

Towards an Implementation of Nitsche's Method on Overlapping Meshes in 3D

Andre Massing^{*,†}, Mats G. Larson^{**,*} and Anders Logg^{*,†}

^{*}Center for Biomedical Computing at Simula Research Laboratory, P.O. Box 134, 1325 Lysaker.

[†]Department of Informatics, University of Oslo, Norway.

^{**}Mathematics, Umeå University, SE-90187 Umeå, Sweden.

Abstract. Nitsche's method may be used to derive a systematic finite element formulation for problems with overlapping meshes that is stable and has optimal order. In this note, we formulate the method for a linear elastic model problem with discontinuous material properties. We discuss the implementation aspects, including computation of intersections of elements and integrals on the resulting polyhedra, and illustrate the method on a three dimensional test problem.

Keywords: Nitsche method, discontinuous Galerkin method, immersed interface

PACS: 02.60.Cb, 02.60.L, 02.70.Dh

INTRODUCTION

Overlapping mesh techniques are important in applications where an object is immersed in a domain, for instance, an elastic body immersed in a fluid flow. The technique is of particular interest in situations with moving objects. It then provides an alternative to ALE techniques and avoids remeshing. Another important application is the representation of material interfaces.

An overlapping mesh is constructed by letting a mesh of the object overlap a mesh of the underlying domain. A finite element space may then be constructed by taking the direct sum of the space of continuous piecewise polynomial functions on the object mesh and the restriction of the space of continuous piecewise polynomial functions to the complement of the object mesh. The resulting finite element space consists of functions that are discontinuous across the interface between the object and the domain. Based on these spaces, Nitsche's method may be used to construct a consistent method of optimal order, see Hansbo, Hansbo, and Larson [1]. In this contribution, we apply Nitsche's method to the problem of a linear elastic body consisting of two different materials with constant material properties. We formulate the method, discuss the implementation in three spatial dimensions, and present an illustrating numerical example.

In order to implement the overlapping mesh Nitsche method, we need to compute the intersection between the underlying and overlapping meshes, which results in elements that are complex polyhedra. In essence, we need to compute the sets that define the intersection between two arbitrary tetrahedra. Furthermore, we need to compute integrals on the resulting polyhedra. By adopting algorithms and search structures from the field of computational geometry, we show how these issues can be handled in an efficient manner.

NITSCHÉ'S METHOD ON OVERLAPPING MESHES FOR LINEAR ELASTICITY

The linear elasticity problem

We consider a linear elastic body occupying a domain $\Omega_0 = \Omega_1 \cup \Omega_2$ in \mathbf{R}^3 consisting of two subdomains Ω_1 and Ω_2 with different material properties. Denoting the interface between the subdomains by $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$, the

corresponding linear elasticity problem takes the form: find the displacements $\mathbf{u} : \Omega_0 \rightarrow \mathbf{R}^3$ such that

$$-\operatorname{div}(\boldsymbol{\sigma}(\mathbf{u}_i)) = \mathbf{f}_i, \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (1)$$

$$[\boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n}] = \mathbf{0}, \quad \text{on } \Gamma, \quad (2)$$

$$[\mathbf{u}] = \mathbf{0}, \quad \text{on } \Gamma, \quad (3)$$

$$\mathbf{u} = \mathbf{0}, \quad \text{on } \partial\Omega_{0,D}, \quad (4)$$

$$\boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{g}, \quad \text{on } \partial\Omega_{0,N}. \quad (5)$$

Here, the stress tensor $\boldsymbol{\sigma}$ is related to the displacements \mathbf{u} by Hooke's law

$$\boldsymbol{\sigma}(\mathbf{u}) = 2\mu_i \boldsymbol{\varepsilon}(\mathbf{u}) + \lambda_i \operatorname{tr}(\boldsymbol{\varepsilon}(\mathbf{u})) \mathbf{I}, \quad \text{in } \Omega_i, \quad i = 1, 2,$$

where λ_i and μ_i are the Lamé parameters in Ω_i for $i = 1, 2$ and $\boldsymbol{\varepsilon} = (\operatorname{grad}(\mathbf{u}) + \operatorname{grad}(\mathbf{u})^T)/2$ is the strain tensor. We also use the notation $v_i = v|_{\Omega_i}$ for the restriction of a function v to the subdomain Ω_i and $[v] = v_1 - v_2$, for the jump in a function over the interface Γ .

