Proceedings of the 7th GACM Colloquium on Computational Mechanics
for Young Scientists from Academia and Industry
October 11-13, 2017 in Stuttgart, Germany

Boundary-Conforming Space-Time Finite Elements for Co-Rotating Intermeshing Domains

Jan Helmig[1]*, Marek Behr[1] and Stefanie Elgeti[1]

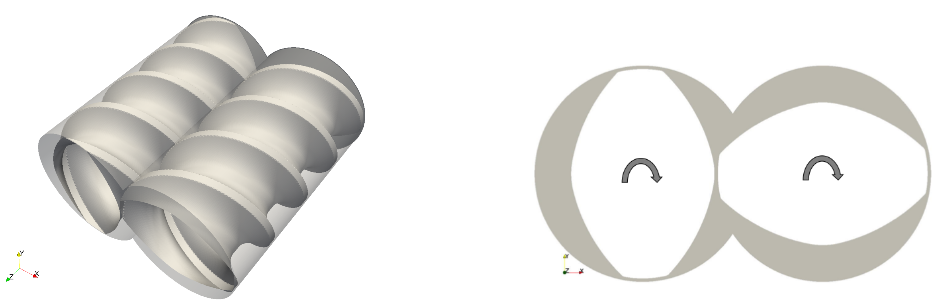
Micro Abstract
Using boundary-conforming methods for co-rotating intermeshing domains is a complex task because the mesh has to be updated in every time step. Traditional methods like the elastic mesh update method require constant re-meshing due to mesh failure. We present a new, efficient approach, the Snapping Reference Mesh Update Method (SRMUM). It is based on a background mesh that constantly adapts to the current geometry. We apply the method to flow computations of plastic melt in twin-screw extruders.

[1]Chair for Computational Analysis of Technical Systems, RWTH Aachen University, Aachen, Germany
***Corresponding author**: helmig@cats.rwth-aachen.de

Introduction

Numerical analysis of flow inside co-rotating intermeshing domains like twin-screw extruders (TSE) is very complex. This is due to small gap widths in addition to constantly rotating, intermeshing surfaces, which makes meshing extremely challenging. An example of a conveying screw element of a TSE is shown in Figure 1a and its 2D cross section in Figure 1b.

In presence of constantly changing computational domains, one can distingish between two numerical approaches that can model the moving boundaries and interfaces, namely interface tracking and interface capturing [1]. For interface capturing or fictitious domain methods (FDM) an implicit description of the moving boundary is used on a constant background mesh. This leads to a very flexible method that does not require intensive re-meshing. However, this kind of methods lack accuracy for co-rotating intermeshing domains with small gap widths [2]. Therefore, specialized interface capturing methods like XFEM [2] have to be employed in order to obtain accurate results. A further disadvantage of FDM is, that the number of degrees of freedom is not constant throughout the simulation which makes load balancing more complex for massive parallel computations.



(a) 3D conveying screw element **(b)** 2D cross section

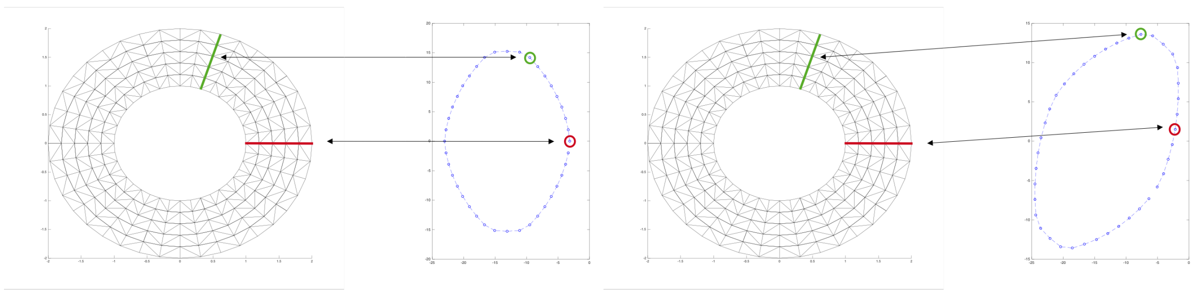
Figure 1. Sketch of a screw element of a twin-screw extruder as an example for a co-rotating intermeshing domain.

