Generating Admissible Space-Time Meshes for Moving Domains in $d + 1$-Dimensions

Elias Karabelas
Institute of Biophysics, Medical University of Graz
Harrachgasse 21/IV, 8010 Graz, Austria

Martin Neumüller
Institute of Computational Mathematics, Johannes Kepler University
Altenberger Str. 69, 4040 Linz, Austria

NuMa-Report No. 2015-07
May 2015
Technical Reports before 1998:

1995

95-1  Hedwig Brandstetter  
      *Was ist neu in Fortran 90?*  
      March 1995

95-2  G. Haase, B. Heise, M. Kuhn, U. Langer  
      *Adaptive Domain Decomposition Methods for Finite and Boundary Element Equations.*  
      August 1995

95-3  Joachim Schöberl  
      *An Automatic Mesh Generator Using Geometric Rules for Two and Three Space Dimensions.*  
      August 1995

1996

96-1  Ferdinand Kickinger  
      *Automatic Mesh Generation for 3D Objects.*  
      February 1996

96-2  Mario Goppold, Gundolf Haase, Bodo Heise und Michael Kuhn  
      *Preprocessing in BE/FE Domain Decomposition Methods.*  
      February 1996

96-3  Bodo Heise  
      *A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation.*  
      February 1996

96-4  Bodo Heise und Michael Jung  
      *Robust Parallel Newton-Multilevel Methods.*  
      February 1996

96-5  Ferdinand Kickinger  
      *Algebraic Multigrid for Discrete Elliptic Second Order Problems.*  
      February 1996

96-6  Bodo Heise  
      *A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation.*  
      May 1996

96-7  Michael Kuhn  
      *Benchmarking for Boundary Element Methods.*  
      June 1996

1997

97-1  Bodo Heise, Michael Kuhn und Ulrich Langer  
      *A Mixed Variational Formulation for 3D Magnetostatics in the Space $H(\text{rot}) \cap H(\text{div})$*  
      February 1997

97-2  Joachim Schöberl  
      *Robust Multigrid Preconditioning for Parameter Dependent Problems I: The Stokes-type Case.*  
      June 1997

97-3  Ferdinand Kickinger, Sergei V. Nepomnyaschikh, Ralf Pfau, Joachim Schöberl  
      *Numerical Estimates of Inequalities in $H^2$.*  
      August 1997

97-4  Joachim Schöberl  
      *Programmbeschreibung NAOMI 2D und Algebraic Multigrid.*  
      September 1997

From 1998 to 2008 technical reports were published by SFB013. Please see  
http://www.sfb013.uni-linz.ac.at/index.php?id=reports

From 2004 on reports were also published by RICAM. Please see  
http://www.ricam.oeaw.ac.at/publications/list/

For a complete list of NuMa reports see  
http://www.numa.uni-linz.ac.at/Publications/List/
Generating admissible space-time meshes for moving domains in $d + 1$-dimensions

Elias Karabelas · Martin Neumüller

Abstract In this paper we present a discontinuous Galerkin finite element method for the solution of the transient Stokes equations on moving domains. For the discretization we use an interior penalty Galerkin approach in space, and an upwind technique in time. The method is based on a decomposition of the space-time cylinder into finite elements. Our focus lies on three-dimensional moving geometries, thus we need to triangulate four dimensional objects. For this we will present an algorithm to generate $d + 1$-dimensional simplex space-time meshes and we show under natural assumptions that the resulting space-time meshes are admissible. Further we will show how one can generate a four-dimensional object resolving the domain movement. First numerical results for the transient Stokes equations on triangulations generated with the newly developed meshing algorithm are presented.

Keywords finite elements · moving domains · four-dimensional mesh generation · parabolic PDE · space-time · discontinuous Galerkin

1 Introduction

The finite element approximation of transient partial differential equations is in most cases based on explicit or implicit time discretization schemes. In particular the simultaneous consideration of different time steps requires an appropriate interpolation to couple the solutions at different time levels. Especially for spatial domains with a moving boundary one encounters various numerical difficulties. One usually relies on an arbitrary Lagrangian-Eulerian formulation. See for example the recent article [13] and references therein for an overview of the ongoing discussion. In this paper we consider the application of finite elements in the whole space-time cylinder $Q$. By this we mean a decomposition of $Q$ into simplical elements. Therefore one replaces the problem of time discretization by a meshing problem. Having this, one can resolve the possible movement of the domain $\Omega$ directly. Simplicial space-time meshes have advantages over tensor-product meshes, since it is easier to decompose complex space-time meshes by those elements.

Space-time finite element methods have been applied to several parabolic model problems. Least square methods for convection-diffusion problems are considered, e.g., in [5,7] and for flow problems, e.g., in [17, 25–28, 22]. Discontinuous Galerkin finite element methods have been applied to solve transient convection-diffusion problems in [24], for fluid dynamics see [30] and problems from solid mechanics see [18, 2, 10, 1]. Very recently a paper concerning the generation of 4D simplicial meshes from a sequence of 3D MRI data has been considered in [12]. Also rather recently, the X-FEM method has been considered in the space-time setting see [14]. In most cases, the time dependent equation is discretized in the space-time domain on space-
time slabs. This allows for local mesh refinement in the space-time domain, see for example [27].

In this paper we consider, similar to [7], a decomposition of the space-time cylinder into simplicial finite elements. In particular for spatial domains \( \Omega \subseteq \mathbb{R}^3 \) the space-time cylinder is a four-dimensional object, which has to be decomposed into finite elements.

