No. 651 March 2022

Mathematical Modeling of Coolant Flow in Drilling Processes with Temperature Coupling

M. Fast, O. Mierka, S. Turek

ISSN: 2190-1767

Mathematical Modeling of Coolant Flow in Drilling Processes with Temperature Coupling

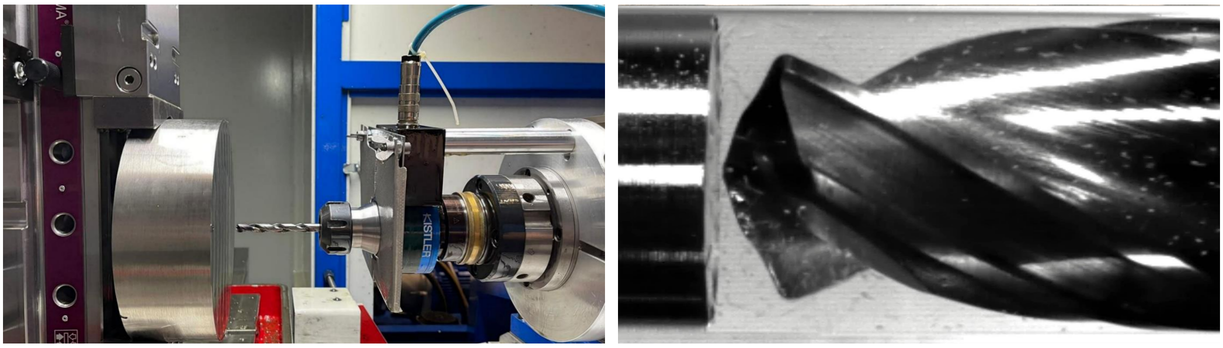
Michael Fast, Otto Mierka, Stefan Turek
Institute for Applied Mathematics, LSIII, TU Dortmund University, D-44227 Dortmund Germany
michael.fast@mathematik.dortmund.de

Abstract: The paper presents a mathematical modeling approach for a novel drilling strategy with coolant flow. Numerical tools for efficient simulations of such drilling applications are explained. We model the fluid flow with the Navier-Stokes equation in a rotational frame of reference and the solid domain is treated with the Fictitious Boundary Method (FBM). This enables us to utilize a unified mesh for the solid and fluid part of the domain and heat transfer between these is treated in an implicit way.

Keywords— drilling simulation, CFD, FEM, discontinuous drilling, Fictitious Boundary Method

1. Introduction

Mathematical modeling and designing efficient numerical methods for application in industrial engineering is a challenging task. Prototypically we consider the numerical simulation of drilling processes of the nickel-base alloy Inconel 718 as depicted in figure 1.1. Inconel 718 is widely used in aerospace industry, power plant construction as well as in chemical industries due to its low thermal conductivity and high toughness. These properties also mean that the alloy is hard to machine which leads to high thermal loads on the cutting edge and thus accelerated tool wear. To reduce the influence of high temperatures on the tool the use of coolant is essential.



(a) Experimental setup

(b) drill with coolant inside acrylic glass

Fig. 1.1. Experimental setup of drilling processes using coolant

Since drilling is a process with a fast rotating tool and the contact zone is inaccessible to measurement instruments, the experimental analysis of coolant flow and heat transfer is nearly impossible. Thus, in order to gain insight into the process, the application of high resolution numerical simulation is a key tool. In this paper we will focus on numerical components to simulate the coolant flow and the heat transfer between tool, workpiece and fluid, omitting chip formation and solid mechanics. The simulation for the fluid flow as well as the temperature development will be performed with the open source Finite Element Method (FEM) CFD software FeatFlow [1], which has proven its efficiency and accuracy in several benchmarks [2, 5, 11]. Fully three-dimensional CFD simulations tend to be very time consuming, so efficient numerical solvers are essential. FeatFlow is a highly parallelized general purpose 3D CFD-simulation software, with great scalability properties making it capable as FEM package for High Performance Computer Clusters. The combination of robust domain decomposition techniques and fast multigrid solvers specially tailored for complex 3D fluid flow simulations ensure computational efficiency. Several extensions concerning numerical methods for non-Newtonian viscosity, shear rate and temperature dependent, and for particulate flow and moving internal obstacles have been implemented making FeatFlow a powerful simulation software for both academic and industrial applications [9, 13, 14].

First let us translate the machining process into algorithmic challenges in the context of Finite Element simulations: rotational movement of the drill results in a highly time-dependent computational domain in addition to the complex geometry of the drilling tool and its coolant channels. An accurate representation of the geometry demands a high-resolution computational mesh during all times. Not only that we have to deal with high angular velocities but also high velocities due to the inflow, resulting in high Reynolds numbers. Further, we need a detailed representation of the fluid and solid domain, solving the Navier-Stokes equation in the fluid part and the heat equation in a coupled manner in the union of the solid and fluid domain. An efficient and flexible combination of the aforementioned task is immensely challenging. The high frequent motion of the drilling tool makes it difficult to construct an appropriate computational grid at all timesteps while still taking fluid structure interaction due to the rigid body motion into account. Very small timestep sizes are needed to track the interface of the solid body and accurately calculate velocities making the simulations time consuming and computational costly. Additionally, construction of an highly resolved mesh for all times is demanding since the mesh nodes have to be adapted frequently to the nonstationary domain. Mesh free methods can circumvent these difficulties, albeit having lower accuracy. Supplementary heat transfer between solid and fluid and consequently the solid-fluid interface has to be captured precisely.

To our knowledge mathematical models for CFD simulations of deep hole drilling and algorithmic details of temperature simulations of drilling applications in its entirely are rare. Steady-state simulation analysis of the flow characteristics of drilling applications were done in [12] using a rotational frame of reference, however effects of nonstationary flow and heat transfer were omitted. Mathematical modeling of fluid flow inside coolant channels for different rotational velocities were conducted in [6], showing that the flow field inside the cooling channels does not depend on the angular velocities. Numerical algorithms for Finite Element simulations of temperature due to friction and deformations during drilling of Inconel 718 with minimal quantity lubrication were introduced in [3], using simulated results to take the deformation of the workpiece due to temperatures into account.

In our approach, we will transform the momentum equation into a rotational frame of reference, thus gaining a time independent computational domain. However, Coriolis forces have to be taken into account and modifications of the solver have to be made. This transformation grants us the possibility to make use of a highly resolved mesh for all time steps for both the fluid and solid part of the domain. The Fictitious Boundary Method (FBM) in combination with mesh deformation techniques enables us to differentiate between solid and fluid domain, the transition boundary from solid to fluid can be described with high accuracy using mesh deformation to align the computational grid with the tool or workpiece geometry. The heat equation is then solved on the unified mesh. The proposed tools are finally applied on prototypical simulations for a novel discontinuous drilling strategy.

2. Numerical components for coolant flow

The fluid motion is modeled by the nonstationary incompressible Navier-Stokes equation

$$\begin{array}{rcll} \rho\frac{\partial \boldsymbol{v}}{\partial t} - \nu\Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p & = & \boldsymbol{f} & \text{in } \Omega_f \times (0,T) \\ \operatorname{div} \boldsymbol{v} & = & 0 & \text{in } \Omega_f \times (0,T) \end{array} \tag{2.1}$$

with given forces $\boldsymbol{f}$, constant density ρ and kinematic viscosity ν. We simplify the rotation of the drill so that the angular velocity $\boldsymbol{\omega} := (0,0,\omega)^\top$ with $\omega \in \mathbb{R}$ is constant and the tool rotates around the z-axis. We denote the fluid and solid domain with Ω_f and Ω_s, respectively, and $\Omega = \Omega_f \cup \Omega_s$. Note that Ω is time dependent and rotates around the z-axis. With a radius vector $\boldsymbol{r}$ we get the transformed velocity $\boldsymbol{u} = \boldsymbol{v} + (\boldsymbol{\omega}\times\boldsymbol{r})$ and hence the Navier-Stokes equation in a rotational frame of reference becomes

$$\begin{array}{rcll} \rho\frac{\partial \boldsymbol{u}}{\partial t} - \nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}) + \nabla p & = & \boldsymbol{f} & \text{in } \Omega_f \times (0,T) \\ \operatorname{div} \boldsymbol{u} & = & 0 & \text{in } \Omega_f \times (0,T) \end{array} \tag{2.2}$$

with Coriolis forces $2\boldsymbol{\omega}\times\boldsymbol{u}$ and centrifugal forces $\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r})$. Using the identity $\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}) = -\frac{1}{2}\nabla(\boldsymbol{\omega}\times\boldsymbol{r})^2$ we can include the centrifugal forces into the pressure with $P = p - \frac{1}{2}(\boldsymbol{\omega}\times\boldsymbol{r})^2$ and finally we have

$$\begin{array}{rcll} \rho\frac{\partial \boldsymbol{u}}{\partial t} - \nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla P & = & \boldsymbol{f} & \text{in } \Omega_f \times (0,T) \\ \operatorname{div} \boldsymbol{u} & = & 0 & \text{in } \Omega_f \times (0,T). \end{array} \tag{2.3}$$

For detailed derivation and analysis of equation (2.3) we like to direct your attention to the literature [16].

