Graz University of Technology

Daniel SCHÖLLHAMMER BSc

Laplace-Beltrami operator on implicitly defined manifolds

MASTER'S THESIS

to achieve the university degree of

Diplom-IngenieurIn

Master's degree programme: Civil Engineering and Structural Engineering

submitted to

Graz University of Technology

Supervisor

Univ. Prof. Dr.Ing. habil. Thomas Peter FRIES

Institute of Structural Analysis

Graz, August 2016

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Acknowledgements

I would like to express my gratitude to my supervisor Thomas Peter Fries for his useful comments, remarks and engagement at any time through the learning process of this master's thesis. I do believe, that this assistance steered me in the right direction and I am thankful for the confidence he has placed in me.

Furthermore I also would like to thank Samir Omerović, Dominik Pölz and Markus Schätzer for their support and helpful discussions about my research and related topics, which increased my interests in this field of engineering. Moreover I need to thank Jakob Steidl and Benjamin Marussig for their help whenever any IT issues have occurred.

Finally, I must express my very profound gratitude to my parents and to my girlfriend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

In this thesis partial differential equations (PDEs) on manifolds (curves and surfaces) are considered. Such PDEs appear in a variety of problems and applications in fluid dynamics, material science, solid mechanics, biology and image processing.

The curves and surfaces are embedded in the Euclidean space and implicitly defined by level-set functions. The meshes for the numerical approach are generated automatically using a higher-order remeshing schemes, which is called reconstruction [5]. The resulting meshes are not necessarily well-shaped, but it is confirmed that optimal approximation properties are achieved.

In contrast to the conventional approach our formulation of the PDEs is based on the concept of the differential tangential calculus, which avoids concepts of the differential geometry such as Christoffel symbols and the discrete system can be established in a Euclidean coordinate system. For the numerical approach the finite element method (FEM) is used. The formulation of the weak form is based on the classical Bubnov-Galerkin method.

As applications the Laplace-Beltrami operator and the in-stationary transport problem are considered herein. Convergence analyses are performed with conventional and reconstructed meshes. It is seen that for both mesh types optimal convergence rates are achieved with respect to the theoretical a priori error estimates.

Zusammenfassung

In dieser Masterarbeit werden partielle Differentialgleichungen (PDEs) auf Mannigfaltigkeiten (Kurven und Oberflächen) betrachtet. Diese Gruppe von PDEs sind von großem Interesse, da diese in vielen Anwendungen im Bereich der Strömungsdynamik, Materialwissenschaften, Strukurmechanik, Biologie und in der Bilderverarbeitung auftreten.

Die Kurven und Oberflächen sind im euklidischen Raum eingebettet und werden über implizite Level-set Funktionen definiert. Die Netze für das numerische Lösungsverfahren werden automatisch, unter Verwendung von konformen Vernetzungs-Algorithmen, generiert. Die Netzgenerierung wird als Rekonstruktion benannt und derartige Netze können, trotz möglichem, nicht optimalen Elementgrößenverhältnissen, optimale Approximationseigenschaften aufweisen [5].

Im Gegensatz zur klassischen Vorgehensweise basiert die Formulierung der PDEs auf dem sogenannten „differential tangential calculus“, wobei auf einige typische Konzepte der Differentialgeometrie verzichtet werden kann, wie etwa die explizite Verwendung von Christoffel-Symbolen. Das diskrete System kann dann vereinfacht in einem euklidischen Koordinatensystem assembliert werden. Als numerisches Verfahren wird die Finite Elemente Methode (FEM) verwendet und die schwache Form wird mit Hilfe des klassischen Bubnov-Galerkin Verfahren formuliert.

Die Vorgehensweise wird exemplarisch am Laplace-Beltrami Operator und am instationären Transportproblem gezeigt. Für die Testfälle wird eine Konvergenzanalyse mit konventionellen und den automatisch rekonstruierten Netzen durchgeführt. Die numerischen Ergebnisse stimmen für beide Netztypen mit den a priori Fehlerabschätzungen überein.

Contents

1 Introduction **5**

2 Preliminaries **7**
2.1 Implicitly defined manifolds . . . 7
2.2 Discretization . . . 8
2.2.1 1D-Reference element . . . 8
2.2.2 2D-Reference element . . . 9
2.2.3 Mapping & Derivatives . . . 9

3 Mesh generation **11**
3.1 Reconstruction . . . 11
3.1.1 Shape and manipulation of the background mesh . . . 13
3.2 Manifold mesh extraction . . . 15
3.2.1 Line mesh in a 2D background mesh . . . 16
3.2.2 Surface mesh in a 3D background mesh . . . 18
3.3 Computational Costs . . . 21

4 FEM-Simulations on manifolds **23**
4.1 General definitions . . . 23
4.2 Weak form of the Laplace-Beltrami operator . . . 23
4.2.1 Discrete weak form . . . 25
4.2.2 Error estimation . . . 26
4.3 Weak form of the in-stationary transport problem . . . 26
4.3.1 Error estimation . . . 27

5 Numerical results **29**
5.1 Laplace-Beltrami operator . . . 29
5.1.1 Manufactured solution . . . 29
5.1.2 Example 1 - Circle . . . 30
5.1.3 Example 2 - Flower . . . 32
5.1.4 Example 3 - S-shaped manifold . . . 35
5.1.5 Example 4 - Quarter Cylinder . . . 38
5.1.6 Example 5 - Sphere . . . 40
5.1.7 Example 6a - Hyperbolic paraboloid with bumps . . . 43
5.1.8 Example 6b - Hyperbolic paraboloid with bumps with fixed nodes . . . 45
5.2 In-stationary transport problem . . . 47
5.2.1 Verification in 1d . . . 47
5.2.2 Verification in 2d . . . 49
5.2.3 Example 1 - S-shaped manifold . . . 51
5.2.4 Example 2 - Hyperbolic paraboloid with bumps . . . 51

6 Conclusion & Future prospects **53**

Notation

a	Scalar
$\boldsymbol{g}$	Vector, matrix or tensor
$\boldsymbol{v}_{i,j}$	Derivative of $\boldsymbol{v}_i$ in respect to j
$\boldsymbol{a} \cdot \boldsymbol{b}$	Dot product
$\nabla \cdot \boldsymbol{v}$	Divergence of a vector field
∇u	Gradient of a scalar field
$\nabla_\Sigma u$	Surface gradient of a scalar field
Δu	Laplace operator of a scalar field $:= \nabla \cdot \nabla u$
$f \circ g$	Composition of the functions f and g
$\hat{\Omega}$	Reference space
Ω	Real domain
Σ	Manifold
Γ_D	Dirichlet boundary
Γ_N	Neumann boundary
$\|\boldsymbol{v}\|$	Euclidean norm of $\boldsymbol{v}$
$\|u\|_{0,2,\Sigma}$	L^2 norm of u in Σ

1 Introduction

Partial differential equations (PDEs) on curves and surfaces appear in a variety of problems and applications in fluid dynamics, material science, solid mechanics, biology and image processing. In general, when the geometry of the domain of interest can be considered as "thin" in one or two directions, one may often simplify the model using a formulation that involves PDEs on a lower dimensional geometry, i.e. on a curve or a surface [9]. Common examples are shells, curved beams or transport problems on interfaces or thin structures.

The curves and surfaces are manifolds embedded in the physical space and in this thesis the hypersurfaces are defined implicitly by the level set method. That is, a scalar function, called "level-set function", is defined everywhere in the physical space and its zero-level set defines the manifold where a PDE is to be solved. Based on this implicit representation higher-order remeshing schemes such as those presented in [3], [4] can be used in order to generate manifold meshes automatically. These methods identify the position of the zero-level set and approximate it by manifold (line and surface) elements. This process is called reconstruction. The resulting meshes typically feature awkward element shapes and apsect ratios. However, it is shown herein that this still allows the approximation of PDEs with optimal accuracy.

In order to solve a PDE on a manifold, the conventional approach is to heavily employ concepts from differential geometry such as Christoffel symbols [2]. However, the approach discussed herein is based on the differential tangential calculus, which simplifies the notation significantly [9], [6]. The finite element method is used for the approximation of the Laplace-Beltrami operator and transport equations on the manifolds. A classical Bubnov-Galerkin approach for the formulation of the weak form is chosen herein [7]. The mathematical analysis of this approach for higher-order elements is described in [11].

2 Preliminaries

In this section implicitly defined curves and surfaces (manifolds) and their discretization is considered. The representation of a manifold by the level-set method is described. Furthermore polynomial, reference elements for the numerical approach are defined.

2.1 Implicitly defined manifolds

The implicit representation of a n-dimensional manifold Σ takes the form

$$\Sigma = \left\{ \boldsymbol{x} \in \mathbb{R}^{n+1} : \phi(\boldsymbol{x} = 0) \right\} \tag{2.1}$$

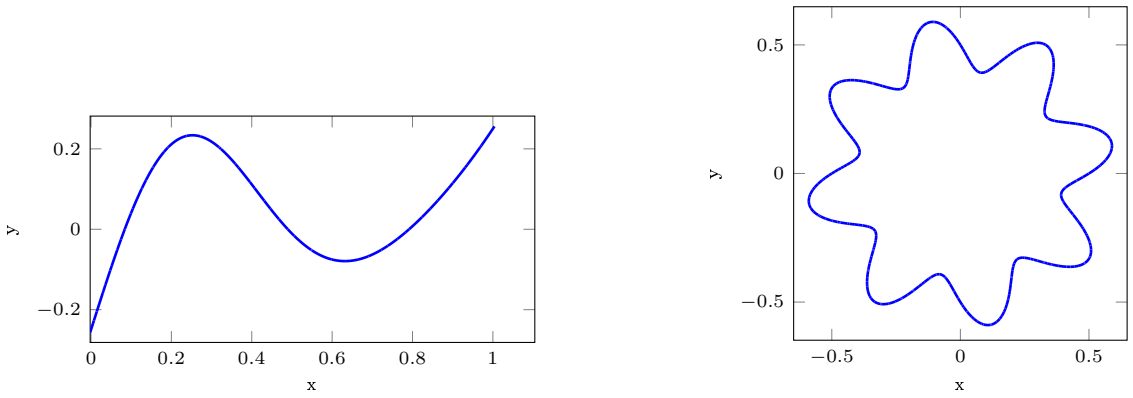
so that $\boldsymbol{x} \in \Sigma$ is a point on the manifold implicitly defined by the function $\phi : \mathbb{R}^{n+1} \to \mathbb{R}$. There are three different cases for the level-set function ϕ:

$$\begin{aligned} \phi(\boldsymbol{x}) > 0, \quad & \boldsymbol{x} \text{ is on the one side of the manifold} \\ \phi(\boldsymbol{x}) < 0, \quad & \boldsymbol{x} \text{ is on the other side of the manifold} \\ \phi(\boldsymbol{x}) = 0, \quad & \boldsymbol{x} \text{ is exactly on the manifold} \end{aligned}$$

For example, a unit sphere can be written as

$$\phi(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 - 1 \ , \qquad \text{where} \quad \|\boldsymbol{x}\|^2 = x^2 + y^2 + z^2 \ .$$

Further information about the implicit definition of a manifold is given in [9]. In this thesis, the manifolds which are considered herein, are lines in $\mathbb{R}^2$ and surfaces in $\mathbb{R}^3$. These geometrical objects may have a boundary or not. If there is a boundary the manifolds are considered as "open" and if not the manifolds are called "closed". In Fig. 2.1 an open and closed manifold embedded in $\mathbb{R}^2$ is illustrated.



(a) Open **(b)** Closed

Fig. 2.1: Manifolds embedded in $\mathbb{R}^2$

2.2 Discretization

The finite element method (FEM) is used to determine a numerical solution of the given PDE. Therefore the domain, in our case the manifold Σ, has to be discretized by a mesh, i.e. a set of elements. The procedure of the mesh generation is described in Chapter 3, but at first the reference elements $\hat{T}$ and the mapping from the reference space $\hat{\Omega}$ to the real domain Ω is defined.

2.2.1 1D-Reference element

The 1D-reference element $\hat{T} \in [-1, 1]$ is based on polynomials and is defined in a 1D reference space $\hat{\Omega}$. The element is divided into m equidistant intervals with $m+1$ nodes. For each node a polynomial (base function), which satisfies $N_\beta(\xi_\alpha) = \delta_{\alpha\beta}$ can be obtained as

$$N_\beta(\xi) = \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{m+1} \frac{\xi - \xi_\alpha}{\xi_\beta - \xi_\alpha}, \qquad \beta = 1, 2, \ldots, m+1 \ . \tag{2.2}$$

This class of polynomials is called Lagrange polynomials [7, Chapter 3] and, hence, the corresponding element is called Lagrange element. A more general definition, which can easily be extended to higher dimensions is

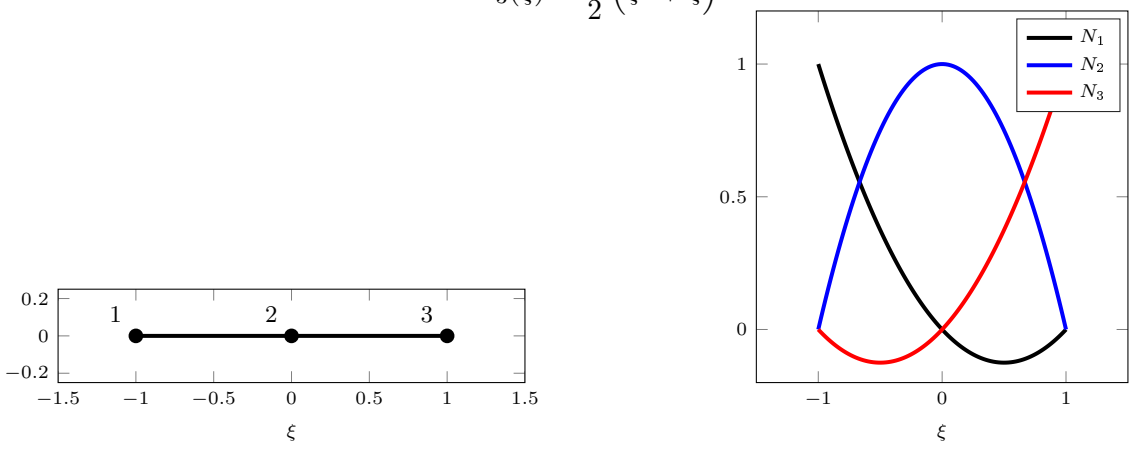
$$\begin{aligned} N_\beta(\xi) &= a_\alpha \xi^{\alpha-1} \qquad \alpha = 1, 2, \ldots, m+1 \\ N_\beta(\xi_\alpha) &= \delta_{\alpha\beta} \ . \end{aligned} \tag{2.3}$$

The coefficients a_i can be determined by solving the system of linear equations:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{a} &= \boldsymbol{\delta} \\ A_{i,j} &= \xi_i^j \ , \quad i = \alpha \ , \ j = \alpha - 1 \\ \boldsymbol{a} &= \{a_i\} \\ \boldsymbol{\delta} &= \{\delta_{i\beta}\} \end{aligned} \tag{2.4}$$

The subscript i indicates the node number and the superscript j the exponent of ξ. Fig. 2.2 shows the reference element and the base functions of a 2nd-order line element.

$$\begin{aligned} N_1(\xi) &= \frac{1}{2}\left(\xi^2 - \xi\right) \\ N_2(\xi) &= 1 - \xi^2 \\ N_3(\xi) &= \frac{1}{2}\left(\xi^2 + \xi\right) \end{aligned}$$



(a) Reference element **(b)** Base functions

Fig. 2.2: 2nd-order 1D-Lagrange element

2.2.2 2D-Reference element

For the description of surfaces triangular and quadrilateral elements are used. The definition of the reference elements $\hat{T}$ is shown for the example of a 2^{nd}-order ($p = 2$) quadrilateral Lagrange element. The base functions can be defined with a polynomial base

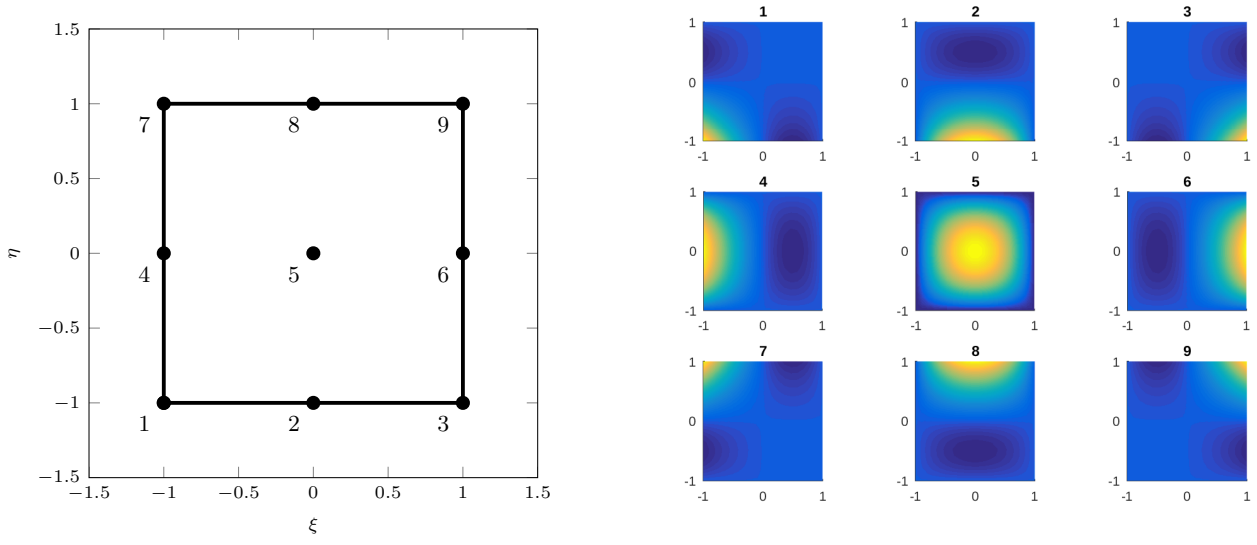
$$\varphi_\beta(\xi, \eta) = \sum_{i=1}^{n=(p+1)^2} a_i \, \xi^j \eta^k \,, \qquad j, k = 0,\, 1,\, 2 \tag{2.5}$$
$$N_\beta(\xi_\alpha, \eta_\alpha) = \delta_{\alpha\beta}$$

and by solving the system of linear equations similar to Eq. 2.4 the coefficients can be determined:

$$\begin{aligned} N_1(\boldsymbol{\xi}) &= n_1 m_1 \quad N_4(\boldsymbol{\xi}) = n_1 m_2 \quad N_7(\boldsymbol{\xi}) = n_1 m_3 \\ N_2(\boldsymbol{\xi}) &= n_2 m_1 \quad N_5(\boldsymbol{\xi}) = n_2 m_2 \quad N_8(\boldsymbol{\xi}) = n_2 m_3 \\ N_3(\boldsymbol{\xi}) &= n_3 m_1 \quad N_6(\boldsymbol{\xi}) = n_3 m_2 \quad N_9(\boldsymbol{\xi}) = n_3 m_3 \end{aligned}$$

$$\begin{aligned} n_1(\xi) &= \frac{1}{2}\xi\,(1-\xi) \quad & m_1(\eta) &= -\frac{1}{2}\eta\,(1-\eta) \\ n_2(\xi) &= 1-\xi^2 & m_2(\eta) &= 1-\eta^2 \\ n_3(\xi) &= \frac{1}{2}\xi\,(1+\xi) & m_3(\eta) &= \frac{1}{2}\eta\,(1+\eta) \end{aligned}$$

In Fig. 2.3 the reference element and the corresponding base functions are visualized.



(a) Reference element **(b)** Base functions

Fig. 2.3: 2^{nd}-order 2D-Lagrange element

2.2.3 Mapping & Derivatives

As usual in the finite element method, the isoparametric concept is used to map the reference element to the physical element in the mesh. Here, the mesh is composed by manifold elements so that this mapping is $\mathbb{R}^n \to \mathbb{R}^{n+1}$. It is defined as $\boldsymbol{\chi}(\boldsymbol{\xi}) := N_i(\boldsymbol{\xi})\boldsymbol{x}_i$ (summing is implied by repetitive indices herein). The points $\boldsymbol{x}_i \in \mathbb{R}^{n+1}$ denote the coordinates of the element nodes in the real domain. In

Fig. 2.4 the mapping of a 2nd order Lagrange element from the reference space $\hat{\Omega}$ to the real domain Ω is illustrated.

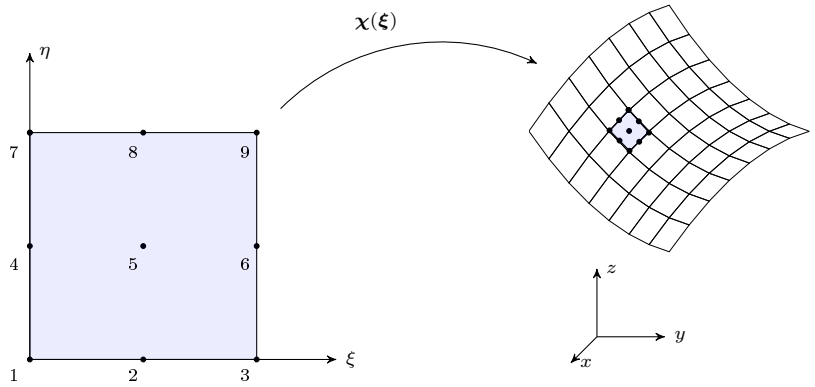


Fig. 2.4: Illustration of a mapping from $\hat{\Omega} \rightarrow \Omega$

Thereby a point of the manifold, in global coordinates, can be expressed in terms of coordinates in the reference space:

$$\boldsymbol{x}(\boldsymbol{\xi}) = N_i(\boldsymbol{\xi})\, \boldsymbol{x}_i \tag{2.6}$$

The Jacobi matrix $\boldsymbol{J}$ is defined by the partial derivatives of the mapping function $\boldsymbol{\chi}(\boldsymbol{\xi})$ with respect to the reference coordinates.

$$\boldsymbol{J} = \frac{\partial x_k}{\partial \xi_j} = \begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \\ z_{,\xi} & z_{,\eta} \end{bmatrix} \tag{2.7}$$

The global derivatives of $\boldsymbol{N}$ can be determined by rearranging

$$\begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{bmatrix} = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \end{bmatrix} \cdot \begin{bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{bmatrix} = \boldsymbol{J}^T \cdot N_{i,\boldsymbol{x}} \tag{2.8}$$

to

$$\begin{aligned} N_{i,\boldsymbol{x}} &= \boldsymbol{J} \cdot \left(\boldsymbol{J}^T \cdot \boldsymbol{J} \right)^{-1} \cdot N_{i,\boldsymbol{\xi}} \\ \nabla_\Sigma N(\boldsymbol{x}) &= \left[\boldsymbol{J} \cdot \boldsymbol{g}^{-1} \cdot \nabla_{\hat{\Omega}} N(\boldsymbol{\xi}) \right] \circ \boldsymbol{\chi}^{-1}(\boldsymbol{\xi}). \end{aligned} \tag{2.9}$$

Here, $\boldsymbol{g}$ is the Gram matrix of the Jacobi matrix. In terms of classical differential geometry $\boldsymbol{g}$ is the metric tensor and the inverse of $\boldsymbol{g}$ is the contra-variant form. The operators $\nabla_{\hat{\Omega}} N(\boldsymbol{\xi})$ denotes the gradient in the reference space and ∇_Σ is the surface gradient, which is defined as the projection of the physical space gradient of an extension of the function onto the tangent space of the surface. The mathematical derivation of the surface gradient is given in [9, Chapter 2.4] and is based on the differential tangential calculus.

