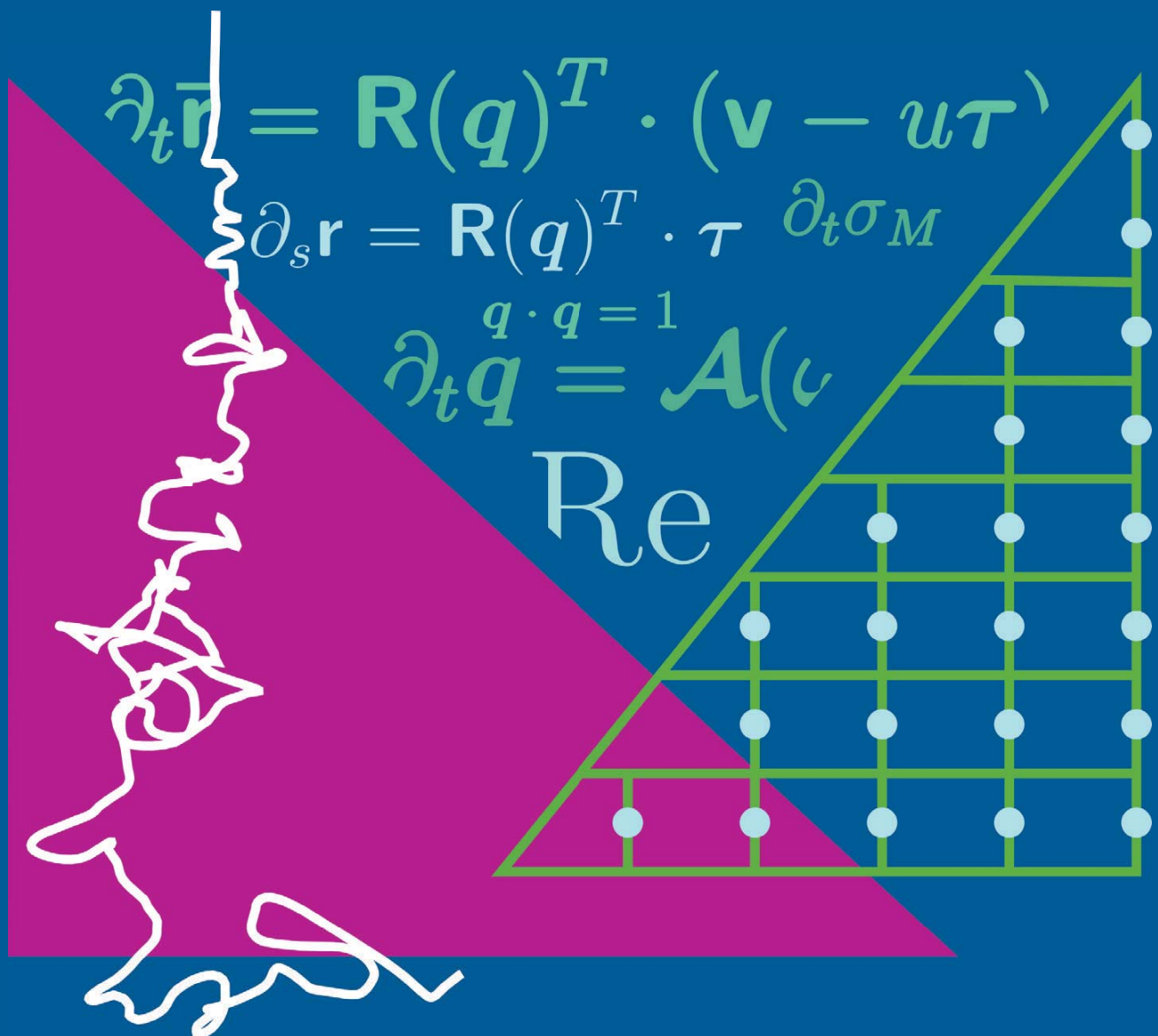


Stefan Schießl

Jet and Fiber Dynamics with high Elongations: Models, Numerical Strategies and Applications



Fraunhofer-Institut für
Techno- und Wirtschaftsmathematik ITWM

Jet and Fiber Dynamics with high Elongations: Models, Numerical Strategies and Applications

Stefan Schießl

FRAUNHOFER VERLAG

Kontakt:

Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM
Fraunhofer-Platz 1
67663 Kaiserslautern
Telefon +49 631/31600-0
Fax +49 631/31600-1099
E-Mail info@itwm.fraunhofer.de
URL www.itwm.fraunhofer.de

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.
ISBN (Print): 978-3-8396-1241-5

D 29

Zugl.: Erlangen-Nürnberg, Univ., Diss., 2017

Druck: Mediendienstleistungen des
Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart

Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© by **FRAUNHOFER VERLAG**, 2017

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 80 04 69, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-25 00
Telefax 07 11 9 70-25 08
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften. Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

Jet and fiber dynamics with high elongations: Models, numerical strategies and applications

Jet- und Fadendynamik mit hohen Dehnungen: Modelle,
numerische Strategien und Anwendungen

Der Naturwissenschaftlichen Fakultät
der Friedrich–Alexander–Universität
Erlangen–Nürnberg
zur
Erlangung des Doktorgrades Dr. rer. nat.

vorgelegt von
Stefan Schießl

aus Erlangen

Als Dissertation genehmigt von der Naturwissenschaftlichen Fakultät
der Friedrich–Alexander–Universität Erlangen–Nürnberg

Tag der mündlichen Prüfung: 18. Juli 2017

Vorsitzender des Promotionsorgans: Prof. Dr. Georg Kreimer

Gutachter: Prof. Dr. Nicole Marheineke
Prof. Dr. Andreas Meister

Zusammenfassung

Diese Arbeit untersucht Modelle und numerische Strategien für das Verhalten von schmalen Objekten, die in Spinnprozessen durch große äußere Kräfte verformt werden. Wir beschäftigen uns mit einem inkompressiblen, hochviskosen und dreidimensionalen *Jet* (engl. Strahl), der durch ein eindimensionales Modell, bestehend aus partiellen differential-algebraischen Gleichungen, beschrieben wird. Dieses Modell wird wiederum mit Hilfe der Mittellinie des Jets und der dazugehörigen, gerichteten Querschnitte ausgedrückt und besteht aus der Kinematik und Dynamik des Jets. Es wird mit einem Geometrie- und Materialmodell vervollständigt. Für die Kinematik und die Parametrisierung des Jets sind unterschiedliche Formulierungen möglich, welche zu unterschiedlichen Varianten des Modells führt. Um den Wechsel zwischen den Formulierungen und des Material- sowie Geometriemodells zu vereinfachen, benutzen wir eine vielseitig verwendbare Modellbeschreibung mit einem passenden numerischen *Framework* (engl. Programmiergerüst).

Unser Ziel ist es, eine robuste Basis für die Simulation von Spinnprozessen zur Verfügung zu stellen. Insbesondere solche Prozesse, die eine zeitabhängige Betrachtung erfordern und dadurch jede sinnvolle Vereinfachung des Modells verhindern. Das generell als zeitabhängig angenommene räumliche Gebiet muss passend in die Diskretisierung übertragen werden. Zu diesem Zweck benutzen wir eine Finite Volumen Methode für beliebige Raum-Zeit-Gebiete und führen eine neue Formulierung der Jet-Kinematik ein, die dem Modell zugrunde liegenden differential-algebraischen Eigenschaften berücksichtigt. Das Modell und die Diskretisierung werden durch numerische Konvergenzaussagen (im Raum, in der Zeit und kombiniert) validiert.

Als Beispiele für die industrielle Anwendung betrachten wir Produktionsprozesse von Dämmstoffen mit Rotationsspinnverfahren und von Vliesstoffen mit *Melt-Blowing*-Verfahren (engl. Schmelzblasverfahren). In beiden Prozessen treten große Dehnungen auf, welche sich in stark variierenden Komponenten der Lösung widerspiegelt. Diese können wiederum numerische Schwierigkeiten verursachen. Hierfür untersuchen wir Möglichkeiten zur adaptiven Gitterverfeinerung, insbesondere der r -Verfeinerung (*Moving Mesh*, engl. bewegtes Gitter), die direkt mit dem Jet-Modell in einer allgemeinen Parametrisierung verwendet werden kann. Ohne die Ergebnisse solcher Moving-Mesh-Strategien vorwegzunehmen, stellen wir zusätzlich Anpassungen der Randbedingungen des Modells vor, die die Simulationen der beiden Produktionsprozesse in den industriell relevanten Parameterbereichen ermöglichen.

Abstract

This work investigates models and numerical strategies for the behavior of a slender object deformed by large external forces in spinning processes. Our main consideration is an incompressible, highly viscous and three-dimensional jet that is described by a one-dimensional model of partial differential-algebraic equations with the help of the jet's centerline and oriented cross-sections. The model consists of the kinematics and dynamics for the jet and is completed with geometric and material models. Considering the jet's kinematics and parameterization, different formulations are possible which lead to different model variants. A versatile model description and numerical framework are provided that facilitate the exchange of the kinematics' formulation, geometric and material model.

We furthermore aim to provide a robust basis for the simulation of production processes that require transient treatment and prevent any meaningful simplification of the model equations. The spatial domain is generally considered time-dependent and requires proper handling by the discrete scheme. For that purpose, a Finite Volume method for an arbitrary space-time domain is proposed in combination with a new kinematics formulation that takes the underlying differential-algebraic character of the model into account. The performance of the model and discrete scheme is validated through numerical convergence order results (in space, time and combined).

As examples for industrial applications, we consider production processes of insulation with the rotational spinning process and nonwoven materials with the melt-blowing process. Both exhibit large elongations that manifest in strongly varying solution components, possibly causing numerical difficulties. We explore the possibilities of adaptive mesh refinement, in particular we use r -refinement (moving mesh) that is easily included through a general parameterization of the jet model. Without anticipating the performance of such moving mesh strategies, we introduce alterations of the model's boundary conditions that enable us to conduct simulation of both production processes in industrially relevant parameter ranges.

I want to deeply express my gratitude to Prof. Dr. Nicole Marheineke and Dr. Raimund Wegener, who suggested this interesting research topic and gave me the opportunity to develop under their wings while still giving me freedom for the course of investigations. Special thanks are owed to them for all the helpful discussions over the years and their constructive, creative and critical input. They always dedicated their time for arising questions and problems, which is not taken for granted and was essential for my development. I could draw from their vast pool of experience of over ten years in the area of jet and fiber dynamics through their valuable suggestions and advices. I could collect many new insights, which I am very grateful for. I also want to thank Dr. Walter Arne for his support and knowledge, especially his experience with numerical details. To all three I want to say thank you for the successful cooperation and the resulting joint research papers. Furthermore, I want to thank Prof. Dr. Nicole Marheineke, Dr. Raimund Wegener, Nadine Stahl, Björn Liljigren-Sailer and Alexander Vibe for their helpful remarks during the creation of this dissertation.

I would like to thank all the colleagues of the department of Applied Mathematics 1 of the Friedrich-Alexander-University Erlangen-Nürnberg for their motivation and support, and very pleasant working atmosphere. In particular to Prof. Dr. Knabner as the head of the department, Dr. Alexander Prectel and Balthasar Reuter for their network, software and hardware administration, and Astrid Bigott and Cornelia Weber for their organizational help. During the course of this work I had many visits to the department of Transport Processes of the Fraunhofer ITWM in Kaiserslautern, here I want to extend my thanks to Dr. Raimund Wegener and Dr. Dietmar Hietel, who enabled those visits. I am grateful for the warm and continuous welcome and the opportunity to conduct my research there, in particular I want to mention Dr. Walter Arne and Dr. Simone Gramsch. I could greatly benefit from the direct ties to industrial research partners, certainly also through providing experimental data of the considered applications.

For the finances I want to acknowledge the Fraunhofer Institute for Industrial Mathematics (ITWM) and the Deutsche Forschungsgemeinschaft (DFG) through the project 'Simulation hochdynamischer Spinnprozesse unter Turbulenzeinfluss' (MA 4526/2-1).

At last but not least, I would particularly like to thank my partner and soon-to-be wife Sara and my whole family for their unconditional support and belief in me.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 9 |
| 2 | Modeling the behavior of jets and fibers | 13 |
| 2.1 | Theory of rods | 13 |
| 2.1.1 | Kinematics and dynamics | 14 |
| 2.1.2 | Geometric model | 17 |
| 2.1.3 | Material laws | 18 |
| 2.1.4 | Energy balance | 21 |
| 2.1.5 | External loads | 21 |
| 2.2 | Formulations for the viscous jet model | 21 |
| 2.2.1 | General parameterization | 22 |
| 2.2.2 | Non-dimensionalization | 27 |
| 2.2.3 | Parameterization of the rotational group | 28 |
| 2.2.4 | Index analysis and a stabilized reduced index formulation | 30 |
| 2.2.5 | Viscous jet model | 36 |
| 2.3 | Boundaries and space-time domains for set-ups | 37 |
| 2.3.1 | Position and tension-free boundary | 41 |
| 2.3.2 | Both tension-free boundaries | 47 |
| 2.3.3 | Both position boundaries | 48 |
| 3 | Adaptive meshes: r-refinement | 51 |
| 3.1 | Three parameterization layers | 52 |
| 3.2 | Adaption strategy | 54 |
| 3.3 | Desired re-parameterization | 61 |
| 4 | Numerical scheme | 69 |
| 4.1 | Finite Volume method | 69 |
| 4.1.1 | Scheme and truncation error | 70 |
| 4.1.2 | Initial and boundary conditions | 78 |
| 4.1.3 | Algorithm | 82 |
| 4.1.4 | Solution reconstruction | 83 |
| 4.2 | Application to the jet model | 84 |
| 4.3 | Application to r -refinement strategies | 86 |
| 5 | Convergence and performance | 89 |
| 5.1 | Academic scenario: Viscous cantilever | 89 |
| 5.2 | Convergence order | 91 |
| 5.3 | Avoiding a singular Jacobian | 95 |

| | | |
|---------------------|--|------------|
| 5.4 | Adaptivity | 96 |
| 5.4.1 | Cost of general parameterization | 97 |
| 5.4.2 | Suitable re-parameterization | 99 |
| 6 | Applications | 105 |
| 6.1 | Rotational spinning process | 105 |
| 6.1.1 | Model alteration, external loads and parameters | 107 |
| 6.1.2 | Numerical method and investigation | 108 |
| 6.1.3 | Parameter study | 114 |
| 6.1.4 | Industrial application | 115 |
| 6.2 | Melt-blowing process | 118 |
| 6.2.1 | Air drag model | 122 |
| 6.2.2 | Turbulent reconstruction | 123 |
| 6.2.3 | Numerical method and investigation | 124 |
| 6.2.4 | Industrial example | 126 |
| 7 | Conclusion | 135 |
| Appendices | | |
| A | Other material laws | 139 |
| A.1 | Elastic | 139 |
| A.2 | Viscoelastic | 140 |
| A.3 | Thermal simulations | 140 |
| B | Modeling and reduction of dimensions | 145 |
| B.1 | GGL projection correction for the kinematics | 145 |
| B.2 | Two-dimensional jet model | 145 |
| B.3 | Uniaxial jet model | 146 |
| C | Adaptivity | 149 |
| C.1 | Analytical example | 150 |
| C.2 | Burger's equation | 157 |
| C.3 | Growing Eulerian parameterization with Lagrange tracking | 160 |
| C.4 | Rotational spinning process | 161 |
| D | Numerical scheme and implementation | 163 |
| D.1 | Details to derivation of the Finite Volume method | 163 |
| D.2 | Details on the application of the jet model | 169 |
| Notations | | 173 |
| Bibliography | | 179 |

1 Introduction

The spinning of jets and fibers provides the core of various industrial manufacturing processes. Examples are the production of nonwoven fabrics, insulating and light-weight materials, and technical, specialized textures. There exist broad application realms, ranging from highly absorbent fabrics (diapers) to high-tech products like microporous synthetics (battery separators). The wording jet and fiber are distinguishably used here. Both mean a slender object, whereas a jet is fluid-like, governed by a viscous material behavior, and a fiber is solid-like, governed by elastic material behavior. The exemplary area of applications in this work is textile manufacturing, in particular highly dynamic spinning processes for nonwovens, e.g., the rotational spinning, spunbound and melt-blowing processes. The common core is that some hot, viscous material (for example a molten polymer or glass) is extruded through small nozzles, forming jets that are influenced by viscous strains, surface tension and strong external forces (e.g. turbulent aerodynamic forces). The polymer is vigorously deformed and stretched and solidifies during the cooldown and is subsequently collected by a conveyor belt for further processing. The quality of this lay-down web and the resulting material depends essentially on the dynamics of the jets.

Of particular interest for industrial applications is the behavior close to the nozzle and achievable thickness of the jet. The thickness of the jet can become several orders of magnitude smaller than the initial thickness at the nozzle, e.g., up to order 10^6 in melt-blowing. Thinking of a realization of such processes in order to estimate material properties requires a robust simulation. This task is especially challenging as the jet is growing over time (and therewith the spatial domain to be considered) and the large stretching of the jet manifests in unknowns that vary greatly in their scale and possibly large gradients. Hence, to enable the simulation of highly dynamic spinning processes an appropriate jet model combined with a robust numerical approach for the growing domain and appearing computational challenges has to be provided.

The following literature overview makes no attempt to offer a complete historical portrayal. Instead, we try to sketch a cohesive overview from our point of view. The analysis on spinning processes dates back as early as to the 1960s to the works of Kase and Matsuo [63], and Pearson and Matovich [81, 76] who investigated steady, isothermal and viscous dominated Stokes' flow on a reduced slender, straight (uniaxial) and axisymmetric jet. Entov and Yarin [39] derived an instationary, quasi-one-dimensional model that considers angular momentum to investigate viscous jets moving in air. In [35, 36] a perturbation expansion of the full Stokes' flow problem is investigated and leading-order equations for the extensional flow from the three-dimensional Navier-Stokes equations are derived, which is the first systematic derivation of an one-dimensional model, that has subsequently been extended in [55] to non-axisymmetric models and more complex geometries, also accounting for surface tension [31]. The restriction to nearly straight bodies was alleviated to curved and coiled ones by Descent et al. [33] and [85] and their following studies. An

asymptotic model without any restriction on the jet motion and shape is developed by Panda, Marheineke and Wegener [80] and extended in [74], which is also the basis of this work.

The slender-body theory can be separated in two classes of one-dimensional models. On the one hand there are the string models, e.g. [63, 81, 76, 31, 35, 36, 74] and on the other hand the rod models, e.g. [39, 111, 85]. For the derivation of those models the slenderness ratio is of key importance. Said ratio is the relation between length and diameter of the jet, which is of course very small for long and thin jets. The string models are asymptotic systems of leading order that result from a strict systematic derivation using expansions in the slenderness ratio from three-dimensional free boundary value problems of Newtonian fluid flows. They consist of balance laws for mass and linear momentum. The rod models are considered more sophisticated and also include an angular momentum balance law. The derivation of the rod models builds on cross-sectional averaging of the underlying three-dimensional balance equations under certain assumptions. The constitutive elements of the special Cosserat theory of rods [3] are a curve (center-line, not necessarily center of mass) and director triad specifying the position and the orientation of the cross-sections, the model is completed with heuristically motivated material and geometrical laws. The Cosserat rod model contains the slenderness ratio explicitly in the angular momentum balance, but it is not an asymptotic system of leading order. Nevertheless it reduces to a string model as the slenderness ratio goes to zero, thus it can be considered as a regularized string. The string model has restrictions in their applicability concerning parameter ranges, in particular when dealing with transient cases, which can be overcome by the rod model [44, 54, 6, 9].

In contrast to most of the literature, that either focuses on stationary or uniaxial behavior, we seek to provide a versatile model for the transient dynamics of a slender object without any restriction on the position or shape. This is due to our particular interest in highly dynamic nonwoven production processes. They are influenced by large external forces, e.g. turbulent aerodynamic forces [73, 60] and fictitious ones (Coriolis and centrifugal) for rotational processes [5], which hinders any meaningful simplification of the used models (e.g. a dimensional reduction of the physical space). Due to the restriction in the parameter ranges of the string model we will solely be using the Cosserat rod model with a new stabilized, index-reduced formulation for its kinematics, that is non-dimensionalized in a way that facilitates the exchange of the underlying material behavior (viscous, elastic and viscoelastic). A robust discrete scheme is proposed that is applicable to arbitrary space-time domains.

The main scope of this work is incompressible, viscous material behavior. We want to lay out the origins and numerical progress of said the corresponding model shortly: A viscous string model is presented in [79, 80]. Based on the work of Ribe [85] a rod model was proposed and numerically investigated by Arne et al. in [6, 9]. The transient behavior of the rod model was considered in [7] in an Eulerian framework. Numerical challenges lie in the stiffness (through the slenderness parameter) and the general differential-algebraic character of the model, as well as the preservation of the rotation group (director triad). The Eulerian framework allows a time-independent domain, whereas in a Lagrangian framework the enlarging flow-domain also requires proper numerical treatment. Two numerical methods for general viscous rods in a Lagrangian framework can be found in literature: The

first one is a discrete geometric Lagrangian approach by Audoly et al. [12], which utilizes the center-line-spin formulation of the jet and is applied to the viscous fluid-mechanical “sewing machine” [28]. The second method originated from [4], was published in [5] and laid out the groundwork for this work. In [5], we used a Finite Volume semi-discretization in space combined with a stiffly-accurate Runge-Kutta method suitable for the resulting differential-algebraic equations (DAEs). That discrete growing process has flaws that are present independently of the chosen formulation of the jet kinematics [87]. Both approaches lack in robustness when it comes to scenarios with large deformations, which is where we pick up and continue in this work.

We aim to provide a versatile model description and numerical framework with this thesis that facilitate the exchange of the kinematics’ formulation, geometric and material model, and that embody a robust basis for the simulation of production processes with time-dependent spatial domains. The considered processes generally require transient treatment and exhibit large elongations that manifest in strongly varying solution components that have to be handled properly by the numerics.

This thesis starts with an introduction to Cosserat rod models in Chapter 2. Focusing on the viscous jet model we introduce its dimensionless formulation in a general parameterization (special cases of the parameterization are the Lagrangian and the Eulerian parameterization) and introduce a stabilized, index-reduced formulation for the kinematics. Possible set-ups in terms of the underlying space-time domain and initial and boundary conditions are investigated subsequently in great detail. The advantage of the general parameterization is that it enables the use of moving meshes, or r -refinement. In a continuous setting it would be a re-parameterization of the model that is more suitable for a numerical approximation. In Chapter 3 we propose a general moving mesh framework that is based on three parameterization layers and discuss different adaption strategies. In contrast to the many existing strategies in literature we use time-dependent spatial domains. In Chapter 4 we propose a Finite Volume method for a general partial differential-algebraic equation (PDAE) with a proper approximation of the space-time domain and a staggered approach that allows a simple central approximation of spatial fluxes and a fully implicit one for the temporal fluxes. The performance and convergence properties of the discrete scheme and the jet model are validated in Chapter 5. The moving mesh approaches are promising and show comparable performance on simple benchmark models from literature (cf. Appendix C), yet our discrete scheme with simple, central spatial flux approximations appears to be unsuitable for the jet model. Restricting to a Lagrangian parameterization we investigate the robustness of our scheme with two production processes in Chapter 6, the rotational spinning process and the melt-blowing process. In the following Appendix A we show simulations of viscoelastic material behavior to demonstrate the versatility of our model core formulation and discrete scheme. Following that, we highlight in Appendix B an alternative modeling possibility for a stabilized, index-reduced formulation of the jet kinematics that follows more closely the ideas from literature and introduce the two-dimensional and uniaxial reduction of the jet model for completeness. As already stated above, we investigate the moving mesh strategies combined with our discrete scheme and model problems taken from literature in Appendix C. In particular, we use the Burger’s equation, which is a common benchmark model for such refinement strategies. In Appendix D we present details on the derivation of our space-time Finite Volume method and give some explanatory

details on the application of the discrete scheme to the jet model. This thesis is concluded with the used notational conventions and symbols and the bibliography.

All research that was published in the course of this work is listed below.

Published works within the scope of this thesis

- W. ARNE, N. MARHEINEKE, A. MEISTER, S. SCHIESSL, AND R. WEGENER, *Finite Volume approach for the instationary Cosserat rod model describing the spinning of viscous jets*, Journal of Computational Physics, 294 (2015), pp. 20–37. [5]
- S. SCHIESSL, W. ARNE, N. MARHEINEKE, AND R. WEGENER, *DAE-index monitoring for semidiscretized viscous Cosserat rod models*, in Proceedings in Applied Mathematics and Mechanics (PAMM), Wiley, 2013, pp. 501–502. [87]
- S. SCHIESSL, N. MARHEINEKE, W. ARNE, AND R. WEGENER, *An adaptive moving mesh approach for hyperbolic conservation laws on time-dependent domains*, in Proceedings in Applied Mathematics and Mechanics (PAMM), Wiley, 2014, pp. 957–958. [88]
- S. SCHIESSL, N. MARHEINEKE, W. ARNE, AND R. WEGENER, *A Finite Volume method with staggered grid on time-dependent domains for viscous fiber spinning*, in Progress in Industrial Mathematics at ECMI 2016, Springer, 2017, to appear. [89]
- S. SCHIESSL, N. MARHEINEKE, AND R. WEGENER, *A moving mesh framework based on three parametrization layers for 1d PDEs*, in Progress in Industrial Mathematics at ECMI 2014, Springer, 2017, pp. 945–952. [90]

2 Modeling the behavior of jets and fibers

The focus of this work are one-dimensional models that can be used to describe the behavior of slender objects. A slender object is essentially a very long and thin body, therefore its 3d-dynamics can be reduced with the idea of averaging over its cross-sections, e.g. [81, 35] and more recently [74]. We use models that are based on the *Special Cosserat Theory of Rods* (see [3]) consisting of two constitutive elements, namely a curve specifying the position, and an orthonormal triad for the oriented cross-sections. In its core it uses balance laws for mass, linear and angular momentum and is applicable to all materials and physical set-ups. The rod model is completed with geometrical relations and material laws and can be formulated in various ways with respect to the used basis (invariant, director and outer), parameterization (Lagrangian, Eulerian, general), set-up (time-dependent and time-independent domain) and dimensions (with or without units). We are specifically interested in formulations of the model that allow the simple exchange of geometric relations and material laws. We will focus on one particular choice in the course of this work, which is incompressible, viscous behavior. We call a viscous rod a jet, whereas an elastic one is a fiber. We start with the invariant formulation in Lagrangian parameterization and give an overview over the formulations and parameterizations that are relevant for this work. We stick to a great extent to the outline in [107, 5].

This section is structured as follows. First we introduce the core of the rod model which is applicable to arbitrary geometric and material models. The focus in this work is on incompressible viscous behavior, nevertheless we introduce three possible geometric relations, namely inertia-free, compressible and incompressible behavior, and three material laws, namely viscous, elastic and viscoelastic. Then the possible parameterizations are discussed and the model is made dimensionless without affecting the versatility of the core. We summarize the model equations for the incompressible viscous case in Section 2.2.5 and introduce subsequently the initial and boundary conditions that will be relevant for our later numerical studies.

2.1 Theory of rods

According to the Special Cosserat Theory of Rods a three-dimensional rod in the Euclidean space \mathbb{E}^3 , equipped with the Euclidean norm $\|\cdot\|$, consists of two constitutive elements. The curve $\vec{r}: \hat{Q} \rightarrow \mathbb{E}^3$ that describes the midpoints of the cross-sections of the rod – we furthermore choose \vec{r} as mass-associated center-line – and the orientation of the cross-sections that is described with the orthonormal directors $\vec{d}_1, \vec{d}_2: \hat{Q} \rightarrow \mathbb{E}^3$. We define an orthonormal director triad $\{\vec{d}_1, \vec{d}_2, \vec{d}_3\}$ for every point of the curve by introducing

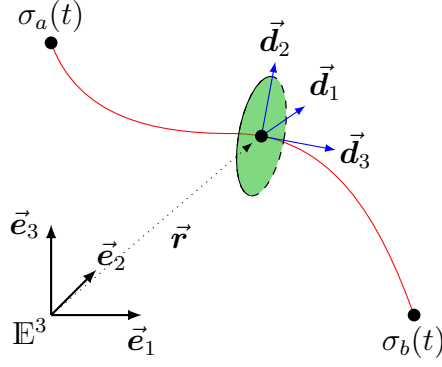


Figure 2.1: The constitutive elements of a Cosserat rod: Jet curve and orientation, denoted by \vec{r} and $\{\vec{d}_1, \vec{d}_2, \vec{d}_3\}$.

$\vec{d}_3 = \vec{d}_1 \times \vec{d}_2$. The space-time domain is

$$\hat{\mathcal{Q}} = \{(\sigma, t) \in \mathbb{R}^2 \mid \sigma \in \hat{\Omega}(t), t \in [0, T]\}$$

with the spatial domain $\hat{\Omega}(t) = [\sigma_a(t), \sigma_b(t)]$, where each material cross-section (material point) of the rod is identified with its coordinate σ , and t is the time. The interval borders are $\sigma_a, \sigma_b: [0, T] \rightarrow \mathbb{R}$ with $\sigma_a(t) < \sigma_b(t)$ for all $t \in [0, T]$. We call $\sigma_a(t)$ the left and $\sigma_b(t)$ the right side of the domain. An illustration is given in Figure 2.1.

All following quantities are assumed to be in Lagrangian parameterization and thus are dependent on $\hat{\mathcal{Q}}$. Said parameterization is determined up to a constant that gives the initial orientation and the material arrangement in the time-independent reference state.

2.1.1 Kinematics and dynamics

The kinematics of the rod are now introduced. The velocity \vec{v} and the tangent field $\vec{\tau}$ are obtained by taking the derivatives of the curve \vec{r} ,

$$\partial_t \vec{r} = \vec{v}, \quad (2.1a)$$

$$\partial_\sigma \vec{r} = \vec{\tau}. \quad (2.1b)$$

Assuming sufficient regularity, (2.1a) and (2.1b) together imply the compatibility condition for \vec{r}

$$\partial_t \vec{\tau} = \partial_\sigma \vec{v}. \quad (2.1c)$$

The director triad is orthonormal, therefore there exist vector-valued functions for the angular velocity $\vec{\omega}$ and the generalized curvature $\vec{\kappa}$. They satisfy

$$\partial_t \vec{d}_k = \vec{\omega} \times \vec{d}_k, \quad (2.2a)$$

$$\partial_\sigma \vec{d}_k = \vec{\kappa} \times \vec{d}_k, \quad (k = 1, 2, 3). \quad (2.2b)$$

Assuming sufficient regularity, (2.2a) and (2.2b) together imply the compatibility condition

$$\partial_t \vec{\kappa} = \partial_\sigma \vec{\omega} + \vec{\omega} \times \vec{\kappa}. \quad (2.2c)$$

The dynamics of the rod

$$\partial_t \sigma_M = 0, \quad (2.3a)$$

$$\partial_t \vec{p} = \partial_\sigma \vec{n} + \vec{f}, \quad (2.3b)$$

$$\partial_t \vec{h} = \partial_\sigma \vec{m} + \vec{\tau} \times \vec{n} + \vec{l} \quad (2.3c)$$

consist of a conservation law for the mass line density σ_M and balance laws for the linear and angular momentum line density, respectively given by $\vec{p} = \sigma_M \vec{v}$ and $\vec{h} = \mathbf{J}_M \cdot \vec{\omega}$. Notice that σ_M only depends on the reference state and is constant in time in the Lagrangian parameterization. A geometric model is required for the inertia tensor $\mathbf{J}_M \in \mathbb{E}^3 \otimes \mathbb{E}^3$ of the cross-sections, as well as the specification of the stress-strain response through a material law that involves the contact force \vec{n} and couple \vec{m} . External loads can be modeled within \vec{f} and \vec{l} .

The full invariant formulation of a rod [3] consists of the kinematics together with the dynamics, we call that the core. In general, for the kinematics any two of the equations of (2.1) and also any two of (2.2) can be chosen, thus different formulations of the framework are possible. We present the core with the original kinematics, an overview of the relevant combinations is shown in Remark 2.2.

System 2.1 (Invariant core). The full, invariant framework with original kinematics consists of (2.1a), (2.1b), (2.2a), (2.2b) and (2.3) and reads

$$\begin{aligned} \partial_t \vec{r} &= \vec{v}, \\ \partial_\sigma \vec{r} &= \vec{\tau}, \\ \partial_t \vec{d}_k &= \vec{\omega} \times \vec{d}_k, \\ \partial_\sigma \vec{d}_k &= \vec{\kappa} \times \vec{d}_k, & (k = 1, 2, 3) \\ \partial_t \sigma_M &= 0, \\ \partial_t \vec{p} &= \partial_\sigma \vec{n} + \vec{f}, & \vec{p} = \sigma_M \vec{v}, \\ \partial_t \vec{h} &= \partial_\sigma \vec{m} + \vec{\tau} \times \vec{n} + \vec{l}, & \vec{h} = \mathbf{J}_M \cdot \vec{\omega}. \end{aligned}$$

Remark 2.2 (Addressing different formulations). *In this work the formulation of the kinematics will be important in later discussion. Therefore we introduce a short way to address the model equations in different formulations. System 2.1 is stated with original kinematics, if at some later point in this work we need it with a different formulation (the dynamics remain untouched) we add a subscript to the system number:*

| Identifier | Kinematics | Dynamics | Description |
|-------------------------|--------------------------------|----------|-------------------------|
| System 2.1 _M | (2.1a), (2.1b), (2.2a), (2.2b) | (2.3) | Mixed (original) |
| System 2.1 _T | (2.1a), (2.1c), (2.2a), (2.2c) | (2.3) | Time evolution-biased |
| System 2.1 _S | (2.1b), (2.1c), (2.2b), (2.2c) | (2.3) | Space derivative-biased |

We used the M and T formulation already in [87] for viscous jets, the first investigation for all three formulations was done in [15] for elastic fibers.

For the discussion of closure relations for the geometry and the material we represent the invariant vector fields in terms of a chosen basis, namely an outer basis and the director basis. Thus we introduce the necessary notation.

Remark 2.3 (Notation and calculus for director basis and outer basis). *Every invariant vector field $\vec{x} \in \mathbb{E}^3$ can be decomposed in different bases. Using an outer orthonormal basis $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$ and the director basis $\{\vec{d}_1, \vec{d}_2, \vec{d}_3\}$, the vector \vec{x} reads*

$$\vec{x} = \sum_{k=1}^3 \bar{x}_k \vec{a}_k = \sum_{k=1}^3 x_k \vec{d}_k.$$

The corresponding component triples are $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) \in \mathbb{R}^3$ and $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$. To switch between different bases, transformations can be applied. The time-dependent director basis can be transformed into the outer basis with the help of the tensor-valued rotation \mathcal{R} ,

$$\mathcal{R} = \vec{a}_i \otimes \vec{d}_i = R_{ij} \vec{a}_i \otimes \vec{a}_j \in \mathbb{E}^3 \otimes \mathbb{E}^3$$

with associated orthogonal matrix $\mathbf{R} = (R_{ij}) = (\vec{d}_i \cdot \vec{a}_j) \in SO(3)$. For the coordinate triples,

$$\mathbf{x} = \mathbf{R} \cdot \bar{\mathbf{x}}$$

holds. The component triples of the partial derivatives with respect to t and σ denoted in director basis are

$$(\mathbf{R} \cdot \partial_t \bar{\mathbf{x}}) = \partial_t \mathbf{x} + (\boldsymbol{\omega} - \boldsymbol{\omega}_a) \times \mathbf{x}, \quad (\mathbf{R} \cdot \partial_\sigma \bar{\mathbf{x}}) = \partial_\sigma \mathbf{x} + \boldsymbol{\kappa} \times \mathbf{x}$$

with the angular velocity $\boldsymbol{\omega}_a$ induced by the motion of the outer basis. It follows for the time and space derivative of \mathbf{R} that

$$\partial_t \mathbf{R} = -(\boldsymbol{\omega} - \boldsymbol{\omega}_a) \times \mathbf{R} \quad \partial_\sigma \mathbf{R} = -\boldsymbol{\kappa} \times \mathbf{R}.$$

The cross-product $\mathbf{a} \times \mathbf{A} \in \mathbb{R}^{3 \times 3}$ between the vector $\mathbf{a} \in \mathbb{R}^3$ and a matrix $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ is defined by $(\mathbf{a} \times \mathbf{A}) \cdot \mathbf{x} = \mathbf{a} \times (\mathbf{A} \cdot \mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^3$. If not otherwise mentioned, the outer basis is chosen to be time-independent.

With the help of Remark 2.3 the full invariant system is denoted in director and outer basis. The curve $\vec{\mathbf{r}}$ and the director triad $\vec{\mathbf{d}}_i, i = 1, 2, 3$, are formulated in the outer basis, the remaining vectors in director basis. The tangent $\vec{\boldsymbol{\tau}}$ and curvature $\vec{\boldsymbol{\kappa}}$ represented in director basis with $\boldsymbol{\tau}$ and $\boldsymbol{\kappa}$ are indicators for the distortion of the rod. In particular we have in material parameterization: τ_1, τ_2 as shear strains, τ_3 as stretching strain and $\|\boldsymbol{\tau}\|$ as the overall dilatation. Furthermore the measures for bending are κ_1, κ_2 and κ_3 is the torsion of the rod. Analogously $\mathbf{n}_1, \mathbf{n}_2$ measure the shear stress and \mathbf{n}_3 the tension, as well as $\mathbf{m}_1, \mathbf{m}_2$ the bending torque and \mathbf{m}_3 the twisting torque.

System 2.4 (Core in director basis). The core of the rod model in director and outer basis reads

$$\begin{aligned}
 \partial_t \bar{\mathbf{r}} &= \mathbf{R}^T \cdot \mathbf{v}, \\
 \partial_\sigma \bar{\mathbf{r}} &= \mathbf{R}^T \cdot \boldsymbol{\tau}, \\
 \partial_t \mathbf{R} &= -\boldsymbol{\omega} \times \mathbf{R}, \\
 \partial_\sigma \mathbf{R} &= -\boldsymbol{\kappa} \times \mathbf{R}, \\
 \partial_t \sigma_M &= 0, \\
 \partial_t \mathbf{p} &= \mathbf{p} \times \boldsymbol{\omega} + \partial_\sigma \mathbf{n} + \boldsymbol{\kappa} \times \mathbf{n} + \mathbf{f}, & \mathbf{p} &= \sigma_M \mathbf{v}, \\
 \partial_t \mathbf{h} &= \mathbf{h} \times \boldsymbol{\omega} + \partial_\sigma \mathbf{m} + \boldsymbol{\kappa} \times \mathbf{m} + \boldsymbol{\tau} \times \mathbf{n} + \mathbf{l}, & \mathbf{h} &= \mathbf{J}_M \cdot \boldsymbol{\omega}.
 \end{aligned}$$

The matrix $\mathbf{J}_M \in \mathbb{R}^3 \otimes \mathbb{R}^3$ contains the components of the inertia tensor with respect to the director basis.

Remark 2.5 (Compatibility conditions). *The invariant compatibility conditions (2.1c) and (2.2c) can be expressed in director basis by*

$$\begin{aligned}
 \partial_t \boldsymbol{\tau} &= \partial_\sigma \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega}, \\
 \partial_t \boldsymbol{\kappa} &= \partial_\sigma \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega}.
 \end{aligned}$$

2.1.2 Geometric model

Now the geometric relations of the rod are specified, more precisely the model for \mathbf{J}_M . We always assume homogeneous, circular cross-sections. Here, homogeneous means that the mass density is constant within the cross-section. Then the inertia tensor \mathbf{J}_M is, contrary to the mass line density σ_M , time-dependent because it is linked to the dynamic of the director triad through

$$\mathbf{J}_M = (J_M)_{ij} \vec{\mathbf{d}}_i \otimes \vec{\mathbf{d}}_j, \quad \mathbf{J}_M = (J_M)_{ij} = \frac{\sigma_M \sigma_V}{\|\boldsymbol{\tau}\|} \mathbf{M}_{in}, \quad \mathbf{M}_{in} = \mathbf{L}_2. \quad (2.4)$$

with the volume line density σ_V , that implies that the area of the cross-section is given by $A = \sigma_V / \|\boldsymbol{\tau}\|$ and the matrix $\mathbf{L}_p = \text{diag}(1, 1, p)/(4\pi)$ for $p \in \mathbb{R}$. The matrix \mathbf{M}_{in} is specific to a circular shape of the cross-sections. Other cross-section shapes of the rod give the same structure, solely the matrix \mathbf{M}_{in} is changed. A remark on the choice of the director basis: Note the link between \mathbf{J}_M and \mathbf{J}_M : the inertia matrix \mathbf{J}_M contains the components of \mathbf{J}_M in director basis, thus using the director basis is a great simplification here. Now we need an ansatz how the three-dimensional geometry changes with respect to the deformations of the Cosserat rod. There are three possibilities considered, namely

$$\mathbf{J}_M = \mathbf{0}, \quad (2.5a)$$

$$\partial_t \mathbf{J}_M = \mathbf{0}, \quad (2.5b)$$

$$\partial_t (\|\boldsymbol{\tau}\| \mathbf{J}_M) = \mathbf{0}. \quad (2.5c)$$

representing inertia free behavior, immutable cross-sectional area (compressible, standard interpretation [3]) and immutable volume (shape-preserving, incompressible [6]), respectively. For (2.5a) the actual shape of the cross-section has no influence, thus $\vec{h} = \mathbf{0}$. Equations (2.5b) and (2.5c) are fulfilled if the two equations

$$\partial_t A = 0, \quad (2.6a)$$

$$\partial_t \sigma_V = 0. \quad (2.6b)$$

are fulfilled, respectively. This can be seen by using (2.4) in (2.5) and applying (2.3a).

Remark 2.6 (Line densities). *From the embedding in the 3d-continuum mechanics the mass line density σ_M is the integral of the mass density over the cross-section of the jet in the reference state and thus time-independent [3]. If the cross-sections with referential area A^o have homogeneous mass distribution with referential density ρ^o and referential elongation τ^o , then the mass line density is $\sigma_M = \rho^o A^o \tau^o$. It can be feasible to introduce ρ as the density, then we have*

- $\rho = \rho^o / \|\boldsymbol{\tau}\|$, $\sigma_V = A^o \|\boldsymbol{\tau}\|$ for the compressible geometry model and
- $\rho = \rho^o / \tau^o$, $\sigma_V = A^o \tau^o$ for the incompressible one.

2.1.3 Material laws

In the theory of rods a material law specifies the relationship between the contact force \vec{n} and couple \vec{m} , and the change of shape of the rod represented with $\vec{\tau}$ and $\vec{\kappa}$. Such a relationship is also called a constitutive relation and it must distinguish the material response of e.g. a rubber band to that of a stream of honey. The material laws have to follow the *Principle of Frame-Indifference* [3] that states that the material response should be unaffected by rigid body-motions and time translations. Invariance to time translations is easily achieved by avoiding direct dependence on time. The invariance towards rigid-body motion is ensured by formulating the material law in director basis as a relation between the stresses \mathbf{n} , \mathbf{m} and the distortion measures $\boldsymbol{\tau}$, $\boldsymbol{\kappa}$ [3]. We present material laws for elastic and viscous behavior and point out one possibility to combine them both for viscoelastic behavior. We do this to give a deeper insight into the diversity of the constitutive relations. After this section we will focus on viscous behavior.

Elastic fiber There exist numerous elastic variants suitable for the Cosserat rod theory, whereas the Bernoulli-Euler model is the one mainly used for fiber applications and also in [107]. A modified Kirchhoff-constraint that enforces a unstretchable and shear free rod is used

$$\boldsymbol{\tau} = \tau^o \mathbf{e}_3 \quad (2.7a)$$

and furthermore equipped with a affine linear relation for the contact couple (cf. [3, 15])

$$\boldsymbol{\kappa} = \mathbf{J}_E^{-1} \cdot \mathbf{m} + \boldsymbol{\kappa}^o. \quad (2.7b)$$

Here \mathbf{e}_i , $i = 1, 2, 3$, is the canonical basis tuple and \mathbf{J}_E a positive-definite tensor-valued function. The reference distortion measures $\boldsymbol{\kappa}^o$ and τ^o denote the stress-free state of the jet, a popular choice is $\tau^o = 1, \boldsymbol{\kappa}^o \equiv \mathbf{0}$. That means that the stress-free state of the rod is unbent and arc-length parameterized. The material properties refer to the time-independent reference state, thus

$$\partial_t \mathbf{J}_E = \mathbf{0}, \quad \partial_t \tau^o = \mathbf{0}, \quad \partial_t \boldsymbol{\kappa}^o = \mathbf{0}. \quad (2.8)$$

With homogeneous, circular cross-sections we have

$$\mathbf{J}_E = \frac{EA^2}{\tau^o} \mathbf{M}_E, \quad \mathbf{M}_E = \mathbf{L}_{1/(1+\nu_P)} \quad (2.9)$$

with the Poisson number ν_P that expresses the relation of Young's modulus E and shear modulus G , namely $E/G = 2(1 + \nu_P)$. In the three-dimensional incompressible case the relation $G = E/3$ holds as $\nu_P = 1/2$. The material law is completed with a model for E , we choose it as a constant as shown in Remark 2.7.

Remark 2.7 (Constant Young's modulus). *Assuming that Young's modulus is constant implies by using (2.8) with (2.9) inserted that*

$$\partial_t A = 0 \quad (2.10)$$

has to hold. Both the incompressible and compressible geometry model ensure that, for the inertia-free geometry model we can use (2.10) as an additional requirement.

Viscous jet Arne et al. [6] model the viscous material with a constitutive law that corresponds to the proposition of Ribe for an Eulerian framework [85, 86]. There, a generalized Kirchhoff constraint

$$\boldsymbol{\tau} = \tau_3 \mathbf{e}_3$$

is used. Note that the overall dilatation $\|\boldsymbol{\tau}\| = \tau_3$ now only represents the elongation. The material model specifies the tension \mathbf{n}_3 and the torques \mathbf{m} through a linear ordinary differential equation (ODE) in terms of the distortion measures. The shear stresses \mathbf{n}_1 and \mathbf{n}_2 become Lagrangian multipliers, overall we have

$$\begin{aligned} \partial_t \tau_3 &= \tau_3 A_\mu^{-1} \mathbf{n}_3, & \tau_1 &= 0, \quad \tau_2 = 0, \\ \partial_t \boldsymbol{\kappa} &= \tau_3 \mathbf{J}_\mu^{-1} \cdot \mathbf{m}. \end{aligned}$$

The material properties \mathbf{J}_μ and A_μ refer to the time-independent reference state in terms of powers of τ_3

$$\partial_t (\tau_3^2 \mathbf{J}_\mu) = \mathbf{0}, \quad \partial_t (\tau_3 A_\mu) = 0. \quad (2.11)$$

With homogeneous, circular cross-sections it follows that

$$\begin{aligned} A_\mu &= 3\mu A, \\ \mathbf{J}_\mu &= 3\mu A^2 \mathbf{M}_\mu, & \mathbf{M}_\mu &= \mathbf{L}_{2/3}, \end{aligned}$$

where the unknown dynamic (or shear) viscosity μ appears. The term 3μ is for Newtonian fluids the elongational viscosity (Trouton's law). It is understood as the resistance of a fluid to any elongational flow ([108] and references within). The material law is completed with a model for μ . If not otherwise mentioned we choose it as a constant as shown in Remark 2.8.

Remark 2.8 (Constant viscosity). *Assuming a constant viscosity ($\partial_t \mu = 0$) implies by using (2.11) that*

$$\partial_t \sigma_V = 0 \tag{2.12}$$

has to hold. The sole geometry model that is compatible with (2.12) is the incompressible one, (2.5c).

Viscoelastic jet With the industrial application in mind that motivated this work, the need for a viscoelastic model that unifies the elastic and viscous behavior comes to mind. Just recently, Arne, Marheineke and Wegener introduced a incompressible viscoelastic model for the Cosserat rod theory [10] by combining the elastic and viscous models. They utilize a generalized Kirchhoff constraint and the viscous relation for the tension and a Maxwell-like relaxation of the viscous constitutive law for the torques. The model contains the incompressible viscous and elastic rod as asymptotic limits (cf. Appendix A.2) and is given by

$$\begin{aligned} \partial_t \tau_3 &= \tau_3 A_\mu^{-1} \mathbf{n}_3, & \tau_1 &= 0, \quad \tau_2 = 0, \\ \partial_t \boldsymbol{\kappa} &= \tau_3 \mathbf{J}_\mu^{-1} \cdot \mathbf{m} + \mathbf{J}_E^{-1} \cdot \partial_t \mathbf{m}. \end{aligned}$$

Following [10], an incompressible geometry model is assumed (giving $\nu_P = 1/2$ as the Poisson number), which implies a scalar relaxation time $\theta = 3\mu/E$ (time scale of the viscoelastic response [23]) that is derived by calculating $\mathbf{J}_\mu \cdot \mathbf{J}_E^{-1}/\tau_3$, i.e. we have

$$\partial_t \boldsymbol{\kappa} = \tau_3 \mathbf{J}_\mu^{-1} \cdot (\mathbf{m} + \theta \partial_t \mathbf{m}).$$

The model is completed with further evolution equations for μ and θ . In particular, Arne, Marheineke and Wegener [10] extend the rod model furthermore by considering an additional balance law for the temperature T (see the following Section 2.1.4) and prescribe a temperature-dependent viscosity by applying an Arrhenius law, i.e.

$$\mu(T) = c_1 \exp(c_2/T), \tag{2.13}$$

whereas the Young's modulus is assumed constant. Note that c_1 and c_2 are dimensioned material constants.

Remark 2.9. *The temperature-dependent model (2.13) for the viscosity could also be used in combination with our viscous material law, but intrinsically correct material behavior requires the above viscoelastic material law. The viscous behavior in the limit $\mu \rightarrow \infty$ ($T \rightarrow 0$) is a rigid body, but the materials considered in our industrial application transition to elastic solids for decreasing temperature.*

2.1.4 Energy balance

The previous introduction of the theory of rods only included mass, linear and angular momentum balances. To account for industrial spinning processes that are significantly affected by the cooling of the material we would need to consider temperature dependent material modeling. For that purpose we follow [107] and introduce a energy balance law. Consider the specific enthalpy as a function of the temperature T and its associated derivative with respect to the temperature, that is, the specific heat capacity c_p of the rod. Using the mass conservation law, the energy balance simplifies to

$$c_p \partial_t (\sigma_M T) = q.$$

We restrict warming and cooling effects to the thermal exchange with a surrounding air stream of temperature T_{air} , which is described using a heat transfer function α_{air} . The thermal exchange happens across the circumference πd with diameter $d = (2/\sqrt{\pi})\sqrt{A}$ and cross-sectional area A of the fiber. The source term for convective air cooling of the fiber then becomes

$$q = - \|\boldsymbol{\tau}\| \pi d \alpha_{air} (T - T_{air}).$$

The heat transfer function α_{air} uses a heuristic based model that was initially formulated for a vertical incident flow on a cylinder and then modified for arbitrary incident flow direction on grounds of experimental data, see [8, 107] and references within. We introduce the energy balance for completeness, it will solely be used in the appendix, more precisely in Appendix A.3.

2.1.5 External loads

The model for the behavior of a fiber is driven by external loads \mathbf{f} and \mathbf{l} . As an example, consider the gravitational force acting in a specified direction $\bar{\mathbf{e}}_g$ and given by

$$\mathbf{f}_g = g \sigma_M \mathbf{R} \cdot \bar{\mathbf{e}}_g$$

with the gravitational constant g . Only considering the gravitational force would give $\mathbf{f} = \mathbf{f}_g$ and $\mathbf{l} = \mathbf{0}$ for the external loads. In our scenarios we will later consider the more complicated aerodynamic forces for the melt-blowing process and fictitious rotational loads for the rotational spinning process. Models for both are already available in literature, e.g. [75, 5]. We will introduce them in non-dimensionalized form in the context of the application that they are used for, namely in Section 6.1 (rotational spinning process) and Section 6.2 (melt-blowing process).

2.2 Formulations for the viscous jet model

Up until now the theory was introduced in Lagrange parameterization with dimensioned quantities. In this section we first introduce a type concept that facilitates the formulation in any time-dependent parameterization, with the most prominent example of an Eulerian

| | model core | viscous |
|--------------|--|------------------------------|
| type-0-field | $\bar{\mathbf{r}}, \mathbf{R}, \mathbf{v}, \boldsymbol{\omega}, \mathbf{m}, \mathbf{n}$ | $A_\mu, \mathbf{J}_\mu, \mu$ |
| type-1-field | $\boldsymbol{\tau}, \boldsymbol{\kappa}, \sigma_M, \sigma_V, \mathbf{J}_M, \mathbf{f}, \mathbf{l}, \mathbf{p}, \mathbf{h}$ | |

Table 2.1: Type categorization of the quantities appearing in System 2.4 and the viscous material law.

parameterization (arc-length at all times). The Eulerian parameterization is especially important since its arc-length constraint allows to facilitate the model equations, which might not be possible in a general parameterization. We furthermore discuss the non-dimensionalization of the model and the parameterization of the rotational group as well as our approach to formulate a set of kinematics that takes the underlying DAE characteristics of the model into account. We conclude this section with an overview of the finalized viscous jet model.

2.2.1 General parameterization

Following [107], a type concept for the unknowns is introduced that facilitates the reformulation of the model equations in any time-dependent parameterization. A suitable transformation to a general (time-dependent) parameterization is given with the bijective function

$$\Psi(\cdot, t): \Omega(t) \rightarrow \hat{\Omega}(t)$$

for $t \in [0, T]$. The spatial domain $\hat{\Omega}(t) = [\sigma_a(t), \sigma_b(t)]$ represents the material (or referential) parameterization and $\Omega(t) = [s_a(t), s_b(t)]$ represents the general parameterization. We call σ_a, s_a the left side, and σ_b, s_b the right side of the domain. The inverse transformation is denoted with $\Phi(\cdot, t) := \Psi^{-1}(\cdot, t)$ for fixed t . It holds that $\Psi(\Phi(\sigma, t), t) = \sigma$. We always assume sufficient regularity of Ψ , in particular for the following Ψ has to be continuously differentiable in both variables. The unknowns are categorized in type- n -fields ($n \in \mathbb{Z}$) and will be transformed according to their type with the help of

$$\vec{\mathbf{f}}(s, t) = \chi(s, t)^n \hat{\vec{\mathbf{f}}}(\Psi(s, t), t), \quad \chi(s, t) := \partial_s \Psi(s, t) \quad (2.14)$$

where $\hat{\vec{\mathbf{f}}}$ is an unknown in in Lagrange parameterization and $\vec{\mathbf{f}}$ the respective unknown in general parameterization with its general space-time domain given by

$$\mathcal{Q} = \{(s, t) \in \mathbb{R}^2 | s \in \Omega(t), t \in [0, T]\}. \quad (2.15)$$

To distinguish between the unknowns, the Lagrangian ones are from now on denoted with a hat $\hat{\cdot}$ (in contrast to the beginning of this chapter) and all unknowns in the general parameterization receive no special mark. The types of the unknowns of the viscous jet model are listed in Table 2.1. The type categorization allows that the fields keep their physical behavior and relations (point-related observables, densities, derivatives, etc.), furthermore the equations will be mostly invariant with respect to re-parameterization. Any

time-dependent transformation that is applied to the model equations will solely introduce additional convective terms using the parameter speed u , which is defined as

$$u(\Phi(\sigma, t), t) := \partial_t \Phi(\sigma, t) = - \frac{\partial_t \Psi(s, t)}{\chi(s, t)} \Big|_{s=\Phi(\sigma, t)}. \quad (2.16)$$

It represents the rate of change of $s = \Phi(\sigma, t)$ that is identified with the material point σ . Moreover, with the chain rule it follows for type- n -fields:

$$\partial_t \hat{\vec{f}}(\sigma, t) = \frac{1}{\chi(s, t)^n} \left[\partial_t \vec{f}(s, t) + u(s, t) \partial_s \vec{f}(s, t) + n \vec{f}(s, t) \partial_s u(s, t) \right] \Big|_{s=\Phi(\sigma, t)}, \quad (2.17a)$$

$$\partial_\sigma \hat{\vec{f}}(\sigma, t) = \frac{1}{\chi(s, t)^{n+1}} \left(\partial_s \vec{f}(s, t) - n \frac{\partial_s \chi(s, t)}{\chi(s, t)} \vec{f}(s, t) \right) \Big|_{s=\Phi(\sigma, t)}. \quad (2.17b)$$

The parameter speed u is a new unknown in the system and is yet to be determined. The simple case $u \equiv 0$ yields a time-independent re-parameterization. If additionally $\chi \equiv 1$ is chosen we obtain the original Lagrangian parameterization. Another very well known special case is the Eulerian parameterization, which is an arc-length parameterization. It holds that $\|\hat{\vec{\tau}}\| \equiv 1$. This relation is added as a constraint to the model equation making u the corresponding Lagrangian parameter, and since the dilatation measure is treated as a type-1-field we have

$$\|\hat{\vec{\tau}}\| \equiv \frac{1}{\chi}.$$

In this work we aim for a general parameterization that is neither strictly Lagrangian nor Eulerian – both cases have already been studied e.g. in [7, 5]. Here, the identification between material points and coordinates $s \in \Omega(t)$ will be arbitrarily specified, similar approaches are e.g. studied in [52, 37], they call it *arbitrary Lagrangian-Eulerian* (ALE) description. In particular they started with the idea to have a parameterization that is Lagrangian in some parts and Eulerian in other parts of the domain. We consider in this work to determine the parameterization by an *adaptive mesh refinement* strategy (AMR), in particular by a r -refinement. We will get to this in Section 3.1. Nevertheless we want to mention already one important property in the moving mesh context that our model inherently fulfills due to the type definition.

Remark 2.10 (Geometric conservation law). *Take the spatial derivative of (2.16) to obtain*

$$\partial_t \chi + \partial_s (u \chi) = 0. \quad (2.18)$$

Equation (2.18) is known as the geometric conservation law (GCL) [100] and governs volume conservation under an arbitrary mapping Ψ . Assume now an arbitrary field \hat{k} that follows a conservation law in Lagrangian parameterization, for simplicity of the demonstration we choose $\partial_t \hat{k} = 0$. If k is chosen to be type-0, we obtain with $k(s, t) = \hat{k}(\Psi(s, t), t)$ the non-conservative equation

$$\partial_t k + u \partial_s k = 0. \quad (2.19)$$

Combining (2.18) and (2.19) yields the conservative form, i.e.

$$\partial_t(\chi k) + \partial_s(u\chi k) = 0. \quad (2.20)$$

If k is chosen to be type-1, i.e., $k(s, t) = \chi(s, t)\hat{k}(\Psi(s, t), t)$, the conservative form is already implicitly given, its equation then becomes

$$\partial_t k + \partial_s(uk) = 0.$$

For numerical conservation the conservative form is inevitable, which in turn means that the natural choice would be type-1. The choice of the conservative or non-conservative form can have a significant effect on the accuracy and stability of a numerical scheme. Guillard and Farhat [45] found that satisfying the discrete GCL is a sufficient condition for ensuring that the underlying (proper) numerical method is at least first-order time-accurate. In that sense the GCL ensures that a constant solution is reproduced, independent of the magnitude of u and the distortion of the mesh. Our viscous jet model inherently fulfills the GCL by its type definitions of the unknowns. In general this does not have to be the case, e.g. the model equations could be formulated in non-conservative form (cf. (2.19)). However the general recommendation is to formulate conservation laws in conservative form (cf. [100, 51, 22] and more recently [59, 110]), especially in the context of moving mesh methods that we discuss in Chapter 3, e.g. [27, 13, 49]).

Transforming the domain The transformation $\Phi(\cdot, t)$ is bijective for all $t \in [0, T]$ and we want to identify the left and right side of $\hat{\Omega}(t)$ and $\Omega(t)$ with each other, thus we assume

$$\Phi(\sigma_a(t), t) = s_a(t), \quad (2.21a)$$

$$\Phi(\sigma_b(t), t) = s_b(t). \quad (2.21b)$$

This means that $\partial_\sigma \Phi > 0$. Applying a time derivative to (2.21) yields

$$\frac{d}{dt}s_a(t) = \partial_t \Phi(\sigma_a(t), t) + \partial_\sigma \Phi(\sigma, t)|_{\sigma=\sigma_a(t)} \frac{d}{dt}\sigma_a(t), \quad (2.22a)$$

$$\frac{d}{dt}s_b(t) = \partial_t \Phi(\sigma_b(t), t) + \partial_\sigma \Phi(\sigma, t)|_{\sigma=\sigma_b(t)} \frac{d}{dt}\sigma_b(t). \quad (2.22b)$$

We investigate u at the boundaries

$$u(s_a(t), t) \stackrel{(2.21a)}{=} u(\Phi(\sigma_a(t), t), t) \stackrel{(2.16)}{=} \partial_t \Phi(\sigma_a(t), t) \stackrel{(2.22a)}{=} \frac{d}{dt}s_a(t) - \partial_\sigma \Phi(\sigma, t)|_{\sigma=\sigma_a(t)} \frac{d}{dt}\sigma_a(t), \quad (2.23a)$$

$$u(s_b(t), t) \stackrel{(2.21b)}{=} u(\Phi(\sigma_b(t), t), t) \stackrel{(2.16)}{=} \partial_t \Phi(\sigma_b(t), t) \stackrel{(2.22b)}{=} \frac{d}{dt}s_b(t) - \partial_\sigma \Phi(\sigma, t)|_{\sigma=\sigma_b(t)} \frac{d}{dt}\sigma_b(t). \quad (2.23b)$$

The choice of the actual domains is linked to the chosen physical set-up and yields different conditions for u . They will be presented when the set-ups are introduced in Section 2.3. Nevertheless we show now how we represent the space-time domain in general for our application.

Notation 2.11 (Time-dependent parameter domains). In this work we denote space-time domains with time-dependent spatial domains as

$$\hat{\Omega}(t) = [-l(t), -r(t)] \quad \text{and} \quad \Omega(t) = [-\mathcal{L}(t), -\mathcal{R}(t)]$$

and assume without loss of generality $r(0) = \mathcal{R}(0) = 0$. We call them time-dependent domains (material and general, respectively), which refers to the interval boundaries of the spatial domains $\hat{\Omega}$ and Ω being time-dependent. The initial parameter interval lengths are denoted with $|\hat{\Omega}(0)| = l(0) = l_0 \geq 0$ and $|\Omega(0)| = \mathcal{L}(0) = \mathcal{L}_0 \geq 0$. The sign change of the boundary functions will give a physical interpretation suitable to our application later in this work. For a better readability we address l, \mathcal{L} as the *left* side and r, \mathcal{R} as the *right* side of the domain. We repeat (2.23) and use the relation between the derivatives of a function and its inverse $\partial_\sigma \Phi(\sigma, t)|_{\sigma=\Psi(s,t)} = 1/\partial_s \Psi(s, t) = 1/\chi(s, t)$

$$\begin{aligned} u(-\mathcal{L}(t), t) &= -\frac{d}{dt} \mathcal{L}(t) + \frac{1}{\chi(-\mathcal{L}(t), t)} \frac{d}{dt} l(t), \\ u(-\mathcal{R}(t), t) &= -\frac{d}{dt} \mathcal{R}(t) + \frac{1}{\chi(-\mathcal{R}(t), t)} \frac{d}{dt} r(t). \end{aligned} \quad (2.24)$$

Looking at Notation 2.11, the time-independent domains are a special case.

Notation 2.12 (Time-independent parameter domains). Any space-time domains with $|\hat{\Omega}| = l_0 > 0$ and $|\Omega| = \mathcal{L}_0 > 0$ for all times can be denoted without loss of generality as

$$\hat{\Omega} = [-l_0, 0] \quad \text{and} \quad \Omega = [-\mathcal{L}_0, 0].$$

We call them time-independent domains (material and general, respectively), which refers to the interval boundaries of the spatial domains $\hat{\Omega}$ and Ω being time-independent.

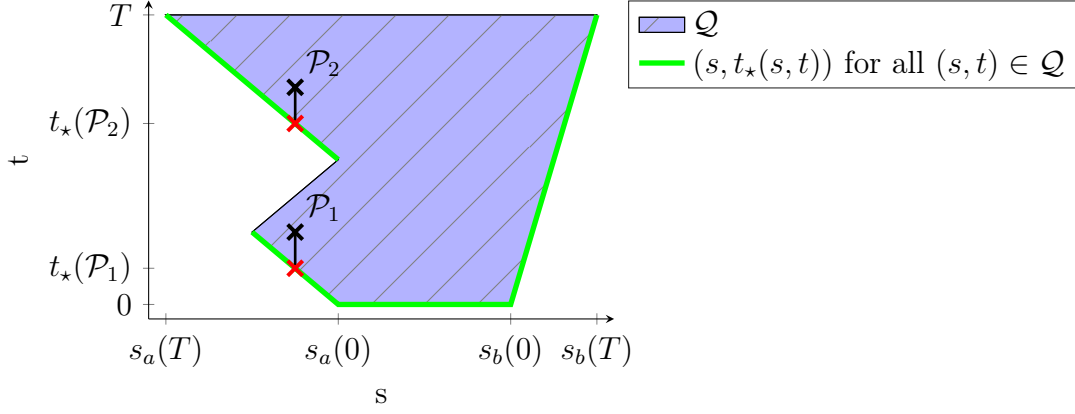
Before we get to the transformation of the model equations, we introduce the creation time. It will be helpful for later explanation and is a function that assigns every $(s, t) \in \mathcal{Q}$ the point in time from which on it was continuously within \mathcal{Q} .

Definition 2.13 (Creation time). Let a function $t_\star: \mathcal{Q} \rightarrow \mathbb{R}_0^+$ be given by

$$t_\star(s, t) = \min\{t' \mid (s, t'') \in \mathcal{Q} \text{ for all } t'' \in [t', t]\}. \quad (2.25)$$

It assigns every tuple (s, t) the time of entry into the domain and we call it creation time.

The abstract definition of the creation time seems complicated, yet it does not restrict the shape of the space-time domain in any way. As an example we illustrate a domain that grows and shrinks on one side, cf. Figure 2.2. In it the exemplary space-time tuples \mathcal{P}_1 and \mathcal{P}_2 have the same spatial coordinate, but different creation times.


 Figure 2.2: Illustration of the creation time t_* .

Transforming the model equations The transformation introduced in (2.14) is now applied to the jet model. Since we are solely using viscous material behavior in the remainder of this work, we give a summary for other material behavior in Appendix A.1.

System 2.14 (Incompressible viscous jet in director basis). The behavior of a viscous jet with homogeneous, circular cross-sections, incompressible geometry and generalized Kirchhoff constraint, expressed in outer and director basis and put in general parameterization, is given by

$$\begin{aligned}
 \partial_t \bar{\mathbf{r}} &= \mathbf{R}^T \cdot (\mathbf{v} - u\boldsymbol{\tau}), \\
 \partial_s \bar{\mathbf{r}} &= \mathbf{R}^T \cdot \boldsymbol{\tau}, \\
 \partial_t \mathbf{R} &= -(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \times \mathbf{R}, \\
 \partial_s \mathbf{R} &= -\boldsymbol{\kappa} \times \mathbf{R}, \\
 \partial_t \sigma_M + \partial_s(u\sigma_M) &= 0, \quad \partial_t \sigma_V + \partial_s(u\sigma_V) = 0, \\
 \partial_t \mathbf{p} + \partial_s(u\mathbf{p}) &= \mathbf{p} \times \boldsymbol{\omega} + \partial_s \mathbf{n} + \boldsymbol{\kappa} \times \mathbf{n} + \mathbf{f}, & \mathbf{p} &= \sigma_M \mathbf{v}, \\
 \partial_t \mathbf{h} + \partial_s(u\mathbf{h}) &= \mathbf{h} \times \boldsymbol{\omega} + \partial_s \mathbf{m} + \boldsymbol{\kappa} \times \mathbf{m} + \boldsymbol{\tau} \times \mathbf{n} + \mathbf{l}, & \mathbf{h} &= \mathbf{J}_M \cdot \boldsymbol{\omega}, \\
 \partial_t \tau_3 + \partial_s(u\tau_3) &= \frac{1}{3\mu} \frac{\tau_3^2}{\sigma_V} \mathbf{n}_3, \quad \tau_1 = 0, \quad \tau_2 = 0, \\
 \partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa}) &= \frac{1}{3\mu} \frac{\tau_3^3}{\sigma_V^2} \mathbf{M}_\mu^{-1} \cdot \mathbf{m}, \\
 \partial_t \mu + u\partial_s \mu &= 0
 \end{aligned}$$

with $\mathbf{J}_M = (\sigma_M \sigma_V / \tau_3) \mathbf{M}_{in}$. The parameter speed u is still a degree of freedom in the system.

Remark 2.15 (Compatibility conditions). The invariant compatibility conditions are expressed in director basis and transformed to general parameterization

$$\partial_t \boldsymbol{\tau} + \partial_s(u\boldsymbol{\tau}) = \partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega}, \quad (2.26a)$$

$$\partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa}) = \partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega}. \quad (2.26b)$$

2.2.2 Non-dimensionalization

We want to remove the physical units from the model equations and reduce the number of parameters as well as give the possibility to consider certain quantities in the system with different scales. Any quantity z of the model can be decomposed to $z = \tilde{z}z_0$ with \tilde{z} as the dimensionless quantity and z_0 as reference value. The model core has various physical properties involving three physical SI-units, namely length, time and weight. We introduce three typical constant values for scaling (marked with the subscript \star) representing the typical length (r_\star [m]), the typical velocity (v_\star [m/s]) and typical weight ($\sigma_{M\star}$ [kg/m]). The introduction of an additional typical length for the cross-section diameter (using $\sigma_{V\star}$ [m²]) creates the dimensionless parameter

$$\varepsilon = \frac{\sqrt{\sigma_{V\star}}}{r_\star},$$

also called slenderness ratio. All quantities were made dimensionless by scaling with the following reference values

$$\begin{aligned} r_0 = s_0 = r_\star, & \quad v_0 = v_\star, & \quad \sigma_{M0} = \sigma_{M\star}, & \quad \sigma_{V0} = \sigma_{V\star}, \\ t_0 = r_\star/v_\star, & \quad \kappa_0 = 1/r_\star, & \quad \omega_0 = v_\star/r_\star, & \quad n_0 = \sigma_{M\star}v_\star^2, \\ m_0 = \sigma_{M\star}v_\star^2r_\star, & \quad u_0 = v_\star, & \quad f_0 = \sigma_{M\star}v_\star^2/r_\star, & \quad l_0 = \sigma_{M\star}v_\star^2\varepsilon^2. \end{aligned} \quad (2.27)$$

The viscous material law is scaled with an additional reference value by using $\mu_0 = \mu_\star$ [kg/(m s)]. This leads to the dimensionless Reynolds number (ratio between inertia and viscosity)

$$\text{Re} = \frac{\sigma_{M\star}v_\star r_\star}{\mu_\star \sigma_{V\star}}.$$

System 2.16 (Dimensionless incompressible viscous jet in director basis).

System 2.14 is now stated in dimensionless form using the above reference values. To simplify the notation we remove the $\tilde{\cdot}$ mark of dimensionless unknowns.

$$\begin{aligned} \partial_t \bar{\mathbf{r}} &= \mathbf{R}^T \cdot (\mathbf{v} - u\boldsymbol{\tau}), \\ \partial_s \bar{\mathbf{r}} &= \mathbf{R}^T \cdot \boldsymbol{\tau}, \\ \partial_t \mathbf{R} &= -(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \times \mathbf{R}, \\ \partial_s \mathbf{R} &= -\boldsymbol{\kappa} \times \mathbf{R}, \\ \partial_t \sigma_M + \partial_s(u\sigma_M) &= 0, & \quad \partial_t \sigma_V + \partial_s(u\sigma_V) &= 0, \\ \partial_t \mathbf{p} + \partial_s(u\mathbf{p}) &= \mathbf{p} \times \boldsymbol{\omega} + \partial_s \mathbf{n} + \boldsymbol{\kappa} \times \mathbf{n} + \mathbf{f}, & \quad \mathbf{p} &= \sigma_M \mathbf{v}, \\ \partial_t \mathbf{h} + \partial_s(u\mathbf{h}) &= \mathbf{h} \times \boldsymbol{\omega} + \frac{1}{\varepsilon^2} (\partial_s \mathbf{m} + \boldsymbol{\kappa} \times \mathbf{m} + \boldsymbol{\tau} \times \mathbf{n}) + \mathbf{l} & \quad \mathbf{h} &= \mathbf{J}_M \cdot \boldsymbol{\omega}, \\ \partial_t \tau_3 + \partial_s(u\tau_3) &= \frac{\text{Re}}{3\mu} \frac{\tau_3^2}{\sigma_V} \mathbf{n}_3, & \quad \tau_1 &= 0, \quad \tau_2 = 0, \\ \partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa}) &= \frac{\text{Re}}{\varepsilon^2 3\mu} \frac{\tau_3^3}{\sigma_V} \mathbf{M}_\mu^{-1} \cdot \mathbf{m}, \\ \partial_t \mu + u\partial_s \mu &= 0 \end{aligned}$$

with $\mathbf{J}_M = (\sigma_M \sigma_V / \tau_3) \mathbf{M}_{in}$. The parameter speed u is a degree of freedom in the system.

Remark 2.17 (Dimensionless numbers). *There exist many possibilities in the choice of the typical values, in particular the choice of $\sigma_{M\star}$ and $\sigma_{V\star}$ can have an impact on the dimensionless numbers. Assume some given physical parameters of the jet: cross-section diameter D , length L and velocity U as well as density ρ and viscosity μ . We want to choose the typical values such that the Reynolds number becomes*

$$Re = \frac{\sigma_{M\star} v_\star r_\star}{\mu_\star \sigma_{V\star}} \stackrel{!}{=} \frac{\rho U L}{\mu},$$

thus $r_\star = L$, $v_\star = U$, $\mu_\star = \mu$ and $\sigma_{M\star}$, $\sigma_{V\star}$ have to be chosen with respect to the same typical area. We use $A_0 = (\pi/4)D^2$, i.e. $\sigma_M = \rho A_0$ and $\sigma_{V\star} = A_0$. This has a consequence on the slenderness parameter ε , it is not directly the ratio between the cross-section diameter and the typical length, unlike in [5]. The slenderness ratio deviates with a factor of $\sqrt{\pi}/2 \approx 0.8862$:

$$\varepsilon = \frac{\sqrt{\sigma_{V\star}}}{r_\star} = \frac{\sqrt{\pi}}{2} \frac{D}{L}.$$

Remark 2.18 (Constant viscosity). *Assume that the dynamic viscosity is a constant in the initial values (meaning it is constant in space). Then its evolution equation ensures that it is also constant in time and we can omit its equation and simply use it as a global constant. Furthermore the typical value μ_\star can be chosen such that the dimensionless viscosity is $\mu = 1$.*

2.2.3 Parameterization of the rotational group

The rotation \mathbf{R} introduced in Remark 2.3 allows the transformation between the outer and director basis. We depict a rotation in \mathbb{R}^3 with the help of unit quaternions, other choices would be Euler angles or rotation vectors. We use the calculus for quaternions that is presented in detail in the same context as we have here in [4], also in a summarized way in [5].