First we discretize the transformed Navier-Stokes Equation (2.3) in time with the θ-scheme. Let $\boldsymbol{u}^n$ be the velocity and a given time t_n, solve for $\boldsymbol{u}^{n+1}$ at time t_{n+1} with timestep $\Delta t = t_{n+1} - t_n$

$$\begin{array}{rcll} \rho\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\Delta t} + \theta\big((\boldsymbol{u}^*\cdot\nabla)\boldsymbol{u}^{n+1} - \nu\Delta\boldsymbol{u}^{n+1} + 2\boldsymbol{\omega}\times\boldsymbol{u}^{n+1}\big) + \nabla P^{n+1} & = & \boldsymbol{rhs}^{n+1} & \text{in } \Omega_f \times (0,T) \\ \operatorname{div} \boldsymbol{u}^{n+1} & = & 0 & \text{in } \Omega_f \times (0,T) \end{array} \tag{2.4}$$

with right-hand side

$$\boldsymbol{rhs}^{n+1} = \theta\boldsymbol{f}^{n+1} + (1-\theta)\boldsymbol{f}^n - (1-\theta)\big((\boldsymbol{u}^*\cdot\nabla)\boldsymbol{u}^n - \nu\Delta\boldsymbol{u}^n + 2\boldsymbol{\omega}\times\boldsymbol{u}^n\big)$$

with a suitable approximation $\boldsymbol{u}^*$ of $\boldsymbol{u}^{n+1}$. Setting $\boldsymbol{u}^* = \boldsymbol{u}^{n+1}$ or $\boldsymbol{u}^* = \boldsymbol{u}^n$ results in an implicit or explicit scheme, respectively. We consider the mixed Finite Element spaces Q_2P_1 for the velocity and pressure, respectively, as the spatial discretization. It is well known that this element is numerically stable and satisfies the Babuska-Brezzi condition

$$\inf_{p_h\in P_1}\sup_{\boldsymbol{u}_h\in Q_2}\frac{(p_h,\nabla\cdot\boldsymbol{u}_h)}{||p_h||_0||\nabla_h\boldsymbol{u}_h||_0} \geq \gamma > 0$$

with a mesh independent constant $\gamma > 0$ [4]. Defining $N(\boldsymbol{u})$ and L as the discrete counterpart to the continuous operators $\boldsymbol{u} \cdot \nabla$ and Δ, respectively, the discrete gradient B and the mass matrix M, we get the discrete coupled equation of (2.4)

$$\begin{pmatrix} A & -2\Delta t\theta\omega M & 0 & \Delta t B_1 \\ 2\Delta t\theta\omega M & A & 0 & \Delta t B_2 \\ 0 & 0 & A & \Delta t B_3 \\ B_1^\top & B_2^\top & B_3^\top & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{rhs}_1^{n+1} \\ \boldsymbol{rhs}_2^{n+1} \\ \boldsymbol{rhs}_3^{n+1} \\ \boldsymbol{0} \end{pmatrix} \tag{2.5}$$

with $A = M + \theta\Delta t(N(\boldsymbol{u}) + \nu L)$. For the sake of simplicity we set $B^\top = (B_1^\top \; B_2^\top \; B_3^\top)$, $\boldsymbol{rhs}^\top = (\boldsymbol{rhs}_1^{n+1^\top} \; \boldsymbol{rhs}_2^{n+1^\top} \; \boldsymbol{rhs}_3^{n+1^\top})$ and

$$S := \begin{pmatrix} A & -2\Delta t\theta\omega M & 0 \\ 2\Delta t\theta\omega M & A & 0 \\ 0 & 0 & A \end{pmatrix} \tag{2.6}$$

and hence we obtain the saddlepoint problem

$$\begin{pmatrix} S & B \\ B^\top & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{rhs} \\ \boldsymbol{0} \end{pmatrix}. \tag{2.7}$$

Now the well-known Discrete Projection Method (DPM) [17] can be applied to equation (2.7).

2.1. *Modified Discrete Projection Method*

The DPM consists of 3 main steps for each time step. Each linear system is solved by a geometric multigrid solver [18]. We will give an outline of the algorithm and give remarks concerning modifications due to Coriolis forces. In the first step of the DPM the nonlinear Burgers equation is solved using several steps of fixpoint iterations. The inclusion of Coriolis forces in $S(\boldsymbol{u}^n)$ does not call for modifications in the solver. An investigation of different preconditioners can be found in [16]. The second is the main step to ensure efficiency for DPM with Coriolis forces. The inverse matrix S^{-1} is usually dense and its computation is too expensive. We choose some variant of the mass matrix denoted by $M_{(.)}$ as an approximation of S^{-1}. The use of the lumped mass matrix M_l would be an appropriate choice, since its inverse is easy to calculate. Our goal is to construct a modified matrix that is as easy to invert but also includes the screw-symmetric part of (2.6). The straightforward option is to include the full screw symmetric part, so we have

Algorithm 2.1 Discrete Projection Method

1. Solve the viscous Burgers equation for the intermediate solution $\bar{\boldsymbol{u}}$

$$S(\boldsymbol{u}^n)\bar{\boldsymbol{u}} = \boldsymbol{rhs} - \theta\Delta t B\boldsymbol{p}^n$$

2. Solve the Pressure Poisson problem for the intermediate solution $\boldsymbol{q}$

$$B^\top S^{-1} B\boldsymbol{q} = \frac{1}{\Delta t} B^\top \bar{\boldsymbol{u}}$$

3. Correction of the pressure and velocity

$$\begin{aligned} \boldsymbol{p}^{n+1} &= \boldsymbol{p}^n + \boldsymbol{q} + \alpha M_p^{-1} B^\top \bar{\boldsymbol{u}} \\ \boldsymbol{u}^{n+1} &= \bar{\boldsymbol{u}} - \Delta t S^{-1} B\boldsymbol{q} \end{aligned}$$

$$M_{coriolis} := \begin{pmatrix} M_l & -2\omega\Delta t\theta M_l & 0 \\ 2\omega\Delta t\theta M_l & M_l & 0 \\ 0 & 0 & M_l \end{pmatrix} \tag{2.8}$$

and its inverse is given by

$$M_{coriolis}^{-1} := \frac{1}{(1 + [2\omega\Delta t\theta]^2)} \begin{pmatrix} M_l^{-1} & 2\omega\Delta t\theta M_l^{-1} & 0 \\ -2\omega\Delta t\theta M_l^{-1} & M_l^{-1} & 0 \\ 0 & 0 & (1 + [2\omega\Delta t\theta]^2)M_l^{-1} \end{pmatrix}. \tag{2.9}$$

