Isogeometric analysis framework for the numerical simulation of rotary screw machines. I. General concept and early applications

Möller, Matthias; Hinz, Jochen

DOI
10.1088/1757-899X/425/1/012032

Publication date
2018

Document Version
Final published version

Published in
IOP Conference Series: Materials Science and Engineering

Isogeometric analysis framework for the numerical simulation of rotary screw machines. I. General concept and early applications

IOP Conf. Series: Materials Science and Engineering **425** (2018) 012032 doi:10.1088/1757-899X/425/1/012032

Isogeometric analysis framework for the numerical simulation of rotary screw machines. I. General concept and early applications

Matthias Möller and Jochen Hinz

Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Department of Applied Mathematics, Van Mourik Broekmanweg 6, 2628XE Delft, The Netherlands

E-mail: {m.moller,j.p.hinz}@tudelft.nl

Abstract. This paper reports on the current status of an isogeometric modeling and analysis framework for rotary twin-screw machines that is being developed by an international consortium of academic partners within the EU-funded MOTOR project. The approach aims at combining accurate geometry modeling capabilities with modern high-performance computing techniques to enable efficient multi-physics simulations. The paper discusses the overall design philosophy of our isogeometric modeling and simulation pipeline and presents an isogeometric grid generator that makes it possible to integrate ANSYS CFX. Preliminary computational results for Sod's shock tube problem and an asymmetric SRM profile are shown.

1. Introduction

Rotary twin screw compressors are positive displacement (PD) machines that use the mechanical power of two counter-rotating rotors to suck in a fluid at the inlet, transport it in the chambers that are formed between the two rotors and reduce their volume in the course of the rotation, and finally release the compressed fluid with higher pressure at the exit. Likewise, screw expanders adopt the inverse working principle and recover mechanical power from the fluid that enters with high pressure at the suction side and is released with lower pressure at the outlet. Both PD machine types are nowadays widely used in industrial applications.

The accurate numerical analysis of screw machines and the simulation-based optimization of their design remains a challenging task. In the simplest case, high-speed turbulent compressible fluid flow through the chambers, heat exchange between the fluid and the rotors and the casing through the respective surfaces, and the resulting thermal expansion of the rotor profiles need to be taken into account, which leads to a mathematical model consisting of a coupled system of partial differential equations (PDEs) that is closed by one or more constitutive laws. The complexity of this multi-physics (MP) problem can be further enhanced by including additional physical phenomena like condensation and vaporization. The second major challenge stems from the complicated rotor geometries and the resulting fluid chambers that, by definition of the working principle of PD machines, vary in time, thereby giving rise to small gap sections.

Current best practice is to utilize special purpose mesh generation tools like TwinMesh$^{\text{TM}}$ [1] and SCORG$^{\text{TM}}$ [2] to pre-generate a sequence of computational meshes for the fluid and solid part of the domain at pre-selected time instances and apply commercial simulation packages

like ANSYS CFX [3] and STAR-CCM+ [4] to perform multi-physics simulations using finite elements (FEM) and finite volumes (FVM) or a combination of both. This *mesh-based coupling approach* via the discrete unknowns of the numerical approximation is particularly attractive since it simplifies the combination of different simulation packages and makes the full potential of each of these tools immediately available for the numerical analysis of PD machines.

On the other hand, mesh-based coupling approaches have several conceptual limitations, which arise from the lack of a parametric master geometry model. This is not a particular problem of the numerical analysis of PD machines or of the particular mesh generation tools but it is inherent to the 'discreteness' of mesh-based coupling approaches.

Some of the major shortcomings that are frequently observed in practice are as follows:

1. Non-matching meshes at the MP domain interface make it difficult to exchange data between disciplines without introducing excessive interpolation errors. Moreover, the identification of 'neighboring' unknowns that communicate via the MP domain interface might become computationally expensive and calls for efficient search data structures.
2. Modeling of the MP domain interface as discrete surface mesh imposes an *a priori* maximum resolution on the geometry model and makes solution-adaptive mesh refinement beyond this level ineffective. Moreover, the lack of a parametric geometry model counteracts advanced computational approaches such as adjoint-based shape optimization or sensitivity analysis, which perform best if applied in the parameter space rather than on mesh points.
3. Combining the computational modeling of the geometry with the generation of high-quality computational meshes complicates this tasks and makes it particularly challenging to develop mathematical methods that can be proven to produce 'admissible' meshes.
4. Load-balancing strategies for distributed high-performance computing have no access to topology information of the geometry but need to schedule hundreds of thousands of individual cells to the different processors trying to minimize communication overhead.

In addition, TwinMesh$^{\text{TM}}$ [1] and SCORG$^{\text{TM}}$ [2] are particularly designed as pre-processings tools for ANSYS CFX$^{\text{TM}}$ [3], which can only handle computational meshes consisting of straight-sided cells. Our interest is mainly in high-order methods (see Section 2.2 for the technical details of the adopted approach), which require computational meshes (actually, parametrizations) with curved element boundaries in order to represent complex geometries more accurately. High-order mesh generation is a topic of active research in itself and, to the best of our knowledge, neither TwinMesh$^{\text{TM}}$ [1] and SCORG$^{\text{TM}}$ [2] are capable of producing such meshes.

Geometry-aware parametrization-based multi-physics coupling. This work is the first in a series of publications that propose a novel approach for the numerical analysis of PD machines based on a parametric master geometry model with the ultimate aim to overcome the aforementioned shortcomings and to enable much more accurate predictions. Our approach builds on the concept of Isogeometric Analysis (IGA), which is a recent simulation technology that was introduced by Hughes et al. [5] in 2005 with the explicit vision to unify the worlds of Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE). The central idea is to utilize the same mathematical language, namely, (non-uniform rational) B-Splines for describing the geometry in design and analysis and for seeking approximate solutions to the multi-physics problems.

The full realization of the envisaged isogeometric analysis framework is ongoing work that is carried out by an international consortium of academic partners and will be reported in subsequent publications. This paper describes the general concept including a short introduction to IGA and computational geometry modeling in Section 2. Section 3 discusses our recently developed isogeometric mesh generation tool, which makes it possible to couple any of the IGA applications that are provided by the open-source G+Smo (Geometry + Simulation Modules)

library [6], which serves as common software base for all IGA-related developments, with ANSYS CFX [3]. This development is both meant as intermediate step to test the new approach with an established flow solver and to enable early adopters of this new technology to exploit the benefits of using a parametric master geometry model within an established simulation pipeline consisting of commercial solver packages. Preliminary computational results for fluid flow through an asymmetric SRM profile and a variation of Sod's shock tube benchmark problem [7] are reported in Section 4. Conclusions and an outlook on future research are given in Section 5.

2. Isogeometric Modeling and Analysis Pipeline

This section introduces the general concepts of our isogeometric modeling and simulation approach in an abstract mathematical formulation. Our aim is to highlight the conceptual differences of what we consider as *parametric master geometry model* to existing mesh generation techniques such as those adopted by TwinMesh$^{\text{TM}}$ [1] and SCORG$^{\text{TM}}$ [2], respectively.

2.1. Parametric curves

Mathematically speaking, the interfaces between MP domains are given by so-called *functional free-form surfaces*, which can be considered as two-dimensional manifolds that are exposed to the three-dimensional Cartesian space $\mathbb{R}^3$. These geometric objects are typically constructed by defining a parametric mapping $\mathbf{F}$ between the *parameter domain* $\hat{\Omega}$ and the *physical domain* Ω.

In order to illustrate the underlying principles of parametric curves and their algorithmic manipulation, consider the parametric curve depicted in Figure 1, which results from defining

$$\mathbf{F}(u) := \begin{pmatrix} \cos(2u)\cos(u) \\ \cos(2u)\sin(u) \end{pmatrix} \quad \text{where} \quad u \in \hat{\Omega} := [0,1]. \tag{1}$$

The so-defined mapping is bijective (one-to-one and onto), which means that the parameter value u^* of an arbitrary point (x^*, y^*) on the curve can be recovered from is inverse, that is

$$u^* = \mathbf{F}^{-1}(x^*, y^*). \tag{2}$$

The red circles in Figure 1 illustrate the location of the equidistantly distributed parameter values $u_i = i/9$, $i = 0, 1, \ldots, 9$ along the curve. Likewise, black dots stand for the non-equidistantly distributed parameter values $u'_i := \mathbf{G}(u_i)$ with bijective function $\mathbf{G} : [0,1] \to [0,1]$ defined as

$$u' = \mathbf{G}(u) := \sqrt{u} \quad \text{with} \quad u = \mathbf{G}^{-1}(u') = (u')^2. \tag{3}$$

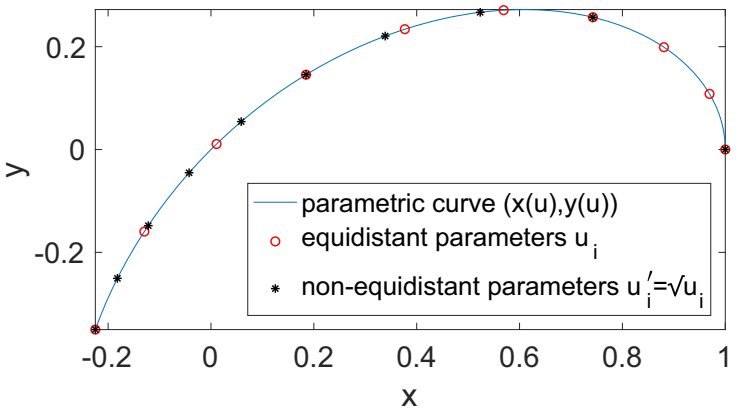


Figure 1. Illustration of a univariate parametric curves $\mathbf{F}(u) = (\cos(2u)\cos(u), \cos(2u)\sin(u))$ with equidistantly and non-equidistantly distributed parameter values $0 \leq u, u' \leq 1$.