A quaternion is an element in \mathbb{R}^4 and defined as $\mathbf{q} = (q_0, \mathbf{q}_v) = (q_0, q_1, q_2, q_3)$ with scalar part q_0 and vector part \mathbf{q}_v . The norm is $\|\mathbf{q}\| = q_0^2 + q_1^2 + q_2^2 + q_3^2$. For a unit quaternion it holds that $\|\mathbf{q}\| = 1$ and the rotation can then be depicted as

$$\mathbf{R}(\mathbf{q}) = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_0^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & -q_1^2 + q_2^2 - q_3^2 + q_0^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & -q_1^2 - q_2^2 + q_3^2 + q_0^2 \end{pmatrix}.$$

The time and space derivative of \mathbf{R}

$$\partial_t \mathbf{R}(\mathbf{q}) = -(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \times \mathbf{R}(\mathbf{q}), \quad (2.28a)$$

$$\partial_s \mathbf{R}(\mathbf{q}) = -\boldsymbol{\kappa} \times \mathbf{R}(\mathbf{q}), \quad (2.28b)$$

can be equivalently expressed as evolution equations (time and space) of a unit quaternion

$$\partial_t \mathbf{q} = \mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q}, \quad (2.29a)$$

$$\partial_s \mathbf{q} = \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q}, \quad (2.29b)$$

with the skew-symmetric matrix \mathcal{A} defined as

$$\mathcal{A}(\mathbf{z}) = \frac{1}{2} \begin{pmatrix} 0 & z_1 & z_2 & z_3 \\ -z_1 & 0 & z_3 & -z_2 \\ -z_2 & -z_3 & 0 & z_1 \\ -z_3 & z_2 & -z_1 & 0 \end{pmatrix}$$

for all $\mathbf{z} \in \mathbb{R}^3$. The equivalence only holds if \mathbf{q} is a unit quaternion in its initial values for the time evolution and in its boundary values for the space evolution. The ODEs in (2.29) then ensure analytically that \mathbf{q} stays a unit quaternion since \mathcal{A} is skew-symmetric (e.g. [34]). Numerically this can be ensured with an isometric method, a non-isometric method does not maintain the norm. One way to overcome this issue is to enforce the unit quaternion by adding the constraint

$$\|\mathbf{q}\| = 1 \tag{2.30a}$$

to the system with appropriate penalty quantities λ_t and λ_s in (2.29)

$$\partial_t \mathbf{q} = \mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_t \mathbf{q}, \tag{2.30b}$$

$$\partial_s \mathbf{q} = \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_s \mathbf{q}, \tag{2.30c}$$

with reference values $\lambda_{t,0} = v_\star/r_\star$, $\lambda_{s,0} = 1/r_\star$ for the dimensionless system. This also addresses another issue: The system (2.29) for the kinematics is overdetermined (eight equations for seven unknowns). It is still consistent because the unit quaternion constraint is inherently enforced by both the temporal and spatial evolution of the quaternion, thus creating two dependent equations. The adjusted equations (2.30) are balanced, there are nine equations and nine unknowns (\mathbf{q} , $\boldsymbol{\kappa}$, λ_t , λ_s).

Lemma 2.19 (Penalty quantities vanish). *Let some $\boldsymbol{\kappa}, \boldsymbol{\omega}, u$ and initial and boundary conditions for \mathbf{q} be given. If the initial and boundary conditions are unit quaternions, then (2.29) and (2.30) are equivalent.*

Proof. Constraint (2.30a) is rewritten as $\mathbf{q} \cdot \mathbf{q} = 1$ and its total time and space derivative imply

$$\mathbf{q} \cdot \partial_t \mathbf{q} = 0, \quad \mathbf{q} \cdot \partial_s \mathbf{q} = 0.$$

Then we insert (2.30b) and (2.30c) to obtain

$$\mathbf{q} \cdot (\mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_t \mathbf{q}) = 0, \quad \mathbf{q} \cdot (\mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_s \mathbf{q}) = 0. \tag{2.31}$$

For all $\mathbf{q} \in \mathbb{R}^4$ and all $\mathbf{z} \in \mathbb{R}^3$ it holds that $\mathbf{q} \cdot \mathcal{A}(\mathbf{z}) \cdot \mathbf{q} = 0$ and thus (2.31) yields $\lambda_t \equiv \lambda_s \equiv 0$. \square

For the remaining part of this work we always use quaternions to parameterize the rotation matrix \mathbf{R} and use equations (2.30) for the kinematics of a jet in formulation M, T and S (cf. Remark 2.2). The compatibility conditions are not touched by this adjustment. We state the system for the sake of proper referencing.

System 2.20. System 2.16 is given and its rotation group is parameterized with unit quaternions, more precisely both evolution equations for the rotation matrix are replaced with

$$\begin{aligned}\partial_t \mathbf{q} &= \mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_t \mathbf{q}, \\ \partial_s \mathbf{q} &= \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_s \mathbf{q}, \\ \mathbf{q} \cdot \mathbf{q} &= 1.\end{aligned}$$

2.2.4 Index analysis and a stabilized reduced index formulation

The jet model (System 2.20) is a non-linear PDAE system and becomes a system of DAEs after a semi-discretization (in time or space). When thinking about an appropriate numerical scheme we have to take the so-called *index* of the DAEs into account. It reflects the grade of how disturbances affect the solution and has been taken as one of the major criteria to classify DAEs. In general, a direct discretization of higher index DAEs may suffer from significant numerical perturbations, e.g. instabilities and convergence order reductions [46, 47]. Various definitions of the index exist in literature, we focus on the differentiation index introduced by Gear in [42], which characterizes DAEs by converting it to ODEs (basically counting the total number of differentiations needed to obtain an ODE). For a PDAE – like the viscous jet model – we can think of two different indices: The index in time after a semi-discretization in space, and the index in space after a semi-discretization in time.

Remark 2.21. *We want to mention another index definition: The perturbation index introduced by Hairer, Lubich and Roche [46], which measures the sensitivity of the solution with respect to perturbations of the given problem, in particular in dependence of derivatives of the perturbations. It is important for convergence analysis and a much stronger indicator for the possibly occurring problems, but in general quite difficult to determine. The differentiation index is equal or less than the perturbation index.*

The index concept was introduced as a rule of thumb for the expected numerical difficulties of a DAE, thus it is beneficial to not only choose a proper discretization scheme but to analyze the DAE in terms of possible equivalent transformations, often called stabilization or regularization techniques. A classical index-reduction is obtained by replacing the constraint with its derivative, also called the *hidden constraint*. The downside is that this introduces the *drift-off* effect which distorts the solution over time. One approach to stabilize the system was introduced by Gear, Gupta and Leimkuhler in [43]. It was originally introduced for mechanical systems with algebraic constraints. If the solution leaves the constraint manifold due to some arbitrary reason, a projection correction is added to the time evolution equations that pulls the solution back to the constraint manifold. In that process uniqueness of the solution has to be ensured which is generally possible due to specific assumptions and the construction of the so-called Gear-Gupta-Leimkuhler (GGL) correction, a more detailed description is given in Appendix B.1. The GGL idea can be transferred to PDAE systems by introducing discrete operators representing a semi-discretization. If a temporal correction is sought, we replace the spatial derivatives with

| temporal index | | | spatial index | | |
|----------------|---|---|---------------|---|---|
| M | T | S | M | T | S |
| 3 | 2 | 2 | 1 | 1 | 1 |

Table 2.2: Overview of the temporal and spatial index of different formulations (cf. Remark 2.2) of the jet model (cf. System 2.20).

said operator obtaining a DAE. After applying the stabilization we re-interpret the discrete operator analytically to obtain the stabilized PDAE model.

The jet model formulated with its original kinematics yields index three in time and one in space (an overview of the indices of the different kinematics formulation introduced in Remark 2.2 is given in Table 2.2 for completeness). Theoretical results for general index three system can only be given with higher order integration schemes [46]. If the index 3 problem is linear in its algebraic variables – which is the case for the jet model – then linear convergence order for the implicit Euler with constant stepsize is achieved [71, 46]. For reasons that will be clearer later we seek to reduce the index in time to two and use stabilization techniques to avoid drift-off effects. Inspired by the index-reduction and stabilization techniques of [43] we propose a general projection correction for the kinematics.

Formulation 2.22 (Stabilized kinematics). The index-reduced, stabilized formulation of the kinematics consists of the original kinematics and the linearly dependent compatibility condition and the unit quaternion constraint. To avoid an overdetermined system we incorporate projection corrections by introducing respective type-0 Lagrange multipliers $\lambda_s \in \mathbb{R}$, $\bar{\lambda}_1 \in \mathbb{R}^3$ and $\lambda_2 \in \mathbb{R}^4$. Whereas λ_s is required for the spatial evolution of the quaternion, $\bar{\lambda}_1$ and λ_2 are used in the functionals $\bar{\Lambda}_{r,1}(\cdot)$, $\Lambda_{r,2}(\cdot)$, $\Lambda_{q,2}(\cdot) \in \mathbb{R}^3$, $\Lambda_{q,1}(\cdot) \in \mathbb{R}^4$ in the kinematics. They can also be dependent on other unknowns of the solution and will be filled subsequently. The sole assumption at this point is that $\bar{\Lambda}_{r,1}$, $\Lambda_{r,2}$ vanish when $\bar{\lambda}_1$ vanishes and $\Lambda_{q,1}$, $\Lambda_{q,2}$ vanish when λ_2 vanishes. Overall the kinematics in material parameterization consist of the unknowns $\bar{\mathbf{r}}$, $\boldsymbol{\tau}$, \mathbf{q} , $\boldsymbol{\kappa}$, λ_s , $\bar{\lambda}_1$, λ_2 – together 21 degrees of freedom – and 21 equations

$$\partial_t \bar{\mathbf{r}} = \mathbf{R}(\mathbf{q})^T \cdot \mathbf{v} + \bar{\Lambda}_{r,1}, \quad \partial_t \mathbf{q} = \mathcal{A}(\boldsymbol{\omega}) \cdot \mathbf{q} + \Lambda_{q,1}, \quad (2.32a)$$

$$\mathbf{0} = \partial_s \bar{\mathbf{r}} - \mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\tau}, \quad \mathbf{0} = \partial_s \mathbf{q} - \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} - \lambda_s \mathbf{q}, \quad (2.32b)$$

$$\partial_t \boldsymbol{\tau} = \partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega} + \Lambda_{r,2}, \quad \partial_t \boldsymbol{\kappa} = \partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega} + \Lambda_{q,2}, \quad (2.32c)$$

$$0 = \mathbf{q} \cdot \mathbf{q} - 1. \quad (2.32d)$$

Remark 2.23 (Minimal change for an index 2 system). *Considering only the temporal index of the system it would be sufficient to introduce the compatibility condition for the kinematics of the jet curve, i.e. the left column in (2.32), to obtain an index 2 system. The introduction of both compatibility conditions further reduce the index of some variables, yet the whole system remains index 2. It turns out that the presence of both compatibility conditions is beneficial for our numerics, see Remark 2.24.*

Remark 2.24 (Necessity of both compatibility conditions). *One could argue that, to avoid drift-off effects, the original high-index formulation should be used. Like expected, simulations with our discrete approach and time-independent spatial domains have linear convergence, but for time-dependent simulations this yields highly unstable behavior of the inner forces of the jet. The investigation of the source of this effect is quite cumbersome, yet one symptom could be found: The discrete compatibility conditions are violated (very marginally though) in the proximity of the time-dependent side of the domain, which corrupts the DAE consistency of the initial values. Our proposed kinematics formulation includes the original spatial constraints and the compatibility conditions and can overcome said instability while avoiding the drift-off.*

The correction terms in Formulation 2.22 are yet to be defined. We want to show for all possible correction terms that a solution with the original kinematics is equivalent to a solution with the stabilized kinematics. Assume that kinematics in Formulation 2.22 are given. We show now an inherent relation that directly results from them. The Lagrangian multiplier λ_s can be shown to be zero with the help of the unit quaternion constraint analogously to Lemma 2.19. The spatial constraints (2.32b) are then stated with the help of the functionals

$$\begin{aligned} \mathbf{g}_r(\bar{\mathbf{r}}, \boldsymbol{\tau}, \mathbf{q}) &:= \partial_s \bar{\mathbf{r}} - \mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\tau} = \mathbf{0}, \\ \mathbf{g}_q(\mathbf{q}, \boldsymbol{\omega}) &:= \partial_s \mathbf{q} - \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} = \mathbf{0} \end{aligned}$$

and their Jacobian

$$\begin{aligned} D\mathbf{g}_r &= (D_{\bar{\mathbf{r}}}\mathbf{g}_r, D_{\boldsymbol{\tau}}\mathbf{g}_r, D_{\mathbf{q}}\mathbf{g}_r) = (\partial_s, -\mathbf{R}(\mathbf{q})^T, -D_{\mathbf{q}}(\mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\tau})), \\ D\mathbf{g}_q &= (D_{\mathbf{q}}\mathbf{g}_q, D_{\boldsymbol{\kappa}}\mathbf{g}_q) = (\partial_s - \mathcal{A}(\boldsymbol{\kappa}), -\mathcal{E}(\mathbf{q})) \end{aligned}$$

with a matrix \mathcal{E} such that $\mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} = \mathcal{E}(\mathbf{q}) \cdot \boldsymbol{\kappa}$ holds (such a matrix exists because \mathcal{A} is linear). We take the total time derivative of \mathbf{g}_r and \mathbf{g}_q and insert (2.32a)-(2.32c)

$$\begin{aligned} \mathbf{0} &\stackrel{!}{=} \frac{d}{dt} \mathbf{g}_r(\bar{\mathbf{r}}, \boldsymbol{\tau}, \mathbf{q}) = D\mathbf{g}_r \cdot \begin{pmatrix} \partial_t \bar{\mathbf{r}} \\ \partial_t \boldsymbol{\tau} \\ \partial_t \mathbf{q} \end{pmatrix} = D\mathbf{g}_r \cdot \underbrace{\begin{pmatrix} \mathbf{R}(\mathbf{q})^T \cdot \mathbf{v} \\ \partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega} \\ \partial_t \mathbf{q} \end{pmatrix}}_{=: \mathbf{A} = \mathbf{0}, \text{ see Remark 2.25}} + D\mathbf{g}_r \cdot \begin{pmatrix} \bar{\boldsymbol{\Lambda}}_{r,1} \\ \boldsymbol{\Lambda}_{r,2} \\ \mathbf{0} \end{pmatrix} \\ &= \partial_s \bar{\boldsymbol{\Lambda}}_{r,1} - \mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\Lambda}_{r,2}, \end{aligned} \tag{2.33a}$$

$$\begin{aligned} \mathbf{0} &\stackrel{!}{=} \frac{d}{dt} \mathbf{g}_q(\mathbf{q}, \boldsymbol{\omega}) = D\mathbf{g}_q \cdot \begin{pmatrix} \partial_t \mathbf{q} \\ \partial_t \boldsymbol{\kappa} \end{pmatrix} = D\mathbf{g}_q \cdot \underbrace{\begin{pmatrix} \mathcal{A}(\boldsymbol{\omega}) \cdot \mathbf{q} \\ \partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega} \end{pmatrix}}_{=: \mathbf{B} = \mathbf{0}, \text{ see Remark 2.25}} + D\mathbf{g}_q \cdot \begin{pmatrix} \boldsymbol{\Lambda}_{q,1} \\ \boldsymbol{\Lambda}_{q,2} \end{pmatrix} \\ &= \partial_s \boldsymbol{\Lambda}_{q,1} - \mathcal{A}(\boldsymbol{\kappa}) \cdot \boldsymbol{\Lambda}_{q,1} - \mathcal{A}(\boldsymbol{\Lambda}_{q,2}) \cdot \mathbf{q}. \end{aligned} \tag{2.33b}$$

Remark 2.25. *The term \mathbf{A} in (2.33a) vanishes because it is merely recreating the compatibility conditions, more precisely after the expansion of \mathbf{A} we use Remark 2.3 and reverse the chain and product rule for the time derivative of $\bar{\boldsymbol{\tau}} = \mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\tau}$, i.e.*

$$\begin{aligned} \mathbf{A} &= \partial_s (\mathbf{R}(\mathbf{q})^T \cdot \mathbf{v}) - \mathbf{R}(\mathbf{q})^T \cdot (\partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega}) - D_{\mathbf{q}}(\mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\tau}) \cdot \partial_t \mathbf{q} \\ &= \partial_s \bar{\mathbf{v}} - \mathbf{R}(\mathbf{q})^T \cdot (\partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v}) - (\mathbf{R}(\mathbf{q})^T \cdot (\boldsymbol{\tau} \times \boldsymbol{\omega}) + \partial_t (\mathbf{R}(\mathbf{q})^T) \cdot \boldsymbol{\tau}) \\ &= \partial_t \bar{\boldsymbol{\tau}} - \mathbf{R}(\mathbf{q})^T \cdot \partial_t \boldsymbol{\tau} - \partial_t (\mathbf{R}(\mathbf{q})^T) \cdot \boldsymbol{\tau} = \mathbf{0}. \end{aligned}$$

The term B in (2.33b) also vanishes due to the special structure of \mathcal{A} , namely $2\mathcal{A}(\mathbf{x}) \cdot \mathcal{A}(\mathbf{y}) = \mathcal{A}(\mathbf{y} \times \mathbf{x})$ for all $\mathbf{y}, \mathbf{x} \in \mathbb{R}^3$. The expansion of B gives

$$\begin{aligned} B &= \partial_s (\mathcal{A}(\boldsymbol{\omega}) \cdot \mathbf{q}) - \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathcal{A}(\boldsymbol{\omega}) \cdot \mathbf{q} - \mathcal{E}(\mathbf{q}) \cdot (\partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega}) \\ &= \mathcal{A}(\partial_s \boldsymbol{\omega}) \cdot \mathbf{q} + \mathcal{A}(\boldsymbol{\kappa} \times \boldsymbol{\omega}) \cdot \mathbf{q} - \mathcal{E}(\mathbf{q}) \cdot (\partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega}) = \mathbf{0}. \end{aligned}$$

Both equations (2.33) are inherently enforced if the stabilized formulation 2.22 of the kinematics is used. We want to use that to show that the correction terms themselves vanish and formulate beforehand a general lemma for the projection correction.

Lemma 2.26 (Equivalence of the stabilized kinematics). *Assume at some time t a solution of System 2.20_M (A), and a solution of System 2.20_P (B) with proper initial and boundary conditions are given. The subscript P – for projection – denotes System 2.20 with the stabilized formulation 2.22 for the kinematics. Assume furthermore that the correction terms and initial and boundary conditions are chosen such that for any t the inherently enforced equations (2.33) have the unique solution that the multipliers $\bar{\boldsymbol{\lambda}}_1$ and $\boldsymbol{\lambda}_2$ vanish. Then the solution of (A) is equivalent to the solution of (B).*

Proof. We restrict the scope to the variables $\bar{\mathbf{r}}, \boldsymbol{\tau}, \mathbf{q}, \boldsymbol{\kappa}$ and $\lambda_s, \bar{\boldsymbol{\lambda}}_1, \boldsymbol{\lambda}_2$ and their equations, the remaining variables of the solution are untouched. At first, consider a solution of (A). We expand the solution by adding the corresponding multipliers with value zero. Due to the assumption in Formulation 2.22 the correction terms are also zero. Then we also have a solution of (B) since the additional equations (2.32c) reduce to the compatibility conditions. Second, assume a solution of (B). The system inherently requires (2.33) which forces the multipliers to vanish due to the assumption. In consequence of Formulation 2.22 the correction terms vanish as well and (2.32c) reduce to the compatibility conditions. Then the solution of (B) is a solution of (A) with an added linear dependent equation. \square

After this preparatory work we now get to the actual choice of the correction terms. The original GGL correction transferred to our PDAE model (System 2.20_M) leads to correction terms only for the temporal evolution of the kinematics, i.e. $\bar{\boldsymbol{\Lambda}}_{r,2}$ and $\boldsymbol{\Lambda}_{q,2}$ vanish (the GGL correction terms are presented in Appendix B.1). We extend the projection ideas of the GGL correction and propose a correction term for the compatibility conditions as well. They are formulated in the material parameterization, but to keep the correction terms transformation invariant we introduce a fictitious type-1 scalar k that fulfills $k \equiv 1$.

Formulation 2.27 (Schiessl-Arne-Marheineke-Wegener (SAMW) correction).

Assume Formulation 2.22 for the kinematics. The correction terms are given by

$$\begin{aligned} \bar{\boldsymbol{\Lambda}}_{r,1}^{\text{SAMW}} &:= \frac{1}{k} \partial_s \bar{\boldsymbol{\Lambda}}_{\tau}, & \boldsymbol{\Lambda}_{q,1}^{\text{SAMW}} &:= \frac{1}{k} \mathcal{A}(\partial_s \boldsymbol{\lambda}_{\kappa}) \cdot \mathbf{q} + \lambda_t \mathbf{q}, \\ \boldsymbol{\Lambda}_{r,2}^{\text{SAMW}} &:= Ck \mathbf{R}(\mathbf{q}) \cdot \bar{\boldsymbol{\Lambda}}_{\tau}, & \boldsymbol{\Lambda}_{q,2}^{\text{SAMW}} &:= Ck \boldsymbol{\lambda}_{\kappa} \end{aligned}$$

with $\bar{\boldsymbol{\Lambda}}_{\tau} := \bar{\boldsymbol{\lambda}}_1$, $(\boldsymbol{\lambda}_{\kappa}, \lambda_t) := \boldsymbol{\lambda}_2$ as the multipliers and some constant $C > 0$.

The correction terms are not obvious at first, but they have significant advantages on an analytical and numerical level compared to the GGL correction, or in that matter any correction that has $\boldsymbol{\Lambda}_{r,2} \equiv \mathbf{0}$, $\boldsymbol{\Lambda}_{q,2} \equiv \mathbf{0}$. This is highlighted in the following Remark 2.28 and the later Remark 4.13.

Remark 2.28 (Linear dependent equation in GGL correction). Assume that the GGL correction, or in that matter, any correction that uses $\mathbf{\Lambda}_{r,2} \equiv \mathbf{0}$, $\mathbf{\Lambda}_{q,2} \equiv \mathbf{0}$, is used for the stabilized kinematics 2.22. Observe that the constraints (2.32b) have the same dimension as the temporal evolution equations (2.32a) that contain the correction term. Consequently the additional multipliers have the same dimension as the corrected equations, thus said equations only determine the multipliers and are linearly dependent in the system. In consequence the corrected equations as well as the multipliers can be removed from the system giving the S formulation. Unfortunately, the S formulation is not applicable to our discrete scheme due to the staggered grid, cf. Remark 4.12.

As the last task it remains to show that the assumptions of Lemma 2.26 hold true.

Lemma 2.29 (Equivalence of the SAMW correction). Assume a solution of System 2.20 with properly given initial and boundary conditions and stabilized kinematics with the correction terms of Formulation 2.27. Furthermore we assume that for all $t \in [0, T]$ and $s', s'' \in \{-\mathcal{L}(t), -\mathcal{R}(t)\}$ we have

$$\bar{\lambda}_\tau(s', t) = \mathbf{0}, \quad \partial_s \bar{\lambda}_\tau(s'', t) = \mathbf{0}, \quad (2.34a)$$

for λ_τ and

$$\lambda_\kappa(s', t) = \mathbf{0}, \quad \partial_s \lambda_\kappa(s'', t) = \mathbf{0}, \quad (2.34b)$$

for λ_κ . Then the assumptions of Lemma 2.26 are fulfilled.

Proof. First we show analogously to Lemma 2.19 that $\lambda_t \equiv 0$ holds. Then the equations (2.33) become

$$\mathbf{0} = \partial_s \left(\frac{1}{k} \partial_s \bar{\lambda}_\tau \right) - Ck \bar{\lambda}_\tau, \quad (2.35a)$$

$$\begin{aligned} \mathbf{0} &= \mathcal{A} \left(\frac{d}{ds} \left(\frac{1}{k} \partial_s \lambda_\kappa \right) \right) \cdot \mathbf{q} + \mathcal{A} \left(\frac{1}{k} \partial_s \lambda_\kappa \right) \cdot \mathcal{A}(\kappa) \cdot \mathbf{q} \\ &\quad - \mathcal{A}(\kappa) \cdot \mathcal{A} \left(\frac{1}{k} \partial_s \lambda_\kappa \right) \cdot \mathbf{q} - \mathcal{A}(Ck \lambda_\kappa) \cdot \mathbf{q} \\ &= \mathcal{A} \left(\frac{1}{k} \partial_{ss} \lambda_\kappa - \frac{\partial_s k}{k^2} \partial_s \lambda_\kappa + \kappa \times \frac{1}{k} \partial_s \lambda_\kappa - Ck \lambda_\kappa \right) \cdot \mathbf{q}. \end{aligned} \quad (2.35b)$$

Since \mathbf{q} is a unit quaternion, (2.35b) is only fulfilled if

$$\begin{aligned} \mathbf{0} &= \partial_{ss} \lambda_\kappa - \frac{\partial_s k}{k} \partial_s \lambda_\kappa + \kappa \times \partial_s \lambda_\kappa - Ck^2 \lambda_\kappa \\ &= \partial_{ss} \lambda_\kappa + \left(\mathcal{K} - \frac{\partial_s k}{k} \mathbf{1} \right) \cdot \partial_s \lambda_\kappa - Ck^2 \lambda_\kappa \quad \text{with } \mathcal{K} = \begin{pmatrix} 0 & -\kappa_3 & \kappa_2 \\ \kappa_3 & 0 & -\kappa_1 \\ -\kappa_2 & \kappa_1 & 0 \end{pmatrix}, \end{aligned} \quad (2.36)$$

which is a system of homogeneous, differential equations of second order with variable coefficients and compact domain for every t . We reformulate (2.36) together with (2.34b) as

a first-order system and initial value problem. This is possible, since the initial assumption is that a solution exists and is given and either we already have $s' = s''$, or we rewrite the system with a substitution that puts the Dirichlet condition on the same side as the Neumann condition, i.e. we introduce λ'_κ with $\lambda'_\kappa(s, t) = \lambda_\kappa(s, t) - \lambda_\kappa(s'', t)$, where s'' is the position of the Neumann condition. Assuming sufficient regularity, we can apply common uniqueness results (we refer e.g. to [50]) and derive that $\lambda_\kappa \equiv \mathbf{0}$ must be the unique solution. Equation (2.35a) is expanded to

$$\mathbf{0} = \partial_{ss}\bar{\lambda}_\tau - \frac{\partial_s k}{k}\partial_s\bar{\lambda}_\tau - Ck^2\bar{\lambda}_\tau.$$

Analogously to λ_κ we can show with (2.34a) that the unique solution is $\bar{\lambda}_\tau \equiv 0$. The assumptions of Lemma 2.26 are fulfilled. \square

We only introduce one set of correction terms, yet there might be other possible choices for the them that have numerical advantages, one aspect that motivated the SAMW correction was maintaining the spatial constraint. In this work we always adress the SAMW correction if we talk about *stabilized kinematics*. Furthermore there exist other fundamentally different approaches how a stabilized, index-reduced formulation of the kinematics can be obtained. We want to mention selected ones in the following remark.

Remark 2.30 (Alternative methods for a stabilized formulation).

In literature there exist different approaches for the stabilization of higher index DAE systems. We want to mention some of them.

- *Baumgarte stabilization [14]: In the Baumgarte stabilization for index 3 system, the constraint and both hidden constraints are linearly combined to an parametric and asymptotically stable differential equation. The choice in the parameter influence the quality of the solution and has to be fitted to the model.*
- *Regularization involving the concepts around the strangeness index [69], for example, the projected-strangeness-free form for quasi-linear DAEs [95]: A strangeness-free and index-reduced DAE is obtained through algebraic transformations which are suitable for the numerical integration with stiffly accurate methods.*
- *Projected implicit Runge-Kutta method (PIRK): Stiffly-accurate Runge-Kutta methods have convergence properties for DAEs, nevertheless they are not isometric which e.g. enforces special handling of the unit quaternion in the jet model (cf. Section 2.2.3). Symmetric discretizations that would maintain the unit quaternion may suffer from instability, oscillation and loss of accuracy. Ascher and Petzold introduce the PIRK methods to overcome those difficulties and do so by including a projection correction step within the numerical formula. Their method shows interesting results regarding the drift-off when applied index 2 systems that were obtained from a index 3 system (similar to our System 2.20_T), more precisely the unstabilized index 2 formulation of the mathematical pendulum shows no drift-off (up to machine precision) due to the quadratic constraint. For details we refer to [11].*

2.2.5 Viscous jet model

The parts presented in the last subsections are now put together. We use the kinematics in the stabilized formulation together with the incompressible geometry model and the viscous material law and present the jet model in dimensionless form put in the general parameterization.

System 2.31. The system of the jet model for the unknowns $(\bar{\mathbf{r}}, \bar{\boldsymbol{\lambda}}_\tau, \mathbf{q}, \boldsymbol{\lambda}_\kappa, \lambda_s, \boldsymbol{\tau}, \boldsymbol{\kappa}, \lambda_t, \sigma_M, \sigma_V, k, \mathbf{v}, \boldsymbol{\omega}, \mathbf{n}, \mathbf{m}, u)$ on the domain \mathcal{Q} (cf. Notation 2.11) is given by

Viscous jet model

$$\partial_t \bar{\mathbf{r}} = \mathbf{R}(\mathbf{q})^T \cdot (\mathbf{v} - u\boldsymbol{\tau}) + \frac{1}{k} \partial_s \bar{\boldsymbol{\lambda}}_\tau, \quad (2.37a)$$

$$\partial_s \bar{\mathbf{r}} = \mathbf{R}(\mathbf{q})^T \cdot \boldsymbol{\tau}, \quad (2.37b)$$

$$\partial_t \mathbf{q} = \mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q} + \frac{1}{k} \mathcal{A}(\partial_s \boldsymbol{\lambda}_\kappa) \cdot \mathbf{q} + \lambda_t \mathbf{q}, \quad (2.37c)$$

$$\partial_s \mathbf{q} = \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_s \mathbf{q}, \quad (2.37d)$$

$$\partial_t \boldsymbol{\tau} + \partial_s(u\boldsymbol{\tau}) = \partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega} + Ck\mathbf{R}(\mathbf{q}) \cdot \bar{\boldsymbol{\lambda}}_\tau, \quad (2.37e)$$

$$\partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa}) = \partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega} + Ck\boldsymbol{\lambda}_\kappa, \quad (2.37f)$$

$$\mathbf{q} \cdot \mathbf{q} = 1, \quad (2.37g)$$

$$\partial_t \sigma_M + \partial_s(u\sigma_M) = 0, \quad \partial_t \sigma_V + \partial_s(u\sigma_V) = 0, \quad \partial_t k + \partial_s(uk) = 0, \quad (2.37h)$$

$$\partial_t \mathbf{p} + \partial_s(u\mathbf{p}) = \mathbf{p} \times \boldsymbol{\omega} + \partial_s \mathbf{n} + \boldsymbol{\kappa} \times \mathbf{n} + \mathbf{f}, \quad \mathbf{p} = \sigma_M \mathbf{v}, \quad (2.37i)$$

$$\varepsilon^2(\partial_t \mathbf{h} + \partial_s(u\mathbf{h})) = \varepsilon^2 \mathbf{h} \times \boldsymbol{\omega} + \partial_s \mathbf{m} + \boldsymbol{\kappa} \times \mathbf{m} + \boldsymbol{\tau} \times \mathbf{n} + \varepsilon^2 \mathbf{l}, \quad \mathbf{h} = \mathbf{J}_M \cdot \boldsymbol{\omega}, \quad (2.37j)$$

$$\partial_t \tau_3 + \partial_s(u\tau_3) = \frac{\text{Re } \tau_3^2}{3\mu \sigma_V} \mathbf{n}_3, \quad \tau_1 = 0, \quad \tau_2 = 0, \quad (2.37k)$$

$$\varepsilon^2(\partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa})) = \frac{\text{Re } \tau_3^3}{3\mu \sigma_V^2} \mathbf{M}_\mu^{-1} \cdot \mathbf{m}. \quad (2.37l)$$

Here, the inertia matrix is $\mathbf{J}_M = (\sigma_M \sigma_V / \tau_3) \mathbf{M}_{in}$ and the parameter speed u is still a degree of freedom in the system, but assumed to be globally zero if not otherwise mentioned.

Notation 2.32 (Formulation of the kinematics). Other formulations of the kinematics of viscous jet model are possible as well and could be addressed analogously to Remark 2.2. For the sake of clarity the following discussions are tailored to the formulation which is printed in System 2.31 – it contains the stabilized, index-reduced kinematics with SAMW-correction. Only if we seek to address different formulations we add the respective subscript.

System 2.31 is not yet complete. We need to define initial and boundary conditions on an appropriate space-time domain, which we do in the following Section 2.3.

Remark 2.33 (Comparison to Audoly et al. (2013)). *The modified Kirchhoff constraint relates the tangent of the jet to the director triad which allows to express the angular velocity with the help of the scalar-valued spin (tangential angular speed). The angular*

momentum balance becomes scalar-valued and the director triad can be computed a posteriori, making its temporal evolution redundant. Audoly et al. use this center-line-spin formulation (utilizing the compatibility conditions in the kinematics) and propose a discrete viscous rods method [19, 12] that was based on their experience with the discrete elastic rod method [20] and focuses on geometrically exact space discretization for the rod model. They performed instationary simulations for a viscous jet lay-down [25] and also discussed the effect of inertia in the angular momentum balance and argued that it can be neglected [24].

2.3 Boundaries and space-time domains for set-ups

We use the wording *set-up* in combination with the jet model when we specify initial and boundary conditions and give a framework for the determination of appropriate space-time domains, that both together complete the jet model. Initial conditions will be set according to the application at hand, for the boundary conditions we consider two different physical boundaries that can be put on either side of the domain: One gives the position and orientation of the jet (position boundary), the other the inner tensions (tension boundary).

At first we describe the physical boundaries. We present all conditions in detail to show where the dependencies are. Note that the following discussion is not meant to give a list of complete boundary conditions for the jet model. It is merely a list of relations that hold for the considered boundary. The discussion of the resulting initial and boundary conditions is done afterwards.

Position boundary At a position boundary all that is needed to formulate the boundary conditions for our jet model is a prescribed position, orientation and material velocity. The remaining required unknowns are given by inherent relations, choice of parameterization and corresponding typical values. By means of the material velocity we can differentiate the physical meaning of the boundary. If it is zero, the jet is assumed to be fixated at a wall, whereas a positive / negative value indicates a inflow / outflow at the left side and vice-versa on the right side. Inflow indicates that the jet is extruded through a nozzle and outflow that the jet is collected in some way. The following part is formulated exemplarily for the left side of the domain, thus we mark the given values with the superscript \cdot^L . Notice that fields in Lagrangian parameterization are marked with a hat $\hat{\cdot}$, whereas ones in general parameterization receive no special mark. The location of the left side of the boundary is $\sigma = -l(t)$, and $s = -\mathcal{L}(t)$.

Assume a given position $\bar{\mathbf{r}}^L: [0, T] \rightarrow \mathbb{R}^3$ and orientation $\mathbf{q}^L: [0, T] \rightarrow \mathbb{R}^4$, i.e.

$$\bar{\mathbf{r}}(-\mathcal{L}(t), t) = \bar{\mathbf{r}}^L(t), \quad (2.38a)$$

$$\mathbf{q}(-\mathcal{L}(t), t) = \mathbf{q}^L(t), \quad (2.38b)$$

and material velocity $\frac{d}{dt}l(t) = v^L(t)$ for all $t \in [0, T]$. In this work we always call $v^L(t)$ the mass inflow, or extrusion speed. The jet model contains substitutions for the temporal derivatives of $\bar{\mathbf{r}}$ and \mathbf{q} which require consistent boundary conditions enforced by inherent relations. We take the total time derivative of the jet curve at the boundary in material parameterization and insert the original kinematics of the jet to obtain a relation between

the jet velocity, dilatation and material velocity

$$\mathbf{v}(-\mathcal{L}(t), t) = \mathbf{R}(\mathbf{q}^L(t)) \frac{d}{dt} \bar{\mathbf{r}}^L(t) + \hat{\boldsymbol{\tau}}(-l(t), t) v^L(t) \quad (2.38c)$$

with the type-0 field $\mathbf{v}(-\mathcal{L}(t), t) = \hat{\mathbf{v}}(-l(t), t)$. Analogously an inherent condition can be derived from the jet orientation, i.e.

$$\mathcal{A}(\hat{\boldsymbol{\omega}}(-l(t), t)) \cdot \mathbf{q}^L(t) - \mathcal{A}(\hat{\boldsymbol{\kappa}}(-l(t), t)) \cdot \mathbf{q}^L(t) v^L(t) = \frac{d}{dt} \mathbf{q}^L(t).$$

The time-dependent \mathbf{q}^L induces an angular velocity $\boldsymbol{\omega}^l$ through $\frac{d}{dt} \mathbf{q}^L(t) = \mathcal{A}(\boldsymbol{\omega}^L(t)) \cdot \mathbf{q}^L(t)$, for details see Remark 2.35. Since \mathcal{A} is linear the terms can be merged and the equation is only fulfilled if the argument becomes zero, thus

$$\boldsymbol{\omega}(-\mathcal{L}(t), t) = \boldsymbol{\omega}^L(t) + \hat{\boldsymbol{\kappa}}(-l(t), t) v^L(t) \quad (2.38d)$$

has to hold, which is the inherent condition for the type-0 field $\hat{\boldsymbol{\omega}}(-l(t), t) = \boldsymbol{\omega}(-\mathcal{L}(t), t)$. We assume further knowledge of the distortion measures $\hat{\boldsymbol{\tau}}^L, \hat{\boldsymbol{\kappa}}^L: [0, T] \rightarrow \mathbb{R}^3$, i.e. we have

$$\boldsymbol{\tau}(-\mathcal{L}(t), t) = \chi(-l(t), t) \hat{\boldsymbol{\tau}}(-l(t), t) = \chi(-l(t), t) \hat{\boldsymbol{\tau}}^L(t), \quad (2.38e)$$

$$\boldsymbol{\kappa}(-\mathcal{L}(t), t) = \chi(-l(t), t) \hat{\boldsymbol{\kappa}}(-l(t), t) = \chi(-l(t), t) \hat{\boldsymbol{\kappa}}^L(t). \quad (2.38f)$$

In the set-ups that we are considering we assume unstretched and unbent jets, i.e. $\hat{\boldsymbol{\tau}}^L(t) = \mathbf{e}_3$ and $\hat{\boldsymbol{\kappa}}^L(t) = \mathbf{0}$. The parameterization is left open at this point.

Remark 2.34 (Application to the stabilized kinematics). Assume now that a position boundary fulfilling (2.38) is given and used with the stabilized formulation of the jet model (cf. Formulation 2.22). Its kinematics induce inherent relations for \mathbf{v} and $\boldsymbol{\omega}$ – analogously to (2.38c) and (2.38d) – that involve the correction terms and therewith the Lagrangian multipliers. More precisely we obtain

$$\begin{aligned} \hat{\mathbf{v}}(-l(t), t) - \hat{\boldsymbol{\tau}}^L(t) v^L(t) &= \mathbf{R}(\mathbf{q}^L(t)) \left(\frac{d}{dt} \bar{\mathbf{r}}^L(t) - \hat{\boldsymbol{\Lambda}}_{r,1}(-l(t), t) \right), \\ \mathcal{A}(\hat{\boldsymbol{\omega}}(-l(t), t) - \hat{\boldsymbol{\kappa}}^L(t) v^L(t) - \boldsymbol{\omega}^L(t)) \cdot \mathbf{q}^L(t) &= \lambda_s(-l(t), t) \mathbf{q}^L(t) v^L(t) - \hat{\boldsymbol{\Lambda}}_{q,1}(-l(t), t). \end{aligned}$$

We plug in (2.38) and conclude that

$$\begin{aligned} \hat{\boldsymbol{\Lambda}}_{r,1}(-l(t), t) &= \mathbf{0}, \\ \lambda_s(-l(t), t) \mathbf{q}^L(t) v^L(t) - \hat{\boldsymbol{\Lambda}}_{q,1}(-l(t), t) &= \mathbf{0} \end{aligned}$$

have to hold. With the fact that λ_s can be shown to be globally zero (cf. Lemma 2.19), we know that the correction terms vanish at a position boundary.

Remark 2.35 (Induced angular velocity). The rotation matrix is parameterized with unit quaternions. Its temporal evolution is given in terms of the quaternions by

$$\partial_t \mathbf{q} = \mathcal{A}(\boldsymbol{\omega}) \cdot \mathbf{q}. \quad (2.39)$$

It can be depicted with the quaternion multiplication which is defined as follows: Let \mathbf{q} and \mathbf{p} be two arbitrary quaternions, then we have

$$\mathbf{q} \star \mathbf{p} = (q_0 p_0 - \mathbf{q}_v \cdot \mathbf{p}_v, q_0 \mathbf{p}_v + p_0 \mathbf{q}_v + \mathbf{q}_v \times \mathbf{p}_v).$$

We rewrite (2.39) to

$$\partial_t \mathbf{q} = -\frac{1}{2}(0, \boldsymbol{\omega}) \star \mathbf{q}$$

and multiply it with the inverse $\mathbf{q}^{-1} = (q_0, -\mathbf{q}_v) / \|\mathbf{q}\|$, which is always defined for unit quaternions. It follows that

$$\boldsymbol{\omega} = 2(\mathbf{q}_v \partial_t q_0 - q_0 \partial_t \mathbf{q}_v - \mathbf{q}_v \times \partial_t \mathbf{q}_v). \quad (2.40)$$

With the quaternion $\mathbf{q}(-\mathcal{L}(t), t) = \mathbf{q}^L(t)$ given at the boundary for all t we can herewith obtain $\boldsymbol{\omega}^l(t)$. Note that $\boldsymbol{\omega}^L(t)$ is induced by the total derivative of \mathbf{q} whereas $\boldsymbol{\omega}(-\mathcal{L}(t), t)$ in (2.40) is induced by the partial derivative.

Tension boundary A tension boundary is stipulated with prescribed inner tensions and possible mass outflow. The following part is formulated exemplarily for the right side of the domain, thus we mark the values with the superscript \cdot^R . The location of the right side of the boundary is $\sigma = -r(t)$, and $s = -\mathcal{R}(t)$.

Assume a given contact force $\mathbf{n}^R: [0, T] \rightarrow \mathbb{R}^3$ and couple $\mathbf{m}^R: [0, T] \rightarrow \mathbb{R}^3$ at the boundary

$$\mathbf{n}(-\mathcal{R}(t), t) = \mathbf{n}^R(t), \quad (2.41a)$$

$$\mathbf{m}(-\mathcal{R}(t), t) = \mathbf{m}^R(t), \quad (2.41b)$$

as well as a non-negative mass outflow velocity $\frac{d}{dt}r(t) = v^R(t) \geq 0$. The material law of the jet model gives a relation of the inner tensions to the distortion measures. We take the total derivative of τ_3 and $\boldsymbol{\kappa}$ at the boundary (in general parameterization) and plug in the material law to obtain inherent equations for them:

$$\begin{aligned} \frac{d}{dt}\tau_3(-\mathcal{R}(t), t) &= \frac{\text{Re } \tau_3^2(-\mathcal{R}(t), t)}{3\mu \sigma_V(-\mathcal{R}(t), t)} \mathbf{n}_3^R(t) \\ &\quad - \partial_s(u(s, t)\tau_3(s, t))\big|_{s=-\mathcal{R}(t)} - \partial_s\tau_3(s, t)\big|_{s=-\mathcal{R}(t)} \frac{d}{dt}\mathcal{R}(t), \end{aligned} \quad (2.41c)$$

$$\begin{aligned} \frac{d}{dt}\boldsymbol{\kappa}(-\mathcal{R}(t), t) &= \frac{\text{Re } \tau_3^3(-\mathcal{R}(t), t)}{\varepsilon^2 3\mu \sigma_V^2(-\mathcal{R}(t), t)} \mathbf{M}_\mu^{-1} \cdot \mathbf{m}^R(t) \\ &\quad - \partial_s(u(s, t)\boldsymbol{\kappa}(s, t))\big|_{s=-\mathcal{R}(t)} - \partial_s\boldsymbol{\kappa}(s, t)\big|_{s=-\mathcal{R}(t)} \frac{d}{dt}\mathcal{R}(t). \end{aligned} \quad (2.41d)$$

The shear measures τ_1, τ_2 can be dealt with the same way, but since they are globally zero their inherent conditions are trivial and always fulfilled in any case. The special case that \mathbf{n}^R and \mathbf{m}^R vanish (tension-free boundary) significantly simplifies (2.41c) and

(2.41d). Furthermore, if a time-independent boundary is given, i.e. $\frac{d}{dt}\mathcal{R}(t) = 0$, we obtain conservation for the prescribed values, i.e.,

$$\begin{aligned}\frac{d}{dt}\tau_3(-\mathcal{R}(t), t) + \partial_s(u(s, t)\tau_3(s, t))\big|_{s=-\mathcal{R}(t)} &= 0, \\ \frac{d}{dt}\kappa(-\mathcal{R}(t), t) + \partial_s(u(s, t)\kappa(s, t))\big|_{s=-\mathcal{R}(t)} &= 0.\end{aligned}$$

That means that the value at the boundary can be obtained by transforming the initial values:

$$\begin{aligned}\tau_3(-\mathcal{R}(t), t) &= \chi(-\mathcal{R}(t), t)\tau_3(-\mathcal{R}(0), 0), \\ \kappa(-\mathcal{R}(t), t) &= \chi(-\mathcal{R}(t), t)\kappa(-\mathcal{R}(0), 0).\end{aligned}$$

Remark 2.36 (Application to the stabilized kinematics). *Assume a tension boundary fulfilling (2.41) is given. There is no inherent relation when used with the stabilized kinematics 2.22 that affect the multipliers or correction terms. For the equivalency of the solutions of the original model and the stabilized model we require the multipliers to vanish at the boundaries and thus we set*

$$\begin{aligned}\bar{\lambda}_1(-\mathcal{R}(t), t) &= 0, \\ \lambda_2(-\mathcal{R}(t), t) &= 0.\end{aligned}$$

Combining the boundaries We have introduced two possible boundaries for our viscous jet model and construct now three set-ups by altering the possible combinations of the boundaries:

- mixed boundary conditions (Section 2.3.1) that can be generally used for jet extrusion processes with an out-stream element,
- both tension-free boundaries (Section 2.3.2) for a jet that moves freely in a surrounding medium,
- both position boundaries (Section 2.3.3) that could prescribe a jet clamped on both sides, or extruded on one side and rolled up on the other.

No claim for completeness is made here, of course other set-ups and other forms of boundaries are possible as well, but that is outside the scope of this work. Note that the two latter set-ups are introduced as additional examples, they are not used in the later numerical simulations.

In the following subsections we discuss the impact of the set-ups on the parameterization domain and state initial and boundary values. The conditions are presented in dimensionless quantities, thus we assume that the corresponding typical values are appropriately chosen (e.g. the mass line density at the nozzle of a position boundary is scaled to one). For the discussion of the set-ups it is not important how the general parameterization is determined within the domain, only the shape of the space-time domain is. The jet model

can be given in various kinematic formulations that differ in the usage of the compatibility conditions (see Remark 2.2). Depending on the formulation, the model may require different sets of initial and boundary conditions. Our discussion is tailored to the index-reduced, stabilized formulation with SAMW correction that is stated in Section 2.2.5, other formulations are analogously.

2.3.1 Position and tension-free boundary

In this set-up with mixed boundaries the jet is extruded through a nozzle on the left side of the domain with a given extrusion speed. The nozzle has a fixed position and orientation and the jet leaves the nozzle unstretched in the same direction. Hence the jet at the nozzle neither has curvature nor angular velocity. On the right side the jet is moving freely in a surrounding medium – in our case air – and the stresses vanish. Furthermore mass outflow is possible. If an initial jet exists, it has a given length and is chosen to be straight and stress-free everywhere, furthermore the initial mass and volume line density are assumed to be constant in space. The initial parameterization is chosen to coincide with the Lagrangian one and to be arc-length. This is called the jet extrusion set-up. Note that this includes the case with zero extrusion speed – meaning that the jet is simply fixated at a wall – but we do not differentiate in our wording.

Assume given functions $\bar{\mathbf{r}}^A, \mathbf{q}^A, v^A$ and initial length $l_0 = \mathcal{L}_0 = \int_{-\mathcal{L}_0}^0 \frac{d}{ds} \bar{\mathbf{r}}^A(s') ds' \geq 0$. Initial conditions for all $s \in [-\mathcal{L}_0, 0]$ are required for $\bar{\mathbf{r}}, \mathbf{q}, \boldsymbol{\tau}, \boldsymbol{\kappa}, \sigma_M, \sigma_V, k, \sigma_M \mathbf{v}, \mathbf{J}_M \cdot \boldsymbol{\omega}$. We set the initial conditions by using

$$\begin{aligned} \bar{\mathbf{r}}(s, 0) &= \bar{\mathbf{r}}^A(s), & \mathbf{q}(s, 0) &= \mathbf{q}^A(s), \\ \mathbf{v}(s, 0) &= \mathbf{e}_3 v^A(s), & \boldsymbol{\omega}(s, 0) &= \mathbf{0}, \\ \boldsymbol{\tau}(s, 0) &= \mathbf{e}_3 \chi(s, 0), & \boldsymbol{\kappa}(s, 0) &= \mathbf{0}, \\ \sigma_M(s, 0) &= \sigma_V(s, 0) = k(s, 0) = \chi(s, 0). \end{aligned} \tag{2.42}$$

The initial parameterization fulfills $\chi(s, 0) = 1$ since it is chosen to be arc-length. In the case of a time-dependent domain we need auxiliary conditions for the temporal fluxes at $t \in (0, T]$, $s = -\mathcal{L}(t)$ and set them analogously to $t = 0$, knowledge of χ is assumed. Note that $\boldsymbol{\kappa}$ is a type-1 field and would also need to be transformed with χ in a non-vanishing case.

On the left side a position boundary with given time-independent values $\bar{\mathbf{r}}^L, \mathbf{q}^L$ and $v^L \leq 0$ is imposed, furthermore we have the parameter speed $u^L: [0, T] \rightarrow \mathbb{R}$ and the Jacobian of the transformation $\chi^L: [0, T] \rightarrow \mathbb{R}$. On this side we set boundary conditions for the spatial fluxes of $\bar{\mathbf{r}}, \mathbf{q}$ and $\mathbf{v}, \boldsymbol{\omega}$ and due to the general parameterization as well for the artificial fluxes $u\boldsymbol{\tau}, u\boldsymbol{\kappa}, u\sigma_M, u\sigma_V, uk, u\sigma_M \mathbf{v}, u\mathbf{J}_M \cdot \boldsymbol{\omega}$. The initial conditions have to be compatible to the boundary conditions, consequently we require $\bar{\mathbf{r}}^A(-\mathcal{L}_0) = \bar{\mathbf{r}}^L$, $\mathbf{q}^A(-\mathcal{L}_0) = \mathbf{q}^L$ and $v^A(-\mathcal{L}_0) = v^L$. On the right side a tension-free boundary with outflow is imposed. Here we set boundary conditions for the spatial fluxes of \mathbf{n}, \mathbf{m} and $\bar{\boldsymbol{\lambda}}_\tau, \boldsymbol{\lambda}_\kappa$. We set the boundary conditions according to (2.38) and (2.41) by using (2.43).

Position and tension-free boundary

For material fluxes:

$$\begin{aligned}
 \bar{\mathbf{r}}(-\mathcal{L}(t), t) &= \bar{\mathbf{r}}^L, & \mathbf{q}(-\mathcal{L}(t), t) &= \mathbf{q}^L, \\
 \mathbf{v}(-\mathcal{L}(t), t) &= \mathbf{e}_3 v^L, & \boldsymbol{\omega}(-\mathcal{L}(t), t) &= \mathbf{0}, \\
 \mathbf{m}(-\mathcal{R}(t), t) &= \mathbf{0}, & \mathbf{n}(-\mathcal{R}(t), t) &= \mathbf{0}, \\
 \bar{\boldsymbol{\lambda}}_\tau(-\mathcal{R}(t), t) &= \mathbf{0}, & \boldsymbol{\lambda}_\kappa(-\mathcal{R}(t), t) &= \mathbf{0}.
 \end{aligned} \tag{2.43a}$$

For artificial fluxes:

$$\begin{aligned}
 \boldsymbol{\tau}(-\mathcal{L}(t), t) &= \mathbf{e}_3 \chi^L(t), & \boldsymbol{\kappa}(-\mathcal{L}(t), t) &= \mathbf{0}, \\
 \sigma_M(-\mathcal{L}(t), t) &= \chi^L(t), & \sigma_V(-\mathcal{L}(t), t) &= \chi^L(t), \\
 k(-\mathcal{L}(t), t) &= \chi^L(t), & u(-\mathcal{L}(t), t) &= u^L(t).
 \end{aligned} \tag{2.43b}$$

Remark 2.37 (Conditions for the artificial fluxes). *The boundary conditions for the artificial spatial fluxes are chosen to be located on the left side, but the right side would be possible as well. We do this since we have initial values from the nozzle on the left side that can be re-used. In the trivial case $u^L \equiv 0$, all artificial fluxes are determined by this single condition.*

The boundary conditions (2.43) are complete if both u^L and χ^L are given or can be determined implicitly, which requires more knowledge of the actual parameterization and/or space-time domain. If u^L or χ^L is given, we have a relation given by (2.24) that allows the determination of the respective other and can be used in almost all cases, namely

$$u^L(t) = \frac{1}{\chi^L(t)} v^L - \frac{d}{dt} \mathcal{L}(t). \tag{2.44a}$$

In the case that $\frac{d}{dt} \mathcal{L}(t) + u^L(t) = 0$, the above relation is no longer applicable (cannot be solved for $\chi^L(t)$). Then we can utilize the definition of u , i.e.

$$u^L(t) = \frac{\partial_t \Psi(-\mathcal{L}(t), t)}{\chi^L(t)} \tag{2.44b}$$

to determine $\chi^L(t)$.

Let us look at some examples that illustrate how to properly complete the boundary conditions. We present different inflow-approaches that can be handled, and connect them to existing literature.

Example 2.38 (Inflow-Outflow with Lagrangian boundaries). We set the general parameterization to be Lagrangian at the boundaries, thus we assume

$$u^L \equiv 0$$

and also $u(-\mathcal{R}(t), t) = 0$. Furthermore we have a given, positive extrusion speed $v^L > 0$ and some outflow $v^R(t) \geq 0$ for all $t \in [0, T]$. We decide to keep the material point

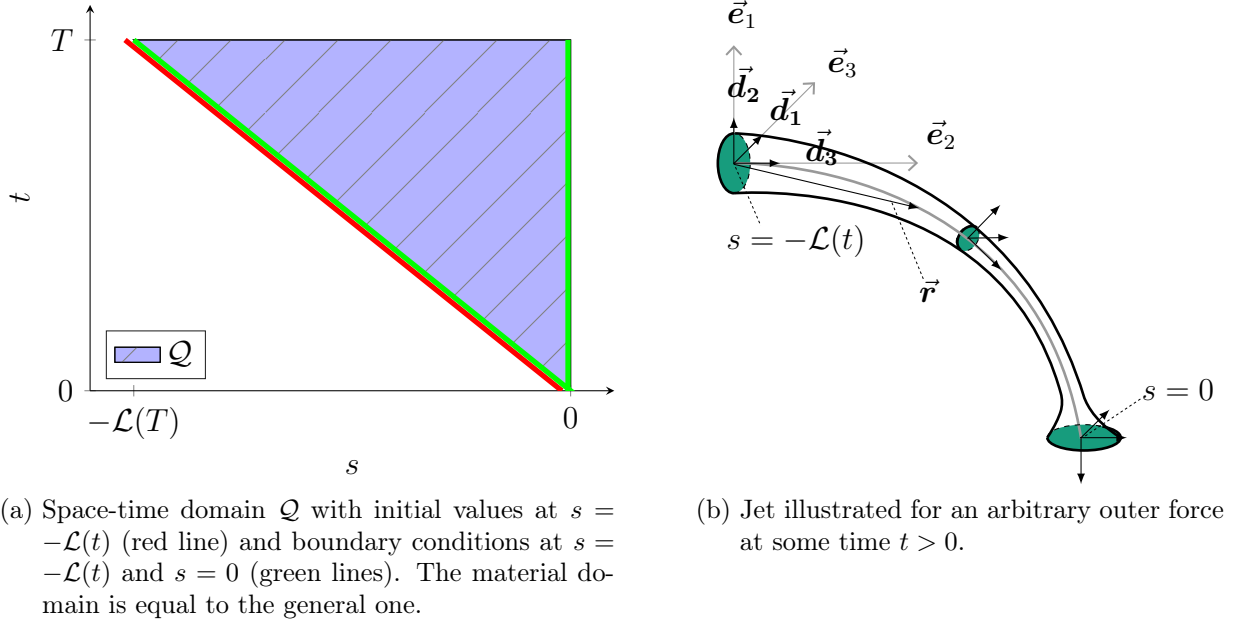


Figure 2.3: Illustration of a space-time domain and the jet curve of Example 2.38 with no outflow.

density unchanged under the re-parameterization and assume therewith $\mathcal{L}(t) = l(t)$ and $\mathcal{R}(t) = r(t)$. This leads with (2.44a) to

$$\chi^L \equiv 1.$$

The parameter speed u is yet to be determined in the inside of the domain, e.g. it vanishes for a globally Lagrangian parameterization. The no-outflow case ($v^r \equiv 0$) with its space-time domain is illustrated in Figure 2.3, it has also been studied in [5, Set-up A]. This case has an inevitable boundary layer that is built-in the system through the inherent condition of τ_3 at the tension-free boundary. The jet will be stretched through external forces within the domain, but τ_3 remains at the initial value (cf. Equation (2.41c)). To avoid that boundary layer we consider an outflow ($v^r > 0$) case. An illustration is given in Figure 2.4.

Example 2.39 (Inflow-Outflow with Eulerian boundaries). The initial material domain and general domain are set to be $[-l(0), -r(0)] = [-\mathcal{L}(t), -\mathcal{R}(t)] = [-1, 0]$, furthermore a positive extrusion speed v^L is given, i.e. $l(t) = -tv^L - 1$. Imposing the arc-length constraint $\boldsymbol{\tau} \equiv \mathbf{e}_3$ at both sides gives Eulerian boundaries. This constraint could also be enforced globally to obtain a Eulerian parameterization. The position boundary at the left side implies at all times that the Jacobian of the transformation is equal to one. Together with (2.44a) we have

$$\chi^L \equiv 1, \quad u^L \equiv v^L.$$

The jet starts out unstretched and undergoes some external loads that cause elongation to build up. To compensate for the elongation we allow mass outflow v^L at the free end,

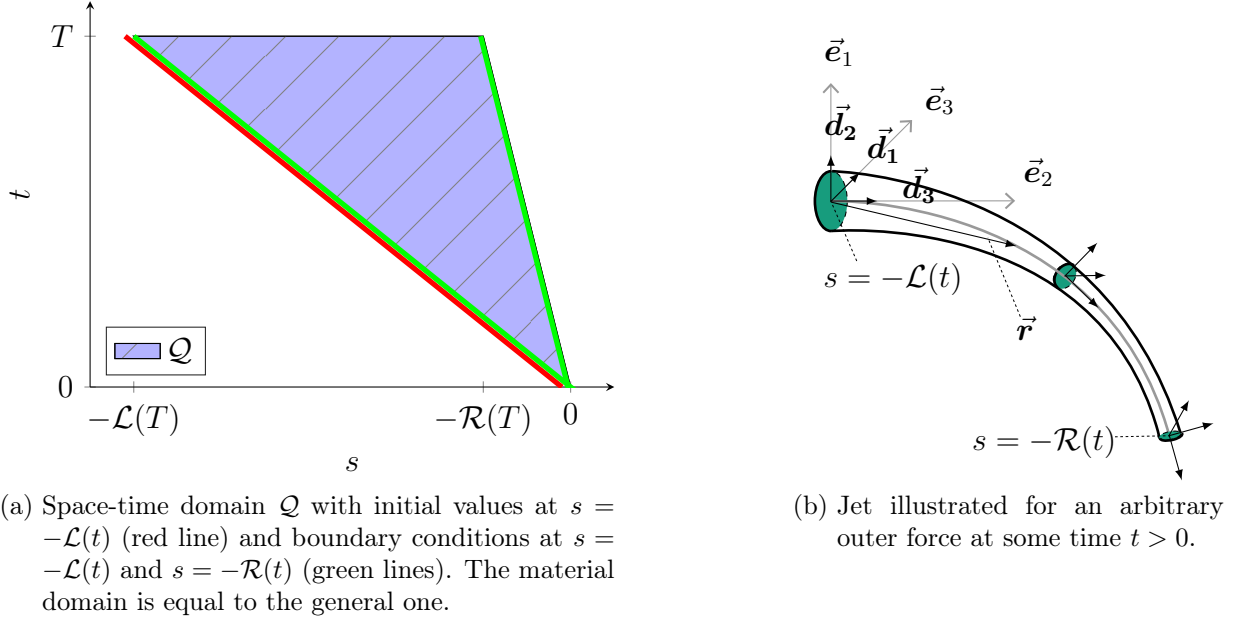


Figure 2.4: Illustration of a space-time domain and the jet curve of Example 2.38 with some given outflow.

which becomes a degree of freedom and is indirectly determined. The jet enters a steady-state after sufficient time has passed (assuming deterministic external loads). The resulting material and general domain and an illustration of the steady-state behavior is shown in Figure 2.5. Arne et al. [5, Set-up B] designed that case to obtain a time-independent general domain.

Example 2.40 (Inflow with growing Eulerian parameterization). A positive extrusion speed v^L is given, the outflow is set to zero. The arc-length constraint $\boldsymbol{\tau} \equiv \mathbf{e}_3$ is globally enforced to obtain an Eulerian parameterization, implying

$$\chi^L \equiv 1.$$

The length of the general spatial domain $|\Omega(t)| = |\mathcal{L}(t) - \mathcal{R}(t)|$ becomes a degree of freedom and thus the system is a free-boundary value problem for which special consideration is necessary. The time-dependency is collected on one side either giving: a) $\mathcal{L}(t) = 0$, $\mathcal{R}(t) = |\Omega(t)|$ or b) $\mathcal{L}(t) = |\Omega(t)|$, $\mathcal{R}(t) = 0$. This example has the same boundary layer as described and depicted in the no outflow case of Example 2.38.

a) We have with (2.44a)

$$u^L \equiv v^L.$$

This case has been studied by Panda [79] with the jet model reduced to be uniaxial, under certain assumptions. We have summarized the uniaxial model in our notation in Appendix B.3. In particular, Panda employs a Finite Volume method and utilizes a specific inherent condition of the system. The mass-line density in Eulerian parameterization remains constant at a tension-free boundary. Since the mass outflow

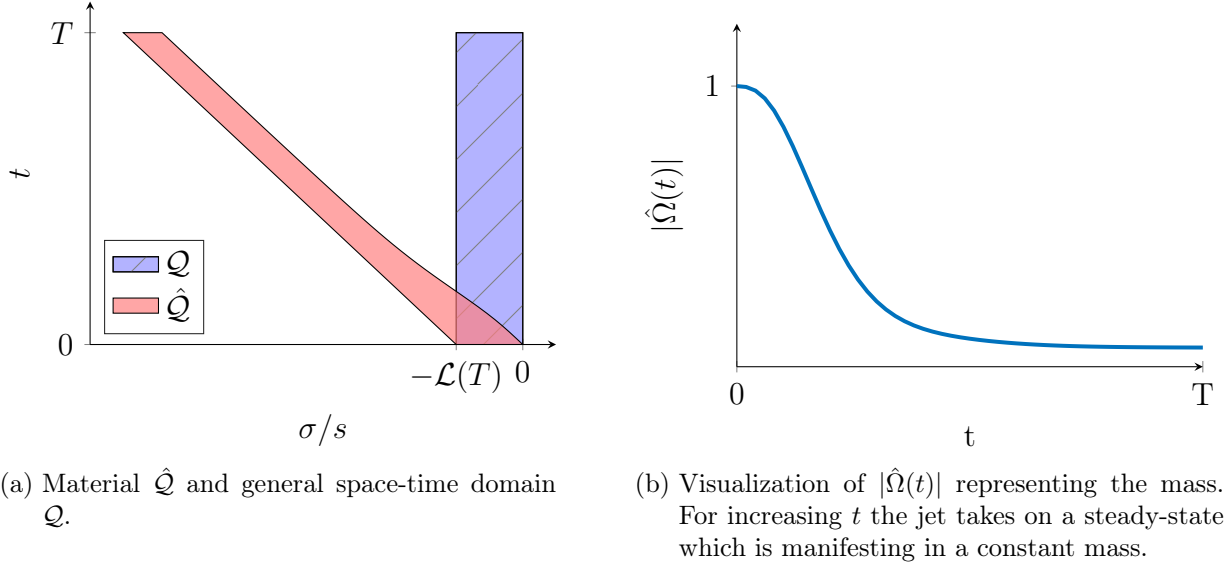


Figure 2.5: Illustration of the space-time domains and total mass of the jet of Example 2.39.

is zero, this condition allows to determine $\mathcal{R}(t)$. The free-boundary value problem is avoided by using a variable cell size for the last cell that is then split up in between time steps. Panda mentions that the method has problems in the discrete growing process, more precisely simulations with large elongations require very small time steps.

b) In this case the boundary relation (2.44a) becomes

$$u^L(t) = v^L - \frac{d}{dt}\mathcal{L}(t),$$

with $\frac{d}{dt}\mathcal{L}(t)$ representing the change in the arc-length of the jet. We study this case in Appendix C.3. To eliminate the free-boundary value problem we use a purely Lagrangian parameterized sub-problem for each time step to calculate the arc-length of the jet beforehand and use that information to advance the Eulerian simulation. The computational time is very high, even for small jet lengths. Even worse, errors in the arc-length $\mathcal{L}(t)$ due to the Lagrangian sub-problem can significantly alter the result.

Example 2.41 (Inflow-Outflow with growing Eulerian parameterization). This example is designed similarly to Example 2.40 b) with the arc-length constraint $\boldsymbol{\tau} \equiv \mathbf{e}_3$ globally enforced to obtain an Eulerian parameterization. Yet, here we seek to completely avoid the problematic of a free-boundary value problem. For that purpose we allow mass outflow on the free end and prescribe the general space-time domain with the help of $\frac{d}{dt}\mathcal{L}(t) = v^L$, $\mathcal{R}(t) = 0$ and $\mathcal{L}(0) = 0$, with a given positive extrusion speed v^L . The outflow will be a degree of freedom similar to Example 2.39, the resulting space-time domains are illustrated in Figure 2.6. With (2.44a) we have

$$\chi^L \equiv 1, \quad u^L \equiv 0.$$

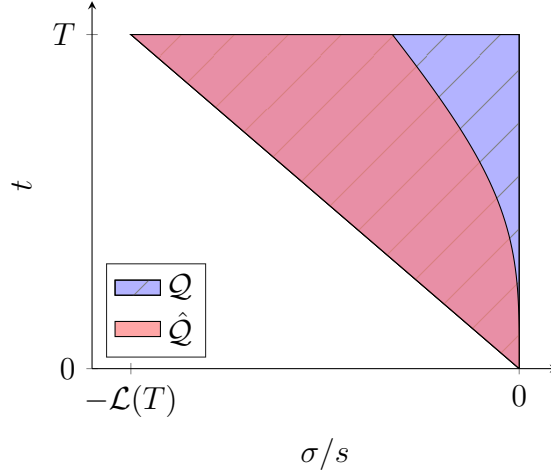
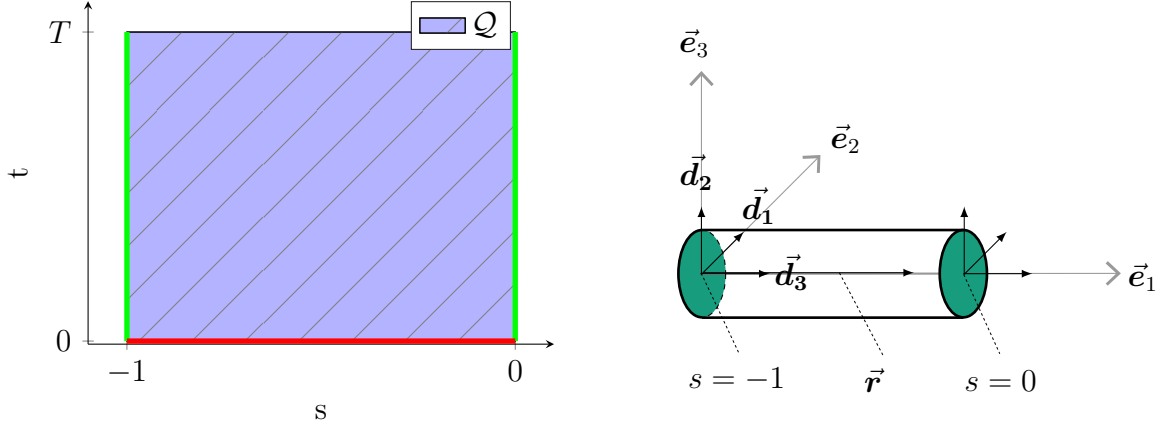


Figure 2.6: Illustration of the material \hat{Q} and general space-time domain Q of Example 2.41.

This example is expected to behave similarly to the outflow case of Example 2.38, Figure 2.4b.

The following Example 2.42 can be seen as a jet that is attached to a wall on the left side and is then undergoing the external forces. No mass enters or leaves the domain.

Example 2.42 (No inflow, no outflow). The material domain is assumed to be time-independent. We set $[-l(t), -r(t)] = [-1, 0]$ for all $t \in [0, T]$. The general domain is not restricted at this point, thus we utilize (2.44) to implicitly determine u^L and χ^L . If we set $[-\mathcal{L}(t), -\mathcal{R}(t)] = [-1, 0]$ as well (illustrated in Figure 2.7), then we can simplify $u^L \equiv 0$. In an elastic context this set-up is a cantilever, a long projecting beam fixed at only one end. Undergoing solely gravitational forces in normal direction to the initial beam, it will continuously bounce up and down (without any numerical dampening). In the viscous context the jet is expected to fall down while the viscosity keeps some cohesion to the wall creating high elongation in its proximity.



(a) Space-time domain \mathcal{Q} with initial values at $t = 0$ (red line) and boundary conditions at $s = \mathcal{L}(t) = -1, s = 0$ (green lines). The material domain coincides.

(b) Initial values for jet curve and director triad.

Figure 2.7: Illustration of the space-time domain and initial values of Example 2.42. Initial values are $r_i(s) = (s, 0, 0)$, $\mathbf{q}^A(s) = (0.5, -0.5, -0.5, -0.5)$.

2.3.2 Both tension-free boundaries

In this set-up we assume an initially straight and tension-free jet with some given length, furthermore the initial mass and volume line density are assumed to be constants in space. The jet is not fixed on either side and moves completely free in the surrounding medium. The initial parameterization is chosen to coincide with the Lagrangian one and to be arc-length.

Assume given functions $\bar{\mathbf{r}}^A, \mathbf{q}^A$, the initial length is $l_0 = \mathcal{L}_0 > 0$. Initial conditions are required analogously to Section 2.3.1 and set by using

$$\begin{aligned}
 \bar{\mathbf{r}}(s, 0) &= \bar{\mathbf{r}}^A(s), & \mathbf{q}(s, 0) &= \mathbf{q}^A(s), \\
 \mathbf{v}(s, 0) &= \mathbf{0}, & \boldsymbol{\omega}(s, 0) &= \mathbf{0}, \\
 \boldsymbol{\tau}(s, 0) &= \mathbf{e}_3 \chi(s, 0), & \boldsymbol{\kappa}(s, 0) &= \mathbf{0}, \\
 \sigma_M(s, 0) &= \sigma_V(s, 0) = k(s, 0) = \chi(s, 0).
 \end{aligned} \tag{2.45}$$

On both sides a tension-free boundary is imposed. We require boundary conditions for the fluxes of $\bar{\mathbf{r}}, \mathbf{q}, \mathbf{v}, \boldsymbol{\omega}, \mathbf{n}, \mathbf{m}, \bar{\boldsymbol{\lambda}}_\tau, \boldsymbol{\lambda}_\kappa$ and use therefor Dirichlet conditions for $\mathbf{n}, \mathbf{m}, \bar{\boldsymbol{\lambda}}_\tau, \boldsymbol{\lambda}_\kappa$ on both sides. To handle the artificial fluxes we restrict ourselves to the case where u vanishes at one side, w.l.o.g. we choose $u^L \equiv 0$. We set the boundary conditions by using (2.46):

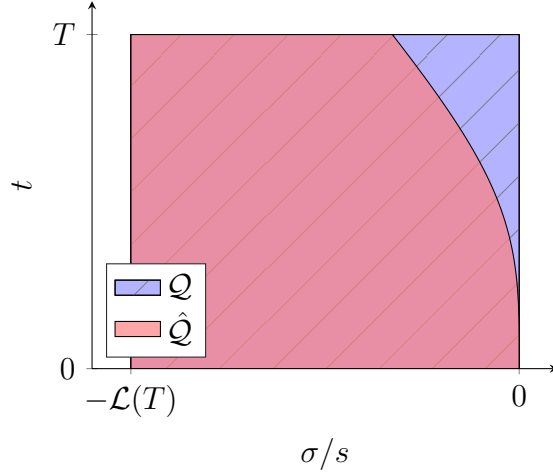


Figure 2.8: Illustration of the material \hat{Q} and general space-time domain Q of Example 2.44.

Both tension-free boundaries

$$\begin{aligned}
 \mathbf{m}(-\mathcal{L}(t), t) &= \mathbf{0}, & \mathbf{n}(-\mathcal{L}(t), t) &= \mathbf{0}, \\
 \bar{\lambda}_\tau(-\mathcal{L}(t), t) &= \mathbf{0}, & \lambda_\kappa(-\mathcal{L}(t), t) &= \mathbf{0}, \\
 \mathbf{m}(-\mathcal{R}(t), t) &= \mathbf{0}, & \mathbf{n}(-\mathcal{R}(t), t) &= \mathbf{0}, \\
 \bar{\lambda}_\tau(-\mathcal{R}(t), t) &= \mathbf{0}, & \lambda_\kappa(-\mathcal{R}(t), t) &= \mathbf{0}, \\
 u(-\mathcal{L}(t), t) &= u^L(t) = 0. & &
 \end{aligned} \tag{2.46}$$

Two examples come to mind:

Example 2.43 (No inflow, no outflow). We assume a time-independent space-time domain $[-\mathcal{L}(t), -\mathcal{R}(t)] = [-l(t), -r(t)] = [-1, 0]$, implying that u is zero on both sides. Boundary layers in τ_3 will develop on both sides due the tension-free boundary analogously to Example 2.38.

Example 2.44 (Outflow with Eulerian parameterization). We assume a time-independent space-time domain $[-\mathcal{L}(t), -\mathcal{R}(t)] = [-1, 0]$ and furthermore $l(t) = -1$, i.e. $u^L = 0$. By imposing the arc-length constraint globally we obtain an Eulerian description. Since the jet will undergo stretching there will be mass outflow on the right side to compensate for it, more precisely we have $v^R(t) = \chi(0, t)u(0, t)$. An illustration is shown in Figure 2.8.

2.3.3 Both position boundaries

The third and last possibility to combine the boundaries is to have two position boundaries. On both sides we have a fixed position and orientation, whereas we assume to have mass inflow on one side and mass outflow on the other side. We assume that a jet with some initial length is given that is straight and tension-free everywhere, furthermore the initial

mass and volume line density are assumed to be constants in space. The jet is fixed on both ends and the initial parameterization is chosen to coincide with the Lagrangian one and to be arc-length.