The matrix $M_{coriolis}^{-1}$ has a block-diagonal structure and can be explicitly calculated. It has to be noted that $M_{coriolis}^{-1}$ is screw-symmetric and thus $B^\top M_{coriolis}^{-1} B$ is in general not symmetric and special care has to be taken in the choice of preconditioners and coarse grid solvers. In [15] Sokolov et al. provided a proof that for the discretization with a nonconforming Stokes element $\tilde{Q}_1 Q_0$, the matrix $B^\top M_{coriolis}^{-1} B$ regains symmetry. In order to obtain a symmetric Pressure Poisson matrix we consider

$$M_{ysm}^{-1} := \frac{1}{2}\left(M_{coriolis}^{-1} + M_{coriolis}^{-\top}\right) = \frac{1}{(1+[2\omega\Delta t\theta]^2)} \begin{pmatrix} M_l^{-1} & 0 & 0 \\ 0 & M_l^{-1} & 0 \\ 0 & 0 & (1+[2\omega\Delta t\theta]^2)M_l^{-1} \end{pmatrix} \tag{2.10}$$

the symmetric part of $M_{coriolis}^{-1}$, but first let us have a closer look at $B^\top M_{coriolis}^{-1} B$ and its continuous counterpart. Defining $\mathscr{M}_{coriolis} := [(\theta\Delta t)^{-1} I + \boldsymbol{\omega}\times]$ with $\boldsymbol{\omega} = (0,0,\omega)^\top$ we get the continuous counterpart of the "Pressure Poisson" matrix as $\nabla \cdot \mathscr{M}_{coriolis}^{-1} \nabla$ with the analytic inverse

$$\mathscr{M}_{coriolis}^{-1} = -\theta\Delta t(1+|\tilde{\boldsymbol{\omega}}|^2)^{-1}\left[I + \tilde{\boldsymbol{\omega}} \otimes \tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\omega}}\times\right], \quad \text{with } \tilde{\boldsymbol{\omega}} = \theta\Delta t\boldsymbol{\omega}.$$

Now note that for a scalar function q it holds $\tilde{\boldsymbol{\omega}} \times \nabla q = \nabla \times q\tilde{\boldsymbol{\omega}}$ and $\nabla \cdot (\nabla\times) \equiv 0$ and thus the screw-symmetric operator $-\tilde{\boldsymbol{\omega}} \times \cdot$ can be omitted and we have

$$\mathscr{M}_{coriolis}^{-1} = -\theta\Delta t(1+|\tilde{\boldsymbol{\omega}}|^2)^{-1}\left[I + \tilde{\boldsymbol{\omega}} \otimes \tilde{\boldsymbol{\omega}}\right]. \tag{2.11}$$

In consequence the continuous counterpart to $B^\top M_{coriolis}^{-1} B$ is in fact symmetric which is not recovered by our choice of the Finite Element space $Q_2 P_1$. However (2.11) is the continuous equivalent operator to the symmetric inverse matrix (2.10). This equivalence justifies the choice if M_{sym}^{-1} as an appropriate preconditioner for the Pressure Poisson problem including Coriolis forces. Finally note that both $M_{coriolis}^{-1}$ and M_{sym}^{-1} converge to the lumped mass matrix as Δt approaches zero.

2.2. *Stabilization via multiscale artificial diffusion techniques*

Looking back at the drilling application, not only high angular velocities due to the rotational movement of the drill have to be taken into account, but also high velocities compared to the length scale of the drilling hole caused by the inflow of the coolant. Consequently large Reynolds numbers (Re) are to be expected and numerical inaccuracies resulting in physics-violating oscillations in the solution may arise, that can even lead to divergence of the iterative solver. To prevent unwanted oscillations we propose a new multiscale artificial diffusion stabilization following the ideas of [8]. In order to counteract the major contribution of the convection dominated flow, artificial diffusion is added to the momentum equation. Let us consider the weak formulation of (2.4)

$$\begin{aligned} \tfrac{1}{\Delta t}\int_{\Omega_f} \boldsymbol{\varphi}_h \boldsymbol{u}_h^{n+1}\,\mathrm{dx} + \nu\int_{\Omega_f} \nabla\boldsymbol{u}_h^{n+1}\nabla\boldsymbol{\varphi}_h\,\mathrm{dx} + \int_{\Omega_f} \boldsymbol{\varphi}_h\cdot\boldsymbol{u}_h^{n+1}\cdot\nabla\boldsymbol{u}_h^{n+1}\,\mathrm{dx} - \int_{\Omega_f}\nabla\cdot\boldsymbol{\varphi}_h P^{n+1}\,\mathrm{dx} &= 0 \\ \int_{\Omega_f}\nabla\cdot\boldsymbol{u}_h^{n+1} q_h\,\mathrm{dx} &= 0 \end{aligned} \quad \text{in } \Omega_f \times (0,T), \tag{2.12}$$

with some suitable testfunctions $\boldsymbol{\varphi}_h$ and q_h. For the sake of simplicity we omit boundary conditions and Coriolis forces in equation (2.12) and set $\theta = 1$ and $\boldsymbol{f} = 0$. Now we add an additional diffusive term to equation (2.12) with a stabilization parameter $\epsilon \geq 0$, which has to be compensated in a reasonable manner and we have

$$\begin{aligned} \tfrac{1}{\Delta t}\int_{\Omega_f} \boldsymbol{\varphi}_h \boldsymbol{u}_h^{n+1}\,\mathrm{dx} + \nu\int_{\Omega_f} \nabla\boldsymbol{u}_h^{n+1}\nabla\boldsymbol{\varphi}_h\,\mathrm{dx} + \int_{\Omega_f} \boldsymbol{\varphi}_h\cdot\boldsymbol{u}_h^{n+1}\cdot\nabla\boldsymbol{u}_h^{n+1}\,\mathrm{dx} - \int_{\Omega_f}\nabla\cdot\boldsymbol{\varphi}_h P^{n+1}\,\mathrm{dx} & \\ + \int_{\Omega_f}\epsilon\nabla\boldsymbol{u}_h^{n+1}\nabla\boldsymbol{\varphi}_h\mathrm{dx} + \int_{\Omega_f}\epsilon\boldsymbol{g}\nabla\boldsymbol{\varphi}_h\mathrm{dx} &= 0 \quad \text{in } \Omega_f \times (0,T), \\ \int_{\Omega_f}(\nabla\boldsymbol{u}_h^{n+1} - \boldsymbol{g})\boldsymbol{\varphi}_H\mathrm{dx} &= 0 \end{aligned} \tag{2.13}$$

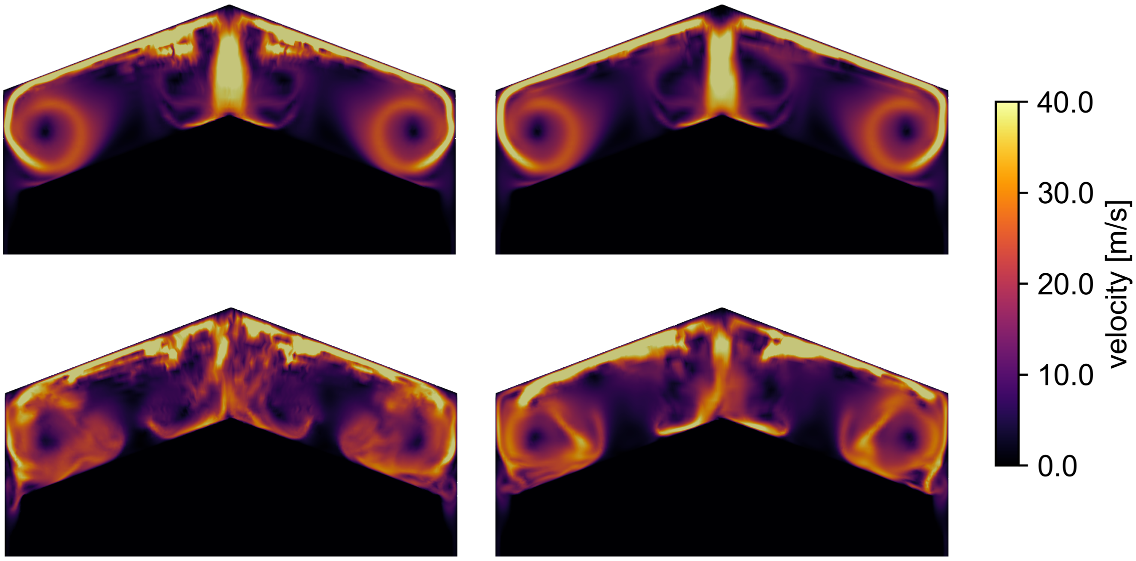
with the incompressibility condition $\int_{\Omega_f}\nabla\cdot\boldsymbol{u}_h^{n+1} q_h\,\mathrm{dx} = 0$ and $\boldsymbol{\varphi}_H \in Q_2$ a higher order testfunction. Note that $\boldsymbol{g}$ is the L^2-projection of $\nabla\boldsymbol{u}_h$ into a function space of higher order and thus the added diffusive term $\int_{\Omega_f}\epsilon(\nabla\boldsymbol{u}_h^{n+1} - \boldsymbol{g})\nabla\boldsymbol{\varphi}_h\mathrm{dx}$ can be interpreted as a local measure for the smoothness of $\nabla\boldsymbol{u}_h$ and so stabilization is only added in region that suffer from non smooth velocity gradients, e.g. regions with oscillations. As an extension to the proposed stabilization technique we define the diffusion parameter ϵ as elementwise parameter such that with a global parameter $\epsilon_{global} \geq 0$ we have

