 1$ and $\lambda = 0$ correspond to the complete integrity and no integrity, respectively. The network integrity influences the viscosity as

$$\mu = \mu_\infty (1 + \alpha \lambda) \quad (1.4)$$

where α is a constant. μ_∞ is the infinity-shear viscosity, and the zero-shear viscosity is $\mu_\infty (1 + \alpha)$. Note that this model also accounts for shear-thinning implicitly. When the fluid is sheared at a constant strain rate with the network integrity complete at the beginning, its destruction progresses gradually according to eq.(1.3), hence gradual changes in viscosity. This is a major difference to the shear thinning models (eqs.(1.1) and (1.2)). There are also different models for network integrity development and its impact on viscosity [89]. While those models, which were developed for polymeric fluids, share the fundamental mechanism with greases, the model verifications are rather limited, and their application to greases would require a thorough investigation.

Viscoelasticity is another material property of grease, and one of its sub-aspects is yield stress, which is relevant to the start-up torque [37]. Yield stress also relates to the choice of grease pumps, and prevention of leakage [191]. Despite its importance, there has not yet been a consensus on the definition and measurement of yield stress with different authors proposing numerous methods. [37] obtained the stress-strain curve from oscillatory measurements of the rheometer. Since there was no clear transition from solid to liquid, the authors defined yield stress as the stress where the linear relation between stress and strain deviates by a predefined arbitrary value. [144] applied various constant shear rates and measured the stress, obtaining stress-shear rate curves, and defined yield stress as the stress at an arbitrary shear rate. [150] gave yet another definition using storage modulus G' and loss modulus G'' , where yield stress was defined as the stress when the solid regime shifts to the fluid regime switch, G'' exceeds G' .

Tackiness is defined as the ability to form threads [2], and is another

important material property of grease. In the railway industry, tackiness corresponds to the ability to carry the grease between the wheels and the railway track[61] (Fig. 1.4). The study used an approach-retraction type experiment, and during the retraction, measured the work required and the amount of the grease picked up. The measurement of tackiness is yet to be standardised, and as an alternative method, [50] used a point contact with a copper ball that reportedly yields better repeatability.



Figure 1.4: Grease threads between the railway track and the wheel[61]

Bleeding or oil separation ability is a material property whereby base oil is slowly released from the bulk grease during the operation [13], [43]. On the one hand, bleeding plays an important role in lubrication of supplying oil to the rolling track of bearings [12]. On the other hand, excessive bleeding shortens the grease life and therefore is considered undesirable [197]. As there are multiple factors relevant to bleeding, such as mechanical stress [197], temperature [43], and centripetal forces [12], more than one method has been developed to quantify grease bleeding correspondingly. Oil separation is tested in a static condition or under a centrifugal force [70], or with a dead weight that presses base oil out of the grease [12].

The bleeding phenomenon has been modelled using Darcy's law, which is the pressure-driven flow through a porous media. According to the law, the flow rate Q past the cross-section area A is expressed as

$$\frac{Q}{A} = \frac{k}{\mu} \frac{\Delta p}{L} \quad (1.5)$$

where k is the permeability constant, Δp the pressure difference, and L the length in the flow direction. In this model, the base oil is assumed to flow inside the static thickener network. In the case of [12], the focus was bleeding under the centrifugal force(Fig. 1.5), which creates a pressure difference as

$$\nabla p = \frac{\omega^2 \rho}{h} \int_{R_0-h}^{R_0} R dR \quad (1.6)$$

where ω is the rotational speed, R_0 the outer radius of the test rig, h the grease fill level, and ρ the base oil density. Combined with eq.(1.5), the oil bleeding rate reads

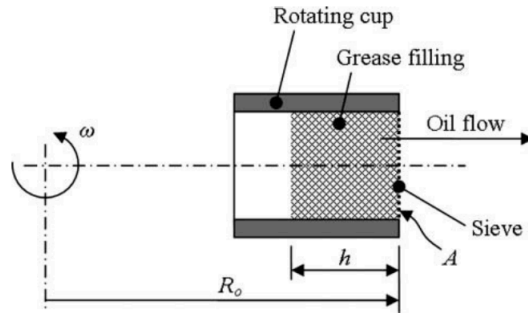


Figure 1.5: The centrifugal force-driven grease bleeding test rig [197]

$$\frac{dV}{dt} = -Q = -A \frac{k}{\mu} \nabla p \quad (1.7)$$

[196] and [65] also used Darcy's law to analyse the oil bleeding from the bulk grease to the blotting paper underneath driven by gravity.

Finally, the dropping point is defined as the temperature where the base oil starts to drop from the bulk grease [88]. It is used to estimate the maximum possible operation temperature [158]. Standardised methods exist, such as ASTM D566 - IP 132 - DIN 51801. Despite the term being different to bleeding, dropping point refers to the same phenomenon that the base oil separates from the bulk grease. "Bleeding" is used when the focus is the amount of bled oil, and "dropping point" is used when the focus is the temperature where the separation starts.

1.2 Experimental study of grease in the application

Various studies can be found on the dynamics of grease in the applications and their resultant performance. Of these, this subchapter will cover some of the main applications, i.e., gearboxes and bearings with a range of performance aspects, such as wear, torque, and life.

[180] investigated the friction of plastic wormgears in the gearbox with a particular focus on the base oil's influence on the gearbox torque. [152] studied the effect of grease consistency, temperature, and fill level on the lubricant supply in the gearbox. [33] investigated the influence of base oil type on the wear and friction torque of the thrust ball bearing. The authors found that some base oils generate more wear particles, hence harsher wear (Fig. 1.6). [84] analysed the relation between thickener types and the service life of the ball bearing, along with the observation of grease distribution after the life tests. [83] conducted a noise test on the bearing with greases of different base oils, and concluded that certain combinations of base oil and thickener yield lower noise. [23] shed light on the failure mechanism of ball bearings by examining the grease distribution and the condition. [86] investigated how different contaminant particles affect the life of the ball bearing by evaluating vibration and subsequent stereoscopic observation.

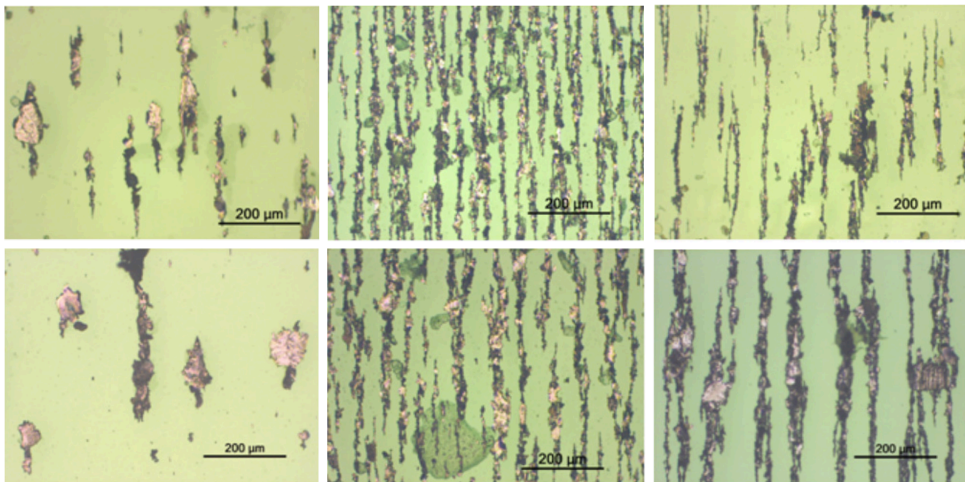


Figure 1.6: Different wear patterns by grease[33]

Whilst these studies effectively evaluate the performance of the actual machinery, other studies deployed apparatus specifically designed to more deeply understand the dynamics of grease in the applications. To analyse the grease flow at the restrictions of the bearing, such as seals, [92] employed Micro Particle Image Velocimetry (μ PIV) to obtain the velocity field with a geometry model that was simplified from the actual bearing. [57] also used μ PIV to observe the velocity field of the grease in a double restriction sealing with a specially crafted transparent bearing housing. [107] designed a test rig where PIV was applied to the raceway of the actual tapered roller bearing, and the velocity field was analysed for each velocity component. [129],

[130] fabricated a ball bearing made of carbon fibre and glass combined with X-ray Computed Tomography (CT), and observed the grease flow inside the bearing. [79] also utilised X-ray CT to observe the grease distribution in the ball bearing to facilitate the cage design optimisation for lower torque. Despite their effectiveness, experimental approaches could be subject to the limited analyses that associate input parameters with output performance along with qualitative discussion. Detailed analyses sometimes require individually designed test environment, which could be expensive. To overcome these difficulties, some authors adopted the approach based on numerical simulation as follows.

1.3 Numerical simulation of grease

This chapter will discuss the studies of grease simulation under three categories for convenience, namely Elastohydrodynamic Lubrication (EHL), micro-scale simulation, and macro-scale simulation.

1.3.1 Elastohydrodynamic Lubrication (EHL) simulation

Elastohydrodynamic Lubrication (EHL) is defined as the lubrication where a high pressure occurs at the contact point, which leads to a significant elastic deformation of the surfaces, eventually forming a lubricant film [105]. This film is significant because it works as a protector at the contact points of gear teeth and bearings [41] (Fig. 1.7), and the simulation of EHL has been performed by researchers.

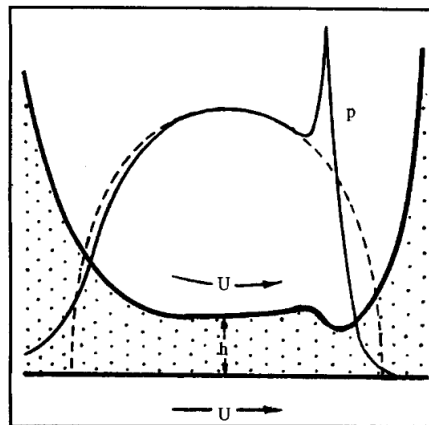


Figure 1.7: Film shape and pressure distribution at the contact in EHL viewed from the side of the primary flow direction [41]

[10] modelled grease using shear-rate dependent viscosity, and studied the influence of load, speed and rheological index or the power constant in the modified Herschel-Bulkley model on the film thickness and pressure spike in the line contact of roller bearings. [69] used shear rate-dependent viscosity and yield stress to investigate the influence of surface texture on the coefficient of friction in the ring-on-disc test. [170] studied the effect of surface morphology in tapered rolling bearings(Fig 1.8), and suggested that there are surface roughness values that can keep full film lubrication. [15] studied the friction forces at the seals in pneumatic spool valves. The authors investigated more than one viscosity model, and concluded that the Newtonian model also yielded similar results to non-Newtonian ones. Using the neural network, [195] studied film thickness when it is comparable to surface roughness, and found that the thermal effect plays a key role under high speeds.

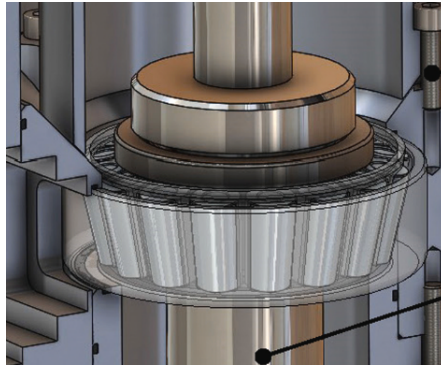


Figure 1.8: A tapered-roller bearing [107]

1.3.2 Micro-scale grease simulation

This subchapter will review the micro-scale simulations, i.e. the simulations that involve the explicit modelling of grease constituents, base oil and thickener. [21] employed Discrete Element Model (DEM) to investigate the interaction between base oil, thickener, and the solid surface. Under the tribometer-like condition, i.e. the condition where grease is subject to a load and friction, the thickener concentration was simulated with different values of phase interaction parameters to facilitate the formulation of better grease. [22] applied a similar technique to investigate the torque peaks of space bearings.

[174] used Molecular Dynamics (MD) to simulate the thermal degradation of silicone grease. The authors investigated the effect of temperature,

and oxygen content, which was found to affect the amount of the thermal degradation products, namely small gaseous molecules and large-molecule products.

Amongst the simulation approaches based on discrete models, such as MD and DEM, the link between the microstructure and the mechanical properties of grease has not been investigated despite the commonly believed theories or experimental attempts.

1.3.3 Macro-scale grease simulation

In mechanical components such as ball bearings and gearboxes, the need for efficient lubrication is paramount. Lubricants must be supplied at points of mechanical contact to mitigate wear and noise. However, an excess of lubricant can cause resistance, leading to unnecessary power consumption.

Computational Fluid Dynamics (CFD) of grease can address this issue in two ways. First, it can provide a detailed analysis of lubricant flow, which can inform product design. Second, it can reduce the need for trial and error in experiments, thereby accelerating product development. Macro-scale grease simulation, in particular, is more focused on these objectives than micro-scale grease simulation.

The bearing manufacturer NSK, the author's employer, aims to optimise the grease injection process, cage geometries, and the material properties of grease, all with the assistance of CFD. The importance of grease CFD is further underscored by the organisations involved in the studies mentioned below. For instance, [123] is from the Japan Aerospace Exploration Agency (JAXA), while [130] and [81] are from NSK and NTN, respectively, both of which are among the largest players in the Japanese ball bearing industry.

To illustrate the state-of-the-art grease simulation, this chapter will review some of the single-phase simulation and all the multi-phase simulation or simulation that handles free surfaces. This is because the unique characteristics of grease only become clear with free surfaces. The somewhat limited number of works regarding the multi-phase simulation is likely due to the challenges associated with simulating grease behaviour.

[111], [112] simulated the grease flow in the gearbox using Finite Volume Method (FVM) along with Volume Of Fluid (VOF), a method for multiphase fluids with sharp interfaces, whilst modelling grease with the shear-rate dependent viscosity. The authors observed the grease flow at different grease fill levels (Fig 1.9), and also calculated the power loss caused by grease flow.

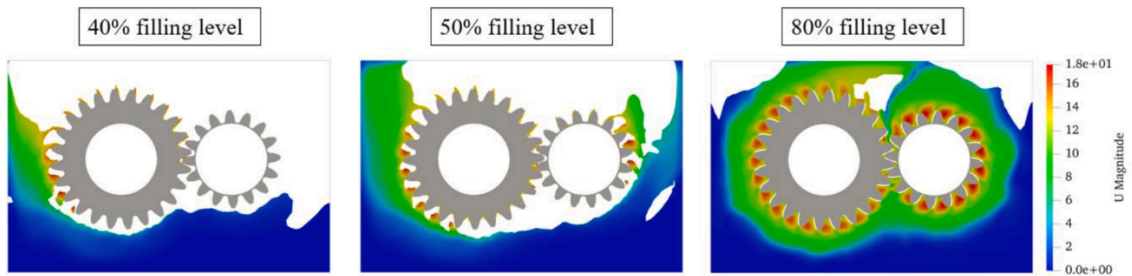


Figure 1.9: Grease distributions in the gearbox [111]

[113] simulated the flow rate and leakage flow of the grease in an air-operated piston pump with a single-phase flow model. The authors nonetheless point out the issue that single-phase modelling did not replicate the cavitation caused by the pump suction. [129], [130] numerically studied the grease flow in a ball bearing using FVM along with VOF whilst modelling grease with the shear-rate dependent viscosity, where grease was initially stirred, and later settled to a stable state. The results were validated with the experiment using X-ray CT imaging (Fig. 1.10)

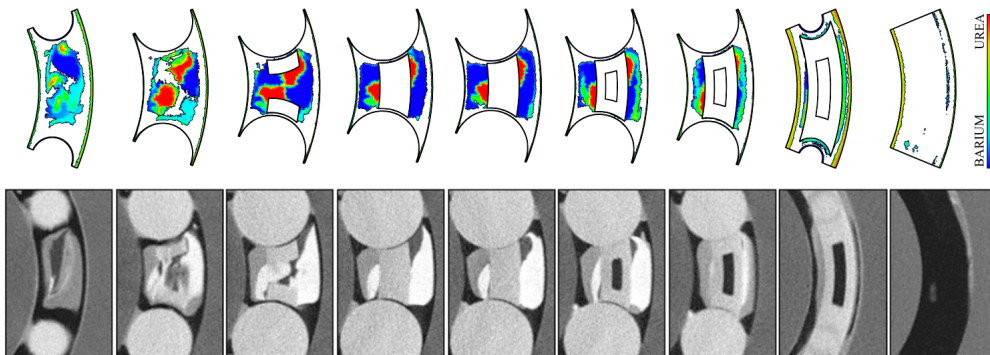


Figure 1.10: Grease distribution after settling. Top: CFD, bottom: X-ray CT [130]

[123] simulated a grease dam break as a benchmark for macro-scale flows using Moving Particle Simulation (MPS), a particle method based on the SPH combined with non-Newtonian viscosity. [124] used MPS for grease in the ball bearing favouring the particle method's suitability for complex geometry and free surface flow, and identified high-shear regions and poorly-lubricated regions. In the ball bearing, grease leakage is occasionally a problem as it shortens the product life and contaminates the environment. [81]

developed a new cage for the ball bearing to reduce grease leakage from the bearing. During the product designing process, they conducted CFD to examine the migration of grease, and stated that grease attached to the balls eventually migrated to the edge of the cage as was the case with the experiment, and that the newly developed design indeed resulted in reduced leakage. Whilst this study illustrates the interest of applying grease CFD to the issues of the product, the details of CFD except VOF being employed and analyses apart from the above results are not mentioned.

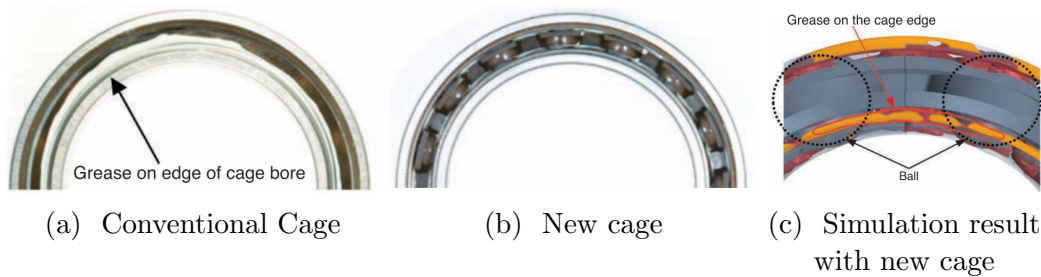


Figure 1.11: Grease distribution in the double-groove ball bearing with the newly designed cage [81]

Whilst all these studies of macro-scale grease simulation shed light on the grease dynamics in the mechanical parts, they did not address the differences between various types of grease. In reality, material properties differ significantly between greases, which in turn makes a difference to the product performance, in the case of the ball bearing, such as torque and life. The experimental studies discussed in the previous subchapter also reflect the interest in this topic. Moreover, the studies of grease simulation mentioned here consider the shear-rate dependency of grease viscosity as one of the key material properties. However, they do not include other material properties, such as thixotropy, elasticity or bleeding despite researchers' interest as discussed in the previous subchapter.

1.4 Simulation techniques potentially applicable to grease

Despite the overall rigorous work by researchers, the grease simulation has not yet explored certain aspects that have been experimentally suggested as important. In the micro-scale approach regarding grease components, such as MD and DEM, it is the link between the thickener microstructure and macroscopic material properties. In the macro-scale simulation, it is the material properties, such as thixotropy, elasticity or bleeding. Nevertheless,

there are studies of other materials than grease, such as jelly, cream and concrete, where the above aspects are included in the simulation. This sub-chapter will review those studies as potentially applicable methods that could further increase the realism of grease simulation.

Tackiness has not yet been handled in grease simulation, but there are examples in different areas. [52], [53] used VOF, where the elastic force was embedded in the Navier-Stokes equation, and the strain was handled in another convection equation. As a result, various flow patterns were replicated with thread-forming observed in some cases (Fig. 1.12).

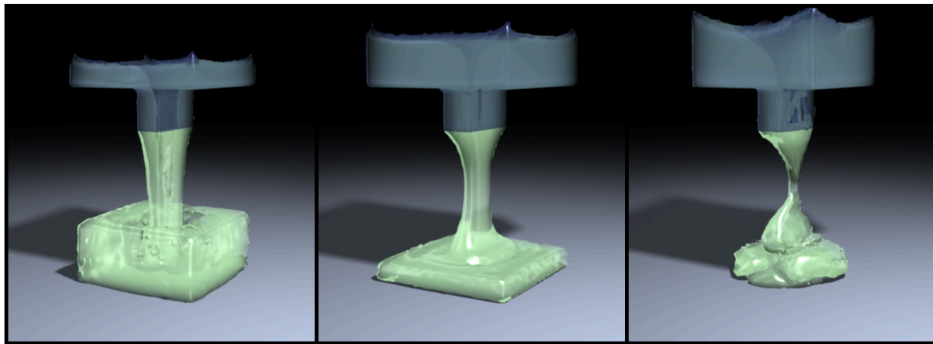


Figure 1.12: Viscoelastic fluids draining from a tank [52]

[59] also embedded strain in the Navier-Stokes equation, using Smoothed Particle Hydrodynamics (SPH). Combined with the newly developed particle re-sampling method, the authors also reproduced the characteristic stretching behaviour of viscoelastic fluids. [7] and [72] simulated the tackiness of the fluid by combining MPS, viscoelasticity and surface tension. [198] used particles connected into simplicial complexes to simulate viscoelastic fluids with an example of filaments of cheese akin to the grease tackiness (Fig. 1.13). [14] simulated viscoelastic materials using Position-Based Dynamics (PBD), and even realised a real-time interactive application under certain conditions.



Figure 1.13: Filaments of a viscoelastic fluid [198]

Thixotropy is also handled in certain simulation. Using a modified Barkhudarov's viscosity model, [188] simulated the gel flow around the orifice, and obtained a more natural viscosity field without viscosity peaks as opposed to shear-thinning models, such as Carreau-Yasuda and Power law.

All these examples are based on macro-scale homogeneous fluid models, but micro-scale discrete modelling of fluid also exists. [91] applied MD to crude oil to study the mechanism of thixotropy. As shear is applied, the oil molecular chains get disentangled and extend in the shear direction, resulting in the reduction of shear stress (Fig. 1.14). This observation agrees with the widely believed mechanisms of grease thixotropy and shear-thinning. [118] applied MD to polyethylene melt, and reproduced the shear-thinning behaviour.

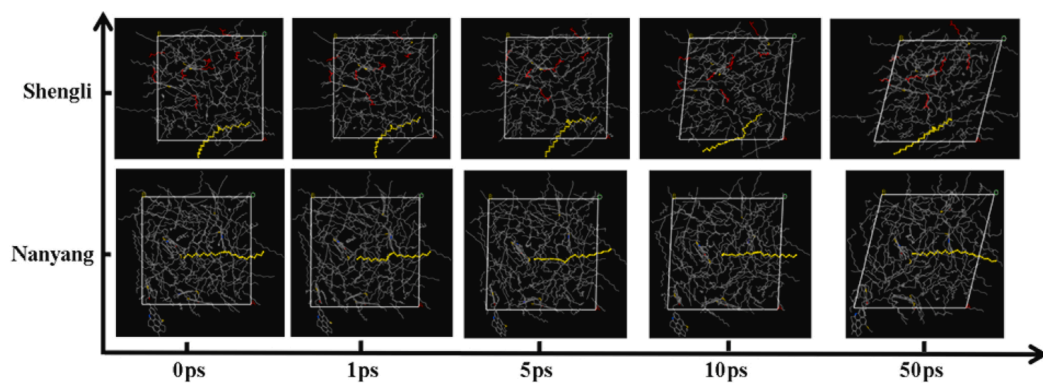


Figure 1.14: Disentanglement of molecular chains [91]

In addition to these studies regarding the physical realism of materials,

the speed-up of simulation has also been addressed, which is seldom mentioned in the grease simulation. [60] implemented SPH entirely on the GPU, and achieved the real-time simulation of free-surface flow with 60,000 particles. The aforementioned interactive viscoelastic fluid simulation [14] was enabled by the improved stability of the solver that allows large time step sizes.

Lastly, SPH is known to be subject to interpolation inaccuracy whilst some studies employ SPH, such as [59], [60]. Nonetheless, there are remedies for this inaccuracy that might be more suitable for engineering applications where accuracy is paramount. Examples include Corrected SPH (CSPH) [76], Mixed Corrected Symmetric SPH (MC-SSPH) [77], Lagrangian gradient smoothing method (L-GSM) [109], and Moving Least Reproducing Kernel particle method (MLSRK) [28].

1.5 The goal of this study

This study is motivated by the ongoing efforts in the ball bearing industry (Fig. 1.15) to utilise Computational Fluid Dynamics (CFD) in product design, as evidenced by studies such as [124] and [130]. Despite these advancements, a gap remains in our understanding: the impact of different greases on the performance of ball bearings has not been thoroughly investigated in simulations.

Moreover, this study is sponsored by the author's employer NSK, one of the largest ball-bearing manufacturers in Japan. They strive to optimise the design of grease-lubricated ball bearings, currently relying on trial and error in experiments. There is a need for grease CFD, which could help establish more theory-based product design whilst also reducing the trial and error.

In light of this, the primary aim of this study is to develop a numerical method for simulating grease behaviour, with the ultimate goal of applying this method to predict the torque of ball bearings when using different types of grease. However, it is important to note that as a first step towards this goal, this research will focus on simple tests involving grease, rather than on the actual ball bearing.

Grease, due to its complex non-Newtonian nature, is believed to undergo various modes of motion within the bearing, including shear, compression, and tension (Fig. 1.16). These motions occur between the various components of the bearing, such as the balls, cage, and inner and outer rings (as illustrated in Fig. 1.17). Each mode of motion could elicit a unique response from the grease, warranting individual investigations to assess the accuracy of numerical simulation.

The tests designed for this study aim to evaluate the behaviour of grease

under these different modes of motion as separately as possible. This approach serves as a stepping stone towards our ultimate goal of accurately simulating the performance of ball bearings.

The subsequent chapters will delve into the methodology employed in this study and discuss the results obtained. By addressing the current gaps in our understanding, this research hopes to contribute to the field of ball bearing design and optimisation.

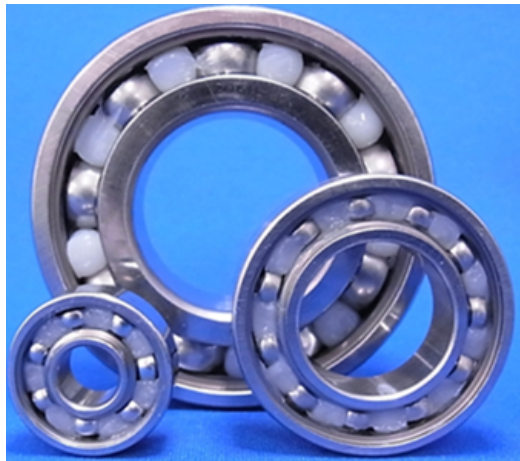


Figure 1.15: Grease in ball bearings of different sizes before the shield installation (courtesy to the author's colleague)



Figure 1.16: Grease stretching when scraped with spatula(author's own)

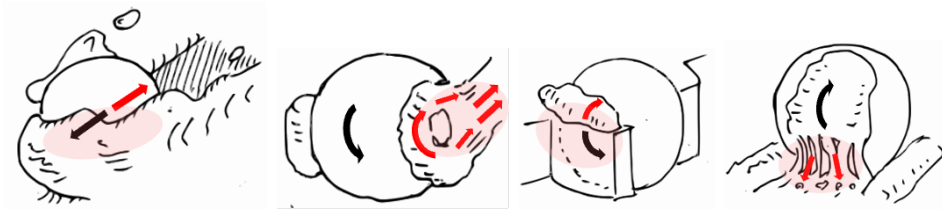


Figure 1.17: Modes of grease motions in the ball bearing
courtesy of the lead author of [130]

2 Simulation method and its implementation in this work

This chapter outlines the simulation method used in this study, which will subsequently be employed in the grease tests, accompanied by benchmarks.

As shown later in this chapter, this study implements the reproduced kernel particle method(RKPM) from scratch while adding extra features, such as elastoplasticity, surface tension and GPU parallelisation. In the implementation phase, C++ and its NVIDIA GPU equivalent, CUDA, were chosen as the programming languages due to their efficiency, object-oriented nature, and the abundance of helpful online resources.

Moreover, two libraries were used to facilitate the implementation. One is OpenGL Mathematics(GLM). The RKPM formulation involves a local matrix inversion(eq.(2.57)) for each particle, which is performed on each GPU thread. GLM is a header-only C++ mathematics library for graphics software based on the OpenGL Shading Language that is compatible with CUDA [1]. In this study, all field variable vectors, such as position and velocity, were instantiated as a GLM vector class. The other library used is Thrust, an equivalent to the C++ Standard Template Library(STL) developed for CUDA by NVIDIA. All field variables are stored as arrays. In the sorting process, those arrays are sorted such that the particles close together in the physical space are also close together in the memory space. The Thrust sorting function(`thrust::sort_by_key`) is more efficient than the STL equivalent as it sorts arrays on the GPU while executing from the CPU side. Other optimisations regarding GPU computing will be mentioned in the relevant subchapter.

2.1 Spatial discretisation of particle methods

In the realm of numerical methods for handling grease, two methods predominantly stand out: the Volume of Fluid (VOF) combined with the Finite Volume Method(FVM), and Smoothed Particle Hydrodynamics(SPH) and its variants, such as RKPM. Each method has its own set of advantages and disadvantages. One of the key advantages of particle methods is the guaranteed conservation of mass, a crucial aspect in grease simulation. For instance, in the context of ball bearings, an overestimation of grease quantity could lead to an overestimation of fluid resistance and lubricating ability, and vice versa.

In terms of memory requirements, particle methods have a distinct advantage. In the context of grease-lubricated ball bearings, typically only

30% of the space is occupied by grease, with the remainder being air. Given the significantly lower viscosity and density of air (approximately $1\text{E-}9$ and $1\text{E-}3$, respectively), it can be largely ignored from a physics standpoint. Particle methods inherently account for this, whereas VOF necessitates solving for the entire space, irrespective of the presence of grease. This implies an unnecessary consumption of computational resources.

Furthermore, particle methods are generally more straightforward to implement than FVM, which is particularly beneficial when customising with additional models, such as elastoplasticity.

However, particle methods do have their drawbacks. They typically require the use of artificial parameters that could influence the solution. While this is acceptable in computer graphics, where the primary goal is to achieve visually plausible results, it can pose a problem in engineering applications like grease simulation, where accurate quantitative discussions about physical parameters are often required.

Taking all these factors into consideration, this study opts for a particle method.

Amongst mesh-free particle methods, SPH has widely been used. It was originally created for astrophysical applications [51], and subsequently found utility in the realm of fluid dynamics [114]. SPH has been applied to CFD due to its advantages over the more conventional volumetric methods. In SPH, for example, moving interfaces can be tracked easily, since the material is represented by particles, each of which carries mass as it moves [114]. In principle, advection can be accurately treated as this only involves integrating the equation of motion for each particle. Moreover, particle methods are free from mesh-related difficulties. In grid-based methods, firstly mesh generation is required, which could be complicated particularly for complex geometries or unstructured meshes. In addition, mesh distortion could occur in the case of mesh morphing for moving boundaries. Remeshing methods exist, but those would make an extra task. On the other hand, being a Lagrangian method, SPH is highly adaptive and free of these problems [97].

Some of the notable contributions are as follows. For a more comprehensive overview, several review papers are available [17], [116], [167], [173], [186].

SPH has been extensively applied in various fields, including the simulation of free-surface flows of water in computer graphics and the simulation of non-Newtonian fluids. Several significant works stand out in terms of methods. For instance, the method was first introduced by [36], who proposed an implicit incompressibility method to address the time step constraint in the explicit method. This was achieved by solving the Poisson equation to ensure the divergence-free condition.

Later, [68] developed a method that combined divergence-free and constant-density methods, emphasising their accuracy and convergence. [183] further compared the accuracy and stability of incompressibility between these methods and developed a new scheme using particle shifts that proved to be both accurate and stable.

Implicit incompressibility schemes are often favoured in the field of computer graphics due to their speed and the visually plausible results they produce. A significant advancement in this area was made by [71], who proposed a new discretisation of the Pressure Poisson Equation (PPE)(eq. 2.1) that greatly improved the convergence of the solver, resulting in a speed-up from previous implicit methods. Similar works on implicit methods have been conducted by [18], [80], and [156].

$$\nabla \cdot \left(\frac{1}{\rho} \nabla P \right) = \frac{\nabla \cdot \mathbf{u}}{\Delta t} \quad (2.1)$$

In addition to water, SPH has also been applied to non-Newtonian fluids. [149] implemented incompressibility to handle non-Newtonian fluids with free surfaces as a remedy for artificial compressibility. Later, [184] proposed a method to simulate non-Newtonian fluids with free surfaces, addressing efficient boundary handling and tensile instability. [44] further implemented implicit viscosity to stably handle high-viscosity fluids and explored the flow pattern of the injection moulding process.

Beyond enabling incompressibility and the simulation of non-Newtonian fluids through implicit schemes, there have been other improvements to the original SPH. [54] proposed assigning different time steps to different fluid regions and solving them at different rates to minimise computational cost, achieving approximately double the speed in their benchmark.

While SPH typically handles wall boundaries by placing boundary particles, which is straightforward to implement but prone to errors, [163] developed a semi-analytical boundary method that enhances feasibility and efficiency with complex geometries. To address the instability sometimes experienced at free surfaces, [3] introduced a surface viscosity method that enables higher accuracies.

Having discussed the significant works related to the methods of SPH, its applications in the engineering field are to be reviewed. Specifically, SPH has been effectively utilised in lubricant flows and marine engineering.

In the context of lubricant flows, SPH has been employed to investigate the wear of material at contact points considering the surface roughness under oil lubrication [133], [134]. It has also been used to optimise the lubrication and cooling in aero engine gearbox applications, with the authors simulating

oil flows and verifying the results against more traditional VOF [82]. Furthermore, SPH has been utilised to optimise the oil level of oil-bath lubrication in gearboxes, with the results compared to Particle Image Velocimetry (PIV) [75]. To analyse the flow in an aero-engine bearing chamber, an SPH code was implemented that can examine the prevailing flow types, such as shear-driven wall-films, droplet-wall- and droplet-film-interaction [179].

SPH has also been deployed in marine engineering, for example, to study the oil spill onto the ocean along with the mechanisms of floating booms, the tool used to respond to an oil spill in marine or inland waters [166]. It has also been used to investigate problems such as the installation of gravity foundations for wind turbines (lowered to the ground by partial flooding), jacket launching for offshore oil and gas production facilities, and the harbour bed erosion induced by the starting propeller of large container vessels [165]. In the context of arctic sailing, water splashing and forming ice on ships is a significant issue. To this end, SPH has been used to investigate the water splash by wave height [25]. Other notable applications in this field include [106] and [6].

To exploit the advantages, such as memory requirements and customisability, this study adopts a particle method that is based on SPH, Reproducing Kernel Particle Method(RKPM). The rest of the chapter will discuss the physical equations and their discretised forms.

In computational fluid dynamics, fluid is assumed to follow the Navier-Stokes equations. The Cauchy momentum equation is a more generalised form, and in the Lagrangian form, it reads

$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \quad (2.2)$$

where ρ is the fluid density, \mathbf{u} the velocity, $\boldsymbol{\sigma}$ the stress, and \mathbf{g} gravity. In the case of purely viscous fluid, the stress involves only viscous stress and pressure, and is expressed as

$$\boldsymbol{\sigma} = \mu \{ \nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{d} (\nabla \cdot \mathbf{u}) \mathbf{I} \} + \zeta (\nabla \cdot \mathbf{u}) \mathbf{I} - p \mathbf{I} \quad (2.3)$$

where μ is the dynamic viscosity, ζ the bulk viscosity, and p the pressure.

In the case of explicit SPH, pressure is usually computed based on density. There are two ways to obtain density. One is the position-based approach, where density is based on the count and the distance of neighbour particles. This method applies the SPH interpolation eq.(2.20) to density to yield

$$\rho_i = \sum_j m_j W(\mathbf{x} - \mathbf{x}_j, h) \quad (2.4)$$

The other method is the divergence-based or velocity-based approach that uses the continuity equation as

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \mathbf{u} \quad (2.5)$$

As the name suggests, this method is based on the velocity of the particles, and the density increases when neighbour particles approach, and decreases when they disperse. In the discretised form, this yields

$$\frac{D\rho_i}{Dt} = \sum_j m_j (\mathbf{u}_i - \mathbf{u}_j) \cdot \nabla W(\mathbf{x} - \mathbf{x}_j, h) \quad (2.6)$$

The position-based density eq.(2.4) and the velocity-based density eq.(2.6) both have pros and cons. The position-based approach exactly conserves mass over time. However, at the same time, the density is inaccurate at the free surface and at the material interface due to the insufficient number of particles. On the other hand, the velocity-based approach does not suffer from these issues, but it does not conserve mass over time due to the accumulation of integration error, requiring employing the position-based approach every several time steps. This study adopts the position-based approach. In either case, pressure is obtained using the equation of state(EOS).

$$p = B(\rho - \rho_0) \quad (2.7)$$

where ρ_0 is the rest density [80]. There are other forms of EOS as in [115], [155], but this study employs eq. (2.7) in preference of brevity.

2.1.1 SPH discretisation

In SPH, a function $f(\mathbf{x})$ is approximated using a smoothing function over the support domain Ω . The approximation starts with the identity

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \quad (2.8)$$

where $\delta(\mathbf{x} - \mathbf{x}')$ is the Dirac delta function denoted as

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}' \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

The SPH approximation is obtained by replacing the Dirac delta function with an SPH smoothing function called a kernel W as

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (2.10)$$

where $\langle f(\mathbf{x}) \rangle$ denotes the SPH interpolation, and h the smoothing length or the influence area of W , where the particles inside are counted as neighbours hence exerting influence.

There are choices for smoothing functions, but three main conditions need to be met. The first is the normalisation condition, which is

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1 \quad (2.11)$$

The second condition is the delta function property as

$$\lim_{h \rightarrow 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}') \quad (2.12)$$

Finally, the third condition is the compact condition:

$$W(\mathbf{x} - \mathbf{x}', h) = 0 \text{ for } |\mathbf{x} - \mathbf{x}'| > h \quad (2.13)$$

The last condition means the neglect of the effect from outside the support domain. This will be useful in the approximation of the derivative of a function, as will be shown shortly. Moreover, it also means that in the discretised form of the SPH approximation, the number of the neighbouring particles to be taken into account is limited, reducing the computational cost.

The approximation for the derivative of a function $\partial f(\mathbf{x})$ starts with substituting $\partial f(\mathbf{x})$ in eq.(2.10) as

$$\langle \partial f(\mathbf{x}) \rangle = \int_{\Omega} \partial f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (2.14)$$

Using the chain rule

$$\partial f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}') = \partial [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}')] - f(\mathbf{x}') \partial W(\mathbf{x} - \mathbf{x}') \quad (2.15)$$

eq.(2.14) becomes

$$\langle \partial f(\mathbf{x}) \rangle = \int_{\Omega} \partial [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}')] d\mathbf{x}' - \int_{\Omega} f(\mathbf{x}') \partial W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (2.16)$$

Using the divergence theorem, this can be rewritten as

$$\langle \partial f(\mathbf{x}) \rangle = \int_S [f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}')]d\mathbf{S} - \int_{\Omega} f(\mathbf{x}')\partial W(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \quad (2.17)$$

where S denotes the surface of the support domain. Because of the compact condition(eq.(2.13)), the first term on the RHS is zero. Therefore, the derivative of a function can be expressed as

$$\langle \partial f(\mathbf{x}) \rangle = - \int_{\Omega} f(\mathbf{x}')\partial W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}' \quad (2.18)$$

This means that the derivative of a function is transmitted to the derivative of the smoothing function, making the differentiation operation possible for a given smoothing function.

The SPH approximations written in the continuous form(eqs.(2.10) and (2.18)) will be converted to the discretised form by replacing the integral with the summation over the particles in the support domain. Let j a particle within the support domain of the particle i , and the mass of the particle j is expressed as

$$m_j = \Delta V_j \rho_j \quad (2.19)$$

Therefore, eq.(2.10) becomes

$$\begin{aligned} \langle f(\mathbf{x}_i) \rangle &= \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x}_i - \mathbf{x}')d\mathbf{x}' \\ &\approx \sum_{j=1}^N f(\mathbf{x}_j)W(\mathbf{x}_i - \mathbf{x}_j, h)\Delta V_j \\ &= \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{x}_j)W(\mathbf{x}_i - \mathbf{x}_j, h) \end{aligned} \quad (2.20)$$

Similarly, the derivative of a function is

$$\langle \partial f(\mathbf{x}_i) \rangle = - \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{x}_j)\partial W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (2.21)$$

2.2 Elastoplasticity

As discussed in the previous chapter, the experiments of grease suggest that grease possesses elasticity in addition to viscosity. Elasticity is thought to

play a key role in the solid-like behaviour of grease, and is relevant in material property tests, such as cone penetration, plate separation, and also the start-up torque of the ball bearing. Therefore, this study attempts to incorporate elasticity. In the field of CFD, several models have been proposed that handle elasticity, and have produced compelling results.