In [7], a method based on Delaunay’s algorithm is given to construct a four-dimensional triangulation out of a given decomposition of the spatial domain \( \Omega \). This method relies on the extension of the given finite elements of the triangulation of \( \Omega \) to four-dimensional prisms. Afterwards a random perturbation of the resulting points is made, to ensure the admissibility of the resulting four dimensional mesh. Here, we present a different approach using similar ideas. Our method does not rely on random perturbations. Furthermore we can ensure and proof that the resulting mesh is admissible and we can also include movements of the domain boundary. We want to stress out, that our approach is possible and we can also include movements of the domain boundary. For any \( t \in (0, T) \) let \( \Omega(t) \subseteq \mathbb{R}^d \), \( d = 1, 2, 3 \) be a bounded Lipschitz domain with boundary \( \Gamma(t) := \partial \Omega(t) \). We assume that the boundary \( \Gamma(t) \) admits the following decomposition for every \( t \in (0, T) \)

\[
\Gamma(t) = \Gamma_D(t) \cup \Gamma_R(t).
\]

We assume that the movement of the domain \( \Omega(t) \) is known for every \( t \in [0, T] \). We define the space-time cylinder \( Q \) as

\[
Q := \{(x, t) \in \mathbb{R}^{d+1} : x \in \Omega(t) \ t \in (0, T)\}.
\]

Further we define the space-time mantle \( \Sigma \) as

\[
\Sigma := \{(x, t) \in \mathbb{R}^{d+1} : x \in \Gamma(t) \ t \in (0, T)\}.
\]

The decomposition (1) induces

\[
\Sigma = \Sigma_D \cup \Sigma_R.
\]

The model problem we intend to study is governed by the transient Stokes equations. It reads as find \((u, p)\) such that

\[
\begin{align*}
\frac{\partial}{\partial t} u - \nu \Delta u + \nabla p &= f \quad \text{in } Q, \\
\text{div}(u) &= 0 \quad \text{in } Q, \\
u u = g_D & \quad \text{on } \Sigma_D, \\
\nabla u \cdot n + \alpha_R u - \nu n &= g_R \quad \text{on } \Sigma_R, \\
u u &= u_0 \quad \text{on } \Sigma_0 := \Omega(0).
\end{align*}
\]

Remark 1 In the case of a non-moving domain the definition of \( Q \) and \( \Sigma \) simplifies to

\[
Q := \Omega \times (0, T), \\
\Sigma := \partial \Omega \times (0, T).
\]

For deriving a discrete variational formulation we need to decompose the space-time cylinder \( Q \) into simplicial elements, see [20]. Let \( T_h \) be a sequence of decompositions

\[
\overline{Q} = T_h = \bigcup_{k=1}^{N} \tau_k
\]

into finite elements of mesh size \( h_k \). For \( d = 1 \) we have triangles, for \( d = 2 \) we use tetrahedrons and for \( d = 3 \) pentatopes are chosen. The generation of such triangulations from a given triangulation of \( \Omega(0) \) is not trivial. We will address this topic in Section 3.

Definition 1 (Admissible decomposition) A decomposition \( T_h \) is called admissible iff the non-empty intersection of the closure of two finite elements is either an edge (for \( d = 1, 2, 3 \)), a triangle (for \( d = 2, 3 \)) or a tetrahedron (for \( d = 3 \)).

It is worth noting that discontinuous Galerkin methods are not restricted to admissible decompositions. However one needs additional technical assumptions, see [9].

Definition 2 (Interior facet) Let \( T_h \) be a decomposition of \( Q \) into finite elements \( \tau_k \). For two neighboring elements \( \tau_k, \tau_l \in T_h \) we call

\[
\Gamma_{kl} := \tau_k \cap \tau_l
\]

an interior facet iff \( \Gamma_{kl} \) is a \( d \)-dimensional manifold. The set of all interior facets will be defined as \( \mathcal{I}_h \).
Any interior element $I_{kl}$ has an exterior normal vector $\mathbf{n}_{kl}$ with a non-unique direction. We fix the direction of the normal vector such that $\mathbf{n}_{kl}$ is the exterior normal vector of the element $\tau_k$ when $k < l$. So the direction of the normal vector $\mathbf{n}_{kl}$ depends on the ordering of the finite elements, but the variational formulation which we are going to use will be independent of this ordering.

**Definition 3** Let $I_{kl} \in I_h$ be an interior facet with outer normal $\mathbf{n}_h = (\mathbf{n}_{x,k}, \mathbf{n}_{t,k})^T \in \mathbb{R}^{d+1}$ for $\tau_k$ and $\mathbf{n}_l = -\mathbf{n}_k$ for $\tau_l$. For a given function $\phi$ smooth enough restricted to either $\tau_k$ or $\tau_l$ one defines:

- The **jump** across $I_{kl}$ as 

  $$ [\phi]_{kl} := \phi|_{\tau_k} \mathbf{n}_k + \phi|_{\tau_l} \mathbf{n}_l. $$

- The **space jump** across $I_{kl}$ as 

  $$ [\phi]_{x,kl} := \phi|_{\tau_k} \mathbf{n}_{x,k} + \phi|_{\tau_l} \mathbf{n}_{x,l}. $$

- The **time jump** across $I_{kl}$ as 

  $$ [\phi]_{t,kl} := \phi|_{\tau_k} \mathbf{n}_{t,k} + \phi|_{\tau_l} \mathbf{n}_{t,l}. $$

- The **average** of $\phi$ on $I_{kl}$ as 

  $$ \langle \phi \rangle_{kl} := \frac{1}{2} \left( \phi|_{\tau_k} + \phi|_{\tau_l} \right). $$

- The **upwind** in direction of $\phi$ is defined as 

  $$ \{\phi\}_{kl} := \begin{cases} 
  \phi|_{\tau_k} & \text{if } n_{k,t} > 0 \\
  0 & \text{if } n_{k,t} = 0 \\
  \phi|_{\tau_l} & \text{if } n_{k,t} < 0 
  \end{cases}. $$

Let $p, q \in \mathbb{N}_0$. Then one defines the spaces of piecewise polynomials 

$$ V^p_h := [S^p_h(T_h)]^d = \left\{ \mathbf{v}_h \in [L^2(Q)]^d : \mathbf{v}_h|_{\tau} \in [P_p(\tau)]^d \right\} $$

for all $\tau \in T_h$, $\mathbf{v}_h|_{\Sigma_{p}} = 0$, and 

$$ Q^q_h := \left\{ q_h \in L^2(Q) : q_h|_{\tau} \in P_q(\tau) \text{ for all } \tau \in T_h \right\}. $$