Assume given functions $\bar{\mathbf{r}}^A, \mathbf{q}^A, v^A$, the initial length is $l_0 = \mathcal{L}_0 > 0$. Initial conditions are required analogously to Section 2.3.1 and set by using

$$\begin{aligned} \bar{\mathbf{r}}(s, 0) &= \bar{\mathbf{r}}^A(s), & \mathbf{q}(s, 0) &= \mathbf{q}^A(s), \\ \mathbf{v}(s, 0) &= \mathbf{e}_3 v^A(s), & \boldsymbol{\omega}(s, 0) &= \mathbf{0}, \\ \boldsymbol{\tau}(s, 0) &= \mathbf{e}_3 \chi(s, 0), & \boldsymbol{\kappa}(s, 0) &= \mathbf{0}, \end{aligned} \quad (2.47)$$

$$\sigma_M(s, 0) = \sigma_V(s, 0) = k(s, 0) = \chi(s, 0). \quad (2.48)$$

Position boundaries are put on both sides using $\bar{\mathbf{r}}^L = \bar{\mathbf{r}}^A(-\mathcal{L}_0)$, $\mathbf{q}^L = \mathbf{q}^A(-\mathcal{L}_0)$, $v^L = v^A(-\mathcal{L}_0)$ and $\bar{\mathbf{r}}^R = \bar{\mathbf{r}}^A(0)$, $\mathbf{q}^R = \mathbf{q}^A(0)$, $v^R = v^A(0)$. We require boundary conditions for the fluxes of $\bar{\mathbf{r}}$, \mathbf{q} , \mathbf{v} , $\boldsymbol{\omega}$, \mathbf{n} , \mathbf{m} , $\bar{\boldsymbol{\lambda}}_\tau$, $\boldsymbol{\lambda}_\kappa$ and use therefor Dirichlet conditions for $\bar{\mathbf{r}}$, \mathbf{q} , \mathbf{v} , $\boldsymbol{\omega}$ on both sides. The artificial fluxes are handled w.l.o.g. analogously to Section 2.3.1 on the left side. Overall we set the boundary conditions by using (2.49):

Both position boundaries

For material fluxes:

$$\begin{aligned} \bar{\mathbf{r}}(-\mathcal{L}(t), t) &= \bar{\mathbf{r}}^L, & \mathbf{q}(-\mathcal{L}(t), t) &= \mathbf{q}^L, \\ \mathbf{v}(-\mathcal{L}(t), t) &= \mathbf{e}_3 v^L, & \boldsymbol{\omega}(-\mathcal{L}(t), t) &= \mathbf{0}, \\ \bar{\mathbf{r}}(-\mathcal{R}(t), t) &= \bar{\mathbf{r}}^R, & \mathbf{q}(-\mathcal{R}(t), t) &= \mathbf{q}^R, \\ \mathbf{v}(-\mathcal{R}(t), t) &= \boldsymbol{\tau}(-\mathcal{R}(t), t) v^R(t) / \chi(-\mathcal{R}(t), t), & \boldsymbol{\omega}(-\mathcal{R}(t), t) &= \mathbf{0}. \end{aligned} \quad (2.49a)$$

For artificial fluxes:

$$\begin{aligned} \boldsymbol{\tau}(-\mathcal{L}(t), t) &= \mathbf{e}_3 \chi^L(t), & \boldsymbol{\kappa}(-\mathcal{L}(t), t) &= \mathbf{0}, \\ \sigma_M(-\mathcal{L}(t), t) &= \chi^L(t), & \sigma_V(-\mathcal{L}(t), t) &= \chi^L(t), \end{aligned} \quad (2.49b)$$

$$k(-\mathcal{L}(t), t) = \chi^L(t), \quad u(-\mathcal{L}(t), t) = u^L(t). \quad (2.49c)$$

Note that the conditions for \mathbf{v} require knowledge of the elongation. At the left side it is prescribed through the initially unstretched assumption of the jet, whereas it is a degree of freedom at the right side. Here, an extrapolation seems feasible since we are dealing with mass outflow.

As an example one could think of a jet that is clamped on both sides, or a jet drawing process with roll-up, meaning the jet is extruded on one side and rolled up much quicker on the other side to stretch it in between (see Example 2.46).

Example 2.45 (Clamped jet with Lagrangian boundaries). The space-time domains are time-independent and set to $[-\mathcal{L}(t), -\mathcal{R}(t)] = [-l(t), -r(t)] = [-\|\bar{\mathbf{r}}^R - \bar{\mathbf{r}}^L\|, 0]$, thus we have no mass inflow or outflow (analogous to Example 2.42) and furthermore

$$u^L \equiv u^R \equiv 0.$$

Example 2.46 (Jet drawing process). The domains are set to $[-\mathcal{L}(t), -\mathcal{R}(t)] = [-l(0), -r(0)] = [-\|\bar{\mathbf{r}}^R - \bar{\mathbf{r}}^L\|, 0]$. A positive extrusion speed v^L as well as a positive roll up speed $\mathbf{v}(-\mathcal{R}(t), t) = \mathbf{e}_3 v^{roll}$ is assumed. The initial arc-length of the jet is $\|\bar{\mathbf{r}}^R - \bar{\mathbf{r}}^L\|$, but the jet length does not have to be constant in simulations when external forces are applied, thus hindering the use of an Eulerian parameterization. One could think of a scaled Eulerian parameterization by prescribing that the elongation is not one, but a constant over the whole jet. The relation (2.44a) holds on both sides of the boundary. For the roll up speed it holds that $v^{roll} = v^R(t)\boldsymbol{\tau}_3(-\mathcal{R}(t), t)/\chi(-\mathcal{R}(t), t)$, which implicitly determines the mass outflow. An attempt with a Lagrangian parameterization is consequently a free boundary value problem, which we want to avoid. The domain situation is similar to Figure 2.5a.

To conclude this chapter we give an overview over the presented examples.

Mixed boundaries

| Example | $\hat{\mathcal{Q}}$ inflow/outflow | \mathcal{Q} | Parameterization | Details |
|---------|---------------------------------------|---------------|-----------------------|-------------------------------|
| 2.38 | yes/yes | time-dep. | Lagrangian boundaries | |
| 2.39 | yes/yes | time-indep. | Eulerian boundaries | Outflow implicitly determined |
| 2.40 | yes/no | time-dep. | Eulerian | Free-boundary value problem |
| 2.41 | yes/yes | time-dep. | Eulerian | Outflow implicitly determined |
| 2.42 | no/no | time-indep. | Eulerian/Lagrangian | |

Both tension-free

| Example | $\hat{\mathcal{Q}}$ inflow/outflow | \mathcal{Q} | Parameterization | Details |
|---------|---------------------------------------|---------------|-----------------------|-------------------------------|
| 2.43 | no/no | time-indep. | Lagrangian boundaries | |
| 2.44 | no/yes | time-indep. | Eulerian | Outflow implicitly determined |

Both position

| Example | $\hat{\mathcal{Q}}$ inflow/outflow | \mathcal{Q} | Parameterization | Details |
|---------|---------------------------------------|---------------|-----------------------|-------------------------------|
| 2.45 | no/no | time-indep. | Lagrangian boundaries | |
| 2.46 | yes/yes | time-indep. | Scaled Eulerian | Outflow implicitly determined |

Table 2.4: Overview of the illustrative examples of the set-ups.

3 Adaptive meshes: r -refinement

The viscous jet model (Section 2.2.5) is formulated in a general parameterization with the help of the artificial parameter speed. In general, said parameter speed has to be determined to be suitable for the underlying physical scenario. The most prominent special cases would be the Lagrangian (material) and Eulerian (arc-length) parameterization. Another one, that is designed for fluid-structure-interaction problems, is called the arbitrary Lagrangian-Eulerian (ALE) parameterization, which is attributed to the flexibility of the method that allows it to be Lagrangian within the structure and to be Eulerian in the fluid with a sharp tracking of the boundary layer [53]. In our case there is no sharp interface or other features of the geometry that can be tracked, our parameterization has to be chosen appropriately for the underlying partial differential equations that can involve large solution variations over small parts of the material domain – e.g. boundary layers or oscillations. A purely Lagrangian parameterization, or in discrete terms, an equidistant mesh whose grid points track the material points, might not be able to resolve such variations properly. The idea is to improve the approximation quality – while keeping the computational effort – by finding a suitable parameterization. In this, the term *suitable* is the key challenge. In literature this can be ascribed to so-called *r-refinement* (r for relocation) strategies. Those strategies relocate the given mesh points towards areas of the domain where they are needed, i.e. areas of large solution variations. Certainly, classical *h*-refinement that inserts additional mesh points in said areas, or even an overall finer mesh, could be used as well, but the improved approximation quality comes hence with the price of increased computational costs and requires interpolation of the solution. Such interpolation strategies suitable for partial differential equations (PDEs) with hyperbolic/parabolic character and conservation laws are found e.g. [18, 16, 17, 62]. In this work we investigate the *r*-refinement, i.e. we decide to stick to the idea of finding a suitable re-parameterization.

The need for mesh adaptivity in the various areas of science is omnipresent, thus numerous strategies have been developed in literature. Huang and Russell [59] give a great summary of a variety of methods. We quote:

”Given the ubiquitous need for (...) [such strategies] in the various areas of science and engineering, a proliferation of methods have been developed in the past ... [It is] a daunting task for the potential user of adaptive mesh techniques to know where to begin looking for a method suitable for his or her particular needs.” [59, p. vii].

We focus on strategies that interpret the adaptivity as time-dependent coordinate transformations and that continuously move the mesh utilizing the mesh velocity. The resulting *moving mesh* method can be formulated in numerous ways (see [57]). At some point they all are based on or linked to the so-called *equidistribution principle*, even though the final moving mesh equations appear to be quite different. Said principle was first introduced by

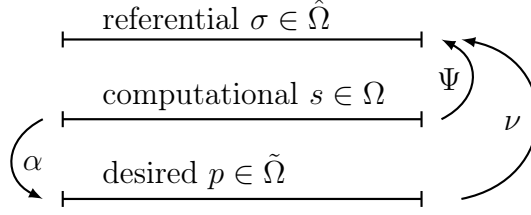


Figure 3.1: Illustration of the three parameterization layers.

de Boor [32] for solving boundary value problems for ODEs and involves selecting mesh points such that some measure of the solution error is equalized over each subinterval. To name prominent moving mesh methods we list the moving finite element method (MFE) of Miller, which computes the solution and the mesh simultaneously by minimizing the residual of the PDEs written in a finite element form, and the moving finite difference method of Dorfi and Drury [38]. The latter was directly obtained from a discrete representation of the equidistribution principle and is used to solve one-dimensional conservation laws. We select the moving mesh PDE strategies of Huang, Ren and Russel [57], which was also applied in the context of Finite Volume methods in [96]. In the process of obtaining a deeper understanding of such strategies we developed a general moving mesh framework, that is based on three parameterization layers for time-dependent domains. We do not claim its superiority, but it opens interesting approaches for the formulation of such strategies.

This section is ordered as follows: In Section 3.1 we present our frame for moving meshes. It basically has two parts: An abstract design of three parameterization layers to help interpret the adaption strategies, and some measure of the solution that indicates where to refine the mesh. In Section 3.2 we discuss different adaption strategies for a suitable re-parameterization based on our preliminary works [88, 90]. In particular, we establish the link back to the moving mesh strategies from Huang et al. [57] and introduce new strategies. Subsequently we discuss in Section 3.3 an approach to formulate refinement measures and their implications.

3.1 Three parameterization layers

The general idea of moving mesh strategies is to improve the approximation quality by re-distributing a fixed number of mesh points within the domain while keeping the computational effort. In a continuous setting the idea corresponds to applying a re-parameterization to the given model equations that is determined in some way by the moving mesh strategy. We restrict to 1d parameterized problems and consider three parameterization layers: *referential* $\sigma \in \hat{\Omega}(t)$, *computational* $s \in \Omega(t)$ and *desired* $p \in \tilde{\Omega}(t)$ parameterizations, see Figure 3.1. We denote $\hat{\Omega}(t) = [\sigma_a(t), \sigma_b(t)]$, $\Omega(t) = [s_a(t), s_b(t)]$ according to Section 2.2.1 and $\tilde{\Omega}(t) = [p_a(t), p_b(t)]$. If not otherwise mentioned we assume that p_a, p_b coincide with σ_a, σ_b . The model equations for an arbitrary application are originally formulated in a depicted parameterization (e.g. Lagrangian or Eulerian description) to which we refer to as the referential parameters. The desired parameters should now reflect some kind of optimal parameterization for the given problem, e.g. the absolute value of a gradient of a solution component becomes constant. The direct use of the desired parameterization may

not be numerically beneficial (see Remark 3.1), therefore the computational parameters are additionally introduced. Consequently we formalize three parameterization layers whereas two are normally used in literature (see Remark 3.2). The core of the framework are the time-dependent parameter transformations $\Xi \in \{\Psi, \alpha, \nu\}$ with $\Xi(\cdot, t)$ being a one-to-one mapping for $t \in [0, T]$, their relations are listed in Fig. 3.1. We assume that the transformations are at least continuously differentiable in both variables, we denote the spatial derivative as $f_\Xi = \partial_x \Xi$. In accordance with (2.21) we want to identify the left and right side of the domains respectively with each other, thus $f_\Xi > 0$. We call the transformations *parameter distribution functions* and their space derivatives *parameter densities* (formulated in Definition 3.5 and 3.6) because of their analogy when fixed domains are used (see Remark 3.3).

Remark 3.1 (Justification to modify the desired parameterization). *In literature extensive studies have been done on moving mesh strategies that consider only the referential and desired parameters, e.g. [29, 84] and for an overview [57]. They all exhibit some kind of stability problem, mostly because the discrete mesh tends not to reflect the wanted desired parameters. For example, consider the perturbation analysis of the moving mesh strategy (MMPDE1) in [57]. This strategy is a third order partial differential equation that enforces the relation between the referential and desired parameterization. If there is a perturbation in the desired parameters, it is not guaranteed that this deviation disappears over time, in fact it might even be worsened, producing an asymptotically unstable mesh.*

Remark 3.2 (Three parameterization layers). *Huang et al. [57] formalize only the referential and computational parameterization, the desired parameterization is implicitly incorporated in the computational parameters. They define what is deemed optimal for the given problem and then modify that definition in some way to obtain the computational parameterization without formally introducing a third parameterization layer, as we do.*

Remark 3.3 (Time-independent domains). *In literature there exist various moving mesh strategies on time-independent domains, e.g. [57, 58]. Following that confinement, the assumption $\hat{\Omega}(t) = \Omega(t) = \tilde{\Omega}(t) = [0, 1]$ could be made w.l.o.g.. Then, the transformations Ξ can be interpreted as probability distribution functions for the parameters. Supposing sufficient regularity $\Xi \in C^1([0, 1] \times [0, T], [0, 1])$, the derivatives f_Ξ describe the probability densities for the parameters, i.e. $\int_0^1 f_\Xi(x, t) dx = 1$.*

To allow for moving meshes, the original equations are transformed into the computational parameters by using

$$\Psi(\cdot, t): \Omega(t) \rightarrow \hat{\Omega}(t)$$

with

$$\Psi(s_a(t), t) = \sigma_a(t), \quad \Psi(s_b(t), t) = \sigma_b(t) \quad (3.1)$$

for $t \in [0, T]$ (cf. the general parameterization in Section 2.2.1). The computational parameters are not identical to the desired ones, but should approach them, i.e. $\alpha(\cdot, t): \Omega(t) \rightarrow$

$\tilde{\Omega}(t)$ should be pulled towards identity id . Consequently, the r -refinement aims at an adaption strategy (see Section 3.2, equation for the unknown Ψ) and a description of a desired re-parameterization (see Section 3.3, choice for α).

The existing moving mesh approaches regard only two parameterizations as the desired layer is implicitly incorporated in the computational layer (cf. Remark 3.2). The advantage of our proposed framework is a clear separation of all three layers. This aims to provide more flexibility in the modeling as we will see in the following. Approaches for time-dependent computational domains also exist, see Remark 3.4.

Remark 3.4 (Hybrid hr -refinement). *In literature there exists the so-called hybrid hr -refinement [78], whose key feature is its ability to add or remove mesh nodes in a smooth manner through the boundaries of the domain. This is certainly also reflected by our framework with the general space-time domains by choosing a time-independent referential domain in combination with a time-dependent computational domain. The time-dependency allows to smoothly add or remove mesh nodes.*

3.2 Adaption strategy

The adaption strategy aims to provide an equation for the transformation Ψ , or the parameter density f_Ψ . Proceeding from a given desired parameter distribution in terms of ν , f_ν , the idea behind the adaption of the computational parameters is a temporal relaxation. In that sense Ψ , f_Ψ fulfills at a later time what is currently deemed optimal with ν , f_ν . In the following we present two different types of strategies: Distribution relaxation (DELAX) and moving mesh partial differential equations (MMPDE). Thereby, the second one was originally proposed and explored in [57]. We choose these strategies to exemplify the embedding in our general framework and because they have been also employed in [96] in the context of Finite Volume methods.

The parameter densities play an important role for the moving mesh strategies. We now state a general definition for a parameter distribution and density.

Definition 3.5 (Parameter distribution Ξ). *Let a function $\Xi(\cdot, t): [A(t), B(t)] \rightarrow [a(t), b(t)]$ with sufficient regularity be given, the interval boundaries are given with some $A, B, a, b: [0, T] \rightarrow \mathbb{R}$ such that $A(t) < B(t)$ and $a(t) < b(t)$ for all $t \in [0, T]$. If $\Xi(\cdot, t)$ is bijective for all $t \in [0, T]$ and if it holds that $\Xi(A(t), t) = a(t)$ and $\Xi(B(t), t) = b(t)$, then Ξ is called a parameter distribution function.*

The parameter distribution are all related to each other, e.g. $\Psi(s, t) = \nu(\alpha(s, t), t)$ for all $(s, t) \in \mathcal{Q}$. We write in short $\Psi = \nu \circ \alpha$.

Definition 3.6 (Parameter density f_Ξ). *Let a continuous function $f_\Xi(\cdot, t): [A(t), B(t)] \rightarrow \mathbb{R}$ be given for all $t \in [0, T]$, the interval boundaries are given with some $A, B: [0, T] \rightarrow \mathbb{R}$ such that $A(t) < B(t)$. If f_Ξ satisfies*

$$\int_{A(t)}^{B(t)} f_\Xi(x, t) \, dx = b(t) - a(t) \quad (3.2)$$

for all $t \in [0, T]$ with $a, b: [0, T] \rightarrow \mathbb{R}^+$ and

$$f_{\Xi} > 0,$$

then it is called a parameter density.

If a parameter density f_{Ξ} is given, we can construct the corresponding parameter distribution by setting

$$\Xi(s, t) = \int_{A(t)}^s f_{\Xi}(x, t) dx + a(t).$$

It can easily be seen that $\Xi(A(t), t) = a(t)$ and $\Xi(B(t), t) = b(t)$ holds for all $t \in [0, T]$. Thus Ξ is the corresponding parameter distribution to f_{Ξ} .

Example 3.7 (Parameter density f_{α}). We want to exemplify Definition 3.6. Assume that a parameter density f_{α} is given. It involves the parameter domains Ω and $\hat{\Omega}$ (cf. Figure 3.1). We restate (3.2)

$$\int_{s_a(t)}^{s_b(t)} f_{\alpha}(s, t) ds = p_b(t) - p_a(t).$$

The corresponding parameter distribution α is

$$\alpha(s, t) = \int_{s_a(t)}^s f_{\alpha}(s', t) ds' + p_a(t).$$

We now want to embed a chosen MMPDE strategy of Huang and Russell [57] in our framework. They utilize a *monitor function* to describe the desired re-parameterization, we point out the link to the parameter density f_{α} afterwards. There exist many MMPDE approaches, we choose a variant that was also used in [96] for conservation laws. Before we get to that we introduce the monitor function.

Definition 3.8 (Monitor function). Let a function $\hat{M}: \hat{\mathcal{Q}} \rightarrow \mathbb{R}^+$, $t \in [0, T]$ be given and assume that $\hat{M}(\cdot, t) \in \mathcal{C}^0(\hat{\Omega}(t), \mathbb{R}^+)$ and

$$0 < \alpha_0 \leq \hat{M}(\sigma, t) \leq \alpha_1 < \infty \quad (3.3)$$

holds for all $\sigma \in \hat{\Omega}(t)$ and $t \in [0, T]$ with $\alpha_0, \alpha_1 > 0$. Then \hat{M} is called a monitor function. On the computational domain \mathcal{Q} the monitor function is denoted by $M(s, t) := \hat{M}(\Psi(s, t), t)$.

Strategy 3.9 (Moving Mesh PDE 1 (MMPDE1)). Let a monitor function M be given, satisfying $M(\cdot, t) \in \mathcal{C}^1(\Omega(t), \mathbb{R}^+)$. Assume that $\Psi(\cdot, t) \in \mathcal{C}^2(\Omega(t), \hat{\Omega}(t))$, then Ψ is determined by the PDE

$$\partial_s(M \partial_t f_{\Psi}) = -\frac{1}{\tau} \partial_s(M f_{\Psi}), \quad t \in (0, T], \quad s \in \Omega(t), \quad (3.4)$$

with appropriate initial conditions and boundary conditions taken from (3.1), and temporal relaxation parameter $\tau > 0$.

MMPDE proceeds from the relaxation ansatz $f_\Psi(s, t + \tau) = f_\nu(\alpha(s, t), t)$ for the density. By the chain rule $f_\nu \circ \alpha = (f_{\nu^{-1}} \circ \Psi)^{-1} = f_\Psi/f_\alpha$ particularly holds. The desired parameterization is modeled in terms of M with $\mathcal{M}(t) = \int_{\hat{\Omega}(t)} \hat{M}(\sigma, t) d\sigma = \int_{\Omega(t)} f_\Psi M(s, t) ds$, i.e.

$$f_{\nu^{-1}} = \frac{\hat{M}|\Omega(t)|}{\mathcal{M}}, \quad \text{implying} \quad f_\nu \circ \alpha = \frac{\mathcal{M}}{M|\Omega(t)|}, \quad f_\alpha = \frac{f_\Psi M|\Omega(t)|}{\mathcal{M}}. \quad (3.5)$$

After doing a Taylor expansion of first order of the relaxation ansatz, we obtain $\partial_t f_\Psi = -(f_\Psi - \mathcal{M}/(M|\Omega|))/\tau$. Multiplying this with M and taking the space derivative, the integral of the monitor function and the length of the domain Ω disappear and the second order MMPDE (3.4) with mixed partial derivatives is obtained. Moreover, in the limit $\tau \rightarrow 0$, $f_\alpha \equiv 1$ is enforced which implies $\alpha \equiv \text{id}$. By dealing with the parameter density f_Ψ , this strategy brings diffusion to the problem. In contrast to the DELAX strategies, the MMPDE1 strategy cannot be formulated directly on the level of the distribution function Ψ .

We introduce a further MMPDE strategy of [57] that is designed to create a spatially balanced mesh [56]. It is based on the idea of attraction and repulsion pseudo forces in between mesh nodes and was first stated in a slight variation in [2]. We choose it due to its simplicity and will show the link to our framework after we introduce it.

Strategy 3.10 (Moving Mesh PDE 2 (MMPDE2)). Let a monitor function M be given, satisfying $M(\cdot, t) \in \mathcal{C}^1(\Omega(t), \mathbb{R}^+)$. Assume that $\Psi(\cdot, t) \in \mathcal{C}^2(\Omega(t), \hat{\Omega}(t))$, then Ψ is determined by the PDE

$$\partial_t \Psi = -\frac{1}{\tau M} \partial_s (M f_\Psi), \quad t \in (0, T], \quad s \in \Omega(t), \quad (3.6)$$

with appropriate initial and boundary conditions taken from (3.1), and temporal relaxation parameter $\tau > 0$. In Section 2.2.1 the artificial parameter speed u has been introduced. It was defined in (2.16) as

$$u(s, t) = \partial_t \Psi^{-1}(\sigma, t)|_{\sigma=\Psi(s, t)} = -\frac{\partial_t \Psi(s, t)}{\partial_s \Psi(s, t)}.$$

We split the second derivative on the right side of Equation (3.6), reorder it and plug in in the definition of u to obtain

$$-\tau u = \frac{\partial_s M}{M} + \frac{\partial_s f_\Psi}{f_\Psi} = \frac{\partial_s (M f_\Psi)}{M f_\Psi}. \quad (3.7)$$

In (3.7) the term $M f_\Psi$ appears. Expressed in terms of our parameter densities it is $f_\alpha \mathcal{M}/|\Omega(t)|$. MMPDE2's equation then becomes

$$-\tau u = \frac{\partial_s f_\alpha}{f_\alpha}.$$

Our ansatz with the three parameterization layers allows a more obvious way to formulate adaption strategies, namely through temporal relaxation of the transformation. In particular, we now present two strategies involving Ψ .

Strategy 3.11 (Distribution relaxation with Ψ (DELAX1)). Let a parameter density f_α with corresponding parameter distribution α be given. Then Ψ is determined by the evolution equation

$$\partial_t \Psi = -\frac{1}{\tau} \left(\Psi - \Psi \circ \alpha^{-1} \Big|_{p=s} \right), \quad t \in (0, T], \quad s \in \Omega(t) \quad (3.8)$$

with appropriate initial conditions and temporal relaxation parameter $\tau > 0$.

DELAX1 proceeds from the relaxation ansatz $\Psi(s, t + \tau) = \nu(s, t) = \Psi(\alpha^{-1}(s, t), t)$ for the distribution. Moreover, in the limit $\tau \rightarrow 0$, (3.8) enforces that $\alpha \equiv \text{id}$. The strategy requires the computation of the inverse of α in (3.8) and the interpolation between the parameterizations. The costs are relatively cheap as the domains are one-dimensional, but the use of the inverse transformation needs special consideration (cf. Remark 3.12). Alternatively, one might also think of an respective evolution equation for f_Ψ

$$\partial_t f_\Psi = -\frac{1}{\tau} \left(f_\Psi - (f_\Psi \circ \alpha^{-1}) f_{\alpha^{-1}} \Big|_{p=s} \right).$$

Remark 3.12. *The transformation α can have steep gradients in the region where the monitor function is large. Obtaining a reasonably accurate numerical inverse thus requires a priori an adaptive mesh that resolves said steep gradient appropriately (which is what we want to compute in the first place). Huang and Russell [59] come across a similar issue when the inverse transformation Ψ^{-1} appears in one of their moving mesh strategies (more precisely, in a strategy they call MMPDE5xi) and recommend the use of an alternate solution procedure in that case (the model equation and mesh equation are decoupled in some way and solved separately). We focus on a simultaneous solution procedure (model and mesh equations are solved together) and will therefore not use DELAX1 in the later numerics of this work.*

Instead of Ψ its inverse Ψ^{-1} can be considered for the ansatz.

Strategy 3.13 (Distribution relaxation with Ψ^{-1} (DELAX2)). Let a parameter density f_α with corresponding parameter distribution α be given. Then Ψ is determined by the transport equation

$$\partial_t \Psi - \frac{1}{\tau} (s - \alpha) \partial_s \Psi = 0, \quad t \in (0, T], \quad s \in \Omega(t) \quad (3.9)$$

with appropriate initial conditions and boundary conditions taken from (3.1), and temporal relaxation parameter $\tau > 0$. Similar to Strategy 3.10 we can express (3.9) in terms of u , i.e.

$$u = -\frac{1}{\tau} (s - \alpha), \quad t \in (0, T], \quad s \in \Omega(t). \quad (3.10)$$

DELAX2 proceeds from the relaxation ansatz $\Psi^{-1}(\sigma, t + \tau) = \nu^{-1}(\sigma, t) = \alpha(\Psi^{-1}(\sigma, t), t)$ for the distribution and uses $\sigma = \Psi(s, t)$. Moreover, in the limit $\tau \rightarrow 0$, (3.9) enforces that $\alpha \equiv \text{id}$. Alternatively, one might also think of an respective evolution equation for $f_{\Psi^{-1}}$

$$\partial_s u = -\frac{1}{\tau} (1 - f_\alpha).$$

We presented four different strategies to determine Ψ . They were constructed by assuming that a parameter density f_α , or a monitor function \hat{M} , was given. It is not clear whether Ψ that was determined by either of those strategies is a parameter distribution function. In the following part we show that this is in fact the case. Before that we give a remark on analogous terminology that is used in literature.

Remark 3.14. *In the context of [58, 59] Huang et al. introduce the so called no node crossing condition. It states that the transformation Ψ has a strictly positive spatial derivative. In our wording that means that Ψ is a parameter distribution according to Definition 3.5. Furthermore, (MMPDE1) is called (MMPDE4) by Huang et al. and (MMPDE2) is called (modified MMPDE5).*

Theorem 3.15. *Let Ψ be determined by MMPDE1 (Strategy 3.9). Additionally assume that the initial transformation satisfies*

$$0 < \partial_s \Psi(s, t_\star(s)) \quad (3.11)$$

with the creation time t_\star from (2.25). Then Ψ is a parameter distribution according to Definition 3.5 for small enough τ .

The proof of theorem 3.15 is analogous to the ideas for the time-independent domain stated in [58]. However, the time-dependent boundaries introduce extra terms that need to be estimated.

Proof. First we take the indefinite integral of MMPDE (3.4) with respect to s , the integration constant $c(t)$ appears:

$$\partial_{ts} \Psi(s, t) + \frac{1}{\tau} \partial_s \Psi(s, t) = \frac{1}{M(s, t)} c(t). \quad (3.12)$$

Then one more time with a definite integral with respect to s

$$\left[\partial_t \Psi(s, t) \right]_{s_a(t)}^{s_b(t)} + \frac{1}{\tau} \left[\Psi(s, t) \right]_{s_a(t)}^{s_b(t)} = c(t) \int_{s_a(t)}^{s_b(t)} \frac{1}{M(s, t)} ds.$$

The boundary conditions of Ψ together with the strictly positive monitor function yield

$$c(t) = \left(\frac{1}{\tau} (\sigma_b(t) - \sigma_a(t)) + \partial_t \Psi(s_b(t), t) - \partial_t \Psi(s_a(t), t) \right) \left(\int_{s_a(t)}^{s_b(t)} \frac{1}{M(s, t)} ds \right)^{-1} > 0 \quad (3.13)$$

for all $t \in (0, T]$ and a sufficiently small enough τ . Equation (3.12) can be seen as an ODE for $\partial_s \Psi$, therefore one can obtain for $s \in [s_a(t), s_b(t)]$, $t \geq t_\star(s)$ the solution

$$\partial_s \Psi(s, t) = e^{-t/\tau} \left(\partial_s \Psi(s, t_\star(s)) + \underbrace{\int_{t_\star(s)}^t \frac{1}{M(s, \tilde{t})} c(\tilde{t}) e^{\tilde{t}/\tau} d\tilde{t}}_{>0 \text{ (3.3) \& (3.13)}} \right). \quad (3.14)$$

and consequently we estimate

$$\partial_s \Psi(s, t) > e^{-t/\tau} \partial_s \Psi(s, t_\star(s)) \stackrel{(3.11)}{>} 0 \quad (3.15)$$

for all $(s, t) \in \mathcal{Q}$. □

Theorem 3.16. *Let Ψ be determined by MMPDE2 (Strategy 3.10). Additionally assume that the transformation satisfies*

$$0 < \partial_s \Psi(s_a(t), t) \quad (3.16)$$

for all $t \in [0, T]$. Then Ψ is a parameter distribution according to Definition 3.5.

Proof. Equation (3.7) is an ODE in space for $M\partial_s \Psi(s, t)$, therefore we obtain for every $t \in [0, T]$

$$\partial_s \Psi(s, t) = \frac{1}{M} e^{-\tau u s} M(s_a(t), t) \partial_s \Psi(s_a(t), t).$$

With (3.16) one immediately sees that $\partial_s \Psi(s, t)$ is always strictly positive because M is strictly positive. \square

Theorem 3.17. *Let a function Ψ be determined by DELAX1 (Strategy 3.11). Then Ψ is a parameter distribution according to Definition 3.5.*

Proof. All properties of α are carried over to Ψ for all $t \in [0, T]$ due to the structure of the equation. Since α is a parameter distribution, it is strictly monotone increasing, and then Ψ is also strictly monotone increasing. Thus Ψ is also a parameter distribution. \square

Theorem 3.18. *Let Ψ be determined by DELAX2 (Strategy 3.13). Then Ψ is a parameter distribution according to Definition 3.5.*

Proof. Strategy 3.13 is an initial-boundary value problem with a linear PDE (3.9) given by

$$\begin{aligned} \partial_t \Psi(s, t) - \partial_s \Psi(s, t) \frac{1}{\tau} (s - \alpha(s, t)) &= 0, & (s, t) \in \mathcal{Q}, \\ \Psi(s_a(t), t) &= \sigma_a(t), \\ \Psi(s_b(t), t) &= \sigma_b(t), \\ \Psi(s, 0) &= \sigma_0(s). \end{aligned} \quad (3.17)$$

Instead of directly solving this transport equation the method of characteristics can be used to set-up an equivalent problem. We define the family of functions $\bar{\Psi}_k(t) := \Psi(\bar{s}_k(t), t)$ with a suitable coordinate transformation. We have

$$\frac{d}{dt} \bar{\Psi}_k(t) = \frac{d}{dt} \Psi(\bar{s}_k(t), t) = \left(\partial_t \Psi(\bar{s}_k(t), t) + \partial_s \Psi(s, t) \Big|_{s=\bar{s}_k(t)} \frac{d}{dt} \bar{s}_k(t) \right) = 0$$

and it follows that

$$\frac{d}{dt} \bar{s}_k(t) = -\frac{1}{\tau} (\bar{s}_k(t) - \alpha(\bar{s}_k(t), t)). \quad (3.18)$$

Here, \bar{s}_k denotes the characteristic with creation time t_k . We assumed that α is continuous in space and time, thus \bar{s}_k is continuously differentiable. The original problem (3.17) is equivalent to the family of problems involving a system of ODEs

$$\begin{aligned} \frac{d}{dt}\bar{\Psi}_k(t) &= 0, & \bar{\Psi}_k(t_k) &= \Psi(\bar{s}_k(t_k), t_k) = \sigma_k, \\ \frac{d}{dt}\bar{s}_k(t) &= -\frac{1}{\tau}(\bar{s}_k(t) - \alpha(\bar{s}_k(t), t)), & \bar{s}_k(t_k) &= \begin{cases} s_{a/b}(t_k), & t_k > 0, \\ s_k, & t_k = 0 \end{cases} \end{aligned} \quad (3.19)$$

for all t such that $\sigma_k \in \hat{\Omega}(t)$ and $s_k \in \Omega(0)$. The notation $s_{a/b}(t_k)$ indicates that the characteristic can have its origin either on the left or the right side of the domain.

We want to show that Ψ is a parameter distribution and we do that by showing that Ψ has strictly positive space derivative. For a solution of (3.19) this is equivalent to

- a) $\bar{s}_i(t') > \bar{s}_j(t')$,
- b) $\bar{\Psi}_i(t') > \bar{\Psi}_j(t')$

for all $t_j > t_i$ and $t' \in [t_j, T]$ with $\bar{s}_i(t_i) > \bar{s}_j(t_j)$ and $\bar{\Psi}_i(t_i) > \bar{\Psi}_j(t_j)$. This can easily be seen by looking at

$$\partial_s \Psi(s, t) = \lim_{h \rightarrow 0} \frac{\bar{\Psi}_k(t) - \bar{\Psi}_l(t)}{\bar{s}_k(t) - \bar{s}_l(t)}$$

where t_k and t_l are chosen such that $\bar{s}_k(t) = s$ and $h = |\bar{s}_k(t) - \bar{s}_l(t)|$.

ad a) Consider a characteristic \bar{s}_i with its origin at the left boundary – the right boundary is analogous. According to the assumption $\alpha(s_a(t), t) = s_a(t)$ we have $\frac{d}{dt}\bar{s}_i(t)|_{t=t_i} = 0$, additionally we assumed $\frac{d}{dt}s_a(t) < 0$. Thus there exists $\epsilon > 0$ such that $s_a(t) < \bar{s}_i(t)$ for all $t \in (t_i, t_i + \epsilon]$. Now consider the characteristic \bar{s}_j with $t_j = t_i + \epsilon$. It holds that

$$\bar{s}_j(t_j) = s_a(t) < \bar{s}_i(t_j).$$

Assume that there exists a $\check{t} \in [t_j, T]$ such that

$$\begin{cases} \bar{s}_j(t) < \bar{s}_i(t), & t \in [t_j, \check{t}), \\ \bar{s}_j(t) = \bar{s}_i(t), & t = \check{t}. \end{cases}$$

Define $w(t) := (\bar{s}_i - \bar{s}_j)(t)$ for all $t \geq t_j$. With the above assumption the relation $w(\check{t}) = 0$ must be fulfilled. With (3.18) we have

$$\frac{d}{dt}w(t) = -\frac{1}{\tau}(w - (\alpha(\bar{s}_i(t), t) - \alpha(\bar{s}_j(t), t))), \quad w(t_j) =: \delta > 0. \quad (3.20)$$

Since α is assumed to be strictly positive increasing, $\alpha(\bar{s}_i(t), t) - \alpha(\bar{s}_j(t), t) > 0$ holds and we can estimate for all $t \in [t_j, \check{t}]$ that

$$\frac{d}{dt}w(t) > -\frac{1}{\tau}w.$$

It follows for the solution of (3.20) that

$$w(t) > \delta \exp \left(-\frac{1}{\tau}(t - t_j) \right)$$

We use continuous extension to infer that

$$w(\check{t}) \geq \delta \exp \left(-\frac{1}{\tau}(\check{t} - t_j) \right)$$

must also hold. This is a contradiction to the assumption that $w(\check{t}) = 0$.

ad b) The solution of the initial value problem (3.19) for $\bar{\Psi}_k$ is $\bar{\Psi}_k(t) = \sigma_k$. Thus the assertion holds since $\bar{\Psi}_i(t_i) > \bar{\Psi}_j(t_j)$ was true. \square

3.3 Desired re-parameterization

The redistribution of the parameters is performed with respect to the chosen density function f_α (or monitor function \hat{M} in MMPDE, respectively), thus we want to discuss the possibilities and restrictions in its modeling in the following part. We call both f_α and M from now on mesh control functions. In general, f_α is a model-dependent arbitrarily complicated functional on the solution that should approach $f_\alpha \equiv 1$ by moving the mesh. We will introduce a special form of the parameter density and show how the parameter density can be constructed following the ideas of the monitor function.

A monitor function can be easily modeled, the only restriction according to Definition 3.8 is continuity and strict positivity. There already exists a vast spectrum of possibilities in literature [59]. When a parameter density is modeled, it needs to be ensured that it fulfills the requirements given in Definition 3.6. We already have a link between the monitor function and the parameter density with (3.5) and suggest the following general structure: Let a continuous and strictly positive function $z: \mathcal{Q} \rightarrow \mathbb{R}^+$ be given (analogously to a monitor function). Then we assume that the parameter density f_α has the form

$$f_\alpha(s, t) = \frac{z(s, t) |\Omega(t)|}{Z(t)}, \quad Z(t) := \int_{\Omega(t)} z(s', t) \, ds' \quad (3.21)$$

and the modeling of a parameter density reduces to the modeling of the function z . This allows to adopt all ideas from already existing monitor functions straightforwardly to the parameter densities. We present two approaches how to remodel the function z from a given monitor function.

Strategy 3.19 (Implicated parameter density). Let an arbitrary monitor function \hat{M} be given. Then this monitor function implicates a parameter density f_α in the form (3.21) through (3.5) by using

$$z(s, t) = f_\Psi(s, t) \hat{M}(\Psi(s, t), t).$$

We call this f_α the implicated parameter density of \hat{M} .

Another idea how to remodel the function z from a given monitor function is to define a comparable measure solely on the computational coordinates. This seems reasonable since the computational parameterization is the one that the actual computation is carried out on.

Strategy 3.20 (Transferred parameter density). Let an arbitrary monitor function \hat{M} be given that involves the solution component \hat{y} and spatial derivatives of it in referential parameterization, i.e. $\hat{M} = \hat{M}(\sigma, t, \hat{y}, \partial_\sigma \hat{y}, \dots)$. Then the idea of this monitor function is transferred to a parameter density f_α in the form (3.21) by using

$$z(s, t, y, \partial_s y, \dots) := \hat{M}(s, t, y, \partial_s y, \dots)$$

with the solution component in computational parameterization y . This means that z uses the expression of the monitor function and puts it in the context of the computational parameterization. One can say, the rule is to replace σ with s and \hat{y} with y in the expression of the monitor function.

The following example illustrates the implicated and transferred parameter density and shows similarities.

Example 3.21. Consider a solution $\hat{y}: \hat{Q} \rightarrow \mathbb{R}$ with large, strictly positive derivative in the referential parameterization. To obtain a moderate (constant) derivative in the computational parameterization, we model the monitor function as

$$\hat{M}(\sigma, t) = \partial_\sigma \hat{y}(\sigma, t). \quad (3.22)$$

The implicated and transferred parameter density f_α then is obtained by using

$$z_{\text{imp}}(s, t) = f_\Psi(s, t) \partial_\sigma \hat{y}(\sigma, t) \big|_{\sigma=\Psi(s, t)}, \quad (3.23)$$

$$z_{\text{tra}}(s, t) = \partial_s y(s, t) \quad (3.24)$$

in (3.21) for z . Remember the type categorization of the unknowns in Section 2.2.1. If \hat{y} is a type-0 field, then we have $y(s, t) = \hat{y}(\Psi(s, t), t)$ and $\partial_\sigma \hat{y}(\sigma, t) \big|_{\sigma=\Psi(s, t)} = \partial_s y(s, t) / f_\Psi(s, t)$ with (2.14) and (2.17b). In consequence the implicated and transferred parameter density coincide. But this is only a special case, in general they are fundamentally different. Consider e.g.

$$\hat{M}(\sigma, t) = \sqrt{1 + |\partial_\sigma y(\sigma, t)|^2}, \quad (3.25)$$

which is called the *arc-length monitor function* [59]. We state z for the implicated and transferred parameter density

$$\begin{aligned} z_{\text{imp}}(s, t) &= f_\Psi(s, t) \sqrt{1 + \left| \partial_\sigma \hat{y}(\sigma, t) \big|_{\sigma=\Psi(s, t)} \right|^2}, \\ z_{\text{tra}}(s, t) &= \sqrt{1 + |\partial_s y(s, t)|^2}. \end{aligned}$$

We emphasize again that both f_α and \hat{M} are arbitrary functions that have to be modeled to be suitable for the given physical problem. It is possible that very large peaks appear in the mesh control function, e.g. when steep gradients are present, or that the mesh control function becomes non-smooth for approximations of high order derivatives. This lack of smoothness can create rapid changes in cell sizes between adjacent cells that translate into larger truncation errors which may effect the convergence of the mesh movement [82]. In [59] this effect is avoided by employing spatial smoothing to the mesh control function. More precisely, the direct smoothing of the mesh control function is based on the use of an elliptic smoothing operator (the Laplacian).

Lemma 3.22 (Spatial smoothing operator G). [58, Lemma 3.1]. *Let a continuous function $f: \mathcal{I} \rightarrow \mathbb{R}$ be given, where \mathcal{I} is a closed interval. We denote the interior with $\overset{\circ}{\mathcal{I}}$ and the boundary with $\partial\mathcal{I}$. Assume $0 < \alpha < f(s) \leq \beta < \infty$ for all $s \in \mathcal{I}$ with $\alpha, \beta > 0$. Then the solution $v: \mathcal{I} \rightarrow \mathbb{R}$ of*

$$\begin{aligned} Gv &\equiv (Id - \lambda_0^{-2} \Delta)v = f & s \in \overset{\circ}{\mathcal{I}} \\ \partial_s v(s) &= 0 & s \in \partial\mathcal{I} \end{aligned} \quad (3.26)$$

exists and is unique for constant $\lambda_0 > 0$. Furthermore, for $v = G^{-1}f$ holds

- i) $v \in \mathcal{C}^2(\overset{\circ}{\mathcal{I}}) \cap \mathcal{C}^1(\partial\mathcal{I})$,
- ii) $0 < \alpha < v(s) \leq \beta < \infty \quad s \in \mathcal{I}$,
- iii) $|\partial_s v/v| \leq \lambda_0, \quad s \in \mathcal{I}$.

In the following we will apply the smoothing operator to arbitrary functions. We use the superscript G to indicate that, i.e. for any function $f: \mathcal{I} \rightarrow \mathbb{R}$ with arbitrary bounded domain \mathcal{I} the usage of the smoothing operator is indicated by $f^G = G^{-1}f$. With its help we introduce the smoothed monitor function M^{smo} and parameter density f_α^{smo} .

Lemma 3.23 (Smoothed monitor function). *Let a monitor function M and a smoothing operator G be given. Then $M^{smo}(\cdot, t) := M^G(\cdot, t)$, $t \in [0, T]$ is also a monitor function with the same upper and lower bound and it follows that*

- i) $M^{smo}(\cdot, t) \in \mathcal{C}^1(\Omega(t), \mathbb{R}^+)$ for $t \in [0, T]$,
- ii) $|\partial_s M^{smo}(s, t)/M^{smo}(s, t)| \leq \lambda_0, \quad s \in \Omega(t), t \in [0, T]$ with λ_0 from Lemma 3.22.

Proof. Both i) and ii) follow with Lemma 3.22. Since M is continuous it follows for the solution of the Poisson problem (3.26) that $M^{smo}(\cdot, t) \in \mathcal{C}^1(\Omega(t), \mathbb{R}^+)$ for all $t \in [0, T]$. The lower and upper bound of M still hold for M^s , thus it is also strictly positive and therefore a monitor function. \square

Lemma 3.24 (Smoothed parameter density). *Let a parameter density f_α in the form (3.21) and a smoothing operator G be given. We define*

$$f_\alpha^{smo}(s, t) := \frac{z^G(s, t) |\tilde{\Omega}(t)|}{\int_{\Omega(t)} z^G(s', t) ds'}.$$

Then f_α^{smo} is also a parameter density for the distribution α .

Proof. Analogous to Lemma 3.23, the function z^G is also strictly positive and continuous, which carries over to f_α^{smo} . Furthermore we have

$$\int_{\Omega(t)} f_\alpha^{smo}(s', t) \, ds' = |\tilde{\Omega}(t)|.$$

This agrees with Definition 3.6, thus f_α^{smo} is a parameter density for the distribution α . \square

The aim is to use the smoothed mesh control functions in the moving mesh strategies to obtain a transformation that produces a spatially smooth mesh. If the mesh, or more precisely the transformation Ψ , is *locally quasi-uniform* (see Definition 3.25), then the spatial smoothness is guaranteed and such meshes normally lead to an approximation error of the same (asymptotic) order as uniform ones, e.g. see [64, 65, 58]. In the discrete case, the local quasi-uniformity translates to a restriction of the cell size variation of direct neighbors in space.

Definition 3.25 (Local quasi-uniformity (LQU)). Assume a transformation Ψ with sufficient regularity, in this case at least $\Psi(\cdot, t) \in C^2(\Omega(t), \hat{\Omega}(t))$ for all $t \in [0, T]$. If

$$\left| \frac{\partial_{ss} \Psi(s, t)}{\partial_s \Psi(s, t)} \right| \leq \lambda \quad (3.27)$$

holds for a constant $\lambda > 0$ and for all $(s, t) \in \mathcal{Q}$, then the transformation is called *local quasi-uniform*.

In Section 3.2 we presented MMPDE strategies taken from literature and adjusted them to our time-dependent domains. The usage of a smoothed mesh control function creates locally quasi-uniform transformations Ψ . This has already been proven for time-independent domains in [58], we now extend the proofs for time-dependent domains.

Theorem 3.26 (LQU with MMPDE1). Let Ψ be determined by MMPDE1 (Strategy 3.9). Let the assumption of Theorem 3.15 be true. Assume additionally that the inequalities

$$\left| \frac{\partial_s M(s, t)}{M(s, t)} \right| \leq \lambda_0 \quad (3.28)$$

(also called smoothness condition) and

$$\left| \frac{\partial_{ss} \Psi(s, t_\star(s))}{\partial_s \Psi(s, t_\star(s))} \right| \leq \lambda_1 \quad (3.29)$$

are satisfied for all $s \in \Omega(t)$, $t \in [0, T]$ with some constants $\lambda_0, \lambda_1 > 0$. Then

$$\left| \frac{\partial_{ss} \Psi(s, t)}{\partial_s \Psi(s, t)} \right| \leq \lambda_0 + \lambda_1 + \frac{\alpha_1}{\alpha_0} C(s)$$

holds for all $s \in \Omega(t)$, $t \in (0, T]$ with some positive and bounded function C .

Proof. Remember the equation for $\partial_s \Psi(s, t)$ from (3.14). We separate the integral, take the s derivative and apply the absolute value to obtain

$$\begin{aligned} H &:= \left| \partial_{ss} \Psi(s, t) - e^{-t/\tau} \left[\partial_{ss} \Psi(s, t_*(s)) + \partial_{ts} \Psi(s, t) \Big|_{t=t_*(s)} \frac{d}{ds} t_*(s) \right] \right| \\ &= \left| e^{-t/\tau} \frac{d}{ds} \int_{t_*(s)}^t \frac{c(\tilde{t}) e^{\tilde{t}/\tau}}{M(s, \tilde{t})} d\tilde{t} \right| \\ &= \left| e^{-t/\tau} \left[\int_{t_*(s)}^t \frac{\partial_s M(s, \tilde{t})}{M(s, \tilde{t})^2} c(\tilde{t}) e^{\tilde{t}/\tau} d\tilde{t} - \frac{c(t_*(s)) e^{t_*(s)/\tau}}{M(s, t_*(s))} \frac{d}{ds} t_*(s) \right] \right|. \end{aligned}$$

The absolute value can be split up due to the strict positivity of c , cf. (3.13). The monitor function M in the right term is estimated with its lower bound (3.3) and the smoothness condition of M , cf. (3.28), is used:

$$H \leq e^{-t/\tau} \left[\lambda_0 \int_{t_*(s)}^t \frac{1}{M(s, \tilde{t})} c(\tilde{t}) e^{\tilde{t}/\tau} d\tilde{t} + \frac{1}{\alpha_0} \left| \frac{d}{ds} t_*(s) \right| c(t_*(s)) e^{t_*(s)/\tau} \right].$$

Equation (3.14) is now plugged in again to replace the integral giving

$$\begin{aligned} H &\leq \lambda_0 (\partial_s \Psi(s, t) - e^{-t/\tau} \partial_s \Psi(s, t_*(s))) + \frac{1}{\alpha_0} \left| \frac{d}{ds} t_*(s) \right| e^{-t/\tau} c(t_*(s)) e^{t_*(s)/\tau} \\ &\stackrel{(3.11)}{\leq} \lambda_0 \partial_s \Psi(s, t) + \frac{1}{\alpha_0} \left| \frac{d}{ds} t_*(s) \right| e^{-t/\tau} c(t_*(s)) e^{t_*(s)/\tau}. \end{aligned}$$

The whole inequality is now divided by the positive $\partial_s \Psi(s, t)$ and the lower bound for $\partial_s \Psi(s, t)$, cf. (3.15), is inserted on all but the first occurrence

$$\begin{aligned} &\left| \frac{\partial_{ss} \Psi(s, t)}{\partial_s \Psi(s, t)} - \frac{1}{\partial_s \Psi(s, t_*(s))} \left[\partial_{ss} \Psi(s, t_*(s)) + \frac{d}{ds} t_*(s) \partial_{ts} \Psi(s, t) \Big|_{t=t_*(s)} \right] \right| \\ &\leq \lambda_0 + \frac{1}{\alpha_0} \frac{\left| \frac{d}{ds} t_*(s) \right|}{\partial_s \Psi(s, t_*(s))} e^{t_*(s)/\tau} c(t_*(s)). \end{aligned}$$

The absolute value on the left side is now estimated with $|a| - |b| \leq |a - b|$ and assumption (3.29) is used:

$$\left| \frac{\partial_{ss} \Psi(s, t)}{\partial_s \Psi(s, t)} \right| \leq \lambda_0 + \lambda_1 + \frac{\left| \frac{d}{ds} t_*(s) \right|}{\partial_s \Psi(s, t_*(s))} \left(\frac{1}{\alpha_0} e^{t_*(s)/\tau} c(t_*(s)) + \left| \partial_{ts} \Psi(s, t) \Big|_{t=t_*(s)} \right| \right).$$

We further estimate with the help of (3.12) and the lower bound of M , cf. (3.3), the mixed-derivative term of Ψ yielding

$$|\partial_{ts} \Psi(s, t)| = \frac{1}{M(s, t)} c(t) - \underbrace{\frac{1}{\tau} \partial_s \Psi(s, t)}_{>0} \leq \frac{1}{\alpha_0} c(t).$$

We insert this result in the inequality and use the total space derivative of the boundary $\Psi(s, t_*(s)) = \sigma_{a/b}(t_*(s))$ to replace $\frac{d}{ds}t_*(s)$ on the right side:

$$\left| \frac{\partial_{ss}\Psi(s, t)}{\partial_s\Psi(s, t)} \right| \leq \lambda_0 + \lambda_1 + \frac{1}{\alpha_0} c(t_*(s)) \frac{\left| \frac{d}{ds}t_*(s) \right|}{\partial_s\Psi(s, t_*(s))} (e^{t_*(s)/\tau} + 1).$$

The integral constant $c(t_*(s))$, cf. (3.13), is bounded by α_1/τ_* with (3.3) and a constant $0 < \tau_* < \tau$ that is chosen small enough such that

$$\frac{1}{\tau_*} \geq \frac{1}{\tau} (\sigma_b(t) - \sigma_a(t)) + \partial_t\Psi(s_b(t), t) - \partial_t\Psi(s_a(t), t) \Big|_{t=t_*(s)}$$

holds. We conclude the complete estimation for $t \in (0, T]$, $s \in \Omega(t)$ with

$$\left| \frac{\partial_{ss}\Psi(s, t)}{\partial_s\Psi(s, t)} \right| \leq \lambda_0 + \lambda_1 + \frac{\alpha_1}{\alpha_0} C(s)$$

that incorporates a factor C that is dependent on the space-time domain and the initial transformation,

$$C(s) := \frac{1}{\tau_*} \frac{\left| \frac{d}{ds}t_*(s) \right|}{\partial_s\Psi(s, t_*(s))} (e^{t_*(s)/\tau} + 1). \quad \square$$

Theorem 3.27 (LQU with MMPDE2). *Let Ψ be determined by MMPDE2 (Strategy 3.10). Let the assumption of Theorem 3.15 be true. Assume additionally that the inequality*

$$\left| \frac{\partial_s M(s, t)}{M(s, t)} \right| \leq \lambda_0 \quad (3.30)$$

is satisfied for all $s \in \Omega(t)$, $t \in [0, T]$ with some constant $\lambda_0 > 0$. Then

$$\left| \frac{\partial_{ss}\Psi(s, t)}{\partial_s\Psi(s, t)} \right| \leq \lambda_0 + \lambda_1$$

holds for all $s \in \Omega(t)$, $t \in (0, T]$ with some constant λ_1 .

Proof. Let us restate MMPDE2 from (3.7) and apply the absolute value to obtain with assumption (3.30)

$$\begin{aligned} \left| \frac{\partial_{ss}\Psi}{\partial_s\Psi} \right| &= \left| \frac{\partial_s M}{M} \right| + \tau |u| \\ &\leq \lambda_0 + \tau |u|. \end{aligned}$$

Since u is a part of the assumed existing solution we now know that the left term does not explode, i.e. there exists some constant λ_1 such that $\tau |u| < \lambda_1$. \square

The necessary assumptions in Theorem 3.26 and Theorem 3.27 are not critical and can be easily fulfilled. If a smoothed monitor function M^{smo} is used, the additional regularity assumption and smoothness condition to obtain LQU are already fulfilled and the constant λ_0 becomes the parameter from the spatial smoothing operator from Lemma 3.22. The only remaining restrictions to choose M are the upper and lower bound and continuity, and that the transformation is locally quasi-uniform in its initial values for Theorem 3.26. (cf. (3.29)).

Remark 3.28. *The MMPDE2 and DELAX2 strategy seem preferable for actual computations because they can be formulated in terms of the parameter speed u , which directly couples them to the applied model equations. The MMPDE1 is a PDE of third order, which might increase the computation time, but the higher regularity could help create a smoother mesh. The DELAX1 strategy is not considered for the numerical investigations later in this work, nevertheless it might be viable when an alternate solution procedure [59] is used, cf. Remark 3.12.*

We have presented the general mechanisms that are needed for the moving mesh strategies with our way of formalizing three parameterization layers. We presented two strategies already existing in literature, MMPDE1, MMPDE2, and proposed two new moving mesh strategies, namely DELAX1, DELAX2. All four strategies determine a valid re-parameterization for the underlying model equations. Unfortunately, we could not show up to this point that the DELAX strategies produce a transformation that is LQU, in contrast to the MMPDE strategies. Nevertheless, this does not mean that it is not possible, or that the produced meshes are of lower quality. The remaining question is now: *How do you choose a suitable mesh control function?* By suitable we mean that it has to be tailored to the physical equations that are being solved. Therefore we will discuss options together with the studies that we are conducting later in Section 5.4.2.

4 Numerical scheme

This chapter is dealing with the numerical scheme that we develop for the jet model. The jet model (cf. Chapter 2) can be rewritten as a first-order system of PDAEs on a space-time domain with time-dependent spatial boundaries. We propose a Finite Volume method on a staggered grid with flux approximation suitable for the underlying differential-algebraic character (cf. Section 2.2.4) and a proper geometric handling of the space-time domain.

The Finite Volume method is introduced in Section 4.1, which is partly already published in [89]. More precisely, in Section 4.1.1 we state the discrete scheme together with a truncation error analysis. In Section 4.1.2 we then discuss details of the incorporation of initial and boundary conditions. After stating a step-by-step algorithm in Section 4.1.3 we show in Section 4.1.4 the reconstruction of a discrete solution, in particular for the comparison of solutions with different spatial grids. Finally we give details about the application to the jet model in Section 4.2 and motivate the necessity of our new formulation of the jet kinematics from Section 2.2.4.

4.1 Finite Volume method

The most common numerical methods for PDEs are the Finite Difference, Finite Volume and Finite Element method. In this work we deal with equations resulting from conservation and balance laws. Conservation properties of the scheme play an important role, we thus favor the Finite Volume method with its natural conservation for cell averages: The given domain is partitioned in small finite volumes (cells) and the underlying equations are solved on them in a weak sense in integral form. After integration over the cells, proper quadrature rules are required for the fluxes and source terms. Whereas the source terms need more difficult approximations on the whole cell, the application of the Gauss integral theorem leads to surface integrals for the fluxes giving natural flux conservation. Every unknown can be evaluated at the cell midpoints or the cell edges. If said unknown mostly appears in the fluxes of the system the edge location is often beneficial because the surface integral can therewith be easily evaluated. When both node and edge location is used for different unknowns, such an approach is known as *staggered grid*. For a general introduction to Finite Volume schemes we refer to [70, 105].

In our case the space-time domain is looked upon as a two-dimensional domain with t and s axis (cf. Figure 2.7a). We have temporal fluxes for derivatives in t and spatial fluxes for the ones in s . We use quadrilateral finite volumes, or more precisely space-time cells, that are rectangular within the domain and possibly trapezoidal at the boundary for a proper approximation of the space-time boundaries. We do this to allow decoupling of the time levels for iterative solving. Further on we employ a central approximation for the spatial fluxes and a fully implicit approximation of the temporal fluxes. The staggered grid

approach allows here the use of a narrow, direct-neighbor stencils. The approximations are tailored for the underlying DAE character of the jet model (see Remark 4.1)

Remark 4.1 (Applicability of the scheme). *Our discrete scheme is clearly motivated by the needs of the jet model which is a PDAE system and not strictly classifiable as parabolic/hyperbolic system. The underlying DAE character of the jet model (cf. Section 2.2.4) is taken into account, thus our scheme is suitable for PDAE with similar character (namely index one or less in space and two or less in time). If we were to interpret our discrete approach as a method of lines we would employ a shifted Gauss's method in space and an implicit Euler's method in time. Considering each direction by itself, we have general theoretical statements for the convergence order of implicit Euler's method when applied to index 2 or less DAEs (also for index 3 when the algebraic, index 3 unknowns only appear in linear or quadratic terms), and ones for the Gauss's method when applied to index 1 DAEs [46]. Nevertheless this is not a hard restriction, since convergence of higher index DAEs is not ruled out, it is simply not guaranteed by the theory. The scheme is designed to be applicable to any system that has similar requirements than the jet model and that fits in the form (4.1). As an example, the index 3 formulation of the jet model, cf. Table 2.2, shows linear convergence order for time-independent domains.*

4.1.1 Scheme and truncation error

Assume a system of partial differential equations on the domain \mathcal{Q} from (2.15) that is given in the form

$$\partial_t \mathbf{a}(\mathbf{y}(s, t)) + \partial_s \mathbf{f}(\mathbf{y}(s, t)) + \mathbf{g}(\mathbf{y}(s, t)) = \mathbf{0} \quad (4.1)$$

with the vector of unknowns $\mathbf{y}: \mathcal{Q} \rightarrow \mathbb{R}^M$, and the vector-valued functions $\mathbf{a}, \mathbf{f}, \mathbf{g}: \mathbb{R}^M \rightarrow \mathbb{R}^M$, $M \in \mathbb{N}$ for the time evolution, spatial fluxes and source terms, respectively. The latter three are explicitly allowed to be zero component-wise.

The following parts are technical and introduce necessary notation and formalizations at length, as for the result: the discrete version of (4.1) is given by (4.3) and its truncation error by (4.5).

The components of the functions in (4.1) will be addressed with a lower index \cdot_j , $j = 1, \dots, M$. We assume that the unknowns and equations are already ordered and uniquely assign the unknowns y_j , $j = 1, \dots, M$ to the corresponding j -th equation of system (4.1)

$$\partial_t a_j(\mathbf{y}(s, t)) + \partial_s f_j(\mathbf{y}(s, t)) + g_j(\mathbf{y}(s, t)) = 0.$$

We seek a consistent numbering of the discrete grid points throughout the whole simulation and thus put a fixed numbering onto the axes – independent from the actual domain. We do this because the spatial domain is time-dependent and we do not want to re-number when the domain is changing. For this purpose we introduce the equidistant spatial grid $s_{i+1/2}$, $i \in \mathbb{Z}$ with $s_{i+1/2} < s_{i+3/2}$ and $s_{1/2} = 0$ as the reference, the grid spacing is Δs . Additionally we set $s_i = (s_{i+1/2} + s_{i-1/2})/2$, thus the entire spatial grid comprises all s_k with $k \in \mathcal{K} := \{\dots, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, \dots\}$. The temporal grid is given by t_n , $n \in \mathbb{N}$ with $t_n < t_{n+1}$ and $t_0 = 0$ as reference, the grid spacing is $\Delta t_n := t_n - t_{n-1}$ (see Figure 4.1). Throughout this section the usage of the indices i , n , k and j will be consistent, meaning i

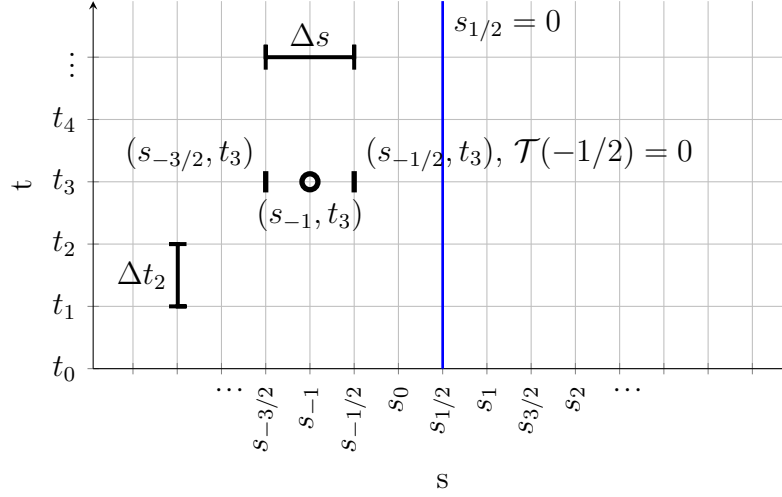


Figure 4.1: Consistent numbering of the $s - t$ plane with $s_{1/2} = 0$ as the reference. The three exemplary tuples indicate where nodes (circles, e.g. (s_{-1}, t_3)) and edges (bars, e.g. $(s_{-3/2}, t_3)$, $(s_{-1/2}, t_3)$) will be located.

will be integers, n natural numbers (zero excluded), $k \in \mathcal{K}$ and $j = 1, \dots, M$. All discrete space-time tuples enclosed by \mathcal{Q} are collected in the set

$$\mathcal{Q}^h = \{(s_k, t_n) \in \mathcal{Q} \mid s_a(t_n) \leq s_k \leq s_b(t_n), t_n \leq T, k \in \mathcal{K}, n \in \mathbb{N}\}.$$

We call those tuples with $k \in \mathbb{Z}$ nodes and those with $k \notin \mathbb{Z}$ edges and formalize this choice by

$$\mathcal{T}(k) = \begin{cases} 1, & \text{if } k \in \mathbb{Z}, & \text{(node)} \\ 0, & \text{if } k \notin \mathbb{Z}. & \text{(edge)} \end{cases}$$

An illustration of \mathcal{Q} and \mathcal{Q}^h for a fixed domain is shown in Figure 4.2a. Aiming for natural narrow stencils of the spatial fluxes (see Remark 4.5), we decide for every unknown whether it should be of node type or of edge type, which means in agreement with the space-time tuples that their discrete counterpart will be located either on nodes or edges. The associated model equation receives the same type and we formalize this decision with

$$\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_M) \in \{0, 1\}^M.$$

To obtain the discrete version of (4.1) we construct the finite volume cells with the help of the nodes and edges. More precisely we use a subset of the space-time tuples that we denote by $\mathcal{Q}^{h,A} \subset \mathcal{Q}^h$ and call it the set of active tuples, or active domain. It is defined by

$$\begin{aligned} \mathcal{Q}^{h,A} = \{ & (s_k, t_n) \in \mathcal{Q}^h \mid [((s_{k-1/2}, t_n) \in \mathcal{Q}^h \wedge \mathcal{T}(k-1/2) = p_l) \vee (s_{k-1}, t_n) \in \mathcal{Q}^h] \\ & \wedge [((s_{k+1/2}, t_n) \in \mathcal{Q}^h \wedge \mathcal{T}(k+1/2) = p_r) \vee (s_{k+1}, t_n) \in \mathcal{Q}^h], k \in \mathcal{K}, n \in \mathbb{N} \} \end{aligned} \quad (4.2)$$

and utilizes $p_l, p_r \in \{0, 1\}$ that have to be chosen appropriately for the model equations (see Remark 4.2). We call p_l the type of the left boundary and p_r the type of the right

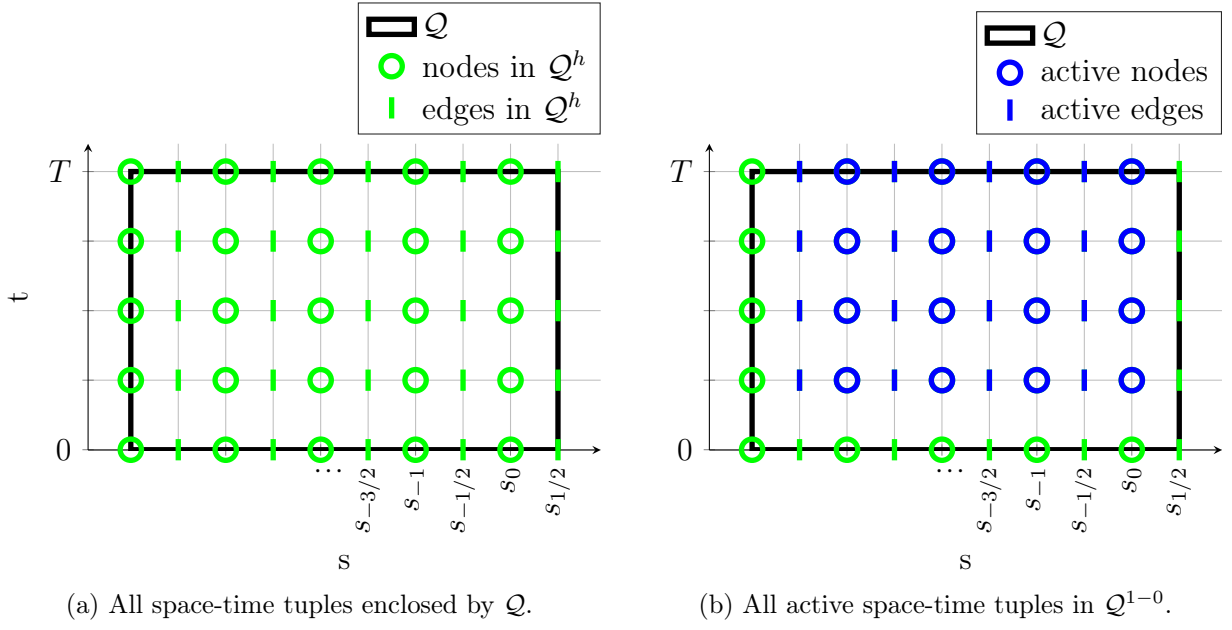


Figure 4.2: Illustration of Q^h and $Q^{h,A}$ for an example of a time-independent spatial domain.

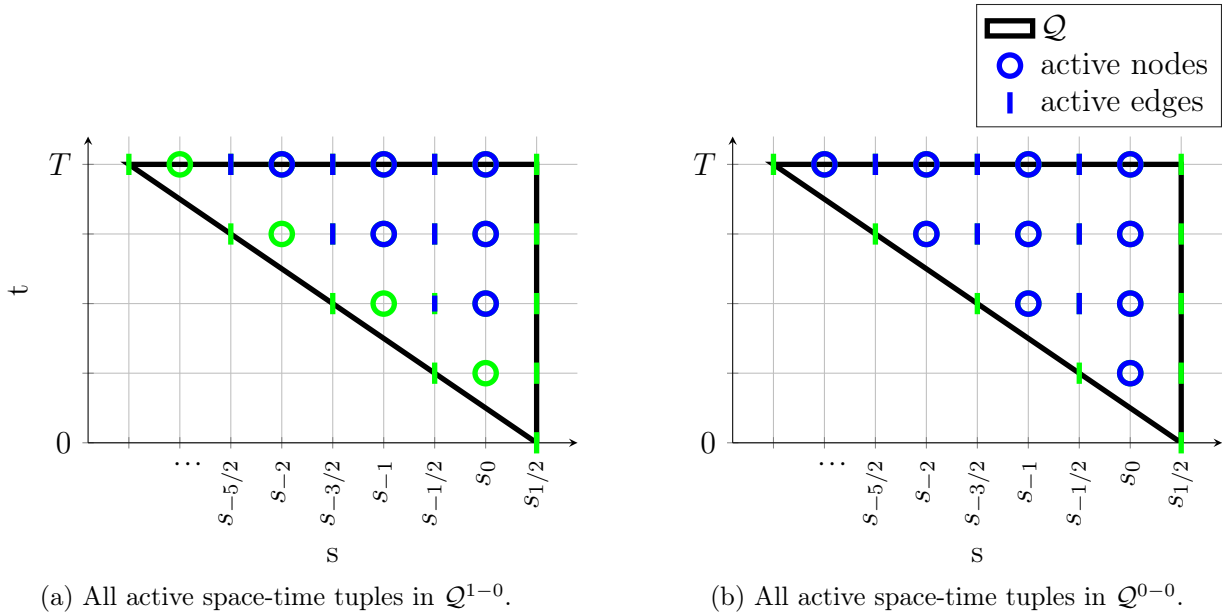


Figure 4.3: Illustration of $Q^{h,A}$ for an example of a time-dependent spatial domain.

boundary. The set $\mathcal{Q}^{h,A}$ can be explained the following way: If the left boundary has the type p_l , then for every time level the tuple in $\mathcal{Q}^{h,A}$, that is closest to this boundary, has the opposite type. The right boundary is handled analogously with p_r . The elements of $\mathcal{Q}^{h,A}$ are called active nodes or edges, exemplarily indicated in blue in Figure 4.2b and Figure 4.3 for a time-independent and time-dependent domain, respectively. As an abbreviation, and to indicate the actual choice of the boundary types, we use $\mathcal{Q}^{p_l-p_r}$ for the active domain $\mathcal{Q}^{h,A}$ with boundary types p_l and p_r .

Remark 4.2 (Choice of active domain). *Note that $\mathcal{Q}^{h,A}$, or more precisely, p_l, p_r for the boundaries has to be chosen appropriately with respect to the model problem. In the numerical approach we need to ensure that all boundary conditions can be coupled naturally through interpolation within the discrete fluxes of (4.1). A carelessly chosen set of active tuples can decouple the boundary conditions from the system due to the staggered grid, creating an unsolvable system in the process. As the most general active domain we could consider all tuples with $t > t_0$ and whose left and right neighbor is also within the domain. In this work we always assume that the active domain has the form (4.2).*

We construct a finite volume cell for every active node and edge in $\mathcal{Q}^{h,A}$ according to Strategy 4.3, examples of the resulting domain partitioning are illustrated in Figure 4.5.

Strategy 4.3 (Construction of finite volume cells). Let one of the active tuples $(s_k, t_n) \in \mathcal{Q}^{h,A}$ be given. We construct its corresponding cell specifically to allow iterative time solving and use four space-time points as the corners. We say that this cell is assigned to time level t_n and of the same type as the original tuple. The two top corners of the cell are always $(s_{k-1/2}, t_n)$, $(s_{k+1/2}, t_n)$ and there are two cases for the two bottom corners (as illustrated in Figure 4.4):

- 1) $(s_k, t_{n-1}) \in \mathcal{Q}^{h,A}$: A rectangular cell is constructed with the top corners and $(s_{k+1/2}, t_{n-1})$, $(s_{k-1/2}, t_{n-1})$ as the bottom corners.
- 2) Otherwise: A trapezoidal cell is constructed with the top corners and the bottom ones $(s_{k+1/2}, t_\star(s_{k+1/2}, t_n))$, $(s_{k-1/2}, t_\star(s_{k-1/2}, t_n))$ with the help of the creation time t_\star , cf. Definition 2.13.