Here, the method by [52], [53] is used, which appears to be one of the first methods in the framework of CFD, and also the fundamental upon which many other models were subsequently built. In this method, strain is computed from strain rate, convected with a convection equation, and finally integrated into the stress in the Navier-Stokes equation. Here, the method will be described for completeness. Firstly, total strain ϵ^{Tot} is separated into elastic strain ϵ^{Elc} and plastic strain ϵ^{Plc} as

$$\epsilon^{Tot} = \epsilon^{Elc} + \epsilon^{Plc} \quad (2.22)$$

The total strain at one point in time is expressed as

$$\epsilon^{Tot} = \epsilon_0^{Tot} + \int_0^t \dot{\epsilon}^{Tot} dt \quad (2.23)$$

where ϵ_0^{Tot} is the total strain at the beginning of the simulation. The derivative of total strain is equal to strain rate $\dot{\epsilon}^{Tot}$, which reads

$$\dot{\epsilon}^{Tot} = \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} \quad (2.24)$$

Since static pressure is already taken into account in the Navier-Stokes equation, the elastic normal stress, or pressure arising from dilation and compression will be removed to avoid double counting. Therefore, only elastic strain deviation ϵ' contributes to elastic stress, which reads

$$\epsilon' = \epsilon^{Elc} - \frac{Tr(\epsilon^{Elc})}{3} \mathbf{I} \quad (2.25)$$

As for plasticity, no plastic deformation occurs as long as the magnitude of strain deviation stays below the yield point γ . Note that unlike the convention in solid mechanics, this yield point refers to yield strain, not yield stress. Once strain deviation exceeds this point, plastic deformation occurs at a certain decay rate α rather than at an infinite rate that instantly brings total strain to the yield point. Hence, the yield stress is expressed as

$$\dot{\epsilon}^{Plc} = \alpha \frac{\epsilon'}{\|\epsilon'\|} \max\{0, \|\epsilon'\| - \gamma\} \quad (2.26)$$

where $\|\epsilon'\|$ is the Frobenius norm, and in the case of 3D, it is

$$\|\boldsymbol{\epsilon}'\| = \{(\epsilon'_{xx})^2 + (\epsilon'_{yy})^2 + (\epsilon'_{zz})^2 + 2(\epsilon'_{xy})^2 + 2(\epsilon'_{yz})^2 + 2(\epsilon'_{zx})^2\}^{\frac{1}{2}} \quad (2.27)$$

Given that the initial strain is zero, combining all these results in the elastic strain(now, simply denoted as $\boldsymbol{\epsilon}$) as

$$\dot{\boldsymbol{\epsilon}} = \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} - \alpha \frac{\boldsymbol{\epsilon}'}{\|\boldsymbol{\epsilon}'\|} \max\{0, \|\boldsymbol{\epsilon}'\| - \gamma\} \quad (2.28)$$

Strain is then advected according to the advection equation. In the Lagrangian form, it would be

$$\dot{\boldsymbol{\epsilon}} = \frac{D\boldsymbol{\epsilon}}{Dt} \quad (2.29)$$

Combined with the eq. (2.28),

$$\frac{D\boldsymbol{\epsilon}}{Dt} = \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} - \alpha \frac{\boldsymbol{\epsilon}'}{\|\boldsymbol{\epsilon}'\|} \max\{0, \|\boldsymbol{\epsilon}'\| - \gamma\} \quad (2.30)$$

Here, when the fluid rotates, the strain tensor needs to be rotated accordingly, which is achieved by using a co-rotational derivative as

$$\dot{\boldsymbol{\epsilon}} = \frac{D\boldsymbol{\epsilon}}{Dt} + \boldsymbol{\epsilon} \mathbf{W} - \mathbf{W} \boldsymbol{\epsilon} \quad (2.31)$$

where \mathbf{W} is the anti-symmetric spin tensor

$$\mathbf{W} = \frac{\nabla \mathbf{u} - (\nabla \mathbf{u})^T}{2} \quad (2.32)$$

Altogether, the rotational invariant form of the strain advection equation is expressed as

$$\frac{D\boldsymbol{\epsilon}}{Dt} = \boldsymbol{\epsilon} \mathbf{W} - \mathbf{W} \boldsymbol{\epsilon} + \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} - \alpha \frac{\boldsymbol{\epsilon}'}{\|\boldsymbol{\epsilon}'\|} \max\{0, \|\boldsymbol{\epsilon}'\| - \gamma\} \quad (2.33)$$

The strain is then integrated as

$$\boldsymbol{\epsilon}^{n+1} = \boldsymbol{\epsilon}^n + \dot{\boldsymbol{\epsilon}} \Delta t \quad (2.34)$$

This strain is eventually integrated into the stress term of eq. (2.3) as

$$\boldsymbol{\sigma} = \mu_v \left\{ \nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{d} (\nabla \cdot \mathbf{u}) \mathbf{I} \right\} + \zeta (\nabla \cdot \mathbf{u}) \mathbf{I} - p \mathbf{I} + \mu_e \boldsymbol{\epsilon} \quad (2.35)$$

where μ_v is the viscosity, and μ_e Young's modulus.

This study uses the above model by incorporating eqs. (2.33) and (2.35), favouring the clear physical meaning and simplicity. However, to clarify the reason for selecting this model, it is worth reviewing other models, many of which are connected to this model in certain ways. [59] used the same method regarding elasticity, but with SPH. Combined with the newly developed particle re-sampling method, the characteristic stretching motion of viscoelastic fluids was reproduced. [14] used a constraint-based model of viscoelasticity, where conformation tensor is handled in a very similar manner to [52]. The authors employed an implicit conformation constraint on fluid velocities, which enables high viscoelasticity simulation efficiently. Whilst the method produced gripping visual effects of viscoelastic fluids, as [175] points out, the model includes parameters that are only meaningful for polymers, and in the case of other fluids, those parameters need to be decided artistically. Some examples employ deformation tensor rather than strain tensor or conformation tensor. [198] simulated viscoelastic phenomena such as thin filaments, which were enabled by the combination of the semi-implicit elasticity scheme and adaptive meshing. [189] simulated dense foam's viscoelastic behaviour, akin to that of grease. This was helped by explicit tear mode, which is particularly helpful in the material point method(MPM) along with particle resampling, where particles fill the void around the stretched region. [121] extended [189] for mixtures, the behaviour of which follows a blending model where each constituent of the mixture is considered by a mixing ratio parameter.

2.2.1 Benchmark

A benchmark test of elasticity was performed, where a fluid cube falls to the floor, both in 2D and 3D. Fig. 2.1 shows the case of 3D.

In the absence of elasticity(Figs. 2.1(a) and 2.2(a) in 2D and 3D respectively), the material continues to deform and flow due to gravity, showing the material's purely viscous behaviour. On the other hand, with elasticity(Figs. 2.1(b) and 2.2(b) in 2D and 3D respectively), the material stops deforming at some point in time, and it holds the shape against gravity, showing the elastic characteristics of the material. Therefore, this benchmark has successfully demonstrated the validity of this method for elasticity albeit only visually. Table 2.2 shows the parameters used in the benchmark. Fig. 2.3 shows the particle files used.

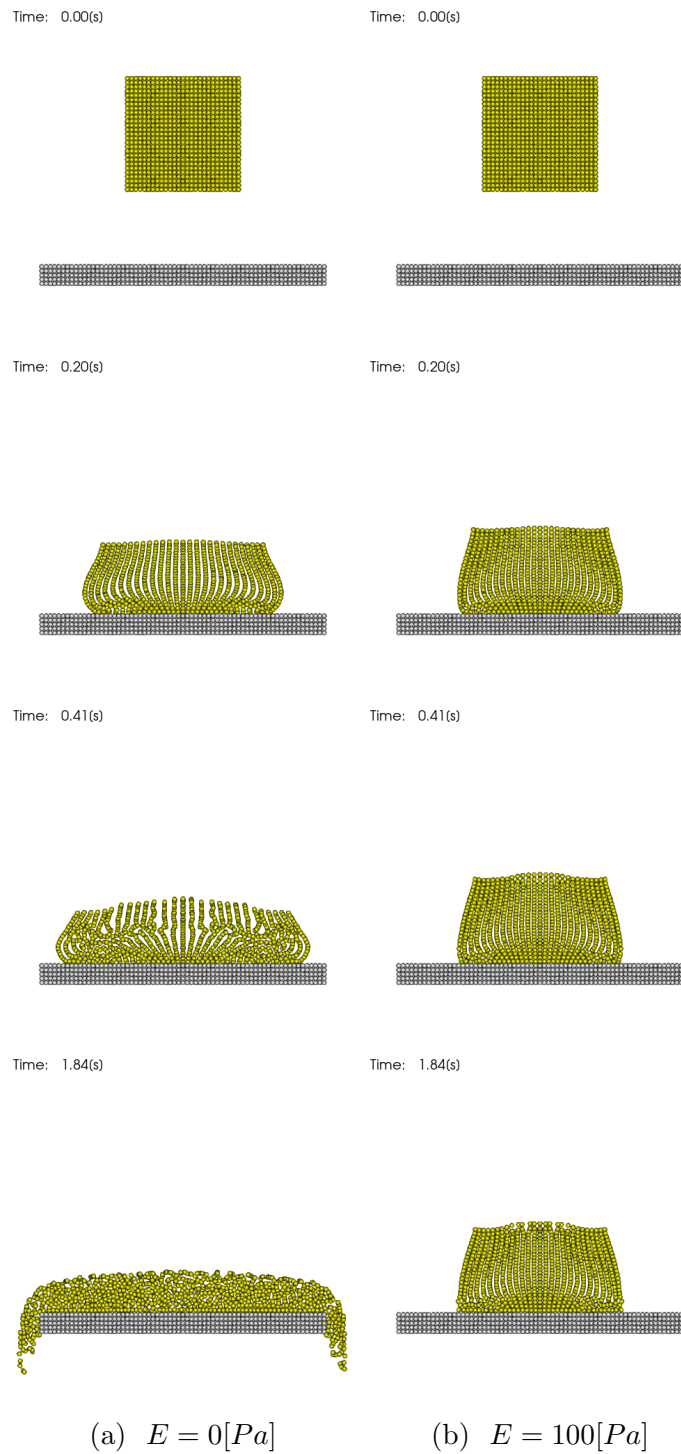


Figure 2.1: Effect of elasticity in 2D

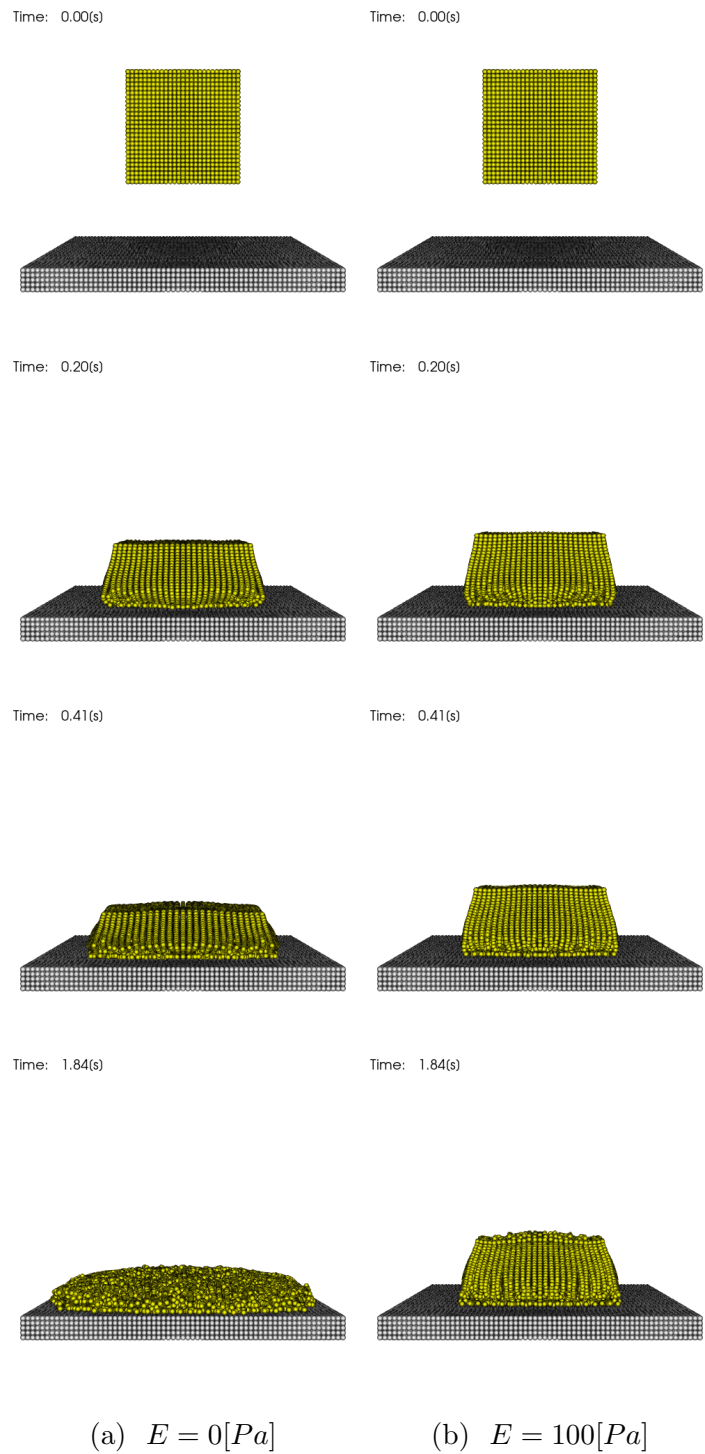


Figure 2.2: Effect of elasticity in 3D

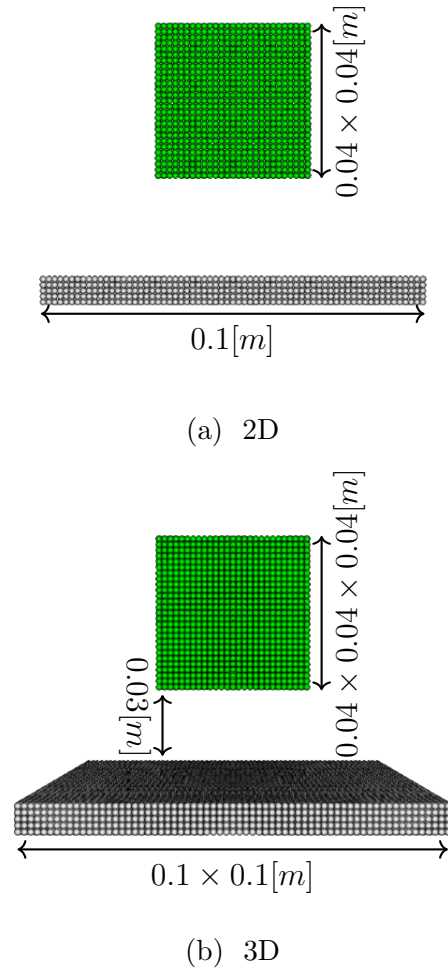
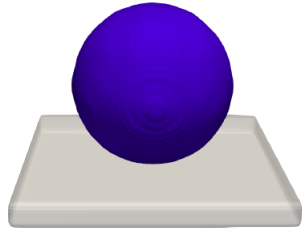


Figure 2.3: Particle files used in the elasticity benchmarks

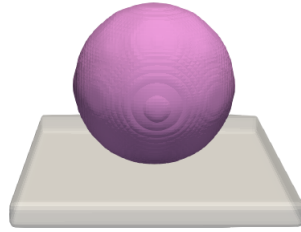
Table 2.1: Parameters used in the benchmark of elasticity

	w/o elasticity	w/ elasticity
particle size[m]	1.5e-3	
kernel radius[m]	3.75e-3	
particle count	1,064(2D), 42,128(3D)	
time step[s]	2.0e-5	
velocity blending coefficient	0.8	
EOS coefficient B	3.0e+3	
shift coefficient	0.0	
surface tension coefficient	0.0	
surface particle volume coefficient	1.07	
surface particle number of neighbours	20(2D), 90(3D)	
density $\rho[kg/m^3]$	1000	
viscosity $\mu_v[Pa \cdot s]$	1000	
Young's modulus $E[Pa]$	0	100
Yield strain ϵ_Y	-	1.0
decay rate $\alpha[/math>/s]$	-	100

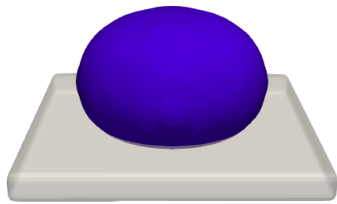
Furthermore, another benchmark of elasticity was performed to compare this study with the previous[59]. In this benchmark, elastic balls are placed on a surface. They deform by gravity, and the final shapes differ by elastic modulus. Two different elastic moduli were tested, and the ball with the lower elastic modulus deformed more than the one with the higher modulus, in agreement with [59](Fig. 2.4). The deformations in this study are smaller than [59] likely due to the values of density and the ball diameter. This is because [59] did not list any parameters except elastic moduli and decay rates, and therefore, this study set its own values. The particle file used in this benchmark is shown in Fig. 2.5, and the parameters in Table 2.2.



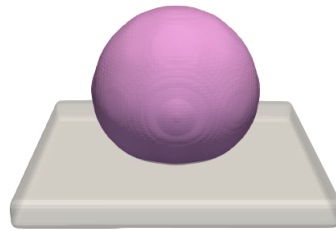
(a) $E = 1e3[Pa]$, $\alpha = 10$,
start, this study



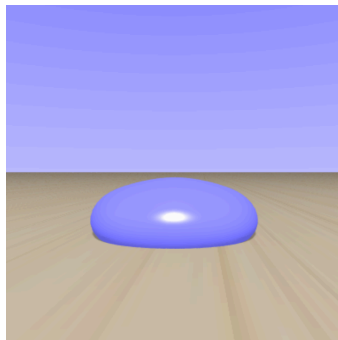
(b) $E = 1e4[Pa]$, $\alpha = 10$,
start, this study



(c) $E = 1e3[Pa]$, $\alpha = 10$,
end, this study



(d) $E = 1e4[Pa]$, $\alpha = 10$,
end, this study



(e) $E = 1e3[Pa]$, $\alpha = 10$,
[59]



(f) $E = 1e4[Pa]$, $\alpha = 10$,
[59]

Figure 2.4: Different elastic moduli in 3D

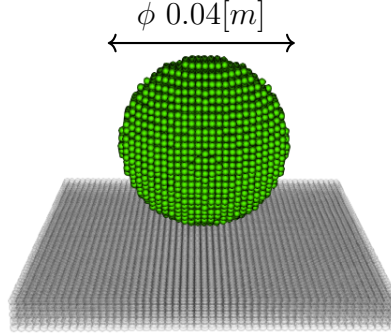


Figure 2.5: Particle file used in the elastic moduli benchmark

Table 2.2: Parameters used in the elastic moduli benchmark

	$E = 1e3[Pa]$	$E = 1e4[Pa]$
particle size[m]	1.5e-3	
kernel radius[m]	3.75e-3	
particle count	1,064	
time step[s]	2.0e-5	
velocity blending coefficient	0.8	
EOS coefficient B	3.0e+3	
shift coefficient	0.0	
surface tension coefficient	0.0	
surface particle volume coefficient	1.1	
surface particle number of neighbours	90	
density $\rho[kg/m^3]$	3000	
viscosity $\mu_v[Pa \cdot s]$	1.0	
Young's modulus $E[Pa]$	1e3	1e4
Yield strain ϵ_Y	1.0	1.0
decay rate $\alpha[s]$	10	10

2.3 Fluid-rigid body interaction

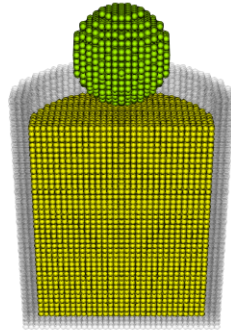
This study will later investigate grease in the cone penetration test, where a metal cone is dropped into grease, which involves free fall. To simulate this test, fluid rigid body interaction is incorporated. Since the cone penetration test involves only one degree of freedom, which is z-direction translation, the rotational motion of the rigid body will not be considered. This is done in a relatively simple manner with particle methods [5]. Rigid bodies are represented as a set of particles in exactly the same way as fluids are. At the end of each time step, the total force exerted on the rigid body is calculated by accumulating over the rigid body particles. However, the direct implementation of [5] would be impractical due to the use of weak formulation as in eqs.(2.78)-(2.82). Instead, the acceleration of the rigid body $\dot{\mathbf{u}}_{rigid}$ is obtained by mass-averaging over the particles as in eq.(2.36), which in turn is used to update the velocity \mathbf{u}_{rigid} . The acceleration of each particle is obtained by eq. (2.82).

$$\begin{aligned}\dot{\mathbf{u}}_{rigid} &= \sum_i \left(\dot{\mathbf{u}}_i \frac{m_i}{M_{rigid}} \right) + \mathbf{g} \\ \mathbf{u}_{rigid}^{n+1} &= \mathbf{u}_{rigid}^n + \dot{\mathbf{u}}_{rigid} \Delta t\end{aligned}\tag{2.36}$$

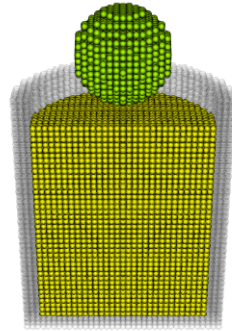
2.3.1 Benchmark

A benchmark has been performed to evaluate this fluid rigid body interaction formulation. Two balls of different densities, 7000 and 500 [kg/m^3] were dropped into a liquid with the density of 1000 [kg/m^3], and their motions were observed. This is by no means a quantitative comparison, but rather a qualitative one. Fig.2.6 shows the results. The actual geometry is axisymmetric, but only half is shown to increase visibility. As expected, the heavier ball sinks completely to the bottom whereas the lighter one stays on the liquid surface, supported by sufficient buoyancy. Therefore, this benchmark confirms the validity of this fluid-rigid body interaction. This can also be confirmed by the displacements of the balls(Fig.2.7). Fig.2.8 and Table 2.3 show the particle files and the parameters used, respectively.

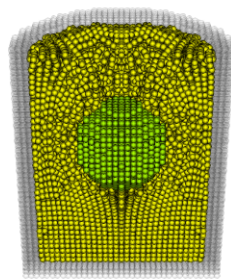
Time: 0.00(s)



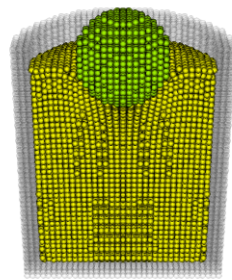
Time: 0.00(s)



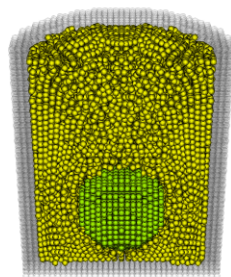
Time: 0.42(s)



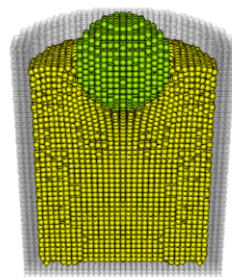
Time: 0.42(s)



Time: 0.83(s)



Time: 0.83(s)



(a) heavier ball

(b) lighter ball

Figure 2.6: Dropping heavier and lighter balls into a fluid

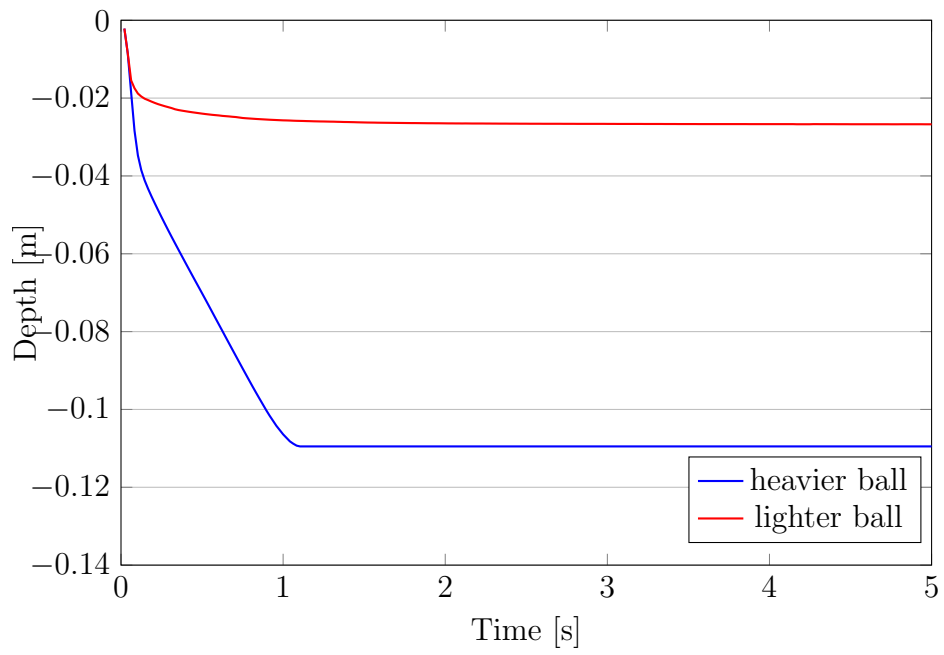


Figure 2.7: Ball displacements over time

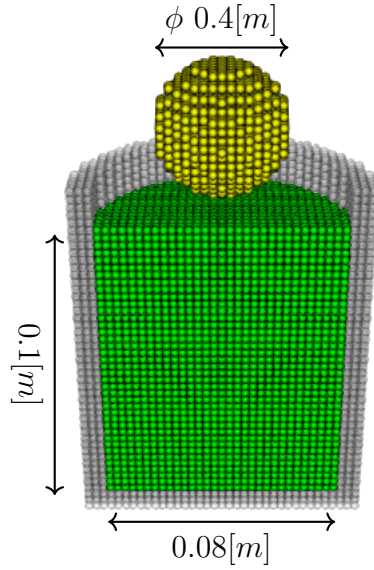


Figure 2.8: Particle file used in the fluid rigid body interaction benchmark

Table 2.3: Parameters used in the fluid rigid body interaction benchmark

	heavier ball	lighter ball
particle size [m]	2.5e-3	
kernel radius[m]	6.25e-3	
particle count	50,229	
time step [s]	5.0e-5	
velocity blending coefficient	0.85	
EOS coefficient B	3.0e+3	
shift coefficient	0.0	0.0
surface particle volume coefficient	1.07	
surface particle number of neighbours	90	
surface tension coefficient [N/m]	0.0	
rigid body density $\rho_{Rigid}[kg/m^3]$	7000	500
fluid density $\rho_f[kg/m^3]$	1000	
viscosity $\mu_v[Pa \cdot s]$	10	

2.4 Reproducing Kernel Particle Method

2.4.1 Inconsistency problem in SPH

Being the most widely used particle method, SPH has various advantages, but nonetheless, it is not free of drawbacks, one of which is inconsistency. SPH interpolation is expected to be "consistent", meaning the interpolation exactly reproduces the original value of a variable. The condition for a constant field function $f(\mathbf{x}) = c$ is

$$f(x) = \int_{\Omega} cW(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = c \quad (2.37)$$

where \mathbf{x} is the coordinate vector of a particle, Ω the support domain, \mathbf{x}' coordinate vector of the infinitesimal volume, $d\mathbf{x}'$ is the infinitesimal volume, and h the support radius. The above equation is equivalent to

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1 \quad (2.38)$$

This is called the normalisation(unity) condition. For a linear function $f(\mathbf{x}) = c_0 + c_1\mathbf{x}$ to be exactly reproduced, the condition is

$$f(x) = \int_{\Omega} (c_0 + c_1\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = c_0 + c_1\mathbf{x} \quad (2.39)$$

Using eq. (2.38),

$$f(x) = \int_{\Omega} \mathbf{x}' W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = \mathbf{x} \quad (2.40)$$

Finally, the condition for a linear field leads to

$$\int_{\Omega} (\mathbf{x} - \mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0 \quad (2.41)$$

In the discretised form, eqs. (2.37), (2.41) are respectively

$$\sum_{j=1}^N W(\mathbf{x} - \mathbf{x}_j, h) \Delta\mathbf{x}_j = 1 \quad (2.42)$$

and

$$\sum_{j=1}^N (\mathbf{x} - \mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j, h) \Delta\mathbf{x}_j = 0 \quad (2.43)$$

where \mathbf{x}_j is the coordinate vector of neighbour particle j , N the number of neighbour particles, and $\Delta\mathbf{x}_j$ the volume of the neighbour particle. In the case of SPH, these conditions are satisfied when particles are evenly distributed at the interval of the particle size as in Fig. 2.9 (a). Nonetheless, this is not always the case. For example, when the particle is near the boundary, such as free surfaces, the lack of neighbour particles results in the truncation of the kernel (Fig. 2.9 (b)). In addition, during the simulation, particles constantly move around, and the distribution or spacings change accordingly. In this case, consistency will be lost in the discretised form (Fig. 2.9 (c)). Therefore, these cases suffer from particle inconsistency, where eqs. (2.42), (2.43) are not satisfied. This issue has been widely discussed ([17], [28], [49], [119]). Inconsistency is undesirable since it can lead to so-called tensile instability, where the velocity explodes due to the numerical instability under tension ([161], [78], [42]).

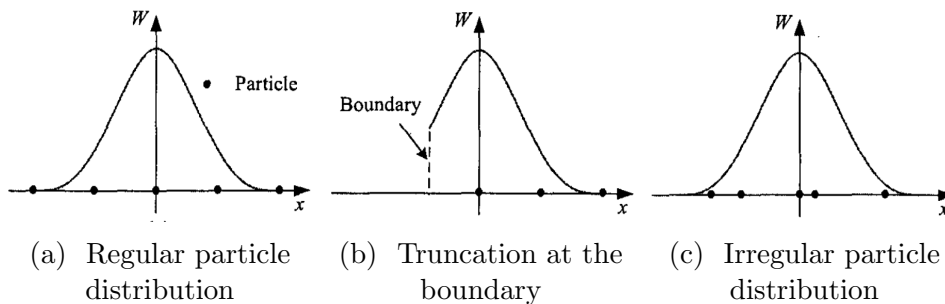


Figure 2.9: Particle distribution and SPH consistency [97]

2.4.2 Reproducing Kernel Particle Method(RKPM)

The target of this study is the simulation of grease for engineering purposes, where high accuracies are desired. Therefore, as a remedy to the SPH inconsistency, the Reproducing Kernel Particle Method(RKPM) is employed in this study, which adds an extra correction term to the original SPH [100]. Since RKPM is a particle method that requires no mesh in the same manner as SPH, it still enjoys the advantages of Lagrangian methods.

After its establishment [100], There have been several applications. The quantity of work is somewhat limited compared to SPH possibly because RKPM is newer than SPH and more complicated to implement. In the field of solid mechanics, [135] studied the fracture of a circular plate subjected to an underwater contact explosion. In this research, a Smoothed Particle Hydrodynamics (SPH)-based fluid solver was coupled with an RKPM-based

solid solver. [126] explored the radial crushing of cylindrical tubes composed of anisotropic fibre-reinforced composites. Their findings favoured the accuracy of RKPM and its ability to adjust the support size to accurately capture the size and shape of finite-size fracture process zones. [151] introduced an innovative numerical technique known as the augmented corrected collocation method to RKPM. This method was designed to capture field quantities across the interfaces between material discontinuities, and its effectiveness was verified in a 2D problem of crack propagation near a material interface. The application of RKPM extends to the simulation of shear band formations induced by cracks, as demonstrated by [93]. Their work used RKPM to alleviate the mesh-alignment sensitivity that is often associated with the numerical simulation of strain localisation, a common issue in mesh-based methods such as the Finite Element Method (FEM). [96] implemented RKPM in a solid mechanics called Sierra SolidMechanics. The benchmarking of this implementation was carried out using the Taylor bar impact problem, underscoring the method's suitability for addressing large deformation and material failure.

RKPM has also been applied to fluid dynamics. [16] established a framework for air-blast Fluid-Structure Interaction (FSI) using RKPM in combination with isogeometric analysis. This approach effectively addressed instability problems and demonstrated its effectiveness in the Sod shock tube problem and the detonation flow in the presence of multiple objects. In a more recent study, [29] implemented RKPM to simulate the flow of fresh concrete in 3D concrete printing, a technique used in autonomous construction. This research aimed to understand the key factors for successful 3D printing, such as material parameters and printing strategies. Lastly, [185] integrated the RKPM differential operators into Updated Lagrangian Particle Hydrodynamics to capture complex and delicate three-dimensional multiphase flows. Examples of these flows include rising bubbles underwater, splashing on a thin liquid film, and Rayleigh Taylor instability. The validity of this approach was confirmed by comparing the results to experimental and numerical results found in the literature.

The reason for those RKPM adoptions is the improved accuracy from SPH, where inconsistency is an issue due to constant interpolation coefficients that ignore neighbour particle distribution. RKPM adds the boundary correction term to improve accuracy. Here, the method used in this study will be explained along with the derivation.

The derivation of RKPM starts with a function $f(\mathbf{x})$ which is represented in terms of linearly independent functions $P_i(\mathbf{x})$ and a set of unknowns c_i as

$$f(\mathbf{x}) = \sum_{i=1}^m P_i(\mathbf{x})c_i \quad (2.44)$$

This can be rewritten in the matrix notation as

$$f(\mathbf{x}) = \mathbf{P}(\mathbf{x})\mathbf{c} \quad (2.45)$$

where $\mathbf{P} = \{P_1(\mathbf{x}), P_2(\mathbf{x}), \dots, P_m(\mathbf{x})\}$ is the vector of linearly independent functions(e.g. Table 2.4), $\mathbf{c} = \{c_1, c_2, \dots, c_m\}$ is the vector of unknown coefficients. Multiplying either side of eq. (2.45) by $\mathbf{P}^T(\mathbf{y})$ and transforming with a positive window function $W(\mathbf{y})$ yields

$$\langle \mathbf{P}^T f, W \rangle = \langle \mathbf{P}^T \mathbf{P}, W \rangle \mathbf{c} \quad (2.46)$$

Here, W is the SPH kernel that satisfies the unity condition.

$$\int_{R_x} W(\mathbf{x}) dR_x = 1 \quad (2.47)$$

Eq. 2.46 can be solved with respect to \mathbf{c} as

$$\mathbf{c} = \mathbf{M}^{-1}(\mathbf{x}) \langle \mathbf{P}^T f, W \rangle \quad (2.48)$$

where $\mathbf{M}(\mathbf{x})$ is a $m \times m$ non-singular matrix

$$\mathbf{M}(\mathbf{x}) = \langle \mathbf{P}^T \mathbf{P}, W \rangle = \int_{R_x} \mathbf{P}^T(\mathbf{y})\mathbf{P}(\mathbf{y})W\left(\frac{\mathbf{y}-\mathbf{x}}{a}\right) dR_y \quad (2.49)$$

$f^h(\mathbf{x})$, which denotes the interpolated value of $f(\mathbf{x})$, is obtained by substituting eq.(2.48) into eq.(2.45) yields as

$$f^h(\mathbf{x}) = \langle \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}^T f, W \rangle \quad (2.50)$$

After the integral window transform, this can be written as

$$f^h(\mathbf{x}) = \int_{R_x} C(a, \mathbf{x}, \mathbf{y})f(\mathbf{y})W\left(\frac{\mathbf{y}-\mathbf{x}}{a}\right) dR_y \quad (2.51)$$

where $C(a, \mathbf{x}, \mathbf{y})$ is a correction function

$$C(a, \mathbf{x}, \mathbf{y}) = \mathbf{P}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}^T(\mathbf{y}) \quad (2.52)$$

Eq. (2.51) is discretised as

$$f^h(\mathbf{x}) = \sum_{j=1}^m N_j(\mathbf{x})f_j \quad (2.53)$$

where $N_j(\mathbf{x})$ is the reproducing kernel particle interpolation function

$$N_j(\mathbf{x}) = C(a, \mathbf{x}, \mathbf{x}_j) W \left(\frac{\mathbf{y} - \mathbf{x}}{a} \right) \Delta V_j \quad (2.54)$$

The correction function in the discretised form $C(a, \mathbf{x}, \mathbf{x}_j)$ is

$$C(a, \mathbf{x}, \mathbf{x}_j) = \mathbf{b}(\mathbf{x}) \mathbf{P}^T(\mathbf{x}_j) \quad (2.55)$$

where

$$\mathbf{b}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \mathbf{M}^{-1}(\mathbf{x}) \quad (2.56)$$

Note that if the linear basis function itself were to be interpolated, i.e., $f(\mathbf{x}) = \mathbf{P}(\mathbf{x})$, eqs.(2.51) and (2.52),

$$\begin{aligned} \mathbf{P}^h(\mathbf{x}) &= \int_{R_x} C(a, \mathbf{x}, \mathbf{y}) W \left(\frac{\mathbf{y} - \mathbf{x}}{a} \right) \mathbf{P}(\mathbf{y}) dR_y \\ &= \mathbf{P}(\mathbf{x}) \mathbf{M}^{-1}(\mathbf{x}) \int_{R_x} \mathbf{P}^T(\mathbf{y}) \mathbf{P}(\mathbf{y}) W \left(\frac{\mathbf{y} - \mathbf{x}}{a} \right) dR_y \\ &= \mathbf{P}(\mathbf{x}) \mathbf{M}^{-1}(\mathbf{x}) \mathbf{P}(\mathbf{x}) \mathbf{M}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \end{aligned} \quad (2.57)$$

because of eq.(2.49). This means the basis function is reproduced, and also the satisfaction of the consistency condition up to the highest order used in $\mathbf{P}(\mathbf{x})$. That is, in the case a first-order polynomial is selected,

$$\begin{aligned} \int_{R_x} C(a, \mathbf{x}, \mathbf{y}) 1 \cdot W \left(\frac{\mathbf{y} - \mathbf{x}}{a} \right) dR_y &= 1 \\ \int_{R_x} C(a, \mathbf{x}, \mathbf{y}) y \cdot W \left(\frac{\mathbf{y} - \mathbf{x}}{a} \right) dR_y &= x \end{aligned} \quad (2.58)$$

The interpolation for the derivative $(\partial u(\mathbf{x}))^h$ is given as

$$(\partial f(\mathbf{x}))_i^h = \sum_j \partial N_j(\mathbf{x}) f_j \quad (2.59)$$

where ∂ can be replaced with either of ∂_x , ∂_y and ∂_z . In the original RKPM, the derivative of shape function $\nabla N_j(\mathbf{x})$ is obtained by directly differentiating eq.(2.54) as

$$\begin{aligned}
\partial \mathbf{M}(\mathbf{x}) = & -\frac{1}{a} \sum_j \left\{ \partial \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) W \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \right. \\
& + \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \partial \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) W \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \\
& \left. + \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \partial W \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \right\} V_i
\end{aligned} \tag{2.60}$$

$$\partial \mathbf{M}^{-1}(\mathbf{x}) = -\mathbf{M}^{-1}(\mathbf{x}) \partial \mathbf{M}(\mathbf{x}) \mathbf{M}^{-1}(\mathbf{x}) \tag{2.61}$$

$$\partial \mathbf{b}(\mathbf{x})^T = \mathbf{h}(\mathbf{0})^T \partial \mathbf{M}^{-1}(\mathbf{x}) \tag{2.62}$$

And finally,

$$\begin{aligned}
\partial N_j(\mathbf{x}) = & \left\{ \partial \mathbf{b}(\mathbf{x})^T \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) W \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \right. \\
& - \frac{1}{a} \mathbf{b}(\mathbf{x})^T \partial \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) W \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \\
& \left. - \frac{1}{a} \mathbf{b}(\mathbf{x})^T \mathbf{P} \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \partial W \left(\frac{\mathbf{x}_j - \mathbf{x}}{a} \right) \right\} V_i
\end{aligned} \tag{2.63}$$

However, this is not the most efficient way since it involves multiple equations along with calculating the inversion of the derivative of the matrix for all particles. Later, an alternative with better efficiency was developed to obtain the derivative, known as the gradient reproducing kernel collocation method (GRKCM) [30]. In GRKCM, the reproducing kernel approximation is given in the same manner as RKPM as

$$f(\mathbf{x}) = \sum_j N_j f_j \tag{2.64}$$

where

$$N_j = \mathbf{P}^T(\mathbf{0}) \mathbf{M}^{-1} \mathbf{P} \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) W \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) \tag{2.65}$$

Similarly to eq.(2.49), but here M is in the discretised form as

$$\mathbf{M}(\mathbf{x}) = \sum_j \mathbf{P}^T \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) \mathbf{P} \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) W \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) \tag{2.66}$$

Note that these equations do not include the particle volume V_j unlike the RKPM equivalents eqs.(2.49), (2.54). The derivative of a function in GRKCM is expressed in the same way as eq.(2.59), but the difference is that the derivative of the shape function, in the case of the first order-derivative, is now assumed to be

$$\partial N_j = \mathbf{P}^T \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) \partial \mathbf{b}'(\mathbf{x}) W \left(\frac{\mathbf{x} - \mathbf{x}_j}{a} \right) \quad (2.67)$$

The coefficient $\mathbf{b}'(\mathbf{x})$ is obtained through the reproducing conditions for the gradient of the function rather than the original function as

$$\sum_j \partial W_j \mathbf{x}_j^\alpha = \partial \mathbf{x}^\alpha \quad (2.68)$$

where α is the degree of moment. Here, the derivation of the coefficient $\partial \mathbf{b}'(\mathbf{x})$ and the resultant ∂N_j is described for the 2D case with the first-order polynomial, i.e. $\mathbf{P}(\mathbf{x}) = [1 \ x \ y]^T$. There is a way to determine the coefficients for general polynomials and also for higher-order derivatives, but as shown later, this study only uses the first-order polynomial. The reproducing condition eq.(2.68) can be expanded as

$$\begin{aligned} \sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) &= 0, \sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) x_j = 1, \sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) y_j = 0 \\ \sum_j \frac{\partial}{\partial y} N_j(\mathbf{x}) &= 0, \sum_j \frac{\partial}{\partial y} N_j(\mathbf{x}) x_j = 0, \sum_j \frac{\partial}{\partial y} N_j(\mathbf{x}) y_j = 1 \end{aligned} \quad (2.69)$$

For, the x-direction derivative, multiplying eq.(2.69-1) by x and subtracting eq.(2.69-2) yields

$$\sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) (x - x_j) = -1 \quad (2.70)$$

Multiplying eq.(2.69-1) by y and subtracting eq.(2.69-3) yields

$$\sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) (y - y_j) = 0 \quad (2.71)$$

Similarly, for the y-direction derivative,

$$\begin{aligned}
\sum_j \frac{\partial}{\partial y} N_j(\mathbf{x})(x - x_j) &= 0 \\
\sum_j \frac{\partial}{\partial y} N_j(\mathbf{x})(y - y_j) &= -1
\end{aligned} \tag{2.72}$$

Therefore, the reproducing conditions eq.(2.69) can be rewritten as

$$\begin{aligned}
\sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) &= 0, \sum_j \frac{\partial}{\partial x} N_j(\mathbf{x})(x - x_j) = -1, \sum_j \frac{\partial}{\partial x} N_j(\mathbf{x})(y - y_j) = 0 \\
\sum_j \frac{\partial}{\partial y} N_j(\mathbf{x}) &= 0, \sum_j \frac{\partial}{\partial y} N_j(\mathbf{x})(x - x_j) = 0, \sum_j \frac{\partial}{\partial y} N_j(\mathbf{x})(y - y_j) = -1
\end{aligned} \tag{2.73}$$

Using $\mathbf{P}(\mathbf{x}) = [1 \ x \ y]^T$, these can be rewritten as

$$\begin{aligned}
\sum_j \frac{\partial}{\partial x} N_j(\mathbf{x}) \mathbf{P}(\mathbf{x} - \mathbf{x}_j) &= -\frac{\partial}{\partial x} \mathbf{P}(\mathbf{0}) \\
\sum_j \frac{\partial}{\partial y} N_j(\mathbf{y}) \mathbf{P}(\mathbf{x} - \mathbf{x}_j) &= -\frac{\partial}{\partial y} \mathbf{P}(\mathbf{0})
\end{aligned} \tag{2.74}$$

Substituting eqs.(2.67) and (2.49) yields

$$\begin{aligned}
\mathbf{M}(\mathbf{x}) \frac{\partial}{\partial x} \mathbf{b}'(\mathbf{x}) &= -\frac{\partial}{\partial x} \mathbf{P}(\mathbf{0}) \\
\mathbf{M}(\mathbf{x}) \frac{\partial}{\partial y} \mathbf{b}'(\mathbf{x}) &= -\frac{\partial}{\partial y} \mathbf{P}(\mathbf{0})
\end{aligned} \tag{2.75}$$

Therefore,

$$\begin{aligned}
\frac{\partial}{\partial x} \mathbf{b}'(\mathbf{x}) &= -\mathbf{M}(\mathbf{x})^{-1} \frac{\partial}{\partial x} \mathbf{P}(\mathbf{0}) \\
\frac{\partial}{\partial y} \mathbf{b}'(\mathbf{x}) &= -\mathbf{M}(\mathbf{x})^{-1} \frac{\partial}{\partial y} \mathbf{P}(\mathbf{0})
\end{aligned} \tag{2.76}$$

Finally, the derivative of the shape function in eq.(2.67) is given as

$$\begin{aligned}\frac{\partial}{\partial x}N_j &= -\frac{\partial}{\partial x}\mathbf{P}(\mathbf{0})^T\mathbf{M}(\mathbf{x})^{-1}\frac{\partial}{\partial x}\mathbf{P}\left(\frac{\mathbf{x}-\mathbf{x}_j}{a}\right) \\ \frac{\partial}{\partial y}N_j &= -\frac{\partial}{\partial y}\mathbf{P}(\mathbf{0})^T\mathbf{M}(\mathbf{x})^{-1}\frac{\partial}{\partial y}\mathbf{P}\left(\frac{\mathbf{x}-\mathbf{x}_j}{a}\right)\end{aligned}\tag{2.77}$$

$(\partial/\partial z)N_j$ for the 3D case can be obtained in the same way by adding conditions regarding z in eq.(2.69).