We would like to emphasize that both $\mathbf{F}$ and $\mathbf{F} \circ \mathbf{G}$ represent the same *geometric* object but the so-called parametric speed at which the curve is traversed when varying u and u' between zero and one differs. Nonetheless, the bijectiveness of all involved mappings makes it possible to convert between both descriptions back and forth. This feature is particularly helpful when exchanging data between the solid and the fluid along a common interface curve.

For illustration purposes let (x^*, y^*) denote the point on the curve at which information should be exchanged. According to (2) and (3) both u^* and $(u^*)^2$ are mapped to the same physical location (x^*, y^*) under the mapping $\mathbf{F}$ and $\mathbf{F} \circ \mathbf{G}$, respectively.

2.2. Multi-patch parametrizations

Consider the benchmark configuration of a two-dimensional test rig depicted in Figure 2, which consists of an asymmetric SRM rotor profile (reddsih), a circular casing (greenish), and the fluid region (blueish) between the two solids. This benchmark has been studied experimentally by Sachs in [8] and numerically using ANSYS CFX$^{\text{TM}}$ by Utri et al. in [9]. The (planar) physical domain that is depicted in Figure 2 (b) consists of 3×4 so-called *patches*, that is, geometric objects that can be mapped onto one of the unit squares defined in the parameter domain; cf. Figure 2 (a). The decomposition of the screw machine geometry into several patches is advisable in order to simplify the parametric treatment of the 'step' profile at the tip of of the rotor; cf. Figure 3. Here and below white numbers denote the patch number.

Mathematically speaking, there exist $k = 1, \ldots, 12$ distinct bijective mappings between the parameter domain $\hat{\Omega}_k = [0,1] \times [0,1]$ and the physical patch $\Omega_k \subset \mathbb{R}^2$ such that

$$\mathbf{F}_k(u_k, v_k) := \begin{pmatrix} x_k(u_k, v_k) \\ y_k(u_k, v_k) \end{pmatrix} \qquad \begin{matrix} (u_k, v_k) & \in \hat{\Omega}_k, \\ (x_k, y_k) & \in \Omega_k, \end{matrix} \tag{4}$$

and $(u_k, 0)$, $(u_k, 1)$, $0 \le u_k \le 1$ and $(0, v_k)$, $(1, v_k)$, $0 \le v_k \le 1$ are mapped onto the four boundary segments Γ_k^s, $s \in (S, N, W, E)$, which together form the boundary of the k-th patch.

Even though it is not mandatory for the method to work, it eases the implementation if the mappings of two adjacent patches, say, Ω_k and Ω_l that share the common boundary, say,

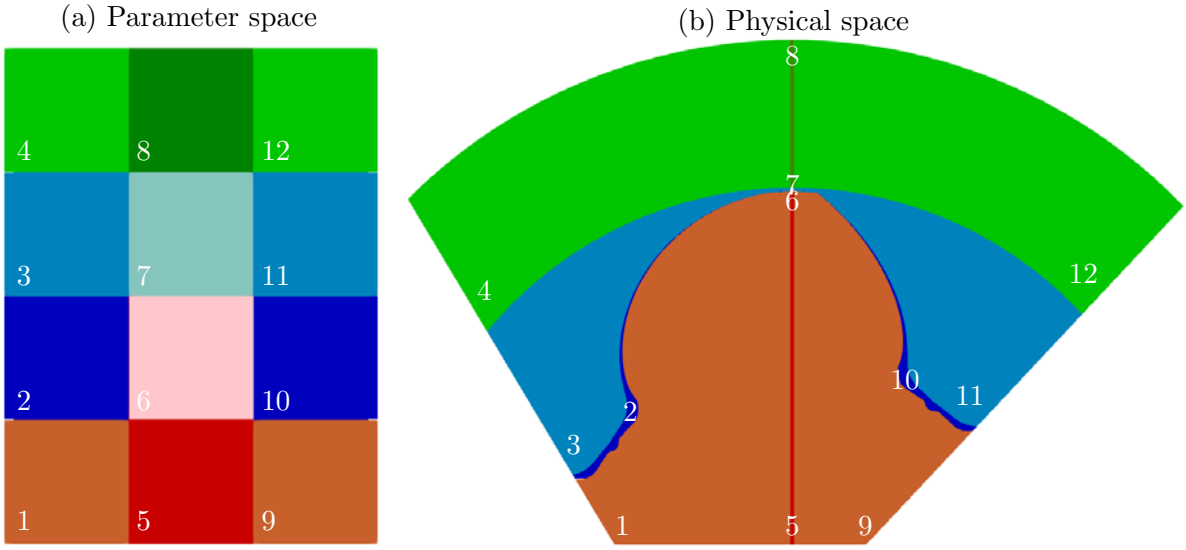


Figure 2. Segmentation of the test rig geometry into 3×4 multi-patch topology in (a) parameter and (b) physical space. Solid rotor (reddish), casing (greenish) and fluid region (blueish).

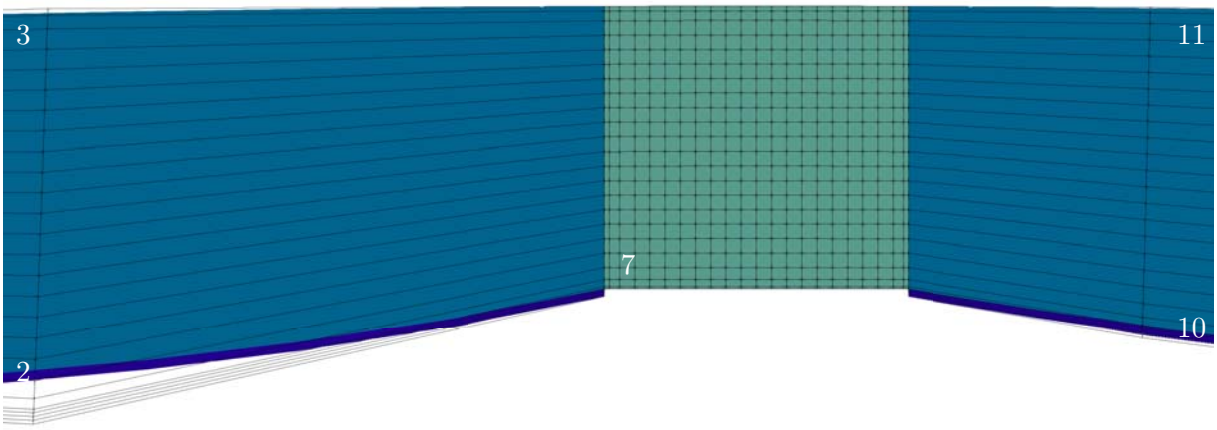


Figure 3. Close-up of the flow passage between casing and rotor tip. The B-Spline based geometry model is visualized by blue-colored patches. Black lines indicate the control net.

$\Gamma_k^S = \Gamma_l^N$ are parametrically compatible, that is, the parametric value u that describes the boundary segment is mapped to the same physical position by both mappings, e.g.

$$\mathbf{F}_k(u,0) = \mathbf{F}_l(u,1), \quad 0 \leq u \leq 1. \tag{5}$$

We will later relax this assumption to enable excessive mesh grading in the fluid domain along the boundaries and utilize the reparametrization (3) to restore parametric compatibility.

The ensemble of patches together with the list of patch neighbors is termed *multi-patch* topology. For the sake of better readability, let us drop subscript k and describe our isogeometric modeling and simulation approach for a single patch (unless explicitly stated otherwise) keeping in mind that all steps have to be applied to all patches Ω_k separately.