For interface tracking methods the computational mesh follows the moving boundary, which makes them boundary-conforming. For deforming domains, this requires a constant update of the computational mesh. It can be achieved by updating the interior nodes based on the movement of the boundary. A commonly used method is the Elastic Mesh Update Method (EMUM) that treats the mesh as an elastic body [4]. However, for TSE applications this method results in mesh tangling after few time steps due to the complex movement of the screws. This makes re-meshing necessary which can be very time-consuming and tedious for complex domains.

Therefore, we present a new efficient mesh update method that allows to use boundary-conforming meshes in the context of co-rotating intermeshing domains without constant re-meshing. It is based on a reference mesh that constantly adapts (snaps) to the moving domain - thus, we refer to it as Snapping Reference Mesh Update Method (SRMUM). Only algebraic operation have to be performed, which allows to update the mesh at a fraction of the cost of the flow solution. In order to naturally account for the moving domain the method is embedded into the Deforming-Spatial Domain / Stabilized Space-Time (DSD/SST) finite element framework [3].

1 Snapping Reference Mesh Update Method

In the following we present the basic ideas for the Snapping Reference Mesh Update Method. As a starting point, we consider a single convex rotating surface inside one barrel. We define a structured background mesh between two circles with n beeing the number of elements in circular direction. The rotating surface is also discretized into n equally spaced elements, where every surface point belongs to an ID going from 1 to n. According to the naming of the method, we want the background mesh to snap onto the rotating surface. Therefore, we have to establish a connection between background mesh and surface discretization. Thus, every node of the background mesh is associated to an ID that is computed as follows: $ID = \alpha/360° * n$, where α is the angle enclosed by the line from the circle origin to the node. Thus, all mesh points lying on one line in normal direction have the same ID. This can be observed in Figure 2a. In order to account for the rotation of the surface the IDs of the surface have to be updated. Whenever the surface rotates more than $\theta = 360°/n$ the IDs of the screw surface have to be shifted, whereas the IDs of the background mesh are constant over time, see Figure 2b.



(a) Initial configuration **(b)** Configuration after some rotation

Figure 2. SRMUM background mesh and screw surface discretization at two different screw orientation.

In addition to the ID, every node needs to have information about its relative position between the inner and outer circle. We denote this as x_{rel}. It is zero for points on the inner circle and one on the outer one. Based on this the position of every node at any point in time can be determined by

$$\mathbf{x}\left(ID\right) = \mathbf{x}_{surf}\left(ID\right) + x_{rel} * \left[\mathbf{x}_{barrel}\left(ID\right) - \mathbf{x}_{surf}\left(ID\right)\right]. \tag{1}$$

We extend the method to co-rotating intermeshing domains, as shown in Figure 1b. Therefore, two circular background meshes are connected and duplicated nodes on the connected interface

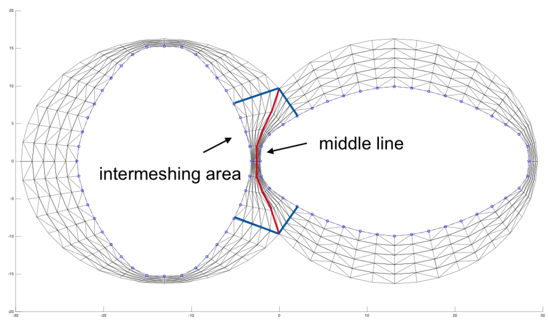


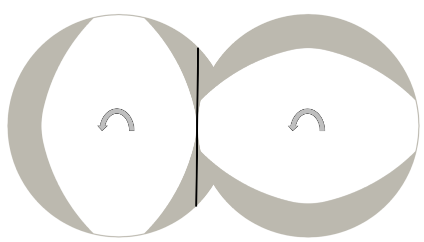
Figure 3. Sketch of the intermeshing area and the middle line for 2D SRMUM.

are eliminated. The update of nodes outside the intermeshing area is the same as for the single surface case. For points inside the intermeshing area the corresponding barrel position is not defined, such that Equation 1 cannot be applied. As a substitute for the barrel position $\mathbf{x}_{barrel}(ID)$, we define a middle line inside this area, see Figure 3. Points on the middle line $\mathbf{x}_{middle\_line}(ID, \mathbf{x}_{surf}(ID))$ are computed based on relations between the corresponding screw positions.