$$\epsilon_T = \epsilon_{global} h_T \|\boldsymbol{u}_h\|_{\infty,T}$$

with the local mesh size h_T of an element T. In this way the local diffusion is dependent of the local velocity and in this way the whole range of velocities is stabilized somewhat equally. Looking again at equation (2.13), the L^2-projection of $\nabla\boldsymbol{u}_h^{n+1}$ is coupled in an implicit way, which can be a drawback solving the stabilized equation. If we assume that the timestep Δt is sufficiently small (which is a viable assumption in context of the drilling application) we can take $\nabla\boldsymbol{u}_h^n$ as an suitable approximation of $\nabla\boldsymbol{u}_h^{n+1}$ and in this manner $\int_{\Omega_f}\epsilon\boldsymbol{g}\nabla\boldsymbol{\varphi}_h\mathrm{dx}$ can explicitly added to the right hand side. In figure 2.1 the effect of the stabilization technique is depicted for $\epsilon_{global} = 0.1$.

3. Geometrical modeling of the drilling process

The transformation of the Navier-Stokes equation into a rotational frame of reference, as described in the previous section 2, allows us to model the drill as a static obstacle inside the fluid domain, and only to prescribe rotational boundary conditions on the cylindrical walls of the drilling hole and on the cone-like shaped drilling ground. As a consequence only one mesh has to be constructed for all time steps, eliminating the need for cumbersome remeshing or surface tracking techniques to account for the high rotational domain. Although the transformation enables us to construct only one computational mesh, the construction for the fluid and solid domain is no trivial task. The geometry of the drill is complex and consists of elements of different length



(a) No stabilization

(b) Stabilization with $\epsilon_{global} = 0.1$

Fig. 2.1. Comparison of different timesteps (top and bottom) with and without stabilization

scales, i.e. the coolant channels and the chip flutes. Constructing a case specific mesh, where the mesh nodes are aligned with the fluid solid interface, is to our knowledge not possible (at least for hexahedral grids) or suffers from the lack of flexibility to study the variation of process parameters, like the distance of the tool to the drilling ground, or tool parameters like the shape of the coolant channels. Our strategy to model the geometry of the drill is twofold: 1) The Fictitious Boundary Method (FBM) is used to assign each element an indicator, based on a surface representation of the tool (or workpiece), which decomposes the computational domain into a solid and fluid part, while still maintaining one unified mesh for the whole domain; 2) Mesh deformation techniques dependent on some monitor function $\mathfrak{M}$ in combination with mesh regularization algorithms let us concentrate mesh nodes in regions of high interest, i.e. in vicinity of the tool surface, without changing the connectivity of the grid. These two points will ensure a flexible and robust grid generation for drilling applications.

3.1. Fictitious Boundary Method

The Fictitious Boundary Method is a filtering-related technique that creates a decomposition of the computational domain into a solid and a fluid part. With an indicator function

$$\alpha(\boldsymbol{x}) := \begin{cases} 1, & \boldsymbol{x} \in \Omega_s \\ 0, & \boldsymbol{x} \in \Omega_f \end{cases} \tag{3.1}$$

for coordinates $\boldsymbol{x} \in \mathbb{R}^3$, each node in the computational grid is either assigned to be solid ($\alpha(\boldsymbol{x}) = 1$) or fluid ($\alpha(\boldsymbol{x}) = 0$). The interaction between solid and fluid is driven by internal Dirichlet boundary condition, e.g. the velocity in the solid nodes are set to zero. Since we model the drill as a static object, the calculation of α only has to be done in a preprocessing step. With a given surface triangulation, that can be derived from the exact CAD data, a signed distance function is computed to determine the inside (solid) and outside (fluid) of the given surface triangulation and thus the values for α are assigned. Note that with the definition (3.1) the solid-fluid interface is implicitly given by the derivative of α, as $\nabla\alpha \neq 0$ in vicinity of the solid boundary and zero everywhere else. This can be used to compute fluid-solid interaction like hydrodynamic forces. Details can be found in [11].

3.2. Mesh deformation

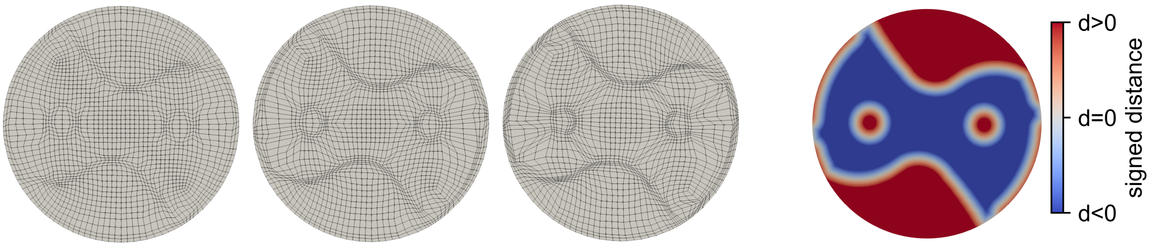
The main idea of the proposed mesh deformation technique is to determine a mesh size distribution using a monitor function $\mathfrak{M}$ and move mesh nodes according to the desired mesh size distribution without changing the connectivity. The monitor function $\mathfrak{M}$ can be constructed arbitrarily, i.e. it can take several goal quantities into account like shear rate, temperature gradients or some custom defined values. Since we aim to construct a computational grid that is highly resolved at the solid-fluid interface, the node distance to the solid surface is a straightforward goal quantity. Further the signed distance function is already given as it was needed in context of FBM to evaluate the indicator function (3.1). Now the monitor function $\mathfrak{M}$ can be used to calculate weights w_i to nodes $\boldsymbol{x}_i$. The weights determine how the mesh nodes are moved as depicted in algorithm 3.1. Algorithm 3.1 is applied iteratively to adjust the mesh successively. An exemplary deformed mesh according to a signed distance function of a drill is depicted in figure 3.1. The relaxation parameter τ is usually set to 0.25 and Laplacian smoothing procedure improves the

Algorithm 3.1 Algebraic mesh deformation

For given weights w_i, edges e_{ij} and node coordinates $\boldsymbol{x}_i$ calculate the new coordinates

```
for $i = 1,\dots,n$ do
    $\boldsymbol{x}_i^{new} = (1-\tau)\boldsymbol{x}_i^{old} + \tau \frac{\sum_{j=1,n}^{\text{if } \exists e_{ij}} w_j \boldsymbol{x}_j^{old}}{\sum_{j=1,n}^{\text{if } \exists e_{ij}} w_j}$
end do
```

with a relaxation parameter $\tau \in (0,1]$.



(a) Successive mesh deformation

(b) Signed distance to tool drill surface

Fig. 3.1. Mesh deformation based on calculated signed distance function

mesh quality. It has to be noted that the properties of the resulting mesh have (yet) to be proven from a mathematical standpoint, but several examples like simulations of co-rotating twin-screw extruders show the robustness and validity of algorithm 3.1 [10]. Mesh deformation is performed purely algebraic, unlike PDE based counterparts, and thus it is computationally cheap and still flexible in regards to complex geometries. Additionally, the use of computational grids generated by algorithm 3.1 in combination with FBM does not deteriorate the convergence behavior and accuracy of the Navier-Stokes solver as shown in [2, 11]. So mesh deformation and FBM enables us to accurately simulate fluid flow around complex geometries like drilling tools.