Inspired by works in [19,21] we will use the following bilinear form defined for $\mathbf{u}_h, \mathbf{v}_h \in V^p_h(T_h)$:

$$ A(\mathbf{u}_h, \mathbf{v}_h) := b_T(\mathbf{u}_h, \mathbf{v}_h) + a_h(\mathbf{u}_h, \mathbf{v}_h). $$

The individual components read as 

$$ a_h(\mathbf{u}_h, \mathbf{v}_h) := \nu \sum_{l=1}^{N} \int_{\Sigma_{l}} \nabla_{x} \mathbf{u}_h \cdot \nabla_{x} \mathbf{v}_h \, dq $$

and 

$$ b_T(\mathbf{u}_h, \mathbf{v}_h) := \sum_{l=1}^{N} \int_{\tau_l} \mathbf{u}_h \cdot \frac{\partial}{\partial t} \mathbf{v}_h \, dq + \int_{\Sigma_{T}} \mathbf{u}_h \cdot \mathbf{v}_h \, ds_q $$

for a given velocity stabilization parameter $\sigma_v > 0$. Furthermore we define the following pressure bilinear forms 

for $\mathbf{v}_h \in V^p_h(T_h)$ and $(p_h, q_h) \in Q^q_h(T_h)$:

$$ b_p(\mathbf{v}_h, p_h) := \sum_{l=1}^{N} \int_{\tau_l} p_h \text{div}(\mathbf{v}_h) \, dq $$

$$ + \sum_{l \in I_{kl} \in I_h} \int_{I_{kl}} \{ \mathbf{u}_h \}^\tau \{}[v_h]_{\Gamma_{kl,t}} \, ds_q $$

for a given pressure stabilization parameter $\sigma_p$. In all the bilinear forms defined above we have used $\Gamma_{kl} := \frac{1}{2}(n_k + n_l)$. Hence we have to find $\mathbf{u}_h^0 \in V^p_h(T_h)$ and 

$$ p_h \in Q^q_h(T_h) $$

such that 

$$ A(\mathbf{u}_h^0, \mathbf{v}_h) - b_p(\mathbf{v}_h, p_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle_Q + \langle \mathbf{u}_0, \mathbf{v}_h \rangle $$

$$ - A(\mathbf{u}_h, \mathbf{v}_h), $$

$$ b_p(\mathbf{u}_h^0, q_h) + d_p(p_h, q_h) = -b_p(\mathbf{u}_h^0, q_h). $$

Here we used an discrete extension $\mathbf{u}_g^h$ of the given Dirichlet data, for example a $L^2$-projection.

### 3 Triangulations in $d + 1$ dimensions

In this section we will introduce an algorithm to decompose a hyper prism into simplices to generate a $d + 1$ simplex space-time mesh. Moreover we will show that the resulting mesh is admissible if the nodes of the simplices from the initial mesh are sorted in a special way.

#### 3.1 Tensor product extensions

A simple idea for constructing a space-time mesh for a given three-dimensional simplicial spatial mesh is to extrude the mesh in time direction by a tensor product extension, see also Figure 1. Afterwards we decompose the upcoming prisms or so called hyperprisms into simplicial elements.

Before we can start we need a precise definition of a $d$-dimensional simplex.
Definition 4 (d-dimensional simplex) Let 
\[ \{p_1, \ldots, p_{d+1}\} \subset \mathbb{R}^d, \]
d \( \in \mathbb{N} \) be a set of nodes, then a d-dimensional simplex \( S_d \) is defined as 
\[ S_d := \{p_1, \ldots, p_{d+1}\} := \text{conv}(\{p_1, \ldots, p_{d+1}\}), \]
where \( \text{conv}(\cdot) \) is the convex hull of a set of nodes. Note that we also fix the ordering of the nodes in the definition of a d-dimensional simplex.

Now we can extrude one simplex in time direction and we obtain the following definition.

Definition 5 (Hyperprism) For a given simplex \( S_d = [p_1, \ldots, p_{d+1}] \) the tensor product extension in time direction for a given time interval \( [0, \tau] \) or the so called hyperprism \( H_{d+1} \) is given by
\[ P_{d+1} := [p_1, \ldots, p_{d+1}; \tau] := \text{conv}(\{p_1, \ldots, p_{d+1}^\tau, \ldots, p_{d+1}\}) \subset \mathbb{R}^{d+1}, \]
with
\[ p_i := (p_i^\top, 0)^\top, \]
\[ p_i^\tau := (p_i^\top, \tau)^\top, \]
for \( i = 1, \ldots, d+1. \)

3.2 Decomposing Hyperprisms

In this section we will give an algorithm to decompose the hyperprisms given in Definition 5 into simplices.

Definition 6 (Decomposed hyperprism) Let \( S_d \) be a given simplex and \( P_{d+1} \) the hyperprism with respect to the simplex \( S_d \) and \( \tau > 0 \). Then we define the following simplices
\[ S_{d+1}^1 := [p_1', p_2', p_3', \ldots, p_{d+1}', p_1''], \]
\[ S_{d+1}^2 := [p_2', p_3', \ldots, p_{d+1}', p_1'', p_2''], \]
\[ S_{d+1}^3 := [p_3', \ldots, p_{d+1}', p_1'', p_2', p_3''], \]
\[ \vdots \]
\[ S_{d+1}^{d+1} := [p_{d+1}', p_1'', p_2', p_3'', \ldots, p_{d+1}'', p_1''']. \]

Furthermore we define the set of simplices \( T_P(S_d, \tau) := \{S_{d+1}^1, \ldots, S_{d+1}^{d+1}\} \).