3 Mesh generation

In general, there are several possibilities to define a manifold and generate a mesh. For instance one may start with a planar mesh and apply any map to $\mathbb{R}^3$ to obtain a surface mesh being a manifold.

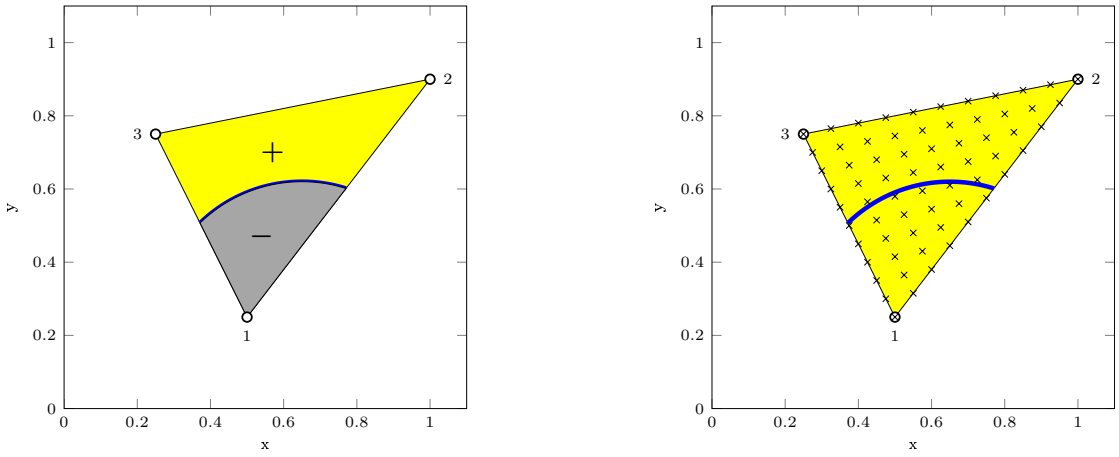
Our approach is to generate a manifold mesh using a higher-order re-meshing scheme. A direct consequence of this approach is, that an arbitrary manifold mesh can be generated automatically. Therefore a background mesh is introduced and the zero-level set is meshed by manifold (line or surface) elements. This procedure is called reconstruction and is described in detail in [3], [4].

In this section an overview of the reconstruction procedure, the requirements of the background mesh and the manifolds mesh extraction is given. Finally, in Section 3.3 the computational costs of the whole manifold mesh generation are considered.

3.1 Reconstruction

As mentioned above the aim of the reconstruction is to approximate the zero-level set with line or surface elements. The procedure of the reconstruction is shown for the example of a curve embedded in the 2D physical space.

At first a background mesh composed by higher-order Lagrange elements is introduced and the cut elements have to be determined. Therefore a simple grid, with a user defined resolution, is applied to each element and at each grid point the level set function is evaluated. If a sign change within the element is detected, the element is considered as cut. In Fig. 3.1 the zero-level set cuts two edges of the background element. In the sub-figures the sign of the level set function (a) and a sample grid (b) is illustrated.

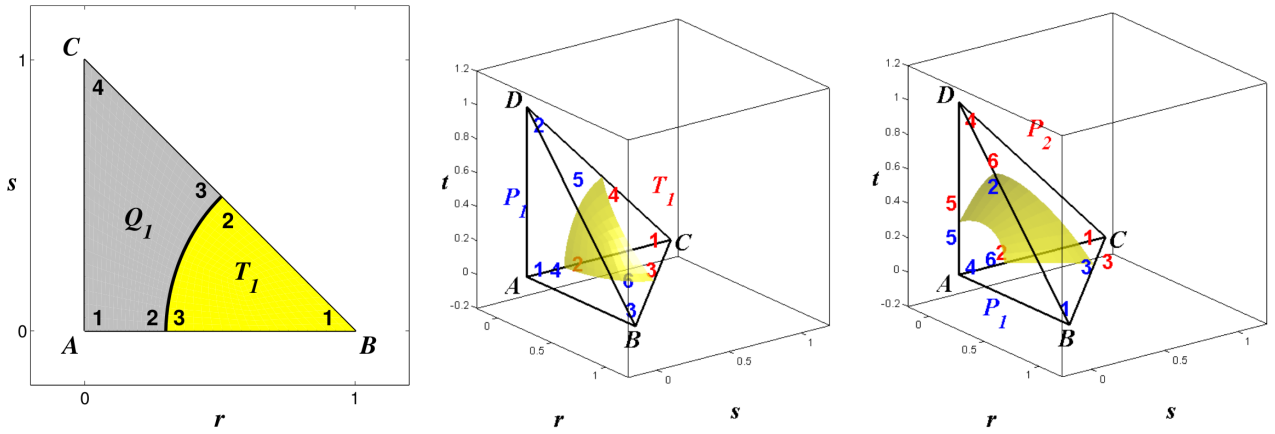


(a) Sign of the level set function **(b)** Sample grid

Fig. 3.1: Detection of a cut, triangular element

Note that in general just the nodal values of the element are not sufficient for a successful detection, see [3, Section 3.1] for further information.

If the level set function is more complex, many different cut scenarios may occur. For instance one edge is cut twice or all three edges are cut. Therefore a limitation to the cut scenarios might be necessary for an efficient implementation. In Fig. 3.2 “valid” cut scenarios for triangular and tetrahedral elements are illustrated.

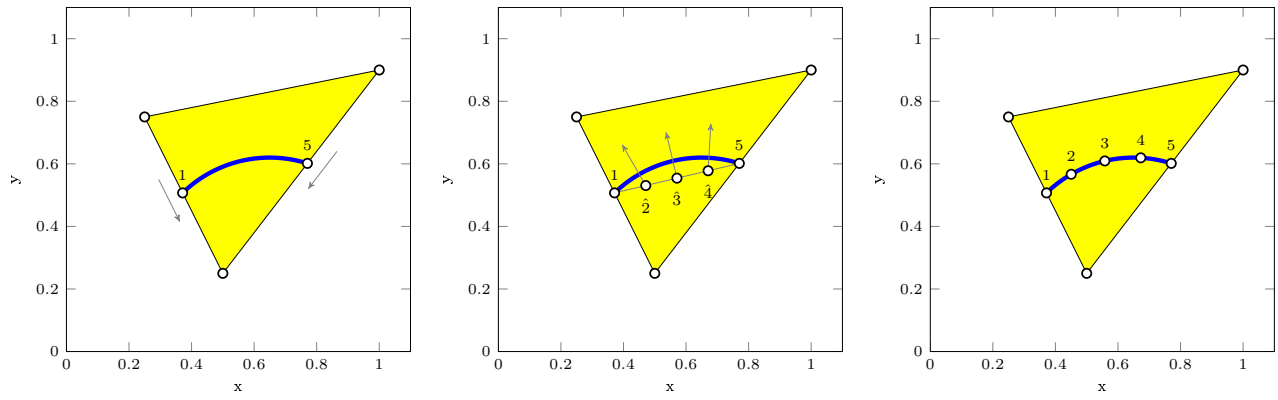


(a) Triangular element **(b)** Tetrahedral element, case 1 **(c)** Tetrahedral element, case 2

Fig. 3.2: Topological cut scenarios of a background element. (a) for the triangular element a valid cut scenario is if *just* two edges are cut. (b) and (c) for the tetrahedral element, depending on the signs of the level set function on the corners, the shape of the zero-level set may be triangular or quadrilateral [4, Section 3].

With this restriction invalid cut scenarios can occur and further steps for a successful reconstruction are necessary. These steps are discussed in Section 3.1.1.

After a successful detection of a cut element the zero-level set will be approximated by a manifold element. The nodes of the element are defined by a Newton-Raphson procedure with prescribed search path. As mentioned and investigated in [4] the node distribution is crucial for the approximation properties of the resulting manifold element. In this thesis, for line elements, the gradient of the level-set function is used for the search path. In Fig. 3.3 the procedure is visualized for a 4-th order line element. The gray arrows indicate the search path for the Newton-Raphson algorithm and the numbered nodes are the reconstructed line element nodes. The reconstruction starts with



(a) Corner nodes **(b)** Inner nodes **(c)** Line element

Fig. 3.3: Reconstruction procedure. (a) Reconstruction of the corner nodes along the edges of the background element. (b) Determination of the inner nodes with the gradient as search path. (c) Reconstructed line element.

the determination of the corner nodes, see Fig. 3.3(a). In Fig. 3.3(b) the start values ($\hat{2}$, $\hat{3}$, $\hat{4}$) are distributed equidistantly between the two corner nodes. The search paths (gray arrows) are the

gradients of the level set function at the start points. Finally, in Fig. 3.3(c) the whole line element is reconstructed.

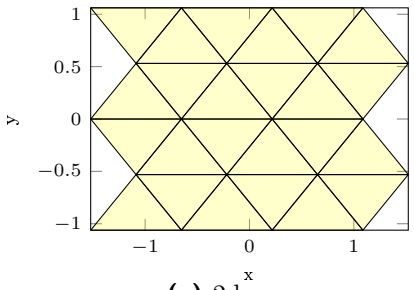
For surface elements the reconstruction is analogous. At first the edges are reconstructed and with a certain mapping start values at the inner nodes are determined. Search directions are defined and the inner element nodes of the surface element are found iteratively. Note that if the mapping is not smooth enough, suboptimal convergence rates may occur. Further information is given in [3], [4].

In contrast to the procedure in [3], [4], where the level-set data is transfered to the reference space, herein the whole procedure takes place on the level of the physical element. This leads to a restriction to the background mesh, because the corner nodes of a manifold element are determined by a Newton-Raphson iteration along the edges of the cut element, see Fig. 3.3(a). Therefore straight edges of the background mesh are a crucial requirement. Anyway, a linear background mesh is sufficient to define a manifold mesh composed by elements of higher-order.

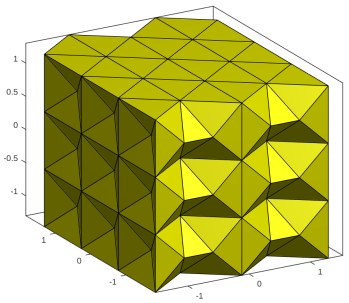
3.1.1 Shape and manipulation of the background mesh

In general there are many possibilities to define a background mesh, but as mentioned above the properties of the mesh influence the resulting reconstruction. In this thesis just simplex elements in 2d (triangular) and 3d (tetrahedral) shall be used for the background mesh although a higher-order manifold mesh is reconstructed.

Herein, the background meshes are based on universal meshes [10], [8] as illustrated in Fig. 3.4. These kind of meshes have, in our context, two important properties: (1) They are very regular so that nodes may be moved quite flexible. (2) They feature non-smooth boundaries, which is an issue when open manifolds are considered. For a closed manifold these class of background meshes is used in all cases. However, for the case of open manifolds, the background mesh does have the additional task



(a) 2d



(b) 3d

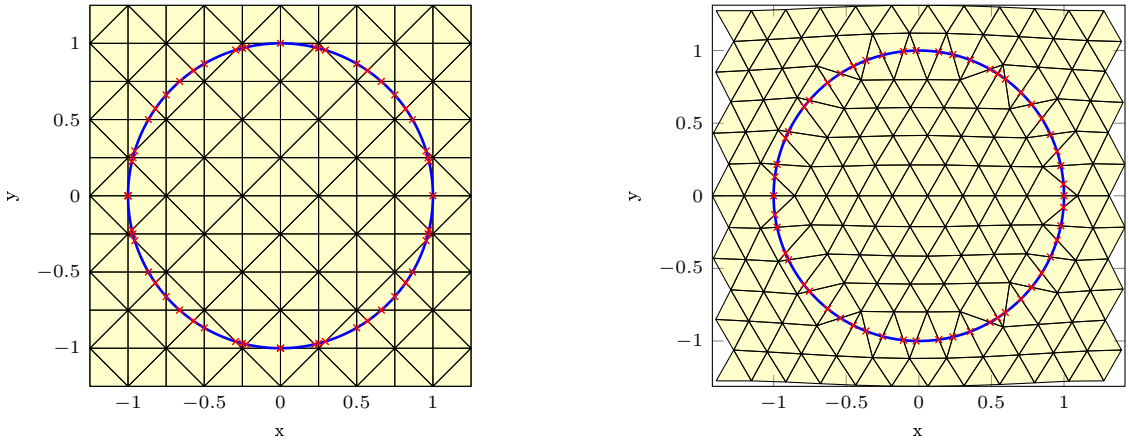
Fig. 3.4: Universal mesh

to imply the boundary of the manifold. It is then quite unlikely that a universal mesh generates the desired boundary. Then, it may be more useful to use background meshes with straight boundaries, e.g. union-jack like meshes as shown in Fig. 3.5.

(a) 2d **(b)** 3d

Fig. 3.5: Union-jack mesh

For a successful reconstruction of manifold elements in *every* cut background element, it is necessary to avoid special cases where the topology of the cut situation is not valid as discussed above. We found that a movement of some of the nodes in the background mesh is highly useful to "guarantee" valid cut situations whenver the resolution of the background mesh is reasonable with respect to the complexity of the zero-level sets. This approach moves the nodes of the background elements, which are close to the zero-level set, away to a moderate distance and tries to avoid invalid cut scenarios as much as possible. Furthermore this algorithm also ensures a more balanced element size ratio. In Fig. 3.6 a circle with a conventional triangular(left) and a universal mesh(right) is reconstructed. Moreover in the left figure the mesh manipulation is disabled and in the right figure enabled. The number of background elements in Fig. 3.6(a) is 56 and in Fig. 3.6(b) 54. The blue line is the manifold and the red crosses indicate the element corners. In Fig. 3.6(a) the ratio between $h_{\max}/h_{\min} \approx 2.38 \times 10^5$ and in Fig. 3.6(b) $h_{\max}/h_{\min} \approx 4.07$, which is way better than the ratio of the conventional mesh without node manipulations. With the implemented mesh manipulation it is



(a) Variant 1 **(b)** Variant 2

Fig. 3.6: Reconstructed circle, $r = 1$, (a) Conventional mesh with disabled mesh manipulation, (b) Universal mesh with enabled mesh manipulation

possible that all nodes of the background mesh are moved to another position, but if we consider an open manifold and the mesh manipulation is enabled, the shape of the boundary is changed. In order to avoid this situation the boundary of the background mesh can be forced back to the original location. Unfortunately, this may again lead to bad element size ratios for both types of meshes (conventional and universal) again. In Section 5.1.7 and Section 5.1.8 the impact of this constraint is demonstrated.

Another approach to define the boundary of an open manifold is to define additional level-set functions [5]. Note that these approach is not implemented yet, herein. In this thesis open manifolds

are reconstructed with universal meshes if the boundary is through the smooth edges. If not, a conventional block-mesh with union-jack element pattern is used. The mesh manipulation is enabled in all cases and the phenomena of the slightly changed boundary will be ignored and in the numerical examples the Dirichlet-boundary condition will be evaluated at the reconstructed boundary.

Another possibility to increase the robustness of the algorithm is a recursive refinement of the invalid background element until just valid cut scenarios occur [4, Section 3.5]. Despite of the fact that recursive refinement is a powerful tool, this approach is not an option herein because this would naturally lead to hanging nodes which we wish to avoid. For details about this constraint, see Section 3.2.

3.2 Manifold mesh extraction

The result of the reconstruction is a set of elements, each defined by the coordinates of the element nodes. There is not yet a concept of nodes which can be associated to degrees of freedoms in a finite element context. That is, nodes with identical coordinates have to be identified. A node number is associated to each unique node position and a connectivity matrix has to be set up. Thereafter, a finite element simulation may be carried out in the standard way. In this section the mesh generation is explained in detail. The whole mesh extraction can be divided into two major steps and in general this procedure is similar for lines and surfaces:

1. Reconstruction and storing of all manifold elements
2. Identification and removal of duplicate nodes:
   a) Determining the topological relationships of the cut background elements
   b) Updating the connectivity matrix

In the first step the coordinates of the generated manifold elements are stored and a “brute force” connectivity matrix is assembled. Each new element is just added by new node numbers. Therefore on shared edges or corners duplicate nodes occur. This can not be avoided, because the topological relation of the cut background elements and the location and number of the manifold elements is not known before the reconstruction procedure starts. Therefore the topological relationships are neglected by assembling the “brute force” connectivity matrix.

In step two the duplicate nodes are removed. In general there are many algorithms to identify and remove identical nodes and most of them are based on comparing the coordinates with tolerances. For instance in Fig. 3.6(a) the element size might be indefinite small. If such a small element occurs, it can happen that nodes of the same element or the whole element collapse to one node. Fig. 3.7 shows a path with quadratic line elements. The inner node 5 of element 2 is also in the range of the tolerance, due to the small element size. Therefore the nodes 3, 4, 5 and 6, 7 would be merged to node 3 and 6, but we just want to merge the nodes 3, 4 and 6, 7. Note that the points 3, 4 and 6, 7 are identical, due to a better visualization these nodes are moved apart slightly. Therefore these kind of algorithms do not satisfy the requirements especially when the elements may be unboundedly small.

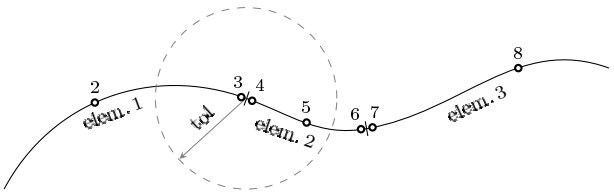


Fig. 3.7: Node collapse

Our approach is to merge the nodes 3, 4 and 6, 7 using the topological relationships. After the reconstruction it is known, which background element is cut and based on this information the neighbor relationships of the cut background elements can be determined. It is evident that, if the neighbor situation of the cut background elements is available, the topological relationships of the manifold mesh is determined too. With the cut scenarios and the neighbor information, identical nodes may be merged without comparing coordinates. In Section 3.2.1 and Section 3.2.2 these algorithms for line and surfaces meshes are explained in detail.

3.2.1 Line mesh in a 2D background mesh

As mentioned above, at first the topological relationships have to be determined and with this information the duplicate nodes can be identified and removed efficiently. The procedure is shown on a s-shaped manifold. The manifold is defined by a parameter $x \in [0,\, 1]$

$$\Sigma = \begin{bmatrix} x \\ f(x) \end{bmatrix} = \begin{bmatrix} x \\ \frac{x^3}{2} + \sin(\pi(1-x)) \sin^5\left(\frac{\pi}{2}(1-x)\right) - \frac{1}{4} \end{bmatrix} \tag{3.1}$$

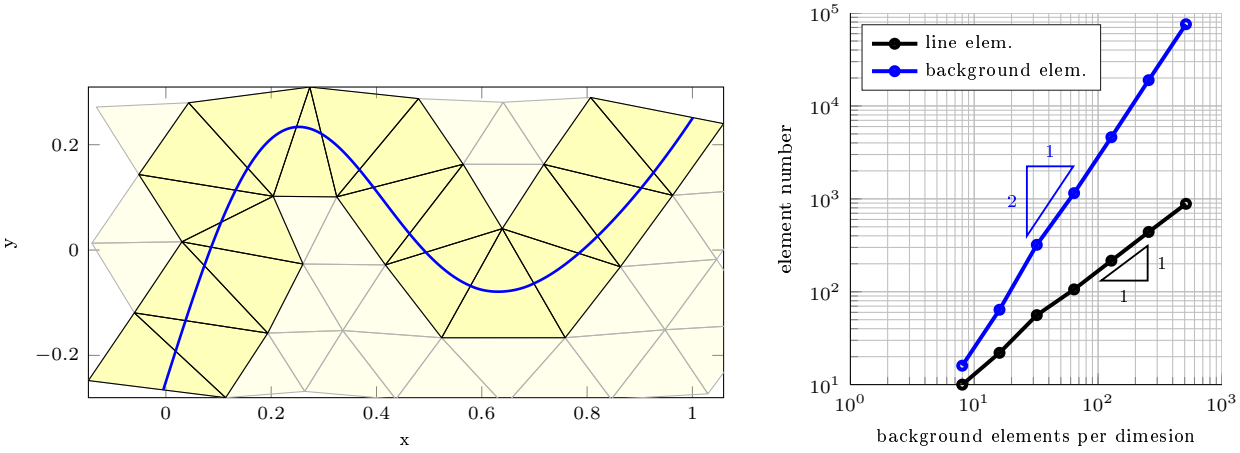
and the level-set function can be obtained as

$$\phi(\boldsymbol{x}) = f(x) - y \ . \tag{3.2}$$

Topological relationships

As mentioned before, a 2D background mesh is composed by linear triangular elements. Therefore each element can have a maximum of three neighbors. The topological relationships are stored in an integer matrix with the dimension $n_{\text{Elem}} \times 3$. The first index indicates the element number and second index indicates the reference edge number. If there is a neighbor, the corresponding element number will be placed at this position, otherwise an edge is on the boundary and a zero is placed.