2.3. B-Spline based geometry modeling

In our isogeometric framework we resort to tensor-product B-Splines rather than non-uniform rational B-Splines (NURBS), which are most commonly used in Computer-Aided Design (CAD) applications but would render our approach more complicated. To begin with, let $U = [u_1, u_2, \ldots, u_{n+p+1}]$ be a sequence of non-decreasing values between zero and one with the additional constraint that the first and last $p+1$ entries are identical. This yields the so-called open knot vector U, which is used to construct univariate B-Spline basis functions $\{N_{i,p}(u)\}_{i=1}^n$ of degree p with the aid of the Cox-de-Boor recursion formula [10]:

$$p = 0: \quad N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$$p > 0: \quad N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{7}$$

The above definition assumes the convention "$\frac{0}{0} = 0$" to prevent division by zero. B-Spline basis functions $\{N_{j,q}(v)\}_{j=1}^m$ of degree q are constructed from the knot vector $V = [v_1, v_2, \ldots, v_{m+q+1}]$.

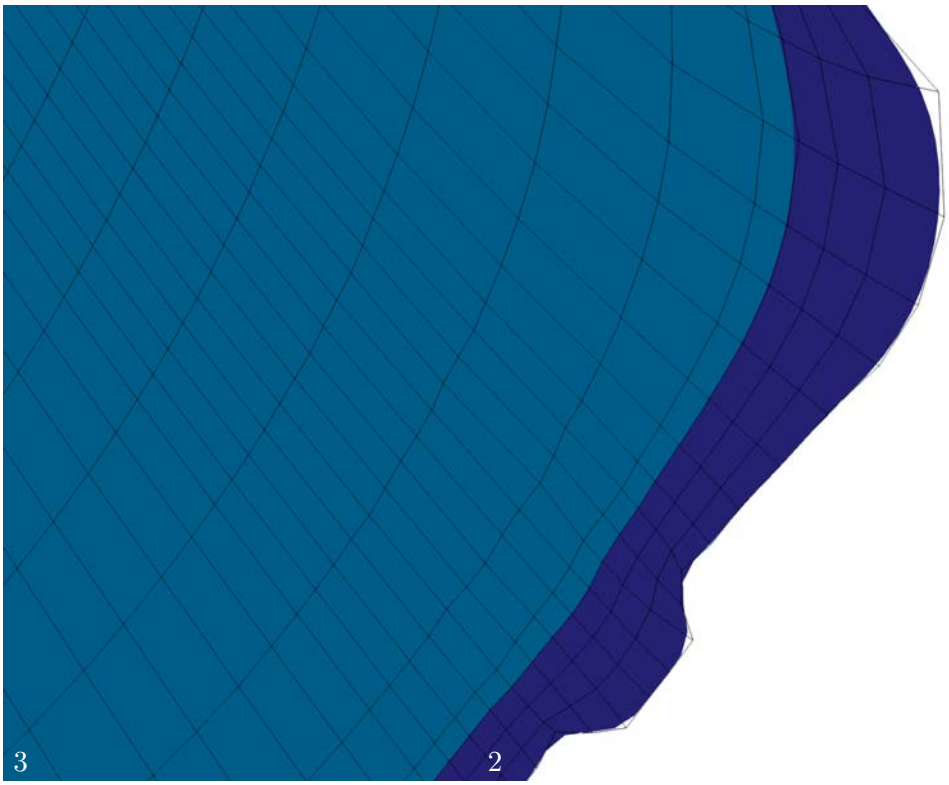


Figure 4. Close-up of the flow passage between casing and rotor tip. The B-Spline based geometry model is visualized by blue-colored patches. Black lines indicate the control net.

Our overall aim is to represent the geometry mapping (4) as a bivariate B-Spline surface that is expressed in terms of the tensor product of univariate B-Spline basis functions. That is,

$$\mathbf{F}_h(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{c}_{i,j} N_{i,p}(u) N_{j,q}(v), \quad 0 \le u, v \le 1. \tag{8}$$

It should be noted that the above mapping does not necessarily yield an exact representation of the true physical domain Ω but rather an approximation $\Omega_h \approx \Omega$, which is indicated by subscript 'h'. The quality of the approximation can often be enhanced by increasing the number of basis functions and distributing the entries of the generating knot vectors U and V in a way that accounts for geometric details; cf. Figures 3 and 4. However, an *exact* B-Spline representation of a domain that is described, say, by trigonometric functions (see Figure 1) is impossible.

In the above expression, $\mathbf{c}_{i,j} \in \mathbb{R}^2$ are the *control points* that 'steer' the deviation of the physical patch from the unit square as sketched schematically in Figure 5. It should be noted that only the four corner points are interpolated, e.g., $\mathbf{F}_h(0,0) = \mathbf{c}_{1,1}$, whereas all other control points can attain arbitrary values that may even lie outside of the physical patch, cf. Figures 3–5. Control points are connected in horizontal and vertical parametric direction by the so-called *control net*, which is indicated by black lines. Let us remark that these are not the grid lines of the computational mesh but only serve the purpose of giving control to the parametrization (8).

The univariate boundary parametrizations are obtained from the bivariate planar

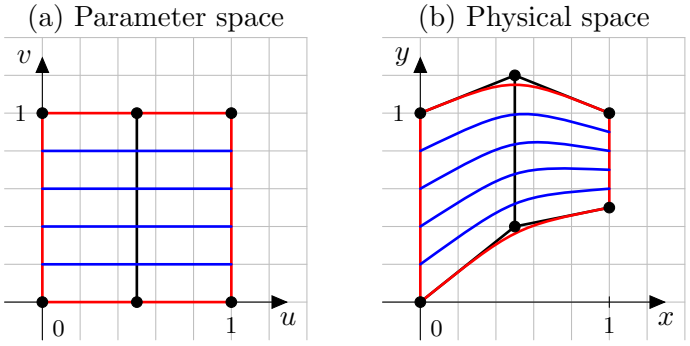


Figure 5. Mapping between (a) parameter and (b) physical domain. Black lines indicate the control net with control points $\mathbf{c}_{i,j}$ given by dots. Boundary segments are in red. Grid lines with parameter values varying in u-direction are sketched in blue for illustration purposes.

parametrization (8) by setting one of the two parameters to zero or one, e.g.

$$\mathbf{F}_h^S(u) := \mathbf{F}_h(u,0) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{c}_{i,j} N_{i,p}(u) N_{j,q}(0) = \sum_{i=1}^{n} \mathbf{c}_{i,1} N_{i,p}(u), \quad 0 \le u \le 1. \tag{9}$$

Let us switch to vector notation, i.e. $\mathbf{x} := (x, y)$ and $\mathbf{u} := (u, v)$, to improve readability and also abbreviate the tensor-product of univariate B-Spline basis functions as $B_I(\mathbf{u}) := N_{i,p}(u) N_{j,q}(v)$, which allows us to express the geometry mapping (8) in a more compact way as follows

$$\mathbf{x} = \mathbf{F}_h(\mathbf{u}) = \sum_{I=1}^{n\times m} \mathbf{c}_I B_I(\mathbf{u}), \quad \mathbf{u} \in \hat{\Omega}. \tag{10}$$

Here, the global index $I := j \times (n-1) + i$ is computed from the local indices of univariate basis functions. Since the forward mapping $\mathbf{F}_h$ is assumed to be bijective, the inverse mapping

$$\mathbf{u} = \mathbf{F}_h^{-1}(\mathbf{x}), \quad \mathbf{x} \in \Omega_h. \tag{11}$$

from the physical domain Ω_h onto the parameter domain $\hat{\Omega}$ exists. Since it pulls any physical coordinate back into the parameter space $\hat{\Omega}$ it is termed the 'pull-back' mapping.

To further simplify the notation we drop subscript 'h' whenever it is clear from the context whether the exact object or its approximation is meant.

2.4. Geometry pre-processor

A geometry pre-processor has been developed in [11, 12, 13] that generates planar mappings of the form (8) from a point cloud $\{\mathbf{p}_a\}$ that represents the boundary profile $\partial\Omega$. In practice, this point cloud is generated by an in-house tool for designing screw machine rotor profiles provided by the Chair of Fluidics at TU Dortmund University but other point clouds are also accepted.

