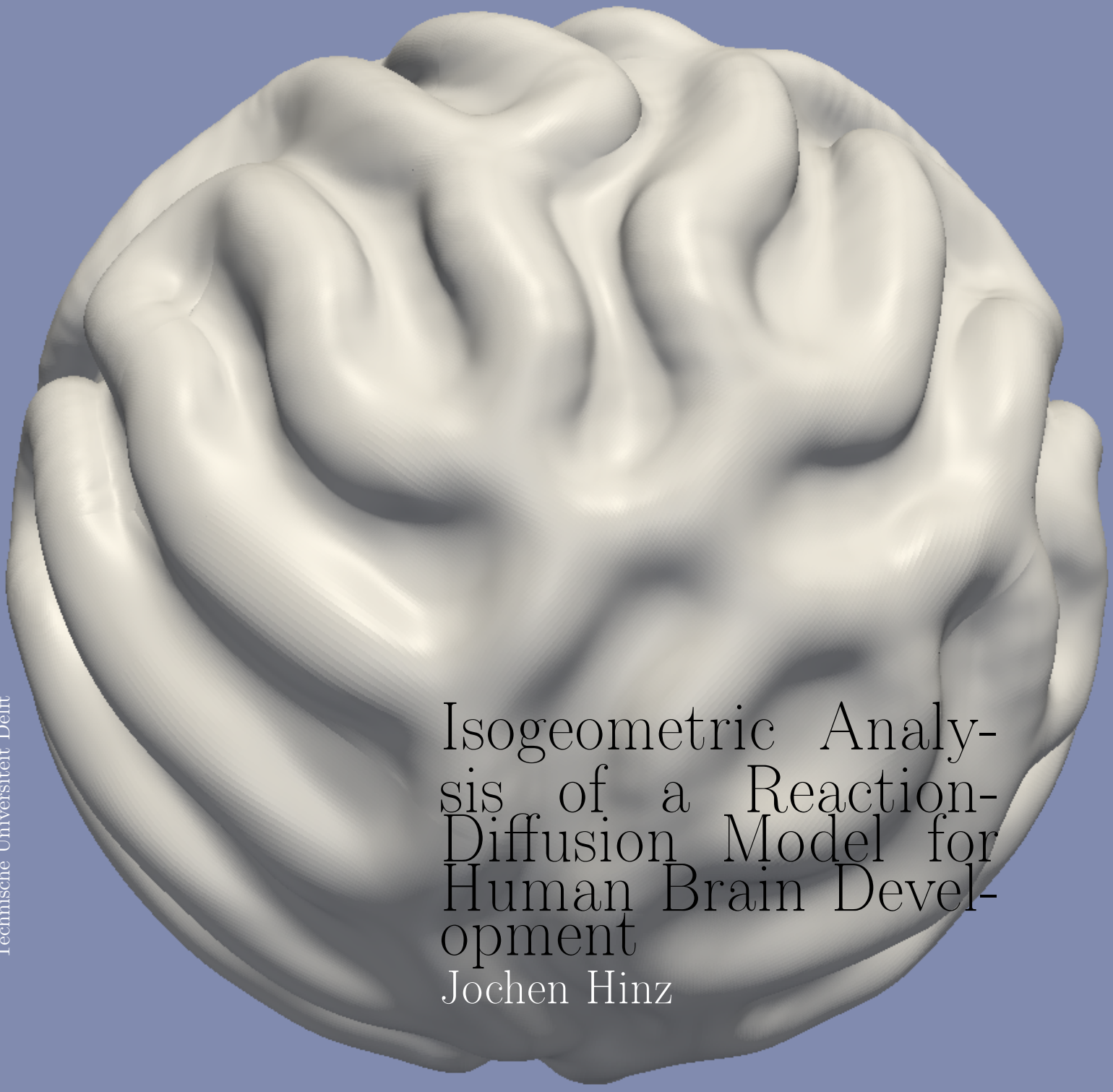


Isogeometric Analysis of a Reaction-Diffusion Model for Human Brain Development

Jochen Hinz

Technische Universiteit Delft

TU Delft

Isogeometric Analysis of a Reaction-Diffusion Model for Human Brain Development

by

Jochen Hinz

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday January 28, 2016 at 02:00 PM.

Student number:	1545337	
Project duration:	March 18, 2015 – January 28, 2016	
Thesis committee:	Dr. ir. F. J. Vermolen,	TU Delft, supervisor
	Dr. M. Möller,	TU Delft, supervisor
	Prof. dr. ir. C. Vuik,	TU Delft
	Dr. ir. W. T. van Horssen,	TU Delft

This thesis is confidential and cannot be made public until January 28, 2016.

TU Delft

Preface

Before you lies the thesis "Isogeometric Analysis of a Reaction-Diffusion Model for Human Brain Development" which has been written to fulfill the graduation requirements of the Applied Mathematics Program at the Delft University of Technology. I was engaged in this project from March 2015 to January 2016.
I chose for a subject related to the finite element method (FEM) since I had already been confronted with topics from numerical linear algebra during my project "The Scattering Algorithm for Sparse Linear Hermitian Systems", which would have been my first choice otherwise.
This thesis is written for people with a background in (applied) mathematics and numerical analysis in particular. I have done my best to translate the basic principles of differential geometry into a formalism that is compatible with the formalism that is common in FEM-related topics.
I would like to thank my thesis supervisors Dr. ir. F.J. Vermolen and Dr. M. Möller for providing me with this exciting and challenging project and for their support and plenty of helpful advice during my time at the department. Furthermore, I would like to thank Joost for his help with the implementation of the numerical solver and for helping me improve my mathematical, as well as my programming skills with his useful tips.
I would like to thank my parents for encouraging me to pursue a double degree despite the additional study costs associated with it.
Last but not least, I would like to thank everyone who takes his time to read this (rather lengthy) thesis.

Jochen Hinz
Delft, January 2016

Contents

1

Introduction

In this chapter we introduce the problem that will be the subject of this thesis. We outline the shortcomings of an existing numerical implementation and how we aim to improve upon it. We end this chapter on introducing the notation that we shall adhere to in the remainder of this thesis.

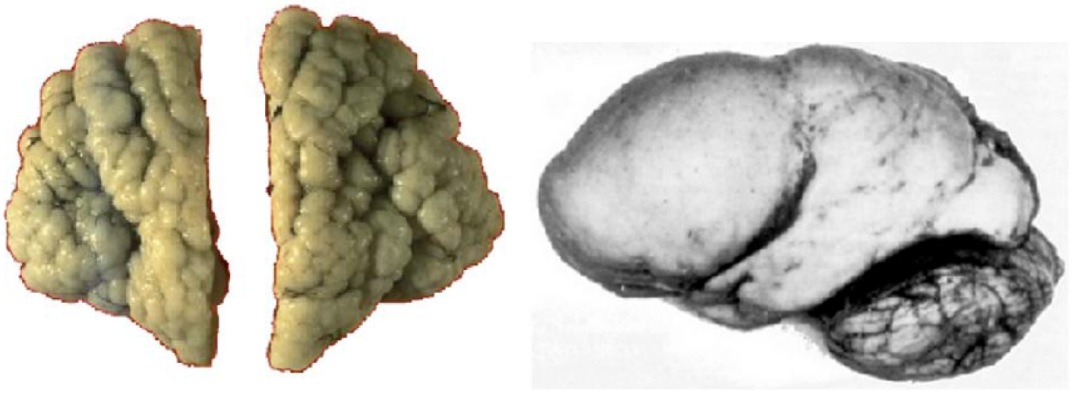
1.1. Problem Description

Neural development has become a topic of growing interest in the past decades. On the one hand healthy adult individuals exhibit qualitatively similar neural structures, on the other hand neural development exhibits a substantial degree of randomness, which is largely confirmed by the observation that even monozygotic twins exhibit significant anatomical differences [7]. Among other factors, this neural 'fingerprint' manifests itself mainly through the patterns formed in the neural folding and buckling process occurring naturally after the twentieth week of fetal development (see figure 1.1).
This suggests that environmental factors can have a profound influence on the course of neural development, which in turn suggests that the underlying biological process, mathematically, exhibits

Figure 1.1: Typical patterns formed at the surface of human brains.

a high degree of sensitivity toward perturbations in the initial condition. On the other hand, a proficient model for human brain development should be capable of producing qualitatively similar outcomes for similar setups and explain neural pathologies like lissencephaly and polymicrogyria (see figure 1.2) by quantitatively different starting conditions. The derivation of proficient models for human brain development is greatly hindered by the unethicalness of experimentation on human fetuses.

For this reason existing models postulate various driving forces behind pattern formation and assess



(a) (b)

Figure 1.2: Polymicrogyria (a) and Lissencephaly (b).

their validity by comparing the results of simulations to existent brains. Furthermore quality is assessed through above criteria. Existing models for brain pattern formation can be split into two broad categories: intrinsic and extrinsic. Intrinsic models postulate a chemical process, whereas extrinsic models a physical process as the driving force behind neural buckling and folding. Researchers in the field aim to identify the most-likely driving force by comparing various models in order to reduce the amount of experimentation needed to a minimum.

The model proposed by Lefèvre et al. [25] shall be the subject of this thesis. It adopts a modified version of the Gray-Scott reaction-diffusion equations as a basic model for pattern formation. The resulting system of equations shall be tackled numerically using the principles of *Isogeometric Analysis* (IgA) [17].

The concentration of one of the two chemical species considered in the model is postulated as growth-activator, leading to deformations in the geometry that resemble typical folding patterns found in human brains. The implicit assumption of the model is that neural growth can, to a good approximation, be considered as taking place only at the surface. The results from the numerical implementation presented in the article exhibit a high degree of qualitative similarities for similar setups as well as quantitative differences resulting from perturbations in the initial condition. The outcomes also show qualitative differences for different reaction rates and the authors were able to reproduce certain characteristics from various brain anomalies by changing the numerical values of the parameters. Since these findings are well-establish by a great amount of simulations in the paper, in this thesis we will only present three simulations with different setups.

1.2. Motivation

The model proposed by the authors of the article results in a complex system of equations that cannot be solved analytically. The main challenge is the fact that the chemical species affect the local topological properties of the geometry which in turn affects the local expressions of the differential operators acting on the concentrations of the chemical species, leading to a highly nonlinear system.

Complexity is further aggravated by the existence of nonlinear reaction terms. The authors present a numerical scheme that utilizes a finite-difference discretization in the temporal component, treating some terms implicitly and others explicitly, as well as a classical finite-element approach in the spatial components. The initial geometry is given by a triangular tessellated spherical shell.
Growth is incorporated by extracting the value $v_{\mathbf{x}}$ of the concentration of one of the chemical species at triangle vertex $\mathbf{x}$ and shifting its position by an amount proportional to $v_{\mathbf{x}}$ in the direction of the normal vector at $\mathbf{x}$. On a tessellated surface, due to the non-smooth transition between triangles, the normal vector does not exist at the triangle vertices. It is not explicitly stated in the article how the normal vector is computed but it is apparent that it has been computed by a weighed average of the normal vectors of the surrounding (planar) triangles. Within an implementation that is mainly focused on minimizing computational costs (in order to allow for a large amount of simulations), such an approach is reasonable. I does, however, lack overall mathematical rigor.
Thus, being able to construct smooth geometries would obviously constitute an improvement of above shortcoming. Furthermore, smoothness in the geometry will most likely result in more appealing outcomes since non-smooth geometries are not realistic from a biological standpoint.
To achieve smoothness, the evident choice is to replace the classical finite-element approach by an approach based on isgoeometric analysis. This approach, apart from some restrictions, allows for the replacement of the piecewise-linear basis functions by smooth B-spline basis functions of arbitrary polynomial order. It is customary to construct approximations of the geometry from a global mapping that uses the same set of (smooth) basis functions that is used to approximate the unknowns on the geometry. The geometry will thus inherit the continuity properties of the basis. Another advantage of smoothness is that it allows for a (non-discrete) measure of curvature, which can subsequently serve as local refinement criterion.
The computational costs are expected to be higher than in the approach proposed in the article. This is why this approach should be considered quality-oriented.

1.3. Notation

We represent vectors and vector-valued functions utilizing bold-faced letters. The n-th entry of vector(-valued function) $\mathbf{r}$ is denoted by r_n or $(\mathbf{r})_n$. Matrices are presented in square brackets. The n-th entry in the m-th column of matrix $[A]$ is denoted by $[A]_{n,m}$.
Let $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)^T$ and $\mathbf{x} = (x_1, \dots, x_m)^T$. We define the vector-by-vector derivative of $\mathbf{x}$ with respect to $\boldsymbol{\xi}$ as follows

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \cdots & \frac{\partial x_1}{\partial \xi_n} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \cdots & \frac{\partial x_2}{\partial \xi_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial \xi_1} & \frac{\partial x_m}{\partial \xi_2} & \cdots & \frac{\partial x_m}{\partial \xi_n} \end{bmatrix}, \tag{1.1}$$

this is also referred to as the *Jacobian matrix* $[J]$ of $\mathbf{x}$.
In this thesis we will frequently work with finite-element bases. Usually the basis shall we denoted by $\Sigma = \{w_1, \dots, w_N\}$. Whenever we refer to span Σ, we mean the set of all possible linear combinations of the $w_i \in \operatorname{span} \Sigma$. Here we will also allow for weights form $\mathbb{R}^n$ with $n \neq 1$. The value of n will always be clear from context.

1.4. Thesis Overview

In chapter 2, we will present a brief introduction the principles of differential calculus on geometric objects. Here, we will discuss the generalizations of frequently used identities to cases in which the

Jacobian matrix (see equation (1.1)) of the mapping between domain and geometry is not square. Furthermore, we will present several measures for the curvature of a parametric surface.
In chapter 3, we will briefly discuss the basics of finite-element analysis (FEA) and in chapter 4 we will extend it with the principles from IgA.
Numerical integration and matrix assembly techniques will be treated in chapter 5 which constitutes the last chapter treating the theoretical aspects of this report.
In chapter 6, we will present the model and corresponding equations as proposed by Lefèvre et al. and in chapter 7 we present a general IgA implementation of this model.
Concrete implementations based on the principles from chapter 7 will be the subject of chapters 8 and 9 which will be extended with refinement strategies in chapter 10. The results of both implementations will be presented in chapter 11.
We will present some numerical experiments testing the numerical scheme from chapter 7 in chapter 12 and in chapter 13, we will present a possible alternative scheme for future implementations of this model.

2

Calculus on Geometric Objects

In this thesis we shall make frequent use of parameterizations. The mapping $\mathbf{s} : \Omega \to \mathscr{M}$, with $\Omega \subset \mathbb{R}^n$ and $\mathscr{M} \subset \mathbb{R}^m$, that parameterizes the geometry $\mathscr{M}$ with points from the parametric domain Ω shall be of major importance. 'Local' functions living on Ω shall generally be represented utilizing lower-case letters, for example $w : \Omega \to \mathbb{R}$. Functions living in parameter space Ω can be made 'global' by utilizing the inverse $\mathbf{s}^{-1} : \mathscr{M} \to \Omega$ of $\mathbf{s}$ and are generally represented by the corresponding upper-case letter, here: $W : \mathscr{M} \to \mathbb{R}$. Their relationship is given by

$$W = w \circ \mathbf{s}^{-1}, \tag{2.1}$$

see figure 2.1.
In practice, we will rarely need W (so we do not have to find $\mathbf{s}^{-1}$) since most calculations are carried out in Ω. Thus, W will only be important when explaining mathematical concepts. Sometimes, we shall refer to W as *'the projection of w onto the geometry $\mathscr{M}$'*.
The mapping $\mathbf{s}$ is presented in bold-faced type since it is, in fact, a vector-valued function with its vector-dimensionality matching that of the target space (here $\mathscr{M} \subset \mathbb{R}^m$, so the dimensionality is m). In general, a vector-valued function whose domain is given by Ω shall receive a bold-faced lower-case letter, whereas a vector-valued function over $\mathscr{M}$ receives a bold-faced upper-case letter.

Remark. *In this thesis it is more natural to adopt the relation*

$$W \circ \mathbf{s} = w, \tag{2.2}$$

as opposed to (2.1). This is especially true whenever $\mathbf{s}$ *fails to be bijective, in which case we require that w assumes the same value on segments of Ω that overlap on $\mathscr{M}$, in order to avoid having to work with open sets (see figure 2.2).*

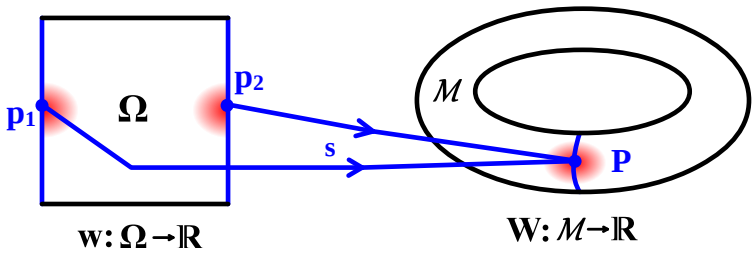


Figure 2.2: The point $P \in \mathscr{M}$ has two points $\{p_1, p_2\} \subset \Omega$ that point to it via the mapping $\mathbf{s}$, such that $\mathbf{s}$ can not be invertible. Periodic boundary conditions have been imposed to ensure that w is single-valued when projected onto $\mathscr{M}$.

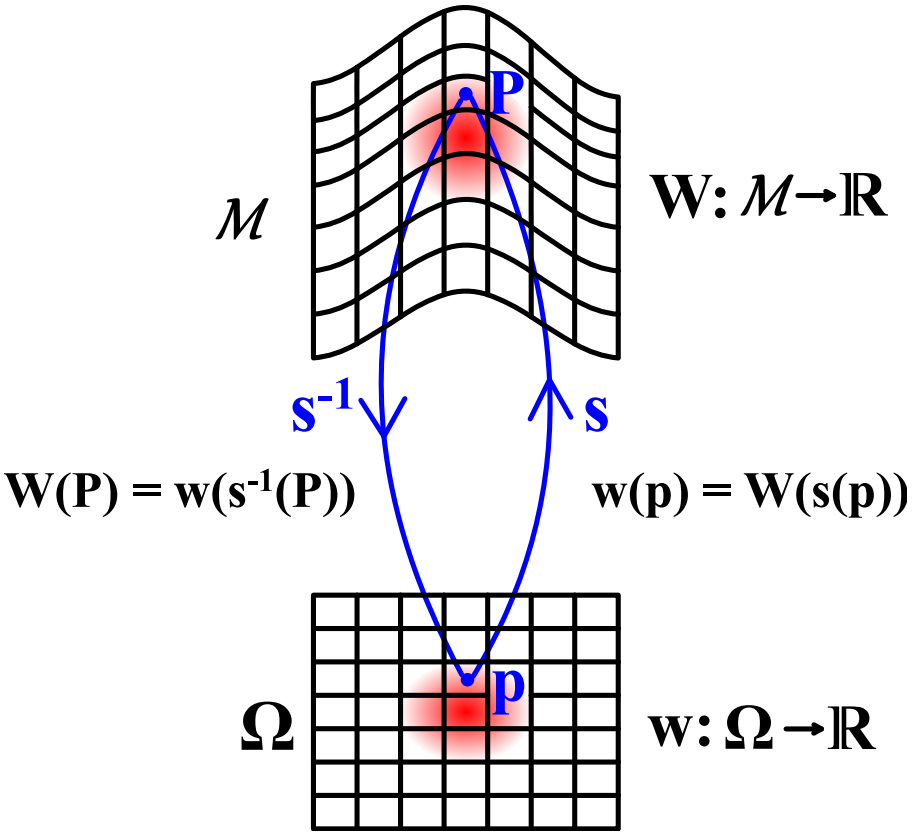


Figure 2.1: Schematic representation of the relation between local and global functions.

2.1. Preliminaries

Let $\mathbf{s}:\Omega\to M$ be a mapping that parameterizes an n-dimensional geometry $\mathscr{M}\subset\mathbb{R}^m, m>n$. Thus, $\Omega\subset\mathbb{R}^n$ and we denote its free variables by $\xi_1,\dots,\xi_n$ such that $(x_1,\dots,x_m)^T=\mathbf{s}(\xi_1,\dots,\xi_n)$.
As a next step we will characterize the *tangent plane* $T_P\mathscr{M}$ of $\mathscr{M}$ at point $P\in\mathscr{M}$ by the vectors that span it. The tangent plane has the property that every curve $\mathscr{C}\subset\mathscr{M}$ that passes through P is tangential to $T_P\mathscr{M}$ at point P. $T_P\mathscr{M}$ is spanned by

$$\left\{\left.\frac{\partial\mathbf{s}}{\partial\xi_1}\right|_p,\dots,\left.\frac{\partial\mathbf{s}}{\partial\xi_n}\right|_p\right\}, \tag{2.3}$$

where $p\in\Omega$ is the point that satisfies $\mathbf{s}(p)=P$. Note that above vectors are exactly the column vectors of the *Jacobian matrix* $[J]$ of $\mathbf{s}$ at point p. Since $\mathbf{x}=\mathbf{s}(\boldsymbol{\xi})$, we have

$$[J]=\frac{\partial\mathbf{x}}{\partial\boldsymbol{\xi}}. \tag{2.4}$$

Remark. *Note that $[J]$ is not square whenever $n\neq m$.*
In the remainder of this chapter, $p\in\Omega$ and $P\in\mathscr{M}$, will always be related by $P=\mathbf{s}(p)$.

The set $\{(P,\mathbf{V}):P\in\mathscr{M},\mathbf{V}\in T_p\mathscr{M}\}$, i.e. the set of all vectors that are tangent to $\mathscr{M}$ along with the information to which point of $\mathscr{M}$ they are tangent, is referred to as the *'tangent bundle'* [23, p. 65].
Let $\mathbf{U}:\mathscr{M}\to T_P\mathscr{M}$, i.e. let $\mathbf{U}$ be a vector-valued function that assumes some $\mathbf{V}\in T_P\mathscr{M}$ at each point

$P \in \mathscr{M}$. As a general rule, the vector-valued function $\mathbf{u} : \Omega \to \mathbb{R}^n$ is related to $\mathbf{U}$ in the following way

$$\mathbf{U} \circ \mathbf{s} = [J]\mathbf{u}. \tag{2.5}$$

Thus, $\mathbf{u}(p)$ represents $\mathbf{U}(P)$ in the canonical basis of $T_P\mathscr{M}$ and is referred to as *the local counterpart of* $\mathbf{U}$ or $\mathbf{U}$ *in local coordinates*. Whenever the domain of a function is $\mathscr{M}$ and its co-domain is $T_P\mathscr{M}$, it will receive a bold-faced upper-case letter and its local counterpart will receive the corresponding lower-case letter. Furthermore, we shall simply utilize $\mathbf{U}$ as opposed to $\mathbf{U} \circ \mathbf{s}$ whenever it is clear from the context that we are evaluating in $\boldsymbol{\xi}$ as opposed to $\mathbf{x}$. This is equivalent to saying

$$\mathbf{U}(\boldsymbol{\xi}) \equiv \mathbf{U} \circ \mathbf{s}. \tag{2.6}$$

With (2.5) in mind, we define the *surface metric* $[g]$ in the following way

$$[g] = [J]^T[J]. \tag{2.7}$$

The terminology 'surface metric' stems from the fact that it induces a canonical inner product for vectors or vector-valued functions in local coordinates. Using (2.5) and (2.6), we have

$$\begin{aligned} \mathbf{A} \cdot \mathbf{B}(\boldsymbol{\xi}) &= \mathbf{a}^T[J]^T[J]\mathbf{b} \\ &= \mathbf{a}^T[g]\mathbf{b} \\ &\equiv \langle \mathbf{a}, \mathbf{b} \rangle_g, \end{aligned} \tag{2.8}$$

where '$\cdot$' stands for the standard inner-product in $\mathbb{R}^m$.

2.2. Integration over Geometric Objects

Since direct integration over complex geometries like $\mathscr{M}$ is generally not possible, the idea is to replace the domain of integration $\mathscr{M}$ by the simpler parametric domain Ω. Since geometries are usually parametrized by relatively simple domains (for instance quadrilaterals in $\mathbb{R}^2$), the limits of integration follow very naturally from the shape of Ω.
An integral of any function $W : \mathscr{M} \to \mathbb{R}$ over the geometry $\mathscr{M}$ can be related to an equivalent integral in parameter space Ω as follows [23, p. 654]

$$\int_{\mathscr{M}} W \mathrm{d}\mathbf{x} = \int_{\Omega} w \sqrt{\det[g]}\, \mathrm{d}\boldsymbol{\xi}. \tag{2.9}$$

The function $\sqrt{\det[g]}$ acts as a scaling factor, relating infinitesimal (hyper-)surface elements on $\mathscr{M}$ to those from Ω. It is referred to as the *Riemannian volume form* [23, p. 389]. For the sake of brevity, we define $\sqrt{\det[g]} \equiv \sqrt{g}$ and $\mathrm{d}\Omega \equiv \sqrt{g}\mathrm{d}\boldsymbol{\xi}$. In this context, $\mathrm{d}\Omega$ represents the infinitesimal surface element on Ω in the presence of scaling and $\mathrm{d}\boldsymbol{\xi}$ in the absence of scaling. It is seen that the change of domain is accompanied by a change of integrand. As a result of the scaling $\sqrt{g}$, the complexity of the integrand usually increases which is why one might favor an approximation with an appropriate quadrature-scheme over a symbolic approach (see chapter 5).

2.3. Differential Operators on Geometric Objects

Having defined the notion of *tangent plane* and the relation between local and global vector-valued functions, we can proceed to defining the geometrical counterparts of the gradient, divergence and Laplace operators.

Let $W : \mathbb{R}^m \to \mathbb{R}$, we define the *directional derivative* of W in the direction of $\mathbf{V}$ (where $\|\mathbf{V}\| = 1$), denoted by $\partial_{\mathbf{V}} W$, as follows

$$\partial_{\mathbf{V}} W(P) = \frac{\mathrm{d}}{\mathrm{d}t} W(P + t\mathbf{V})\bigg|_{t=0}. \tag{2.10}$$

With the chain rule, it evaluates to

$$\partial_{\mathbf{V}} W = \nabla W \cdot \mathbf{V}. \tag{2.11}$$

The definition from (2.10) is equivalent to [30, p. 190]

$$\partial_{\mathbf{V}} W(P) = \frac{\mathrm{d}}{\mathrm{d}t} W(\gamma(t))\bigg|_{t=0}, \tag{2.12}$$

where the parameterization $\gamma(t)$, $t \in [a,b]$ of the curve $\mathscr{C}$ satisfies

$$\gamma(0) = P \text{ and } \frac{\mathrm{d}}{\mathrm{d}t}\gamma(t)\bigg|_{t=0} = \mathbf{V}. \tag{2.13}$$

Equation (2.12) can be utilized for functions on $\mathscr{M}$ by requiring $\mathscr{C} \subset \mathscr{M}$, since then $W(\gamma(t))$ is defined for all t. Note that $\mathbf{V} \in T_P\mathscr{M}$. In light of the ordinary gradient ∇W producing a vector that points in the direction of the steepest ascent in $\mathbb{R}^m$, it is obvious that its geometry-counterpart $\nabla_{\mathscr{M}} W$ should point into the direction of the steepest ascent on $\mathscr{M}$. Therefore, it obviously may not possess a component in the direction orthogonal to the geometry. Thus, for each $W : \mathscr{M} \to \mathbb{R}$, we have $\nabla_{\mathscr{M}} W(P) \in T_P\mathscr{M}$.
Since ∇W satisfies

$$\nabla W(P) \cdot \mathbf{V} = \partial_{\mathbf{V}} W(P), \quad \forall \quad \mathbf{V} \in \mathbb{R}^m, \tag{2.14}$$

the natural definition of $\nabla_{\mathscr{M}} W$, for functions $W : \mathscr{M} \to \mathbb{R}$, is as follows

$$\nabla_{\mathscr{M}} W(P) \cdot \mathbf{V} = \partial_{\mathbf{V}} W(P), \quad \forall \quad \mathbf{V} \in T_P\mathscr{M}. \tag{2.15}$$

This definition can be translated to local coordinates

$$\langle \nabla_{\mathscr{M}} w(p), \mathbf{v} \rangle_{g(p)} = \partial_{\mathbf{V}} W(P), \quad \forall \quad \mathbf{V} \in T_P\mathscr{M}, \tag{2.16}$$

where $[J(p)]\mathbf{v} = \mathbf{V}$. The function $\nabla_{\mathscr{M}} W$ that satisfies (2.15) is, in local coordinates, given by [18, p.62]

$$\nabla_{\mathscr{M}} w = [g]^{-1} \hat{\nabla} w, \tag{2.17}$$

where $\hat{\nabla}$ denotes the ordinary nabla operator in local coordinates.
Its global counterpart thus satisfies

$$\nabla_{\mathscr{M}} W(\boldsymbol{\xi}) = [J] \nabla_{\mathscr{M}} w. \tag{2.18}$$

In the remainder, we shall replace $\nabla_{\mathscr{M}} \to \nabla$ for convenience. In most textbooks, ∇W is referred to as *the surface gradient.*

Remark. *Just like the ordinary gradient, $\nabla W : \mathscr{M} \to T_P\mathscr{M}$ points in the direction of steepest increase of W on $\mathscr{M}$. This is easily seen from (2.15) by noting that $\partial_{\mathbf{V}} W$, with $\|\mathbf{V}\| = 1$, is maximized over $\mathbf{V} \in T_P\mathscr{M}$ whenever $\mathbf{V}$ points into the same direction as ∇W.*

We can utilize the above definition of the surface gradient to find an equivalent expression for the geometric divergence. In $\mathbb{R}^n$, for smooth compactly supported functions W and vector-valued functions $\mathbf{U}$, the divergence can be regarded as the negative adjoint operator of the gradient, i.e.

$$\int_{\mathbb{R}^n} W \nabla \cdot \mathbf{U} \mathrm{d}\mathbf{x} = -\int_{\mathbb{R}^n} \nabla W \cdot \mathbf{U} \mathrm{d}\mathbf{x}. \tag{2.19}$$

Let $\mathbf{U} : \mathscr{M} \to T_P\mathscr{M}$ and $W : \mathscr{M} \to \mathbb{R}$, analogous to the standard divergence, the geometric divergence is defined as the negative adjoint of the surface gradient. This translates to local coordinates as follows

$$\int_\Omega \langle w, \nabla \cdot \mathbf{u} \rangle_g \sqrt{g} \mathrm{d}\boldsymbol{\xi} = -\int_{\mathscr{M}} \langle \nabla w, \mathbf{u} \rangle_g \sqrt{g} \mathrm{d}\boldsymbol{\xi}, \tag{2.20}$$

for all functions W that vanish on $\partial\mathscr{M}$. The function that satisfies (2.20) is, in local coordinates, given by [34, p. 18]

$$\nabla \cdot \mathbf{u} = \frac{1}{\sqrt{g}} \sum_{i=1}^{n} \frac{\partial}{\partial \xi_i} \left(\sqrt{g} u_i \right). \tag{2.21}$$

The global counterpart of $\nabla \cdot \mathbf{u}$ satisfies

$$\nabla \cdot \mathbf{U}(\boldsymbol{\xi}) = \nabla \cdot \mathbf{u}. \tag{2.22}$$

The Laplace-Beltrami operator, being the counterpart of the ordinary Laplace-operator, consequently satisfies $\Delta w \equiv \nabla \cdot \nabla w$. After some rearrangement, we find [20]

$$\Delta w = \frac{1}{\sqrt{g}} \sum_{i,j=1}^{n} \frac{\partial}{\partial \xi_i} \left(\sqrt{g} g^{i,j} \frac{\partial}{\partial \xi_j} w \right), \tag{2.23}$$

where the $g^{i,j}$ are the entries of $[g]^{-1}$. As before, its global counterpart satisfies

$$\Delta W(\boldsymbol{\xi}) = \Delta w. \tag{2.24}$$

Remark. *Above expressions for the standard differential operators on geometries can exclusively be computed in Ω, not on $\mathscr{M}$ directly. This appears like a shortcoming since mapping the expressions in local coordinates onto their corresponding global counterparts is a laborious task. In practice, however, this mapping is never explicitly carried out: all computations are carried out in Ω, such that the global expressions only possess a conceptual relevance.*

In (2.3), the definition of the surface gradient, we have related the local and global expression via the Jacobian matrix $[J]$. As a general rule, a pair of local and global vector-valued functions carrying the same letter are related by

$$\mathbf{U}(\boldsymbol{\xi}) = [J]\mathbf{u}, \tag{2.25}$$

whenever $\mathbf{U} : \mathscr{M} \to T_P\mathscr{M}$. If the target space of $\mathbf{U}$ is not $T_P\mathscr{M}$ (but, for instance $\mathbb{R}^m$), we adopt the same relation as defined for scalar functions

$$\mathbf{U}(\boldsymbol{\xi}) = \mathbf{u}. \tag{2.26}$$

Finally, we can state the following two lemmas.

