

## IMPLEMENTATION OF A 3D SOLVER FOR ELECTRIC ARC WELDING, COUPLING FLUID MECHANICS WITH ELECTROMAGNETICS

Nilsson H.<sup>\*1</sup>, Choquet I.<sup>2</sup> and Sass-Tisovskaya M.<sup>2</sup>

<sup>\*</sup>Author for correspondence

1: Department of Applied Mechanics, Chalmers University of Technology, 412 96 Gothenburg, Sweden

2: Department of Engineering Science, University West, Trollhättan, Sweden

E-mail: hani@chalmers.se

### ABSTRACT

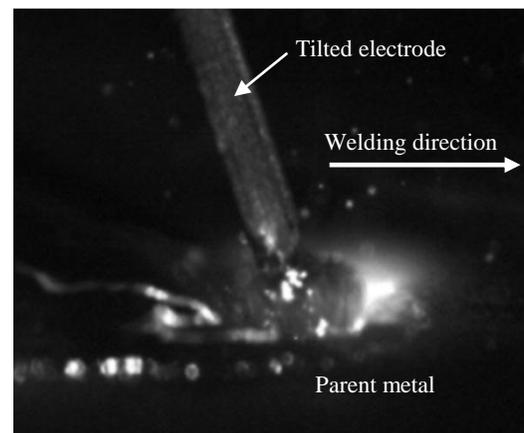
This paper describes the implementation of a 3D numerical solver for electric arc welding, where the fluid mechanics of the shielding gas is strongly influenced by the electromagnetic fields. The implementation is done in the OpenFOAM-1.6.x OpenSource Computational Fluid Dynamics (CFD) tool ([www.openfoam.com](http://www.openfoam.com)). OpenFOAM is basically a general library of C++ classes for numerical simulation of continuum mechanic problems, but it is mainly used in CFD. The basics of high-level programming in OpenFOAM is described briefly, while the main components of the implementation done in the present work are described in high detail. The implementation is validated against an analytical solution of the electromagnetic field of an infinite electrically conducting rod, and against an experimental study of GTAW (Gas Tungsten Arc Welding). The numerical results agree very well with both the analytical and experimental results. A grid-dependency study has been made for the GTAW case, showing that the main features of the presented solutions are independent of the mesh size.

### INTRODUCTION

Welding is a manufacturing process developed to join materials, most commonly metals. It is based on a heat source used to melt a work piece, and form a molten pool. A filler material is often added to fill in the pool. During cooling, the pool becomes a strong joint. It is important to control the process as much as possible, so that the final construction becomes strong, the shape of the welding region becomes as desired, and that the welding process is efficient. This is important both from economical and environmental sustainability points of views.

The present work focuses on electric discharges (arcs) as a heat source. The final aim is to numerically predict the most important features of electric arcs used in practice for welding. Most of the standard numerical methods used in studies of welding assume that the configuration is fully axi-symmetric. This is the case for the documented reference test cases

available in the literature. These test cases are made using water cooled parent metal, tungsten electrode, no added filler material and steady process. Those conditions allow doing spectroscopic measurements and interpreting the results so as to determine the arc temperature. Such experimental data are important for testing and validating the simulation models.



**Figure 1** Picture of electric arc welding with moving tilted electrode, metal transfer across the arc and metal provided by the electrode.

However, the electric arc configurations used in practice for welding are not axi-symmetric, see Fig. 1. First of all because the electrode (or instead the parent metal) is in motion. Most of the time, the electrode is also tilted. When welding with a tungsten electrode, filler material often needs to be added. The metal rod used for providing the filler material is also a source of non axi-symmetry. If instead the filler material is provided by the electrode itself, as in Fig. 1, the metal transfer across the arc is anyway usually not axi-symmetric.

This work presents the implementation of a 3D solver, which has the purpose of analysing arc welding processes used in production. The electromagnetic part of the solver is

validated against analytical results of an electrically conducting rod and its surrounding. The coupling between the electromagnetism and the shielding gas flow, which occurs through Joule heating in the energy equation and modification of temperature dependent fluid properties, is validated against an experimental study of GTAW (Gas Tungsten Arc Welding).