The geometry pre-processor uses elliptic grid generation techniques and extends them into an *isogeometric* variational framework, thereby especially exploiting the higher continuity, namely, C^{p-1} of B-Spline basis functions of degree p. The main algorithmic steps are as follows:

1. Generate individual parametrizations of the four boundary segments by fitting univariate B-Splines to the discrete points ensuring that the curves meet at the respective corners

$$\{\mathbf{p}_a^N\} \mapsto \mathbf{F}^N(u), \qquad \{\mathbf{p}_a^S\} \mapsto \mathbf{F}^S(u), \tag{12}$$

$$\{\mathbf{p}_a^E\} \mapsto \mathbf{F}^E(v), \qquad \{\mathbf{p}_a^W\} \mapsto \mathbf{F}^W(v). \tag{13}$$

2. Reparameterize opposite boundary segments (if necessary) to improve grid orthogonality.
3. Merge knot vectors of opposite boundary segments to obtained unified B-Spline bases.
4. Solve the vector-valued nonlinear problem [14] on the computational domain $\Omega_h \approx \Omega$

$$\mathbf{g}_{vv}\mathbf{x}_{uu} - 2\mathbf{g}_{uv}\mathbf{x}_{uv} + \mathbf{g}_{uu}\mathbf{x}_{vv} = 0, \quad \text{s.t.} \quad \mathbf{x}|_{\partial\Omega_h} = \partial\Omega \tag{14}$$

with $\mathbf{g}_{uv} := \mathbf{F}_u \cdot \mathbf{F}_v$ using a variational approach to directly obtain the 'inner' control points of (8) as unknowns and imposing the boundary condition on the $\mathbf{c}_{i,j}$'s at the boundary.

It can be shown [15] that the exact solution to (14) yields indeed a bijective mapping. It is assumed that this property carries over to the numerical solution at least almost everywhere.

The efficient and robust implementation of the isogeometric mesh pre-processor involves many nifty details, which are described in [11, 12, 13] together with an extension of the underlying algorithm for generating volumetric parametrizations of twin-screw machine geometries from a sequence of planar ones sampled at different locations along the rotation axes. In practice, the input point cloud typically represents the entire geometry and may not even satisfy any ordering. It therefore needs to be segmented into point clouds that can be associated with the different segments of the different patches and possibly further pre-processed (e.g., removal of duplicate points) before the above algorithm can be applied to the individual patches/segments.

2.5. Isogeometric variational formulation

As stated in the previous section, the PDE problem (14) is solved by a variational approach, which is derived in a similar way as in *parametric* finite elements. In short, the residual of the strong form of the PDE problem is multiplied by a test function w from the test space W, integrated over the physical domain Ω and required to vanish for all admissible test functions $w \in W$. For illustration purposes, consider the generic linear PDE problem

$$\mathcal{L}\varphi = f \quad \text{in } \Omega, \tag{15}$$

which is cast into the variational formulation: Find a solution $\varphi \in V$ such that

$$\int_\Omega w\,[f - \mathcal{L}\varphi]\ \mathrm{d}\mathbf{x} = 0 \quad \text{for all } w \in W. \tag{16}$$

Here, V denotes the space of admissible solutions, which needs to account for any Dirichlet boundary conditions that are imposed on the differential equation (15). Neumann boundary conditions are easily implemented by performing integration by parts and overriding the corresponding expressions in the arising boundary integral terms. Next, trial and test spaces V and W are approximated by the finite dimensional polynomial space $\mathcal{S}^{p,q}(U,V)$, which is spanned by the tensor-product B-Spline basis functions $\{B_I\}_{I=1}^{n\times m}$ constructed with the Cox-de-Boor recursion formula (6)–(7) based on the knot vectors U and V, respectively.

The major difference between classical FEM and IGA is the use of the *same* basis for constructing the geometry mapping (8) and approximating the solution to problem (16), i.e.

$$\varphi_h(\mathbf{x}) = \sum_{I=1}^{n\times m} \varphi_I B_I \circ \mathbf{F}^{-1}(\mathbf{x}), \quad \mathbf{x} \in \Omega_h. \tag{17}$$

Since the B_I's are defined on $\hat{\Omega}$, the physical coordinates need to be pulled back. The solution coefficients φ_I are obtained by inserting (17) into (16), replacing w by the $n \times m$ basis functions and solving the resulting linear system of the form $A_h\varphi_h = f_h$ for φ_h. Due to the properties of B-Spline basis functions, the so-defined solution coefficients are *not* interpolatory, that is, $\varphi_h(u_i, v_j) \neq \varphi_{i,j}$, which is the case for classical FEM with nodal Lagrange basis functions.

In what follows we apply this generic approach to derive the isogeometric variational formulations for the equations of gas dynamics. The derivation of the variational formulation for the governing equation of the geometry pre-processor (14) is described in [11, 12, 13].

Compressible Euler equations. Consider the divergence form of the problem at hand

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F}(U) = 0 \quad \text{in } \Omega^{\text{fluid}} \times [0, t_{\text{end}}], \tag{18}$$

where $U : \mathbb{R}^d \to \mathbb{R}^{d+2}$ denotes the state vector of conservative variables

$$U = \left[U_1, \ldots, U_{d+2}\right]^\top = \left[\rho, \rho\mathbf{v}, \rho E\right]^\top, \tag{19}$$

and $\mathcal{F} : \mathbb{R}^{d+2} \to \mathbb{R}^{(d+2)\times d}$ stands for the tensor of inviscid fluxes

$$\mathcal{F} = \begin{bmatrix} F_1^1 & \cdots & F_1^d \\ \vdots & \ddots & \vdots \\ F_{d+2}^1 & \cdots & F_{d+2}^d \end{bmatrix} = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \otimes \mathbf{v} + p\mathcal{I} \\ \rho E\mathbf{v} + p\mathbf{v} \end{bmatrix} \tag{20}$$

with mass density ρ, velocity $\mathbf{v}$, total energy E, and $\mathcal{I}$ denoting the d-dimensional identity tensor. For an ideal polytropic gas, the pressure p is given by the equation of state

$$p = (\gamma - 1)\rho e, \tag{21}$$

where $\gamma = c_p/c_v$ denotes the heat capacity ratio, which equals $\gamma = 1.4$ for dry air and

$$\rho e = \left(\rho E - 0.5\rho|\mathbf{v}|^2\right) \tag{22}$$

is the internal energy per unit volume, from which temperature T can be computed as follows

$$T = \frac{e}{c_v}. \tag{23}$$

The governing equations are equipped with initial conditions prescribed at time $t = 0$

$$U(\mathbf{x}, 0) = U_0(\mathbf{x}) \ \text{ in } \Omega^{\text{fluid}}, \tag{24}$$

and boundary conditions of Dirichlet and Neumann type, respectively

$$U = \mathcal{G}(U, U_\infty) \quad \text{on } \Gamma_D^{\text{fluid}}, \tag{25}$$

$$\mathbf{n} \cdot \mathcal{F} = \mathcal{F}_n(U, U_\infty) \ \text{ on } \Gamma_N^{\text{fluid}}. \tag{26}$$

Here, $\mathbf{n}$ is the outward unit normal vector and U_∞ denotes the vector of 'free stream' solution values, which are calculated and imposed as described in detail in [16].

After applying integration by parts to incorporate the Neumann-type boundary fluxes, the variational formulation of the problem at hand reads: Find $U(t) \in V$ such that

$$\int_{\Omega^{\text{fluid}}} w\frac{dU}{dt} - \nabla w \cdot \mathcal{F}(U)\,\mathrm{d}\mathbf{x} + \int_{\Gamma_N^{\text{fluid}}} w\mathcal{F}_n(U, U_\infty)\,\mathrm{d}s = 0 \tag{27}$$

for all test functions $w \in W$. Let us approximate the state vector and the inviscid fluxes as

$$U_h(\mathbf{x}, t) = \sum_{I=1}^{n\times m} U_I(t) B_I \circ \mathbf{F}^{-1}(\mathbf{x}), \tag{28}$$

$$\mathcal{F}_h(\mathbf{x}, t) = \sum_{I=1}^{n\times m} \mathcal{F}_I(t) B_I \circ \mathbf{F}^{-1}(\mathbf{x}), \tag{29}$$

where $\mathcal{F}_I(t) := \mathcal{F}(U_I(t))$ denotes the value of (20) evaluated at the I-th solution coefficient $U_I(t)$ at time t. This approach is known as Fletcher's group formulation [17] and it is frequently used in finite element methods for conservation laws to approximate the first-order flux function.