Lemma 1. *The divergence theorem holds on $\mathscr{M}$. Let $\mathbf{U} : \mathscr{M} \to T_P\mathscr{M}$ be a sufficiently smooth function, then we have*

$$\int_{\mathscr{M}} \nabla \cdot \mathbf{U} \mathrm{d}\mathbf{x} = \int_{\partial\mathscr{M}} \mathbf{U} \cdot \mathbf{N}_{\partial\mathscr{M}} \mathrm{d}l, \tag{2.27}$$

where $\mathbf{N}_{\partial\mathscr{M}} : \mathscr{M} \to T_P\mathscr{M}$ is the unit outward normal along $\partial\mathscr{M}$ (see figure 2.3) and $\mathrm{d}l$ is the infinitesimal (hyper-) length element along $\partial\mathscr{M}$.

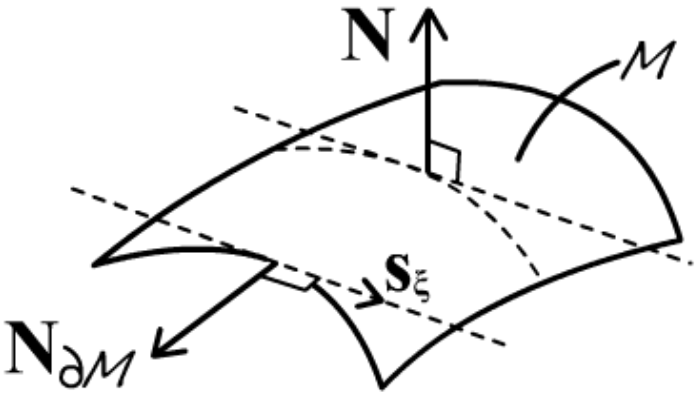


Figure 2.3: The difference between $\mathbf{N}$ and $\mathbf{N}_{\partial\mathscr{M}}$. In this case it is normal to the plane spanned by $\frac{\partial \mathbf{s}}{\partial \xi}$ and $\mathbf{N}$.

Proof. See [23, p. 424]. □

Note that the right-hand side of equation (2.27) vanishes whenever $\mathscr{M}$ has no boundary (for example when $\mathscr{M}$ is a spherical shell in $\mathbb{R}^3$). With lemma 1 in mind, we can state the following

Lemma 2. *On geometries without boundary, we have*

$$\int_{\mathscr{M}} W \Delta U \mathrm{d}\mathbf{x} = -\int_{\mathscr{M}} \nabla W \cdot \nabla U \mathrm{d}\mathbf{x}, \tag{2.28}$$

or in local coordinates

$$\int_{\Omega} w \Delta u \sqrt{g} \mathrm{d}\boldsymbol{\xi} = -\int_{\Omega} \langle \nabla w, \nabla u \rangle_g \sqrt{g} \mathrm{d}\boldsymbol{\xi}. \tag{2.29}$$

Proof. See lemma A.1.2 in the appendix. □

2.4. Curvature of Parametric Surfaces

In this section, we will introduce several measures for the curvature of a parametric surface. To this end, we first define the notions of *'first and second fundamental form'* and *'shape operator'*.
Let $\mathbf{s} : \Omega \to \mathscr{M}$, with $\Omega \subset \mathbb{R}^2$, $\mathscr{M} \subset \mathbb{R}^3$ and $\mathbf{x} = \mathbf{s}(\xi, \eta)$. With (2.7) in mind, we have

$$[g] = \begin{bmatrix} \frac{\partial \mathbf{s}}{\partial \xi} \cdot \frac{\partial \mathbf{s}}{\partial \xi} & \frac{\partial \mathbf{s}}{\partial \xi} \cdot \frac{\partial \mathbf{s}}{\partial \eta} \\ \frac{\partial \mathbf{s}}{\partial \eta} \cdot \frac{\partial \mathbf{s}}{\partial \xi} & \frac{\partial \mathbf{s}}{\partial \eta} \cdot \frac{\partial \mathbf{s}}{\partial \eta} \end{bmatrix}. \tag{2.30}$$

In some textbooks, the metric $[g]$ is referred to as *'the first fundamental form'*. Analogously, the *'second fundamental form'* is given by [26, p. 109]

$$[L] = \begin{bmatrix} \frac{\partial^2 \mathbf{s}}{\partial \xi^2} \cdot \mathbf{n} & \frac{\partial^2 \mathbf{s}}{\partial \xi \partial \eta} \cdot \mathbf{n} \\ \frac{\partial^2 \mathbf{s}}{\partial \xi \partial \eta} \cdot \mathbf{n} & \frac{\partial^2 \mathbf{s}}{\partial \eta^2} \cdot \mathbf{n} \end{bmatrix}, \tag{2.31}$$

where

$$\mathbf{n} = \frac{1}{\left\| \frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right\|} \left(\frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right), \tag{2.32}$$

for positively oriented geometries. Finally, the *'shape operator'* is defined as

$$[S] = [g]^{-1}[L]. \tag{2.33}$$

In light of (2.12), the directional derivative of the unit outward normal $\mathbf{N} : \mathscr{M} \to \mathbb{R}^3$ of $\mathscr{M}$ in the direction $\mathbf{T} \in T_P\mathscr{M}$, with $\|\mathbf{T}\| = 1$ at P is given by

$$\partial_{\mathbf{T}}\mathbf{N}(P) = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}(\gamma(t))\bigg|_{t=0}, \tag{2.34}$$

where γ satisfies

$$\gamma(0) = P \text{ and } \frac{\mathrm{d}}{\mathrm{d}t}\gamma(t)\bigg|_{t=0} = \mathbf{T}. \tag{2.35}$$

Note that $\partial_{\mathbf{T}}\mathbf{N}$ is itself a vector-valued function.
Let us define $\mathrm{d}\mathbf{N} : T_P\mathscr{M} \times \mathscr{M} \to \mathbb{R}^3$, via

$$\mathrm{d}\mathbf{N}(\mathbf{T}, P) = \partial_{\mathbf{T}}\mathbf{N}(P) \tag{2.36}$$

It can be shown that [30, p. 193] $\mathrm{d}\mathbf{N}(\mathbf{T}, P)$ satisfies $\mathrm{d}\mathbf{N}(\mathbf{T}, P) \in T_P\mathscr{M}$. Thus, let

$$\mathbf{T} = [J(p)]\mathbf{t}, \tag{2.37}$$

$\mathrm{d}\mathbf{N}$ is related to (2.33) as follows [43]

$$\mathrm{d}\mathbf{N}(\mathbf{T}, P) = -[J(p)][S(p)]\mathbf{t}. \tag{2.38}$$

Hence, the local counterpart $\mathrm{d}\mathbf{n}(\mathbf{t}, p)$ of $\mathrm{d}\mathbf{N}(\mathbf{T}, P)$ satisfies

$$\mathrm{d}\mathbf{n}(\mathbf{t}, p) = -[S(p)]\mathbf{t}. \tag{2.39}$$

It is seen that the shape operator $[S]$ at p represents the operator $\mathrm{d}\mathbf{N}(\mathbf{T}, P)$ in the canonical basis

$$\left\{ \frac{\partial \mathbf{s}}{\partial \xi}\bigg|_p, \frac{\partial \mathbf{s}}{\partial \eta}\bigg|_p \right\} \tag{2.40}$$

of the tangent bundle at point P (i.e. in local coordinates).

2.4.1. Principal Curvature

The *principal curvatures* at point P, denoted by $\kappa_1(P)$ and $\kappa_2(P)$, are defined as the eigenvalues of $[S(p)]$ [30, p. 200]. For an example of the idea behind principal curvature, see figure 2.4.

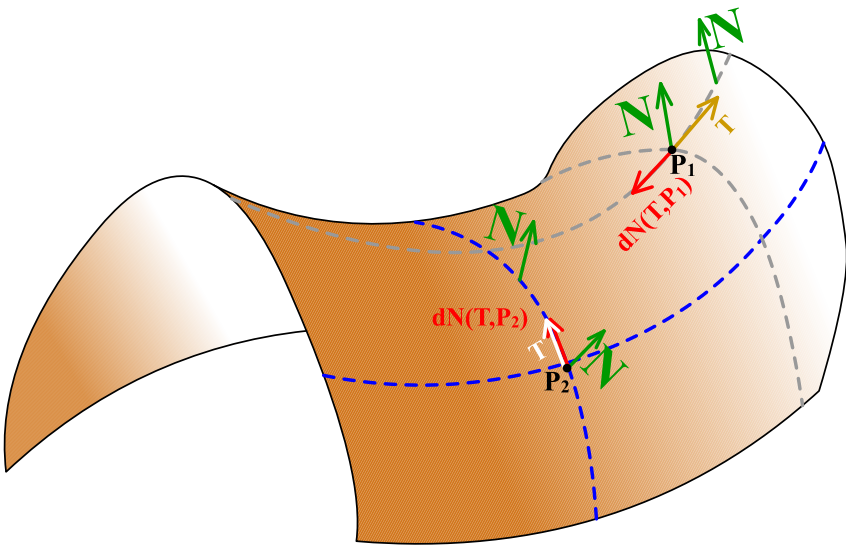


Figure 2.4: Schematic explanation of the idea behind principal curvature. At P_1, $\mathrm{d}N(\mathbf{T}, P_1)$ points in the opposite direction of $\mathbf{T}$ and the surface bends toward the normal, which makes the principal curvature corresponding to $\mathbf{T}$ positive. At P_2, $\mathrm{d}N(\mathbf{T}, P_2)$ points in the direction of $\mathbf{T}$ and the surface bends away from the normal which makes the principal curvature negative.

The *principal directions*, in local coordinates, are given by the corresponding eigenvectors of $[S]$ and can be made global by multiplication with $[J]$.

Loosely speaking, the principal curvatures at P are the reciprocals of the radii of the two circles that pass through point P and are tangent to one of the principal directions (see figure 2.5). They correspond to the minimum and maximum radii of all possible tangent circles trough P. The sign in front of κ_i is positive whenever the surface curves toward the normal vector and negative else. For parametric surfaces, they can be utilized as a measure for curvature. The question is under which conditions the shape operator is diagonalizable (and hence two principal curvatures with corresponding linearly independent principal directions exist at $P \in \mathscr{M}$). To this end, let us state the following proposition

Proposition 1. *Let* $[A]$ *be symmetric and let* $[B]$ *be symmetric positive definite (SPD). Then* $[B][A]$ *is diagonalizable.*

Proof. See proposition A.1.1 in the appendix. □

From proposition (1), we may conclude that the shape operator is guaranteed to be diagonalizable whenever $[g]^{-1}$ is SPD. Since $[g]$ is SPD whenever $\sqrt{g} \neq 0$ ($[g]$ being an inner-product matrix), we may conclude the same for $[g]^{-1}$. It follows that the shape operator is diagonalizable at points on the geometry where $[g]$ is non-singular.

2.4.2. Mean Curvature

In the previous subsection, we have introduced the principal curvatures κ_1 and κ_2 at point $P \in \mathscr{M}$ as the eigenvalues of the shape operator $[S]$ at point $p \in \Omega$. The *mean curvature* κ_m is simply given by

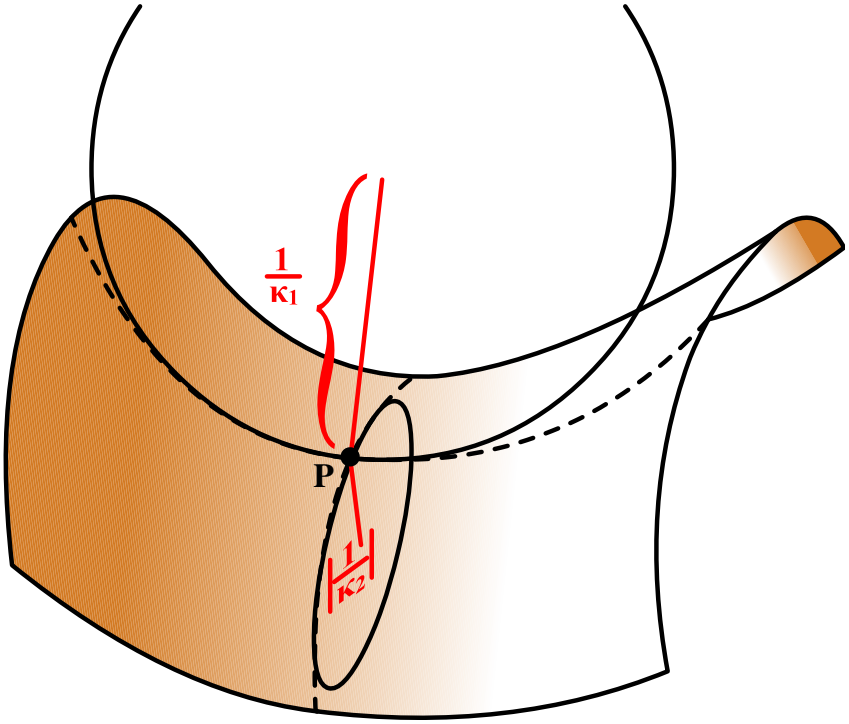


Figure 2.5: The principal curvatures at point P are the reciprocals of the minimum and maximum radii of all possible osculating circles through P.

the average of the two principal curvatures

$$\kappa_m(P) = \frac{1}{2}\left(\kappa_1(P) + \kappa_2(P)\right). \tag{2.41}$$

As the trace of a diagonalizable matrix is equal to the sum of its eigenvalues (including multiplicities), we have

$$\kappa_m = \frac{1}{2}\mathrm{Tr}[S]. \tag{2.42}$$

Note that κ_m can be zero even though the two principal curvatures are nonzero. This happens whenever $\kappa_1 = -\kappa_2$, thus, saddle points can potentially constitute points with nonzero principal curvatures but zero mean curvature.

2.4.3. Gaussian Curvature

The Gaussian curvature κ_g is defined as follows

$$\kappa_g = \kappa_1\kappa_2. \tag{2.43}$$

As the determinant of a matrix is equal to the product of its eigenvalues, assuming that $[g]$ contains no singularities, we may equivalently write

$$\kappa_g = \det[S] = \frac{\det[L]}{\det[g]}.$$

Note that κ_g is zero whenever one or both of the κ_i is zero in a point.

2.4.4. Relating Principal Curvature to Gaussian and Mean Curvature

With the expressions of κ_g and κ_m in mind, we can relate them to the principal curvatures as follows

$$\begin{aligned} \kappa_1 &= \kappa_m + \sqrt{\kappa_m^2 - \kappa_g} \\ \kappa_2 &= \kappa_m - \sqrt{\kappa_m^2 - \kappa_g}. \end{aligned} \tag{2.44}$$

These relations are a useful tool whenever κ_1 and κ_2 are of relevance, since the computations of matrix trace and determinant are relatively cheap operations. In chapter 10, we will make use of the various measures for curvature as a basis for refinement.

3

Finite-Element Analysis

In this chapter, we will briefly discuss the basics of finite element analysis (FEA).

3.1. General Idea

The finite element method (FEM) is a proficient tool to approximate solutions to partial differential equations (PDEs). Given an equation of the form

$$\text{find } U : \mathcal{M} \to \mathbb{R} \quad \text{s.t.} \quad L(U) = F \quad \text{on } \mathcal{M}, \text{ subject to boundary conditions (BCs)} \tag{3.1}$$

with some differential operator L and geometry $\mathcal{M}$, the first step is to multiply (3.1) by a test function W and integrate over $\mathcal{M}$ to derive the *weak form*

$$\text{find } U \in \rho \quad \text{s.t.} \quad \int_{\mathcal{M}} W L(U) \mathrm{d}\mathbf{x} = \int_{\mathcal{M}} W F \mathrm{d}\mathbf{x}, \quad \forall W \in \sigma \tag{3.2}$$

or more compactly

$$\text{find } U \in \rho \quad \text{s.t.} \quad (W, U)_L = (W, F), \quad \forall W \in \sigma. \tag{3.3}$$

Here ρ is called the *trial space* and σ the *test space*. They usually follow from the continuity properties that (3.2) imposes on W and U and depend on the application.
Usually some sort of partial integration is performed on (3.2) to equalize (and minimize) the order of the differential operators applied to W and U and to implement natural boundary conditions. The discretization is carried out by introducing a finite basis $\Sigma = \{W_1, \ldots, W_N\}$ and replacing both ρ and σ by $\operatorname{span}\Sigma$, whereby essential BCs are built into test and trial spaces.
The discretized counterpart of (3.3) is as follows

$$\text{find } U \in \operatorname{span}\Sigma \quad \text{s.t.} \quad (W_i, U)_L = (W_i, F), \quad \forall W_i \in \Sigma, \tag{3.4}$$

assuming that $(\cdot,\cdot)_L$ is still linear in its first argument after partial integration.
By substituting

$$U = \sum_i c_i W_i \tag{3.5}$$

into (3.4), one can derive a system of equations for the c_i. Let us define

$$R(W) = (W, U)_L - (W, F). \tag{3.6}$$

It is seen that U satisfies

$$\forall i \in \{1,\dots,N\}: \quad R(W_i) = 0, \tag{3.7}$$

which is weaker than

$$R(W) = 0, \quad \forall W \in \rho. \tag{3.8}$$

3.2. Elements and FEA Bases

So far, we have not talked about the choice of Σ. Before we proceed to this topic, we will talk about elements and tesselation.
The geometry $\mathscr{M}$ on which the differential equation takes place does not necessarily have to be planar but can be curved. In that case it is customary to build and approximation $\mathscr{M}^*$ of $\mathscr{M}$ that is comprised of a finite set of triangles, referred to as triangular tesselation (see figure *[reference forthcoming]*). The triangle vertices $\{P_1,\dots,P_N\} \subset \mathscr{M}^*$ that result from this tesselation are, in a typical FEM setting, utilized to construct a piecewise polynomial basis $\Sigma = \{W_1,\dots,W_N\}$ that satisfies

$$W_i(P_j) = \delta_{i,j}. \tag{3.9}$$

The most straight-forward tesselation technique is to select a set of points $\{P_1,\dots,P_N\}$ from $\mathscr{M}$ itself and interpolate linearly between neighbouring points. The triangles $\mathscr{A} = \{\epsilon_1,\dots,\epsilon_m\}$ that emerge this way are referred to as *elements*. Since their amount is finite, as well as the cardinality of Σ, the terminology *finite element method* follows. A typical feature of FEM bases, apart from (3.9), is their C^0-continuity across certain element boundaries, regardless of their polynomial order. This is a direct consequence of the fact [40] that higher order FEM basis functions are usually constructed by combining several Lagrangian polynomials into one function with compact support (to acquire a banded system matrix). Integration of basis functions (or their derivatives) is then carried out element-wise. For example, consider the geometrical mass matrix

$$[A]_{i,j} = \int_{\mathscr{M}^*} W_i W_j \mathrm{d}\mathbf{x}. \tag{3.10}$$

Let $\sigma_{i,j} = \left\{\epsilon_k \in \mathscr{A} \mid \epsilon_k \subset \left(\operatorname{supp} W_i \cap \operatorname{supp} W_j\right)\right\}$, then

$$[A]_{i,j} = \sum_{\epsilon_k \in \sigma_{i,j}} \int_{\epsilon_k} W_i W_j \mathrm{d}\mathbf{x}. \tag{3.11}$$

Each triangle ϵ_k is parameterized by a reference triangle and a local mapping $\mathbf{s}_i$ that induces a canonical metric in each of the integrals of (3.11) (see chapter 2).

Remark. *It is possible, though less common, to tesselate using curved triangles as long as they all can be parametrized from a reference triangle.*

3.3. Properties

Whenever L is an *inner product operator*, there exist some easy to derive properties of the approximate solution U over $\operatorname{span}\Sigma$. As before, we define

$$(U,V)_L \equiv \int_{\mathscr{M}} UL(V)\mathrm{d}\mathbf{x}. \tag{3.12}$$

Lemma 3. *Let $(\cdot,\cdot)_L$ be an inner product, i.e. let L satisfy*

$$\begin{aligned}(U,V)_L &= (V,U)_L\\ (\alpha U,V)_L &= \alpha(U,V)_L\\ (U+V,W)_L &= (U,W)_L+(V,W)_L\\ (U,U)_L \geq 0 \text{ and } (U,U)_L = 0 &\implies U=0.\end{aligned} \tag{3.13}$$

Furthermore, let $\Sigma = \{W_1,\dots,W_N\}$ be a (linearly independent) basis over $\mathscr{M}$. Then the matrix

$$[M] = \begin{bmatrix}(W_1,W_1)_L & \dots & (W_1,W_N)_L\\ \vdots & & \vdots\\ (W_N,W_1)_L & \dots & (W_N,W_N)_L\end{bmatrix} \tag{3.14}$$

is symmetric positive-definite (SPD), and thus non-singular.

Proof. See lemma A.2.1 in the appendix. □

Lemma 4. *Let $(\cdot,\cdot)_L$ be an inner product and let U^* be the exact solution of $L(U) = F$ on $\mathscr{M}$. Furthermore let $\Sigma = \{W_1,\dots,W_N\}$ be a (linearly independent) basis over $\mathscr{M}$. If $\mathbf{c}$ satisfies*

$$[M]\mathbf{c} = \mathbf{F} \tag{3.15}$$

with $[M]$ as in (3.14) and

$$F_i = \int_{\mathscr{M}} W_i F \mathrm{d}\mathbf{x}, \tag{3.16}$$

then $U = \sum_j c_j W_j$ minimizes $\|U^* - U\|_L$, where

$$\|U\|_L^2 = \left(U^*-U, U^*-U\right)_L \tag{3.17}$$

over $\operatorname{span}\Sigma$.

Proof. See [10]. □

Whenever we approximate a known function F by some $U \in \operatorname{span}\Sigma$ with a finite-element approach, we are confronted with a problem of the form

$$\text{find } U \in \operatorname{span}\Sigma \quad \text{s.t.} \quad (W_i,U) = (W_i,F), \quad \forall W_i \in \Sigma, \tag{3.18}$$

thus L is the identity operator.
In this case, lemma 4 tells us that the solution U is the best approximation of F from $\operatorname{span}\Sigma$ w.r.t. the $L_2(\mathscr{M})$ norm. We shall therefore refer to such an approximation as *the L_2-projection of F onto Σ*. For projections, $[M]$ is simply given by the mass-matrix $[A]$ of Σ

$$[A]_{i,j} = \int_{\mathscr{M}} W_i W_j \mathrm{d}\mathbf{x}. \tag{3.19}$$

We will discuss L_2-projection in more detail in section 4.4.

3.4. Basic Example: Diffusion on a Monge Patch

In this section, we will give an explicit example of a typical FEM approach to finding an approximate solution to a diffusion problem on a curved surface.
Consider the PDE

$$\begin{cases} \partial_t U = d\Delta_{\mathscr{M}} U & \text{on } \mathscr{M} \\ \nabla U \cdot \mathbf{N}_{\partial\mathscr{M}} = 0 & \text{on } \partial\mathscr{M} \\ U(t=0) = U_0 \end{cases}, \tag{3.20}$$

where d is a positive constant.
$\mathscr{M}$ is parameterized by $\mathbf{s} : \Omega \to \mathscr{M}$, where $\Omega = [0,1] \times [0,1]$ and

$$\mathbf{s}(\xi,\eta) = \begin{bmatrix} \xi \\ \eta \\ 1-(\xi-1/2)^2 \end{bmatrix}. \tag{3.21}$$

As a first step, we discretize in time utilizing an implicit backward Euler scheme

$$\begin{aligned} \int_{t_k}^{t_{k+1}} \partial_t U \mathrm{d}t &= \int_{t_k}^{t_{k+1}} d\Delta_{\mathscr{M}} U \mathrm{d}t \\ &\simeq h_k D\Delta_{\mathscr{M}} U^{k+1}, \end{aligned} \tag{3.22}$$

where $h_k = t_{k+1} - t_k$ and $U^k = U(t_k)$. We can utilize the fundamental theorem of calculus on the left-hand side to derive a system for U^{k+1} in terms of U^k.

$$\underbrace{(I - h_k d\Delta_{\mathscr{M}})}_{L(U^{k+1})} U^{k+1} = U^k. \tag{3.23}$$

As a next step we multiply by a test function W and integrate, in order to derive the weak form

$$\begin{aligned} & \int_{\mathscr{M}} W L(U^{k+1}) \mathrm{d}\mathbf{x} = \int_{\mathscr{M}} W U^k \mathrm{d}\mathbf{x}. \\ \iff \quad & \left(W, U^{k+1}\right) + h_k d \left(\nabla_{\mathscr{M}} W, \nabla_{\mathscr{M}} U^{k+1}\right)_{\mathbf{2}} = \left(W, U^k\right), \end{aligned} \tag{3.24}$$

where we made use of lemma (2) in conjunction with the fact that $\nabla U^{k+1} \cdot \mathbf{N}_{\partial\mathscr{M}} = 0$.
Here,

$$\left(\nabla_{\mathscr{M}} W, \nabla_{\mathscr{M}} U^{k+1}\right)_{\mathbf{2}} = \int_{\mathscr{M}} \nabla_{\mathscr{M}} W_i \cdot \nabla_{\mathscr{M}} W_j \mathrm{d}\mathbf{x}. \tag{3.25}$$

As a next step, we build a triangularly tessellated approximation $\mathscr{M}^*$ of $\mathscr{M}$. This can, for example, be achieved by evaluating $\mathbf{s}$ at a set of points $\{p_1, \ldots, p_n\} \subset \Omega$, such that one acquires a set of (equally spaced) points $\{P_1, \ldots, P_n\} \subset \mathscr{M}$. $\mathscr{M}^*$ is then built by linear interpolation between the P_i. The triangles $\{\epsilon_1, \ldots, \epsilon_m\} \equiv \mathscr{A}$, one acquires this way form the elements of the FEM approach and

$$\bigcup_{i=1}^{m} \epsilon_i = \mathscr{M}^*. \tag{3.26}$$

As a next step, we build a piecewise linear FEM basis that satisfies

$$W_i(P_j) = \delta_{i,j}. \tag{3.27}$$

In (3.24), we replace $\mathscr{M} \to \mathscr{M}^*$, $W \to W_i$ and $U^{k+1} = \sum_i c_i^{k+1} W_i$. We derive the following system of equations for the c_i^{k+1}

$$\{[A] + h_k d[D]\}\,\mathbf{c}^{k+1} = [A]\mathbf{c}^k, \tag{3.28}$$

where

$$\begin{aligned}[A]_{i,j} &= \int_{\mathscr{M}^*} W_i W_j \mathrm{d}\mathbf{x} \\ [D]_{i,j} &= \int_{\mathscr{M}^*} \nabla_{\mathscr{M}^*} W_i \cdot \nabla_{\mathscr{M}^*} W_j \mathrm{d}\mathbf{x}.\end{aligned} \tag{3.29}$$

Let $\sigma_{i,j} = \left\{\epsilon_k \in \mathscr{A} \;\middle|\; \epsilon_k \subset \left(\operatorname{supp} W_i \cap \operatorname{supp} W_j\right)\right\}$, then above expressions reduce to

$$\begin{aligned}[A]_{i,j} &= \sum_{\epsilon_k \in \sigma_{i,j}} \int_{\epsilon_k} W_i W_j \mathrm{d}\mathbf{x} \\ [D]_{i,j} &= \sum_{\epsilon_k \in \sigma_{i,j}} \int_{\epsilon_k} \nabla_{\mathscr{M}^*} W_i \cdot \nabla_{\mathscr{M}^*} W_j \mathrm{d}\mathbf{x}.\end{aligned} \tag{3.30}$$

Given that element ϵ_k has area A_k and defining $\mathbf{H}_i$ as the height (vector) of triangle ϵ_k from P_i, assuming that W_i and W_j are have common support on ϵ_k, the following identities hold [25]

$$\int_{\epsilon_k} W_i W_j \mathrm{d}\mathbf{x} = \begin{cases} \frac{A_k}{6} & i = j \\ \\ \frac{A_k}{12} & i \neq j \end{cases} \tag{3.31}$$

and

$$\int_{\epsilon_k} \nabla_{\mathscr{M}^*} W_i \cdot \nabla_{\mathscr{M}^*} W_j \mathrm{d}\mathbf{x} = \frac{\mathbf{H}_i}{\|\mathbf{H}_i\|} \cdot \frac{\mathbf{H}_j}{\|\mathbf{H}_j\|} A_k. \tag{3.32}$$

We can now approximate the initial condition via polynomial interpolation

$$U^0 \simeq \sum_i U_0(P_i) W_i, \tag{3.33}$$

choose a fixed or variable time-step h_k and commence the time-stepping procedure to solve for the U^k.