The implementation is done in the OpenFOAM-1.6.x [1] OpenSource CFD tool (www.openfoam.com). OpenFOAM is basically a library of C++ classes for continuum mechanic problems. It is mainly used for CFD, and it provides a large number of examples of how the classes can be used to solve problems in heat transfer, fluid mechanics, and MHD (Magneto Hydro Dynamics). The principle of how the OpenFOAM classes are used to build a solver for continuums mechanics is described, with a focus on the implementation done in the present work.

### HIGH-LEVEL PROGRAMMING IN OPENFOAM

OpenFOAM is a C++ library of object-oriented classes that can be used for implementing solvers for continuum mechanics. OpenFOAM was distributed as OpenSource in 2004, and has since then grown in popularity and is now a true competitor to proprietary alternatives. OpenFOAM is distributed with a number of solvers for different continuum mechanical problems, but due to the availability of the source code, its libraries can be used to implement new solvers for other applications. Here, a brief introduction to high-level programming in OpenFOAM is presented, together with a short discussion on object orientation. Text that is related to implementation is surrounded by boxes and in different font.

OpenFOAM is highly object oriented, meaning that the focus is on objects rather than the functions. Consider e.g. a tensor as an object, constructed as tensor T, where T is the object and tensor is the class that defines a tensor. The operators that can operate on the tensor are functions that are members of (belong to) the tensor class (thereby called *member functions*). The values of the tensor are also members of the tensor class (thereby called *member data*). It is thus the tensor class that holds all the member functions and member data, and an object that belongs to that class can use that functionality. This is beneficial since a class and its member data/functions can be very well defined and debugged for its specific purposes. C++ and object orientation also provides functionality for inheriting functionality from sub-classes, so that more complex classes can be built from simpler debugged classes. An example is a tensor field, which is simply a field of tensors where the functionality of each tensor is untouched, but additional functionality for fields of tensors is added. Discretized partial differential equations from continuum mechanics are simply tensor fields that belong to a mesh and use discretization schemes to determine the corresponding linear system. The functions and operators of the equations thus belong to classes that determine how the linear system should be assembled and solved. An example is the Laplace equation, given by  $\nabla \cdot [\Gamma \nabla \phi] = 0$ , where the equation is discretized to its linear form in the finite volume method class, `fvm`, using the high-level OpenFOAM code `fvm::laplacian(gamma,phi)`, and solved using a function that recognizes the type of output from the Laplacian

function as `solve(fvm::laplacian(gamma,phi)`. The right-hand side of the equation is here omitted, and is thus automatically treated as zero. The discretization schemes and linear solver methods are implemented so that they must be chosen when running the code, rather than hard-coding those choices into the solver. If gamma is a known scalar or scalar field, this is all it takes to get a solution for phi. There is also a finite volume calculus class, `fv`, that is used to explicitly calculate values or fields for the linear system rather than constructing entries in the coefficient matrix.

Except for regular functions, it is also possible in C++ to define the functionality of a large number of operators, such as +, -, \*, =, ^, &, etc. This means that a specific implementation of the operator will be used, depending on the type of the operands involved in the operation. E.g. the cross-product between two vector fields A and B is written as A^B, and the dot product is written as A&B. In both cases the operations are performed for each pair of vectors in the two fields that must thus be of the same size.

In the following section, some details of the implementation of the new solver for 3D electric discharge modeling is discussed.

### GOVERNING EQUATIONS AND IMPLEMENTATION

Numerical electric discharge modeling requires the solution of the electromagnetic equations (Maxwell), the fluid flow equations (Navier-Stokes), and the energy equation. The current implementation, that has been done in OpenFOAM-1.6.x, is based on the buoyantSimpleFoam solver, which is a steady-state solver for buoyant, turbulent flow of compressible fluids. Only the differences between the implementations of the original and the new solvers are discussed in detail below, since buoyantSimpleFoam is a standard OpenFOAM solver. Again, text that is related to the implementation is surrounded by boxes and in different font.