Note, that the ordering of the nodes of a hyperprism \( P_{d+1} \) is essential for the resulting decomposition (6). In order to ensure that the simplices \( S_{d+1}^1, \ldots, S_{d+1}^{d+1} \) defined in (6) decompose the hyperprism \( P_{d+1} \) we need the following lemma.

Lemma 1 Let \( P_{d+1} \) be some given hyperprism with respect to the simplex \( S_d \) and \( \tau > 0 \). Then the set of simplices
\[ T_P(S_d, \tau) := \{S_{d+1}^1, \ldots, S_{d+1}^{d+1}\} \]
defined in (6) is an admissible decomposition of the hyperprism \( P_{d+1} \).

Proof By construction the set of simplices \( T_P(S_d, \tau) = \{S_{d+1}^1, \ldots, S_{d+1}^{d+1}\} \) is admissible. Furthermore, every simplex \( S_{d+1}^i \) for \( i = 1, \ldots, d+1 \) is contained in the hyperprism \( P_{d+1} \) since \( P_{d+1} \) is convex. It remains to show, that the union of all simplices \( T_P(S_d, \tau) \) is equal to the hyperprism, i.e. we have to show, that the volume of the union of all simplices \( T_P(S_d, \tau) \) coincides with the volume of the hyperprism. To do so, we transform the hyperprism \( P_{d+1} \) to a reference hyperprism \( \tilde{P}_{d+1} \) where we easily can compute all the volume terms. For this, we define the reference Simplex \( \tilde{S}_d \subset \mathbb{R}^d \) as
\[ \tilde{S}_d := [e_0, e_1, \ldots, e_d] = \text{conv}(\{e_0, e_1, \ldots, e_d\}), \]
with
\[ e_0 := (0, 0, \ldots, 0)^\top, \]
\[ e_1 := (1, 0, \ldots, 0)^\top, \]
\[ e_2 := (0, 1, \ldots, 0)^\top, \]
\[ \vdots \]
\[ e_d := (0, 0, \ldots, 1)^\top. \]

Then we define the reference hyperprism \( \tilde{P}_{d+1} \) as
\[ \tilde{P}_{d+1} := [e_0, \ldots, e_{d+1}; 1]. \]

With the standard affine transformation we have a bijective mapping between the reference hyperprism \( P_{d+1} \) and the hyperprism \( P_{d+1} \). This affine transformation consists of the standard transformation for d-dimensional simplices and a scaling in time direction. So we only have to compare the volume for the reference hyperprism. Now the volume of the reference simplex \( \tilde{S}_d \) is given by \( \frac{\tilde{S}_d}{d!} \). Hence the volume of the reference hyperprism is
\[ |\tilde{P}_{d+1}| = \frac{1}{d!}. \]
The simplices of our decomposition in the reference domain are given by

\[
S^{1}_{d+1} := [e_0, e_1', e_2', \ldots, e_d', e_0'], \\
S^{2}_{d+1} := [e_1', e_2', \ldots, e_d', e_0''', e_1''', e_2'''], \\
S^{3}_{d+1} := [e_2', e_3', \ldots, e_d', e_0'', e_1'', e_2''], \\
\vdots \\
S^{d+1}_{d+1} := [e_d', e_0'', e_1'', e_2'', \ldots, e_d'].
\]

It is easy to see, that these simplices have the same volume, i.e.

\[
|S^{i}_{d+1}| = \frac{1}{(d+1)!}, \quad \text{for } i = 1, \ldots, d+1.
\]

Hence we have

\[
\bigcup_{i=1}^{d+1} S^{i}_{d+1} = (d+1) \left( \frac{1}{(d+1)!} \right) = \frac{1}{d!} = |\hat{P}_{d+1}|,
\]

which completes the proof.

### 3.3 Admissible tensor product triangulations

For a given d-dimensional triangulation $T_h$, we now want to construct a tensor product extension by applying the algorithm (6) for every simplex of the simplicial mesh $T_h$. With Lemma 1 we know that, every hyper prism can be decomposed admissible into simplices. In this section we want to formulate conditions such that the overall space-time mesh is admissible. For this we need that the nodes of the simplices are ordered in a special way.

**Definition 7 (Consistently numbered)** Let

\[
T_h = \{S^i_d : S^i_d = [p^i_1, \ldots, p^i_{d+1}]\},
\]

be an admissible d-dimensional simplex mesh. Then $T_h$ is called **consistently numbered**, if for any two simplices $S^i_d, S^j_d \in T_h$ with non-empty intersection, i.e. $S^i_d \cap S^j_d \neq \emptyset$, there exists indices $k_1 < \ldots < k_n$ and $\ell_1 < \ldots < \ell_n$ with $n \in \mathbb{N}, n \leq d+1$, such that

\[
S^i_d \cap S^j_d = [p^i_{k_1}, \ldots, p^i_{k_n}] \equiv [p^j_{\ell_1}, \ldots, p^j_{\ell_n}].
\]

Here "$=\$" means that the two sets are the same and "$\equiv\$" means that the two sets are equal and that also the numbering of the nodes is the same, i.e.

\[
p^i_{k_1} = p^j_{\ell_1}, \ldots, p^i_{k_n} = p^j_{\ell_n}.
\]

The definition of a consistently numbered triangulation can also be found in [8] and it is important for the refinement of d-dimensional simplices, especially for $d \geq 4$. If an admissible mesh is consistently numbered we can prove the next Theorem.

**Theorem 1** Let $T_h$ be an admissible d-dimensional triangulation which is consistently numbered and let $\tau > 0$. Furthermore let

\[
T_{h,\tau} := \{T_P(S^i_d, \tau) : S^i_d \in T_h\}
\]

be the (d + 1)-dimensional simplex mesh resulting by decomposing every hyper prism with the algorithm given in (6). Then the space-time mesh $T_{h,\tau}$ is admissible.

