Computational Particle Mechanics
https://doi.org/10.1007/s40571-022-00465-x

# Quantitative evaluation of SPH in TIG spot welding

**Stefan Rhys Jeske**[1] · **Marek Sebastian Simon**[2] · **Oleksii Semenov**[3] · **Jan Kruska**[2] · **Oleg Mokrov**[2] · **Rahul Sharma**[2] · **Uwe Reisgen**[2] · **Jan Bender**[1]

Received: 16 August 2021 / Revised: 14 January 2022 / Accepted: 7 February 2022

## Abstract

While the application of the Smoothed Particle Hydrodynamics (SPH) method for the modeling of welding processes has become increasingly popular in recent years, little is yet known about the quantitative predictive capability of this method. We propose a novel SPH model for the simulation of the tungsten inert gas (TIG) spot welding process and conduct a thorough comparison between our SPH implementation and two finite element method (FEM)-based models. In order to be able to quantitatively compare the results of our SPH simulation method with grid-based methods, we additionally propose an improved particle to grid interpolation method based on linear least-squares with an optional hole-filling pass which accounts for missing particles. We show that SPH is able to yield excellent results, especially given the observed deviations between the investigated FEM methods and as such, we validate the accuracy of the method for an industrially relevant engineering application.

**Keywords** SPH · Electric arc welding · FEM · Particle-to-grid interpolation · Incompressible flow · Solidification

## List of symbols

| | |
|---|---|
| $\boldsymbol{a}$ | Acceleration |
| $\boldsymbol{A}_m$ | Magnetic vector potential |
| $\boldsymbol{B}$ | Magnetic induction |
| $\boldsymbol{F_L}$ | Volumetric Lorentz force |
| $f_{\text{ext}}$ | External forces |
| $\boldsymbol{J}$ | Current density |
| $\boldsymbol{v}$ | Velocity |
| $\boldsymbol{x}$ | Position |
| $\dot{q}'''$ | Volumetric heat source |
| $\lambda$ | Heat conductivity |
| $\mathcal{N}_x$ | Neighboring particles of position $\boldsymbol{x}$ |
| $\rho$ | Density |
| $\varphi$ | Scalar electric potential |
| $C$ | Morphological constant |
| $c_p$ | Specific heat capacity |
| $h$ | Specific enthalpy |
| $I_w$ | Welding current |
| $m$ | Mass |
| $p$ | Pressure |
| $T$ | Temperature |
| $t$ | Time |
| $V$ | Volume |

✉ Stefan Rhys Jeske
jeske@cs.rwth-aachen.de

Marek Sebastian Simon
marek.simon@rwth-aachen.de

Oleksii Semenov
oleksii.semenov@ukr.net

Oleg Mokrov
mokrov@isf.rwth-aachen.de

[1] Visual Computing Institute, Computer Animation, Ahornstrasse 55, 52074 Aachen, Germany

[2] Welding and Joining Institute, Pontstrasse 49, 52062 Aachen, Germany

[3] Dep. Gas Discharge Physics and Plasma Devices, E.O.Paton Electric Welding Institute of NASU, Kazemir Malevich 11, 03150 Kiev, Ukraine

## 1 Introduction

The smoothed particle hydrodynamics method (SPH) is a Lagrangian discretization method, which has been investigated in recent years for the simulation of manufacturing processes, such as arc welding processes, see [16,17,20,34]. However, until now, the method has not been quantitatively compared for the application to arc welding processes, to assess its performance regarding calculation speed and computational accuracy. Therefore, in this work, the open-source implementation of the SPH method SPlisHSPlasH [4] has

Published online: 01 April 2022

been extended to allow for the consideration of heat transfer and respective boundary conditions as well as the influence of external forces (such as the Lorentz force). In order to study the applicability of our SPH method for the simulation of arc welding processes, a test case was set up modeling a simplified spot tungsten inert gas (TIG) welding process. The test problem was solved separately by means of our SPH implementation and two different FEM models implemented in Wolfram Mathematica (WM) [31] and COMSOL Multiphysics, allowing for the simulation of weld pool formation under the axisymmetric assumption.

For this comparison, the problem was idealized by using constant material properties in the simulation and assuming a flat free surface. Typically, this kind of problem is well within the strengths of Eulerian methods, whereas the SPH method is known to have difficulties with strictly bounded domains. While other works only consider the application of SPH to the entire process, we aim to first establish the accuracy of the SPH method in this idealized and bounded scenario. This is done by rigorously comparing against known Eulerian methods, whose properties have been well-studied for many years. The electromagnetic Lorentz force, which is the main driving force for the convective heat transfer in the melt pool, was calculated in advance by the WM-Code. The resulting vector field was exported and applied to the SPH simulation as an axisymmetric external force, while it was solved within COMSOL given the same boundary conditions.

We summarize our main contributions as follows:

- The proposition of a novel SPH algorithm for the simulation of the TIG spot-welding process, which utilizes implicit time integration to significantly improve simulation stability, and enables the use of larger time steps.
- The construction and implementation of a fast projection scheme for the integration of surface effects for SPH fluids.
- A novel particle to grid interpolation method which, compared to existing methods such as Shepard interpolation, is able to generate smoother fields while maintaining very high interpolation accuracy. This is extended by a smoothing algorithm in order to fill in holes of the interpolated grid due to missing particles.
- The quantitative comparison and in depth analysis of our SPH method with the WM-FEM and the COMSOL-FEM on the simulation of TIG spot-welding.

Overall, we are able to show remarkable agreement between our SPH method and the two FEM methods, especially the solution of COMSOL, while at the same time being entirely in the margin of error exhibited by the comparison of the WM-FEM method and the COMSOL-FEM method. This result is made all the more remarkable considering that our SPH model solves the problem in full 3D, while both WM and COMSOL solve the problem in 2D with rotational symmetry.

## 1.1 Modeling of TIG spot-welding

The tungsten inert-gas (TIG) welding processes is a widely applied welding process, that, while it has a lower productivity than gas metal arc welding processes, has its advantages when it comes to high requirements on weld seam quality and optical appearance. It is especially suitable for welding root passes, thin metal sheets and materials prone to oxidation. It is therefore often applied in chemical or food processing plants.

For the present validation, the "spot-welding" variant of the process is being studied for simplicity reasons, where the welding torch position is fixed, and no filler material is supplied. In addition, we consider high-frequency TIG (HF-TIG), in which the electromagnetic Lorentz force, as a result of the concentration of the arc, causes the formation of significant flow patterns in the weld pool. For the usual process, in which the distribution of electric current on the surface of the liquid is typically less concentrated, the Lorentz force is less dominant. Therefore, only the weld pool formation caused by the heat flux from the arc and the influence of the Lorentz force is studied, which is sufficient for a first qualitative comparison with Eulerian methods.

To facilitate the comparison, the nonlinearities of the material parameters, like latent heat and temperature-dependent thermal and electrical conductivity or viscosity were neglected, instead applying constant values for the relevant material parameters, which allowed the Lorentz force to be pre-computed. Also, since the modeling of the free surface of the weld pool requires a special treatment in the Eulerian framework, like volume of fluid (VOF) or level-set, the free surface deformation and Marangoni effect were neglected, although their modeling would pose a great advantage for the SPH method. The main focus of the present work is therefore not to present an accurate modeling of the welding process, but rather to investigate the applicability of the SPH method for the conditions present in this process and to quantitatively compare its performance against the common Eulerian methods, under the same conditions.

## 1.2 SPH methods in welding simulation

The smoothed particle hydrodynamics (SPH) method was originally proposed by Lucy [23] and Gingold and Monaghan [13] in the field of Astrophysics. Since then it has been adopted for many different applications, including the simulation of weld pool dynamics. The mesh-free nature of SPH enables the simulation of large deformations and free surface motion and coupling of many physical processes, which makes it an attractive method for many real-world problems.

Das and Cleary [11] use SPH in three-dimensional arc welding simulations in order to study temperature distributions, flow patterns and plastic strain in the filler material and residual thermal stresses in the work piece.

Ito et al. [17] perform full simulations of TIG welding using the SPH method and show results for different material properties due to different sulfur contents and evaluate the flow patterns and the shape of the weld pool.

Trautmann et al. [33] similarly perform weld pool simulations using SPH for a TIG welding process. They consider buoyancy, viscosity and surface tension as flow driving forces and the arc pressure, shear and thermal effect were parameterized using experimental studies. The penetration profiles of three different welding currents were compared to experimental results and show decent agreement.

A hybrid approach is investigated by Komen et al. [20], who use a Eulerian grid and Lagrangian particles in conjunction in order to simulate gas metal arc welding (GMAW). Molten metal is simulated by means of SPH, while the arc plasma and gas are simulated on a grid. The methods are then weakly coupled and executed iteratively.

Komen et al. [19] also simulate the GMAW process under consideration of droplet formation and compare the weld pool shapes against experimental results. In order to more easily visualize results, they use an ensemble averaging in order to transfer particle data onto a regular grid.

These types of simulations are difficult to validate, as they describe very complex systems with many interacting components, and this is evident with Trautmann et al. [33] being one of the few works which attempts to validate their results using experimental data. In this paper, we chose instead to compare our proposed SPH method against Eulerian simulations, as a proof of concept which shows that SPH is able to obtain very good agreement in the resulting weld pool shapes, as well as temperature and velocity distributions. A similar study has also been conducted by Jeske et al. [18], which shows excellent performance of the SPH method when compared against the Eulerian VOF method for the simulation of droplet impacts in thermal spraying. This allows us to justify the usage of SPH to obtain physically meaningful results, also for cases that pose more difficulty for Eulerian methods, for example when considering a free surface. The performed investigations confirm the quantitative accuracy of our proposed method, as well as the SPH method as a whole for these kinds of applications.

### 1.3 Eulerian methods in welding simulation

Eulerian methods are very common for the simulation of arc welding processes and have been used since the advent of computational welding simulation. A common difficulty in these methods is the calculation of free surface flows. There are several numerical approaches which are commonly used for description of these free surface flows. They can be divided into two main groups [32]. The first group includes the so-called Front Capturing Methods (FCM) in which a fixed Eulerian computational mesh is employed and a free surface is ”expanded” along the volume of a certain layer. The thickness of this layer corresponds to several lengths of a computational cell. The most popular FCM are Volume of Fluid (VOF) [14,15] and Level Set (LS) [9] methods.