In order to reduce the computational costs only the cut background elements, which is equal to the number of line elements, are considered. In Fig. 3.8(a) the cut (yellow) and non-cut (shaded) background elements are plotted. The blue line is the zero-level set of the level-set function. In



(a) Background mesh and zero-level set **(b)** Total element numbers

Fig. 3.8: Reduction of the background mesh. (a) cut background elements and zero-level set are highlighted, (b) increment of background and line elements

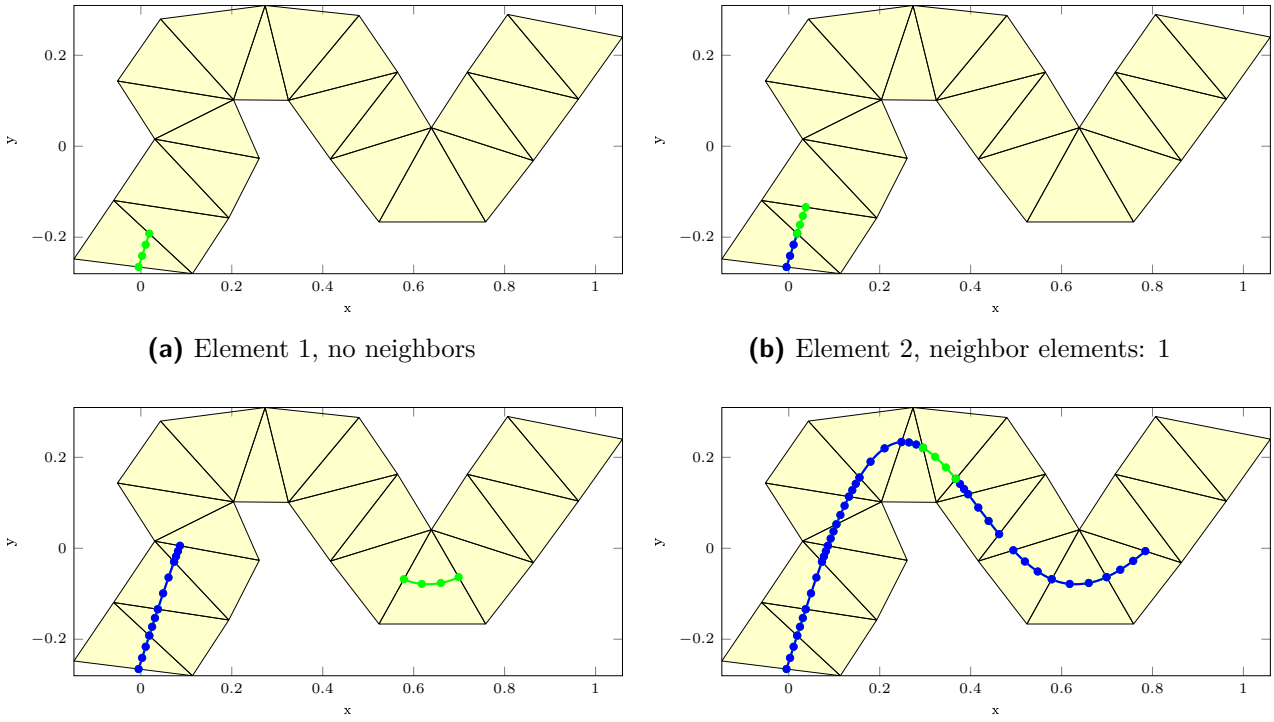
Fig. 3.8(b) the growth of the background and line element number is plotted. It is seen that, the

background mesh element number increases with $\mathcal{O}(n^2)$ and the number of line elements grows with $\mathcal{O}(n^1)$. Therefore the reduction to only considering *cut* background elements decreases the computational costs dramatically.

Remove duplicate nodes in the connectivity matrix

For a line mesh just the corner nodes of the line elements may be shared. In order to identify the duplicate nodes, a loop over the manifold elements is performed and for each element the neighbor situation is used to check if a neighboring line element exists. If there is a neighboring element, which is already added in the updated connectivity matrix, with the cut scenarios the shared nodes are identified and the node numbers are stored accordingly.

Therefore four different situations during the update of the connectivity matrix can appear. In Fig. 3.9 these situations are illustrated. The elements, which are drawn in green are the current elements. The blue elements are already added in the updated connectivity matrix.



(a) Element 1, no neighbors

(b) Element 2, neighbor elements: 1

(c) Element 5, no neighbors

(d) Element 15, neighbor elements: 11, 14

Fig. 3.9: Update situations

In the first situation, see Fig. 3.9(a), element 1 is added. This element is at the boundary and therefore just one neighbor exits, but the neighboring element is not added yet. Therefore for element 1 changes in the connectivity matrix are not necessary.

In Fig. 3.9(b) element 2 is added. We can see that the current element shares the lower, left node with element 1. With the case cut scenario the orientation of the element can be determined and consequently the duplicate node is defined and the shared node number from element 1 can be placed in the connectivity matrix.

In Fig. 3.9(c) element 5 is added. Due to the fact that the reconstruction is not along the level set function, the current neighbor elements are not added yet and therefore this case is similar to the first situation.

In Fig. 3.9(d) both neighbors of element 15 are already added. Therefore the first and last node of the element is shared and the connectivity matrix needs to be updated at both corners.

After the update of the connectivity matrix the node numbering is not consistent due to the duplicate nodes. Therefore the remaining nodes are renumbered and the corresponding coordinates are reordered. Finally, the line mesh is generated, see Fig. 3.10. The blue points indicate the element nodes and the red dots are element borders.

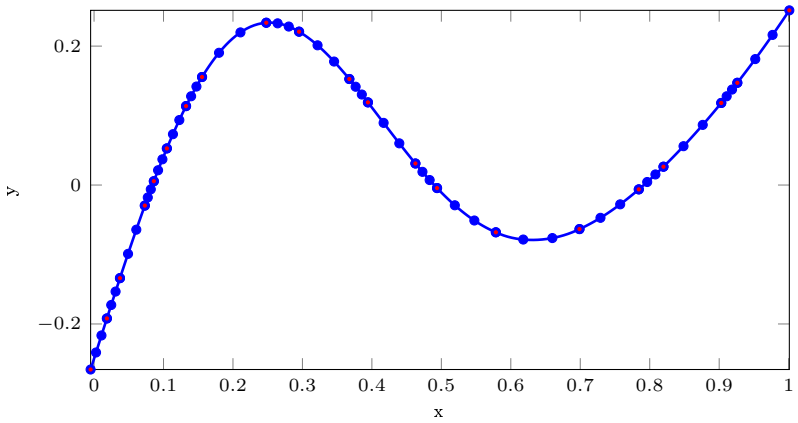


Fig. 3.10: Reconstructed 3rd order line mesh

3.2.2 Surface mesh in a 3D background mesh

In general the procedure for a surface mesh is similar to the algorithm which is shown above, but the topological relationship is more complex. Furthermore the reconstructed surface mesh contains triangular and quadrilateral elements. For the visualizations a sphere with radius $r = 1$ and the center in the origin is chosen. The level set function is defined by:

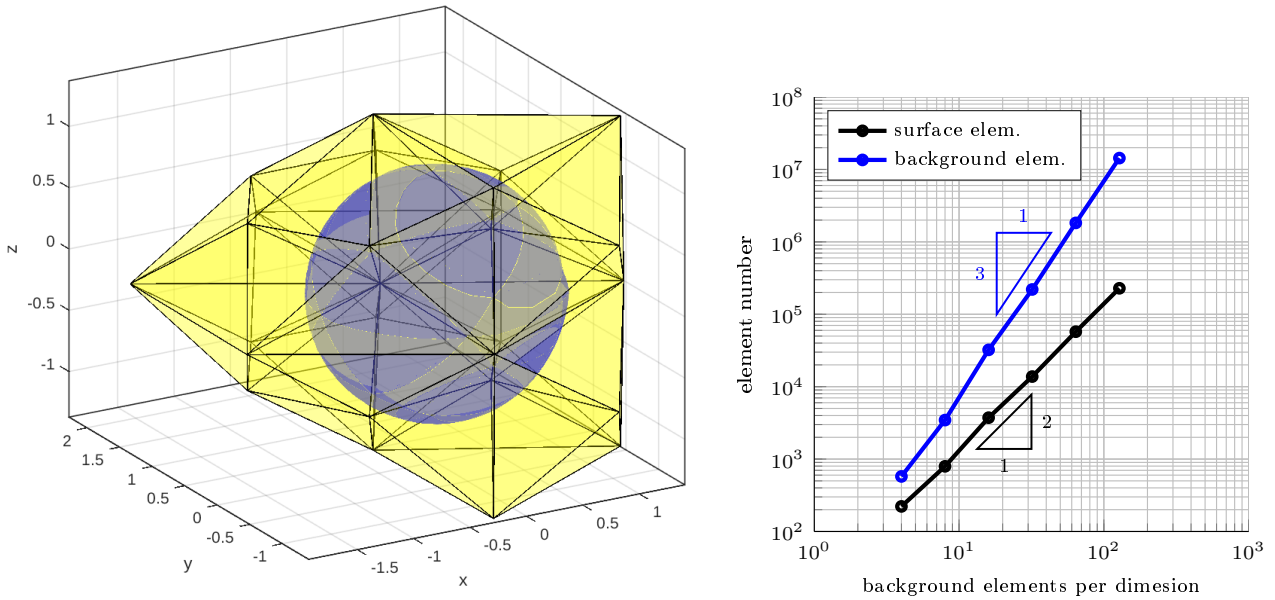
$$\phi(\boldsymbol{x}) = \|\boldsymbol{x}\| - r \tag{3.3}$$

Topological relationships

As mentioned in Section 3.1.1 tetrahedral elements are used. For a 3d background mesh two classes of topological relationships: Face-neighbors and single-edge-neighbors have to be determined.

The algorithm for the face-neighbors is analogous to the edge neighbors of the 2d background mesh, see Section 3.2.1. The topological relationships are stored in an integer matrix with the dimension $n_{\text{Elem}} \times 4$ and the second index indicates the reference face number. The single-edge-neighbor algorithm is based on the methods from the triangle edge neighbors relationship in 3d, which is the same concept like before. Furthermore the already computed face-neighbor information is used to separate the single-edge- from the face-neighbors, otherwise single and multi-edges-neighbors are determined. In contrast to the face-relationship the number of single-edge-neighbors per element is not bounded. Therefore the information will be stored in an array with the dimension nElem $\times$ 6 and the second index indicates the reference edge number. The entries are a vector with the dimension $\hat{n}_{\text{Elem}} \times 1$. $\hat{n}_{\text{Elem}}$ is the number of the single-edge-neighbors per edge, which is not limited.

Again the topological relationships are just determined for the cut background elements in order to decrease the computational costs. Fig. 3.11(a) shows the cut background elements and the zero-level set. The non-cut elements are neglected due to a better visibility.



(a) Background mesh and zero-level set **(b)** Total element number

Fig. 3.11: Reduction of the background mesh. (a) cut background elements and zero-level set of a sphere, (b) increment of background and surface elements

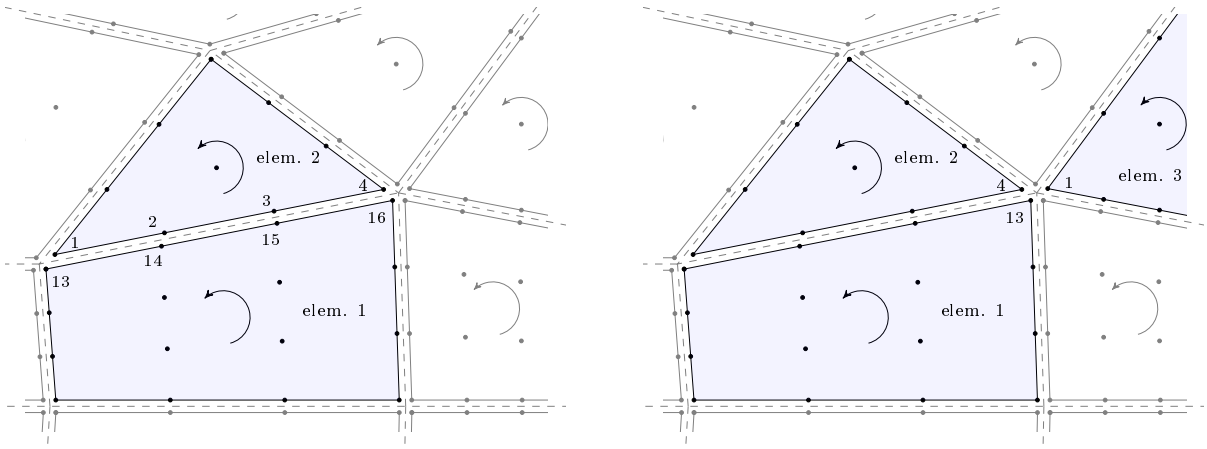
In Fig. 3.11(b) the element number of the background and surface mesh grows with $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$, respectively. Therefore the reduction to only considering *cut* background elements decreases the computational costs dramatically.

Remove duplicate nodes in the connectivity matrix

For surface meshes due to the more complex topological relationships and cut scenarios the update of the connectivity matrix is more challenging than for a line mesh. As is shown in Fig. 3.2 the generated surface mesh contains triangular and quadrilateral elements. In general the identification of the duplicate nodes is similar to the algorithm for a line mesh. A loop over the reconstructed surface elements is performed and with the neighbor relationships the connectivity matrix is updated and renumbered at the end.

Additionally to the four update situations in Fig. 3.9, the neighboring elements may share an *edge* (case 1) or just *one corner* (case 2). The first case can be identified with the face-neighbor relationship and second case with the single-edge-neighbor relationship. Furthermore if a shared edge occurs, the direction of the reference node-numbering has to be considered too. With the constraint that the sense of rotation, which can be described by the normal vector, must be the same for each element, it can be said that the reference node numbering between two elements is always opposite.

In Fig. 3.12(a) element 1 and 2, which share one edge and their sense of rotation is plotted. Furthermore the reference node numbering for the shared edge is shown. The shaded elements are not yet added. It is seen that the global orientation of the corresponding edges is opposite. Before the global node numbers are placed in the updated connectivity matrix the shared edge nodes have to be flipped.



(a) Two elements sharing an edge **(b)** Elements share a corner node

Fig. 3.12: Neighbor situations for a reconstructed surface mesh

In Fig. 3.12(b) element 3 is added to the updated connectivity matrix. In this case element 3 shares one corner node with the elements 1 and 2. After updating and renumbering of the whole connectivity matrix the surface mesh is generated. In Fig. 3.13 a sphere is reconstructed with third order elements.

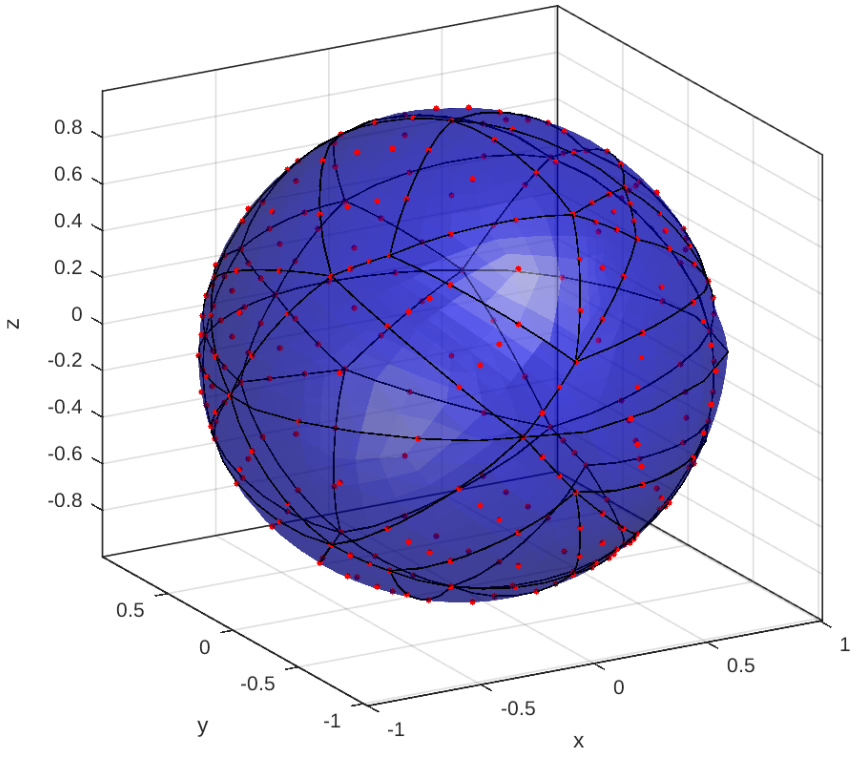
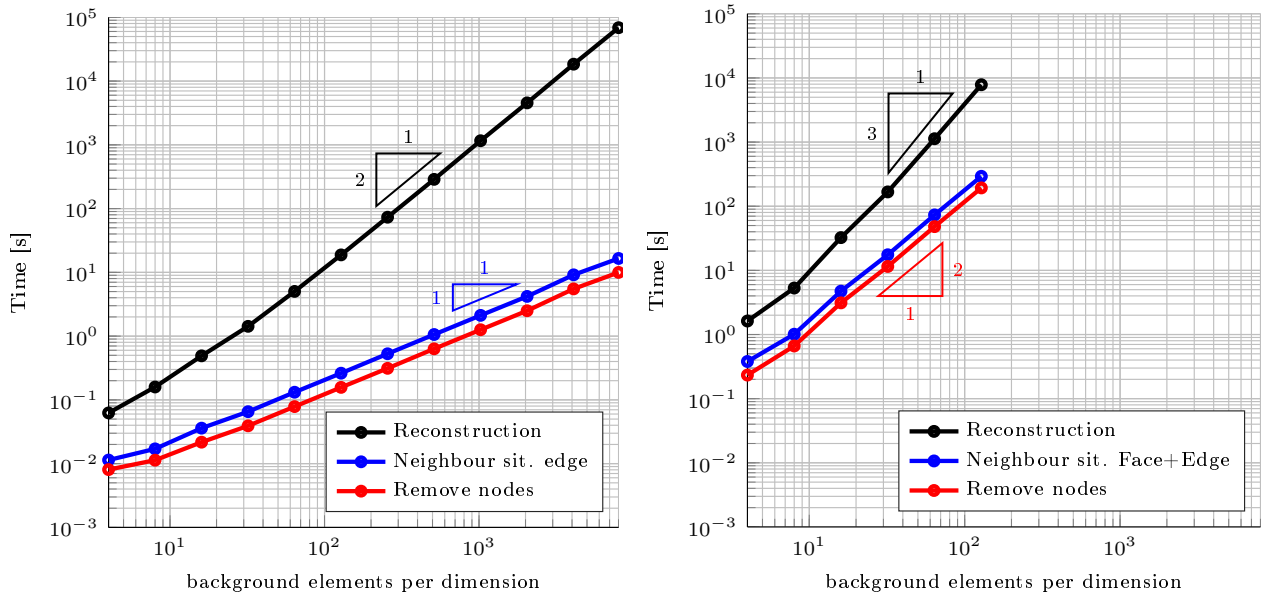


Fig. 3.13: Surface mesh of a sphere

3.3 Computational Costs

After the verification of the implemented algorithms, also the complexity is investigated. Especially for the surface mesh extraction the complexity is important for the architecture of the implemented algorithms.

The following figure shows the computational time subjected to the background elements per dimension in logarithmic scale. Note that for this survey one CPU core is used. In order to get an overview of each step (reconstruction, topological relationships, node removal) the computational time is plotted separately.



(a) Circle in a 2d background mesh

(b) Sphere in a 3d background mesh

Fig. 3.14: Computational costs

In Fig. 3.14(a) a circle with the background-mesh-scale-factor $\hat{n} = 2^a$, $a = 1, 2, \ldots, 13$ and third order elements is reconstructed. The algorithms for the topological relationships and the node removal shows a optimal, linear complexity, within the investigated interval. The reconstruction scales quadratic, because the identification algorithm for the cut elements has to be applied to the whole background mesh $\mathcal{O}(\hat{n}^2)$, which is dominating the procedure. The total computational time for 5.55×10^5 line elements is $\approx 6.88 \times 10^4$ s.

In Fig. 3.14(b) a sphere with the background-mesh-scale-factor $\hat{n} = 2^a$, $a = 2, 3, \ldots, 7$ and third order elements is reconstructed. The reconstruction algorithm scales with a cubic complexity in the tested interval. The reason is the identification algorithm again, because the total background element number scales with $\mathcal{O}(\hat{n}^3)$. The topological relationships and the node removal shows a optimal, quadratic complexity with different constants. The total computational time for 2.28×10^5 surface elements is $\approx 8.32 \times 10^3$ s.

4 FEM-Simulations on manifolds

In general, analytic solutions of PDEs on manifolds are just in special cases available. With the finite element method (FEM), one may obtain numerical solutions, which converge asymptotically to the exact solution. Therefore the variational or weak form of the PDEs is under consideration. In this thesis the Bubnov-Galerkin method is used to determine the weak form [7, Section 4.4].

4.1 General definitions

Before we start with the derivation of the weak form, two important function spaces $L^2(\Sigma)$, $H^1(\Sigma)$ have to be defined.

$$L^2(\Sigma) = \left\{ u : \exists \int_\Sigma (u(\boldsymbol{x}))^2 \, \mathrm{d}s < \infty \right\} \tag{4.1}$$

$$H^1(\Sigma) = \left\{ u \in L^2(\Sigma) : \exists \text{ weak derivitave } \frac{\partial u}{\partial x_i} \in L^2(\Sigma) \ , i = 1, \ldots, n \right\} \tag{4.2}$$

Weak derivative by *Sobolev*:

The integrable function $w(\boldsymbol{x}) = \frac{\partial u}{\partial x_i}$ is a weak derivative of u in respect to x_i, if

$$\int_\Sigma u(\boldsymbol{x}) \frac{\partial \varphi}{\partial x_i} \, \mathrm{d}s = - \int_\Sigma w(\boldsymbol{x}) \, \varphi(\boldsymbol{x}) \, \mathrm{d}s \tag{4.3}$$

is valid $\forall \ \varphi \in \overset{\circ}{C}{}^1(\Sigma)$. The function space $\overset{\circ}{C}{}^1(\Sigma)$ contains all once continuously differentiable functions φ with $\varphi|_\Gamma = 0$, [7, Cha. 4.2].