**Proof** With Lemma 1 we know, that every hyper prism is decomposed admissible into simplices. To obtain a global admissible mesh we have to prove, that the tensor product triangulations $T_P(S^i_d, \tau)$ and $T_P(S^j_d, \tau)$ for each neighboring elements $S^i_d, S^j_d \in T_h$ are matching. Let $S^i_d = [p^i_1, \ldots, p^i_{d+1}]$ and $S^j_d = [p^j_1, \ldots, p^j_{d+1}]$ with $S^i_d, S^j_d \in T_h$ be some neighboring simplices and $P_d := P^i_{d+1} \cap P^j_{d+1}$, with $P^i_{d+1} := [p^i_1', \ldots, p^i_{d+1}; \tau]$ and $P^j_{d+1} := [p^j_1', \ldots, p^j_{d+1}; \tau]$ be the intersecting hyper prism and

\[
T_P := \{S_{d+1} \cap P_d : S^i_d \in T_h(S^i_d, \tau)\}, \\
T_P := \{S_{d+1} \cap P_d : S^j_d \in T_h(S^j_d, \tau)\},
\]

be the corresponding triangulations of $P_d$ obtained by $T_P(S^i_d, \tau)$ and $T_P(S^j_d, \tau)$. It remains to show, that the intersecting hyper prism $P_d$ is decomposed in the same way from both sides, i.e. that $T_h = T_P$. Since $T_h$ is consistently numbered, there exists indices $k_1 < \ldots < k_n$ and $\ell_1 < \ldots < \ell_n$ with $n = d$, such that

\[
S^i_d \cap S^j_d \equiv S^i_d := [p^i_{k_1}, \ldots, p^i_{k_n}], \\
S^j_d \equiv S^j_d := [p^j_{\ell_1}, \ldots, p^j_{\ell_n}].
\]

Therefore, the intersecting simplex $S^i_d \cap S^j_d$ is obtained by simply removing the nodes from $S^i_d$ or $S^j_d$ which are not shared together and furthermore they have the same ordering of the nodes. For the intersecting hyper prism $P_d$ the decompositions from both sides $T_P$ and $T_P$ are given by removing the nodes which are not shared together from the formula (6) and with (7) we have

\[
T_P = T_P(S^{i-1}_{d-1}, \tau) \quad \text{and} \quad T_P = T_P(S^{j-1}_{d-1}, \tau).
\]

Since in equation (7) also the node ordering of $S^i_{d-1}$ and $S^j_{d-1}$ is the same we also obtain

\[
T_P(S^{i-1}_{d-1}, \tau) = T_P(S^{j-1}_{d-1}, \tau),
\]

which implies that $T_P = T_P$.

**Remark 2** To obtain an admissible space-time mesh $T_{h,\tau}$ we only have to ensure, that the nodes of the spatial mesh $T_h$ are consistently numbered. This can be easily obtained by sorting for each simplex $S^i_d \in T_h$ the local nodes with respect to the global node numbers.
3.4 Tensor product triangulations for moving domains

If the movement of a computational domain is known in advance we can generate admissible space-time meshes by applying the methods from above. The idea is to move the points at the top of the tensor-product extension. Assuming that the displacement of points on the boundary $\Gamma(t)$ is governed by a function

$$g_{\text{mov}}(X,t) : \Gamma(0) \times (0,T) \to \mathbb{R}^d.$$ 

Then a point $x \in \Gamma(t)$ can be written as

$$x = X + g_{\text{mov}}(X,t),$$

where $X \in \Gamma(0)$. Recall the definition of a hyperprism in Definition 5. Instead of using $p_i^e \equiv (p_i^1, \tau)^\top$ on the surface we can apply the displacement and use $p_i^e \equiv (p_i^1 + g_{\text{mov}}(p_i, \tau)^\top, \tau)^\top$ for all boundary points of the simplex mesh that are subject to a movement. The remaining generation of the 4D mesh stays untouched. For boundary movements that are of small magnitude and do not change the topology of the initial geometry this can be sufficient. For stronger yet topology preserving movements this concept would create degenerating simplex elements. A remedy to this is to use the movement $g_{\text{mov}}$ as Dirichlet datum for a vector Laplacian or a linear elasticity problem. Then the resulting displacement is applied to all simplex points in the domain. For more on mesh smoothing we refer to [16, 15].

In the case of stronger displacements or even topology changes re-meshing would be required and we need further meshing algorithms to connect different spatial domains in space and time, especially for four-dimensional space-time meshes this remains a future research topic.

If the movement of the computational domain is not known in advance we can solve the problem on a coarse spatial grid with coarse time steps to obtain a coarse approximation for the movement. Afterwards we can construct the coarse space-time mesh with the methods given in this work. By using adaptive schemes in space and time we further can refine the space-time domain adaptively and move the points in the space time domain by the computed finer approximations. Note that the movement of the points has to be only done in the range of the approximation error, which is usually small. Of course this is also considered as a further research topic.

3.5 Visualization

Here we want to address the issue of visualizing results for four-dimensional triangulations $\mathcal{T}_h$. In applications it is desired to visualize results at given time instances $t_k \in [0,T]$. The main idea is to cut the decomposition $\mathcal{T}_h$ into a finite number of three-dimensional manifolds. For this we need to have a hyperplane to calculate the intersections with the decomposition.

**Definition 8 (Hyperplane)** Let $p_0 \in \mathbb{R}^4$ be arbitrary and let $p_1, p_2, p_3$ and $p_4 \in \mathbb{R}^4$ be linear independent. Then the set

$$H_4 := \left\{ x \in \mathbb{R}^4 : x = p_0 + \mu_1 p_1 + \mu_2 p_2 + \mu_3 p_3 \right\}$$

for $\mu_1, \mu_2, \mu_3 \in \mathbb{R}$

is called a hyperplane.