There exist also a number of other approaches where the free surface is considered as a sharp interface between two media, e.g., [24,25,27]. There the Arbitrary–Lagrangian–Eulerian method (ALE) is used, which allows for a deformation of the mesh, but it does not allow to solve problems with significant topological changes, like flow-splitting. However, most common arc welding processes involve a melting, detachment and an impingement of a filler material into a weld pool. Additionally, the attachment of the electric arc to the liquid electrodes is determined by thin sheaths (anode and cathode sheaths), which are located exactly at the surface of the liquid electrodes, and therefore a sharp definition of the free surface is necessary to consider these phenomena [26,30]. Therefore, neither the VOF/LS approach nor the ALE approach are completely satisfactory to accurately capture the process. Due to its strengths in modeling free surfaces as well as considering discontinuities and large topological changes and deformations, the SPH method became an interesting approach for modeling arc welding processes. However, little is yet known about the quantitative performance for the calculation of conductive/convective heat transfer compared to the established Eulerian methods, in the context of arc welding processes.

## 2 Method

Firstly, the following table summarizes the unique symbols used in this document. Please refer to this table when the meaning of symbols does not become clear from the immediate context.

In this section, we briefly describe our SPH model and discretization of the governing equations for incompressible weld pool dynamics. The SPH method discretizes mass at particles in space. These particles are advected and tracked through time and carry associated field quantities with them. The SPH method uses interpolation in order to compute unknown quantities and derivatives needed to solve partial differential equations (PDEs). An arbitrary scalar quantity $A(\boldsymbol{x})$ can thus be computed by interpolation from surrounding particles

$$A(\boldsymbol{x}) = \sum_{j \in \mathcal{N}_x} V_j A_j W(\boldsymbol{x} - \boldsymbol{x}_j; h), \tag{1}$$

where $W(\boldsymbol{x} - \boldsymbol{x}_j; h)$ is a compactly supported weighting function around position $\boldsymbol{x} \in \mathcal{R}^3$ with smoothing length $h$. The term $\sum_{j \in \mathcal{N}_x}$ denotes a summation over the neighboring particles $j$, with position $\boldsymbol{x}_j$ and volume $V_j$, around position $\boldsymbol{x}$, which lie within the compact support radius of weighting function $W(\boldsymbol{x})$. Derivative operators are typically shifted to the kernel function $W$ which significantly simplifies the discretization of PDEs. For more information about the background of the SPH method, the reader is referred to Koschier et al. [21] and Price [28].

### 2.1 Fluid flow

The Navier–Stokes equations for incompressible fluid flow are given by

$$\rho \frac{D\boldsymbol{v}}{Dt} = -\nabla p + \mu \nabla^2 \boldsymbol{v} + \boldsymbol{f}_{\text{ext}}. \tag{2}$$

Here, $\rho$ denotes the fluid density [kg m$^{-3}$], $\boldsymbol{v}$ the velocity [m s$^{-1}$], $p$ the pressure [N m$^{-2}$], $\mu$ the dynamic viscosity [Pa s] and $\boldsymbol{f}_{\text{ext}}$ the external volumetric forces [N m$^{-3}$]. In addition to the Navier–Stokes equations, the continuity equation has to be fulfilled

$$\frac{D\rho}{Dt} = 0, \tag{3}$$

which states that there must be no change in density, a necessary condition for incompressible fluids. This equation is also often used in convective form which relates the local density change to the divergence of the velocity field.

$$\frac{\partial \rho}{\partial t} = -\rho \nabla \cdot \boldsymbol{v} = 0 \tag{4}$$

The equations are solved using operator splitting. First the density of all particles $i$, $\forall i \in [1, N]$, where $N$ is the total number of particles in the simulation, is computed using the SPH interpolation function

$$\rho_i = \sum_{j \in \mathcal{N}_i} m_j W_{ij}, \tag{5}$$

where $W_{ij}$ is shorthand for $W(\boldsymbol{x}_i - \boldsymbol{x}_j; h)$. Afterward, the acceleration due to pressure forces is computed using the divergence-free SPH (DFSPH) method, as presented by Bender and Koschier [5]. Using this method, a set of momentum conserving forces is obtained which guarantees both constant density as well as a divergence-free velocity field. Other established methods, such as advection of density using the continuity equation [11] or explicit computation of pressure forces using a state equation [3] may suffer from volume losses in the former case and typically require small simulation time steps for the latter in order to ensure incompressibility. In contrast to this, we have found DFSPH, which implicitly computes pressure forces, to perform very well in all conditions, and it enabled the usage of large time steps while still remaining stable.

The viscosity term is also computed implicitly using the viscosity model by Weiler et al. [36]. This method computes accelerations

$$\boldsymbol{a}_i = \frac{\boldsymbol{v}_i^{t+1} - \boldsymbol{v}_i^t}{\Delta t} \tag{6}$$

due to viscous forces, by solving for $\boldsymbol{v}_\mu^{t+1}$ in the following equation:

$$\boldsymbol{v}^{t+1} = \boldsymbol{v}^t + \Delta t \frac{\mu}{\rho} \nabla^2 \boldsymbol{v}^{t+1}. \tag{7}$$

This can be written as a square symmetric positive definite linear system and solved efficiently in a matrix-free context using the conjugate gradient method. To the best of our knowledge, implicit viscosity solvers have not yet been employed for the simulation of weld pool dynamics, and we have found this model to also be very stable when using large time steps. Further discussion is given in Sect. 4.

The remaining forces in Eq. (2) which are part of the external volumetric force are the Lorentz force and the Momentum Sink which accounts for the morphological material change during solidification and melting.

The Lorentz force is computed in advance of the simulation and interpolated bilinearly from a grid onto the particles. This is analogous to the widely used approach for calculating Lorentz forces in arc welding process simulation, as for example used by Cho and Na [10], where also the stationary analytical solution developed by Kou and Sun [22] is used to calculate the Lorentz forces.

The Momentum Sink is computed using the Darcy-term method, which is a common approach in the literature to model the solidification of pure metals [7] and alloys [35], which is also a common approach in arc welding simulation, see e.g., [10]. The method adds an acceleration to the Navier–Stokes equation which has a strong flow inhibiting effect on the fluid, once the temperature of the fluid reaches below melting temperature $T_l$

$$\boldsymbol{a}_{porosity} = -\boldsymbol{v}C(1 - f_l(T)) \tag{8}$$

$$f_l(T) = \begin{cases} 1 & T > T_l \\ 0 & T \leq T_l. \end{cases} \tag{9}$$

In the above equations, $C$ denotes the morphological constant and $f_l(T)$ is the temperature-dependent liquid fraction, which is modeled using the heaviside function. The values

**Algorithm 1** Fluid Solver

```
procedure FluidSolve
    SolveDivergenceFreeVelocity($i$)                ▷ [5]
    SolveViscosity($i$)                             ▷ Eq. (7)
    for all particles $i$ do
        SolveMomentumSink($i$)                      ▷ Eq. (12)
    end for
    SolveConstantDensityPressure($i$)               ▷ [5]
    $\Delta t \leftarrow$ ComputeCFLTimeStep()
    for all particles $i$ do
        $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \Delta t \left(\boldsymbol{a}_i + \frac{f_{\text{ext}}}{\rho}\right)$
        $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \Delta t \boldsymbol{v}_i$
    end for
end procedure
```

for $C$ are often very large, resulting in very large deceleration as soon as $T \leq T_l$, so large in fact that simulations using explicit time stepping can become unstable. We remedy this by constructing an algebraic equation which computes the acceleration using the projected velocity of the next time step as follows:

$$\boldsymbol{a}_{porosity} = \frac{\boldsymbol{v}^{t+1} - \boldsymbol{v}^t}{\Delta t} = -\boldsymbol{v}^{t+1} C(1 - f_l(T)) \tag{10}$$

$$\boldsymbol{v}^{t+1} = \boldsymbol{v}^t \frac{1}{1 + C\Delta t(1 - f_l(T))}. \tag{11}$$

The acceleration is then computed by inserting the expression for $\boldsymbol{v}^{t+1}$

$$\boldsymbol{a}_{porosity} = \frac{\boldsymbol{v}^t}{\Delta t}\left(\frac{1}{1 + C\Delta t(1 - f_l(T))} - 1\right). \tag{12}$$

This semi-implicit formulation allows the usage of larger time steps in the simulation without causing instabilities. It should be noted that using large time steps with implicit Euler integration is known to add a significant damping effect, which diminishes the effect of the Darcy-term. This can be controlled by carefully determining a time step in the order of magnitude of $C^{-1}$ which is typically still very small, $\mathcal{O}(1 \times 10^{-8}\text{s}$ to $1 \times 10^{-4}\text{s})$, but not necessarily as small as required for explicit integration.

The main fluid driving forces in our simplified model of TIG spot welding is the Lorentz force, while the fluid behavior is otherwise significantly influenced by viscosity and the melting of solid material. The extensive use of implicit solvers enables us to have stable simulations when using large simulation time steps as well as large morphological constants $C$.

Finally, an overview of our SPH fluid solver is presented in Algorithm 1.

## 2.2 Heat transfer

Heat transfer is computed using the Fourier equation

$$\frac{D(\rho c_p T)}{Dt} = \nabla \cdot (\lambda \nabla T) + \dot{q}''', \tag{13}$$

where $\rho$ denotes the material density [kg m$^{-3}$], $c_p$ the specific heat capacity [J kg$^{-1}$ K$^{-1}$], $T$ the temperature [K], $\lambda$ the thermal conductivity [W K$^{-1}$ m$^{-1}$] and $\dot{q}'''$ the contribution from volumetric heat sources [W m$^{-3}$]. For the purpose of our SPH simulation, we make use of the relationship between specific enthalpy $h$ [J kg$^{-1}$] and the temperature when the observed medium is incompressible

$$h(T) = \int_0^T c_p(T)dT. \tag{14}$$

In this formulation, $c_p$ may also be a function of temperature, taking into account, e.g., the latent heat of melting. Resulting in the following formulation in terms of specific enthalpy

$$h\frac{D\rho}{Dt} + \rho\frac{Dh}{Dt} = \nabla \cdot (\lambda \nabla T) + \dot{q}'''. \tag{15}$$

$\frac{D\rho}{Dt} = 0$ is already enforced by the constant density component of the implicit pressure solver so that this term immediately drops out. The above equation is discretized using SPH and explicit Euler time integration, resulting in the following discrete equation for the fluid particle with index $i$

$$\rho_i \frac{h_i^{t+1} - h_i^t}{\Delta t} = \nabla \cdot (\lambda \nabla T)_i^t + \dot{q}_i^{t'''}. \tag{16}$$

The heat conduction term is discretized by

$$\nabla \cdot (\lambda \nabla T)_i = \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \frac{4\lambda_i \lambda_j}{\lambda_i + \lambda_j} (T_i - T_j) \frac{\nabla_i W_{ij} \cdot \boldsymbol{r}_{ij}}{||\boldsymbol{r}_{ij}||^2} \tag{17}$$

as proposed by Brookshaw [8] and is well-documented in the literature, e.g., by Trautmann et al. [33] and Das and Cleary [11]. It should be noted that the thermal conductivity $\lambda_i = \lambda(T_i)$ is generally a function of temperature and $\boldsymbol{r}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$.