We apply Gauss' Theorem to express the integral of the fluxes over the cells through an integral over the faces and use a fully implicit approximation (with box quadrature rule) for the temporal fluxes and a central approximation (with midpoint quadrature rule) for the spatial fluxes and analogous approximations for the source integral. The incorporation of the time-dependent boundary is facilitated with the help of the creation time t_\star . Overall our approach yields component-wise for $j = 1, \dots, M$ and every $(s_k, t_n) \in \mathcal{Q}^{h,A}$ with $\mathcal{D}_j = \mathcal{T}(k)$ the discrete equation

$$\begin{aligned} \Delta s [a_j(\mathbf{Y}_k(t_n)) - a_j(\mathbf{Y}_k(t_{n-1}^k))] + \Delta t_n^k [f_j(\mathbf{Y}_{k+1/2}(t_n)) - f_j(\mathbf{Y}_{k-1/2}(t_n))] \\ + \Delta s \Delta t_n^k g_j(\mathbf{Y}_k(t_n)) = \mathbf{0} \end{aligned} \quad (4.3)$$

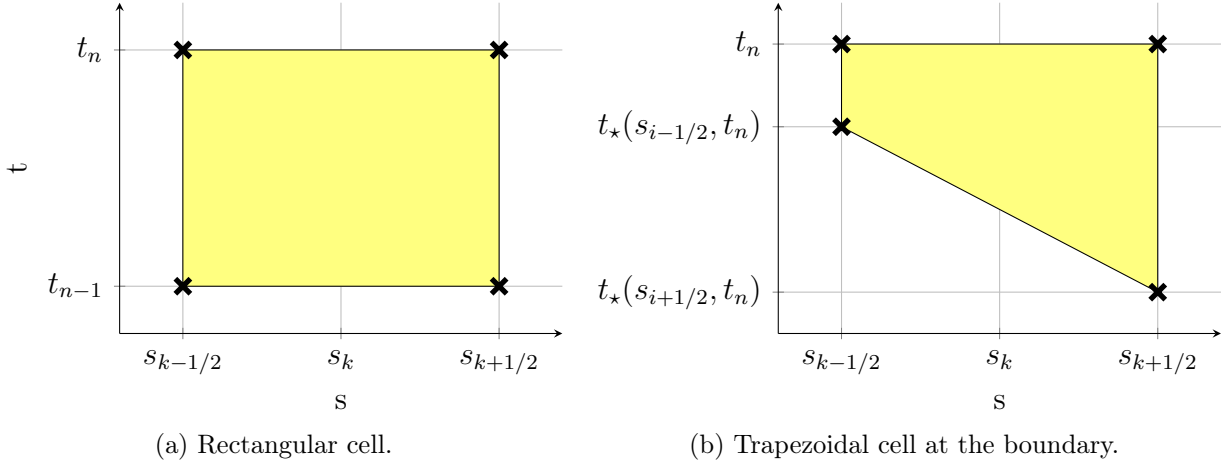
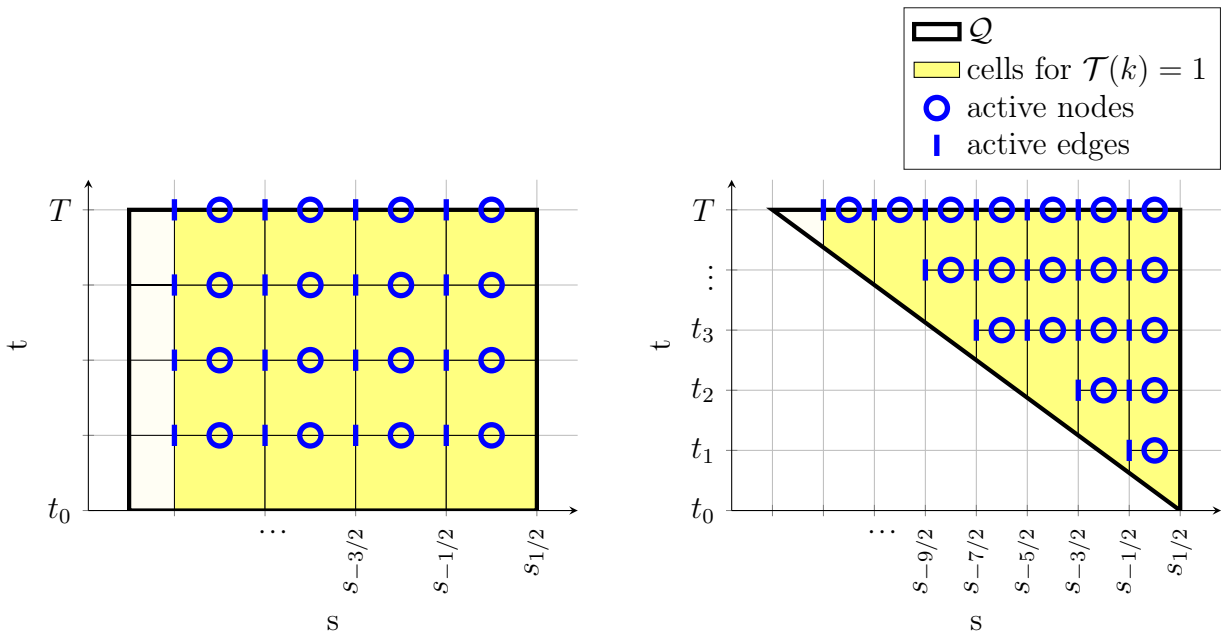


Figure 4.4: Illustration of a slanted space-time cell. The crosses exemplify the space-time tuples used for the construction of the cell and can be either nodes or edges, depending on k . The function t_* is the creation time, cf. Definition 2.13.



(a) Example for a time-independent domain, e.g., Example 2.42. Note that the cells are not fully covering the domain, on the left side there are non-active half-cells which will be set by boundary conditions.

(b) Example for a time-dependent domain, e.g., Example 2.38. Notice the extension of the cells at the boundary over multiple time levels. For example, the edge at $(s_{-5/2}, t_2)$ is not active. It is within the domain, but its left-adjacent node is not. Very small cells are avoided this way.

Figure 4.5: Illustration of the domain partitioning with cells built around all active nodes (blue circles) used for node type equations (analogously for edge type).

where the adjusted time step is given by $\Delta t_n^k = t_n - t_{n-1}^k$ with

$$t_{n-1}^k = \begin{cases} t_{n-1}, & \text{if } (s_k, t_{n-1}) \in \mathcal{Q}^{h,A}, \\ t_\star(s_k, t_n), & \text{otherwise.} \end{cases}$$

The approximations $\mathbf{Y}_k = (Y_{k,1}, \dots, Y_{k,M}): [0, T] \rightarrow \mathbb{R}^M$ in (4.3) are given for all $(s_k, t_n) \in \mathcal{Q}^{h,A}$ by the recursive description

$$Y_{k,j}(t_n) = \begin{cases} Y_{k,j}^n, & \text{if } \mathcal{D}_j = \mathcal{T}(k), \\ \frac{1}{2} (Y_{k-1/2,j}(t_n) + Y_{k+1/2,j}(t_n)), & \text{otherwise} \end{cases} \quad (4.4)$$

with the spatial averages

$$Y_{k,j}^n = \frac{1}{\Delta s} \int_{s_{k-1/2}}^{s_{k+1/2}} y_j(s, t_n) ds.$$

That means that $Y_{k,j}^n$ agrees with the value of y_j at the midpoint of the interval to $\mathcal{O}(\Delta s^2)$ and that we use linear interpolation for \mathbf{Y}_k in (4.4). Higher order interpolation is certainly possible here, but only feasible if numerically beneficial, cf. Section 5.2. For all other $(s_k, t) \in \mathcal{Q} \setminus \mathcal{Q}^{h,A}$ (non-active) that are required in (4.3) or (4.4), we set the corresponding $\mathbf{Y}_k(t)$ with help of the initial and boundary conditions. Initial values can be set directly due to the adjusted time step, whereas boundary conditions need to be interpolated to reflect the actual distances. We use a first-order interpolation here and formalize it in the following Section 4.1.2. Overall, the resulting non-linear system for the unknowns $Y_{k,j}^n$ is solved successively for every time level t_n starting with t_1 and using Newton's method.

To show consistency we need the local truncation error of our proposed scheme, which directly follows from its derivation that is presented in Appendix D.1. It was obtained with simple quadrature rules and Taylor expansions for the approximation of the integrals. We do a first order approximation for temporal fluxes and a second order approximation for spatial ones for cells away from the boundary. Cells adjacent to the boundary are first order due to the applied linear interpolation. Source terms are approximated with a one-point quadrature rule of the two-dimensional cell integral. One noteworthy step in the derivation of the method is the handling of the slanted, south face of the trapezoidal cells, cf. Figure 4.5b. Its integral requires the evaluation of the temporal and spatial flux. With the help of another Taylor expansion we obtain a fully implicit evaluation of the spatial flux on the edges of the north face. Assuming that no interpolation in (4.4) is necessary, the local truncation error of the discrete scheme (4.3) is

$$\mathbf{E}_{rec} = \mathcal{O}(\Delta t_{max}^2 + \Delta s^3 + \Delta s \Delta t_{max}^2 + \Delta s^3 \Delta t_{max}) \quad (4.5a)$$

for rectangular cells with $\Delta t_{max} := \max\{\Delta t_n \mid t_n \leq T, n \in \mathbb{N}\}$ and

$$\mathbf{E}_{sla} = \mathcal{O}(\Delta t_{sla}^2 + \Delta s_{sla}^2 \Delta s + \Delta s \Delta t_{sla}^2 + \Delta s_{sla}^2 \Delta t_{sla}) \quad (4.5b)$$

for slanted cells at the boundary. Note that the last two terms in (4.5a) result solely from the source terms whereas in (4.5b) the last term is also shaped through the spatial

flux approximation in the south, slanted face. In the case that interpolation in (4.4) is necessary, the error $\Delta s^2 \Delta t_{max}$ additionally appears in E_{rec} (and analogously in E_{sla} , but the error is not worsened there). The error E_{sla} is dependent on the space-time domain through the worst-case time step

$$\Delta t_{sla} = \max\{\Delta t_n^k \mid (s_k, t_n) \in \mathcal{Q}^{h,A}\},$$

that reflects the maximum time that a node or edge stays non-active but is inside of \mathcal{Q} , and the worst-case spatial step

$$\Delta s_{sla} = \max(\{|s_{k_l} - s_a(t_n)| \mid n \in \mathbb{N}\} \cup \{|s_b(t_n) - s_{k_r}| \mid n \in \mathbb{N}\})$$

that reflects the maximum distance of the sampling points for the linear interpolation. In the definition of Δs_{sla} the indices of the first and last active tuple of the time level t_n appear, they are explained in Notation 4.4. Due to the construction of $\mathcal{Q}^{h,A}$ the worst-case spatial step has the upper bound of $2\Delta s$, the worst-case time step is proportional to Δt with a constant dependent on the space-time domain. In general it is feasible to assume that the number of slanted cells is much smaller than the number of rectangular cells, thus the error is expected to be dominated by E_{rec} in the global error.

Notation 4.4 (First and last active tuple). Consider all space-time tuples of an arbitrary time level t_n in $\mathcal{Q}^{h,A}$. There exists $k_l^n \in \mathcal{K}$ such that $(s_{k_l^n}, t_n) \in \mathcal{Q}^{h,A}$ and $(s_{k_l^n-1/2}, t_n) \notin \mathcal{Q}^{h,A}$ holds, and there exists $k_r^n \in \mathcal{K}$ such that $(s_{k_r^n}, t_n) \in \mathcal{Q}^{h,A}$ and $(s_{k_r^n+1/2}, t_n) \notin \mathcal{Q}^{h,A}$ holds. The indices k_l^n, k_r^n will always appear in a way that the context allows to read at which time level they are addressed, thus we omit the superscript argument and call (s_{k_l}, t_n) the first active tuple and (s_{k_r}, t_n) the last active tuple for the time level t_n .

To conclude this section we give some further remarks on our discrete approach.

Remark 4.5 (Motivation for the staggered approach). *So far the structure of the numerical formula and truncation error of a node and edge equation are analogously, the assignment of node and edge type seems random. The full potential of this staggered approach unfolds due to the fact that for the jet model (and most likely many others) the node/edge assignment of the unknowns and equations can be done in such a way that a_j, g_j solely depends on unknowns of type \mathcal{D}_j and f_j solely depends unknowns of the opposite type. We call the assignment then ideally staggered. The consequence is that no interpolation in (4.4) is necessary and the approximation of the spatial flux is central using only direct neighboring values (interpreted as finite differences we obtain a narrow, central stencil).*

Remark 4.6 (Higher order methods). *Note that there exist ideas in literature for higher order implicit time integrations with special consideration of hyperbolic conservation laws and stiff systems, e.g. [30] study schemes that are strong stability preserving (SSP) and therefore total variation diminishing (TVD). A scheme that is TVD enables sharper shock predictions without creating spurious oscillations [48]. The implicit Euler method is always SSP, but higher order methods have restrictions on the applicable time step. The upper bound for the time step is defined with the help of the radius of absolute monotonicity [68],*

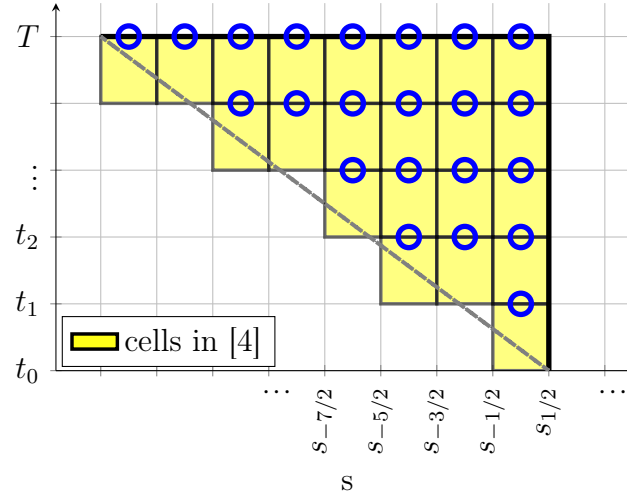


Figure 4.6: Illustration of the zig-zag approximation of the space-time domain.

which is finite for all higher order Runge Kutta methods [66]. The presented Finite Volume scheme has second order flux approximations in space, thus a method that is at least second order in time seems suitable. The Radau IIA method with stage $s = 2$ comes to mind, yet the above restriction on the time step requires investigation. Numerically optimal in terms of the time step restriction are the singly diagonally implicit Runge-Kutta (SDIRK) methods of second and third order, which are presented in [66] and based on [41] and the diagonally implicit Runge-Kutta (DIRK) methods [1]. Nevertheless, we stick to the implicit Euler's method in this work.

Remark 4.7 (Comparison to Arne et al. (2015)). In contrast to the approach in this work, we use a first-order Finite Volume scheme in space on a non-staggered grid with upwinding flux approximations in [5]. After the semi-discretization in space, the resulting DAE system is integrated in time with the stiffly accurate Radau IIA method. Another difference is in handling of time-dependent domains, more precisely in the approximation of the slanted boundary. If we interpret the approach in [5] with our space-time cells in Figure 4.5, the boundary is approximated by a zig-zag line that solely moves in the t or s direction (cf. Figure 4.6). That results in "jumping" boundary values, which imposes additional difficulties onto the numerical solution. Furthermore, the approach in [5] is restricted to $\mathcal{L}(t) = t$, $\mathcal{R}(t) = 0$ for the space-time domain.

In [5] a study was made to examine the temporal convergence order. The expected orders could only be achieved for very small Δt , higher order time integration methods showed similar behavior (in particular, the Radau IIA method with stage one and two is used in [5]). The new approach in this work does not suffer from such a convergence order reduction effect, as we will see in Section 5.2.

Remark 4.8 (Alternative cell shapes). The sole restriction in the modeling of the cells being adjacent to a slanted boundary of the domain (introduced as trapezoidal cells) is our wish to decouple the time integration for iterative solving. Geometrically speaking that means that every cell has to have the north face aligned with one of the horizontal t_n lines. The idea of fan-shaped cells becomes apparent because it gives a better distribution of the

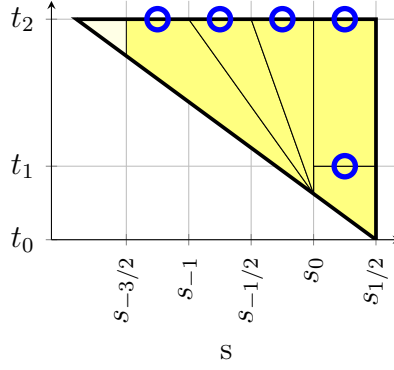


Figure 4.7: Illustration of cells in fan shape.

cell volumina when compared to the trapezoidal shape, this is illustrated in Figure 4.7. The disadvantage of this approach is that in case of multiple fan-shaped cells for one time level all those cells would undergo the same discrete time step. That creates physically unrealistic behavior in the jet, because their actual dynamics influence the system over different timespan. This effect is corrected with our trapezoidal-shaped cells, yet we do not claim that this choice is the only possible one.

4.1.2 Initial and boundary conditions

The model equations are given together with a set of initial and boundary conditions to build a well-defined system. In this section we want to present how they are incorporated in the discrete formula (4.3) and highlight the impact of the choice of the active domain. We state our abstract structure followed by an example and present a hands-on algorithm in Section 4.1.3.

Initial conditions Like already mentioned in Section 4.1.1 we can incorporate the initial conditions directly due to the adjusted time step in (4.3). We assume that initial conditions are given for all $(s, t_0) \in \mathcal{Q}$ in the form

$$y_j(s, t_0) = A_{j'}(s, t_0)$$

with $\mathbf{A}: \Omega(t_0) \times t_0 \rightarrow \mathbb{R}^{M^*}$ and $M^* \leq M$. The indices $j' \in \{1, \dots, M^*\}$ in \mathbf{A} are uniquely assigned to one index j in \mathbf{y} . In the case of time-dependent boundaries $s_a(t)$ and $s_b(t)$ we require initial conditions at the left side for all $(s, t) \in \{(s_a(t), t) \in \mathcal{Q} \mid \frac{d}{dt}s_a(t) < 0, t \in (0, T]\} =: \mathcal{I}_l$ in the form

$$y_j(s_a(t), t) = A_{L,j'}(s_a(t), t),$$

with $\mathbf{A}_L: \mathcal{I}_l \rightarrow \mathbb{R}^{M^*}$ and at the right side for all $(s, t) \in \{(s_b(t), t) \in \mathcal{Q} \mid \frac{d}{dt}s_b(t) > 0, t \in (0, T]\} =: \mathcal{I}_r$ in the form

$$y_j(s_b(t), t) = A_{R,j'}(s_b(t), t),$$

with $\mathbf{A}_R: \mathcal{I}_r \rightarrow \mathbb{R}^{M^*}$. Assume that we want to solve (4.3) at some time level t_n . We set the required non-active $\mathbf{Y}_k(t_{n-1}^k)$ that have $(s_k, t_{n-1}^k) \in \mathcal{Q} \setminus \mathcal{Q}^{h,A}$ with the help of the respective $\mathbf{A}, \mathbf{A}_L, \mathbf{A}_R$.

Remark 4.9. *The initial conditions that are required for the continuous model equations to be well-defined may not cover all unknowns, thus we introduce $M^* \leq M$. To facilitate the notation we decide to extend the image of the initial value functions $\mathbf{A}, \mathbf{A}_L, \mathbf{A}_R$ to \mathbb{R}^M and put the unknowns at their respective index j consistent with the order in \mathbf{Y}_k and use a dummy value for the remaining components, i.e., they are set to some constant $C_{un} \ll 1$, the value is chosen for technical reasons within the implementation.*

Boundary conditions The boundary conditions need more careful examination, in particular there are two effects to be considered: The boundary conditions are only coupled naturally into the system through the fluxes in (4.3) if the active domain is chosen properly, and the staggered grid can require knowledge of the unknowns at the boundary even without corresponding fluxes in the model equation. Before we discuss the first issue in Example 4.11 we address the second one. Examine the numerical formula (4.3), and more importantly, the actual dependence of a_j and q_j on the unknowns in the model equations. If a_j, q_j depend on unknowns of type not equal to \mathcal{D}_j , then we will require interpolation in (4.4) for the evaluation. If said evaluation now takes place on the first or last active tuple, then information of said unknown at the boundary is required for a well-defined discrete system. We call those artificial boundary conditions.

We assume that boundary conditions for all $t \in [0, T]$ are given in the form

$$\mathbf{L}(s_a(t), t, \mathbf{y}(s_a(t), t)) = \mathbf{0}$$

on the left side of the domain and

$$\mathbf{R}(s_b(t), t, \mathbf{y}(s_b(t), t)) = \mathbf{0}$$

on the right side with $\mathbf{L}, \mathbf{R}: \mathbb{R} \times [0, T] \times \mathbb{R}^M \rightarrow \mathbb{R}^M$. They have three ingredients:

- i) The set of boundary conditions that are necessary for the continuous model equations to be well-defined.
- ii) All required artificial boundary conditions. If possible, such values are derived analytically with the help of the given model equations and conditions from i), otherwise an extrapolation boundary condition can ultimately be used – as needed on the left or right side

$$y_j(s_a(t_n), t_n) = \frac{1}{\Delta s_{k_l+1/2}} \left[(s_{k_l+1} - s_a(t_n)) Y_{k_l,j}(t_n) - (s_{k_l} - s_a(t_n)) Y_{k_l+1,j}(t_n) \right],$$

$$y_j(s_b(t_n), t_n) = \frac{1}{\Delta s_{k_r-1/2}} \left[(s_b(t_n) - s_{k_r-1}) Y_{k_r,j}(t_n) - (s_b(t_n) - s_{k_r}) Y_{k_r-1,j}(t_n) \right].$$

We use $E_{l,j}(t_n)$ and $E_{r,j}(t_n)$, respectively, to address such an extrapolation boundary from now on.

- iii) All other components of $\mathbf{y}(\cdot)$ that are not required by the discrete scheme are set to dummy values. This is already anticipating the thoughts of Remark 4.9 and is solely done to facilitate the notation.

Remark 4.10. *Certainly it would be possible to alter the discrete formula (4.3), when adjacent to the boundary, in a way that no artificial boundary conditions are necessary in the first place. We decide to use them in order to maintain the generality and simplicity of our scheme. With our approach those artificial boundary conditions can be easily replaced, e.g. the linear extrapolation in ii) with a higher order one.*

We solve the boundary conditions as separate equations in the non-linear system to determine the boundary unknowns $\mathbf{Y}_L^n, \mathbf{Y}_R^n \in \mathbb{R}^M$, i.e., when solving for the time level t_n we add the equations

$$\mathbf{L}(s_a(t_n), t_n, \mathbf{Y}_L^n) = \mathbf{0}, \quad \mathbf{R}(s_b(t_n), t_n, \mathbf{Y}_R^n) = \mathbf{0} \quad (4.6)$$

to the system. The boundary conditions then enter the discrete formula (4.3) through those non-active $Y_{k_l-1/2,j}(t_n)$ and $Y_{k_r+1/2,j}(t_n)$ that are required in (4.3), we interpolate them linearly with the help of the boundary unknowns

$$\begin{aligned} Y_{k_l-1/2,j}(t_n) &= \frac{s_{k_l} - s_{k_l-1/2}}{s_{k_l} - s_a(t_n)} Y_{L,j}^n + \frac{s_{k_l-1/2} - s_a(t_n)}{s_{k_l} - s_a(t_n)} Y_{k_l,j}(t_n), \\ Y_{k_r+1/2,j}(t_n) &= \frac{s_{k_r+1/2} - s_{k_r}}{s_b(t_n) - s_{k_r}} Y_{R,j}^n + \frac{s_b(t_n) - s_{k_r+1/2}}{s_b(t_n) - s_{k_r}} Y_{k_r,j}(t_n). \end{aligned} \quad (4.7)$$

We mentioned earlier that the choice of the active domain has an impact on the coupling of the boundary conditions. The following example illustrates that and also shows when artificial boundary conditions are required.

Example 4.11 (Choice of node and edge). Let the following system of equations

$$\partial_t y_1 + \partial_s y_2 = 0, \quad (4.8a)$$

$$\partial_t y_2 = 1, \quad \text{on } \mathcal{Q} = [0, 1] \times [-1, 0] \quad (4.8b)$$

$$\mathbf{y}(s, 0) = \mathbf{y}^i(s), \quad s \in [-1, 0] \quad (4.8c)$$

$$y_2(0, t) = y^r(t), \quad t \in [0, 1] \quad (4.8d)$$

for the unknowns $\mathbf{y} = (y_1, y_2)$ be given. The general functions used in (4.1) are

$$\mathbf{a}(\mathbf{y}) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad \mathbf{f}(\mathbf{y}) = \begin{pmatrix} y_2 \\ 0 \end{pmatrix}, \quad \mathbf{g}(\mathbf{y}) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

In an ideal case no interpolation of the spatial averages and no artificial auxiliary conditions are necessary in the discrete version of System (4.8). One key feature that should be ensured is the narrow central approximation of the fluxes (important for a conservative approximation), thus we know with (4.8a) that y_1 and y_2 have to have different types. Since the meaning of node/edge is interchangeable it is enough to consider one case. We choose y_1 to be of node type and y_2 to be edge type, the choice vector is $\mathbf{D} = (1, 0)$. With an equidistant grid in time and space the discrete version of (4.8a) at an arbitrary active node (s_i, t_n) within the domain is

$$\Delta s (Y_{i,1}(t_n) - Y_{i,1}(t_{n-1})) + \Delta t (Y_{i+1/2,2}(t_n) - Y_{i-1/2,2}(t_n)) = 0 \quad (4.9a)$$

and the one of (4.8b) at an arbitrary active edge $(s_{i+1/2}, t_n)$ within the domain is

$$\Delta s (Y_{i+1/2,2}(t_n) - Y_{i+1/2,2}(t_{n-1})) - \Delta s \Delta t = 0. \quad (4.9b)$$

Looking back at (4.4) it can be seen that no interpolation of the spatial averages is necessary. The initial values are set for all $(s_k, 0) \in \mathcal{Q}^h$ by

$$\mathbf{Y}_k(0) = \mathbf{A}(s, 0) \quad \text{with} \quad \mathbf{A}(s, 0) = \mathbf{y}^i(s).$$

For the discussion of the boundary conditions the key aspect is the choice of the boundary types p_l and p_r . We have two cases on each side:

- $p_l = 0$: The first active tuple is a node, meaning we solve (4.9a) there and it requires $Y_{k_l-1/2,2}(\cdot)$, which is a non-active tuple. Since no boundary condition for it is given, we use extrapolation

$$\mathbf{L}(-1, t_n, \mathbf{Y}_L^n) = \begin{pmatrix} Y_{L,1}^n - C_{un} \\ Y_{L,2}^n - E_{l,2}(t_n) \end{pmatrix}.$$

We want to mention once here that the extrapolation condition is an artificial boundary conditions and ensures that the numerical system stays uniquely solvable. It is not necessary for the fully continuous model equations to be well-defined though. It is only necessary due to our staggered approach.

- $p_l = 1$: The first active tuple is an edge, meaning that we solve (4.9a) on the active node to the right of it, i.e., the boundary values never enter the discrete system and are simply the dummy values

$$\mathbf{L}(-1, t_n, \mathbf{Y}_L^n) = \begin{pmatrix} Y_{L,1}^n - C_{un} \\ Y_{L,2}^n - C_{un} \end{pmatrix}.$$

- $p_r = 0$: The last active tuple is a node, meaning we solve (4.9a) there. The boundary condition for y_2 couples naturally through the spatial flux that requires $Y_{k_r+1/2,2}(t_{n+1})$, which in turn requires $Y_{R,2}^n$ in (4.7). We set

$$\mathbf{R}(0, t_n, \mathbf{Y}_R^n) = \begin{pmatrix} Y_{R,1}^n - C_{un} \\ Y_{R,2}^n - y^r(t) \end{pmatrix}.$$

- $p_r = 1$: The last active tuple is an edge, meaning that we solve (4.9a) on the active node left of it. The boundary conditions for y_2 never couples through the spatial flux of (4.9a) and the system is underdetermined. A coupling can be induced by replacing (4.9b) of the last active edge with an extrapolation condition that includes the boundary unknown, namely

$$E_{r,2}(t_n) - Y_{R,2}^n = 0.$$

The boundary conditions are then set

$$\mathbf{R}(0, t_n, \mathbf{Y}_R^n) = \begin{pmatrix} Y_{R,1}^n - C_{un} \\ Y_{R,2}^n - y^r(t) \end{pmatrix}.$$

Note that this is comparable to choosing $p_r = 0$ from the start in terms of distance of the discrete unknowns to the boundary.

Considering all cases we recommend to use the one with the least interpolation and extrapolations. The suitable pick is $p_l = 1$, $p_r = 0$, i.e., \mathcal{Q}^{1-0} .

Example 4.11 demonstrates the synergies when choosing the types of unknowns and the active domain. The larger the system the more complex this choice becomes and careful attention is needed.

4.1.3 Algorithm

Assume that well-defined model equations on a space-time domain with their initial and boundary conditions are given. In order to prepare the model for our discrete approach, the following steps are necessary:

1. Formulate the system as a first-order system in the form (4.1) with the help of $\mathbf{a}, \mathbf{f}, \mathbf{g}$.
2. Define node and edge unknowns and the unknown-equation assignment with the goal of small stencils in the derivatives and minimal amount of interpolation in general.
3. Define the types of the boundaries $p_l, p_r \in \{1, 0\}$ – and therewith the active domain $\mathcal{Q}^{h,A}$ – to properly couple the given boundary conditions.
4. Assess the need for artificial boundary conditions and express the complete initial and boundary conditions with the help of $\mathbf{A}, \mathbf{A}_L, \mathbf{A}_R$ and \mathbf{L}, \mathbf{R} .

The prepared model then is ready for our discrete approach. For a condensed description we introduce a notation for all active nodes and edges of the time levels t_n , $n \in \mathbb{N}$:

$$\Omega^h(t_n) = \{s_k \in \Omega(t_n) \mid (s_k, t_n) \in \mathcal{Q}^{h,A}, k \in \mathcal{K}\}.$$

The steps of the iterative solver for the numerical approximation are listed in the following.

Initialization

- i) Choose the equidistant spatial grid spacing Δs and a default Δt and set $t_0 = 0$ and $n = 0$.
- ii) Iterate over all k with $s_k \in \Omega(t_0)$ and set $\mathbf{Y}_k(t_0) = \mathbf{A}(s_k, t_0)$.
- iii) Find t_1 such that $|\Omega^h(t_1)| > 0$ and $t_1 - t_0$ is a multiple of Δt .

Time Stepping

- i) Set $n := n + 1$. In case of no time step control, set $t_n = t_{n-1} + \Delta t$, otherwise determine t_n . Set $\Delta t_n = t_n - t_{n-1}$.
- ii) Determine the spatial boundaries $s_a(t_n)$ and $s_b(t_n)$.
- iii) Initialize $Y_{k,j}(t_{n-1}^k)$, if necessary, and evaluate any explicitly used information.

Iterate over all k with $s_k \in \Omega^h(t_n)$ and $s_k \notin \Omega^h(t_{n-1})$.

Set $\mathbf{Y}_k(t_{n-1}^k) = \mathbf{A}_L(s_k, t_{n-1}^k)$ on the left side and $\mathbf{Y}_k(t_{n-1}^k) = \mathbf{A}_R(s_k, t_{n-1}^k)$ on the right side.

- iv) Determine the unknowns and equations of time level t_n and build a non-linear system.
 - a) Add every active unknown $Y_{k,j}^n$ and their equation (4.3) with $s_k \in \Omega^h(t_n)$ and $\mathcal{D}_j = \mathcal{T}(k)$.
 - b) Add the boundary unknowns \mathbf{Y}_L^n and \mathbf{Y}_R^n and their equations (4.6).
- v) Solve the non-linear system with Newton's method. If $t_n = T$, terminate. Otherwise goto i).

4.1.4 Solution reconstruction

A solution obtained with our algorithm from Section 4.1.3 is piecewise constant on the underlying grid. It essentially has two parts for every time level t_n : The unknowns \mathbf{Y}_L^n , \mathbf{Y}_R^n and all $Y_{k,j}^n$ with k, j such that $\mathcal{D}_j = \mathcal{T}(k)$ and $(s_k, t_n) \in \mathcal{Q}^{h,A}$, as well as the underlying spatial grid. We address such a solution with \mathbb{Y}^n . We introduce the piecewise constant function $Z_j(\cdot; \mathbb{Y}^n): \Omega(t_n) \rightarrow \mathbb{R}$ as follows:

$$Z_j(s; \mathbb{Y}^n) = \begin{cases} Y_{L,j}^n & \text{if } s \in [s_a(t_n), s_{k_l-1/2}), \\ Y_{k,j}^n & \text{if } s \in [s_{k-1/2}, s_{k+1/2}) \text{ and} \\ & k \text{ such that } s_k \in \Omega^h(t_n) \text{ and } \mathcal{D}_j = \mathcal{T}(k), \\ Y_{R,j}^n & \text{if } s \in [s_{k_r+1/2}, s_b(t_n)]. \end{cases}$$

Note that all solution specific components in $Z_j(\cdot; \mathbb{Y}^n)$, e.g., s_a and $\Omega^h(t_n)$, are associated to the solution \mathbb{Y}^n . To compare solutions of the same model problem with each other we need to reconstruct them if they have different spatial grids. Assume now that two solutions \mathbb{Y}_1^n and \mathbb{Y}_2^n at some time t_n for the same model problem are given. We use the L^2 -error on our staggered grid since the discrete unknowns represent cell averages and define

$$\text{err}(\mathbb{Y}_1^n, \mathbb{Y}_2^n, \mathcal{J}) = \sqrt{\sum_{s_k \in \Omega_n^{h,\text{ref}}} \int_{s_{k-1/2}}^{s_{k+1/2}} \sum_{j \in \mathcal{J}} (Z_j(s; \mathbb{Y}_1^n) - Z_j(s; \mathbb{Y}_2^n))^2 ds} \quad (4.10)$$

with $\mathcal{J} \subset \{1, \dots, M\}$ as the list of unknowns to be compared. The appearing integral over the piecewise constant function is evaluated exactly. The reference grid $\Omega_n^{h,\text{ref}}$ is either the node grid or the edge grid and taken from one of the solutions, preferably the one with the coarser grid. The node ($p = 1$) and edge grid ($p = 0$) is given by

$$\Omega_{h,p}^n = \{s_k \in \Omega(t_n) \mid \mathcal{T}(k) = p, s_k \in \Omega^h(t_n)\}.$$

In the remaining part of this work we will address the error (4.10) as the $L^2(t_n)$ -error, the context will allow to determine which solutions and components are compared.

Node/edge assignment of the jet model

| unknowns | $\bar{\mathbf{r}}$ | $\bar{\boldsymbol{\lambda}}_\tau$ | \mathbf{q} | $(\boldsymbol{\lambda}_\kappa, \lambda_s)$ | $\boldsymbol{\tau}$ | $\boldsymbol{\kappa}$ | λ_t | χ | \mathbf{v} | $\boldsymbol{\omega}$ | \mathbf{n} | \mathbf{m} | u |
|--------------------|--------------------|-----------------------------------|--------------|--|---------------------|-----------------------|-------------|--------|--------------|-----------------------|--------------|--------------|-----|
| equation in (2.37) | a | b | c | d | e | f | g | h | i | j | k | l | - |
| type | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

Table 4.1: Type and equation assignment of the jet model (2.31) with the reduced unknowns $\bar{\mathbf{r}}$, $\bar{\boldsymbol{\lambda}}_\tau$, \mathbf{q} , $\boldsymbol{\lambda}_\kappa$, λ_s , $\boldsymbol{\tau}$, $\boldsymbol{\kappa}$, λ_t , χ , \mathbf{v} , $\boldsymbol{\omega}$, \mathbf{n} , \mathbf{m} , u .

4.2 Application to the jet model

The application of the jet model to the discrete scheme is fairly straightforward. In this section we present the choice of node/edge type for the equations and unknowns and a reduction of unknowns that is easily obtainable through the proper choice of typical values. We will also comment on the impact of our chosen kinematics formulation and the choice of the active domain for the set-ups of Section 2.3.

Node/edge assignment and reduction of unknowns The jet model from Section 2.2.5 can be put in the general PDAE form (4.1) with the help of a simple substitution for the spatial derivatives of $\boldsymbol{\lambda}_\tau$ and $\boldsymbol{\lambda}_\kappa$, the other formulations of the model with respect to its kinematics are analogously (see Appendix D.2 for details). Before talking about node/edge type assignments we want to highlight an obvious reduction of unknowns without restricting the applicability of the model. The dimensionless mass and volume line density σ_M and σ_V as well as k are all conserved type-1 quantities, which will all be chosen to be node type. We always assume that they are independent of s in the initial values and scaled to one in the Lagrangian parameterization through their typical values. That means that σ_M , σ_V and k are easily expressed with the spatial functional determinant χ according to the transformation rule (2.14). Consequently all three are replaced with χ in the jet model, which is a conserved quantity itself and thus is determined by its conservation equation (more details are given in Remark D.3). The reduced unknowns of the jet model and their node/edge type assignments are listed in Table 4.1. The unknown order is analogously to the order of the assigned equations of (2.37). Note that the three conservation equations for σ_M , σ_V and k are replaced with one for χ . For completeness, the condensed jet model is printed with unknown/equation and type assignment in (D.6) together with further reasoning for the choice in Table D.1. The type assignment reduces the amount of interpolations to a minimum, nevertheless an ideally staggered assignment according to Remark 4.5 is not possible. Interpolation is still required in the spatial fluxes of the artificial convection u and some source terms, e.g. the term $\boldsymbol{\tau} \times \mathbf{n}$ in the angular momentum balance.

Remarks on the kinematics formulation The above node/edge assignment is analogously transferred to the M and T formulation of the jet model. For the S formulation difficulties arise which are described in Remark 4.12. Furthermore we want to clarify our choice of correction terms in the SAMW correction in Remark 4.13.

Remark 4.12 (Difficulties with the S-formulation). *The jet model in System 2.31 with kinematics in the S-formulation (cf. Remark 2.2) has no node/edge type assignment of the unknowns and equations that avoids interpolation in the spatial fluxes. The system becomes highly unstable due to the decoupled broad stencils in the fluxes and has to be discarded. The stabilized formulation of the kinematics with GGL correction can be seen as an S-formulation with an added linearly dependent equation to repair the node/edge assignment problem (cf. Remark 2.28). This has to be treated with care since it creates another inconsistency, which is described in Remark 4.13.*

Remark 4.13 (Necessity of the SAMW correction). *Let the stabilized formulation of the kinematics be given (cf. Formulation 2.22) with the GGL correction, or in that matter, any correction that uses $\Lambda_{r,2} \equiv \mathbf{0}$, $\Lambda_{q,2} \equiv \mathbf{0}$. The Lagrangian multipliers $\bar{\lambda}_1$ and λ_2 are of edge type and appear in the node equation (2.37a) and (2.37c). Without correction terms in the compatibility conditions, those equations solely determine the multipliers (cf. Remark 2.28), or in other words, a node equation determines an edge unknown. This is only possible if there is an equal amount of node unknowns and edge unknowns, otherwise the system becomes under- or overdetermined. The jet extrusion set-up in Section 2.3.1 has an equal amount of unknowns and thus allows the usage of the GGL correction. The other two set-ups (Section 2.3.2 and 2.3.3) in contrast are not solvable with it. The SAMW correction overcomes this issue by introducing additional correction terms in the compatibility conditions, i.e., an edge equation. This also introduces an error in the compatibility conditions. To keep this error to a minimum we set the constant that appears in (2.37e) and (2.37f) to $C = \Delta t \Delta s$. The correction terms are then of order $\mathcal{O}(\Delta t^2 \Delta s^2)$ in (4.3).*

Choice of active domain for the set-ups The subsequent step for the numerical scheme is to ensure that the boundary conditions are directly incorporated at all times, especially for the time-dependent domain where cells will be added. All set-ups are similar: We have a combination of the position and tension boundary. Consider the position boundary. We have e.g. a boundary condition for the jet curve. It appears in the spatial flux of (2.37b), which is edge type. Consequently, the numerical formula of (2.37b) only couples to the boundary if the adjacent active cell is of edge type, meaning the boundary has to be node type. Now consider the tension boundary. We have boundary conditions, e.g., for the contact forces and couples, which are needed for the fluxes of the balance laws (2.37i) and (2.37j), which are both node type. Consequently we have a coupling if the adjacent active cell is of node type, i.e. the boundary has to be edge type. Taking the above discussion into account we analyze the four possible combinations of the boundaries and present suitable choices of boundary types in Table 4.2. The initial and boundary conditions of the jet model for all set-ups are set according to Section 2.3. Our choice of unknowns requires artificial boundary conditions for χ at a position boundary and for $\boldsymbol{\tau}$, $\boldsymbol{\kappa}$ at a tension boundary. We set them with the help of the initial conditions for χ and use linear extrapolation for $\boldsymbol{\tau}$, $\boldsymbol{\kappa}$ (an alternative is described in Remark 6.2). The resulting functions \mathbf{A} , \mathbf{A}_L , \mathbf{A}_R and \mathbf{L} , \mathbf{R} of the set-ups are presented in Appendix D.2.

Solution comparison with general parameterization The spatial grid of the jet model (2.37) is with respect to the general parameterization. A feasible comparison and inter-

| set-up | boundary | | suitable active domain |
|---------------|----------|----------|------------------------|
| | left | right | |
| Section 2.3.1 | position | tension | \mathcal{Q}^{1-0} |
| Section 2.3.2 | tension | tension | \mathcal{Q}^{0-0} |
| Section 2.3.3 | position | position | \mathcal{Q}^{1-1} |

Table 4.2: Overview of suitable active domains for the jet model with the node/edge assignment from Table 4.1.

pretation of solutions is only possible if they have the same parameterization. Since the model was originally formulated in the Lagrangian parameterization we use it for the comparison. For that purpose we transform the unknowns of a solution \mathbb{Y}^n to the Lagrangian parameterization (if not already given in it) according to their type, use the corresponding Lagrangian grid for Z_j and determine the $L^2(t_n)$ -error. The Lagrangian grid is given by $\{\sigma_k \in \mathbb{R} \mid \sigma_k = \Psi(s_k, t_n), s_k \in \mathcal{Q}^h\}$. The transformation Ψ can be reconstructed with help of one of the interval boundaries $\sigma_a(t), \sigma_b(t)$ and $\partial_s \Psi = \chi$, which we assume is given with the solution. We denote the solution in Lagrangian parameterization with $\hat{\mathbb{Y}}^n$.

4.3 Application to r -refinement strategies

The jet model is extended by the respective equations of the moving mesh strategy from Section 3.1. Depending on the used set-up, the additionally required boundary conditions are either directly given or can be set with one of the equations in (2.44). We rewrite the moving mesh strategies as systems of first order to fit the form (4.1). Both DELAX strategies require the distribution α for which we use a mid-point quadrature rule. The mesh control function M, f_α of the moving mesh strategies are complicated and dependent on derivatives of solution components, furthermore they are spatially smoothed. Thus we decide to evaluate them explicitly, which will create an additional temporal lag in the movement of the mesh similar to the one induced by the temporal relaxation parameter τ . We use finite difference approximations for the derivatives appearing in M, f_α , in particular a second-order central approximation within the domain and first-order, one-sided one at the boundaries. For the spatial smoothing technique we follow [58], see Appendix C.2 for details. The type of the parameter speed u is edge according to Table 4.1. We summarize the equations of the strategies below. Note that all strategies use the temporal relaxation parameter τ , which is not to be confused with the distortion measure $\boldsymbol{\tau}$.

- Strategy 3.11, DELAX1 is combined with a coupling condition for u

$$\begin{aligned}\partial_t \Psi &= -\chi u, \\ \chi u &= \frac{1}{\tau} \left(\Psi - \Psi \circ \alpha^{-1} \Big|_{p=s} \right)\end{aligned}$$

using the parameter speed u and Ψ (edge type). The distribution α is inverted numerically using linear interpolation.

- Strategy 3.13, DELAX2 is used in terms of the parameter speed u , i.e., its equation is

$$u = -\frac{1}{\tau} (s - \alpha).$$

- Strategy 3.9, MMPDE1 is rewritten as a first order system that is given by

$$\begin{aligned} \partial_t \chi &= \chi^t, \\ \partial_s (M (\tau \chi^t + \chi)) &= 0 \end{aligned}$$

with a substitution unknown χ^t (node type) and parameter speed u .

- Strategy 3.10, MMPDE2 is used in terms of the parameter speed u , i.e., its equation is

$$-M \chi \tau u = \partial_s (M \chi).$$

5 Convergence and performance

In Chapter 2 we presented our stabilized, index-reduced formulation of the viscous, incompressible jet model together with versatile set-ups. We proposed a suitable numerical approach in Chapter 4 and gave details on the algorithm. In this section we explore its numerical properties. If not otherwise mentioned all our implementations are done in MatLab 2016a, the underlying solver for the Newton's method is the band solver that is built in the MatLab backslash routine. The simulations were conducted on a workstation with a Intel(R) Xeon(R) CPUs @ 2.40GHz and 24 GB RAM.

We introduce the academic viscous cantilever scenario in Section 5.1 and show the feasibility of our discrete scheme by investigating convergence properties of the scenario in Section 5.2. In particular we are interested in a comparison of the influence of the boundary handling (time-independent and time-dependent domain) and point out the improvements compared to [5]. We also discuss numerical difficulties arising from the M- and T-formulation of the jet kinematics that ultimately lead us to our stabilized, index-reduced formulation. In Section 5.3 we comment on our solver with focus on the condition number of the underlying linear system. Afterwards we shift the focus to the investigation of the general parameterization in Section 5.4 and in particular its impact on the overall computational performance and its adaptive possibilities, e.g. when combined with the mesh refinement strategies from Chapter 3.

5.1 Academic scenario: Viscous cantilever

In the elastic context, the cantilever is a prominent example. The rod is fixed at one side and exposed to gravity that is acting normal to the nozzle direction as the sole external load, cf. Figure 5.1. We design an analogous viscous scenario that serves as an academic example. For that purpose we use the jet model (2.37) together with the external loads

$$\mathbf{f} = -\frac{1}{\text{Fr}^2}\sigma_M\mathbf{R}(\mathbf{q}) \cdot \mathbf{e}_2, \quad \mathbf{l} = 0$$

and employ the jet extrusion set-up in Section 2.3.1 with

$$\begin{aligned} \bar{\mathbf{r}}^A(s) &= (s + \mathcal{L}_0, 0, 0), & \mathbf{q}^A(s) &= \frac{1}{2}(1, -1, -1, -1), \\ \bar{\mathbf{r}}^L &= (0, 0, 0), & \mathbf{q}^L &= \frac{1}{2}(1, -1, -1, -1), \end{aligned}$$

meaning that $\mathbf{R}(\mathbf{q}^L)^T = (\mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_1)$, and consider two cases:

- a) Time-independent case with $l(t) = l_0 = 1$ and $r(t) = 0$, i.e. $v^A(s) = v^L = 0$. We call this the fixed viscous cantilever.

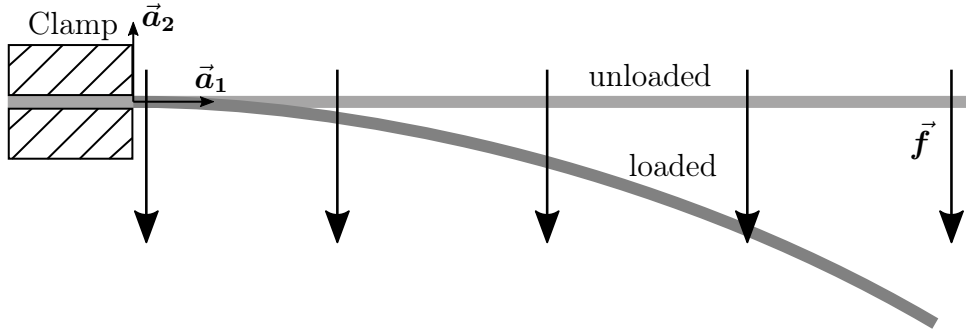
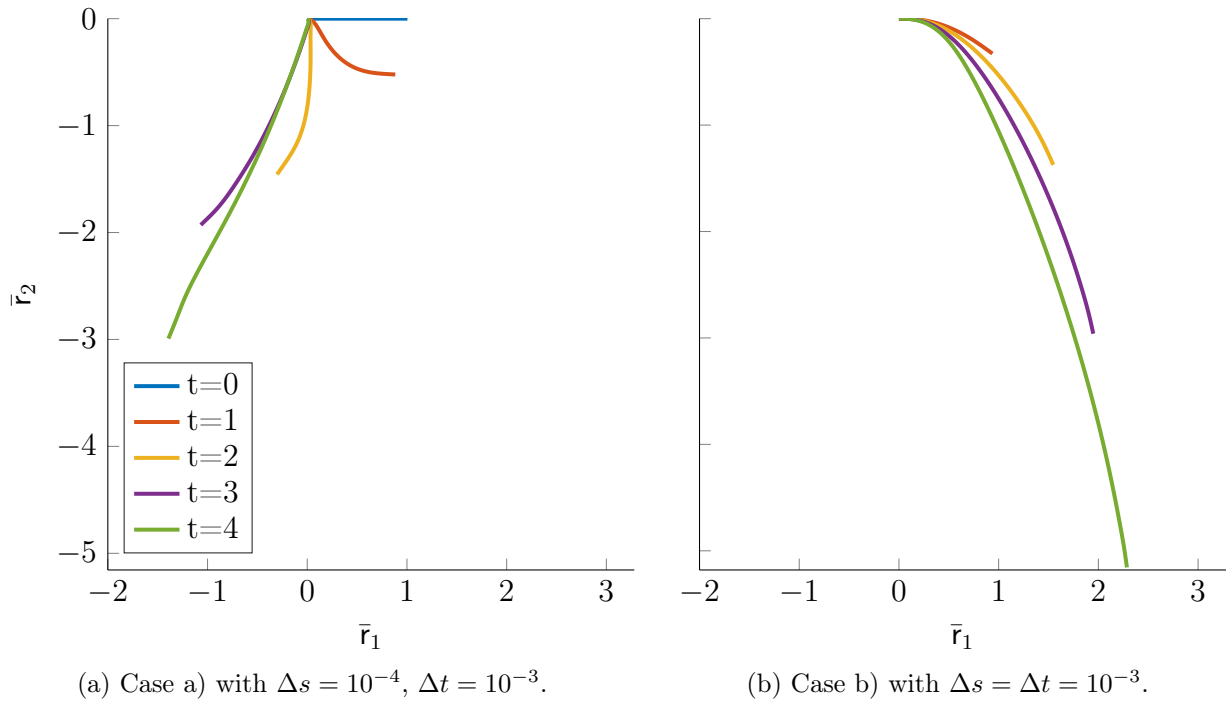


Figure 5.1: Illustration of the classical cantilever in the elastic context.


 Figure 5.2: Simulation of a viscous cantilever with $(\text{Re}, \text{Fr}, \varepsilon, T) = (1, 1, 8.86 \cdot 10^{-2}, 4)$ depicted at interim time points (the value of ε is $\sqrt{\pi}/2 \cdot 10^{-1}$, cf. Remark 2.17).

- b) Time-dependent case with $l(t) = t$, $r(t) = 0$, i.e. $l_0 = 0$ and $v^A(s) = v^L = 1$. We call this the growing viscous cantilever.

Using a Lagrangian parameterization ($u \equiv 0$) we illustrate the jet behavior for both cases in Figure 5.2. In Figure 5.2a the fixed cantilever is shown. The total mass remains constant, yet the jet is stretched while falling down. In Figure 5.2b the growing cantilever is shown. The total mass is constantly increasing due to the inflow, the total length accounts for the mass inflow and the stretching. Notice the difference in Δs , the fixed cantilever case requires a higher spatial resolution due to the large elongation in proximity of the nozzle.

5.2 Convergence order

Our proposed finite volume method employs a second order flux approximation in space and a first order one in time that are both appropriate for the respectively underlying DAEs of the viscous jet model (2.37) (cf. Section 2.2.4), the expected convergence order is one in time and two in space. We use the viscous cantilever scenario from Section 5.1 because it allows the qualitative comparison of a time-independent and time-dependent domain. The computed convergence orders for one set of the physical parameters are depicted in Figure 5.3 and Table 5.1, certainly the convergence properties of the scheme are not influenced by the physical parameters. The theoretical results are partially confirmed: The temporal convergence rate is matched whereas the spatial one is not. For the spatial convergence rate we suspect that the influence of the source term approximation hinders a fully second order convergence order. Regarding the linear interpolation in (4.4) from node to edge position and vice versa, simply exchanging the method with cubic interpolation in the inside and quadratic interpolation at the boundaries (including (4.7)) does not yield an overall second order in space, neither for the fixed or growing domain case. Thus the linear interpolation is not the source of the worsened computed convergence order, which points to the approximation of the source terms as cause. The discrete geometric approach of Audoly et al. ([12], Remark 5.1) can only achieve first order as well. Further investigations are left to future work. We compare the time-independent and time-dependent case and deduct that our approximation of the space-time domain (with slanted cells) does not significantly deteriorate the global error of the solution. Furthermore, the viscous jet model with our new stabilized, index-reduced kinematics formulation contains Lagrangian multipliers, that are analytically zero. This is confirmed for $\Delta s, \Delta t \rightarrow 0$, giving the numerical equivalency of our formulation, cf. Lemma 2.29.

In [5] we use the jet model with an index-reduced, but not stabilized kinematic formulation and combine it with a Finite Volume method for the semi-discretization in space (first order, upwind flux approximations) and a Radau IIA-method in time (with stage one to obtain a first order method (Euler's method) and stage two for a third order method). The investigations in [5] of the convergence order are done with a two-dimensional rotational spinning scenario (cf. Section 6.1 for the process and Appendix B.2 for the dimensional reduction). Other than the drift-off introduced with the kinematic formulation, the scheme in [5] showed a worrisome temporal order reduction effect that rendered approximations for higher convergence orders useless. Regarding the underlying DAE-character of the jet model and handling the approximation of the space-time domain appropriately (cf.

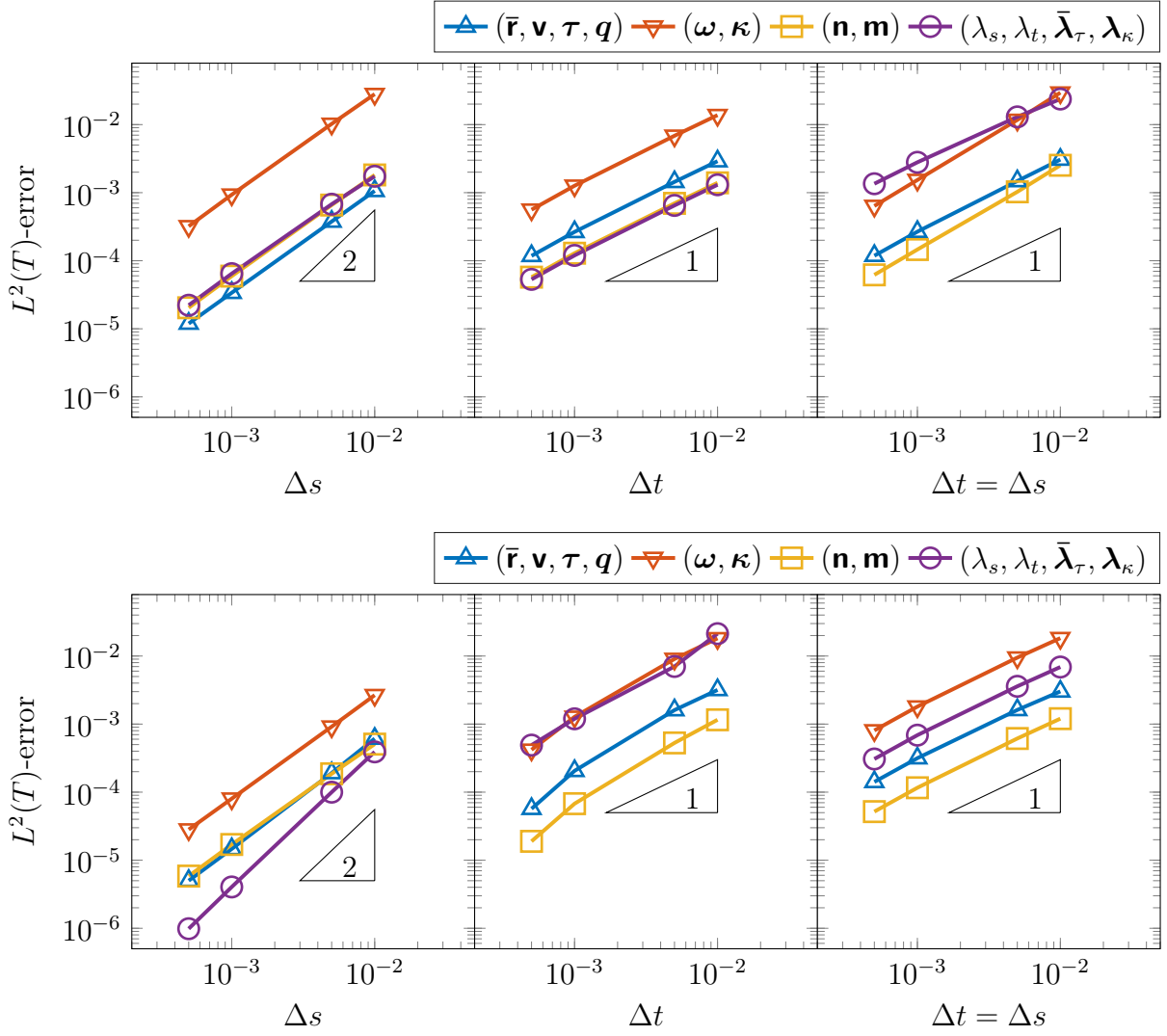


Figure 5.3: From left to right: Spatial, temporal and combined convergence behavior for the cantilever scenario with $(\text{Re}, \text{Fr}, \varepsilon, T) = (1, 1, 8.86 \cdot 10^{-2}, 0.5)$ and default $\Delta t = \Delta s = 10^{-2}$. The unknowns are separated in differential $(\bar{\mathbf{r}}, \mathbf{v}, \boldsymbol{\tau}, \mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\kappa})$ and algebraic ones $(\mathbf{n}, \mathbf{m}, \lambda_s, \lambda_t, \bar{\lambda}_\tau, \lambda_\kappa)$ and split in terms of similar behavior. Top: Time-independent case $(\mathcal{L}(t), \mathcal{R}(t)) = (1, 0)$. Bottom: Time-dependent case $(\mathcal{L}(t), \mathcal{R}(t)) = (t, 0)$. Numerically found orders are listed in Table 5.1.

| Viscous cantilever | | $(\bar{\mathbf{r}}, \mathbf{v}, \boldsymbol{\tau}, \mathbf{q})$ | $(\boldsymbol{\omega}, \boldsymbol{\kappa})$ | (\mathbf{n}, \mathbf{m}) | $(\lambda_s, \lambda_t, \bar{\lambda}_\tau, \lambda_\kappa)$ |
|--------------------|----------|---|--|----------------------------|--|
| fixed case | spatial | 1.50 | 1.54 | 1.55 | 1.54 |
| | temporal | 1.17 | 1.17 | 1.17 | 1.17 |
| | combined | 1.17 | 1.27 | 1.24 | 1.23 |
| growing case | spatial | 1.58 | 1.51 | 1.5 | 2.00 |
| | temporal | 1.42 | 1.31 | 1.42 | 1.15 |
| | combined | 1.05 | 1.06 | 1.07 | 1.06 |

Table 5.1: Spatial, temporal and combined convergence order for the fixed and growing case of the viscous cantilever scenario.

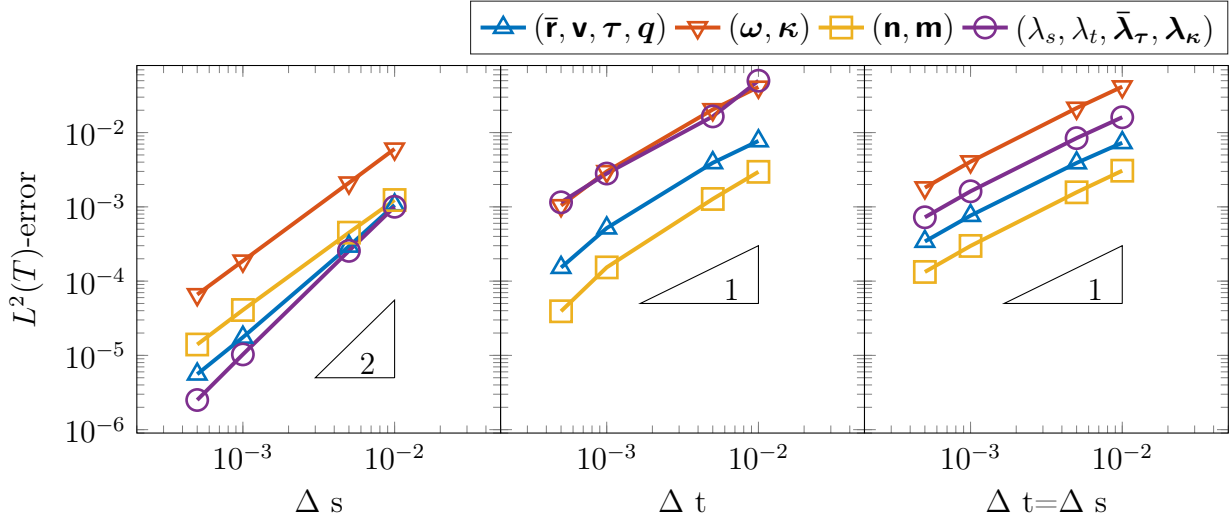


Figure 5.4: From left to right: Spatial, temporal and combined convergence behavior for rotational spinning process with $(\mathcal{L}(t), \mathcal{R}(t)) = (t, 0)$, $(\text{Re}, \text{Rb}, \text{Fr}, \varepsilon, T) = (1, 1, 1, 8.86 \cdot 10^{-2}, 0.5)$ and default $\Delta t = \Delta s = 10^{-2}$. Numerically found orders are listed in Table 5.2.

Remark 4.7), our new scheme and stabilized formulation of the kinematics overcome the hitherto existing restrictions for the two-dimensional scenario and also transfers to the full three-dimensional model. The numerical investigations show spatial, temporal and combined convergence orders analogously to the previous test with the cantilever, see Figure 5.4 and Table 5.2 (we already published these results in [89]).

| Rotational spinning process | $(\bar{\mathbf{r}}, \mathbf{v}, \boldsymbol{\tau}, \mathbf{q})$ | $(\boldsymbol{\omega}, \boldsymbol{\kappa})$ | (\mathbf{n}, \mathbf{m}) | $(\lambda_s, \lambda_t, \bar{\lambda}_\tau, \lambda_\kappa)$ |
|-----------------------------|---|--|----------------------------|--|
| spatial | 1.73 | 1.50 | 1.50 | 2.01 |
| temporal | 1.38 | 1.27 | 1.48 | 1.15 |
| combined | 1.05 | 1.07 | 1.07 | 1.06 |

Table 5.2: Spatial, temporal and combined convergence order of the rotational spinning process. The order is calculated with the three finest solutions, excluding the reference 10^{-4} .

Remark 5.1 (Comparison to Audoly et al. (2013)). *The discrete geometric Lagrangian method in [12] employs a careful spatial discretization of the kinematics of the centerline on a staggered grid and reconstructs the remaining discrete unknowns based on the geometrical and variational principles. The mixed derivatives that appear in the centerline-spin formulation are handled algorithmically by defining local discrete operators for the time derivatives. The time integration is semi-implicit in the viscous forces, in contrast to our fully implicit approach. The discrete geometric Lagrangian method has been validated with experimental data of the fluid-mechanical sewing machine [28, 25] and it shows first order convergence in space-time. In this work we stick to our viscous jet model since it is well suited for (standard) conservative finite volume schemes, which gives great flexibility*

due to the easy exchange of the underlying flux approximations. A direct comparison of the computational performance of both methods is not possible due to a missing benchmark.

Insight into the other kinematics formulations We introduced a index-reduced, stabilized kinematic formulation in the jet model in (2.2.4). Nevertheless, the M- and T-formulations can also be used with our discrete scheme under some restrictions, whereas the S-formulation can not, cf. Remark 4.12:

- The M-formulation has temporal index three, but due to its structure (cf. Section 2.2.4) there exist theoretical convergence results for the implicit Euler's method, which we have employed in our discrete scheme. The M-formulation works well in the fixed domain case, and with $\Delta s \leq v^L \Delta t$ in the growing domain case. Furthermore it is applicable to all three set-ups due to the symmetric boundary conditions. The issue is the temporal convergence of the growing domain case with $\Delta s > v^L \Delta t$. The discrete compatibility conditions are violated in proximity to the nozzle (due to the handling of the boundary cells), which manifests particularly in the contact force and contact couple. They start to fidget when a new discrete point is created and appease after a couple of time steps (see Figure 5.5a). Depending on the actual scenario this effect is more or less developed, yet always visible. In order to fulfill the discrete compatibility conditions our discrete scheme would have to be adjusted in a way that the time steps of all newly initialized points coincide with the time step of the already active cells and their initial values are an extrapolation of the inner points from the previous time step. The consequence is that the actual initial conditions along the slanted space-time boundary are no longer fulfilled exactly, creating a similar fidgeting of the contact force and couple.
- The T-formulation works well with the jet extrusion set-up in Section 2.3.1 in the fixed domain case, and in the growing case if $\Delta s \geq v^L \Delta t$. A sufficiently small time step should be used to minimize the temporal drift-off effect. Nonetheless the T-formulation creates a non-smooth jet curve in the growing domain case if $\Delta s < v^L \Delta t$, i.e. more than one cell is possibly created during one single time step. This issue is consequently transferred to the distortion measures, but most dominantly visible in the curve, as illustrated in Figure 5.5b. This non-smoothness is undesirable, but it seems to have a negligible affect on the robustness of the simulation. Unfortunately the other two set-ups are not applicable because of the non-symmetric boundary conditions.

The SAMW formulation does not show the above problems and is applicable for all set-ups and parameter ranges. We want to mention that all formulations show problems similar to the one above described for the M-formulation when using the discrete scheme according to [5].

Rule of thumb for convergent behavior The stabilized kinematics show numerically proven convergent behavior for $\Delta s, \Delta t \rightarrow 0$ (as do the other formulations). But in industrial scenarios such a test is not feasible in most cases and general answers to the question if a specific simulation already shows convergent behavior are not possible and depend on each

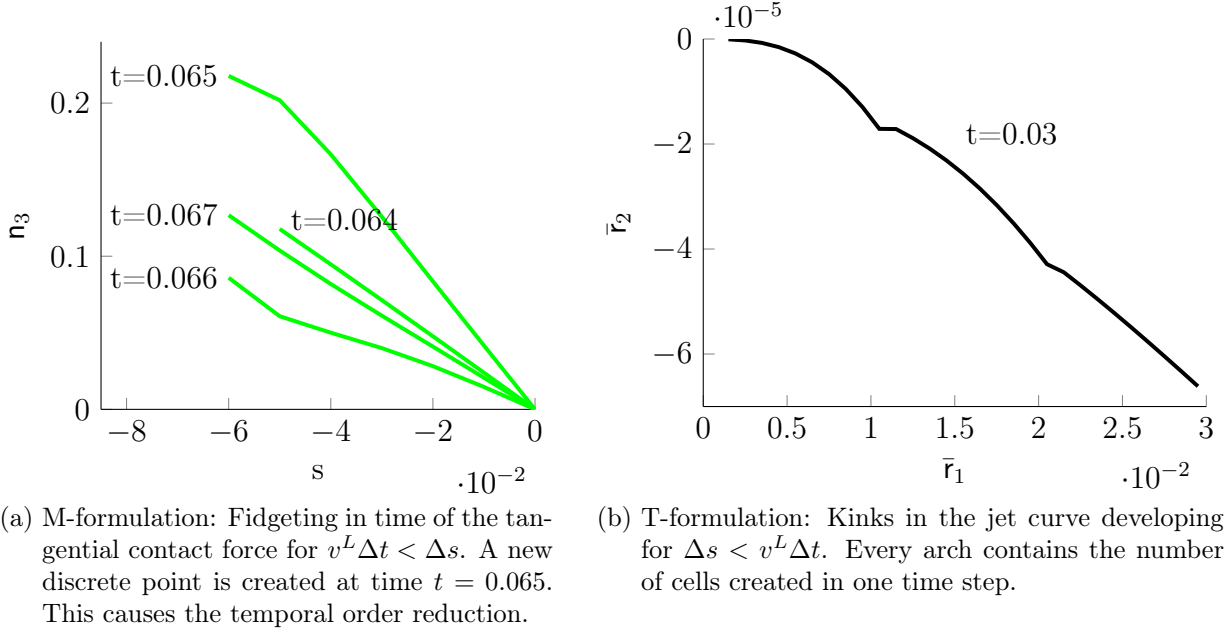


Figure 5.5: Illustration of the critical issues in the M and T-formulation for the growing viscous cantilever with $(\text{Re}, \text{Fr}, \varepsilon) = (0.5, 0.5, 0.01)$ and $\Delta s = 10^{-3}$, $\Delta t = 10^{-2}$, or vice-versa.

case. Nevertheless a rule of thumb can be formulated with the help of the slenderness ratio ε , which correlates with the stiffness of the system. For the purpose of fair comparison between the M-, T- and SAMW-formulation, we choose the growing cantilever scenario with $\Delta s = \Delta t$ and repeat the same simulation with different ε , as illustrated in Figure 5.6. The deviation of the SAMW-formulation to the other two for $\varepsilon = 10^{-4}$ comes from the correction terms in the temporal evolution of the quaternion, but it can be controlled by choosing appropriate grid spacing. In our experience (including other parameter sets), the resolution should be smaller or at least of the same magnitude as ε .

5.3 Avoiding a singular Jacobian

We use Newton's method with an analytical Jacobian to solve the non-linear system arising from our discrete scheme, i.e. we solve repeatedly linear systems. The performance of those iterations is tied to the condition number of the system matrix (which is the Jacobian), up to the point that the matrix becomes numerically singular, meaning that the Newton's method will fail. A common approach to avoid this issue is the use of a preconditioner, but the Jacobi, symmetric Gauss-Seidel and ILU preconditioners did not significantly boost the performance of our solver (the computational trade-off is analyzing, adjusting and solving vs. simply solving the system). There is indeed one thing that one can easily analyze by hand, even with large system matrices: Large entries should be avoided in the system matrix. We already accounted for that in one way by avoiding that we divide by the slenderness ratio in our jet model. Furthermore, in the industrial examples to come

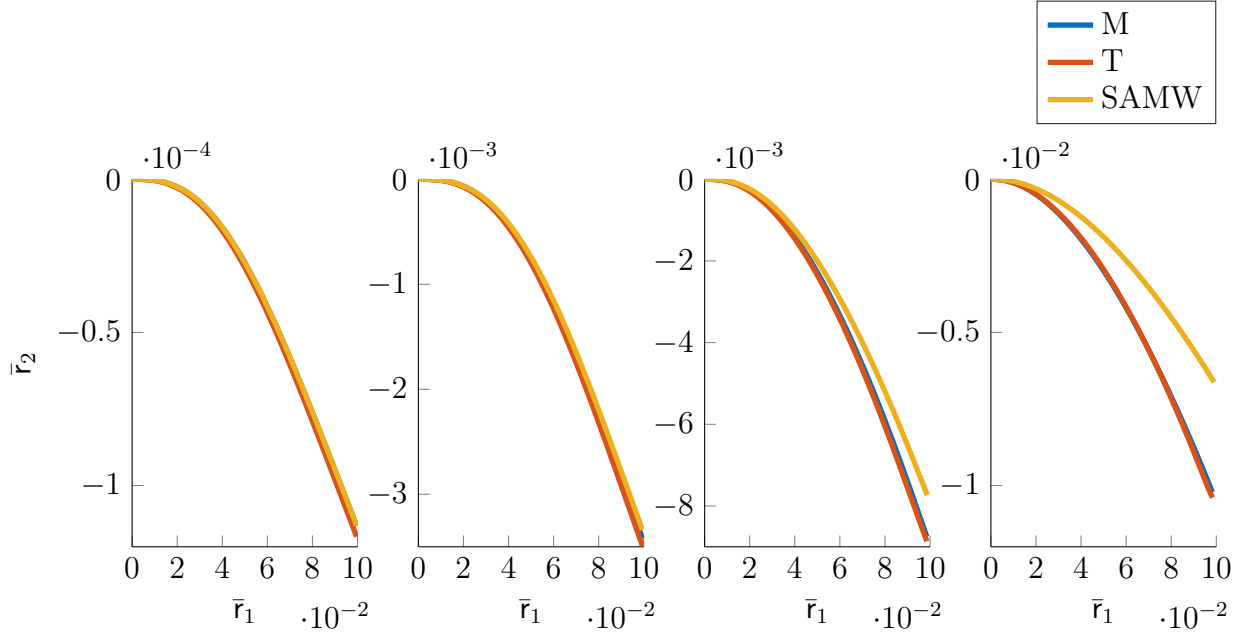


Figure 5.6: Influence of ε on the behavior of the jet model in its different formulations. The growing viscous cantilever with $(\text{Re}, \text{Fr}, T) = (0.5, 0.5, 0.1)$ and $\Delta s = \Delta t = 10^{-3}$ is used. From left to right: $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. Note that the \bar{r}_2 scale of the axes differs.

in Chapter 6 the viscous jets behave highly dynamic in reaction to the external loads causing elongations that are expected to vary several orders of magnitude within the jet. In particular, we investigate the elongation τ_3 in the material law: It appears in second and third power, meaning that we will have very large entries in the Jacobian. This could be avoided by dividing the whole equation by τ_3 , yet our assumed form of the PDAE does not have mixed terms involving the temporal derivative (e.g. the term $(\partial_t \tau_3)/\tau_3^2$ would appear in (2.37k) after dividing, which does not match (4.1)). To avoid an additional substitutional unknown, we decide to do the following: We evaluate both equations (2.37k)-(2.37l) of the material law fully implicitly and then divide them by an explicitly evaluated elongation with the respective power, more precisely we use the maximum elongation from the previous time step.

5.4 Adaptivity

We formulated the viscous jet model specifically in a general parameterization by introducing the parameter speed u and its artificial spatial fluxes into the system. Thus far we assumed that u is somehow determined. At first we analyze the impact of the additional terms and fluxes with an analytically given u . Then we investigate different approaches how to determine u , in particular with the moving mesh approaches in Chapter 3 and a variation of the Eulerian parameterization.

5.4.1 Cost of general parameterization

In the general parameterization of the jet model an additional equation for u is required. For the purpose of the following tests we prescribe $u \equiv 0$, which is added as a separate equation to the model (this gives an Lagrangian parameterization, yet all additional terms involving u are kept). We explore the increase in computation time that results simply by transforming from the Lagrangian parameterization to the general parameterization and then highlight a possible issue within the artificial fluxes.

Computation time Simply using the general parameterization enlarges the system and complicates its Jacobian (illustrated in Figure 5.7). Compared to the Lagrangian system the sparsity pattern of the general parameterization worsens: The number of entries is increased by roughly 50% (the actual size of the Jacobian is only increased by 5%) and the band width size at the boundary is increased by 20% and for the inner points by 60%. This certainly will affect the runtimes in the simulations. The calculation time is broken down into three parts:

- a) pre- and postprocessing of the time step,
- b) evaluation and construction the system matrix and its Jacobian,
- c) solving of the non-linear system.

Timing plots for the viscous cantilever are given in Figure 5.8. The critical part is the performance of the underlying band solver, the remainder in our simulation is mostly MatLab overhead. The computation time of the solver is increased by roughly 95%, the effective runtime of the entire MatLab simulation is increased by roughly 81%, which is substantial. Remember that this example uses a general parameterization with $u \equiv 0$ as separate equation. When an actual moving mesh equation is employed, the costs are even worse: The computation time of the solver is over three times longer compared to a corresponding Lagrangian simulation.

The simulations show roughly quadratically increasing computational effort in t and thus in the number of discretization points. We found that a simulation with Lagrangian parameterization can have roughly half the grid spacing Δs compared to one using the general parameterization, while still being competitive in terms of computation time.