The assumptions made for the electromagnetic part are [2]

- The Debye length,  $\lambda_D$ , is much smaller than the characteristic length of the welding arc, so that there is local electro-neutrality in the plasma core.
- The characteristic time and length of the welding arc allow neglecting the convection current in Ampere's law, resulting in quasi-steady electromagnetic phenomena.
- The Larmor frequency is much smaller than the average collision frequency of electrons, implying a negligible Hall current compared to the conduction current.
- The magnetic Reynolds number is much smaller than unity, leading to a negligible induction current compared to the conduction current.

Thus, the electric potential,  $V$  Elpot, is given by a Laplace equation as

$$\nabla \cdot [\sigma(T) \nabla V] = 0,$$

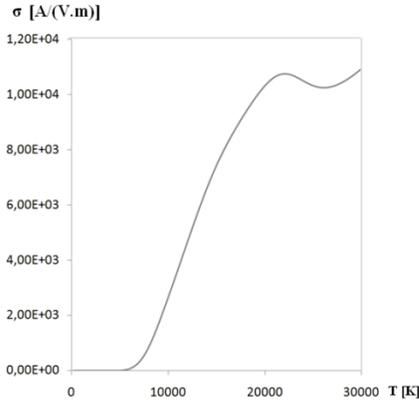
`solve(fvm::laplacian(sigmaMag, Elpot);`

and the magnetic potential,  $\vec{A}$  A, is given by a Poisson equation as

$$\nabla^2 \vec{A} = \sigma(T) \mu_0 \nabla V.$$

```
solve(fvm::laplacian(A) == sigmaMag*muVac*(fvc::grad(Elpot));
```

Here,  $\mu_0$  `[muVac]` is the permeability of vacuum,  $\mu_0 = 4\pi \times 10^{-7} Tm A^{-1}$ . Figure 2 shows that the electric conductivity,  $\sigma(T)$  `[sigmaMag]`, is highly temperature dependent in the range 200-30000K for argon plasma, and is updated every iteration using a linear interpolation of values derived from kinetic theory [3]. The interpolation has been implemented in a thermophysical class in OpenFOAM, which is not described in detail here.



**Figure 2** Electric conductivity of argon gas over a wide range of temperatures.

From the electric potential, the electric field,  $\vec{E}$  `[E]`, and the electric current density,  $\vec{J}$  `[Je]` (from Ohm's law), are derived as

$$\vec{E} = -\nabla V$$

$$\vec{J} = -\sigma(T) \nabla V = \sigma(T) \vec{E}.$$

```
E = -fvc::grad(Elpot);
Je = sigmaMag*E;
```

The magnetic field,  $\vec{B}$  `[B]`, can be derived from the magnetic potential as

$$\vec{B} = \nabla \times \vec{A}.$$

```
B = fvc::curl(A);
```

Finally, the Lorentz force that will be added to the momentum equation is given by

$$\vec{F}_{Lorentz} = \vec{J} \times \vec{B},$$

```
Florenz = Je^B;
```

and the Joule heating that will be added to the energy equation is given by

$$S_{joule} = \vec{J} \cdot \vec{E}$$

```
Sjoule = Je&E;
```

The assumptions made for the fluid mechanic part are [2]

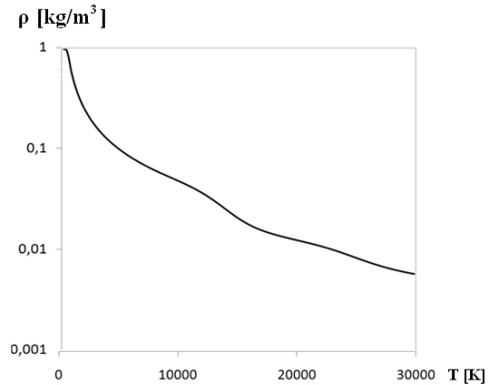
- The flow is steady-state.
- No phase-change is taken into account, i.e. single-phase flow. Inclusion of the plasma sheath would require a two-fluid method.
- There is local thermal equilibrium.
- The fluid is mechanically incompressible because of the small Mach number, but thermally expansible. I.e. the density depends on the temperature, but not on the pressure. The original buoyantSimpleFoam solver has been slightly modified for this purpose.
- The fluid is Newtonian.
- The flow is laminar, which is specified using the option of laminar flow in the OpenFOAM turbulence model class.