4. Mathematical modeling of heat transfer

The previous sections focused on the efficient simulation of coolant flow around drills. While the flow field of the lubricant is an important factor in drilling applications, the main quantity of interest is the temperature of the tool, especially heating due to friction and dissipation in the shear band and how the fluid can transport the heat away from the contact zone to cool the drill down. High temperatures at the cutting edge are a driving factor for tool wear and since measurements of the temperatures are tedious (or nearly impossible), accurate modeling of heat generation and transfer is the key. Here we want to give a brief description of the underlying heat equation and its discretization. We consider the heat equation

$$\partial_t(c_p\rho\Theta) - \nabla\cdot(\lambda\nabla\Theta) + c_p\rho\boldsymbol{u}\cdot\nabla\Theta = h \quad \text{in } \Omega\times(0,T) \tag{4.1}$$

with temperature Θ, velocity $\boldsymbol{u}$, density ρ, heat capacity c_p, heat conductivity λ and volumetric heat source h. In this work we only consider Dirichlet boundary conditions and we omit frictional heat generation. We discretize equation (4.1) in space with Q_1 Finite Elements and with implicit Euler in time and get the discrete heat equation

$$c_p\rho\frac{\Theta_h^{n+1}-\Theta_h^n}{\Delta t} - \nabla\cdot(\lambda\nabla\Theta_h^{n+1}) + c_p\rho\boldsymbol{u}_h\cdot\nabla\Theta_h^{n+1} = h^{n+1} \quad \text{in } \Omega_h\times(0,T). \tag{4.2}$$

The heat equation (4.2) is solved in both the solid and fluid part at the same time by taking advantage of the meshing techniques described in section 3. The mesh is decomposed into a solid and fluid domain using the indicator function (3.1) acting as filter for internal boundary conditions for the velocity in equation (4.2). The velocity is set to zero if $\alpha = 1$ (solid) and thus the convective term vanishes. In consequence we make use of one mesh for the heat equation, the distinction between the different parts of the domain is implicitly managed by FBM. Further, the material parameters for fluid and solid can be automatically assigned according to values of α and the thermal fluid-solid interaction is implicitly governed by different material properties. Temperature distribution in the solid is solely diffusion driven. Heat transport in the fluid is mainly due to convection, which can lead to numerical difficulties like oscillating solution. In order to deal the convection dominated heat transport, an algebraic Flux Correcting is incorporated, details can be found in [7].

5. Modeling of the Drilling process

The previous sections established a toolbox of numerical techniques and algorithms that will now be plugged together to develop an robust simulation model for drilling application. In this context we will give a short summary of the proposed methods and how they are employed. First, from a given surface triangulation (constructed from exact CAD data) of a tool the distance function is evaluated to adapt the computational grid to the fluid-solid interface. The FBM indicator is set accordingly and material properties are assigned to the corresponding elements as described in section 3. From there on a multigrid mesh hierarchy is constructed and matrices for the rotational Navier-Stokes equation (2.4) are assembled. The resulting algebraic equations are solved using DPM with a geometric multigrid solver. Rotational boundary conditions are set at the drilling ground and wall and parabolic inflow is prescribed for the coolant channels. Possible numerical inaccuracies that may arise due to high Reynolds numbers are counteracted by the proposed two scale artificial diffusion technique as explained in section 2. Choice of an appropriate diffusion parameter is a delicate task since it should be large enough to ensure a nonoscillatory solution but not too large so the velocity field is not ”smeared out”. Numerical tests show that $\epsilon_{global} = 0.1$ is viable a choice for our applications, however the influence of different ϵ_{global} values should be investigated in future work. Finally the temperature simulation makes use of FBM. Having a unified mesh for both solid and fluid enables us to solve the heat equation in the drill and coolant simultaneously. The model does not distinguish between fluid and solid domain, instead only assigned material properties and the velocity filter corresponding to α differ, as mentioned in section 4. Needless to say, the timestep size is limited by the convective term in the fluid domain.

5.1. *Algorithmic details of Drilling simulation*

Since the order of velocities (∼20 m/s) is significantly greater than the spatial dimension of the drilling hole (diameter 8 mm), small timesteps are necessary. In consequence fully coupled fluid flow and temperature would be highly inefficient. Alternatively we do not consider the full rotational movement of the drill, and instead we solve the Navier-Stokes equation and the heat equation in a decoupled way. Figure 5.1 illustrates the steps of the simulation model for drilling applications.

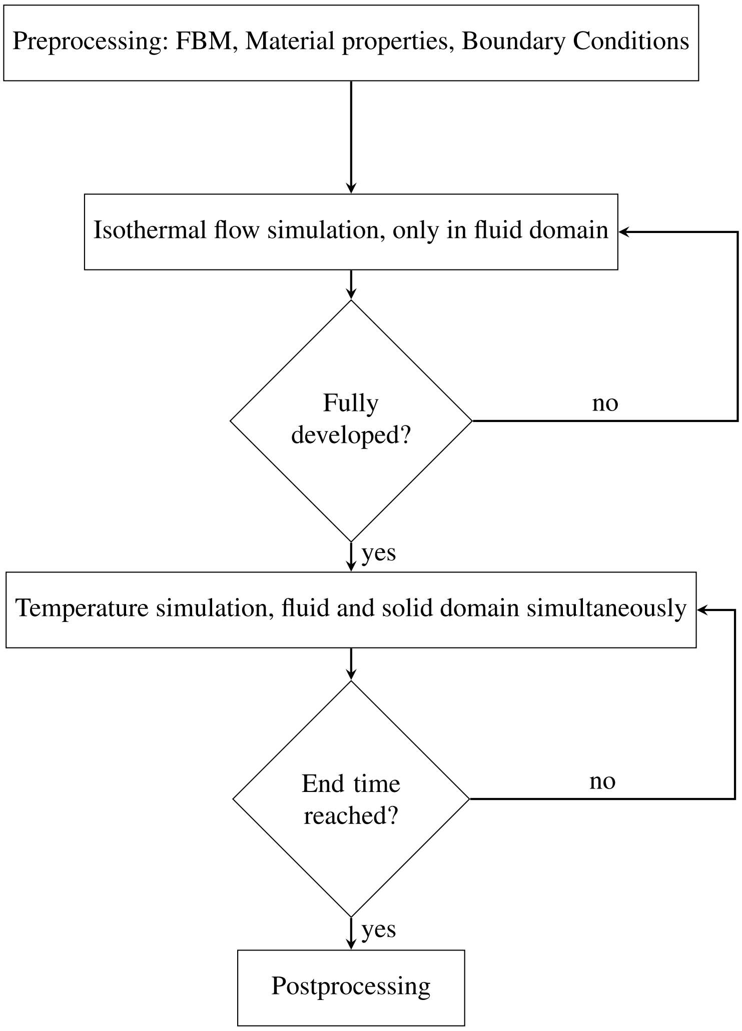

Preprocessing: FBM, Material properties, Boundary Conditions
Isothermal flow simulation, only in fluid domain
Fully developed?
no
yes
Temperature simulation, fluid and solid domain simultaneously
End time reached?
no
yes
Postprocessing

Fig. 5.1. Drilling application simulation flow chart

The central idea to simulate the cooling process of drilling applications due to coolant flow is to calculate the fluid flow and the heat distribution sequentially. First, simulation of the coolant flow is carried until the velocity field is fully developed.