Now, the only matrix involved here is $\mathbf{M}^{-1}(\mathbf{x})$, which would have been already computed in computing the shape function N_j , and therefore this method is significantly faster than the RKPM as in eqs. (2.60) - (2.63). In addition to this increased efficiency, GRKCM still possesses similar convergence rates of L_2 norm to RKPM, and even in some cases, the solution is less oscillatory [30].

The vector of linearly independent functions \mathbf{P} has multiple choices, which is often a set of polynomial functions as in table 2.4. Theoretically, higher-order polynomials can satisfy higher-order consistency conditions, but earlier studies conclude that first-order polynomials are more stable than higher-order ones, and also that they are sufficiently accurate for most purposes[28], [100]. Furthermore, the use of higher-order polynomials would result in drastically larger matrices, which would inevitably increase the computational cost.

Table 2.4: Commonly-used sets of linear functions $\mathbf{P}(\mathbf{x})$

Type	Dimension	Linear function $\mathbf{P}(\mathbf{x})$
Constant	1D	$[1]^T$
	2D	$[1]^T$
	3D	$[1]^T$
1st order	1D	$[1 \ x]^T$
	2D	$[1 \ x \ y]^T$
	3D	$[1 \ x \ y \ z]^T$
2nd order	1D	$[1 \ x \ x^2]^T$
	2D	$[1 \ x \ y \ x^2 \ y^2 \ xy]^T$
	3D	$[1 \ x \ y \ z \ x^2 \ y^2 \ z^2 \ xy \ yz \ zx]^T$

Now, the Cauchy momentum equation eq.(2.2) is discretised for RKPM [28]. In RKPM or GRKCM, the discretisation of Cauchy momentum equation eq.(2.2) requires the weak formulation of the equation, as in [28]

$$\int_{\Omega} \left\{ \rho \left(\frac{D\mathbf{u}}{Dt} - \mathbf{g} \right) v + \nabla v \cdot \boldsymbol{\sigma} \right\} - \oint_{\partial\Omega} n \cdot \boldsymbol{\sigma} v dS = 0 \quad (2.78)$$

where v is a trial function, Ω is the integration region. Here, for brevity, only free boundary cases are considered, and therefore the second term vanishes. Using the shape function N as trial function, the discretised form yields

$$\sum_j \left\{ \rho_j (\dot{\mathbf{u}}_j^h - \mathbf{g}) N_i(\mathbf{x}_j) + \nabla N_i(\mathbf{x}_j) \cdot \boldsymbol{\sigma}_j \right\} V_j = 0 \quad (2.79)$$

where

$$\dot{\mathbf{u}}_j^h = \sum_k \dot{\mathbf{u}}_j^k N_k(\mathbf{x}_j) \quad (2.80)$$

Therefore, eq.(2.79) can be rewritten as

$$\sum_j \left\{ \sum_k N_i(\mathbf{x}_k) N_j(\mathbf{x}_k) \rho_k \right\} (\dot{\mathbf{u}}_j - \mathbf{g}) = -\nabla N_i(\mathbf{x}_j) \cdot \boldsymbol{\sigma}_j V_j \quad (2.81)$$

Considering the mass matrix $\mathbf{M}_{ij} = \sum_k N_i(\mathbf{x}_k) N_j(\mathbf{x}_k) \rho_k$ and the lumped mass matrix $\tilde{\mathbf{M}}_{ij} = \delta_{ij} \mathbf{M}_{ik} = \delta_{ij} \sum_l N_i(\mathbf{x}_l) N_l(\mathbf{x}_l) \rho_l V_l$, the final discretised form yields

$$\dot{\mathbf{u}}_i = -\frac{\sum_j \nabla N_i(\mathbf{x}_j) \cdot \boldsymbol{\sigma}_j V_j}{\sum_j N_i(\mathbf{x}_j) \rho_j V_j} + \mathbf{g} \quad (2.82)$$

Now, nodal integration methods like this are known to suffer from instability due to vanishing derivatives of shape functions at the nodes, and remedies have been proposed as in [27]. However, a far simpler alternative of using artificial viscosity [28] will be used here as

$$\tilde{\mathbf{u}}_i = \alpha_b \mathbf{u}_i + (1 - \alpha_b) \mathbf{u}^h(\mathbf{x}_i) \quad (2.83)$$

where α_b is a constant, $\tilde{\mathbf{u}}_i$ is the velocity after the artificial viscosity is applied.

2.4.3 Other corrected methods

Numerous methods have been proposed other than RKPM that address the consistency issue of SPH. Some of them will be briefly reviewed here to assess the suitability of RKPM for grease CFD.

One of them is corrective SPH(CSPH) [26], which is probably the simplest amongst the corrected methods, but it can reduce tensile instability that traditional SPH suffers from. However, there is a drawback that this method does not apply to problems with discontinuities [173]. In CSPH, a function f is interpolated as

$$f_i = \frac{\sum_j \frac{m_j}{\rho_j} f_j W_{ij}}{\sum_j \frac{m_j}{\rho_j} W_{ij}} \quad (2.84)$$

The derivative of a function is obtained by solving

$$A_{\alpha\beta i} f_{\beta i} = F_{\alpha i} \quad (2.85)$$

where

$$\begin{aligned} A_{\alpha\beta i} f_{\beta i} &= \sum_j (\beta - \beta_i) W_{ij,\alpha} \frac{m_j}{\rho_j} \\ F_{\alpha i} &= \sum_j (f - f_i) W_{ij,\alpha} \frac{m_j}{\rho_j} \end{aligned} \quad (2.86)$$

Modified SPH(MSPH) is an improved version of CSPH [193]. This method has a better accuracy around the boundary than CSPH, but comes with a higher computational cost of solving larger matrices.

[194] proposed symmetric SPH(SSPH), where unlike RKPM, the derivative of a function is obtained without using the derivative of the kernel, which enables a more flexible choice of the kernel. This method requires solving a local matrix, which is symmetric as opposed to MSPH. Therefore, the method is reported to be more efficient than MSPH along with a higher accuracy. One drawback is that H^1 norm of error is larger than RKPM.

Another method is the finite particle method(FPM) [99]. This method uses the Taylor expansion on the original SPH approximation, which was also the case with corrective SPH. However, FPM has a higher order of consistency than corrective SPH. A function is interpolated as

$$f_i = \left| \begin{array}{cc} \sum_j f_j W_{ij} \frac{m_j}{\rho_j} & \sum_j (r_j^\alpha - r_i^\alpha) W_{ij} \frac{m_j}{\rho_j} \\ \sum_j f_j W_{ij,\beta} \frac{m_j}{\rho_j} & \sum_j (r_j^\alpha - r_i^\alpha) W_{ij,\beta} \frac{m_j}{\rho_j} \end{array} \right| \Bigg/ \left| \begin{array}{cc} \sum_j W_{ij} \frac{m_j}{\rho_j} & \sum_j (r_j^\alpha - r_i^\alpha) W_{ij} \frac{m_j}{\rho_j} \\ \sum_j W_{ij,\beta} \frac{m_j}{\rho_j} & \sum_j (r_j^\alpha - r_i^\alpha) W_{ij,\beta} \frac{m_j}{\rho_j} \end{array} \right| \quad (2.87)$$

The derivative of a function is

$$f_{i,\alpha} = \left| \begin{array}{cc} \sum_j f_j W_{ij} \frac{m_j}{\rho_j} & \sum_j W_{ij} \frac{m_j}{\rho_j} \\ \sum_j f_j W_{ij,\beta} \frac{m_j}{\rho_j} & \sum_j W_{ij,\beta} \frac{m_j}{\rho_j} \end{array} \right| \Bigg/ \left| \begin{array}{cc} \sum_j (r_j^\alpha - r_i^\alpha) W_{ij} \frac{m_j}{\rho_j} & \sum_j W_{ij} \frac{m_j}{\rho_j} \\ \sum_j (r_j^\alpha - r_i^\alpha) W_{ij,\beta} \frac{m_j}{\rho_j} & \sum_j W_{ij,\beta} \frac{m_j}{\rho_j} \end{array} \right| \quad (2.88)$$

In comparison to these corrected methods, RKPM possesses relatively high accuracies although the computational cost is also high. However, since the goal of this work is the simulation of grease in the context of engineering rather than just creating graphically compelling results as in existing researches, highly accurate RKPM would fit the purpose.

2.5 Ill-distributed particles

Despite RKPM being a highly accurate method that avoids the tensile instability issue, there is another type of instability issue that cannot be solved by RKPM. That is ill-distributed particles that occur under certain circumstances, which have been tackled by some studies. This study also faced the ill-distributed particle issue occasionally leading to an explosion of particles. Therefore, this subchapter will discuss the issue and remedies.

[78] and [161] studied ill-distributed particles that arises when tension is concerned. In such a case, tensile stress combined with certain smoothing functions causes instability. [78] proposed a normalised smoothing function(NSF) algorithm that mitigates the instability.

[183] discussed the problem of ill-distributed particles in the context of incompressibility. As mentioned in the previous chapter, there are two ways to handle incompressibility, one with the divergence-free condition, and the other with the density-invariance condition. The authors stated that the divergence-free condition can suffer from ill-distributed particles(Fig. 2.10). They reported that the Density-Invariant condition does not exhibit this problem, but numerical accuracy is compromised.

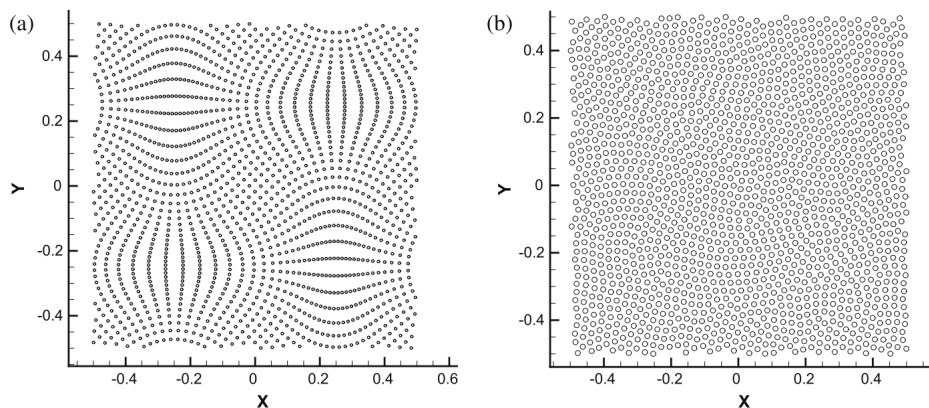


Figure 2.10: TaylorGreen vortices. Left: Particle clustering with Divergence-Free, right: uniform distribution with Divergence-Free and particle shifting [183]

[80] also investigated the ill-distributed particle problem, but stated that this issue arises with the Density-Invariant condition rather than the Divergence-Free as opposed to [183](Fig. 2.11). Therefore, the problem of ill-distributed particles appears to arise in either of the conditions, and whenever it does, an appropriate countermeasure would be necessary.

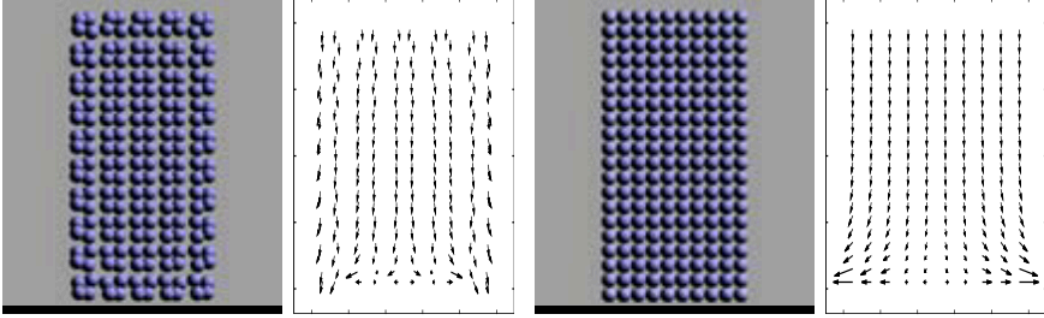


Figure 2.11: Particle clustering under tension. Left: Density-Invariant, right: Divergence-Free [80]

As a remedy, [183] developed a scheme called particle shifting that achieves both accuracy and stability. The position of particles is slightly shifted so that particles are distributed more evenly(Fig. 2.10).

Based on this particle shifting scheme, [95] developed a method that is more suitable for free surface flows. This method is based on Fick's law of diffusion, and particles are shifted so that the particle concentration averages out. The flux \mathbf{J} on Fick's law can be expressed as

$$\mathbf{J} = -\mathcal{D}'\nabla C \quad (2.89)$$

where C is particle concentration, and \mathcal{D}' is an arbitrary diffusion coefficient. The particle shifting $\delta\mathbf{r}_{s,i}$ reads

$$\delta\mathbf{r}_{s,i} = -\mathcal{D}(\nabla C)_i \quad (2.90)$$

where \mathcal{D} is a new diffusion coefficient. The particle concentration is

$$C_i = \sum_j V_j W_{ij} \quad (2.91)$$

where W_{ij} is the SPH kernel between particles i and j . The concentration gradient $(\nabla C)_i$ is

$$(\nabla C)_i = \sum_j V_j \nabla W_{ij} \quad (2.92)$$

The direct use of eq. (2.92) would have a problem of the gradient decreasing as particles get closer, which would not help mitigate the particle clustering. To resolve this, a trick is employed to ensure that the kernel gradient increases as particles get closer.

$$(\nabla C)_i = \sum_j V_j (1 + f_{ij}) \nabla W_{ij} \quad (2.93)$$

where the factor f_{ij} is

$$f_{ij} = R \left(\frac{W_{ij}}{W(dx)} \right)^n \quad (2.94)$$

where R is a constant. In this way, more uniformly distributed particles can be obtained. Physical variables, such as velocity and strain are updated according to the shifting to reduce the numerical error. Using the Taylor series, an arbitrary physical variable ϕ is corrected as

$$\phi_{i'} = \phi_i + \delta \mathbf{r}_{ii'} \cdot (\nabla \phi)_i + \mathcal{O}(\delta r_{ii'}^2) \quad (2.95)$$

[59] stated that ill-distributed particles result in particles clustering and the lack of stretching. They developed a particle resampling scheme. In this method, particles merge when they get close, and new particles are inserted where particles are sparse. This way, the authors successfully reproduced the fluid stretching using SPH(Fig. 2.12).

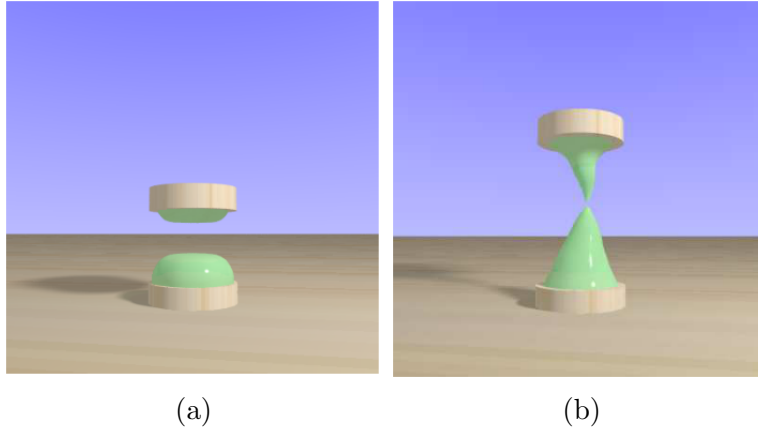


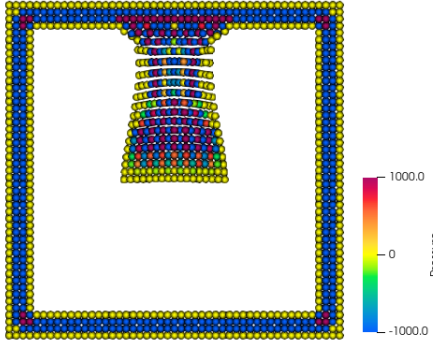
Figure 2.12: Separation of fluid.
 (a) without re-sampling, (b) with re-sampling [59]

The ill-distributed particles are also found to be relevant in this study, particularly when the stretching motion is concerned, as in the plate separation test. As will be discussed later, the particle shift method is found to be effective.

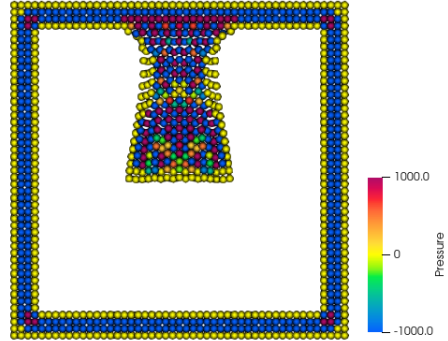
A benchmark was performed where a fluid is hung from the ceiling, which is expected to stretch. In the case of particle methods, one factor that enables fluid stretching is the negative pressure that starts to develop between the fluid and the ceiling, which pulls the fluid towards the centre. However, simply employing negative pressure does not produce the expected result (Fig. 2.13(a)). When position-based density is employed, negative pressure firstly pulls particles sideways. This causes the pressure to rise despite the particles are sparsely distributed vertically, which eventually yields little negative pressure to pull the fluid in the vertical direction. Here, to address this issue, particle shifting is employed rather than particle re-sampling method for simplicity.

Fig. 2.13(b) demonstrates the effectiveness of particle shifting. This allows more even distribution, avoids the issue, and therefore allows the fluid to stretch. Moreover, this method produced similar results to the particle re-sampling however with easier implementation. Fig. 2.23 and Table 2.5 show the particle file and conditions used in this benchmark respectively.

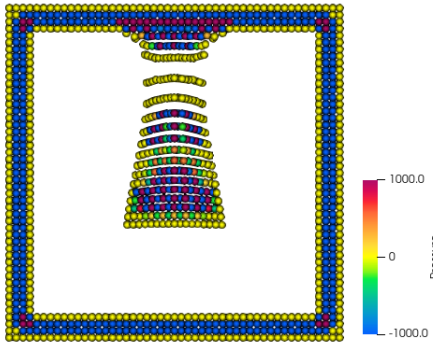
Time: 0.21(s)



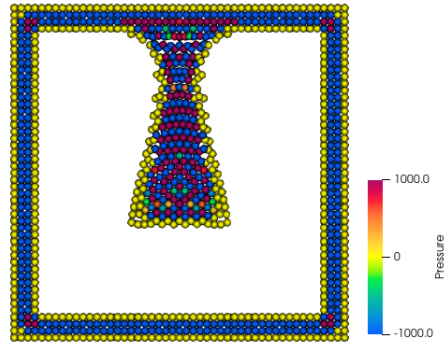
Time: 0.21(s)



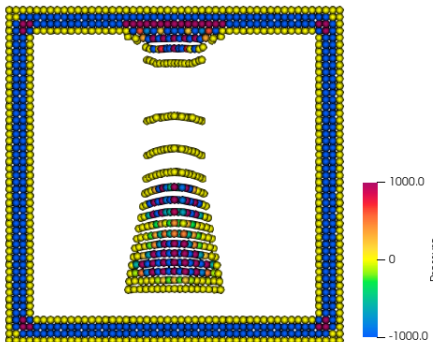
Time: 0.31(s)



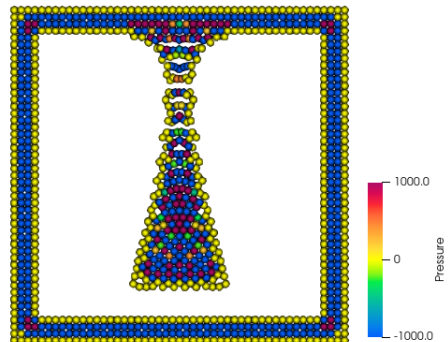
Time: 0.31(s)



Time: 0.42(s)



Time: 0.42(s)



(a) unnatural cut-off w/o shifting

(b) stretching w/ shifting

Figure 2.13: Effect of particle shifting in a chronological order

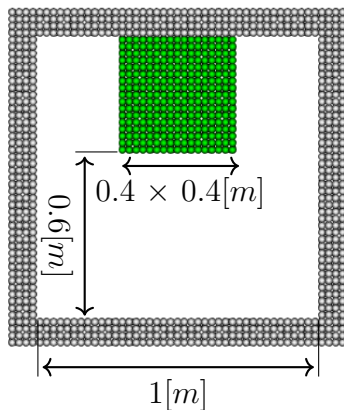


Figure 2.14: Particle file used in shifting benchmark

Table 2.5: Parameters used in the particle shifting benchmark

	w/o shift	w/ shift
particle size [m]	2.5e-2	
kernel radius[m]	6.25e-2	
particle count	1,009	
time step [s]	2.5e-4	
velocity blending coefficient	0.8	
EOS coefficient B	3.0e+3	
shift coefficient	0.0	-0.1
surface particle volume coefficient	1.07	
surface particle number of neighbours	90	
surface tension coefficient [N/m]	0.0	
density ρ [kg/m ³]	1000	
viscosity μ_v [Pa · s]	100	

2.6 Surface cohesion

For the fluid stretching using particle methods, surface tension is found to be helpful. In fact, some studies, such as [7], [72], [120] introduced surface tension in producing the stretch. As seen later in this chapter, this study also finds surface cohesion effective, which is part of surface tension. The method

used in this study is based on [4], which was originally developed to handle large surface tensions in a more realistic manner than the previous methods. The model consists of the cohesion term that helps particles stick together and the surface area minimisation term that makes the surface round. The cohesion term per neighbour particle's contribution is

$$\mathbf{F}_{i \leftarrow j}^{cohesion} = \gamma m_i m_j C(|\mathbf{x}_i - \mathbf{x}_j|) \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|} \quad (2.96)$$

where γ and C are surface tension coefficient and a spline function respectively.

$$C(r) = \frac{32}{\pi h^9} \begin{cases} (h-r)^3 r^3 & 2r > h \wedge r \leq 1 \\ 2(h-r)^3 r^3 - \frac{h^6}{64} & r > 0 \wedge 2r \leq h \\ 0 & \text{otherwise} \end{cases} \quad (2.97)$$

Using the gradient of the smoothing function, surface area minimisation term per neighbour particle's contribution is

$$\mathbf{n}_i = h \sum_j \frac{m_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|) \quad (2.98)$$

$$\mathbf{F}_{i \leftarrow j}^{curvature} = -\gamma m_i (\mathbf{n}_i - \mathbf{n}_j) \quad (2.99)$$

Considering this mechanism, this study employs the aforementioned model [4], but only with the cohesion term without the surface area minimisation term. It makes surfaces rounder, and therefore, in the case of fluid stretching, it would flatten the peaks after the break-up. However, as discussed in the later chapter, grease largely retains its peaks after break-ups. The same logic applies to other surface tension models. If applied to stretching fluids, they may yield longer break-off lengths, but afterwards they would flatten the peaks.

Here, the effect of surface cohesion is investigated under two conditions; one without elasticity, and another with elasticity as listed in Table 2.6. Without surface cohesion, particularly in the 3D cases, the particle distribution under cohesion can be unnaturally sparse as in Figs. 2.15 (a), 2.16(a). This adversely affects the result in two ways. The first is that more particles are categorised as surface since they have only a small number of neighbour particles, which in turn prevents negative pressure and, hence finally less pronounced stretching. The other is that these unnaturally sparse particles, which might be considered to be ill-distributed, can cause instability. In the worst case, the particles completely explode(Fig.2.16(a)). These are the reasons why surface cohesion could be useful. It makes particles stick together,

Table 2.6: Parameters used in the surface cohesion benchmark

Fig.	2.15(a)	2.15(b)	2.16(a)	2.16(b)
particle size [m]	2.5e-3			
kernel radius [m]	6.25e-3			
particle count	5,001			
time step [s]	2e-5			
velocity blending coefficient	0.8			
EOS coefficient B	3.0e+3			
shift coefficient	-0.1			
surface particle volume coefficient	1.07			
surface particle number of neighbours	90			
surface cohesion coefficient [N/m]	0.0	0.1	0.0	0.1
plate speed [m/s]	0.01			
density $\rho[kg/m^3]$	1000			
viscosity $\mu_v[Pa \cdot s]$	100			
Young's modulus $E [N/m^2]$	0.0		100	
yield strain ϵ_Y	-		0.3	
decay rate $\alpha[/math>/s]$	-		100	

negative pressure develops, and finally, it allows the fluid to stretch. In both cases, introducing surface cohesion was indeed found effective in achieving the aim(Figs. 2.15(b), 2.16(b)). The particle distribution became more natural with a thin stripe of fluid forming at the centre. Moreover, surface cohesion stabilised the simulation with elasticity. Despite this effectiveness, attention is needed since the value of the surface coefficient is arbitrarily chosen. Furthermore, this surface cohesion model itself involves SPH-specific functions, and it makes it very challenging to determine the coefficient from the value obtained in the real world. Therefore, this model is purely artificial, and does not reflect the real values of surface cohesion of the material. Fig. 2.17 and Table 2.6 show the particle file and conditions used in this benchmark respectively.

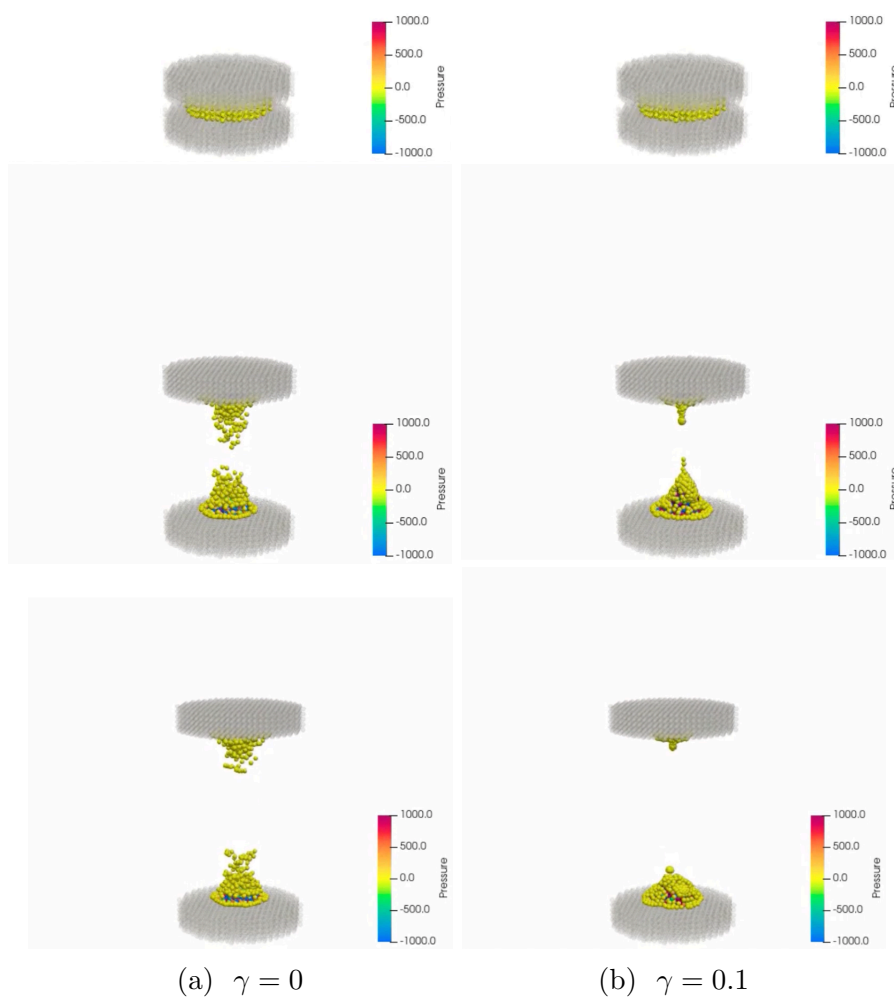


Figure 2.15: Effect of surface cohesion, $E = 0$ in a chronological order

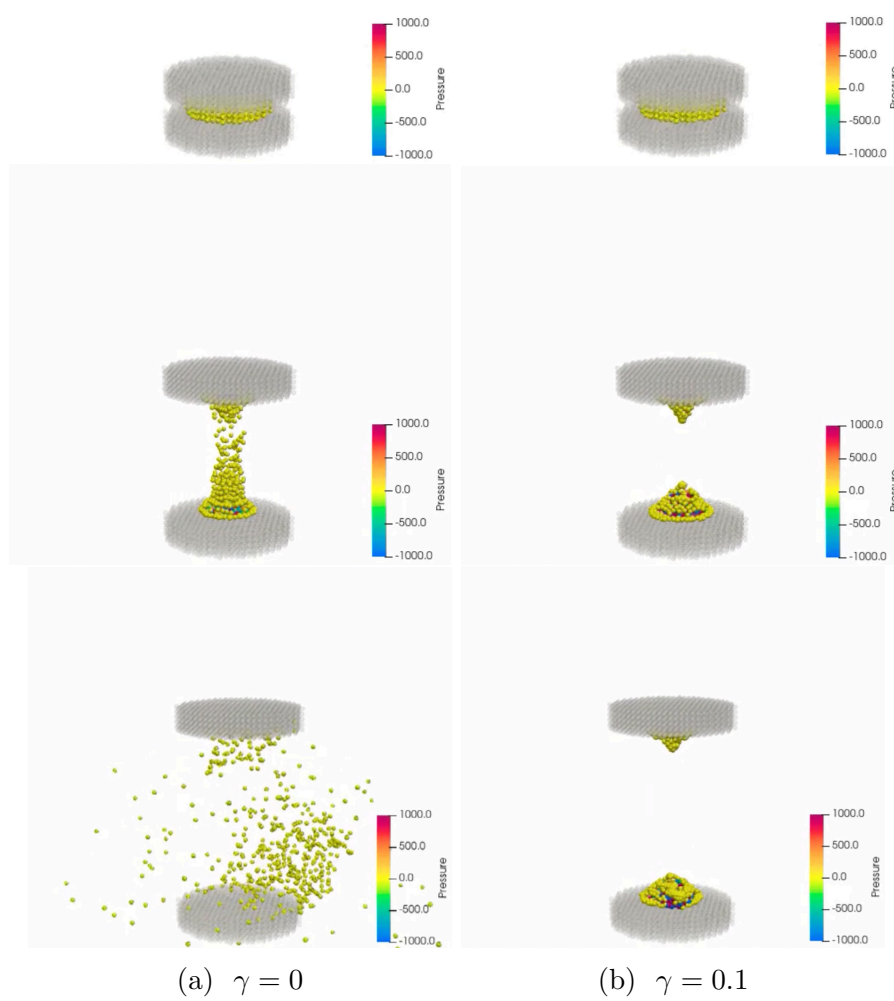


Figure 2.16: Effect of surface tension, $E = 100$ in a chronological order

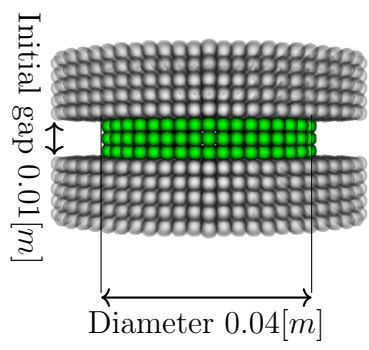


Figure 2.17: Particle file used in Figs. 2.15, 2.16

2.7 GPU parallelisation

2.7.1 GPU computing overview

To achieve a fast simulation of grease, this study incorporates GPU computing to speed up the simulation. Firstly, a brief description of the situation of GPU computing follows. GPUs have rapidly evolved from configurable graphics processors to programmable graphics processors, and today can execute thousands of parallel threads to solve large-scale problems faster than ever whilst being cost-effective [127]. This is because in the early stage, GPU programming was challenging, but later it became more manageable when standard scalar languages such as C were supported.

NVIDIA's Compute Unified Device Architecture(CUDA), an architecture installed on the company's graphics cards facilitated GPU computing even further, allowing writing codes in widely used programming languages, such as C, C++, Fortran, and so on [127]. CUDA programs can be executed on a GPU of any size, automatically adjusting to the available size of the GPU. There are many applications both in academic research and the industry using CUDA, such as quantum chemistry, molecular dynamics, CFD, and finite element analysis.

There have been reports on the GPU acceleration, but the speed-up gains would likely be even greater than they claim. This is because some of them were published some time ago, with [45] in 2004, [60] in 2007, [63] in 2009, [56] in 2010, and [128] in 2015, during which the GPU has drastically evolved. In the case of NVIDIA GPUs, the throughput of Ampere in 2020 is over 20 times more than Fermi in 2010 [38](Fig. 2.18). Therefore, if the same methods of the GPU works mentioned were to be employed, that would be even faster, or instead, the resource can be allocated to achieve higher resolutions with an equivalent runtime. Furthermore, the difference in theoretical peak performance in terms of gigaflops and bandwidth for the fastest available NVIDIA GPUs and Intel CPUs has been growing larger [19].

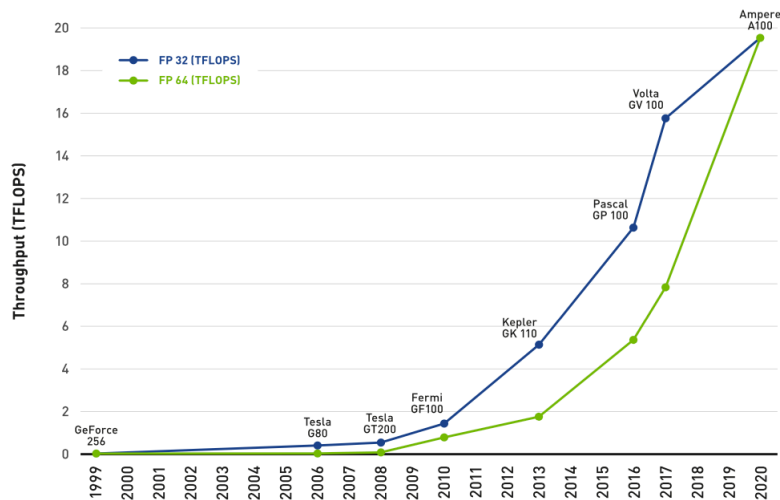


Figure 2.18: Evolution of single GPU performance in throughput [38]

2.7.2 SPH on the GPU

GPU computing has also been exploited in SPH. Among the relevant studies, [60] appears to be the first to implement the entire procedure of SPH on the GPU as opposed to the previous work which partially uses the CPU for mainly neighbour search. Avoiding the use of a CPU enables a drastic speed-up since the communication between the GPU and CPU is slow. However, [60] was before the introduction of CUDA, where great knowledge of computer graphics was needed in implementation. Later, [63] implemented SPH using CUDA, showcasing the ease of implementation and the speed-up by CUDA, even realising real-time simulation in certain cases. The NVIDIA GPUs have the advantages of a large number of cores and high memory bandwidth in relation to the CPU.

The authors also investigated the scale-up by a more powerful GPU, and concluded that whilst the neighbour search did not enjoy a linear scale-up, the force computation and Euler step part did. [87] also implemented SPH on the GPU with CUDA, and achieved a speed-up from earlier studies. [182] simulated shallow wave equations on the GPU, and realised a 7-10 time speed-up against an efficient quad-core CPU enabled by the combination of the quad-tree neighbour search algorithm and the z-curve sorting. [56] realised a real-time simulation on the GPU with better stability and resolution than previous studies. This was enabled by the combination of fast neighbour search(Z-indexing) and a new efficient rendering pipeline. [128] implemented predictive-corrective incompressible SPH(PCISPH) real-time on the GPU,

along with a physical improvement by eliminating the artefact that arises from compressibility. The computational intensity of PCISPH was alleviated by the extensive use of fast read-only memory. All these works of SPH on the GPU use fully explicit schemes with no need to deal with global matrices, some of which even can run in real-time.

In contrast to the explicit schemes, there also exists implicit SPH on the GPU. Implicit schemes inevitably incur solving global (although sparse) matrices, which is nowadays possible due to the linear solver for the GPU, such as AmgX by NVIDIA [137]. NVIDIA investigated the speed-up by the GPU in the case of ANSYS Fluent, a commercial CFD program of fully implicit FVM, and reported a two to three-time speed-up from the CPU version. In particle methods, implicit schemes can be applied to pressure, viscous force, or both. [55] implemented Implicit Incompressible SPH(IISPH), an implicit pressure scheme on the GPU. Whilst this was faster than the CPU version, more standard explicit SPH on the GPU turned out to be about 2.3 times faster than IISPH on the GPU. [31] also performed IISPH on the GPU, reporting better performance than the 16-thread CPU equivalent. Compared to explicit SPH, the advantage of the implicit scheme is the noise-free natural pressure field, but there is no performance comparison against explicit SPH on the GPU. The authors also mentioned the limitation on the particle count due to the limited GPU memory. As opposed to these implicit pressure schemes, [190] handled viscous force implicitly with an explicit pressure scheme. This removed the stability constraint on the time step size arising from high viscosities, and in the best-case scenario, it was reported to be 43 times faster than the explicit GPU version although speed-up by the implicit scheme can be achieved only for extremely high viscosities.