Thus, the continuity equation is given by

$$\nabla \cdot [\rho(T) \vec{u}] = 0,$$

which is satisfied through a `[SIMPLE pressure-velocity coupling]` method that is already available in buoyantSimpleFoam.

Figure 3 shows the temperature dependence of the density,  $\rho(T)$ , of argon gas over the required temperature range [3].



**Figure 3** Density of argon gas over a wide range of temperatures.

The momentum equation is given by

$$\begin{aligned} \nabla \cdot [\rho(T) \vec{u} \vec{u}] - \vec{u} \nabla \cdot [\rho(T) \vec{u}] \\ - \nabla \cdot \left[ \mu(T) (\nabla \vec{u} + (\nabla \vec{u})^T) - \frac{2}{3} \mu(T) (\nabla \cdot \vec{u}) I \right] \\ = -\nabla P + \rho(T) g + \vec{J} \times \vec{B}, \end{aligned}$$

where the last term is the Lorentz force `[Florenz]`, described above, which is the only modification to the original buoyantSimpleFoam momentum equation.

Notice that the gravitational force per unit volume,  $\rho(T)g$ , could be neglected in the present context, but it will become important in the forthcoming developments with metal transfer across the plasma arc.

The conservation of energy is formulated as an enthalpy equation, as

$$\nabla \cdot [\rho(T)\vec{u}h] - h\nabla \cdot [\rho(T)\vec{u}] - \nabla \cdot [\alpha(T)\nabla h] - \nabla \cdot \left[ \frac{5k_B\vec{j}}{2eC_p(T)}h \right] = \nabla \cdot (\vec{u}P) - P\nabla \cdot \vec{u} + \vec{j} \cdot \vec{E} - Q_{rad},$$

where  $h$  is the specific enthalpy,  $\alpha$  is the thermal diffusivity,  $k_B = 1.380658 \times 10^{-23} JK^{-1}$  is the Boltzmann constant,  $e = 1.60217733 \times 10^{-19} coulombs$  is the elementary charge of an electron, and  $C_p(T)$  is the temperature dependent specific heat at constant pressure. The term  $\vec{j} \cdot \vec{E}$  [Sjoule], described above, is the Joule heating. The last term on the left-hand side is the transport of electron enthalpy, implemented into the coefficient matrix as

`-fv::div((linearInterpolate(kBe*pow(Cp,-1)*Je)&mesh.Sf()),h)`, where  $kBE = (5*kB)/(2*e)$ . The temperature dependent radiation heat loss  $Q_{rad}$  [4], as well as the thermodynamic and transport properties [3], are linearly interpolated from tabulated values in the range 200-30000K. The interpolation has been implemented in a thermophysical class in OpenFOAM, which is not described in detail here. It is thus only the last term on the left hand side, and the two last terms on the right hand side that are added to the original implementation in buoyantSimpleFoam. The temperature is derived from the enthalpy using the definition of specific heat, i.e.

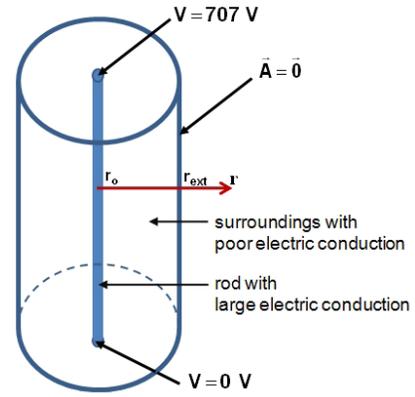
$$C_p(T) = \left( \frac{dh}{dT} \right)_p.$$

## VALIDATION OF THE IMPLEMENTATION

In the following sub-sections, the new implementation is validated against the analytical solution of the electromagnetics of an electrically conducting cylindrical rod, and experimental results of a water cooled GTAW (Gas Tungsten Arc Welding) case.