To further simplify the notation, let us define the consistent mass matrix $M_C := \{m_{IJ}\}$ and the discretized divergence operator $\mathbf{C} := \{\mathbf{c}_{IJ}\}$ whose entries are given by

$$m_{IJ} = \int_{\hat{\Omega}} B_I(\mathbf{u}) B_J(\mathbf{u}) \, |\det \mathbf{DF}(\mathbf{u})| \, \mathrm{d}\mathbf{u}, \tag{30}$$

$$\mathbf{c}_{IJ} = \int_{\hat{\Omega}} \left(\nabla B_I(\mathbf{u}) \mathbf{DF}(\mathbf{u})^{-1}\right) B_J(\mathbf{u}) \, |\det \mathbf{DF}(\mathbf{u})| \, \mathrm{d}\mathbf{u}. \tag{31}$$

The above expressions naturally arise from transforming all integrals over the physical domain Ω^{fluid} into integrals over the parametric domain $\hat{\Omega}$. Here, $\mathbf{DF}$ denotes the Jacobian matrix of the geometry mapping (8) and $\det \mathbf{DF}$ is its determinant, which is strictly non-zero as long as $\mathbf{F}$ is bijective. In contrast to parametric finite elements, where this technique is applied individually for each element ($\mathbf{F}$ is defined at the element level), IGA employs a patch-wise global mapping. In a practical implementation, the integral terms are evaluated by applying numerical quadrature, e.g., a sufficiently accurate Gauss quadrature rule, to each 'virtual' element $\omega_{i,j} = [u_i, u_{i+1}] \times [v_j, v_{j+1}]$, where u_i and v_j denote the knots introduced in Section 2.3.

At the end of the day, the semi-discrete form of the Euler equations reads [18]

$$M_C \frac{dU_h}{dt} - \sum_{b=1}^{d} C^b F^b(U_h) + S(U_h) = 0, \tag{32}$$

where superscript $b = 1, \ldots, d$ refers to the b-th spatial component of the discrete divergence operator $\mathbf{C}$ and the tensor of inviscid fluxes $\mathcal{F}$, respectively. Here, $S(U_h)$ accounts for the contribution of the boundary fluxes; see publication [16] for more details.

Equation (32) is discretized in time by an explicit strong stability preserving Runge-Kutta (SSP-RK) method of order three [19]. The resulting scheme is known to produce non-physical oscillations in the vicinity of shocks and steep gradients, which can lead to negative density and pressure values and, hence, to the complete failure of the simulation. This shortcoming is cured by applying algebraic flux correction (AFC), which has been originally developed for low-order FEM [16] and has been extended to isogeometric schemes recently [18, 20]. Preliminary results are given in Section 4.1 to demonstrate the general applicability of our isogeometric approach.

A word on parametric multi-physics coupling. Even though the numerical results presented in this publication are restricted to single-phase compressible flows, we give a brief conceptual outlook on how IGA simplifies the exchange of data between multiple disciplines. Let

$$q_h := (\nabla T_h(\mathbf{x}, t)) \cdot \mathbf{n}^{\text{fluid}} = \sum_{I=1}^{n \times m} T_I(t) \nabla \left(B_I \circ \mathbf{F}^{-1}(\mathbf{x})\right) \cdot \mathbf{n}^{\text{fluid}} \tag{33}$$

be the numerically computed temperature gradient in outward normal direction at the boundary segment Γ_N^{fluid} of the fluid domain Ω^{fluid} at time t and assume that its values should be prescribed as flux boundary condition $(\kappa \nabla T) \cdot \mathbf{n}^{\text{heat}} = -\kappa q$ at segment Γ_N^{heat} for the heat equation

$$\rho c_p \frac{dT}{dt} - \nabla \cdot (\kappa \nabla T) = \dot{q}_V \quad \text{in } \Omega^{\text{heat}} \times [0, t_{\text{end}}], \tag{34}$$

where $\dot{q}_V$ the heat energy generated per unit volume per unit time and κ the thermal conductivity of the material. The variational form reads: Find $T(t) \in V$ such that

$$\int_{\Omega^{\text{heat}}} w \left[\rho c_p \frac{dT}{dt} - \dot{q}_V\right] + \nabla w \cdot (\kappa \nabla T) \, \mathrm{d}\mathbf{x} + \int_{\Gamma_N^{\text{heat}}} w \kappa q \, \mathrm{d}s = 0 \tag{35}$$

for all test functions $w \in W$. After the discretization of the temperature variable similarly to (17) and substitution of (33) into the boundary integral we obtain the semi-discrete system

$$M_C \frac{dT_h}{dt} + KT_h + S(T_h) = 0 \tag{36}$$

where $K = \{k_{IJ}\}$ denotes the stiffness matrix whose entries are given by

$$k_{IJ} = \int_{\hat{\Omega}} \left(\nabla B_I(\mathbf{u})\mathbf{DF}(\mathbf{u})^{-1}\right) \cdot \left(\nabla B_J(\mathbf{u})\mathbf{DF}(\mathbf{u})^{-1}\right) \, |\det \mathbf{DF}(\mathbf{u})| \, \mathrm{d}\mathbf{u} \tag{37}$$

and $S(T_h)$ accounts for the boundary contribution. As before, the integral terms are evaluated by transforming them to the parametric space and applying numerical quadrature. Let $\mathbf{F}^{\text{fluid}} : [0,1] \to \Gamma_N^{\text{fluid}}$ and $\mathbf{F}^{\text{heat}} : [0,1] \to \Gamma_N^{\text{heat}}$ be two bijective parametrizations of the common interface between the fluid and heat domain. Since $\Gamma_N^{\text{heat}} \equiv \Gamma_N^{\text{fluid}}$ in the isogeometric multi-patch geometry model, the parametric values are related by the following expression

$$\mathbf{F}^{\text{fluid}}(t^{\text{fluid}}) = \mathbf{F}^{\text{heat}}(t^{\text{heat}}). \tag{38}$$

Hence, the temperature gradient stemming from the fluid solution can be directly evaluated in the boundary integral term of the heat equation without extra interpolation steps, i.e.

$$s_I = \int_0^1 B_I(t) \, \kappa \, q_h(\mathbf{F}^{\text{heat}}(t)) \, |\det \mathbf{Df}^{\text{heat}}| \, \mathrm{d}t. \tag{39}$$

Equation (38) can also be used to apply a similar parametric coupling approach when imposing the wall temperature of the rotor as Dirichlet boundary condition to the fluid solver.

3. Isogeometric mesh generator for ANSYS CFX

The complete realization of the isogeometric modeling and analysis pipeline as described in the previous section requires that all computational building blocks, i.e., the geometry pre-processor and the flow solver, are implemented and operational at the maturity level of commercial tools in order to enable realistic simulations of twin-screw machines. To validate the quality of the isogeometric geometry pre-processor, we have realized an isogeometric mesh generator, which converts the parametric master geometry model into a block-structured mesh for ANSYS CFX$^{\text{TM}}$ [3]. The block structure immediately follows from the multi-patch topology; cf. Figure 2.

The computational mesh for a single block, say, k is created by constructing a tensor-product mesh with equidistant mesh widths Δu and Δv in the parameter domain $\hat{\Omega}_k$ and transforming it with the corresponding forward mapping $\mathbf{F}_k(u,v)$ into the physical domain Ω_k. The resulting planar quadrilateral mesh is extruded in the z-direction to obtain the mesh shown in Figure 6.

Since the mapping is assumed to be bijective, an admissible tensor-product mesh in $\hat{\Omega}$ will yield an admissible mesh in Ω_k with the same connectivity pattern and curved mesh lines. The term admissible means that no twisted mesh cells occur but it does not account for mesh quality.

For use in ANSYS CFX$^{\text{TM}}$, curved grid lines are replaced by straight lines after which the admissibility of the mesh needs to be re-assessed by checking that the (oriented) element volume is strictly positive for each hexahedra (cf. Eq. (11.28) in [21]). In our experience, non-admissible elements rarely occurr for highly curved elements with extreme aspect ratios. A simple cure is to split the respective element into two in the parameter domain and map the newly introduced vectices to their positions in the physical domain by virtue of the master geometry mapping.