4

Isogeometric Analysis

Isogeometric analysis is a recent development that aims to combine the techniques of FEA with the flexibility of spline basis functions. Splines constitute piecewise polynomial functions that can be constructed so as to satisfy various continuity properties at the places where the polynomials connect.
Splines are widely utilized within the field of computer-aided design (CAD). Their versatility is taken advantage of when constructing parameterizations for complicated geometries. Within the realm of industrial mathematics, they find applications in FEM approximations of PDEs that impose certain continuity properties on the trial and test spaces (see chapter 3). Even though there exist various types of spline functions, we will exclusively utilize B-splines (basis splines).
The B-spline basis functions $N_1, \dots, N_n$ live in parameter space, in one dimension usually the interval $\Omega = [0,1]$, i.e. $N_i : \Omega \to \mathbb{R}$. This interval is utilized to parametrize a B-spline curve in $\mathbb{R}^k$ (the one-dimensional geometry $\mathscr{M}$) in the following way: $\mathbf{s}(\xi) = \sum_{i=1}^{n} N_i(\xi)\mathbf{B}_i$, where $\xi \in \Omega$ and the $\mathbf{B}_i$ are points in $\mathbb{R}^k$ (referred to as the *'control points'*).
Bivariate B-splines, or B-splines with higher-order variable dependencies are straight-forwardly constructed from univariate B-splines through tensor-products and can be utilized to construct B-spline surfaces, solids or higher-order geometric objects (see section 4.5). When $\mathbf{s}$ is a bivariate mapping, quadrilaterals in the parametric domain are referred to as *'patch'*. In many application a single patch suffices to parameterize $\mathscr{M}$ but in more complicated settings several patches are needed. In this case it is customary to either construct $\mathscr{M}$ from several patch-specific mappings or craft a global mapping from a parametric domain consisting of several patches.

Remark. *For convenience, we will assume that $\mathscr{M}$ is always the result of one global mapping operator $\mathbf{s} : \Omega \to \mathscr{M}$ (so Ω can be comprised of several patches).*

The basic idea of isogeometric analysis (IgA) is to utilize the same set of spline-functions as a basis for the geometry $\mathscr{M}$ as well as the solution space. Thus, any function on $\mathscr{M}$ is approximated by an element from the linear span of the very same basis that is utilized to parameterize it via the operator $\mathbf{s} : \Omega \to \mathscr{M}$. In this context one might regard the global mapping $\mathbf{s}$ as a vector-valued function comprised of $k + l$, where l is the amount of unknows on $\mathscr{M}$, functions from span Σ with scalar weights, such that all components of $\mathbf{s}$ and all relevant functions on $\mathscr{M}$ jointly make up a collection of scalar functions all from the linear span of the same basis Σ. This is referred to as the isoparametric concept.
Unlike in standard FEM, tessellation of $\mathscr{M}$ is very uncommon. This is a result of the fact that a cleverly-chosen spline basis is often capable of parametrizing $\mathscr{M}$ exactly. If $\mathscr{M}$ is too complicated for

an exact parameterization, it is customary to build an approximation from Σ, utilizing L_2-projection (see section 4.4) or polynomial interpolation.

Remark. *The main motivation to utilize B-splines in this work, is the resulting smoothness of the geometry.*

The most striking difference between IgA and FEM, apart from the basis functions used, is the utilization of a global operator $\mathbf{s}$ to parameterize $\mathcal{M}$ as opposed to the parameterization of each individual element from a reference element. In IgA the individual patches of the parametric domain are thus local to the geometry as opposed to individual elements.
We start off by introducing the concept of univariate B-spline basis functions and one-dimensional refinement techniques. After that, we generalize these principles to bivariate B-spline functions and present refinement techniques in two dimensions. We end this chapter with an example for a typical IgA-approach to the problem from section 3.4.

4.1. Knot Vectors

B-splines come in various shapes and with various continuity properties. Their properties follow from the way they are constructed using the so-called knot vector. A knot vector is a partition of the parameter space (here: $\Omega = [0,1]$) into segments. It is written in the following way:

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\},$$

where n is the amount of (desired) basis functions, p their polynomial order and $\xi_i \in \Omega$ is the i^{th} knot. The knot vector is an non-decreasing sequence of knots (i.e. $\xi_j \geq \xi_i$ for $j > i$), that usually starts on $\xi_0 = 0$ and ends on $\xi_{n+p+1} = 1$. The knot vector also allows for the repetition of knots, the cardinality of each repetition in conjunction with the knot-spacing leading to the great variety of different possible shapes.
The knot vector without the repeated knots forms the equivalent of a grid and each segment of the the grid, fenced off by two unequal consecutive knots, is called an element (analogous to standard-FEA).

4.2. Constructing B-Splines

Given a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, the i^{th} basis function $N_{i,p}(\xi)$ with polynomial order p is constructed recursively starting from

$$N_{i,0} = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise} \end{cases}, \tag{4.1}$$

utilizing the recurrence relation (with $\frac{0}{0} \equiv 0$)

$$N_{i,q}(\xi) = \frac{\xi - \xi_i}{\xi_{i+q} - \xi_i} N_{i,q-1}(\xi) + \frac{\xi_{i+q+1} - \xi}{\xi_{i+q+1} - \xi_{i+1}} N_{i+1,q-1}(\xi), \tag{4.2}$$

and iterating until $q = p$ [12].
In general, the support of basis function $N_{i,p}(\xi)$ is given by the interval $[\xi_i, \xi_{i+p+1}]$ in parameter space, thus higher order basis functions tend to have a larger support unless they are accompanied by repeated knots.
It is easily seen that for $p = 0$ and $p = 1$, the basis functions that are constructed this way coincide with

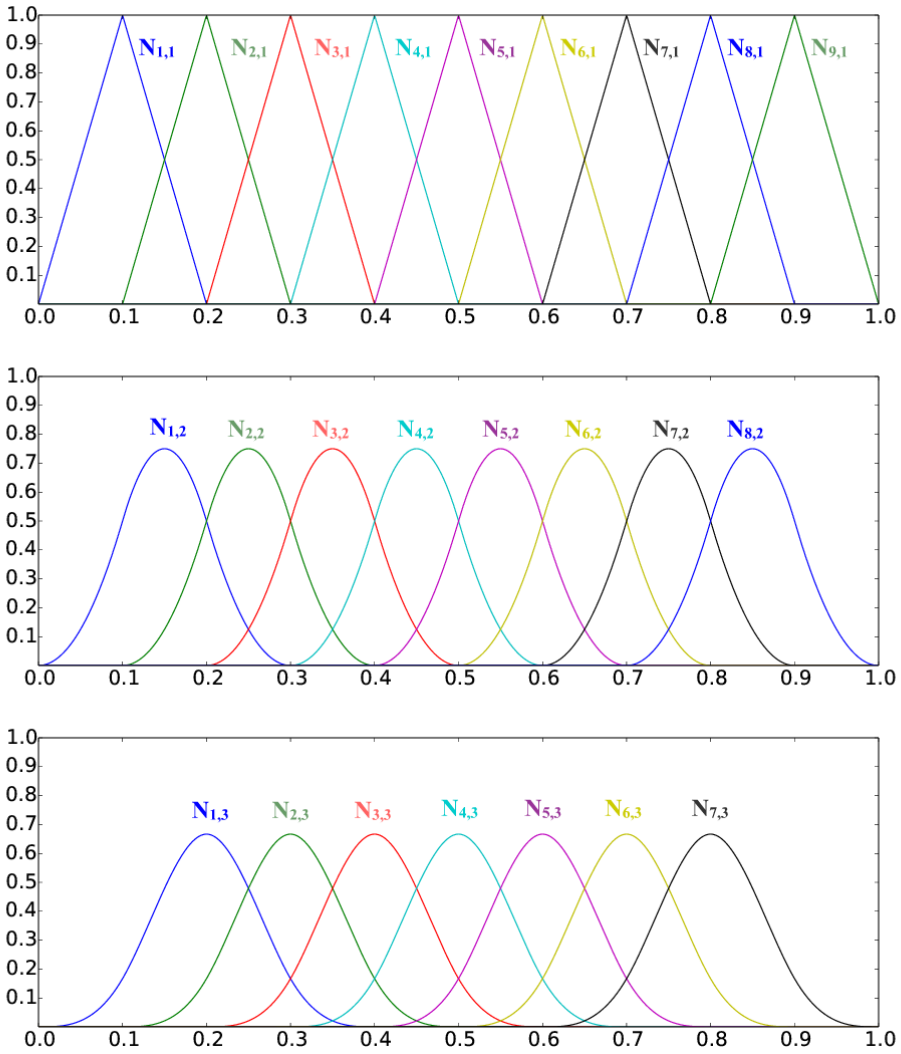


Figure 4.1: B-spline basis functions with $p = 1, 2$ and 3, all constructed from the knot vector $\Xi = \{0, 1/10, 2/10, \ldots, 1\}$ utilizing formulae (4.1) and (4.2). Individual functions are represented in various colors. Note that $\xi_1 = 0$ and $\xi_{10} = 1$ such that the amount of basis functions is given by $n(p) = 10 - (p + 1)$. Furthermore note that the maximum value that is attained by each individual function decreases as p increases while their supports broaden. Each basis with $p < 3$ can be regarded as the result of an intermediate iteration of the iterative process prescribed by (4.1) and (4.2) and each basis with $p = q$ is constructed from the basis with $p = q - 1$.

the ordinary zeroth and first order basis functions from FEA. This changes for $p \geq 2$ (see figure 4.1). From the way the basis functions are constructed utilizing the knot vector, it is seenthat in general $N_{i,p}(\xi)$ has $p - m_j$ continuous derivatives across knot $\xi_j \in \{\xi_i, \ldots, \xi_{i+p+1}\}$, where m_j is the multiplicity of the value of ξ_j in $\{\xi_i, \ldots, \xi_{i+p+1}\}$. This means that p^{th} order basis functions constructed from a knot vector without repeated knot values have at least $p - 1$ continuous derivatives everywhere, unlike ordinary higher order basis functions that remain interpolatory across the boundaries of their support irrespective of their polynomial order.
Additionally, we can state the following lemma about B-splines

Lemma 5. *Let Σ be the p^{th}-order B-spline basis constructed from $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$, then the*

$N_{i,p} \in \Sigma$ form a partition of unity on the interval $[\xi_{p+1}, \xi_{n-p-1}]$, i.e.

$$\sum_i N_{i,p}(\xi) = 1, \quad \forall \xi \in [\xi_{p+1}, \xi_{n+1}]. \tag{4.3}$$

Furthermore, B-spline functions are non-negative on the entire parametric domain.

Proof. See [12, p. 22]. □

Knot-vectors over the interval $[0,1]$ with the property

$$\Xi = \left\{ \underbrace{0,\ldots,0}_{p+1 \text{ times}}, \underbrace{\xi_{p+2},\ldots,\xi_n}_{n-p-1 \text{ terms}}, \underbrace{1,\ldots,1}_{p+1 \text{ times}} \right\}, \tag{4.4}$$

with $0 < \xi_i < 1$, can be utilized to construct *'clamped'* bases. For $p > 1$, B-spline functions do not, in general, assume the value 1 at some element boundary which means that B-spline curves do not necessarily cross their control points. The term clamped refers to the fact that the functions $N_{0,p}$ and $N_{n,p}$ resulting from a clamped knot vector do assume the value 1 at $\xi = 0$ and $\xi = 1$, respectively. This means that B-spline curves constructed from a clamped basis have the property that the control points $\mathbf{B}_1$ and $\mathbf{B}_n$ constitute the starting and end points of the B-spline curve, irrespective of the polynomial order p. Thus the terminology stems from the fact that the curve is *'clamped'* between $\mathbf{B}_1$ and $\mathbf{B}_n$.
In general we can state that if a knot value has multiplicity p, the basis will be interpolatory at that knot (meaning C^0 continuous and some basis function will attain the value 1 there such that all others assume the value 0). If it has multiplicity $p+1$ the basis will acquire C^{-1} continuity, which implies the existence of discontinuity. For an example of an interpolatory knot see figure 4.2.

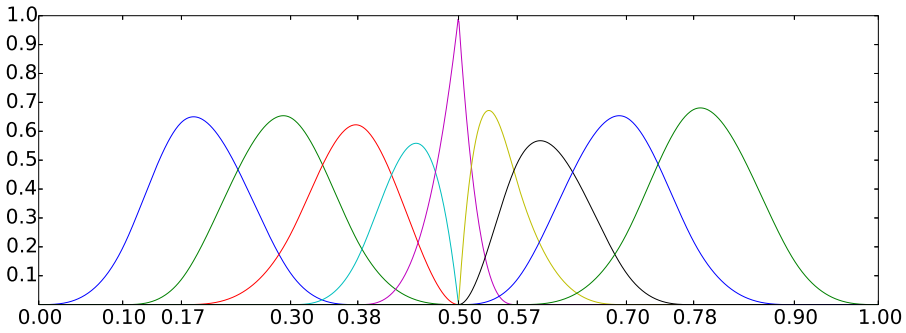


Figure 4.2: B-spline basis with $p = 3$, resulting from the knot vector $\Xi = \{0, 0.1, 0.17, 0.3, 0.38, 0.5, 0.5, 0.5, 0.57, 0.7, 0.78, 0.9, 1\}$. It is seen that the knot value 0.5 has multiplicity $3 = p$ such that the basis is interpolatory at $\xi = 0.5$. Additionally, this B-spline basis is non-uniform since it has unequally spaced knots. The basis forms a partition of unity on the interval $[0.3, 0.7]$.

According to lemma 5, clamped B-spline bases form a partition of unity on the entire parametric interval $[0,1]$, i.e.

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1, \quad \forall \xi \in \Omega. \tag{4.5}$$

In the remainder of this thesis knot vectors of the form

$$\Xi = \left\{ \underbrace{0,\ldots,0}_{p+1 \text{ times}}, \underbrace{\frac{1}{n-p}, \frac{2}{n-p}, \ldots, \frac{n-p-1}{n-p}}_{n-p-1 \text{ terms}}, \underbrace{1,\ldots,1}_{p+1 \text{ times}} \right\} \tag{4.6}$$

will be of major importance. We shall refer to them as *'uniform, clamped knot vector of length n and order p'*.

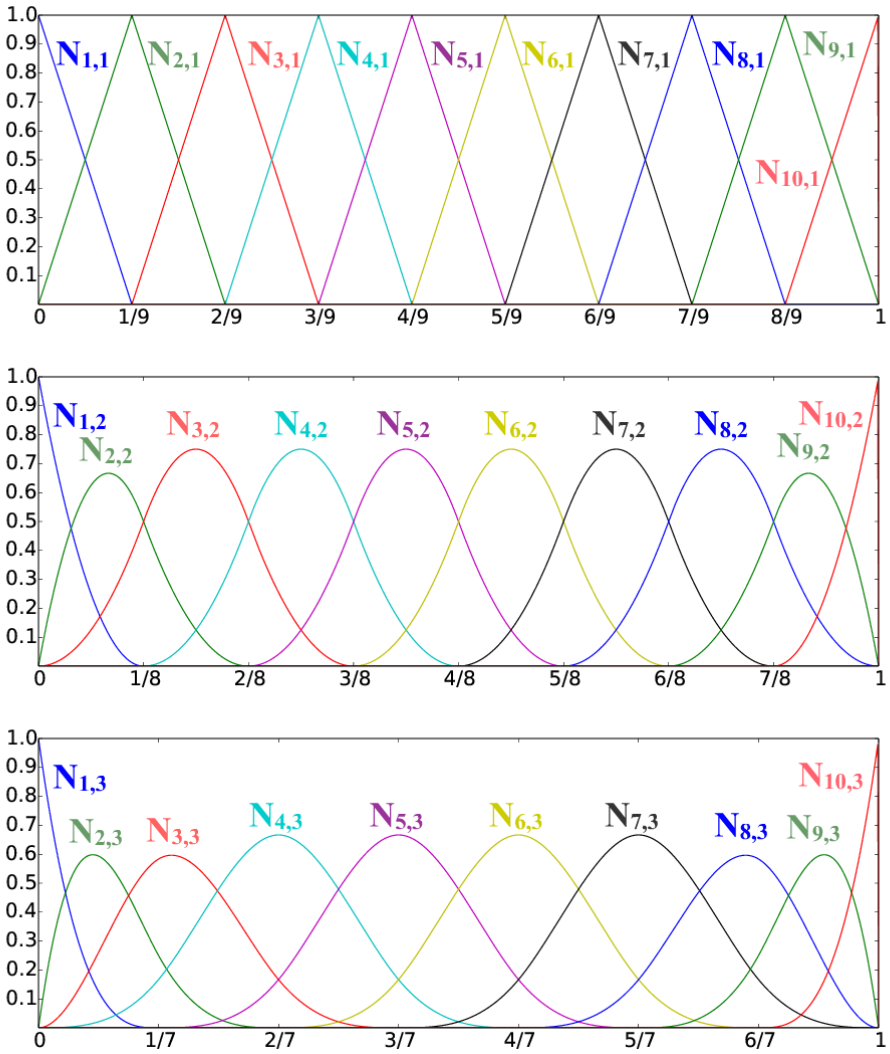


Figure 4.3: The B-spline bases constructed from uniform knot vectors of length 10 and orders $p = 1, 2$ and 3. Note that the amount of basis functions that are *'different'* from functions that are supported by internal elements is exactly equal to p at each interval boundary. Since the multiplicity of knot values 0 and 1 is equal to $p+1$, the outermost basis functions at either side attain the value 1, irrespective of the polynomial degree p. If the interval were to be slightly extended, a discontinuity (C^{-1} continuity) in the outermost functions would become visible.

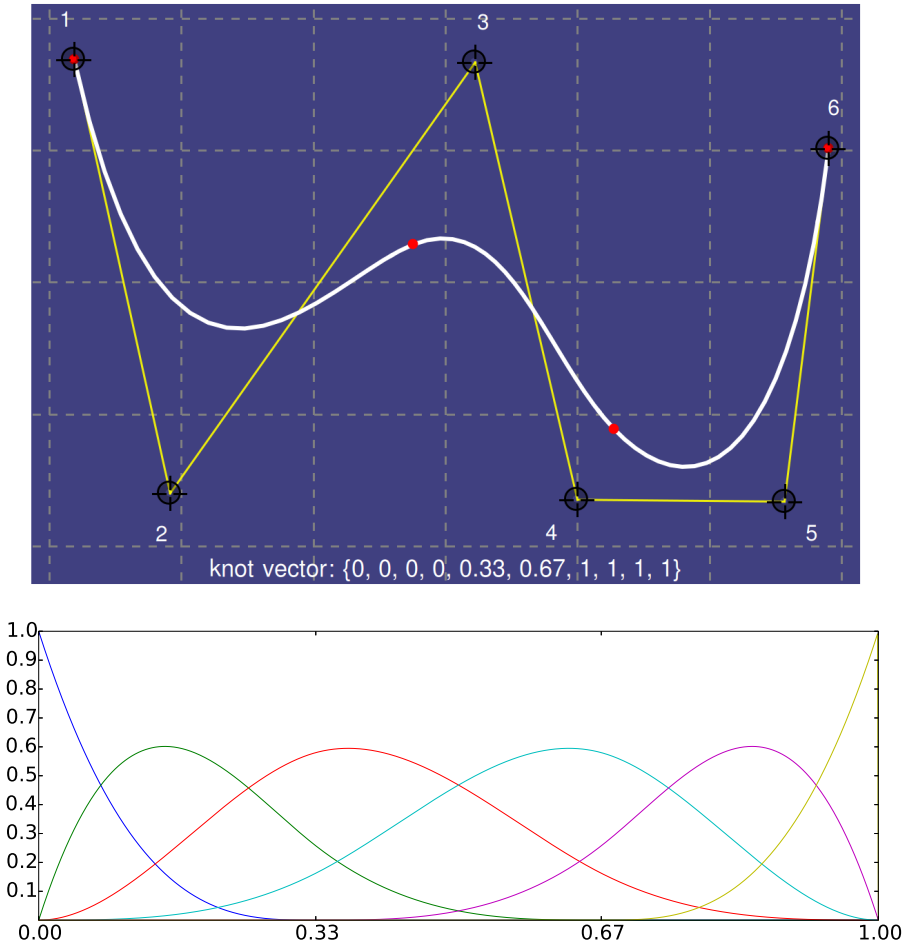


Figure 4.4: A B-spline curve resulting from the $p = 3$ basis with $\Xi = \{0,0,0,0,0.33,0.67,1,1,1,1\}$ and a set of control points. Notice that the curve intersects only with control points $\mathbf{B}_1$ and $\mathbf{B}_6$, which is a result of the clamped basis property.

Examples of bases resulting from uniform, clamped knot vectors are depicted in figure 4.3. An example of a one-dimensional B-spline curve in $\mathbb{R}^2$ can be found in figure 4.4.

4.3. Refinement in One Dimension

In numerical simulations, especially in time-stepping methods, one is frequently confronted with problems of the form

$$\text{solve for } \mathbf{c}^{k+1} \text{ from } F(\mathbf{c}^{k+1}) = G(\mathbf{c}^k), \tag{4.7}$$

where the $\mathbf{c}^k$ contains the weights of a functional from span Σ at time instance t^k and F, G are (usually linear) operators.
In certain settings it can make sense to replace an element from the linear span of the coarse basis Σ, by an element from the finer basis $\bar{\Sigma}$ and continue the iterative process utilizing the finer basis. This can, for instance, become a necessity in problems that enforce local increase of the resolution like shock waves or in settings with a time-dependent geometry. Thanks to the possibility to construct a finer basis $\bar{\Sigma}$ that satisfies span $\Sigma \subset$ span $\bar{\Sigma}$, this replacement can be carried out without projection errors. In light of (4.7), if refinement is carried out after the k^{th} iteration, $\mathbf{c}^k$ is replaced by its corresponding vector with respect to $\bar{\Sigma}$ and the iterative process is continued using $\bar{\Sigma}$. It is convenient to

be able to carry out this basis transformation without projection errors as it may be hard to determine how they propagate in (4.7).
All refinement techniques can be divided into three broad categories: refinement based on order elevation, refinement based on knot-insertion and a third that has no classical FEM counterpart [12, p. 41]. The technique that closest resembles the traditional h-refinement from FEA is the technique of knot-insertion and it will be the subject of the discussion that follows (in this thesis order elevation will not be discussed as we shall exclusively utilize bases of uniform order).

Remark. *Generally speaking, the density of (unequal) knots in a given segment of Ω is positively correlated with the 'resolution' of the basis there.*

This is a result of the fact that a higher density of knots implies the presence of a higher density of basis functions, leading to a locally higher resolution.
Given a knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, we can refine the grid by adding m new knots such that we acquire $\Xi \subset \bar{\Xi} = \{\xi_1 = \bar{\xi}_1, \bar{\xi}_2, \dots, \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}\}$.
We shall refer to $\bar{\Xi}$ as an *enriched knot-vector*. As an example, consider $\Xi = \{0, 0.5, 1\}$, then $\bar{\Xi} = \{0, 0.25, 0.5, 0.75, 1\}$ is an enriched knot vector of Ξ. We can state the following lemma

Lemma 6. *Let Σ be a B-spline basis constructed from the knot-vector Ξ. If $\bar{\Sigma}$ is a B-spline basis constructed from an enriched knot-vector $\bar{\Xi}$ of Ξ, then* $\operatorname{span}\Sigma \subset \operatorname{span}\bar{\Sigma}$.

Proof. See [13] □

According to lemma (6), any element from the coarse function space Σ is also contained in $\bar{\Sigma}$, we will now present a method to find the new control points from the old control points. We can construct a new set of control points $\bar{\mathbf{B}} = \{\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, \dots \bar{\mathbf{B}}_{n+m}\}$ from the original set of control points $\mathbf{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n\}$, utilizing a suitable linear transformation of the vectors contained in $\mathbf{B}$. The matrix $[T^p]$ associated with the transformation can be constructed recursively in the following way [12, p. 37]

$$[T^0]_{i,j} = \begin{cases} 1 & \bar{\xi}_i \in [\xi_j, \xi_{j+1}) \\ 0 & \text{else} \end{cases} \tag{4.8}$$

and

$$[T^{q+1}]_{i,j} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j}[T^q]_{i,j} + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}}[T^q]_{i,j+1} \quad \text{for} \quad q = 0, 1, 2, \dots, p-1. \tag{4.9}$$

The new and old weights are related as follows

$$\forall\, i \in \{1, \dots, n+m\}: \quad \bar{\mathbf{B}}_i = \sum_{j=1}^{n} [T^p]_{i,j}\mathbf{B}_j. \tag{4.10}$$

Above transformation ensures that the parametrization is not altered, i.e.

$$\sum_{i=1}^{n+m} \bar{N}_{i,p}\bar{\mathbf{B}}_i(\xi) = \sum_{i=1}^{n} N_{i,p}\mathbf{B}_i(\xi), \quad \forall \xi \in \Omega, \tag{4.11}$$

where the $\bar{N}_{i,p}$ are the new basis functions constructed from $\bar{\Xi}$. Note that above procedure is compatible with functions from $\operatorname{span}\Sigma$ with scalar weights.
Refinement is a proficient tool to increase the local resolution of the function space. With above principles, we are capable of refining the basis locally, by only adding knots in a subregion of Ω. The

trade-off between precision and computational costs is the main motivation for local refinement as a global refinement may increase the resolution (and by that the computational costs) in regions where it is unnecessary.
For an example of a locally refined basis, see figure 4.5.

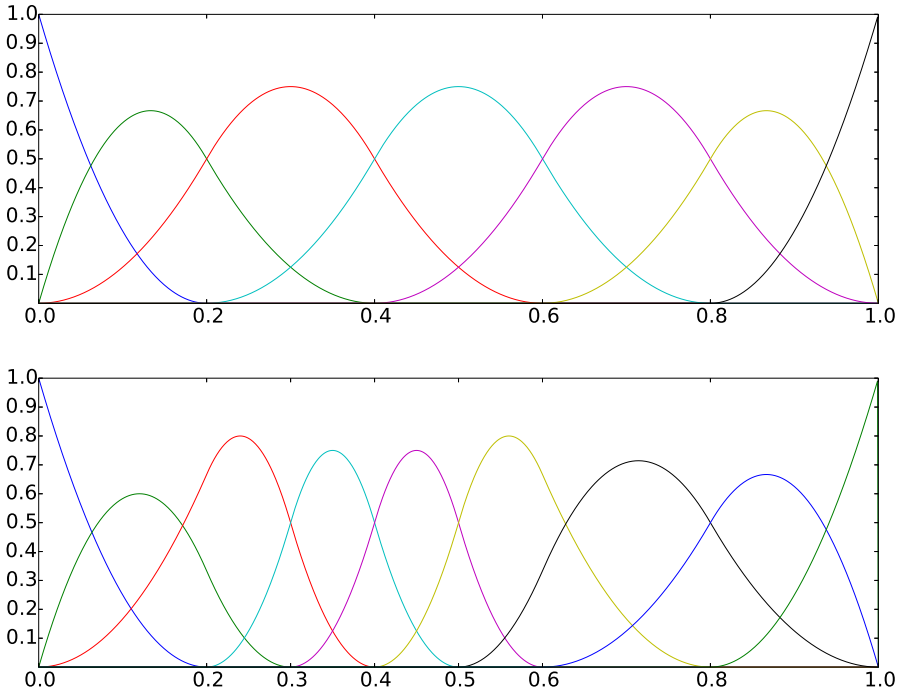


Figure 4.5: The $p = 2$ bases resulting from $\Xi = \{0,0,0,0.2,0.4,0.6,0.8,1,1,1\}$ and the locally enriched knot vector $\bar{\Xi} = \{0,0,0,0.2,0.3,0.4,0.5,0.6,0.8,1,1,1\}$. The bases satisfy $\operatorname{span}\Sigma \subset \operatorname{span}\bar{\Sigma}$.

4.4. L_2-Projection

In section 4.3, we introduced the concept of refinement via knot-insertion. We have introduced a general method to construct the new control points (scalars) $\bar{\mathbf{B}}_i$ that preserve the geometry (function) under refinement. In this section we will present an alternate way to determine the weights with respect to the refined basis. This approach constitutes a Galerkin-based projection as opposed to a collocation technique.
To determine the $\bar{\mathbf{B}}_i$, we can simply project each component of the vector-valued function $\mathbf{s}(\xi)$ onto the refined basis $\bar{\Sigma} = \{\bar{w}_1, \ldots, \bar{w}_m\}$, utilizing L_2-projection (see section 3.3). The objective is

$$\begin{aligned} &\text{find } \mathbf{u} = \mathbf{s}, \\ &\text{s.t. } \mathbf{u}_i \in \operatorname{span}\bar{\Sigma}. \end{aligned} \tag{4.12}$$

Since $\mathbf{s}(\xi) : \Omega \to \mathbb{R}^k$, we are, in fact, confronted with a problem of the form

$$\forall i \in \{1,\ldots,k\} : \begin{cases} \text{find} & u_i(\xi) = s_i(\xi) \\ \text{s.t.} & u_i \in \operatorname{span}\bar{\Sigma} \end{cases} . \tag{4.13}$$

Each individual system can now be solved utilizing an L_2-projection. It is customary to carry out this projection over the geometry (i.e. minimizing the $L_2(\mathcal{M})$-error as opposed to the $L_2(\Omega)$-error), since

usually the geometrical mass-matrix

$$[A]_{i,j} = \int_\Omega \bar{w}_i \bar{w}_j \sqrt{g} \mathrm{d}\boldsymbol{\xi}, \tag{4.14}$$

is needed for other computations as well (and thus has to be assembled only once).
Again, since $\operatorname{span}\bar{\Sigma} \subset \operatorname{span}\Sigma$, this projection should in principle be possible without projection errors.
L_2-projection is applicable outside of refinement, as well. Going back to some IgA basis Σ, in general, if the local counterpart $\mathbf{f}$ of the right hand side function $\mathbf{F} : \mathscr{M} \to \mathbb{R}^k$ is not contained in $\operatorname{span}\Sigma$, we can nevertheless find a 'good' approximation from $\operatorname{span}\Sigma$ utilizing an L_2-projection. Again, it is recommended to carry out this projection over $\mathscr{M}$ as opposed to Ω since the geometrical mass-matrix has to be assembled either way and the $L_2(\mathscr{M})$-norm is 'canonical' for functions $\mathbf{U} : \mathscr{M} \to \mathbb{R}^k$. The problem can then be formulated as follows

$$\forall i \in \{1,\dots,k\} : \text{find} \min_{u_i \in \operatorname{span}\Sigma} \int_\Omega \left(u_i - f_i\right)^2 \sqrt{g} \mathrm{d}\boldsymbol{\xi}. \tag{4.15}$$

The weights $c_1^i,\dots,c_n^i$ of u_i with respect to $\Sigma = \{w_1,\dots,w_n\}$, are found by solving

$$[A]\mathbf{c}^i = \mathbf{f}^i, \tag{4.16}$$

where

$$\mathbf{f}_j^i = \int_\Omega f_i w_j \sqrt{g} \mathrm{d}\boldsymbol{\xi}, \tag{4.17}$$

and $[A]$ is the mass-matrix corresponding to Σ.
For its ease of implementation and the availability of the geometrical mass-matrix, the L_2-projection method is, in fact, more popular than the method presented at the end of section 4.3.
We end this section with the following proposition

Proposition 2. *Let the B-spline basis $\bar{\Sigma}$ satisfy $1 \in \operatorname{span}\Sigma$, then the L_2-projection is* mass conserving.