### Electrically conducting cylindrical rod

Under the assumptions discussed in the description of the implementation, the electromagnetic equations are independent of the flow solution except for the temperature dependent electric conductivity. Thus, the electromagnetic part of the implementation can be validated without solving the Navier-Stokes and energy equations, but with given distributions of the electric conductivity. The first validation test case is thus the electromagnetics of an electrically conducting cylindrical rod. Figure 4 shows the case set-up and boundary conditions used.



**Figure 4** Case description for the electrically conducting cylindrical rod.

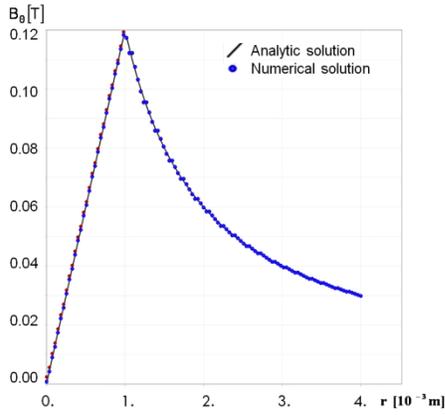
The rod radius is  $r_0 = 1mm$  and the radius of the computational domain is  $r_{ext} = 16mm$ . The length of the rod is 10mm. Due to the axi-symmetry of the geometry and the solution, the simulation is performed using a 2D axi-symmetric approach. In OpenFOAM, that is done using a wedge sector of an angle of 5 degrees, with one control volume in the tangential direction and centered about the X-Y-plane, and a symmetry axis along the x-axis. The full 3D equations are solved, but the symmetry is taken into account in the discretization process. The electric potential,  $V$ , was set to 707V and 0V at the ends of the rod, corresponding to a current intensity of 600A, and zero normal gradient at all other boundaries. The magnetic potential,  $\vec{A}$ , was set to zero at  $r = r_{ext}$ , and zero normal gradient at all other boundaries. The hexahedral equidistant mesh consists of 50 cells along the rod, 100 radial cells in the rod, and 320 radial cells in the surrounding. The electric conductivity,  $\sigma$ , is chosen to correspond to argon plasma at 10600K in the rod, and 300K in the surrounding, i.e.  $\sigma_{rod} = 2700A(Vm)^{-1}$  and  $\sigma_{sur} = 10^{-5}A(Vm)^{-1}$ .

The analytic solution for the magnetic field in and around an infinite electrically conducting cylindrical rod is given by

$$B_{\theta}(r) = \frac{\mu_0 J_{axial} r}{2}, \quad \text{if } r < r_0$$

$$B_{\theta}(r) = \frac{\mu_0 J_{axial} r_0^2}{2r}, \quad \text{if } r \geq r_0$$

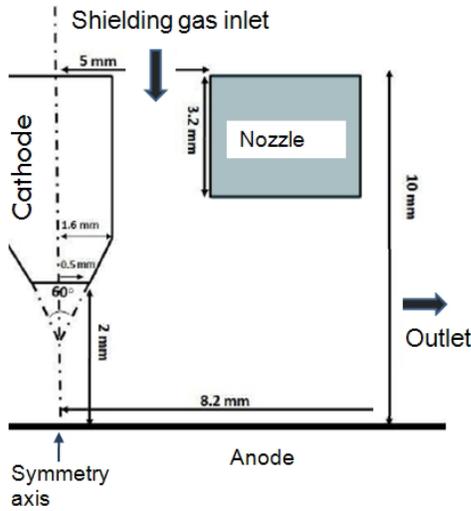
where  $J_{axial} = I/(\pi r_0)$  is the current density along the rod axis and  $I$  is the current intensity. Figure 5 shows a comparison between the analytical and numerical results, and the results are identical.



**Figure 5** Radial distribution of the angular component of the magnetic field, for the electrically conducting cylindrical rod.