## 2.3 Heat sources

In this work, a surface heat source $\dot{q}_i^{t'''}$ with constant power and Gaussian distribution was used. There are many available surface classification techniques available; however, the more accurate they are, the more computationally expensive they typically become. In this work, it was found that the coverage vector technique, as presented by Barecasco et al. [2],

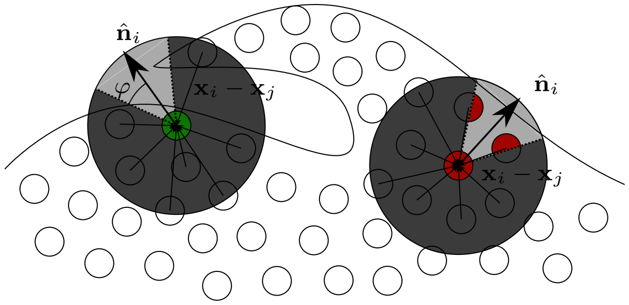


**Fig. 1** Surface particle classification using Eq. (20). The green particle is classified as a surface particle, while the red particle is classified as an interior particle

works quite well, since this results in almost flawless classification for the initial grid configuration and remains stable throughout. The normalized coverage vector $\hat{\boldsymbol{n}}_i$ for particle $i$ is computed as

$$\boldsymbol{n}_i = \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_i - \boldsymbol{x}_j) \tag{18}$$

$$\hat{\boldsymbol{n}}_i = \frac{\boldsymbol{n}_i}{||\boldsymbol{n}_i||}. \tag{19}$$

This vector points in the direction of the lowest particle density in the particle neighborhood. Particle $i$ is classified as a surface particle, if there is no other particle $j$ in the direction of $\hat{\boldsymbol{n}}_i$ in a cone with an angle of

$$\varphi < \cos^{-1}\left(\hat{\boldsymbol{n}}_{\boldsymbol{i}} \cdot \frac{(\boldsymbol{x}_j - \boldsymbol{x}_i)}{||\boldsymbol{x}_j - \boldsymbol{x}_i||}\right), \forall j \in \mathcal{N}_i. \tag{20}$$

The angle $\varphi$ is set to a fixed value. We have observed excellent classification results using an angle of $\varphi = 35°$. Here, $\mathcal{N}_i$ contains the indices of particles in the compact support radius of particle $i$. Surface classification using the coverage vector is shown in Fig. 1.

The particle shaded in green is classified as a surface particle, because no other particles are present in a cone of $\varphi$ around the coverage vector $\hat{\boldsymbol{n}}_{\boldsymbol{i}}$. The red particle is classified as an interior particle since there are two other particles in the cone around the coverage vector.

For the particles classified as surface particles, we propose the following fast projection method in order to compute the influence of a directed heat source. In general, however, this method could also be used to apply arbitrary boundary conditions at the fluid surface. The heat source is computed by additionally computing the visibility of the surface particles from an area heat source. This is shown in Fig. 2.

All surface particles outside of the region which lies in normal direction of the area source (white area) are discarded. Then, all remaining surface particles are checked for other neighboring particles in a cone around the vector in normal direction to the source. The green particles are eventually classified as being visible from the surface, while the red one is covered by a neighboring particle in the direction of the source. The yellow particle would be classified as visible, but can be discarded by an additional depth test.

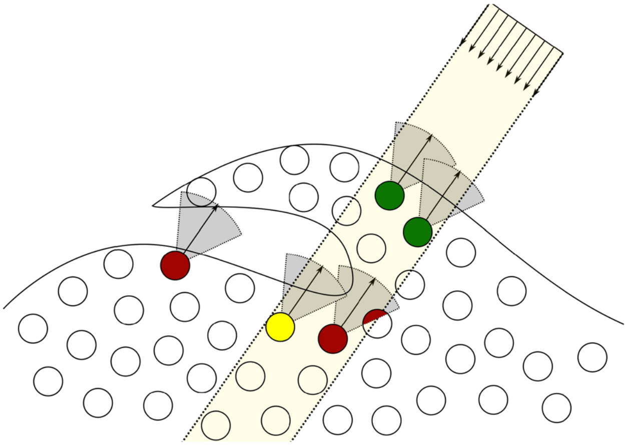


**Fig. 2** Projection of an area heat source onto previously detected surface particles. The green particles are visible, the yellow particle is discarded by a depth test, and the red particles are invisible to the heat source

In this work, the heat source is parameterized as a Gaussian and in order to ensure constant power of the heat source, weighted normalization is applied using the previously computed Gaussian weights. This results in the expression

$$\dot{q}''_{i,surface} = P_{source} w_i \frac{3}{\pi r_{HS}^2} \exp\left(-3\frac{||\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_{\boldsymbol{0}}||^2}{r_{HS}^2}\right), \tag{21}$$

where $r_{HS}$ is the standard deviation, $\boldsymbol{x}_{\boldsymbol{0}}$ the center of the Gaussian and $w_i$ the weight which is needed for normalization of the total power to the target power $P_{source}$.

The subscript $i, surface$ denotes only the particles which have in the previous step been identified as surface particles and have been determined to be “visible” from the heat source, while $\dot{q}''$ denotes the heat source as Watts per unit area $[Wm^{-2}]$.

In our TIG simulation setup, all boundaries are adiabatic and do not require any special handling for heat transfer. Since heat can only be conducted within the material itself, the SPH formulation is adiabatic by construction. Finally, the enthalpy in terms of the temperature (and vice versa) is precomputed by integration of Eq. (14). Our heat solver algorithm is outlined in Algorithm 2, while the full algorithm containing both the heat solver and fluid solver is shown in Algorithm 3.

**Algorithm 2** Heat Transfer Computation

```
procedure ComputeHeatTransfer
    for all particles $i$ do
        ComputeTemperature($i$)            ▷ Eq. (14)
    end for
    for all particles $i$ do
        ClassifySurface($i$)               ▷ Eq. (20)
    end for
    for all particles $i$ do
        ApplyHeatSource($i$)               ▷ Eq. (21)
    end for
    for all particles $i$ do
        ComputeHeatConduction($i$)         ▷ Eq. (17)
    end for
    for all particles $i$ do
        ExplicitEulerIntegration($i$)      ▷ Eq. (16)
    end for
end procedure
```

**Algorithm 3** SPH Solver

```
procedure SolveFluidAndHeatConduction
    $t \leftarrow t_{\text{start}}$
    while $t < t_{\text{end}}$ do
        ComputeHeatTransfer()              ▷ Algorithm 2
        FluidSolve()                       ▷ Algorithm 1
        $t \leftarrow t + \Delta t$
    end while
end procedure
```

## 2.4 Particle to grid transfer

For scientific evaluations, it can be quite cumbersome to use particle data, since many visualization techniques, i.e., line plots, contour plots, stream lines, etc., rely on the ability to interpolate the underlying data at any point in space. While this can be achieved by using the common SPH summation from Equation (1), this method suffers from the issue of particle deficiency at fluid boundaries where the interpolation quality will decrease rapidly. The ability to reconstruct constant functions can be restored by using normalization, as is also done by Komen et al. [19], which often yields satisfactory results in practice. This normalized SPH interpolation is also often called Shepard interpolation in the literature and simply normalizes Eq. (1)

$$A(\boldsymbol{x}) = \frac{\sum_{j \in \mathcal{N}_x} V_j A_j W(\boldsymbol{x} - \boldsymbol{x}_j; h)}{\sum_{j \in \mathcal{N}_x} V_j W(\boldsymbol{x} - \boldsymbol{x}_j; h)}, \tag{22}$$

which locally restores the property of accurately representing constant functions.

Nevertheless, we propose a novel method which does not rely on the SPH kernel for interpolation of particle data onto a grid and provides an efficient and accurate method of transferring the underlying continuous field from the particle nodes to specific grid nodes. Our proposed method overlays a regular grid (in theory this could be any kind of unstructured grid) and solves a linear least squares problem for the values at the grid nodes

$$\min \sum_{i=1}^{N} (f(\boldsymbol{x}_i; \boldsymbol{\theta}) - y_i)^2, \tag{23}$$

where $f(\boldsymbol{x}_i; \boldsymbol{\theta})$ is the function which linearly interpolates the values at the nodes of the grid $\boldsymbol{\theta} = (\theta^0, \ldots, \theta^M) \in \mathcal{R}^M$ to the position $\boldsymbol{x}_i$ of particle $i$ and $y_i$ is the field value at particle $i$. The normal equations for this problem are given by

$$\mathbf{M}^T \mathbf{M} \boldsymbol{\theta} = \mathbf{M}^T \boldsymbol{y} \quad \mathbf{M}_{ij} = w_{ij}. \tag{24}$$

The matrix $\mathbf{M} \in \mathcal{R}^{N \times M}$ has as many rows as there are particles and as many columns as there are grid nodes. $w_{ij}$ denotes the linear interpolation weight of particle $i$ from grid node $j$. Matrix $\mathbf{M}$ is very sparse and is solved using the matrix-free conjugate gradient solver of Eigen. Since there are not always particles in every grid cell, the matrix $\mathbf{M}$ is extended by a set of identity equations for grid nodes where there are no particles in the vicinity and the right-hand side is equal to a default value for the interpolated field.

Using this, or other interpolation methods, and depending on the desired grid resolution, it may be possible for gaps to appear in the interpolation grid, where no particles contribute to the interpolated value. In order to avoid this, we perform a smoothing step where the values in the missing cells may be filled in by using Laplacian smoothing

$$\Delta \boldsymbol{\theta} = 0. \tag{25}$$

Other smoothing kernels, e.g., Bi-Laplacian or Gaussian kernels, may also be used, but we have found the simple 7-point 3D Laplacian kernel to work well for our purposes. To do the smoothing step, we implemented a fast marching algorithm to select the grid nodes which are directly adjacent to grid nodes with neighboring particles. For the smoothing operation, the already computed grid values serve as boundary conditions such that gradients from the already computed fields are preserved and the gaps can be filled in with meaningful values. The actual boundary of the enclosing grid is treated using zero-gradient boundary conditions. In practice, we frequently used a grid size equal to the particle diameter and smoothed for up to a distance of 1 to 3 nodes. This yielded very good results in the regions with gaps as is discussed in the evaluation of Fig. 7.

In comparison with the usual interpolation technique which uses SPH to average local information at a specific position, our method attempts to approximate the underlying field globally and continuously over all particles. In practice, we have found this to result in smoother interpolated fields while at the same time having equal or better approximation quality. A brief evaluation of our interpolation method is shown in Sect. 3.1.