Nitsche's method on overlapping meshes

We consider an overlapping mesh consisting of a mesh $\mathcal{T}_0 = \{T\}$ on the whole domain Ω_0 overlapped by a mesh $\mathcal{T}_1 = \{T\}$ on the subdomain Ω_1 . Both meshes consist of shape-regular tetrahedra T . We let $\mathbf{V}_{h,i}$ denote the space of vector-valued continuous piecewise linear polynomials defined on \mathcal{T}_i for $i = 0, 1$ and define the space

$$\mathbf{W}_h = (\mathbf{V}_{h,0}|_{\Omega_2}) \oplus \mathbf{V}_{h,1},$$

where $\mathbf{V}_{h,0}|_{\Omega_2}$ denotes the restriction of functions in $\mathbf{V}_{h,0}$ to Ω_2 .

Nitsche' method proposed in [1] then takes the form: find $\mathbf{u}_h \in \mathbf{W}_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{W}_h, \quad (6)$$

where

$$a(u, v) = \sum_{i=1}^2 \int_{\Omega_i} \boldsymbol{\sigma}(\mathbf{u}_i) : \operatorname{grad}(\mathbf{v}_i) - \underbrace{\int_{\Gamma} \boldsymbol{\sigma}(\mathbf{u}_1) \cdot \mathbf{n}_1 [\mathbf{v}]}_{\text{Stress balance}} - \underbrace{\int_{\Gamma} \boldsymbol{\sigma}(\mathbf{v}_1) \cdot \mathbf{n}_1 [\mathbf{u}]}_{\text{Symmetrization}} + \underbrace{\gamma \int_{\Gamma} h^{-1} [\mathbf{u}] \cdot [\mathbf{v}]}_{\text{Penalty/Stabilization}},$$

$$l(v) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\partial\Omega} \mathbf{g} \cdot \mathbf{v},$$

with γ a positive penalty parameter. When the penalty parameter is chosen large enough, the form $a(\cdot, \cdot)$ is coercive on the discrete space \mathbf{W}_h and one can derive optimal order a priori error estimates for arbitrary order polynomials, see [1] for details.

IMPLEMENTATION OF NITSCHÉ'S OVERLAPPING MESH METHOD

General implementation considerations

A realization of Nitsche's method on overlapping meshes is challenging because of geometry related issues, similar to those arising in the implementation of the XFEM, see for example [2]. To assemble the stiffness matrix and the load vector, the local tensors

$$\int_{P_0^m} \boldsymbol{\sigma}(\phi_i) : \operatorname{grad}(\phi_j), \quad \int_{P_0^m} \mathbf{f} \cdot \phi_j \quad \text{and} \quad - \int_{\Gamma^{mm}} \boldsymbol{\sigma}(\phi_i) \cdot \mathbf{n}_1 [\phi_j] - \int_{\Gamma^{mn}} \boldsymbol{\sigma}(\phi_i) \cdot \mathbf{n}_1 [\phi_j] + \gamma \int_{\Gamma^{mn}} h^{-1} [\phi_i] \cdot [\phi_j]$$

have to be computed on cut cells (polyhedra) $P_0^m = T_0^m \setminus \Omega_1$ and cut facets $\Gamma^{mm} = \Gamma_1^m \cap T_0^n$ respectively. Additionally, the interface integrands involve products of test and trial functions on *different* meshes, making a partition of the

interface facet with respect to its intersecting cells necessary, as Figure 1 illustrates. Basically, the assembly of the integral contributions in the intersection zone can be split up into three main steps:

- i) *Collision detection*: As a first essential step, we have to determine the involved mesh entities. These can be described by an intersection map that maps from cells in \mathcal{T}_0 to the interface facets in \mathcal{T}_1 they are intersected by.
- ii) *Intersection*: Next, the volume of the polyhedra P_0^m and the partition $\{\Gamma^{mn}\}_n$ of the interface facets Γ^n have to be computed.
- iii) *Integration*: As a last step, the local contributions to the stiffness matrix and load vector have to be computed on the intersected entities P_0^m and Γ^{mn} .