Proof. See proposition A.3.1 in the appendix. □

4.5. B-Spline Surfaces

We shall now generalize the principle of B-splines to two dimensions.
Two-dimensional parametric surfaces embedded in $\mathbb{R}^k$ require two free variables instead of one in order to be parameterized. We shall use ξ and η as free variables in parameter space. We utilize two knot vectors $\Xi = \{\xi_1,\xi_2,\dots,\xi_{n+p+1}\}$ and $\mathscr{H} = \{\eta_1,\eta_2,\dots,\eta_{m+q+1}\}$ to construct the p^{th}- and q^{th}-order bases $\{N_{1,p},\dots,N_{n,p}\}$ and $\{M_{1,q},\dots,M_{m,q}\}$. B-spline surfaces are then parameterized in the following way

$$\mathbf{s}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B}_{i,j}, \quad (\xi,\eta) \in \Omega \tag{4.18}$$

where $\Omega = [0,1] \times [0,1]$.
Here the $N_{i,p}(\xi)M_{j,q}(\eta)$ become the $N = n \times m$ new bivariate basis functions and the $\mathbf{B}_{i,j}$ are the

control points in $\mathbb{R}^k$. One may choose to replace the tensor-product numbering i, j by a single index such that

$$\mathbf{s}(\xi,\eta) = \sum_{i=1}^{N} w_i(\xi,\eta)\mathbf{B}_i, \tag{4.19}$$

where the global index usually follows from a lexicographical numbering. The bivariate B-spline basis Σ is then simply given by $\Sigma = \{w_1, \ldots, w_N\}$.
If the two individual, one-dimensional bases are a partition of unity, then so will be their tensor-product basis, since

$$\sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta) = \left(\sum_{i=1}^{n} N_{i,p}(\xi)\right)\left(\sum_{j=1}^{m} M_{j,q}(\eta)\right) = 1. \tag{4.20}$$

For an example of a bivariate tensor-product basis with lexicographical global numbering, see figure 4.6.

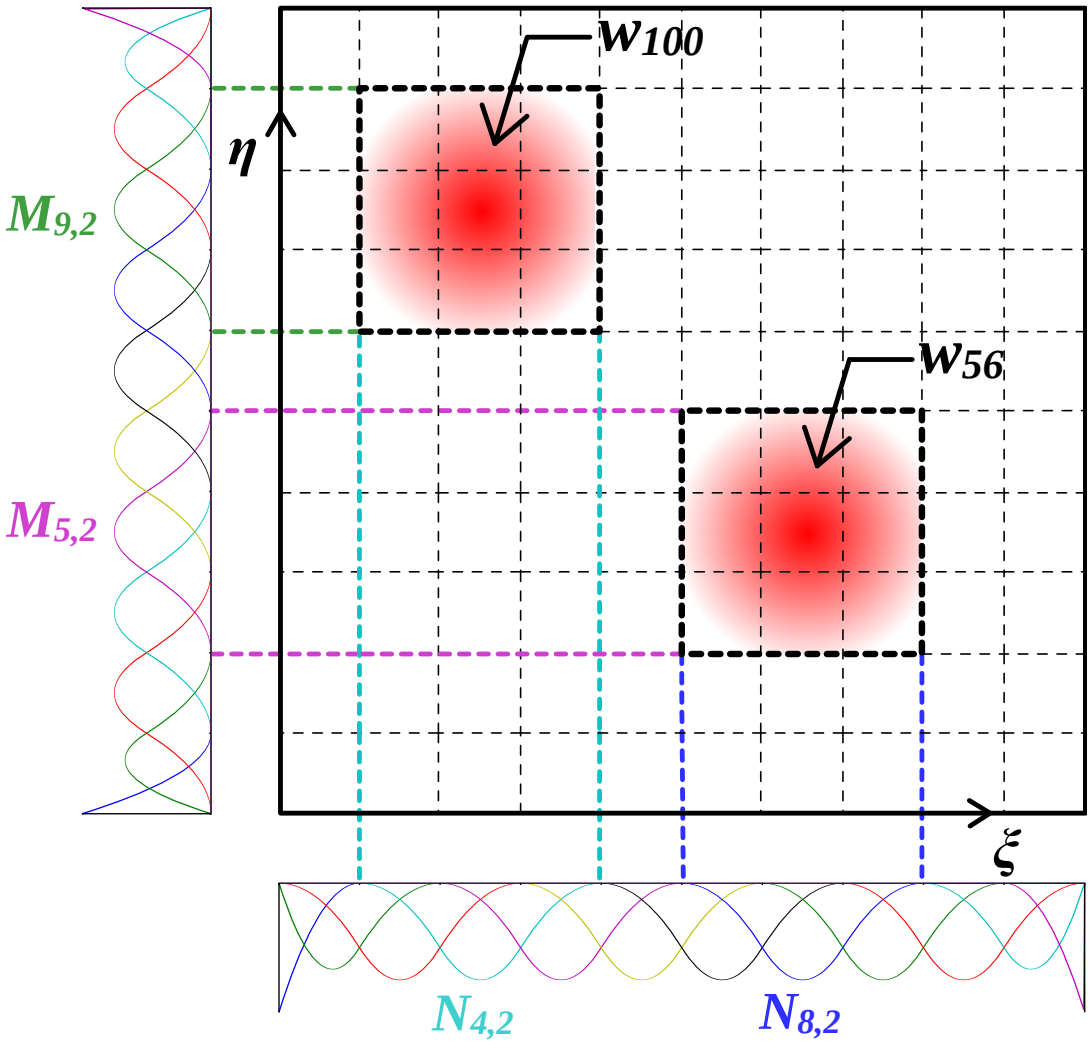


Figure 4.6: Bivariate tensor-product basis constructed from the knot vectors $\Xi = \mathcal{H} = \{0,0,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,1\}$, with $p = q = 2$. The indices of the w_i follow from a lexicographical ordering of the form $(k,l) \to k + (l-1)n$, with $n = 12$.

Remark. *In the bivariate case, the two-dimensional elements follow from a tensor-product of the elements belonging to each of the univariate bases. The techniques from section 4.4 are applicable in the bivariate case in the exact same way.*

4.6. Refinement in Two Dimensions

The knot-insertion technique from section 4.3 is also applicable in two dimensions. We can refine in ξ and η by adding knots to the corresponding knot vectors, Ξ and $\mathscr{H}$, respectively. Thanks to the tensor-product property of two-dimensional B-spline basis functions, the basis $\bar{\Sigma}$ constructed from the enriched knot-vectors $\bar{\Xi}$ and $\bar{\mathscr{H}}$ satisfies $\operatorname{span}\bar{\Sigma} \subset \operatorname{span}\Sigma$, too.
The drawback of refinement via enriched knot vectors in two dimensions, however, is its non-local nature. Since we build a new basis from a global knot vector, refinement in one direction stretches out over the entirety of the complementary direction (see figure 4.7). This leads to a possibly too rich refinement, that demands heavier computations in regions where refinement might not even be necessary.
A more advanced refinement technique is hierarchical refinement. To understand hierarchical re-

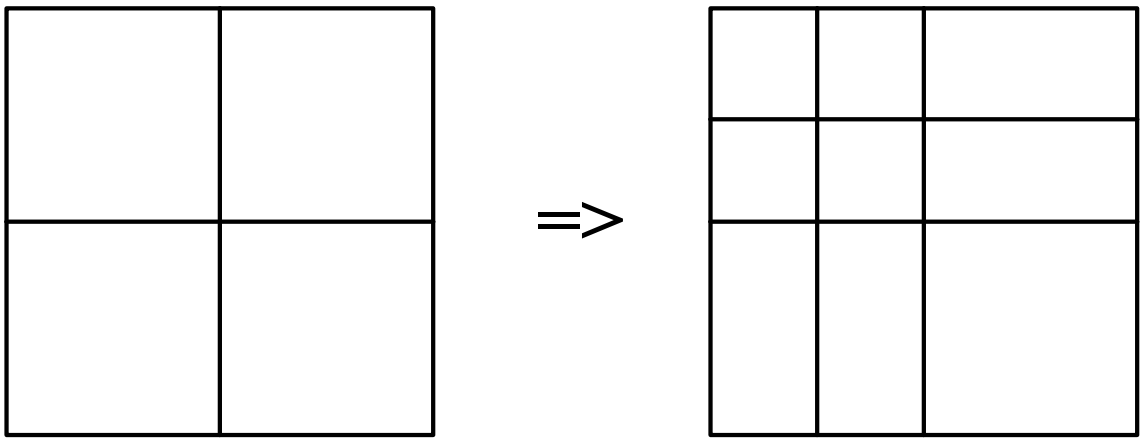


Figure 4.7: The non-local nature of refinement via knot-vectors. Even though only a single cell should be refined, the refinement stretches out over the entire domain leading to a refinement in regions where it might not be necessary.

finement, let us assume that we start with a basis $\mathscr{B}_1$ that is constructed from two uniform knot vectors Ξ_1 and $\mathscr{H}_1$. We can construct the finer bases $\mathscr{B}_k$ recursively by taking the knot vectors Ξ_{k-1} and $\mathscr{H}_{k-1}$ and adding new knots. Let us assume that we enrich the knot vectors in a straight-forward fashion by simply halving the spacing between all (unequal) knots. Carrying out a tensor-product, we acquire the bases $\mathscr{B}_1, \mathscr{B}_2, \ldots$ that live on the increasingly finer grids $\mathscr{T}_1, \mathscr{T}_2, \ldots$.
The basic idea of hierarchical refinement is to divide Ω into a disjoint set of elements $\mathscr{A}$ from the hierarchical grids $\mathscr{T}_1, \mathscr{T}_2, \ldots$, such that the elements from $\mathscr{A}$ jointly make up Ω. Usually, $\mathscr{A}$ is initialized to $\mathscr{A} = \mathscr{T}_1$. Grid refinement is then carried out by selecting elements from $\mathscr{A}$ and replacing them with several elements from the finer grid $\mathscr{T}_2$ and so on. From the hierarchical knot-structure of the $\mathscr{T}_i$ elements from coarser grids can always be completely replaced by elements from finer grids, such that the grids we acquire with this procedure always make up Ω. Element replacement is accompanied by basis refinement, replacing coarse basis functions living on finer elements by their finer counterparts. The following lemma will help us derive an efficient way to do so

Lemma 7. *Let $w_i^k \in \mathscr{B}_k$ and let*

$$\sigma_{i,k}^l = \{w_j^l \in \mathscr{B}_l | \operatorname{supp} w_j^l \subset \operatorname{supp} w_i^k\}.$$

If $l > k$, then $w_i^k \in \operatorname{span} \sigma_{i,k}^l$, i.e. w_i^k can be represented by a linear combination of the $w_j^l \in \sigma_{i,k}^l$.

Proof. See [42, p. 86]. □

In hierarchical refinement it is important to note that it is usually not recommended to refine single cells. The reason being that these single cells might be too small to contain a full basis function of the current basis and/or to harbor finer basis functions (thus making the refinement pointless). It might therefore be necessary to choose the refinement region slightly larger than anticipated, which makes the refinement technique not fully local but sufficiently local in practice.
Nevertheless, we present a refinement algorithm that refines $\mathscr{A}$ utilizing no more than a set of elements ϵ from $\mathscr{A}$ as input. The basic idea is to refine the grid and replace a coarse function by its finer counterparts whenever it is nonvanishing exclusively on elements from a higher level in the element hierarchy after refinement. For simplicity we assume that $\mathscr{A} \supset \epsilon \subset \mathscr{T}_k$, i.e. ϵ only contains elements from $\mathscr{A}$ from one level in the element hierarchy. Efficient element selection depends on the application and will be further discussed in chapter 10. Algorithm 1 gives the pseudocode to refining $\mathscr{A}$

Algorithm 1 Refine $\mathscr{A}$ (ϵ) * $\epsilon \subset \mathscr{A}$ should only contain elements from the same level $\mathscr{T}_k$ in the element hierarchy

1: **for** all $\epsilon_i^k \in \epsilon$ **do**
2: replace ϵ_i^k by $\{\epsilon_j^{k+1} \in \mathscr{T}_{k+1} | \epsilon_j^{k+1} \subseteq \epsilon_i^k\}$ in $\mathscr{A}$
3: **end for**
4: $\sigma \leftarrow \left\{\Sigma \ni w_i^k \in \mathscr{B}_k | w_i^k \text{ is nonvanishing exclusively on elements from } \mathscr{T}_{k+1} \text{ contained in } \mathscr{A}\right\}$
5: **if** $\sigma \neq \emptyset$ **then**
6: **for** all $w_i^k \in \sigma$ **do**
7: Project w_i^k onto $\sigma_{i,k}^{k+1} = \{w_j^{k+1} \in \mathscr{B}_{k+1} | \operatorname{supp} w_j^{k+1} \subset \operatorname{supp} w_i^k\}$
8: In each function from $\operatorname{span} \Sigma$, replace w_i^k by its projection from $\operatorname{span} \sigma_{i,k}^{k+1}$
9: $\Sigma \leftarrow \Sigma \setminus \{w_i^k\} \cup \sigma_{i,k}^{k+1}$
10: **end for**
11: **end if**

from a set of elements ϵ. It is loosely based on the algorithm given in [42] but has been modified for the purposes of this thesis. The if clause in line number (5) serves as to ensure that whenever a coarse basis function w_i^k is replaced by finer ones, each function of the form

$$u = \sum_{j \neq i} c_j w_j + c_i w_i^k \tag{4.21}$$

is replaced by

$$u = \sum_{j \neq i} c_j w_j + c_i \left(c_\alpha w_\alpha^{k+1} + \ldots + c_\gamma w_\gamma^{k+1} \right), \tag{4.22}$$

where $c_\alpha w_\alpha^{k+1} + \ldots + c_\gamma w_\gamma^{k+1}$ is the projection of w_i^k onto $\sigma_{i,k}^{k+1}$. Above procedure transforms each function into an element from the linear span of the updated basis.

Remark. *Being able to carry out a global basis transformation by a sequential projection of all components onto the relevant elements from the refined basis as in line number (8) of algorithm (1), is a result of lemma 7. A sequential approach can be cheaper than a global projection when only a few basis functions are nonvanishing exclusively on the refinement region. If the refinement region supports many coarse basis functions, one might choose to update the basis first and utilize the principles from section 4.4 to project each function onto the newly formed basis in a global fashion. Algorithm (2) represents the pseudocode corresponding to refinement with global projection.*

Algorithm 2 Refine Globally $\mathscr{A}$ (ϵ) * $\epsilon \subset \mathscr{A}$ should only contain elements from the same level $\mathscr{T}_k$ in the element hierarchy

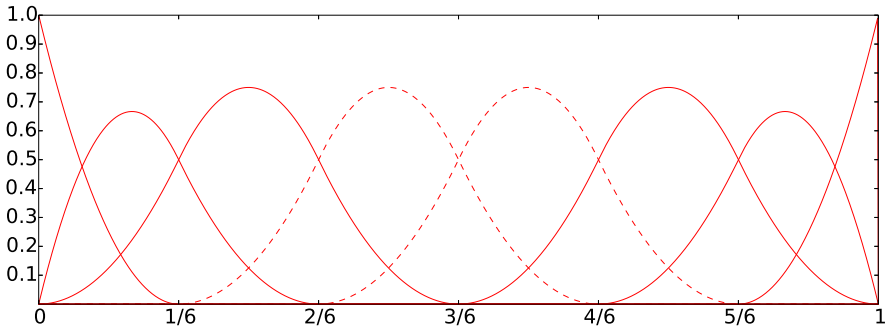
1: **for** all $\epsilon_i^k \in \epsilon$ **do**
2: replace ϵ_i^k by $\{\epsilon_j^{k+1} \in \mathscr{T}_{k+1} | \epsilon_j^{k+1} \subseteq \epsilon_i^k\}$ in $\mathscr{A}$
3: **end for**
4: $\sigma \leftarrow \{\Sigma \ni w_i^k \in \mathscr{B}_k | w_i^k$ is nonvanishing exclusively on elements from $\mathscr{T}_{k+1}$ contained in $\mathscr{A}\}$
5: **if** $\sigma \neq \emptyset$ **then**
6: $\Sigma_{\text{old}} \leftarrow \Sigma$
7: **for** all $w_i^k \in \sigma$ **do**
8: $\sigma_{i,k}^{k+1} \leftarrow \{w_j^{k+1} \in \mathscr{B}_{k+1} | \operatorname{supp} w_j^{k+1} \subset \operatorname{supp} w_i^k\}$
9: $\Sigma \leftarrow \Sigma \setminus \{w_i^k\} \cup \sigma_{i,k}^{k+1}$
10: **end for**
11: Project each function $u \in \operatorname{span} \Sigma_{\text{old}}$ onto Σ, using an L_2-projection
12: **end if**

The if-clause also ensures that whenever the refinement of $\mathscr{A}$ does not lead to the case in which some $\mathscr{B}_k \ni w_i^k \in \Sigma$ is completely supported by elements from $\mathscr{T}_{k+1}$, the basis is not changed (note, however, that the elements in $\mathscr{A}$ are nevertheless refined). In principle basis extension without replacement is a possibility, as long as the linear independence is not compromised (this would lead to singular system matrices). Allowing for the possibility to add fine functions without removing coarse functions is, in principle, possible. We will, however, not allow for it since it would require a complicated mechanism in algorithm (1) to preserve the linear independence of Σ. The refinement criteria encountered in this thesis will be solely based on function replacement as opposed to extension.
With algorithm (1) in mind, note that coarse functions may actually be partially supported by finer elements contained in $\mathscr{A}$. This does not pose a problem. Thanks to the piecewise-polynomial nature of the basis functions, functions from $\mathscr{B}_k$ are C^∞-continuous on elements from $\mathscr{T}_k$ and thus, by extension, on elements from $\mathscr{T}_{k+1}$ (the converse is not true and will lead to quadrature errors, see chapter 5).

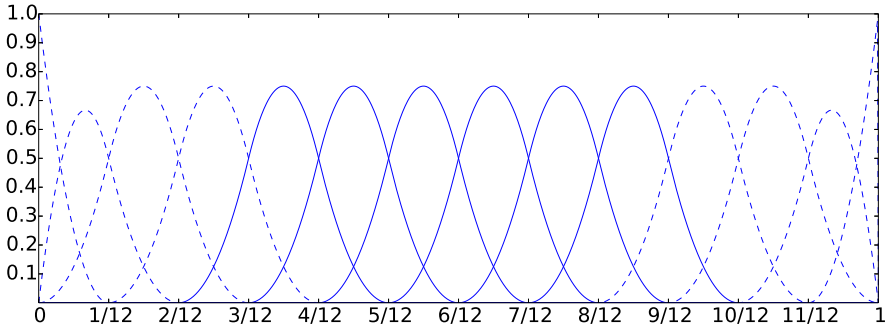
Remark. *After refining hierarchically, the basis Σ is likely to lose its partition of unity property. This is easily seen by noting that if $\sum_i w_i^1 = 1$ and w_j^1 is replaced by $w_\alpha^2, \ldots, w_\gamma^2$, then the updated basis satisfies $\sum_{i \neq j} w_i^1 + c_\alpha w_\alpha^2 + \ldots + c_\gamma w_\gamma^2 = 1$, in which, generally, $c_\alpha, \ldots, c_\gamma \neq 1$. Note that nevertheless, $1 \in \operatorname{span} \Sigma$.*
For an example of a hierarchically refined univariate basis, see figure 4.8.

As a result of the hierarchical refinement, the amount of nonzero basis functions per element usually increases. This results in an increased bandwidth of the system matrix and by that in increased assembly costs. This effect can be partially suppressed by the usage of *truncated hierarchical B-splines* (THB-splines). For more information, see [15].

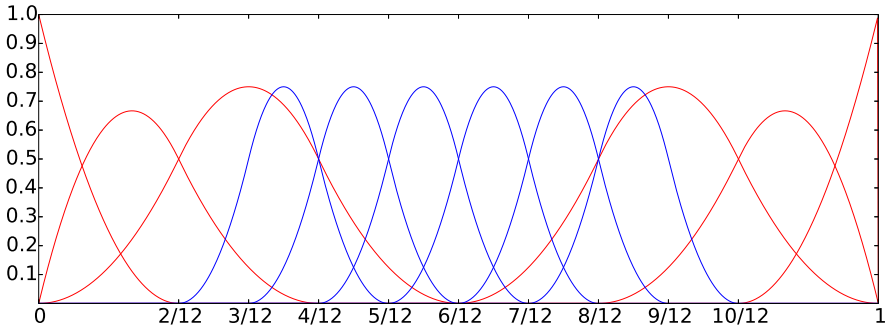
The key-features of algorithms (1) and (2) are schematically illustrated in figure 4.9



(a)



(b)



(c)

Figure 4.8: The univariate $p = 2$ bases $\mathscr{B}_1$ and $\mathscr{B}_2$ resulting from the uniform. clamped knot vectors with spacing $1/6$ and $1/12$, respectively and the hierarchically refined basis Σ resulting from the set of to be refined elements $\epsilon = \left\{[\frac{1}{6},\frac{2}{6}],[\frac{2}{6},\frac{3}{6}],[\frac{3}{6},\frac{4}{6}],[\frac{4}{6},\frac{5}{6}]\right\}$. The dotted functions in a) and the undotted ones in b) represent the functions from $\mathscr{B}_1$ and $\mathscr{B}_2$ that are completely supported by elements from $\mathscr{T}_2$ in $\mathscr{A}$ after refinement. They are removed from and added to the basis Σ initialized to $\Sigma = \mathscr{B}_1$. The boundaries of the newly formed set of elements contained in $\mathscr{A}$ which make up $\Omega = [0,1]$, are drawn on the x-axis of basis c), as well.

4.7. Example: IgA on a Monge Patch

In this section, we consider the same problem as in section 3.4, i.e.

$$\begin{cases} \partial_t U = d\Delta_{\mathscr{M}} U & \text{on } \mathscr{M} \\ \nabla U \cdot \mathbf{N}_{\partial\mathscr{M}} = 0 & \text{on } \partial\mathscr{M} \\ U(t=0) = U_0. \end{cases} \tag{4.23}$$

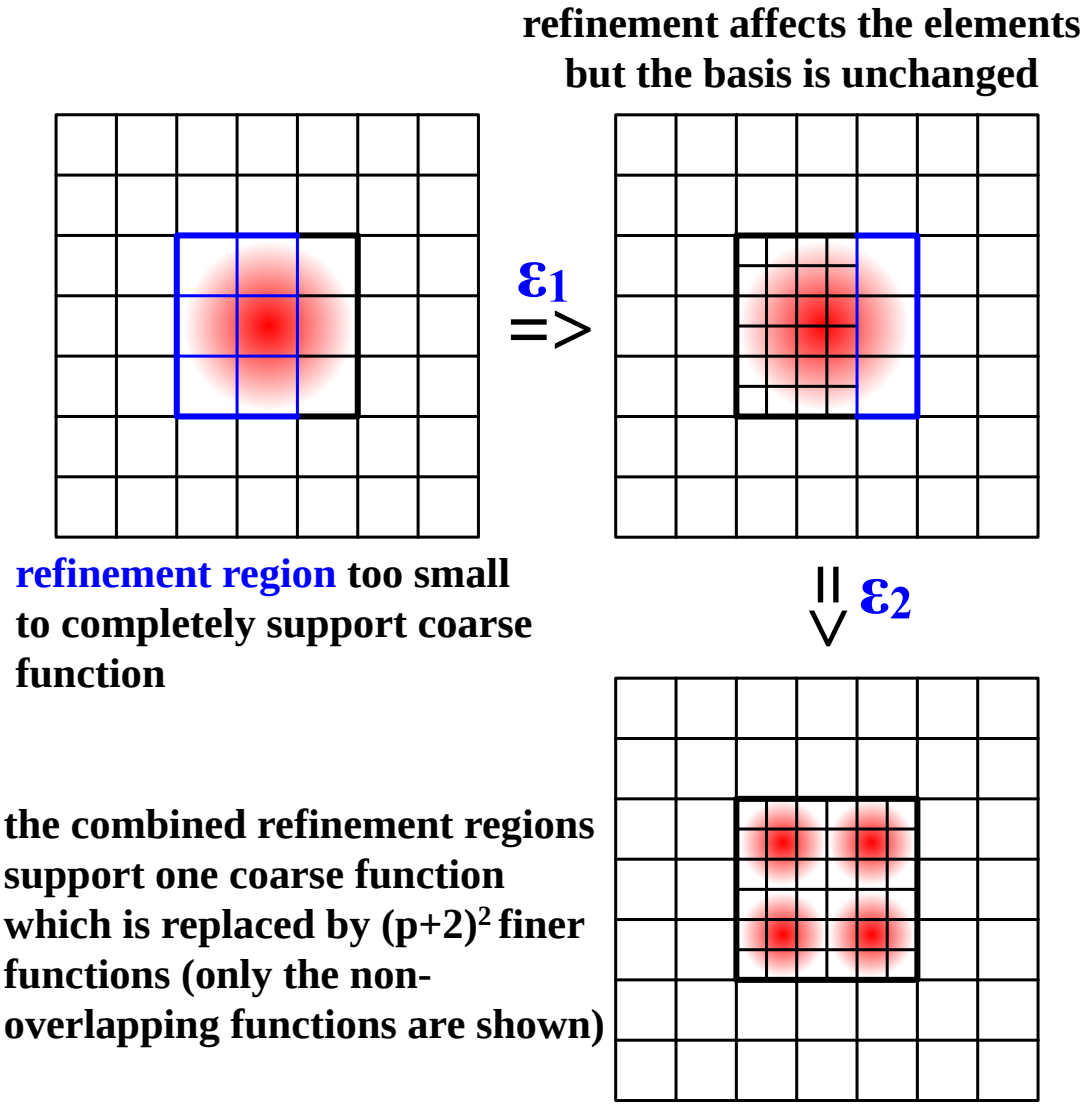


Figure 4.9: The key-features of refinement via replacement.

Again, $\mathscr{M}$ is parameterized by $\mathbf{s}^* : \Omega \to \mathscr{M}$, with

$$\mathbf{s}^*(\xi,\eta) = \begin{bmatrix} \xi \\ \eta \\ 1-(\xi-1/2)^2 \end{bmatrix}. \tag{4.24}$$

The first step, just like in section (3.4), is to derive the weak form of the implicit Euler-scheme, with boundary condition handled by partial integration

$$\left(W, U^{k+1}\right) + h_k d\left(\nabla_{\mathscr{M}} W, \nabla_{\mathscr{M}} U^{k+1}\right)_{\mathbf{2}} = \left(W, U^k\right). \tag{4.25}$$

As a next step, we introduce two uniform, clamped knot-vectors

$$\begin{aligned} \Xi_1 &= \left\{0,0,0,\tfrac{1}{10},\tfrac{2}{10},\ldots,\tfrac{9}{10},1,1,1\right\} \\ \Xi_2 &= \left\{0,0,0,\tfrac{1}{20},\tfrac{2}{20},\ldots,\tfrac{19}{20},1,1,1\right\}, \end{aligned} \tag{4.26}$$

to construct the univariate $p = 2$ bases σ_1 and σ_2.
As a next step, we construct the bivariate tensor-product bases $\mathscr{B}_1$ and $\mathscr{B}_2$ from $\sigma_1 \times \sigma_1$ and $\sigma_2 \times \sigma_2$,

respectively. We initialize the basis Σ to $\Sigma = \mathscr{B}_1$ and the elements to $\mathscr{A} = \mathscr{T}_1$. Since elements from the centering region of $\Omega = [0,1] \times [0,1]$ are larger than boundary elements when projected onto $\mathscr{M}$, we refine the region $[\frac{4}{10}, \frac{8}{10}] \times [\frac{4}{10}, \frac{8}{10}] \subset \Omega$ and update Σ. Finally, we project U_0 and $\mathbf{s}^*$ onto Σ (refined) in order to acquire the first iterand $\mathbf{c}^0$ and the mapping $\mathbf{s}$. As the components of $\mathbf{s}^*$ all have polynomial order ≤ 2, they are contained in span Σ [14]. We commence the time-stepping procedure where we iterate according to

$$\left\{[A] + h^k d[D]\right\} \mathbf{c}^{k+1} = [A]\mathbf{c}^k, \tag{4.27}$$

where

$$\begin{aligned} [A]_{i,j} &= \int_\Omega w_i w_j \sqrt{g} \mathrm{d}\boldsymbol{\xi} \\ [D]_{i,j} &= \int_\Omega \nabla w_i \cdot \nabla w_j \sqrt{g} \mathrm{d}\boldsymbol{\xi}. \end{aligned} \tag{4.28}$$

The key-steps of the IgA-approach after the weak form has been derived are outlined in algorithm (3).

Algorithm 3 IgA approach to the Monge Patch Problem

1: Construct $\mathscr{B}_1, \mathscr{T}_1$ and $\mathscr{B}_2, \mathscr{T}_2$ from $\sigma_1 \times \sigma_1$ and $\sigma_2 \times \sigma_2$, respectively.
2: $\mathscr{A} \leftarrow \mathscr{T}_1$
3: $\Sigma \leftarrow \mathscr{B}_1$
4: $\epsilon \leftarrow \{\epsilon_i^1 \in \mathscr{T}_1 \mid \epsilon_i^1 \subset [\frac{4}{10}, \frac{8}{10}] \times [\frac{4}{10}, \frac{8}{10}]$
5: call *Refine Globally*(ϵ)
6: $\mathbf{c}^0 \leftarrow$ projection of U_0 w.r.t. Σ
7: $\mathbf{s} \leftarrow$ projection of $\mathbf{s}^*$ onto Σ
8: Assemble $[A]$ and $[D]$, choose h^k and commence the time-stepping procedure

5

Computational Aspects

5.1. Integration Techniques

In order to construct the matrices and vectors involved in the weak formulation, it is necessary to evaluate an array of integrals. This can be done using the integration technique called 'Gaussian quadrature'.
We start off by introducing the concept for one-dimensional integrals, which we shall generalize to two-dimensional integrals in subsection 5.1.2.