4.2 Weak form of the Laplace-Beltrami operator

The procedure of the derivation of the weak form of a PDE on manifolds is shown by the inhomogeneous Laplace-Beltrami operator. In [9, Section 2.5] the derivation is given with homogeneous boundary conditions. The strong form of the Laplace-Beltrami operator on a manifold is similar to the Poisson problem in a domain $\Omega \in \mathbb{R}^{n+1}$. With a given source function $f \in L^2(\Sigma)$ find $u : \Sigma \to \mathbb{R}$ such that:

$$-\Delta_\Sigma u = f \quad \text{on} \quad \Sigma, \tag{4.4a}$$

$$u|_{\Gamma_\mathrm{D}} = g_1 \quad \text{on} \quad \Gamma_\mathrm{D}, \tag{4.4b}$$

$$\boldsymbol{n}_\Gamma \cdot \nabla_\Sigma u = 0 \quad \text{on} \quad \Gamma_\mathrm{N}, \tag{4.4c}$$

where $\boldsymbol{n}_\Gamma$ is the unit normal vector on the boundary. For a closed manifold, without boundary, we need an additional condition for the problem to be well-posed. The reason is that we can choose

u = const., which makes the left hand side of Eq. 4.4a to vanish. Since the null space of the Laplace-Beltrami operator on a closed manifold is the space of constant functions, we need to add an additional constraint to only allow the zero function. This zero mean constraint is

$$\int_{\Sigma} u \, \mathrm{d}s = 0 \ . \tag{4.5}$$

Multiplying the strong form (see Eq. 4.4a) with a test function $v \in \mathcal{V}_0$, where $\mathcal{V}_0$ is a suitable function space and integrate in respect to ds we get

$$-\int_{\Sigma} \Delta_{\Sigma} u \, v \, \mathrm{d}s = \int_{\Sigma} v \, f \, \mathrm{d}s \ . \tag{4.6}$$

Using the Green's formula for tangential operators [9, Section 2.5] the variational form can be obtained as:

$$-\int_{\Sigma} \Delta_{\Sigma} u \, v \, \mathrm{d}s = \int_{\Sigma} \nabla_{\Sigma} v \cdot \nabla_{\Sigma} u \, \mathrm{d}s - \int_{\Gamma_{\mathrm{D}}} \boldsymbol{n}_{\Gamma} \cdot \nabla_{\Sigma} u \, v \, \mathrm{d}s - \int_{\Gamma_{\mathrm{N}}} \boldsymbol{n}_{\Gamma} \cdot \nabla_{\Sigma} u \, v \, \mathrm{d}s = \int_{\Sigma} v \cdot f \, \mathrm{d}s \tag{4.7}$$

Due to the choice of the test function space and the homogeneous Neumann condition the integrals over Γ_{D} and Γ_{N} are zero. In particular, the weak form of the Laplace-Beltrami operator is:

Find $u \in \mathcal{V}_{g_1}$, such that

$$a(u, v) = \langle F, v \rangle \qquad \forall v \in \mathcal{V}_0 \tag{4.8}$$

where

$$a(u, v) = \int_{\Sigma} \nabla_{\Sigma} v \cdot \nabla_{\Sigma} u \, \mathrm{d}s$$

$$\langle F, v \rangle = \int_{\Sigma} f \, v \, \mathrm{d}s$$

$$\mathcal{V}_{g_1} = \left\{ u \in H^1(\Sigma) \ : \ u|_{\Gamma_{\mathrm{D}}} = g_1 \right\}$$

For manifolds with boundary:

$$\mathcal{V}_0 = \left\{ v \in H^1(\Sigma) \ : \ v|_{\Gamma_{\mathrm{D}}} = 0 \right\}$$

For closed manifolds:

$$\mathcal{V}_0 = \left\{ v \in H^1(\Sigma) \ : \ \int_{\Sigma} v \, \mathrm{d}s = 0 \right\}$$

Here, $a(u, v)$ is the bilinear form and $\langle F, v \rangle$ is the linear form of the inhomogeneous boundary value problem. For theoretical investigations it is common to consider a homogeneous boundary value problems. With the procedure of the homogenization an inhomogeneous boundary value problem can be reduced to a substituted boundary value problem with homogeneous boundary conditions.

The procedure is shown in [7, Section 4.3] and is applied to Eq. 4.8. Extend $g_1 \in H^1(\Gamma_\mathrm{D})$ properly to $\tilde{g}_1 \in H^1(\Sigma)$ and with the properties of the bilinear form we can replace u by

$$u = w + \tilde{g}_1 \tag{4.9}$$

and solve the substituted boundary value problem from Eq. 4.8:

Find $w \in \mathcal{V}_0$, such that

$$a(w, v) = \left\langle \tilde{F}, v \right\rangle \qquad \forall v \in \mathcal{V}_0$$

where

$$\begin{aligned} a(w, v) &= \int_\Sigma \nabla_\Sigma v \cdot \nabla_\Sigma w \, \mathrm{d}s \\ \left\langle \tilde{F}, v \right\rangle &= \int_\Sigma v \cdot f \, \mathrm{d}s - a(\tilde{g}_1, v) \end{aligned}$$

The existence and uniqueness of the solution is guaranteed by the Lax-Milgram lemma and proven in [9, Section 2.5].

4.2.1 Discrete weak form

Using the iso-parametric concept we choose the element functions N as ansatz for the solution u and test functions v to

$$u \approx u_h = \hat{u}_k N_k(\boldsymbol{x}(\boldsymbol{\xi})) \quad , \qquad v \approx v_h = v_j N_j(\boldsymbol{x}(\boldsymbol{\xi})) \ , \tag{4.10}$$

where, $\hat{u}_k$ denotes the approximative solution at the element nodes k and $v_j = \delta_{kj}$. Therefore the weak form (Eq. 4.8) leads into the discrete weak form:

$$\begin{aligned} \int_\Sigma N_{j,i} \, N_{k,i} \, \mathrm{d}s \; \hat{u}_k &= \int_\Sigma N_j f \, \mathrm{d}s \\ \int_\Sigma \boldsymbol{N}_{,i}^\mathrm{T} \cdot \boldsymbol{N}_{,i} \, \mathrm{d}s \; \hat{\boldsymbol{u}} &= \int_\Sigma \boldsymbol{N} f \, \mathrm{d}s \\ \boldsymbol{K} \hat{\boldsymbol{u}} &= \boldsymbol{b} \end{aligned} \tag{4.11}$$

Here, $\boldsymbol{K}$ is the stiffness matrix and $\boldsymbol{b}$ is the load vector. Note that the test functions v_h, for manifolds with a boundary, satisfy $v_h \in \mathcal{V}_0$ after the insertion of the boundary conditions.

As mentioned above, for closed manifolds the introduction of the zero-constraint is necessary. For the implementation the Lagrange multiplier μ is used [9, Section 3.2].

$$\int_\Sigma u \, \mathrm{d}s \approx \int_\Sigma u_\mathrm{h} \, \mathrm{d}s = \int_\Sigma \boldsymbol{N}^\mathrm{T} \, \mathrm{d}s \cdot \hat{\boldsymbol{u}} = \boldsymbol{C}^\mathrm{T} \cdot \hat{\boldsymbol{u}} = 0 \tag{4.12}$$

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C} \\ \boldsymbol{C}^\mathrm{T} & 0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}} \\ \mu \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix} \tag{4.13}$$

With this modification $v_h \in \mathcal{V}_0$.

4.2.2 Error estimation

In [7, Section 3.8] an a-priori error estimation in the L^2-norm is given:

$$\|u - u_h\|_{0,2,\Sigma} = \sqrt{\int_\Sigma (u(\boldsymbol{x}) - u_h(\boldsymbol{x}))^2 \,\mathrm{d}s}$$
$$\|u - u_h\|_{0,2,\Sigma} \leq c_1 h^{p+1} \tag{4.14}$$

Note that c_1 is a positive constant and is a function of u, but not from the discretization parameter h. p is the order of the element functions. Especially for p-FEM it is important to reach the $p+1$ convergence order for an arbitrary element order.

4.3 Weak form of the in-stationary transport problem

Again, starting with the inhomogeneous strong form in the Euclidean space $\mathbb{R}^{n+1}$ [7, Section 2.2]:

$$c\rho\, u_{,t} + \hat{a}_i\, u_{,i} + (\lambda_i u_{,i})_{,i} = f \quad \text{on} \quad \Sigma$$

Assuming $c\rho = 1$, $\lambda_i = \lambda$ the strong form can be simplified to:

$$u_{,t} + \hat{\boldsymbol{a}}_\Sigma \cdot \nabla_\Sigma u + \lambda \Delta_\Sigma u = f \qquad \text{on} \quad \Sigma, \tag{4.15a}$$
$$u|_{\Gamma_\mathrm{D}} = g_1 \qquad \text{on} \quad \Gamma_\mathrm{D}, \tag{4.15b}$$
$$\boldsymbol{n}_\Gamma \cdot \nabla_\Sigma u = 0 \qquad \text{on} \quad \Gamma_\mathrm{N}, \tag{4.15c}$$
$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) \quad \text{at} \quad t = 0, \text{ and } \forall \boldsymbol{x} \in \Sigma \tag{4.15d}$$

The parameter $\hat{\boldsymbol{a}}_\Sigma$ is the tangential convection term. Applying the procedure from Section 4.2 on Eq. 4.15a the semi-discrete weak form can be written in terms of [7, Section 7.1]:

Find $u_h \in \mathcal{V}_{g_{1h}}$ with $u_{h,t} \in L^2(\Sigma)$, such that

$$(u_{h,t},\, v_h)_0 + a(t; u_h, v_h) = \langle F(t), v\rangle \qquad \forall v \in \mathcal{V}_{0h} \tag{4.16}$$

where

$$(u_{h,t},\, v_h)_0 = \int_\Sigma u_{h,t}\, v_h \,\mathrm{d}s$$
$$a(t; u_h, v_h) = \hat{\boldsymbol{a}}_\Sigma \cdot \int_\Sigma v_h \nabla_\Sigma u_h \,\mathrm{d}s + \lambda \int_\Sigma \nabla_\Sigma v_h \cdot \nabla_\Sigma u_h \,\mathrm{d}s$$
$$\langle F(t), v_h\rangle = \int_\Sigma f(\boldsymbol{x}, t)\, v_h \,\mathrm{d}s$$

for almost all $t \in (0, \mathrm{T})$ and

$$(u_h(\boldsymbol{x}, 0),\, v_h)_0 = (u_0,\, v_h)_0 \quad \forall v \in \mathcal{V}_{0h}$$

satisfies.

Here,

$$\mathcal{V}_{g_1 h} = \{u_h(\boldsymbol{x}, t) \ : \ u_h(\boldsymbol{x}, t) = \hat{u}_i(t) N_i(\boldsymbol{x}) + g_1(x_j, t) N_j(\boldsymbol{x})\} \subset V_{g_1}$$
$$\mathcal{V}_{0h} = \{v_h(\boldsymbol{x}) \ : \ v_h = v_i N_i(\boldsymbol{x})\} \subset \mathcal{V}_0$$

Integrating in respect to the space the semi-discrete weak form yields to:

$$\boldsymbol{M}\hat{\boldsymbol{u}}_{,t}(t) + \boldsymbol{K}\hat{\boldsymbol{u}}(t) = \boldsymbol{b} \tag{4.17}$$

Where

$$\boldsymbol{M} = \int_\Sigma \boldsymbol{N}^{\mathrm{T}} \cdot \boldsymbol{N} \, \mathrm{d}s$$
$$\boldsymbol{K} = \hat{a}_i \int_\Sigma \boldsymbol{N}^{\mathrm{T}} \cdot \boldsymbol{N}_{,i} \, \mathrm{d}s + \lambda \int_\Sigma \boldsymbol{N}_{,i}^{\mathrm{T}} \cdot \boldsymbol{N}_{,i} \, \mathrm{d}s$$

The solution on the element nodes $\hat{\boldsymbol{u}}(t)$ is still a function of the time. Therefore a numerical time integration in order to get the full-discrete weak form is introduced. In this thesis the Crank-Nicolson method ($\sigma = 1/2$) is used [7, Section 7.1]. The continuous time space $(0, \mathrm{T})$ is divided into $0 = t_0 < t_1 < t_m < \ldots t_{m_\mathrm{T}} = \mathrm{T}$ discrete time layers with $\tau = t_{m+1} - t_m$. The full-discrete weak form can be obtained as:

$$(\boldsymbol{M} + \sigma\tau\boldsymbol{K})\,\hat{\boldsymbol{u}}^{(m+1)} = [\boldsymbol{M} - (1-\sigma)\tau\boldsymbol{K}]\,\hat{\boldsymbol{u}}^{(m)} + \tau\,[\sigma\boldsymbol{b}(t_{m+1}) + (1-\sigma)\boldsymbol{b}(t_m)] \tag{4.18}$$

Note that, if the time step $\tau = \text{const.}$ the matrices on the left and right hand side are constant for all time steps.

4.3.1 Error estimation

For time dependent PDEs the total error is depend of the error of the space discretization and the error of the numerical time integration. If both errors converge with the same order, optimal convergence orders can be archived.

The stability and error measurement for a numerical time integration is considered in [7, Section 8]. The Crank-Nicolson is unconditionally stable and has a consistence order of $\mathcal{O}(\tau^2)$ if the error of the space discretization is not dominating.

5 Numerical results

In this section the Laplace-Beltrami operator and the in-stationary transport problem are solved on several manifolds. Herein, we use the method of manufactured solutions Section 5.1.1 to obtain analytic solutions for the convergence studies. The following examples are solved with a regular mesh, in order to validate the implemented source code and with the reconstructed meshes, from Chapter 3, in order to investigate the properties of these meshes.

5.1 Laplace-Beltrami operator

The derivation of the discrete variational form of the Laplace-Beltrami operator is shown in Section 4.2. The analytic solutions are constructed by the method of manufactured solution, which is shown in Section 5.1.1. For each example a convergence analysis is performed. The error is measured in the relative L^2-norm. Furthermore the condition number of the stiffness matrix and the ratio of the element sizes is investigated too.

5.1.1 Manufactured solution

In order to construct an example with analytic solution for an arbitrary manifold, a solution $u(\boldsymbol{x})$ is assumed and the Laplace-Beltrami operator is applied on this function. This yields to the source function, which is used in the numerical approach as source term at the right hand side of the linear system of equations. Therefore the Laplace-Beltrami operator is written in local coordinates [2]:

$$-\frac{1}{\sqrt{|\det \boldsymbol{g}|}}\frac{\partial}{\partial x_i}\left(g^{ij}\sqrt{|\det \boldsymbol{g}|}\frac{\partial u}{\partial x_j}\right) = f, \tag{5.1}$$

where $\boldsymbol{g} = \boldsymbol{J}^{\mathrm{T}} \cdot \boldsymbol{J}$ is the first fundamental form in the differential geometry and g^{ij} are the contravariant entries of the metric tensor. For manifolds (curves) $\Sigma \in \mathbb{R}^1$ Eq. 5.1 yields into:

$$-\Delta u = f = \boldsymbol{J}_{,1}^{\mathrm{T}} \cdot \boldsymbol{J} g_{11}^{-2} u_{,1} - g_{11}^{-1} u_{,11} \tag{5.2}$$

For manifolds (surfaces) $\Sigma \in \mathbb{R}^2$ with the form

$$\boldsymbol{x}(\boldsymbol{r}) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \\ s \\ F(r,s) \end{bmatrix} \tag{5.3}$$

the Laplace-Beltrami operator (Eq. 5.1) leads to:

$$\begin{aligned} -\Delta u = f = & -g^{11}u_{,11} - 2g^{12}u_{,12} - g^{22}u_{,22} \\ & - u_{,1}\left[\frac{1}{\kappa}\left(\kappa_{,1}g^{11} + \kappa_{,2}g^{12}\right) + g_{,1}^{11} + g_{,2}^{12}\right] \\ & - u_{,2}\left[\frac{1}{\kappa}\left(\kappa_{,1}g^{12} + \kappa_{,2}g^{22}\right) + g_{,1}^{12} + g_{,2}^{22}\right] \end{aligned} \tag{5.4}$$

With:

$$\kappa = \sqrt{|\det \boldsymbol{g}|} = \sqrt{F_{,r}^2 + F_{,s}^2 + 1} \quad \rightarrow \quad \det \boldsymbol{g} > 0 \quad \forall\, r,\, s \in \mathbb{R}$$

5.1.2 Example 1 - Circle

At first, the Laplace-Beltrami opterator is solved on a circle. Using polar coordinates $(r,\, \varphi)$ the metric tensor yields to

$$\boldsymbol{g} = \begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix} . \tag{5.5}$$

The solution in polar coordinates is chosen as in [9] to

$$u(r, \varphi) = 12 \sin 3\varphi \tag{5.6}$$

and the source term from Eq. 5.2 yields to

$$f = \frac{108}{r^2} \sin 3\varphi \ . \tag{5.7}$$

In Fig. 5.1 the numerical and analytic solution is illustrated. The numerical approach seems to be quite accurate.

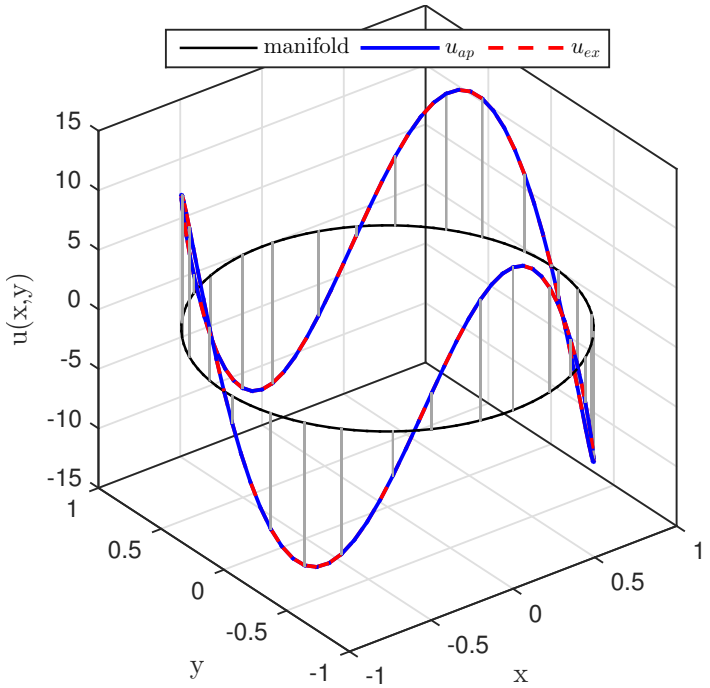
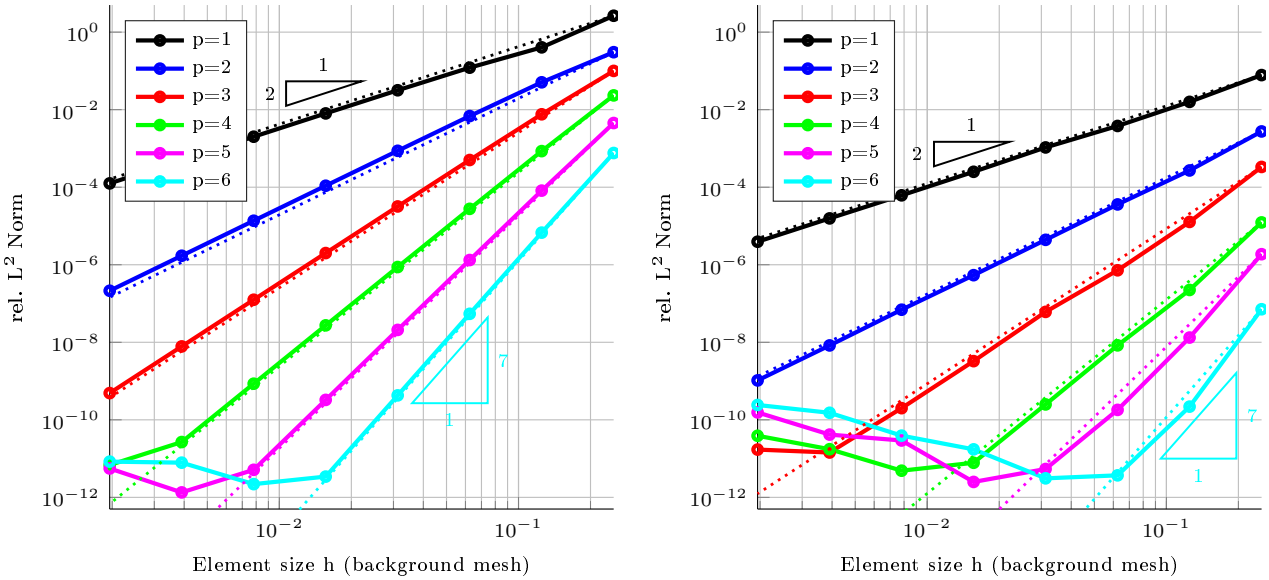


Fig. 5.1: Numerical u_{ap} and analytical u_{ex} solution for a circle

In the Fig. 5.2 - Fig. 5.4 the convergence analysis is performed with the following set of parameters:

Element order	$p \in [1, \ldots, 6]$
Element scale factor (regular mesh)	$n_E = [4, 8, 16, 32, 64, 128, 256, 512]$
Element scale factor (reconstructed mesh)	$\hat{n}_E = [4, 8, 16, 32, 64, 128, 256, 512]$

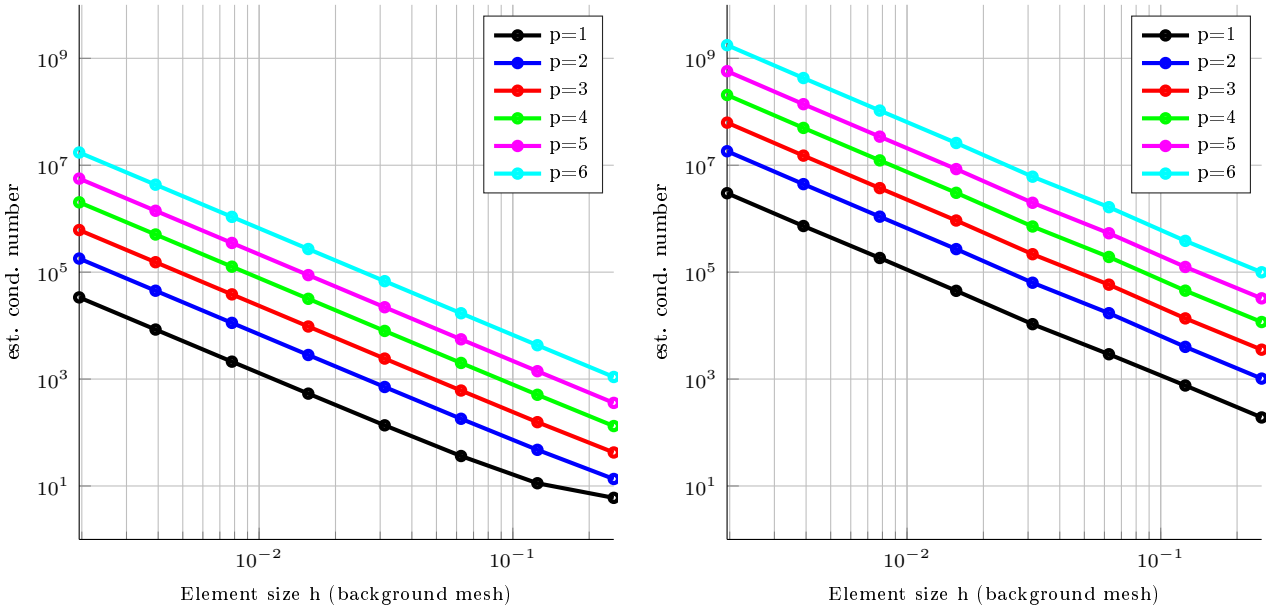
In the left column the analysis is calculated with regular meshes[1] and in the right column meshes from the reconstruction procedure are used. The x-axis of each plot indicates the element size factor $h_j = {}^1/_{n_j}$.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.2: L^2 error analysis

Fig. 5.2 shows the relative L^2 error subjected to the element scale factor h in logarithmic scale and p is the order of the element functions. The dotted lines are the theoretical, optimal convergence orders $\mathcal{O}(p+1)$ [7]. Both types of meshes show optimal convergence rates.