Broad stencil issue In Section 4.2 we have shown the type and equation assignment of the unknowns of the jet model. A careful examination reveals that no ideally staggered grid for the spatial fluxes is possible, which is due to the artificial convection terms. That means that the scheme creates broad stencils that can decouple in the worst case, e.g. for the edge type-1 unknowns τ and κ in the compatibility conditions and the material law. The involved unknowns still appear in the source terms, thus we monitor at all times if actual decoupling of the solution occurs, at which point the simulation is aborted. An Eulerian parameterization can relax this issue by enforcing the arc-length constraint which simplifies the equations, but it is still present.

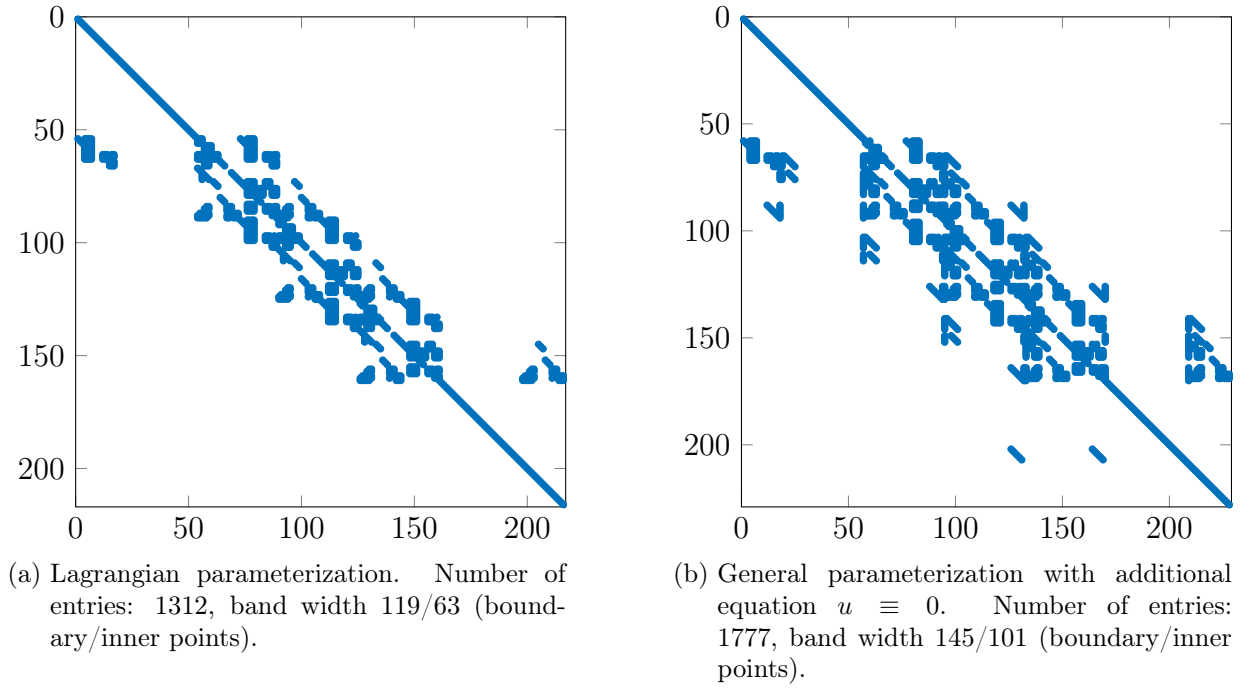


Figure 5.7: Pattern of the Jacobian of the non-linear system with an example of the viscous cantilever scenario with 3 inner points.

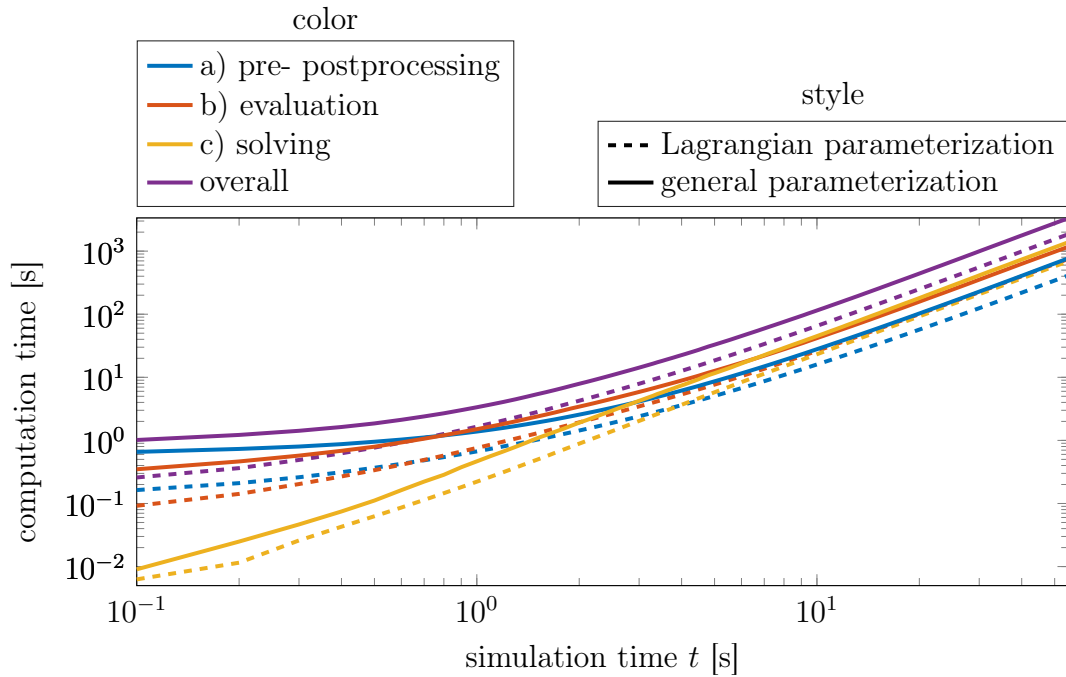


Figure 5.8: Computation time of the Lagrangian (dashed line, total computation time 31min) and general (solid line, total computation time 56min) parameterization of the growing viscous cantilever with $(\text{Re}, \text{Fr}, \varepsilon, T) = (1, 1, 8.86 \cdot 10^{-2}, \infty)$ and $\Delta s = \Delta t = 10^{-2}$.

5.4.2 Suitable re-parameterization

The determination of the parameter speed u in the viscous jet model was left open up to this point. Other than using the two most prominent examples – the Lagrangian and the Eulerian parameterization – we will consider the moving mesh approaches of Chapter 3. The performance of the MMPDE1, MMPDE2 and DELAX2 strategy is now under investigation, the DELAX1 strategy is discarded due to its direct dependence on Ψ (cf. Remark 3.28). Additionally, we think of a scaled Eulerian parameterization that is not arc-length (i.e. elongation equal to one), but that enforces a constant elongation in space. This parameterization is only added as another example of an analytically prescribed parameterization, but it is not used in the later industrial simulations. The following approaches for the determination of u are considered:

- (L) Lagrangian (material) description: $u \equiv 0$,
- (SE) Scaled Eulerian description: u becomes the multiplier of $\partial_s \tau_3 = 0$ with appropriate boundary condition,
- (D) ^{p} u is determined by DELAX2 (Strategy 3.13). The superscript $p = i$ means that the implicated parameter density is used and $p = t$ means the transferred one, the underlying monitor function is (5.1).
- (M1) u is determined by MMPDE1 (Strategy 3.9) and the monitor function given according to (5.1).
- (M2) u is determined by MMPDE2 (Strategy 3.10) and the monitor function given according to (5.1).

We aim to control rapid changes in the jet curve and its orientation (like it likely happens close to the nozzle) and thus we employ a mesh control function targeting $\mathbf{y} = (\bar{\mathbf{r}}, \mathbf{q})$ for the moving mesh strategies. We use the optimal curvature monitor function [59] and state it in the computational parameterization

$$M^{cur}(s, t) = \left(1 + 1/\alpha(t) \left\| \partial_{\sigma\sigma} \hat{\mathbf{y}}(\sigma, t) \big|_{\sigma=\Psi(s,t)} \right\|^2\right)^{1/3}, \quad (5.1a)$$

$$\alpha^{cur}(t) = \left(\frac{1}{|\Omega|} \int_{\Omega} \left\| \partial_{\sigma\sigma} \hat{\mathbf{y}}(\sigma, t) \big|_{\sigma=\Psi(s,t)} \right\|^{2/3} ds \right)^3 \quad (5.1b)$$

with

$$\partial_{\sigma\sigma} \hat{\mathbf{y}}(\sigma, t) \big|_{\sigma=\Psi(s,t)} = \frac{1}{\chi(s, t)} \partial_s \left(\frac{\partial_s \mathbf{y}(s, t)}{\chi(s, t)} \right)$$

and its implicated and transferred parameter density (cf. Strategy 3.19 and 3.20).

Remark 5.2. *The way the monitor function (5.1) is formed is not apparent, for a in-depth motivation we refer to [59]. The additive term in the monitor function ensures that it is strictly positive since derivatives could vanish in the solution. The (adaption) intensity parameter α determines the level of impact the derivatives have on the mesh distribution.*

A carefully chosen α can improve the interpolation error bounds of the adapted mesh and creates a more spatially balanced mesh since large peaks in the derivative are weakened (large peaks create locally very dense meshes). The choice of α for the monitor function (5.1) is motivated by finding an optimal error bound of an optimal equidistributing mesh.

We investigate the performance of the fixed viscous cantilever scenario, which has a time-independent domain, in combination with our stabilized kinematics. The jet is attached to a wall and we expect that large elongation will build up in proximity to the wall. In a Lagrangian parameterization all material points are constantly moving away from the wall which ultimately causes the simulation to fail or behave physically unrealistic. At first we compare the different re-parameterizations and a Lagrangian simulation with $\Delta s = 10^{-2}$ and $\Delta t = 0.1$ (plotted in Figure 5.9). The moving mesh approaches behave like expected and concentrate the discrete points close to the nozzle, cf. Figure 5.10. The $L^2(T)$ -error with respect to the Lagrangian reference solution ($\Delta s = 10^{-4}$, $\Delta t = 10^{-3}$) is calculated and printed in Table 5.3. Comparing (M2), (D)ⁱ and (D)^t to (L) the error is improved for every unknown. Notice the increase in computation time, which is the price for the adaptivity. Keeping that in mind, we refine the spatial grid spacing and redo the Lagrangian simulation in two ways: i) we seek similar computation time compared to the moving mesh solutions, and ii) we seek to obtain a similar error in (ω, κ) . For i) we find $\Delta s = 4.3 \cdot 10^{-3}$ to be suitable and for ii) $\Delta s = 2.1 \cdot 10^{-3}$. The respective Lagrangian solutions are marked with a subscript and also printed in Table 5.3. The results are in favor of the moving mesh approaches: The computation time for a similar error is roughly doubled compared to the moving mesh simulations (compare (L)_{2.1·10⁻³} to e.g. (M2)), whereas a similar computation time does not lead to comparable errors (compare (L)_{4.3·10⁻³} to e.g. (M2)).

| strategy | band solver [s] | $L^2(4)$ -error | | | |
|------------------------------------|-----------------|---|--|----------------------------|--|
| | | $(\bar{\mathbf{r}}, \mathbf{v}, \boldsymbol{\tau}, \mathbf{q})$ | $(\boldsymbol{\omega}, \boldsymbol{\kappa})$ | (\mathbf{n}, \mathbf{m}) | $(\lambda_s, \lambda_t, \bar{\lambda}_\tau, \lambda_\kappa)$ |
| (L) | 1.76 | 0.8884 | 9.3951 | 0.0261 | 0.0285 |
| (SE) | 4.37 | 0.5127 | 12.7525 | 0.0231 | 0.0391 |
| (D) ⁱ | 4.22 | 0.7721 | 2.6085 | 0.0130 | 0.0012 |
| (D) ^t | 4.27 | 0.7272 | 2.6779 | 0.0172 | 0.0015 |
| (M1) | 4.94 | 0.7841 | 2.4549 | 0.0140 | 0.0014 |
| (M2) | 4.42 | 0.6741 | 2.2549 | 0.0099 | 0.0014 |
| (L) _{4.3·10⁻³} | 4.32 | 0.8164 | 3.9755 | 0.0238 | 0.0076 |
| (L) _{2.1·10⁻³} | 8.92 | 0.8073 | 2.3608 | 0.0232 | 0.0071 |

Table 5.3: Comparison of the band solver's computation time and resulting error for the fixed viscous cantilever of Figure 5.2.

Another advantage of the moving mesh approaches can be seen in the simulation runtime of the example. The Lagrangian simulation quickly fails after the time $t = 4$ (over- and undershooting is already visibly in κ_2 , Figure 5.9) whereas the moving mesh approaches allow a continued simulation. The scaled Eulerian case also fails when the overall elongation becomes too large causing insufficient resolution close to the wall.

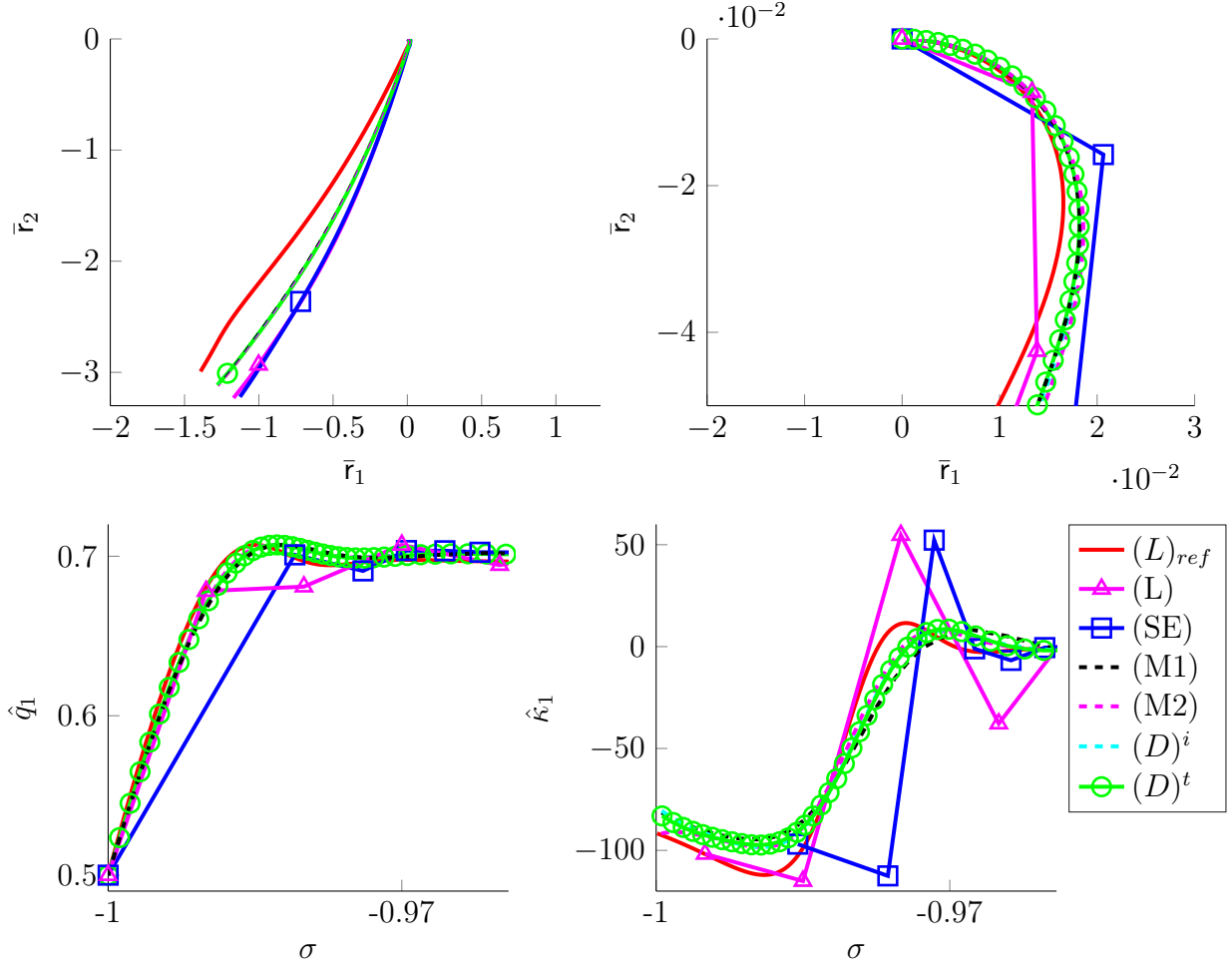


Figure 5.9: Cantilever scenario: Comparison of solutions without and with a moving mesh with $(Re, Fr, \varepsilon, T) = (1, 1, 8.86 \cdot 10^{-2}, 4)$ and $\Delta s = 10^{-2}$, $\Delta t = 10^{-1}$ against the reference from Figure 5.2. Top: Side view on the fiber and zoom-in close to the nozzle. Bottom: Quaternion q_1 . In the top right and bottom plot the markers represent the actual discrete grid for the (L) , (SE) and $(D)^t$ solution. The solution of $(M1)$, $(M2)$ and $(D)^i$ are indicated with a dashed line since they are only marginally different from the $(D)^t$ solution.

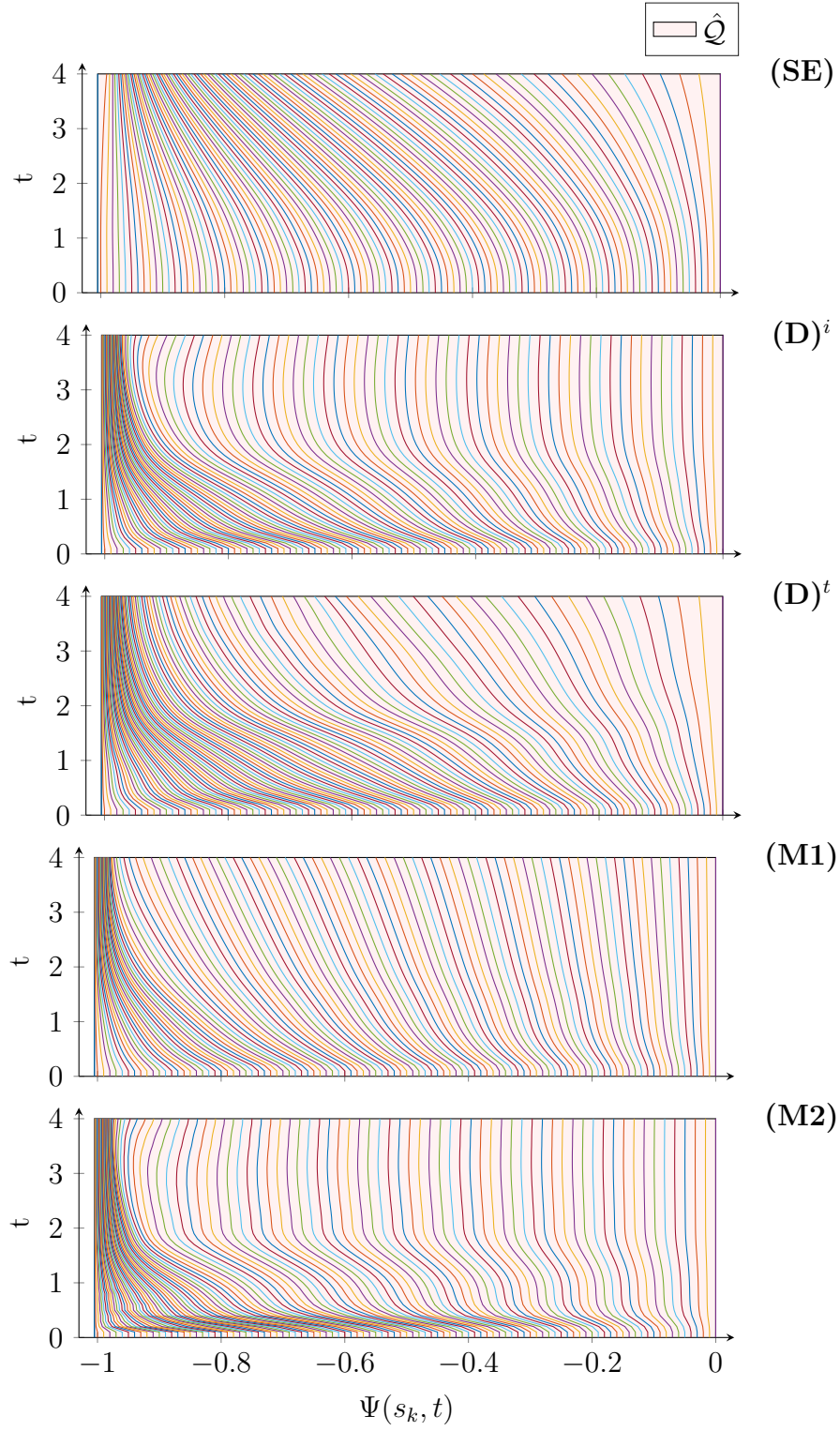


Figure 5.10: Cantilever scenario with parameters from Figure 5.9: Comparison of the mesh movement for (M1), (M2) with monitor function (5.1), and (D) with implicated and transferred parameter density and $\tau = 5\Delta t$.

Large external loads The former cantilever scenario with a moderate Froude number does not include locally large forces like we are expecting to face in the industrial spinning scenarios. In order to test our discrete scheme we decrease the Froude number to $Fr = 0.1$ and redo the simulations. We find that the moving mesh solutions start to oscillate and ultimately break down quicker (in simulation runtime) than the respective Lagrangian simulation, likely due to the broad stencil of the general parameterization (described in the end of Section 5.4.1). The oscillations are illustrated for the rotational spinning process in Appendix C.4. The use of a higher order interpolation (Remark 5.3) or the usage of $\boldsymbol{\tau}$ and $\boldsymbol{\kappa}$ as type-0 fields in the material law and compatibility conditions (cf. Remark 5.4) could not overcome this issue.

Remark 5.3 (Higher order interpolation). *The oscillations in the moving mesh solution are possibly caused by decoupled, broad stencils that can be addressed by a higher order node to edge interpolation (and vice-versa). In particular, we employ cubic interpolation in the inside and quadratic interpolation at the boundaries for (4.4) and (4.7). This can prevent a sudden blow up of the oscillations, but they are not completely avoided.*

Remark 5.4 (Using the distortion measures as unknowns). *All physical meaningful conservation quantities are node unknowns which led us to the decision to put the artificial convection speed u on edges in order to maintain perfect conservation for them. Nevertheless we also have edge unknowns that are type-1 and thus obtain a conservation form in the general parameterization, namely $\boldsymbol{\tau}$ and $\boldsymbol{\kappa}$. Conservation is not guaranteed due to the interpolation that is required in their fluxes, which leads to decoupled, broad stencils for the flux approximation. Since conservation is not guaranteed anyways in this case, it is not too far-fetched to give up the conservative form, i.e. treat $\boldsymbol{\tau}$ and $\boldsymbol{\kappa}$ as type-0 fields. Then the transformation is*

$$\begin{aligned}\hat{\boldsymbol{\tau}}|_{\sigma=\Psi(s,t)} &= \boldsymbol{\tau}, & \hat{\boldsymbol{\kappa}}|_{\sigma=\Psi(s,t)} &= \boldsymbol{\kappa}, \\ \partial_t \hat{\boldsymbol{\tau}}|_{\sigma=\Psi(s,t)} &= \partial_t \boldsymbol{\tau} + u \partial_s \boldsymbol{\tau}, & \partial_t \hat{\boldsymbol{\kappa}}|_{\sigma=\Psi(s,t)} &= \partial_t \boldsymbol{\kappa} + u \partial_s \boldsymbol{\kappa}.\end{aligned}$$

We do not restate the whole jet model at this point. As an example, the material law (2.37k) becomes

$$\partial_t \tau_3 + u \partial_s \tau_3 = \frac{\text{Re}}{3\mu} \frac{\chi \tau_3^2}{\sigma_V} \mathbf{n}_3.$$

This avoids an entirely decoupled broad stencil because u is now outside of the flux. The price is the transformation invariance of the jet model and some additional interpolation when χ is required on the edge grid. This adjustment does not improve the performance of the moving mesh strategies, it even worsens it in general (to be more precise, numerical oscillations occur earlier). We conclude that the non-conservative form introduces more aggravating errors than the broad stencil.

The example of this section is just an illustration, the issue of an oscillating solution transfers to all cases relevant in this work. The mesh control function (5.1) is solely an example as well, even though it is working best for the jet model in our experience. We do not claim that this is in fact the best one, simply because there is such a vast

variety of possibilities. In particular, the use of the arc-length and optimal slope monitor function [59] (that use the first derivative of the solution components), as well as the component-averaged monitor function as used in [96, 102] (see also Appendix C.2) does not overcome the issue of an oscillating solution. Furthermore, the mesh control function is not restricted to which components of the solution are controlled, e.g. the usage of the distortion measures or contact force and couple with one of the above monitor functions is feasible as well, but does not improve the results. Our experience leads us to the assumption that more sophisticated spatial flux approximations are required for the artificial fluxes (see Remark 5.5). On the other hand, benchmarks of our discrete scheme with simple, central approximation and the moving mesh strategies in combination with the viscous Burger's equation (cf. Appendix C.2) perform comparably to results found in literature [59]. Further investigation is necessary here to find what is actually causing the oscillations. An idea to adjust the temporal relaxation parameter τ is stated in Remark 5.6.

Remark 5.5 (High resolution schemes). *There are many ideas in literature for flux splitting strategies and high resolution schemes with special consideration of adaptive mesh refinement (AMR) and conservation laws – e.g. in [96, 40, 99, 102, 83]. In particular, the third- and fifth-order weighted essentially nonoscillatory (WENO) method [92] used in [110] seems promising. WENO uses the idea of adaptive stencils in the reconstruction procedure of the derivatives. A convex combination of candidate stencils is employed, each being assigned a nonlinear weight depending on the local smoothness of the numerical solution based on that stencil. Such a WENO reconstruction has a high computational effort and would be, in a first attempt, only be applied to the artificial fluxes of the jet model. The use of a monotone upstream-centered scheme for conservation laws (MUSCL) method [103], which is more straight-forward to implement and computationally efficient than WENO is certainly feasible to be explored as well.*

Remark 5.6 (Time step control). *The time step is taken to be constant in the simulation. We specifically do not want to employ a time step control since that might drastically increase the computation time for longtime simulations. If the mesh movement cannot be properly resolved by the chosen time step, mesh racing effects can occur [96, 26]. This might be overcome by using a time-dependent temporal relaxation parameter τ that is chosen in accordance with the current parameter speed. The basic idea is that the mesh movement is slowed down to avoid mesh racing effects without the reduction of the time step [94]. Even so, the oscillations found in the example above with the jet model are likely not due to mesh racing because the mesh movement is properly resolved by the chosen time step when the oscillations occur.*

Nevertheless, our central flux approximation works well for the jet model in Lagrangian parameterization due to the ideally staggered fluxes, cf. Remark 4.5. We will examine its capabilities in the following chapter with the industrial applications at hand.

6 Applications

The industrial applications that motivated this work are highly dynamic jet spinning processes with an out-stream element. In such a process a jet consisting of a liquid polymer is extruded through a nozzle and then moves due to viscous friction, surface tension and gravity while it is cooling down. Aerodynamic forces and other external loads may also influence the jet [74, 6, 8]. In the last sections we introduced the viscous jet model – neglecting surface tension – with suitable frames for initial and boundary conditions, as well as a discrete scheme that will now be used. We discuss capabilities of said model and scheme with two industrial examples at hand, namely the rotational spinning and the melt-blowing process. The selection of those two production processes is motivated through the research cooperation with the Fraunhofer ITWM and their ties to industrial partners (in particular the physical parameters, and for the melt-blowing process the airflow data, is provided by the Fraunhofer ITWM, as well as key routines for the aerodynamic force and turbulent reconstruction, Remark 6.5). In those highly-viscous processes the transient behavior of the jet is an essential feature, which hinders any meaningful reduction of the model and underlines the significance of transient simulations. We aim to provide a framework for both industrial applications.

From our point of view, special interest is on robust long-time simulations and the applicability of industrially relevant parameter ranges, as well as predictions of the jet behavior close to the nozzle and of the jet's thickness. We discuss the key aspects that have to be considered for a successful simulation and show exemplary simulations for both processes, yet neglecting temperature effects. In particular we introduce the rotational spinning process in Section 6.1 and describe the required model alteration as well as the external loads in Section 6.1.1. In Section 6.1.2 we present details of the numerical method, explore alterations of the boundary conditions of the tension-free end and estimate resolution requirements of the process. In Section 6.1.3 the impact of the characteristic numbers (Reynolds and Rossby number) is studied. At last, simulations for industrially relevant parameter ranges are shown in Section 6.1.4. Our course of action is similar for the melt-blowing process in Section 6.2. We introduce the necessary models for the turbulent aerodynamic force in Section 6.2.1 and Section 6.2.2. Following that, we present details of the numerical method in Section 6.2.3 and analyze resolution requirements imposed by the turbulent scales of the used airflow. Approaches for the industrial example are shown in Section 6.2.4.

6.1 Rotational spinning process

In the rotational spinning process [72] many small nozzles are located on the curved face of a circular cylindrical drum that rotates around its symmetry axis with high frequency. Like

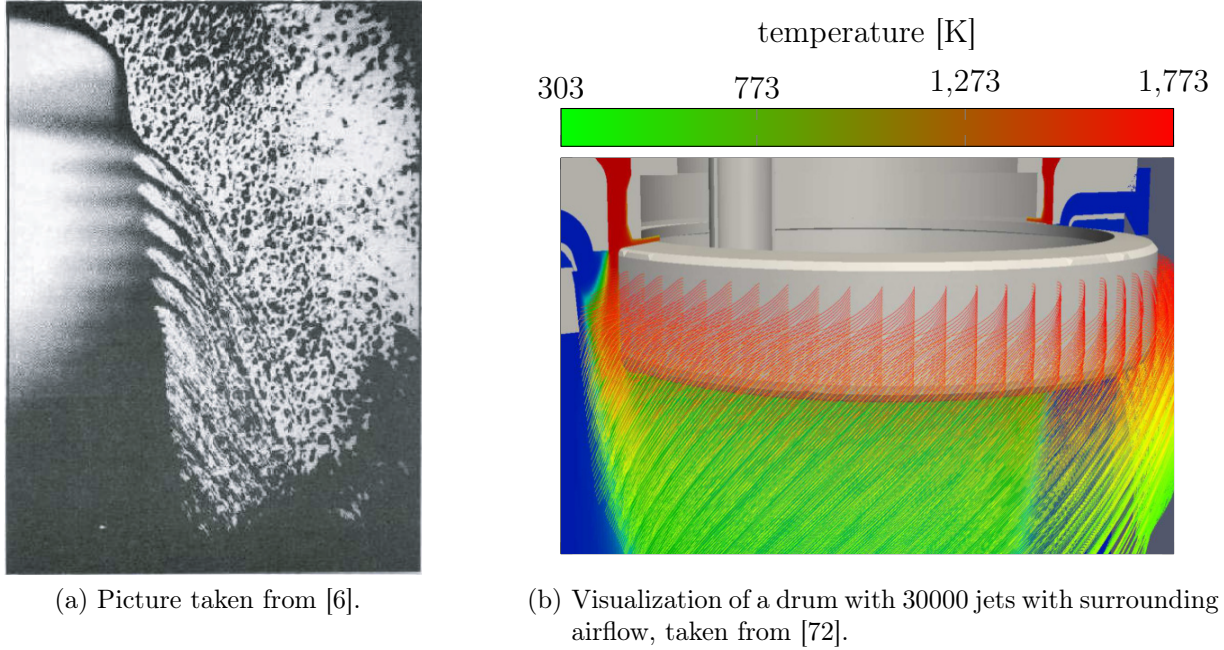


Figure 6.1: Illustration of the rotational spinning process.

we mentioned in the beginning of Section 6.1, the jets deformation is driven by external loads, in this process additionally of centrifugal and Coriolis forces. The temperature of the polymer at the nozzle is $1303K$, the surrounding airflow has a temperature range from $1773K$ to $303K$, thus the deformation of the jet is highly dependent on the temperature (see Figure 6.1). The jets are being swirled by the aerodynamic forces while solidifying and are normally deposited onto a conveyor belt. In this work we are focusing on the extrusion process and the deformation of the jet caused by gravitational and rotational forces, aerodynamic forces are neglected. During the cooling process the jet behaves more and more elastic, which we expect has a beneficial effect on the simulation since the deformations become less drastic. Nevertheless, we simulate this process with the viscous jet model and a constant viscosity instead of using a viscoelastic material law and a temperature-dependent model for the viscosity (cf. Section 2.1.3 and Section 2.1.4), as required for a realistic behavior. Our purely viscous approach can be seen as a worst-case scenario, which justifies its usage.

In [6, 9, 8, 5] the use of the viscous jet model in the rotational spinning process is analyzed in a stationary case – both theoretically and numerically – in great detail. The numerical investigation was done by reducing the process to the extrusion process of one jet and thus neglecting jet interaction, which we will do as well. A sketch of the drum with one jet is shown in Figure 6.2. The numerical approach for the instationary case in [5] has a worrisome convergence order reduction effect in the temporal convergence, which is due to their zig-zag approximation of the space-time domain, cf. Remark 4.7. Furthermore, in [5] we used a formulation of the jet kinematics that introduces drift-off effects that might alter solutions in long-term studies. We avoid them here by using our viscous jet model with the stabilized kinematics (cf. Section 2.2.5), that, together with our discrete scheme,

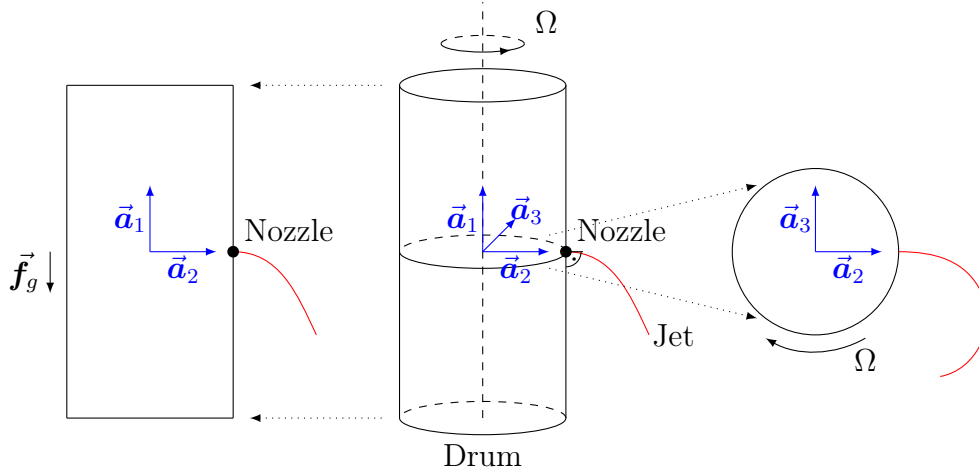


Figure 6.2: Sketch of the drum with one nozzle on the side. The jet is indicated with a red line. On the left a side view of the drum, on the right a top view. Inspired by [6].

can overcome the order reduction effect, cf. Section 5.2, Figure 5.4.

6.1.1 Model alteration, external loads and parameters

The symmetry axis of the rotating drum is $\vec{\Omega} = \Omega \vec{e}_\Omega$ with rotational frequency Ω . To fit the rotational spinning process into our definition of the jet extrusion set-up in Section 2.3.1 we follow [6, 5] and use a reference frame that rotates with the drum. Let $\vec{\Omega} = \Omega \vec{e}_\Omega$ be the angular frequency of the rotating device with the rotational axis \vec{e}_Ω . We set the outer basis $\{\vec{a}_1(t), \vec{a}_2(t), \vec{a}_3(t)\}$ such that $\vec{a}_1 \equiv \vec{e}_\Omega$ and $\partial_t \vec{a}_i(t) = \vec{\Omega} \times \vec{a}_i(t)$ holds for $i = 2, 3$. This makes the position of the nozzle and the direction of the inflow time-independent, but introduces rotational body forces and couples in the dynamic equations due to inertia (cf. (6.1)). We incorporate them into the $\vec{\Omega}$ -adapted linear and angular velocities, i.e. $\vec{v}_\Omega = \vec{v} - (\vec{\Omega} \times \vec{r})$ and $\vec{\omega}_\Omega = \vec{\omega} - \vec{\Omega}$. Those adapted fields are now substituted into the jet model (2.37), which means that the kinematics keep the form stated in (2.37). The substitution adds fictitious external loads for Coriolis as well as centrifugal effects in the balance laws for linear and angular momentum. Overall, the external loads cover the mentioned fictitious loads and gravity and are characterized by the dimensionless Rossby number Rb and Froude number Fr. The gravitational force is acting in direction of $-\vec{a}_1$ and the gravitational constant $g = 9.81 \text{ [m/s}^2\text{]}$ induces the Froude number (ratio of flow inertia to gravity)

$$\text{Fr}^2 = \frac{v_\star^2}{gr_\star}.$$

The symmetry axis of the drum is \vec{a}_1 and the rotational effects use the typical angular frequency $\Omega_0 = \Omega_\star [1/s]$. To relate both time scales of our model we introduce the Rossby number

$$\text{Rb} = \frac{v_\star}{\Omega_\star r_\star},$$

which expresses the ratio of inertial to centrifugal forces. The underlying choice of the typical values is the following: $r_\star = R$ is the radius of the rotating drum and $\Omega_\star = \Omega$ the rotational frequency. Furthermore $v_\star = U$ is the extrusion speed, $\sigma_{M\star} = \rho(\pi/4)D^2$ the mass line density and $\sigma_{V\star} = (\pi/4)D^2$ the cross-section area of the nozzle. The slenderness ratio ε is now a relation between the cross-section diameter and the radius of the drum. Furthermore we set $\mu_\star = \mu$ as the dynamic viscosity (μ is the viscosity of a typical polymer at the nozzle with temperature $1323K$). The appearing physical parameters are listed in Table 6.1. Summing up, the external loads are given by $\mathbf{f} = \mathbf{f}_g + \mathbf{f}_\Omega$ and $\mathbf{l} = \mathbf{l}_\Omega$ with

$$\begin{aligned}\mathbf{f}_g &= -\frac{1}{\text{Fr}^2}\sigma_M \mathbf{R} \cdot \mathbf{e}_1 \\ \mathbf{f}_\Omega &= -\frac{2}{\text{Rb}}\sigma_M(\mathbf{R} \cdot \Omega \mathbf{e}_1) \times \mathbf{v} - \frac{1}{\text{Rb}^2}\sigma_M \mathbf{R} \cdot (\Omega \mathbf{e}_1 \times (\Omega \mathbf{e}_1 \times \bar{\mathbf{r}})), \\ \mathbf{l}_\Omega &= \frac{1}{\text{Rb}} \left[(\mathbf{J}_M \cdot (\mathbf{R} \cdot \Omega \mathbf{e}_1)) \times \left(\boldsymbol{\omega} + \frac{1}{\text{Rb}}(\mathbf{R} \cdot \Omega \mathbf{e}_1) \right) + (\mathbf{J}_M \cdot \boldsymbol{\omega}) \times (\mathbf{R} \cdot \Omega \mathbf{e}_1) \right. \\ &\quad \left. + \mathbf{J}_M \cdot (\boldsymbol{\omega} \times (\mathbf{R} \cdot \Omega \mathbf{e}_1)) - \partial_t \mathbf{J}_M \cdot (\mathbf{R} \cdot \Omega \mathbf{e}_1) \right]\end{aligned}\tag{6.1}$$

with rotation $\mathbf{R} = \mathbf{R}(\mathbf{q})$. Note that the usage of the incompressible geometry model gives the simplification $\partial_t \mathbf{J}_M = -(\partial_t \tau_3 / \tau_3) \mathbf{J}_M$. The jet model with the $\vec{\Omega}$ -adapted velocities is used in this section without restating the complete model equations, also no special marking is introduced for the rotating outer basis since the context allows to distinguish it. Only keep in mind that we deal with adapted quantities and that the outer basis is changed, i.e., the director triad represented by the quaternions is also with respect to the rotating outer basis.

Remark 6.1 (External loads compared to Arne et al. [5]). *Note that we summarize the external loads of the rotational spinning process in a different way compared to [5]. In particular, consider the angular momentum balance law (2.37). In [5] the whole term $\mathbf{h} \times \boldsymbol{\omega}$ is moved into the torque \mathbf{l} . One of our goals is to keep the form of the model equations, therefore we split it up. Furthermore, Arne et al. use a two dimensional simplification of the process for their convergence studies by neglecting the gravitational force ($\text{Fr} \rightarrow \infty$). For completeness we present it in our formulation in Appendix B.2.*

6.1.2 Numerical method and investigation

We use the framework for the space-time domain and initial and boundary conditions provided by the jet extrusion set-up in Section 2.3.1. In particular we use the setting prescribed by Example 2.38, with its two cases: with and without mass outflow at the right side. Starting with $l(0) = r(0) = 0$ we set

$$\bar{\mathbf{r}}^L = \mathbf{e}_2, \quad \mathbf{q}^L = \frac{1}{\sqrt{2}}(1, 1, 0, 0)$$

and assume $\mathcal{L}(t) = l(t) = t$, $\mathcal{R}(t) = r(t)$ if not otherwise mentioned. This yields $v^L = 1$ for the mass inflow, yet the mass outflow v^r varies in the simulations and is stated with

| Parameters | | | |
|---------------------------------|-------------------------------------|----------------------|-------------------|
| Description | Symbol | Value | Unit |
| Fiber density | ρ | 2500 | kg/m ³ |
| Fiber dynamic viscosity | μ | 162.3577 | kg/m/s |
| Nozzle diameter | D | $7.4 \cdot 10^{-4}$ | m |
| Extrusion speed | U | $6.7 \cdot 10^{-3}$ | m/s |
| Drum rotational frequency | Ω | $2.3 \cdot 10^2$ | 1/s |
| Drum radius | R | $2 \cdot 10^{-1}$ | m |
| Dimensionless quantities | | | |
| Description | Formula | Value | |
| Reynolds | $Re = \rho U R / \mu$ | $2.06 \cdot 10^{-2}$ | |
| Rossby | $Rb = U / (\Omega R)$ | $1.46 \cdot 10^{-4}$ | |
| Froude | $Fr = U / \sqrt{g R}$ | $4.78 \cdot 10^{-3}$ | |
| Slenderness | $\varepsilon = (\sqrt{\pi}/2)(D/R)$ | $3.28 \cdot 10^{-3}$ | |

Table 6.1: Overview of the physical parameters of the industrial example of the rotational spinning process illustrated in Figure 6.1, taken from [8]. The typical values are plugged in and the dimensionless numbers computed.

them. We now analyze the impact of different approaches for the tension-free boundary and estimate resolution requirements depending on the Rossby number.

Outflow at the tension-free boundary

Industrial partners are mostly interested in the behavior close to the nozzle and the achievable thickness of the jet. Thus, simulations of the rotational spinning process leave some freedom in choosing the boundary conditions for the tension-free end. We explore two cases, that we already published in [89]: a) without mass outflow, i.e. $r(t) = 0$, and b) with mass outflow, i.e. $r(t) > 0$, more precisely we utilize $r(t) = 0.3t$ here.

- a) A natural boundary layer at the tension-free end arises because the distortion measures keep their initial values due to the viscous material law. This creates physically unrealistic behavior and ultimately causes the simulation to fail – in our explanatory simulation in Figure 6.3 at time $t = 1.52$.
- b) The jet behaves comparable to a) over its length (cf. τ_3 in Figure 6.4), but the imminent boundary layer is suppressed due to mass outflow. Different attempts to find a suitable outflow function have shown that the concentration of mass at the tension-free end is crucial for realistic deformation of the jet. Too much mass outflow creates quasi-stationary state in the jet long before the expected deformation is reached, too little outflow creates physically unrealistic behavior due to the boundary layer which manifests most obviously in the fact that the bulk mass at the jet end starts to move against gravity, at which point the simulation quickly fails. Our choice of mass outflow is illustrative to show the possibilities for a prolonged simulation. While the boundary layer can be avoided, at some point however the normal velocities start to

oscillate due to errors in the director triad (the jet appears to be twisting). When the jet becomes thinner and the normal velocities increase drastically they cannot be properly resolved with a static grid – in our explanatory simulation in Figure 6.5 at time $t = 5.02$. The use of the moving mesh strategies of Chapter 3 with mesh control functions (like shown in Section 5.4.2) targeting the jet curve, orientation, contact force and couple, as well as linear and angular velocity (or any combination of those) can not overcome this issue, the respective simulations show an oscillating solution similar to Section 5.4.2.

The results show that case b) is preferable. We expect simulations of even longer run time (more importantly producing an even longer jet) when using a more suitable outflow function.

Artificial boundary conditions at the tension-free boundary

In general, when an outflow is present, the viscous jet model requires artificial boundary conditions for its distortion measures. We use an extrapolation boundary as described in Section 4.2. The use of alternate boundary conditions that are possible through the tension-free boundary do not give improved results, cf. Remark 6.2.

Remark 6.2 (Artificial condition for τ_3 and κ). *In Section 4.2 we discuss the necessary artificial boundary conditions for a tension-free boundary. In particular we use an extrapolation boundary for τ and κ . Alternatively, we use the inherent equations (2.41c) and (2.41d) along the boundary, more precisely we use one-sided, first-order finite difference approximation of them. We do this by replacing all equations j in \mathbf{R} (the function that provides the boundary conditions, cf. Section 4.1.2), belonging to τ and κ by*

$$\begin{aligned} (Y_{R,j}^{n+1} - Y_{R,j}^n) - \frac{\mathcal{R}(t_{n+1}) - \mathcal{R}(t_n)}{\mathcal{R}(t_{n+1}) + s_{k_r-1/2}} (Y_{R,j}^{n+1} - Y_{k_r-1/2,j}^{n+1}) \\ - \frac{\Delta t}{\mathcal{R}(t_{n+1}) + s_{k_r-1/2}} (Y_{R,u}^{n+1} Y_{R,j}^{n+1} - Y_{k_r-1/2,u}^{n+1} Y_{k_r-1/2,j}^{n+1}) = 0 \end{aligned}$$

with $Y_{R,u}^{n+1}$ as the component representing the parameter speed u . In the special case that $u \equiv 0$ and $\mathcal{R}(t) = 0$ this can be simplified even further and the boundary unknowns can be set according to $\tau_3(\mathcal{R}(t), t) = \mathbf{e}_3$ and $\kappa(\mathcal{R}(t), t) = \mathbf{0}_3$. We repeat the longtime studies shown in Figures 6.3 and 6.5 with the replaced boundary conditions. The difference at time $t = 5.01$ for mass outflow case is less than 1.5% in the unknowns of the last active cell and this quickly vanishes further into the domain. It also has no effect on the runtime, the simulation breaks down at the same time as the former one with extrapolation boundaries. Overall, the influence on the discrete solution of these replaced boundary conditions is marginal, since we only consider mass outflow on the right side. We always use the extrapolation boundary.

Spatial and temporal resolution

The applied spatial and temporal grid spacing critically depends on the considered problem parameters. Smaller parameter values of Rb , Fr and ε , or bigger ones for Re imply faster,

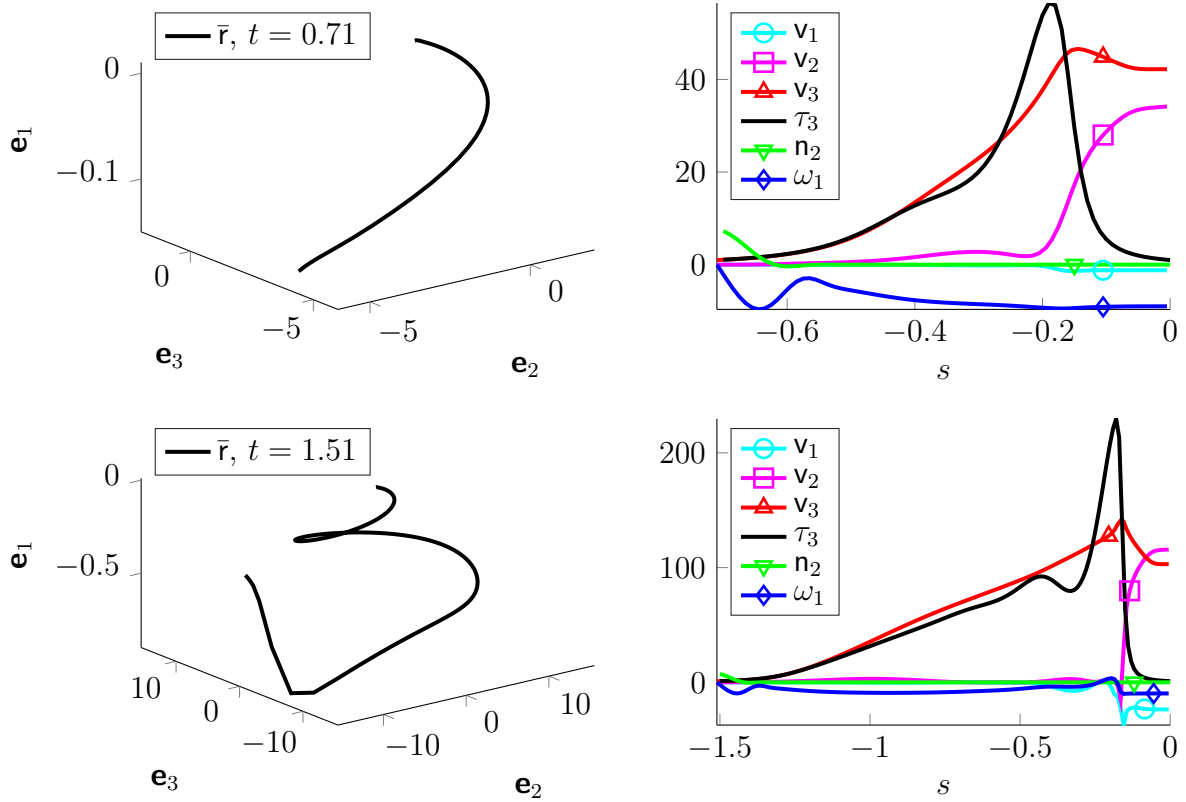


Figure 6.3: Rotational spinning scenario with $(\mathcal{L}(t), \mathcal{R}(t)) = (t, 0)$, $(\text{Re}, \text{Rb}, \text{Fr}, \epsilon) = (1, 0.1, 1, 8.86 \cdot 10^{-2})$ and $\Delta t = \Delta s = 10^{-2}$. Notice the inaccuracies due to overshooting in the velocities of the jet (e.g. v_1) which are caused by the boundary layer. It creates unrealistic behavior and ultimately causes the simulation to fail (cf. the tension-free end is moving against the gravitational force).

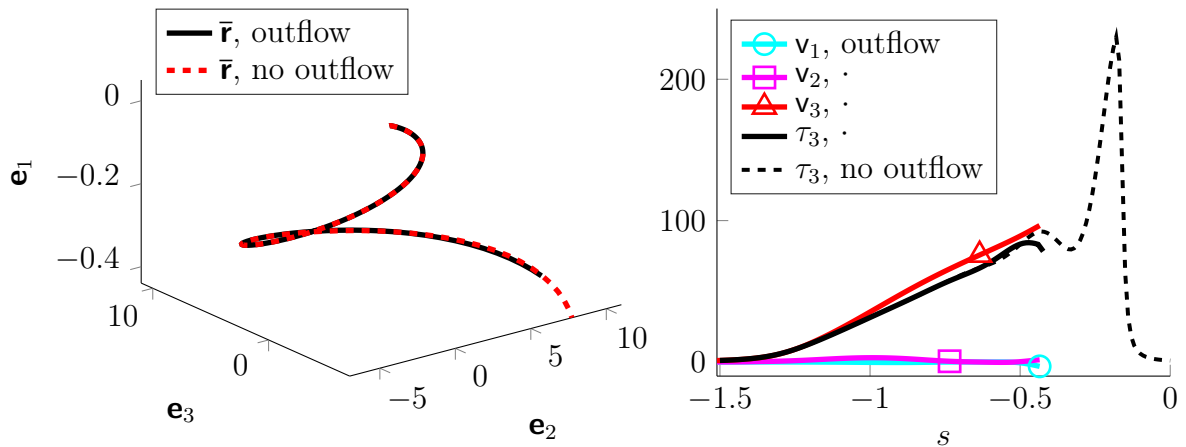


Figure 6.4: Comparison of the effects of tension-free boundary with and without mass outflow at $t = 1.51$, remaining unknowns are comparable in the similarity. [89]

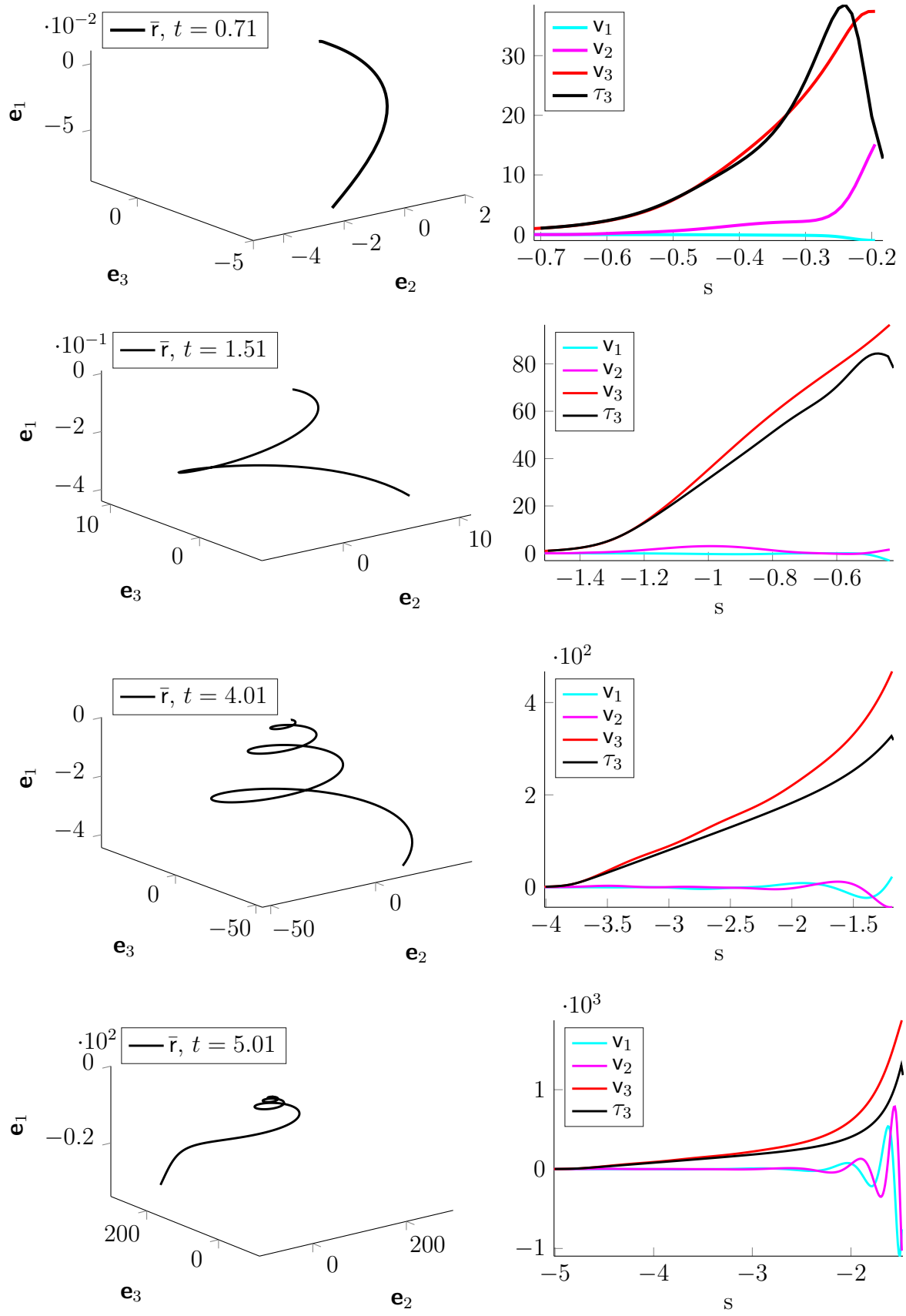


Figure 6.5: Longtime simulation of the rotational spinning process with $(\text{Re}, \text{Rb}, \text{Fr}, \epsilon) = (1, 0.1, 1, 8.86 \cdot 10^{-2})$ and $\Delta t = \Delta s = 10^{-2}$, given mass outflow $v^r = 0.3$.

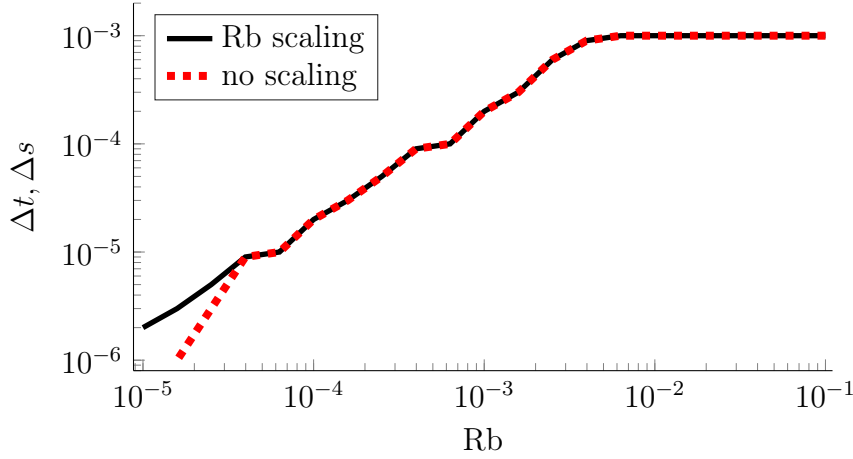


Figure 6.6: Estimation of the safety factor C_{Rb} for the rotational spinning process. For every Rb the graph indicates the minimal grid spacing ($\Delta s = \Delta t$) that is necessary for a successful simulation.

larger changes in the dynamics and higher elongation, which requires a finer resolution. The impact of Rb is now explored, since it is the driving parameter for the rotational spinning process. The Rossby number is linked to the dimensionless time that the drum needs for one rotation, namely $2\pi Rb$, and is very small in industrially relevant parameter ranges, cf. Table 6.1. This revolution time has to be properly resolved, thus Δt has to be equal or less than $C_{Rb} Rb$, where C_{Rb} is a positive constant and $2\pi/C_{Rb}$ estimates how many time steps within one revolution are necessary for proper resolution. The jet leaves the nozzle unstretched and unbent and then undergoes external loads in normal direction that aggravate with $Rb \rightarrow 0$. This manifests in a boundary layer close to the nozzle that develops quickly in the normal inner forces, curvature and angular velocity but do not worsen over time (cf. n_2, ω_1 in Figure 6.3). To find C_{Rb} for the example in Table 6.1 we set $(Re, Fr, \epsilon) = (2.06 \cdot 10^{-3}, 4.78 \cdot 10^{-3}, 3.28 \cdot 10^{-3})$ and $\Delta t = \Delta s = 10^{-2}$. We start a simulation for every $Rb \in \{10^{-5}, 2 \cdot 10^{-5}, \dots, 10^{-1}\}$ and keep track of the boundary layer that builds up towards the nozzle. If it behaves unrealistic after 100 time steps or the simulation breaks down, we equally refine the grid spacing until success. The estimation of the required resolution for every Rb is presented in Figure 6.6. For $Rb > 5 \cdot 10^{-3}$ we require at least $\Delta t = \Delta s = 10^{-3}$, due to the stiffness of the system – observe that ϵ is of similar magnitude. For small Rb numbers we estimate $C_{Rb} \approx 2 \cdot 10^{-1}$ ($2\pi/C_{Rb} \approx 30$). The longtime study in Figure 6.5 has moderate physical parameters, especially in the Rossby number. Due to the fact that we are dealing with small Rossby numbers and Rb appears quadratic in the external loads (6.1) of the conservation laws, we divide both equations with Rb^2 to avoid small number division – we call this Rb -scaling. As expected the Rb -scaling has a big impact when Rb^2 gets close to the accuracy of the Newton's method: Without scaling the simulation fails for $Rb \leq 2 \cdot 10^{-5}$, cf. Figure 6.6, dashed red line. In general the condition of the Jacobian is improved, thus we always use the Rb -scaling.

6.1.3 Parameter study

We continue the investigations in the spirit of [5] regarding the robustness and applicability in terms of parameter ranges. We mentioned in the introduction (Chapter 1) that former attempts for the instationary simulations of rotational spinning processes use a string model, which has a restricted applicable parameter range in terms of Re and Rb [80, 6]. The more sophisticated rod model opens the full parameter range to simulations [5], which is confirmed here by simulations with our three-dimensional viscous model and the stabilized kinematics (deviations compared to the original kinematics are less than 0.8%).

For the no outflow case (cf. Section 6.1.2) we exemplarily visualize the elongation in dependence on Re and Rb for $T = 1$ in Figure 6.7, red line, with a fixed grid spacing for all parameter sets. We start with a Rossby number $Rb = 1$ and compute the solutions for the different Reynolds numbers. Then we halven the Rossby number and repeat the computations until the simulation breaks down, which is the case for $Re = 10^{0.2}$, $Rb = 3.125 \cdot 10^{-2}$. In theory all parameter settings can be computed with the help of our discrete scheme, but the simulations with no outflow are practically restricted to moderate parameter ranges by the appearance and therein lack of resolution at boundary layers at the free jet end. Certainly, those layers can be resolved with a finer static grid, but this study employs a fixed grid spacing. Attempts with the moving mesh approaches (Chapter 3) help with the proper resolution of the layers, but they can not achieve the required runtime due to appearing oscillations similar to Section 5.4.2. We adjust the tension-free boundary conditions by introducing mass outflow, which avoids an aggravating boundary layer and broadens the applicable parameters range. We repeat the study of Figure 6.7 for an arbitrarily chosen outflow of 30% (cf. Figure 6.7, black line) and an adjusted end time $T = 1.3$ (cf. Figure 6.7, dashed black line), for which the extruded length is the same as in the no outflow case. The elongation behavior is similar in all three cases and simulations up to $Re = 10^{0.5}$, $Rb = 1.5625 \cdot 10^{-2}$ are now possible. The simulation for bigger Reynolds or smaller Rossby numbers still breaks down due to the fact that the outflow magnitude is not fitted to the problem, it still allows some mass lumping at the free jet end. This lumping also creates the difference of the curvature behavior for small Reynolds numbers. For bigger Reynolds numbers, and generally for small Rossby numbers the impact of the mass lumping at the tension-free end on the elongation behavior is decreasing. For example, consider the outflow case with $T = 1.3$ and the no outflow case (those two cases have the same extruded length) with $Rb = 3.125 \cdot 10^{-2}$ in Figure 6.7, that have very similar maximum elongation.

Our results for the no outflow case are comparable to [5], but the resolution requirements are greatly relaxed. The discrete approach in [5] was quite restrictive and required a resolution of magnitude $\mathcal{O}(10^{-4})$ for $Rb = 1.25 \cdot 10^{-1}$, whereas our new approach reaches $Rb = 3.125 \cdot 10^{-2}$ with $\Delta t = \Delta s = 10^{-3}$, and an even broader parameter range when an outflow is used. We expect that the Rossby numbers can be decreased further if the outflow function is chosen more carefully and fitted to the problem. We will do this for the industrial application in the following Section 6.1.4.

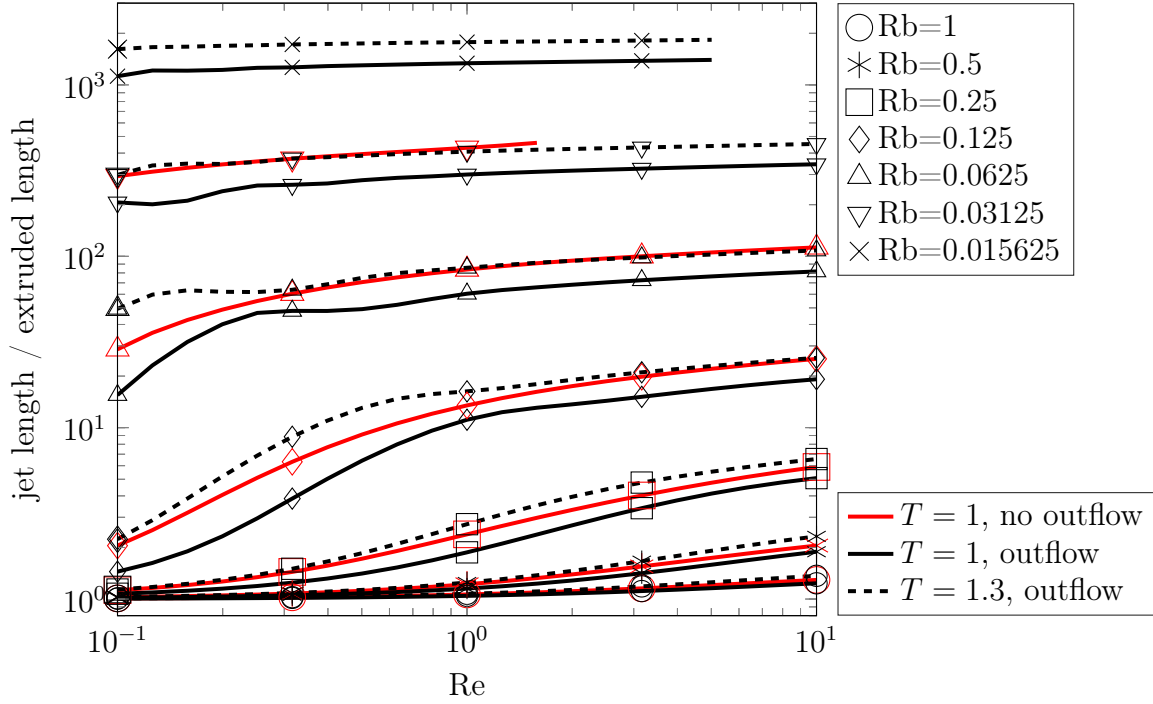


Figure 6.7: Total elongation in dependence of Re and Rb for the rotational spinning process with $(Fr, \varepsilon) = (1, 8.86 \cdot 10^{-2})$ and $\Delta s = \Delta t = 10^{-3}$. The no outflow is compared to two outflow cases with $v^R(t) = 0.3$ and the indicated end times. The parameter ranges are $Rb = \{1, 0.5, 0.25, \dots\}$ and $Re = \{10^{-1}, 10^{-0.9}, \dots, 10^1\}$, those parameter tuples that allow a successful simulation are indicated. Here, the jet length is $\int_{-l(T)}^{-r(T)} \partial_s \bar{r}(s', T) ds'$ and the extruded length $|\hat{\Omega}| = r(T) - l(T)$.

6.1.4 Industrial application

The results of Figure 6.6 allow to estimate the resolution requirements for the example in Table 6.1, which is $\Delta s = \Delta t < C_{Rb} Rb \approx 2.92 \cdot 10^{-5}$. This estimate takes the effects close to the nozzle and the revolution speed of the drum into account. As the jet extrudes and rotates around the drum we expect large elongation to form. It varies several orders of magnitude within the jet, which requires special consideration to avoid singular matrices of the non-linear system. Unsurprisingly, the approach of Section 5.3 is the key for a successful simulation: The elongation appears in second and third power in the material law. We divide the respective equation by the equal power of the maximum elongation from the previous time step. Additionally, an aggravation of the boundary layer on the tension-free end is avoided by choosing the mass outflow appropriately. We come to know by several targeted simulations that an outflow function $v^r(t) = 0.98$ enables the simulation of the relevant area between the nozzle and a possible conveyor belt. This choice seems drastic, but it avoids that the boundary layer arising from the tension-free end is affecting the solution, see Remark 6.3 for more details. The viscous jet and the resulting domain of the simulation are depicted in Figure 6.8, the simulation is stopped when sufficient height is produced. Our investigation shows that the jet is properly resolved, although a material parameterization is used. The deformation gradually increases within the jet but never

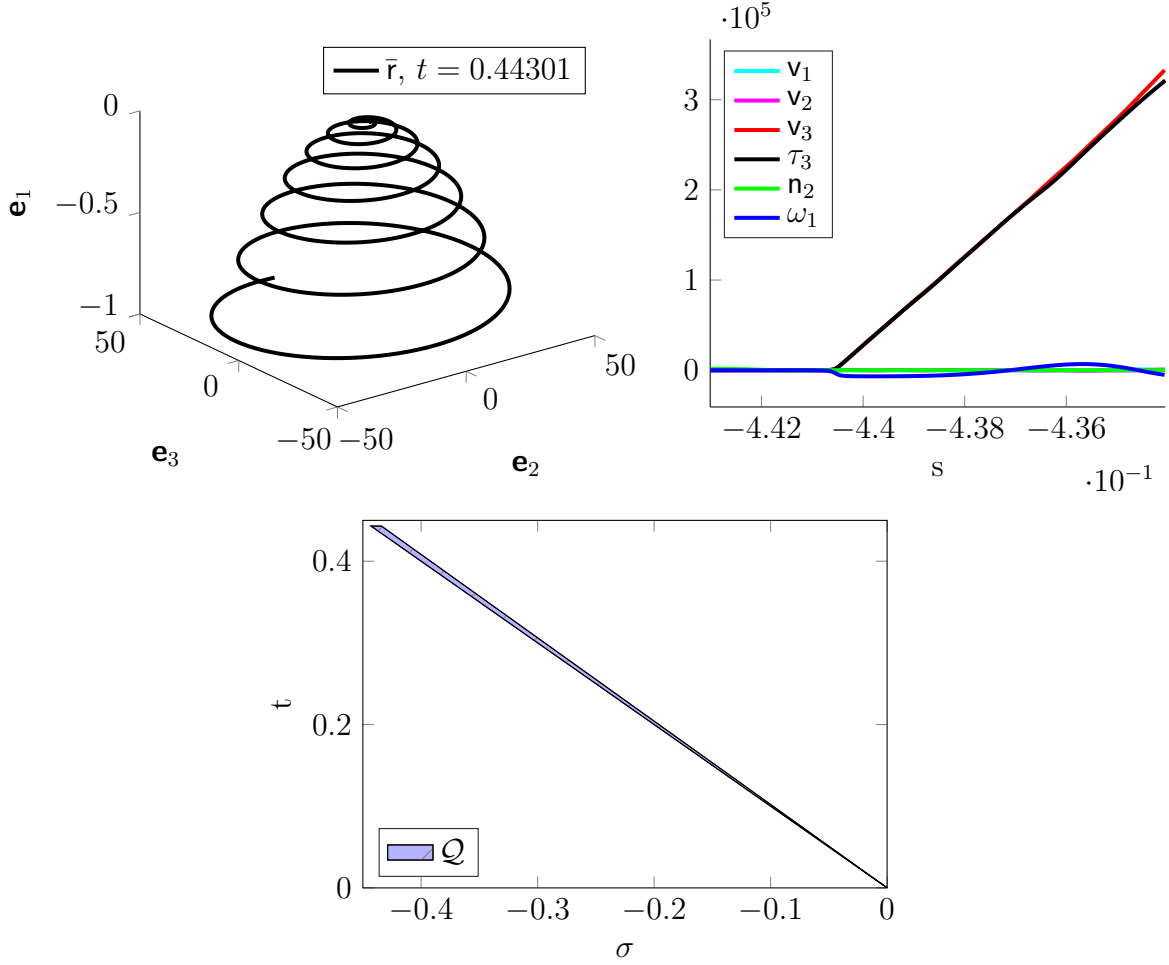


Figure 6.8: Industrial example of the rotational spinning process with parameters from Table 6.1 and $\Delta t = \Delta s = 10^{-5}$. The mass outflow is $v^r(t) = 0.98$.

explodes. Without the explicit elongation scaling of the material law the simulation breaks down due to a numerically singular system matrix when the elongation reaches roughly a magnitude of $5 \cdot 10^3$. The oscillation in the normal velocities, that are the bottleneck in the example of Figure 6.5, are properly resolved here and seem to be less severe. We suspect that this is due to the dominating rotational forces.

Remark 6.3 (Finding the outflow function). *The outflow function in Figure 6.8 is not an intuitive choice. We identify one reason the simulation breaks down, namely the influence of the free end that creates boundary layers (Figure 6.3) due to the inherent condition (2.41c). If those boundary layers are not properly resolved, the jet, and in particular its end, will behave unrealistic. We define two states for the jet in the rotational spinning process:*

- *Valid: The jet end moves with gravity, i.e.*

$$\operatorname{argmin}_{s \in \Omega(t_n)} \bar{r}_1(s, t_n) = \mathcal{R}(t_n).$$

- *Invalid: The jet end moves against gravity.*

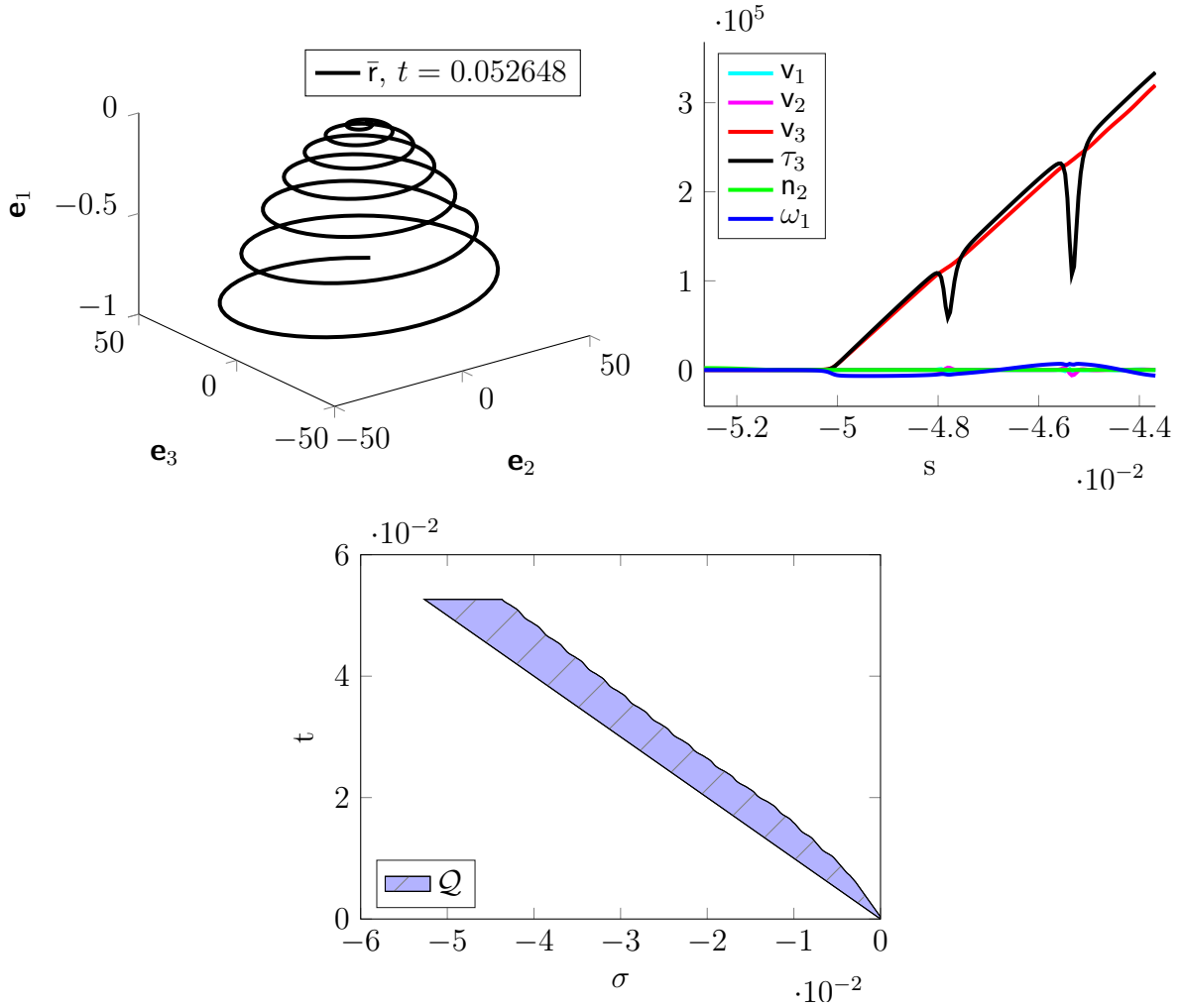


Figure 6.9: Industrial example of the rotational spinning process with parameters from Table 6.1 and $\Delta t = \Delta s = 2 \cdot 10^{-6}$. The heuristic mass outflow of Remark 6.3 was used.