To cut a given decomposition $\mathcal{T}_h$ with a hyperplane $H_4$, we have to cut every element $\tau_k \in \mathcal{T}_h$ with the hyperplane. For this we have to calculate for every edge $e_i = (x_{i1}, x_{i2})$, $i = 1, \ldots, 10$ of $\tau_k$, the intersection with the hyperplane. A point $x \in e_i$ can be written as

$$x = x_{i1} + \lambda(x_{i2} - x_{i1})$$

for a given $\lambda \in [0,1]$. Hence, an intersection point $\xi_i$ of the edge $e_i$ with the hyperplane $H_4$ has to satisfy

$$x_{i1} + \lambda(x_{i2} - x_{i1}) = p_0 + \mu_1 p_1 + \mu_2 p_2 + \mu_3 p_3$$

or in matrix notation

$$A_i := \begin{pmatrix} p_1 & p_2 & p_3 & x_{i1} - x_{i2} \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \lambda \end{pmatrix} = x_{i1} - p_0.$$

The matrix $A_i$ is invertible iff the vector $x_{i1} - x_{i2}$ is linear independent to the vectors $p_1, p_2, p_3$. In fact, the matrix $A_i$ is not invertible if the edge $e_i$ is parallel to the hyperplane $H_4$. In this case there exists either no intersection point or infinitely many. If the matrix is invertible we can calculate the coefficients $\mu_1, \mu_2, \mu_3$ and $\lambda \in \mathbb{R}$ uniquely. Let $D_k$ denote the set of all intersection points of the element $\tau_k \in \mathcal{T}_h$ with the hyperplane $H_4$.

We distinguish two relevant cases

1. If $|D_k| = 4$, then the intersection points form a tetrahedron
2. If $|D_k| = 6$, then the intersection points for a general irregular prism.

If we use the special vectors

$$p_0 := t_e e_t, \quad p_1 := e_x, p_2 := e_y, p_3 := e_z$$

for a given $t_e \in [0,T]$ we can now calculate a three-dimensional object which can be visualized with existing software tools for example [4].
4 Numerical Results

In this section we will present first numerical examples. Starting point is the discrete variational formulation (3)-(5). This can be equivalently written as the following block system

\[
\begin{pmatrix}
K_h - B_h^T D_h & U \\
B_h D_h & P
\end{pmatrix} = \begin{pmatrix}
F_1 \\
F_2
\end{pmatrix}.
\]

(8)

It is worth noting, that due to the discretization of the time derivative we have that \(K_h \neq K_h^T\). The four dimensional computational geometries as well as the resulting linear systems in the subsequent numerical examples were solved with the software package Neshmet developed by the authors. In particular we used a preconditioned GMRes method. As preconditioner we used the following:

\[
P := \begin{pmatrix}
K_h \\
S_h
\end{pmatrix}
\]

(9)

where \(K_h\) is chosen as a component-wise algebraic multi-grid and \(S_h\) is chosen as ILU(2)-factorization of \(D_h + B_h \text{diag}(K_h)^{-1} B_h^T\). These preconditioners were taken from the HYPRE library [11].

4.1 Robin Boundary Conditions for Simulating Valves

In the subsequent examples we want to simulate opening and closing valves. This means, that we need to switch between and inflow and a no-slip condition. To this end we used the following configuration for (2): We set \(g_R \equiv 0\). Further we use the following Robin coefficients for outflow

\[
\alpha_R(x, t) := \begin{cases} 
10^6 & \text{if } t \in [0, \frac{1}{2}) \\
0 & \text{if } t \in [\frac{1}{2}, 1],
\end{cases}
\]

and the following for inflow:

\[
\alpha_R(x, t) := \begin{cases} 
0 & \text{if } t \in [0, \frac{1}{2}) \\
10^6 & \text{if } t \in [\frac{1}{2}, 1],
\end{cases}
\]

4.2 First Example

In the first example we consider Stokes flow in a diaphragm pump. The geometry consists of the intersection of two cylinders. The first one has its main axis aligned with the \(z\)-axis with a radius of 0.8 and ranges between \(z = -0.4\) and \(z = 0.4\). The second cylinder has its main axis aligned with the \(x\)-axis with a radius of 0.2 and ranges from \(x = -1\) to \(x = 1\). A front view of the geometry is depicted in Figure 2. The movement of the boundary \(\Gamma(t)\) was prescribed as follows

\[
g_{\text{mov}}(t, X) := (0.4 + \sin^2(\pi t) \left(1 - \frac{X(0)^2 + X(1)^2}{0.75^2}\right)) e_z - X
\]

for \(X \in \Gamma_{D,m}(0)\) and 0 else. The following boundary conditions are used:

- \(u = 0\) on \(\Gamma_D\)
- \(u = \frac{\partial}{\partial t} g_{\text{mov}}\) on \(\Gamma_{D,m}(t)\)
- On \(\Gamma_{R,\text{in}}\) and \(\Gamma_{R,\text{out}}\) we used the Robin boundary conditions discussed in Section 4.1
- The initial condition for \(u\) was set to \(u(0, x) = 0\).

The triangulation of the resulting 4D geometry was accomplished using the tools described in Section 3. The resulting mesh consisted of 951360 pentatopes.
degree for $u_h$ was set to $p = 1$ and $q = 0$ for the pressure variable $p_h$. This resulted in 14270400 degrees of freedom for $u_h$ and 951360 degrees of freedom for $p_h$. We needed 95 GMRes-iterations for achieving a relative error of $1E^{-5}$. In Figure 5 one can see the resulting flow and pressure at given time stamps, which were again produced by slicing the 4D geometry along the time axis.

4.3 Second Example

For the second example we considered a Y-shaped pipe. A schematic view is depicted in Figure 2. We prescribed the following movement of $\Gamma(t)$:

$$g_{\text{mov}}(X,t) = \begin{cases} 0 & \text{for } X \notin \Gamma_{D,m} \cup \Gamma_{D,m}^\infty \\ \frac{2}{\pi} g_{\text{mov}} & \text{for } X \in \Gamma_{D,m} \cup \Gamma_{D,m}^\infty \end{cases}$$

Some snapshots of the domain movement are depicted in Figure 8. The boundary conditions were set to

- $u = 0$ on $\Gamma_D \cup \Gamma_{D,m}(t)$
- $u = \frac{2}{\pi} g_{\text{mov}}$ on $\Gamma_{D,m}(t)$
- On $\Gamma_{R,\text{in}}$ and $\Gamma_{R,\text{out}}$ we used the Robin boundary conditions discussed in Section 4.1
- The initial condition for $u$ was set to $u(0,x) = 0$.