Overall, those implicit schemes can be faster than the CPU equivalent, or can produce more natural results than the GPU explicit schemes, but they are not without difficulty. In terms of speed, they are behind the explicit GPU counterparts with minor exceptions reported by [190], and in other cases, the implicit schemes are slower, or no comparison is made against explicit schemes. Moreover, the implementation becomes significantly more complicated due to the memory management and the appropriate choice of linear library and its handling. If implicit pressure or viscosity were to be applied to corrected methods, such as CSPH, SSPH and RKPM, the matrices would be asymmetric unlike SPH, which would further increase memory usage and runtime. Further performance improvements in relation to explicit schemes do not appear promising either due to the lack of recent developments of linear solvers. Two iterative solvers have been used in SPH on the GPU, one of which is ViennaCL, used by [31], preferring its speed. Nonetheless, [142] published in 2016 is the first and last publication about

ViennaCL, and no update has been released until the time of writing. The other iterative solver is AmgX, used by [132] for Eulerian incompressible smoothed SPH. This solver is two to five times faster than a competitive implementation on the CPU [122], but similarly to ViennaCL, no update has been made since its publication in 2015. Therefore, the expected speed-up of linear solvers themselves has not happened for several years, and there have not been any new solvers that outperform those. Therefore, the speed of implicit schemes on the GPU is not expected to surpass explicit schemes for the foreseeable future. For these reasons, this study employs an explicit scheme for both pressure and viscosity.

2.7.3 GPU implementation in this study

This chapter discusses the technical points of the GPU implementation in this study. As discussed in the previous chapter, GPUs can achieve significant speed-ups, but their unique architecture and complicated memory structure necessitate carefully planned coding strategies to realise optimal memory use and exploit the potential of GPUs. The code was designed to minimise the global memory access, and maximise the use of register, the latter of which is significantly faster than the former. The entire process of simulation is performed on the GPU apart from the initialisation and result dumping part, where the data is output to text files. Here are the considerations taken for optimal performance.

- **Kernel fusion**

As shown later, the most time-consuming part of this implementation is the force computation. This was accelerated by kernel fusion, which means merging independent kernels into one. A kernel is a function that is called from the CPU and executed on the GPU threads concurrently. There are three kinds of fusions, inner thread fusion, inner thread block fusion, and inter-thread block fusion [169]. This study only uses inner thread fusion, where the same kernels are applied to all threads, such as neighbour search, RKPM coefficient computation, pressure computation, and so on. Effective use of inner thread fusion can reduce reading from and writing to the global memory repeatedly, where many of the variables are stored on the register to be accessed multiple times during a single kernel execution. In some cases, as a by-product, kernel fusion is also reported to reduce energy consumption [169]. One such example of the register usage is the neighbour particle list and RKPM coefficients, which are accessed multiple times within the force computation kernel, and they do not need storing in

the global memory as they are not used elsewhere. Therefore, merging kernels and storing them only on the register saves unnecessary global memory accesses and memory consumption. Only physical variables are written to the global memory by the force computation, which is necessary to output the simulation results. Despite the effectiveness of kernel fusion, force computation and time integration still need to be executed in separate kernels. This is because during time integration, velocity and position are updated. Therefore, if time integration were to happen in the same kernel, this would cause a data race, where some particles would already have new values \mathbf{x}_j^{n+1} and \mathbf{u}_j^{n+1} whereas others particles old values \mathbf{x}_j^n and \mathbf{u}_j^n , and random combinations of these would be used in computing variables, such as RKPM coefficients.

- **Loop fusion**

Loop fusion is also extensively implemented inside the force computation kernel, which aims to aggregate multiple loops into one. Loop fusion helps minimise runtime due to redundant loops. For example, in the force computation kernel, one single loop over neighbours is responsible for neighbour search, part of RKPM coefficients \mathbf{b} and $\partial\mathbf{b}$, and particle volume. Moreover, loop fusion will increase the opportunity for intermediate data to reside in cache [138], which in turn will reduce slow global memory use. In the case mentioned above, all the variables use positions of neighbour particles. The increased use of cache and decreased use of global memory is similar to kernel fusion, but loop fusion occurs at a smaller level within one kernel.

- **Floating-point division**

Floating-point division is a slow operation[48], so minimising its use will also be effective for improved performance, particularly on the GPU, which relies on its throughput rather than latency. In this implementation, positions of neighbour particles \mathbf{x}_j are used multiple times, but always in the form of $(\mathbf{x}_j - \mathbf{x}_i)/a$. Therefore, this term is computed and cached rather than bare \mathbf{x}_j cached.

- **Data structure**

Data structure on CUDA needs to be carefully designed considering the data access pattern to exploit DRAM burst, where the data that are close in address, or in other words coalesced, are cached at once. There are essentially two options to construct arrays that store variables, such as particle positions \mathbf{x}_i , velocities \mathbf{u}_i , and pressure p_i . One is arrays of particle class that stores variables per particle(Algorithm 2.2). The other is individual arrays of each variable(Algorithm 2.1).

As stated earlier, each thread, which corresponds to each particle in the physical sense, has loops where the positions of neighbour particles are accessed. By storing them close together in the memory space to increase coalescence, DRAM burst would kick in, and the number of memory access would be minimised, resulting in increased efficiency. Individual arrays of each variable enable this because of the sorting method deployed, where particles that are physically located close to each other are also located close to each other in the memory space. In contrast, arrays of particle class that store variables per particle would not be able to exploit DRAM burst since there are not many occasions where variables of one particle are used, such as position, velocity, and pressure of one single particle all at the same time. Moreover, using such a class on CUDA would pose another problem. Object instances created on the host reside only on the host, but not on the device, or vice versa. This would make data transfer between the host and the device challenging. Therefore, this study employs individual arrays of each variable.

- **Register**

A register is the fastest on-chip storage in the memory hierarchy. The variables that are used multiple times in the compute force kernel are cached in the register, which includes neighbour list, positions of neighbours, and RKPM coefficient $\partial N_i(\mathbf{x}_j)$ to reduce the latency caused by global memory access. However, care needs to be taken at register allocation. Each particle has different memory requirements because in Lagrangian particle methods, particles move around every time step, and each particle has different numbers of neighbouring particles. Nevertheless, as [63] pointed out, dynamic register allocation is not possible in CUDA. Therefore, the register is allocated to each particle at the compilation time such that it can accommodate the largest possible number of neighbour particles, which is decided upon trial and error. At the same time, register needs to be allocated sparingly because an excessive amount of memory per particle, i.e. per thread would mean a smaller number of blocks that can run concurrently, slowing down the process as a whole. Moreover, this could cause memory spill, where the register is run out of, and compensated by global memory in the background, posing another possible cause for slow-down. In this study, memory spill was not detected, or more precisely, there was no way to tell whether this happened or not even though analyses were made using Nvidia visual profiler.

Algorithm 2.1: Independent arrays of variables

```
1 CuArray<dvec3> position[N];
2 CuArray<dvec3> velocity[N];
3 CuArray<double> pressure[N];
4 ... // arrays of other variables
```

Algorithm 2.2: Particle class and its array

```
1 Class Particle contains
2   | dvec3 position; // dvec3 is data type for 3D vector
3   | dvec3 velocity;
4   | double pressure;
5   | ... // other variables
6 end
7
8 CuArray<Particle> particles[N]; // CuArray is data type for
   CUDA array
```

2.7.4 GPU parallelisation benchmark

To validate the GPU version and evaluate the speed-up achieved, a benchmark was performed. The test cases are 2D dam break (Fig. 2.20), 3D small dam break (Fig. 2.21), and 3D large dam break (Fig. 2.22). Firstly, as in Fig. 2.20, the GPU version yielded very similar results in the 2D dam break benchmark to the CPU version, which is also the case with the 3D small dam break (Fig. 2.21), demonstrating the validity of the GPU code. It also yielded a reasonable result in the 3D large dam break benchmark although this case was not carried out with the CPU code until the end of the simulation due to the unreasonably long runtime, and was instead done only for the first several time steps to compare the speeds. The GPU often produces slightly different results every time even when exactly the same code is executed even with the same conditions. This is because in the parallelisation process, the GPU handles some calculations, such as summation in different orders every time, which introduces subtle differences in round-off errors [178].

Fig. 2.19 shows the speed-up by the GPU. A minimum speed-up was 4.9x for the 2D small dam break, the case with the fewest particles. As the particle count increased, the speed-up brought by the GPU parallelisation was also more pronounced. The maximum speed-up as large as over 42x was achieved for the 3D large dam break, the case with the most particles.

Overall, this benchmark has demonstrated that the GPU implementation of this study is significantly faster than the CPU equivalent. In the

cone penetration test of grease in the later chapter, the particle count is around 100,000, and therefore the benefit of the GPU should be considerable although a performance comparison with the CPU was not made. The conditions used in the benchmark are listed in Table 2.7.

Table 2.8 shows the specs of the graphics card used in this study. The model is the Nvidia GeForce GTX 1080, released in 2016, so considering the evolution of graphics cards since then (Fig. 2.18), the same code on the latest GPU would run much faster with the newest graphics card.

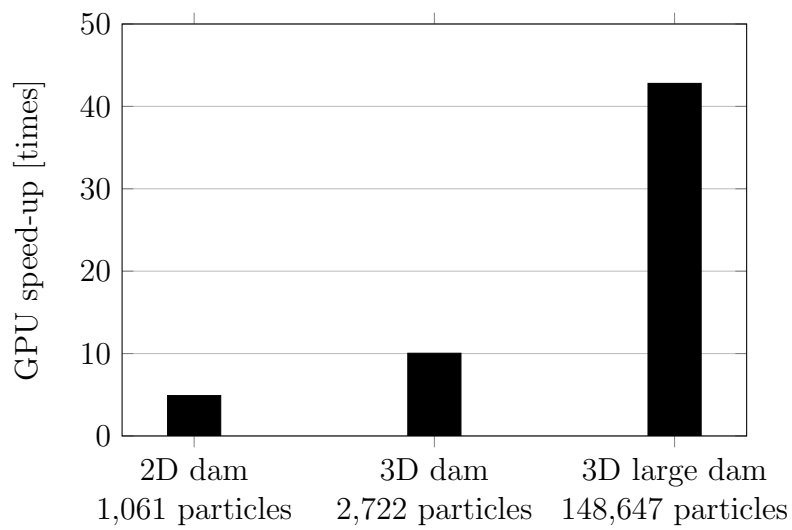


Figure 2.19: Speed-up with the GPU from the CPU equivalent

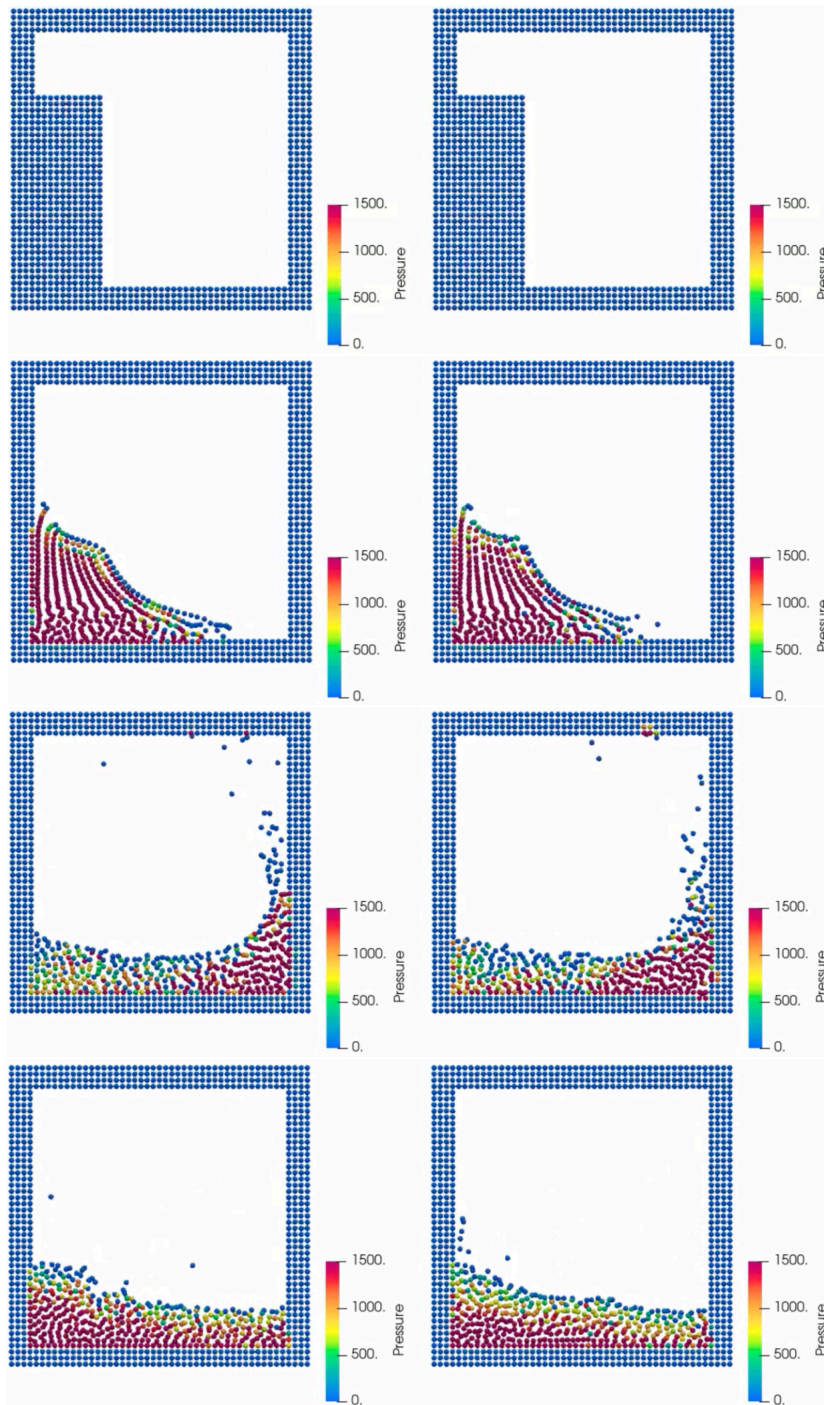


Figure 2.20: CPU(left) vs GPU(right) for the 2D dam break

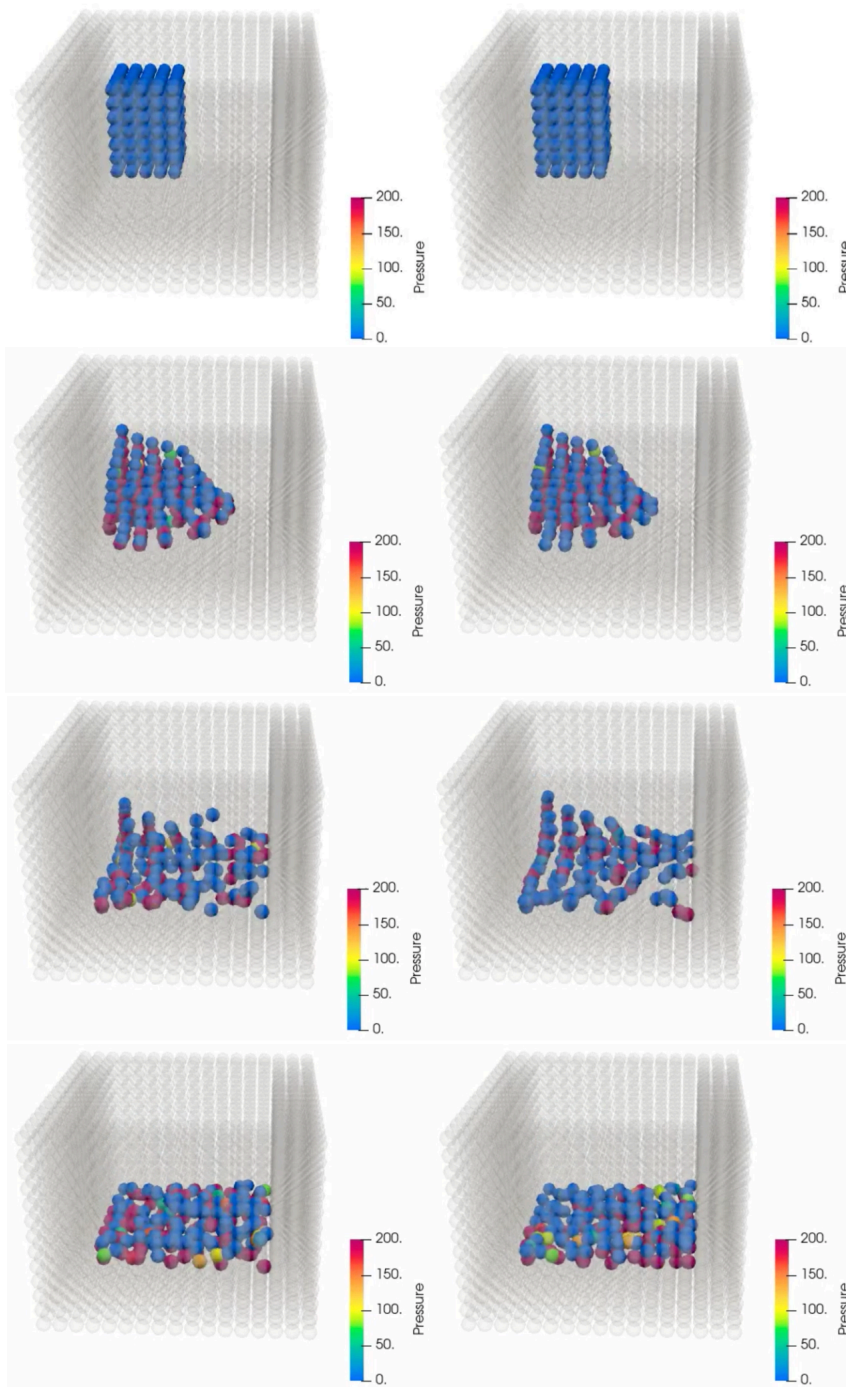


Figure 2.21: CPU(left) vs GPU(right) for the 3D small dam break

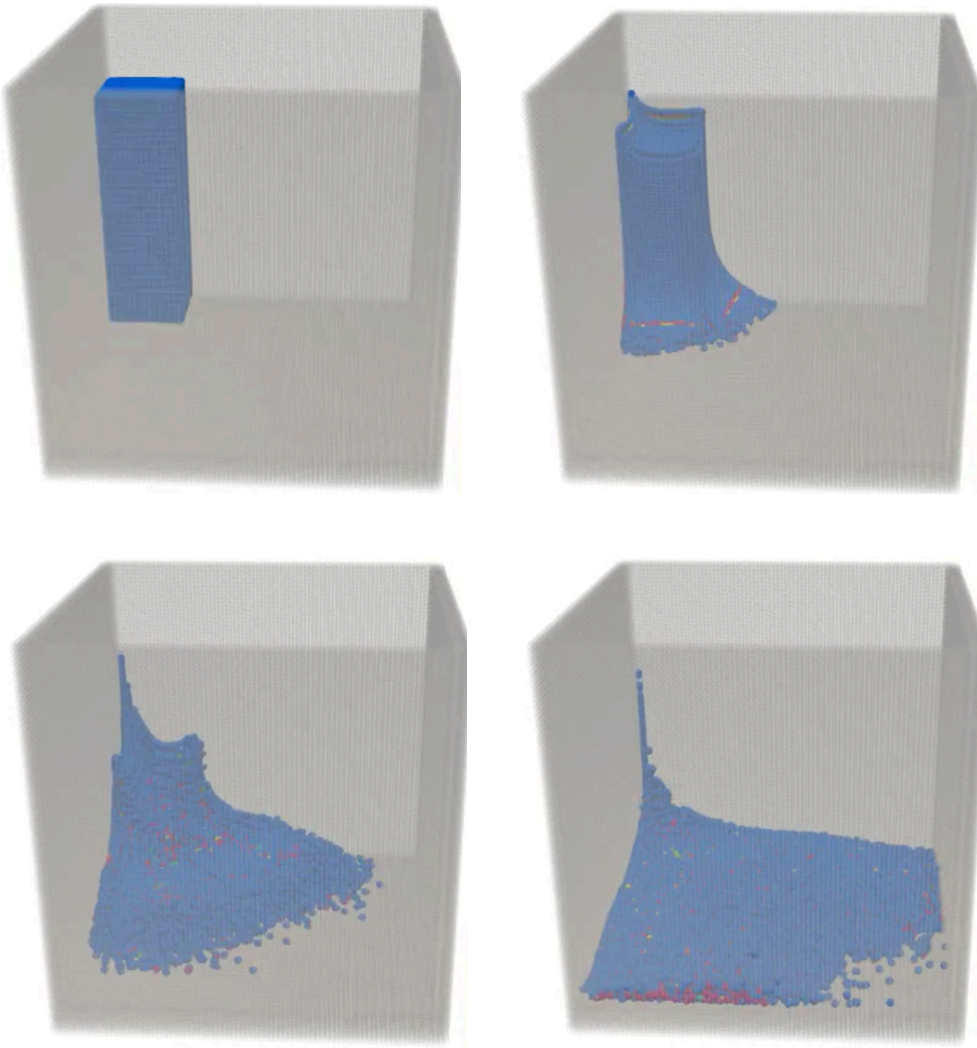
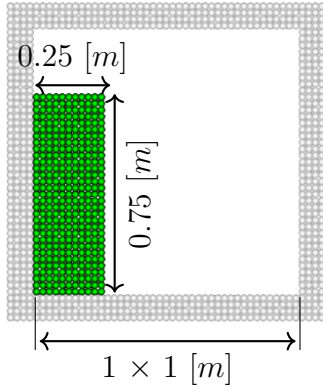
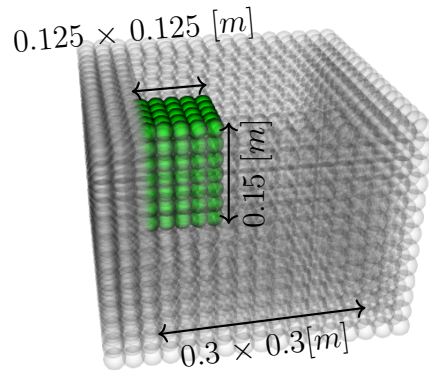


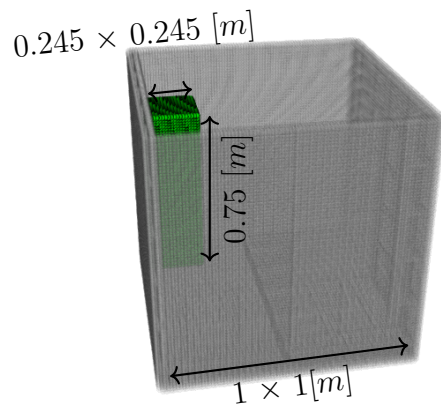
Figure 2.22: 3D large dam break with the GPU



(a) 2D dam break(Fig. 2.20)



(b) 3D dam break(Fig. 2.21)



(c) 3D dam break large(Fig. 2.22)

Figure 2.23: Particle file used in the GPU benchmarks

Table 2.7: Parameters used in the dam break GPU benchmark

	2D dam	3D dam	3D dam large
particle size [m]	2.5e-2		1.25e-2
kernel radius [m]	5.0e-2		2.5e-2
particle count	1,061	2,722	148.647
time step [s]	1.0e-4		2.5e-4
velocity blending coefficient	0.99		0.9
EOS coefficient B	3000		
shift coefficient	0.0		
surface particle volume coefficient	-		
surface particle number of neighbours	-		
surface cohesion coefficient [N/m]	-		
density $\rho[kg/m^3]$	1000		
viscosity $\mu_v[Pa \cdot s]$	0.1		

Table 2.8: Graphics card specs

model	Nvidia GeForce GTX 1080
Cores	2560
memory clock	10[Gbps]
memory bandwidth	320[Gbps]
CUDA compute capability	6.1
Compiler	CUDA 11.8

2.8 Overall algorithm

The overall algorithm used in this study is shown in Algorithm 2.3. Two things are worth noting. Firstly, as stated in §2.7, kernel fusion was applied extensively. Apart from the sorting part, the code has only two kernels, namely `compute_force()` and `time_integration_fluid()`. This extensive kernel fusion is partly helped by trick in implementing artificial viscosity, where the old value $\mathbf{u}_i^{h^n}$ is used rather than the new value $\mathbf{u}_i^{h^{n+1}}$ (line 23), eliminating the need to split this kernel. After each of the kernels, all the streams are synchronised to avoid data race.

Secondly, a trick was used in calculating the acceleration (eq.(2.82)) to minimise memory consumption. Computing $\dot{\mathbf{u}}_i$ in the i th thread (particle) requires the values that involve the neighbour particles, such as σ_j , V_j , and $\nabla N_i(\mathbf{x}_j)$ due to the employment of weak form. If the code were to exactly follow the summation to compute $\dot{\mathbf{u}}_i$ in the i th thread, i.e. "collect j th particle's influence", it would be necessary to store all the values listed above in the global memory in one kernel and to execute the summation part in another, which would require massive memory as each particle has many neighbours. Moreover, the kernel would need to be split for those values and acceleration. Rather than "collecting the j th particle's influence" like this, the code "distributes the i th particle's influence to the neighbours". This means that instead of storing those values in the global memory, arrays for the denominator and numerator in eq.(2.82) were stored, and i th thread computes its contribution to the numerator and denominator of j th particle. In other words, the i th particle works as a neighbour of the j th particle. This trick saves memory and leads to a higher efficiency for RKPM, where computationally intensive weak form and correction coefficients are involved. Nonetheless, due to the nature of GPU parallelism, a particle's denominator and numerator could be simultaneously overwritten by more than one particle, which would amount to a data race. However, this is avoided by the CUDA function `atomicAdd`, which executes the addition operation from the kernel with a safeguard against such simultaneous writing.

Algorithm 2.3: Overall algorithm at each time step

```
1 Sort according to [35];
2 global function compute_force() // the first kernel
3   size_t i = threadIdx.x + blockIdx.x * blockDim.x;
4   if i < N then
5     for particles in neighbour cells do
6       Create neighbour particle list;
7       Compute particle volume  $V_i$  (eq.(2.4)),  $\mathbf{b}$  and  $\partial\mathbf{b}$ 
          (eqs.(2.54), (2.62)), and  $(\mathbf{x}_j - \mathbf{x}_i)/a$ 
8     end
9     for particles in neighbour list do
10      Compute RKPM coefficient  $\nabla N_i(\mathbf{x}_j)$  (eq.(2.63)), strain
          rate  $(\nabla\mathbf{u})_i$  (eq.), shift (eq.(2.90)), blended velocity  $\mathbf{u}_i^h$ 
          (eq.(2.80))
11    end
12    Compute viscosity  $\mu_i$  (eq.(2.7)) and pressure  $p_i$  (eq.(2.7));
13    for particles in neighbour list do
14      Compute RKPM coefficient  $N_j(\mathbf{x}_i)$  (eq.(2.65)), stress  $\boldsymbol{\sigma}_i$ 
          (eq.(2.35)), numerator  $\nabla N_i(\mathbf{x}_j) \cdot \sigma_j V_j$  and denominator
           $N_i(\mathbf{x}_j)\rho_j V_j$  of Cauchy eq.(eq.(2.82));
15    end
16  end
17 end
18 global function time_integration_fluid() // the second kernel
19   size_t i = threadIdx.x + blockIdx.x * blockDim.x;
20   if i < N then
21     Compute acceleration
           $\dot{\mathbf{u}}_i = -(\mathbf{numerator})/(\mathbf{denominator}) + \mathbf{g}$ ; // eq. (2.82)
22     Update velocity  $\mathbf{u}_i+ = \dot{\mathbf{u}}_i * \Delta t$ ;
23     Artificial viscosity  $\mathbf{u}_i^{n+1} = \alpha\mathbf{u}_i^{n+1} + (1 - \alpha)\mathbf{u}_i^{h^n}$ ; // eq. (2.83)
24     Update position  $\mathbf{x}_i+ = \mathbf{u}_i * \Delta t + \mathbf{shift}$ ;
25     Update strain  $\boldsymbol{\epsilon}_i$ ; // eqs. (2.33), (2.34)
26   end
27 end
28 host function time_integration_rigid_body()
29   Update velocity  $\mathbf{u}_{rigid}+ = \dot{\mathbf{u}}_{rigid} * \Delta t$ ;
30   Update position  $\mathbf{x}_{rigid}+ = \mathbf{u}_{rigid} * \Delta t$ ;
31 end
```

2.9 Verification of the Method through Benchmark; lid-driven cavity flow

Quantitative verification of the code along with the numerical method will be performed in this subchapter since the code used in this study was built in-house from scratch rather than using an established program. This is done through the benchmark of the 2D lid-driven cavity flow with comparisons to existing studies.

Fig. 2.24 shows the boundary condition along with the particle file. The top wall has a velocity of $1[m/s]$ to the right. As shown in Table 2.9, Reynolds numbers 1, 10, 50, and 100 were tested by changing the value of viscosity whilst keeping other parameters fixed. These Reynolds numbers might be considered rather low, considering some studies handle larger values, such as 1,000 and 8,000([20], [102]). However, the target of this study is grease, the viscosity of which can be as high as $600[Pa \cdot s]$, and Reynolds numbers will be low, overall. Therefore, considering only cases with low Reynolds numbers here would suffice in this context.

The velocity profile and streamline patterns are examined in each case. Figs. 2.25 to 2.28 show the velocity fields, more specifically, normalised x-direction component u/U along the vertical centre line and y-direction component v/V along the horizontal centre line, compared with [110] except $Re = 1$. In all the cases, good agreements between this study and [110] were obtained with the velocity profiles overlapping with one another in the charts. Figs. 2.29 to 2.32 show the streamlines, in the case of $Re = 1$ and 100, compared with [102]. When $Re = 1$, the locations of the vortex were similar between this study and [102], around the centre in the x direction, and slightly below the top horizontal grid line. When $Re = 100$, although there was a slight difference in the vortex positions in the x direction, the results are still similar in that both have the vortex shifted towards the right in comparison to the case of $Re = 1$. When $Re = 10$ (Fig. 2.30), the vortex position is almost the same as when $Re = 1$. When $Re = 50$, the vortex position was between $Re = 10$ and 100, indicating the validity of the result.

Table 2.9: Parameters used in the lid-driven cavity flow benchmark

Reynolds number	1	10	50	100
particle size [m]	2.5e-2			
kernel radius[m]	6.25e-2			
particle count	2023			
time step [s]	1.0e-3			
velocity blending coefficient	0.9			
EOS coefficient B	3000			
shift coefficient	0.0			
surface particle volume coefficient	-			
surface particle number of neighbours	-			
surface cohesion coefficient [N/m]	-			
density ρ [kg/m ³]	1000			
viscosity μ_v [Pa · s]	1000	100	20	10
Top wall velocity U [m/s]	1.0			

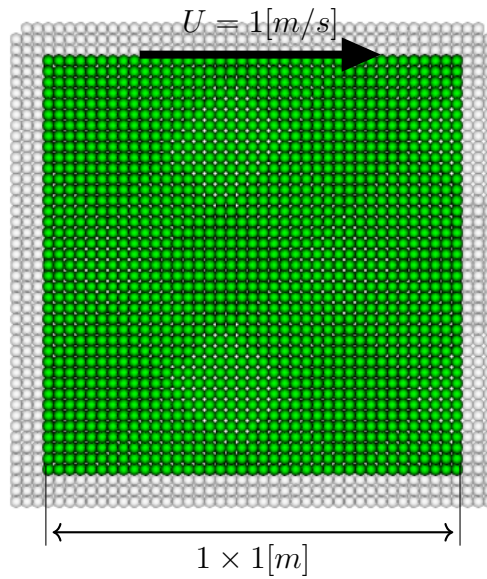


Figure 2.24: Particle file with velocity condition used in Figs. 2.15, 2.16

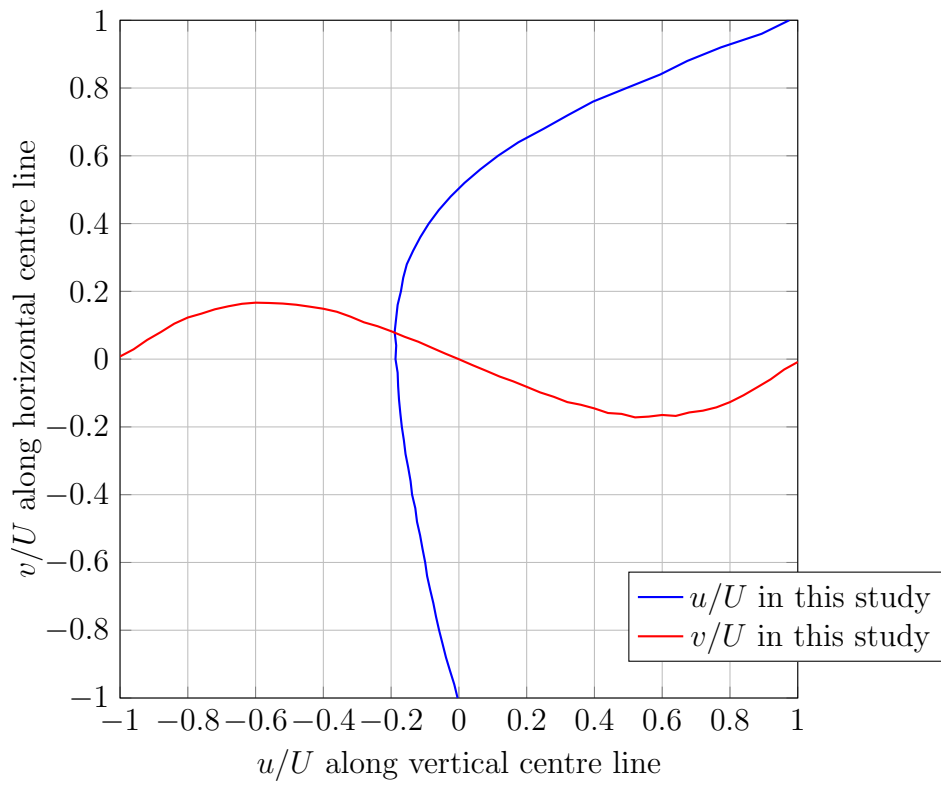


Figure 2.25: Normalised velocities in lid-driven cavity flow, $Re = 1$

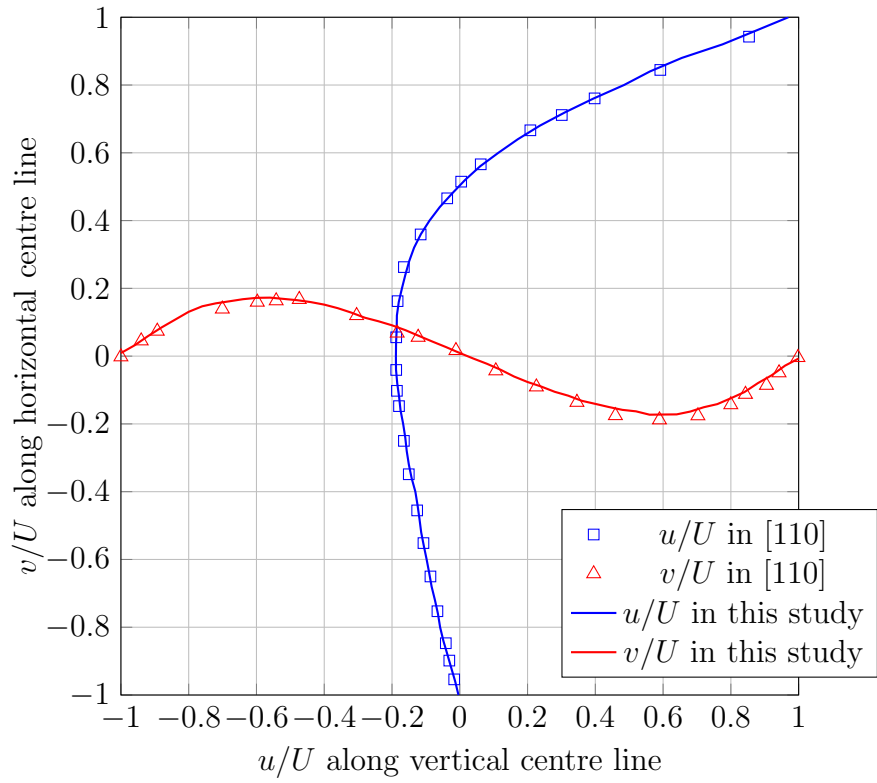


Figure 2.26: Normalised velocities in lid-driven cavity flow, $Re = 10$

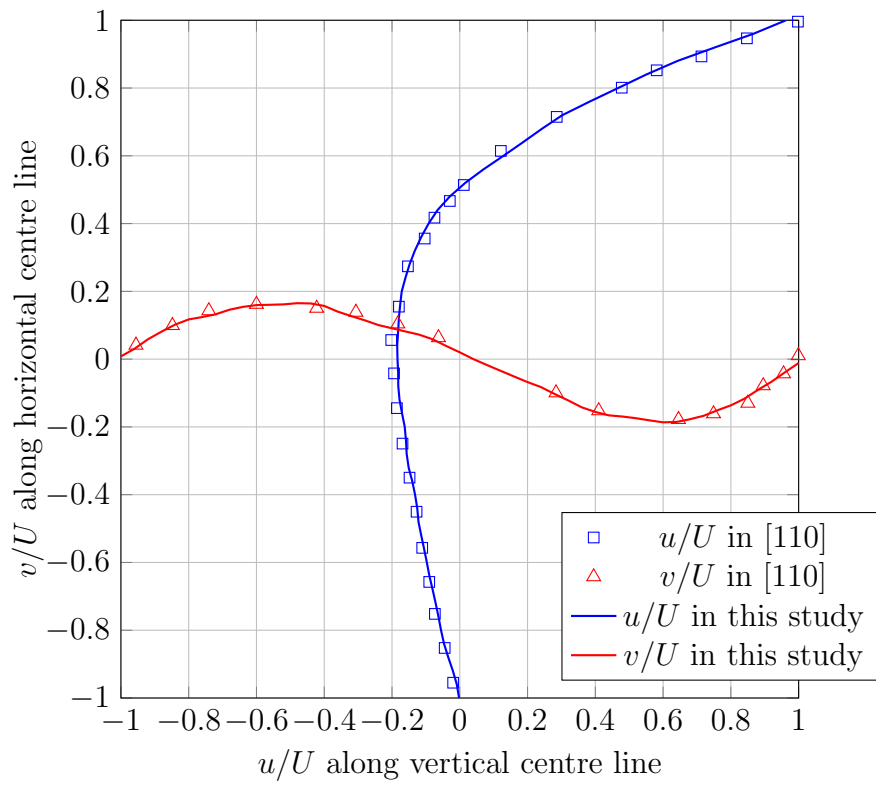


Figure 2.27: Normalised velocities in lid-driven cavity flow, $Re = 50$

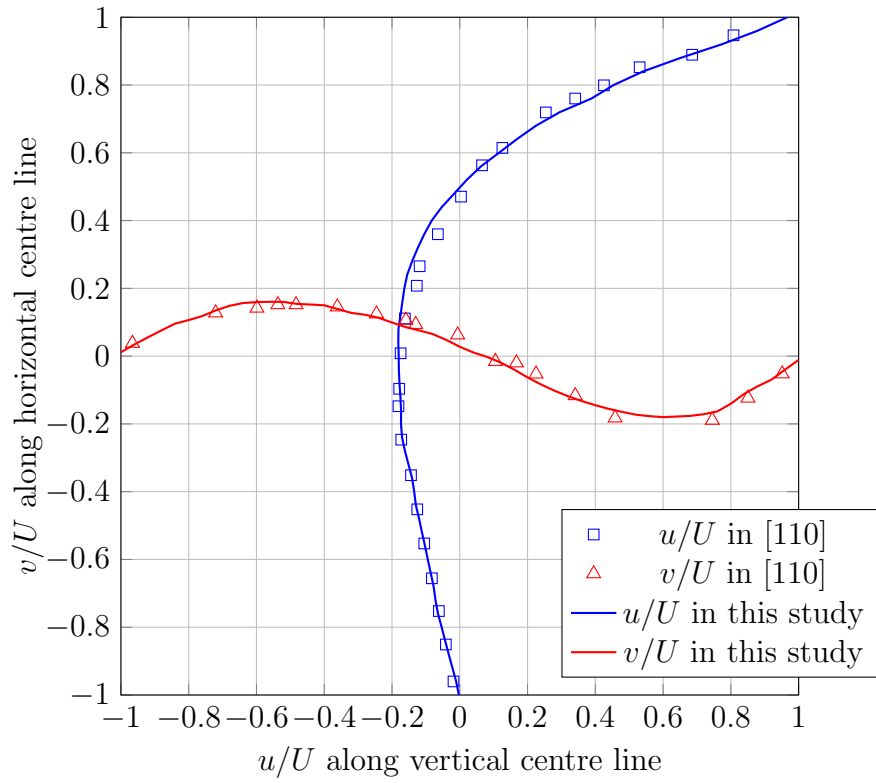
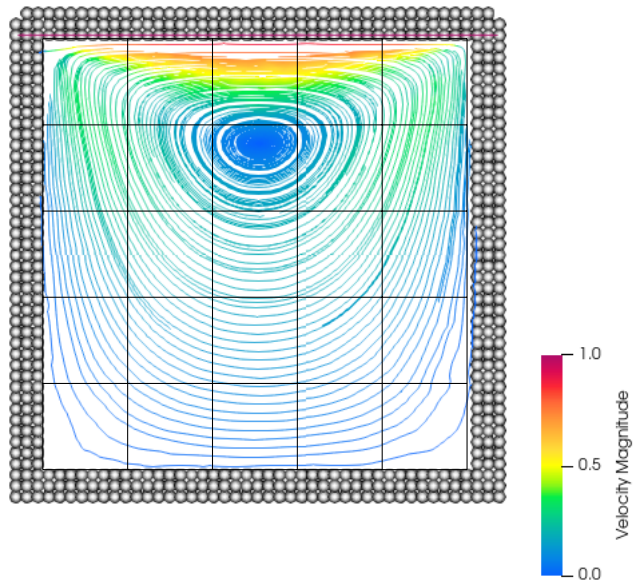
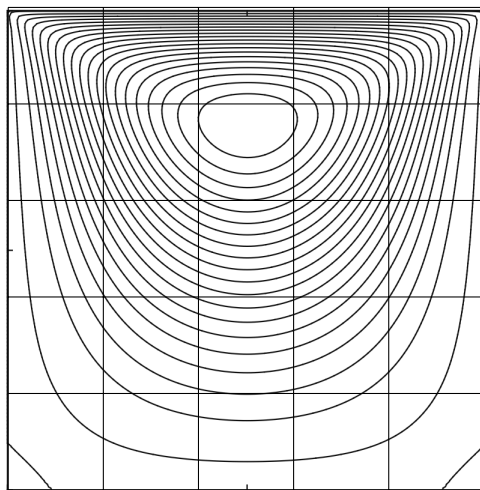