5.1.1. Univariate Case

Let

$$Q(f) = \int_a^b f(x)\mathrm{d}x, \tag{5.1}$$

the main idea behind quadrature, in general, is to find $\{c_1, \ldots, c_n\}$ and $\{x_1, \ldots, x_n\}$ such that

$$Q(f) \simeq \sum_{i=1}^{n} c_i f(x_i). \tag{5.2}$$

Here n is called the order of the quadrature scheme. In a Gaussian quadrature scheme, the weights c_i and abscissa x_i are chosen such that (5.2) holds for polynomials up to order $2n-1$ exactly. For $a=-1$, $b=1$ and $n=2$, for instance, this means that we have to solve the system

$$\begin{cases} f=1 & \Longrightarrow & Q(f) & = & 2 & = & c_1 + c_2 \\ f=x & \Longrightarrow & Q(f) & = & 0 & = & c_1 x_1 + c_2 x_2 \\ f=x^2 & \Longrightarrow & Q(f) & = & \frac{2}{3} & = & c_1 x_1^2 + c_2 x_2^2 \\ f=x^3 & \Longrightarrow & Q(f) & = & 0 & = & c_1 x_1^3 + c_2 x_2^3 \end{cases}. \tag{5.3}$$

(5.3) is solved by

$$\begin{aligned} (c_1, c_2) &= (1, 1) \\ (x_1, x_2) &= \left(-\tfrac{1}{\sqrt{3}}, \tfrac{1}{\sqrt{3}}\right). \end{aligned} \tag{5.4}$$

Usually the system that belongs to a certain scheme is solved using a multidimensional nonlinear solver. In general, it follows from the linearity of integrals that we can evaluate the integral of any

$\leq (2n-1)$-dimensional polynomial exactly, utilizing a total of n function evaluations. For a table of coefficients for the interval $[a,b]=[-1,1]$ corresponding to values of n up to 64, see [1].
To go from the interval $[-1,1]$ to any interval $[a,b]$, one can use substitution. Utilizing the coordinate transformation $x=\frac{b-a}{2}t+\frac{b+a}{2}$, any integral of the form

$$I=\int_a^b f(x)\mathrm{d}x \tag{5.5}$$

can be brought to the form

$$I=\int_{-1}^{1} f\left(\frac{b-a}{2}t+\frac{b+a}{2}\right)\left(\frac{b-a}{2}\right)\mathrm{d}t, \tag{5.6}$$

and be tackled with a suitable quadrature scheme.
Since not all functions are polynomials, the question is how accurate the method is for non-polynomial functions. It can be shown [19] that for a function that possesses $2n$ continuous derivatives, the error is bounded by

$$\left|Q(f)-\sum_i^n c_i f(x_i)\right| \leq \frac{(b-a)^{n+1}(n!)^4}{(2n+1)[(2n)!]^3}\left|f^{(2n)}(\xi)\right|, \quad \text{for some } \xi\in(a,b), \tag{5.7}$$

which tends to decrease for increasing n.
Note that B-spline basis functions are no polynomials and may fail to possess $2n$ continuous derivatives everywhere. However, they are piecewise polynomial and therefore Gaussian quadrature can be applied piecewise and above error estimate holds on polynomial segments. In practice, the integrals are carried out element-wise, i.e. the integral over Ω is split into a sum of integrals that are then approximated utilizing the above principle. In the remainder, we will always ensure that quadrature of piecewise-polynomial functions is performed over a set elements that coincides with the polynomial segments of the integrand.

5.1.2. Bivariate Case

We shall now extend the principles from section 5.1.1 to two dimensions. Given

$$Q(f)=\int_a^b\int_c^d f(x,y)\mathrm{d}y\mathrm{d}x, \tag{5.8}$$

we first utilize substitution to transform the integral into an integral over $[-1,1]\times[-1,1]$ as follows

$$\begin{aligned} Q(f) &= \int_{-1}^{1}\int_{-1}^{1} f\left(\frac{b-a}{2}s+\frac{b+a}{2},\frac{d-c}{2}t+\frac{d+c}{2}\right)\left(\frac{b-a}{2}\right)\left(\frac{d-c}{2}\right)\mathrm{d}s\mathrm{d}t \\ &= \int_{-1}^{1}\int_{-1}^{1} g(s,t)\,\mathrm{d}s\mathrm{d}t. \end{aligned} \tag{5.9}$$

We utilize an n^{th}-order Gauss-scheme with weights $\{c_1,\ldots,c_n\}$ and abscissa $\{x_1,\ldots,x_n\}$ to approximate the inner integral

$$\begin{aligned} Q(f) &= \int_{-1}^{1}\int_{-1}^{1} g(s,t)\,\mathrm{d}s\mathrm{d}t \\ &\simeq \int_{-1}^{1}\sum_{i=1}^{n} c_i g(x_i,t)\mathrm{d}t \\ &= \sum_{i=1}^{n} c_i\int_{-1}^{1} g(x_i,t)\mathrm{d}t. \end{aligned} \tag{5.10}$$

Since $g(x_i, t) = h_i(t)$, each $h_i(t)$ can be approximated by an m^{th}-order Gauss-scheme with weights $\{d_1, \ldots, d_m\}$ and abscissa $\{y_1, \ldots, y_m\}$

$$Q(f) \simeq \sum_{i=1}^{n} \sum_{j=1}^{m} c_i d_j g(x_i, y_j). \tag{5.11}$$

It is customary to utilize the same scheme for both one-dimensional integrals. It is seen that the two-dimensional scheme has the same form as (5.2) with weights $c_i d_j$ and abscissa (x_i, y_j), where $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, m\}$.

Remark. *Suppose we are given some tensor-product Gauss-scheme over* $[-1,1]\times[-1,1]$ *with weights* $c_{i,j} = c_i d_j$ *and abscissa* (x_i, y_j)*, where* $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, m\}$*, we can see from (5.9) that converting any integral over* $[a, b] \times [c, d]$ *to an integral over* $[-1,1] \times [-1,1]$ *is equivalent to replacing*

$$c_i d_j \to c_i d_j \left(\frac{b-a}{2}\right)\left(\frac{d-c}{2}\right) \tag{5.12}$$

and

$$(x_i, y_j) \to \left(\frac{b-a}{2} x_i + \frac{b+a}{2}, \frac{d-c}{2} y_j + \frac{d+c}{2}\right), \tag{5.13}$$

and carrying the integral out over the original interval $[a, b] \times [c, d]$.
In the remainder of this chapter, whenever we present a scheme over an arbitrary quadrilateral $[a, b] \times [c, d]$*, we will assume that the weights and abscissa suit this interval through an appropriate conversion utilizing (5.12) and (5.13).*

5.2. Matrix Assembly and Choice of Linear Solver

The bivariate integration-techniques from subsection 5.1.2 can be utilized to assemble system matrices of IgA-problems. Suppose, for example, we are dealing with the geometry $\mathscr{M}$ parameterized by $\mathbf{s} : \Omega \to \mathbb{R}^3$, and we would like to project the function $U : \mathscr{M} \to \mathbb{R}$ onto the basis $\Sigma = \{w_1, \ldots, w_N\}$, utilizing the principles from section 4.4. In order to do so, we first have to assemble the *mass matrix* $[A]$ of Σ. Thus, we first have to evaluate (or at least approximate) an array of integrals of the form

$$\begin{aligned} [A]_{i,j} &= \int_{\mathscr{M}} W_i W_j \mathrm{d}\mathbf{x} \\ &= \int_{\Omega} w_i w_j \sqrt{g} \mathrm{d}\boldsymbol{\xi}. \end{aligned} \tag{5.14}$$

Of course, in any IgA-setting, the integrals have to be carried out element-wise. Thus, let $\mathscr{A} = \{\epsilon_1, \ldots, \epsilon_m\}$ be the set of elements that corresponds to the basis Σ, naively, each entry of $[A]$ could be computed in the following way

$$\forall (i, j) \in \{1, \ldots, N\} \times \{1, \ldots, N\} : \quad [A]_{i,j} = \sum_{\epsilon_k \in \mathscr{A}} \int_{\epsilon_k} w_i w_j \sqrt{g} \mathrm{d}\boldsymbol{\xi}, \tag{5.15}$$

where each integral from the sum on the right-hand side can is approximated by repeated one-dimensional Gaussian quadrature (since elements are rectangularly shaped on Ω, the limits of integration follow naturally from their vertices). This approach, however, is highly inefficient for several reasons:

- The sum is carried out over all elements even though only the elements on which both w_i and w_j are nonzero should be taken into account.
- $n \times n$ (where n is the order of the quadrature-scheme) function evaluations of $\sqrt{g}$ can be expected whenever a quadrature over element ϵ_k is carried out. Since ϵ_k is integrated over N^2 times, this makes for a total of $N^2 n^2$ function evaluations of $\sqrt{g}$ per element.
- $[A]_{i,j}$ and $[A]_{j,i}$ are computed separately even though they are equal.

It is thus recommendable to look for more efficient ways to assemble. Let an array of to-be-evaluated integrals be given

$$\forall (i,j) \in \{1,\dots,N\} \times \{1,\dots,N\}: \quad [M]_{i,j} = \int_\Omega f_{i,j}(\xi,\eta)\sqrt{g(\xi,\eta)}\mathrm{d}\boldsymbol{\xi}, \tag{5.16}$$

with $f_{i,j} = f_{j,i}$ (as for example in (5.14)).
Choosing some tensor-product Gauss-scheme with weights $c^k_{q,r}$ and abscissa (x^k_q, y^k_r) on element ϵ_k, we can proceed as follows: for each element ϵ_k we first evaluate $\sqrt{g(\xi,\eta)}$ in the abscissa (x^k_q, y^k_r) and store the result $\sqrt{g(x^k_q, y^k_r)}$ in memory. As a next step, we select all $f_{i,j}$ with $i \geq j$ that are nonvanishing on ϵ_k and evaluate them in (x^k_q, y^k_r), in order to compute

$$\int_{\epsilon_k} f_{i,j}\sqrt{g}\mathrm{d}\boldsymbol{\xi} \simeq \sum_{q,r} c^k_{q,r} f_{i,j}(x^k_q, y^k_r)\sqrt{g(x^k_q, y^k_r)}, \tag{5.17}$$

and add it to the corresponding entry in $[M]$. After looping over all elements, the matrix $[M]$ can then be *reflected* in the diagonal in order to incorporate all entries with $j > i$, even though in a sparse-matrix setting, it makes more sense to only store the upper or lower triangular entries of a symmetric matrix. The corresponding pseudocode can be found in algorithm 4. Thus, $\sqrt{g}$ is evaluated only n^2

Algorithm 4 Matrix Assembly

1: $[M] \leftarrow \emptyset^{N\times N}$
2: **for** all $\epsilon_k \in \mathscr{A}$ **do**
3: allocate memory and store $\sqrt{g(x^k_q, y^k_r)}$, $\forall (i,j) \in \{1,\dots,n\} \times \{1,\dots,n\}$
4: $\lambda \leftarrow \{f_{i,j} \mid f_{i,j} \text{ is nonvanishing on } \epsilon_k\}$
5: **for** all $f_{i,j} \in \lambda$ **do**
6: $[M]_{i,j} \leftarrow [M]_{i,j} + \sum_{q,r} c^k_{q,r} f_{i,j}(x^k_q, y^k_r)\sqrt{g(x^k_q, y^k_r)}$
7: **end for**
8: **end for**

times per element, only the $f_{i,j}$ that are nonzero on ϵ_k are looped over and $[M]_{i,j}$ is computed only for $i \geq j$. Algorithm 4 still constitutes a rather crude implementation, efficiency can be improved further using the tensor-product property of B-spline basis functions, for more information see [4].

Remark. *Since most IgA matrices contain integrals over either $W_i W_j$ or $\nabla W_i \cdot \nabla W_j$ in each entry, determining λ in line 4 of algorithm 4 is just a matter of book-keeping.*
Due to the additional factor $\sqrt{g}$ when integrating over $\mathscr{M}$, it is advisable to utilize a somewhat higher-order Gauss-scheme than the one one would normally utilize.

The sparsity of IgA matrices suggest the usage of an iterative solver. This is especially true in time-stepping methods, where the solution of the previous time-step can be utilized as an initial guess for the current time-step, to reduce the amount of iterations needed until convergence. Whenever

it can be guaranteed that the differential equation that is tackled with an IgA-approach satisfies the inner-product property, lemma 3 holds and the best choice is an CG-type algorithm [16]. Should the opposite be the case, the usage of a solver based on GMRES [35] or BiCGSTAB [39] is an option. Preconditioning is recommended to speed up the convergence.

5.3. Concluding Remarks

The assembly of right-hand side vectors can straight-forwardly be incorporated into the assembly of system matrices. After refinement of the basis Σ (see section 4.6), it is usually necessary to carry out L_2-projections onto the refined basis $\bar{\Sigma}$ in order to acquire representations of the functions in the new basis. Thus given some right-hand side function $F : \mathscr{M} \to \mathbb{R}$, we have to assemble the right-hand side vector $\mathbf{F}$ with

$$F_i = \int_\Omega \bar{w}_i f \sqrt{g} \mathrm{d}\boldsymbol{\xi}. \tag{5.18}$$

This assembly can be carried out over $\bar{\mathscr{A}}$ even though f may be nonvanishing on coarser elements in $\mathscr{A}$. This does not pose a problem since f is C^∞-continuous on each element from $\mathscr{A}$ and thus, by extension, C^∞-continuous on elements from $\bar{\mathscr{A}}$. In chapter 10, we present refinement strategies that do not allow for coarsening, which is why we can carry out the assembly over $\bar{\mathscr{A}}$ after refinement. In section 12.4, however, we will encounter a refinement-strategy that does perform refinement alongside coarsening. In this case it is of importance that a grid is selected that suits both the continuity-properties of the $\bar{w}_i$ as well as the right hand side function f in the assembly process. Details will be given in aforementioned section.

6

The Gray-Scott Reaction-Diffusion Model for Human Brain Development

Now that the theoretical foundation for the mathematical aspects of this thesis has been laid, we can proceed to introducing the actual model that will be the subject of the analysis that follows. The model was proposed in 2010 by Lefèvre et al. [25] and adopts a modified version of the Gray-Scott reaction-diffusion equations as a basic model for human brain growth. It aims to reproduce brain foldings and pattern formation occurring naturally after the 20^{th} week of fetal development. The original Gray-Scott equations are extended to include surface curvature and by a mechanism that leads to surface deformation, leading to geometric pattern formation. In order to acquaint the reader with all the key-features of this model, we shall start off by presenting the Gray-Scott reaction-diffusion model and the corresponding system of equations. After the basic features have been established, we shall modify the reaction-diffusion equations by the novel features proposed in the paper.

6.1. The Gray-Scott Reaction-Diffusion Equations

The Gray-Scott reaction-diffusion model features three chemical species u, v and p that undergo mutual conversion. Species u and v are assumed to be fed and drained by an outside source so as to keep their concentrations U and V at a reference level in the absence of mutual reaction. The reference concentration of u is given by one and the reference concentration of v is given by zero. The feeding / drainage process takes place at a constant rate F. Mathematically, the aforementioned process takes the form

$$\begin{aligned} \partial_t U &= F(1-U) \\ \partial_t V &= -FV. \end{aligned} \tag{6.1}$$

On top of the drainage through the external source, species v is assumed to be converted into species p at a constant rate H

$$v \xrightarrow{H} p. \tag{6.2}$$

This essentially acts as an additional drainage of v since p exhibits additional reaction properties with neither u nor the external source. Apart from its draining effect on v, p can and shall henceforth be disregarded in this model.
The species u and v undergo mutual transformation at a constant rate of one in the following way

$$u + 2v \to 3v. \tag{6.3}$$

Here the numerical values of H and F should be regarded in relation to the reference reaction rate at which (6.3) takes place. Assuming that the reaction takes place on a planar surface, the Gray-Scott model for human brain growth reads [25]

$$\begin{cases} \partial_t U &= d_1 \Delta U + F(1-U) - UV^2 \\ \partial_t V &= d_2 \Delta V - (F+H)V + UV^2 \end{cases}, \tag{6.4}$$

subject to initial and boundary conditions.
Here d_1 and d_2 are the diffusion constants of U and V on the surface and Δ is the ordinary two-dimensional Laplace operator. The additional terms $-UV^2$ and $+UV^2$ are a result of (6.3), while the additional drainage term $-HV$ in the governing equation for concentration V is a result of (6.2).
The function $(U,V) = (1,0)$ constitutes a linearly stable steady-state solution of (6.4) [32].
The (non-equilibrium) solutions of (6.4) are characterized by a large number of patterns for different values of F and H (alternating between low and high concentration). Pattern formation has been verified both experimentally [24] and computationally [32] and the field has broadened, greatly facilitated by computational simulation. For an overview of the typical patterns that species V exhibits for various compositions of (F,H), see figure 6.1. Patterns produced by the Gray-Scott reaction-diffusion - and similar models - resemble many patterns seen in nature. Connections have been proposed, for instance, in hair follicles [29], leaves [36], butterfly wings and mammalian coat markings [28].

6.2. Including Curvature

The reaction-diffusion process need not take place on a planar geometry. The geometry $\mathscr{M}$ can, for instance, be a curved surface parameterized by the mapping $\mathbf{s} : \Omega \to \mathscr{M}$, where $\Omega \subset \mathbb{R}^2$. As reaction-diffusion type equations commonly follow from mass-conservation considerations, the local curvature of the surface must have an effect on the system of PDEs. According to [33], in the presence of curvature, one has to replace $\Delta \to \Delta_{\mathscr{M}}$ in (6.4) (see chapter 2). Thus, in the global sense, the reaction-diffusion equation reads

$$\begin{cases} \partial_t U &= d_1 \Delta_{\mathscr{M}} U + F(1-U) - UV^2 \\ \partial_t V &= d_2 \Delta_{\mathscr{M}} V - (F+H)V + UV^2 \end{cases}, \quad \text{on } \mathscr{M}, \tag{6.5}$$

subject to initial and boundary conditions.
Above system of PDEs is easily translated to local coordinates by replacing $U \to u$ and $V \to v$.

Remark. *In the remainder, u and v will refer to the local counterparts of U and V, as opposed to the chemical species from section 6.1. We adopt the standard relation between local and global functions that we established in chapter 2.*

6.3. Including Growth

The most distinguishing feature of the model proposed by Lefèvre et al. is the inclusion of surface deformation. It is assumed that species V acts as a growth activator and U as inhibitor. To be specific, it is suggested that the species affect the mapping operator $\mathbf{s}(\xi,\eta,t)$ in the following way

$$\partial_t \mathbf{s}(\xi,\eta,t) = l\left(u(\xi,\eta,t), v(\xi,\eta,t)\right)\mathbf{n}(\xi,\eta,t), \tag{6.6}$$

where $l(u,v)$ is some growth function and $\mathbf{n}$ the unit outward normal vector in local coordinates. The most straight-forward choice for $l(u,v)$ is

$$l(u,v) = Kv, \tag{6.7}$$

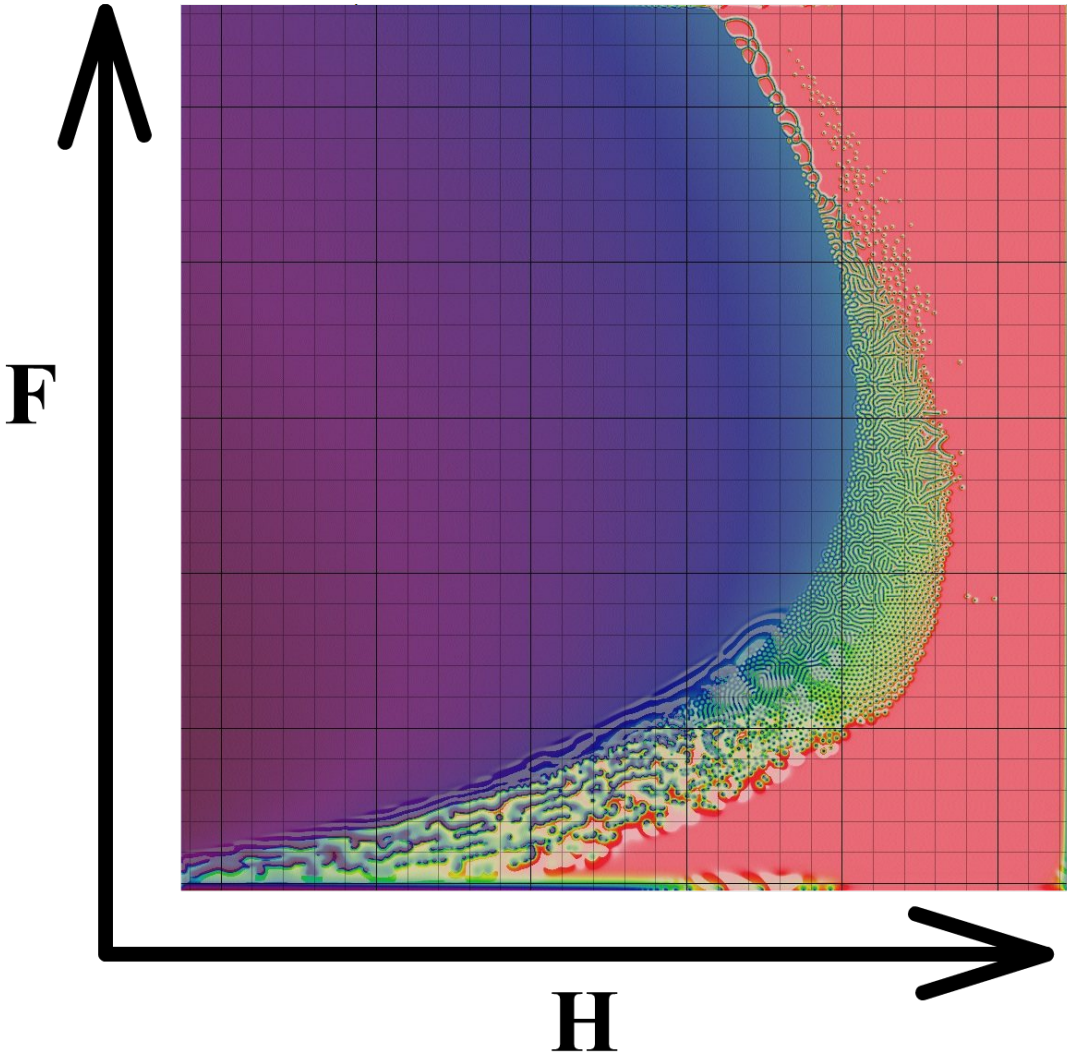


Figure 6.1: A color-coded plot of the typical kinds of patterns that V forms as a function of F and H (source: [2]). Dark regions refer to low and bright to high concentrations of V. It is seen that there exists a narrow band in the $H-F$-plane in which V exhibits a variety of different patterns. The numerical values of H and F at which this band is localed depend on the diffusion constants d_1, d_2 and the geometry on which the reaction takes place.

for some $K > 0$. Thus, with this choice of growth function, the geometry $\mathscr{M}$ will deform at places where V is nonzero, which is why V acts as a *growth activator*.

Remark. *Since U is not directly involved in the surface deformation, one might regard it as a passive* growth inhibitor. *Even though one could easily find a growth function $l(u,v)$ in which u does actively inhibit growth, in the remainder of this thesis, we will restrict out attention to the choice from (6.7).*

As **s**, in the presence of growth, becomes a time-dependent function, the geometry will receive a time-subscript $\mathscr{M} \to \mathscr{M}_t$, where $\mathscr{M}_t$ is parameterized by $\mathbf{s}(\xi,\eta,t)$. Similarly, the surface metric receives a time-indicator $[g] \to [g_t]$ as well as the Riemannian volume form $\sqrt{g} \to \sqrt{g_t}$.
Obviously, the inclusion of geometry deformation has an influence on mass-conservation considerations over control surfaces. According to [33], in equation (6.5) the governing equations for the

concentrations U and V have to be modified so as to contain an additional term. This term is, in local coordinates, given by $-u\partial_t(\ln\sqrt{g_t})$ and $-v\partial_t(\ln\sqrt{g_t})$, respectively. With that in mind, the modified equations in local coordinates read

$$\begin{cases} \partial_t u &= -u\partial_t\left(\ln\sqrt{g_t}\right) + d_1\Delta_t u + F(1-u) - uv^2 \\ \partial_t v &= -v\partial_t\left(\ln\sqrt{g_t}\right) + d_2\Delta_t v - (F+H)v + uv^2 \end{cases}, \quad \text{on } \Omega, \tag{6.8}$$

where $\Delta_t \equiv \Delta_{\mathscr{M}_t}$.
These additional terms ensure that the average concentrations decrease upon surface expansion and increase upon surface contraction. Since $-\partial_t\left(\ln\sqrt{g_t}\right) < 0$ whenever the surface locally expands and $-\partial_t\left(\ln\sqrt{g_t}\right) > 0$ whenever it (locally) contracts, the modification makes intuitive sense.
The basic idea behind the model is that patterns formed on the surface will manifest themselves in surface deformations through the extension of the model via (6.6) and (6.7). With the right choice for F and H, the patterns formed resemble the typical patterns found on the surface of human brains (see figure 6.1).
Note that we did not explicitly impose any BCs on u and v. Whenever the geometry $\mathscr{M}_t$ has no boundary for all t, there exist no explicit spatial BCs on U and V but only an initial condition. Under all circumstances, however, BCs have to be imposed on u and v. Whenever there are spatial BCs on the global concentrations, the local BCs follow straight-forwardly from an appropriate conversion into local coordinates. Whenever there are no BCs on the global concentrations, the boundary conditions of the local counterparts should be chosen so as to ensure that the function values of u and v are equal on the segments of $\partial\Omega$ that overlap on $\mathscr{M}_t$. On a domain comprised of a single patch, for example, this usually reduces to periodic boundary conditions across pairs of boundaries. Thus, local BCs follow from the way in which Ω is mapped onto $\mathscr{M}_t$ but statements can not in general be made. We will elaborate upon this topic in sections 7.4, 8.2.1 and 9.3.

7

Isogeometric Implementation

In this section, we present a general numerical scheme with which the differential equation from chapter 6 is tackled. We will not presume any specific form of Ω but derive conditions that any IgA-basis should satisfy in the most general way. Tangible applications of these principles for various initial conditions shall be the subject of chapters 8 and 9.
In the remainder of this chapter we will assume that the Riemannian volume form $\sqrt{g_t}$ satisfies the following condition: for all t, there exist strictly positive constants m_t, M_t, with $M_t > m_t$, such that $m_t \leq \sqrt{g_t} \leq M_t$.

7.1. Formulation as a System of Equations

We can formulate the system comprised of substrates U and V and geometry $\mathscr{M}_t$ parameterized by $\mathbf{s}$ as a system of equations in local coordinates

$$\partial_t \begin{bmatrix} u \\ v \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} -u\partial_t\left(\ln\sqrt{g_t}\right) + d_1\Delta_t u + F(1-u) - uv^2 \\ -v\partial_t\left(\ln\sqrt{g_t}\right) + d_2\Delta_t v - (F+H)v + uv^2 \\ K v\mathbf{n} \end{bmatrix}, \tag{7.1}$$

with

$$\mathbf{n}(\mathbf{s}) = \frac{1}{\left\| \frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right\|} \left(\frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right). \tag{7.2}$$

Defining

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ \mathbf{s} \end{bmatrix} \tag{7.3}$$

and

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} -u\partial_t\left(\ln\sqrt{g_t}\right) + d_1\Delta_t u + F(1-u) - uv^2 \\ -v\partial_t\left(\ln\sqrt{g_t}\right) + d_2\Delta_t v - (F+H)v + uv^2 \\ K v\mathbf{n} \end{bmatrix}, \tag{7.4}$$

the system of equations can be written as

$$\partial_t \mathbf{u} = \mathbf{f}(\mathbf{u}). \tag{7.5}$$

The individual components of $\mathbf{f}(\mathbf{u})$ shall be referred to as f_u, f_v and $f_\mathbf{s}$. Note that $f_\mathbf{s}$ is itself a vector-valued function.

7.2. Temporal Discretization

We integrate both sides of (7.5)

$$\begin{aligned} \int_{t^k}^{t^{k+1}} \partial_t \mathbf{u} \mathrm{d}t \quad &= \int_{t^k}^{t^{k+1}} \mathbf{f}(\mathbf{u}) \mathrm{d}t \\ \implies \quad \mathbf{u}\left(t^{k+1}\right) - \mathbf{u}\left(t^k\right) &= \int_{t^k}^{t^{k+1}} \mathbf{f}(\mathbf{u}) \mathrm{d}t. \end{aligned} \tag{7.6}$$

Before proceeding to the discretization, we define $\mathbf{u}^k$ as the approximation of $\mathbf{u}\left(t^k\right)$, resulting from the discretized scheme.
The right-hand-side integral of (7.6) is approximated by a mixed implicit / explicit quadrature. Defining $h_k = t^{k+1} - t^k$, we utilize

$$\int_{t^k}^{t^{k+1}} \mathbf{f}(u, v, \mathbf{s}) \mathrm{d}t \simeq h_k \left[\mathbf{g}(u^{k+1}, v^{k+1}, \mathbf{s}^k, \mathbf{s}^{k-1}) + \mathbf{h}(u^k, v^k, v^{k+1}, \mathbf{s}^k) \right], \tag{7.7}$$

where

$$\mathbf{g}(u^{k+1}, v^{k+1}, \mathbf{s}^k, \mathbf{s}^{k-1}) = \begin{bmatrix} -u^{k+1} \partial_t^h \left(\ln \sqrt{g_k}\right) + d_1 \Delta_k u^{k+1} - F u^{k+1} \\ -v^{k+1} \partial_t^h \left(\ln \sqrt{g_k}\right) + d_2 \Delta_k v^{k+1} - (F+H) v^{k+1} \\ \mathbf{0} \end{bmatrix} \tag{7.8}$$

and

$$\mathbf{h}(u^k, v^k, v^{k+1}, \mathbf{s}^k) = \begin{bmatrix} -u^k \left(v^k\right)^2 + F \\ u^k \left(v^k\right)^2 \\ K \mathbf{n}^k v^{k+1} \end{bmatrix}, \tag{7.9}$$

with $\mathbf{n}^k = \mathbf{n}(\mathbf{s}^k)$, $\Delta_k \equiv \Delta_{t^k}$ and $\sqrt{g_k} \equiv \sqrt{g_{t^k}}$.
The expression $\partial_t^h \left(\ln \sqrt{g_k}\right)$ represents the time-discretization of $\partial_t \left(\ln \sqrt{g_k}\right)$

$$\partial_t^h \left(\ln \sqrt{g_k}\right) = \frac{\ln \sqrt{g_k} - \ln \sqrt{g_{k-1}}}{h_{k-1}}, \tag{7.10}$$

we utilize a backward-difference scheme to avoid having to include $\sqrt{g_{k+1}}$ which is unknown at time-instance $t = t^k$. In this setting, it also becomes apparent why $\mathbf{g}$ is a function of $\mathbf{s}^k$ and $\mathbf{s}^{k-1}$: $\mathbf{s}^{k-1}$ appears in $\sqrt{g_{k-1}}$ and $\mathbf{s}^k$ in $\sqrt{g_k}$ as well as the operator Δ_k (see section 2.3).
We can cast the discretized system into the form

$$\mathbf{L}(\mathbf{u}^{k+1}) = \mathbf{u}^k + h^k \mathbf{h}\left(u^k, v^k, v^{k+1}\right), \tag{7.11}$$

where

$$\mathbf{L}\left(\mathbf{u}^{k+1}\right) = \begin{bmatrix} \left\{h_k \left[\partial_t^h \left(\ln \sqrt{g_k}\right) - d_1 \Delta_k + F\right] + 1\right\} u^{k+1} \\ \left\{h_k \left[\partial_t^h \left(\ln \sqrt{g_k}\right) - d_2 \Delta_k + F + H\right] + 1\right\} v^{k+1} \\ \mathbf{s}^{k+1} \end{bmatrix}. \tag{7.12}$$

As before, the individual components of $\mathbf{L}, \mathbf{g}$ and $\mathbf{h}$ are referred to utilizing u, v and $\mathbf{s}$ subscripts. Note that $\mathbf{L}$ satisfies

$$\mathbf{L}(a_1\mathbf{u}_1 + a_2\mathbf{u}_2) = a_1\mathbf{L}(\mathbf{u}_1) + a_2\mathbf{L}(\mathbf{u}_2) \tag{7.13}$$

and that each component of $\mathbf{L}(\mathbf{u}^{k+1})$ only depends on the corresponding component in $\mathbf{u}^{k+1}$.