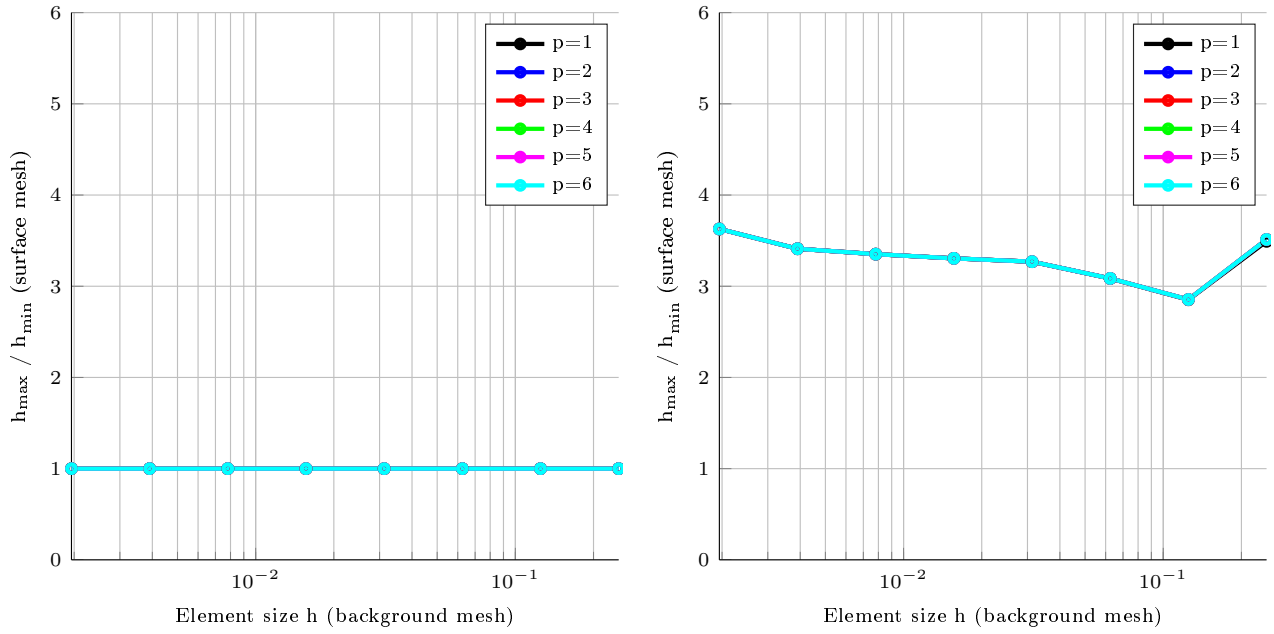


(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.3: Estimated condition Number

In Fig. 5.3 the estimated condition number of the stiffness matrix $\boldsymbol{K}$ is plotted. It is seen that the characteristic of the curves in Fig. 5.3(a) and Fig. 5.3(b) is similar, but for the reconstructed meshes the condition numbers are higher by the factor $\approx 1 \times 10^2$.

[1] These are tailored meshes for the circle where all elements feature the same size.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.4: Ratio between $h_{\max}/h_{\min}$ surface mesh

In Fig. 5.4 the ratio between the largest and smallest element of the surface mesh is plotted. Here the ratio in Fig. 5.4(b) is converging to a constant factor < 4.

5.1.3 Example 2 - Flower

In this example the Laplace-Beltrami operator is solved on a flower shaped manifold. The manifold is defined with the parameter $s \in [0, 2\pi]$

$$\begin{aligned} x(s) &= \left[r_0 + \frac{1}{10}\sin(\Omega s)\right]\cos s \\ y(s) &= \left[r_0 + \frac{1}{10}\sin(\Omega s)\right]\sin s \end{aligned} \tag{5.8}$$

With:

$$s = \arctan\left(\frac{y}{x}\right), \quad r_0 = \frac{1}{2}, \quad \Omega = 8$$

The analytic solution u is chosen to:

$$u(s) = 12\sin 3s \tag{5.9}$$

Inserting this relations in Eq. 5.2 a source term can be generated. In Fig. 5.5 the numerical and analytic solution is presented. The numerical approach seems to be quite accurate.

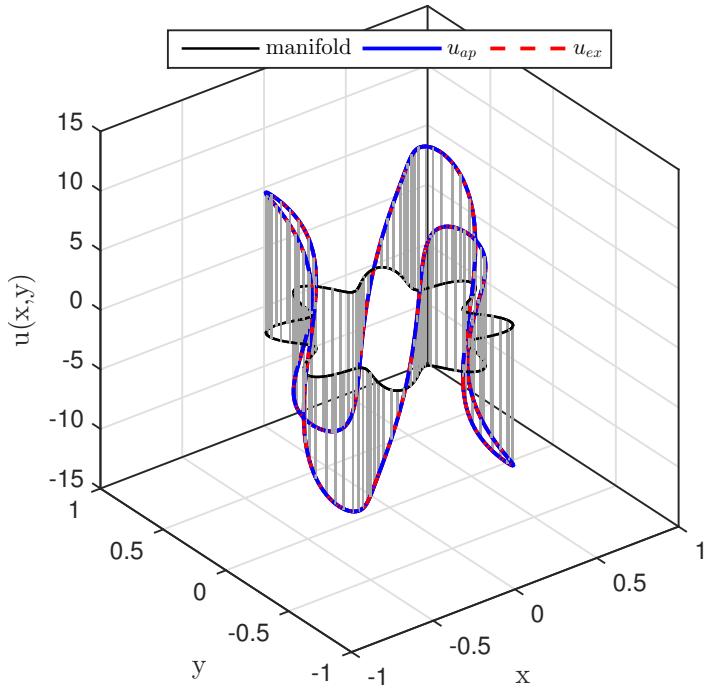
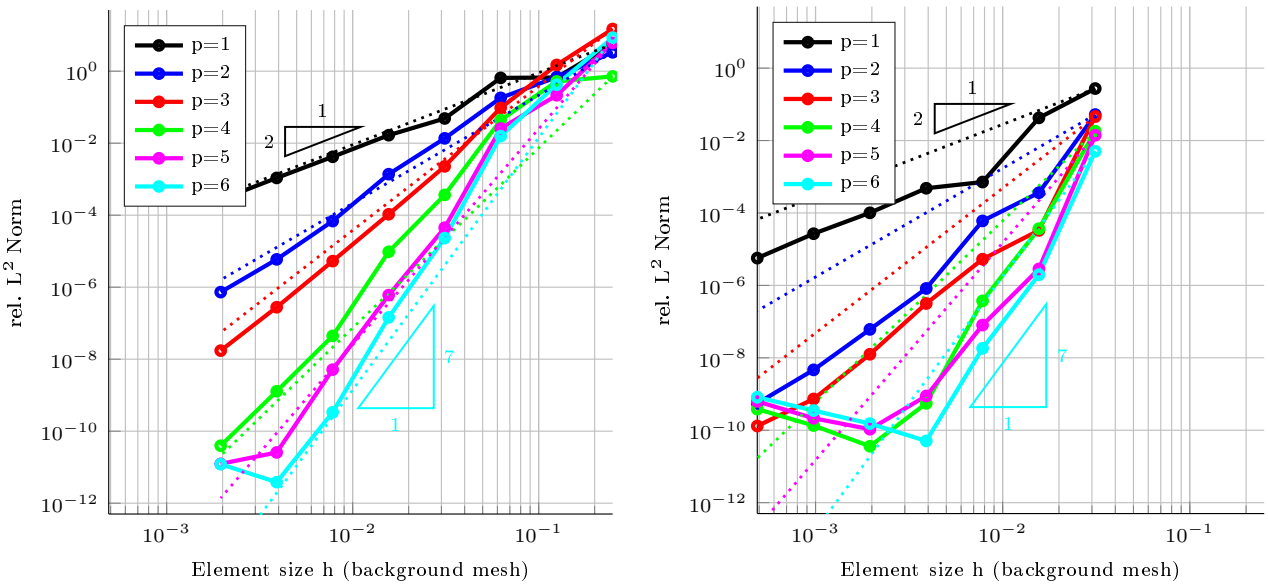


Fig. 5.5: Numerical u_{ap} and analytical u_{ex} solution for a flower-shaped manifold

In the Fig. 5.6 - Fig. 5.8 a convergence analysis is performed with the following set of parameters:

Element order	$p \in [1, \ldots, 6]$
Element scale factor (regular mesh)	$n_E = [4, 8, 16, 32, 64, 128, 256, 512]$
Element scale factor (reconstructed mesh)	$\hat{n}_E = [32, 64, 128, 256, 512, 1024, 2048]$

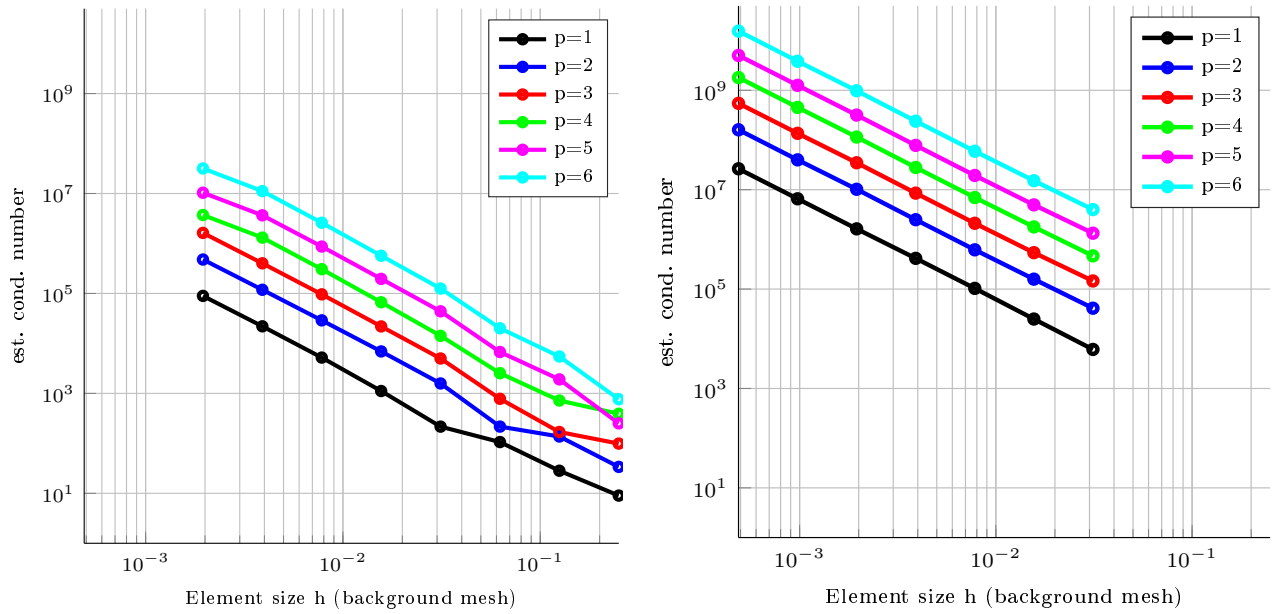
The increase of the element scale factor $\hat{n}_E$ is due to the high curvature of the zero-level set, because for low $\hat{n}_E$ invalid cut scenarios and the Runge-phenomena may occur.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.6: L^2 error analysis

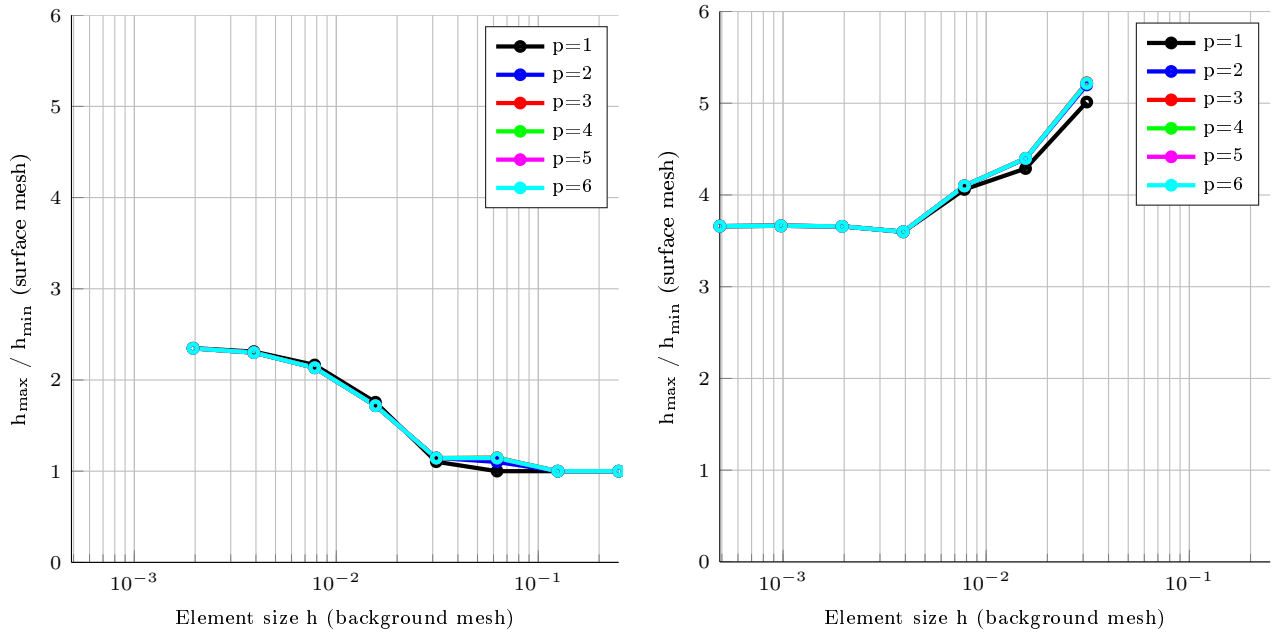
Again optimal convergence rates $\mathcal{O}(p+1)$ can be observed, but the pre-asymptotic range is larger compared to the range of the example with the circle.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.7: Estimated condition Number

In Fig. 5.3 the estimated condition number of the stiffness matrix $\boldsymbol{K}$ is plotted. The difference at each level is approximately the factor 1×10^3.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.8: Ratio between $h_{\max}/h_{\min}$

In Fig. 5.4 the ratio between the largest and smallest element of the surface mesh is plotted and in Fig. 5.8(b) after the first levels the ratio is almost constant and converges again to a limited value < 4.

5.1.4 Example 3 - S-shaped manifold

In this example an "arbitrary" manifold with boundaries is considered. The shape of the manifold and corresponding level-set function is defined in Section 3.2.1. The analytic solution is chosen to:

$$u(\boldsymbol{x}) = \exp(2x) \tag{5.10}$$

Inserting Eq. 3.1 and the analytic solution u in Eq. 5.2 a source term f can be generated. The Dirichlet boundary condition u_{Dir} is evaluated on the boundaries. In Fig. 5.9 the numerical and exact solution are presented.

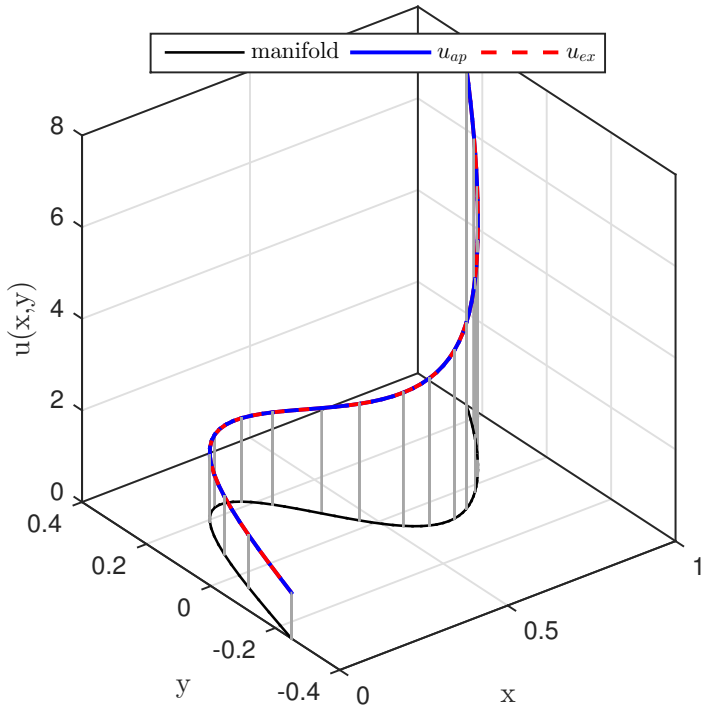
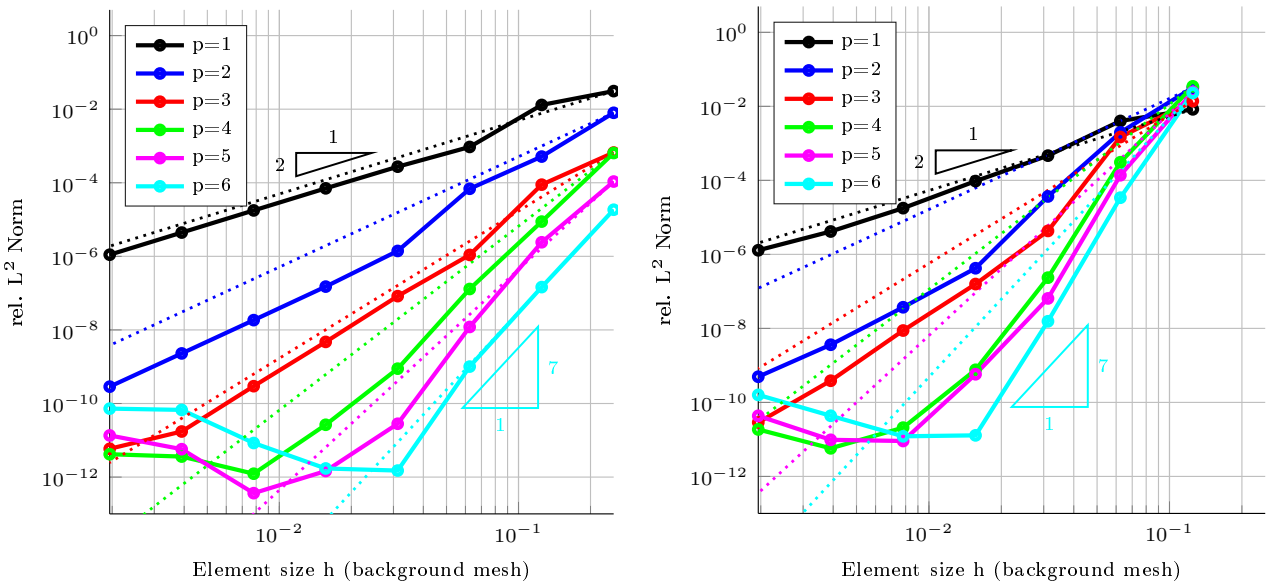


Fig. 5.9: Numerical u_{ap} and analytical u_{ex} solution for a curved manifold.

In the Fig. 5.10 - Fig. 5.12 the convergence analysis is performed with the following set of parameters:

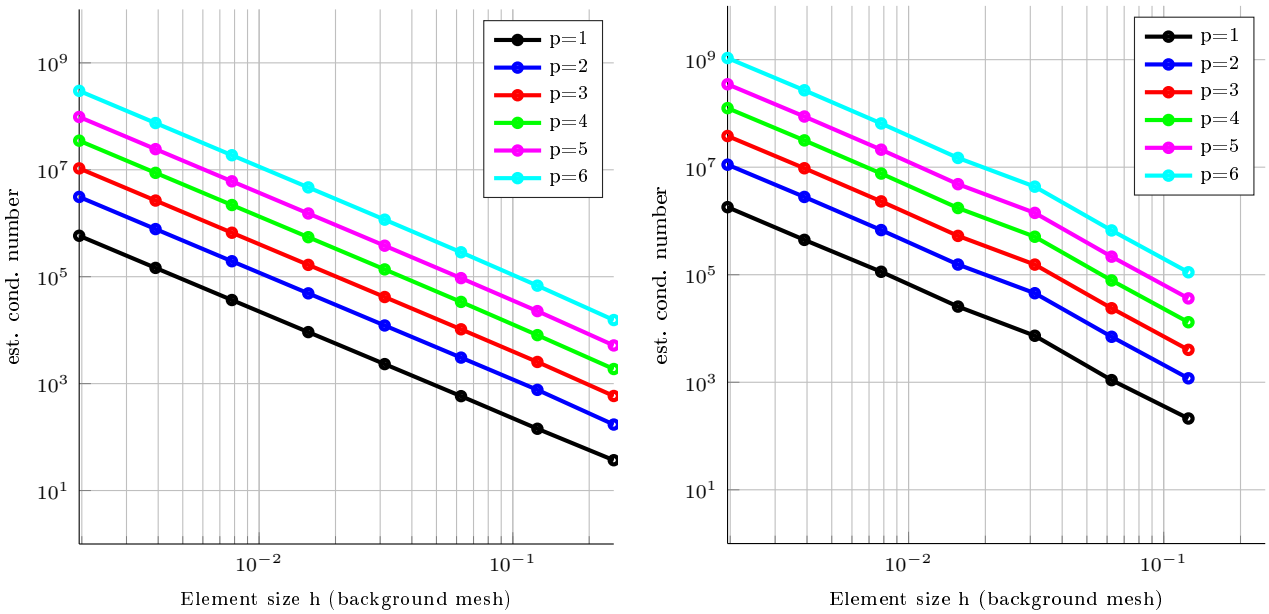
Element order	$p \in [1, \ldots, 6]$
Element scale factor (regular mesh)	$n_E = [4, 8, 16, 32, 64, 128, 256, 512]$
Element scale factor (reconstructed mesh)	$\hat{n}_E = [8, 16, 32, 64, 128, 256, 512]$



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.10: L^2 error analysis

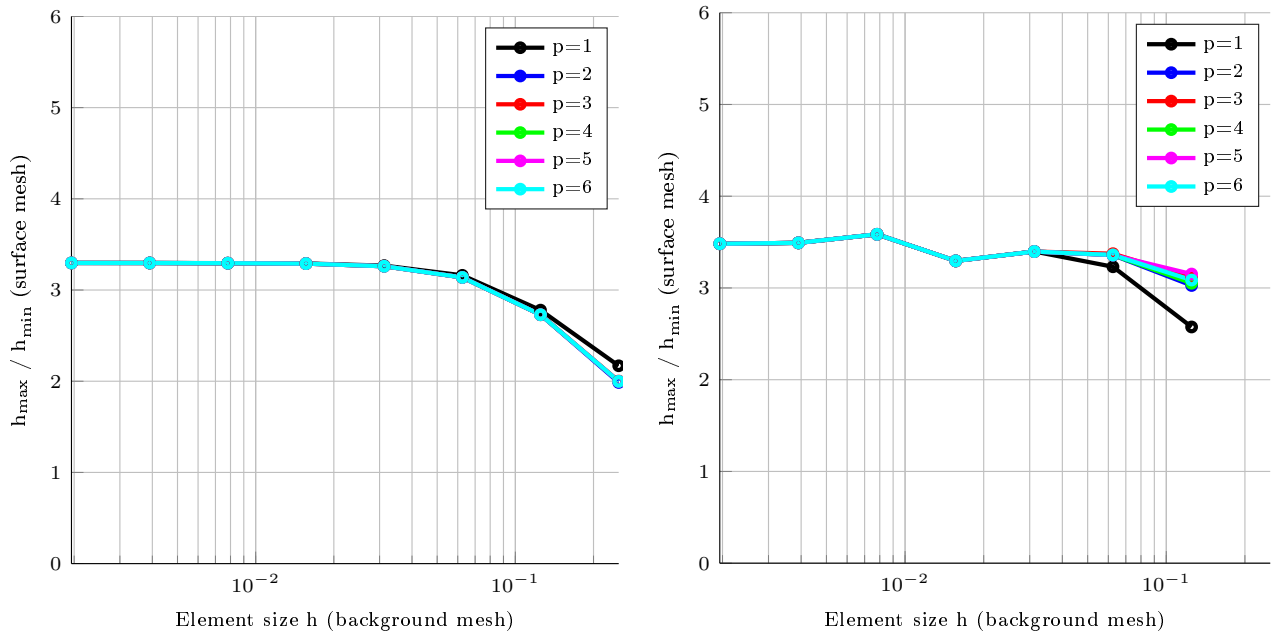
Fig. 5.10 shows the relative L^2 error subjected to the element scale factor h. Both types of meshes achieve optimal convergence orders $\mathcal{O}(p+1)$. Furthermore the reconstructed meshes show a super-convergent behavior.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.11: Estimated condition Number

In Fig. 5.11 the estimated condition number of the stiffness matrix $\boldsymbol{K}$ is plotted. The difference at each level is approximately the factor 1×10^1. The small difference might be due to the almost same element size ratio.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.12: Ration between h_{max}/h_{min}

In Fig. 5.12(b) the ratio between the largest and smallest element of the manifold mesh is plotted and similar to the examples before.