Figure 2.28: Velocities in lid-driven cavity flow, $Re = 100$



(a) This study



(b) [102]

Figure 2.29: Streamline patterns in lid-driven cavity flow, $Re = 1$

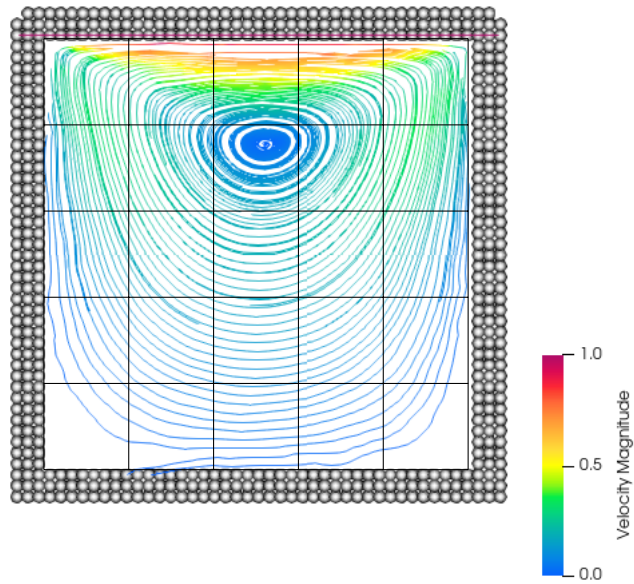


Figure 2.30: Streamline patterns in lid-driven cavity flow, $Re = 10$

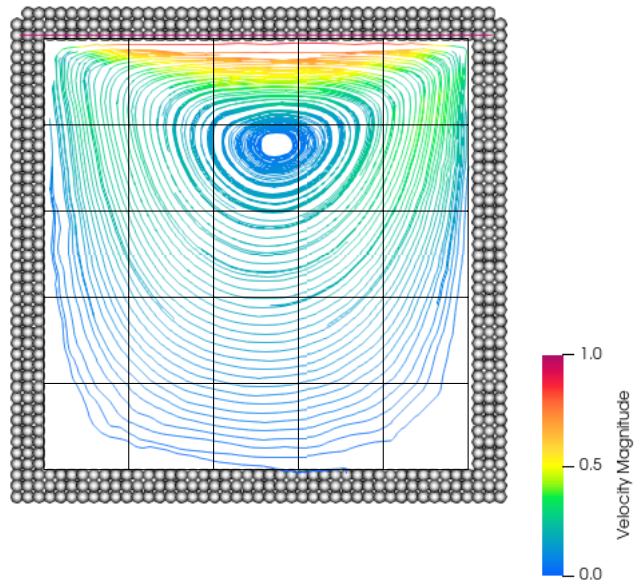
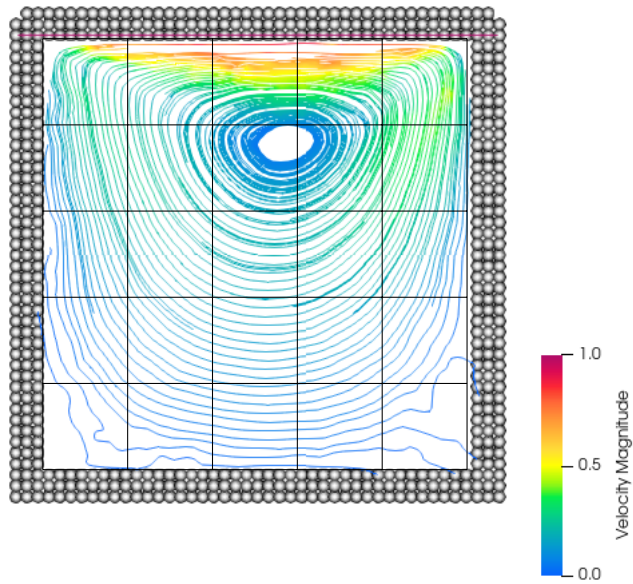
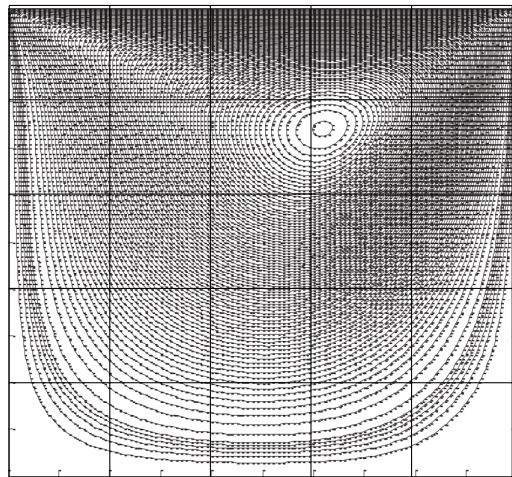


Figure 2.31: Streamline patterns in lid-driven cavity flow, $Re = 50$



(a) This study



(b) [102]

Figure 2.32: Streamline patterns in lid-driven cavity flow, $Re = 100$

In the later chapter, greases will be modelled as a non-Newtonian fluid. Therefore, in addition to the lid-driven cavity flows with different Reynolds numbers for Newtonian fluids, another case for a non-Newtonian fluid, more

specifically a shear-thinning fluid, was performed. The shear-thinning in this benchmark was modelled with the Cross model(eq. 2.100) as opposed to the Bingham model(eq. 1.2) used in [125], but the fitted curves suggest that the error by this difference is considered relatively small(Fig. 2.33).

$$\mu = \frac{\mu_0 + \mu_\infty (K\dot{\gamma})^m}{1 + (K\dot{\gamma})^m} \quad (2.100)$$

μ_0 , μ_∞ , K , and m are viscosity at zero shear rate, viscosity at infinite shear rate, consistency coefficient, and power constant respectively. The values of the parameters are shown in Table 2.10. The Reynolds number is 10, based on the zero strain-rate viscosity μ_0 as $Re = \rho LU/\mu_0$. Fig. 2.34 shows the result. The overall result is close to the previous study with the velocity profiles having local maxima and minima at almost identical points. The velocity component of the y direction is small in magnitude in this study across the section, and the x direction below $y = 0$ compared to [125]. However, it is difficult to say which of the two studies are more accurate as no analytical solutions exist for this flow, and also there are a very limited number of studies of non-Newtonian lid-driven cavity flows.

To conclude, the plausible results that have been obtained in these lid-driven cavity flow benchmarks show the reasonableness of the method and code used in this study.

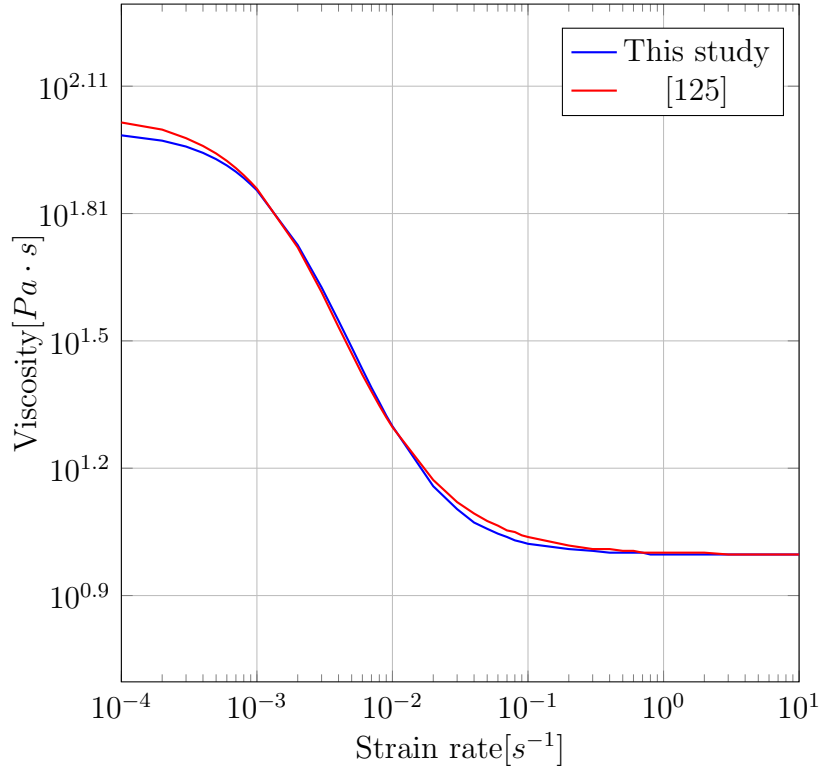


Figure 2.33: Flow curves

Table 2.10: Fluid parameters used in non-Newtonian lid-driven cavity flow

density [kg/m^3]	cross μ_0	cross μ_∞	cross m	cross K
1000	100	10	1.257	519

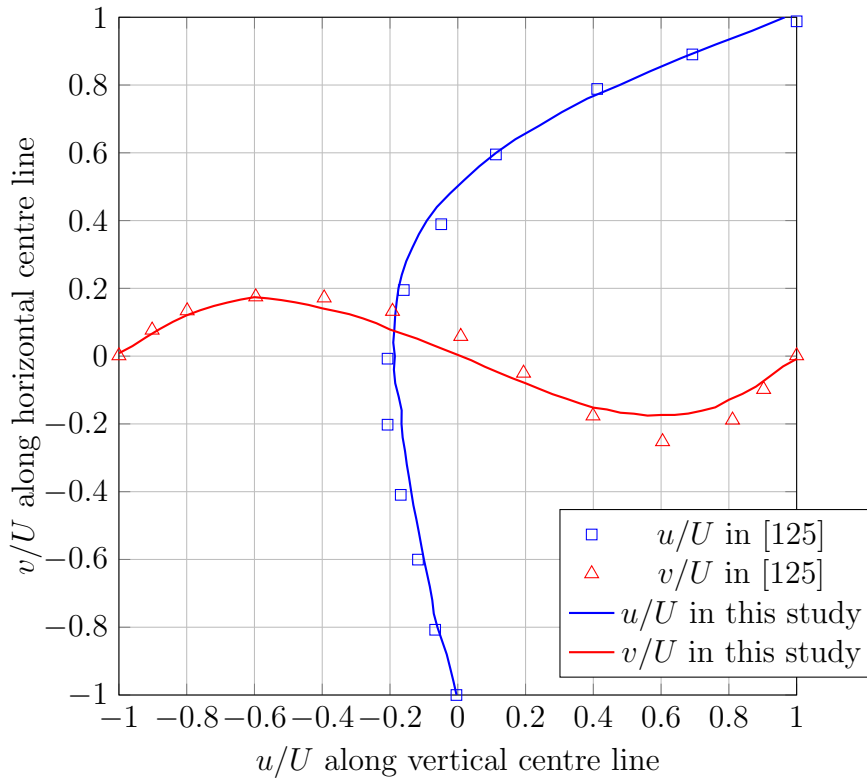


Figure 2.34: Normalised velocities in non-Newtonian lid-driven cavity flow, $Re = 10$

2.10 Conclusions

Considering those needs and challenges, this study attempted to establish a fast and accurate CFD method. More specifically, this study implemented the RKPM on the GPU from scratch. The RKPM formulation used was discussed along with its basis, the SPH. The techniques employed to realise the tests in the next chapter were explained and implemented, i.e. elastoplasticity, fluid-rigid body coupling, and particle shifting. Moreover, the tactics specific to the GPU implementation to maximise the performance were described. The performance was compared to the serial CPU equivalent, and the gain by GPU parallelisation was significant, but at the same time, the overall performance was somewhat limited due to the time step size constraint arising from the high viscosities of grease. However, this issue is not specific to CPU computing. Implicit schemes would allow larger time steps, but this would increase the computational cost per time step due to the

global matrices, and the estimated cost exceeds the benefit. A higher-speed linear solver for the GPU is a much-needed yet unfulfilled requirement.

3 Numerical simulation of grease; cone penetration and plate separation tests

3.1 Modelling of grease

This chapter describes the way the material properties of grease are handled in the simulation. As discussed in the previous chapter, grease possesses various material properties, some of which are unique to grease as opposed to oil, another widely used lubricant. These include shear rate-dependent viscosity, temperature-dependent viscosity, elasticity, tackiness, phase separation, and consistency.

Among these material properties, firstly, the density of grease will be taken into account in the simulation using the values obtained from the experiment.

Secondly, shear rate-dependent viscosity will be taken into consideration. Grease behaves almost like a solid at low shear rates. When grease is treated as a purely viscous fluid, i.e. without elasticity, several models exist that can describe this behaviour. Among these models, the Herschel–Bulkley model is one of the simplest with the yield stress and the following stress steadily increasing with strain rate. However, when treating fluids as purely viscous, the model needs modification by replacing the yield stress with a very high viscosity. Cross model is one such option, which is adopted in this study.

The viscosity μ is expressed as

$$\mu = \frac{\mu_0 + \mu_\infty (K\dot{\gamma})^m}{1 + (K\dot{\gamma})^m} \quad (3.1)$$

where μ_0 , μ_∞ , K , and m are viscosity at zero shear rate, viscosity at infinite shear rate, consistency coefficient, and power constant respectively. As the strain rate approaches zero, the viscosity approaches the constant value μ_0 , when the fibre structure of the thickener is thought to be the most complete. On the other hand, when the strain rate approaches infinity, the viscosity approaches its minimum. This minimum is thought to be the same as the viscosity of the base oil where the thickener is not structured at all. This feature is preferred in previous studies of grease CFD[146][177]. In addition to grease, the Cross model has also been used to model molten polymer, another shear-thinning fluid [64][85].

Four greases shown in Fig. 3.1 are used in this study. Fig. 3.2 shows the stress-strain rate curves. The dots show the result from the experiment, done by the author's colleague at NSK on 9th July 2020 using the Anton Paar's rotational rheometer MCR302 with the 25mm diameter parallel plate

attachment. The temperature was maintained at 25 °C by the instrument. The curves indicate the Cross model fitting with the parameters listed in Table 3.1.



Figure 3.1: Greases used in this study, stored in the container(author's own)

Table 3.1: Grease parameters with only viscosity

	Grease 1	Grease 2	Grease 3	Grease 4
density [kg/m^3]	863	920	830	850
cross μ_0	300	600	500	400
cross μ_∞	2.0	0.05	0.25	0.3
cross m	0.65	0.78	0.95	0.8
cross K	1.0	1.0	1.0	0.5

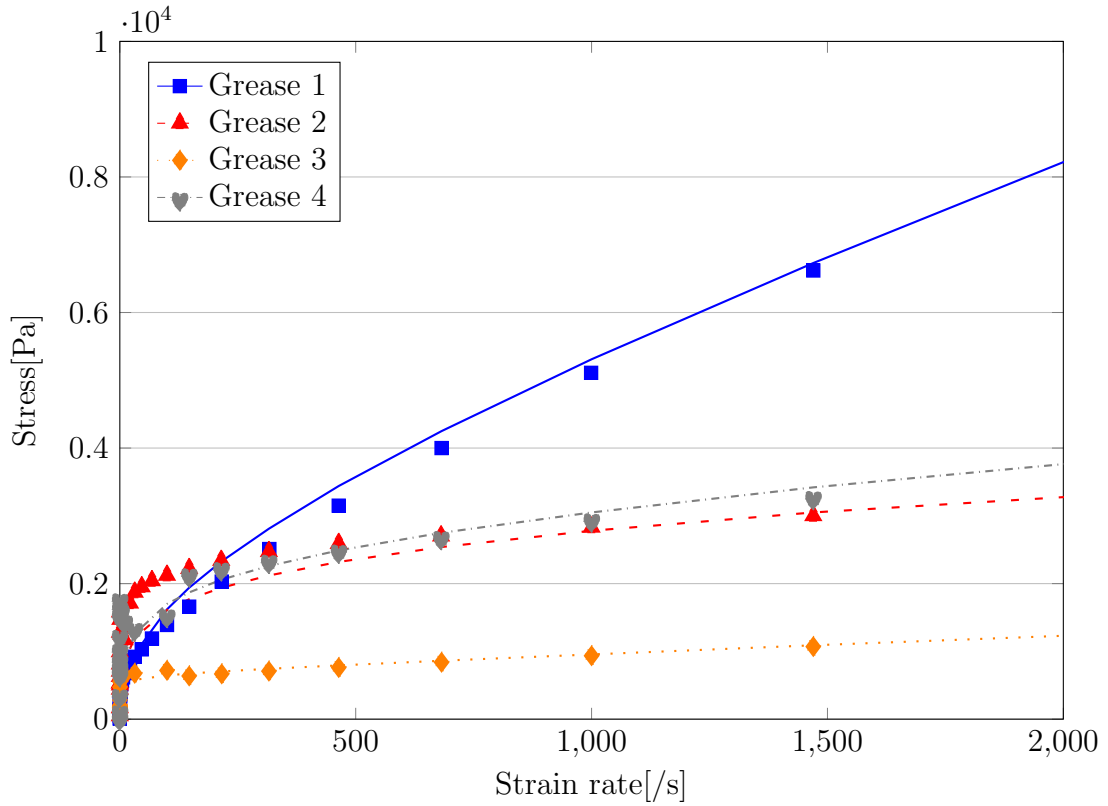


Figure 3.2: Stress-strain rate curves with viscosity only. Symbols for data from experiment and lines for fitted curve.

Elasticity will also be taken into account to incorporate the solid-like behaviour of grease, most notably, yielding. Modelling with only viscosity will result in creep, i.e. no matter how small the applied force might be, the material continues to flow. In the case of the widely-used cone penetration test, this will overestimate the value with the cone continuing to fall. This implies that in applications like ball bearings, the start-up torque could be underestimated, or grease may not resist the centrifugal force to sit still in certain locations.

In this study, the model introduced in §2.2 is used, which includes elastic modulus, yield strain, and decay rate. The reason for this selection is the relative simplicity, for example in comparison to the Oldroyd-B model, sometimes used in CFD [139] [94]. As a first step to consider the elasticity of grease, this can be considered a reasonable choice.

As for parameter fitting, the yield stress is firstly determined by finding the intersection of the stress-strain curves with the vertical axis in Fig. 3.3.

It is not feasible to independently measure the elastic modulus E and yield point ϵ_Y , and they had to be chosen arbitrarily. More precise determination is a topic for future research. Table 3.2 and Fig. 3.3 show the parameters and the curves with elasticity fitted to the experiment data (the same as Fig. 3.2) respectively.

The measurement and determination of the parameters regarding viscosity and elasticity are a difficult subject. The parameters of viscosity are determined by fitting the Cross model to the data obtained from the rotational rheometer. The parameters of elasticity are determined from the same data, but this makes it almost impossible to separate viscosity and elasticity. This is because the rheometer measures only the torque, which is the combination of both viscosity and elasticity, meaning the contributions from those are indistinguishable from the data. For convenience in this study, amongst all these parameters, yield stress σ_Y was firstly determined from the approximate intercept in the stress-strain rate chart (Fig. 3.3). This method is not perfect because grease does not exhibit a clear intercept with some gradient visible when zooming in to the stress-shear rate curve around the low strain-rate region. After this, the parameters of viscosity are determined by fitting the model so that the sum of the stress by the fluid and the yield stress fit the measured data. As for elasticity, in addition to the yield stress, elastic modulus and decay rate also need determining, where again, there were no definite ways of determination. The already determined yield stress is a product of elastic modulus and yield strain, and one of them had to be determined after trial and error whilst the other was automatically found. As for decay rate, at which the strain beyond the yield point reduces, even though arbitrary values are used, it was empirically found in the test conducted later that it exerts very little influence on the result.

There is a measurement method that separates the components of elasticity and viscosity, using the rotational rheometer with oscillatory displacement. In this method, the storage modulus corresponds to the elastic component in phase with the displacement, and the loss modulus viscous component out of phase with the displacement. Despite this advantage of separation, this study did not adopt this since those moduli do not directly correspond to elastic modulus or viscosity, the very values that are necessary as input parameters. Moreover, it is arguable how well the oscillatory movement within a limited strain amplitude represents the grease flow in the bearing.

There are certain material properties of grease that are not taken into account in this study despite researchers' interest.

Grease viscosity has temperature dependency, but viscosity in this study is not an explicit function of temperature. The main target is the bearing torque, and in most cases, torque is measured only for the first several minutes

due to practical reasons in product benchmarking despite actual applications operating much longer. Temperature rise in the ball bearing takes tens of hours even for a relatively high speed(5800rpm) [172], a condition that leads to a quick temperature rise. Moreover, temperature dependency is implicitly considered in this study because the viscosity measurement and other tests are all carried out approximately at the same temperature.

Oil separates from the bulk grease over time either in storage or operation in machinery. It is, however, neglected in this study like the previous studies on grease CFD. This is because the time scale for oil separation is long. Its measurement often involves some acceleration technique, for example, centrifugal force or heat applied to the grease sample [12][147]. With the latter, non-trivial separation still takes at least hours [73].

Thixotropy will be neglected for simplicity. The modelling and parameter determination are already complicated by taking into account elastoplasticity and strain rate-dependent viscosity. Adding an extra material property would be appropriate after fully investigating the abilities and shortcomings of the proposed method. Therefore, along with temperature-dependent viscosity and oil separation, thixotropy is a topic for future work.

There is one difference to note amongst material properties in simulation. Some of them are direct input parameters of simulation that explicitly appear in the Navier-Stokes equation, such as density, viscosity, and elastic modulus. There is another type of material properties that on the one hand, still characterise grease but on the other hand, do not appear in the Navier-Stokes equation, and can only be the output of simulation. Those include consistency and tackiness, to cite a few. Consistency and tackiness are nonetheless thought relevant to this study, so they will be evaluated as results of simulation in this study.

To summarise, this study models grease considering density, viscosity, and elasticity as input parameters, and simulates plate separation and consistency tests. In this way, the validity of the grease modelling will be evaluated.

Table 3.2: Grease parameters with elasticity

	Grease 1	Grease 2	Grease 3	Grease 4
density [kg/m^3]	863	920	830	850
cross μ_0	300	300	400	200
cross μ_∞	2.0	0.15	0.25	0.4
cross m	0.65	0.8	0.95	0.4
cross K	1.0	1.0	1.0	0.5
σ_Y	500	1400	300	1500
$\sigma_{Y,low}$	20	300	200	400
$E_{(high)}$	500	1400	300	1500
E_{low}	20	300	200	400
ϵ_Y	1	1	1	1
α	100	100	100	100

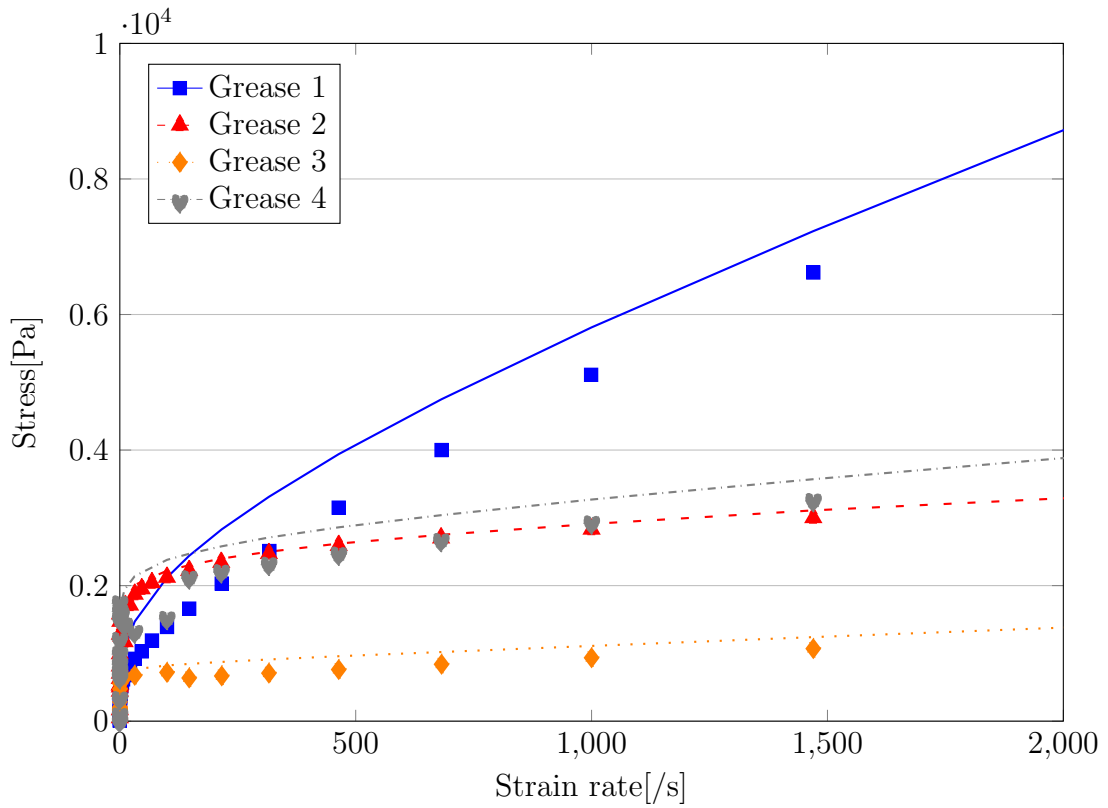


Figure 3.3: Stress-strain rate curves with elasticity. Symbols for data from experiment and lines for fitted curve.

In the following subchapters, grease is simulated in two different tests. In §3.2, cone penetration is done, roughly representing compression, and in In §3.3, plate separation is done, roughly corresponding to dilation. As will be discussed, simulation without elasticity using the parameters shown in Table 3.1 resulted in excessively large penetration values although a brief test showed a good agreement in the plate separation (Table. 3.3). On the other hand, simulation with elasticity using the parameters shown in Table 3.2 resulted in a good agreement in the cone penetration whereas again a brief test in the plate separation resulted in an excessively diminished stretch (Table. 3.4).

Table 3.3: Cone penetration with grease 1

exp.	sim. w/o elasticity	sim. w/ elasticity
305	409	327

Table 3.4: Cut-off length in *mm* in plate separation with grease 1

exp.	sim. w/o elasticity	sim. w/ elasticity
29.5	28.4	14.5

This implies that elasticity matters in grease, but is not perfectly modelled in simulation. Grease possesses multifaceted aspects, and the above results imply that grease responds differently to compression and tension, i.e. it is harder under compression, and softer under tension. Therefore, variable elastic modulus is introduced as in eq. (3.2). A higher value is used under compression and a lower one under dilation based on volumetric strain $tr(\epsilon)$.

$$E = \begin{cases} E_{(high)} & \text{if } tr(\epsilon) \leq 0 \\ E_{low} & \text{otherwise} \end{cases} \quad (3.2)$$

This method introduces extra arbitrariness, but it helps replicate grease behaviour under different settings, namely compression and tension, which itself is a novel attempt. From this viewpoint, the measure has somewhat effectively balanced the cone penetration and plate separation whilst the input parameters regarding viscosity and elasticity are still based on the data obtained with the rotational rheometer.

Despite the arbitrary determination of the elasticity under tension, there is evidence for materials' different responses to compression and tension. For example, [62] obtained different elastic moduli under compression and tension for polypropylene composites, referring to it as tension-compression asymmetry, and attributed it to the filler orientations in the material. Since both polypropylene composites and greases are suspensions, a similar phenomenon is also plausible for grease. More precise measurement and modelling of two elastic moduli will be a future research topic. The surface cohesion coefficient was also determined without quantitative links to experiments. It was chosen as a minimum value that would alleviate unnaturally sparse particles observed under tension.

Despite the determination processes of those arbitrarily chosen parameters, they help better replicate grease in the tests conducted in the following chapter, and the results suggest that the grease behaviour in the actual application, such as ball bearings, would also be better reproduced due to those parameters.

3.2 Cone penetration

3.2.1 Introduction

As mentioned in the previous chapter, the consistency of grease, measured in the cone penetration test, is widely used as an indicator of the suitable usage of grease, which is also standardised by the National Lubricating Grease Institute(NLGI). In fact, grease catalogues by bearing manufacturers or lubricant manufacturers include the NLGI indices [140], [154], [164].

Consistency is also mentioned in the context of machine performance, such as noise [192], bearing life [84], mechanical stability of grease [39], leakage [11], friction in seals [9]. It is also analysed from the viewpoint of the thickener microstructure [141]. Furthermore, consistency is also mentioned in numerical simulation contaminant motion in the bearing[176], [177], [168].

Those examples demonstrate the extensive use of consistency as one of the prominent characteristics of grease. Nonetheless, there have not been any examples where cone penetration is simulated. Therefore, this study simulates cone penetration as a benchmark as to how well grease is replicated, particularly discussing the significance of elasticity.

Moreover, the cone penetration test is largely the measurement of the response of grease to compression, one of the modes of the flows that grease experiences in the ball bearing(Fig. 1.17). Therefore, consistency, measured in the cone penetration test, partially examines grease in a setting simpler than the actual application.

The experimental results of the cone penetration test in this work were obtained from the private grease database of NSK, the ball bearing manufacturer and the author's sponsor. The test was performed with the quarter-size cone equipment, following the industry standard procedure specified in [157]. The quarter-size is adopted as it requires smaller amounts of grease than full-size and half-size. In the test, firstly grease is filled in the chamber of the grease worker(Fig. 3.4), and the grease is "worked", meaning it undergoes 60 strokes in the worker manually with the handle within 60 seconds. The working process is meant to better represent the consistency of greases in use as grease in the actual use is thought to experience some degree of shear for example when grease is injected inside the mechanical part. The handle is subsequently removed, and the grease surface in the chamber is flattened with a spatula. Finally, the cone is dropped in the same chamber(Fig. 1.2), and the depth after five seconds of free fall is measured by the machine. All the tests were conducted in January 2020. The specific dates were unavailable on the system.

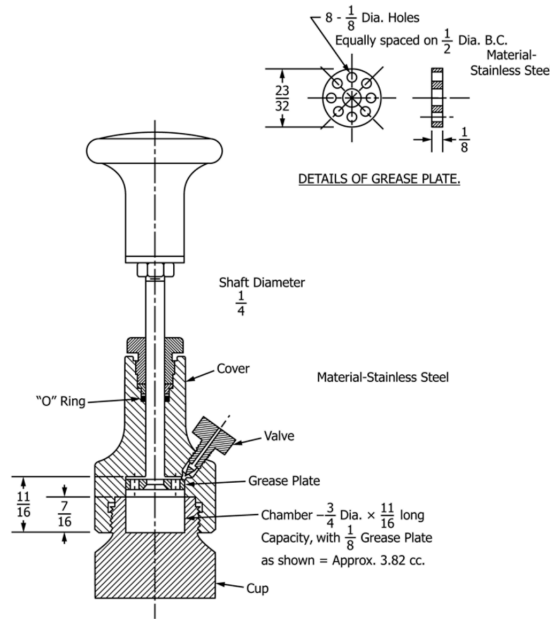


Figure 3.4: Industry standard grease worker [157]

3.2.2 Results of simulation

In this simulation, half-size cone penetration was employed as opposed to the quarter-size that was used in the experiment. This is intended for simulation to balance decently high resolution and moderate particle count simultaneously. Nonetheless, there is a widely recognised conversion equation which is provided by ASTM [157], and therefore this comparison between the simulation and the experiment with different sizes can be considered reasonable. The equation reads

$$P = 2r + 5 \quad (3.3)$$

where P is the penetration in the full-size test, and r is the cone penetration depth in the half-size test, both measured in tenth of a millimetre. To replicate the free fall of the cone, fluid-rigid body interaction was implemented. Figs. 3.15 to 3.18 shows the beginning of the simulation and at the completion after five seconds.

The values of consistency without elasticity are shown in Fig. 3.7 along with the results from the experiment. There are two points to note. Firstly, simulation without elasticity overestimated the consistency for all greases. This is because the cone continues to penetrate for the entire five seconds as can be seen in the cone displacements(Figs. 3.9). Secondly, the difference

between each grease is hardly seen at all with almost no correlation with the experiment. Nonetheless, including elasticity in simulation drastically changes the results. As in Fig. 3.8, the absolute values are closer to the experiment than the cases without elasticity. This is because, with elasticity, the cone stops almost completely after the initial fall (Figs. 3.10, 3.14). In the cases of Grease 2, Grease 3, and Grease 4, after the initial fall, the cone slightly bounces upwards, strongly suggesting the effect of elastic force. The bounce-back of the cone was also confirmed experimentally in a study [160] although this study did not track the cone displacement over time to confirm this due to the lack of facility. Moreover, the trend agrees with the experiment. Overall, it can be concluded that elasticity plays an important role in the cone penetration test. A good agreement was obtained between the simulation and experiment, with grease 1 getting the smallest consistency, and grease 4 the largest. However, as can be seen from the absolute values, the simulation overestimated the consistency in all the cases, or in other words the grease is evaluated softer than it is in the experiment.

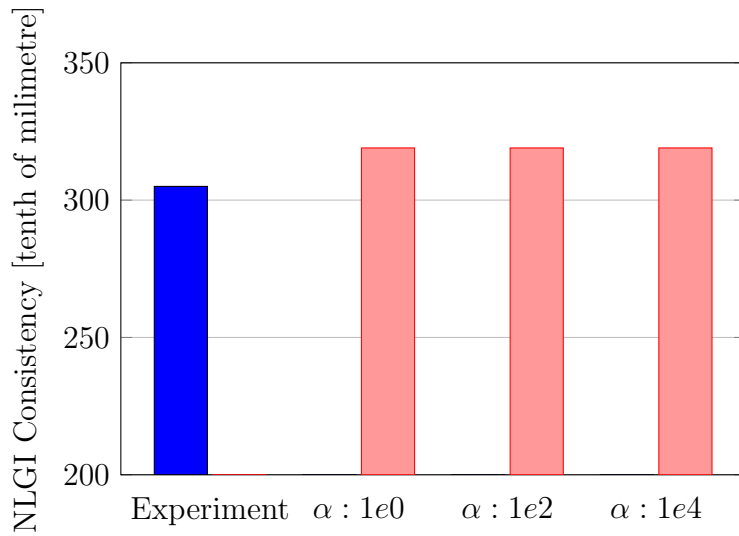


Figure 3.5: Effect of decay rate α on consistency for grease Grease 1

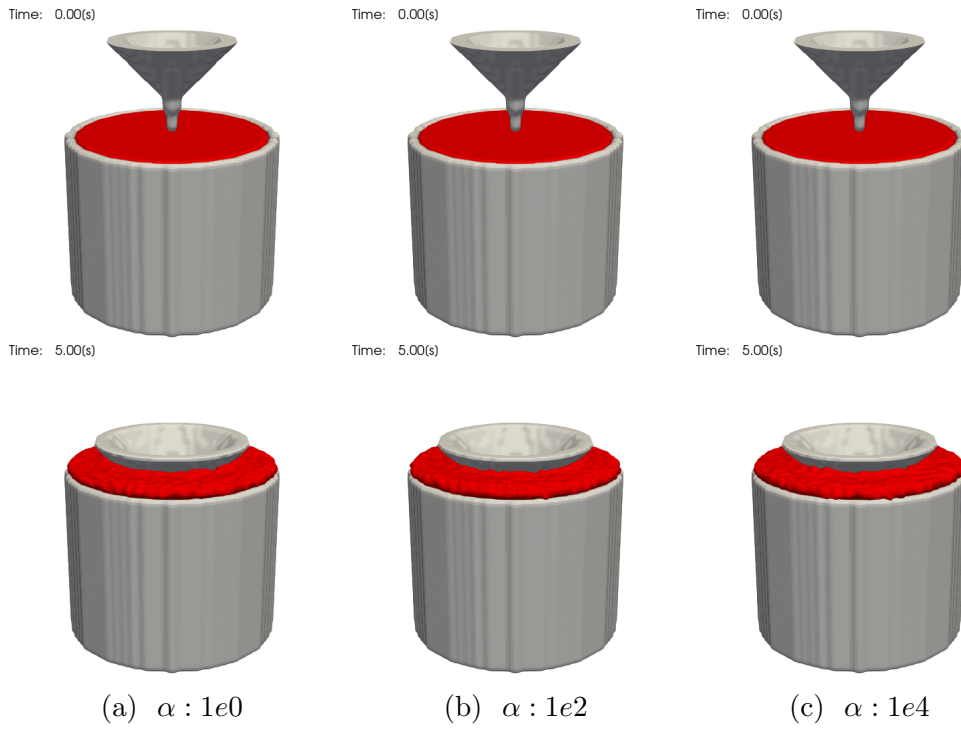


Figure 3.6: Effect of decay rate α in half-size penetrometer for grease SHC460 in half-size penetrometer. Top: initial state, bottom: after 5[s]