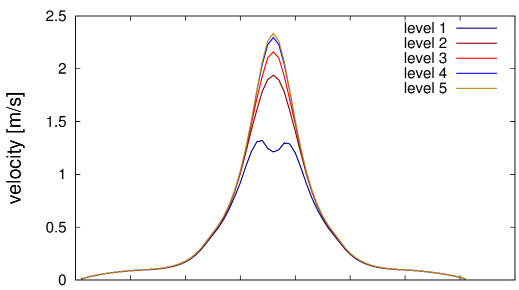
Extending SRMUM to 3D is straight forward. The longitudinal direction (z-axis) can be interpreted as a 2D space-time surface rotating in time. We simply connect 2D meshes in the xy-plane along the z-axis. This is simple due to the well defined structure of the background mesh. The resulting prisms are split into tetrahedrons.

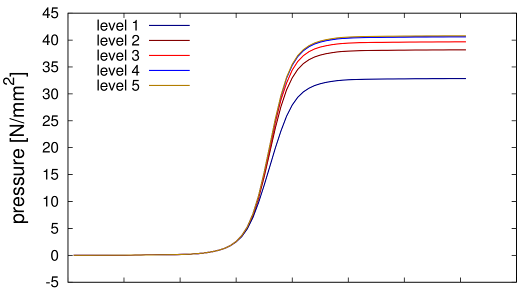
2 Numerical Results

2.1 2D Convergence Study



(a) Screw configuration and line used for plotting.





(b) Velocity

(c) Pressure

Figure 4. Pressure and velocity plot of a 2D TSE flow computation on 5 different computational meshes along a vertical line.

In order to validate the SRMUM we use a testcase that computes the flow inside a 2D TSE configuration presented in [2]. The configuration is shown in Figure 4a. The plastic melt is modeled using stokes flow and in addition to that, the Yasuda model is employed to account for shear-thinning effects. The properties for comparison are pressure and velocity values along a line through the intermeshing area. In order to show convergence of our method we compute the flow solution on 5 different mesh levels generated via SRMUM. The number of elements of the screw discretization varies from 1000 to 4000. This results in a total number of elements from 120 000 to 700 000. The results are shown in Figure 4b and 4c. Mesh convergence can be shown for velocity and pressure. Furthermore, the results match those shown in [2].

2.2 3D Temperature-Dependent Flow of Plastic Melt in TSE

In the following, we compute the velocity and temperature distribution of plastic melt inside a 3D TSE with forward conveying screw elements using SRMUM. Viscosity is not only dependent on the flow field but also on temperature through a WLF model, s.t. the heat equation is not decoupled from the energy equation anymore, which makes the system highly nonlinear. The plastic melt heats up due to viscous heating effects resulting from high shear rates in the gap regions. This effect is shown in Figure 5.

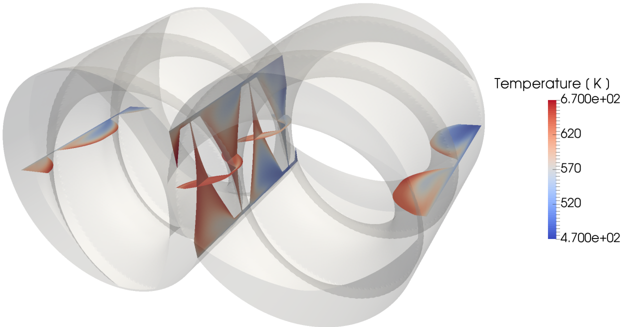


Figure 5. Temperature distribution inside 3D TSE due to viscous heating.

Conclusions

We presented SRMUM as an efficient boundary-conforming mesh update method, that allows to conduct unsteady computations on co-rotating intermeshing domains. The functionality and mesh convergence of the method has been validated using a 2D TSE example. Furthermore, results for more complex simulation of temperature-dependent flow inside 3D TSE have been shown.

References

[1] S. Elgeti and H. Sauerland. Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison. *Archives of Computational Methods in Engineering*, 23(2):323–361, 2016.

[2] A. Sarhangi Fard, M. Hulsen, H. Meijer, N. Famili, and P. Anderson. Adaptive non-conformal mesh refinement and extended finite element method for viscous flow inside complex moving geometries. *International Journal for Numerical Methods in Fluids*, 68(8):1031–1052, 2012.

[3] T. E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces–the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, Vol. 94(3):339 – 351, 1992.

[4] T. E. Tezduyar, M. Behr, S. Mittal, and A. Johnson. Computation of unsteady incompressible flows with the stabilized finite element methods: space-time formulations, iterative strategies and massively parallel implementations. *Asme Pressure Vessels Piping Div Publ PVP*, 246:7–24, 1992.