The resulting 4D mesh consisted of 2618880 pentatopes. With the same ansatz spaces as used for Example 1 we have 39283200 degrees of freedom for $u_h$ and 2618880 degrees of freedom for $p_h$. We needed 107 GMRes-iterations for achieving a relative error of $1E^{-5}$. In Figure 9 we have depicted some results.

Fig. 4: Snapshots of the Triangulations $\Omega(t)$

Fig. 6: Front view of the initial geometry $\Omega(0)$. Blue boundaries belong to $\Gamma_D$. The dashed line represents the plane $z = -3$. The height from top to bottom is 17. The base of the pipe is located at $z = -10$. The radius of the pipe is 3.

Fig. 7: Initial Triangulation $\Omega(0)$

Fig. 8: Snapshots of the Triangulations $\Omega(t)$

Fig. 9: Some results.
Generating admissible space-time meshes for moving domains in $d + 1$-dimensions

(a) $t = 0.1$

(b) $t = 0.35$

(c) $t = 0.7$

(d) $t = 1.0$

Fig. 5: Snapshots of the solution. Additionally we have cut along the $y$-axis.

5 Conclusions

In this paper we have presented a novel approach to construct four-dimensional triangulations for moving domains. This was done by extending the elements of the space triangulation into hyperprisms. Assuming a consistent numbered spatial triangulation we were able to proof that our algorithm produces admissible space-time meshes. We implemented the presented algorithm and applied it to solve the transient Stokes equations with a space-time discontinuous Galerkin finite element method. In the future one could start investigating Navier-Stokes equations. Furthermore, optimal control problems with time dependent partial differential equations render themselves interesting candidates for applying space-time methods, since one has to solve a forward and backward problem which are coupled in space and time. Another attractive aspect of general space-time meshes is the possibility to apply adaptive refinement strategies to resolve local behaviors in space and time. Considering solvers, one could think about domain decomposition approaches or space-time multigrid methods for example, which are a future research topic.

Acknowledgments

This research was supported by the grant F3210-N18 from the Austrian Science Fund (FWF).
Fig. 9: Snapshots of the solution. Additionally we have cut along the $y$-axis.

References

1. Abedi, R., Hawker, M., Haber, R., Matouš, K.: An adaptive spacetime discontinuous Galerkin method for cohesive models of elastodynamic fracture. Internat. J. Numer. Methods Engrg. 81, 1207–1241 (2010)
2. Abedi, R., Petracovici, B., Haber, R.B.: A space–time discontinuous galerkin method for linearized elastodynamics with element-wise momentum balance. Computer Methods in Applied Mechanics and Engineering 195(25), 3247–3273 (2006)
3. Arnold, D., Brezzi, F., Cockburn, B., Marini, L.: Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. SIAM J. Numer. Anal. 39, 1749–1779 (2002)
4. Ayachit., U.: The ParaView Guide: A Parallel Visualization Application. Kitware Inc., Clifton Park (2015)
5. Bank, R., Metti, M.: An error analysis of some higher order space-time moving finite elements. Computing and Visualization in Science 16(5), 219–229 (2013). DOI 10.1007/s00791-015-0235-1. URL http://dx.doi.org/10.1007/s00791-015-0235-1
6. Baumann, C., Oden, J.: A discontinuous hp finite element method for convection-diffusion problems. Comput. Methods Appl. Mech. Engrg. 175, 311–341 (1999)
7. Behr, M.: Simplex space–time meshes in finite element simulations. Internat. J. Numer. Methods Fluids 57, 1421–1434 (2008)
8. Bey, J.: Simplicial grid refinement: on Freudenthal’s algorithm and the optimal number of congruence classes. Numer. Math. 85, 1–29 (2000)
9. Buffa, A., Ortner, C.: Compact embeddings of broken Sobolev spaces and applications. IMA J. Numer. Anal. 29(4), 827–855 (2009). DOI 10.1093/imanum/dra038. URL http://dx.doi.org/10.1093/imanum/dra038
10. Dumont, S., Jourdan, F.: A space-time finite element method for elastodynamics problems with a moving loading zone. In: CST 2012 (2012)
11. Falgout, R., Jones, J., Yang, U.: The design and implementation of hypre, a library of parallel high performance preconditioners. Lect. Notes Comput. Sci. Eng. 51, 267–294 (2006)
12. Foteinos, P., Chrisochoides, N.: 4d space-time delaunay meshing for medical images. In: Proceedings of the 22nd International Meshing Roundtable, pp. 223–240. Springer (2014)
13. Gawlik, E.S., Lew, A.J.: High-order finite element methods for moving boundary problems with prescribed boundary evolution. Computer Methods in Applied Mechanics and Engineering 278, 314–346 (2014)
14. Lehrenfeld, C.: The nitsche xfem-dg space-time method and its implementation in three space dimensions. SIAM Journal on Scientific Computing 37(1), A245–A270 (2015)
15. Liakopoulos, P.I., Giannakoglou, K.C.: Unstructured remeshing using an efficient smoothing scheme approach.
Generating admissible space-time meshes for moving domains in $d+1$-dimensions

Fig. 8: Snapshots of the Triangulations $\Omega(t)$

In: ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS) (2006)

16. López, E.J., Nigro, N.M., Storti, M.A.: Simultaneous untangling and smoothing of moving grids. International journal for numerical methods in engineering 76(7), 994–1019 (2008)

17. Masud, A., Hughes, T.: A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems. Comput. Methods Appl. Mech. Engrg. 146, 91–126 (1997)