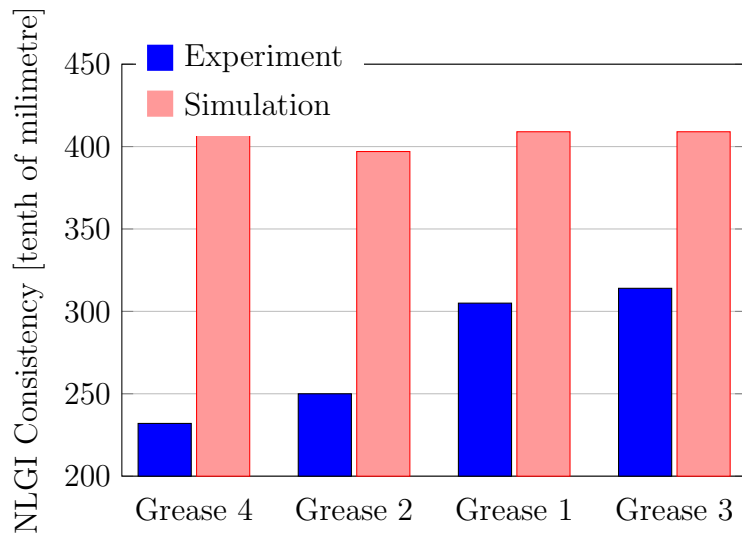


Figure 3.7: Grease consistencies without elasticity

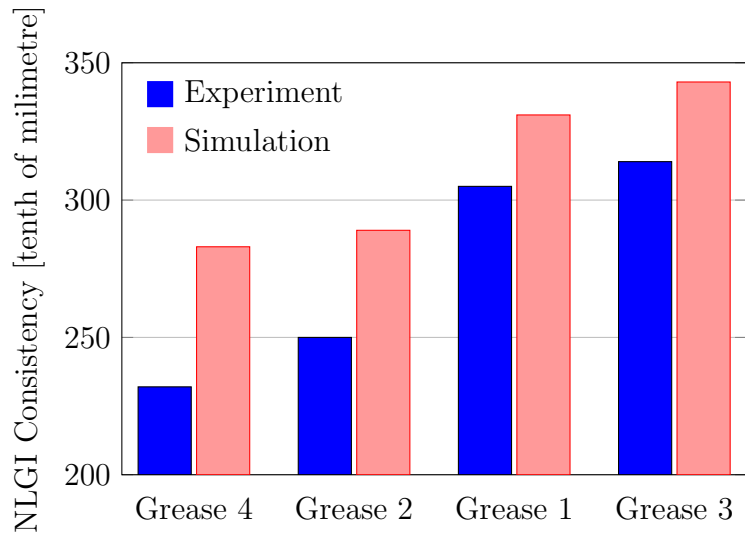


Figure 3.8: Grease consistencies with elasticity

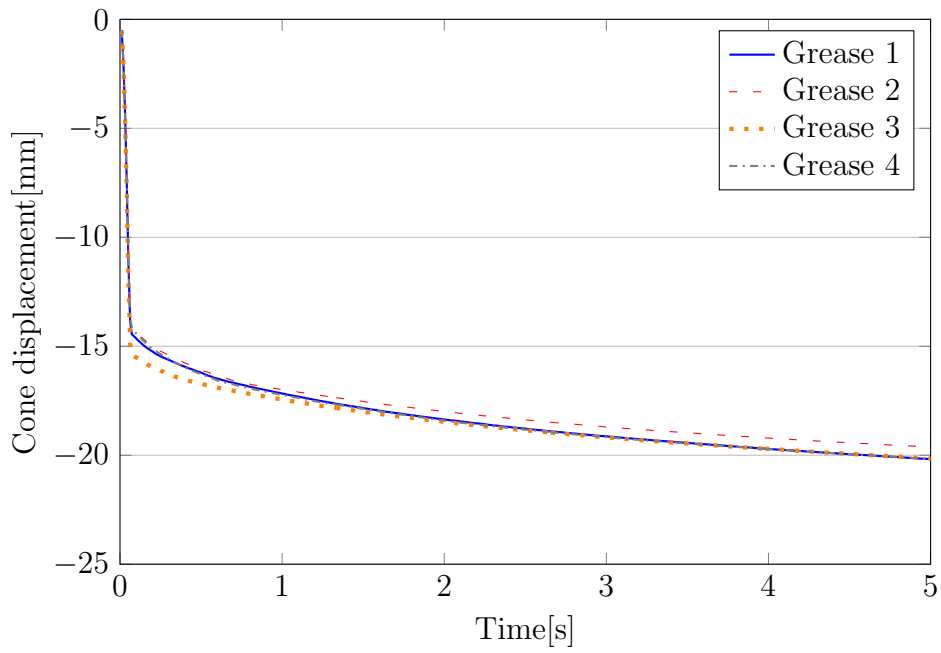


Figure 3.9: Cone displacements over time without elasticity

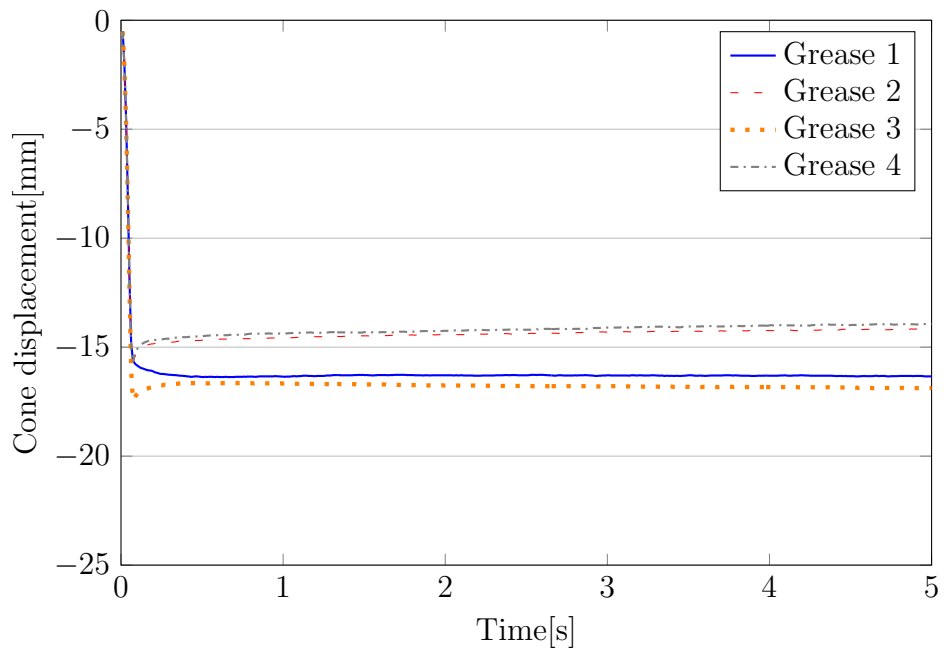


Figure 3.10: Cone displacements over time

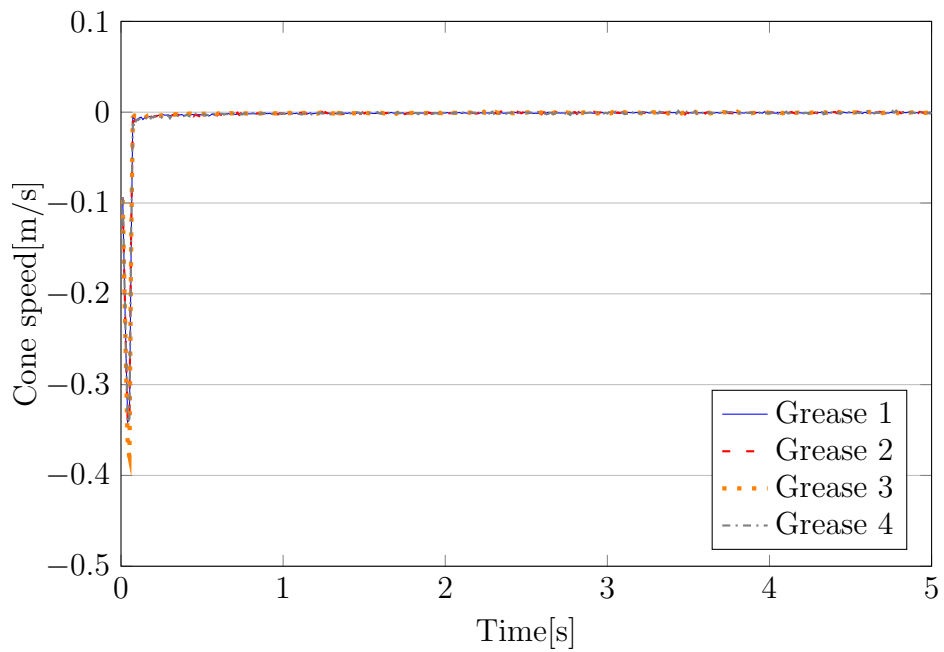


Figure 3.11: Cone speeds over time without elasticity

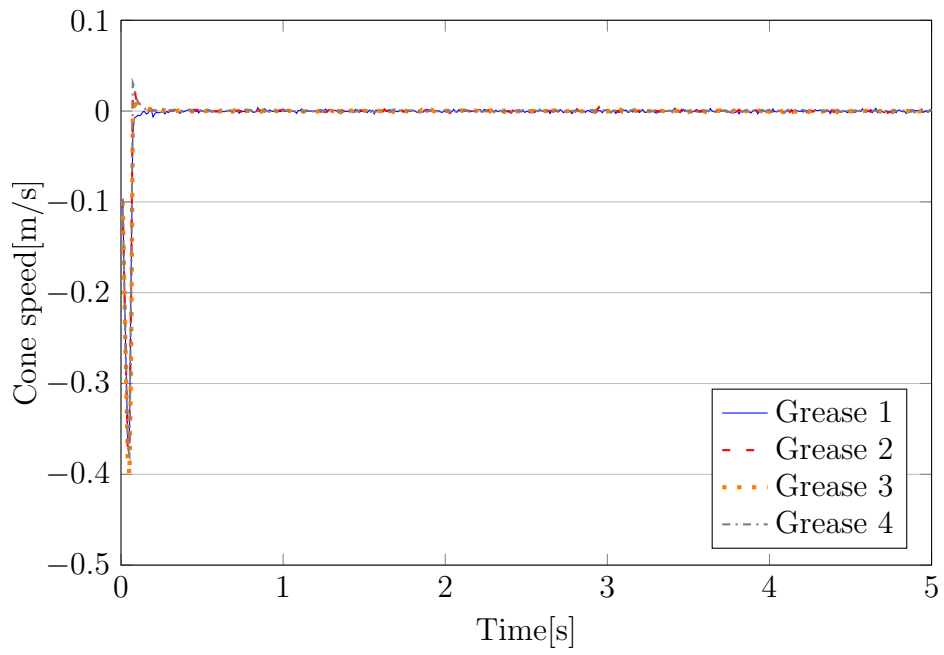
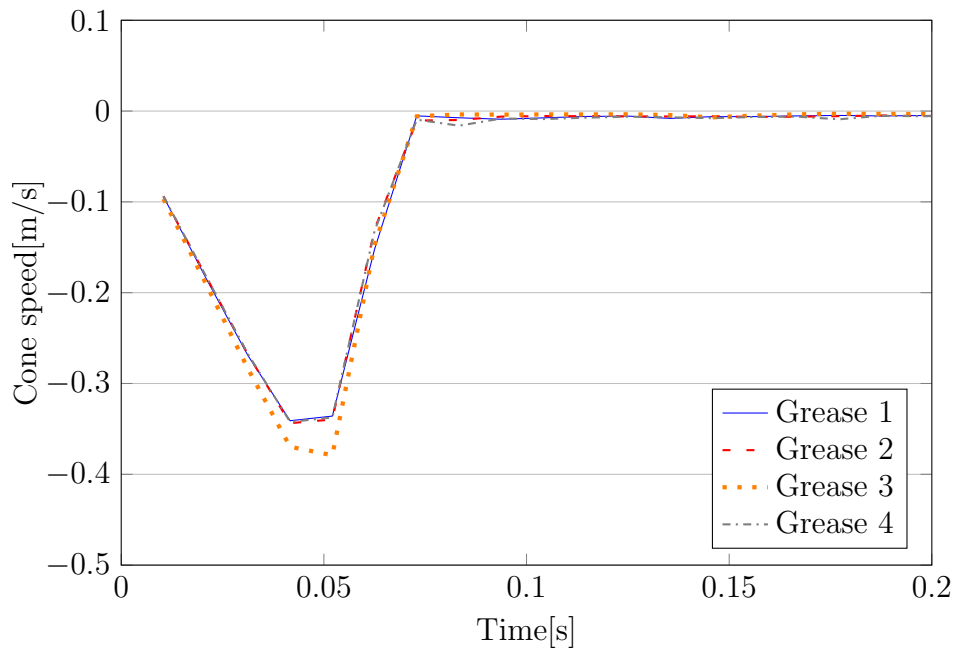
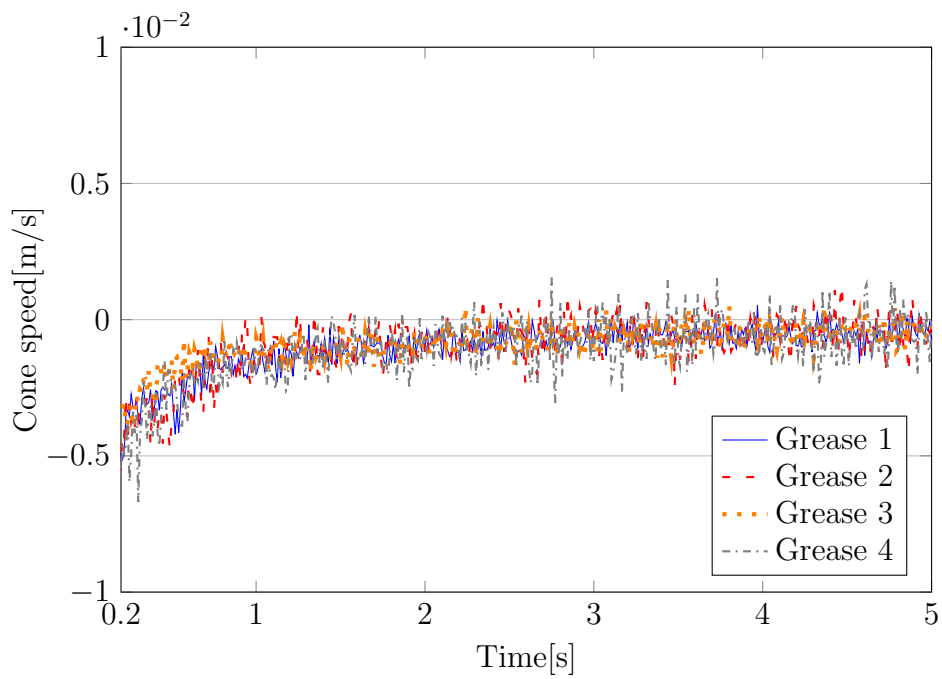


Figure 3.12: Cone speeds over time with elasticity

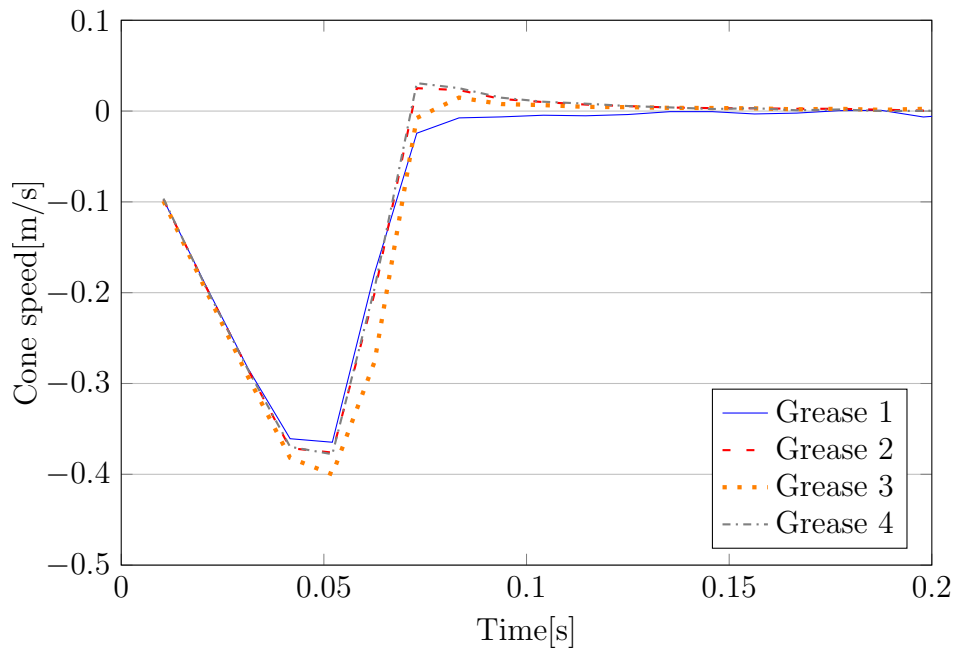


(a) During the initial drop

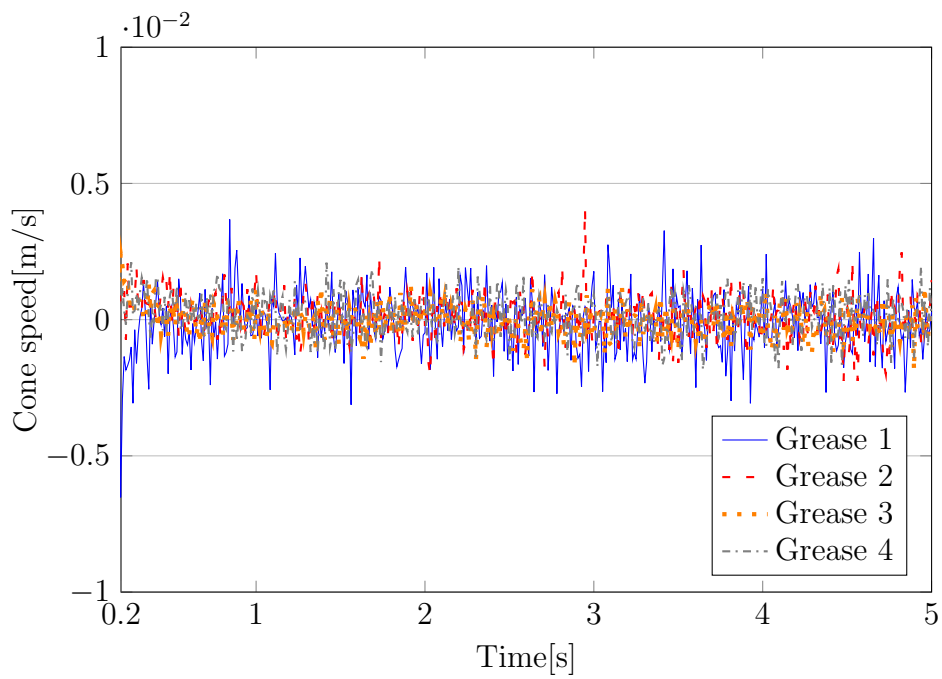


(b) After the initial drop

Figure 3.13: Cone speeds breakdown



(a) During the initial drop



(b) After the initial drop

Figure 3.14: Cone speeds breakdown

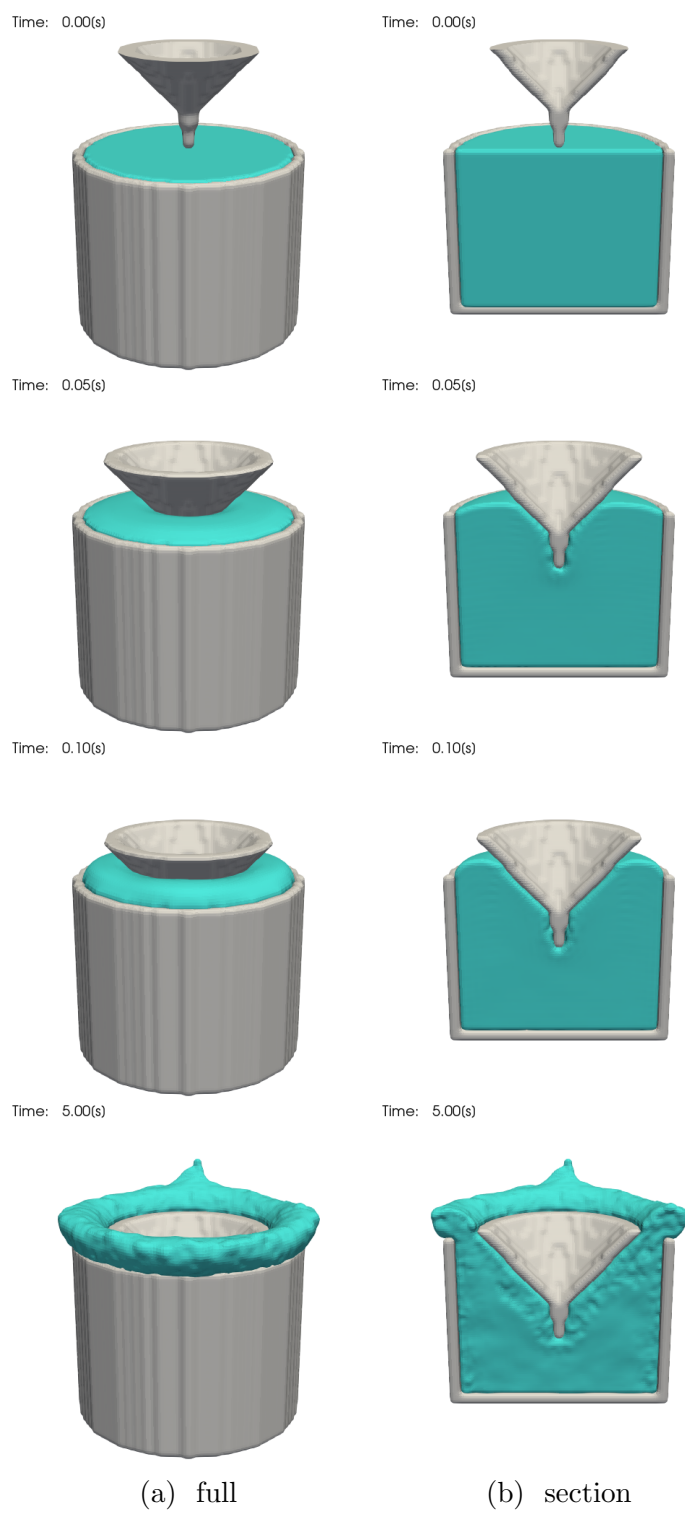


Figure 3.15: Grease 4 in half-size penetrometer in a chronological order without elasticity
109

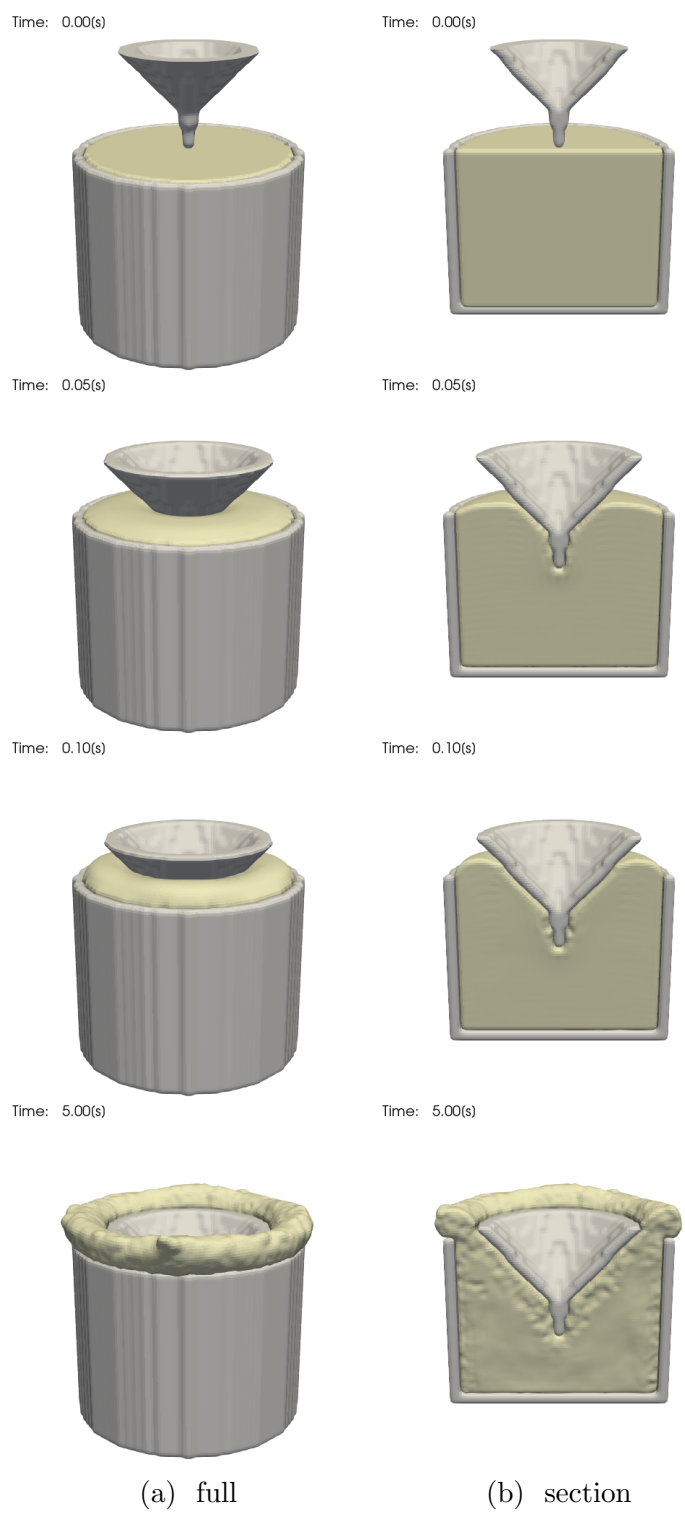


Figure 3.16: Grease 2 in half-size penetrometer in a chronological order without elasticity

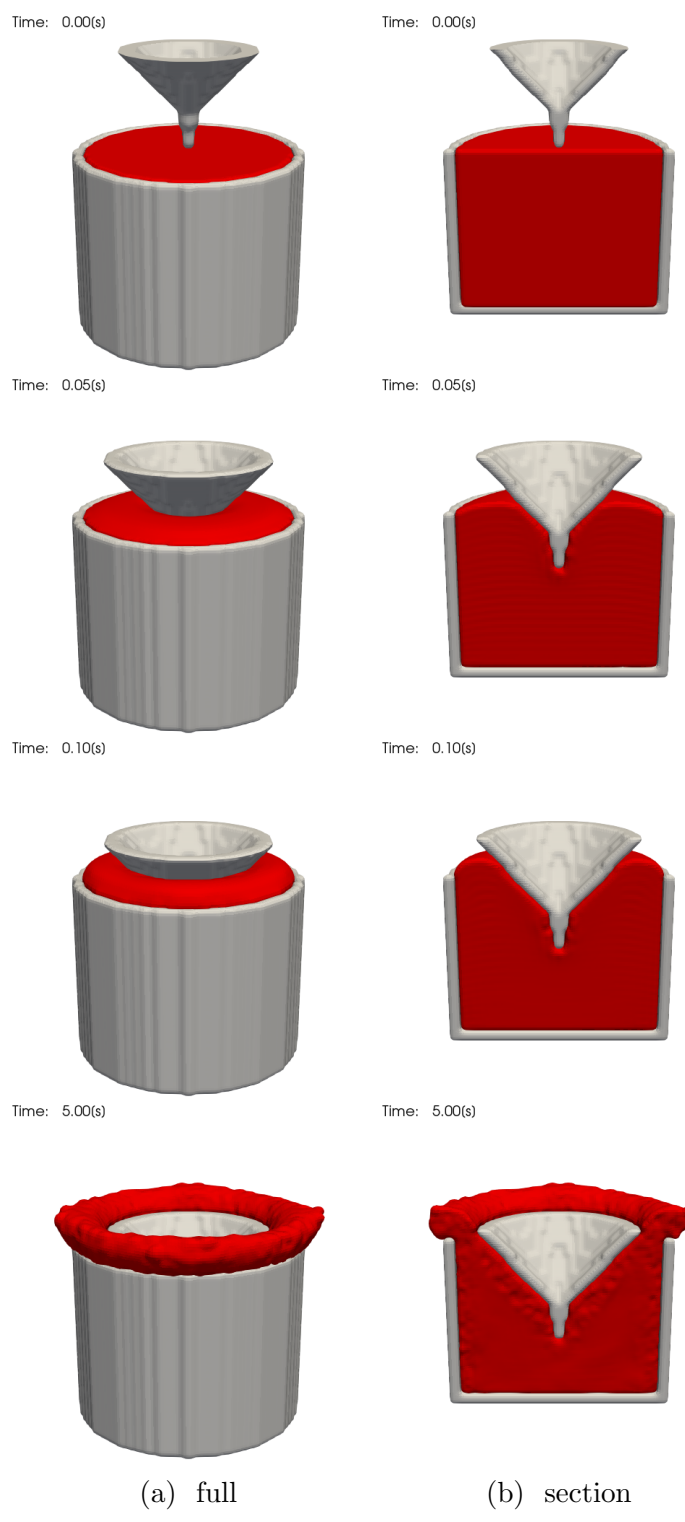


Figure 3.17: Grease 1 in half-size penetrometer in a chronological order without elasticity

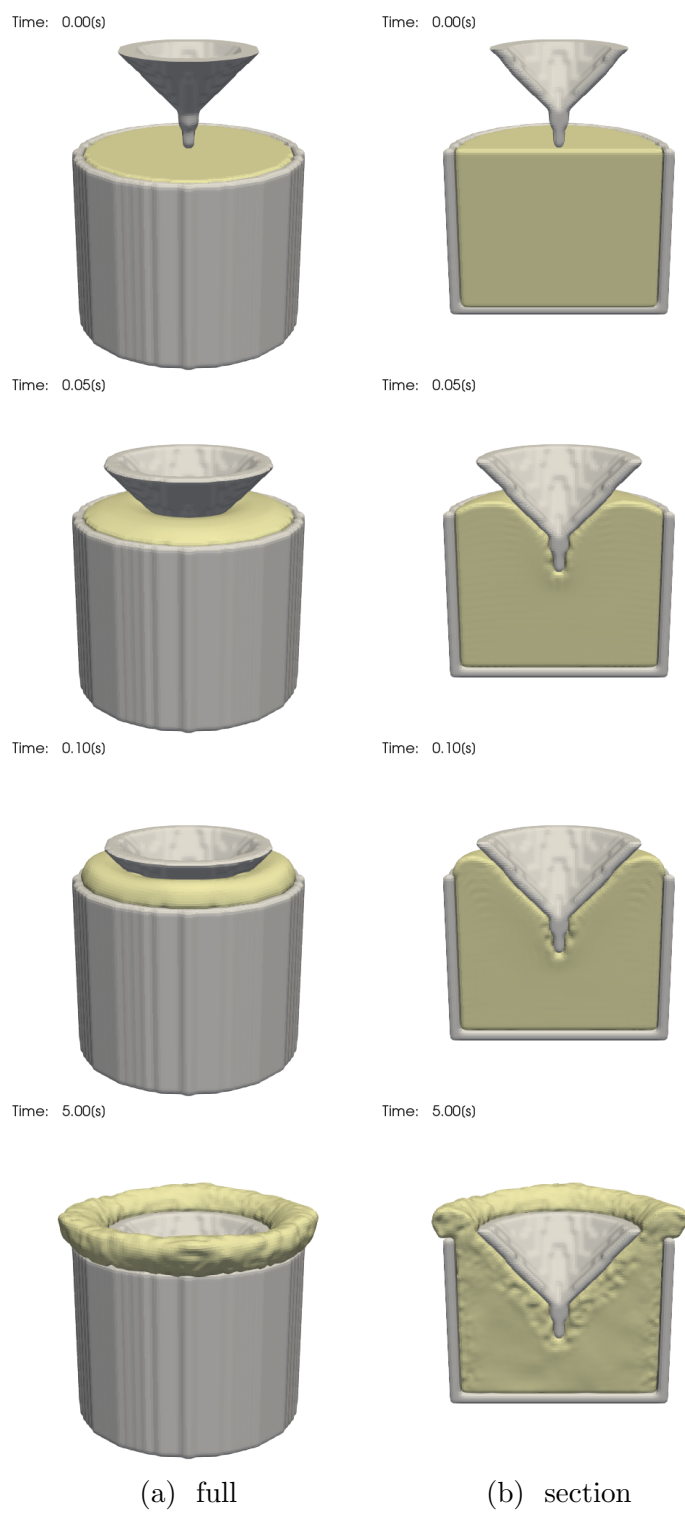


Figure 3.18: Grease 3 in half-size penetrometer in a chronological order without elasticity
112

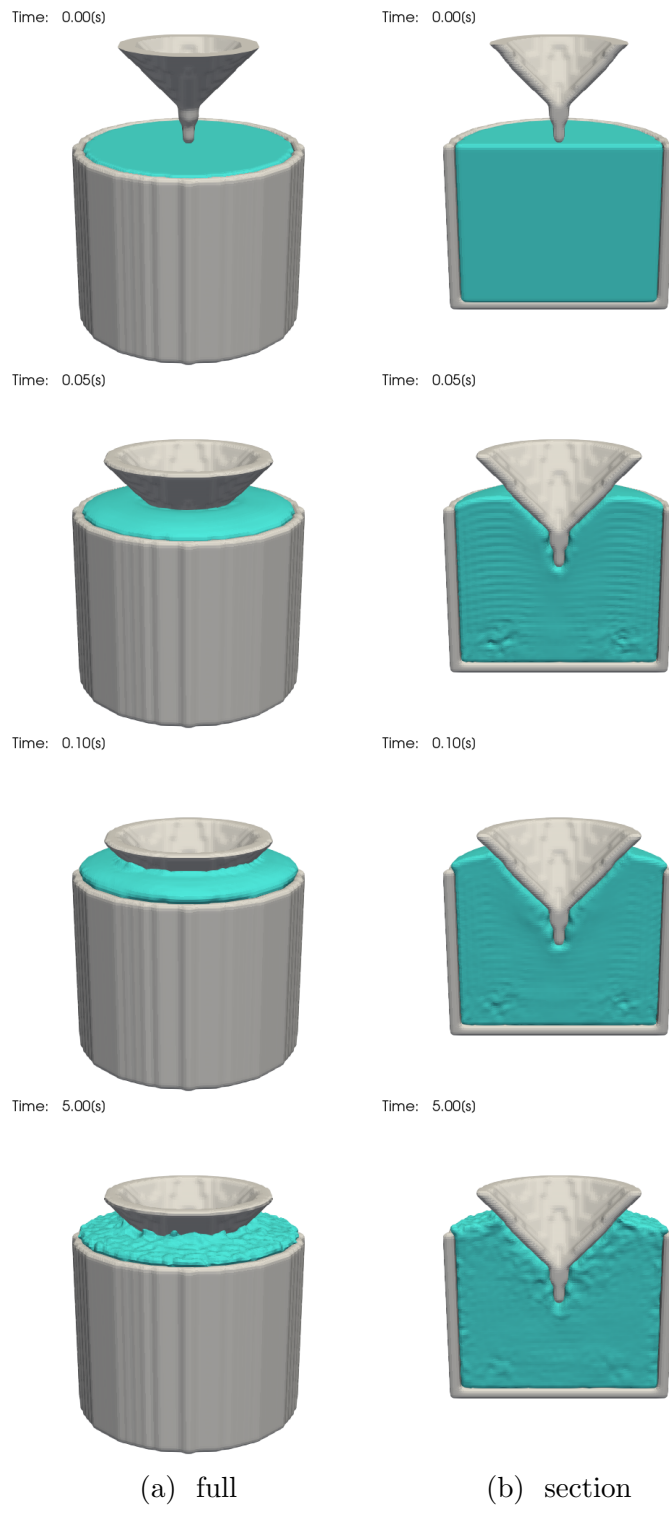


Figure 3.19: Grease 4 in half-size penetrometer in a chronological order with elasticity

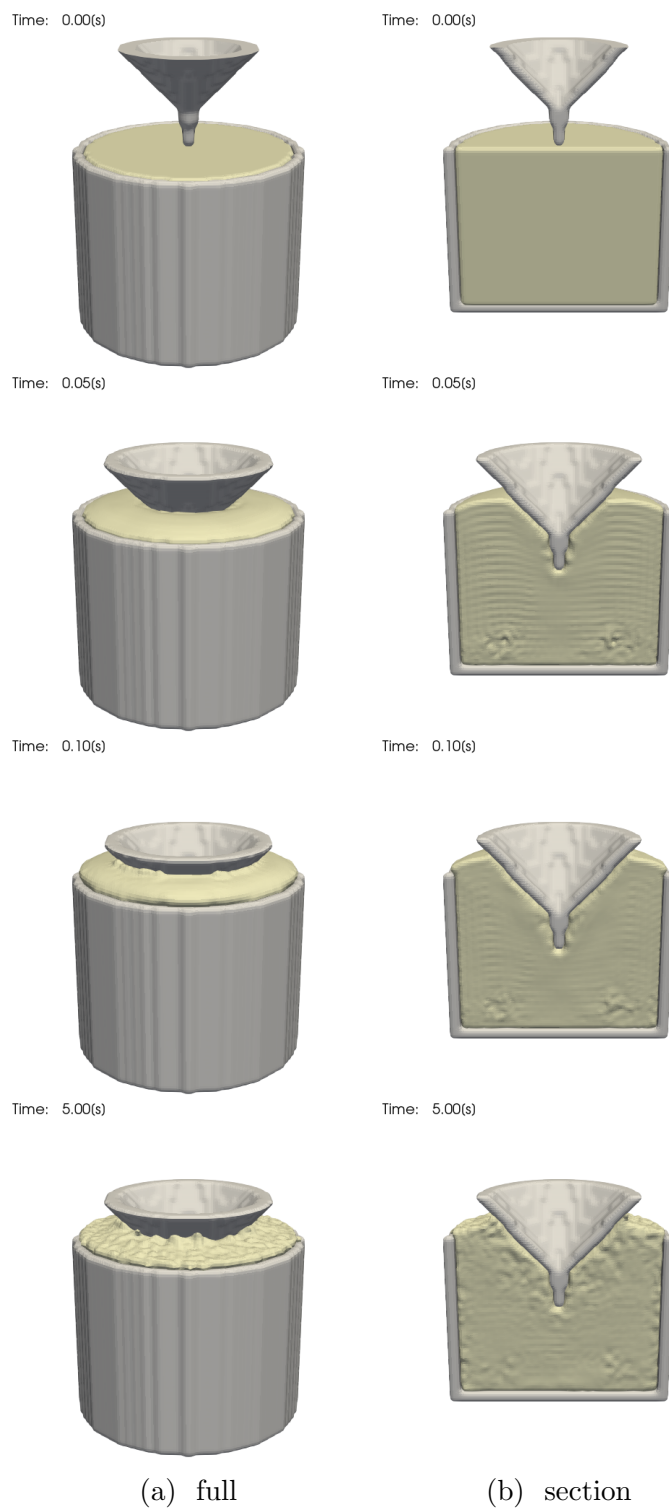


Figure 3.20: Grease 1 in half-size penetrometer in a chronological order with elasticity

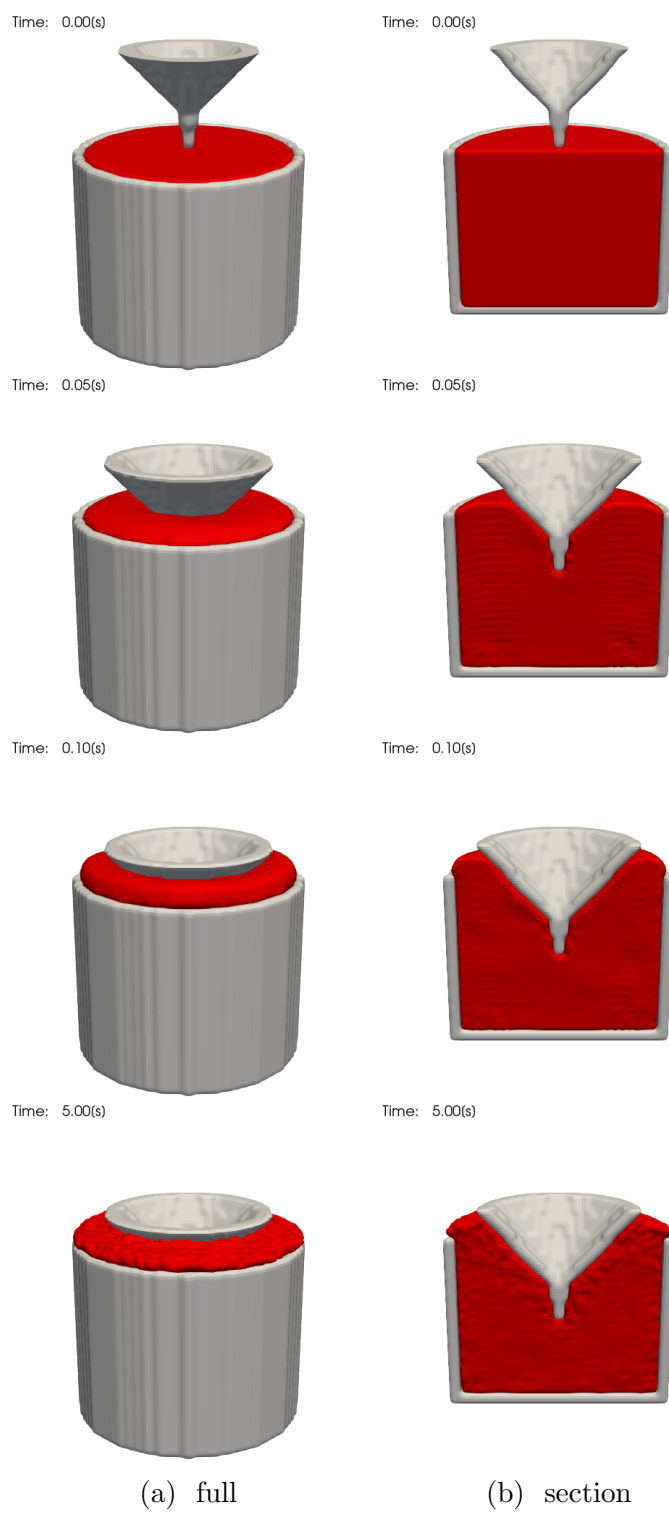


Figure 3.21: Grease 1 in half-size penetrometer in a chronological order with elasticity

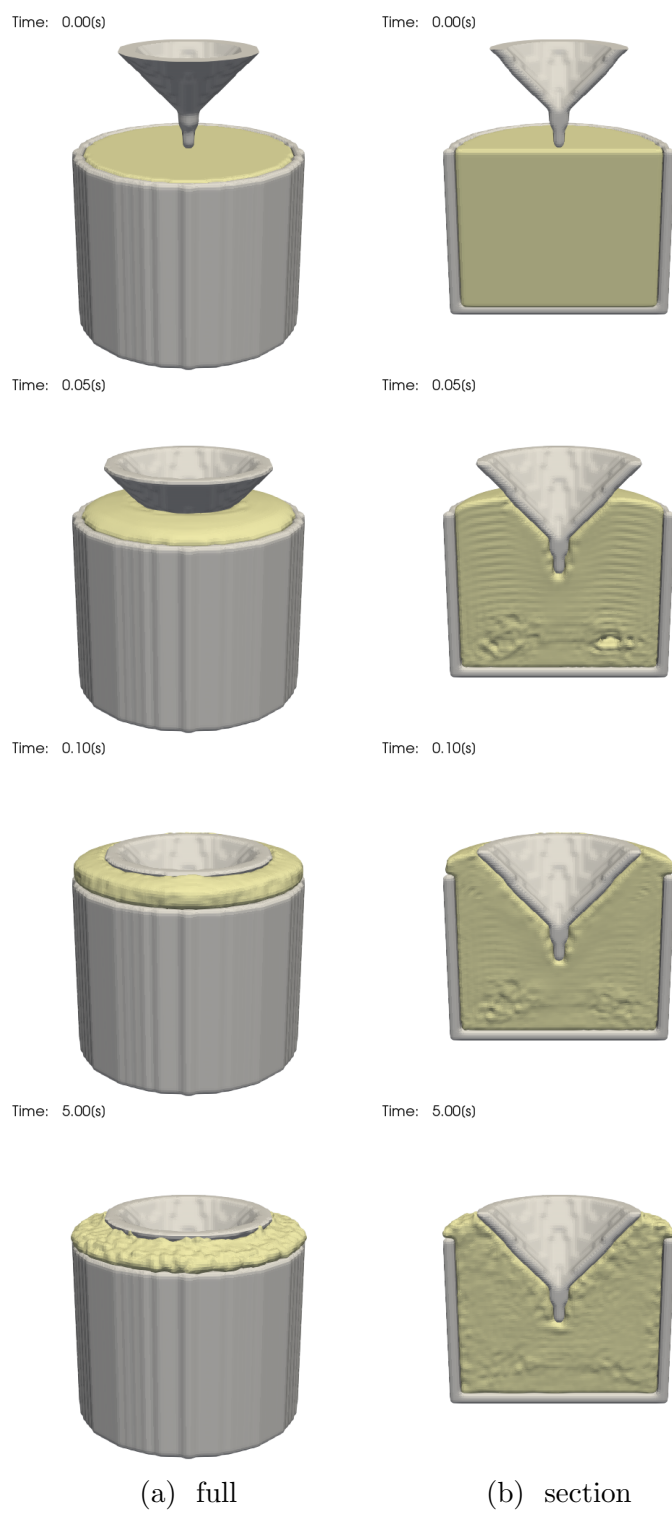


Figure 3.22: Grease 3 in half-size penetrometer in a chronological order with elasticity

Considering that the results of the simulation became significantly closer to the experiment when elasticity was taken into consideration(Figs. 3.7, 3.8), the difference between greases can be largely attributed to elasticity. In fact, the elastic moduli and NLGI consistency in Fig. 3.8 are exactly in the opposite order. To verify this theory, further analyses have been made by breaking down the force that grease exerts on the cone into elastic, viscous, and static pressure components.