Despite all effort that is invested into the creation of the mapping (8) to ensure that vertical and horizontal grid lines intersect each other 'orthogonally' and are perpendicular to the boundary by applying boundary re-parametrization, no attention is paid to additional mesh

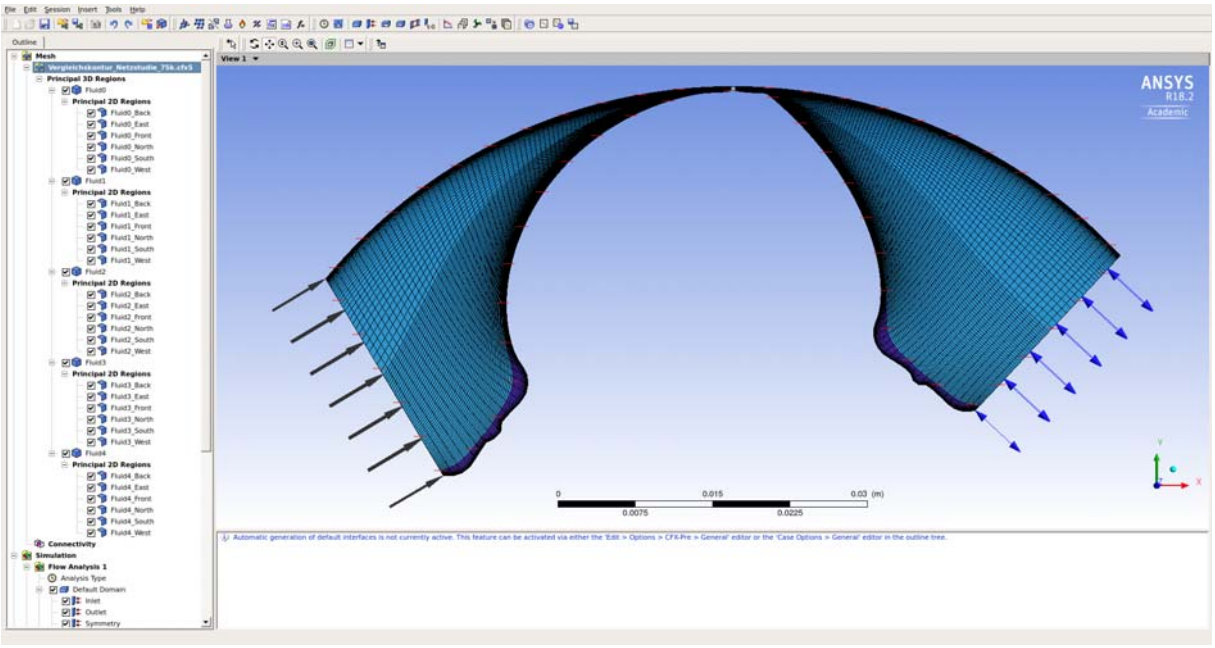


Figure 6. Block-structured hexahedral mesh for fluid domain (5-patch blue region) in CFX5-Pre. Adaptive mesh refinement towards boundary layers is imposed via pre-parametrization.

quality criteria like the accurate resolution of boundary layers. Figure 5 demonstrates that grid lines that run horizontally in the $u-v$ plane, and thus have constant distance to the domain boundary $\hat{\Gamma}$, can have varying distance to the boundary Γ in the $x-y$ plane.

Even though it is possible to augment the homogeneous problem (14) with a source term to modulate the mesh density locally, there are no clear guidelines on how to choose the source term in order to ensure a fixed distance between any two parallel grid lines or the first grid line and the wall. Moreover, this approach would require the numerical solution of the nonlinear PDE problem (14) for each computational grid, which is computationally expensive.

We therefore introduced an additional re-parametrization step, cf. Figure 7, which does not influence the shape of the geometry but only the distribution of grid lines. Let $\mathbf{G}(s,t) = (u,v)$

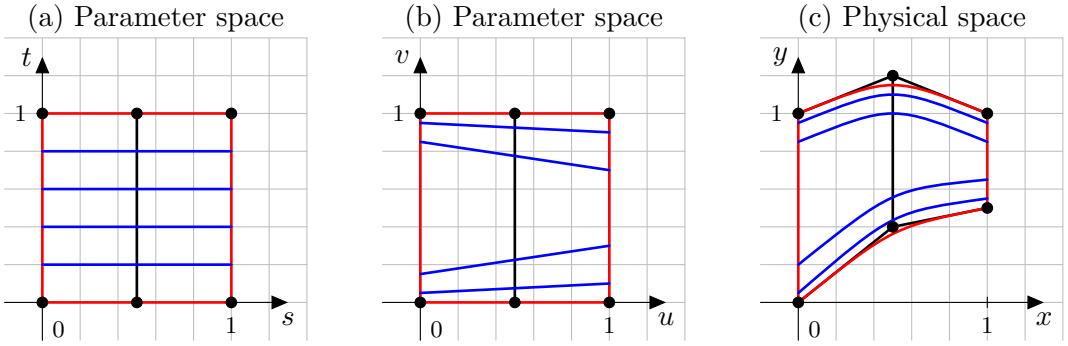


Figure 7. Mapping between (a)+(b) parameter and (c) physical domains. Black lines indicate the control net with control points $\mathbf{c}_{i,j}$ given by dots. Boundary segments are in red. Grid lines with parameter values varying in u-direction are sketched in blue for illustration purposes.

be a bijective mapping from $\tilde{\Omega} = [0,1]^2$ to $\hat{\Omega} = [0,1]^2$ that is defined as transfinite interpolation

$$\mathbf{G}(s,t) = \begin{pmatrix} (1-t)\mathbf{G}^W(s) + t\mathbf{G}^E(s) \\ (1-s)\mathbf{G}^S(t) + s\mathbf{G}^N(t) \end{pmatrix}, \quad 0 \le s,t \le 1, \tag{40}$$

where $\mathbf{G}^s$ are univariate mappings defined along the four boundary segments of the unit square.

The resulting mapping $(x,y) = \mathbf{F} \circ \mathbf{G}(s,t)$ is bijective as long as both $\mathbf{F}(u,v)$ and $\mathbf{G}(s,t)$ exhibit this property. As an example, the mesh grading function $(0 \le \gamma \le 1)$

$$\mathbf{G}_{LR}(\gamma) = 0.5 + \frac{\operatorname{atan}(\alpha(\gamma - 0.5))}{2\operatorname{atan}(\alpha/2)} \tag{41}$$

with $\alpha > 0$ being a user-definable parameter, clusters the grid lines towards the two ends of the unit interval, whereas mesh grading towards one end of the interval is achieved by

$$\mathbf{G}_L(\gamma) = 1 + \frac{\tanh((\gamma-1)\beta/2)}{\tanh(\beta/2)}, \tag{42}$$

$$\mathbf{G}_R(\gamma) = 1 - \frac{\tanh((0-\gamma)\beta/2)}{\tanh(\beta/2)}, \tag{43}$$

with $\beta > 0$. The computational mesh depicted in Figure 6 has been generated using the above functions and piece-wise combinations thereof, e.g., to achieve a moderate grading of the horizontal mesh width Δu_i over the entire length of the inlet section (patch numbers 2 and 3) with a rapid reduction close to the 'step' profile in the gap section; cf. Figure 9.

The quality of the resulting mesh has been assessed using ANSYS ICEM CFD$^{\text{TM}}$, which reports a mean quality value of 0.9561; cf. Figure 8. Numerical simulation results obtained on this mesh, which consists of 42,808 hexahedral elements are presented in Section 4.2.

4. Numerical Results

This section presents preliminary results from single-phase compressible flow simulations computed by our isogeometric solver that has been implemented in the open-source G+Smo (Geometry + Simulation Modules) library [6] and by the commercial tool ANSYS CFX$^{\text{TM}}$ [3].

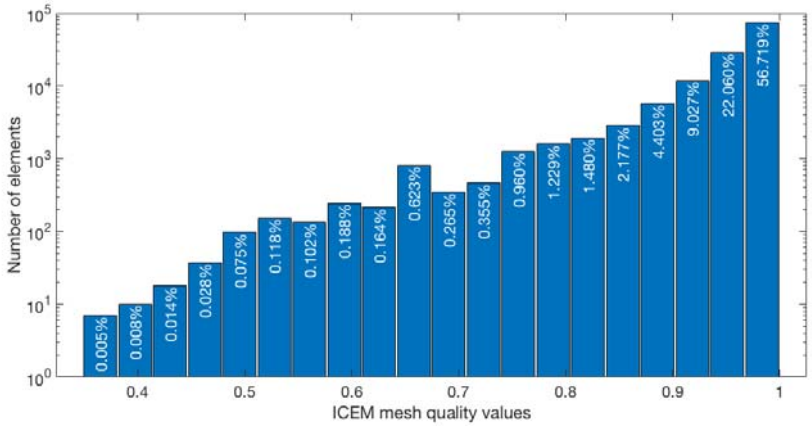


Figure 8. ANSYS ICEM$^{\text{TM}}$ mesh quality criterion: Min = 0.3489, max = 1, mean = 0.9561.