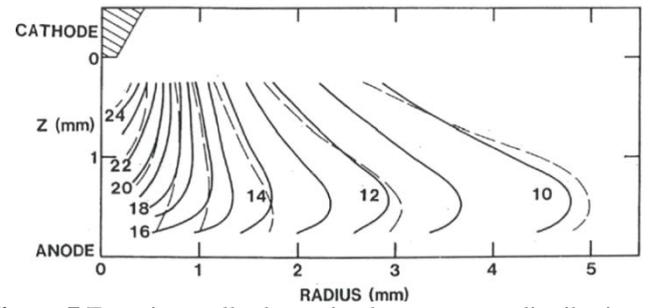
### Water cooled GTAW

A water cooled GTAW (Gas Tungsten Arc Welding) test case with experimental results [5] is used to validate the full solver, including both electromagnetics and fluid mechanics. Figure 6 shows a schematic representation of the test case. As for the rod case, due to the axi-symmetry of the geometry and the solution, the simulation is performed using a 2D axi-symmetric approach. In OpenFOAM, that is done using a wedge sector of an angle of 5 degrees, with one control volume in the tangential direction and centered about the X-Y-plane, and a symmetry axis along the x-axis. The full 3D equations are solved, but the symmetry is taken into account in the discretization process.



**Figure 6** Schematic description of the GTAW case.

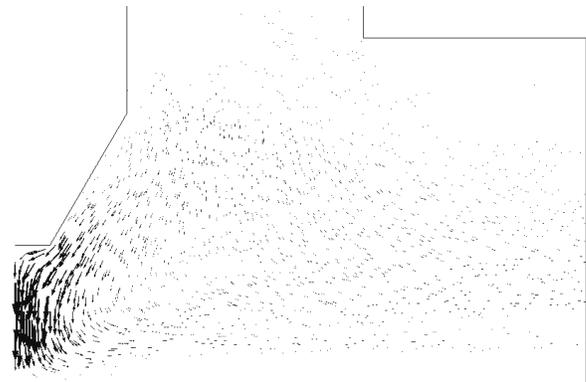
The argon shielding gas enters the nozzle at  $T = 300K$ , with a mass flow rate of  $\dot{m} = 1.66 \times 10^{-4} kg/s$ , and a parabolic velocity profile. At the outlet, at  $r = 8.2mm$ , the normal gradient of temperature and velocity is set to zero. At the walls, a no-slip condition is used for the velocity, and extrapolated values from the experiments, shown in Figure 7, are used for the temperature at the anode boundary.



**Figure 7** Experimentally determined temperature distribution.

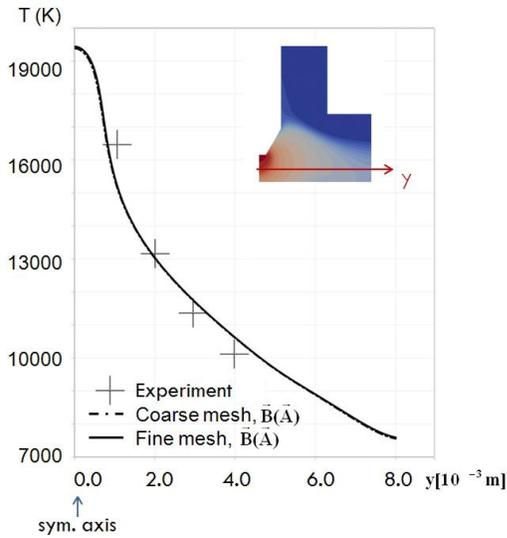
Explicitly specified current density and temperature distributions are set at the cathode boundary [6]. The cathode tip temperature is  $T = 20000K$ , the cylindrical cathode boundary temperature is  $T = 300K$ , and at the conical part of the cathode the temperature is linearly distributed between  $300K$  and  $20000K$ . The fixed current density boundary condition yields a fixed gradient boundary condition for the electric potential as  $\frac{\partial V}{\partial n} = -6755.1V/m$  at the tip, a rapid linear reduction of this gradient down to zero close to the tip along the conical section, and zero normal gradient along the rest of the cathode. At the anode boundary,  $V = 0$ , and a zero normal gradient is used everywhere else. The boundary conditions for the magnetic potential,  $\vec{A}$ , are set to zero normal gradient at all boundaries except at the outlet, where it is set to  $\vec{A} = \vec{0}$ . Two different fully equidistant hexahedral meshes have been used in the present study. The coarse and fine meshes had 25 (resp. 100) cells along the 0.5 mm tip radius, and 100 (resp. 200) cells between the electrodes along the symmetry axis. The coarse and fine meshes consist of 136250 and 192500 control volumes respectively.