With the help of those two states we set the outflow heuristically. We define a desired outflow function $v^{r,desired}$ that can be discontinuous and consists of three parts: An initial growth phase ($t_n \leq t_{growth} = 6 \cdot 10^{-3}$) in which we set $v^{r,desired}(t_n) = 0.5$ (these values have been determined by trial and error). After the initial growth we set $v^{r,desired}(t_n) = 0.5$ if the jet was valid at the time t_{n-1} , otherwise $v^{r,desired}(t_n) = 1.5$. The actual mass outflow is now chosen to approach the desired mass outflow in a smooth manner. This is done to avoid difficulties in the iteration of the Newton's method, the actual smoothing process has no significant effect (we choose to restrict the acceleration of the mass outflow by $\Delta t/10^{-3}$). The simulation and the resulting domain are depicted in Figure 6.9. We observe that the tension-free end is still affecting the solution, visible in the reverberating elongation kinks in Figure 6.9, right plot. The beginning of such a kink can also be seen for moderate parameters, cf. Figure 6.4. The shape of the domain suggests that instead of an heuristic approach for the mass outflow one could pick a analytic linear mass outflow function that approximately reflects the heuristic one. After the initial growth phase we find a slope of

0.88 to be a good approximation. We repeat the simulation by setting the linear outflow $v^r(t) = 0.88$ and obtain comparable results. Gradually increasing the mass outflow we find that the kinks disappear with an outflow of $v^r(t) \geq 0.98$.

We conclude that our jet model, set-up and discrete scheme greatly extends the applicable parameter range of the rotational spinning process, including the ones for industrially relevant simulations. A key aspect is the introduction of some mass outflow at the tension-free boundary to compensate for the boundary layer in the elongation at the tension-free jet end. The achievable elongation of the jet in the outflow case is seemingly not restricted due to proper scaling of the material law (cf. Section 5.3). This naive statement has to be re-evaluated in a realistic scenario with intrinsically correct material behavior (viscoelastic material law with temperature-dependent viscosity, cf. Section 2.1.3) and aerodynamic forces. Nevertheless, the capabilities of our approach are demonstrated in Figure 6.8, elongations of magnitude $\mathcal{O}(10^5)$ are achieved.

Remark 6.4. *We expect that the viscoelastic model facilitates the numerics whereas the aerodynamic forces worsen them due to their dominating effect on the jet. Even so, the aerodynamic forces are expected to be the highest in the area close to the nozzle, which will in turn allow to give a reasonable estimate on the necessary resolution. Since this is the main effect restricting the capabilities of the static grid (Lagrangian parameterization) in the rotational spinning process, we expect a viscoelastic model with aerodynamic forces to perform equally well if the initial resolution requirements can be met. Exemplary simulations of the thermal, viscoelastic jet in an academic scenario are shown in Appendix A.3.*

6.2 Melt-blowing process

The melt-blowing process is designed to manufacture nanofibers. Directed, highly turbulent air streams are used to rapidly deform polymer jets that are extruded through multiple nozzles. A typical production design of a melt-blowing machine is sketched in Figure 6.10. There is a chamber of some molten polymer that is extruded through the nozzles and undergoes aerodynamic forces that push the stretched jets onto some collecting structure, e.g. a conveyor belt, while cooling it down. The turbulent nature of the airflow is a key feature of such production processes since it forms the desired stochastic microstructures through overlapping of the deposited jets. The melt-blowing process is characterized economically by low production costs compared to the material costs, which makes it attractive for commercial use. A successful simulation would provide a valuable contribution to optimization efforts in terms of material properties of the microstructures and achievable jet thickness without reducing the machine throughput. So far there has been a gap in literature that deals with simulations of the melt-blowing process in regards to the computed jet thickness, in particular results from experiments and simulations deviate several orders of magnitude ([101] and more recently [112, 109]). This likely due to insufficient consideration of turbulent effects. To exemplify this, consider the elongation τ_3 , which is the ratio of the cross section at the nozzle and the deformed jet. In a stationary treatment of incompressible jets, the maximal elongation can be expressed as the ratio of the velocities (nozzle velocity vs. maximal jet velocity). The attainable maximal jet velocity is characterized by the

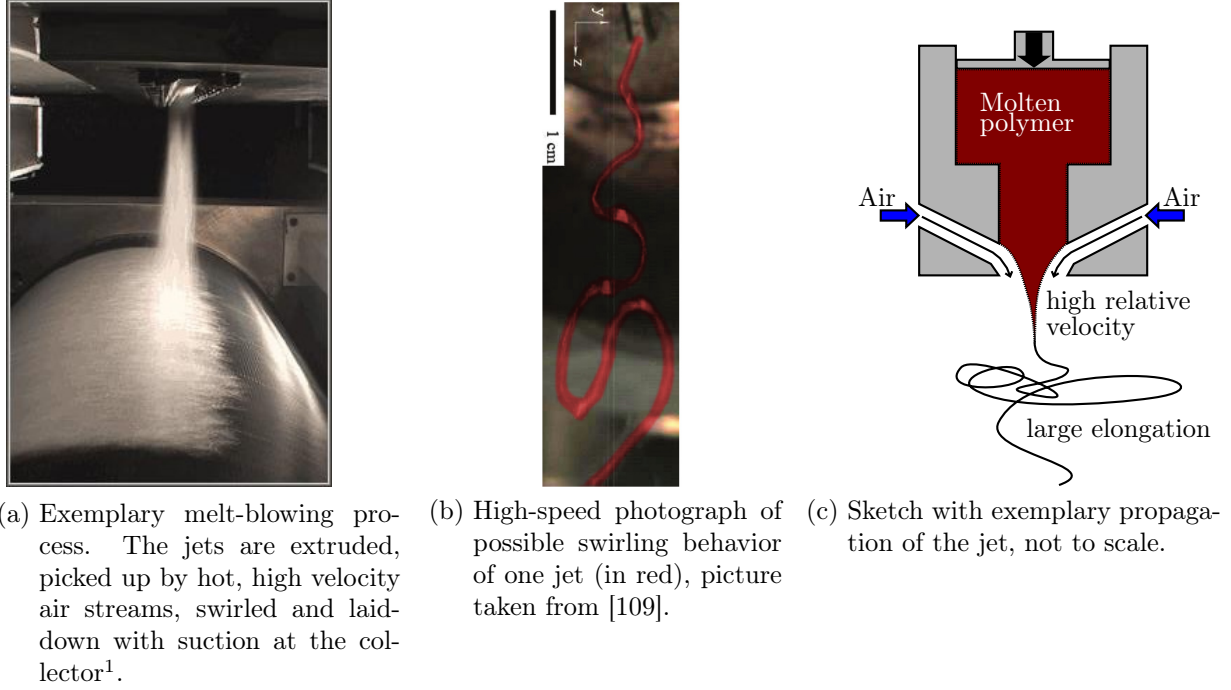


Figure 6.10: Illustration of a melt-blowing process.

mean airflow velocity and would cause an elongation of $\tau_3 \sim \mathcal{O}(10^4)$, whereas measurements of actual experiments show elongations of $\tau_3 \sim \mathcal{O}(10^6)$. Experiments in [93, 109] indicate the relevance of the turbulent fluctuations for the jet thinning, which prohibits any stationarity assumptions. This assumption was investigated numerically with a simplified ODE model for the jet by Hübsch et al. [60], who achieved promising results that raise hope to close the melt-blowing gap that is reported in literature. We aim to provide a basis for the robust simulation of the melt-blowing process with our more sophisticated viscous model and to provide more insight into the physical effects that can explain said melt-blowing gap.

Our interest is on the behavior of the jet in the flow situation, and especially the elongation behavior in in longtime simulations. The immersed jet moves exclusively in the distinct region prescribed by the air stream. Whereas its initial motion is determined strongly by the mean flow pulling the jet, the flow fluctuations cause slight bouncing that reinforce as the jet accelerates. When the jet velocity starts to follow and finally adjusts to the flow velocity, i.e. the relative velocity of jet and air goes to zero, the deformation of the jet is presumably dominated by the flow fluctuations. The aerodynamic forces are clearly the key player in the deformation of the jet, overall the applied external loads cover turbulent aerodynamic forces and gravity

Considering a representative melt-blowing scenario (Figure 6.11 and Figure 6.12), the jet behavior was explored with an ODE model and correlated aerodynamic force resulting from the turbulence reconstruction in [60]. In said scenario an airflow is directed vertically downwards and enters the domain via thin slot dies, the many nozzles are located inbetween

¹Picture reproduced with permission of the Textile Centre of Excellence, Textile House, Red Doles Lane, Huddersfield, HD2 1YF, enquiries@textilehouse.co.uk

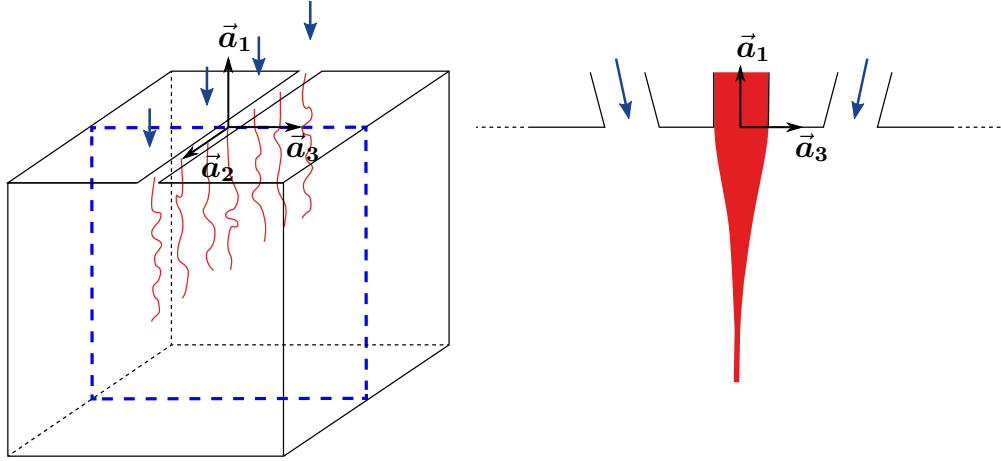


Figure 6.11: Illustration of the considered melt-blowing process. Left: Flow domain with immersed jets and indicated two-dimensional cut (\vec{a}_3 - \vec{a}_1 -plane, marked by dashed line) that is representative for the whole domain due to the homogeneity in \vec{a}_2 -direction. Right: Side view of the \vec{a}_3 - \vec{a}_1 -plane with zoom on slot dies with the nozzle inbetween. [60]

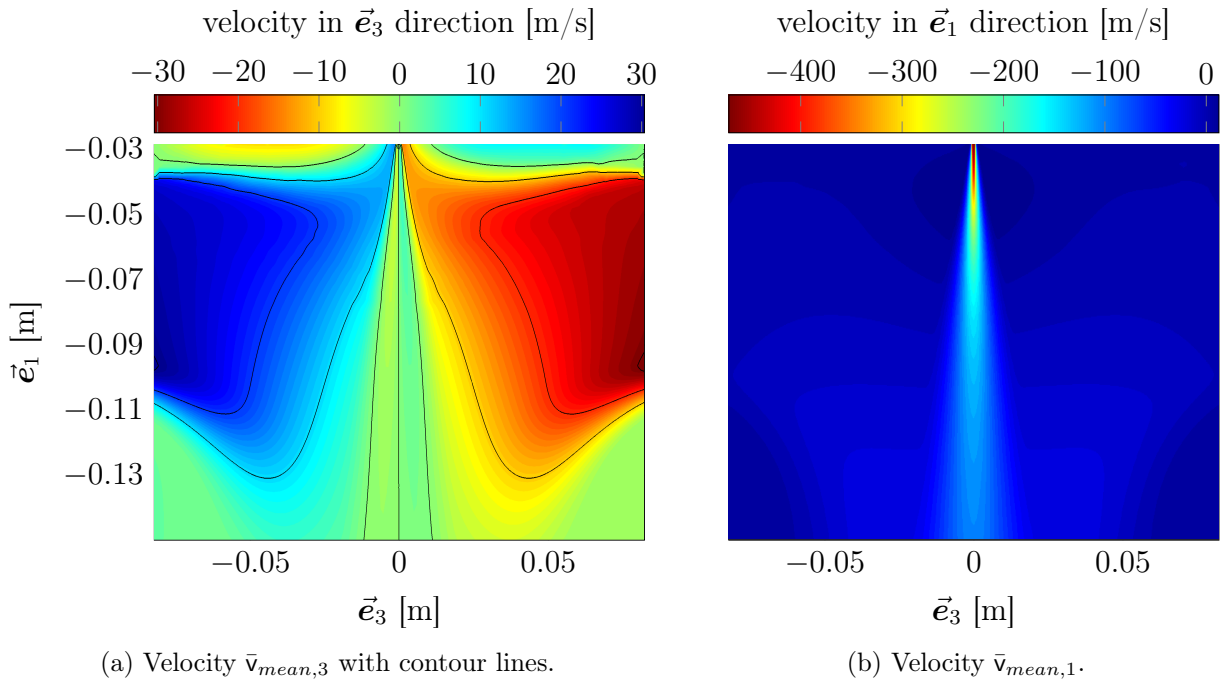


Figure 6.12: Components of the mean airflow velocity of the representative two-dimensional flow domain in Figure 6.11.

and point in the same direction. We study the scenario by help of our jet model and follow [60] by neglecting temperature effects and influences of the jets on the airflow (one-sided coupling through the aerodynamic line force). Furthermore, the outer basis in the sketch is chosen such that $\vec{\mathbf{a}}_2$ is aligned with the slot inlet, $\vec{\mathbf{a}}_3$ is in normal direction of the nozzles and $\vec{\mathbf{a}}_1$ in negative tangential direction. The airflow is time-independent and homogeneous in direction of the slot, thus a representative two-dimensional airflow can be used. The scenario is simplified to only consider one of the nozzles, i.e. jet interaction is neglected, and we put the origin of the fixed, outer basis of the jet model $\{\vec{\mathbf{a}}_1, \vec{\mathbf{a}}_2, \vec{\mathbf{a}}_3\}$ on top of the nozzle (that means that the boundary conditions for the jet position are always the origin). For the sake of an unambiguous visualization we introduce yet another outer basis $\{\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2, \vec{\mathbf{e}}_3\}$ that only differs in the choice of the origin from $\{\vec{\mathbf{a}}_1, \vec{\mathbf{a}}_2, \vec{\mathbf{a}}_3\}$ and assumes the coordinates given by the external flow data (Remark 6.5). In it, the nozzle position in the airflow field is at $-0.03\vec{\mathbf{e}}_1$ [m]). The typical values for the jet model are chosen analogously to the rotational spinning process (Section 6.1) with the exception of the typical length. It is chosen to be the height of the airflow field $r_\star = H$. The appearing physical parameters are described in Table 6.2. The kinematic viscosity of the airflow is assumed to be constant, whereas the density can not be considered constant due to the high velocities.

Summing up, the external loads consist of aerodynamic and gravitational forces:

$$\mathbf{f} = \mathbf{f}_g + \mathbf{f}_{air}, \quad \mathbf{l} = \mathbf{0}.$$

We handle the delicate jet-turbulence problem by help of two models: a drag force model for an incompressible flow acting on a slender fiber by Marheineke and Wegener [75] (cf. Section 6.2.1) and an approach for correlated turbulent fluctuations by Hübsch et al. [60] (cf. Section 6.2.2).

| Parameters | | | |
|---------------------------------|-------------------------------------|------------------------|-------------------|
| Description | Symbol | Value | Unit |
| Fiber density | ρ | $7 \cdot 10^2$ | kg/m ³ |
| Fiber dynamic viscosity | μ | 1 | kg/m/s |
| Nozzle diameter | D | $4 \cdot 10^{-4}$ | m |
| Extrusion speed | U | $1 \cdot 10^{-2}$ | m/s |
| Height of airflow field | H | $1.1986 \cdot 10^{-1}$ | m |
| Air kinematic viscosity | ν_{air} | $1.5 \cdot 10^{-5}$ | m ² /s |
| Dimensionless quantities | | | |
| Description | Formula | Value | |
| Reynolds | $\text{Re} = \rho U H / \mu$ | $8.39 \cdot 10^{-1}$ | |
| Froude | $\text{Fr} = U / \sqrt{g H}$ | $9.22 \cdot 10^{-3}$ | |
| Slenderness | $\varepsilon = (\sqrt{\pi}/2)(D/H)$ | $2.96 \cdot 10^{-3}$ | |
| mixed Reynolds | $\text{Re}_* = D U / \nu_{air}$ | $2.66 \cdot 10^{-1}$ | |

Table 6.2: Overview of the physical parameters of representative melt-blowing process from Figure 6.11, taken from [60]. The typical values are plugged in and the dimensionless quantities computed.

In the following subsections we introduce the models for the air drag force and the turbulent fluctuations. Both require the diameter of the slender jet, that is given in dimensionless form by $d : \mathcal{Q} \rightarrow \mathbb{R}^+$ with its typical value d_0 in accordance with Remark 2.17, i.e.

$$d = \sqrt{\frac{\sigma_V}{\|\boldsymbol{\tau}\|}}, \quad d_0 = 2\sqrt{\frac{\sigma_{V*}}{\pi}}. \quad (6.2)$$

6.2.1 Air drag model

The dynamics of a jet surrounded by an airflow depend on the drag forces that are imposed on the jet by the air. In principle, it can be computed by a coupling of jet and airflow with no-slip interface conditions. However, the needed high resolution (even with potential adaptive grid refinement) makes the direct numerical simulation of the three-dimensional fluid-solid-problem practically impossible for relevant applications. We adapt the air drag model from Marheineke and Wegener [75] that was specifically created for slender fibers in an incompressible flow and imposes a model for the jets line force density. It is constructed in an Eulerian parameterization and universally valid for all Reynolds number regimes, incident flow directions and all kind of fibers or jets. The model uses the Global-from-Local concept of [73] and globalizes by superposition. This simplified globalization strategy is applicable because the jet stays in the area of the free stream (depicted in Figure 6.12) and does not get close to the boundaries, i.e. it stays in an area where the Global-from-Local assumption holds. Locally, the jet is assumed to be a thin circular cylinder, and the force acting on it is exclusively caused by friction and inertia due to the incompressible flow assumption. The model for the aerodynamic line force is dependent on the material and geometrical properties as well as the specific inflow situation. The concept of [75] allows for a one-sided consideration of the airflow effect on the jets by using a filament-free airflow as a basis, but also for the complete coupling of jet and aerodynamics via additional source terms in the flow external computation. We use one-sided coupling.

The aerodynamic force \mathbf{f}_{MW} modeled in [75] is non-dimensionalized with typical mass, length and time chosen to be $\rho_{air} D_{jet}^3$, D_{jet} and D_{jet}^2/ν_{air} , respectively. The density ρ_{air} [kg/m³] and kinematic viscosity ν_{air} [m²/s] of the airflow and the diameter D_{jet} [m] of the jet are used. Note that ν_{air} is assumed to be constant whereas ρ_{air} and D_{jet} are functions on \mathcal{Q} . More precisely, ρ_{air} is given through an external computation, and $D_{jet} = dd_0$, cf. (6.2). The aerodynamic line force in director basis that is suitable for the jet model in terms of its non-dimensionalization is given by

$$\mathbf{f}_{air} = F \frac{\tau_3}{d} \mathbf{f}_{MW}(\mathbf{e}_3, \text{Re}_* d\mathbf{v}_{rel}), \quad (6.3)$$

$$F = \frac{\sqrt{\pi}}{2} \frac{\rho_{air} \nu_{air}^2 r_*}{\sigma_{M*} v_*^2 \sqrt{\sigma_{V*}}} = \frac{1}{\varepsilon^2} \frac{1}{\text{Re} \text{Re}_*} \text{Vi}_*, \quad \text{Vi}_* = \frac{\rho_{air} \nu_{air}}{\mu_*}, \quad \text{Re}_* = \frac{2}{\sqrt{\pi}} \frac{\sqrt{\sigma_{V*}} v_*}{\nu_{air}}$$

with the typical values of the jet model, cf. Section 2.2.2, and the dimensionless numbers Re , Re_* and Vi_* , where Re is the Reynolds number of the jet, Re_* is a Reynolds number relating the inertial forces of the jet to the viscous ones of the airflow and Vi_* relates the dynamic viscosity of the airflow and the jet. Note that Vi_* (and therewith F) is not a constant but a function dependent on \mathcal{Q} through ρ_{air} and F is the conversion factor

between jet and air drag model. Whereas \mathbf{f}_{air} is referenced to the jet parameter s , note that \mathbf{f}_{MW} is referenced to arc-length, thus the jet elongation $\|\boldsymbol{\tau}\| = \tau_3$ additionally appears in (6.2.1). The model behind \mathbf{f}_{MW} depends on the normed tangent of the jet and the flow velocity relative to the jet (we refer to [75] for details). Since the air drag model is frame-invariant and we are using the director basis, the normed tangent simplifies to \mathbf{e}_3 . The relative incident velocity is

$$\mathbf{v}_{rel} = \mathbf{R}(\mathbf{q}) \cdot \bar{\mathbf{v}}_{air} - \mathbf{v}. \quad (6.4)$$

Here we require the dimensionless airflow velocity $\bar{\mathbf{v}}_{air}$ at the current position of the jet. It is provided by an external computation and interpolated onto the jet position, cf. Remark 6.5. The underlying model assumes that the airflow velocity consists of the mean flow and turbulent fluctuations

$$\bar{\mathbf{v}}_{air} = \bar{\mathbf{v}}_{mean} + \bar{\mathbf{v}}_{turb}. \quad (6.5)$$

The turbulent fluctuations are either zero for purely deterministic simulations or will be given by the turbulence reconstruction that is described in Section 6.2.2.

6.2.2 Turbulent reconstruction

Marheineke and Wegener [73] derive a correlated aerodynamic force concept and argue the asymptotic transition to Gaussian white noise with flow-dependent amplitude. If the bending scales of the jet are significantly larger than the turbulent length scales, then the white noise limit is qualitatively and quantitatively justifiable and gives an enormous simplification and reduction of computational time and memory usage (investigated numerically and validated in [67, 75]). The impact of the correlated global force has been explored in [60]. The reconstruction for the fluctuations is on top of a k - ϵ formulation with a simplified energy spectrum from [73]. The underlying Global-from-Local concept assumes that the local velocity fluctuations (fine-scale structure) can be modeled as homogeneous, isotropic Gaussian random fields. This is ad-hoc possible if the kinetic energy k_{air} [m^2/s^2] of the turbulent fluctuations and the viscous dissipation of the turbulent motions per unit mass ϵ_{air} [m^2/s^3] only vary gradually within the considered airflow. The local fluctuations are globalized by superposition to form the large-scale structure of the global turbulence. In [60] Hübsch et al. propose a fast and accurate sampling procedure for the random fields that shows qualitatively appropriate jet thinning in magnitude for a simple isothermal ODE jet model. The effort is linear in the discretization making a realization possible for industrial applications. We apply their approach to the more complex viscous jet model.

The airflow velocity (6.5) is assumed to consist of a mean flow and turbulent fluctuations. The stochastic part is reconstructed with the help of the mean flow $\bar{\mathbf{v}}_{mean}$, the kinetic energy k_{air} and viscous dissipation ϵ_{air} . All three are functions on \mathcal{Q} whose values are interpolated from data given by an external flow simulation, cf. Remark 6.5. The turbulent velocity fluctuations $\bar{\mathbf{v}}_{HM}$ modeled in [60] depend on the current position in the airflow field (given by the jet position) and time, as well as the viscosity and mean velocity of the airflow at said position, and are made dimensionless with the turbulent length $l_T = k_{air}^{3/2}/\epsilon_{air}$ and

time $t_T = k_{air}/\epsilon_{air}$. The velocity fluctuations $\bar{\mathbf{v}}_{turb}$ suited for the jet model in terms of its non-dimensionalization are given by

$$\begin{aligned}\bar{\mathbf{v}}_{turb} &= \frac{k_{air}^{1/2}}{v_\star} \bar{\mathbf{v}}_{HM} \left(\frac{\epsilon_{air}}{k_{air}^{3/2}} r_\star \bar{\mathbf{r}}, \frac{\epsilon_{air}}{k_{air}} \frac{r_\star}{v_\star} t, \frac{\epsilon_{air}}{k^2} \nu_{air}, \frac{v_\star}{k_{air}^{1/2}} \bar{\mathbf{v}}_{mean} \right) \\ &= Tu \bar{\mathbf{v}}_{HM} \left(\frac{Tt}{Tu} \bar{\mathbf{r}}, Tt, \frac{2\varepsilon}{\sqrt{\pi} Re_\star} \frac{Tt}{Tu^2}, \frac{1}{Tu} \bar{\mathbf{v}}_{mean} \right), \quad Tu = \frac{k_{air}^{1/2}}{v_\star}, \quad Tt = \frac{\epsilon_{air} r_\star}{k_{air} v_\star}\end{aligned}$$

with the dimensionless degree of turbulence Tu and time scale ratio Tt , both of which are not constants but functions dependent on \mathcal{Q} .

6.2.3 Numerical method and investigation

Hübsch et al. were investigating the turbulent fluctuations as possible cause for the large elongation measured in melt-blowing processes [60]. We continue their study by applying the turbulence reconstruction to the viscous jet model as the consequent advancement from the ODE model in [60]. We consider the jet extrusion set-up (Section 2.3.1) and use the frame for the initial and boundary conditions provided by (2.43). Starting with $l(0) = r(0) = 0$ we set

$$\bar{\mathbf{r}}^L = \mathbf{0}, \quad \mathbf{q}^L = \frac{1}{2}(-1, -1, -1, 1), \quad v^L = 1.$$

If not otherwise mentioned, we assume $\mathcal{L}(t) = l(t)$, $\mathcal{R}(t) = r(t)$ and $v^r = 0$.

Remark 6.5 (External data and routines). *All external data and routines are provided by the Fraunhofer ITWM. The flow data is provided with permission from their industrial partner and was obtained by an external simulation of the incompressible Navier-Stokes equations, namely via the commercial tool FLUENT². The air drag force model is available as a MatLab routine (according to [75]) whereas the turbulence reconstruction is made in MatLab through a `.mex` interface written in C++ (for performance reasons, according to [60, Algorithmus 9]). The computation of the air force is quite CPU-intensive, also derivatives of the turbulent reconstruction are not available. Thus the air force is best used explicitly in the jet model.*

Necessary spatial and temporal resolution The model for the turbulence reconstruction (cf. Section 6.2) is built with the turbulent large-scale length t_T [s] (expected length of the large-scale vortices) and time l_T [m] (expected creation and break-up time of the vortices). Both depend on the position in the airflow and have to be considered when choosing an appropriate resolution. We are interested in the resulting force on the jet as a slender object that moves freely in the airflow. Thus there is an additional challenge for the temporal resolution: the time that a vortex needs to pass a fixed material point of the slender object due to their relative velocity. Without a doubt those scales have to be resolved properly in the discretization, otherwise the turbulent force is not affecting the jet

²commercial finite volume-based software by ANSYS, <http://www.fluent.com/>.

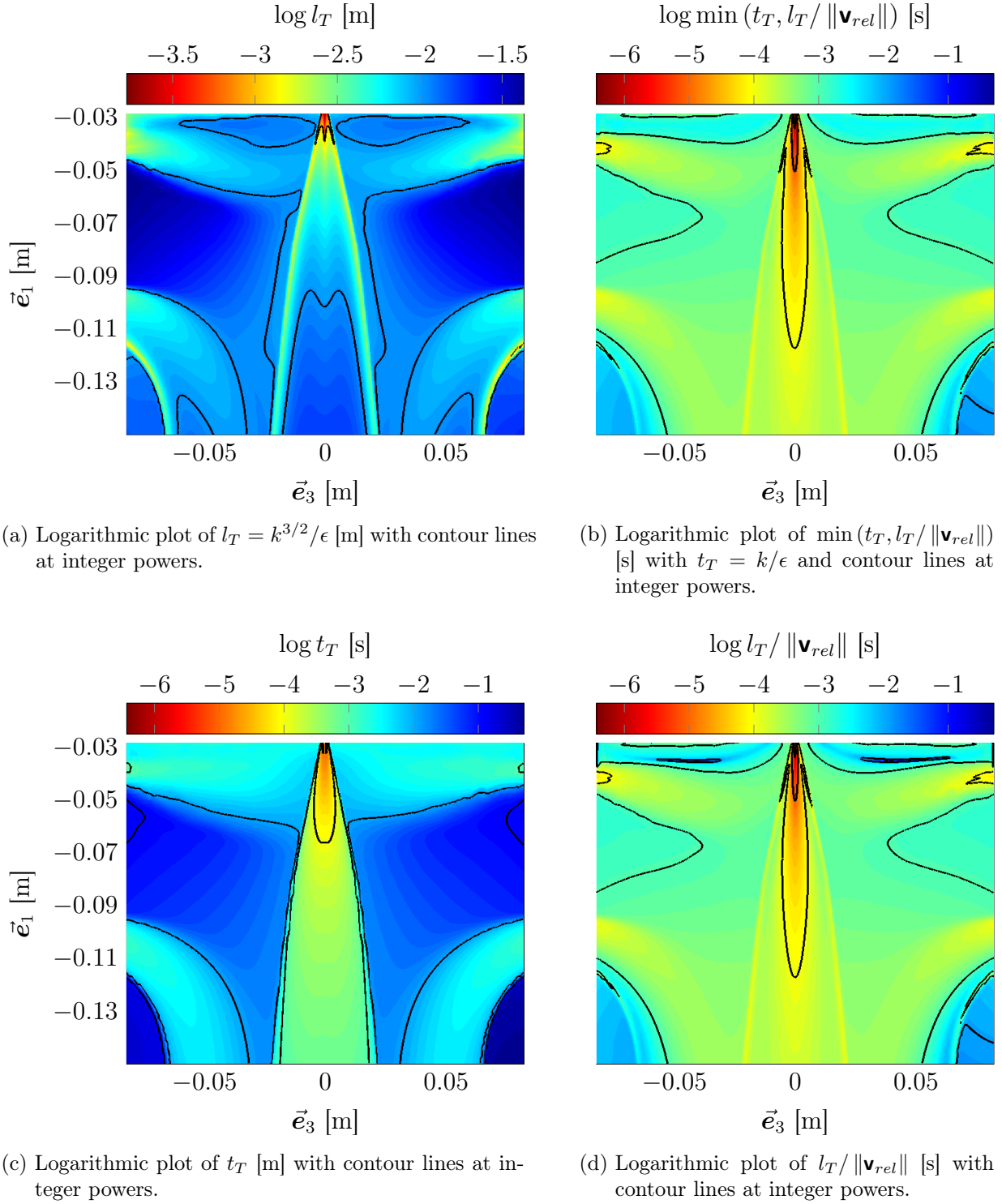


Figure 6.13: Dimensioned turbulent space and time scales for the industrial example of Figure 6.11 in a) and b). Comparison of the two relevant turbulent time scales in c) and d).

accumulatively and becomes white noise. The requirements for a successful simulation in terms of Δs , Δt depend on the underlying non-dimensionalization and parameterization. We formulate the upper bounds along the jet for every $(s, t) \in \mathcal{Q}$

$$\Delta s \|\boldsymbol{\tau}\| \leq \frac{l_T}{H}, \quad (6.6a)$$

$$\Delta t \leq \frac{U}{H} \min \left(t_T, \frac{l_T}{\|\mathbf{v}_{rel}\| U} \right) \quad (6.6b)$$

with \mathbf{v}_{rel} from (6.4) and the typical values from the jet model given in Table 6.2. An estimate for the upper bound of the temporal and spatial resolution is given by the right sides of (6.6) and can be obtained by assuming that the jet velocity is simply the typical velocity. To find the global minimum of said bounds we calculate it for all airflow data that is in front of the nozzle, i.e. with a negative $\vec{\mathbf{a}}_1$ coordinate. The minimum is then computed and it is, as expected, located close to the nozzle (cf. Figure 6.13) and given by

$$\Delta s \leq \frac{1.53 \cdot 10^{-4}}{H} = 1.18 \cdot 10^{-3}, \quad (6.7a)$$

$$\Delta t \leq \frac{4.12 \cdot 10^{-7}}{H/U} = 3.05 \cdot 10^{-8}. \quad (6.7b)$$

6.2.4 Industrial example

The time step restriction (6.7) seems too harsh for actual simulations, yet this is the worst-case only in close proximity to the nozzle. A simulation is indeed possible, but the runtime is insufficient. In Figure 6.14 several representative jets due to the turbulence reconstruction are depicted, the simulation runtime is quite short. Longtime simulations are not feasible, since the expected elongation of this industrial example is of magnitude 10^6 . Considering the impact of the elongation in (6.6), the spatial grid spacing would need to be at least $\Delta s = 10^{-7}$, in an optimistic guess according to Figure 6.13a (the worst case in (6.7a) would be of magnitude 10^{-9} , but the large elongations do not occur that close to the nozzle). Using said resolution, simply extruding the jet until it reaches the arc-length depicted in Figure 6.14 exceeds our computational capabilities, or more precisely, are no longer feasibly in terms of computation time.

Deterministic dominance We alter our simulation approach by hypothesizing that the turbulent fluctuations have neglectable influence on the deformation of the jet when the relative velocity is very high, i.e. when the jet just exited the nozzle. In other words, in a certain area directly after the nozzle the jet is dominantly deformed by the mean airflow. The question is the extent of said area such that our assumption is reasonable.

We estimate the velocity induced solely by the mean flow. For that purpose we assume that the sole external force is induced by the mean flow in nozzle direction. This simplifies the scenario drastically to an uniaxial model, which is presented in Appendix B.3. Since we are interested in longtime simulations we can assume that the jet enters a stationary state within the nozzle-adjacent area. Consequently the velocity profile of the jet can be

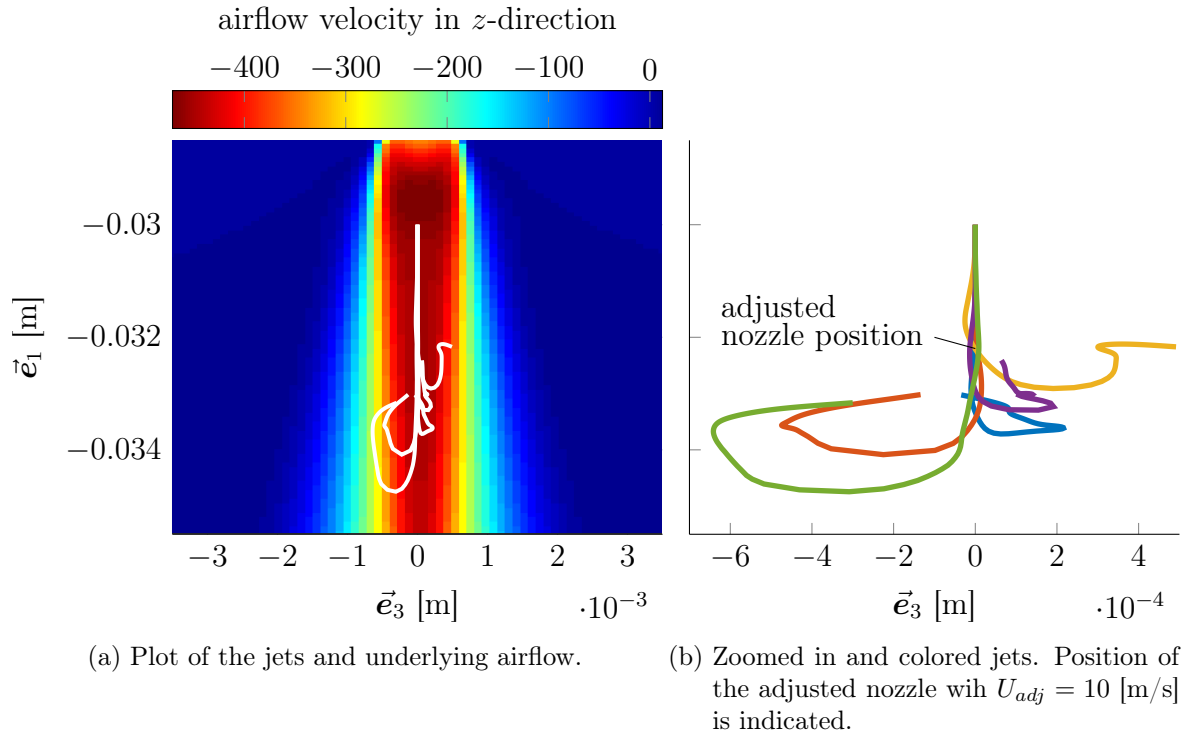


Figure 6.14: Simulation of the melt-blowing process with parameters from Table 6.2 and $\Delta s = \Delta t = 5 \cdot 10^{-7}$. The plots show representatives of jet curve at time $t = 1.8 \cdot 10^{-3}$ due to the turbulence reconstruction.

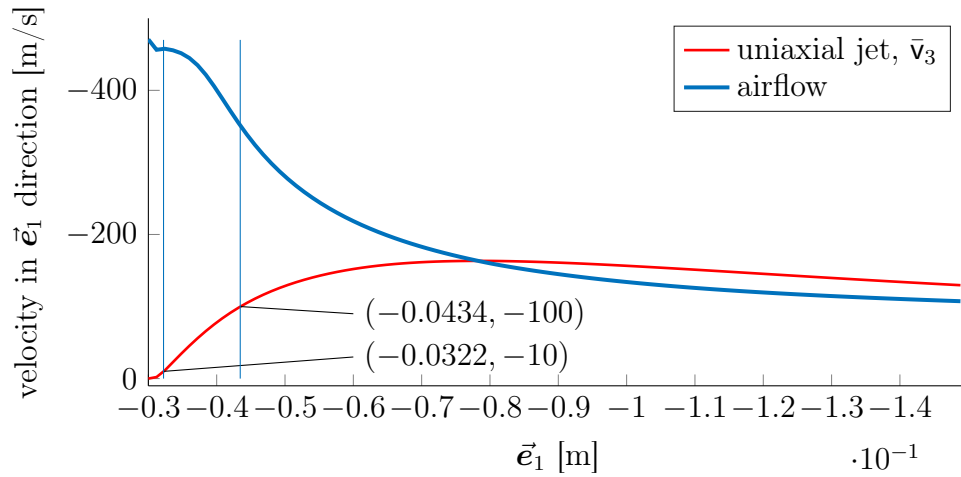


Figure 6.15: Velocity profile of a jet only considering non-turbulent airflow in the direction of the nozzle. The jet position and its velocity are depicted for $(\bar{r}_1, \bar{v}_3) = (-0.0322$ [m], -10 [m/s]) and $(\bar{r}_1, \bar{v}_3) = (-0.0434$ [m], -100 [m/s]).

calculated with a stationary simulation in Eulerian parameterization that spans the whole area. The velocity profile is plotted in Figure 6.15.

The simulation is altered with the following idea: Choose an arbitrary speed U_{adj} greater than the extrusion speed U of the nozzle. Then the steady-state simulation allows to estimate the jet position with the help of the velocity profile in Figure 6.15 (meaning: Read the position of the jet when it has velocity equal to U_{adj}). We put a fictive nozzle at said position (which will be the origin of the outer basis $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$) and adjust the typical values of the simulation to obtain the same mass flow as with the original nozzle (cf. Table 6.2), i.e.

$$v_\star = U_{adj}, \quad \sigma_{M\star} = \rho \frac{\pi D^2}{4} \frac{U}{U_{adj}}, \quad \sigma_{V\star} = \frac{\pi D^2}{4} \frac{U}{U_{adj}},$$

which means that the diameter of the fictive nozzle is adjusted by a factor of $\sqrt{U/U_{adj}}$. With the adjusted velocity and position, the dimensionless number and the necessary turbulent scales relax (we simply neglect the airflow that is behind the nozzle, and the relative velocity decreases). We choose $U_{adj} = 10$ [m/s], which puts the adjusted nozzle position in the airflow field at $-0.0322\vec{e}_1$ according to Figure 6.15. The diameter of the adjusted nozzle becomes $D = 1.26 \cdot 10^{-5}$, which translated to a preassumed elongation compared to the original nozzle of $U_{adj}/U = 10^3$, and the dimensionless numbers become $(\text{Re}, \text{Fr}, \varepsilon) = (8.23 \cdot 10^2, 9.31, 9.53 \cdot 10^{-5})$. Of course the jets extruded with the original parameters are not passing exactly through the adjusted nozzle position $-0.0322\vec{e}_1$, but this deviation is neglected (cf. Figure 6.14b). We will be using the adjusted nozzle for the following investigation.

Investigation of the turbulent effect causing large elongations We run a simulation with parameters of Table 6.2 and the adjusted nozzle with $U_{adj} = 10$. The applied airflow field is simply the velocity 400 [m/s] in negative \vec{e}_1 direction. We observe that the turbulent vortices create swirls in the jet and pull it in opposite directions due to velocity gradients (cf. Figure 6.16). Sooner or later the fluctuations cause the jet to position itself normal to the high-velocity airflow, creating large normal aerodynamic forces due to high relative velocity gradients (cf. Figure 6.17a). If those external forces can accumulate over time, a spike in the elongation is formed that easily exceeds the deterministic expectations. In Figure 6.17b those forces have created such an elongation peak. We can see in the jet and airflow velocities that the jet is closely following the airflow, the induced forces are more uniform indicating that the jet will only be stretched up to a certain maximum elongation, at which point the jet is simply following the airflow low.

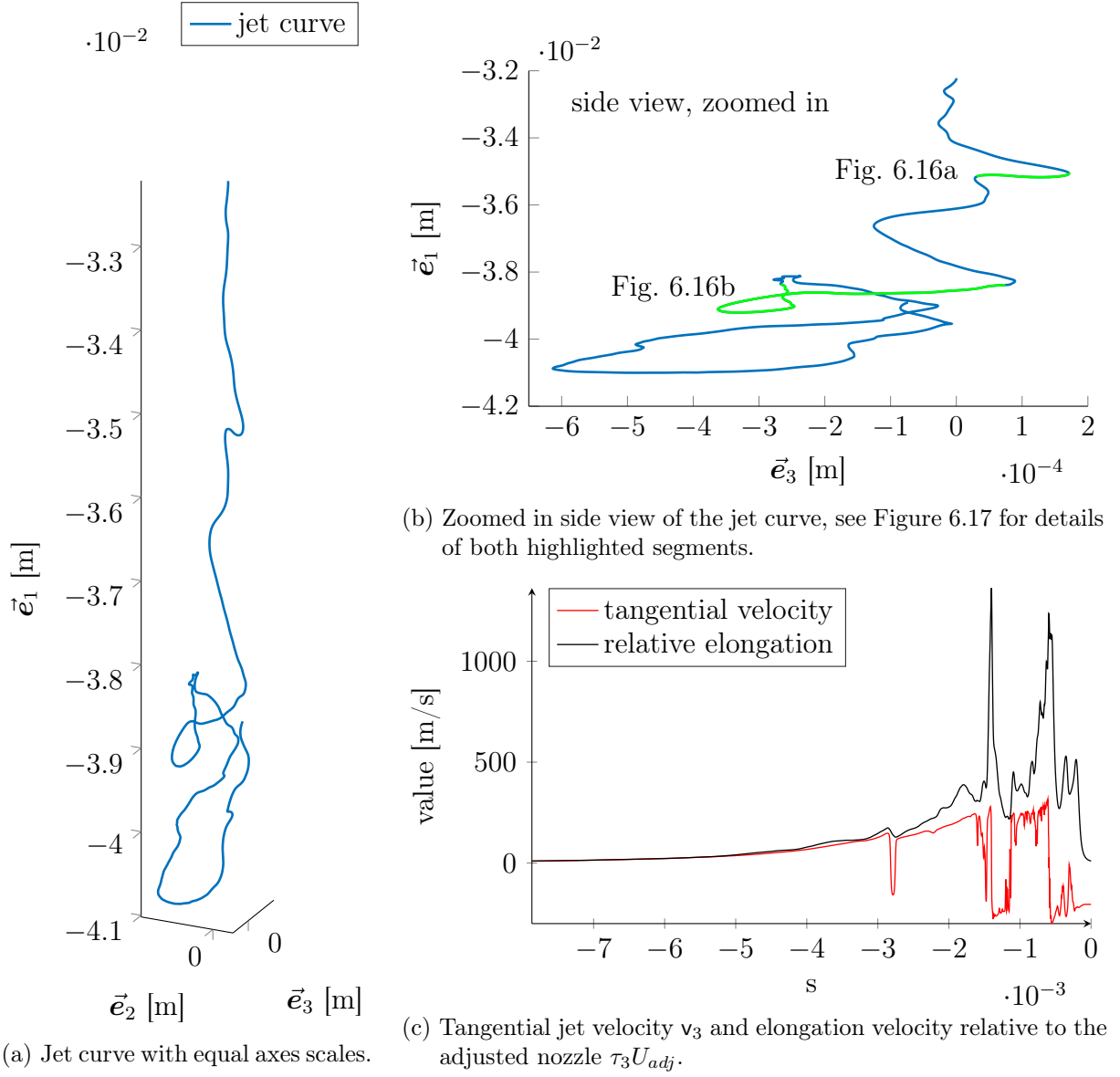
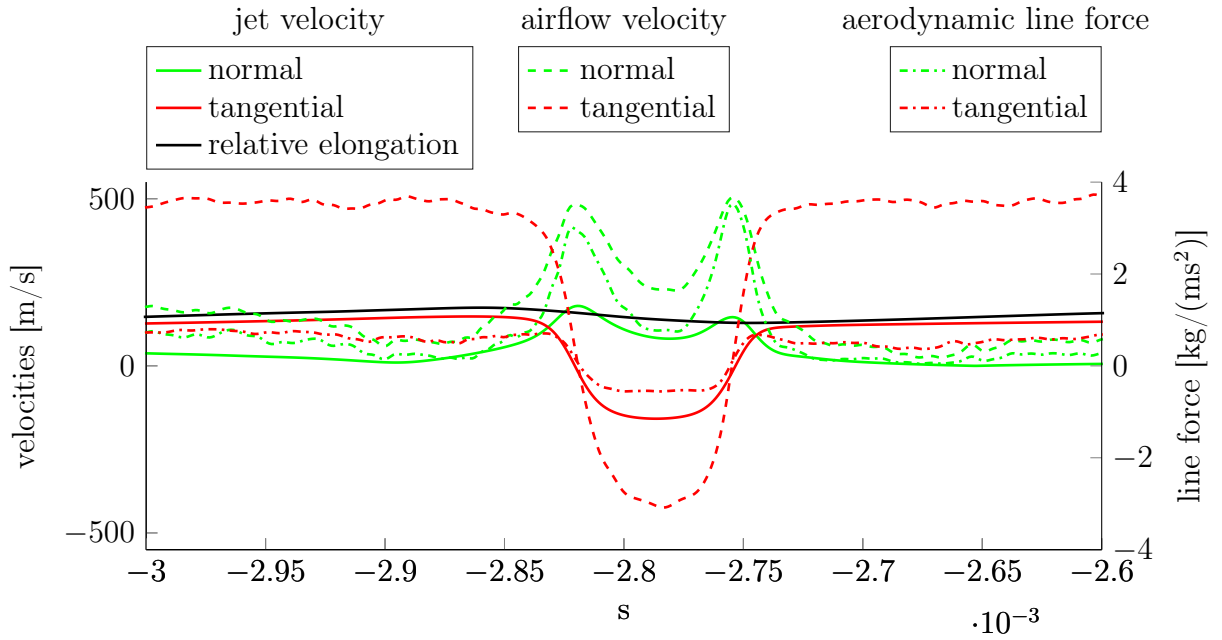
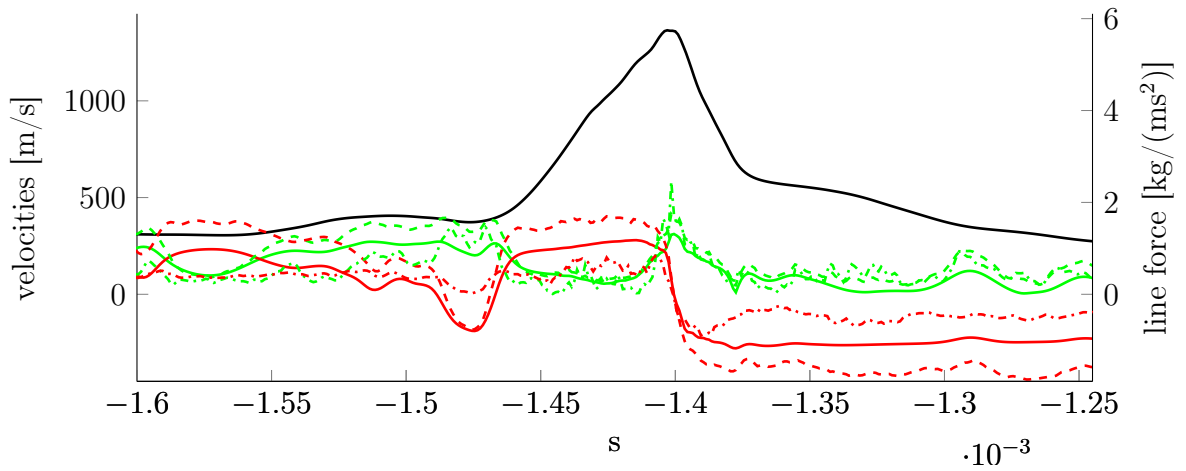


Figure 6.16: Simulation of the melt-blowing scenario with parameters from Table 6.2 and $\Delta s = \Delta t = 10^{-6}$ with an adjusted nozzle and airflow, the time is $t = 0.0078$.



(a) First highlighted area in Figure 6.16. Due to the strong pull in normal direction of the jet, the jets tangential velocity changes its sign.



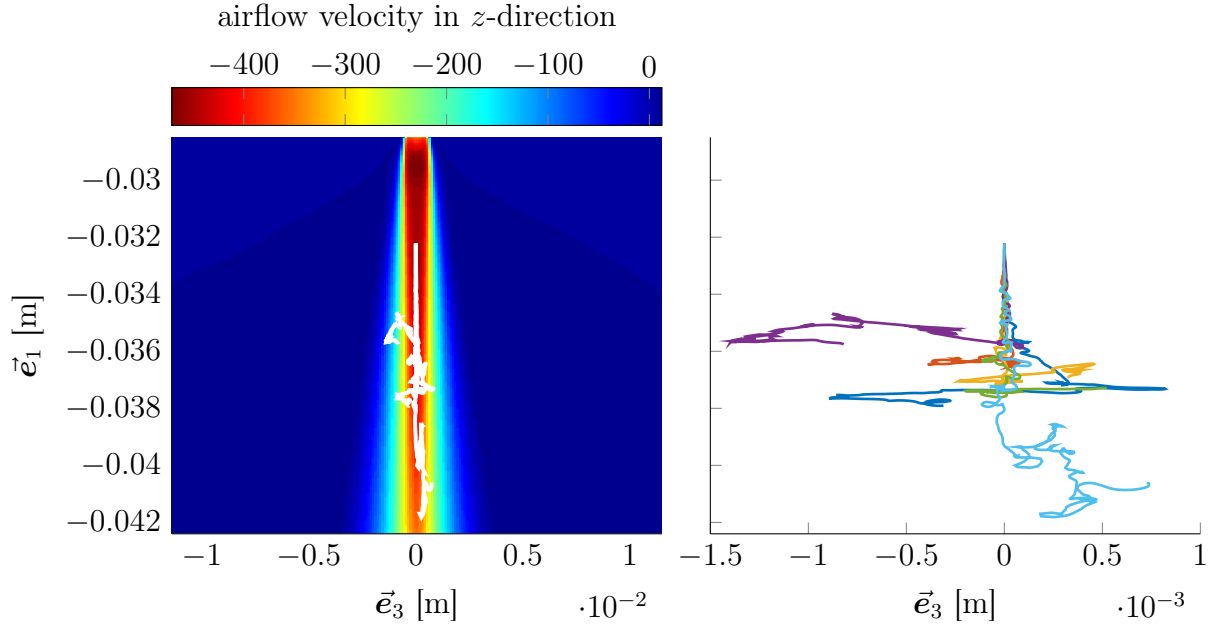
(b) Second highlighted area in Figure 6.16.

Figure 6.17: Details of the simulation depicted in Figure 6.16: The plots show the dimensioned normal ($\|\cdot \times \mathbf{e}_3\|$) and tangential component (\cdot_3) of the jet \mathbf{v} and airflow velocity \mathbf{v}_{air} as well as the induced aerodynamic line force \mathbf{f}_{air} and the elongation velocity relative to the adjusted nozzle $\tau_3 U_{adj}$.

We repeat the simulation with the actual airflow from Figure 6.11. The jet curve of different representations due to the turbulence are shown in Figure 6.18. We can observe maximum total elongations (compared to the original nozzle) of 10^{-5} to 10^{-6} within those jets. They all exhibited similar effects, the tangential velocity and elongation of one representation is depicted in Figure 6.19. Very high elongation peaks appear and we expect in a prolonged simulation that the jet is evenly deformed due to inertial effects (if the jet is thick, it is not following the airflow and velocity gradients can occur during swirling. If it is thin, it simply follows the airflow without anymore significant deformation). Notice the many sign changes of the tangential velocity in Figure 6.19c, in all those areas high normal forces are acting on the jet, possibly creating elongation peaks.

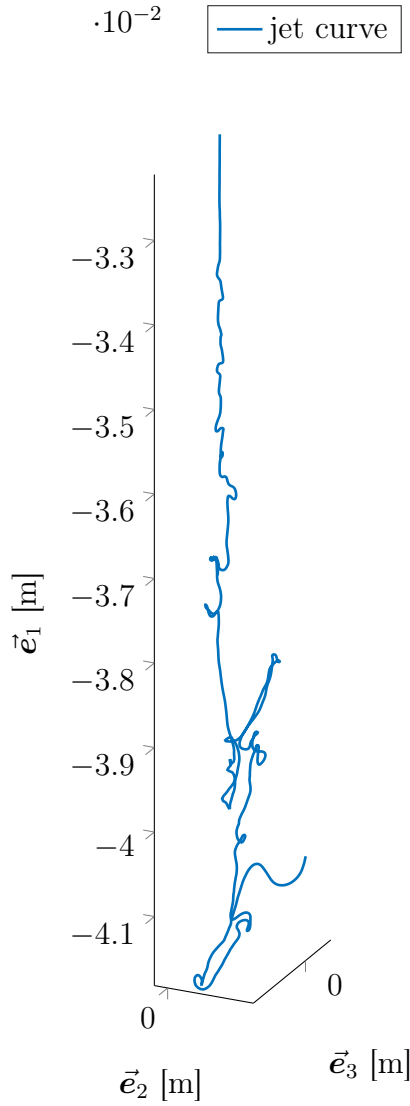
Unfortunately, our computational capabilities are not sufficient to run longtime simulations with a Lagrangian parameterization to reproduce jets that have an evenly distributed elongation of the same magnitude. The moving mesh approaches from Chapter 3 are tailored exactly for this problem of locally strongly varying solution components, but such simulations (more precisely, with the strategies and mesh control function from Section 5.4.2) start to oscillate and break down before its Lagrangian counterpart does. We hypothesize that a smoother, higher-order approximation for the artificial spatial fluxes of the jet model in general parameterization can overcome that issue, cf. Remark 5.5. Even so, the performance of the jet model in Lagrangian parameterization is already promising and we do not preclude that the use of those smoother, higher-order approximation for the material spatial fluxes can extend the simulation runtime to be suitable for longtime considerations.

Despite all that, our viscous jet model and discrete scheme allows the general simulation of the melt-blowing process in industrially relevant parameter ranges if an adjusted nozzle is used. The impact of such an adjusted nozzle needs to be investigated in more detail, but simulations show that the underlying turbulent fluctuations can in fact create jet swirling that creates elongation peaks that well surpass the results induced by a non-turbulent airflow. Furthermore we observe that there is no significant further deformation when the jet's inertia becomes so small that it simply follows the airflow, which explains that there is an expected maximum elongation of the process.



(a) Plot with equal scales and the underlying airflow. (b) Zoomed in on the jets that are distinguished by color.

Figure 6.18: Industrial example of the melt-blowing process with parameters from Table 6.2. A fictive nozzle with $U_{adj} = 10$ [m/s] and $\Delta t = \Delta s = 10^{-6}$ is used. The plot shows the jet curve of various representations depicted at $t = 0.0113 \pm 0.01$, the maximum elongation compared to the original nozzle $(U_{adj}/U)\tau_3$ is stated in the legend. A continued simulation of the jets would quickly break down or behave unrealistic due to a lack of resolution in the elongation peaks.



(a) Jet curve with equal scales.

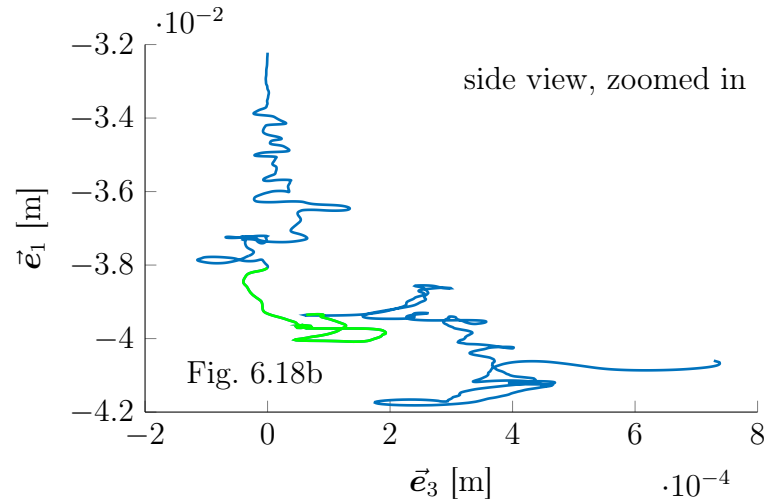
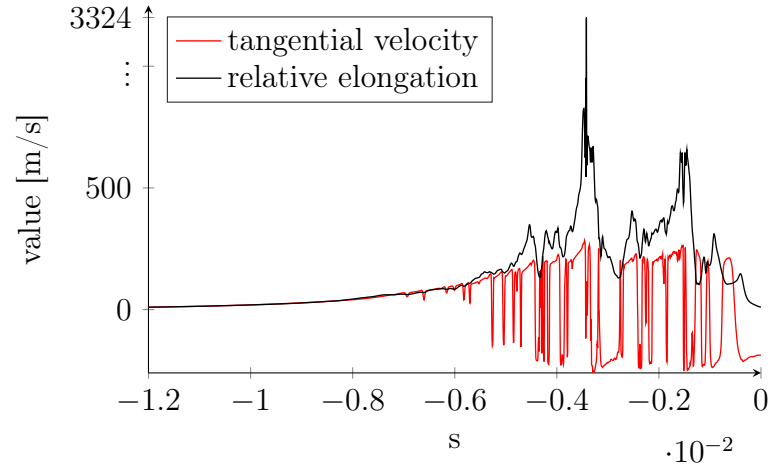
(b) Zoomed in on the jet curve with the area around the highest elongation (at $s = -3.423 \cdot 10^{-3}$) highlighted.(c) Tangential jet velocity v_3 and elongation relative to the adjusted nozzle $\tau_3 U_{adj}$. Highest peak at $s = -3.423 \cdot 10^{-3}$ (not in scale), which is compared to the original nozzle a total elongation of $3.32 \cdot 10^5$.

Figure 6.19: A detailed plot of one of the representations of Figure 6.18, the time is $t = 0.012002$. Notice the plenty sign changes in the tangential velocity indicating the swirling of the jet that accumulatively can cause elongation peaks.

7 Conclusion

We gave a detailed derivation of the one-dimensional Cosserat rod model describing the dynamics of a three-dimensional slender object in Section 2.1. The general form of the model (Section 2.2) allows the simple exchange of geometric and material behavior. Furthermore a stabilized, index-reduced formulation of the kinematics is used that allows the usage of various set-ups (including, but not restricted to, jet extrusion with free end, free falling jet, jet drawing), and more importantly, does not suffer from convergence order reduction effects compared to [5]. The considered set-ups are built with different combinations of the position and tension boundary, which are discussed in Section 2.3, especially with respect to the underlying inherent relations.

The model is formulated in a general parameterization that allows the use of r -refinement, or moving mesh, which we introduced in Chapter 3 with a new view on the inner structures, interpreted through a framework of three parameterization layers. We showed the link to two existing moving mesh strategies with the help of our framework and introduce two new strategies, yet no claim on their competitiveness is made. All introduced strategies do in fact produce valid meshes for time-dependent spatial domain – which was not shown so far in literature [59] – and they are applicable to the jet model. The strategies are validated numerically with the broadly used Burger’s equation in Appendix C.

In Section 4.1 we proposed a discrete scheme that looks upon the space-time domain as a two-dimensional domain and employs a Finite Volume method on it. We use simple quadrilateral cells for the proper approximation of the space-time boundaries, while allowing decoupling in time for iterative solving. Further on we employ a central approximation for the spatial fluxes and a fully implicit approximation of the temporal fluxes. The staggered grid approach allows here the use of a narrow, direct-neighbor stencil. The entire discrete scheme is formulated completely independent of the underlying model equations, nevertheless the approximations are tailored for the underlying DAE character of the jet model. In general, the scheme is applicable to PDAEs with index 1 in space and index 2 in time, and even higher indices when special structures are present. Details for the application to the jet model and the moving mesh strategies are given in Section 4.2 and Section 4.3.

The performance of the model and discrete scheme is validated through numerical convergence order results in Chapter 5. For that purpose we introduced an academic scenario – the viscous cantilever, cf. Section 5.1. In Section 5.2 we obtained linear/super-linear convergence behavior in space, in time and combined for our jet model with the stabilized, index-reduced kinematics, independent of the used space-time domain (time-independent, time-dependent). Furthermore the convergence order reduction effect that was present in [5] could be overcome. In Section 5.3 we remark on possibly occurring singular Jacobian of the underlying linear solver. The use of r -refinement requires the jet’s general parameterization, which we investigated in Section 5.4. We compared the computational requirements

that are imposed by using said parameterization and the moving mesh strategies and found an increase of roughly factor three in computation time. Nevertheless, the moving mesh strategies could outperform the Lagrangian parameterization in terms of computation time and achieved error for the considered case of the viscous cantilever. We explored the usage of the r -refinement when large external forces are present and discovered that it does not perform like expected, the simulation breaks down even before its Lagrangian counterpart does due to oscillations. We assume that our premise to employ simple temporal and spatial flux approximations in the discrete scheme is hindering the successful application.

As examples for industrial applications we consider the rotational spinning process and the melt-blowing process in Chapter 6. Both processes are highly dynamic in the deformation of the jet which suggests the use of r -refinement, but unfortunately it fails like already mentioned in the previous paragraph. Even so, our discrete scheme greatly extends the applicable parameter range (physical parameters and discretization parameters) of the considered production processes with a Lagrangian parameterization. To include industrially relevant parameter ranges we altered the boundary conditions with the aim of easing the computational effort. In particular we include mass outflow at the tension-free jet end for the rotational spinning process and adjusted the extrusion speed and nozzle position for the melt-blowing process. Both alteration now enable the simulation of industrially relevant parameter ranges and therewith gives a valuable contribution to the existing simulation approaches for those processes.

To encourage a continued research we now talk about insights gained from this work. The moving mesh strategies do show great promise with the Burger's equation, and applications for jet model with moderate external forces, but fail altogether for large external forces. In order to overcome the oscillatory effects we propose the exploration of second order MUSCL-type, or third-/fifth-order WENO-type approximations for the artificial spatial fluxes (cf. Remark 5.5). Since the general parameterization of the jet model is computationally ponderous, we think the use of those high resolution schemes for all spatial fluxes in the Lagrangian parameterization is worth investigation as well. Additionally, the moving mesh strategies are possible to be used with the Lagrangian parameterization by employing a rezoning approach, i.e. the model is used in Lagrangian parameterization, but remeshing inbetween timesteps is allowed. For that purpose an appropriate interpolation is required that considers the physical meaning of the unknowns. With such an interpolation at hand, the classical h -refinement is easily incorporated as well, which would certainly be an enhancement to the existing discrete scheme as well. As for the industrial applications, the consideration of temperature-dependent viscoelastic material behavior is the next logical step. Exemplary academic viscoelastic simulations are presented in Appendix A.

Appendices

A Other material laws

In this work we focused on the viscous jet model. Nevertheless our modelling efforts, discrete scheme and its implementation aim at providing a framework that facilitates the exchange of the underlying geometric model and material law. For completeness we now state the non-dimensionalization of the elastic and viscoelastic material law and show exemplary simulations for the viscous and viscoelastic jet with temperature-dependent viscosity.

A.1 Elastic

Consider the elastic material law presented in Section 2.1.3. We introduce the typical Young's modulus $E_0 = E_\star$ [kg/(m s²)] for the nondimensionalization (also used for the shear modulus $G_0 = E_0$) implying the characteristic Mach number

$$\text{Ma} = v_\star \sqrt{\frac{\sigma_{M\star}}{\sigma_{V\star} E_\star}}.$$

The inextensible, unshearable elastic fiber model in the general parameterization is given by

$$\boldsymbol{\tau} = \tau^o \mathbf{e}_3, \tag{A.1a}$$

$$\epsilon^2(\boldsymbol{\kappa} - \boldsymbol{\kappa}^o) = \text{Ma}^2 \frac{\tau^o}{EA^2} \mathbf{M}_E^{-1} \cdot \mathbf{m}. \tag{A.1b}$$

The referential state $\tau^o, \boldsymbol{\kappa}^o$ is handled as type-1 fields

$$\partial_t \tau^o + \partial_s(u \tau^o) = 0, \quad \partial_t \boldsymbol{\kappa}^o + \partial_s(u \boldsymbol{\kappa}^o) = \mathbf{0}, \tag{A.1c}$$

whereas the material property E is handled as a type-0 field as well as $A = \sigma_V / \tau_3$ due to its definition. In case of a constant Young's modulus this yields

$$\partial_t E + u \partial_s E = 0, \quad \partial_t A + u \partial_s A = 0 \tag{A.1d}$$

which completes the material model. A full model for a elastic fiber is given by exchanging the material law of the viscous model (2.37). By furthermore neglecting the inertia term in the angular momentum balance, the classical Kirchhoff beam (also known as Kirchhoff-Love equations [3]) is contained in the elastic fiber model. It is given by the slenderness-low Mach number limit as $\varepsilon \rightarrow 0$, $\text{Ma} \rightarrow 0$ and $\varepsilon/\text{Ma} = \text{constant}$ (cf. [15]).

A.2 Viscoelastic

Our industrial examples both start with an initially incompressible viscous, liquid polymer that undergoes some degree of cooling and becomes inextensible elastic in the process. To approach a monolithic numerical simulation and design of such spinning processes from nozzle to deposition the obvious choice would be viscoelastic material behavior. Recent works of such models applied to Cosserat rod models are [107, 77, 10]. In Section 2.1.3 we follow [10] that use one that has the viscous and elastic behavior as asymptotic limit cases in its characteristic numbers. The relaxation time θ is handled as a type-0 field with its typical time $\theta_0 = \mu_0/E_0$ inducing the Deborah number De . It can be expressed as function of Re and Ma , i.e.

$$De = \frac{\mu_* v_*}{E_* r_*} = \frac{Ma^2}{Re}.$$

The dimensionless model is then formulated in general parameterization and given by

$$\begin{aligned} \partial_t \tau_3 + \partial_s(u\tau_3) &= \frac{Re}{3\mu} \frac{\tau_3^2}{\sigma_V} \mathbf{n}_3, & \tau_1 &= 0, & \tau_2 &= 0, \\ \varepsilon^2(\partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa})) &= \frac{Re}{3\mu} \frac{\tau_3^3}{\sigma_V^2} \mathbf{M}_\mu^{-1} \cdot (\mathbf{m} + De\theta(\partial_t \mathbf{m} + u\partial_s \mathbf{m})). \end{aligned}$$

The viscous limit is achieved by $De \rightarrow 0$, whereas the elastic one follows by $De \rightarrow \infty$, $Re \rightarrow 0$ such that $DeRe = Ma$. We have $3\mu/\theta = E$ due to the initial incompressibility assumption. For a more detailed discussion we refer to [10]. A full model for a viscoelastic jet is given by exchanging the material law of the viscous model (2.37).

A.3 Thermal simulations

The aim is a temperature-dependent simulation, thus we add the energy balance with convective air cooling (cf. Section 2.1.4) with a model for the heat transfer coefficient α_{air} that is based on the treatment of a cylindrical incident flow analogously to the air drag model in Section 6.2. We use a heuristic based model based on the Nusselt number Nu (ratio of convective to conductive heat transfer across the exchanging surface) that was proposed by Wegener and Arne [106]. It is the consecutive advancement to the one initially formulated in [97, 104], which was only valid for vertical incident flow and thus extended by [8] for arbitrary incident flow directions on the basis of experimental data. Wegener and Arne [106] added modifications to include meaningful values for limit cases (cf. [107]), avoid singularities and ensure continuous differentiability. The Nusselt number uses the diameter of the jet $D_{jet} = dd_0$ (cf. Section 6.2, (6.2)) as typical length and is dependent on specific dimensionless numbers, namely the tangential Reynolds, Reynolds and Prandtl number as well as the degree of turbulence which are given by

$$Re'_\tau = Re_* d \mathbf{e}_3 \cdot \mathbf{v}_{rel}, \quad Re' = Re_* d \|\mathbf{v}_{rel}\|, \quad Pr' = Pr = \frac{\rho_{air} \nu_{air} c_{p,air}}{\lambda_{air}}, \quad Tu' = Tu \frac{1}{\|\mathbf{v}_{rel}\|}.$$

Here, Pr is the Prandtl number of the airflow (ratio of viscosity to thermal diffusivity), the appearing physical constants are the thermal conductivity λ_{air} [W/(mK)= kg m/(s³ K)] and the specific heat capacity $c_{p,air}$ [J/(kg K)=m²/(s² K)] of the airflow. The Reynolds number Re_* was already introduced in Section 6.2.1 and the degree of turbulence Tu in Section 6.2.2. The suggested model for the Nusselt number is given by

$$Nu(Re'_\tau, Re', Pr', Tu') = \left(1 - 0.5h(Re'_\tau, Re')\right) Nu_{lam}(Re', Pr') \left(1 + Nu_{turb}(Re', Tu')\right)$$

with the help of

$$Nu_{lam}(Re', Pr') = \begin{cases} 0.462(Re'Pr')^{0.1} + f(Pr') \frac{(Re'Pr')^{7/10}}{1+2.79(Re'Pr')^{1/5}}, & Re'Pr' \geq 7.3 \cdot 10^{-5}, \\ a(Re'Pr')^3 + b(Re'Pr')^2 + c, & Re'Pr' < 7.3 \cdot 10^{-5} \end{cases}$$

$$Nu_{turb}(Re', Tu') = \frac{1.3 \cdot 10^{-2} Tu'^{1/2} Re'^{2Tu'^{1/10}}}{3.08 \cdot 10^3 + Tu'^{1/4} Re'^{2Tu'^{1/10}-0.5}}$$

and

$$f(Pr') = \frac{2.5}{\left(1 + \left(1.25Pr'^{1/6}\right)^{5/2}\right)^{2/5}},$$

$$a = -8.70481 \cdot 10^{11} + 5.14116 \cdot 10^{12}L - 3.128 \cdot 10^9 f(Pr'),$$

$$b = 9.69899 \cdot 10^7 - 5.62957 \cdot 10^8 L + 3.97675 \cdot 10^5 f(Pr'),$$

$$c = L.$$

The formula in Nu_{lam} for $Re'Pr' \geq 7.3 \cdot 10^{-5}$ has the limit $L = 0.1$ for $Re'Pr' \rightarrow 0$. The coefficients a, b, c are chosen to ensure continuous differentiability of Nu_{lam} . The function

$$h(Re'_\tau, Re') = \begin{cases} \left(\frac{Re'_\tau}{Re'}\right)^2, & Re' \geq \delta, \\ \left(1 - \left(\frac{Re'}{\delta}\right)^2\right)^2 + \left(3 - 2\left(\frac{Re'}{\delta}\right)^2\right) \left(\frac{Re'}{\delta}\right)^2 \left(\frac{Re'_\tau}{\delta}\right)^2, & Re' < \delta \end{cases}$$

extends the applicability of the model to arbitrary incident flow directions (for vertical flow we have $h(Re'_\tau, Re') = 0$). The appropriate regularization with the associated parameter $\delta \ll 1$ (here, $\delta = 10^{-7}$) guarantees a smooth transition for vanishing relative velocity to the case of parallel incident flow (namely $\lim_{Re' \rightarrow 0} h(Re'_\tau, Re') = 1$). This was also done in [107], but the above choice is additionally continuous differentiable. Note that

$$Re'_\tau/Re' = \mathbf{e}_3 \cdot \frac{\mathbf{v}_{rel}}{\|\mathbf{v}_{rel}\|}$$

is the cosine of the incident flow angle. The construction with Re'_τ and Re avoids singularities for vanishing relative velocity. The term Nu_{turb} can be neglected in stationary flows.

The dimensionless energy balance solely considering convective air cooling of the jet (cf. Section 2.1.4) is now given by

$$A(\partial_t(\sigma_M T) + \partial_s(u\sigma_M T)) = -2\|\boldsymbol{\tau}\|(T - T_{air})Nu \left(Re_* d\mathbf{e}_3 \cdot \mathbf{v}_{rel}, Re_* d\|\mathbf{v}_{rel}\|, Pr, Tu \frac{1}{\|\mathbf{v}_{rel}\|} \right)$$

using the dimensionless quantity

$$A = \frac{\sigma_{M\star} v_\star}{r_\star} \frac{c_p}{\lambda_{air}} = \varepsilon^2 \text{RePr}_\star, \quad \text{Pr}_\star = \frac{c_p \mu_\star}{\lambda_{air}}$$

with a mixed air-jet Prandtl number Pr_\star (ratio of viscosity to thermal diffusivity from air to jet) that contains the constant specific heat capacity c_p [J/(kg K)=m²/(s² K)] of the jet. Note that Pr and Tu are functions dependent on \mathcal{Q} , whereas A and Re_\star are constant.