18. Miller, S.T., Kraczek, B., Haber, R.B., Johnson, D.D.: Multi-field spacetime discontinuous galerkin methods for linearized elastodynamics. Computer Methods in Applied Mechanics and Engineering 199(1), 34–47 (2009)

19. Neumüller, M.: Space-time methods: Fast solvers and applications. Ph.D. thesis, University of Graz (2013)

20. Neumüller, M., Steinbach, O.: A DG space–time domain decomposition method. Domain Decomposition Methods in Science and Engineering XX (2013)

21. Di Pietro, D., Ern, A.: Mathematical Aspects of Discontinuous Galerkin Methods. Springer, Berlin, Heidelberg (2012)

22. Rendell, T., Allen, C.B., Power, E.D.: Conservative unsteady aerodynamic simulation of arbitrary boundary motion using structured and unstructured meshes in time. International Journal for Numerical Methods in Fluids 70(12), 1518–1542 (2012)

23. Rivièrè, B.: Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations. Cambridge University Press (2008)

24. Sudirham, J., van der Vegt, J., van Damme, R.: Space–time discontinuous Galerkin method for advection–diffusion problems on time-dependent domains. Appl. Num. Math. 56, 1491–1518 (2006)

25. Tezduyar, T., Behr, M., Liu, J.: A new strategy for finite element computations involving moving boundaries and interfaces/the deforming-spatial-domain/space-time procedure: i. computation of freesurface flows, two-liquid flows, and flows with drifting cylinders. Computer methods in applied mechanics and engineering 94(3), 353–371 (1992)

26. Tezduyar, T., Behr, M., Mittal, S., Liu, J.: A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: ii. computation of freesurface flows, two-liquid flows, and flows with drifting cylinders. Computer methods in applied mechanics and engineering 94(3), 339–359 (1992)

27. Tezduyar, T., Sathe, S.: Enhanced-discretization space-time technique (EDSTT). Comput. Methods Appl. Mech. Engrg. 193, 1385–1401 (2004)

28. Tezduyar, T.E.: Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. Computer Methods in Applied Mechanics and Engineering 195(23), 2983–3000 (2006)

29. Thomée, V.: Galerkin Finite Element Methods for Parabolic Problems. Springer, New York (2006)

30. van der Vegt, J., Sudirham, J.: A space–time discontinuous galerkin method for the time-dependent oseen equations. Applied Numerical Mathematics 58(12), 1892–1917 (2008)
## Latest Reports in this series

### 2009 - 2013

[...]

### 2014

[...]

| Year | Number | Authors | Title | Date |
|------|--------|---------|-------|------|
| 2014-04 | Ulrich Langer, Sergey Repin and Monika Wolfmayr | Functional A Posteriori Error Estimates for Parabolic Time-Periodic Boundary Value Problems | July 2014 |
| 2014-05 | Irina Georgieva and Clemens Hofreither | Interpolating Solutions of the Poisson Equation in the Disk Based on Radon Projections | July 2014 |
| 2014-06 | Wolfgang Kreindl and Walter Zulehner | A Decomposition Result for Biharmonic Problems and the Hellan-Herrmann-Johnson Method | July 2014 |
| 2014-07 | Martin Jakob Gander and Martin Neumüller | Analysis of a Time Multigrid Algorithm for DG-Discretizations in Time | September 2014 |
| 2014-08 | Martin Jakob Gander and Martin Neumüller | Analysis of a New Space-Time Parallel Multigrid Algorithm for Parabolic Problems | November 2014 |
| 2014-09 | Helmut Gfrerer and Jiří V. Outrata | On Computation of Limiting Coderivatives of the Normal-Cone Mapping to Inequality Systems and their Applications | December 2014 |
| 2014-10 | Clemens Hofreither and Walter Zulehner | Spectral Analysis of Geometric Multigrid Methods for Isogeometric Analysis | December 2014 |
| 2014-11 | Clemens Hofreither and Walter Zulehner | Mass Smoothers in Geometric Multigrid for Isogeometric Analysis | December 2014 |

### 2015

| Year | Number | Authors | Title | Date |
|------|--------|---------|-------|------|
| 2015-01 | Peter Gangl, Ulrich Langer, Antoine Laurain, Houcine Meftahi and Kevin Sturm | Shape Optimization of an Electric Motor Subject to Nonlinear Magnetostatics | January 2015 |
| 2015-02 | Stefan Takacs and Thomas Takacs | Approximation Error Estimates and Inverse Inequalities for B-Splines of Maximum Smoothness | February 2015 |
| 2015-03 | Clemens Hofreither, Ulrich Langer and Steffen Weißer | Convection-adapted BEM-based FEM | February 2015 |
| 2015-04 | Helmut Gfrerer and Boris S. Mordukhovich | Complete Characterizations of Tilt Stability in Nonlinear Programming under Weakest Qualification Conditions | March 2015 |
| 2015-05 | Matúš Benko and Helmut Gfrerer | An SQP Method for Mathematical Programs with Complementarity Constraints with Strong Convergence Properties | May 2015 |
| 2015-06 | Irina Georgieva and Clemens Hofreither | An Algorithm for Low-rank Approximation of Bivariate Functions Using Splines | May 2015 |
| 2015-07 | Elias Karabelas and Martin Neumüller | Generating Admissible Space-Time Meshes for Moving Domains in d + 1-Dimensions | May 2015 |

From 1998 to 2008 reports were published by SFB013. Please see [http://www.sfb013.uni-linz.ac.at/index.php?id=reports](http://www.sfb013.uni-linz.ac.at/index.php?id=reports)

From 2004 on reports were also published by RICAM. Please see [http://www.ricam.oeaw.ac.at/publications/list/](http://www.ricam.oeaw.ac.at/publications/list/)

For a complete list of NuMa reports see [http://www.numa.uni-linz.ac.at/Publications/List/](http://www.numa.uni-linz.ac.at/Publications/List/)