5.1.5 Example 4 - Quarter Cylinder

Here, the Laplace-Beltrami operator is solved on a surface of a quarter cylinder with the radius $r = 1$ and height $L = 4$. The analytical solution in cylindrical coordinates $(r,\, \varphi,\, z)$ is given in [9] with the following parameters and functions:

$$g_{\varphi,1}(\varphi) = (1 - \cos\varphi)\,(1 - \sin\varphi)\,, \tag{5.11}$$

$$g_{\varphi,2}(\varphi) = (\cos\varphi + \sin\varphi - 4\sin\varphi\cos\varphi)\,, \tag{5.12}$$

$$g_z(z) = \sin\frac{\alpha\pi z}{L}, \tag{5.13}$$

$$\alpha = 3\,, \qquad \beta = \frac{1}{\frac{3}{2} - \sqrt{2}}\,. \tag{5.14}$$

The analytic solution is defined by

$$u(r,\, \varphi,\, z) = \beta g_{\varphi,1}(\varphi) g_z(z)\,. \tag{5.15}$$

Using cylindrical coordinates the metric tensor yields to

$$\boldsymbol{g} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.16}$$

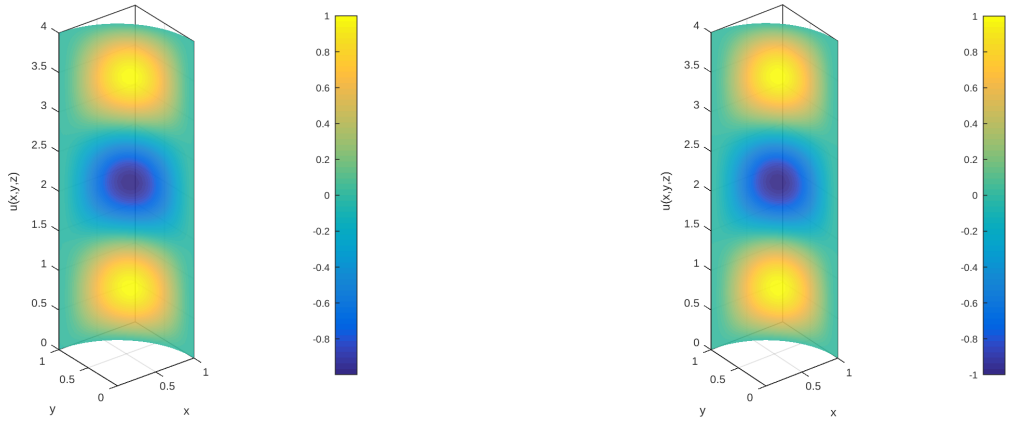
and Eq. 5.1 reduces to

$$-\frac{1}{r}\left(r\, u_{,r}\right)_{,r} - \frac{1}{r^2} u_{,\varphi\varphi} - \frac{1}{r} u_{,zz} = f\,. \tag{5.17}$$

Applying the Laplace-Beltrami operator on the given solution, the source function is obtained as:

$$\begin{aligned} u_{\varphi\varphi} &= \beta g_z(z)\left[\cos\varphi(1 - \sin\varphi) + \sin\varphi(1 - \cos\varphi) - 2\sin\varphi\cos\varphi\right] \\ u_{zz} &= -\beta g_{\varphi,1}(\varphi)\left(\frac{\alpha\pi}{L}\right)^2 \sin\left(\frac{\alpha\pi z}{L}\right) \\ f &= \beta g_z(z)\left[\left(\frac{\alpha\pi}{L}\right)^2 g_{\varphi,1}(\varphi) - g_{\varphi,2}(\varphi)\right] \end{aligned} \tag{5.18}$$

The Dirichlet boundary condition is $u|_{\Gamma_D} = 0$. In the following figure the numerical and exact solution can be compared. The colors on the surface represent the solution of the problem.

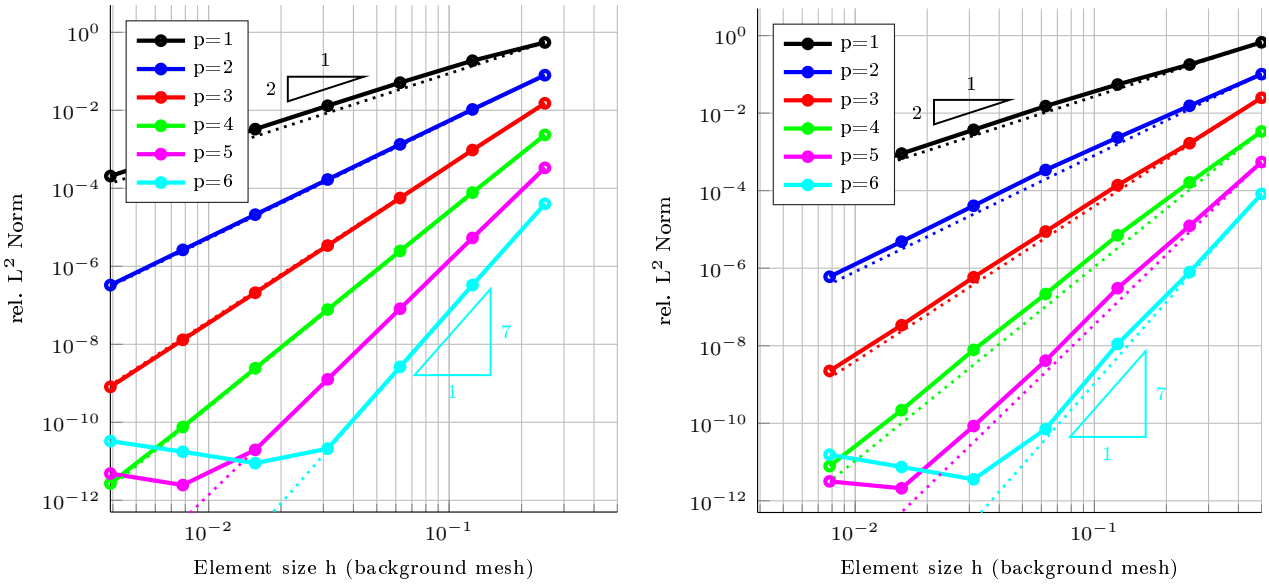


(a) Numerical solution **(b)** Exact solution

Fig. 5.13: Laplace-Beltrami operator on a quarter cylinder

In the Fig. 5.14 - Fig. 5.16 the convergence analysis is performed with the following set of parameters:

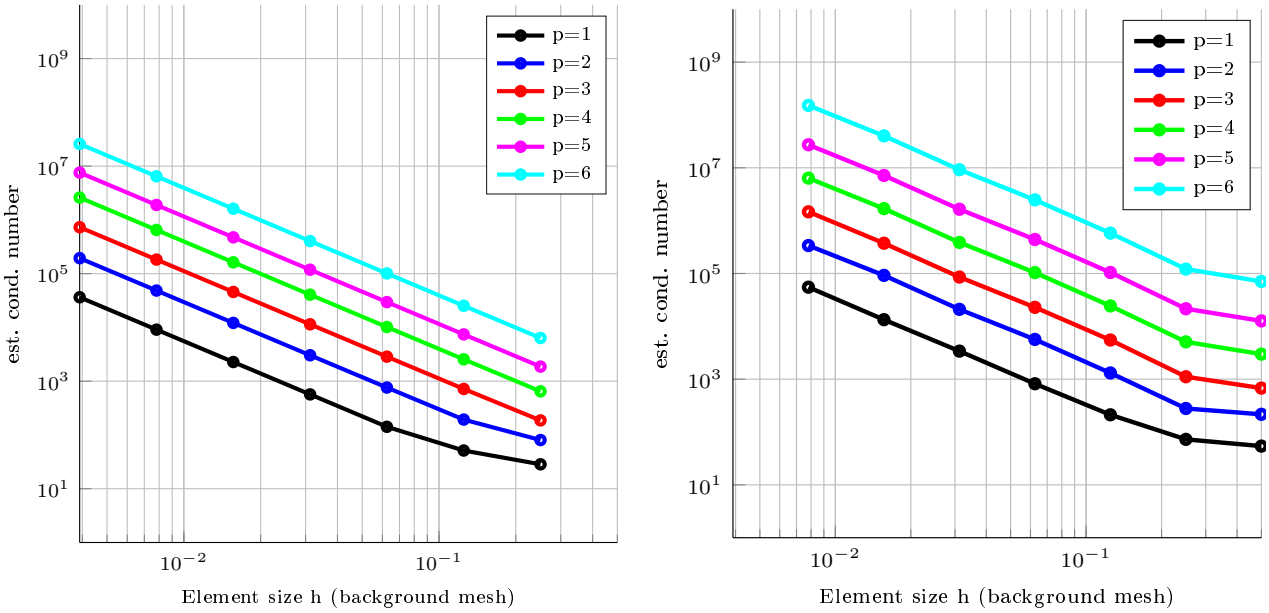
Element order	$p \in [1, \ldots, 6]$
Element scale factor (regular mesh)	$n_E = [2, 4, 8, 16, 32, 64, 128, 256]$
Element scale factor (reconstructed mesh)	$\hat{n}_E = [4, 8, 16, 32, 64, 128]$



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.14: L^2 error analysis

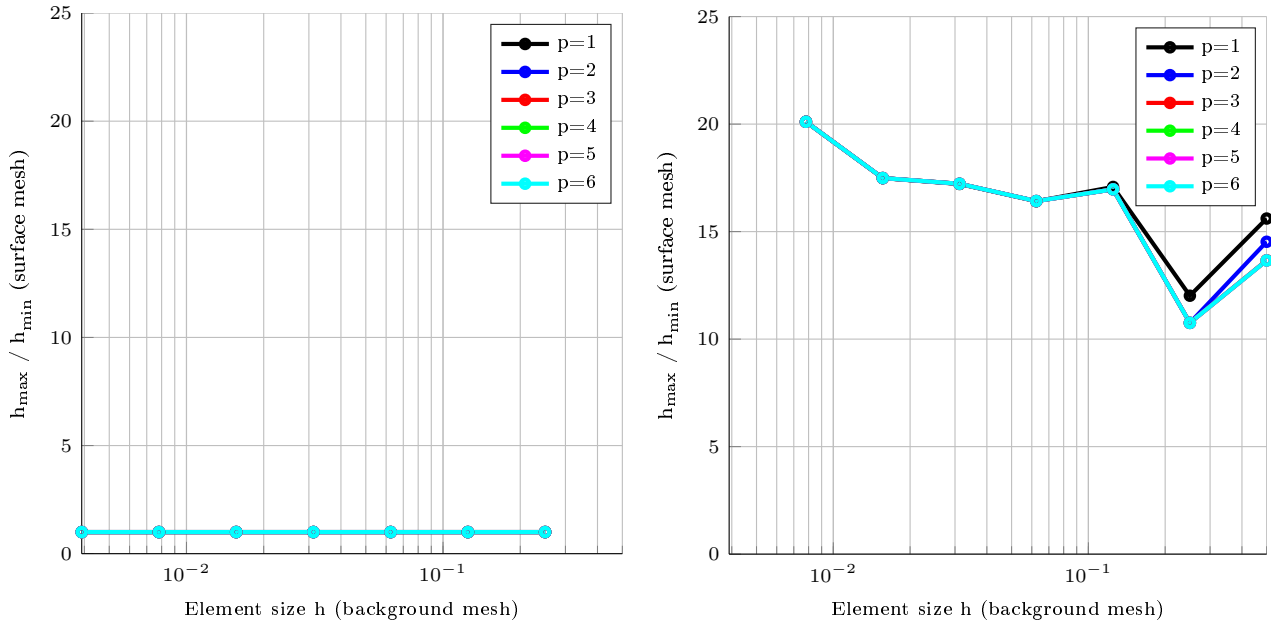
In Fig. 5.14 for both types of meshes optimal convergence orders $\mathcal{O}(p+1)$ [7] can be observed.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.15: Estimated condition Number

Fig. 5.15 shows the estimated condition number of the stiffness matrix $\boldsymbol{K}$. The difference at each level is approximately the factor 1×10^1.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.16: Ration between h_{max}/h_{min}

In Fig. 5.16 the ratio between the largest and smallest element of the surface mesh is plotted. In Fig. 5.16(b) after the first level the element ratio is converging to a constant value around 20. Therefore a significant influence for higher levels on the condition number can not be observed.

5.1.6 Example 5 - Sphere

Here, we consider a sphere with the radius $r = 1$. Using spherical coordinates (r, φ, θ) the metric tensor yields to

$$\boldsymbol{g} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sin^2\theta \end{bmatrix} \tag{5.19}$$

and Eq. 5.1 reduces to

$$-u_{,rr} - \frac{2}{r}u_{,r} - \frac{1}{r^2}u_{,\theta} + \frac{\cot\theta}{r^2}u_{,\theta\theta} - \frac{1}{r^2\sin^2\theta}u_{,\varphi\varphi} = f \ . \tag{5.20}$$

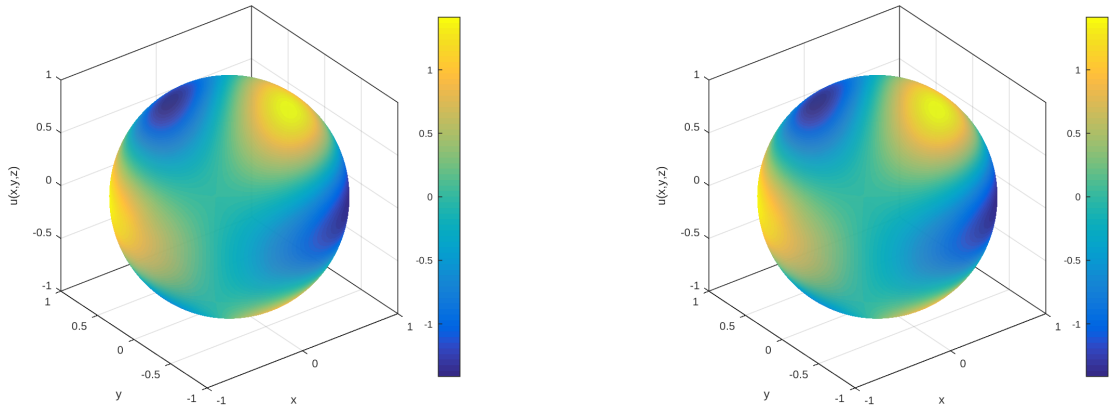
An exact solution $u(r, \varphi, \theta)$, which fulfills the condition from Eq. 4.5 is defined by:

$$u(r, \varphi, \theta) = \sin(3\theta)\,(\cos\varphi - \sin\varphi) \tag{5.21}$$

Applying the Laplace-Beltrami operator on the given solution, the source function is obtained as:

$$f = -\frac{2\sqrt{2}}{\sin(\theta)}\cos\left(\frac{\pi}{4} + \varphi\right)\left(24\cos^4\theta - 29\cos^2\theta + 5\right) \tag{5.22}$$

In the following figure the numeric and exact solution is presented.

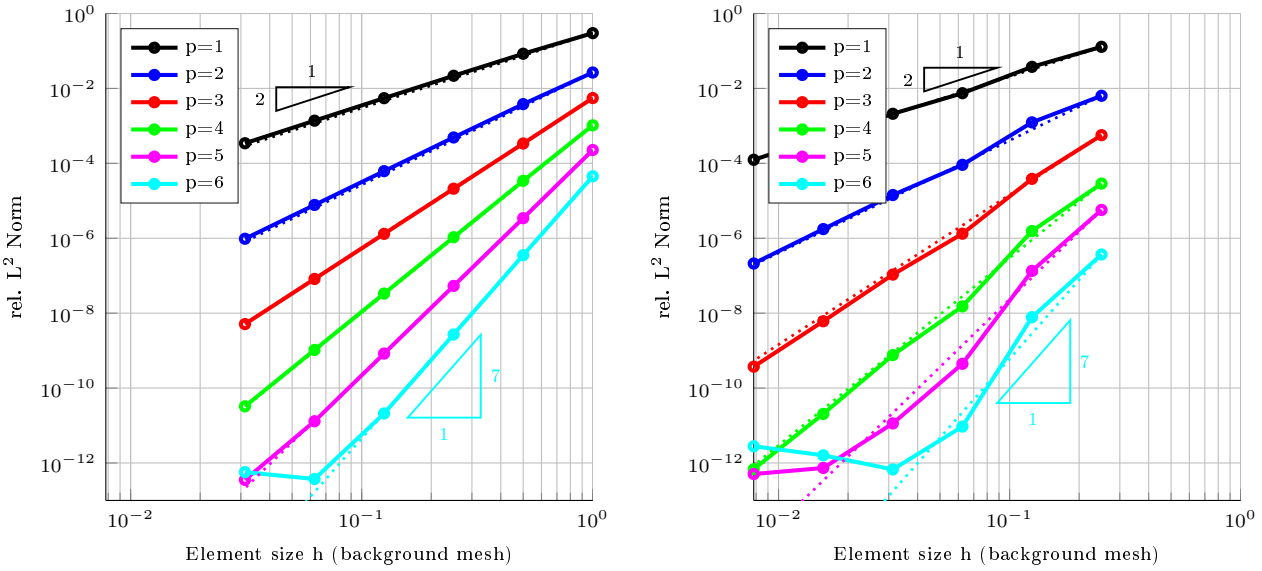


(a) Numerical solution **(b)** Exact solution

Fig. 5.17: Laplace-Beltrami operator on a sphere

In the Fig. 5.18 - Fig. 5.20 the convergence analysis is performed with the following set of parameters:

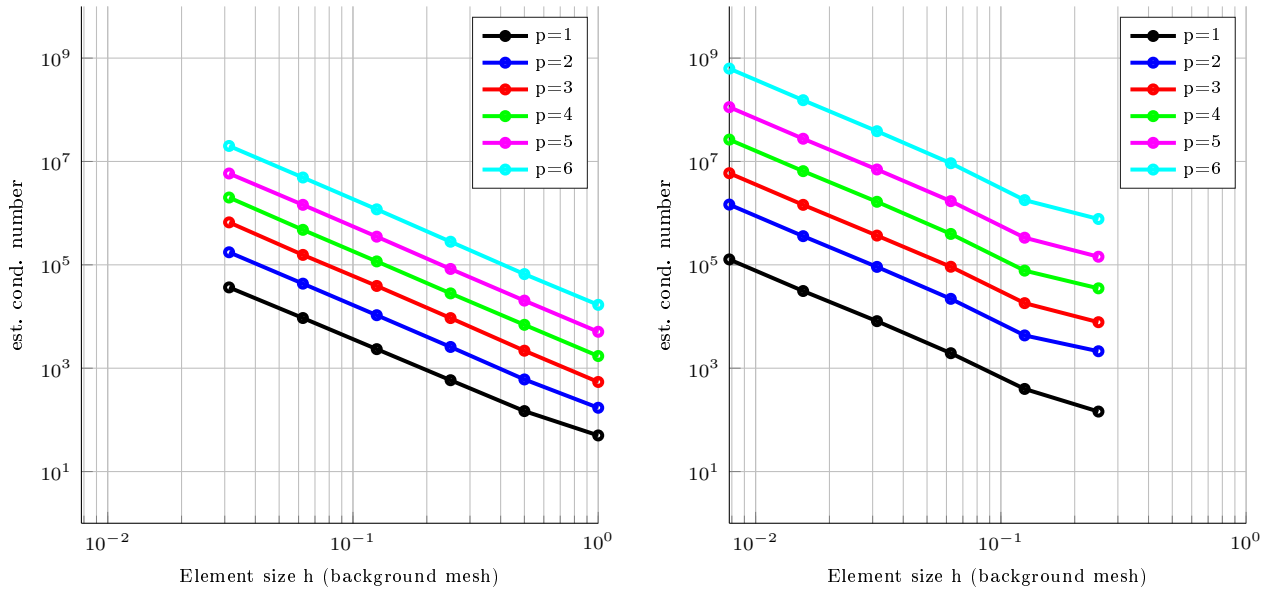
Element order	$p \in [1, \ldots, 6]$
Element scale factor (regular mesh)	$n_E = [1, 2, 4, 8, 16, 32]$
Element scale factor (reconstructed mesh)	$\hat{n}_E = [4, 8, 16, 32, 64, 128]$



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.18: L^2 error analysis

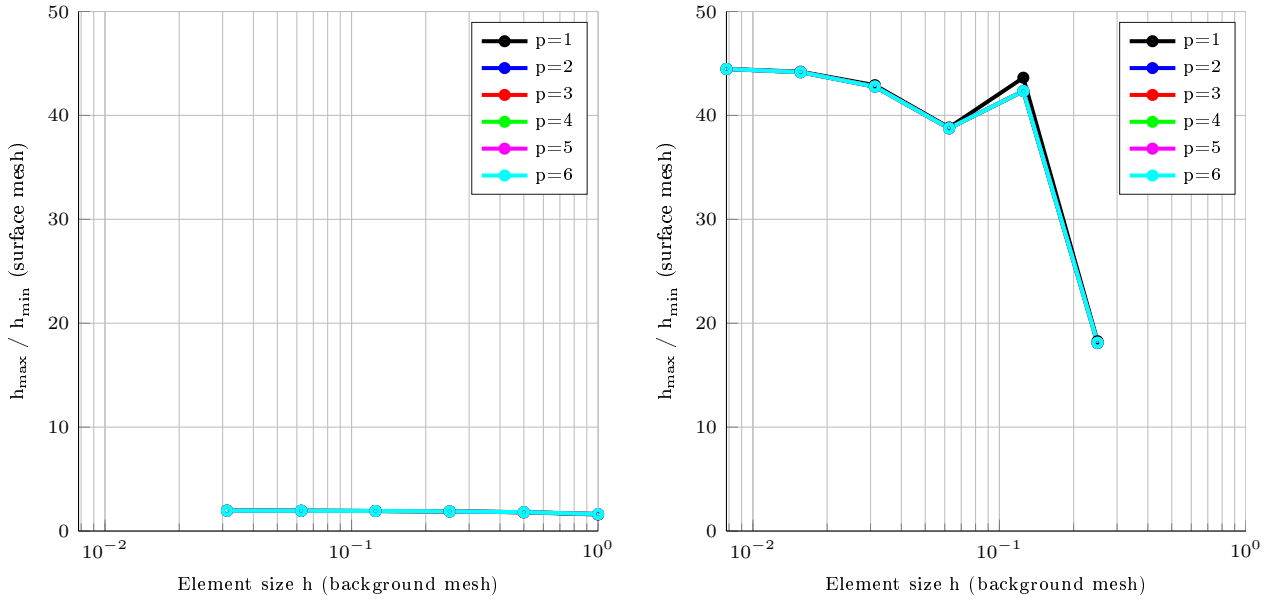
In Fig. 5.18 for both types of meshes optimal convergence orders $\mathcal{O}(p+1)$ [7] can be observed.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.19: Estimated condition Number

Fig. 5.19 shows the estimated condition number of the stiffness matrix $\boldsymbol{K}$. The difference at each level is approximately the factor 3×10^1.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.20: Ration between $h_{\max}/h_{\min}$

In Fig. 5.20 the ratio between the largest and smallest element of the surface mesh is plotted. The ratio in the reconstructed meshes converges to a constant around 45.