## 2.5 Boundary conditions in SPH

Boundary volumes are taken into account using the analytical approach for Volume Maps by Bender et al. [6]. As the name suggests, the method computes analytical volume contributions of solid objects to the SPH integral as a precomputation step, so that it can be evaluated using a simple lookup and polynomial interpolation. As discussed in the original paper, this method has the advantage of accurately representing solid objects and boundary conditions, and the resulting surface varies smoothly with position. When the boundary is sampled using particles, e.g., using the method of Akinci et al. [1], particle movement becomes bumpy due to the unevenness of the discretization.

Using Volume Maps, boundaries can simply be integrated into the typical SPH summation of Eq. (1) resulting in the following equation:

$$A_i = \sum_{j \in \mathcal{N}_i^f} V_j^f A_j^f W_{ij} + V_i^b A_i^b W_{ib}, \tag{26}$$

where the superscript $f$ denotes contributions from fluid particles and $b$ from boundary objects. The volume $V_i^b$ denotes the precomputed volume occupied by the boundary in the compact support radius of particle $i$, while the weight $W_{ib}$ is computed for a representative particle on the surface of the boundary. Finally, $A_i^b$ is chosen to comply with prescribed boundary conditions, e.g., being set to a fixed value for Dirichlet boundary conditions. For our model, the Volume Maps boundaries are mainly used in order to extend the fluid domain in order to avoid particle deficiencies for the computation of density and pressure.

We use free slip boundaries at all surfaces, meaning that

$$\left.\frac{\partial \boldsymbol{v}_t}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = 0, \tag{27}$$

where $\partial\Omega$ denotes the entirety of the bounding cylinder (see Fig. 3), while $\partial/\partial\boldsymbol{n}$ denotes the derivative in normal direction. The velocity in tangential direction to the surface is denoted by $\boldsymbol{v}_t$. This does not require any special treatment in the SPH formulation. Additionally, we impose adiabatic boundary condition for the heat equation

$$\left.\frac{\partial T}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = 0, \tag{28}$$

for the entire boundary, while additionally adding a source term at the top surface (line $DC$ in Fig. 3) of the boundary

$$\left.\dot{q}''\right|_{DC} = \dot{q}''_{\text{surface}}, \tag{29}$$

which is computed using the heat source from Equation (21). The adiabatic condition is also satisfied by construction, while the method described in Sect. 2.3 is used to apply the heat sources.

## 2.6 2D-FEM model

### 2.6.1 Model of electromagnetic processes in COMSOL

For the magnetostatic simulation by means of COMSOL Multiphysics, we use the so-called $\boldsymbol{A}_m - \varphi$ formulation which gives equations for the magnetic vector potential $\boldsymbol{A}_m$ and the scalar electric potential $\varphi$

$$\Delta \boldsymbol{A}_m = -\mu_0 \boldsymbol{J}, \tag{30}$$
$$\boldsymbol{B} = \nabla \times \boldsymbol{A}_m, \tag{31}$$
$$\nabla \cdot (\sigma \nabla \varphi) = 0. \tag{32}$$

Here, $\boldsymbol{J}$ and $\boldsymbol{B}$ are the current density and magnetic induction, respectively, $\mu_0$ is the permeability of vacuum and $\sigma$ is the electrical conductivity. The volumetric Lorentz force is governed by

$$\boldsymbol{F_L} = \boldsymbol{J} \times \boldsymbol{B}. \tag{33}$$

The governing equations, Eq. (30) and Equation (32), are subject to the boundary conditions

$$\left.\boldsymbol{n} \times \boldsymbol{A}_m\right|_{AB \bigcup BC \bigcup CD} = 0, \tag{34}$$
$$\left.\frac{\partial \varphi}{\partial r}\right|_{BC} = 0, \tag{35}$$
$$\left.\varphi\right|_{AB} = 0, \tag{36}$$
$$\left.\sigma \frac{\partial \varphi}{\partial z}\right|_{CD} = J_n, \tag{37}$$

where $J_n$ denotes the current density distribution along surface $CD$.

### 2.6.2 Model of electromagnetic processes in WM

The model of electromagnetic processes is based on the stationary magnetic field diffusion equation [12] which in the case of axial symmetry reads as follows:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial B_\theta}{\partial r}\right) + \frac{\partial^2 B_\theta}{\partial z^2} - \frac{B_\theta}{r^2} = 0, \tag{38}$$

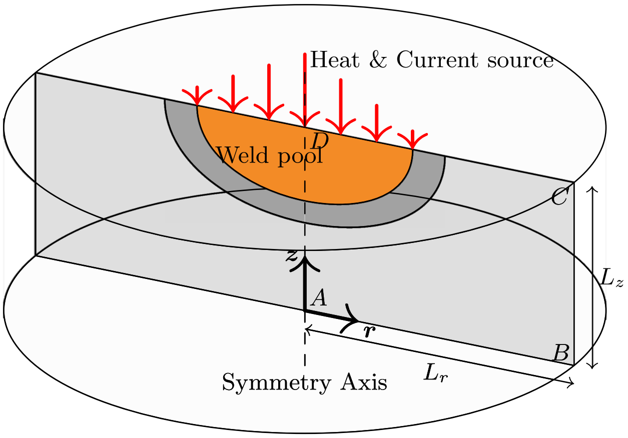


**Fig. 3** Computational domain for both SPH and FEM simulations. The FEM methods solve the problem in rotational symmetry in the $ABCD$ plane, while SPH considers the full cylinder by explicit rotation of $ABCD$ around the denoted symmetry axis

where $B_\theta$ is an azimuthal component of magnetic induction. From Ampere's law

$$\mu_0 \boldsymbol{J} = \nabla \times \boldsymbol{B}, \tag{39}$$

the next boundary conditions for Eq. (38) can be readily obtained

$$B_\theta\Big|_{CD} = \frac{\mu_0}{r}\int_0^r J_n(s)sds, \tag{40}$$

$$B_\theta\Big|_{BC} = \frac{\mu_0 I_w}{2\pi L_r}, \tag{41}$$

$$\frac{\partial B_\theta}{\partial z}\Big|_{AB} = 0. \tag{42}$$

Here, $L_r$ is the radius of computational domain (see Fig. 3), while $I_w$ denotes the welding current. Under the given magnetic field, the current density and Lorentz force are obtained from Eq. (39) and Equation (33), respectively.

### 2.6.3 Numerical procedure in COMSOL Multiphysics

The governing equations in COMSOL are solved using the finite element method (FEM). For temporal discretization, the BDF (backward differentiation formula) was applied, with default settings of COMSOL 5.6. The spatial discretization can be seen in Fig. 4. Here, the nodes used quadratic shape functions for temperature, velocity, magnetic strength and for scalar and vector potential. The shape functions utilized for the solution of the pressure were linear. The sparse direct solver PARDISO was selected to solve the resulting system of linear equations.

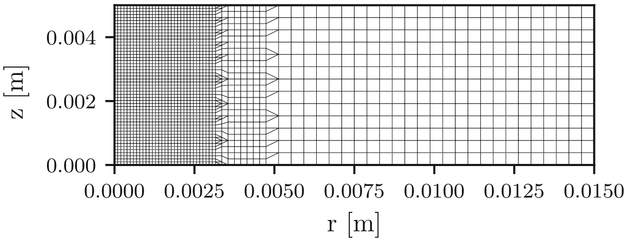


**Fig. 4** COMSOL mesh as mixture of quadrilaterals and triangles

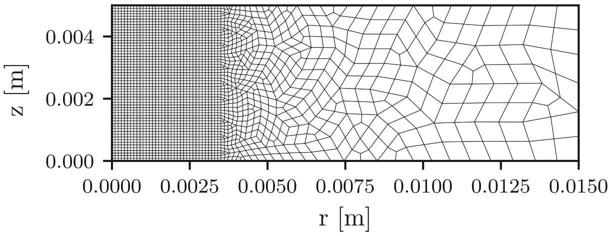


**Fig. 5** FEM mesh as Quad9 quadrilaterals. The Navier–Stokes equations are solved in the region of interest $\Omega = \{0\,\text{mm} < r < 3.5\,\text{mm}, 0\,\text{mm} < z < 5\,\text{mm}\}$, while the heat transport is analyzed in whole domain

### 2.6.4 Numerical procedure in WM

The governing equations are solved using the Galerkin finite element method (FEM) [37]. A characteristic-based scheme [38] was used in temporal discretization of Eqs. (2) and (13) along with FEM approximation in space, see Fig. 5. Nine node biquadratic shape functions are employed for velocity, temperature and magnetic fields, whereas the pressure is approximated by four node bilinear shape functions. Such a choice of mixed interpolation for velocity and pressure fields provides stability of the numerical procedure [37]. The sparse direct solver PARDISO [29] was employed to solve the resulting systems of linear equations. All the numerical algorithms were implemented in the Wolfram Mathematica Language (WM).

## 3 Results

### 3.1 Grid interpolation

As noted before, while all following evaluations could technically be done on particle data alone, many existing evaluation functions and programs, e.g., ParaView and Tecplot, are tailored to grid structures and unstructured meshes. This makes the usage of regular and unstructured meshes very appealing for quantitatively evaluating and comparing scientific computations. To this end, we compare the interpolation accuracy using Naïve SPH interpolation (see Equation (1)), Shepard-corrected SPH interpolation (see Eq. (22)) and our proposed least-squares linear interpolation (see Eq. (23)).

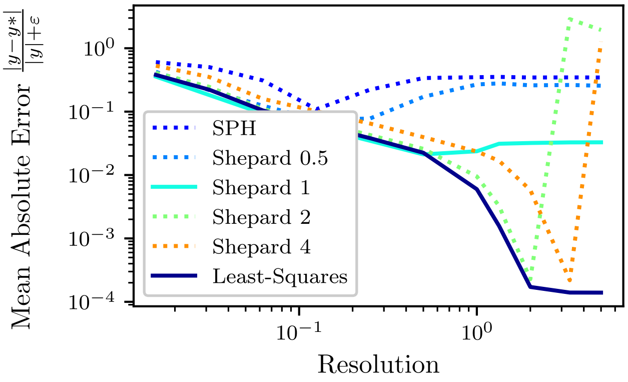


**Fig. 6** Comparison of particle to grid interpolation techniques

Figure 6 shows a comparison of the different interpolation approaches. Each method has been used to compute the particle to grid transfer of the same particle data. The error is computed by interpolating these grids back to the original particle positions and computing the mean absolute error. The error is determined by evaluating the error at the particle positions using linear interpolation from the computed grids. The underlying particle data are the velocity field of a randomly selected frame of a fluid dynamics simulation in motion, but the same trend is visible regardless of which field is approximated. The regular grid spacing $d_{spacing}$ is determined by a multiple $n_{grid}$ of the SPH particle radius $r_{part}$

$$d_{spacing} = n_{grid} \cdot r_{part}. \tag{43}$$

The resolution of the grid shown in Fig. 6 is then computed as the inverse of this factor, $n_{grid}^{-1}$, i.e., a higher resolution corresponds to a smaller grid spacing $d_{spacing}$ and thus a larger number of total grid nodes.