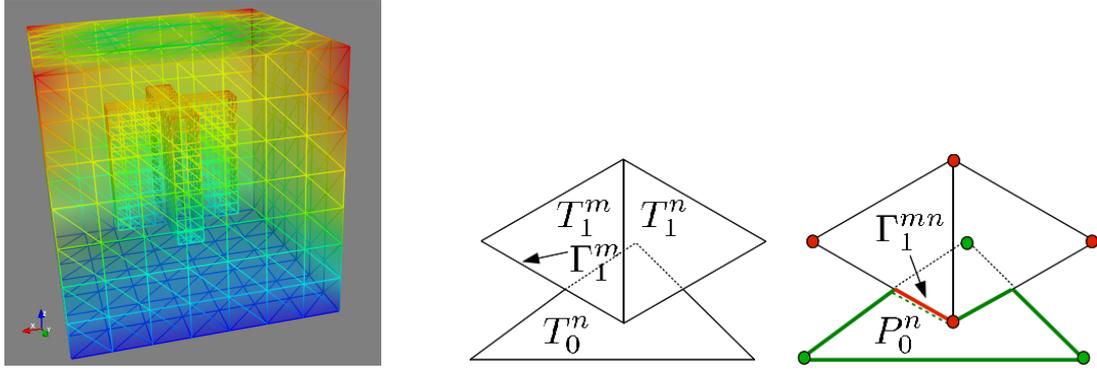


FIGURE 1. Left: Mesh domain Ω_0 containing an interior, overlapping domain Ω_1 . Middle and right: Intersected entities.

Collision detection and intersection computation. A naive approach to computing the intersection of two meshes \mathcal{T}_0 and \mathcal{T}_1 would be to intersect each cell of \mathcal{T}_0 with each cell \mathcal{T}_1 , resulting in an unacceptable $\mathcal{O}(|\mathcal{T}_0| \cdot |\mathcal{T}_1|)$ -complexity. In the field of computational geometry, efficient algorithms and data structures have been developed which reduce the complexity dramatically by fast intersection tests for pairs of simple geometric objects and a hierarchical description of the data set, see [3]. *Geometric predicates* allow robust computation of whether an intersection exists without computing the actual intersection. To handle collision tests for large data sets of geometric objects, so-called **Axis Aligned Bounding Box** trees are used. The basic idea behind AABB trees is to approximate each geometric entity by an AABB and to build up a hierarchy of boxes to avoid checking every single box. Whether two AABBs intersect can be quickly determined and gives a candidate for checking for an actual intersection. Note that to build up the intersection map, in essence only the overlap region between the *boundary* $\partial\Omega_1$ and Ω_0 has to be determined. With the help of the intersection map, the intersected cells P_0^m and partitions $\Gamma^m = \bigcup \Gamma^{mn}$ for the facets can be computed.

Integration on complex polyhedra. A common approach to computing local element matrices on non-standard or run-time dependent element geometries is to compute a sub-triangulation of the element. We here suggest another approach, based on work by Mirtich [4]. Mirtich's algorithm computes efficiently the mass properties in a $\mathcal{O}(\text{number of vertices})$ manner by reducing the moment integrals $\int_P x_i^\alpha$ to 1D integrals. The volume and barycenter coordinates of a polyhedron can be directly used for barycenter based quadrature, which integrates linear polynomials exactly and is therefore sufficient if only P_1 Lagrange elements are considered. If a higher order integration scheme is desirable, one can generalize the approach by computing the required higher order moments and interpolating the integrand.

Realization within the FEniCS framework

Our implementation of Nitsche's method on overlapping meshes is based on the finite element library DOLFIN [5] which is part of the FEniCS project for automated scientific computing. For this work, we have integrated the computational geometry libraries CGAL [6] and GTS [7] with DOLFIN. While CGAL and GTS provide a large part of the functionality needed to compute intersections between tetrahedra, the integration scheme was realized through an adapted version of Mirtich's code. Furthermore, the assembly routine was extended to handle integration over cut cells and meshes. Currently, the code for computing local interface integrals has to be implemented manually by the

user, but we plan to extend FEniCS, in particular DOLFIN and the form compiler FFC, to provide a full automation of Nitsche's method where a user only needs to supply the abstract variational problem (6).