The material that we are considering is a polypropylene that is typical for melt-blowing processes. The parameters necessary for the Arrhenius law of the temperature-dependent viscosity are provided by the Fraunhofer ITWM and were calibrated for temperatures given in Celsius. Its dimensionless form is obtained by introducing the typical temperature $T_0 = T_\star$ [K] (chosen to be the nozzle temperature) and given by

$$\mu(T) = c_1 \exp(c_2/(T - c_3)) \quad (\text{A.2})$$

with the physical constants $c_1 = 0.1352/\mu_0$ and $c_2 = 852.323/T_0$ and $c_3 = 273.15/T_0$. Due to the exponential dependence on the temperature, the viscosity will vary several orders of magnitude, more precisely $\mathcal{O}(10^{-1})$ up to $\mathcal{O}(10^{15})$ for 300 up to 1500 K. Our formulation of the material law (μ only appears in the denominator) is recommended.

To illustrate different material behavior we will now present simulations of the cantilever example in Section 5.1, we use the jet model with original kinematics. As the first example we consider the fixed and growing cantilever with viscous and viscoelastic material, both are equipped with the temperature-dependent viscosity model (A.2). A temperature profile is set that is homogeneous in \bar{r}_1 and \bar{r}_3 direction. It is cold at the nozzle and then gradually gets warmer in negative \bar{r}_2 direction. The parameters are $(\text{Re}, \text{Fr}, \text{De}, \epsilon, T) = (1, 1, 0.1, (\sqrt{\pi}/2)0.1)$ and $\Delta s = \Delta t = 10^{-2}$. Results for the viscoelastic jet are depicted in Figure A.1, the temperature profile of the domain is indicated in the

Parameters

| Description | Symbol | Value | Unit |
|----------------------------|-----------------|----------------------|---|
| Air specific heat capacity | $c_{p,air}$ | 1006.43 | J/(kg K)=m ² /(s ² K) |
| Air thermal conductivity | λ_{air} | $2.42 \cdot 10^{-2}$ | W/(m K)=kg m/(K s ³) |
| Jet specific heat capacity | c_p | 1700 | J/(kg K)=m ² /(s ² K) |
| Jet Young's modulus | E | 10^9 | kg/(m s ²) |

All remaining parameters are taken from Table 6.2

Dimensionless quantities

| Description | Formula | Value |
|---------------|---|----------------------|
| Deborah | $\text{De} = \mu U / EH$ | $8.50 \cdot 10^{-8}$ |
| mixed Prandlt | $\text{Pr}_\star = \mu c_p / \lambda_{air}$ | $7.02 \cdot 10^4$ |

Table A.1: Overview of the physical parameters of an industrial example for viscoelastic simulations. The typical values are plugged in and the dimensionless quantities computed.

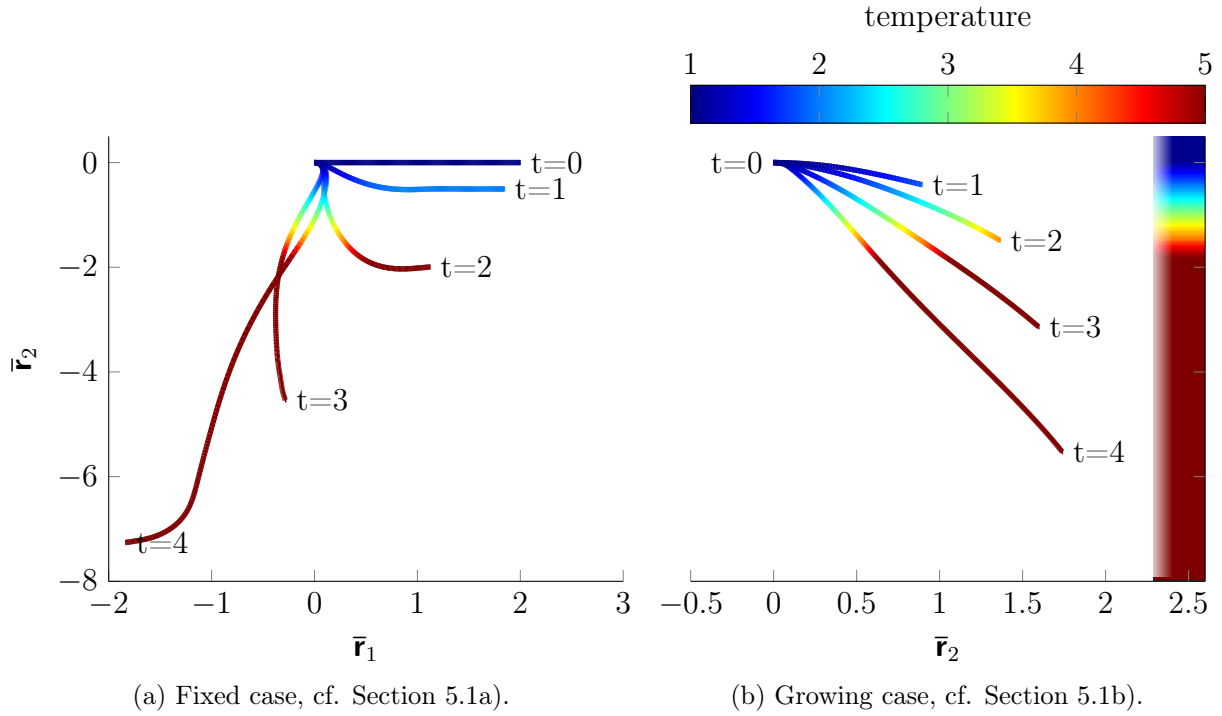


Figure A.1: Simulation of a initially cold viscoelastic Cantilever that is heated up. The bar on the right side indicates the applied temperature profile that ranges from 300 K at the nozzle and 1500 K at the bottom, it is depicted dimensionless.

right plot. As the jet is heating up it becomes more viscous and stretches like expected. The analogous viscous simulation simply extrudes in \bar{r}_1 , since it is a rigid body in that temperature range.

The second example is now with an added airflow that has a reverse temperature profile and has a velocity that is homogeneous in \bar{r}_1 and \bar{r}_3 direction. The airflow starts initially at the nozzle as standing air until $\bar{r}_2 \leq -1.6$, then its component in negative \bar{r}_1 direction increases. To account for the convective thermal exchange between airflow and jet the energy balance is added. A viscous and viscoelastic simulation are depicted in Figure A.2. The difference in the behavior for cold temperature is clearly visible. Whereas the viscous jet keeps its curvature when cooled down, the viscoelastic one tends to straighten and follow the airflow.

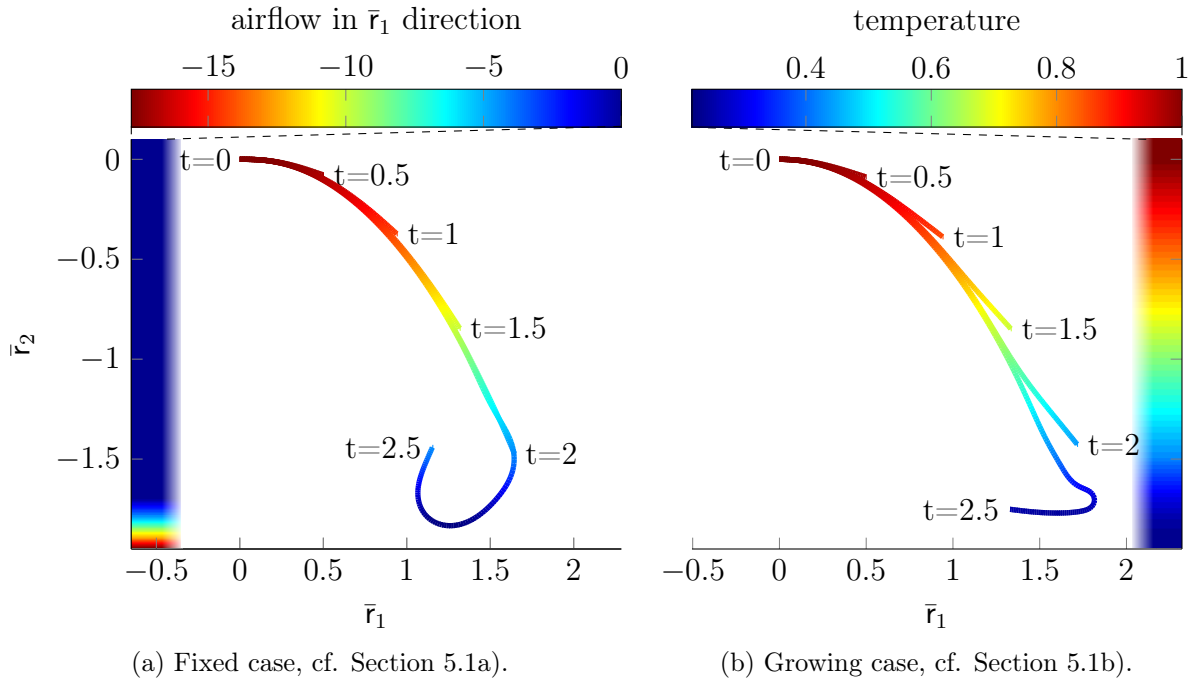


Figure A.2: Simulation of a cantilever that is extruded with high temperature and then cooled by the surrounding air. At the bottom is an airflow layer with high velocity normal to the gravitational force. The bar on the left side indicates the dimensionless velocity profile and the one on the right side the dimensionless temperature of the airflow for both simulations.

B Modeling and reduction of dimensions

B.1 GGL projection correction for the kinematics

In Section 2.2.4 we introduced the stabilized, index-reduced formulation of the kinematics of the jet model. Throughout the work we have used our SAMW-correction that was inspired by the Gear-Gupta-Leimkuhler (GGL) correction [43]. In this appendix we want to present said approach for completeness.

Correction B.1 (GGL correction). Assume Formulation 2.22 for the kinematics. In the beginning of Section 2.2.4 we described our interpretation of the Gear-Gupta-Leimkuhler approach for PDAEs. Following that outline there we obtain

$$\begin{aligned}\bar{\Lambda}_{r,1}^{\text{GGL}} &:= \partial_s \bar{\lambda}_r, & \Lambda_{q,1}^{\text{GGL}} &:= \partial_s \lambda_q - \mathcal{A}(\kappa) \cdot \lambda_q, \\ \Lambda_{r,2}^{\text{GGL}} &:= \mathbf{0}, & \Lambda_{q,2}^{\text{GGL}} &:= \mathbf{0}\end{aligned}$$

with $\bar{\lambda}_r := \bar{\lambda}_1$ and $\lambda_q := \lambda_2$ as the multipliers.

The equivalence of the GGL correction to the original kinematics can be shown analogously to Section 2.2.4. The inherent equations (2.33) enforced by the kinematics become

$$\begin{aligned}\partial_{ss} \bar{\lambda}_r - \mathbf{R}(\mathbf{q})^T \cdot \partial_s \bar{\lambda}_r &= \mathbf{0}, \\ \partial_{ss} \lambda_q - 2\mathbf{A}(\kappa) \partial_s \lambda_q - \mathbf{A}(\partial_s \kappa) \cdot \lambda_q &= \mathbf{0},\end{aligned}$$

which is a system of linear, homogeneous, differential equations of second order with variable coefficients analogous to the SAMW-correction in Lemma 2.29. One could also think of a proof similar to the one in [43], that was done for DAEs. We do not want to reiterate it here, but the key idea to transfer it to PDAEs is to apply a semi-discretization in space – which makes it a DAE – and to consider all discrete equations including boundary conditions in the proof.

B.2 Two-dimensional jet model

The rotational spinning process from Section 6.1 (and of course also the jet model generally) can be simplified to a two-dimensional model by neglecting gravity ($\text{Fr} \rightarrow \infty$). The jet solely undergoes the rotational loads and moves in the exit plane perpendicular to the rotation axis of the drum if the initial and boundary conditions also abide by said plane. We set $\vec{\Omega} \parallel \vec{a}_1$, thus the unknowns $\vec{r}, \vec{v}, \vec{n}$ and $\vec{\tau}$ take shape in the the $\vec{a}_2 - \vec{a}_3$ plane,

whereas $\vec{\kappa}$, $\vec{\omega}$ and \vec{m} are parallel to \vec{a}_1 as a consequence of the kinematic equations and the material law. The direction of the jet can be prescribed by a single angle $\alpha \in [0, 2\pi]$ with

$$\mathbf{R}(\alpha) = \begin{pmatrix} \sin(\alpha) & -\cos(\alpha) \\ \cos(\alpha) & \sin(\alpha) \end{pmatrix}.$$

The three-dimensional unknowns are reduced to

$$\begin{aligned} \bar{\mathbf{r}} &= (\bar{r}_2, \bar{r}_3), & \mathbf{v} &= (v_2, v_3), & \mathbf{n} &= (n_2, n_3) \\ \boldsymbol{\tau} &= (\tau_2, \tau_3), & \omega &= \omega_1, & \kappa &= \kappa_1, & \mathbf{m} &= m_1, & \Omega &= \bar{\Omega}_1 \end{aligned}$$

and the two-dimensional jet model in Lagrangian parameterization with original kinematics is given by

$$\begin{aligned} \partial_t \bar{\mathbf{r}} &= \mathbf{R}(\alpha)^T \cdot \mathbf{v}, \\ \partial_s \bar{\mathbf{r}} &= \mathbf{R}(\alpha)^T \cdot \boldsymbol{\tau}, \\ \partial_t \alpha &= \omega, \\ \partial_s \alpha &= \kappa, \\ \partial_t \mathbf{p} &= -\omega \mathbf{p}^\perp + \partial_s \mathbf{n} + \kappa \mathbf{n}^\perp + \mathbf{f}, & \mathbf{p} &= \sigma_M \mathbf{v}, \\ \partial_t h &= \frac{1}{\varepsilon^2} \partial_s m + \boldsymbol{\tau} \cdot \mathbf{n}^\perp + l, & h &= \mathbf{J}_M \omega, \\ \partial_t \sigma_M &= 0, & \partial_t \sigma_V &= 0, \\ \partial_t \tau_3 &= \frac{\text{Re } \tau_3^2}{3\mu \sigma_V} n_3, & \tau_2 &= 0, \\ \varepsilon^2 \partial_t \kappa &= \frac{\text{Re } \tau_3^3}{3\mu \sigma_V^2} M_\mu^{-1} m \end{aligned}$$

with the artificial outer loads resulting from the rotating outer basis

$$\begin{aligned} \mathbf{f} &= \mathbf{f}_\Omega = -\frac{2}{\text{Rb}} \sigma_M \Omega \mathbf{v}^\perp + \frac{1}{\text{Rb}^2} \sigma_M \Omega^2 \mathbf{R}(\alpha) \cdot \bar{\mathbf{r}}, \\ l &= l_\Omega = -\frac{1}{\text{Rb}} \Omega \partial_t \mathbf{J}_M \end{aligned}$$

and

$$\mathbf{J}_M = \frac{\sigma_M \sigma_V}{\tau_3} M_{in}, \quad M_{in} = M_\mu = \frac{1}{4\pi}.$$

Furthermore $\mathbf{z}^\perp = (-z_2, z_1)^T$ was introduced for any tuple $\mathbf{z} \in \mathbb{R}^2$. We published simulations for the two-dimensional rotational spinning process in [87].

B.3 Uniaxial jet model

The jet model can be reduced to an uniaxial case with restrictions on the initial- and boundary conditions and external loads. Assume that the initial- and boundary conditions

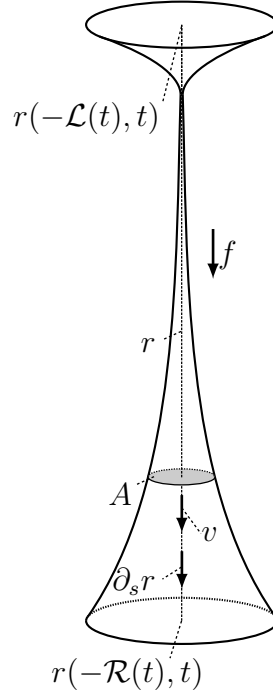


Figure B.1: Illustration of an extruded, uniaxial jet with arbitrary outer force with no outflow, cf. Example 2.38.

are given such that the fiber is a straight line. Assume furthermore that the external forces are of the form $\vec{f} = f\vec{d}_3$ and $\vec{l} = l\vec{d}_3$. Then the fiber always remains a straight line, an illustration is shown in Figure B.1. Respective simulations are published in [88, 90].

Without loss of generality we set $\vec{d}_3 = \vec{a}_1$ and use $r = \bar{r}_1$, $v = v_3$, $\omega = \omega_3$, $\tau = \tau_3$, $\kappa = \kappa_3$, $n = n_3$ and $m = m_3$ as well as $\lambda_\tau = \bar{\lambda}_{\tau,3}$ and $\lambda_\kappa = \lambda_{\kappa,3}$. The viscous jet model (2.37) reduces to

$$\partial_t r = v - ue + \frac{1}{k} \partial_s \lambda_\tau, \quad (\text{B.1a})$$

$$\partial_s r = e, \quad (\text{B.1b})$$

$$\partial_t \tau + \partial_s (u\tau) = \partial_s v + Ck\lambda_\tau, \quad (\text{B.1c})$$

$$\partial_t \sigma_M + \partial_s (u\sigma_M) = 0, \quad \partial_t \sigma_V + \partial_s (u\sigma_V) = 0, \quad (\text{B.1d})$$

$$\partial_t p + \partial_s (up) = \partial_s n + f, \quad p = \sigma_M v \quad (\text{B.1e})$$

$$\partial_t \tau + \partial_s (u\tau) = \frac{\text{Re}}{3\mu} \frac{\tau^2}{\sigma_V} n, \quad (\text{B.1f})$$

$$\partial_t \mu = 0. \quad (\text{B.1g})$$

Bending will not appear and the torsion of the jet can be depicted by a single angle α and the rotation matrix

$$\mathbf{R}(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sin(\alpha) & -\cos(\alpha) \\ 0 & \cos(\alpha) & \sin(\alpha) \end{pmatrix}.$$

Its dynamics are decoupled from Equation (B.1) and given by

$$\partial_t \alpha = \omega - u\kappa + \frac{1}{k}\lambda_\kappa, \quad (\text{B.2a})$$

$$\partial_s \alpha = \kappa, \quad (\text{B.2b})$$

$$\partial_t \kappa + \partial_s(u\kappa) = \partial_s \omega + Ck\lambda_\kappa, \quad (\text{B.2c})$$

$$\varepsilon^2 \partial_t h + \partial_s(uh) = \partial_s m + \varepsilon^2 l, \quad h = (\sigma_M \sigma_V / \tau_3) M_{in}, \quad (\text{B.2d})$$

$$\varepsilon^2 (\partial_t \kappa + \partial_s(u\kappa)) = \frac{\text{Re } \tau^3}{3\mu \sigma_V^2} M_\mu^{-1} m \quad (\text{B.2e})$$

with $M_{in} = 1/(2\pi)$ and $M_\mu = 1/(6\pi)$.

C Adaptivity

We explore the performance of the moving mesh strategies in Chapter 3 with an analytical function to explore how the strategies and mesh control functions influences the mesh and subsequently do a benchmark with the Burger's equation at hand. Before we get to that we present the considered mesh control functions and the spatial smoothing technique.

To illustrate the moving mesh strategies we use mesh control functions according to choices in literature that have proven to work well for the Burger's equation (cf. [59, Section 1.2.2, Section 2.4.4]). The considered possibilities for the monitor function are the arc-length one (3.25), the optimal curvature one (5.1), and the the optimal slope monitor function given by

$$M^{slo}(s, t) = \left(1 + \frac{1}{\alpha(t)} |\partial_\sigma \hat{y}(\sigma, t)|_{\sigma=\Psi(s, t)}|^2 \right)^{1/3}, \quad (\text{C.1a})$$

$$\alpha^{slo}(t) = \left(\frac{1}{|\Omega|} \int_{\Omega} |\partial_\sigma \hat{y}(\sigma, t)|_{\sigma=\Psi(s, t)}|^{2/3} ds \right)^3, \quad (\text{C.1b})$$

with their corresponding implicated and transferred parameter density (cf. Strategy 3.19 and 3.20). The mesh control functions are discretized using central finite differences within the domain and one-sided ones at the boundary, appearing integrals are approximated with a trapezoidal rule.

Remark C.1. *We want to mention another option for the monitor function – yet not used in this work – namely the component-averaged monitor that is developed in [94] for hyperbolic systems. The idea is to tailor a monitor function for some or all unknowns of the system and to combine them in an appropriate fashion such that physical features of the solution are properly resolved. The example in [94] is tailored to a model problem for the one-dimensional Euler equation for an inviscid, compressible and polytropic gas that experiences shocks and contact discontinuity. They use a shock monitor and contact monitor controlling the first spatial derivative of velocity and entropy of the solution. Both are then scaled and combined.*

The spatial smoothing operator G^{-1} from Lemma 3.22 is replaced by the approximation $\mathbb{1} + \lambda^{-2}\Delta + (\lambda^{-2}\Delta)^2 + \dots + (\lambda^{-2}\Delta)^p$ with a properly chosen λ , $p > 0$ and a central finite difference for Δ , i.e. $\Delta M_i = (M_{i+1} - 2M_i + M_{i-1})/\Delta s^2$. This approximation uses $2p + 1$ adjacent cells for the averaging of the monitor function. Huang et al. use two discrete approximations. The first one, that was originally based on [38], is given by

$$M_i^G = \left(\sum_{k=i-p}^{i+p} (M_k)^q \left(\frac{\gamma}{1+\gamma} \right)^{|k-i|} / \sum_{k=i-p}^{i+p} \left(\frac{\gamma}{1+\gamma} \right)^{|k-i|} \right)^{1/q}. \quad (\text{C.2})$$

with $p = 3$, $\gamma = 2$ and $q = 2$. The second one is a simple but effective smoothing scheme utilizing the above formula. By using $p = 1$, $\gamma = 1$ and $q = 1$, (C.2) reduces to the simple averaging of direct neighbors, i.e. $M_i^G = \frac{1}{4}M_{i-1} + \frac{1}{2}M_i + \frac{1}{4}M_{i+1}$. They apply several sweeps of the scheme within one integration step, in particular they redo it four times and in each sweep use the result from the previous sweep as input. We follow the second approach. We use the notation from Section 5.4.2 to address the different moving mesh strategies.

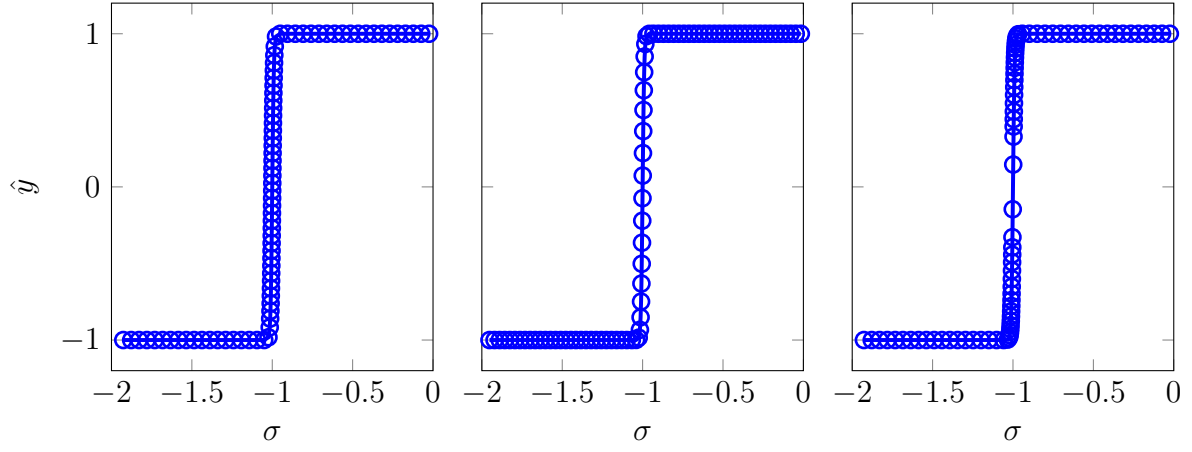
C.1 Analytical example

Analogous to [59, Section 2.5.4] we present an illustrative example here for which the adaptive meshes are generated from a known function to gain insight of the performance of our discrete approach and the suggested moving mesh strategies. Consider the function

$$\hat{y}(\sigma, t) = \tanh(R(\sigma + 1)) \quad (\text{C.3})$$

on $\hat{\mathcal{Q}} = [-2, 0] \times [0, 1]$ with $R = 100$ as a parameter. The general space-time domain is chosen to coincide, i.e. $\hat{\mathcal{Q}} = \mathcal{Q}$. We use the moving mesh strategies from Chapter 3 but do not apply spatial smoothing to the mesh control function. The results for different mesh control functions are depicted in Figures C.2, C.1, C.3 and C.4. In those simulations the temporal relaxation parameter τ was adjusted such that the different strategies behave comparably (if possible). In Figure C.5 we kept τ constant to illustrate the difference in the strategies. All strategies enforce that the mesh movement is following the mesh control function, with exception of $(D2)^t$ (DELAX2 (Strategy 3.13) with the transferred parameter density). Since the transferred parameter density is defined solely on the general parameterization. It seems that the role of the mesh adaption factor is diminished and the spatial balance is lost (observe that e.g. (M2) (MMPDE2 (Strategy 3.10)) reached a converged state of the mesh at the end time). This can lead to a spatially skew mesh and areas with moderate values of the mesh control function are resolved insufficiently, cf. Figure C.5. The lack of the spatial smoothing procedure is clearly visible for the optimal curvature mesh control function. The unbalanced resolution in the steep front is avoided when spatial smoothing is employed, cf. Figure C.2.

The simulations suggest that the performance of (M1), (M2) combined with our discrete scheme agrees with [59]. (D2) performs comparably with the implicated parameter density, but has the disadvantage of possible mesh skewness with the transferred parameter density.



(a) Solution at the end time.

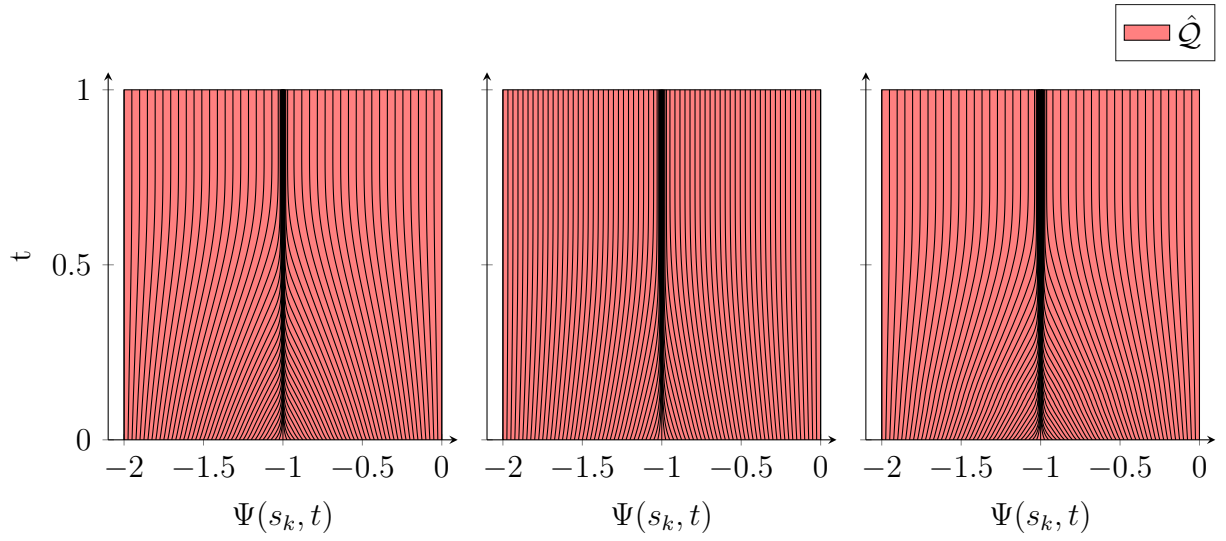
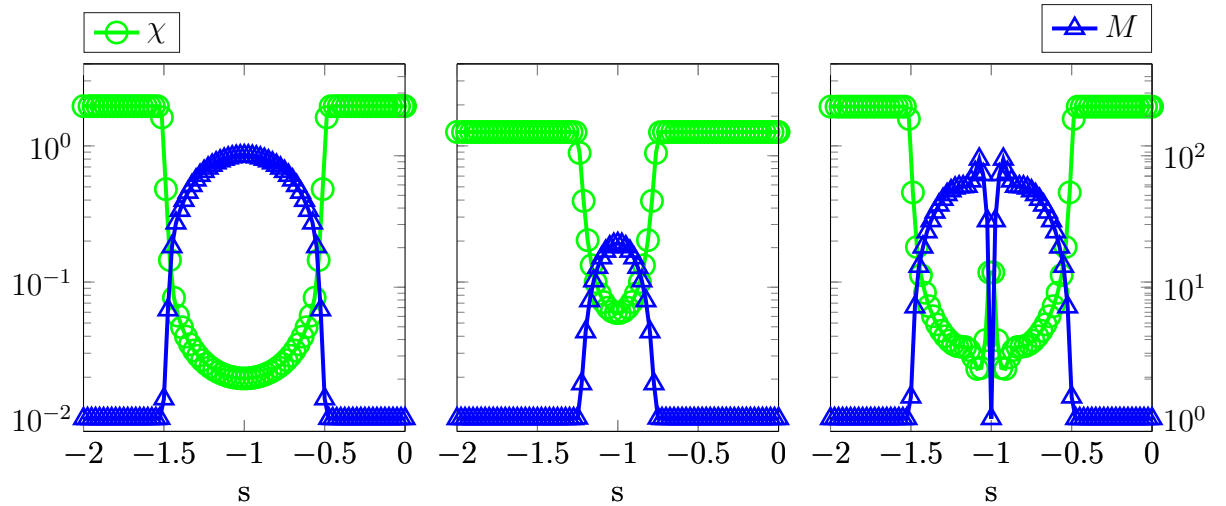
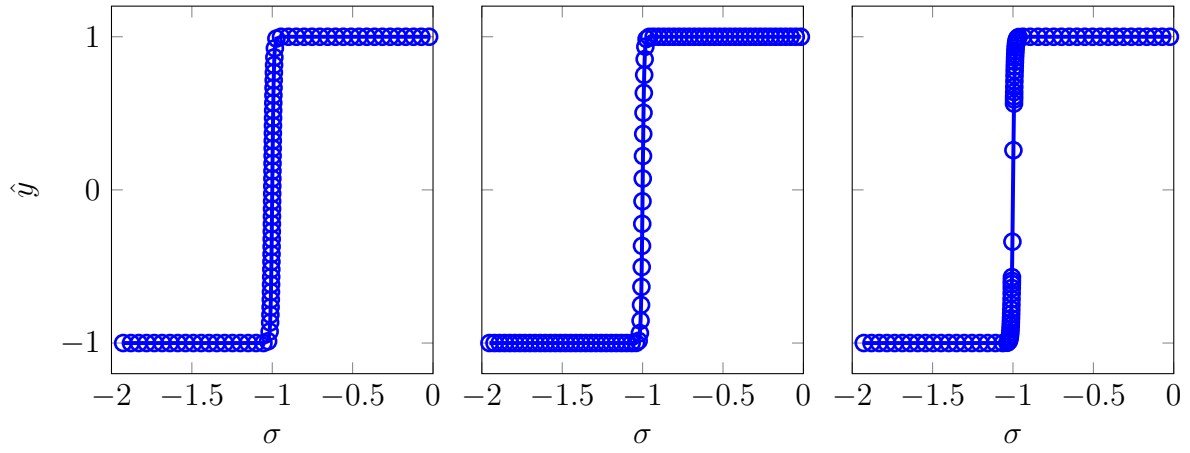
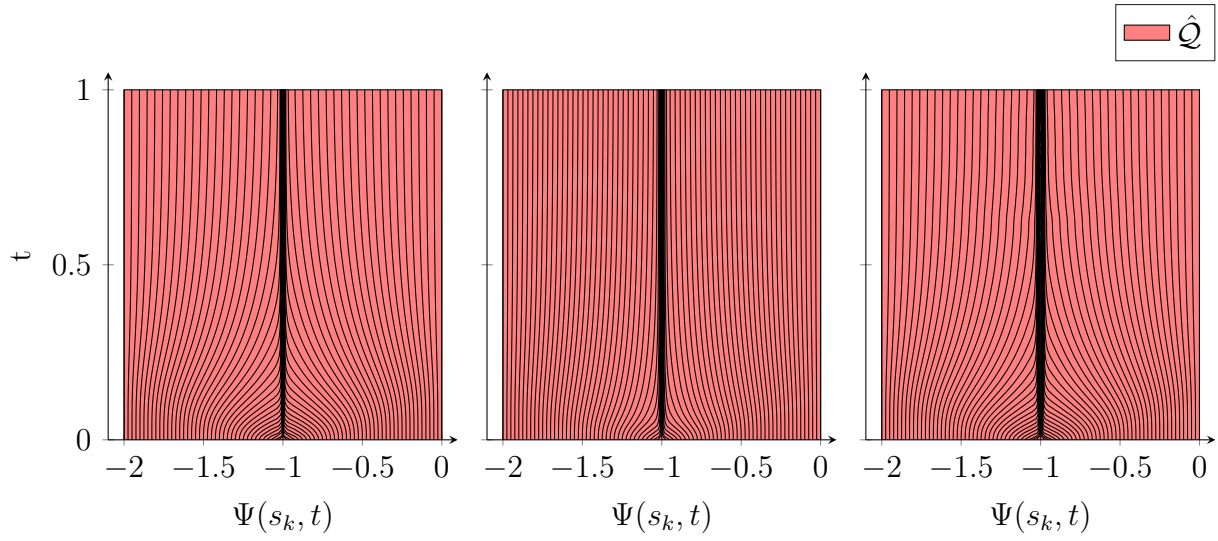
(b) Trajectories of material points identified with s_k over time.(c) Jacobian χ and monitor function M at the end time $t = 1$.

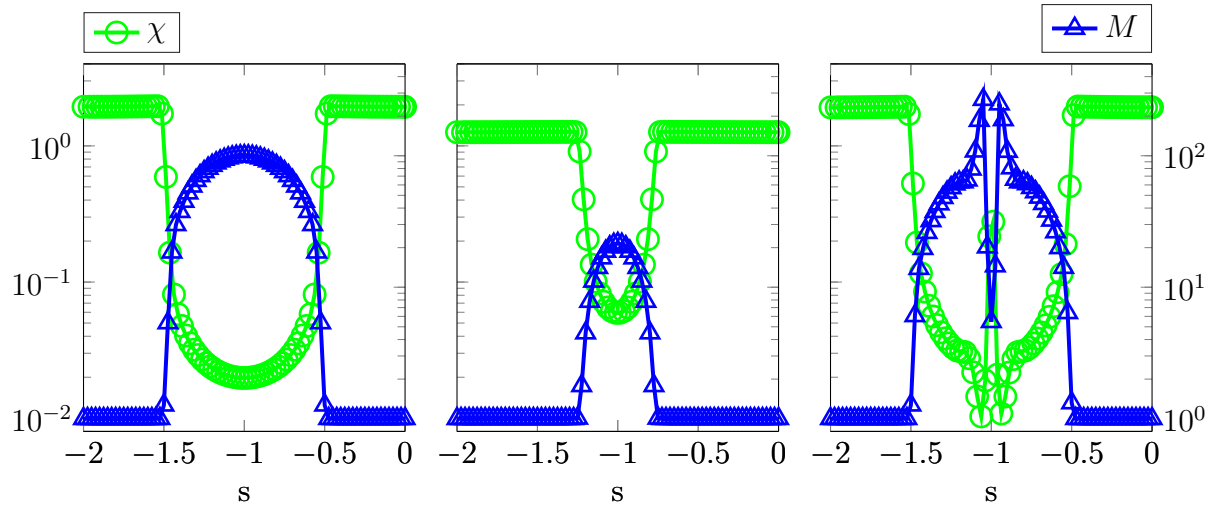
Figure C.1: Details of the analytical example (C.3) together with (M1), $\tau = 2 \cdot 10^{-2}$ and different monitor functions. From left to right: Arc-length, optimal slope and optimal curvature monitor function.



(a) Solution at the end time.

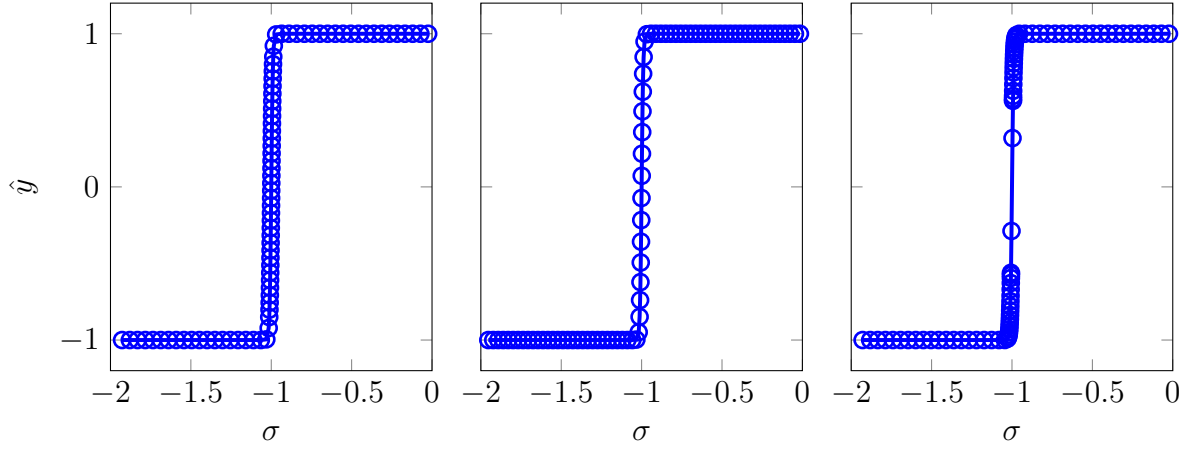


(b) Trajectories of material points identified with s_k over time.



(c) Jacobian χ and monitor function M at the end time $t = 1$.

Figure C.2: Details of the analytical example (C.3) together with (M2), $\tau = 2$ and different monitor functions. From left to right: Arc-length, optimal slope and optimal curvature monitor function.



(a) Solution at the end time.

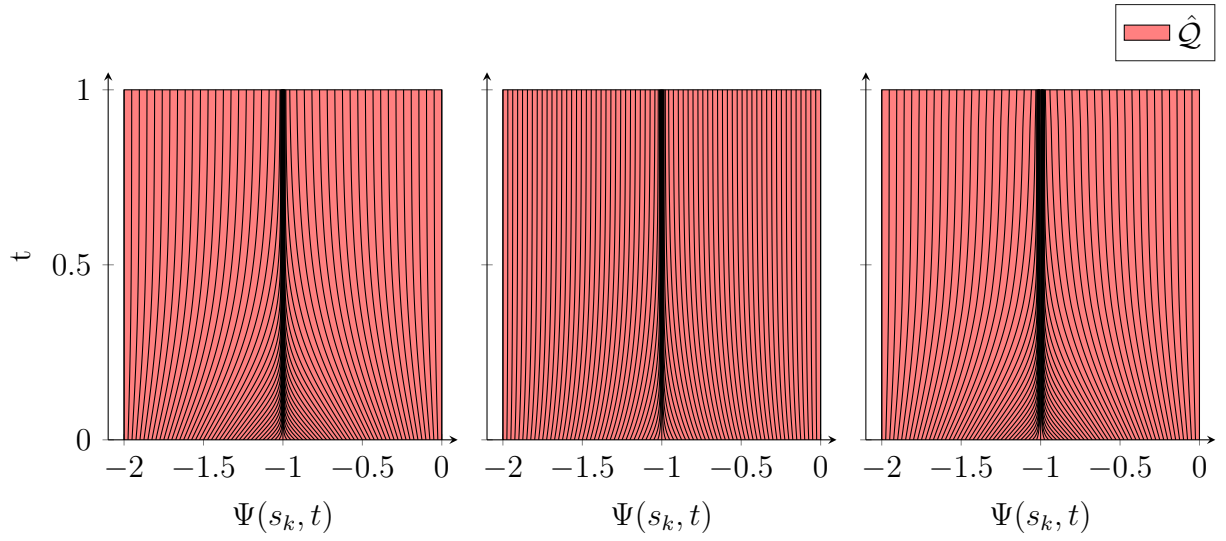
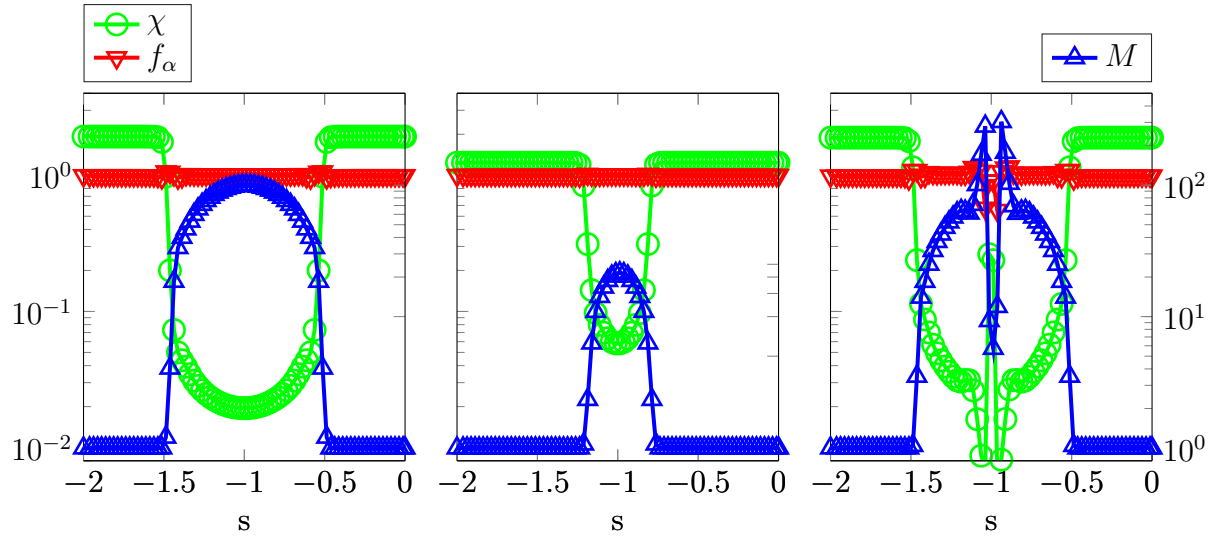
(b) Trajectories of material points identified with s_k over time.(c) Jacobian χ and parameter density f_α , as well as the underlying monitor function z at the end time $t = 1$.

Figure C.3: Details of the analytical example (C.3) together with $(D2)^i$, $\tau = 2 \cdot 10^{-1}$ and different implicated parameter densities. From left to right: Arc-length, optimal slope and optimal curvature parameter densities.

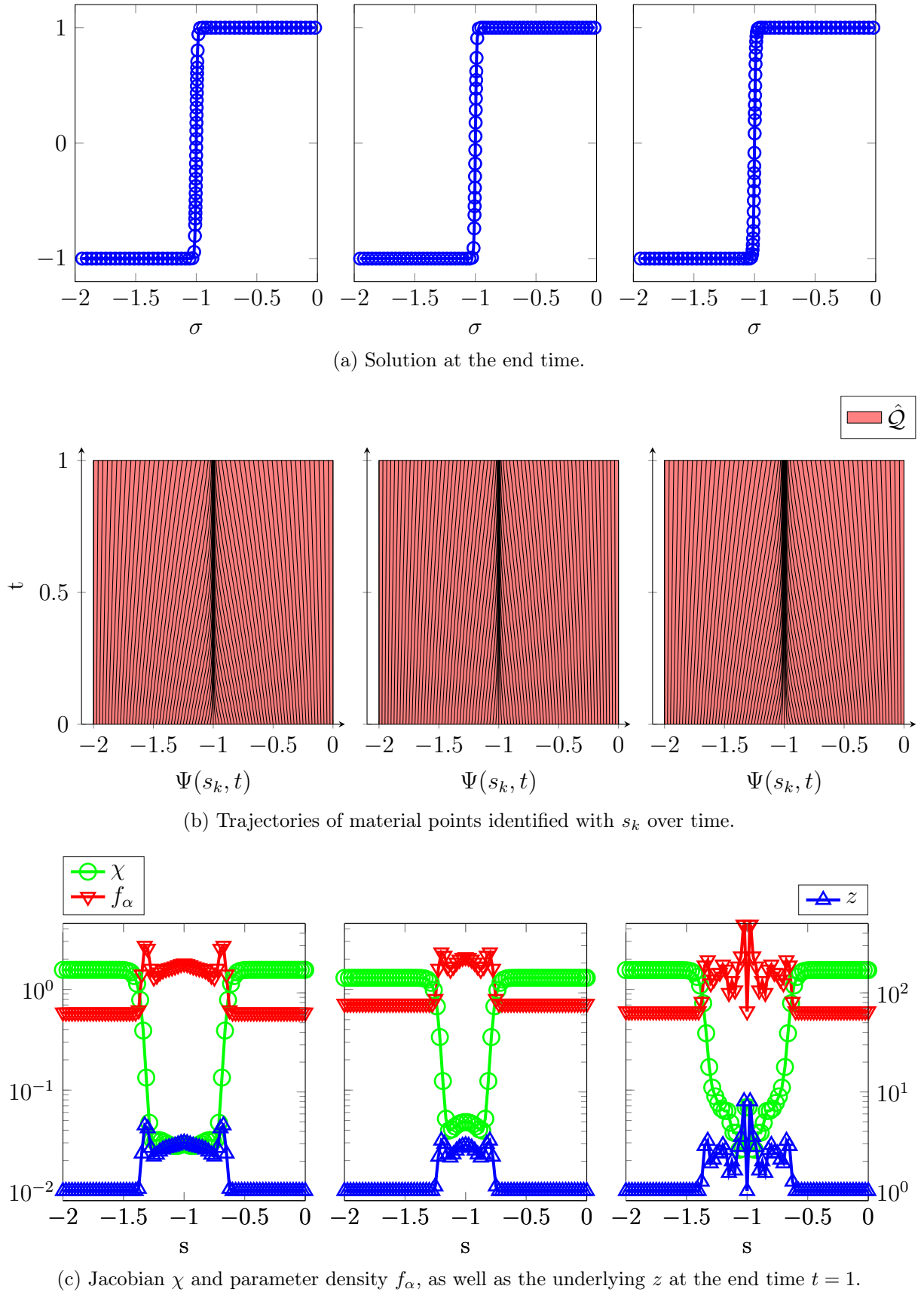


Figure C.4: Details of the analytical example (C.3) together with $(D2)^t$, $\tau = 1$ and different transferred parameter densities. From left to right: Arc-length, optimal slope and optimal curvature parameter densities.

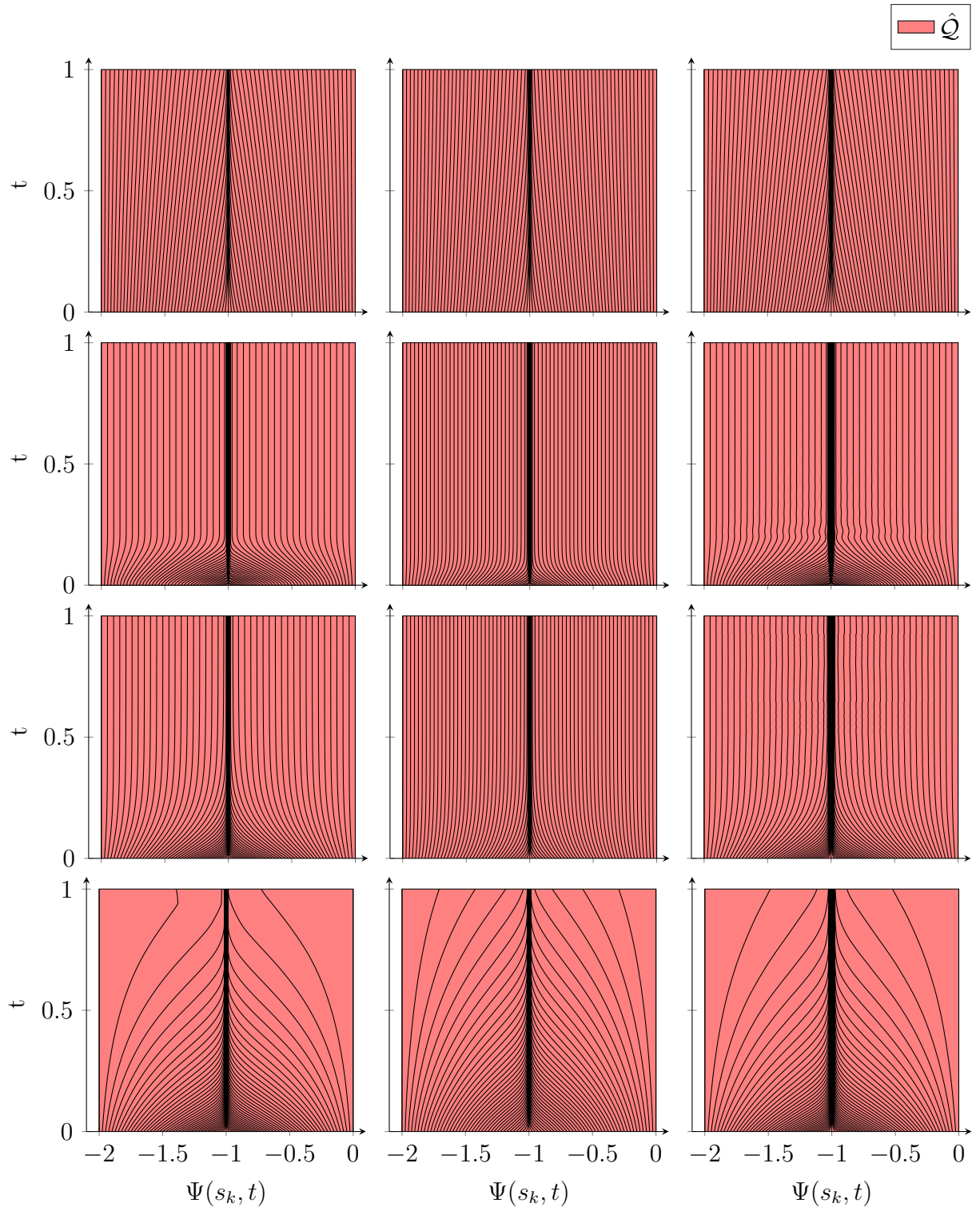


Figure C.5: Mesh trajectories of the analytical example (C.3) with $\tau = 10^{-1}$. From top to bottom: (M1), (M2), $(D2)^i$, $(D2)^t$. From left to right: Arc-length, optimal slope and optimal curvature mesh control function.

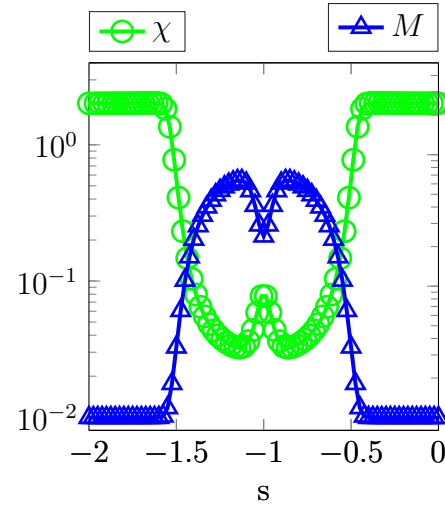
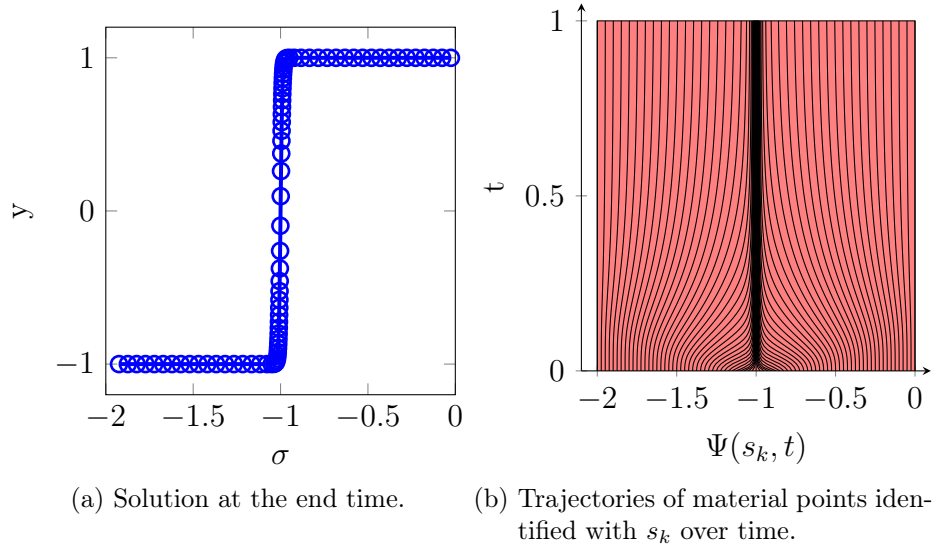


Figure C.6: Details of the analytical example (C.3) together with (M2) and the smoothed optimal curvature monitor functions.

C.2 Burger's equation

Benchmarks are now carried out with the DELAX and MMPDE strategies. The model problem for this investigation is a initial-boundary value problem consisting of the viscous Burger's equation (broadly used in literature, e.g., in [21, 61, 58, 96, 98, 59, 110, 78]) on a fixed domain $\hat{\Omega} = [0, 1]$ in the referential (Lagrangian) parameterization

$$\partial_t \hat{y} + \partial_\sigma (\hat{y}^2/2) = \mu \partial_{\sigma\sigma} \hat{y}, \quad (\text{C.4a})$$

with $\hat{y}: [0, 1] \times [0, T] \rightarrow \mathbb{R}$ and a viscosity parameter $0 \leq \mu \ll 1$ (not to be mistaken with the viscosity of the jet model) and appropriate initial and boundary conditions given by

$$\hat{y}(-1, t) = \hat{y}(0, t) = 0, \quad \hat{y}(\sigma, 0) = \sin(2\pi\sigma) + \frac{1}{2} \sin(\pi\sigma). \quad (\text{C.4b})$$

Starting out with a smooth initial profile a steep front develops whose inclination depends on ε . The front propagates towards the right side while being dampened by the homogeneous Dirichlet boundary condition at $s = 0$. The steep front has to be properly resolved, which is the difficulty of a numerical solution. The Burger's equation is now transformed into the computational (general) parameterization, whereas the general domain is assumed to coincide with the Lagrangian one if not otherwise mentioned. Treating \hat{y} as a type-0 field ($y(s, t) = \hat{y}(\Psi(s, t), t)$) we obtain by using the conservation for χ

$$\partial_t(\chi y) + \partial_s(u\chi y) + \partial_s(y^2/2) = \mu \partial_s \left(\frac{1}{\chi} \partial_s y \right). \quad (\text{C.5})$$

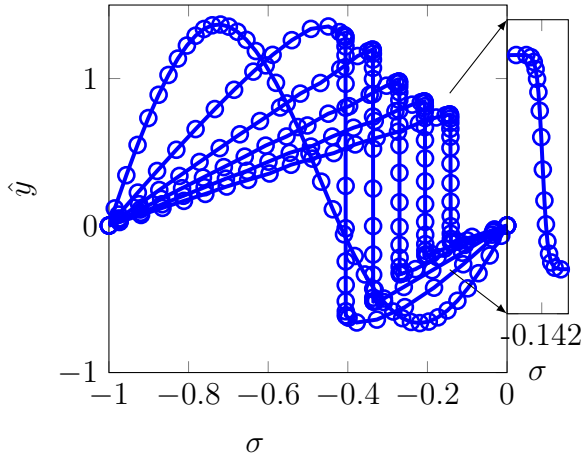
The Burger's equation is discretized using the discrete scheme presented in Chapter 4. To obtain a first-order system the substitution unknown $y^s = \partial_s y$ is used. The derivative at the boundaries is approximated with a one-sided, first-order finite difference. The set of unknowns is y , y^s , χ and u for the Burger's equation as a first-order system and one of the moving mesh strategies. Analogously to the jet model we choose y , y^s to be node type and y^s , u to be edge type.

Remark C.2. *Huang and Russell [59] use a finite difference approximation in space of the Burger's equation, i.e. they use pointwise approximations $y_i = y(s_i, t)$, $\Psi_i = \Psi(s_i, t)$ and $u_i = u(s_i, t)$ ($i = 0, \dots, N$) and obtain the semi-discrete Burger's equation*

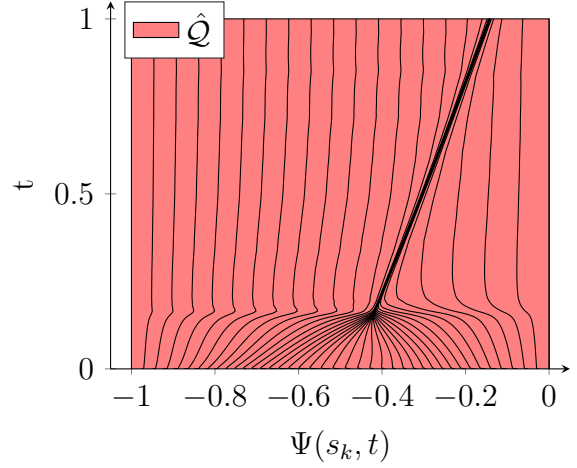
$$\partial_t y_i - u_i \frac{y_{i+1} - y_{i-1}}{2\Delta s} + \frac{(y_{i+1}^2 - y_{i-1}^2)/2}{\Psi_{i+1} - \Psi_{i-1}} = \frac{2\mu}{\Psi_{i+1} - \Psi_{i-1}} \left(\frac{y_{i+1} - y_i}{\Psi_{i+1} - \Psi_i} - \frac{y_i - y_{i-1}}{\Psi_i - \Psi_{i-1}} \right)$$

for $i = 1, \dots, N-1$ and $\Delta s = 1/N$ and given initial conditions for $t = 0$ and boundary conditions y_0 , y_N and Ψ_0 , Ψ_N . The integration in time that they use is fully implicit with MatLab's `ode15i` solver. It employs an automatic time step control [91] whereas our discrete scheme uses a constant time step. Our discrete scheme is prepared for an automatic time stepping, but we focus explicitly on constant time steps in this work.

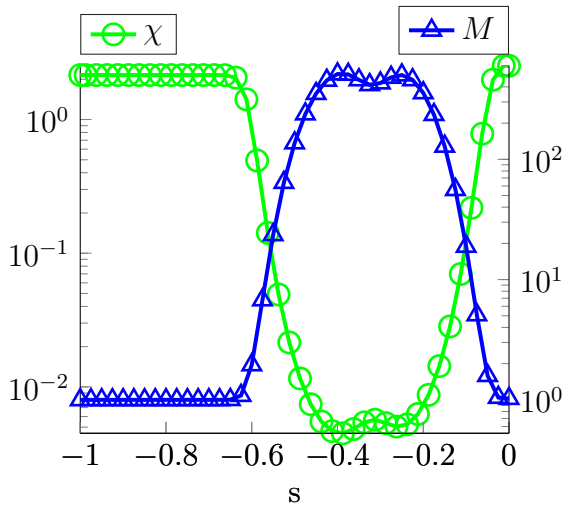
The following exemplary test uses one of the moving mesh strategies and parameters, analogously to what can be found in literature. The Burger's equation (C.5) is



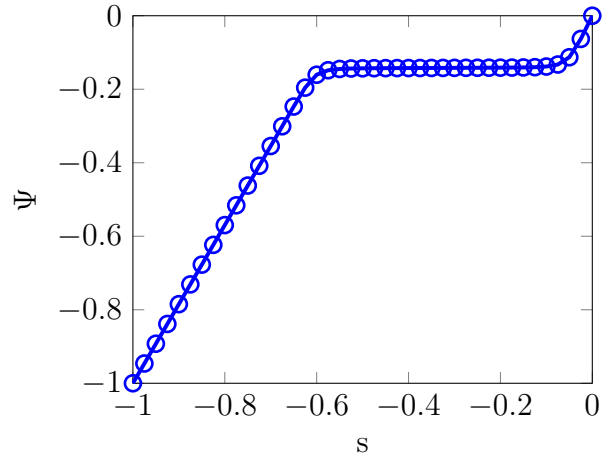
(a) Solution depicted from left to right at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ and zoom in at the jump at the end time $t = 1$.



(b) Trajectories of material points identified with s_k over time.

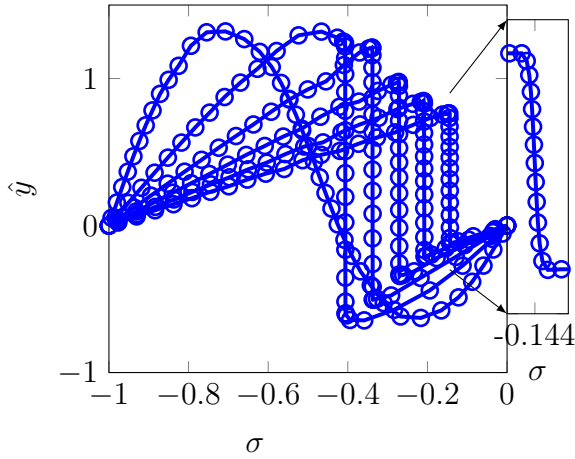


(c) Jacobian χ and monitor function M at the end time $t = 1$.

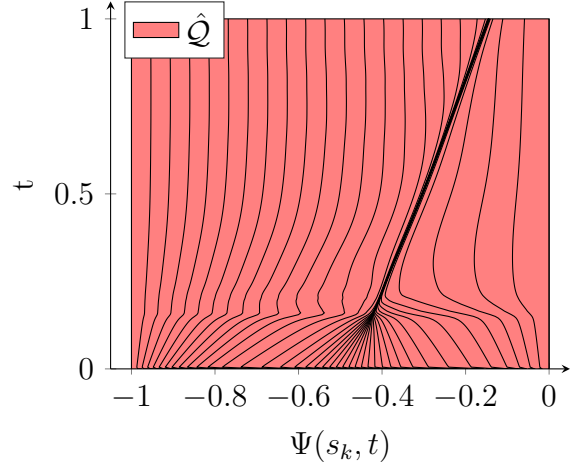


(d) The transformation Ψ at the end time $t = 1$.

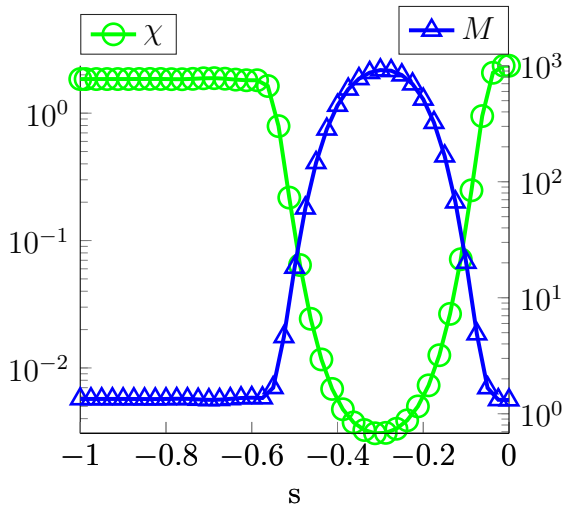
Figure C.7: The computed solution for Burger's equation with $\mu = 10^{-4}$, $\Delta s = 2.5 \cdot 10^{-2}$ and $\Delta t = 10^{-4}$ and the optimal curvature monitor function.



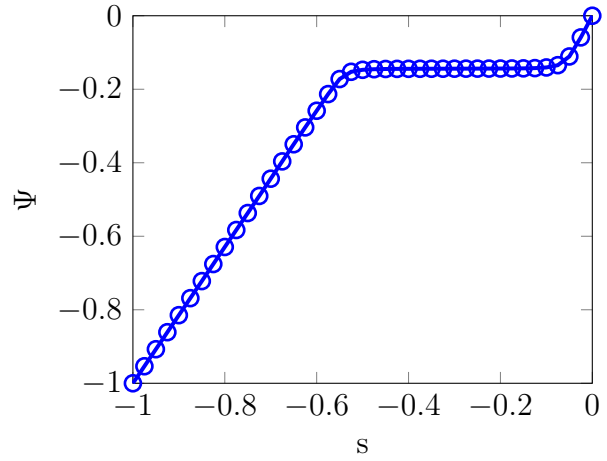
(a) Solution depicted from left to right at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ and zoom in at the jump at the end time $t = 1$.



(b) Trajectories of material points identified with s_k over time.



(c) Jacobian χ and monitor function M at the end time $t = 1$.



(d) The transformation Ψ at the end time $t = 1$.

Figure C.8: The computed solution for Burger's equation with $\mu = 10^{-4}$, $\Delta s = 2.5 \cdot 10^{-2}$ and $\Delta t = 10^{-4}$ and the arc-length monitor function.

solved with Strategy 3.10, (MMPDE2). The parameters are $\mu = 10^{-4}$ and $(\tau, \Delta s, \Delta t) = (10^{-2}, 2.5 \cdot 10^{-2}, \Delta t = 10^{-4})$. Results for the smoothed optimal curvature are illustrated in Figure C.7, and ones for the arc-length monitor function analogously shown in Figure C.8. In Figure C.7a the solution in referential parameterization is shown. No under- or overshooting, or oscillations appear, we magnified the steep front at the final time in the figure. Said steep moving front is properly resolved and tracked with the moving mesh approach, which is visible in the mesh trajectories in Figure C.7b. In Figure C.7c the Jacobian χ and the underlying monitor function at the final time is depicted. The strategy performs like expected by adapting to the monitor function. The transformation Ψ is depicted in Figure C.7d. It is defined to map from the computational to the referential parameterization, thus no steep layers occur when the mesh is very dense (which is a clear advantage compared to other alternative strategies using the inverse transformation [59]). We observe that our discrete scheme performs comparable to results in [59]. The broad stencils caused by our staggered scheme do not pose a problem (in contrast to results from the jet model, cf. Section 5.4).

C.3 Growing Eulerian parameterization with Lagrange tracking

The growing viscous cantilever was simulated in Section 5.1 with a Lagrangian parameterization. We now seek to use the approach described in Example 2.40b) with an Eulerian parameterization that globally prescribes $\tau_3 \equiv 1$. Then $\mathcal{L}(t)$ is the arc-length of the jet at the time t . During an actual simulation, the just mentioned arc-length of the jet of some arbitrary time is not known a priori, a free boundary value problem would have to be solved. We avoid this by precalculating $\mathcal{L}(t)$ with the help of the Lagrangian parameterization. Consider a solution at any time t_n . Before advancing with the normal algorithm the estimation of the arc-length $\mathcal{L}(t_n + \Delta t)$ is done with a purely Lagrangian sub-step. In particular, the given solution at time t_n is Eulerian. It is converted to solution in a Lagrangian parameterization and the time step is performed with it. The new Lagrangian solution now allows the calculation of $\mathcal{L}(t + \Delta t)$ and we switch back to the original simulation and continue to advance with the Eulerian parameterization. We call this approach Eulerian with Lagrange Tracking (ELT).

This approach creates a new issue in the time integration. The total amount of cells for any time can not be estimated a priori of the simulation because the jet growth is not predictable. It is thus possible that multiple cells are added within one time step, i.e. the jet is growing quickly. In those situations the accuracy of the approximation of the arc-length $\mathcal{L}(t + \Delta t)$ through the Lagrangian sub-step decreases and in consequence significantly alter the Eulerian solution. Therefore a time step control is added to avoid having to create multiple cells within one time step. If multiple cells would be inserted within one time step, it is reduced in some linear fashion and then repeated until no or only one new cell is needed. The respective time steps Δt_i are numbered increasingly until the end time is reached, $T = t_0 + \sum_{i=0}^k \Delta t_i, k \in \mathbb{N}$.

An exemplary study is done to get a comparison in computation effort and amount of cells required. System B.1_M is used, the test parameters are $(\text{Re}, \text{Fr}) = (1, 0.4), T = 5$. The

| | Δs | $\Delta t_0 / \min(\Delta t_i)$ | Time (sec) | # Cells | $\max(e)$ | jet arc-length |
|-------|------------|---------------------------------|------------|---------|-----------|----------------|
| (PL) | 1e-1 | 1e-1 | 5.08 | 52 | 26.60 | 55.4690 |
| | 1e-1 | 5e-3 | 71.97 | 52 | 23.75 | 50.9115 |
| | 5e-3 | 5e-3 | 112.29 | 1002 | 23.92 | 50.9089 |
| | 1e-3 | 1e-3 | 2760.76 | 5002 | 23.82 | 50.7245 |
| (ELT) | 1e-1 | 1e-1 / 4.11e-3 | 1018.10 | 513 | 15.80 | 51.1781 |
| | 1e-2 | 1e-2 / 4.15e-4 | 15915 | 5069 | 22.59 | 50.6730 |
| | 1e-3 | 1e-3 / 4.04e-5 | 1081487 | 50660 | 23.65 | 50.6583 |

Table C.1: Comparison of a purely Lagrangian solution (PL) to the Eulerian one with Lagrange tracking (ELT). The computation time, number of cells, maximum elongation and physical length of the jet are shown.

results for a purely Lagrangian solution (PL) and for the Eulerian solution with Lagrangian tracking (ELT) are presented in Table C.1. The results clearly favor (PL). The computation effort for (ELT) are exploding for small discretization parameters. Even if the (ELT) solution would be competitive in terms of computation time and accuracy, the solution produces boundary layers due to the mass lumping at the free end (visible in e.g. in the mass line density). It is unsuitable for further consideration.

C.4 Rotational spinning process

We redo the longtime simulation with the moving mesh strategies from Section 5.4.2 and the mesh control functions considered in the beginning of this chapter. The behavior of all is similar, the simulation breaks down before its Lagrangian counterpart, cf. Figure C.9. The oscillations start to develop early on and aggravate until ultimately causing the underlying Newton's method to diverge.

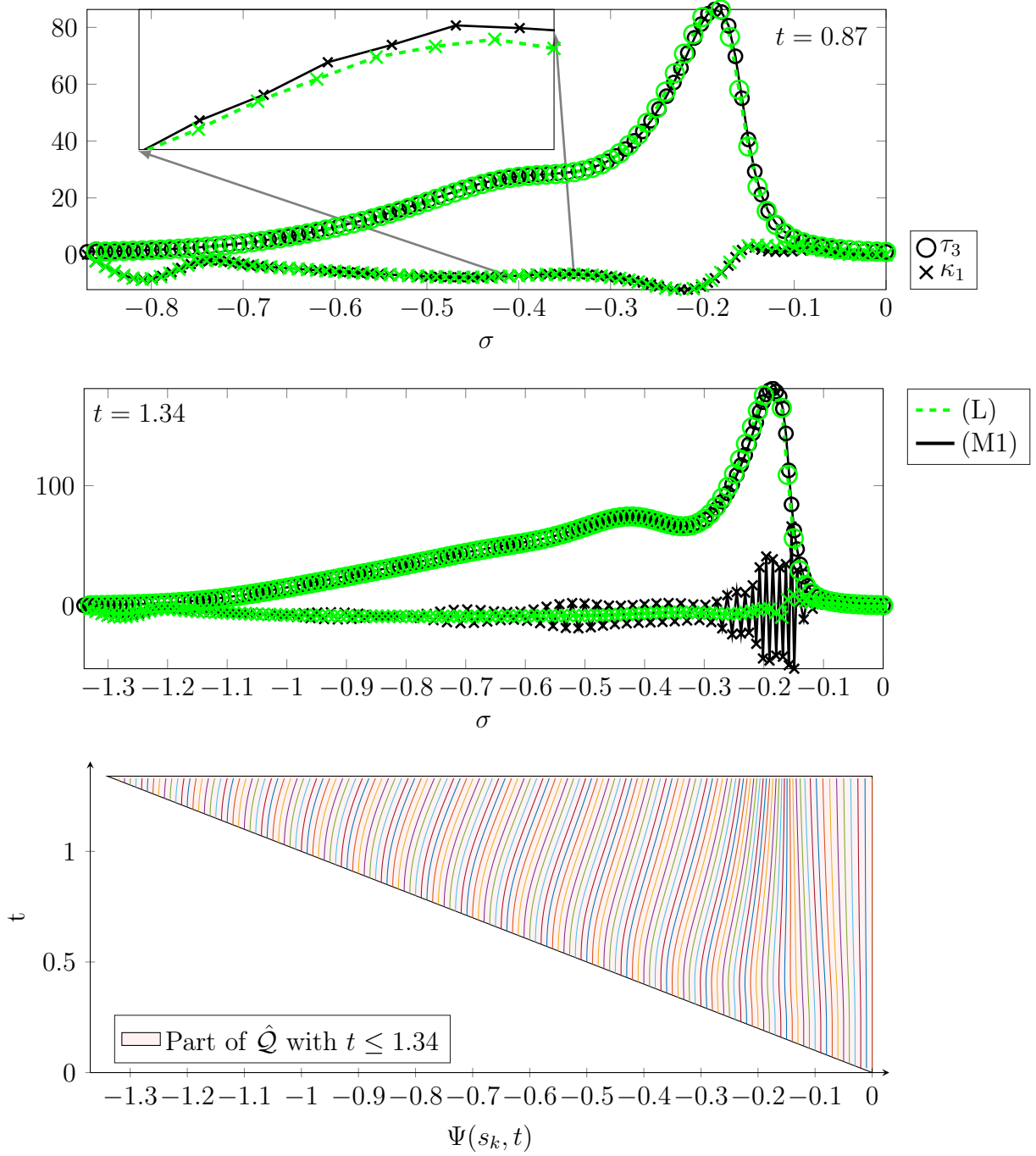


Figure C.9: Rotational spinning process: Comparison of a Lagrangian solution (L) and a moving mesh solution with MMPDE1 (Strategy 3.9) (M1) and the optimal curvature monitor function controlling the jet curve and orientation (cf. (5.1)). Parameters taken from Figure 6.3. Top two: Elongation and bending at two interim times. Observe the oscillation that develop early on and ultimately cause the moving mesh solution to break down before the Lagrangian counterpart. Bottom: Mesh trajectories until $t = 1.34$.

D Numerical scheme and implementation

D.1 Details to derivation of the Finite Volume method

In this appendix we give details on the derivation of the discrete scheme. The shape of the two-dimensional cells can be rectangular and trapezoidal, cf. Strategy 4.3. The rectangular cell can be seen as a special case of the trapezoidal cell, nevertheless we introduce them separately.