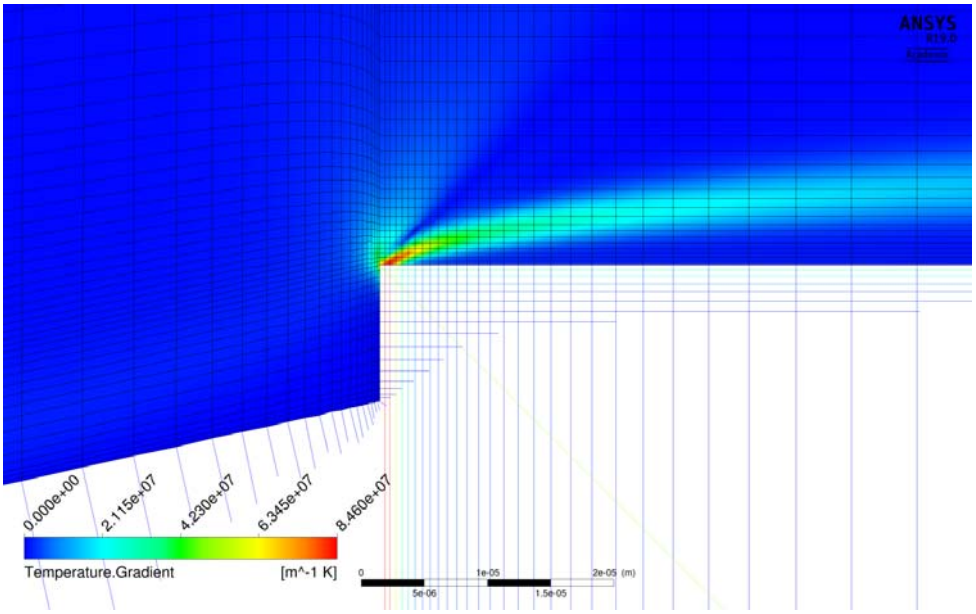


Figure 9. Temperature gradient in normal direction to the wall.

4.1. G+Smo results

The isogeometric flow solver has been tested for a variation of Sod's shock tube problem [7], which is a classical benchmark in computational gas dynamics. Consider a closed system (free-slip boundary conditions are imposed on the entire boundary Γ) with a fluid that is initially at rest and separated into two regions with $\rho_L = 1.0$, $p_L = 1.0$ and $\rho_R = 0.125$, $p_R = 0.1$ by a diaphragm located in the middle of the domain. Its abrupt removal causes the formation of a rarefaction wave, a contact discontinuity, and a shock wave. This setup has been considered on the VKI U-bend geometry [22], which has been modeled by 66×66 equidistantly distributed bi-quadratic tensor-product B-Splines. The density profile at time $t_{\text{end}} = 0.231$ that is shown in Figure 10 has been produced with our isogeometric approach, using the AFC technology for stabilization and a third-order explicit SSP-RK scheme with time-step size $\Delta t = 0.001$.

The black contour lines indicate the location of the three characteristic wave types, whereas 'parametric grid lines' (image of $\mathbf{F}(u, v)$ with one of the two parameters being kept constant) are given in white. The adopted AFC stabilization suppresses the generation of spurious oscillations reliably and furthermore ensures that the numerical solution values remain bounded by the physical upper and lower bounds, i.e. $0 < \rho \leq 1$, $0 < p$, and $0 < \rho E$.

It should be furthermore noted that the C^1 continuous parameterization of the curved boundary by a bi-quadratic B-Spline function yields a unique definition of the outward unit normal vector $\mathbf{n}(\mathbf{x}) = \mathbf{n}(\mathbf{F}(\mathbf{u}))$ in every point on the boundary Γ. Moreover, $\mathbf{n}$ is a continuous function of the boundary parameter value $\mathbf{u}$. As a consequence, the approximate solution is free of 'numerical artifacts', which often occur for polygonal boundary representations in P_1/Q_1 finite elements, where the C^0 'kinks' between two consecutive boundary segments acts as microscopic compression or expansion corners and, moreover, give rise to undetermined normal vectors in the vertices. In practice, this problem is often overcome by averaging the local normal vectors of the adjacent boundary segments but, still, the so-defined normal vector is not a continuous function of the boundary parameterization, which is the case for IGA.

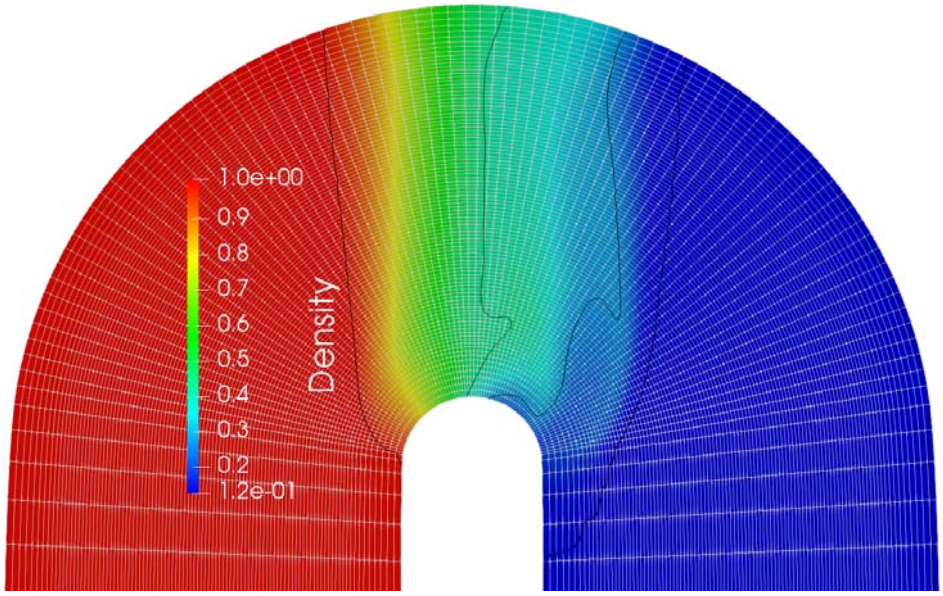


Figure 10. G+Smo solution to Sod's shock tube problem [7] computed on the VKI U-bend [22] geometry with 66×66 bi-quadratic tensor-product B-Splines and third-order SSP-RK scheme.

4.2. ANSYS CFX results

As a second test case, we have performed a fluid flow simulation for the test rig with asymmetric SRM rotor profile (cf. Figure 2) with ANSYS CFX$^{\text{TM}}$ [3]. The computational grid (cf. Figure 6) consists of 42,808 block-structured elements and was generated from the parametric master geometry model with the aid of our isogeometric mesh generator. Further details of the simulation are given in Table 1. Contour plots (51 contour levels) for the pressure, Mach number, and temperature distribution in the gap region are presented in Figure 11 (a)-(c). A close-up view of the temperature gradient is depicted in Figure 9 showing a steep increase by orders of magnitudes at the corner singularity. This might be attributed to the numerically computed first-order approximation of the gradient with extremely small mesh widths in normal direction. Work on improved gradient reconstruction schemes based on high-order Splines is in progress.

The flow becomes supersonic ($Ma_{\max} = 1.775$) behind the smallest part of the flow passage, which goes a long with a significant drop in temperature by about 120 °C relative to the prescribed temperature of 25°C prescribed at the inlet and outlet and the solid walls.

5. Conclusions and Outlook

In this paper we have sketched the overall design principles of our isogeometric modeling and analysis pipeline for rotary twin-screw machines that is currently being developed by an international consortium consisting of research groups in Germany and the Netherlands. The central idea is to link all computational tools to the same parametric master geometry model that can be used in an isogeometric analysis framework or converted to a highly accurate computational mesh for ANSYS CFX$^{\text{TM}}$ with the aid of our isogeometric mesh generator.

Preliminary results have demonstrated the applicability of the developed geometry pre-processor and the isogeometric mesh generator. Moreover, the isogeometric flow solver has been successfully applied to inviscid compressible flows, which have been studied in curved geometries. The extension to viscous turbulent flows and more complex multi-patch topologies is ongoing work. Current research focusses on the two-way coupling of the fluid flow solvers with an isogeometric thermo-dynamical simulation of the expansion of the screw machine rotors [23].

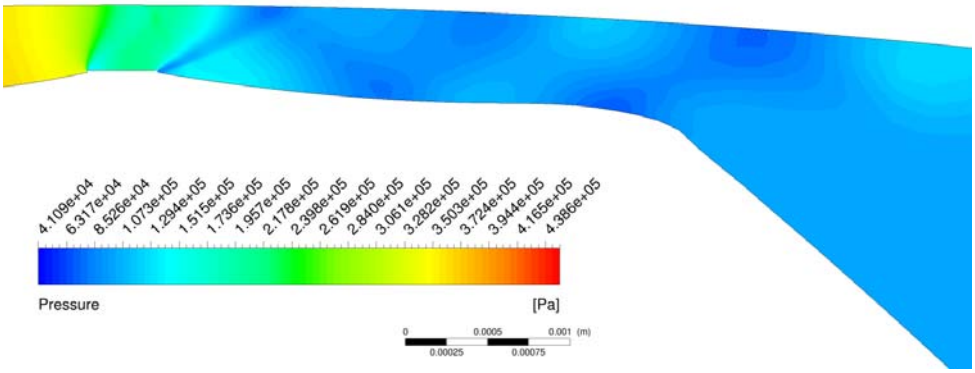
Basic Settings and Fluid Model	
Material	Air Ideal Gas
Reference Pressure	0 [bar]
Buoyancy Model	Non Buoyant
Heat Transfer	Total Energy (Incl. Viscous Work Term)
Turbulence	Shear Stress Transport (medium intensity 5%)
Wall Functions	Automatic
Static Pressure	1 [bar]
Static Temperature	25 [°C]
Inlet Boundary Condition	
Flow Regime	Subsonic
Mass flux rate	$0.04 gs^{-1}$
Heat Transfer	Static Temperature 25 [°C]
Outlet Boundary Condition	
Flow Regime	Subsonic
Opening Pressure	1 [bar]
Opening Temperature	25 [°C]
Wall Boundary Condition	
Wall	No Slip, Smooth
Fixed Temperature	25 [°C]

Table 1. Configuration of the ANSYS CFX^{TM} flow solver for the test rig simulations.