EXAMPLE

We consider the linear elasticity problem (1)–(5) with $\Omega_0 = [-2, 2]^3 \subset \mathbf{R}^3$ which is overlapped by a propeller-like domain $\Omega_1 = [-1, 1] \times P$ where $P = [-1, 1] \times [-0.2, 0.2] \cup [-0.2, 0.2] \times [-1, 1] \subset \mathbf{R}^2$, see Figure 1. The right-hand side is zero, $\mathbf{f} = \mathbf{0}$, and the boundary conditions on $\partial\Omega_0$ are defined by

$$\begin{cases} \mathbf{u} = 0, & \text{on } [-2, 2]^2 \times \{-2\}, \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = 0, & \text{on } \partial[-2, 2]^2 \times [-2, 2], \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{g} & \text{on } [-2, 2]^2 \times \{2\}, \end{cases}$$

where \mathbf{g} represents a combination of pure tangential, rotational force and a normal pressure:

$$\mathbf{g}(x, y, z) = \frac{(-y, x, 0)^\top}{5(\sqrt{x^2 + y^2} + 1)} + (0, 0, -2 + \frac{\sqrt{x^2 + y^2}}{5 \cdot 10^{-3} + \sqrt{x^2 + y^2}})^\top.$$

The Lamé parameters are given by $\mu_i = E_i / (2 + 2\nu_i)$, $\lambda_i = E_i \cdot \nu_i / ((1 + \nu_i)(1 - 2\nu_i))$ in Ω_i for $i = 1, 2$, with $E_1 = 1, E_2 = 0.1, \nu_1 = \nu_2 = 0.3$.

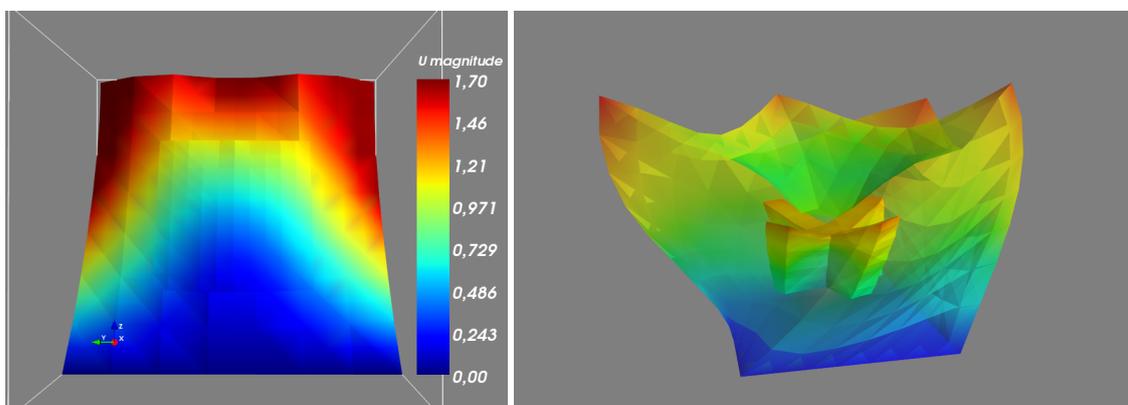


FIGURE 2. Left: Magnitude of displacement \mathbf{u} in yz plane. Right: Deformed domain(s).

ACKNOWLEDGMENTS

We thank Harish Narayanan for valuable discussions regarding the elasticity test case. This work is supported by an Outstanding Young Investigator grant from the Research Council of Norway, NFR 180450. This work is also supported by a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory.

REFERENCES

1. A. Hansbo, P. Hansbo, and M. Larson, *M2AN Math. Model. Numer. Anal* **37**, 495–514 (2003).
2. N. Sukumar, N. Moës, B. Moran, and T. Belytschko, *International Journal for Numerical Methods in Engineering* **48**, 1549–1570 (2000).
3. C. Ericson, *Real-time collision detection*, Morgan Kaufmann, 2005.
4. B. Mirtich, *Journal of graphics, GPU, and game tools* **1**, 31–50 (1996).
5. A. Logg, and G. N. Wells, *ACM Transactions on Mathematical Software* **32**, 1–28 (2010).
6. <http://www.cgal.org/> (2010).
7. <http://gts.sourceforge.net/> (2010).