The Shepard interpolation was evaluated using a varying factor for the SPH compact support radius depending on the factor for the grid spacing $h_{shepard} = d_{spacing} \cdot n_{shepard}$. It can be seen that both our proposed least-squares interpolation as well as Shepard interpolation using the grid spacing with a factor of 1 and 2 as compact support radius perform the best in this comparison, for grid spacings larger than $r_{part}$. Naïve SPH interpolation performs best when the compact support radius is equal to the compact support used in the actual simulation. It also suffers from worse interpolation quality due to particle deficiency at the surface, leading to large errors in the interpolation.

For finer resolutions, it can be seen that the least-squares interpolation performs best in all cases, while the Shepard interpolation with a factor of 2 and subsequently a factor of 4 have a comparable error when a grid spacing of $0.5r_{part}$ and $0.25r_{part}$ is used respectively. From this comparison we conclude that both Shepard interpolation, as well as our proposed least-squares interpolation can work very well for particle data. The interpolation quality of our least-squares approach becomes more apparent when the grid-resolution increases as well.

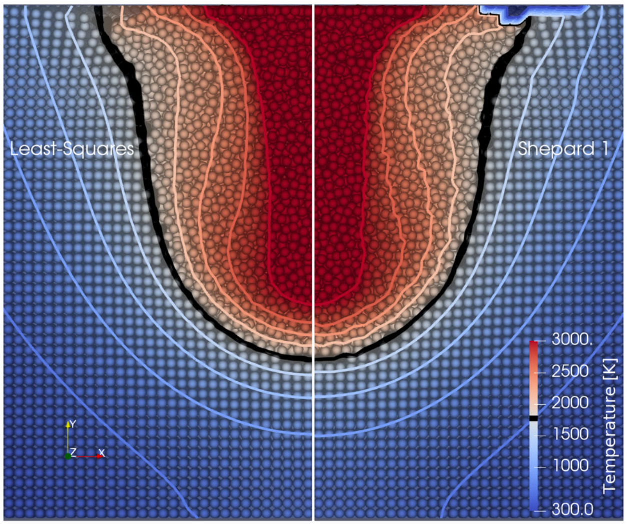


**Fig. 7** A comparison of our proposed least-squares interpolation method using smoothing of one surrounding grid node (left) and Shepard interpolation (right) without smoothing

Another advantage of our interpolation is shown in Fig. 7, where the particle to grid transfer of the temperature field using two different interpolation methods is shown. The left half shows the interpolation of our proposed least-squares method and smoothing for up to one neighboring grid node, and the right side shows Shepard interpolation.

From this qualitative comparison, it is possible to see that while the values are almost identical in the entire domain, the least squares interpolation results in a much smoother field. This is especially visible at the temperature contour lines within the black border. In this figure the advantages of our smoothing approach can also be observed. While the top right of the Shepard interpolation shows gaps in the interpolation due to missing particles, the left side shows that this part (top left) can be filled in with very plausible values using the aforementioned Laplacian smoothing. While the filled in values should be treated with care, this kind of filling in allows the consistent reconstruction of fixed domains from particle data.

For all following particle to grid interpolations, we use our least-squares interpolation with a grid resolution of twice the particle radius, because this is the same resolution as for the WM and COMSOL simulations. In addition, we apply a smoothing of up to 3 particle nodes, although only for times close to the end of the simulation is the full smoothing length actually needed. At this resolution, it can be expected to incur a mean absolute error in the interpolated data of 4% for scalar values.

**Table 1** Material parameters of high-alloy steel

| Property | Unit | Value |
|---|---|---|
| Density $\rho$ | $\mathrm{kg\,m^{-3}}$ | 8100 |
| Surface tension $\gamma$ | $\mathrm{N\,m^{-1}}$ | disabled |
| Viscosity $\mu$ | Pa s | 0.004 |
| Thermal conductivity $\lambda$ | $\mathrm{W\,m^{-1}K^{-1}}$ | 22.9 |
| Electrical conductivity $\sigma$ | $\mathrm{S\,m^{-1}}$ | $10^6$ |
| Heat capacity $c_p$ | $\mathrm{J\,kg^{-1}K^{-1}}$ | 800 |
| Melting temperature $T_l$ | K | 1773 |
| $C$ from Eq. (12) | $\mathrm{s^{-1}}$ | $3.7 \times 10^4$ |

**Table 2** Domain parameters

| Property | Unit | Value |
|---|---|---|
| Radius of the cylinder $L_r$ | m | $15 \times 10^{-3}$ |
| Height of the cylinder $L_z$ | m | $5 \times 10^{-3}$ |
| Initial temperature | K | 300 |
| Heat source center $\boldsymbol{x_0}$ | m | $(0,\ 6 \times 10^{-3},\ 0)$ |
| Heat source dist. radius $r_{HS}$ | m | $2.7 \times 10^{-3}$ |
| Heat source power $P_{source}$ | W | 1213.8 |
| Total current $I_w$ | A | 140 |
| External forces $\boldsymbol{f}_{\text{ext}}$ | $\mathrm{N\,m^{-3}}$ | See Fig. 8 |

### 3.2 Simulation parameters

To allow for a first validation, the relevant material parameters have been simplified to constants, see Table 1. It should be mentioned that the morphological constant $C$ is usually given without division over the density, where it would evaluate as $C = 3 \times 10^8 \mathrm{kgm^{-3}\ s^{-1}}$.

The properties are applied homogeneously over the calculation domain, which consists of a rotationally symmetrical cylinder, as shown in Fig. 3. The details of the computational domain are given in Table 2. It is important to note that the SPH domain is realized in full 3D, while the FEM domain is just considering a 2D plane with rotational symmetry around the symmetry axis AD.

The external Lorentz force field due to electromagnetic forces is pre-computed in advance of the fluid-flow simulation, according to Eq. (33) in order to be able to use the same external forces for both the SPH and WM simulations. The magnetic strength field is obtained from the solution of Eq. (38) by means of the FEM. From these, the Lorentz forces $\boldsymbol{F}_L$ are computed by Eq. (33), and are then included as external volumetric forces $\boldsymbol{f}_{\text{ext}}$ in the Navier–Stokes equation Eq. (2), by a rotation of the 2D plane around the symmetry axis

$$\boldsymbol{f}_{\text{ext}} = \boldsymbol{F}_L. \tag{44}$$

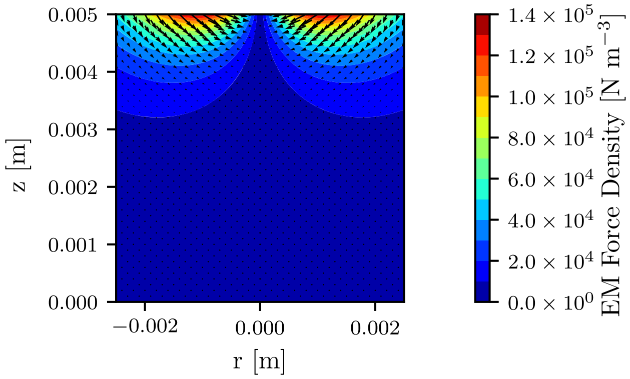


**Fig. 8** Lorentz force in the region of interest with the direction indicated as an arrow and the magnitude indicated as the underlying color. (Color figure online)

The vector field of the forces is calculated from the applied electric current density $\boldsymbol{J_n}(s)$ in Eq. (40), distributed as a Gaussian along the DC boundary (see Fig. 3) in the same way that the heat source distribution is defined in Equation (21), but with the welding current $I_w$

$$\boldsymbol{J_n}(s) = I_{\mathrm{w}} \frac{3}{\pi r_{HS}^2} \exp\left(-3 \frac{||\boldsymbol{x_i} - \boldsymbol{x_0}||^2}{r_{HS}^2}\right). \tag{45}$$

Figure 8 shows the Lorentz force in the region of interest that was pre-calculated with the WM-code, according to Sect. 2.6.2 as the electrical conductivity was constant. The resulting force field was then applied in rotational symmetry on the SPH particles. The Lorentz force for COMSOL were calculated within the software according to Sect. 2.6.1 and they were found to be nearly identical in the region of interest, showing only small deviations far away due to the difference of the system of equations (c.f. Sects. 2.6.1 and 2.6.2).

The convergence of the simulation was achieved using the numerical parameters listed in Tables 3 and 4. The SPH particle diameter is chosen to be equivalent to the element size in the region of interest for the Wolfram Mathematica and COMSOL FEM simulation. The WM method and the SPH method use adaptive time integration using the CFL criterion, with identical maximum time step sizes but different factors. The WM method made use of both second-order consistent implicit time integration as well as higher-order finite element basis functions. As such, the CFL factor in SPH was chosen heuristically to be smaller, since there a only first-order implicit time integration scheme with operator splitting is used. For COMSOL, the backward differentiation formula (BDF) was used for temporal discretization, with default parameters.

The SPH simulation was run on a compute node with 36 cores and took 86 h for 2.5 s of simulated time. The WM simulation was run on a 4 core machine and completed in roughly 9 h. Finally, the COMSOL simulation was run on a

**Table 3** Numerical parameters of SPH

| Property | Unit | Value |
|---|---|---|
| SPH particle radius | m | $5 \times 10^{-5}$ |
| Maximum time step | s | $1 \times 10^{-4}$ |
| CFL criterion | – | 0.1 |

**Table 4** Numerical parameters of WM

| Property | Unit | Value |
|---|---|---|
| Element size in region of interest | m | $10 \times 10^{-5}$ |
| Maximum time step | s | $1 \times 10^{-4}$ |
| CFL criterion | – | 0.5 |

compute node with 48 cores and completed in approximately 20 min. It is strongly noted that these computations are not at all comparable between one another. For one, the simulations were run on very different hardware. Additionally, the SPH method was run in full 3D which has significantly more degrees of freedom than the rotationally symmetric WM and COMSOL simulations. In the SPH simulation, we use roughly 3.5 million particles with 3 degrees of freedom for all vector quantities, while the meshes of the COMSOL and WM simulation have roughly 10 k grid nodes with 2 degrees of freedom for all vector quantities. As such the number of degrees of freedom of the SPH simulation is $\approx 500$ times larger than for the other simulation methods. In our opinion, this shows good performance and scaling of our SPH method.

## 3.3 Simulation results

### 3.3.1 Conductive heat transfer

Before the quantitative comparison of forced convective and conductive heat transfer is performed, a comparison without the external forces $f_{\text{ext}}$ was made, i.e., without any fluid movement and therefore no convection. As can be seen from the shape of the melting temperature iso-contours in Fig. 9, the calculation for the conductive heat transfer was in perfect agreement between SPH, WM and COMSOL for all time steps. Therefore, we consider the purely conductive part of the heat transfer module in SPH as validated.