5.1.7 Example 6a - Hyperbolic paraboloid with bumps

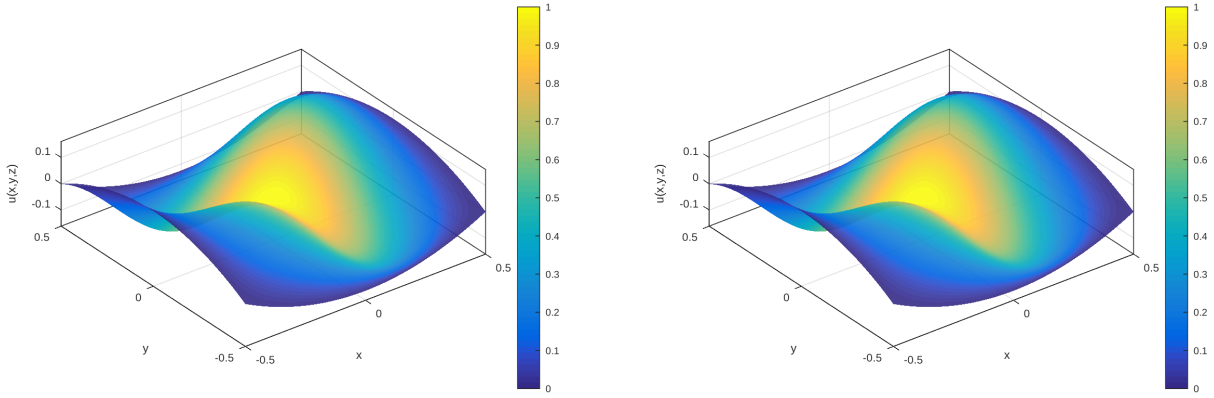
In this example the Laplace-Beltrami operator is solved on a hyperbolic paraboloid with bumps. In this variant the mesh manipulation from Section 3.1.1 is enabled and there is no constraint to the boundary nodes. The manifold is defined by:

$$\begin{aligned} x(r,s) &= r \\ y(r,s) &= s \\ z(r,s) &= \frac{1}{2}\left(r^2 - s^2\right) + \frac{3}{20}\sin\left(2\pi r\right)\sin\left(2\pi s\right) \quad . \end{aligned} \tag{5.23}$$

The analytic solution u is chosen to:

$$u(r,s) = \sin\left[\pi\left(r - \frac{1}{2}\right)\right]\,\sin\left[\pi\left(s - \frac{1}{2}\right)\right] \tag{5.24}$$

Inserting the defined functions in Eq. 5.4 a source term can be generated. In Fig. 5.21 the solution of the numerical and analytic approach is illustrated.

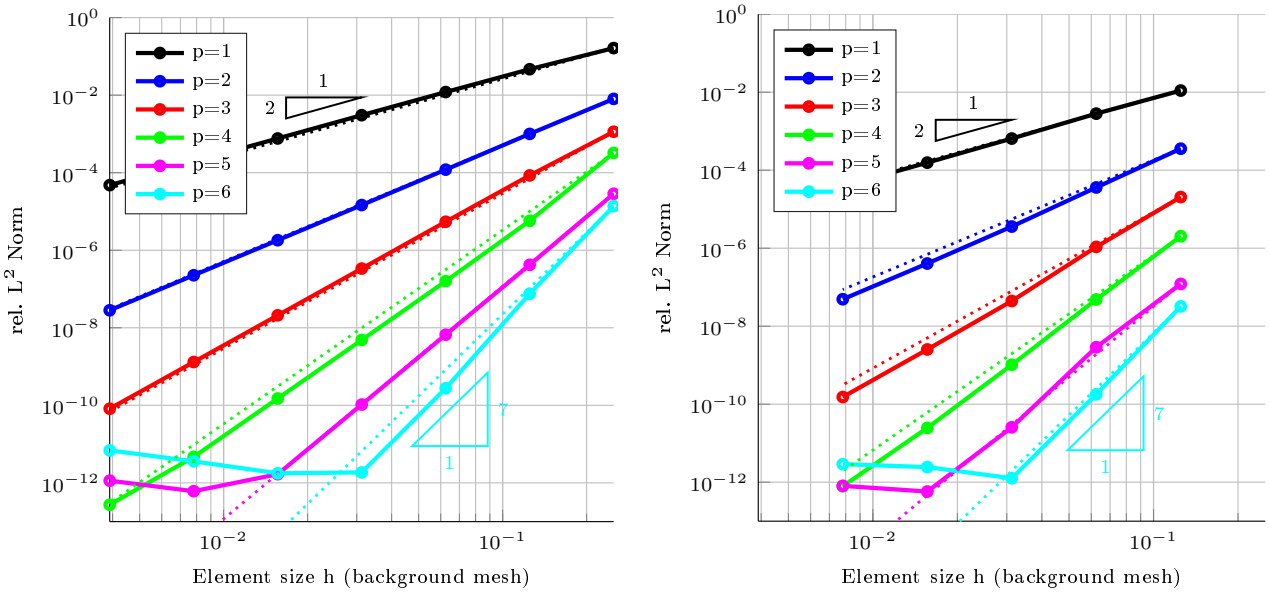


(a) Numerical solution **(b)** Exact solution

Fig. 5.21: Laplace-Beltrami operator on a hyperbolic paraboloid with bumps

In the Fig. 5.22 - Fig. 5.24 the convergence analysis is performed with the following set of parameters:

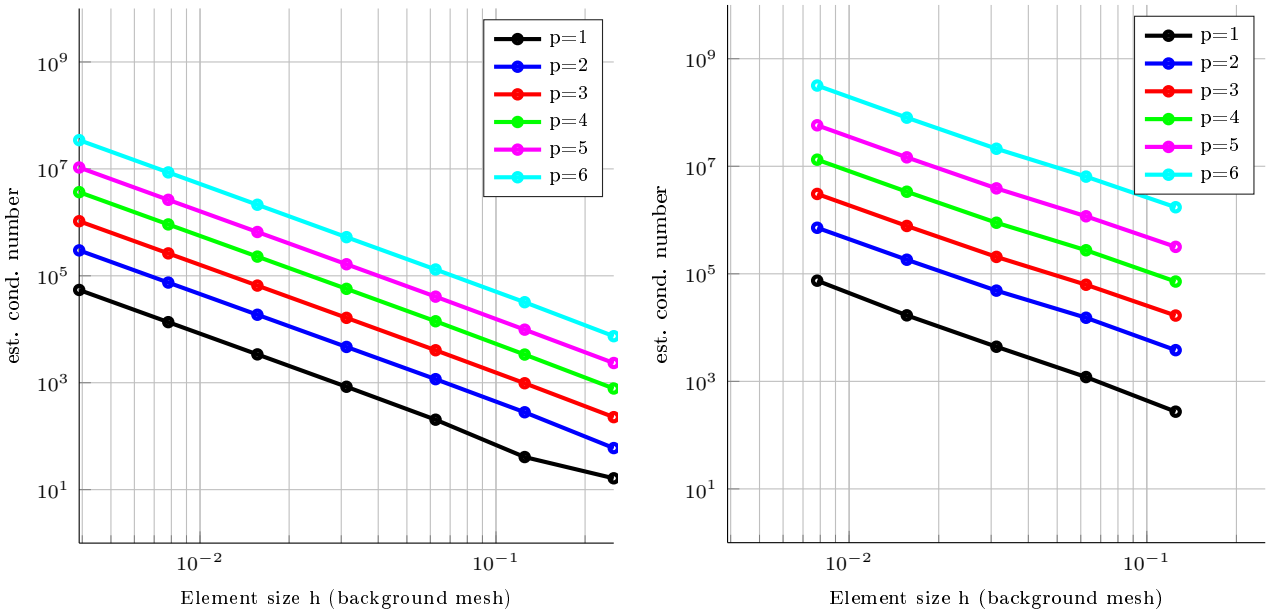
Element order	$p \in [1, \ldots, 6]$
Element scale factor (regular mesh)	$n_E = [4, 8, 16, 32, 64, 128, 256]$
Element scale factor (reconstructed mesh)	$\hat{n}_E = [8, 16, 32, 64, 128]$



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.22: L^2 error analysis

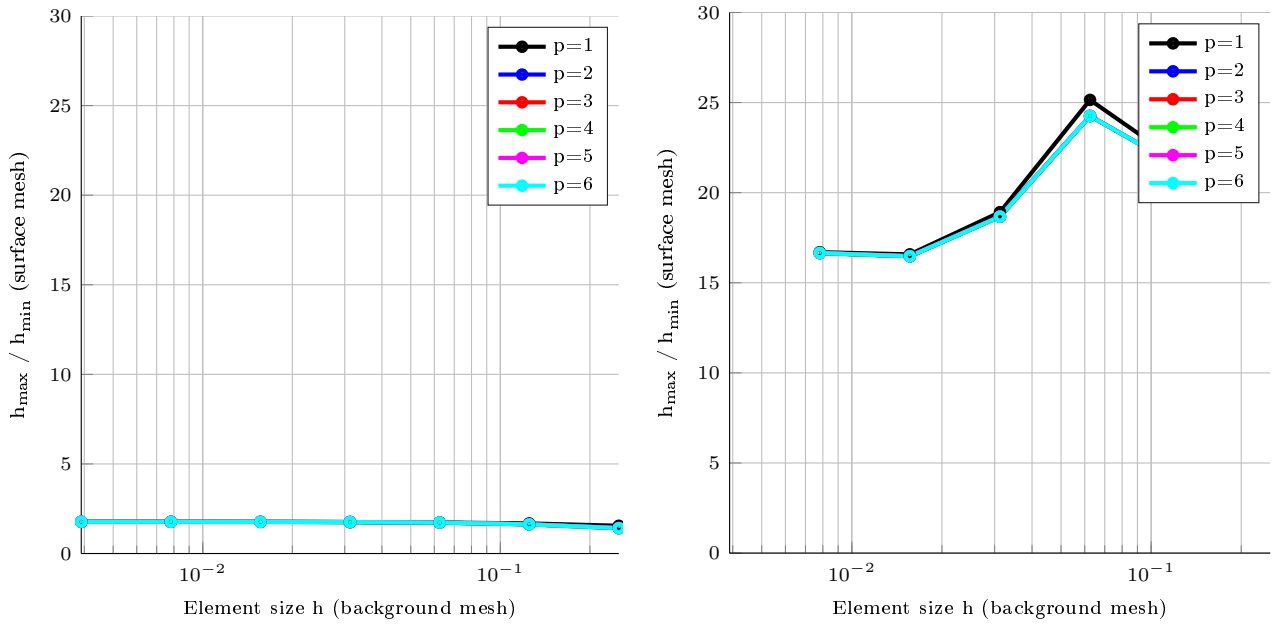
In Fig. 5.22 for both types of meshes optimal convergence orders $\mathcal{O}(p+1)$ [7] can be observed.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.23: Estimated condition Number

Fig. 5.23 shows the estimated condition number of the stiffness matrix $\boldsymbol{K}$. The difference at each level is approximately the factor 1×10^1.



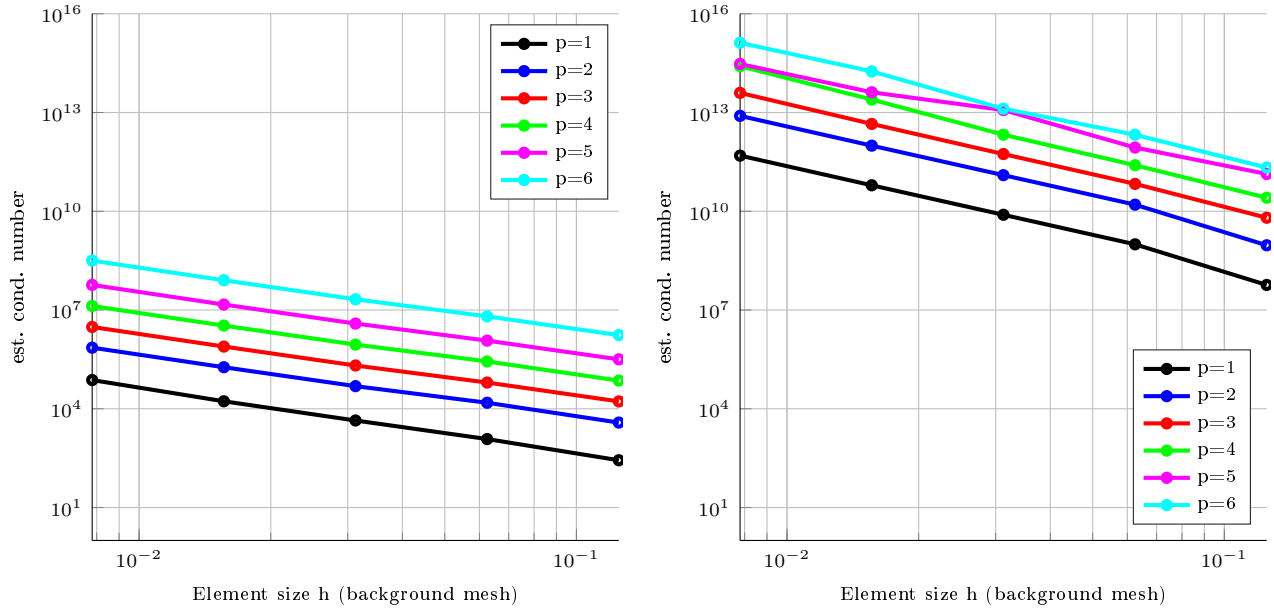
(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.24: Ratio between h_{max}/h_{min}

In Fig. 5.24 the ratio between the largest and smallest element of the surface mesh is plotted. After the second level the ratio in Fig. 5.24(b) is decreasing to ≈ 16.

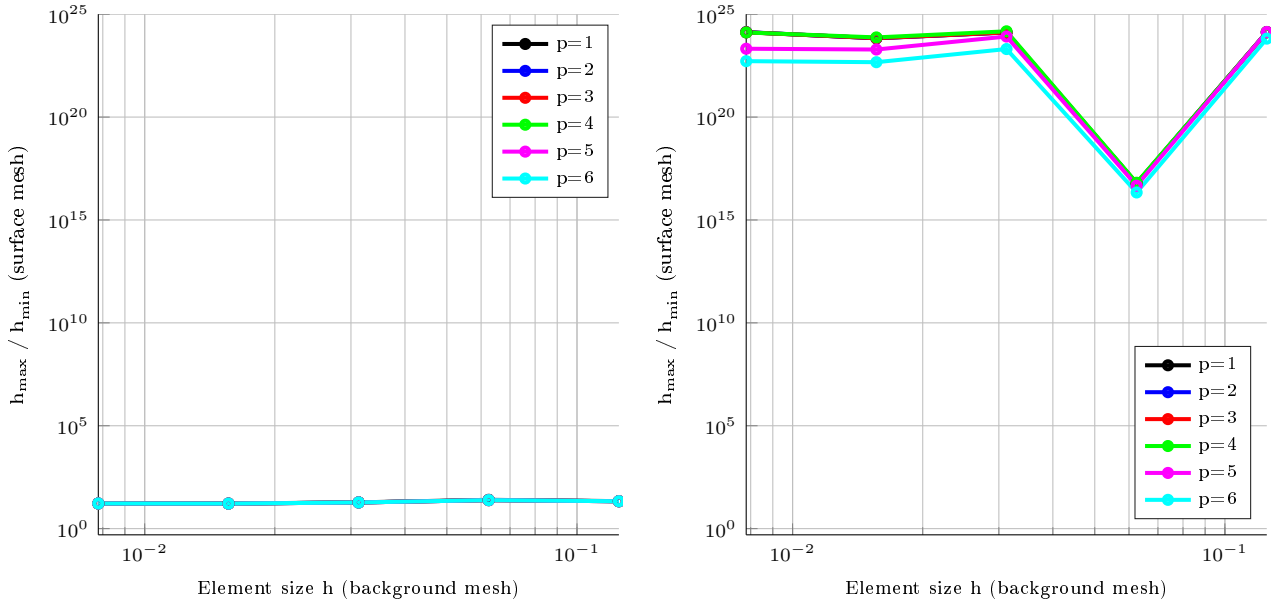
5.1.8 Example 6b - Hyperbolic paraboloid with bumps with fixed nodes

Here, in this variant of example 6a the boundary nodes of the background mesh are forced back to their original position after the mesh manipulation. As in Section 3.1.1 mentioned this constraint may lead to bad element size ratios. In Fig. 5.25 and Fig. 5.26 the consequence of this constraint is illustrated. In the two columns the variants A and B can be compared.



(a) Reconstructed mesh (Variant A) **(b)** Reconstructed mesh (Variant B)

Fig. 5.25: Estimated condition Number



(a) Reconstructed mesh (Variant A)

(b) Reconstructed mesh (Variant B)

Fig. 5.26: Ratio between $h_{\max}/h_{\min}$

The element size ratios differ by the (factor $\approx 6 \times 10^{22}$) and the estimated condition numbers by (factor $\approx 4 \times 10^{6}$). Therefore it can be concluded that, if the boundary nodes are forced back, the mesh manipulation is not able to balance the element sizes anymore and the same problem without mesh manipulation occur again.

5.2 In-stationary transport problem

In this section the in-stationary transport problem is solved. The full-discrete weak form is derived in Section 4.3. In the first two examples the implemented numerical time integration is verified. Therefore on a closed manifold the diffusion term is set to zero and the initial state is "moving" along the manifold without any changes of the shape. In this special case and for simple velocity fields a analytic solution can be constructed. The relative L^2 error is measured at a certain time layer t_m. In order to investigate the error of the time integration, the error of the space discretization should converge with a higher order. Analogous to the Laplace-Beltrami operator the problems are solved on both types of meshes. In the other examples the in-stationary transport problem is illustrated on more complicated manifolds.

5.2.1 Verification in 1d

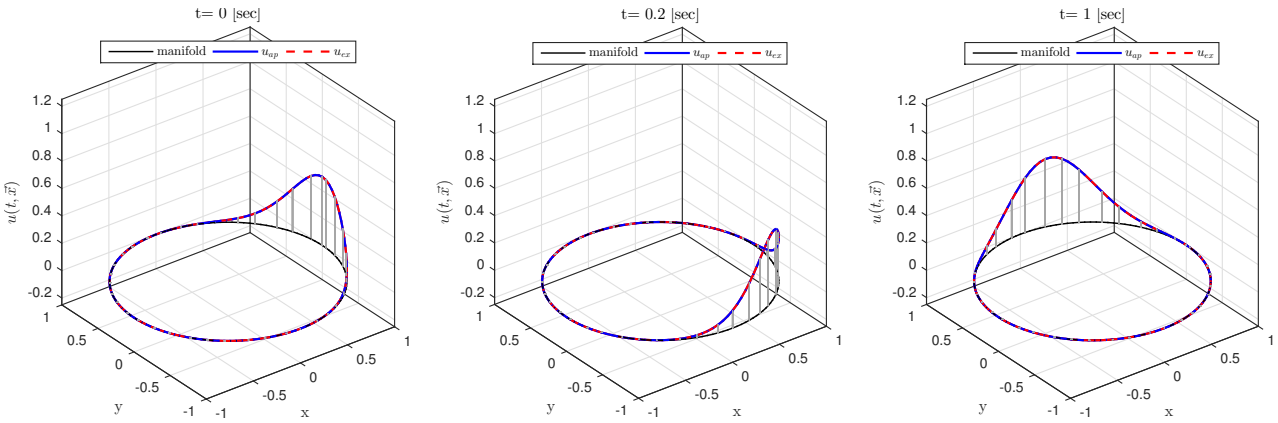
The in-stationary transport problem is solved on a circle $r = 1$ with the following set of parameters:

$$\begin{aligned} \hat{a} &= 5 \qquad \text{(tangential convection)} \\ \lambda &= 0 \\ t_{m_T} &= 1.00\,\text{s} \end{aligned}$$

The initial state is set to

$$u(\boldsymbol{x}, 0) = \exp\left(-4\varphi(\boldsymbol{x})^2\right) \ , \qquad \varphi \in [-\pi,\, \pi] \ . \tag{5.25}$$

The exact solution is obtained as a shift of the angle $\varphi_m = \varphi_0 + \hat{a}\, t_m$. In Fig. 5.27 the solution at $t = [0.00,\, 0.19,\, 1.00]\,\text{s}$ is illustrated.