Figure 8 shows a view of the velocity distribution affected by the electromagnetic field. Although there is a small inlet velocity, the velocity distribution is completely determined by the electromagnetic source terms that have been introduced in the momentum and energy equations. The maximum velocity is approximately 87m/s, while the maximum inlet velocity (parabolic profile) is approximately 2.9m/s.



**Figure 8** Velocity vectors, showing that the inlet velocity is negligible in comparison with the velocities that are driven by the electromagnetic forces.

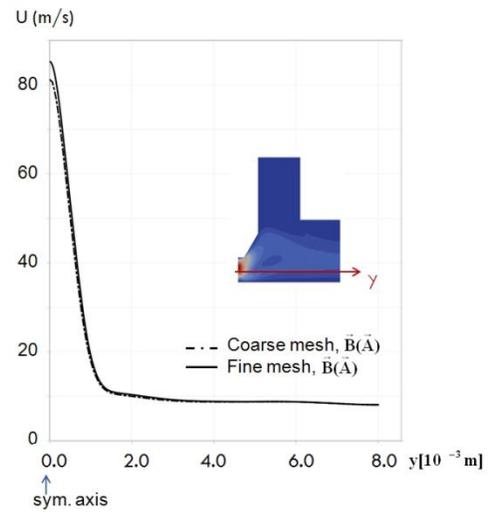
Figure 9 compares the temperature distributions at mid-height between the cathode and anode from the simulations with the two meshes, and the experimental results. The numerical results are identical from the two meshes, and they correspond well to the experimental results.  $\vec{B}(\vec{A})$  here refers to the 3D method described in the present work, distinguishing it from other methods where the magnetic field  $\vec{B}$  is determined by a simplified axi-symmetric approach introduced for long arcs assuming that the current density is invariant by translation along the symmetry axis [7]. This simplified model can not be used for this GTAW case with short arc; see [8] for further details.



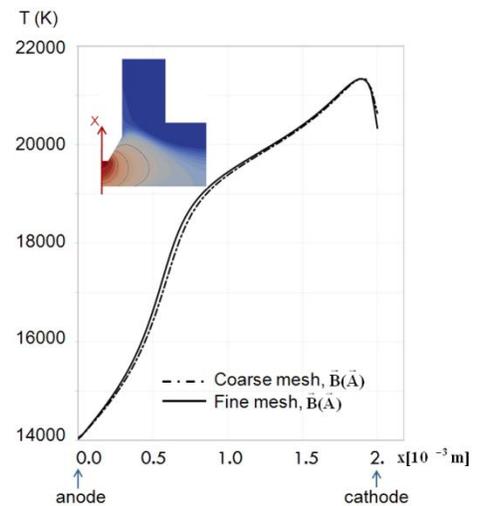
**Figure 9** Temperature distribution at mid-height between cathode and anode.

Figure 10 compares the numerically predicted velocity distributions at mid-height between the cathode and anode. It can be concluded that both the temperature and velocity distributions are quite independent of the mesh resolution except for the region very close to the symmetry axis.

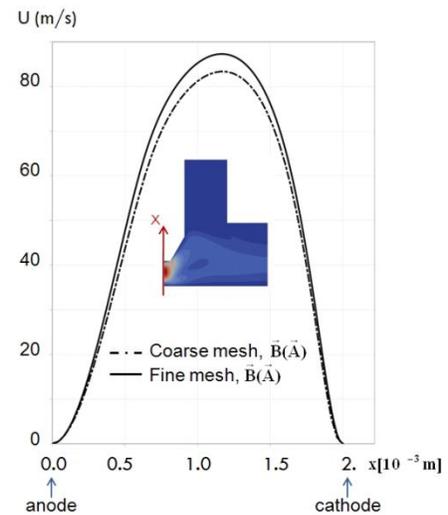
Figures 11 and 12 compares the results from the two meshes along the symmetry axis, showing that the largest difference occurs for the velocity at mid-height between the cathode and anode, as also suggested by Figures 9 and 10.