7.3. Spatial Discretization

System (7.11) can and should not be solved exactly. It cannot in general be solved exactly since the operator Δ_k assumes too complicated expressions on most geometries to allow for a symbolic solution method. Another reason is the fact that $\mathbf{h}(u^k, v^k, v^{k+1}, \mathbf{s}^k)$ contains $\mathbf{n}^k$ which is a function of the derivatives of $\mathbf{s}^k$. A symbolic computation of the derivatives for arbitrary $\mathbf{s}^k$ is usually unfeasible or even impossible, for this reason, u^k and v^k and $\mathbf{s}^k$ are approximated using an IgA-approach. We first multiply by a test-function W and integrate over $\mathscr{M}_k$. Here $\mathscr{M}_k$ refers to the geometry at $t = t^k$ *before* the spatial discretization. Assuming that $\mathscr{M}_0$ constitutes a boundaryless geometry, for each of the k, $\mathbf{s}^k$ parameterizes a geometry without boundary. For the sake of brevity, we utilize $\mathrm{d}\Omega_k \equiv \sqrt{g_k}\mathrm{d}\boldsymbol{\xi}$

$$\begin{aligned}
&\int_{\mathscr{M}_k} W\mathbf{L}_u(U^{k+1})\mathrm{d}\mathbf{x} = \int_{\mathscr{M}_k} W\left[U^k + h_k\mathbf{h}_u\left(U^k, V^k\right)\right]\mathrm{d}\mathbf{x} \\
\iff\quad &h_k d_1 \int_\Omega \langle\nabla_k w, \nabla_k u^{k+1}\rangle_{g_k}\mathrm{d}\Omega_k + \int_\Omega w\left[\left(\partial_t^h\left(\ln\sqrt{g_k}\right) + F\right)h_k + 1\right]u^{k+1}\mathrm{d}\Omega_k \\
&\qquad\qquad = \int_\Omega w\left[u^k + h_k\mathbf{h}_u\left(u^k, v^k\right)\right]\mathrm{d}\Omega_k.
\end{aligned} \tag{7.14}$$

Similarly,

$$\begin{aligned}
&\int_{\mathscr{M}_k} W\mathbf{L}_v(V^{k+1})\mathrm{d}\mathbf{x} = \int_{\mathscr{M}_k} W\left[V^k + h_k\mathbf{h}_v\left(U^k, V^k\right)\right]\mathrm{d}\mathbf{x} \\
\iff\quad &h_k d_2 \int_\Omega \langle\nabla_k w, \nabla_k v^{k+1}\rangle_{g_k}\mathrm{d}\Omega_k + \int_\Omega w\left[\left(\partial_t^h\left(\ln\sqrt{g_k}\right) + F + H\right)h_k + 1\right]v^{k+1}\mathrm{d}\Omega_k \\
&\qquad\qquad = \int_\Omega w\left[v^k + h_k\mathbf{h}_v\left(u^k, v^k\right)\right]\mathrm{d}\Omega_k.
\end{aligned} \tag{7.15}$$

Here, we have made use of lemma 2. As a next step, we introduce a local basis $\Sigma = \{w_1, \ldots, w_N\}$. In (7.14) and (7.15), we successively replace $w \to w_i$ and we approximate u^k, v^k and $\mathbf{s}^k$ by elements from span Σ,

$$\begin{aligned}
u^k &= \sum_j c_j^k w_j \\
v^k &= \sum_j d_j^k w_j \\
\mathbf{s}^k &= \sum_j \mathbf{e}_j^k w_j,
\end{aligned} \tag{7.16}$$

with

$$\mathbf{e}_j^k = \begin{bmatrix} \left(e_j^1\right)^k \\ \left(e_j^2\right)^k \\ \left(e_j^3\right)^k \end{bmatrix} \equiv \begin{bmatrix} e_j^1 \\ e_j^2 \\ e_j^3 \end{bmatrix}^k . \tag{7.17}$$

Note that through the introduction of (7.16) we have implicitly replaced the exact expression of $\mathscr{M}_k$ by its discretized counterpart, parameterized by the discrete $\mathbf{s}^k$. In the following, $\mathscr{M}_k, u^k, v^k$ and $\mathbf{s}^k$ will always refer to the spatially discretized expressions. Introducing $[A]$, $[B]$, $[D]$, $\mathbf{f}^r$ and $\mathbf{w}$ with

$$\begin{aligned}
[A]_{i,j} &= \int_\Omega w_i w_j \sqrt{g_k} \mathrm{d}\boldsymbol{\xi} \\
[D]_{i,j} &= \int_\Omega \langle \nabla_k w_i, \nabla_k w_j \rangle_{g_k} \sqrt{g_k} \mathrm{d}\boldsymbol{\xi} \\
[B]_{i,j} &= \int_\Omega w_i w_j \partial_t^h \left(\ln \sqrt{g_k} \right) \sqrt{g_k} \mathrm{d}\boldsymbol{\xi} \\
(\mathbf{f}^r)_i &= \int_\Omega w_i u^k \left(v^k \right)^2 \sqrt{g_k} \mathrm{d}\boldsymbol{\xi} \\
(\mathbf{w})_i &= \int_\Omega w_i \sqrt{g_k} \mathrm{d}\boldsymbol{\xi},
\end{aligned} \tag{7.18}$$

we can construct a system of equations for $\mathbf{c}^{k+1} = \left(c_1^{k+1}, \ldots, c_N^{k+1}\right)^T$ and $\mathbf{d}^{k+1} = \left(d_1^{k+1}, \ldots, d_N^{k+1}\right)^T$. They satisfy

$$\{[A](1 + h_k F) + d_1 h_k [D] + h_k [B]\} \mathbf{c}^{k+1} = h_k F \mathbf{w} - h_k \mathbf{f}^r + [A]\mathbf{c}^k \tag{7.19}$$

and

$$\{[A](1 + h_k (F + H)) + d_2 h_k [D] + h_k [B]\} \mathbf{d}^{k+1} = h_k \mathbf{f}^r + [A]\mathbf{d}^k. \tag{7.20}$$

Note that, strictly speaking, the matrices and vectors from (7.18) should receive a time-superscript k since they are not constant but have to be recomputed after each iteration (due to the non-static nature of $\mathscr{M}_k$). For the sake of brevity, however, we shall omit this superscript and assume that the reader is aware of the time-dependence involved.
After systems (7.19) and (7.20) have been solved, we are in the position to update the geometry. Equations (7.11) and (7.12) suggest that we should update according to the relation

$$\mathbf{s}^{k+1} = \mathbf{s}^k + h_k \mathbf{h}_\mathbf{s} \left(v^{k+1}, \mathbf{s}^k \right) = \mathbf{s}^k + h_k K v^{k+1} \mathbf{n}^k. \tag{7.21}$$

For reasons mentioned at the beginning of this section, continuing the computations with this parameterization is not feasible which is why we project $\mathbf{s}^{k+1}$ onto Σ. Assuming that $\mathbf{s}^k$ is in the form of (7.16) this is accomplished by solving

$$\forall i \in \{1,2,3\}: \quad [A] \left(\mathbf{e}^i \right)^{k+1} = [A] \left(\mathbf{e}^i \right)^k + h_k K \mathbf{l}^i (v^{k+1}, \mathbf{s}^k), \tag{7.22}$$

with

$$\left(\mathbf{l}^i (v^{k+1}, \mathbf{s}^k) \right)_j = \int_\Omega w_j v^{k+1} \left(\mathbf{n}^k \right)_i \sqrt{g_k} \mathrm{d}\boldsymbol{\xi} \tag{7.23}$$

and

$$\left(\mathbf{e}^i \right)^k = \left(\left(e_1^i \right)^k, \ldots, \left(e_N^i \right)^k \right)^T. \tag{7.24}$$

After (7.22) has been solved, $\mathbf{s}^{k+1}$ is given by

$$\mathbf{s}^{k+1} = \sum_j \mathbf{e}_j^{k+1} w_j, \tag{7.25}$$

with the $\mathbf{e}_j^{k+1}$ as in (7.17).

7.4. Essential Boundary Conditions and Choice of Basis

Assuming that $\mathbf{s}^k$ parameterizes a boundaryless geometry for all k, there exist no essential spatial boundary conditions on the $W_i : \mathscr{M}_k \to \mathbb{R}$. There do, however, exist boundary conditions in Ω and some initial condition

$$\mathbf{u}(t=0) = \mathbf{i}. \tag{7.26}$$

Depending on the complexity of the individual components $\mathbf{i}_u$, $\mathbf{i}_v$ and $\mathbf{i}_\mathbf{s}$, it might be necessary to project the exact expressions onto Σ to acquire approximations from $\operatorname{span}\Sigma$ before the time-stepping procedure commences. In particular, this might be necessary for $\mathbf{i}_\mathbf{s}$ when $\mathbf{n}(\mathbf{i}_\mathbf{s})$ is not known or hard to compute. These approximations then become the first iterands u^0, v^0 and $\mathbf{s}^0$.
The boundary conditions in Ω follow from the following observation: in order for the u^k and v^k to be compatible with the weak forms from (7.14) and (7.15), their global counterparts U^k and V^k must at time instance t^k be elements from $H^1(\mathscr{M}_k)$, the Sobolev space of order one. This is accomplished by requiring that they are elements from $C^0(\mathscr{M}_k)$ everywhere and elements from $C^1(\mathscr{M}_k)$ *almost everywhere*. In this case *almost everywhere* allows for violations on subsets of $\mathscr{M}_k$ that have measure zero. These requirements can be translated to boundary conditions on Ω, which in turn enforce certain constraints on the $w_i \in \Sigma$. Before we derive these constraints, note that $\mathbf{s}^k$, is not bijective when $\mathscr{M}_k$ has no boundary. This is easily seen by noting that each point from $\partial\Omega$ has to coincide with at least one other point from $\partial\Omega$ when mapped onto the geometry $\mathscr{M}_k$, in order for it to possess no boundary.
Thus, suppose that $\gamma \subset \partial\Omega$ is an overlapping set at some time-instance t^k, i.e. it satisfies

$$\mathbf{s}^k(p_i) = P \in \mathscr{M}_k, \quad \forall\, p_i \in \gamma = \{p_1, \dots, p_n\}, \tag{7.27}$$

where γ has a cardinality of at least two. A set $\{p_1, \dots, p_n\}$ that satisfies (7.27) is referred to as a set of *'equivalent points'*. Let us define $\operatorname{int}\Omega = \Omega \setminus \partial\Omega$. Following [9], we call $U : \mathscr{M}_k \to \mathbb{R}$ a function on $\mathscr{M}_k$ if its local counterpart u satisfies:

$$u|_{\operatorname{int}\Omega} \text{ is a function on } \operatorname{int}\Omega \text{ and } u \text{ assumes equal function values on equivalent points.} \tag{7.28}$$

In order for any function $U : \mathscr{M}_k \to \mathbb{R}$ to satisfy $U \in H^1(\mathscr{M}_k)$, we require that its local counterpart satisfy

$$U \text{ is a function w.r.t. } \mathscr{M}_k \text{ and } u \in H^1(\Omega). \tag{7.29}$$

The two conditions from (7.29) ensure that there are no discontinuities on the geometry on equivalent points (see figure 7.1). The easiest way to ensure that each $\mathbf{s}^k$ parameterizes a geometry that does not possess a boundary and that each $u \in \operatorname{span}\Sigma$ satisfies (7.29) for all k, is to impose the following condition on the w_i

$$W_i \text{ is a function w.r.t. } \mathscr{M}_- \text{ and } w_i \in H^1(\Omega), \tag{7.30}$$

where $\mathscr{M}_-$ is parameterized by $\mathbf{i}_\mathbf{s}$ ($\mathscr{M}_0$ already refers to its projection onto Σ).
That (7.30) indeed produces geometries without boundary is easily seen by noting that the condition $w_i \in H^1(\Omega)$ prevents any 'holes' on $\mathscr{M}_k$, resulting from points from $\operatorname{int}\Omega$. Furthermore, the condition that U be a function on $\mathscr{M}_-$ ensures that each $\gamma \subset \partial\Omega$ that constitutes an overlapping set on the initial geometry stays an overlapping set with respect to each of the $\mathbf{s}^k$. As the initial geometry is assumed to possess no boundary, these two conditions ensure that it remains without boundary.
The definition from (7.30) will be applied in chapters 8 and 9 where they reduce to certain periodicity requirements on the w_i on segments of $\partial\Omega$. In sections 8.2.1 and 9.3, we will construct bases that satisfy (7.30) utilizing regular knot-vectors and some additional tweaking.

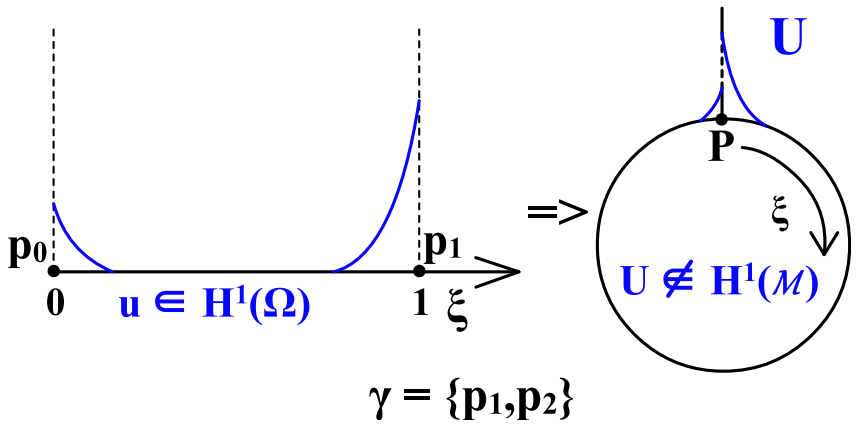


Figure 7.1: One-dimensional example of (7.29). The function u is in $H^1(\Omega)$ but this does not exclude discontinuities in its global counterpart U.

7.5. Natural Boundary Conditions

In section 7.3 we established a spatial discretization of the numerical scheme and were able to reduce the continuity requirements of the basis from second order to first (in the weak sense). This was accomplished by replacing

$$\int_{\mathcal{M}_k} W\Delta_k U \mathrm{d}\mathbf{x} \rightarrow -\int_{\mathcal{M}_k} \nabla_k W \cdot \nabla_k U \mathrm{d}\mathbf{x}, \tag{7.31}$$

under the implicit assumption that the corresponding mapping $\mathbf{s}^k$ is sufficiently smooth and parameterizes a geometry without boundary (such that lemma 2 holds).
We replaced the smooth $\mathbf{s}^k$ by an element from span Σ, which could violate the smoothness criterion (since the w_i are only required to satisfy (7.30)).
Let us assume that Ω is comprised of m patches

$$\Omega = \bigcup_{i=1}^{m} \Omega_i \tag{7.32}$$

and that we choose a basis compliant with (7.30) that satisfies

$$w_i \in C^{p-1}(\mathrm{int}\,\Omega_i), \quad \forall i \in \{1,\ldots,m\} \tag{7.33}$$

with $p \geq 2$.
Violations of the second order smoothness requirement can thus only occur on $\partial\Omega_i$. Thus, we can apply the intermediate result of lemma A.1.2

$$\int_{\mathcal{M}} W\Delta_{\mathcal{M}} U \mathrm{d}\mathbf{x} = -\int_{\mathcal{M}} \nabla_{\mathcal{M}} W \cdot \nabla_{\mathcal{M}} U \mathrm{d}\mathbf{x} + \int_{\partial\mathcal{M}} W \nabla_{\mathcal{M}} U \cdot \mathbf{N}_{\partial\mathcal{M}} \mathrm{d}l \tag{7.34}$$

to each patch individually. Let $\mathcal{M}_k^i \equiv \mathbf{s}^k|_{\Omega_i}$, we have

$$\begin{aligned}
\int_{\mathcal{M}_k} W\Delta_k U \mathrm{d}\mathbf{x} &= \sum_{i=1}^{m} \int_{\mathcal{M}_k^i} W\Delta_k U \mathrm{d}\mathbf{x} \\
&= -\sum_{i=1}^{m} \int_{\mathcal{M}_k^i} \nabla_k W \cdot \nabla_k U \mathrm{d}\mathbf{x} + \sum_{i=1}^{m} \int_{\partial\mathcal{M}_k^i} W \nabla_k U \cdot \mathbf{N}_{\partial\mathcal{M}_k^i} \mathrm{d}l \\
&= -\int_{\mathcal{M}_k} \nabla_k W \cdot \nabla_k U \mathrm{d}\mathbf{x} + \sum_{i=1}^{m} \int_{\partial\mathcal{M}_k^i} W \nabla_k U \cdot \mathbf{N}_{\partial\mathcal{M}_k^i} \mathrm{d}l.
\end{aligned} \tag{7.35}$$

Thus, we implicitly required

$$\sum_{i=1}^{m} \int_{\partial \mathscr{M}_k^i} W \nabla_k U \cdot \mathbf{N}_{\partial \mathscr{M}_k^i} \mathrm{d}l = 0, \tag{7.36}$$

which can be regarded as the weak imposition of the condition

$$\nabla_k U(\mathbf{x}) \cdot \mathbf{N}_{\partial \mathscr{M}_k^i}(\mathbf{x}) = -\nabla U_k(\mathbf{x}) \cdot \mathbf{N}_{\partial \mathscr{M}_k^j}(\mathbf{x}), \quad \text{for } \mathbf{x} \in \left(\partial \mathscr{M}_k^i \cap \partial \mathscr{M}_k^j\right), \quad i \neq j \tag{7.37}$$

that holds for diffusive processes strongly. Note that this also holds for $m = 1$ and that above condition is strongly satisfied when the basis is sufficiently smooth everywhere.

7.6. Properties of the Numerical Scheme

In this section we will present an order-error for the temporal discretization, as well as a condition under which lemma 4 holds for the spatial discretization.

7.6.1. Temporal

In equation (7.7), we introduced the temporal discretization of the right-hand side term $\mathbf{f}(u, v, \mathbf{s})$ with

$$\begin{aligned}
\int_{t^k}^{t^{k+1}} \mathbf{f}_u \mathrm{d}t &\simeq h_k \left[-u^{k+1} \partial_t^h \left(\ln \sqrt{g_k} \right) + d_1 \Delta_k u^{k+1} - F u^{k+1} - u^k \left(v^k \right)^2 + F \right] \\
\int_{t^k}^{t^{k+1}} \mathbf{f}_v \mathrm{d}t &\simeq h_k \left[-v^{k+1} \partial_t^h \left(\ln \sqrt{g_k} \right) + d_1 \Delta_k v^{k+1} - (F+H) v^{k+1} + u^k \left(v^k \right)^2 \right] \\
\int_{t^k}^{t^{k+1}} \mathbf{f}_{\mathbf{s}} \mathrm{d}t &\simeq h_k K \mathbf{n}^k v^{k+1}.
\end{aligned} \tag{7.38}$$

In appendix A.4, we derive the local truncation errors τ_k for the individual components of the temporal discretization. They are given by

$$\begin{aligned}
\tau_k(\mathbf{f}_u) &= \mathscr{O}(h_k) + \mathscr{O}(h_{k-1}) \\
\tau_k(\mathbf{f}_v) &= \mathscr{O}(h_k) + \mathscr{O}(h_{k-1}) \\
\tau_k(\mathbf{f}_{\mathbf{s}})_i &= \mathscr{O}(h_k)
\end{aligned} \tag{7.39}$$

The global error is of the same order [37].

7.6.2. Spatial

In section (7.3), we have derived the weak forms for u^{k+1} and v^{k+1}. We have done so by multiplying by a test-function and integrating over $\mathscr{M}_k$. Choosing $\mathscr{M}_k$ instead of $\mathscr{M}_{k+1}$ is a necessity since $\mathscr{M}_{k+1}$ is unknown at time-instance t^k ($\mathbf{s}^{k+1}$ has not yet been solved for). Assuming that $\mathbf{L}_u$ and $\mathbf{L}_v$ are inner-product operators, according to lemma 4, the approximate solutions u^{k+1} and v^{k+1} constitute the best approximation from span Σ to the exact solutions with respect to the $\mathbf{L}_u(\mathscr{M}_k)$- and $\mathbf{L}_v(\mathscr{M}_k)$-norm, respectively. Since $\mathbf{L}$ is obviously an operator whose components satisfy the linearity as well as the symmetry requirement, the inner-product property rests on positive-definiteness. We now derive a (not necessarily sharp) condition for positive definiteness of the individual components of

L.
We have

$$\begin{aligned}\int_{M_k} U\mathbf{L}_u(U)\mathrm{d}\mathbf{x} &= (h_kF+1)\int_\Omega u^2\mathrm{d}\Omega_k + h_k d_1\int_\Omega \langle\nabla_k u,\nabla_k u\rangle_{g_k}\mathrm{d}\Omega_k + h_k\int_\Omega \partial_t^h\left(\ln\sqrt{g_k}\right)u^2\mathrm{d}\Omega_k\\ &\geq (h_kF+1)\int_\Omega u^2\mathrm{d}\Omega_k + h_k\int_\Omega \partial_t^h\left(\ln\sqrt{g_k}\right)u^2\mathrm{d}\Omega_k\\ &= (h_kF+1)\int\limits_{\operatorname{supp} u} u^2\mathrm{d}\Omega_k + h_k\int\limits_{\operatorname{supp} u}\partial_t^h\left(\ln\sqrt{g_k}\right)u^2\mathrm{d}\Omega_k. \end{aligned} \tag{7.40}$$

Thus, if $\operatorname{supp} u \neq \emptyset$, we need to require

$$\int\limits_{\operatorname{supp} u}\left(h_k\partial_t^h\left(\ln\sqrt{g_k}\right)+h_kF+1\right)u^2\sqrt{g_k}\mathrm{d}\boldsymbol{\xi} > 0. \tag{7.41}$$

Since $u^2\sqrt{g_k} > 0$ on $\operatorname{supp} u$, a sufficient condition for (7.41) to hold is

$$\partial_t^h\left(\ln\sqrt{g_k}\right) > -\frac{1}{h_k} - F, \tag{7.42}$$

pointwise.
A similar analysis leads to the v-counterpart of (7.42)

$$\partial_t^h\left(\ln\sqrt{g_k}\right) > -\frac{1}{h_k} - F - H, \tag{7.43}$$

pointwise.
Exact statements, concerning the conditions under which (7.43) is satisfied are difficult to make in general. However, assuming that K in (7.9) satisfies $K \ll F+H$, in conjunction with the fact that the outward orientation of $\mathbf{n}$ is likely to lead to local growth of $\sqrt{g_k}(\xi,\eta)$ (and thus of $\ln\sqrt{g_k}$), it is reasonable to assume that (7.42) and (7.43) is not violated.
This suggests the usage of a CG-type algorithm to tackle (7.19) and (7.20), as well as each of the projections from (7.22). Of course it is advisable to utilize the solution vector of the previous time-step as initial guess in order to speed up the convergence.

Remark. *Should (7.43) or (7.43) be violated at some time-instance $t = t^k$, this does not necessarily imply that (7.41) is violated over the finite IgA basis Σ.*

7.7. Time-Step Selection

So far we have not discussed how to choose the time-step h_k. Since the system-matrices have to be rebuilt after each iteration, the time-step selection should be based on the following principles:

- as large as possible with respect to numerical stability
- small enough with respect to the characteristic time-scale of the equation to warrant numerical accuracy
- time-step selection should be a cheap operation.

With above principles in mind, time-step selection based on the PID-controller proposed by Valli et al. [38] constitutes a proficient choice. The basic ingredient of this PID-controller is a recursive

time-step selection based on the following scheme

$$h_{k+1} = \left(\frac{e_{k-1}}{e_k}\right)^{k_P} \left(\frac{1}{e_k}\right)^{k_I} \left(\frac{e_{k-1}^2}{e_k e_{k-2}}\right)^{k_D} h_k, \tag{7.44}$$

where

$$e_k = \max\left(e_u, e_v, e_1, e_2, e_3\right), \tag{7.45}$$

and

$$\begin{aligned} e_u &= \frac{e_u^*}{tol} & e_u^* &= \frac{\left\| u^{k+1} - u^k \right\|_2}{\left\| u^{k+1} \right\|_2} \\ e_v &= \frac{e_v^*}{tol} & e_v^* &= \frac{\left\| v^{k+1} - v^k \right\|_2}{\left\| v^{k+1} \right\|_2} \\ e_i &= \frac{e_i^*}{tol} & e_i^* &= \frac{\left\| s_i^{k+1} - s_i^k \right\|_2}{\left\| s_i^{k+1} \right\|_2}. \end{aligned} \tag{7.46}$$

Whenever $K \ll 1$, it is reasonable to remove the $e_i, i \in \{1,2,3\}$ from (7.45) and base the time-step selection purely on e_u and e_v. Here tol constitutes a tolerance to which the relative change in norm (of u^k or v^k) between two iterations is tuned. k_P, k_I and k_D are positive constants. The recursive selection scheme (7.44) is mostly heuristical and serves the purpose to ensure that e_n gets as close as possible to but does not exceed tol. Choosing the value of tol sufficiently small ensures that the time-step is appropriate for the characteristic time-scale of the problem and that no numerical instabilities occur (in which case the relative change would exceed tol as well). Finally, time-step selection is cheap since we can utilize

$$\begin{aligned} \left\| u^k - u^{k-1} \right\|_2^2 &= \int_{\mathcal{M}_k} \left(U^k - U^{k-1}\right)^2 \mathrm{d}\mathbf{x} \\ &= \left(\mathbf{c}^k - \mathbf{c}^{k-1}\right)^T [A] \left(\mathbf{c}^k - \mathbf{c}^{k-1}\right), \end{aligned} \tag{7.47}$$

and similar expressions for the other norms involved. The overall time-stepping scheme is then comprised of four parts [21, p. 275]

The *PID-controller* by Valli et al.

1. Carry out the k^{th} time-step and compute the relative changes of the chosen indicator variable e_k.
2. If $e_k > tol$, reject $\mathbf{u}^k$ and recompute it using

$$h_k \leftarrow \frac{tol}{e_k} h_k.$$

3. Adjust the time-step for the next iteration *smoothly* using

$$h_{k+1} = \left(\frac{e_{k-1}}{e_k}\right)^{k_P} \left(\frac{1}{e_k}\right)^{k_I} \left(\frac{e_{k-1}^2}{e_k e_{k-2}}\right)^{k_D} h_k.$$

4. Limit the growth and reduction of the time-step so that

$$h_{\min} \le h_{k+1} \le h_{\max} \qquad m \le \frac{h_{k+1}}{h_k} \le M.$$

Adjusting the time-step smoothly as in (7.44) also aims to ensure that the rejection of $\mathbf{u}^k$ is a seldom occurrence. The default values of the PID parameters are given by $K_P = 0.075$, $k_I = 0.175$ and $k_D = 0.01$, which we shall utilize throughout the remaining chapters.

8

Implementation on a Torus

Now that we have presented a general isogeometric implementation and established the requirements that the basis functions have to satisfy, we can commence with an explicit implementation. Within the framework of human brain development, a sphere-like shaped initial geometry is the obvious choice. As a sphere constitutes a challenging geometry in any discrete setting, it is advisable to start with a simpler geometry. The torus constitutes a good starting point since it can be parameterized from a single patch and its parameterization does not contain any singularities (unlike the sphere from a single patch).

8.1. Constructing a Torus

As parametric domain, we choose $\Omega = [0,1] \times [0,1]$ and parameterize the torus via

$$\mathbf{i_s} = \begin{bmatrix} \big(R_1 + R_2 \cos(2\pi\eta - \pi)\big) \cos(2\pi\xi - \pi) \\ \big(R_1 + R_2 \cos(2\pi\eta - \pi)\big) \sin(2\pi\xi - \pi) \\ R_2 \sin(2\pi\eta - \pi) \end{bmatrix}. \tag{8.1}$$

Here R_1 constitutes the inner radius of the torus and R_2 the outer. It is advisable to initialize the radii to values that satisfy $R_1 > CR_2$, for some $C > 1$. In fact, it is recommended to choose $C > 2$ since that reduces the chances of the geometry clashing with itself at later stages of the growth process.
As a next step, we will characterize the overlapping sets of $\mathbf{i_s}$. The parameterization of the torus can be regarded as a two-step process, step one glues together the western and eastern boundaries of Ω to form an open cylinder and step two fuses the two open ends together to form the torus $\mathscr{M}_-$. Thus, the western and eastern boundaries overlap on $\mathscr{M}_-$, as well as the northern and southern. Hence, the collection Γ_T of all overlapping sets is given by the collection of pairs of points that satisfy

$$\Gamma_T = \left\{ \{p_1, p_2\} \subset \Omega \quad | \quad \left(p_1 = (0,s)^T \wedge p_2 = (1,s)^T\right) \vee \left(p_1 = (s,0)^T \wedge p_2 = (s,1)^T\right), \quad s \in [0,1] \right\}. \tag{8.2}$$

Thus, on the torus we are essential dealing with periodic boundary conditions across the western and eastern, and southern and northern boundaries.
As stated in chapter 7, it is advisable to project the initial geometry onto a suitable IgA basis. Constructing bases that are compliant with (7.30) will be the subject of the next section.

8.2. Constructing a Basis

Now that we have introduced the parameterization of the torus and the collection of all overlapping sets with respect to that parameterization, we are in the position to construct a basis that satisfies (7.30). The basic ingredient of this basis shall be a set of bivariate tensor-product B-spline functions (see chapter 4), that we shall slightly modify in order to satisfy (7.30).