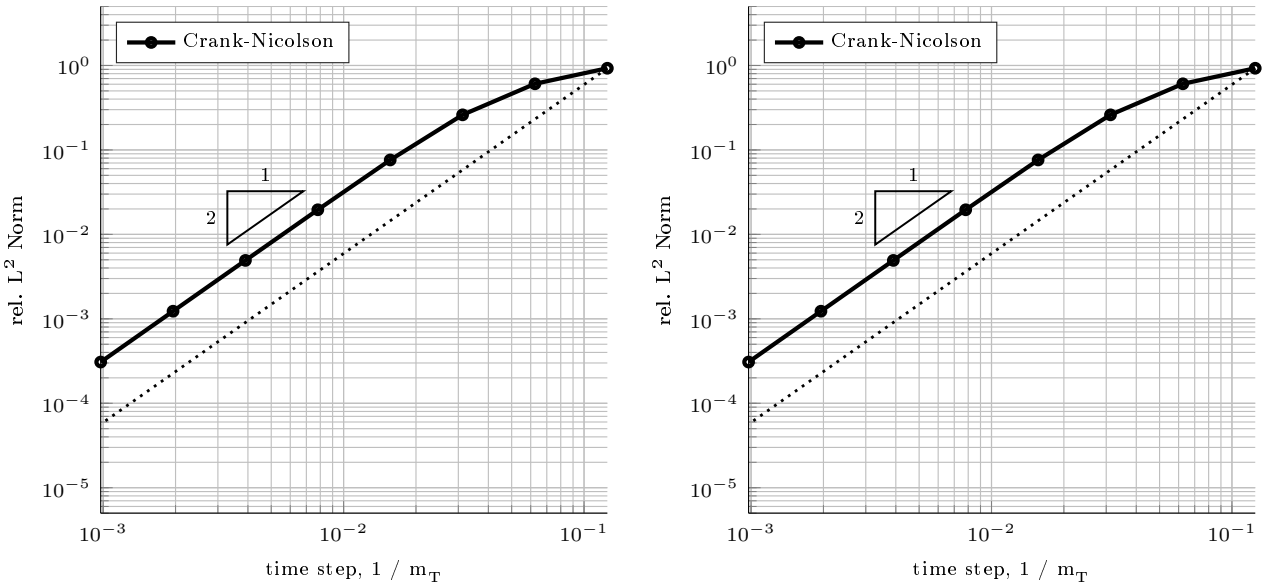


(a) Inital state **(b)** State at $t = 0.19\,\text{s}$ **(c)** State at $t = 1.00\,\text{s}$

Fig. 5.27: Solution of in-stationary transport problem on a circle without diffusion

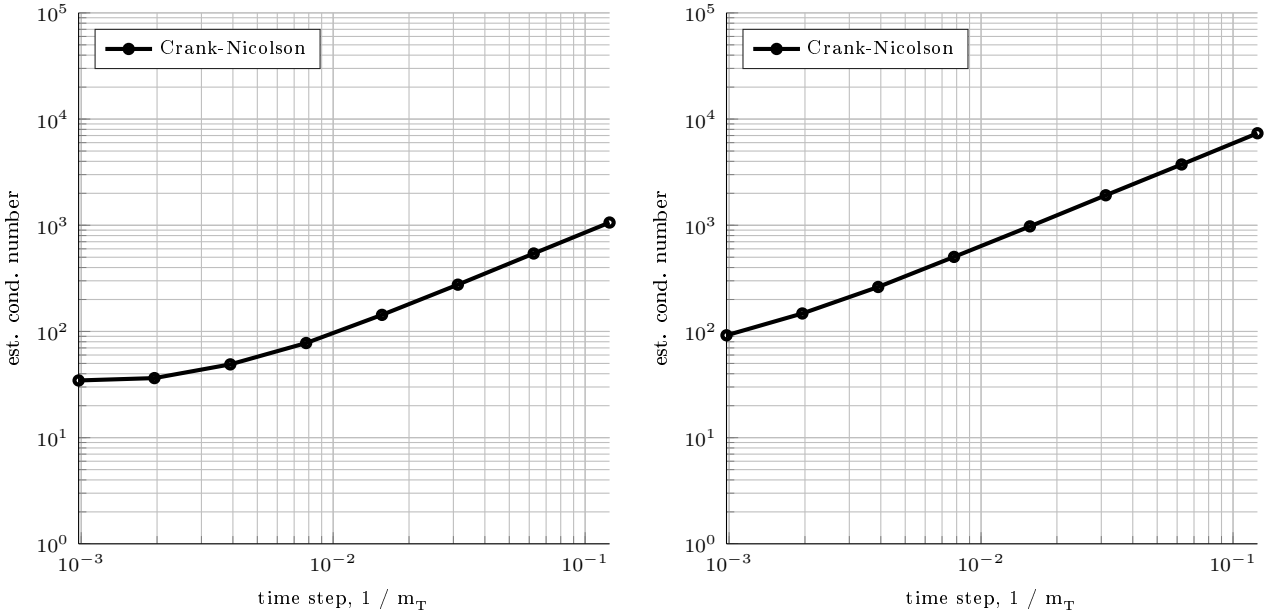
It is seen that the shape of the solution is not changing. The convergence analysis for the time integration is performed with the following set of parameters:

Element order	$p = 6$
Element scale factor (regular mesh)	$n_E = 128$
Element scale factor (reconstructed mesh)	$\hat{n}_E = 128$
Time steps	$m_T = [8,\, 16,\, 32,\, 64,\, 128,\, 256,\, 512,\, 1024]$



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.28: L^2 error analysis



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.29: Estimated condition Number

For both types of meshes the optimal convergence order of $\mathcal{O}(\tau^2)$ can be achieved after the pre-asymptotic range. The condition numbers show the same characteristic and difference is approximately the factor 1×10^1.

5.2.2 Verification in 2d

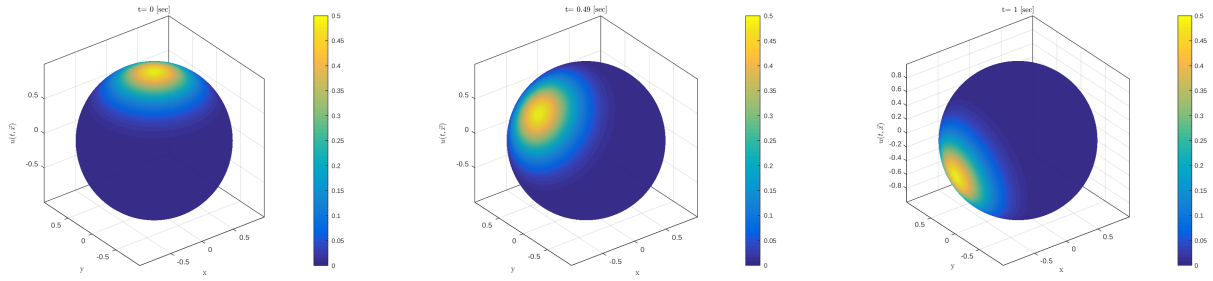
The in-stationary transport problem is solved on a sphere $r = 1$ with the following set of parameters:

$$\begin{aligned} \hat{a} &= -1.75 \qquad \text{(tangential convection in the xz plane)} \\ \lambda &= 0 \\ t_{m_T} &= 1.00\,\mathrm{s} \end{aligned}$$

The initial state is set to

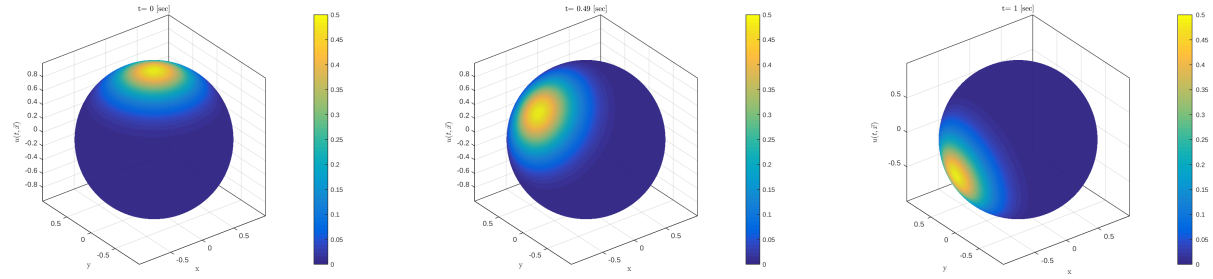
$$u(\boldsymbol{x}, 0) = \exp\left(-4\theta(\boldsymbol{x})^2\right) \ , \qquad \theta \in [0,\, \pi] \ . \tag{5.26}$$

Analogous to the example before the exact solution is obtained as a shift of the θ and the corresponding coordinates. In Fig. 5.30 and Fig. 5.31 the numerical and analytical solution at $t = [0.00,\, 0.49,\, 1.00]\,\mathrm{s}$ is illustrated.



(a) Inital state **(b)** State at $t = 0.49\,\mathrm{s}$ **(c)** State at $t = 1.00\,\mathrm{s}$

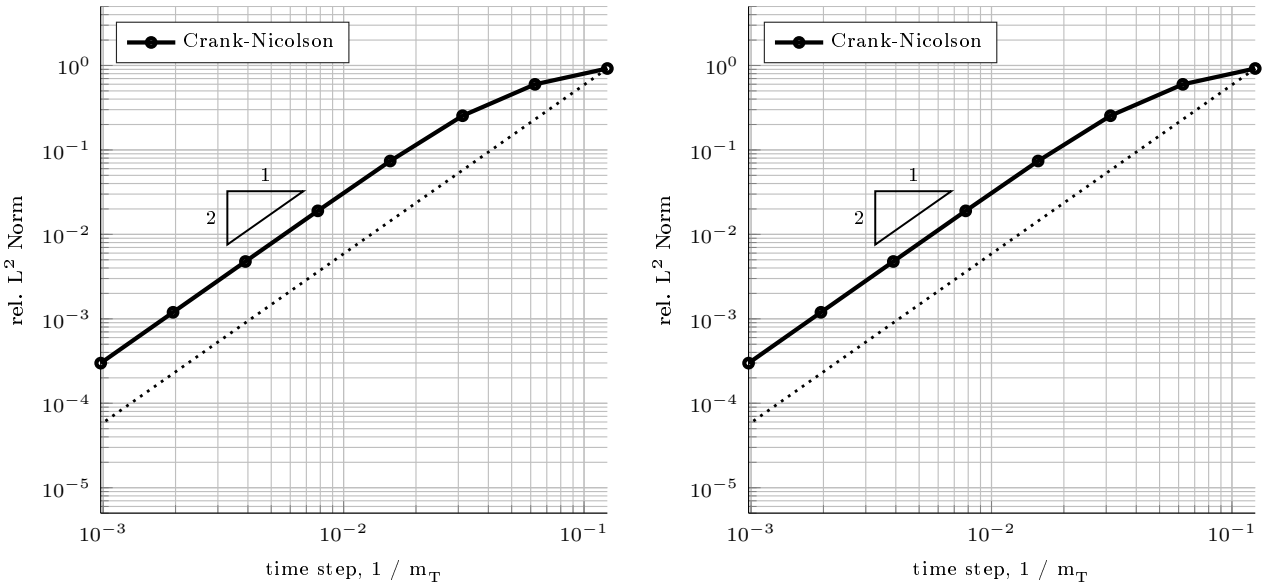
Fig. 5.30: Numerical solution of in-stationary transport problem on a sphere without diffusion



(a) Inital state **(b)** State at $t = 0.49\,\mathrm{s}$ **(c)** State at $t = 1.00\,\mathrm{s}$

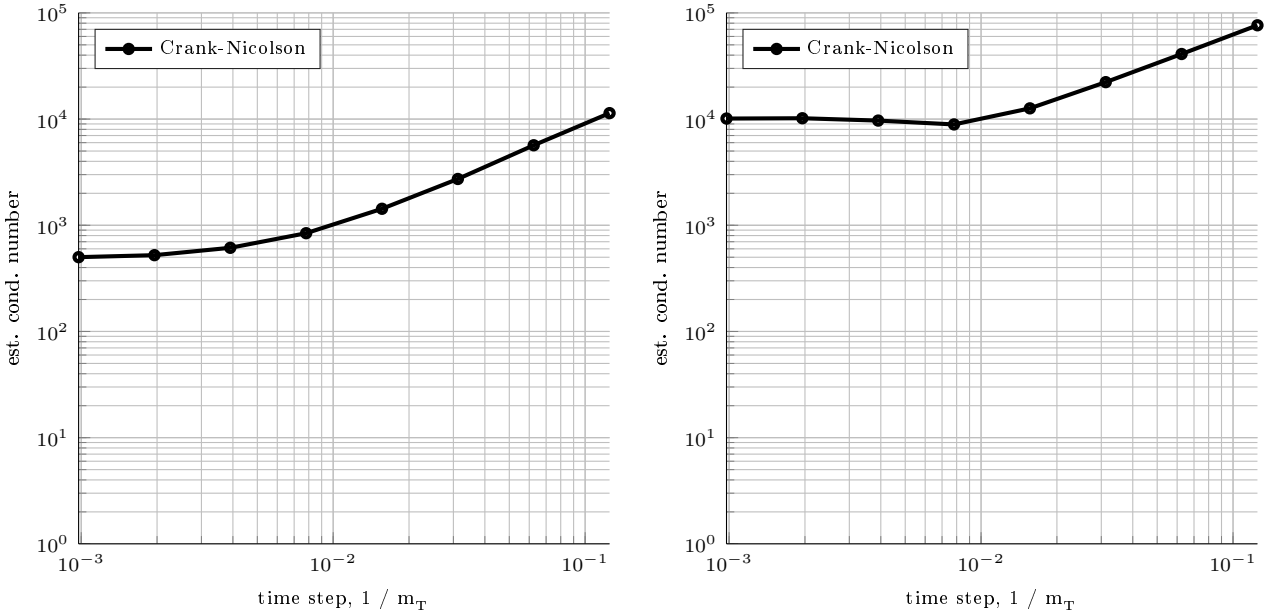
Fig. 5.31: Exact solution of in-stationary transport problem on a sphere without diffusion

The convergence analysis is performed and the results are analogous to the example before. Both types of meshes achieved optimal convergence orders and the condition numbers show the same behavior.



(a) Regular mesh **(b)** Reconstructed mesh

Fig. 5.32: L^2 error analysis



(a) Regular mesh **(b)** Reconstructed mesh

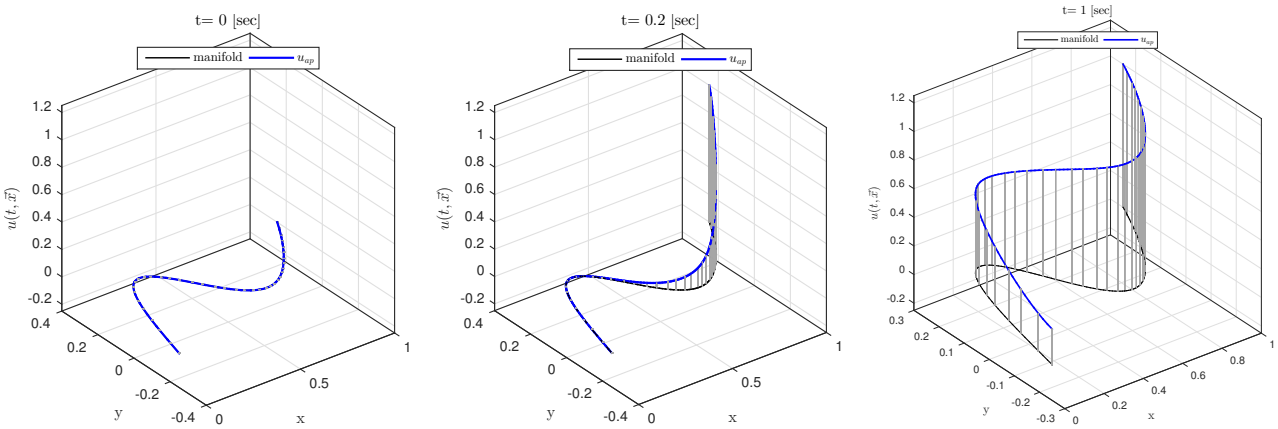
Fig. 5.33: Estimated condition Number

5.2.3 Example 1 - S-shaped manifold

Here, the in-stationary transport problem is solved on a s-shaped manifold. The manifold is defined in Section 5.1.4. The parameters are set to:

$$\begin{aligned} \hat{a} &= 1.00 & \hat{n}_E &= 32 \\ \lambda &= 0.15 & p &= 4 \\ t_{m_T} &= 1.00\,\mathrm{s} & m_T &= 100 \end{aligned}$$

The inital state is set to zero in the whole domain and as Dirichlet-boundary condition the right node is set to 1. In Fig. 5.34 we can see that the distribution is growing from the right boundary to the left boundary. After enough time steps the whole distribution would be one in the whole domain.



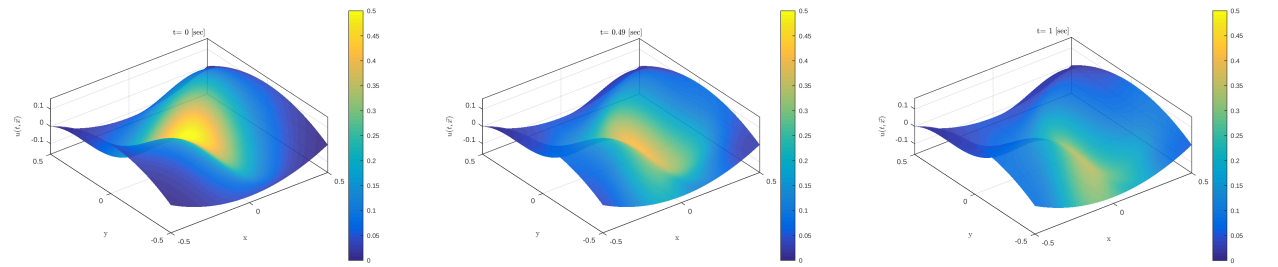
(a) Inital state **(b)** State at $t = 0.20\,\mathrm{s}$ **(c)** State at $t = 1.00\,\mathrm{s}$

Fig. 5.34: Solution of an in-stationary transport problem

5.2.4 Example 2 - Hyperbolic paraboloid with bumps

Here, a initial distribution $u(\boldsymbol{x}, 0)$, which is diffusing and flowing in the global y-direction is assumed. The manifold is defined in Section 5.1.7 and the parameters and initial state are set to:

$$\begin{aligned} \hat{a} &= 1.50 & \hat{n}_E &= 32 \\ \lambda &= 0.01 & p &= 4 \\ t_{m_T} &= 1.00\,\mathrm{s} & m_T &= 100 \\ u(\boldsymbol{x}, 0) &= \frac{1}{2} \exp\left[-10\left(x^2 + y^2\right)\right] . \end{aligned}$$



(a) Inital state **(b)** State at $t = 0.49\,\mathrm{s}$ **(c)** State at $t = 1.00\,\mathrm{s}$

Fig. 5.35: Solution of an in-stationary transport problem

6 Conclusion & Future prospects

With the described meshing strategies, it is possible to generate higher-order meshes automatically. The resulting meshes may feature awkward element size ratios, but with the implemented background mesh manipulation (node movements) this property can be controlled and optimal approximation properties are achieved. To sum up, in order to determine an arbitrary higher-order manifold mesh it is sufficient just to define an implicit level-set function and the corresponding background mesh.

In the context of simulations the convergence rates are optimal and coincide with the theoretical error estimates. When appropriately manipulating the background meshes, the element ratios remain bounded and the condition numbers are comparable to those obtained with classical meshes, however, shifted by a certain factor. An alternative to manipulating the background mesh is to employ stabilization approaches such as in [11], [1]. As mentioned before, the exact description of boundaries of open manifolds is still an issue. A solution for this problem is the definition of additional level-set functions which imply the boundaries.

In the future we will apply the described meshing strategies to a number of further applications in engineering, including the analysis of shells on implicitly defined manifolds.

Bibliography

[1] E. Burman et al. “Full gradient stabilized cut finite element methods for surface partial differential equations”. In: *Computer Methods in Applied Mechanics and Engineering* Vol. 310 (2016), pp. 278–296.

[2] Y. Canzani. *The Laplacian on a Riemannian manifold,* URL *http://www.math.harvard.edu/~canzani/math253/Lecture7.pdf*. Online ressouce, visited on: 25.04.2016.

[3] T.P. Fries and S. Omerovic. “Conformal higher-order remeshing schemes for implicitly defined interface problems”. In: *International Journal for numerical methods in engineering* (2016).

[4] T.P. Fries and S. Omerovic. “Higher order accurate integration of implicit geometries”. In: *International Journal for numerical methods in engineering* Vol. 106 (2016), pp. 323–371.

[5] T.P. Fries et al. *Higher-order meshing of implicit geometries – part I: Integration and interpolation in cut elements*. Graz University of Technology, 2016.

[6] P. Hansbo, M. G. Larson, and F. Larsson. “Tangential differential calculus and the finite element modeling of a large deformation elastic membrane problem”. In: *Computational Mechanics* Vol. 56 (2015), pp. 87–95.

[7] M. Jung and U. Langer. *Methode der finiten Elemente für Ingenieure: Eine Einführung in die numerischen Grundlagen und Computersimulation*. Springer Vieweg, 2013.

[8] H. Kabaria and A. J. Lew. *Universal Meshes: Computing Tetrahedralization Conforming to Curved Surfaces,* URL *http://www.imr.sandia.gov/papers/imr23/IMR23_ResearchNote1_Kabaria.pdf*. Online ressouce, visited on: 03.08.2016.

[9] D. Kamilis. *Numerical methods for the PDES on curves and surfaces*. Umeå university, 2013.

[10] R. Rangarajan and A. J. Lew. “Universal meshes: A method for triangulating planar curved domains immersed in nonconforming meshes”. In: *International Journal for numerical methods in engineering* Vol. 98 (2014), pp. 236–264.

[11] A. Reusken. “Analysis of trace finite element methods for surface partial differential euqations”. In: *Jounral of Numerical Analysis* Vol. 35 (2015), pp. 1568–1590.