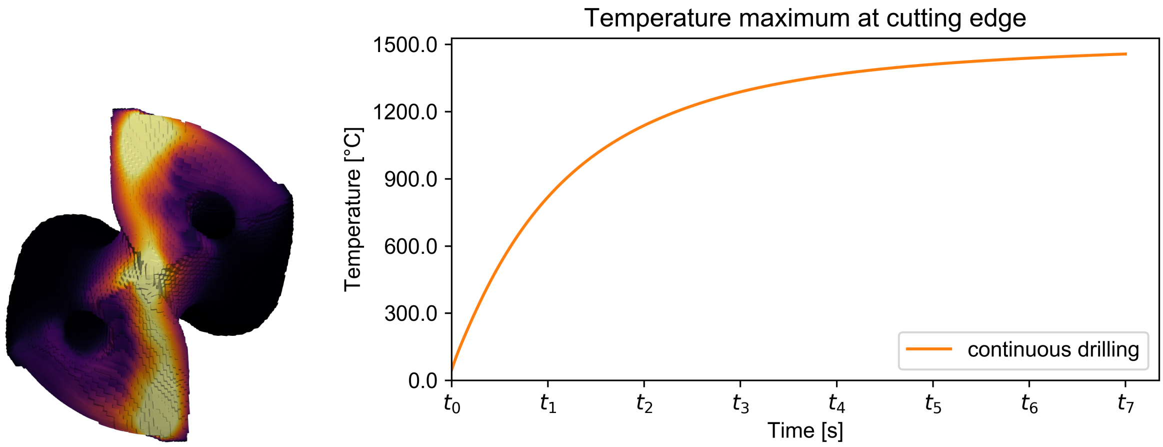
Assuming the velocities to be statistically in nature, taking the velocity field of a fully developed flow field as an representation for the nonstationary flow field is a viable simplification. Time averaging techniques or looping quasi-periodically over a set of flow fields are considered for future work. The velocity field is used as input data for the temperature simulation in the second step. The timestep size is increased in order to increase the efficiency of the simulation. By construction of the simulation model and exploiting FBM, the same grid is used for the heat equation. We omit chip formation and heat generation due to frictional contact for now and internal heat sources are set conceptional in the elements in vicinity of the cutting edge that geometrically penetrate the drilling ground.

5.1.1. Numerical Test

In order to test the algorithmic model as described in figure 5.4 conceptually, we will use parameters found in the actual experimental setup. We consider a lubrication oil, carbide tool (90% WC, 10 %Co) and workpiece made of Inconel 718. Process and material parameters are summarized in table 5.1. The heat source h is taken as a fictional value, and calibration with experiments or chip formation simulations has to be done in future works. A coolant flow simulation with $\Delta t = 5 \cdot 10^{-6}$ is carried out until a fully developed flow is reached. Thereafter temperature is simulated using the previously calculated velocities with a time step size $\Delta t = 10^{-4}$ s until $T = 3.6 \cdot 10^{-2}$ s is reached. The resulting temperature at the cutting edge at the end of the simulation and the maximum temperature at the cutting edge can be seen in figure 5.2. As seen in figure 5.2a the temperature increases gradually, resembling a saturation function, until a temperature of ∼1400 °C is reached at $T = 3.6 \cdot 10^{-2}$ s. Time stamps t_i with $i = 0, \dots, 7$, $t_{i+1} - t_i = 6 \cdot 10^{-3}$ s and $t_7 = T$ denote certain times of interest that will be of importance in the following section.

Processparameters	Coolant	Workpiece	Tool
$\dot{Q}: 3.3\ l/min$ $U: 995\ min^{-1}$ $h: 70\ kW$	$\rho: 0.85\ g/cm^3$ $c_p: 2.02\ J/(g \cdot K)$ $\lambda: 0.00139\ W/(cm \cdot K)$ $\nu: 0.02\ Pa \cdot s$	$\rho: 8.19\ g/cm^3$ $c_p: 0.45\ J/(g \cdot K)$ $\lambda: 0.095\ W/(cm \cdot K)$	$\rho: 14.45\ g/cm^3$ $c_p: 0.2\ J/(g \cdot K)$ $\lambda: 0.86\ W/(cm \cdot K)$

Table 5.1. Process- and materialparameters of coolant, Inconel718 and tool



(a) Temperature at cutting edge at last timestep

(b) Maximum Temperature over time

Fig. 5.2. Conceptional simulation of heating process with coolant flow

5.2. *Outlook Discontinuous Drilling*

With the established simulation model for drilling applications, we want take the proposed algorithmic toolbox further. Let us consider the newly developed discontinuous drilling strategy for hard to machine alloys, like Inconel 718, as proposed in [19]. Continuously drilling until a certain depth is reached results in high temperature and thus high tool wear. The main idea of discontinuous drilling strategy is to interrupt the process by moving the rotating drill away from the contact zone in order to reduce the temperature of the cutting edge. We want to construct a simulation model of the temperature during discontinuous drilling, considering both the 'contact' and 'no contact' phases in one simulation, extending the algorithmic ideas of section 5.1. In a preprocessing step we simulate the flow field of the 'contact' and 'no contact' phase, in order to obtain a velocity field for both cases that will be used for the corresponding temperature simulations. FBM and mesh deformation are crucial for the efficiency of the developed model, since one single mesh is constructed for both configurations ('contact' and 'no contact') simply by switching the corresponding α values assigned to the workpiece. Hence no projection schemes are needed to transfer temperature values to another mesh. Figure 5.3 illustrates this idea: First a mesh is constructed for the 'contact' setup and elements belonging to the drilling ground are marked as workpiece elements (red in figure 5.3). Since we utilize a rotational frame of reference, rotational boundary condition is set in those elements. Switching from one configuration to the other, only the temperature values are carried over, and velocities and the indicator α from preprocessed fluid flow simulations are used as input values. Configuration 'contact' and 'no contact' are alternated until an end time T is reached. The modelling idea is summarized in figure 5.4.

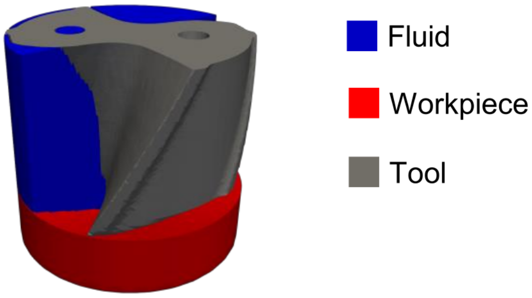


Fig. 5.3. FBM decomposition of the domain into fluid, workpiece and tool elements

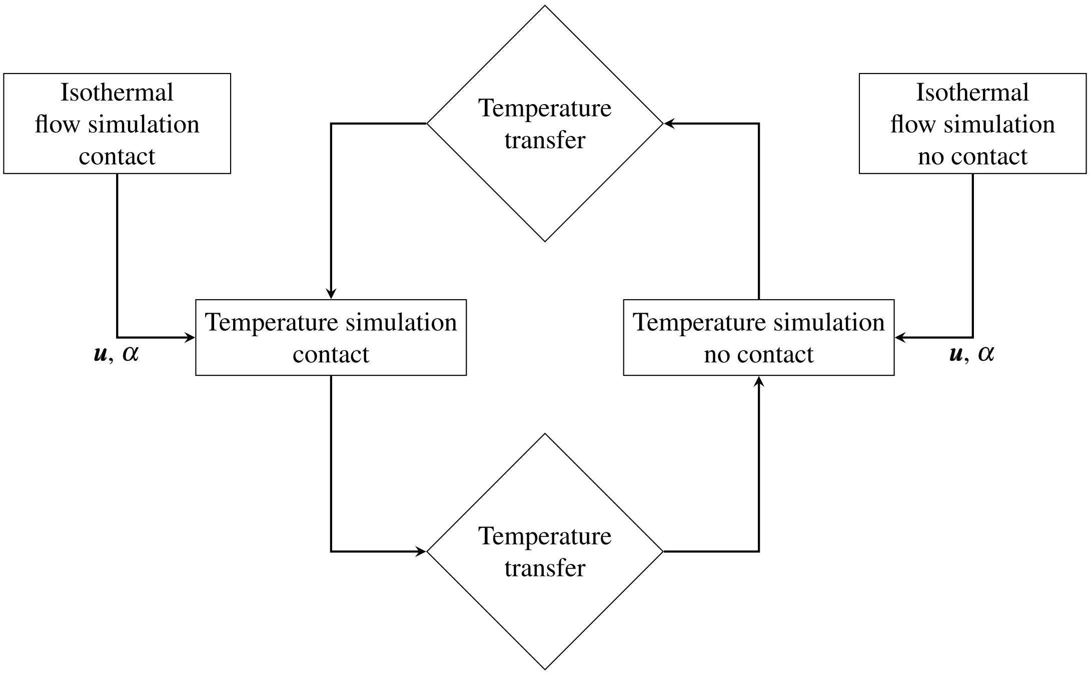
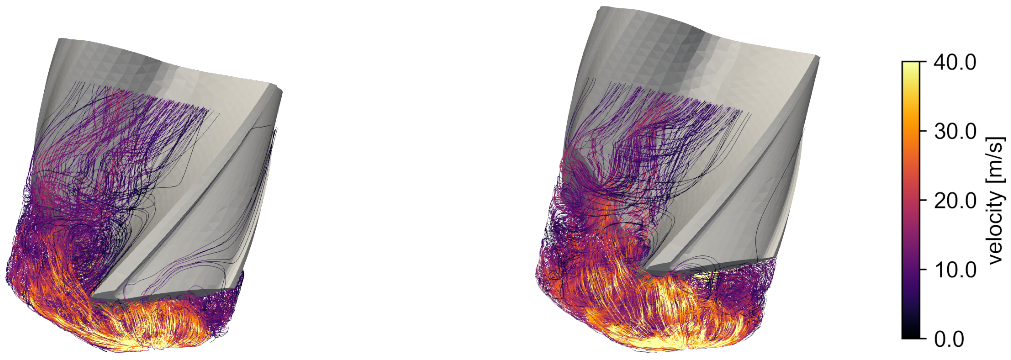