8.2.1. Utilizing Clamped Knot Vectors

The most obvious choice is a basis based on clamped knot vectors. Given some order $p \geq 1$ and $n \geq p+1$, we introduce the uniform, clamped knot vector

$$\Xi_{n,p} = \left\{ \underbrace{0,\ldots,0}_{p+1 \text{ times}}, \underbrace{\frac{1}{n-p},\ldots,\frac{n-p-1}{n-p}}_{n-p-1 \text{ terms}}, \underbrace{1,\ldots,1}_{p+1 \text{ times}} \right\}, \tag{8.3}$$

to construct the univariate basis

$$\sigma_{n,p} = \{N_{1,p},\ldots,N_{n,p}\}. \tag{8.4}$$

The univariate basis resulting from $\Xi_{13,3}$ is depicted in figure 8.1. By γ^1, γ^2, γ^3 and γ^4, we denote the

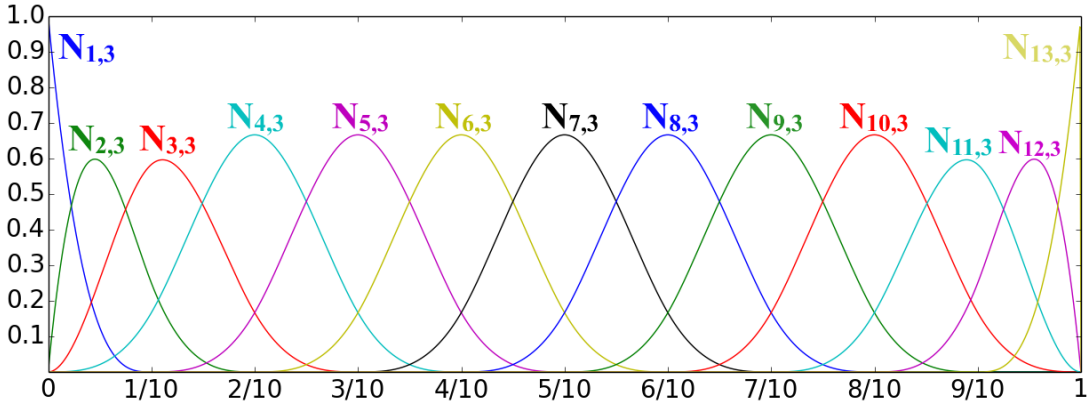


Figure 8.1: An univariate basis resulting from $\Xi_{13,3}$

southern, eastern, northern and western boundaries of Ω. In the previous section we discovered that we are essentially dealing with periodic boundary conditions across γ^1 and γ^3, as well as γ^2 and γ^4. Thus, in order to construct a basis compliant with (7.30), we have to construct basis functions that are periodic across these segments, as well. As a result of the relatively simple topology of the torus, this can be done on a univariate level. To be specific, the only thing we have to require is that $N_{1,p}$ and $N_{n,p}$ be combined into one function, i.e.

$$N_{1,p} \rightarrow N_{1,p} + N_{n,p}. \tag{8.5}$$

The univariate basis $\sigma^*_{n,p} = \{N_{1,p},\ldots,N_{n-1,p}\}$, with $N_{1,p}$ as in (8.5) (see figure 8.2) can then be utilized to construct the bivariate tensor-product basis $\mathscr{B}_1$ from $\sigma^*_{n,p} \times \sigma^*_{n,p}$.

Remark. *Being able to couple in the univariate basis as opposed to the bivariate basis is a result of the relatively simple topology of the torus. We will see in chapter 9 that coupling is usually a more laborious task on more complicated domains in which case it might not be possible on a*

univariate level.
The idea behind coupling on a univariate level is schematically depicted in figure 8.3.

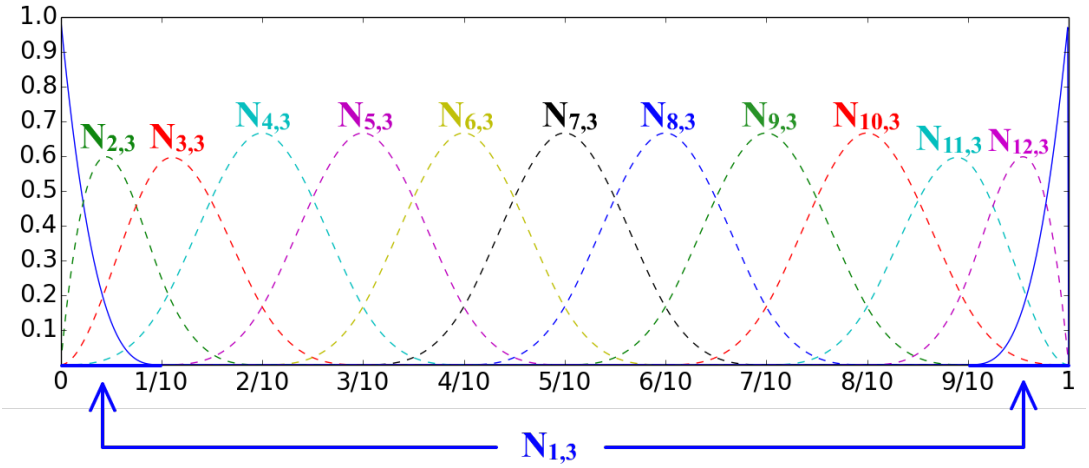


Figure 8.2: The coupled counterpart $\sigma^*_{n,p}$ of the univariate basis $\sigma_{n,p}$ constructed from $\Xi_{13,3}$. The dotted functions are left unaffected by the coupling process but it is seen that $N_{1,3}$ and $N_{13,3}$ have been combined to form one function supported by elements $\epsilon_1 = [0, \frac{1}{10}]$ and $\epsilon_{10} = [\frac{9}{10}, 1]$.

The basis $\mathscr{B}_1$, containing the coupled functions then constitutes the basis to which Σ is initialized. Similarly, the set of elements $\mathscr{A}$ is initialized to $\mathscr{T}_1$ whose elements are simply given by the tensor-product elements resulting from $\Xi_{n,p} \times \Xi_{n,p}$ (without knot repetitions). Acquiring hierarchically finer bases $\mathscr{B}_2, \mathscr{B}_3, \ldots$ and grids $\mathscr{T}_1, \mathscr{T}_2, \ldots$ is accomplished by repeating above steps with $n \to 2n-p$, $n \to 4n-3p$, etc.

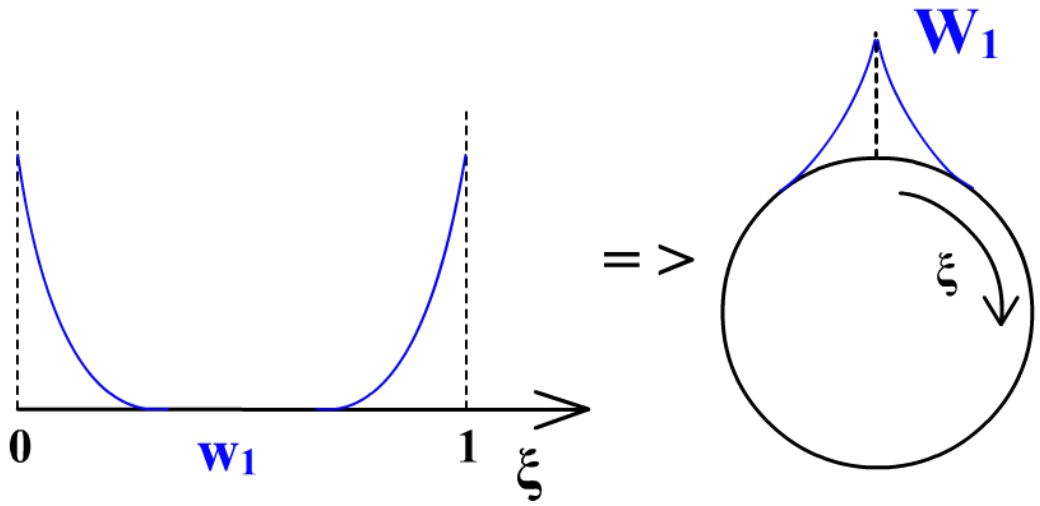


Figure 8.3: One-dimensional, schematic representation of the idea behind function coupling. It is seen that the coupling in w_1 prevents any discontinuities of W_1 at the spot where the two boundaries of $\Omega = [0,1]$ connect on the circle. In the absence of coupling there would have been a discontinuity there which we avoided by having w_1 satisfy (7.30). Note that this one-dimensional example on the circle can be regarded as a cross-sectional representation of the parametric coupling in one direction on the torus.

8.2.2. Higher Order Continuity

The basis presented in section 8.2.1 satisfies the condition from (7.30) but is only C^0-continuous at the places where the boundaries of Ω connect on $\mathscr{M}_-$. Achieving higher order continuity is generally a challenging task, however, in the case of the torus it can be done with relative ease. We have learned in section 8.2.1 that coupling on a univariate level is equivalent to coupling on a bivariate level. We shall take advantage of this observation and present a method to construct bases that exhibit C^{p-1} continuity everywhere. Again, we shall choose some fixed n and $p > 1$ and introduce the uniform knot-vector

$$\mathscr{H}_{n,p} = \left\{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n+p}{n}\right\}. \tag{8.6}$$

An example of the basis $\sigma_{n,p}$ that $\mathscr{H}_{n,p}$ produces is depicted in figure (8.4). Note that each function has at least $p-1$ continuous derivatives at each point. As usual, we are aiming at expressing the

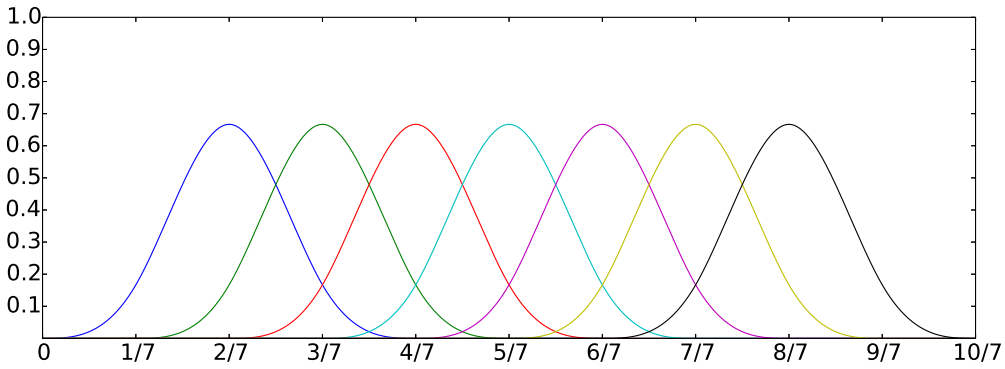


Figure 8.4: The univariate basis resulting from $\mathscr{H}_{7,3}$

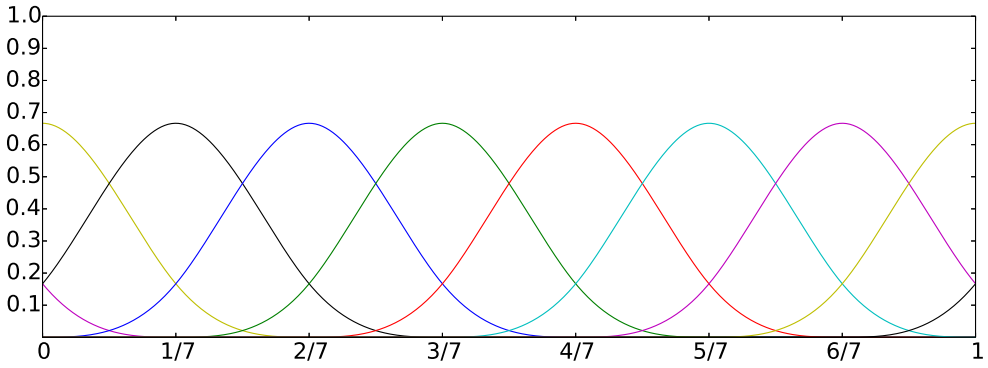


Figure 8.5: The duplicated counterpart of the basis depicted in figure (8.4). The functions are given in various colors without repetitions. It is seen that there are three (= p) functions that are cut off at one interval boundary and continue at the other boundary in a periodic fashion.

basis $\sigma_{n,p}$ on the interval $[0,1]$. However, it is seen that $\mathscr{H}_{n,p}$ does not end on one. The idea is to cut off the part that protrudes $\xi_{n+1} = 1$ and have it continue at $\xi = 0$. In fact, it is not even necessary to cut, the only thing we have to do in $\sigma_{n,p} = \{N_{1,p}, \ldots, N_{n,p}\}$ is to require

$$\forall N_{i,p} \in \sigma_{n,p}: \quad N_{i,p}(\xi) \to N_{i,p}(\xi) + N_{i,p}(\xi+1). \tag{8.7}$$

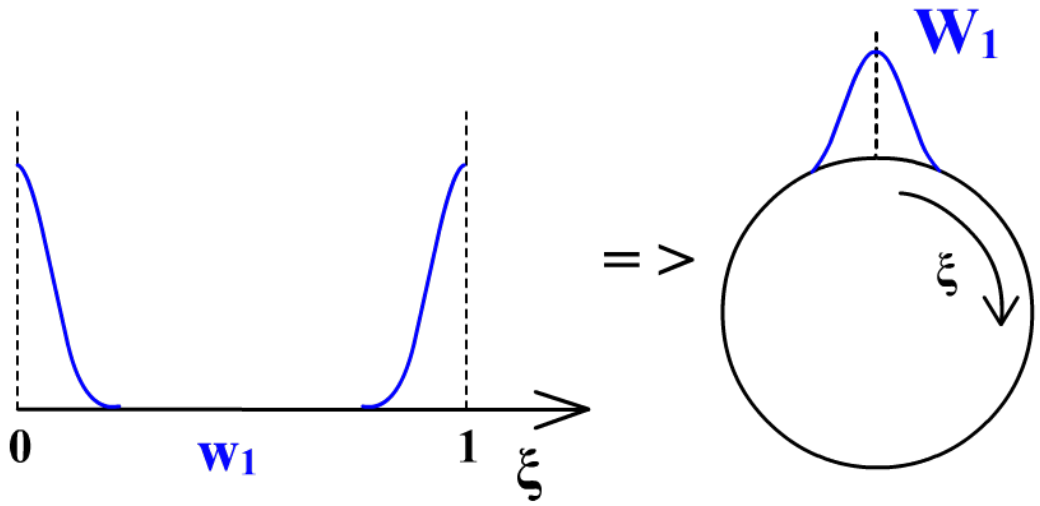


Figure 8.6: One-dimensional, schematic representation of the effect that the shifted duplication has on a global level.

Restricting our attention to $\xi \in [0,1]$, this shifted duplication results in a basis that satisfies periodic boundary conditions on the parametric interval $[0,1]$. It is seen that the $N_{i,p} \in \sigma_{n,p}$ inherit the C^{p-1}-continuity they possess on $[0,1]$ whenever the interval is mapped onto a smooth and closed curve (a circle, for instance). For an example of the duplicated counterpart of the basis depicted in figure 8.4, see figure 8.5. The higher-order continuity counterpart of figure 8.3 is depicted in figure 8.6.
As before, the basis to which Σ is initialized is given by $\mathscr{B}_1 = \sigma_{n,p} \times \sigma_{n,p}$ and $\mathscr{A}$ is initialized to $\mathscr{T}_1$. Finer bases and grids can be acquired by repeating above steps with $n \to 2n$, $2n \to 4n$ and so on.
The torus from (8.1), after projection onto a basis with $p = 3$ and $n = 50$, is depicted in figure 8.7.

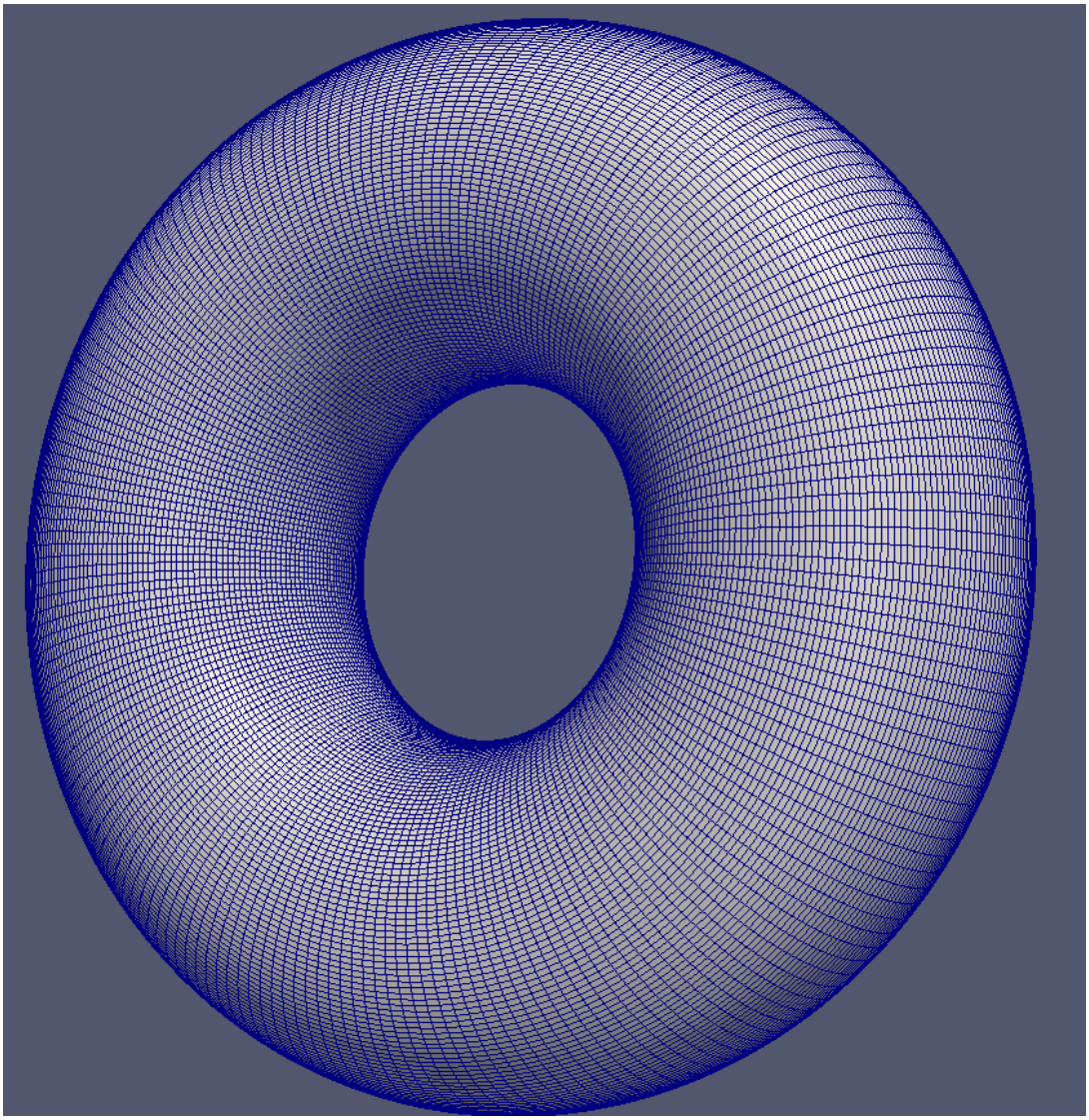


Figure 8.7: The torus resulting from a projection onto a smooth IgA-basis. The grid has been turned on in order to demonstrate the smoothness of the projection. There are no fringing effects by the segments where patch boundaries meet. The grid lines form smooth curves which would not have been the case when using the basis from section 8.2.1.

9

Implementation on a Sphere-Like Shaped Initial Geometry

The implementation on a torus is relatively simple but within the context of brain pattern formation, unrealistic. We would like a sphere-like shaped initial geometry on which pattern-formation and growth will lead to a final geometry that more closely resembles a real brain and its folds. The usual way to achieve a sphere-like shaped initial geometry $\mathscr{M}_-$ from a parametric domain given by $\Omega = [0,1] \times [0,1]$ is to utilize the parameterization $\mathbf{i}_s : \Omega \to \mathscr{M}_-$

$$\mathbf{i}_s = \begin{bmatrix} R\sin(\pi\eta)\cos(2\pi\xi) \\ R\sin(\pi\eta)\sin(2\pi\xi) \\ R\cos(\pi\eta) \end{bmatrix} \tag{9.1}$$

and map the quadrilateral Ω onto a sphere of radius R.
This parametrization, however, poses numerous problems, for instance

- the segments of $\partial\Omega$, given by $\gamma^1 = \{(\xi,0) : \xi \in [0,1]\}$ and $\gamma^3 = \{(\xi,1) : \xi \in [0,1]\}$ are contracted into the points $(0,0,R) \in \mathscr{M}_0$ and $(0,0,-R) \in \mathscr{M}_0$, respectively. This leads to a singularity in the metric at these points. In chapter 7 we required that $\sqrt{g_t}$ contain no singularities.
- The cells corresponding to a uniform grid on Ω become increasingly small near the poles $(0,0,R)$ and $(0,0,-R)$ on $\mathscr{M}_0$. This effect intensifies on finer grids.
- The singularities will lead to issues in the quadrature of certain functions.

Overall, it is seen that the drawbacks of this approach outweigh the advantages, which is why we shall choose for a different approach.

9.1. Multipatch Approach

Instead of utilizing a parametric domain comprised of a single patch of the form $\Omega = [0,1] \times [0,1]$, we shall now utilize six patches. To be specific, we shall choose

$$\Omega = \bigcup_{i=1}^{6} \Omega_i. \tag{9.2}$$

Defining

$$\begin{aligned}
\mathbf{q}^1 &= (0,0)^T & \mathbf{q}^2 &= (1,0)^T \\
\mathbf{q}^3 &= (1,-1)^T & \mathbf{q}^4 &= (1,1)^T \\
\mathbf{q}^5 &= (2,0)^T & \mathbf{q}^6 &= (3,0)^T,
\end{aligned} \tag{9.3}$$

patch Ω_i is given by

$$\Omega_i = [q_1^i, q_1^i + 1] \times [q_2^i, q_2^i + 1]. \tag{9.4}$$

The parametric domain Ω resulting from 9.2 is depicted in figure 9.1.

It is easily seen that Ω can be 'folded' into a (hollow) cube Ω^* with faces at -1 and 1 in either di-

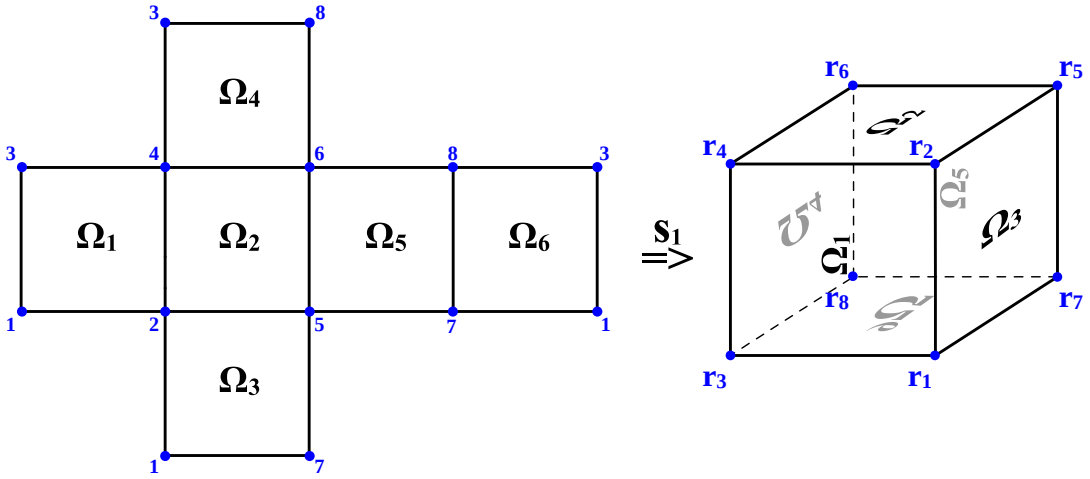


Figure 9.1: The parametric domain and the cube onto which it is mapped via linear interpolation.

rection by performing a set of elementary translation and rotation operations on the patches $\Omega_i, i \in \{1,\ldots,6\}$. We shall, however, utilize a linear isogeometric basis, constructed from the knot-vector $\Xi_\square = \{0,0,1,1\}$ in order to construct the cube Ω^* from Ω. To be specific, we construct the linear basis $\{N_1^\square(\xi), N_2^\square(\xi)\}$ utilizing formulae (4.1) and (4.2) with $\Xi = \Xi_\square$ and $p = 1$ and construct the two-dimensional tensor-product basis $\Sigma^\square \equiv \{w_1^\square, w_2^\square, w_3^\square, w_4^\square\}$ with

$$\{w_1^\square, w_2^\square, w_3^\square, w_4^\square\} = \{N_1^\square(\xi)N_1^\square(\eta), N_2^\square(\xi)N_1^\square(\eta), N_1^\square(\xi)N_2^\square(\eta), N_2^\square(\xi)N_2^\square(\eta)\}. \tag{9.5}$$

As a next step, we assign a number to each of the eight vertices of the cube Ω^*: $\mathbf{r}_i, i \in \{1,\ldots,8\}$ and characterize each patch Ω_i by a patch-vector $\mathbf{p}^i$ that contains the indices of the patch vertices that become its corner-points on Ω^*. We define the $\mathbf{p}^i$ as follows:

$$\begin{aligned}
\mathbf{p}^1 &= (1,2,4,3)^T & \mathbf{p}^2 &= (2,5,6,4)^T \\
\mathbf{p}^3 &= (1,7,5,2)^T & \mathbf{p}^4 &= (4,6,8,3)^T \\
\mathbf{p}^5 &= (5,7,8,6)^T & \mathbf{p}^6 &= (7,1,3,8)^T.
\end{aligned} \tag{9.6}$$

The cube vertices are given by

$$\begin{aligned}
\mathbf{r}_1 &= (1,-1,-1)^T & \mathbf{r}_2 &= (1,-1,1)^T \\
\mathbf{r}_3 &= (-1,-1,-1)^T & \mathbf{r}_4 &= (-1,-1,1)^T \\
\mathbf{r}_5 &= (1,1,1)^T & \mathbf{r}_6 &= (-1,1,1)^T \\
\mathbf{r}_7 &= (1,1,-1)^T & \mathbf{r}_8 &= (-1,1,-1)^T.
\end{aligned} \tag{9.7}$$

The patch vector $\mathbf{p}^i = (a_1, a_2, a_3, a_4)^T$ serves as to indicate that the vertices of patch Ω_i (in counter-clockwise direction) correspond to the points $\{\mathbf{r}_{a_1}, \mathbf{r}_{a_2}, \mathbf{r}_{a_3}, \mathbf{r}_{a_4}\}$ on Ω^*.
The mapping $\mathbf{s}_1 : \Omega \to \Omega^*$ that maps points from the parametric domain Ω onto the cube Ω^* is constructed via linear interpolation, utilizing the basis $\Sigma^{\square}$, the $\mathbf{q}^1, \ldots, \mathbf{q}^6$, the patch vectors $\mathbf{p}^1, \ldots, \mathbf{p}^6$ and the vertices $\mathbf{r}_1, \ldots, \mathbf{r}_8$. It is given by

$$\mathbf{s}_1(\xi, \eta) = \sum_{i=1}^{6} \sum_{j=1}^{4} w_j(\xi - q_1^i, \eta - q_2^i)\mathbf{r}_{p_j^i}. \tag{9.8}$$

The free variables of Ω^* are given by $(x, y, z)^T$.
It is easily seen that (9.8) is orientation-preserving. Therefore, the normal vector on the cube satisfies

$$\mathbf{n}^* = \frac{1}{\left\| \frac{\partial \mathbf{s}_1}{\partial \xi} \times \frac{\partial \mathbf{s}_1}{\partial \eta} \right\|} \left(\frac{\partial \mathbf{s}_1}{\partial \xi} \times \frac{\partial \mathbf{s}_1}{\partial \eta} \right). \tag{9.9}$$

Remark. *One may choose not to build a global topology as in (9.2) but regard the six patches as separate spaces with their interactions established through the 'domain manifold'. Whenever an integral over $\mathcal{M}_k$ is carried out, the integral over Ω with the Riemannian volume form $\sqrt{g}$ is replaced by a sum over the Ω_i with the local volume form $\sqrt{g^i}$ of each individual patch. For details, see [9].*
We will continue with the single domain as it fits well into the framework of the previous chapters, where we exclusively discussed settings with one parametric domain.

9.2. Constructing a Sphere

The cube that the mapping from (9.8) produces can be 'inflated' into a sphere $\mathcal{M}_-$ of radius R, utilizing the mapping $\mathbf{s}_2 : \Omega^* \to \mathcal{M}_-$, where

$$\mathbf{s}_2 = \frac{R}{\sqrt{x^2 + y^2 + z^2}} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{9.10}$$

(see figure 9.2). Regarding each $P^* \in \Omega^*$ as a vector $\mathbf{P}^*$ pointing from the origin to P^*, the mapping from (9.10) takes P^* and maps it onto the point $P \in \mathcal{M}_-$ with $\mathbf{P}$ the vector of length R in the direction of $\mathbf{P}^*$. Since no two points P_1 and P_2 on Ω^* are such that $\mathbf{P}_1 = \lambda \mathbf{P}_2$ for some $\lambda \in \mathbb{R}$, the mapping $\mathbf{s}^2 : \Omega^* \to \mathcal{M}_-$ is bijective. It remains to mention that $\mathbf{s}_2$ produces a sphere since $\|\mathbf{s}_2(\mathbf{x})\| = R, \ \ \forall \mathbf{x} \in \Omega^*$.
We can combine $\mathbf{s}_1$ and $\mathbf{s}_2$ to form the operator $\mathbf{i}_s : \Omega \to \mathcal{M}_-$ (the initial geometry) as follows

$$\mathbf{i}_s = \mathbf{s}_2 \circ \mathbf{s}_1. \tag{9.11}$$

The sphere that (9.11) produces is depicted in figure 9.2. Henceforth, we shall refer to the sphere that (9.11) produces as the *'ordinary sphere'*.
An alternate way to map Ω^* to a sphere, is to use the following mapping

$$\mathbf{s}_2^* = R \begin{pmatrix} x\sqrt{1 - \frac{y^2}{2} - \frac{z^2}{2} + \frac{y^2 z^2}{3}} \\ y\sqrt{1 - \frac{z^2}{2} - \frac{x^2}{2} + \frac{z^2 x^2}{3}} \\ z\sqrt{1 - \frac{x^2}{2} - \frac{y^2}{2} + \frac{x^2 y^2}{3}} \end{pmatrix}. \tag{9.12}$$

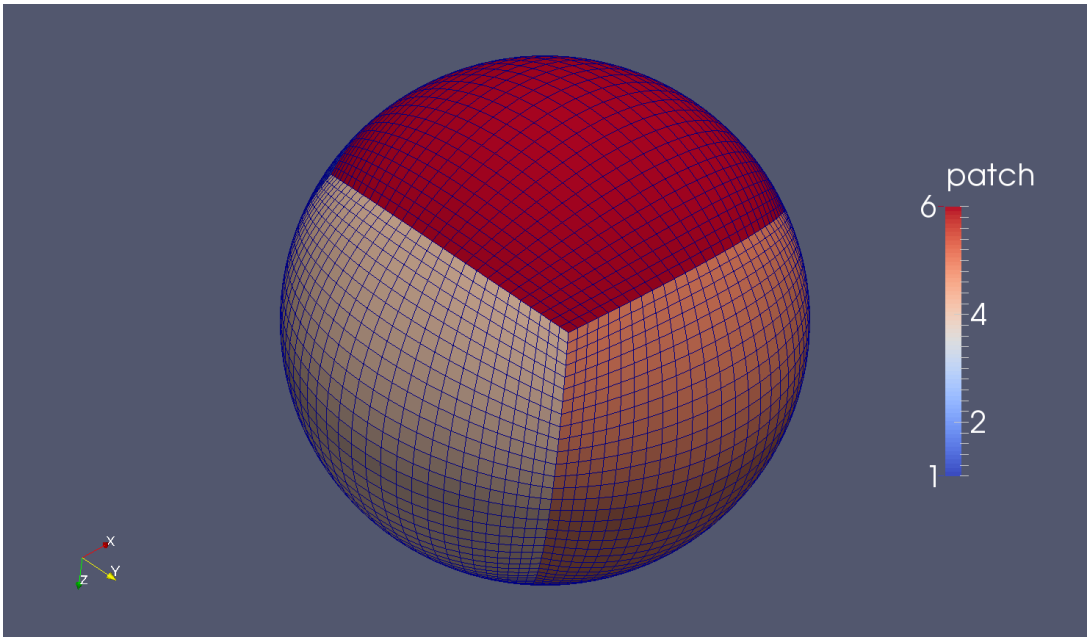


Figure 9.2: The result of (9.11) with various patches highlighted in different colors. The cells become smaller toward the patch boundaries but it is seen that there are no singularities in the parametrization. Note that the transition between the patch boundaries is non-smooth !