**Figure 10** Velocity magnitude distribution at mid-height between cathode and anode.



**Figure 11** Temperature distribution along the symmetry axis between the cathode and anode.



**Figure 12** Velocity magnitude distribution along the symmetry axis between the cathode and anode.

## FUTURE WORK

In on-going work, [8], the sensitivity to different choices of boundary conditions is studied and the preliminary results show that the choice of boundary condition at the electrodes can greatly affect the numerical results. Therefore, future work will move the boundary conditions further away from the region of interest by including the electromagnetic and temperature fields also in the electrodes. That requires a coupled solver that can solve the electromagnetic and temperature fields in the entire domain, and the fluid mechanic equations only in a part of the domain.

In the present work the method has been applied to an electric rod and an axi-symmetric Gas Tungsten Arc Welding case. The aim of the development is however to use the implementation for arc welding used in production, requiring fully 3D simulations. For a 3D case, the region between the electrodes should not be meshed using wedge-shaped control volumes, since such may influence the numerical results.

Further steps are to include the melting and transportation of filler material, and weld pool formation, requiring a multi phase approach.

## CONCLUSION

The implementation of a solver, using the OpenFOAM-1.6.x libraries, for 3D simulations of electric discharge welding has been described. The simulations presented show that the basic features of electric discharge welding can be accurately predicted. The electromagnetic part of the solver has been validated against the electromagnetic fields of an electric rod, while the full solver has been validated against a Gas Tungsten Arc Welding case. A grid study shows that the numerical results are independent of the mesh size except at the symmetry axis, where the results differ slightly. A possible reason for the deviation is the use of a wedge shaped computational domain, yielding control volumes of poor quality at the symmetry axis.

## ACKNOWLEDGEMENTS

This work was supported by the KK-foundation in collaboration with ESAB, Volvo Construction Equipment and SSAB. Håkan Nilsson was also in this work financed by the Sustainable Production Initiative and the Production Area of Advance at Chalmers. Their support is gratefully acknowledged.

Prof. Jacques Auberton and Prof. Marie-Françoise Elchinger are greatly acknowledged for providing the data tables of thermodynamic and transport properties.

## REFERENCES

- [1] Weller H.G., Tabor G., Jasak H., and Fureby C., A Tensorial Approach to Computational Continuum Mechanics Using Object Oriented Techniques, *Computers in Physics*, 1998, Vol 12, No.6, pp.620-631. Doi: 10.1063/1.168744
- [2] Sass-Tisovskaya M., Plasma Arc Welding Simulation with OpenFOAM, *Thesis for Licentiate of Engineering no.2009:10*, Chalmers University of Technology, Gothenburg, Sweden, 2009, ISSN 1652-8565

- [3] Rat V., Pascal A., Auberton J., Elchinger M.F., Fauchais P., and Lefort A., Transport properties in a two-temperature plasma: theory and application, *Physical Review E*, **64**, 2

- [4] Delalondre C., Modélisation aérothermodynamique d'arcs électriques à forte intensité avec prise en compte du déséquilibre thermodynamique local et du transfert thermique à la cathode, *PhD. Thesis, Rouen University, 1990.*

- [5] Haddad G.N., and Farmer A.J.D., Temperature measurements in gas tungsten arcs. *Welding Journal*, **24**, pp.339-342.

- [6] Tsai M.C., and Sindo K., Heat transfer and fluid flow in welding arcs produced by sharpened and flat electrodes, *Int. J. Heat and Mass Transfer*, **33**, 10, pp. 2089-2098.

- [7] Hsu K.C., Etemadi K., and Pfender E., Study of the free-burning high intensity argon arc, *J. Appl. Phys.*, **54** (3), pp. 1293-1301.

- [8] Choquet I., Nilsson H., and Sass-Tisovskaya M., Modeling and simulation of a heat source in electric welding, *In Proc. Swedish Production Symposium, SPS11*, May 3-5, 2011, Lund, Sweden. To appear.