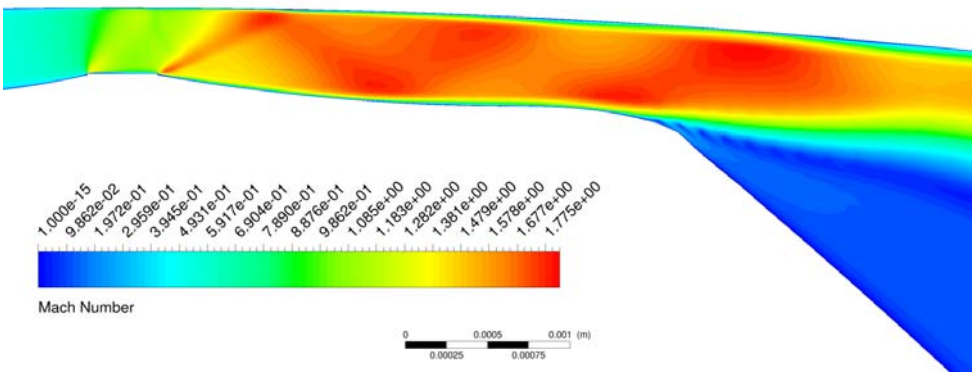
Acknowledgement
The authors would like to thank the anonymous reviewers for their valuable feedback. Support from Matthias Utri from TU Dortmund University by providing the test rig data and assisting in the ANSYS CFX^{TM} simulations is gratefully acknowledged. We finally thank Andrzej Jaeschke from TU Lodz for performing the isogeometric flow simulations with G+Smo.

This project (MOTOR) has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 678727.

(a) Pressure



(b) Mach number



(c) Temperature

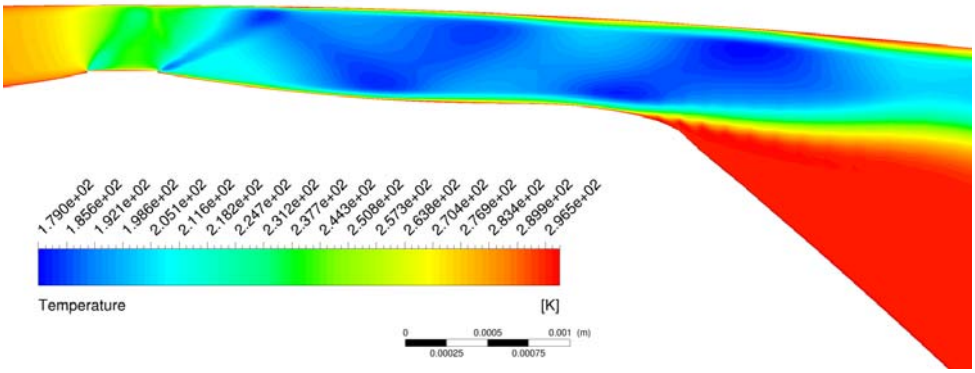


Figure 11. ANSYS CFX^{TM} solution computed on a block-structured hexahedral mesh: (a) pressure, (b) Mach number, and (c) temperature profile in the narrow gap of the test rig.

Notation

Ω	Exact physical domain (= screw machine geometry)
$\Gamma = \partial\Omega$	Boundary of the physical domain
Ω_h	Approximate computational physical domain
$\Gamma_h = \partial\Omega_h$	Boundary of the computational domain
$\hat{\Omega}$	Parameter domain of the geometry mapping
$\tilde{\Omega}$	Parameter domain of the re-parametrization mapping
$\mathbf{x} := (x, y)$	Coordinates in physical domains Ω and Ω_h
$\mathbf{u} := (u, v)$	Coordinates in parameter domain $\hat{\Omega}$
$\mathbf{s} := (s, t)$	Coordinates in parameter domain $\tilde{\Omega}$
U, V	Knot vector in horizontal and vertical direction, resp.
u_i, v_j	Knot entry of the open knot vector U and V, resp.
$N_{i,p}$	i-th B-Spline basis function of polynomial order p
B_I	I-th tensor-product B-Spline basis function
$\mathbf{F}$	Geometry mapping from $\hat{\Omega}$ to $\Omega_{(h)}$
$\mathbf{G}$	Reparameterization mapping from $\tilde{\Omega}$ to $\hat{\Omega}$
$\mathbf{c}_{i,j}$	Control points of the geometry mapping
$\mathbf{p}_a$	Discrete point in physical coordinates
Subscripts	
p, q	Polynomial order of B-Spline basis functions
i, j	Index of univariate B-Spline basis functions
I, J	Global index of basis functions
k, l	Patch number
u, v	First derivative in the parameter space
x, y	First derivative in the physical space
Superscripts	
$s \in (S, N, W, E)$	Face of a planar patch

Bibliography

[1] TwinMesh^{TM} `https://www.twinmesh.com`
[2] SCORG^{TM} `http://pdmanalysis.co.uk`
[3] ANSYS CFX `https://www.ansys.com/products/fluids/ansys-cfx`
[4] STAR-CCM+ `https://mdx.plm.automation.siemens.com/star-ccm-plus`
[5] Hughes T J, Cottrell J A and Bazilevs Y 2005 *Computer Methods in Applied Mechanics and Engineering* **194** 4135–4195
[6] Jüttler B, Langer U, Mantzaflaris A, Moore S E and Zulehner W 2014 *Proceedings in Applied Mathematics and Mechanics* **14** 961–962
[7] Sod G 1978 *Journal of Computational Physics* **27** 1 – 31
[8] Sachs R 2002 *Experimentelle Untersuchung von Gasströmungen in Schraubenmaschinen* Ph.D. thesis TU Dortmund University, Faculty of Mechanical Engineering
[9] Utri M, Höckenkamp S and Brümmer A 2018 *Proceedings of the 10th International Conference on Screw Machines (ICSM 2018)* (Dortmund, Germany)
[10] de Boor C 1971 Subroutine package for calculating with B-splines Tech. Rep. LA-4728-MS Los Alamos Scient. Lab.
[11] Hinz J, Möller M and Vuik C 2018 *Proceedings of the 10th International Conference on Screw Machines (ICSM 2018)* (Dortmund, Germany)
[12] Hinz J, Möller M and Vuik C 2018 *Computer Aided Geometric Design* **65** 48–75
[13] Hinz J, Möller M and Vuik C 2018 *Proceedings of the 7th European Conference on Computational Fluid Dynamics (ECFD 2018)* (Glasgow, United Kingdom)
[14] Thompson J F, Soni B K and Weatherill N P 1998 *Handbook of grid generation* (CRC press)
[15] Azarenok B N 2009 *Computational Mathematics and Mathematical Physics* **49** 797–809
[16] Kuzmin D, Möller M and Gurris M 2012 *Flux-Corrected Transport* (Springer) chap Algebraic flux correction II. Compressible flow problems
[17] Fletcher C 1983 *Computer Methods in Applied Mechanics and Engineering* **37** 225–243
[18] Möller M and Jaeschke A 2018 *Proceedings of the 19th International Conference on Finite Elements in Flow Problems (FEF 2017)* (Rome, Italy)
[19] Gottlieb S, Shu C and Tadmor E 2001 *SIAM Review* **43** 89–112
[20] Jaeschke A and Möller M 2018 *Proceedings of the 19th International Conference on Finite Elements in Flow Problems (FEF 2017)* (Rome, Italy)
[21] Wesseling P 2001 *Principles of Computational Fluid Dynamics* (*Springer Series in Computational Mathematics* vol 29) (Springer)
[22] Verstraete T 2016 The VKI U-bend optimization test case Tech. rep. Von Karman Institute for Fluid Dyanmics
[23] Shamanskiy A and Simeon B 2018 *Proceedings of the 10th International Conference on Screw Machines (ICSM 2018)* (Dortmund, Germany)