Isothermal flow simulation contact
Temperature transfer
Isothermal flow simulation no contact
$\boldsymbol{u}$, α
Temperature simulation contact
Temperature simulation no contact
$\boldsymbol{u}$, α
Temperature transfer

Fig. 5.4. Discontinuous drilling application simulation flow chart

5.2.1. Numerical Tests

Before simulations of the full discontinuous drilling setup are considered, the question arises, which distance during the 'no contact' phase is more beneficial for the cooling process. Therefore we examine two different distances, 1 *mm* and 2 *mm*, with same process and material parameters as listed in table 5.1. A visualization of streamlines and velocities can be found in figure 5.5. The flow fields can further be used for the full discontinuous drilling configuration. In order to study the properties of the simulated flow fields, we analyze how fast the volume of the fluid is exchanged, rinsing the possible heated lubricant. So residence times of the streamlines are calculated and the distribution of cumulative residence times are evaluated, as seen in figure 5.6. The distance of 1 *mm* seems to have better properties to rinse hot coolant out, as the volume exchange is faster in



(a) Distance 1 *mm*

(b) Distance 2 *mm*

Fig. 5.5. Velocity magnitude of streamlines for different distances between drill and drilling ground

comparison to 2 *mm* distance of the tool to the drilling ground. To exchange 60% of the fluid volume, with a distance of 1 *mm* it takes 0.59 *s* and with a distance of 2 *mm* it takes 0.73 *s*. To put the difference of 0.14 *s* into perspective, converting this implies that the difference is more than two full rotations of the drill. With this interpretation, a smaller distance is more favorable for heat transport during the 'no contact' phase. A detailed analysis of the influence of the distance in context of discontinuous drilling strategy is planned for future works. Consequently, we will use a distance 1 *mm* for further simulations of the full discontinuous setup. Same parameters as in section 5.1 are incorporated except that we alternate between 'contact' and 'no contact' configuration at times t_i, $i = 0,\dots,7$ with $t_{i+1} - t_i = 6 \cdot 10^{-3}$ according to figure 5.4. The comparison of maximum temperature at the cutting edge over time to the temperature resulting from the continuous drilling strategy is depicted in figure 5.7. As expected, we observe that the temperature declines during the 'no contact' phases, leading to significant smaller temperature overall compared to continuous drilling. The cooling and heating process for last simulated interval respectively is portrayed in figure 5.8

6. Summary and Outlook

In this work we presented numerical components and algorithmic details for a simulation model of a novel drilling strategy with coolant, the discontinuous drilling approach. Modification of the underlying partial differential equations were made and discretization and the solver routines were adapted. Further, a flexible and robust meshing technique was introduced that, for given geometries, works in a black-box fashion, thus (almost) arbitrary driller shapes, and also possibly workpiece shapes, can be studied. Combining the introduced numerical algorithms, a temperature simulation for discontinuous drilling application was conducted. Extensions concerning the presence of uncut chips and subsequently approximations of realistic heat sources due to friction or deformation are planned. Finally, the developed simulation model is a substantial tool in order to analyze the novel discontinuous drilling approach without costly experiments and giving insight where no measurements are possible.

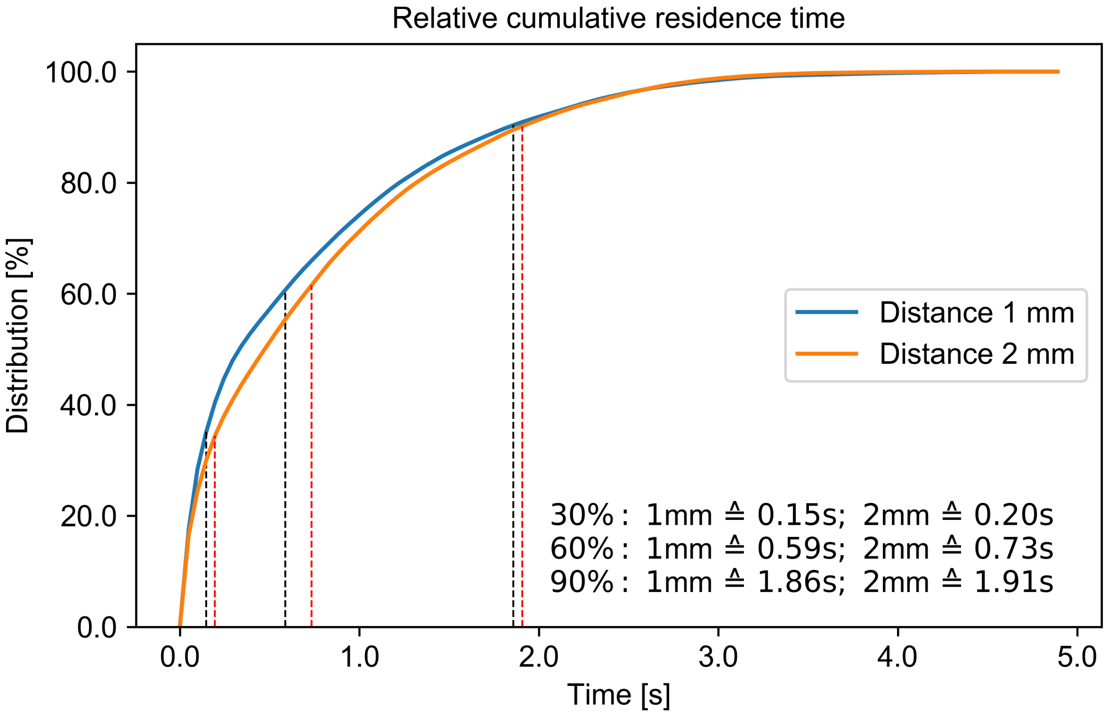


Fig. 5.6. Cumulative residence time for different distances between drill and drilling ground