Rectangular cell Let one equations with index $j \in \{1, \dots, M\}$ and $\mathcal{D}_j = \text{node}$ be given. Consequently the domain splitting of node type is used. Assume a rectangular cells of node type with corners $(s_{i-1/2}, t_{n+1})$, $(s_{i+1/2}, t_{n+1})$, $(s_{i+1/2}, t_n)$ and $(s_{i-1/2}, t_n) \in \mathcal{Q}^h$. The cell spans the area C , the faces are named N, S, W, E (illustrated in Figure D.1). Define $\phi = (s, t)^T$, $\mathbf{F}(\phi) = (f_j, a_j)^T(\mathbf{y}(s, t))$ and $G(\phi) = g_j(\mathbf{y}(s, t))$. The divergence of \mathbf{F} then is $\nabla \cdot \mathbf{F}(\phi) = \partial_t a_j(\mathbf{y}(s, t)) + \partial_s f_j(\mathbf{y}(s, t))$. We integrate the node equations of (4.1) denoted through \mathbf{F} and ϕ over C

$$\int_C \nabla \cdot \mathbf{F} \, dC + \int_C G \, dC = 0 \quad (\text{D.1})$$

and apply Gauss' Theorem to express the integral over C through integrals over the faces

$$\sum_{K \in \{N, S, W, E\}} \int_K \mathbf{F} \cdot \boldsymbol{\eta}(K) \, dK + \int_C G \, dC = 0,$$

with outer normal $\boldsymbol{\eta}$ of the faces given by

$$\begin{aligned} \boldsymbol{\eta}(N) &= (0, 1)^T, & \boldsymbol{\eta}(W) &= (-1, 0)^T, \\ \boldsymbol{\eta}(S) &= (0, -1)^T, & \boldsymbol{\eta}(E) &= (1, 0)^T. \end{aligned}$$

All five integrals are approximated with quadrature rules. We choose the midpoint rule for N and S

$$\begin{aligned} \int_N \mathbf{F} \cdot \boldsymbol{\eta}(N) \, dN &= \int_{s_{i-1/2}}^{s_{i+1/2}} a_j(\mathbf{y}(s, t_{n+1})) \, ds \\ &= \Delta s a_j(\mathbf{y}(s_i, t_{n+1})) + \mathcal{O}(\Delta s^3), \\ \int_S \mathbf{F} \cdot \boldsymbol{\eta}(S) \, dS &= \int_{s_{i-1/2}}^{s_{i+1/2}} -a_j(\mathbf{y}(s, t_n)) \, ds \\ &= -\Delta s a_j(\mathbf{y}(s_i, t_n)) + \mathcal{O}(\Delta s^3) \end{aligned}$$

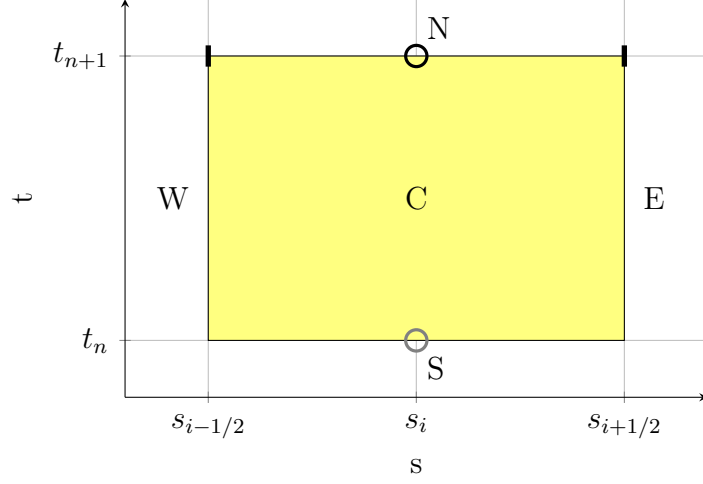


Figure D.1: Illustration of a singled out space-time cell. The nodes and edges that will be relevant for the numerical approach are highlighted.

and box rules for the integration of W and E

$$\begin{aligned} \int_W \mathbf{F} \cdot \boldsymbol{\eta}(W) dW &= \int_{t_n}^{t_{n+1}} -f_j(\mathbf{y}(s_{i-1/2}, t)) dt \\ &= -\Delta t_{n+1} f_j(\mathbf{y}(s_{i-1/2}, t_{n+1})) + \mathcal{O}(\Delta t_{n+1}^2), \\ \int_E \mathbf{F} \cdot \boldsymbol{\eta}(E) dE &= \int_{t_n}^{t_{n+1}} f_j(\mathbf{y}(s_{i+1/2}, t)) dt \\ &= \Delta t_{n+1} f_j(\mathbf{y}(s_{i+1/2}, t_{n+1})) + \mathcal{O}(\Delta t_{n+1}^2). \end{aligned}$$

The quadrature rule for the time integral would be equivalent to the well-known implicit Euler method in a semi-discretization. For the source terms we decompose the integral into space and time direction. Then we choose the quadrature rule analogously to the faces with midpoint rule for the spatial direction and box rule for the time direction

$$\begin{aligned} \int_C Q dC &= \int_{t_n}^{t_{n+1}} \int_{s_{i-1/2}}^{s_{i+1/2}} g_j(\mathbf{y}(s, t)) ds dt \\ &= \int_{t_n}^{t_{n+1}} \Delta s g_j(\mathbf{y}(s_i, t)) + \mathcal{O}(\Delta s^3) dt \\ &= \Delta s [\Delta t_{n+1} g_j(\mathbf{y}(s_i, t_{n+1})) + \mathcal{O}(\Delta t_{n+1}^2)] + \mathcal{O}(\Delta s^3 \Delta t_{n+1}) \\ &= \Delta s \Delta t_{n+1} g_j(\mathbf{y}(s_i, t_{n+1})) + \mathcal{O}(\Delta s \Delta t_{n+1}^2 + \Delta s^3 \Delta t_{n+1}) \end{aligned}$$

Putting everything together we obtain the approximations of the integrals for one cell

$$\begin{aligned} &\Delta s [a_j(\mathbf{y}(s_i, t_{n+1})) - a_j(\mathbf{y}(s_i, t_n))] \\ &+ \Delta t_{n+1} [f_j(\mathbf{y}(s_{i+1/2}, t_{n+1})) - f_j(\mathbf{y}(s_{i-1/2}, t_{n+1}))] + \Delta t_{n+1} \Delta s g_j(\mathbf{y}(s_i, t_{n+1})) \\ &+ \mathcal{O}(\Delta t_{n+1}^2 + \Delta s^3 + \Delta s \Delta t_{n+1}^2 + \Delta s^3 \Delta t_{n+1}) = 0. \end{aligned} \quad (\text{D.2})$$

A first-order Finite Volume method with constant cell-values is the goal. We want to decouple the time-integration, thus we defined spatial averages. Since we have unknowns of

node and edge type we define spatial averages for both, cf. (4.4). In (D.2) the functions are evaluated point-wise, we now evaluate them with the spatial averages $\mathbf{Y}_k(t_n) = \mathbf{y}(s_k, t_n) + \mathcal{O}(\Delta s^2)$, see Remark D.1.

Remark D.1 (Approximation with cell averages). *The unknowns at a given point s_k are approximated with spatial averages. We derive the approximation error for the different combination of node and edge in the following. Assume that y_j is node type, i.e. $\mathcal{D}_j = 1$. We analyze the situation for some time t at a node s_i*

$$\begin{aligned} y_j(s_i, t) - \Delta s^2 &= y_j(s_i, t) - \frac{1}{\Delta s} \int_{s_{i-1/2}}^{s_{i+1/2}} y_j(s, t) ds, \\ &= y_j(s_i, t) - \frac{1}{\Delta s} (\Delta s y_j(s_i, t) + \mathcal{O}(\Delta s^3)) \\ &= \mathcal{O}(\Delta s^2). \end{aligned}$$

Analogously at an edge $s_{i+1/2}$ with an additional Taylor expansion

$$\begin{aligned} y_j(s_{i+1/2}, t) - Y_{i+1/2,j}(t) &= y_j(s_{i+1/2}, t) - \frac{1}{2} \left[\frac{1}{\Delta s} \int_{s_{i-1/2}}^{s_{i+1/2}} y_j(s, t) ds + \frac{1}{\Delta s_{i+1}} \int_{s_{i+1/2}}^{s_{i+3/2}} y_j(s, t) ds \right] \\ &= y_j(s_{i+1/2}, t) - \frac{1}{2} [y_j(s_i, t) + y_j(s_{i+1}, t) + \mathcal{O}(\Delta s^2)] = \mathcal{O}(\Delta s^2). \end{aligned}$$

Now assume that y_j is edge type, i.e. $\mathcal{D}_j = 0$, the procedure is analogous:

$$\begin{aligned} y_j(s_i, t) - Y_{i,j}(t) &= \mathcal{O}(\Delta s^2) \\ y_j(s_{i+1/2}, t) - Y_{i+1/2,j}(t) &= \mathcal{O}(\Delta s^2). \end{aligned}$$

The time evolution and source term in (D.2) are evaluated at the node s_i

$$\begin{aligned} a_j(\mathbf{y}(s_i, t_{n+1})) &= a_j(\mathbf{Y}_i(t_{n+1})) + \mathcal{O}(\Delta s^2), \\ g_j(\mathbf{y}(s_i, t_{n+1})) &= g_j(\mathbf{Y}_i(t_{n+1})) + \mathcal{O}(\Delta s^2), \end{aligned}$$

the fluxes are to be evaluated at the edges $s_{i+1/2}$ and $s_{i-1/2}$, thus we have analogously

$$\begin{aligned} f_j(\mathbf{y}(s_{i+1/2}, t_{n+1})) &= f_j(\mathbf{Y}_{i+1/2}(t_{n+1})) + \mathcal{O}(\Delta s^2), \\ f_j(\mathbf{y}(s_{i-1/2}, t_{n+1})) &= f_j(\mathbf{Y}_{i-1/2}(t_{n+1})) + \mathcal{O}(\Delta s^2). \end{aligned}$$

The overall approximation of (D.2) then becomes

$$\begin{aligned} \Delta s [a_j(\mathbf{Y}_i(t_{n+1})) - a_j(\mathbf{Y}_i(t_n))] + \Delta t_{n+1} [f_j(\mathbf{Y}_{i+1/2}(t_{n+1})) - f_j(\mathbf{Y}_{i-1/2}(t_{n+1}))] \\ + \Delta s \Delta t_{n+1} g_j(\mathbf{Y}_i(t_{n+1})) + \mathbf{E}_{rec} = 0 \end{aligned} \quad (\text{D.3a})$$

and analogously if we do the approximation with an equation of edge type. The error is

$$\mathbf{E}_{rec} := \mathcal{O}(\Delta t_{n+1}^2 + \Delta s^3 + \Delta s \Delta t_{n+1}^2 + \Delta s^3 \Delta t_{n+1} + \Delta s^2 \Delta t_{n+1}).$$

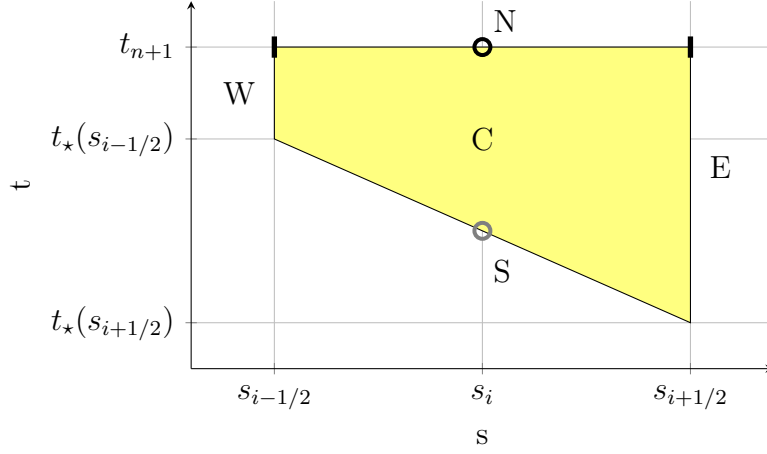


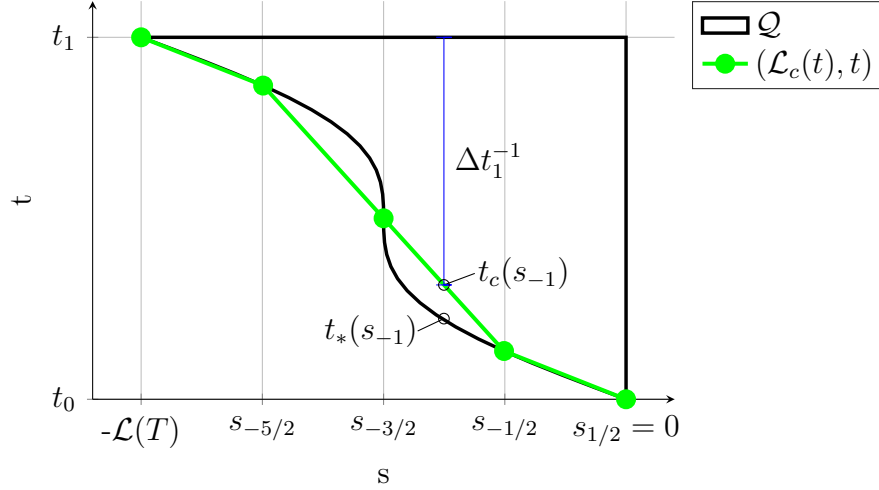
Figure D.2: Illustration of a slanted space-time cell. The circles and bars exemplify the sampling points that will be used in the numerical approach, the cross marks the location of the boundary at time t_{n+1} , the function t_* is the creation time.

The last term $\Delta s^2 \Delta t_{n+1}$ only appears due to the constant cell averages and the possible interpolation in (4.4). If the cell averages are exact for the midpoint of the cell, then this error appears only if interpolation occurs in (4.4).

So far the structure of the numerical formula and error order of a node and edge equation are analogously, the grouping according to node and edge type seems random. The full potential of this staggered approach unfolds due to the fact that for the jet model (and most likely many others) the node/edge assignment of the unknowns and equations can be done in such a way that a_j, g_j solely depends unknowns of type \mathcal{D}_j and f_j solely depends unknowns of the opposite type. We call the assignment then *ideally staggered*. The consequence is that no interpolation in Equation (4.4) is necessary and the approximation of the spatial flux is central using only direct neighboring values (interpreted as finite differences we obtain a narrow, central stencil with second-order accuracy).

The rectangular cell is now generalized to a trapezoidal one.

Slanted cell For a time-dependent space-time domain (see Figure 4.5b) we additionally need a trapezoidal cell to completely cover the domain. We explain the mechanism at hand with the example of a time-dependency on the left side, a time-dependent right side would be treated analogously. Assume one equations with index $j \in \{1, \dots, M\}$ and $\mathcal{D}_j = 1$. Assume a trapezoidal cell of node type with corners $(s_{i-1/2}, t_{n+1})$, $(s_{i+1/2}, t_{n+1})$, $(s_{i+1/2}, t_*(s_{i+1/2}, t_{n+1}))$ and $(s_{i-1/2}, t_*(s_{i-1/2}, t_{n+1})) \in \mathcal{Q}$. The creation time $t_*(s)$ of the point s appears, cf. Definition 2.13. An illustration of such a cell is given with Figure D.2. The difference to the rectangular cell are the faces W, E and S . We follow the outline of the rectangular cell and approximate the integral on every face separately. The faces W and E have an adjusted length of $\Delta t_{n+1}^{i-1/2}$ and $\Delta t_{n+1}^{i+1/2}$ respectively, with $\Delta t_{n+1}^k = t_{n+1} - t_*(s_k)$


 Figure D.3: Illustration of the linear approximation \mathcal{L}_c to \mathcal{L} .

(cf. Equation (4.3)) Their integrals are approximated with the box rule

$$\begin{aligned} \int_W \mathbf{F} \cdot \boldsymbol{\eta}(W) dW &= \int_{t_*(s_{i-1/2}, t_{n+1})}^{t_{n+1}} -f_j(\mathbf{y}(s_{i-1/2}, t)) dt \\ &= -\Delta t_{n+1}^{i-1/2} f_j(\mathbf{y}(s_{i-1/2}, t_{n+1})) + \mathcal{O}((\Delta t_{n+1}^{i-1/2})^2), \\ \int_E \mathbf{F} \cdot \boldsymbol{\eta}(E) dE &= \int_{t_*(s_{i+1/2}, t_{n+1})}^{t_{n+1}} f_j(\mathbf{y}(s_{i+1/2}, t)) dt \\ &= \Delta t_{n+1}^{i+1/2} f_j(\mathbf{y}(s_{i+1/2}, t_{n+1})) + \mathcal{O}((\Delta t_{n+1}^{i+1/2})^2) \end{aligned}$$

The face S is a slanted line with constant slope defined by

$$m_i = \frac{t_*(s_{i+1/2}, t_{n+1}) - t_*(s_{i-1/2}, t_{n+1})}{\Delta s}.$$

We restrict our choices of the domain to those cases where S exactly reflects the border of the domain, a general case is discussed in Remark D.2.

Remark D.2. *If there is no restriction on the border of the space-time domain, the face S gives only a linear approximation. The error can be incorporated in the scheme by introducing the polygonal chain \mathcal{L}_c defined through \mathcal{L} at the sampling points $t_*(s_{k+1/2})$, $k \in \mathbb{N}_0$, where $\mathcal{L}_c(s_{k+1/2}) = \mathcal{L}(s_{k+1/2})$. Furthermore the creation time has to be adjusted analogously, giving the approximated creation time t_c . An illustration of the distances and the linear approximation to the domain boundary is shown in Figure D.3.*

The outer normal of S is constant in space and time within one cell and given by

$$\boldsymbol{\eta}(S) = \left(\frac{m_i}{\sqrt{m_i^2 + 1}}, -\frac{1}{\sqrt{m_i^2 + 1}} \right) =: (\eta_s, \eta_t).$$

The face S of the slanted cell can be formalized by

$$S = \{(s, t) \in \mathbb{R}^- \times \mathbb{R}^+ | s \in [s_{i-1/2}, s_{i+1/2}], t = t_*(s)\},$$

we use the midpoint rule for the approximation of the integral

$$\begin{aligned} \int_S \mathbf{F} \cdot \boldsymbol{\eta}(S) dS &= \int_S \eta_t a_j(\mathbf{y}(s, t_\star(s))) + \eta_s f_j(\mathbf{y}(s, t_\star(s))) dS \\ &= \Delta s \eta_t a_j(\mathbf{y}(s_i, t_\star(s_i, t_{n+1}))) + \Delta s \eta_s f_j(\mathbf{y}(s_i, t_\star(s_i, t_{n+1}))) + \mathcal{O}(\Delta s^3) \end{aligned}$$

with $\Delta s = \sqrt{(t_\star(s_{k-1/2}) - t_\star(s_{k+1/2}))^2 + \Delta s^2}$. We simplify $\Delta s \eta_t = -\Delta s$ and $\Delta s \eta_s = \Delta t_{i,*} := t_\star(s_{i+1/2}, t_{n+1}) - t_\star(s_{i-1/2}, t_{n+1})$. The spatial fluxes have to be evaluated strictly implicit in time, thus we do Taylor expansions in time to pull it to the implicit time $t_{n+1} = t_\star(s_i, t_{n+1}) + \Delta t_{n+1}^i$ and one in space to split it equally to the edges $s_{i+1/2}$ and $s_{i-1/2}$:

$$\begin{aligned} \Delta s \eta_s f_j(\mathbf{y}(s_i, t_\star(s_i, t_{n+1}))) &= \\ \frac{\Delta t_{i,*}}{2} (f_j(\mathbf{y}(s_{i-1/2}, t_{n+1})) + f_j(\mathbf{y}(s_{i+1/2}, t_{n+1}))) &+ \mathcal{O}(\Delta t_{i,*} \Delta t_{n+1}^i + \Delta s^2 \Delta t_{i,*}). \end{aligned}$$

The combined weights for the fluxes $f_j(\mathbf{y}(s_{i+1/2}, t_{n+1}))$ and $f_j(\mathbf{y}(s_{i-1/2}, t_{n+1}))$ are

$$\begin{aligned} \left(\Delta t_{n+1}^{i+1/2} + \frac{\Delta t_{i,*}}{2} \right) &= t_{n+1} - t_\star(s_{i+1/2}, t_{n+1}) + \frac{1}{2} (t_\star(s_{i+1/2}, t_{n+1}) - t_\star(s_{i-1/2}, t_{n+1})) \\ &= t_{n+1} - \frac{1}{2} (t_\star(s_{i+1/2}, t_{n+1}) + t_\star(s_{i-1/2}, t_{n+1})) \\ &= t_{n+1} - t_\star(s_i) = \Delta t_{n+1}^i, \\ - \left(\Delta t_{n+1}^{i-1/2} - \frac{\Delta t_{i,*}}{2} \right) &= - \left(t_{n+1} - t_\star(s_{i-1/2}, t_{n+1}) - \frac{1}{2} (t_\star(s_{i+1/2}, t_{n+1}) - t_\star(s_{i-1/2}, t_{n+1})) \right) \\ &= - \left(t_{n+1} - \frac{1}{2} (t_\star(s_{i+1/2}, t_{n+1}) + t_\star(s_{i-1/2}, t_{n+1})) \right) \\ &= - (t_{n+1} - t_\star(s_i)) = -\Delta t_{n+1}^i \end{aligned}$$

respectively. We need to integrate the source term as well, we do that by decomposing the integral in two one-dimensional parts analogously to the rectangular cell

$$\begin{aligned} \int_C Q dC &= \int_{s_{i-1/2}}^{s_{i+1/2}} \int_{t_\star(s)}^{t_{n+1}} g_j(\mathbf{y}(s, t)) dt ds \\ &= \int_{s_{i-1/2}}^{s_{i+1/2}} \Delta t_{n+1}(s) g_j(\mathbf{y}(s, t_{n+1})) + \mathcal{O}((\Delta t_{n+1})^2) ds \\ &= \Delta s \Delta t_{n+1}^i g_j(\mathbf{y}(s_i, t_{n+1})) + \mathcal{O}(\Delta s (\Delta t_{n+1}^i)^2 + \Delta s^3 \Delta t_{n+1}^i) \end{aligned}$$

Overall we have for the approximations of the integrals for the slanted cell

$$\begin{aligned} \Delta s [a_j(\mathbf{y}(s_i, t_{n+1})) - a_j(\mathbf{y}(s_i, t_\star(s_i, t_{n+1})))] &+ \Delta t_{n+1}(s_i) [f_j(\mathbf{y}(s_{i+1/2}, t_{n+1})) - f_j(\mathbf{y}(s_{i-1/2}, t_{n+1}))] \\ &+ \Delta s \Delta t_{n+1}^i g_j(\mathbf{y}(s_i, t_{n+1})) \\ &+ \mathcal{O}((\Delta t_{n+1}^{i+1/2})^2 + (\Delta t_{n+1}^{i-1/2})^2 + \Delta s^3 + \Delta s (\Delta t_{n+1}^i)^2 + \Delta s^3 \Delta t_{n+1}^i) = 0. \end{aligned} \quad (\text{D.4})$$

In the final step we add the approximation with the spatial averages. The complete approximation then is, written in general notation for equations of node and edge type

$$\begin{aligned} \Delta s [a_j(\mathbf{Y}_k(t_{n+1})) - a_j(\mathbf{Y}_k(t_*(s_k)))] + \Delta t_{n+1}(s_k) [f_j(\mathbf{Y}_{k+1/2}(t_{n+1})) - f_j(\mathbf{Y}_{k-1/2}(t_{n+1}))] \\ + \Delta s \Delta t_{n+1}(s_k) g_j(\mathbf{Y}_k(t_{n+1})) + \mathbf{E}_{sla,k} = 0 \end{aligned} \quad (\text{D.5})$$

with

$$\mathbf{E}_{sla,k} = \mathcal{O}((\Delta t_{n+1}^{k+1/2})^2 + (\Delta t_{n+1}^{k-1/2})^2 + \Delta s^3 + \Delta s \Delta t_{n+1}(s_k)^2 + \Delta s^3 \Delta t_{n+1}^k + \Delta s^2 \Delta t_{n+1}^k).$$

D.2 Details on the application of the jet model

The viscous jet model from System 2.31 is presented with node/edge assignment. The unknowns and their type are printed preceding their assigned equation (here, **node** means that the respective $\mathcal{D}_j = 1$ and **edge** means $\mathcal{D}_j = 0$), the reasoning for the choice is given in Table D.1:

$$\bar{\mathbf{r}}, \text{node} : \quad \partial_t \bar{\mathbf{r}} = \mathbf{R}^T(\mathbf{q}) \cdot (\mathbf{v} - u\boldsymbol{\tau}) + \frac{1}{k} \partial_s \bar{\boldsymbol{\lambda}}_\tau, \quad (\text{D.6a})$$

$$\bar{\boldsymbol{\lambda}}_\tau, \text{edge} : \quad \partial_s \bar{\mathbf{r}} = \mathbf{R}^T(\mathbf{q}) \cdot \boldsymbol{\tau}, \quad (\text{D.6b})$$

$$\mathbf{q}, \text{node} : \quad \partial_t \mathbf{q} = \mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q} + \frac{1}{k} \mathcal{A}(\partial_s \boldsymbol{\lambda}_\kappa) \cdot \mathbf{q} + \lambda_t \mathbf{q}, \quad (\text{D.6c})$$

$$\boldsymbol{\lambda}_\kappa, \lambda_s, \text{edge} : \quad \partial_s \mathbf{q} = \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} + \lambda_s \mathbf{q}, \quad (\text{D.6d})$$

$$\boldsymbol{\tau}, \text{edge} : \quad \partial_t \boldsymbol{\tau} + \partial_s(u\boldsymbol{\tau}) = \partial_s \mathbf{v} + \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega} + Ck \mathbf{R}(\mathbf{q}) \cdot \bar{\boldsymbol{\lambda}}_\tau, \quad (\text{D.6e})$$

$$\boldsymbol{\kappa}, \text{edge} : \quad \partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa}) = \partial_s \boldsymbol{\omega} + \boldsymbol{\kappa} \times \boldsymbol{\omega} + Ck \boldsymbol{\lambda}_\kappa, \quad (\text{D.6f})$$

$$\boldsymbol{\lambda}_t, \text{node} : \quad \mathbf{q} \cdot \mathbf{q} = 1, \quad (\text{D.6g})$$

$$\chi, \text{node} : \quad \partial_t \chi + \partial_s(u\chi) = 0, \quad (\text{D.6h})$$

$$\mathbf{v}, \text{node} : \quad \partial_t \mathbf{p} + \partial_s(u\mathbf{p}) = \mathbf{p} \times \boldsymbol{\omega} + \partial_s \mathbf{n} + \boldsymbol{\kappa} \times \mathbf{n} + \mathbf{f}, \quad \mathbf{p} = \sigma_M \mathbf{v}, \quad (\text{D.6i})$$

$$\boldsymbol{\omega}, \text{node} : \quad \varepsilon^2 (\partial_t \mathbf{h} + \partial_s(u\mathbf{h})) = \varepsilon^2 \mathbf{h} \times \boldsymbol{\omega} + \partial_s \mathbf{m} + \boldsymbol{\kappa} \times \mathbf{m} + \boldsymbol{\tau} \times \mathbf{n} + \varepsilon^2 \mathbf{l}, \quad \mathbf{h} = \mathbf{J}_\mathbf{M} \cdot \boldsymbol{\omega}, \quad (\text{D.6j})$$

$$\mathbf{n}, \text{edge} : \quad \partial_t \tau_3 + \partial_s(u\tau_3) = \frac{\text{Re } \tau_3^2}{3\mu \sigma_V} \mathbf{n}_3, \quad \tau_1 = 0, \quad \tau_2 = 0, \quad (\text{D.6k})$$

$$\mathbf{m}, \text{edge} : \quad \varepsilon^2 (\partial_t \boldsymbol{\kappa} + \partial_s(u\boldsymbol{\kappa})) = \frac{\text{Re } \tau_3^3}{3\mu \sigma_V^2} \mathbf{M}_\mu^{-1} \cdot \mathbf{m}. \quad (\text{D.6l})$$

with $\mathbf{J}_\mathbf{M} = (\sigma_M \sigma_V / \tau_3) \mathbf{M}_{in}$. The parameter speed u is still a degree of freedom in the system, but assumed to be globally zero if not otherwise mentioned. The terms σ_V , σ_M , k are left in the system for clarity, but are not meant as unknowns anymore. They are to be substituted with χ , which is introduced according to Remark D.3.

| node | edge | description |
|---|---|---|
| $\chi, \mathbf{v}, \boldsymbol{\omega}$ | \mathbf{m}, \mathbf{n} | Stress free-jet end at $s = 0$ in every set-up, and $s_{1/2} = 0$ is the reference edge. |
| | $\boldsymbol{\tau}, \boldsymbol{\kappa}$ | Minimum amount of interpolation (narrow stencils) in conservation laws (D.6i), (D.6j) (\mathbf{m}, \mathbf{n} in their spatial flux). |
| $\bar{\mathbf{r}}, \mathbf{q}$ | | Material models (D.6l), (D.6k) are ODEs with \mathbf{m}, \mathbf{n} , which are edge type. |
| λ_t | | Connected through (D.6a) and (D.6c) to $\mathbf{v}, \boldsymbol{\omega}$, which are node type. |
| | $\boldsymbol{\lambda}_\tau, \boldsymbol{\lambda}_\kappa, \lambda_s$ | Unit quaternion constraint (D.6g) needs the same type as \mathbf{q} . |
| | | Connected through (D.6b) and (D.6d) to $\boldsymbol{\tau}, \boldsymbol{\kappa}$ (edge type) in the source terms and to $\bar{\mathbf{r}}$ and \mathbf{q} (node type) in the fluxes. |
| | u | Maintain perfect conservation in (D.6i), (D.6j) and (D.6h). |

Table D.1: Overview of the choice of node and edge unknowns.

Remark D.3. *There is a shortcut to calculate type-1 fields with simple conservation equations under certain assumptions. Remember the transformation rule for a type-1 field $f: \mathcal{Q} \rightarrow \mathbb{R}$:*

$$f(s, t) = \partial_s \Psi(s, t) \hat{f}(\Psi(s, t), t).$$

Assume that the Lagrangian quantity is independent of t and also independent of s in the initial values, thus $\hat{f}(\Psi(s, t), t) = f_0 \in \mathbb{R}$ for all $(s, t) \in \mathcal{Q}$. Then it holds that

$$f(s, t) = \chi(s, t) f_0. \quad (\text{D.7a})$$

The spatial functional determinant of the transformation $\chi(s, t) = \partial_s \Psi(s, t)$ appears, which is a conserved quantity itself by definition. Remember that the whole theory of this work emerged from the 3d continuum mechanics, in which we consider volume conservation in integral depiction under a time-dependent transformation as one of the very first basics. It can be seen by taking (2.16) and applying a spatial derivative to it to obtain

$$\partial_t \chi + \partial_s (u \chi) = 0. \quad (\text{D.7b})$$

That means all similar type-1 fields can be calculated with (D.7) – in summary only one equation to solve instead of one for every field.

To facilitate the application of our discrete scheme onto the viscous jet model (D.6) is now expressed with the functions \mathbf{a}, \mathbf{f} and \mathbf{q} in the general form (4.1) with the ordered unknowns $\mathbf{y} = (\bar{\mathbf{r}}, \bar{\boldsymbol{\lambda}}_\tau, \mathbf{q}, \boldsymbol{\lambda}_\kappa, \lambda_s, \boldsymbol{\tau}, \boldsymbol{\kappa}, \lambda_t, \mathbf{v}, \boldsymbol{\omega}, \chi, \mathbf{n}, \mathbf{m}, \bar{\boldsymbol{\lambda}}_\tau^s, \lambda_\kappa^s, u)$. The last equation is $u \equiv 0$, which would be exchanged when using a moving mesh strategy. The substitutes

$\bar{\lambda}_\tau^s = \partial_s \bar{\lambda}_\tau$ and $\lambda_\kappa^s = \partial_s \lambda_\kappa$ appear, and overall we have

$$\begin{aligned}
 \mathbf{a}(\mathbf{y}) = & \begin{pmatrix} \bar{\mathbf{r}} \\ \mathbf{0}_3 \\ \mathbf{q} \\ \mathbf{0}_4 \\ \boldsymbol{\tau} \\ \boldsymbol{\kappa} \\ 0 \\ \chi \mathbf{v} \\ \varepsilon^2 \mathbf{h} \\ \chi \\ \mathbf{0}_2 \\ \tau_3 \\ \boldsymbol{\kappa} \\ \mathbf{0}_3, \\ \mathbf{0}_3 \\ 0 \end{pmatrix}, & \mathbf{f}(\mathbf{y}) = & \begin{pmatrix} \mathbf{0}_3 \\ \bar{\mathbf{r}} \\ \mathbf{0}_4, \\ \mathbf{q} \\ u\boldsymbol{\tau} - \mathbf{v} \\ u\boldsymbol{\kappa} - \boldsymbol{\omega} \\ 0, \\ u\chi \mathbf{v} - \mathbf{n}, \\ \varepsilon^2 u \mathbf{h} - \mathbf{m} \\ u\chi \\ \mathbf{0}_2 \\ u\tau_3 \\ u\boldsymbol{\kappa} \\ \bar{\lambda}_\tau \\ \lambda_\kappa \\ 0 \end{pmatrix}, & \mathbf{q}(\mathbf{y}) = & \begin{pmatrix} \mathbf{R}^T(\mathbf{q}) \cdot (\mathbf{v} - u\boldsymbol{\tau}) + \frac{1}{k} \bar{\lambda}_\tau^s \\ \mathbf{R}^T(\mathbf{q}) \cdot \boldsymbol{\tau} \\ \mathcal{A}(\boldsymbol{\omega} - u\boldsymbol{\kappa}) \cdot \mathbf{q} + \frac{1}{k} \mathcal{A}(\lambda_\kappa^s) \cdot \mathbf{q} - \lambda_t \mathbf{q} \\ \mathcal{A}(\boldsymbol{\kappa}) \cdot \mathbf{q} + \mathbf{q} \lambda_s \\ \boldsymbol{\kappa} \times \mathbf{v} + \boldsymbol{\tau} \times \boldsymbol{\omega} + Ck \mathbf{R}(\mathbf{q}) \cdot \bar{\lambda}_\tau \\ \boldsymbol{\kappa} \times \boldsymbol{\omega} + Ck \lambda_\kappa \\ \mathbf{q} \cdot \mathbf{q} - 1 \\ \chi \mathbf{v} \times \boldsymbol{\omega} + \boldsymbol{\kappa} \times \mathbf{n} + \mathbf{f} \\ \varepsilon^2 \mathbf{h} \times \boldsymbol{\omega} + \boldsymbol{\kappa} \times \mathbf{m} + \boldsymbol{\tau} \times \mathbf{n} + \varepsilon^2 \mathbf{l} \\ 0 \\ \mathbf{0}_2 \\ (\text{Re} \tau_3^2 / 3\mu\chi) \mathbf{n}_3 \\ (\text{Re} \tau_3^3 / 3\mu\chi^2) \mathbf{M}_\mu^{-1} \cdot \mathbf{m} \\ \bar{\lambda}_\tau^s \\ \lambda_\kappa^s \\ u \end{pmatrix}
 \end{aligned}$$

with $\mathbf{h} = (\chi^2/\tau_3) \mathbf{M}_{in} \cdot \boldsymbol{\omega}$. Initial and boundary conditions are set according to Section 4.1.2. As an example we apply the jet extrusion process of Section 2.3.1 and use

$$\begin{aligned}
 \mathbf{A}(s) = & \begin{pmatrix} \bar{\mathbf{r}}^A(s) \\ C_{un,3} \\ \mathbf{q}^A(s) \\ C_{un,3} \\ C_{un} \\ \mathbf{e}_3 \\ \mathbf{0}_3 \\ C_{un} \\ \mathbf{e}_3 v^A(s) \\ \mathbf{0}_3 \\ 1 \\ C_{un,3} \\ C_{un,3} \\ C_{un,3} \\ C_{un,3} \\ C_{un} \end{pmatrix}, & \mathbf{A}_L(-\mathcal{L}(t), t) = & \begin{pmatrix} \bar{\mathbf{r}}^L \\ C_{un,3} \\ \mathbf{q}^L \\ C_{un,3} \\ C_{un} \\ \mathbf{e}_3 \chi^L(t) \\ \mathbf{0}_3 \\ C_{un} \\ \mathbf{e}_3 v^L \\ \mathbf{0}_3 \\ \chi^L(t) \\ C_{un,3} \\ C_{un,3} \\ C_{un,3} \\ C_{un,3} \\ C_{un} \end{pmatrix}.
 \end{aligned}$$

for the initial conditions and

$$\begin{aligned}
 \mathbf{L}(-\mathcal{L}(t), t, \mathbf{y}(-\mathcal{L}(t), t)) = & \quad \mathbf{R}(-\mathcal{R}(t), t, \mathbf{y}(-\mathcal{R}(t), t)) = \\
 \mathbf{y}(-\mathcal{L}(t), t) - \begin{pmatrix} \bar{\mathbf{r}}^L \\ \mathbf{C}_{un,3} \\ \mathbf{q}^L \\ \mathbf{C}_{un,3} \\ C_{un} \\ \star \mathbf{e}_3 \chi^L(t) \\ \star \mathbf{0}_3 \\ C_{un} \\ \mathbf{e}_3 v^L \\ \mathbf{0}_3 \\ \star \chi^L(t) \\ \mathbf{C}_{un,3} \\ \mathbf{C}_{un,3} \\ \mathbf{C}_{un,3} \\ \mathbf{C}_{un,3} \\ v^L / \chi^L(t) - \frac{d}{dt} \mathcal{L}(t) \end{pmatrix}, & \quad \mathbf{y}(-\mathcal{R}(t), t) - \begin{pmatrix} \mathbf{C}_{un,3} \\ \mathbf{0}_3 \\ \mathbf{C}_{un,3} \\ \mathbf{0}_3 \\ C_{un} \\ \star E_{r,\tau_3}(t) \\ \star E_{r,\kappa}(t) \\ C_{un} \\ \star \mathbf{E}_{r,\mathbf{v}}(t) \\ \star \mathbf{E}_{r,\omega}(t) \\ \star E_{r,\chi}(t^{n+1}) \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{C}_{un,3} \\ \mathbf{C}_{un,3} \\ (v^R / \chi^R - \frac{d}{dt} \mathcal{R})(t) \end{pmatrix}
 \end{aligned}$$

for the spatial boundaries. We use abbreviations $\mathbf{0}_{x,x} \in \mathbb{N}$ for a n -dimensional zero and $\mathbf{C}_{un,x}$ for an n -dimensional vector with C_{un} as the components and $E_{r,x}(t^{n+1})$ as the extrapolation boundary of the unknown x . Artificial auxiliary conditions have been marked with a leading \star , conditions that are only required when u is not zero are marked with $*$. Additionally, it is assumed that $\chi^L(t)$ and $\chi^R(t)$ are given by the grid motion induced by the artificial convection speed. In the special case of a Lagrangian parameterization ($u \equiv 0$) with arc-length reference, we have $\chi^L(t) = \chi^R(t) = 1$ for all $t \in [0, T]$.

Notations

Notational conventions used in the chapters are summarized here.

Scalars, vectors, matrices and more

| | |
|--|--|
| x | Scalar-valued quantity |
| \vec{x} | Invariant vector in \mathbb{E}^3 |
| \mathcal{A} | Tensor-valued quantity |
| \boldsymbol{x} | Components of a vector depicted in canonical basis of \mathbb{R}^n |
| \mathbf{x} | Component triple in \mathbb{R}^3 of a vector depicted in the director basis |
| $\bar{\mathbf{x}}$ | Component triple in \mathbb{R}^3 of a vector depicted in the outer basis |
| $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ | Canonical basis of the respective basis |
| \mathbf{A} | Matrix in $\mathbb{R}^{n \times n}$ |
| \mathbb{N} | Natural number: $\mathbb{N} = \{1, 2, 3, \dots\}$ |
| \mathbb{Z} | Integer numbers: $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ |
| \mathbb{R} | Rational numbers |
| $ x $ | Absolute value of x |
| $\ \mathbf{x}\ $ | Euclidean norm |
| $\mathbf{x} \cdot \mathbf{y}$ | Inner / scalar product |
| $\mathbf{x} \times \mathbf{y}$ | Cross-product |
| $\mathbf{x} \otimes \mathbf{y}$ | Dyadic product |
| $\mathbb{1}, \text{id}$ | Identity matrix resp. mapping |
| \mathbf{A}^T | Transpose matrix of \mathbf{A} |
| \mathbf{A}^{-1} | Inverse of the matrix \mathbf{A} |
| \mathbf{f}^{-1} | Inverse of a bijective mapping \mathbf{f} |
| x^{-1} | Reciprocal of the scalar x |
| $\text{diag}(\mathbf{x})$ | Diagonal matrix with \mathbf{x} as the diagonal entries |
| \mathbf{f} | Mapping from \mathbb{R}^m to \mathbb{R}^n |
| $\partial_x \mathbf{f}$ | Component-wise partial derivative of \mathbf{f} with respect to x , $\partial_x \mathbf{f} = \sum \mathbf{e}_i \partial_x f_i$ |
| $\frac{d}{dx} \mathbf{f}$ | Component-wise total derivative of \mathbf{f} with respect to x , $\frac{d}{dx} \mathbf{f} = \sum \mathbf{e}_i \frac{d}{dx} f_i$ |
| $\nabla \mathbf{f}$ | Component-wise gradient of \mathbf{f} , i.e. for a $\nabla \mathbf{f} = (\nabla f_1 \dots \nabla f_n)$ |
| $D\mathbf{f}$ | Jacobian of \mathbf{f} , $D\mathbf{f} = (\nabla \mathbf{f})^T$ |
| \mathcal{I} | Interval in \mathbb{R} , $\mathcal{I} = [a, b]$ with $a, b \in \mathbb{R}$ and $a < b$ |

| | |
|---|---|
| $\overset{\circ}{\mathcal{I}}$ | Interior of \mathcal{I} |
| $\partial\mathcal{I}$ | Boundary of \mathcal{I} |
| $\mathcal{O}(f)$ | Function satisfying $ \mathcal{O}(f)/f \leq k$ with a positive constant k |
| $\mathcal{C}^l(\mathbb{R}^m, \mathbb{R}^n)$ | l -times continuously differentiable functions on \mathbb{R}^m with image in \mathbb{R}^n . |

Jet model

| | |
|--|---|
| σ | Spatial coordinate in Lagrangian parameterization |
| s | Spatial coordinate in general parameterization |
| t | Time |
| $\hat{\mathcal{Q}}$ | Lagrangian space-time domain |
| \mathcal{Q} | General space-time domain |
| Ψ | Transformation from the general to the Lagrangian domain |
| $\hat{\mathbf{x}}, \hat{\mathbf{x}}, \hat{\mathbf{x}}, \hat{\mathbf{x}}$ | Unknown in Lagrangian parameterization (starting Section 2.2.1) |
| $\vec{\mathbf{x}}, \mathbf{x}, \bar{\mathbf{x}}, \mathbf{x}$ | Unknown in general parameterization (starting Section 2.2.1) |
| $\vec{\mathbf{r}}, \bar{\mathbf{r}}$ | Jet curve (invariant, in outer basis) |
| $\vec{\mathbf{d}}_1, \vec{\mathbf{d}}_2, \vec{\mathbf{d}}_3$ | Orthonormal director triad, director basis |
| $\{\vec{\mathbf{a}}_1, \vec{\mathbf{a}}_2, \vec{\mathbf{a}}_3\}$ | Outer basis, possibly time-dependent |
| \mathcal{R}, \mathbf{R} | Rotation between outer and director basis and its associated matrix in SO(3) |
| \mathbf{q} | Quaternion to parameterize the rotation, relative to the outer basis |
| $\vec{\mathbf{v}}, \mathbf{v}$ | Velocity (invariant, in director basis) |
| $\vec{\boldsymbol{\omega}}, \boldsymbol{\omega}$ | Angular velocity (invariant, in director basis) |
| $\vec{\boldsymbol{\tau}}$ | Tangent (invariant), |
| $\boldsymbol{\tau}$ | Distortion measure $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$. Shear strains: τ_1, τ_2 . Stretching strain: τ_3 . Elongation: $\ \boldsymbol{\tau}\ $ |
| $\vec{\boldsymbol{\kappa}}$ | Curvature (invariant) |
| $\boldsymbol{\kappa}$ | Distortion measure $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \kappa_3)$. Bending: κ_1, κ_2 . Torsion: κ_3 |
| σ_M | Mass line density |
| σ_V | Volume line density |
| $\vec{\mathbf{n}}$ | Contact force |
| \mathbf{n} | $\mathbf{n} = (n_1, n_2, n_3)$. Shear stress: n_1, n_2 . Tension: n_3 |
| $\vec{\mathbf{m}}$ | Contact couple |
| \mathbf{m} | $\mathbf{m} = (m_1, m_2, m_3)$. Bending torque: m_1, m_2 . Twisting torque: m_3 |
| $\vec{\mathbf{f}}, \mathbf{f}$ | External body force line density |
| $\vec{\mathbf{l}}, \mathbf{l}$ | External body couple line density |
| μ | Dynamic viscosity |
| u | Parameter speed |
| \mathbf{J}_M | Inertia matrix |
| M_{in} | Cross-section specific part of inertia |
| M_v | Cross-section specific part of viscous torques |
| \mathbf{L}_p | Helper matrix, given by $\mathbf{L}_p = \text{diag}(1, 1, p)/(4\pi)$ for $p \in \mathbb{R}$ |
| A | Area of the cross-section, given by $A = \sigma_V / \ \boldsymbol{\tau}\ $ |

| | |
|-----------------------------|---|
| Re | Reynolds number (ratio between inertia and viscosity) |
| ε | Slenderness number (ratio of the length to the cross section diameter) |
| $\lambda_T, \lambda_\kappa$ | Multiplier of the SAMW correction |
| k | Fictitious type-1 field |
| C | Constant $\ll 1$ used in the SAMW correction |
| χ | Jacobian of the transformation Ψ ; Substitute for type-1 fields with simple conservation law |

Applications

| | |
|---------------------------|--|
| Ω | Rotational frequency |
| \mathbf{f}_Ω | Fictitious rotational body force line density |
| \mathbf{l}_Ω | Fictitious rotational body couple line density |
| Rb | Rossby number (ratio of inertial to centrifugal forces) |
| \mathbf{f}_g | Gravitational force |
| g | Gravitational constant |
| Fr | Froude number (ratio of inertia to gravity) |
| \mathbf{v}_{rel} | Relative velocity of jet and airflow |
| $\bar{\mathbf{v}}_{air}$ | Velocity of the airflow |
| $\bar{\mathbf{v}}_{mean}$ | Mean velocity of the airflow |
| $\bar{\mathbf{v}}_{turb}$ | Velocity of the turbulent fluctuations |
| ρ_{air} | Density of the airflow |
| ν_{air} | Kinematic viscosity of the airflow |
| \mathbf{f}_{air} | Aerodynamic line density suitable for the jet model |
| \mathbf{f}_{MW} | Aerodynamic line density with its local non-dimensionalization according to [75] |
| Vi_* | Relation of the dynamic viscosities of the airflow and the jet |
| Re_* | Mixed Reynolds number (ratio of inertia of the jet to the viscosity of the airflow) |
| k_{air} | Kinetic energy of the turbulent fluctuations |
| ϵ_{air} | Viscous dissipation of the turbulent fluctuations |
| l_T | Turbulent large-scale length, $l_T = k^{3/2}/\epsilon$ |
| t_T | Turbulent time for decay of large-scale vortices, $t_T = k/\epsilon$ |
| $\bar{\mathbf{v}}_{HM}$ | Velocity of the turbulent fluctuations with its local non-dimensionalization according to [60] |
| Tu | Degree of turbulence |
| Tt | Time scale ratio of jet model and turbulence model |
| T_{air} | Temperature of the airflow |
| α_{air} | Heat transfer coefficient |
| λ_{air} | Thermal conductivity |

| | |
|-----------------|---|
| $c_{p,air}$ | Specific heat capacity of the airflow |
| c_p | Specific heat capacity of the jet |
| T | Jet temperature |
| Nu | Nusselt number (ratio of convective to conductive heat transfer normal to the surface of two fluids) modeled according to [107] |
| Pr | Prandtl number (ratio of viscosity to thermal diffusivity) |
| Pr_* | Mixed Prandtl number (ratio of viscosity to thermal diffusivity from air to jet) |
| c_1, c_2, c_3 | Physical constants for the Arrhenius law of the viscosity |
| De | Deborah number (ratio of the time of relaxation to the time of observation) |

r -refinement

| | |
|------------------------|---|
| $\sigma, \hat{\Omega}$ | Parameter and corresponding domain of the referential parameterization |
| s, Ω | Parameter and corresponding domain of the computational parameterization |
| $p, \hat{\Omega}$ | Parameter and corresponding domain of the desired parameterization |
| t | Time |
| Ψ | Transformation of the computational parameterization to the referential parameterization, or parameter distribution |
| α | Transformation of the computational parameterization to the desired parameterization, or parameter distribution |
| ν | Transformation of the desired parameterization to the referential parameterization, or parameter distribution |
| Ξ, f_Ξ | Generic parameter distribution $\Xi \in \{\Psi, \alpha, \nu\}$ and the corresponding parameter density. |
| \hat{M}, M | Monitor function in referential and computational parameterization, i.e. $\hat{M}(\Psi(s, t), t) = M(s, t)$ |
| τ | Temporal relaxation parameter |
| G | Spatial smoothing operator |
| f^G | Application of the smoothing operator to a scalar-valued function f , i.e. $f^G = G^{-1}f$ |
| M^{smo} | Smoothed monitor function |
| f_α^{smo} | Smoothed parameter density |

Discrete scheme

| | |
|--------------|--|
| \mathbf{y} | Vector of unknowns |
| \mathbf{a} | Temporal flux function |
| \mathbf{f} | Spatial flux function |
| \mathbf{g} | Source function |
| s | Spatial parameter |
| t | Time |
| i | Index for integer values |
| n | Index for natural numbers (zero excluded) |
| k | Index for spatial grid, $k \in \mathcal{K} = \{\dots, -\frac{3}{2}, -1, -\frac{1}{2}, 0, \frac{1}{2}, 1, \frac{3}{2}, \dots\}$ |

| | |
|--|---|
| j | Index for counting the vector of unknowns |
| \mathcal{Q}^h | Discrete space-time domain: Set of all discrete points enclosed by \mathcal{Q} |
| $\Omega^h(t_n)$ | Set of all discrete points with $t = t_n$ in \mathcal{Q}^h |
| $\mathcal{T}(k)$ | Type of all discrete points with $s = s_k$, whereas $\mathcal{T}(k) = 1$ means node and $\mathcal{T}(k) = 0$ edge . |
| \mathcal{D} | Type of the unknowns \mathbf{y} where y_j has type \mathcal{D}_j |
| $\mathcal{Q}^{h,A}$ | Active space-time domain: Set of all active discrete points in \mathcal{Q}^h |
| $\mathcal{Q}^{p_l-p_r}$ | Special active space-time domain: p_l and p_r give the type of the boundary, meaning that considering all time levels separately, the adjacent active discrete point in $\mathcal{Q}^{p_l-p_r}$ is of opposite type |
| $Y_{k,j}^n$ | Discrete unknown (spatial average) at $s = s_k$ and t_n if $\mathcal{T}(k) = \mathcal{D}_j$, otherwise not defined |
| $\mathbf{Y}_k(t)$ | Approximation of the spatial averages for all points in the active space-time domain for $t \in \{t_1, \dots\}$, otherwise initial values. |
| Δs | Equidistant spatial grid spacing |
| $\Delta t_n, \Delta t_n^k$ | Temporal grid spacing and adjusted temporal grid spacing (incorporates slanted boundary cells) |
| k_l^n, k_r^n | First and last active discrete point (counting starting with smallest s) |
| $\mathbf{A}, \mathbf{A}_L, \mathbf{A}_R$ | Initial value functions |
| \mathbf{L}, \mathbf{R} | Boundary functions |
| $\mathbf{Y}_L^n, \mathbf{Y}_R^n$ | Spatial boundary unknowns |
| \mathbb{Y}^n | Solution calculated with the discrete scheme at time level t_n |
| $Z_j(s; \mathbb{Y}^n)$ | Spatial reconstruction of the solution for $s \in \Omega$ |
| $err(\mathbb{Y}_1^n, \mathbb{Y}_2^n, \mathcal{J})$ | L^2 -error calculated of the two given solutions for all components given in the index vector \mathcal{J} |
| $\Omega_{h,n}^p$ | All discrete points in \mathcal{Q}^h with $t = t_n$ and $s = s_k$ with k such that $\mathcal{T}(k) = p$ |

Bibliography

- [1] R. ALEXANDER, *Diagonally implicit Runge–Kutta methods for stiff O.D.E. 's*, SIAM Journal on Numerical Analysis, 14 (1977), pp. 1006–1021.
- [2] D. A. ANDERSON, *Application of adaptive grids to transient problems*, in Adaptive Computational Methods for Partial Differential Equations, Proceedings in Applied Mathematics Series, Society for Industrial and Applied Mathematics, 1983, pp. 208–223.
- [3] S. S. ANTMAN, *Nonlinear Problems of Elasticity*, vol. 107 of Applied Mathematical Sciences, Springer-Verlag, New York, 2005.
- [4] W. ARNE, *Viskose Jets in rotatorischen Spinnprozessen*, PhD thesis, Universität Kassel, 2013.
- [5] W. ARNE, N. MARHEINEKE, A. MEISTER, S. SCHIESSL, AND R. WEGENER, *Finite Volume approach for the instationary Cosserat rod model describing the spinning of viscous jets*, Journal of Computational Physics, 294 (2015), pp. 20–37.
- [6] W. ARNE, N. MARHEINEKE, A. MEISTER, AND R. WEGENER, *Numerical analysis of Cosserat rod and string models for viscous jets in rotational spinning processes*, Mathematical Models and Methods in Applied Sciences, 20 (2010), pp. 1941–1965.
- [7] —, *Numerical treatment of non-stationary viscous Cosserat rod in a two-dimensional Eulerian framework*, in Progress in Industrial Mathematics at ECMI 2012, Springer International Publishing, 2014, pp. 99–107.
- [8] W. ARNE, N. MARHEINEKE, J. SCHNEBELE, AND R. WEGENER, *Fluid-fiber-interactions in rotational spinning process of glass wool production*, Journal of Mathematics in Industry, 1 (2011), p. 2.
- [9] W. ARNE, N. MARHEINEKE, AND R. WEGENER, *Asymptotic transition from Cosserat rod to string models for curved viscous inertial jets*, Mathematical Models and Methods in Applied Sciences, 21 (2011), pp. 1987–2018.
- [10] —, *Viscoelastic Cosserat rod model for spinning processes*, in Progress in Industrial Mathematics at ECMI 2016, to appear, Springer, 2017.
- [11] U. M. ASCHER AND L. R. PETZOLD, *Projected implicit Runge–Kutta methods for differential-algebraic equations*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 1097–1120.

- [12] B. AUDOLY, N. CLAUVELIN, P.-T. BRUN, M. BERGOU, E. GRINSPUN, AND M. WARDETZKY, *A discrete geometric approach for simulating the dynamics of thin viscous threads*, Journal of Computational Physics, 253 (2013), pp. 18–49.
- [13] M. J. BAINES, M. E. HUBBARD, AND P. K. JIMACK, *Velocity-based moving mesh methods for nonlinear partial differential equations*, Communications in Computational Physics, 10 (2011), pp. 509–576.
- [14] J. BAUMGARTE, *Stabilization of constraints and integrals of motion in dynamical systems*, Computer Methods in Applied Mechanics and Engineering, 1 (1972), pp. 1–16.
- [15] F. BAUS, A. KLAR, N. MARHEINEKE, AND R. WEGENER, *Low-Mach-number-slenderness limit for elastic Cosserat rods*, arXiv:1507.03432, (2015).
- [16] M. J. BERGER, P. COLELLA, AND O. F. COMPUTATIONAL, *Local adaptive mesh refinement for shock hydrodynamics*, Journal of Computational Physics, 82 (1989), pp. 64–84.
- [17] M. J. BERGER AND R. J. LEVEQUE, *Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems*, SIAM Journal on Numerical Analysis, 35 (1998), pp. 2298–2316.
- [18] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, Journal of Computational Physics, 53 (1984), pp. 484–512.
- [19] M. BERGOU, B. AUDOLY, E. VOUGA, M. WARDETZKY, AND E. GRINSPUN, *Discrete viscous threads*, ACM Transactions on Graphics, 29 (2010), pp. 116:1–116:10.
- [20] M. BERGOU, M. WARDETZKY, S. ROBINSON, B. AUDOLY, AND E. GRINSPUN, *Discrete elastic rods*, ACM Transactions on Graphics, 27 (2008), pp. 63:1–63:12.
- [21] J. G. BLOM AND J. G. VERWER, *On simple moving grid methods for one-dimensional evolutionary partial differential equations*, Journal of Computational Physics, 213 (1988), pp. 191–213.
- [22] D. BOFFI AND L. GASTALDI, *Stability and geometric conservation laws for ALE formulations*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 4717–4739.
- [23] H. F. BRINSON AND L. C. BRINSON, *Polymer Engineering Science and Viscoelasticity*, Springer US, Boston, MA, 2015.
- [24] P.-T. BRUN, B. AUDOLY, N. M. RIBE, T. S. EAVES, AND J. R. LISTER, *Liquid Ropes: A geometrical model for thin viscous jet instabilities*, Physical Review Letters, 114 (2015), p. 174501.
- [25] P.-T. BRUN, N. M. RIBE, AND B. AUDOLY, *A numerical investigation of the fluid mechanical sewing machine*, Physics of Fluids, 24 (2012), p. 043102.

-
- [26] C. J. BUDD, W. HUANG, AND R. D. RUSSELL, *Adaptivity with moving grids*, Acta Numerica, 18 (2009), pp. 111–241.
 - [27] W. CAO, W. HUANG, AND R. D. RUSSELL, *A moving mesh method based on the geometric conservation law*, SIAM Journal on Scientific Computing, 24 (2002), pp. 118–142.
 - [28] S. CHIU-WEBSTER AND J. R. LISTER, *The fall of a viscous thread onto a moving surface: a ‘fluid-mechanical sewing machine’*, Journal of Fluid Mechanics, 569 (2006), pp. 89–111.
 - [29] J. COYLE, J. E. FLAHERTY, AND R. LUDWIG, *On the stability of mesh equidistribution strategies for time-dependent partial differential equations*, Journal of Computational Physics, 62 (1986), pp. 26–39.
 - [30] N. ČRNJARIĆ-ŽIC, B. CRNKOVIĆ, AND S. MAĆEŠIĆ, *A numerical study of SSP time integration methods for hyperbolic conservation laws*, Mathematical Communications, 15 (2010), pp. 613–633.
 - [31] L. J. CUMMINGS AND P. D. HOWELL, *On the evolution of non-axisymmetric viscous fibres with surface tension, inertia and gravity*, Journal of Fluid Mechanics, 389 (1999), pp. 361–389.
 - [32] C. DE BOOR, *Good approximation by splines with variable knots. II*, in Conference on the Numerical Solution of Differential Equations, 1974, pp. 12–20.
 - [33] S. DECENT, A. KING, AND I. WALLWORK, *Free jets spun from a prilling tower*, Journal of Engineering Mathematics, 42 (2002), pp. 265–282.
 - [34] P. DEUFLHARD, W. C. RHEINBOLDT, AND F. BORNEMANN, *Scientific Computing with Ordinary Differential Equations*, vol. 42 of Texts in Applied Mathematics, Springer New York, 2012.
 - [35] J. N. DEWYNNE, J. R. OCKENDON, AND P. WILMOTT, *A systematic derivation of the leading-order equations for extensional flows in slender geometries*, Journal of Fluid Mechanics, 244 (1992), pp. 323–338.
 - [36] J. N. DEWYNNE AND P. WILMOTT, *Slender axisymmetric fluid jets*, Mathematical and Computer Modelling, 18 (1993), pp. 69–82.
 - [37] J. DONEA AND A. HUERTA, *Arbitrary Lagrangian–Eulerian methods*, Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd, 2004.
 - [38] E. DORFI AND L. DRURY, *Simple adaptive grids for 1-D initial value problems*, Journal of Computational Physics, 69 (1987), pp. 175–195.
 - [39] V. M. ENTOV AND A. L. YARIN, *The dynamics of thin liquid jets in air*, Journal of Fluid Mechanics, 140 (1984), pp. 91–111.

- [40] R. FAZIO AND R. J. LEVEQUE, *Moving-mesh methods for one-dimensional hyperbolic problems using CLAWPACK*, Computers & Mathematics with Applications, 45 (2003), pp. 273–298.
- [41] L. FERRACINA AND M. N. N. SPIJKER, *Strong stability of singly-diagonally-implicit Runge–Kutta methods*, Applied Numerical Mathematics, 58 (2008), pp. 1675–1686.
- [42] C. W. GEAR, *Differential-algebraic equation index transformations*, SIAM Journal on Scientific and Statistical Computing, 9 (1988), pp. 39–47.
- [43] C. W. GEAR, B. LEIMKUHLER, AND G. K. GUPTA, *Automatic integration of Euler-Lagrange equations with constraints*, Journal of Computational and Applied Mathematics, 12-13 (1985), pp. 77–90.
- [44] T. GÖTZ, A. KLAR, A. UNTERREITER, AND R. WEGENER, *Numerical evidence for the non-existence of stationary solutions to the equations describing rotational fiber spinning*, Mathematical Models and Methods in Applied Sciences, 18 (2008), pp. 1829–1844.
- [45] H. GUILLARD AND C. FARHAT, *On the significance of the geometric conservation law for flow computations on moving meshes*, Computer Methods in Applied Mechanics and Engineering, 190 (2000), pp. 1467–1482.
- [46] E. HAIRER, C. LUBICH, AND M. ROCHE, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, Springer-Verlag, Berlin Heidelberg, 1989.
- [47] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II*, vol. 14 of Springer Series in Computational Mathematics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.
- [48] A. HARTEN, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics, 135 (1997), pp. 260–278.
- [49] C. HERTEL, M. SCHÜMICHEN, J. LANG, AND J. FRÖHLICH, *Using a moving mesh PDE for cell centres to adapt a Finite Volume grid*, Flow, Turbulence and Combustion, 90 (2013), pp. 785–812.
- [50] H. HEUSER, *Gewöhnliche Differentialgleichungen: Einführung in Lehre und Gebrauch*, Aus dem Programm Mathematik, Vieweg+Teubner Verlag, 2009.
- [51] R. G. HINDMAN, *Generalized coordinate forms of governing fluid equations and associated geometrically induced errors*, AIAA Journal, 20 (1982), pp. 1359–1367.
- [52] C. HIRT, A. AMSDEN, AND J. COOK, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, Journal of Computational Physics, 14 (1974), pp. 227–253.
- [53] C. W. HIRT, *An arbitrary Lagrangian-Eulerian computing technique*, in Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Berlin, Heidelberg, 1971, Springer Berlin Heidelberg, pp. 350–355.

-
- [54] A. V. HLOD, *Curved Jets of Viscous Fluid: Interactions with a Moving Wall*, PhD thesis, Eindhoven University of Technology, 2009.
- [55] P. D. HOWELL, *Extensional Thin Layer Flows*, PhD thesis, St. Catherine's College Oxford, 1994.
- [56] W. HUANG, *Practical aspects of formulation and solution of moving mesh partial differential equations*, Journal of Computational Physics, 171 (2001), pp. 753–775.
- [57] W. HUANG, Y. REN, AND R. D. RUSSELL, *Moving mesh partial differential equations (MMPDES) based on the equidistribution principle*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 709–730.
- [58] W. HUANG AND R. D. RUSSELL, *Analysis of moving mesh partial differential equations with spatial smoothing*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 1106–1126.
- [59] —, *Adaptive Moving Mesh Methods*, vol. 17 of Applied Mathematical Sciences, Springer Science & Business Media, New York Dordrecht Heidelberg London, 2010.
- [60] F. HÜBSCH, N. MARHEINEKE, K. RITTER, AND R. WEGENER, *Random field sampling for a simplified model of Melt-Blowing considering turbulent velocity fluctuations*, Journal of Statistical Physics, 150 (2013), pp. 1115–1137.
- [61] J. M. HYMAN AND B. LARROUTUROU, *Dynamic rezone methods for partial differential equations in one space dimension*, Applied Numerical Mathematics (IMACS)umerical mathematics, 5 (1989), pp. 435–450.
- [62] H. JASAK AND A. D. GOSMAN, *Automatic resolution control for the Finite-Volume method, Part 2: Adaptive mesh refinement and coarsening*, Numerical Heat Transfer, Part B, 38 (2000), pp. 257–271.
- [63] S. KASE AND T. MATSUO, *Studies on melt spinning. I. Fundamental equations on the dynamics of melt spinning*, Journal of Polymer Science Part A: General Papers, 3 (1965), pp. 2541–2554.
- [64] J. KAUTSKY AND N. K. NICHOLS, *Equidistributing meshes with constraints*, SIAM Journal on Scientific and Statistical Computing, 1 (1980), pp. 499–511.
- [65] J. KAUTSKY AND N. K. NICHOLS, *Smooth regridding of discretized data*, SIAM Journal on Scientific and Statistical Computing, 3 (1982), pp. 145–159.
- [66] D. I. KETCHESON, C. B. MACDONALD, AND S. GOTTLIEB, *Optimal implicit strong stability preserving Runge–Kutta methods*, Applied Numerical Mathematics, 59 (2009), pp. 373–392.
- [67] A. KLAR, N. MARHEINEKE, AND R. WEGENER, *Hierarchy of mathematical models for production processes of technical textiles*, ZAMM, 89 (2009), pp. 941–961.

- [68] J. F. B. M. KRAAIJEVANGER, *Contractivity of Runge-Kutta methods*, BIT, 31 (1991), pp. 482–528.
- [69] P. KUNKEL AND V. MEHRMANN, *A new class of discretization methods for the solution of linear differential-algebraic equations with variable coefficients*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 1941–1961.
- [70] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, vol. 54 of Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
- [71] P. LÖTSTEDT AND L. PETZOLD, *Numerical solution of nonlinear differential equations with algebraic constraints I: Convergence results for backward differentiation formulas*, Mathematics of Computation, 46 (1986), pp. 491–491.
- [72] N. MARHEINEKE, J. LILJO, J. MOHRING, J. SCHNEBELE, AND R. WEGENER, *Multiphysics and multimethods problem of rotational glass fiber melt-spinning*, International Journal of Numerical Analysis and Modeling, Series B, 3 (2012), pp. 330–344.
- [73] N. MARHEINEKE AND R. WEGENER, *Fiber dynamics in turbulent flows: General modeling framework*, SIAM Journal on Applied Mathematics, 66 (2006), pp. 1703–1726.
- [74] —, *Asymptotic model for the dynamics of curved viscous fibres with surface tension*, Journal of Fluid Mechanics, 622 (2009), pp. 345–369.
- [75] —, *Modeling and application of a stochastic drag for fibers in turbulent flows*, International Journal of Multiphase Flow, 37 (2011), pp. 136–148.
- [76] M. A. MATOVICH AND J. R. A. PEARSON, *Spinning a molten threadline. Steady-state isothermal viscous flows*, Industrial & Engineering Chemistry Fundamentals, 8 (1969), pp. 512–520.
- [77] K. MELIGAN, *Asymptotical and Numerical Analysis of Viscoelastic Material Laws for Fiber-Modelling*, master thesis, Friedrich-Alexander-University Erlangen-Nürnberg, 2016.
- [78] B. ONG, R. D. RUSSELL, AND S. RUUTH, *An hr moving mesh method for one-dimensional time-dependent PDEs*, Proceedings of the 21st International Meshing Roundtable, (2013), pp. 1–16.
- [79] S. PANDA, *The Dynamics of Viscous Fibers*, PhD thesis, Technische Universität Kaiserslautern, 2006.
- [80] S. PANDA, N. MARHEINEKE, AND R. WEGENER, *Systematic derivation of an asymptotic model for the dynamics of curved viscous fibers*, Mathematical Methods in the Applied Sciences, 31 (2008), pp. 1153–1173.
- [81] J. R. A. PEARSON AND M. A. MATOVICH, *Spinning a molten threadline. Stability*, Industrial & Engineering Chemistry Fundamentals, 8 (1969), pp. 605–609.

-
- [82] J. D. D. PRYCE, *On the convergence of iterated remeshing*, IMA Journal of Numerical Analysis, 9 (1989), pp. 315–335.
 - [83] J.-M. QIU AND C.-W. SHU, *Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow*, Journal of Computational Physics, 230 (2011), pp. 863–889.
 - [84] Y. REN AND R. D. RUSSELL, *Moving mesh techniques based upon equidistribution, and their stability*, SIAM Journal on Scientific and Statistical Computing, 13 (1992), pp. 1265–1286.
 - [85] N. M. RIBE, *Coiling of viscous jets*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 460 (2004), pp. 3223–3239.
 - [86] N. M. N. RIBE, M. HABIBI, AND D. BONN, *Stability of liquid rope coiling*, Physics of Fluids, 74 (2006), pp. 1–12.
 - [87] S. SCHIESSL, W. ARNE, N. MARHEINEKE, AND R. WEGENER, *DAE-index monitoring for semidiscretized viscous Cosserat rod models*, in Proceedings in Applied Mathematics and Mechanics (PAMM), Wiley, 2013, pp. 501–502.
 - [88] S. SCHIESSL, N. MARHEINEKE, W. ARNE, AND R. WEGENER, *An adaptive moving mesh approach for hyperbolic conservation laws on time-dependent domains*, in Proceedings in Applied Mathematics and Mechanics (PAMM), Wiley, 2014, pp. 957–958.
 - [89] —, *A Finite Volume method with staggered grid on time-dependent domains for viscous fiber spinning*, in Progress in Industrial Mathematics at ECMI 2016, to appear, Springer, 2017.
 - [90] S. SCHIESSL, N. MARHEINEKE, AND R. WEGENER, *A moving mesh framework based on three parametrization layers for 1d PDEs*, in Progress in Industrial Mathematics at ECMI 2014, Springer, 2017, pp. 945–951.
 - [91] L. F. SHAMPINE, *Solving $0 = F(t, y(t), y'(t))$ in MatLab*, Journal of Numerical Mathematics, 10 (2002), pp. 291–310.
 - [92] C.-W. SHU, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer Berlin Heidelberg, 1998, pp. 325–432.
 - [93] S. SINHA-RAY, A. L. YARIN, AND B. POURDEYHIMI, *Meltblowing: I-basic physical mechanisms and threadline model*, Journal of Applied Physics, 108 (2010), p. 034912.
 - [94] A. R. SOHEILI AND J. M. STOCKIE, *A moving mesh method with variable mesh relaxation time*, Applied Numerical Mathematics, 58 (2008), pp. 249–263.
 - [95] A. STEINBRECHER, *Numerical Solution of Quasi-Linear Differential-Algebraic Equations and Industrial Simulation of Multibody Systems*, PhD thesis, Technischen Universität Berlin, 2006.

- [96] J. M. STOCKIE, J. A. MACKENZIE, AND R. D. RUSSELL, *A moving mesh method for one-dimensional hyperbolic conservation laws*, SIAM Journal on Scientific Computing, 22 (2001), pp. 1791–1813.
- [97] D. SUCKER AND H. BRAUER, *Stationärer Stoff- und Wärmeübergang an stationär quer angeströmten Zylindern*, Wärme- Und Stoffübertragung, 9 (1976), pp. 1–12.
- [98] H. TANG AND T. TANG, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 487–515.
- [99] T. TANG, *Moving mesh methods for computational fluid dynamics*, Contemporary mathematics, 383 (2005), pp. 1–33.
- [100] P. D. THOMAS AND C. K. LOMBARD, *Geometric conservation law and Its application to flow computations on moving grids*, AIAA Journal, 17 (1979), pp. 1030–1037.
- [101] M. A. J. UYTENDAELE AND R. L. SHAMBAUGH, *Melt blowing: General equation development and experimental verification*, AIChE Journal, 36 (1990), pp. 175–186.
- [102] A. VAN DAM AND P. ZEGELING, *A robust moving mesh Finite Volume method applied to 1D hyperbolic conservation laws from magnetohydrodynamics*, Journal of Computational Physics, 216 (2006), pp. 526–546.
- [103] B. VAN LEER, *Towards the ultimate conservative difference scheme*, Journal of Computational Physics, 135 (1997), pp. 229–248.
- [104] VDI-GESELLSCHAFT, *VDI-Wärmeatlas*, VDI Buch, Springer Berlin Heidelberg, Berlin, Heidelberg, 10 ed., 2006.
- [105] H. K. VERSTEEG AND W. MALALASEKERA, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Pearson Education, New York, 2nd ed., 2007.
- [106] R. WEGENER AND W. ARNE, *Private communication*, 2017.
- [107] R. WEGENER, N. MARHEINEKE, AND D. HIETEL, *Virtual production of filaments and fleeces*, Springer, 2015.
- [108] J. WISNIAK, *Frederick Thomas Trouton: The Man, the Rule, and the Ratio*, The Chemical Educator, 6 (2001), pp. 55–61.
- [109] S. XIE AND Y. ZENG, *Turbulent air flow field and fiber whipping motion in the melt blowing process: Experimental study*, Industrial & Engineering Chemistry Research, 51 (2012), pp. 5346–5352.
- [110] X. YANG, W. HUANG, AND J. QIU, *A moving mesh WENO method for one-dimensional conservation laws*, SIAM Journal on Scientific Computing, 34 (2012), pp. A2317–A2343.

- [111] A. L. YARIN, *Free Liquid Jets and Films: Hydrodynamics and Rheology*, Longman, New York, 1993.
- [112] Y. C. ZENG, Y. F. SUN, AND X. H. WANG, *Numerical approach to modeling fiber motion during melt blowing*, Journal of Applied Polymer Science, 119 (2011), pp. 2112–2123.

Curriculum vitae

Name

Stefan Schiefl,
born 21. April 1986 in Erlangen

Education

| | |
|------------------------|---|
| Sept. 1992 - Jul. 2005 | Abitur, Ortenburg Gymnasium, Oberviechtach |
| Sept. 2005 - Feb. 2008 | IT specialist, Apprenticeship, Klug integrierte Systeme, Teunz |
| Oct. 2008 - Sept. 2011 | Bachelor of Science Technical Mathematics, Friedrich-Alexander University Erlangen-Nürnberg (FAU) |
| Aug. 2011 - May 2012 | Study abroad, Minnesota State University, Mankato |
| Oct. 2011 - Mar. 2013 | Master of Science Technical Mathematics, FAU |
| Mar. 2013 - now | Phd student in mathematics (Dr. rer. nat.), FAU, doctoral advisor Prof. Dr. Nicole Marheineke |

Internships / Professional career

| | |
|-----------------------|---|
| Feb. 2008 - Sep. 2008 | IT specialist, Klug integrierte Systeme, Teunz |
| Dez. 2008 - Mar. 2010 | Student intern, Siemens AG Healthcare Sector, Erlangen |
| May 2010 - Feb. 2013 | Student assistant, FAU, overall 7 months in 3 positions |
| Aug. 2011 - May 2012 | Graduate assistant, Minnesota State University, Mankato |
| Jun. 2012 - Aug. 2012 | Internship, Fraunhofer Institute for Industrial Mathematics, Kaiserslautern |
| Mar. 2013 - now | Research assistant, FAU, Erlangen |

This work investigates models and numerical strategies for the behavior of a slender object deformed by large external forces in spinning processes. Our main consideration is an incompressible, highly viscous and three-dimensional jet that is described by a one-dimensional model of partial differential-algebraic equations. We aim to provide a robust basis for the simulation of production processes that require transient treatment and prevent any meaningful simplification of the model equations. The spatial domain is considered time-dependent and requires proper handling. For that purpose, a Finite Volume method for an arbitrary space-time domain is proposed. The performance of the model and discrete scheme is validated through numerical convergence order results (in space, time and combined). As examples for industrial applications we consider production processes of insulation with the rotational spinning process and nonwoven materials with the melt-blowing process. Both exhibit large elongations that manifest in strongly varying solution components, possibly causing numerical difficulties. Possibilities with adaptive mesh refinement (in particular r-refinement) are also explored.



FRAUNHOFER VERLAG