The elastic component(Fig. 3.23) indeed supports the above argument. Except the initial peaks of greases 3 and 4, the larger the elastic component is, the smaller the NLGI consistency.

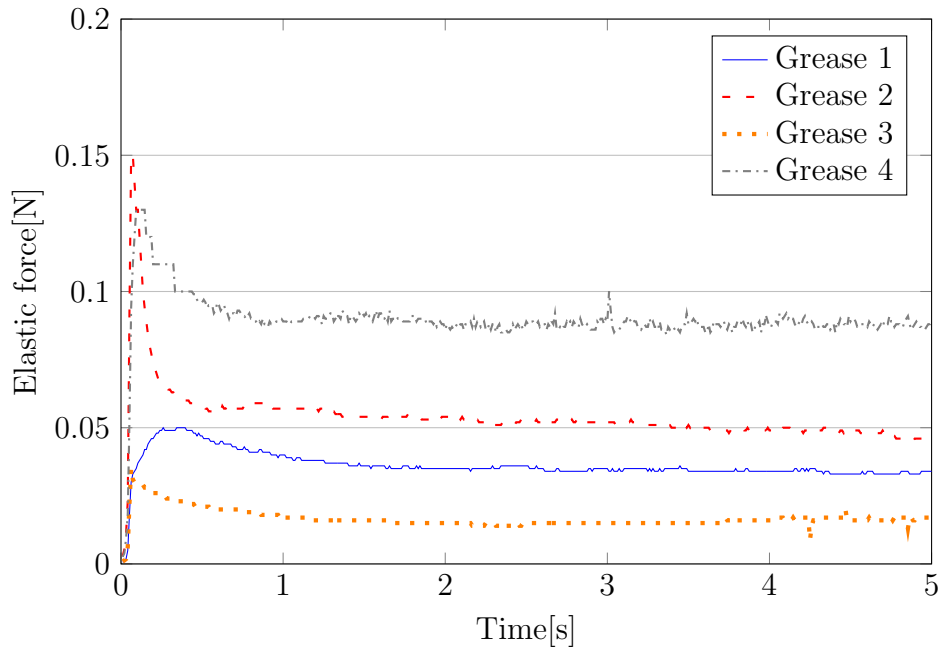


Figure 3.23: Elastic component of force exerting on cone

After the initial fall, the cone drastically decelerates, but at different depths depending on grease. At this stage, the cone is partly supported by the static pressure to stay afloat, and the magnitude should be larger when the cone has penetrated deeply, i.e. for softer greases. Fig. 3.24 shows the static pressure, calculated by subtracting the elastic and viscous components from the total force that fluid exerts on the cone. Despite fluctuations, which will be addressed shortly, a trend can be seen, where grease 4 is the smallest, 1 and 3 are the largest, with 2 in the middle, meaning larger static pressure components for deeper penetrations.

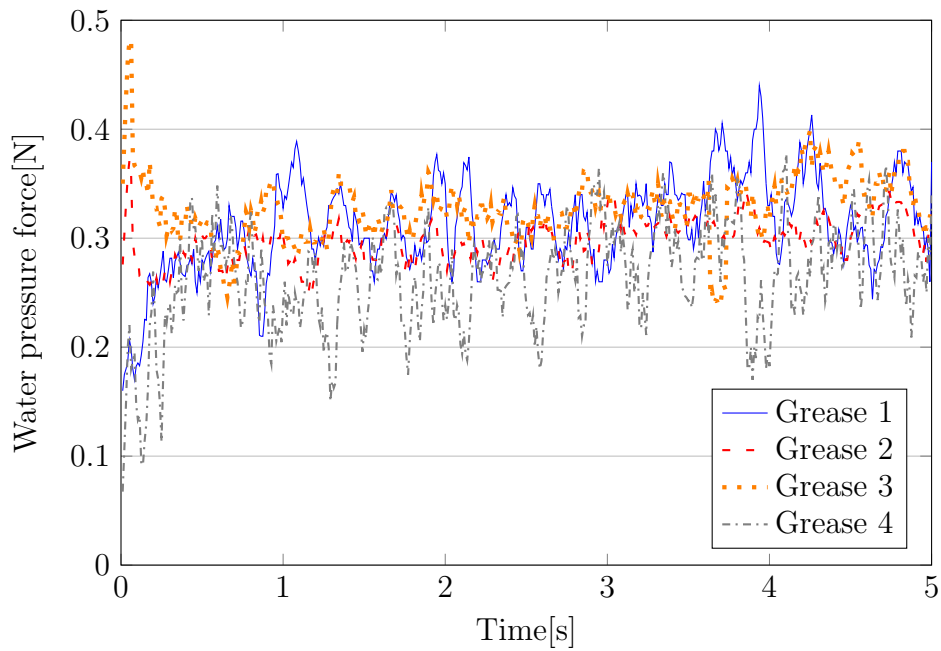


Figure 3.24: Static pressure component of force exerting on cone

Finally, Fig. 3.25 shows the viscous component. The fluctuation is likely caused by numerical inaccuracy. This could happen, for example, because in particle methods, the neighbour particles change constantly, meaning their contribution can also constantly appear and disappear. There were no notable fluctuations in the elastic component, and this is likely because the viscous component is based on the strain rate field or the velocity field, and the elastic component on the strain, which is essentially the strain rate integrated over time, smoothing out the fluctuations. Over the five seconds, the graphs cross each other from the beginning until the end, and finding a trend is implausible.

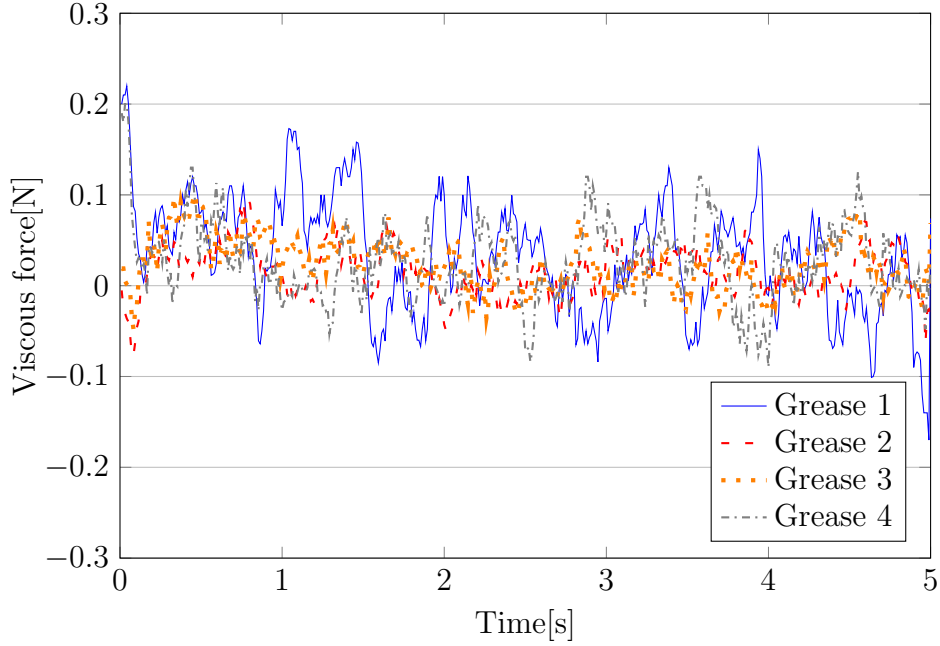


Figure 3.25: Viscous component of force exerting on cone

One of the research objectives was the speed-up of simulation, and therefore the runtime will be analysed. As discussed in the previous chapter, the GPU code in this study has achieved a drastic speed-up from the CPU equivalent. In the case of the cone penetration test, the estimated speed-up is about 40 times since the particle count is similar to the 3D large dam break benchmark. Nonetheless, the runtime was still 17.6 hours for 5 physical seconds, which might be slow considering some earlier studies even achieved real-time simulation in some cases. One reason for such a long runtime is the time step size required by the CFL condition for diffusion(eq. 3.4).

$$\begin{cases} CFL^{conv} &= \frac{|\mathbf{v}|\Delta t}{h} \\ CFL^{diff} &= \frac{\mu\Delta t}{\rho h^2} \end{cases} \quad (3.4)$$

Studies of particle methods empirically suggest both CFL^{conv} and CFL^{diff} need to be lower than 0.1 or 0.25 for stability. In this study, the former was satisfied, but the latter was not, as in Table 3.8.

Table 3.5: Maximum CFL in cone penetration

	grease 1	grease 2	grease 3	grease 4
$CFL^{conv}_{,max}$	4.6e-3	5.0e-3	5.0e-3	5.0e-3
$CFL^{diff}_{,max}$	0.86	0.86	1.2	0.64

Trial and error revealed that larger time steps result in instability, where some particles fly away in an unphysical manner. In the worst case, the vast majority of the particles did so, yielding no results that deserve any physical analysis. In addition to the CFL condition analysis, runtime breakdown was performed for further code optimisation as in Fig. 3.26. The breakdown revealed that the vast majority of the runtime is occupied by `compute_force`. Therefore, it would be essential to tweak this part if any optimisations were to be attempted although tricks have already been extensively implemented.

The particle file used in the half-size cone penetration test is shown in Fig. 3.27. The parameters are shown in 3.7 except grease material properties, which are shown in Table 3.2.

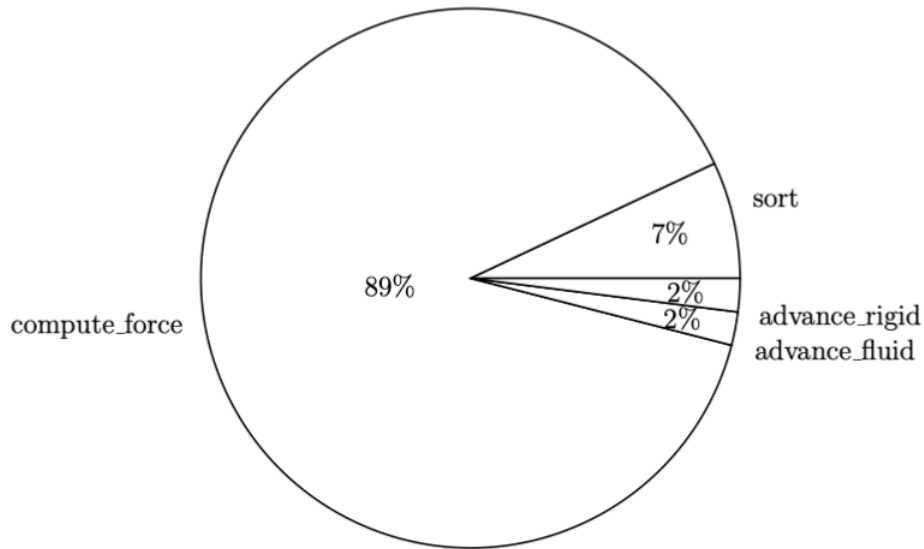


Figure 3.26: Runtime breakdown in cone penetration

Table 3.6: Runtime and breakdown in cone penetration

total physical time [<i>s</i>]	5.0
total runtime [<i>hr</i>]	17.6
whole time step [<i>ms</i>]	127
sort per step [<i>ms</i>]	9
compute force per step [<i>ms</i>]	113
advance fluid per step [<i>ms</i>]	1.9
advance rigid per step [<i>ms</i>]	2.6

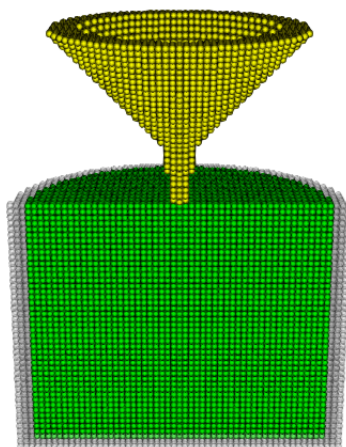


Figure 3.27: Section of particle file used in half-size penetrometer

Table 3.7: Parameters used in in half-size penetrometer.
 Physics parameters are shown in Table 3.2

particle size [m]	7.5e-4
kernel radius [m]	1.875e-3
particle count	118,051
time step [s]	1e-5
velocity blending coefficient	0.8
EOS coefficient B	3.0e+3
shift coefficient	-0.1
surface particle volume coefficient	1.1
surface particle number of neighbours	90
surface cohesion coefficient [N/m]	0.5
cone weight [g]	37.5

3.3 Plate separation

3.3.1 Introduction

Grease exhibits tackiness, which in the case of railway lubrication, was a key feature since it lets transfer to the train wheel [61]. Tackiness can also be a key for the lubrication of rolling bearings. This is because as the rolling elements move, certain grease will be pulled apart, and its response will be affected by tackiness, which will in turn affect the grease distribution. This will eventually affect the performance, such as torque, noise, and life. Moreover, measuring tackiness along with cone penetration would allow a more thorough evaluation of the grease behaviour, since grease undergoes very different forces in these two tests. In cone penetration, grease resists the pressing cone, whereas in plate separation, grease is pulled apart. In applications such as bearings or gearboxes, grease experiences both pressing and pulling apart, and therefore the combination of these tests is thought to be effective.

However, tackiness has not yet been examined in the previous studies of grease simulation. Nor does there exist any standardised measurement method. Here, this study proposes a new method to quantify an aspect of tackiness by measuring the break-off length of the sample when it is placed between two parallel plates and the top plate lifts as opposed to measuring force in previous studies [2][50][61]. The motivation is to focus on how grease flows where. This is because as mentioned in the previous chapter, grease is thought to be pulled apart in mechanical parts, for example, around the ball of the ball bearing (Fig. 1.17), and the degree of grease stretch will affect the subsequent grease distribution, which will eventually impact the performance, such as torque and life of the ball bearing. As discussed earlier, unlike the cone penetration test, the tackiness test is yet to be standardised, and it would be possible to use different geometries other than the rheometer or to use pull-off force as the indicator depending on the purpose.

The experiment was conducted by an NSK employee on the author's behalf on 17th November 2022. The Anton Paar's rotational rheometer MCR302 owned by NSK was used, the same instrument used to obtain the grease viscosities mentioned in §3.1.

In addition to the more conventionally used rotation and oscillation, this instrument has a speed-controlled lift function. The attachment used in this case is a 25mm diameter parallel plate in this case. This lifting function was used in the plate separation test. Each grease was tested twice. The average value from these tests will be compared to the simulation results, which are shown with error bars in Fig. 3.28. The temperature is controlled at 25 °C

by the instrument.

3.3.2 Results of simulation

Fig. 3.28 compares the break-off lengths in the experiment and the simulation. An agreement was obtained in the trend, but the break-off length in the simulation was smaller for all the greases tested, with the largest discrepancy occurring for Grease 1. Figs. 3.29 - 3.32 show the motion of the greases during the plate separation test. For all greases, as the top plate ascended, necking occurred around the middle part (the second frame in each of the figures), which agrees with the experiment results. When it comes to the individual greases, in the case of Grease 4, which resulted in the smallest break-off length both in experiment and simulation, one notable discrepancy is seen in the second frame. In the simulation, the necking is more drastic with the thinnest middle part significantly thinner than experiment. Moreover, the pointed peaks on the grease surface disappear after the break-off whereas they clearly keep their shape in the experiment. Grease 2 showed a very similar result to Grease 4 except the necking was not as significant in the second frame (Fig. 3.30). In this sense, the discrepancy with the experiment was smaller than Grease 4. Grease 3 showed the best agreement with the experiment. The necking in the second frame was milder than the previous two greases (Fig. 3.31). Furthermore, the peaks remained after the break-off. In terms of the necking and the remaining peaks, Grease 1 was similar to Grease 3. Nonetheless, the cut-off length was significantly smaller than the experiment. One possible reason is the spatial resolution of the simulation. When the grease is about to break off, the thinnest part becomes extremely thin in the experiment, presumably exceeding the resolution. The simulation with these conditions cannot capture this and yields a less pronounced cut-off length.

One possible reason for the above-mentioned discrepancy is the use of arbitrary parameters in the simulation. Although surface cohesion was necessary to replicate the stretching behaviour of fluids, the value of the surface cohesion was arbitrarily chosen for numerical stability. Moreover, the measurement of the surface cohesion of grease would be challenging, and the surface cohesion model does not perfectly correspond to the physical equation as mentioned in the previous chapter. The handling of pressure is another instance where arbitrariness was unavoidable. The EOS requires the use of an arbitrary low coefficient for stability. Negative pressure needs to be clamped to zero on the free surface, where the surface detection is based on the number of neighbour particles and the particle volume, and both these values are again decided arbitrarily to balance stability and stretching motion. A small

threshold for neighbour particles, or a large threshold for particle volume categorises more particles as surface. This leads to more stability, but less pronounced stretch. The opposite is more stretch, but stability would be compromised. Surface detection appears to be one of the difficulties when simulating free surfaces with particle methods. All the above arbitrariness combined can be the reason for the discrepancy in terms of the numerical method. There is another possible reason for the discrepancy in terms of physics. Elasticity is assumed to follow Hooke's law for simplicity, but this may not necessarily apply to actual grease. A more complex model may be more appropriate, but this would be a topic for future research.

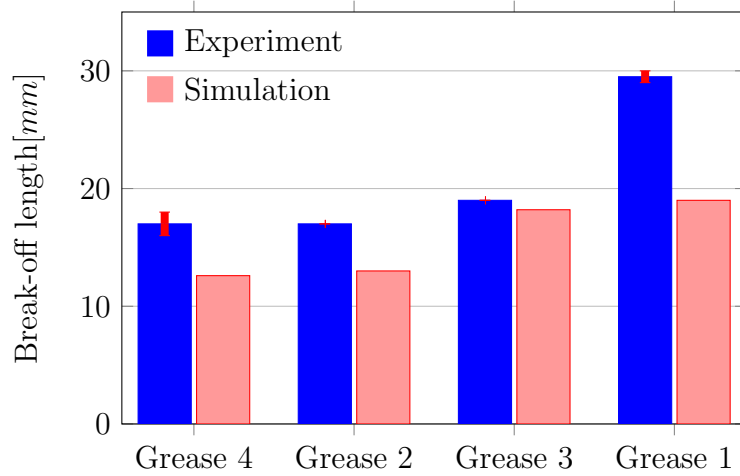


Figure 3.28: Break-off lengths in plate separation test

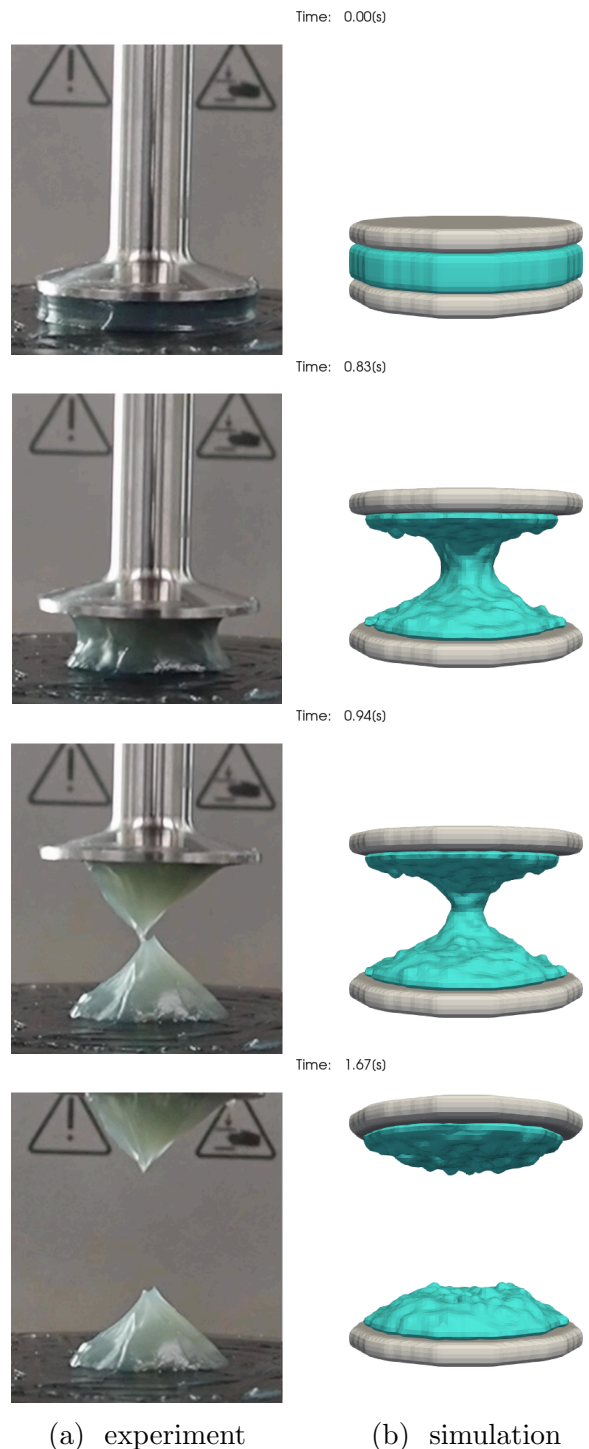


Figure 3.29: Grease Grease 4 in plate separation in plate separation, beginning, during stretch, break-off moment, after break-off

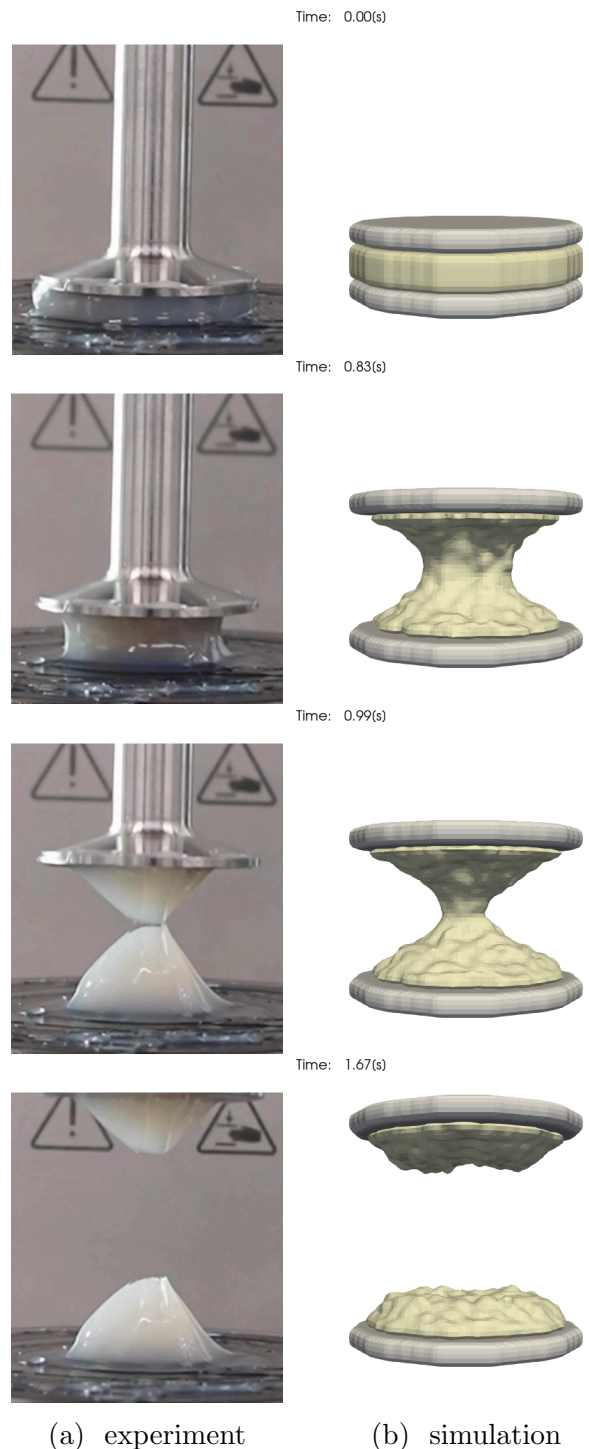


Figure 3.30: Grease Grease 2 in plate separation in plate separation, beginning, during stretch, break-off moment, after break-off

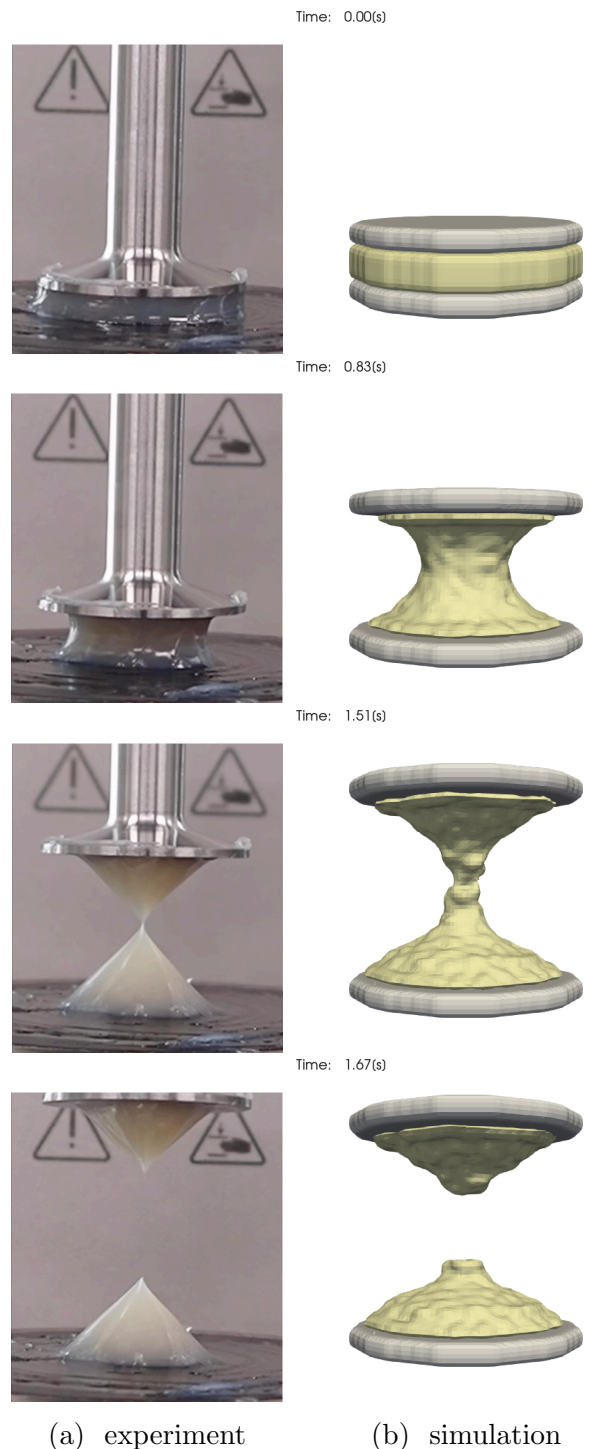


Figure 3.31: Grease Grease 3 in plate separation in plate separation, beginning, during stretch, break-off moment, after break-off

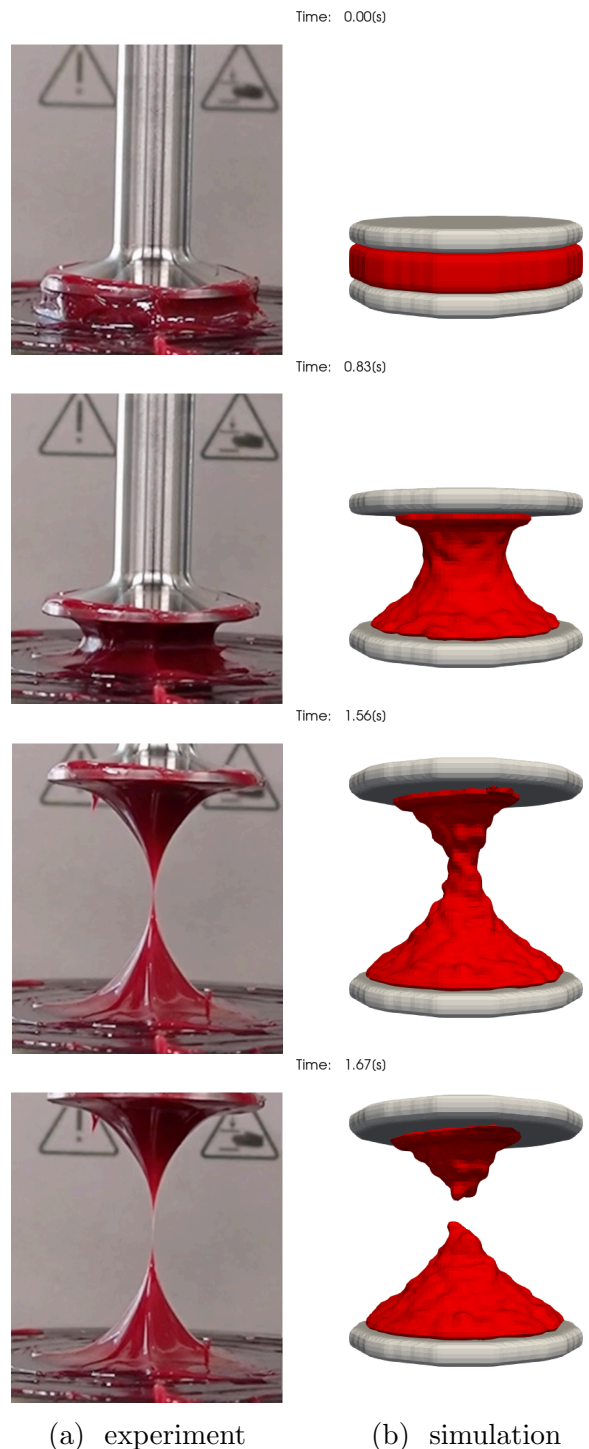
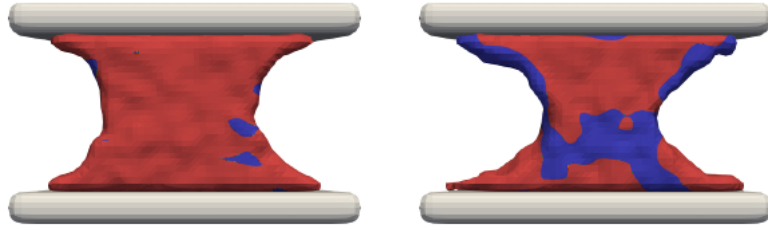
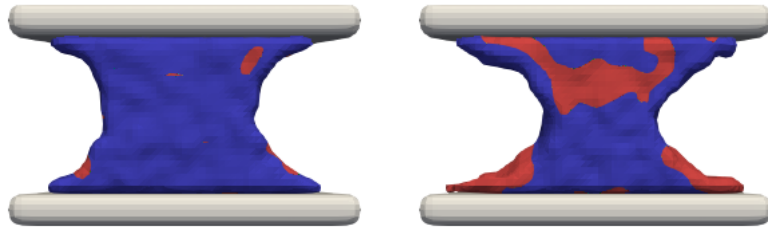


Figure 3.32: Grease Grease 1 in plate separation, beginning, during stretch, break-off moment, after break-off

To further analyse the different cut-off lengths by grease, the effect of elastic force was investigated for greases 1 and 4, which had the longest and shortest cut-off lengths respectively. These greases had the smallest and largest extensional elastic moduli (20 and 400 $[Pa]$ respectively), which is thought to be the main reason for the difference in cut-off length. Fig. 3.33 shows the sections of the greases coloured with the equivalent of elastic force, the contribution of elasticity to the acceleration in the downward/upward directions. In the top two subfigures, blue indicates the regions where the downward acceleration due to elasticity is larger than a certain value ($50m/s^2$). In the bottom centre parts below the necking parts, grease 4 on the right has a broader area coloured blue than grease 1. This means that elasticity pulls the bottom grease downwards, acting as a force to cut the grease off. The asymmetric pattern is likely due to numerical errors. Similarly in the bottom two subfigures, red indicates the regions where the upward acceleration due to elasticity is larger than a certain value ($50m/s^2$). Again in grease 4, there is a broader area coloured red in the top centre part, where the top grease is pulled upwards. To summarise, a larger elastic modulus results in a force that separates the bulk grease in this test.



(a) Downward acceleration higher than $50m/s^2$ marked blue



(b) Upward acceleration higher than $50m/s^2$ marked red

Figure 3.33: Sections of greases coloured with downward/upward acceleration due to elasticity at $0.73[s]$. Left: grease1, right: grease 4

In the interest of the simulation speed, the runtime was analysed. Table 3.9 shows the breakdown in the plate separation test. The whole time step took $12[ms]$, which is over 10 times faster than $127[ms]$ of the cone penetration test. This is likely down to the scale of each case, with the particle counts approximately 9k and 118k in the plate separation and the cone penetration respectively. However, even the plate separation fell short of real-time simulation with the entire 3 physical seconds taking $82[min]$. The small time step appears to be the main reason, which was necessary for stability as was the case with the cone penetration test. Again as in Table 3.8, CFL^{diff} was above the stable threshold whereas CFL^{conv} was.

Fig. 3.34 shows the breakdown of the functions in the code for the total runtime. Similar to the cone penetration, `compute_force` occupied the vast majority of 80%, which again indicates that any further speed-up attempts by code optimisation would require modifying this part.

Fig. 3.35 and Table 3.10 show the particle file and conditions in this test respectively.

Table 3.8: Maximum CFL in plate separation

	grease 1	grease 2	grease 3	grease 4
$CFL^{conv}_{,max}$	1.3e-3	1.3e-3	1.2e-3	1.3e-3
$CFL^{diff}_{,max}$	0.71	0.70	1.0	0.50

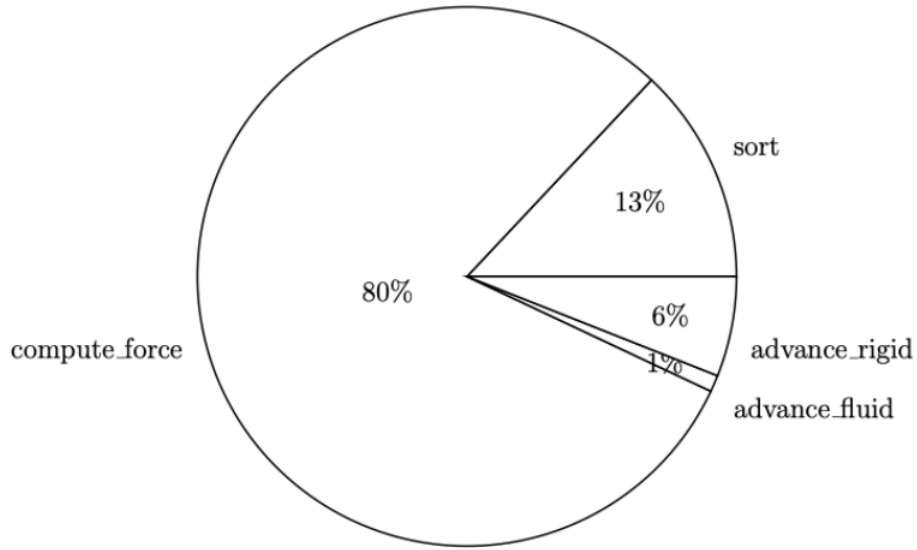


Figure 3.34: Runtime breakdown in plate separation

Table 3.9: Runtime breakdown in plate separation

total physical time [s]	3.0
total runtime [min]	82
Whole time step [ms]	12
sort per step [ms]	1.6
compute force per step [ms]	9.8
advance fluid per step [ms]	0.1
advance rigid per step [ms]	0.7

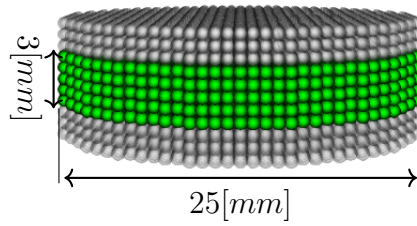


Figure 3.35: Particle file used in plate separation

Table 3.10: Parameters used in in plate separation.
Physics parameters are shown in table 3.2

particle size [m]	7.5e-4
kernel radius [m]	1.875e-3
particle count	9,471
time step [s]	7.5e-6
velocity blending coefficient	0.8
EOS coefficient B	3.0e+3
shift coefficient	-0.1
surface particle volume coefficient	1.1
surface particle number of neighbours	90
surface coheison coefficient [N/m]	0.5

3.4 Conclusions

In previous studies, the difference between greases was hardly ever discussed in CFD. This study addressed this topic in two different aspects and quantified how well the greases were simulated. One is the widely used cone penetration test, representing the response to compression, and the other is

the plate separation test proposed by this study, representing the response to tension.

In terms of grease modelling, this study is the first to incorporate elastoplasticity into grease CFD in order to better replicate the solid-like aspect of grease. In fact, it was found highly effective in replicating the trend between greases in the cone penetration test, which the model without elastoplasticity failed to. This result suggests that it is mostly elasticity that the test largely quantifies, which previous studies referred to as "firmness", without specifying either elasticity or viscosity. In the plate separation test, the trend between greases roughly agreed between the simulation and experiment, but the simulation underestimated the cut-off distance in all the cases.

The speed of the simulation was also measured, but due to the time step size constraint, the overall performance was somewhat limited.

4 Conclusions and future work

4.1 Conclusions

Thanks to its unique desirable features, grease is widely used to lubricate mechanical parts, notably ball bearings. In designing grease-lubricated ball bearings, trial and error is currently required in the experiment since the relations between the design parameters and the product performance are not fully understood. Reducing such trial and error in the experiment would be helpful by executing simulation whilst taking the characteristics of each grease into consideration along with the speed of simulation, which were hardly ever discussed in previous studies.

Considering those needs and challenges, this study attempted to establish a fast and accurate CFD method. More specifically, this study implemented the RKPM on the GPU from scratch with additional techniques, such as particle shifting and surface cohesion that enabled natural results in the 3D surface flow with stability. The performance was compared to the serial CPU equivalent, and the gain by GPU parallelisation was significant, but at the same time, the overall performance was somewhat limited due to the time step size constraint arising from the high viscosities of grease. However, this issue is not specific to CPU computing. Implicit schemes would allow larger time steps, but this would increase the computational cost per time step due to the global matrices, and the estimated cost exceeds the benefit. A higher-speed linear solver for the GPU is a much-needed yet unfulfilled requirement.

Previously, the difference between greases was hardly ever discussed in CFD. This study addressed this topic in two different aspects and quantified how well the greases were simulated. One is the widely used cone penetration test, representing the response to compression, and the other is the plate separation test proposed by this study, representing the response to tension.

In terms of grease modelling, this study is the first to incorporate elastoplasticity into grease CFD in order to better replicate the solid-like aspect of grease. In fact, it was found highly effective in replicating the trend between greases in the cone penetration test, which the model without elastoplasticity failed to. This result suggests that it is mostly elasticity that the test largely quantifies, which previous studies referred to as "firmness", without specifying either elasticity or viscosity. In the plate separation test, the trend between greases roughly agreed between the simulation and experiment, but the simulation underestimated the cut-off distance in all the cases.