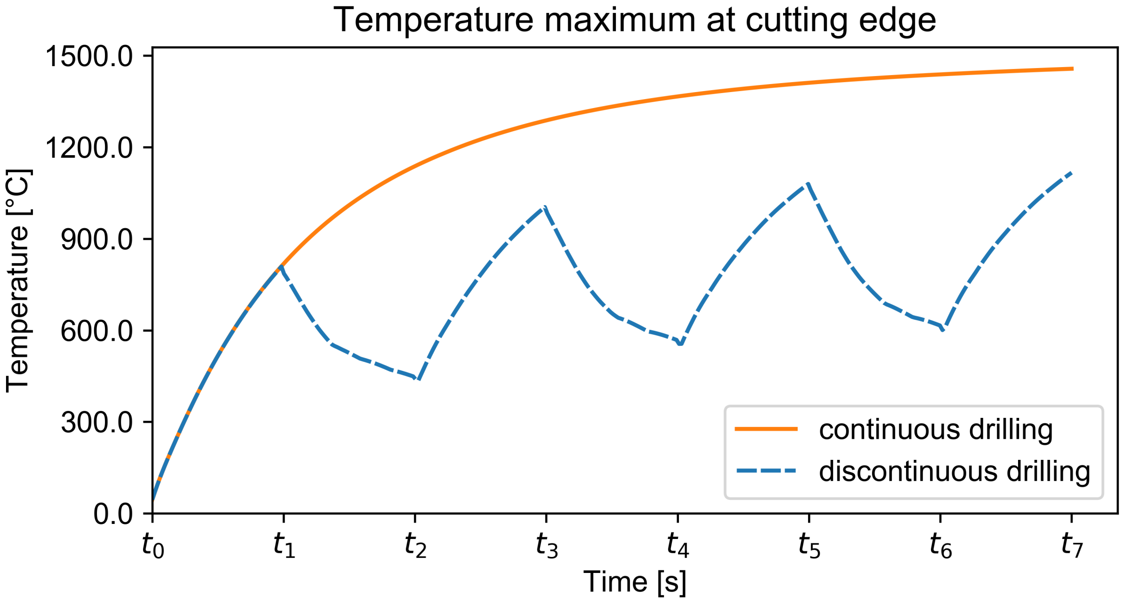
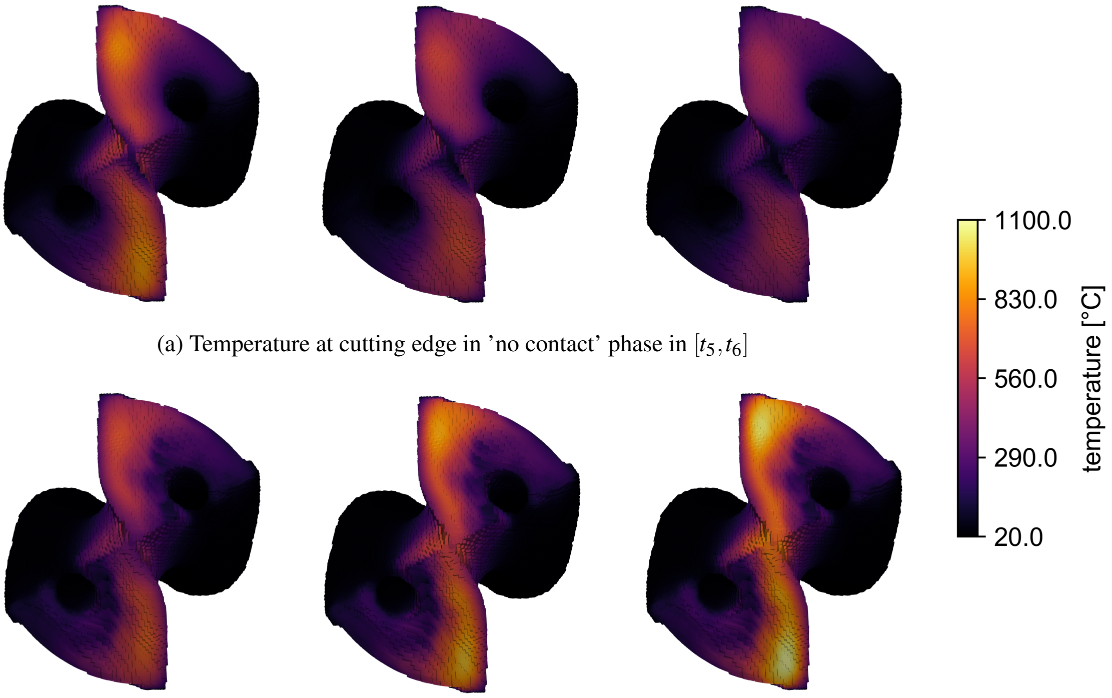


Fig. 5.7. Maximum temperature over time, comparison of continuous and discontinuous drilling strategy



(a) Temperature at cutting edge in ’no contact’ phase in $[t_5, t_6]$

(b) Temperature at cutting edge in ’contact’ phase in $[t_6, t_7]$

Fig. 5.8. Comparison of temperature distribution for ’no contact’ and ’contact’ phase

Acknowledgments

This work was supported by the German Research Association (DFG) within the project 'Simulation and optimization of the coolant flow to reduce the thermal tool load during discontinuous drilling of Inconel 718' (grant TU 102/75-1). The computations have been carried out on the LiDO cluster at TU Dortmund University. We would like to thank the LiDO cluster team for their help and support.

References

[1] Featflow. http://www.mathematik.tu-dortmund.de/~featflow/en/. Accessed: 2022-03-19.

[2] E. Bayraktar, O. Mierka, and S. Turek. Benchmark computations of 3d laminar flow around a cylinder with cfx, openfoam and featflow. *International Journal of Computational Science and Engineering*, 7(3):253–266, 2012.

[3] D. Biermann, H. Blum, J. Frohne, I. Iovkov, A. Rademacher, and K. Rosin. Simulation of mql deep hole drilling for predicting thermally induced workpiece deformations. *Procedia CIRP*, 31:148–153, 2015.

[4] D. Boffi and L. Gastaldi. On the quadrilateral q2–p1 element for the stokes problem. *International journal for numerical methods in fluids*, 39(11):1001–1011, 2002.

[5] H. Damanik, J. Hron, A. Ouazzi, and S. Turek. Monolithic newton-multigrid solution techniques for incompressible nonlinear flow models. *International Journal for Numerical Methods in Fluids*, 71(2):208–222, 2013.

[6] A. Johns, R. Hewson, E. Merson, J. Summers, and H. Thompson. Internal twist drill coolant channel modelling using computational fluid dynamics. In *Proceedings, vol. II*, pages 1114–1122. International Center for Numerical Methods in Engineering (CIMNE), 2014.

[7] D. Kuzmin. Algebraic flux correction i. In *Flux-corrected transport*, pages 145–192. Springer, 2012.

[8] C. Lohmann, D. Kuzmin, J. N. Shadid, and S. Mabuza. Flux-corrected transport algorithms for continuous galerkin methods based on high order bernstein finite elements. *Journal of Computational Physics*, 344:151–186, 2017.

[9] O. Mierka, M. Munir, C. Spille, J. Timmermann, M. Schlüter, and S. Turek. Reactive liquid-flow simulation of micromixers based on grid deformation techniques. *Chemical Engineering & Technology*, 40(8):1408–1417, 2017.

[10] O. Mierka, T. Theis, T. Herken, S. Turek, V. Schöppner, and F. Platte. Mesh deformation based finite element – fictitious boundary method (fem–fbm) for the simulation of twin–screw extruders. Technical report, Fakultät für Mathematik, TU Dortmund, Dec. 2014. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 509.

[11] R. Münster, O. Mierka, and S. Turek. Finite element-fictitious boundary methods (fem-fbm) for 3d particulate flow. *International Journal for Numerical Methods in Fluids*, 69(2):294–313, 2012.

[12] E. Oezkaya, N. Beer, and D. Biermann. Experimental studies and cfd simulation of the internal cooling conditions when drilling inconel 718. *International Journal of Machine Tools and Manufacture*, 108:52–65, 2016.

[13] T. Qiu, T. Lee, A. Mark, K. Morozov, R. Münster, O. Mierka, S. Turek, A. Leshansky, and P. Fischer. Swimming by reciprocal motion at low Reynolds number. *Nature Communications*, 2014. online: doi http://www.nature.com/ncomms/2014/141104/ncomms6119/full/ncomms6119.html.

[14] P. Sarmadi, O. Mierka, S. Turek, S. Hormozi, and I. Frigaard. Three dimensional simulation of flow development of triple–layer lubricated pipeline transport. *Journal of Non-Newtonian Fluid Mechanics*, (274), 2019.

[15] A. Sokolov. *Analysis and numerical realisation of discrete projection methods for rotation incompressible flows*. PhD thesis, Technische Universität Dortmund, Nov. 2008.

[16] A. Sokolov, S. Turek, and M. A. Olshanskii. Numerical study of a discrete projection method for rotating incompressible flows. *Electronic Transactions on Numerical Analysis*, 32:49–62, 2008.

[17] S. Turek. On discrete projection methods for the incompressible navier-stokes equations: An algorithmical approach. *Computer Methods in Applied Mechanics and Engineering*, 143(3-4):271–288, 1997.

[18] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approache*, volume 6. Springer Science & Business Media, 1999.

[19] T. Wolf, I. Iovkov, and D. Biermann. Influence of a discontinuous process strategy on microstructure and microhardness in drilling inconel 718. *Journal of Manufacturing and Materials Processing*, 5(2):43, 2021.