### 3.3.2 Forced convective and conductive heat transfer

**Quantitative comparison** The dimensions of the melt-pool over the course of the simulations are shown in Fig. 10, for the case when the external forces were set to the Lorentz force as shown in Eq. (44). The weld pool dimensions are computed by solving a nonlinear algebraic equation for the position of the melting front. Overall both characteristic melt-pool dimensions, the depth and the width, show very good agreement between all three methods and exhibit a similar development over time. The depth of the weld pool for the SPH and COMSOL simulation match almost perfectly across all time steps. During the first $t < 0.5$ s, the FEM weld pool is deeper, but afterward agreement between all three methods is again observed for $t > 0.5$ s. Starting at roughly $t > 1.25$ s, the development of the depth increases for the WM solver, reaching full penetration about $\Delta t \sim 0.1$ s earlier than the SPH method, while the COMSOL solution is a bit slower, reaching full penetration $\Delta t \sim 0.03$ s later. Also the depth of the COMSOL solution does not seem to develop entirely smoothly, displaying minor kinks at $t = 1.35$ s and $t = 1.5$ s.

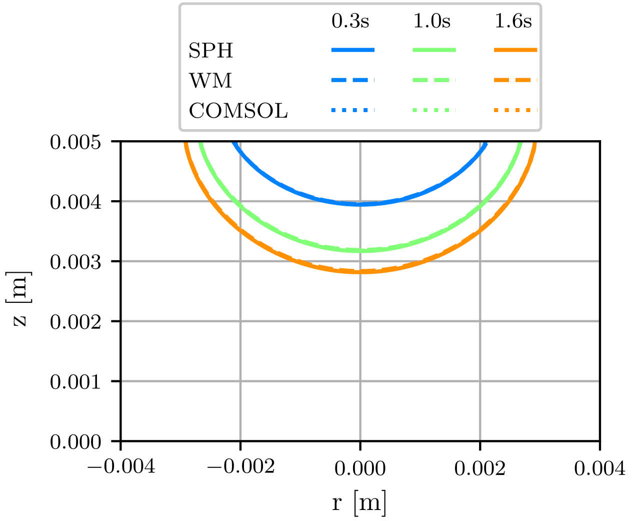


**Fig. 9** Melting temperature (1773K) iso-contours at different times for SPH, WM and COMSOL without Lorentz force $f_{\text{ext}} = 0\,\text{N}\,\text{m}^{-3}$, i.e., pure heat conduction

As the interpolation of the SPH solution into a grid has limitations close to the boundaries, since fewer particles were present to interpolate the results, the width was not computed exactly on the boundary, but 0.5 mm below the surface. While we use our proposed interpolation method to obtain a plausible filling in of the gap, a better result is still obtained when looking at the weld pool width a short distance into the domain. Here, the width matches up nearly perfectly between all three solver methods, with a minor deviation for $t < 0.5$ s, where the COMSOL solution appears right between the SPH and WM solution.

In Fig. 11, the melting temperature iso-contour can be seen, at characteristic time steps of the process for these developments of 0.3 s, 1.0 s and 1.6 s. These characteristic times were also chosen for all future comparisons. At 0.3 s, the melt-pool is more shallow in SPH, compared to WM and COMSOL, with its melting contour having a very similar shape to that shown in Fig. 9 for the same time. This

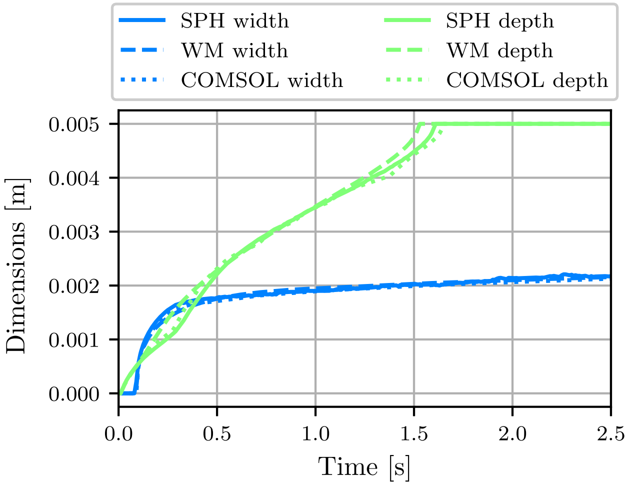


**Fig. 10** Evolution over time of the melt-pool dimensions depth and width at 4.5 mm

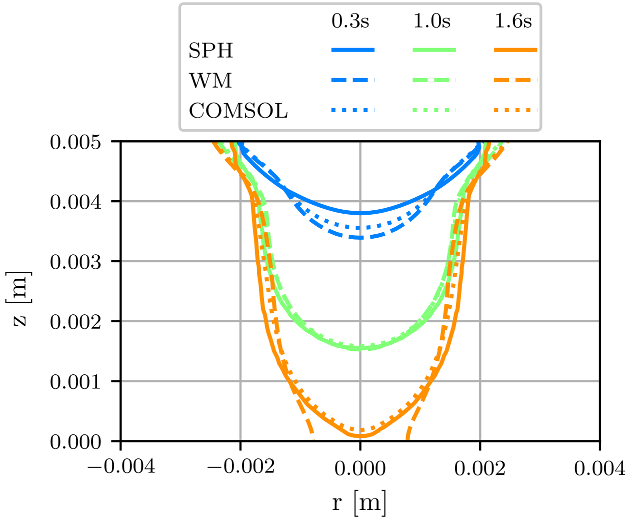


**Fig. 11** Melting temperature (1773K) iso-contours at different times for SPH and WM, including conductive heat transfer driven by the Lorentz force

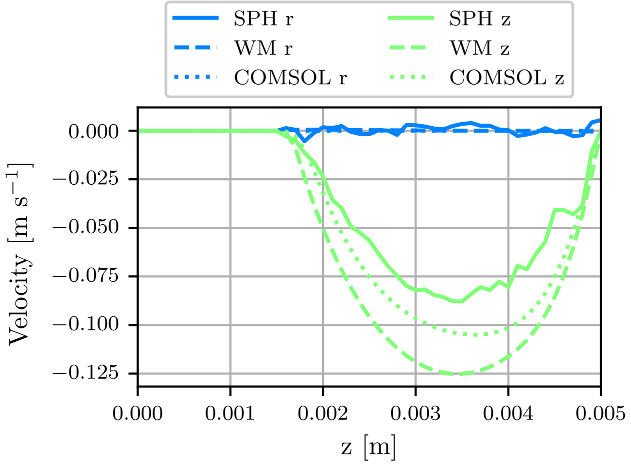


**Fig. 12** Velocity by component along the symmetry axis at $t = 1.0$ s

suggests that the convective heat transfer has not yet started in the SPH case, which could be due to insufficient amount of particles to allow for any fluid movement. At 1.0 s, we can see excellent agreement between the SPH, WM and COMSOL method, not just in terms of maximal width or depth, but also in the general shape of the melt-pool. At 1.6 s, the agreement between SPH and COMSOL can still be regarded as very good, while WM deviates for these solutions in the lower part of the melt-pool $z \le 0.001$ m, as full penetration of the work piece is already achieved for WM at this time step, whereas both SPH and COMSOL are just about to reach the bottom boundary. For the intermediate part 0.001 m $\le z \le$ 0.0045 m, the COMSOL solution is right in between the WM and SPH solution. It should be mentioned that there are slight deviations in the top part $z > 0.0045$ m for the SPH solution, which occur due to missing particles.

To obtain a better understanding of the significant velocities, the velocity distributions at $t = 1.0$ s along the symmetry axis (z) are shown in Fig. 12. The shape of the velocity distribution is very similar for all three solutions, while the difference between WM and COMSOL is quite comparable to the difference between COMSOL and SPH. This is especially remarkable since both WM and COMSOL are FEM solvers of the same problem. In the light of the observed differences between both FEM approaches, the disagreement of the SPH method with the FEM solvers appear to actually be within the margin of error that can be expected from the application of a FEM in general. On the contrary, the overall level of agreement is especially remarkable when considering that in the SPH method an entirely different discretization is used and that a whole three dimensional domain is simulated. The agreement in the radial direction is very good for all three methods. For SPH, the velocities fluctuate around 0 m s$^{-1}$, due to the three dimensional nature of the simulation. The fluctuations of the velocity for SPH may also be partly attributed to the fact that only a slice of full 3D data is examined, where no constraints explicitly enforce strict movement within the radial plane. As such, it is only to be expected for a certain amount of statistical noise to be present in the SPH velocity components, compared to the FEM methods where the considered axis was realized as a rotational symmetry boundary, and as such does not permit movement in radial direction.

Lastly, the temperature for the chosen characteristic time steps along the symmetry axis (z) is shown in Fig. 13. Again similar observations regarding agreement as before can be made. We see excellent agreement in the temperature distribution for $t = 1.0$ s between all three methods, while some deviation can be observed for times close to the beginning $t = 0.3$ s and for times close to full penetration $t = 1.6$ s.

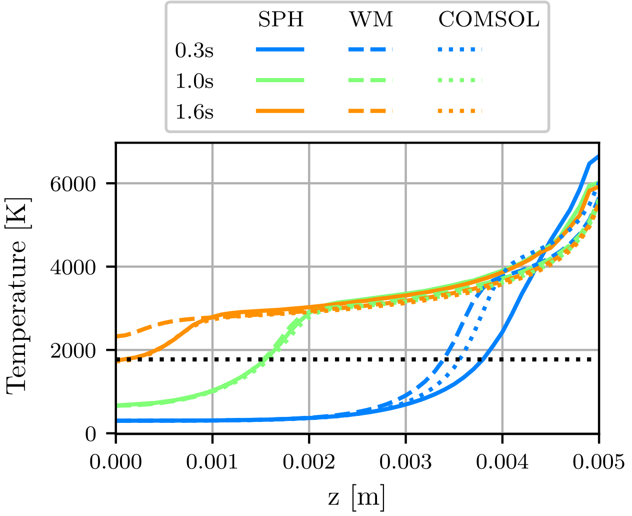


**Fig. 13** Temperature along symmetry axis at different times. The dotted black line marks the melting temperature (1773K)

At time $t = 0.3$ s, it should be pointed out that SPH reaches a higher maximal temperature at $z = 0.005$ m and does not share the characteristic shape we observe for the other distributions, but exhibits a more uniform temperature decay as one would expect from a purely conductive heat transfer. As mentioned earlier in the description of Fig. 11, this is most likely due to inhibited fluid movement at early times, as can also be observed later in Fig. 14b for $t = 0.3$ s. Again, for this time step $t = 0.3$ s, the COMSOL aligns more closely with the WM method, although not reaching perfect agreement. The shape of the COMSOL solution at this time seems to indicate weaker conductive heat transfer than WM, therefore most likely representing an intermediate state between the SPH and WM solution, which is an observation already made for some of the previous data, e.g., Fig. 12.