4.2 Future work

The combination of strain rate-dependent viscosity and elastoplasticity improved the solidness of grease compared to the case with only strain rate-dependent viscosity. However, more precise modelling of grease would be helpful. Firstly, parameter determination with less arbitrariness to capture the grease characteristics more accurately. Moreover, there could be better modelling for elastoplasticity other than the combination of linear elasticity and von Mises criterion. Storage and loss moduli are often mentioned in the works of grease, but without any links to constitutive models.

Some of the well-known material properties of grease were not considered in this study, i.e. phase separation, thixotropy, and temperature dependency of viscosity. The combination of viscosity and elastoplasticity involved the relations between stress and strain and strain rate, but adding any of the said characteristics would add new dimensions like time, temperature and phase concentration, which would make the modelling process even more complex.

Nonetheless, to start with, thixotropy could be an important factor in replicating the torque of ball bearings. It could be tested by simulating stress response under a constant strain rate using a rotational rheometer. Inventing a new method to measure its effect on the free-surface flow would also be beneficial as the uniqueness of grease often emerges in the presence of free surfaces.

Phase separation of grease, in the context of ball bearings, is believed to be the source of oil to the contact points during hundreds of hours of operation. The simulation of the flow of separated oil could provide a useful insight into the lubrication mechanism and a hint for better lubrication conditions including the initial grease placement and separation characteristics. However, this would require clever tactics to avoid the expected long computation as simulating hundreds of hours of physical time would be impractical.

Modelling viscosity as an explicit function could be useful when the grease behaviour over tens and hundreds of hours is of interest, where grease will soften with the temperature rise. However, this means tens and hundreds of hours of simulation needs to be realised first, which would be a major challenge.

References

- [1] (2024). <https://github.com/g-truc/glm>, accessed on 21st Maay 2024.
- [2] Achanta, S., Jungk, M., and Drees, D. (2011). Characterisation of cohesion, adhesion, and tackiness of lubricating greases using approachretraction experiments. *Tribology International*, 44(10):1127–1133.
- [3] Akbari, H. (2017). Simulation of wave overtopping using an improved SPH method. *Coastal Engineering*, 126(October 2016):51–68.
- [4] Akinci, N., Akinci, G., and Teschner, M. (2013). Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics*, 32(6).
- [5] Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., and Teschner, M. (2012). Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics*, 31(4).
- [6] Altomare, C., Tafuni, A., Domínguez, J. M., Crespo, A. J., Gironella, X., and Sospedra, J. (2020). SPH simulations of real sea waves impacting a large-scale structure. *Journal of Marine Science and Engineering*, 8(10):1–21.
- [7] Arimitsu, K., Chang, Y., and Mukai, N. (2012). Spinnability Simulation of Viscoelastic Fluid based on Combined Constitutive Equation. *ITE Annual Convention 2012*, (3):2–3.
- [8] ASTM International (2015). Standard Test Methods for Cone Penetration of Lubricating Grease. *Annual Book of ASTM Standards*, 05(92):1–9.
- [9] Augusto, L. L., Sinatora, A., Negrão, C. O., and Cousseau, T. (2020). A numerical investigation of grease friction losses in labyrinth seals. *Tribology International*, 141(September 2019).
- [10] Awati, B. V. and Naik, S. (2017). A multigrid method for EHL line contact problem with Grease as lubricant. *Mathematical Models in Engineering*, 3(2):126–134.
- [11] Baart, P., Lugt, P. M., and Prakash, B. (2009). Review of the lubrication, sealing, and pumping mechanisms in oil- and grease-lubricated radial lip seals. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, 223(3):347–358.

- [12] Baart, P., Vorst, B. V. D., Lugt, P. M., and Van Ostayen, R. A. (2010). Oil-bleeding model for lubricating grease based on viscous flow through a porous microstructure. *Tribology Transactions*, 53(3):340–348.
- [13] Baker, A. (1957). Grease Bleeding A Factor in Ball Bearing Performance. *NLGI (September)*, 22:271–277.
- [14] Barreiro, H., García-Fernández, I., Alduán, I., and Otaduy, M. A. (2017). Conformation constraints for efficient viscoelastic fluid simulation. *ACM Transactions on Graphics*, 36(6).
- [15] Bauer, N., Hahn, S., Feldmeth, S., Bauer, F., and Schmitz, K. (2021). Rheological Characterization and EHL Simulation of a Grease in a Lubricated Sealing Contact. *Tribologie und Schmierungstechnik*, 68(6):20–28.
- [16] Bazilevs, Y., Moutsanidis, G., Bueno, J., Kamran, K., Kamensky, D., Hillman, M. C., Gomez, H., and Chen, J. S. (2017). A new formulation for air-blast fluidstructure interaction using an immersed approach: part II coupling of IGA and meshfree discretizations. *Computational Mechanics*, 60(1):101–116.
- [17] Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P. (1996). Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47.
- [18] Bender, J. and Koschier, D. (2015). Divergence-Free Smoothed Particle Hydrodynamics. (1).
- [19] Brodtkorb, A. R., Hagen, T. R., and Sætra, M. L. (2013). Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13.
- [20] Budiana, E. P., Indarto, I., Deendarlianto, D., and Pranowo, P. (2018). Numerical study of meshless radial basis functions in the lid driven cavity problem. In *Frontiers in Neuroscience*, volume 14, page 060011.
- [21] Busquet, M., Renouf, M., Berthier, Y., and Sicre, J. (2017a). Space grease lubrication modeling: A discrete element approach. *Tribology International*, 111(November 2016):159–166.
- [22] Busquet, M., Renouf, M., Leveque, D., Berthier, Y., Bouscharain, N., and Group, I. T. (2017b). Space Grease Tribological Behavior for Reformulation : Numerical. (September):20–22.

- [23] Cann, P. M., Webster, M. N., Doner, J. P., Wikstrom, V., and Lugt, P. (2007). Grease degradation in R0F bearing tests. *Tribology Transactions*, 50(2):187–197.
- [24] Chang, H., Lan, C. W., Chen, C. H., Kao, M. J., and Guo, J. B. (2014). Anti-wear and friction properties of nanoparticles as additives in the lithium grease. *International Journal of Precision Engineering and Manufacturing*, 15(10):2059–2063.
- [25] Chen, J., Bai, X., Wang, J., Chen, G., and Zhang, T. (2022). Research on Sea Spray Distribution of Marine Vessels Based on SPH-FEM Coupling Numerical Simulation Method. *Water (Switzerland)*, 14(23).
- [26] Chen, J. K. and Beraun, J. E. (2000). A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. *Computer Methods in Applied Mechanics and Engineering*, 190(1-2):225–239.
- [27] Chen, J. S., Wu, C. T., Yoon, S., and You, Y. (2001). Stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466.
- [28] Chen, X. S., Li, C. F., Cao, G. C., Jiang, Y. T., and Hu, S. M. (2020). A moving least square reproducing kernel particle method for unified multiphase continuum simulation. *ACM Transactions on Graphics*, 39(6).
- [29] Cheng, H., Radlińska, A., Hillman, M., Liu, F., and Wang, J. (2024). Modeling concrete deposition via 3D printing using reproducing kernel particle method. *Cement and Concrete Research*, 181(April).
- [30] Chi, S.-W., Chen, J.-S., Hu, H.-Y., and P, Y. J. (2012). A gradient reproducing kernel collocation method for boundary value problems. *International*, (February):1381–1402.
- [31] Chow, A. D., Rogers, B. D., Lind, S. J., and Stansby, P. K. (2018). Incompressible SPH (ISPH) with fast Poisson solver on a GPU. *Computer Physics Communications*, 226:81–103.
- [32] Chudnovsky, B. H. (2005). Lubrication of electrical contacts. *Electrical Contacts, Proceedings of the Annual Holm Conference on Electrical Contacts*, 2005:107–114.
- [33] Cousseau, T., Graça, B., Campos, A., and Seabra, J. (2011). Friction torque in grease lubricated thrust ball bearings. *Tribology International*, 44(5):523–531.

- [34] Cousseau, T., Graça, B. M., Campos, A. V., and Seabra, J. H. (2010). Influence of grease formulation on thrust bearings power loss. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, 224(9):935–946.
- [35] Crespo, A. C., Dominguez, J. M., Barreiro, A., Gómez-Gesteira, M., and Rogers, B. D. (2011). GPUs, a new tool of acceleration in CFD: Efficiency and reliability on smoothed particle hydrodynamics methods. *PLoS ONE*, 6(6).
- [36] Cummins, S. J. and Rudman, M. (1999). An SPH Projection Method. *Journal of Computational Physics*, 152(2):584–607.
- [37] Cyriac, F., Lugt, P. M., and Bosman, R. (2016). Yield Stress and Low-Temperature Start-Up Torque of Lubricating Greases. *Tribology Letters*, 63(1):1–10.
- [38] Dally, W. J., Keckler, S. W., and Kirk, D. B. (2021). Evolution of the Graphics Processing Unit (GPU). *IEEE Micro*, 41(6):42–51.
- [39] Delgado, M. A., Sánchez, M. C., Valencia, C., Franco, J. M., and Gallegos, C. (2005). Relationship among microstructure, rheology and processing of a lithium lubricating grease. *Chemical Engineering Research and Design*, 83(9 A):1085–1092.
- [40] Donahue, C. J. (2006). Lubricating grease: A chemical primer. *Journal of Chemical Education*, 83(6):862–869.
- [41] Dowson, D. (1995). Elastohydrodynamic and micro-elastohydrodynamic lubrication. *Wear*, 190(2):125–138.
- [42] Dyka, C. T. and Ngel, R. P. (1994). Addressing Tension Instability in SPH Methods. *Interim Report Naval Research Lab., Washington, DC*.
- [43] Fan, X., Li, W., Li, H., Zhu, M., Xia, Y., and Wang, J. (2018). Probing the effect of thickener on tribological properties of lubricating greases. *Tribology International*, 118(September 2017):128–139.
- [44] Fan, X. J., Tanner, R. I., and Zheng, R. (2010). Smoothed particle hydrodynamics simulation of non-Newtonian moulding flow. *Journal of Non-Newtonian Fluid Mechanics*, 165(5-6):219–226.
- [45] Fan, Z., Qiu, F., Kaufman, A., and Yoakum-Stover, S. (2004). GPU cluster for high performance computing. *Proceedings of the ACM/IEEE SC 2004 Conference: Bridging Communities*, 00(1):47.

- [46] Fikry, R. M., El-Adly, R. A., Ismail, N. A., El-Tabei, A. S., and Al-Aidy, H. (2013). Some azine and azole derivatives as antioxidant additives for lithium lubricating grease. *Egyptian Journal of Petroleum*, 22(1):61–71.
- [47] Florea, O., Luca, M., and Steliean, C. (2004). Ecological lubricating greases. *Tribology in Industry*, 26(1-2):52–57.
- [48] Fog, A. (2014). Optimizing software in C++ An optimization guide. *Online book*, <http://www.agner.org/optimize>.
- [49] Fraga Filho, C. A., Chacaltana, J. T., and Pinto, W. J. (2018). Meshless Lagrangian SPH method applied to isothermal lid-driven cavity flow at low-Re numbers. *Computational Particle Mechanics*, 5(4):467–475.
- [50] Georgiou, E. P., Drees, D., De Bilde, M., Anderson, M., Carlstedt, M., and Mollenhauer, O. (2021). Quantification of tackiness of a grease: The road to a method. *Lubricants*, 9(3).
- [51] Gingold, R. A. and Monaghan, J. J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389.
- [52] Goktekin, T. G. (2011). Animating Viscoelastic Fluids. *Electrical Engineering and Computer Sciences University of California at Berkeley*, (Technical Report No. UCB/EECS-2011-111):70.
- [53] Goktekin, T. G., Bargteil, A. W., and O’Brien, J. F. (2004). A method for animating viscoelastic fluids. *ACM SIGGRAPH 2004 Papers, SIGGRAPH 2004*, m:463–468.
- [54] Goswami, P. and Batty, C. (2014). Regional Time Stepping for SPH. *Eurographics 2014 - Short Papers*, pages 1–5.
- [55] Goswami, P., Eliasson, A., and Franzén, P. (2015). Implicit incompressible SPH on the GPU. *12th Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2015*, pages 23–29.
- [56] Goswami, P., Schlegel, P., Solenthaler, B., and Pajarola, R. (2010). Interactive SPH simulation and rendering on the GPU. *Computer Animation 2010 - ACM SIGGRAPH / Eurographics Symposium Proceedings, SCA 2010*, pages 55–64.
- [57] Green, T. M., Baart, P., Westerberg, L. G., Lundström, T. S., Höglund, E., Lugt, P. M., and Li, J. X. (2011). A new method to visualize grease

- flow in a double restriction seal using microparticle image velocimetry. *Tribology Transactions*, 54(5):784–792.
- [58] Gurt, A. and Khonsari, M. M. (2021). Testing grease consistency. *Lubricants*, 9(2):1–16.
- [59] Hai Mao, Y.-H. Y. (2005). A Particle-Based Model for Non-Newtonian Fluid Animation. *Technical Reports (Computing Science)*.
- [60] Harada, T., Koshizuka, S., and Kawaguchi, Y. (2007). Smoothed Particle Hydrodynamics on GPUs. *Particle-Based Methods II - Fundamentals and Applications*, pages 63–70.
- [61] Harmon, M., Powell, B., Barlebo-Larsen, I., and Lewis, R. (2019). Development of Grease Tackiness Test. *Tribology Transactions*, 62(2):207–217.
- [62] Hartl, A. M., Jerabek, M., and Lang, R. W. (2015). Anisotropy and compression/tension asymmetry of PP containing soft and hard particles and short glass fibers. *Express Polymer Letters*, 9(7):658–670.
- [63] HÉRAULT, A. (2009). SPH on GPU with CUDA. *Journal of Hydraulic Research*, 48(extra):000.
- [64] Hieber, C. A. and Chiang, H. H. (1992). Shear rate dependence modeling of polymer melt viscosity. *Polymer Engineering & Science*, 32(14):931–938.
- [65] Hogenberk, F., Osara, J. A., van den Ende, D., and Lugt, P. M. (2023). On the evolution of oil-separation properties of lubricating greases under shear degradation. *Tribology International*, 179(November 2022):108154.
- [66] Holmberg, K. and Erdemir, A. (2019). The impact of tribology on energy use and CO2 emission globally and in combustion engine and electric cars. *Tribology International*, 135(January):389–396.
- [67] Hu, J. Q., Wei, X. Y., and Zong, Z. M. (2006). Study on tribological and corrosion inhibiting properties of thiadiazole derivative as lubricant grease additive. *Industrial Lubrication and Tribology*, 58(6):320–323.
- [68] Hu, X. Y. and Adams, N. A. (2007). An incompressible multi-phase SPH method. *Journal of Computational Physics*, 227(1):264–278.
- [69] Hua, X., Puoza, J. C., Zhang, P., Yin, B., Xie, X., and Din, J. (2019). Numerical Simulation and Experimental Analysis of Grease Friction Properties on Textured Surface. *Iranian Journal of Science and Technology - Transactions of Mechanical Engineering*, 43(s1):357–369.

- [70] Huang, L., Guo, D., and Shizhu, W. (2016). Film thickness decay and replenishment in point contact lubricated with different greases: A study into oil bleeding and the evolution of lubricant reservoir. *Tribology International*, 93:620–627.
- [71] Ihmsen, M., Cornelis, J., Solenthaler, B., Horvath, C., and Teschner, M. (2014). Implicit incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):426–435.
- [72] ITO, K., MUKAI, N., and KOSUGI, M. (2010). CG . *ITE Technical Report*, 34(15):5–8.
- [73] Jacob, K. H. (2023). Oil Separation of Lubricating Greases under Static Conditions: Analytical Photo-Centrifuge and DIN 51817. *Lubricants*, 11(3).
- [74] Jang, J. Y. and Khonsari, M. M. (1997). Performance Analysis of Grease- Lubricated Journal Bearings. 119(October).
- [75] Ji, Z., Stanic, M., Hartono, E. A., and Chernoray, V. (2018). Numerical simulations of oil flow inside a gearbox by Smoothed Particle Hydrodynamics (SPH) method. *Tribology International*, 127(April):47–58.
- [76] Jiang, T., Ouyang, J., Li, Q., Ren, J., and Yang, B. (2011). A corrected smoothed particle hydrodynamics method for solving transient viscoelastic fluid flows. *Applied Mathematical Modelling*, 35(8):3833–3853.
- [77] Jiang, T., Ouyang, J., Ren, J.-l., Yang, B.-x., and Xu, X.-y. (2012). A mixed corrected symmetric SPH (MC-SSPH) method for computational dynamic problems. *Computer Physics Communications*, 183(1):50–62.
- [78] Johnson, G. R. and Beissel, S. R. (1996). Normalized smoothing functions for sph impact computations. *International Journal for Numerical Methods in Engineering*, 39(16):2725–2741.
- [79] KAMAMOTO, S., MURATA, J., and SHISHIHAR, Y. (2019). Study on Low Torque Deep Groove Ball Bearing by Cage Profile Optimization. , 64(3):186–192.
- [80] Kang, N. and Sagong, D. (2014). Incompressible SPH using the Divergence-Free Condition. *Computer Graphics Forum*, 33(7):219–228.
- [81] Kawamura, M., Sato, N., Sakaguchi, T., and Izumi, M. (2013). Grease Leakage Prevention for Double-Row Angular Ball Bearings. *NTN TECHNICAL REVIEW*, (81):92–99.

- [82] Keller, M. C., Braun, S., Wieth, L., Chaussonnet, G., Dauch, T., Koch, R., Höfler, C., and Bauer, H.-J. (2016). Numerical Modeling of Oil-Jet Lubrication for Spur Gears using Smoothed Particle Hydrodynamics. *Proceedings of the 11th International SPHERIC Workshop*, pages 69–76.
- [83] Kimura, H. and Onuki, Y. (2003). Properties and applications of synthetic greases. *Journal of Synthetic Lubrication*, 20(3):241–255.
- [84] Komoriya, T., Ichimura, R., Kochi, T., Yoshihara, M., Sakai, M., Dong, D., and Kimura, Y. (2021). Service life of lubricating grease in ball bearings (Part 1) behavior of grease and its base oil in a ball bearing. *Tribology Online*, 16(4):236–245.
- [85] Koszkul, J. and Nabialek, J. (2004). Viscosity models in simulation of the filling stage of the injection molding process. *Journal of Materials Processing Technology*, 157-158(SPEC. ISS.):183–187.
- [86] Koulocheris, D., Stathis, A., Costopoulos, T., and Tsantiotis, D. (2014). Experimental study of the impact of grease particle contaminants on wear and fatigue life of ball bearings. *Engineering Failure Analysis*, 39:164–180.
- [87] Krog, Ø. E. and Elster, A. C. (2012). Fast GPU-Based fluid simulations using SPH. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7134 LNCS(PART 2):98–109.
- [88] Kumar, K. B. S. S., Reddy, C. J., and Ramesh, K. V. (2020). Experimental Study on Drop Point of Grease Samples for Various Lubricating Applications. *International Journal of Engineering Research & Technology*, 8(16):138–139.
- [89] Larson, R. G. and Wei, Y. (2019). A review of thixotropy and its rheological modeling. *Journal of Rheology*, 63(3):477–501.
- [90] Lewis, W. J. (1990). The design of high speed grease lubricated spindles for machine tool production equipment. *SAE Technical Papers*, 99(1990):1061–1070.
- [91] Li, B., Li, N., Yang, M., Liu, G., Zheng, L., Yang, C., Qiao, W., and Jiao, T. (2022). Molecular dynamics simulation of yield thixotropy of crude oil systems. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 643(March):1–9.

- [92] Li, J. X., Höglund, E., Westerberg, L. G., Green, T. M., Lundström, T. S., Lugt, P. M., and Baart, P. (2012). μ pIV measurement of grease velocity profiles in channels with two different types of flow restrictions. *Tribology International*, 54:94–99.
- [93] Li, S., Hao, W., and Liu, W. K. (2000). Mesh-free simulations of shear banding in large deformation. *International Journal of Solids and Structures*, 37(48):7185–7206.
- [94] Li, Y., Li, Y., and Feng, Z. (2023). A coupled particle model with particle shifting technology for simulating transient viscoelastic fluid flow with free surface. *Journal of Computational Physics*, 488:112213.
- [95] Lind, S. J., Xu, R., Stansby, P. K., and Rogers, B. D. (2012). Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4):1499–1523.
- [96] Littlewood, D., Hillman, M., Yreux, E., Bishop, J., Beckwith, F., and Chen, J.-s. (2015). Implementation and Verification of RKPM in the Sierra/SolidMechanics Analysis Code. In *Volume 9: Mechanics of Solids, Structures and Fluids*, pages 1–10. American Society of Mechanical Engineers.
- [97] Liu, G. and Liu, M. (2003). Smoothed Particle Hydrodynamics; A Mesh-free Particle Method. *World Scientific, Singapore*.
- [98] Liu, J., Yan, Z., and Shao, Y. (2018). An investigation for the friction torque of a needle roller bearing with the roundness error. *Mechanism and Machine Theory*, 121:259–272.
- [99] Liu, M. B., Xie, W. P., and Liu, G. R. (2005). Modeling incompressible flows using a finite particle method. *Applied Mathematical Modelling*, 29(12):1252–1270.
- [100] Liu, W. K., Jun, S., and Zhang, Y. F. (1995). Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106.
- [101] Liu, Z. and Zhang, L. (2020). A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings. *Measurement: Journal of the International Measurement Confederation*, 149:107002.

- [102] Loiseau, J.-C., Bucci, M. A., Cherubini, S., and Robinet, J.-C. (2019). *Computational Modelling of Bifurcations and Instabilities in Fluid Dynamics*, volume 50.
- [103] Lugt, P. M. (2009). A review on grease lubrication in rolling bearings. *Tribology Transactions*, 52(4):470–480.
- [104] Lugt, P. M. (2016). Modern advancements in lubricating grease technology. *Tribology International*, 97:467–477.
- [105] Lugt, P. M. and Morales-Espejel, G. E. (2011). A review of elasto-hydrodynamic lubrication theory. *Tribology Transactions*, 54(3):470–496.
- [106] Lyu, H. G., Sun, P. N., Huang, X. T., Zhong, S. Y., Peng, Y. X., Jiang, T., and Ji, C. N. (2022). A Review of SPH Techniques for Hydrodynamic Simulations of Ocean Energy Devices. *Energies*, 15(2):1–48.
- [107] Maccioni, L., Chernoray, V. G., Bohnert, C., and Concli, F. (2022). Particle Image Velocimetry measurements inside a tapered roller bearing with an outer ring made of sapphire: Design and operation of an innovative test rig. *Tribology International*, 165(September 2021):107313.
- [108] Mang, T. (2014). Encyclopedia of Lubricants and Lubrication. *Encyclopedia of Lubricants and Lubrication*.
- [109] Mao, Z. and Liu, G. R. (2018). A Lagrangian gradient smoothing method for solid-flow problems using simplicial mesh. *International Journal for Numerical Methods in Engineering*, 113(5):858–890.
- [110] Marques, A. C. H. and Doricio, J. L. (2006). Numerical investigation of the flow in a two-dimensional cavity: Meshless, finite volumes and finite differences methods. *Latin American Journal of Solids and Structures*, 3(3):301–324.
- [111] Mastrone, M. N. and Concli, F. (2021). CFD simulation of grease lubrication: Analysis of the power losses and lubricant flows inside a back-to-back test rig gearbox. *Journal of Non-Newtonian Fluid Mechanics*, 297(May):104652.
- [112] Mastrone, M. N. and Concli, F. (2022). A Multi Domain Modeling Approach for the CFD Simulation of Multi-Stage Gearboxes. *Energies*, 15(3).

- [113] Menéndez Blanco, A. and Fernández Oro, J. M. (2012). Unsteady numerical simulation of an air-operated piston pump for lubricating greases using dynamic meshes. *Computers and Fluids*, 57:138–150.
- [114] Monaghan, J. J. (1985). Particle Method for Hydrodynamics. *Computer Physics Reports*, 3(2):71–124.
- [115] Monaghan, J. J. (2000). SPH without a Tensile Instability. *Journal of Computational Physics*, 159(2):290–311.
- [116] Monaghan, J. J. (2011). Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*, 44:323–346.
- [117] Moon, M. (2007). How clean are your lubricants? *Trends in Food Science and Technology*, 18(SUPPL. 1):74–79.
- [118] Moore, J. D., Cui, S. T., Cochran, H. D., and Cummings, P. T. (2000). A molecular dynamics study of a short-chain polyethylene melt. I. Steady-state shear. *Journal of Non-Newtonian Fluid Mechanics*, 93(1):83–99.
- [119] Morris, J. P. (1996). Analysis of smoothed particle hydrodynamics with applications. *Monash University*, (July):177.
- [120] Mukai, N., Matsui, E., and Chang, Y. (2019). Investigation on viscoelastic fluid behavior by modifying deviatoric stress tensor. *SIMULTECH 2019 - Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, (Simultech):216–222.
- [121] Nagasawa, K., Suzuki, T., Seto, R., Okada, M., and Yue, Y. (2019). Mixing sauces: A viscosity blending model for shear thinning fluids. *ACM Transactions on Graphics*, 38(4).
- [122] Naumov, M., Arsaev, M., Castonguay, P., Cohen, J., Demouth, J., Eaton, J., Layton, S., Markovskiy, N., Regulý, I., Sakharnykh, N., Sellappan, V., and Strzodka, R. (2015). AMGX: A library for GPU accelerated algebraic multigrid and preconditioned iterative methods. *SIAM Journal on Scientific Computing*, 37(5):S602–S626.
- [123] NEGISHI, H., AMAKAWA, H., MANIWA, K., OBARA, S., HAYAMA, M., and DONG, D. (2019). Numerical analysis of dam breaking behavior of Bingham pseudoplastic fluid by using the explicit MPS methodExplicit MPS. *Transactions of the JSME (in Japanese)*, 85(875):19–00086–19–00086.

- [124] Negishi, H., Fujiwara, H., Maniwa, K., Obara, S., and Shibata, K. (2018). MPS . , pages 1–6.
- [125] Neofytou, P. (2005). A 3rd order upwind finite volume method for generalised Newtonian fluid flows. *Advances in Engineering Software*, 36(10):664–680.
- [126] Nguyen, H., Wang, J., and Bazilevs, Y. (2024). A smooth Crack-Band Model for anisotropic materials: Continuum theory and computations with the RKPM meshfree method. *International Journal of Solids and Structures*, 288(December 2023):112618.
- [127] Nickolls, J. and Dally, W. J. (2010). The GPU Computing Era. *Ieee Micro*, 30(2):56–70.
- [128] Nie, X., Chen, L., and Xiang, T. (2015). Real-time incompressible fluid simulation on the GPU. *International Journal of Computer Games Technology*, 2015.
- [129] Noda, T., Shibasaki, K., Miyata, S., and Taniguchi, M. (2016). X . , 61:275–284.
- [130] Noda, T., Shibasaki, K., Miyata, S., and Taniguchi, M. (2020). X-ray CT imaging of grease behavior in ball bearing and numerical validation of multi-phase flows simulation. *Tribology Online*, 15(1):36–44.
- [131] Norton, T. and Sun, D. W. (2006). Computational fluid dynamics (CFD) - an effective and efficient design and analysis tool for the food industry: A review. *Trends in Food Science and Technology*, 17(11):600–620.
- [132] O’Connor, J., Domínguez, J. M., Rogers, B. D., Lind, S. J., and Stansby, P. K. (2022). Eulerian incompressible smoothed particle hydrodynamics on multiple GPUs. *Computer Physics Communications*, 273:108263.
- [133] Paggi, M., Amicarelli, A., and Lenarda, P. (2019). SPH modelling of hydrodynamic lubrication along rough surfaces. *Lubricants*, 7(12).
- [134] Paggi, M., Amicarelli, A., and Lenarda, P. (2021). SPH modelling of hydrodynamic lubrication: laminar fluid flowstructure interaction with no-slip conditions for slider bearings. *Computational Particle Mechanics*, 8(4):665–679.

- [135] Peng, Y. X., Zhang, A. M., and Ming, F. R. (2021). Numerical simulation of structural damage subjected to the near-field underwater explosion based on SPH and RKPM. *Ocean Engineering*, 222(October 2020):108576.
- [136] Pinchin, R. and Boer, M. D. (2003). Production and use of food-grade lubricants. *Trends in Food Science and Technology*, 14(4):157–162.
- [137] Posey, S., See, S., and Wang, M. (2015). GPU PROGRESS AND DIRECTIONS IN APPLIED CFD. *11th International Conference on CFD in the Minerals and Process Industries*, (December):1–6.
- [138] Qiao, B., Reiche, O., Hannig, F., and Teich, J. (2019). From Loop Fusion to Kernel Fusion: A Domain-Specific Approach to Locality Optimization. *CGO 2019 - Proceedings of the 2019 IEEE/ACM International Symposium on Code Generation and Optimization*, pages 242–253.
- [139] Rafiee, A., Manzari, M. T., and Hosseini, M. (2007). An incompressible SPH method for simulation of unsteady viscoelastic free-surface flows. *International Journal of Non-Linear Mechanics*, 42(10):1210–1223.
- [140] Repsol (2023). https://www.repsol.com/imagenes/global/en/greases_catalog_tcm14-37191.pdf Accessed on 30th June 2023.
- [141] Roman, C., Valencia, C., and Franco, J. M. (2016). AFM and SEM Assessment of Lubricating Grease Microstructures: Influence of Sample Preparation Protocol, Frictional Working Conditions and Composition. *Tribology Letters*, 63(2).
- [142] RUPP, K., TILLET, P., RUDOLF, F., WEINBUB, J., MORHAMMER, A., GRASSER, T., JUNGEL, A., SELBERHERR, S., and Abstract. (2016). Viennacl - linear algebra library for multi- and many-core architectures. *SIAM Journal on Scientific Computing*, 38(5).
- [143] Ružbarský, J., Majerník, R., Hrabě, P., and Valášek, P. (2016). Dynamics of treatment device for die casting of metals. *Key Engineering Materials*, 669(2):327–334.
- [144] Salomonsson, L., Stang, G., and Zhmud, B. (2007). Oil/thickener interactions and rheology of lubricating greases. *Tribology Transactions*, 50(3):302–309.
- [145] Samman, N. and Lau, S. N. (2002). Grease-based open gear lubricants: Multi-service products - Development and evaluation. *CIM Bulletin*, 95(1059):133–139.

- [146] Sarkar, C., Westerberg, L. G., Höglund, E., and Lundström, T. S. (2018). Numerical Simulations of Lubricating Grease Flow in a Rectangular Channel with and without Restrictions. *Tribology Transactions*, 61(1):144–156.
- [147] Sathwik Chatra, K. R., Osara, J. A., and Lugt, P. M. (2022). Impact of grease churning on grease leakage, oil bleeding and grease rheology. *Tribology International*, 176(July):107926.
- [148] Shah, R., Tung, S., Chen, R., and Miller, R. (2021). Grease performance requirements and future perspectives for electric and hybrid vehicle applications. *Lubricants*, 9(4):1–16.
- [149] Shao, S. and Lo, E. Y. (2003). Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources*, 26(7):787–800.
- [150] Shih, W. Y., Shih, W. H., and Aksay, I. A. (1999). Elastic and yield behavior of strongly flocculated colloids. *Journal of the American Ceramic Society*, 82(3):616–624.
- [151] Shodja, H. M., Khezri, M., Hashemian, A., and Behzadan, A. (2010). RKPM with augmented corrected collocation method for treatment of material discontinuities. *CMES - Computer Modeling in Engineering and Sciences*, 62(2):171–204.
- [152] Siewerin, B. J., Winkler, K. J., Tobie, T., and Stahl, K. (2021). Influences on the lubricant supply of grease lubricated gears. *IOP Conference Series: Materials Science and Engineering*, 1097(1):012015.
- [153] Sikora, M., Adamczyk, G., Krystyjan, M., Dobosz, A., Tomasik, P., Berski, W., Lukasiewicz, M., and Izak, P. (2015). Thixotropic properties of normal potato starch depending on the degree of the granules pasting. *Carbohydrate Polymers*, 121:254–264.
- [154] SKF (2023). https://www.skf.com/binaries/pub12/Images/0901d196802103bc-13238EN_GreaseSelectionChart_tcm_12-99598.pdf Accessed on 30th June 2023.
- [155] Solenthaler, B. and Pajarola, R. (2008). Density contrast SPH interfaces. *Computer Animation 2008 - ACM SIGGRAPH / Eurographics Symposium, SCA 2008 - Proceedings*, pages 211–218.

- [156] Solenthaler, B. and Pajarola, R. (2009). Predictive-corrective incompressible SPH. *ACM Transactions on Graphics*, 28(3).
- [157] standards, A. (2016). Standard Test Methods for Cone Penetration of Lubricating Grease Using One-Quarter and One-Half Scale Cone Equipment. *Annual Book of ASTM Standards*, 84(92):1–9.
- [158] Sterpu, A. E., Prodan, G., Teodorescu, N., Prodea, I. M., Dumitru, A. I., and Koncsag, C. I. (2016). Lubricating greases from olive oil, corn oil and palm oil. *Revista de Chimie*, 67(8):1575–1582.
- [159] Storey, B. T. and Merrill, E. W. (1958). The rheology of aqueous solutions of amylose and amylopectin with reference to molecular configuration and intermolecular association. *Journal of Polymer Science*, 33(126):361–375.
- [160] Suetsugu, Y., Sekiguchi, H., Nakanishi, Y., Fujinami, Y., and Ohno, T. (2013). Basic study of grease rheology and correlation with grease properties. *Tribology Online*, 8(1):83–89.
- [161] Swegle, J. W., Hicks, D. L., and Attaway, S. W. (1995). Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116(1):123–134.
- [162] Tandon, N., Ramakrishna, K. M., and Yadava, G. S. (2007). Condition monitoring of electric motor ball bearings for the detection of grease contaminants. *Tribology International*, 40(1):29–36.
- [163] Tang, Y., Chen, S., and Jiang, Q. (2020). A conservative SPH scheme using exact projection with semi-analytical boundary method for free-surface flows. *Applied Mathematical Modelling*, 82:607–635.
- [164] Total (2023). https://totalenergies.jp/sites/g/files/wompnd1361/f/atoms/files/total-grease-brochure-fa-1003_hd_file.pdf Accessed on 30th June 2023.
- [165] Ulrich, C., Leonardi, M., and Rung, T. (2013). Multi-physics SPH simulation of complex marine-engineering hydrodynamic problems. *Ocean Engineering*, 64:109–121.
- [166] Violeau, D., Buvat, C., Abed-Meraim, K., and de Nanteuil, E. (2007). Numerical modelling of boom and oil spill with SPH. *Coastal Engineering*, 54(12):895–913.

- [167] Violeau, D. and Rogers, B. D. (2016). Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future. *Journal of Hydraulic Research*, 54(1):1–26.
- [168] Wang, B., Wang, Z., Sun, C., and Wu, Y. (2021). Numerical investigation of the heat-fluid characteristic inside high-speed angular contact ball bearing lubricated with grease. *International Journal of Engineering, Transactions B: Applications*, 34(5):1313–1320.
- [169] Wang, G., Lin, Y. S., and Yi, W. (2010). Kernel fusion: An effective method for better power efficiency on multithreaded GPU. *Proceedings - 2010 IEEE/ACM International Conference on Green Computing and Communications, GreenCom 2010, 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing, CPSCoM 2010*, pages 344–350.
- [170] Wang, H., Huang, H., Yu, S., and Gu, W. (2020a). The effect of surface morphology of tapered rolling bearings in high-speed train on grease lubrication. *Lubricants*, 8(7):1–14.
- [171] Wang, X. Y., Zhou, C. G., and Ou, Y. (2019). Experimental analysis of the wear coefficient for the rolling linear guide. *Advances in Mechanical Engineering*, 11(1):1–7.
- [172] Wang, Y., Cao, J., Tong, Q., An, G., Liu, R., Zhang, Y., and Yan, H. (2020b). Study on the thermal performance and temperature distribution of ball bearings in the traction motor of a high-speed EMU. *Applied Sciences (Switzerland)*, 10(12).
- [173] Wang, Z. B., Chen, R., Wang, H., Liao, Q., Zhu, X., and Li, S. Z. (2016). An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modelling*, 40(23-24):9625–9655.
- [174] Wei, X., Xu, C., Jia, Z., and Wang, X. (2021). Molecular dynamics simulation of thermal degradation of silicone grease using reactive force field. *Journal of Applied Polymer Science*, 138(15):1–11.
- [175] Weiler, M., Koschier, D., Brand, M., and Bender, J. (2018). A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum*, 37(2):145–155.
- [176] Westerberg, L. G., Farré-Lladós, J., Sarkar, C., and Casals-Terré, J. (2018). Contaminant particle motion in lubricating grease flow: A computational fluid dynamics approach. *Lubricants*, 6(1).

- [177] Westerberg, L. G., Sarkar, C., Lladós, J. F., Lundström, T. S., and Höglund, E. (2017). Lubricating Grease Flow in a Double Restriction Seal Geometry: A Computational Fluid Dynamics Approach. *Tribology Letters*, 65(3).
- [178] Whitehead, N. and Fit-Florea, A. (2011). Precision & Performance: Floating Point and IEEE 754 Compliance for NVIDIA GPUs. *IEE Conference Publication*.
- [179] Wieth, L., Lieber, C., Kurz, W., Braun, S., Koch, R., and Bauer, H.-J. (2015). Numerical Modeling of an Aero-Engine Bearing Chamber Using the Meshless Smoothed Particle Hydrodynamics Method. In *Volume 2B: Turbomachinery*, pages 1–13. American Society of Mechanical Engineers.
- [180] (Winston) Cheng, W. and Tu, W. (2002). Tribological Study of Grease Lubricated Plastic Wormgear for Automotive Applications. *Tribology Transactions*, 45(4):563–567.
- [181] Wu, L. and Tan, Q. (2016). A study of cooling system in a grease-lubricated precision spindle. *Advances in Mechanical Engineering*, 8(8):1–15.
- [182] Xia, X. and Liang, Q. (2016). A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. *Environmental Modelling and Software*, 75:28–43.
- [183] Xu, R., Stansby, P., and Laurence, D. (2009). Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics*, 228(18):6703–6725.
- [184] Xu, X., Ouyang, J., Yang, B., and Liu, Z. (2013). SPH simulations of three-dimensional non-Newtonian free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 256:101–116.
- [185] Yan, J., Li, S., Kan, X., Zhang, A. M., and Lai, X. (2020). Higher-order nonlocal theory of Updated Lagrangian Particle Hydrodynamics (ULPH) and simulations of multiphase flows. *Computer Methods in Applied Mechanics and Engineering*, 368:113176.
- [186] Ye, T., Pan, D., Huang, C., and Liu, M. (2019). Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications. *Physics of Fluids*, 31(1):1–41.

- [187] Yonggang, M. and Jie, Z. (1998). Rheological model for lithium lubricating grease. *Tribology International*, 31(10):619–625.
- [188] Yoon, C., Heister, S. D., and Campanella, O. H. (2014). Modeling gelled fluid flow with thixotropy and rheological hysteresis effects. *Fuel*, 128:467–475.
- [189] Yue, Y., Smith, B., Batty, C., Zheng, C., and Grinspun, E. (2015). Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics*, 34(5).
- [190] Zago, V., Bilotta, G., Hérault, A., Dalrymple, R. A., Fortuna, L., Cappello, A., Ganci, G., and Del Negro, C. (2018). Semi-implicit 3D SPH on GPU for lava flows. *Journal of Computational Physics*, 375:854–870.
- [191] Zakani, B. and Grecov, D. (2018). Yield Stress Analysis of a Fumed Silica Lubricating Grease. *Tribology Transactions*, 61(6):1131–1140.
- [192] Zhang, E., Li, W., Zhao, G., Wang, Z., and Wang, X. (2021a). A Study on Microstructure, Friction and Rheology of Four Lithium Greases Formulated with Four Different Base Oils. *Tribology Letters*, 69(3):1–9.
- [193] Zhang, G. M. and Batra, R. C. (2004). Modified smoothed particle hydrodynamics method and its application to transient problems. *Computational Mechanics*, 34(2):137–146.
- [194] Zhang, G. M. and Batra, R. C. (2009). Symmetric smoothed particle hydrodynamics (SSPH) method and its application to elastic problems. *Computational Mechanics*, 43(3):321–340.
- [195] Zhang, K., Peng, X., Zhang, Y., Zhou, H., and Ma, M. (2019). Numerical thermal analysis of grease-lubrication in limited line contacts considering asperity contact. *Tribology International*, 134(December 2018):372–384.
- [196] Zhang, Q., Mugele, F., van den Ende, D., and Lugt, P. M. (2021b). A Model Configuration For Studying Stationary Grease Bleed In Rolling Bearings. *Tribology Transactions*, 64(6):1127–1137.
- [197] Zhou, C., Ren, G., Fan, X., and Lv, Y. (2022). Probing the effect of thickener microstructure on rheological and tribological properties of grease. *Journal of Industrial and Engineering Chemistry*, 111:51–63.
- [198] Zhu, B., Lee, M., Quigley, E., and Fedkiw, R. (2015). Codimensional non-Newtonian fluids. *ACM Transactions on Graphics*, 34(4).