For $t = 1.0$ s, we see excellent agreement between all three solutions for the lower half of the temperature distributions ($z > 0.0025$ m); however, in the upper half, we observe slightly higher temperatures in SPH compared to WM and COMSOL.

At $t = 1.6$ s, the largest discrepancy between the three solutions can be found at the bottom of the melt-pool. While the SPH and COMSOL solution show very good agreement at the lower part of the axis ($z \leq 0.001$ m), the WM solution differs strongly from both. This could be, as previously mentioned, due to the effect of the lower boundary, as WM has already achieved full penetration at this time step, while the SPH and COMSOL solution have not yet achieved it. However, for the upper part of the axis ($z > 0.001$ m), the agreement between COMSOL and WM is very good, while the temperatures calculated by SPH is slightly higher, with the deviation further increasing slightly with increasing z. All distributions have the previously mentioned characteristic shape, however the WM solution consistently predicts a deeper weld pool, indicating a stronger convective heat transfer in the entire melt-pool, consistent with the observations on the velocities in Fig. 12. Regarding the temperature profile along the height axis, it can also be said that our SPH method lies well within the margin of error that can be expected between two different FEM solvers.

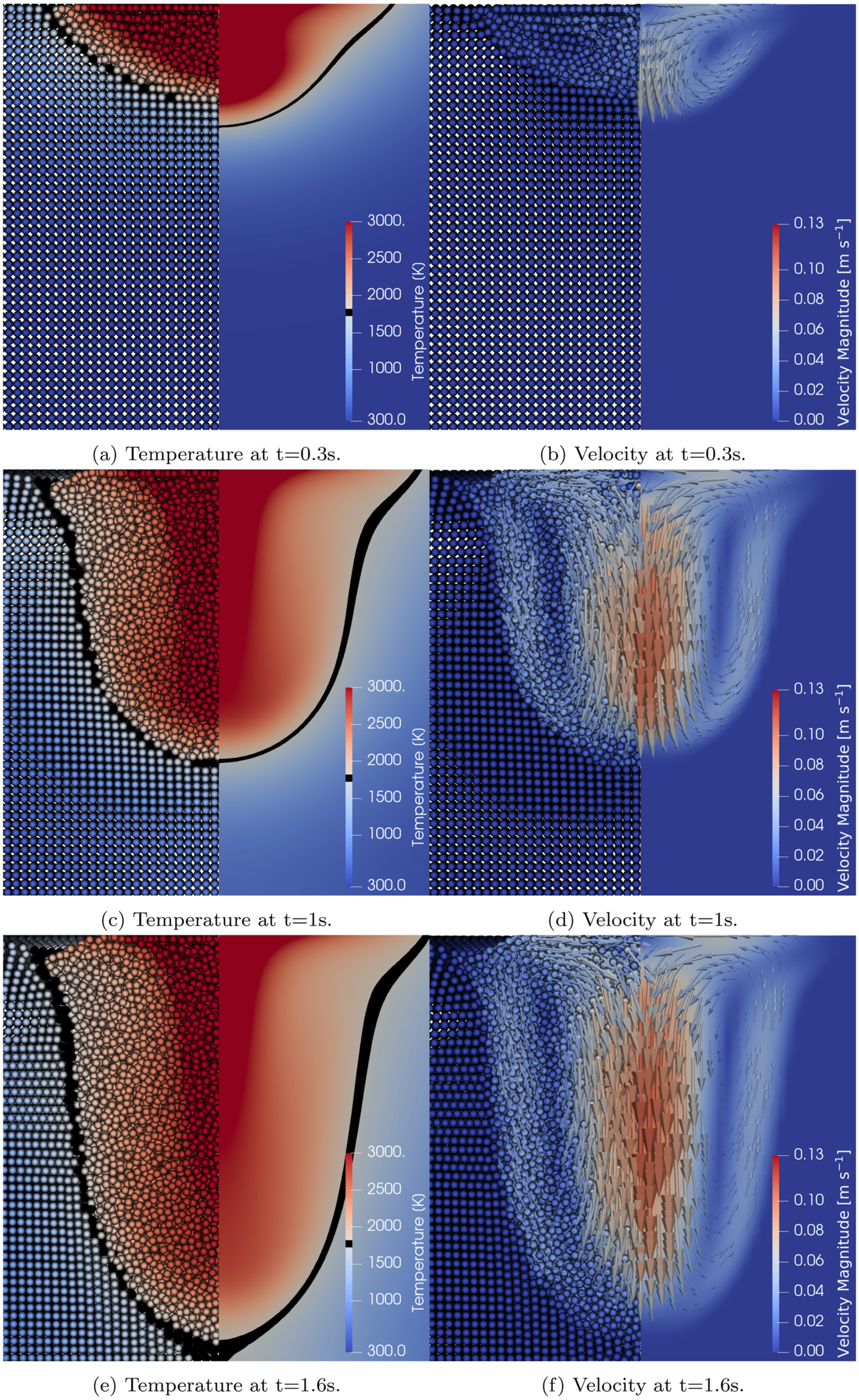
*Qualitative comparison*

Following the quantitative analysis, we will now show a qualitative comparison of the SPH and COMSOL solutions for the half of the cross section at times $t = 0.3$ s, $t = 1.0$ s and $t = 1.6$ s. These times were selected, as both $t = 0.3$ s as well as $t = 1.6$ s showed the greatest difference between the methods with regard to the melt-pool dimensions shown in Fig. 10. While the deviations were strongest for the comparison with the WM solution, the qualitative comparison will compare the SPH and the COMSOL solution where the agreement with of the melt-pool dimensions was best.

Over the whole duration, the time $t = 0.3$ s showed the larges disagreement between both methods, as can be seen also from the study of Figs. 10 and 11. While the disagreement is within the expected margin of error given by the comparison of the two FEM methods, looking in more detail at Fig. 14a, b, one can see that the melt-pool (as indicated by the melting temperature iso-countour as a black line) in the COMSOL simulation is slightly deeper than the melt-pool of the SPH simulation, while the melt-pool radius appears comparable. It is also possible to observe the distinct deviation in the velocity field between the two simulations, which is in line with the observation of the velocity in Fig. 12. A stable vortex seems to have already developed in the COMSOL simulation, while the SPH particles are barely in motion. We explain this by the very restrictive boundary condition in the SPH simulation. Until there is enough space for particles to overwhelm, or circumvent, the pressure forces keeping the fluid incompressible, the fluid will remain motionless. Since we are using an implicit pressure solver, the pressure forces will be quite large until a suitable large region of the fluid has melted, allowing for more movement while still maintaining incompressiblity. The difference in velocity also explains the deviation in melt-pool shape, since at $t = 0.3$ s the SPH simulation is still only transferring heat by conduction, while the melt-pool of the FEM simulation is increasing in depth due to an increase in convection.

Next, in Fig. 14c, d, we show the temperature and velocity distributions at $t = 1.0$ s for both SPH and COMSOL. Overall these results show excellent agreement, both in terms of the melt-pool dimensions (width and height) as well as the actual temperature and velocity field. The temperature fields appear almost identical, however, the melt-pool contour exhibits small deviations, especially close to the top boundary. In addition, the velocity magnitude of the SPH simulation at the symmetry axis is slightly lower than that

**Fig. 14** Side-by-by comparison of SPH and FEM results of temperature (left column) and velocity (right column). The SPH solution is always shown on the left within each figure, while the WM solution is shown on the right



(a) Temperature at t=0.3s.

(b) Velocity at t=0.3s.

(c) Temperature at t=1s.

(d) Velocity at t=1s.

(e) Temperature at t=1.6s.

(f) Velocity at t=1.6s.

of the COMSOL simulation. We are unsure about this deviation and can only speculate that due to the increased pressure forces (slight over-pressure) as mentioned above, the velocity slightly decelerated. Otherwise, there are no movement inhibiting forces acting on the fluid which are not shared across both simulations, i.e., viscosity and momentum-sink. Regardless of the difference in velocity magnitude, the flow pattern is, similar to the temperature field, almost identical.

Finally, considering the simulations at $t = 1.6$ s in Fig. 14e, f, we show the instance where full penetration is reached for both solvers, almost simultaneously, see also Fig. 10. One can see that the comparison of the temperature fields show very good agreement as well, where the general shape of the melting temperature iso-countour is captured accurately almost in the entire region, except for a small area close to the top surface. The velocity field also exhibits the same flow pattern as at $t = 1$ s, where the velocity magnitude is again slightly lower in the SPH simulation than in the COMSOL simulation.

Overall, we can observe very good agreement between the flow patterns and temperature distributions of both simulation methods. We speculate that the observed differences at $t = 0.3$ s results from restricted motion in SPH due to high pressure forces, which would also explain the diminished velocity magnitude and subsequent time steps, but could not confirm nor deny this speculation in a brief investigation. While this discrepancy warrants further study, we have shown successfully that the observed flow pattern and temperature field agree very well over a long duration and retain distinct similarities when the melt-pool depth differs.

## 4 Discussion

The comparison of the results from the SPH, WM and COMSOL solvers yields intriguing results. The simplifications strongly favor the application of an Eulerian method, and one of the main goals of this comparison was to investigate whether the SPH method is able to achieve good results in such a setting. Under consideration of the previous evaluation, we believe this to absolutely be the case, as we were able to observe very good quantitative agreement between the SPH and Comsol methods, while observing the same overall trends with the WM method.

Firstly, the conductive heat transfer was confirmed to work as expected, up to differences which can be accounted for by differences in spatial and temporal discretization. Secondly, the convective heat transfer seems to be in good agreement between SPH and COMSOL as could be seen for example by the melting temperature iso-contours of Fig. 9. The convective heat transfer in WM seemed to occur slightly faster. This is in line with the observations on the velocities, see Fig. 12, that seem to exhibit some deviation for all three methods of $\approx \pm 20\%$. In the light of these deviations, the excellent agreement for the melt pool depth at 0.6 s $< t <$ 1.3 s seems somewhat surprising. Figure 13 seems to confirm that the convective heat transfer in downward direction is slightly slower at all times of the SPH simulations as the temperature at the top, close to the heat source, is slightly higher. Nevertheless, the temperature profiles align almost perfectly for SPH and COMSOL toward the bottom of the melt pool, where the WM solution deviates significantly from both. The movement of the melt pool depth in SPH and COMSOL between 0.2 s $< t <$ 0.5 s, as shown in Fig. 10 seems especially intriguing, as both methods agree, while the WM solution shows a deviation. We can only assume that this deviation between the two FEM solvers is due to implementation-specific details.

Another observation relates to the fact that the SPH solution deviates from the other two mostly for earlier times, i.e., $t <$ 0.3 s, where the number of particles in the melt pool seems to be insufficient to allow for a development of a flow pattern, while at the same time, the implementation of a fixed boundary wall is not native to SPH, as SPH has its strength especially in the treatment of unbounded domains, and to a certain degree free fluid surfaces. The pressure forces in SPH might actually block the flow, while for WM the consideration of fixed wall boundary conditions is natural to the method, and no such restrictions exist when a free-slip condition is used.

It should be kept in mind however that for a real process, the free surface is not actually confined in such a way. Of course, for a comparison of melt pool development with the real process, also the nonlinearity of the material parameters must be considered. Here, the inclusion of the latent heat in terms of the Stefan Problem is known to present a significant challenge, among others. Nevertheless, even established Eulerian approaches have difficulty reaching satisfactory agreement without the use of strong compensation models and all the peculiarities that derive from the challenges related to the free surface deformation, discontinuities and strong topological changes that are typical for arc welding processes. As such, while the inclusion of latent heat in terms of the Stefan Problem is also nontrivial task for SPH methods, we consider our SPH model to be a suitable candidate for performing simulations of the more complex interactions and physics of the real process, due to the improved ability to model surface deformation, discontinuities and significant topological changes.

## 5 Conclusion

In this work, a full 3D SPH model for the simulation of a TIG spot-welding process was proposed and compared quantitatively with a 2 + 1D FEM simulation implemented in Wolfram Mathematica and COMSOL. We were able to show a remarkable agreement of our SPH method and the WM and COMSOL methods, especially given the differences that we were able to observe between the WM-FEM and the COMSOL-FEM solving entirely identical problems.

Overall, our work confirms the expectation that the SPH method represents an at least equivalent capability of quantitatively modeling welding processes. This holds even under conditions that are simplified in such a way to compensate for the limitations of the Eulerian methods (free-surface), while at the same time strongly favoring the application of a Eulerian method (bounded domain). Therefore, we expect that the SPH method can outperform the approach posed by Eulerian methods under more realistic conditions, i.e., allowing for a free surface, and requiring full 3D also for the Eulerian Method, especially once the numerical treatment of the non-linear material parameters has been realized.

**Acknowledgements** The presented investigations were carried out at RWTH Aachen University within the framework of the Collaborative Research Centre SFB1120-236616214 “Bau- teilpräzision durch Beherrschung von Schmelze und Erstarrung in Produktionsprozessen” and funded by the Deutsche Forschungsgemeinschaft e.V. (DFG, German Research Foundation). The sponsorship and support is gratefully acknowledged. The COMSOL and WM models were developed in project DFG-390246097, funded by the Deutsche Forschungsgemeinschaft e.V. (DFG, German Research Foundation). Simulations were performed in part with computing resources granted by RWTH Aachen University under project rwth0398.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

1. Akinci N, Ihmsen M, Akinci G, Solenthaler B, Teschner M (2012) Versatile rigid–fluid coupling for incompressible SPH. ACM Trans Gr 31(4):1–8. https://doi.org/10.1145/2185520.2335413.
2. Barecasco A, Terissa H, Naa CF (2013) Simple free-surface detection in two and three-dimensional sph solver
3. Becker M, Teschner M (2007) Weakly compressible SPH for free surface flows. In: ACM SIGGRAPH/Eurographics symposium on computer animation, pp 1–8. http://portal.acm.org/citation.cfm?id=1272690.1272719%5Cn
4. Bender J (2021) SPlisHSPlasH. https://github.com/InteractiveComputerGraphics/SPlisHSPlasH
5. Bender J, Koschier D (2015) Divergence-free smoothed particle hydrodynamics. In: ACM SIGGRAPH/Eurographics symposium on computer animation, pp 1–9
6. Bender J, Kugelstadt T, Weiler M, Koschier D (2019) Volume maps: an implicit boundary representation for sph. In: Motion, interaction and games. ACM, p 26
7. Brent AD, Voller VR, Reid KJ (1988) Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal. Numer Heat Transf 13(3):297–318. https://doi.org/10.1080/10407788808913615
8. Brookshaw L (1985) A method of calculating radiative heat diffusion in particle simulations. Publ Astron Soc Aust 6(2):207–210. https://doi.org/10.1017/s1323358000018117
9. Cadiou S, Courtois M, Carin M, Berckmans W, Le masson P (2020) 3d heat transfer, fluid flow and electromagnetic model for cold metal transfer wire arc additive manufacturing (cmt-waam). Addit Manuf 36:101541. https://doi.org/10.1016/j.addma.2020.101541
10. Cho WI, Na SJ (2021) Impact of driving forces on molten pool in gas metal arc welding. Weld World. https://doi.org/10.1007/s40194-021-01138-8
11. Das R, Cleary P (2016) Three-dimensional modelling of coupled flow dynamics, heat transfer and residual stress generation in arc welding processes using the mesh-free SPH method. J Comput Sci 16:200–216. https://doi.org/10.1016/j.jocs.2016.03.006
12. Davidson P (2001) An introduction to magnetohydrodynamics. Cambridge University Press, Cambridge
13. Gingold R, Monaghan J (1997) Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon Not R Astron Soc 181:375–389. https://doi.org/10.1093/mnras/181.3.375
14. Haidar J, Lowke J (1996) Predictions of metal droplet formation in arc welding. J Phys D Appl Phys 29(12):2951
15. Hertel M, Füssel U, Schnick M (2014) Numerical simulation of the plasma-mig process-interactions of the arcs, droplet detachment and weld pool formation. Weld World 58(1):85–92
16. Ito M, Izawa S, Fukunishi Y, Shigeta M (2012) Sph simulation of gas arc welding process. In: Proceedings of the seventh international conference on computational fluid dynamics (Hawaii, 2012), ICCFD7-3706 . https://iccfd.org/iccfd7/assets/pdf/papers/ICCFD7-3706_paper.pdf
17. Ito M, Nishio Y, Izawa S, Fukunishi Y, Shigeta M (2015) Numerical simulation of joining process in a TIG welding system using incompressible SPH method. Q J Jpn Weld Soc 33(2):34s–38s. https://doi.org/10.2207/qjjws.33.34s
18. Jeske SR, Bender J, Bobzin K, Heinemann H, Jasutyn K, Simon M, Mokrov O, Sharma R, Reisgen U (2022) Application and benchmark of sph for modeling the impact in thermal spraying. Comput Particle Mech
19. Komen H, Shigeta M, Tanaka M (2018) Numerical simulation of molten metal droplet transfer and weld pool convection during gas metal arc welding using incompressible smoothed particle hydrodynamics method. Int J Heat Mass Transf 121:978–985. https://doi.org/10.1016/j.ijheatmasstransfer.2018.01.059
20. Komen H, Tanaka M, Terasaki H (2020) Three-dimensional simulation of gas metal arc welding process using particle-grid hybrid method. Q J Jpn Weld Soc 38(2):25s–29s. https://doi.org/10.2207/qjjws.38.25s
21. Koschier D, Bender J, Solenthaler B, Teschner M (2019) Smoothed particle hydrodynamics techniques for the physics based sim-

ulation of fluids and solids. In: Eurographics 2019 tutorials. Eurographics Association

22. Kou S, Sun DK (1985) Fluid flow and weld penetration in stationary arc welds. Metall Trans A 16(2):203–213. https://doi.org/10.1007/bf02816047
23. Lucy LB (1977) A numerical approach to the testing of the fission hypothesis. Astron J 82:1013–1024. https://doi.org/10.1086/112164
24. Medale M, Rabier S, Xhaard C (2004) A thermo-hydraulic numerical model for high energy welding processes. Revue Européenne des Eléments 13(3–4):207–229
25. Medale M, Touvrey C, Fabbro R (2008) An axi-symmetric thermo-hydraulic model to better understand spot laser welding. European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 17(5–7):795–806
26. Mokrov O, Simon M, Shvartc I, Sharma R, Reisgen U (2021) Validation of the EDACC model for GMAW process simulation by weld pool dimension comparison. In: Lecture notes in mechanical engineering, pp 51–59. Springer. https://doi.org/10.1007/978-3-030-70332-5_5
27. Nguyen MC, Medale M, Asserin O, Gounand S, Gilles P (2017) Sensitivity to welding positions and parameters in gta welding with a 3d multiphysics numerical model. Numer Heat Transf Part A Appl 71(3):233–249
28. Price DJ (2012) Smoothed particle hydrodynamics and magnetohydrodynamics. J Comput Phys 231(3):759–794. https://doi.org/10.1016/j.jcp.2010.12.011
29. Schenk O, Gärtner K (2004) Solving unsymmetric sparse systems of linear equations with Pardiso. Future Gener Comput Syst 20(3):475–487
30. Semenov O, Demchenko V, Krivtsun, I, et al (2012) Modelling of the droplet formation process in GMA welding. In: Proceedings of the 10th international symposium on numerical analysis of weldability, pp 83–94
31. Semenov O, Krivtsun I, Lykhoshva A, Hluchenkyi O, Bondar O (2021) Comparative analysis of the results of computer simulation of heat transfer and hydrodynamic processes in the metal being welded by means of different software tools. Paton Weld J. https://doi.org/10.37434/as2021.01.04
32. Semyonov A (2014) Methods of mathematical modelling of the processes of electrode metal drop formation and transfer in consumable electrode welding. Paton Weld J 10:2–10
33. Trautmann M, Hertel M, Füssel U (2017) Numerical simulation of TIG weld pool dynamics using smoothed particle hydrodynamics. Int J Heat Mass Transf 115:842–853. https://doi.org/10.1016/j.ijheatmasstransfer.2017.08.060
34. Trautmann M, Hertel M, Füssel U (2018) Numerical simulation of weld pool dynamics using a SPH approach. Weld World 62(5):1013–1020. https://doi.org/10.1007/s40194-018-0615-5
35. Voller V, Brent A, Prakash C (1990) Modelling the mushy region in a binary alloy. Appl Math Model 14(6):320–326. https://doi.org/10.1016/0307-904x(90)90084-i
36. Weiler M, Koschier D, Brand M, Bender J (2018) A physically consistent implicit viscosity solver for sph fluids. CGF 37(2)
37. Zienkiewicz O, Taylor R, Nithiarasu P (2014) The finite element method for fluid dynamics. Butterworth-Heinemann, London
38. Zienkiewicz OC, Codina R (1995) A general algorithm for compressible and incompressible flow-part I. The split, characteristic-based scheme. Int J Numer Methods Fluids 20(8–9):869–885