This parameterization is based on the observation that the function

$$S^*(x,y,z) = x^2 + y^2 + z^2 - x^2y^2 - y^2z^2 - z^2x^2 + x^2y^2z^2 \tag{9.13}$$

satisfies

$$S^*|_{\Omega^*} = 1. \tag{9.14}$$

Equation (9.13) is manipulated as follows

$$\begin{aligned} R^2S^*(x,y,z) =& R^2\left(x^2 + y^2 + z^2 - x^2y^2 - y^2z^2 - z^2x^2 + x^2y^2z^2\right) \\ =& R^2x^2\left(1 - \frac{y^2}{2} - \frac{z^2}{2} + \frac{y^2z^2}{3}\right) + \\ & R^2y^2\left(1 - \frac{z^2}{2} - \frac{x^2}{2} + \frac{x^2z^2}{3}\right) + \\ & R^2z^2\left(1 - \frac{x^2}{2} - \frac{y^2}{2} + \frac{x^2y^2}{3}\right), \end{aligned} \tag{9.15}$$

from which (9.12) follows.
(9.14) and (9.15) imply that (9.12) indeed maps points from Ω^* to a sphere. The operator

$$\mathbf{i}_s = \mathbf{s}_2^* \circ \mathbf{s}_1 \tag{9.16}$$

produces a sphere that is depicted in figure 9.3. Henceforth, we shall refer to the sphere that (9.16) produces as the *'gaming sphere'*. The sphere $\mathscr{M}_-$ that (9.11) or (9.16) produces can then be projected onto an IgA basis Σ in order to form $\mathscr{M}_0$. Constructing a suitable IgA basis shall be the topic of the next section.

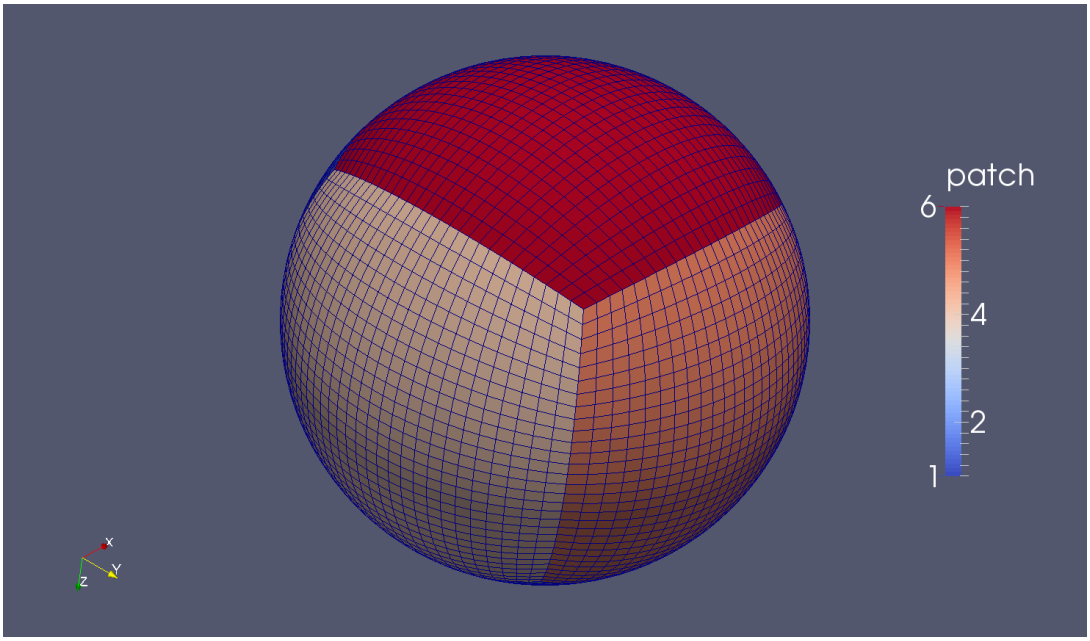


Figure 9.3: The result of (9.16) with various patches highlighted in different colors. The cells become smaller toward the patch boundaries but the effect is somewhat smaller than in figure 9.2. Note also that the angle that the cells make with one another at the boundaries is reduced.

9.3. Constructing a Basis

In section 7.4, we have deduced the general properties that any basis Σ should satisfy in order to be compatible with the temporal and the spatial discretization schemes. We will now present a concrete procedure to construct a basis compliant of the principles from section 7.4, utilizing clamped knot vectors (see section 4.2) and some additional tweaking. Given some order $p \geq 1$, some $n \geq p+1$ and the uniform, clamped knot vector

$$\Xi_{n,p} = \left\{ \underbrace{0,\ldots,0}_{p+1 \text{ times}}, \underbrace{\frac{1}{n-p},\ldots,\frac{n-p-1}{n-p}}_{n-p-1 \text{ terms}}, \underbrace{1,\ldots,1}_{p+1 \text{ times}} \right\}, \tag{9.17}$$

we construct the univariate basis (see figure 8.1)

$$\sigma_{n,p} = \{N_{1,p},\ldots,N_{n,p}\} \tag{9.18}$$

to construct the bivariate tensor-product basis

$$\Sigma^1 = \{w_1^1,\ldots,w_{n^2}^1\}, \tag{9.19}$$

from $\sigma_{n,p} \times \sigma_{n,p}$. The $w_i^1 \in \Sigma^1$ are only supported on Ω_1. We can 'copy' the basis Σ^1 to form Σ^i, containing functions supported on Ω_i as follows

$$\Sigma^i = \{w_1^1(\xi - q_1^i, \eta - q_2^i),\ldots,w_{n^2}^1(\xi - q_1^i, \eta - q_2^i)\}. \tag{9.20}$$

The corresponding elements are acquired from the elements corresponding to Σ^1 by the same shifted duplication. Some functions from

$$\tilde{\Sigma} = \bigcup_{i=1}^{6} \Sigma^i \tag{9.21}$$

do not satisfy (7.30). These are exactly the functions containing the clamped functions $N_{1,p}$, $N_{n,p}$, or a shifted copy in their tensor-product structure. Their global counterparts will exhibit a discontinuity when projected from Ω onto $\mathscr{M}_-$. As a result of the clamped basis property, the set

$$\{w_i^k \in \tilde{\Sigma} \mid W_i^k \text{ does not satisfy (7.30)}\}$$

is identical to the set

$$\Sigma_d = \{w_i^k \in \tilde{\Sigma} \mid w_i^k|_{\partial\Omega_k} \neq 0\}. \tag{9.22}$$

Like in chapter 8, a remedy is the coupling of degrees of freedom (DOFs). We combine some functions $w_i^k \in \Sigma_d$ in $\tilde{\Sigma}$ to form a single function such that the resulting basis Σ satisfies (7.30). The terminology stems from the fact that the merging of a set of functions is equivalent to requiring that their weights be equal in each function from span $\tilde{\Sigma}$. Thanks to the Σ^i being shifted copies of one another, functions on either side of the patch boundaries are 'compatible' with one another in the sense that they can be coupled so as to form a basis compliant with (7.30).
Let us assume that each index i of $w_i^k \in \Sigma^k$ is the result of a lexicographical numbering, i.e.

$$w^k_{(j-1)n+i} = N_{i,p}(\xi - q_1^k, \eta - q_2^k) N_{j,p}(\xi - q_1^k, \eta - q_2^k). \tag{9.23}$$

Furthermore, let us refer to the southern, eastern, northern and western boundary of patch Ω_k by γ_k^1, γ_k^2, γ_k^3 and γ_k^4, respectively. With the numbering from (9.23) in mind, the sets

$$\Sigma_j^k \equiv \left\{ w_i^k \in \Sigma^k \mid w_i^k|_{\gamma_k^j} \neq 0 \right\}, \tag{9.24}$$

are given by

$$\begin{aligned}
\Sigma_1^k &= \{w_1^k, w_2^k, \ldots, w_n^k\} \\
\Sigma_2^k &= \{w_n^k, w_{2n}^k, \ldots, w_{n^2}^k\} \\
\Sigma_3^k &= \{w^k_{(n-1)n+1}, w^k_{(n-1)n+2}, \ldots, w_{n^2}^k\} \\
\Sigma_4^k &= \{w_1^k, w_{1+n}^k, \ldots, w^k_{(n-1)n+1}\}.
\end{aligned} \tag{9.25}$$

A pair of functions $w_i^j \in \Sigma_k^j$, $w_q^r \in \Sigma_s^r$, $j \neq r$ should be coupled whenever their global counterparts are each other's reflection across some patch boundary on the sphere (or cube, see figure 9.4)

Remark. *In (9.24) some of the sets will, strictly speaking, be empty. This is a result of the fact that according to (4.1), $N_{n,p}(1) = 0$ (since the C^{-1} discontinuity has been chosen to occur at $\xi = 1^-$ as opposed to $\xi = 1^+$). Here we may assume that $N_{n,p}(1) = 1$. In the remainder we will disregard such mathematical subtleties for the sake of convenience and assume that the reader is aware of the subtleties involved.*

Table (9.1) lists the boundary sets that ought to be coupled. The functions w_i^k should be coupled via their index i in ascending order. Whenever a column in (9.1) is marked by an *'r'*, the coupling should be carried out in reverse, as for example in column 4:

$$\begin{array}{lccccc}
\Sigma_3^1 = & \{ & w^1_{(n-1)n+1}, & w^1_{(n-1)n+2}, & \ldots, & w^1_{n^2} \quad \} \\
 & & \Updownarrow & \Updownarrow & & \Updownarrow \\
\Sigma_4^4 = & \{ & w^4_{(n-1)n+1}, & w^4_{(n-2)n+1}, & \ldots, & w^4_1 \quad \}.
\end{array} \tag{9.26}$$

Note that there will be eight functions that are each comprised of three functions from $\tilde{\Sigma}$. These

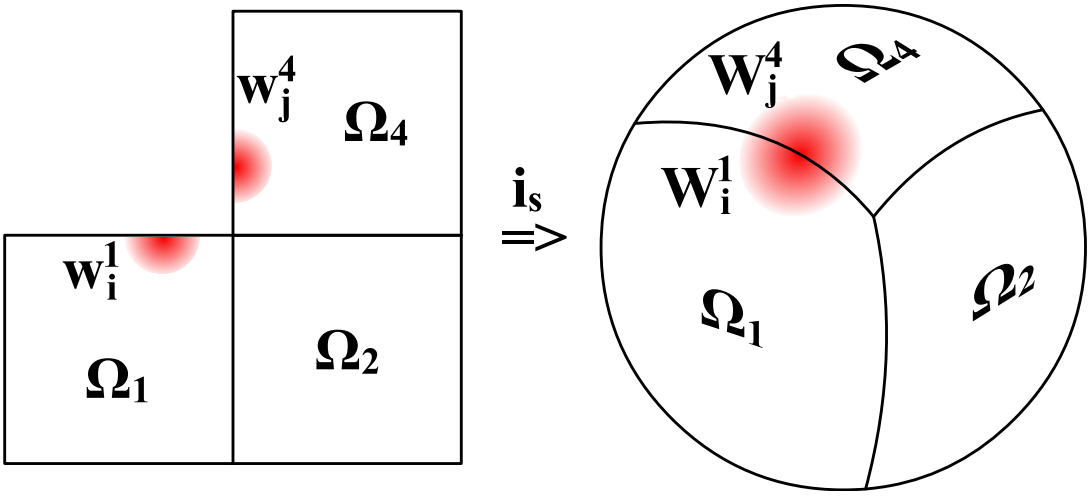


Figure 9.4: Functions w_i^1 and w_j^4 should be coupled because their global counterparts meet on the crossing between the projections of Ω_1 and Ω_4 on $\mathscr{M}_-$.

Boundary set Σ_x^y	Σ_1^1	Σ_1^2	Σ_1^3	Σ_1^4	Σ_2^1	Σ_2^2	Σ_2^3	Σ_3^1	Σ_3^2	Σ_4^2	Σ_4^3	Σ_5^2
To be coupled with	Σ_3^4	Σ_2^4	Σ_4^4, r	Σ_6^2	Σ_3^3	Σ_5^4	Σ_4^1	Σ_6^1, r	Σ_5^1, r	Σ_5^3	Σ_6^3, r	Σ_6^4

Table 9.1: All pairs of patch boundaries whose functions have to be coupled. The *'r'* indicates that functions have to be coupled in reverse.

functions are comprised result from the coupling of three functions of the form

$$w_i^k = N_{r,p}(\xi - q_1^k, \eta - q_2^k) N_{s,p}(\xi - q_1^k, \eta - q_2^k), \tag{9.27}$$

with $\{r,s\} \in \{0,n\} \times \{0,n\}$.
After the coupling has been completed, we form the basis $\mathscr{B}_1$ comprised of all coupled *'boundary'* functions and uncoupled *'internal'* functions. Finer bases $\mathscr{B}_2, \ldots$ can be constructed by repeating above steps with $n \to 2n-p$, $n \to 4n-3p$, etc. The basis Σ can be initialized to $\Sigma = \mathscr{B}_1$ after which the time-stepping procedure can commence.

9.4. Improving Smoothness

The basis from section 9.3 is a mathematically valid basis but it only possesses C^0-continuity by the patch boundaries. As the mappings $\mathbf{s}^k$ are elements from spanΣ, this lack of higher order smoothness will most likely manifest itself through small *'kinks'* in $\mathscr{M}_k$ located at the patch boundaries. It would hence be convenient to be able to construct bases with higher order smoothness across patch boundaries. On first glance, constructing a basis similar to the one from section 8.2.2 on each patch individually and coupling *'compatible'* basis functions across patch boundaries might seem like an option. This coupling, however, will lead to compatibility problems at the *'triple points'*, the spots where three patches meet (in Ω and $\mathscr{M}_-$). The vertex at which three patches meet is also referred to as *extraordinary vertex of valence three* [11]. Higher-order smoothness is generally hard to achieve in the vicinity of extraordinary vertices. This is also seen in the tensor-product structure of the C^{p-1}-continuous basis from section 8.2.2 that can be extended to acquire smoothness between pairs of patches but not in the vicinity of extraordinary vertices (see figure 9.5). Thus, improving the smoothness of the geometry (and the functions on it) proves to be a more challenging task than in the case of the torus. Following [9], we will nevertheless present a method to improve the smoothness between pairs of patches at segments of patch boundaries located sufficiently far from the patch vertices.

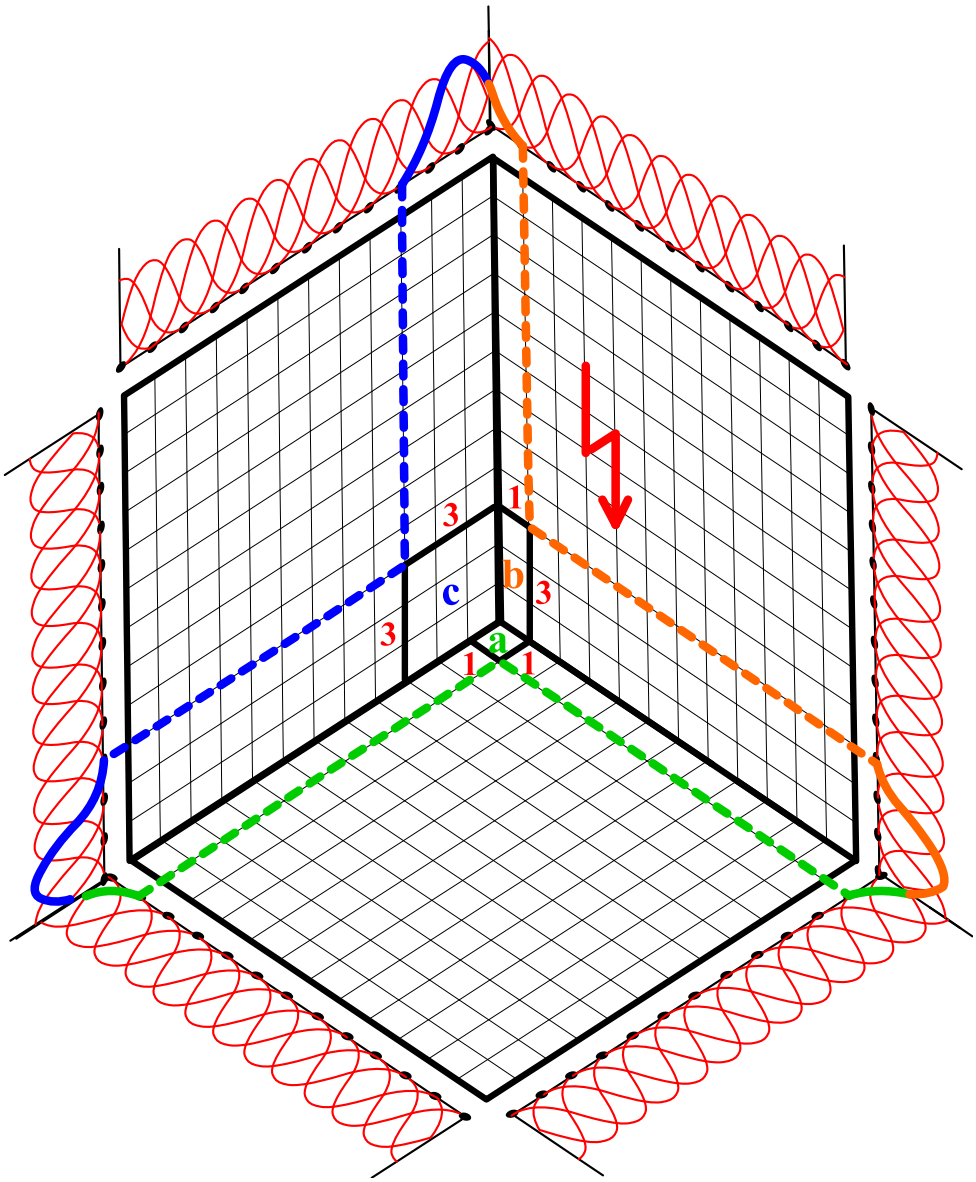
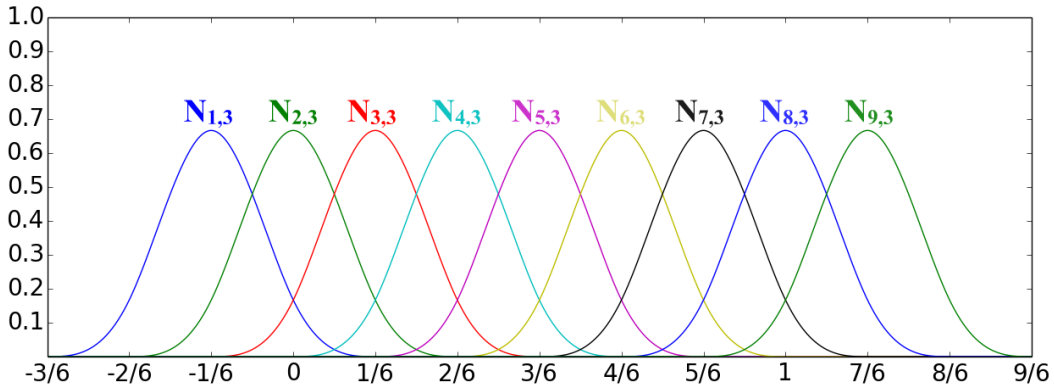


Figure 9.5: As an example of the incompatibility of basis functions near the triple points, we consider the tensor-product function resulting from the two univariate functions plotted in green. The function is supported by segment a) and to achieve smoothness in the vertical direction, we have to couple with the function supported by b). Finally, to achieve smoothness in the horizontal direction we have no choice but to couple b) with c) but then c) is not compatible with a).

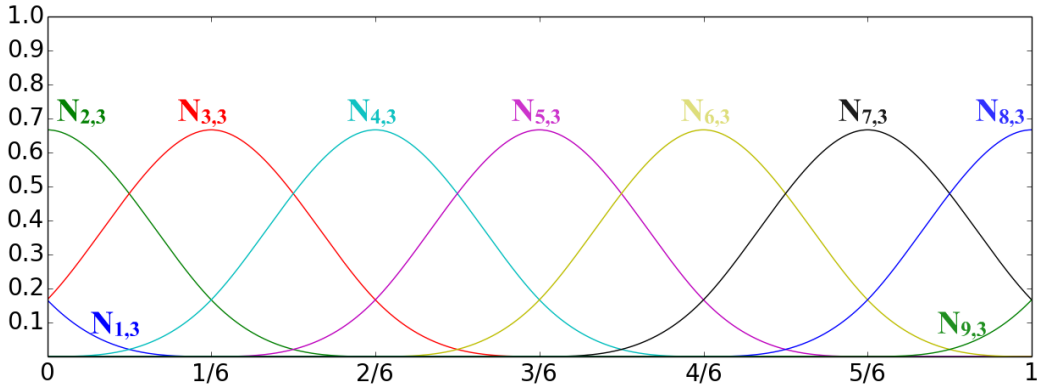
As indicated in figure 9.5, there is no non-trivial way to couple functions in the vicinity of triple points in order to achieve higher-order continuity. Further away from the triple points, coupling can take place in a way similar to section 8.2.2. The idea is to introduce the knot-vector

$$\mathscr{H}_{n,p} = \left\{ -\frac{p}{n-p}, -\frac{p+1}{n-p}, \ldots, 0, \frac{1}{n-p}, \frac{2}{n-p}, \ldots, 1, \frac{n-p+1}{n-p}, \ldots, \frac{n}{n-p} \right\}, \tag{9.28}$$

and restrict the functions to the interval $[0,1]$ by disregarding all the parts that are not supported by this interval (see figure 9.6). As before, the restricted basis $\tilde{\sigma}_{n,p} = \{\tilde{N}_{1,p}, \ldots, \tilde{N}_{n,p}\}$ is utilized to construct the tensor-product basis $\tilde{\Sigma}^1 = \tilde{\sigma}_{n,p} \times \tilde{\sigma}_{n,p}$.
$\tilde{\Sigma}^1$ is copied and shifted as in (9.20) to form the bases $\tilde{\Sigma}^i$, $i \in \{1, \ldots, 6\}$ with $\tilde{w}_i^k \in \tilde{\Sigma}^k$ satisfying supp $w_i^k \subset$



(a)



(b)

Figure 9.6: Example of a basis resulting from the knot-vector $\mathscr{H}_{9,3}$. (a) shows the basis in the absence of restriction and (b) in the presence.

Ω_k. Here,

$$\tilde{\Sigma}^k = \left\{ \tilde{w}^k_1, \ldots, \tilde{w}^k_{n^2} \right\}. \tag{9.29}$$

Again, we shall assume that each global index i of a function $\tilde{w}^k_i \in \tilde{\Sigma}^k$ is the result of a lexicographical numbering. As a next step the DOFs are coupled. The coupling is carried out in a way that is compliant with table (9.1) with the difference that this time several layers are coupled. It is seen from figure 9.6 that exactly p layers have to be coupled (per pair of patch boundaries). Let us define the sets

$$\begin{aligned}
\Sigma^k_{1,q} &= \left\{ \tilde{w}^k_{1+(q-1)n}, \tilde{w}^k_{2+(q-1)n}, \ldots, \tilde{w}^k_{n+(q-1)n} \right\} \\
\Sigma^k_{2,q} &= \left\{ \tilde{w}^k_{n-(q-1)}, \tilde{w}^k_{2n-(q-1)}, \ldots, \tilde{w}^k_{n^2-(q-1)} \right\} \\
\Sigma^k_{3,q} &= \left\{ \tilde{w}^k_{(n-q)n+1}, \tilde{w}^k_{(n-q)n+2}, \ldots, \tilde{w}^k_{(n-q+1)n} \right\} \\
\Sigma^k_{4,q} &= \left\{ \tilde{w}^k_{1+q}, \tilde{w}^k_{1+q+n}, \ldots, \tilde{w}^k_{1+q+(n-1)n} \right\},
\end{aligned} \tag{9.30}$$

with $q \in \{1, \ldots, p\}$.
These sets are coupled in a fashion similar to that of section 9.3, with the difference that whenever table (9.1) suggests that Σ^i_j and Σ^k_l should be coupled, we couple $\Sigma^i_{j,q}$ with $\Sigma^k_{l,p-q+1}$ instead (for all

$q \in \{1,\dots,p\}$ and coupling in reverse whenever the corresponding column contains an *'r'*). The basis $\mathscr{B}_1^*$ containing all the coupled basis functions as well as all internal functions from the $\tilde{\Sigma}^k$ exhibits improved smoothness across patch boundaries but will suffer from the problem depicted in figure 9.5 that occurs nearby the triple points. The idea is to remove all functions that suffer from this issue and replace them by functions that are C^0-continuous across patch boundaries but smooth everywhere else (and do not suffer from the same issue).
To this end, let us assume that we are in possession of the bases $\mathscr{B}_1^*$ and $\mathscr{B}_1$, where $\mathscr{B}_1$ is constructed using the same parameters n, p (see section 9.3), we initialize $\mathscr{B}_1^+$ to $\Sigma = \mathscr{B}_1^*$. As a next step we define the security layers. Defining the intervals

$$\begin{aligned} \chi_{i^+} &\equiv \left[i, i + \tfrac{p}{n-p} \right] \\ \chi_{i^-} &\equiv \left[i - \tfrac{p}{n-p}, i \right], \end{aligned} \tag{9.31}$$

the security layers β_i are given by

$$\begin{aligned} \beta_1 &= \left(\chi_{0^+} \times \chi_{0^+}\right) \cup \left(\chi_{1^+} \times \chi_{-1^+}\right) \cup \left(\chi_{4^-} \times \chi_{0^+}\right) \\ \beta_2 &= \left(\chi_{1^-} \times \chi_{0^+}\right) \cup \left(\chi_{1^+} \times \chi_{0^+}\right) \cup \left(\chi_{1^+} \times \chi_{0^-}\right) \\ \beta_3 &= \left(\chi_{0^+} \times \chi_{1^-}\right) \cup \left(\chi_{1^+} \times \chi_{2^-}\right) \cup \left(\chi_{4^-} \times \chi_{1^-}\right) \\ \beta_4 &= \left(\chi_{1^-} \times \chi_{1^-}\right) \cup \left(\chi_{1^+} \times \chi_{1^-}\right) \cup \left(\chi_{1^+} \times \chi_{1^+}\right) \\ \beta_5 &= \left(\chi_{2^-} \times \chi_{0^+}\right) \cup \left(\chi_{2^-} \times \chi_{0^-}\right) \cup \left(\chi_{2^+} \times \chi_{0^+}\right) \\ \beta_6 &= \left(\chi_{2^-} \times \chi_{1^-}\right) \cup \left(\chi_{2^-} \times \chi_{1^+}\right) \cup \left(\chi_{2^+} \times \chi_{1^-}\right) \\ \beta_7 &= \left(\chi_{3^-} \times \chi_{0^+}\right) \cup \left(\chi_{3^+} \times \chi_{0^+}\right) \cup \left(\chi_{2^-} \times \chi_{-1^+}\right) \\ \beta_8 &= \left(\chi_{3^-} \times \chi_{1^-}\right) \cup \left(\chi_{3^+} \times \chi_{1^-}\right) \cup \left(\chi_{2^-} \times \chi_{2^-}\right). \end{aligned} \tag{9.32}$$

Here, the index i matches the index of $\mathbf{r}_i$ that forms the center of the i^{th} triple point on the cube Ω^*. around which the β_i are located. The security layers for a setting with $p = 3$ and $n = 18$ are depicted in figure 9.7.
The next step is to remove all functions $w_i \in \mathscr{B}_1^+$ with $\operatorname{supp} w_i \subset \beta_j$, for some $j \in \{1,\dots,8\}$. The reduced basis is enriched with all functions $w_i \in \mathscr{B}_1$ with $\operatorname{supp} w_i \subset \beta_j$, for some $j \in \{1,\dots,8\}$. Upon completion of these steps, the basis $\mathscr{B}_1^+$ forms a nonzero partition of unity on Ω with improved smoothness [9]. C^{p-1}-continuity is achieved everywhere apart from the $\beta_i \subset \Omega$. Furthermore, the relative size of the security layers decreases for finer initial grids. Given some n, p and initial basis $\mathscr{B}_1^+$, hierarchically finer bases are constructed by repeating above steps with halved spacing but using the *same* set of security layers β_i that correspond to $\mathscr{B}_1^+$. Decreasing the size of the β_i with the spacing is not recommended since finer bases do not satisfy $\operatorname{span} \mathscr{B}_i^+ \subset \operatorname{span} \mathscr{B}_j^+$ in that case.
Constructing finer bases while adjusting the size of the security layers according to the new spacing will lead to projection errors upon refinement but increases the relative size of the segments of Ω on which C^{p-1}-continuity is achieved.

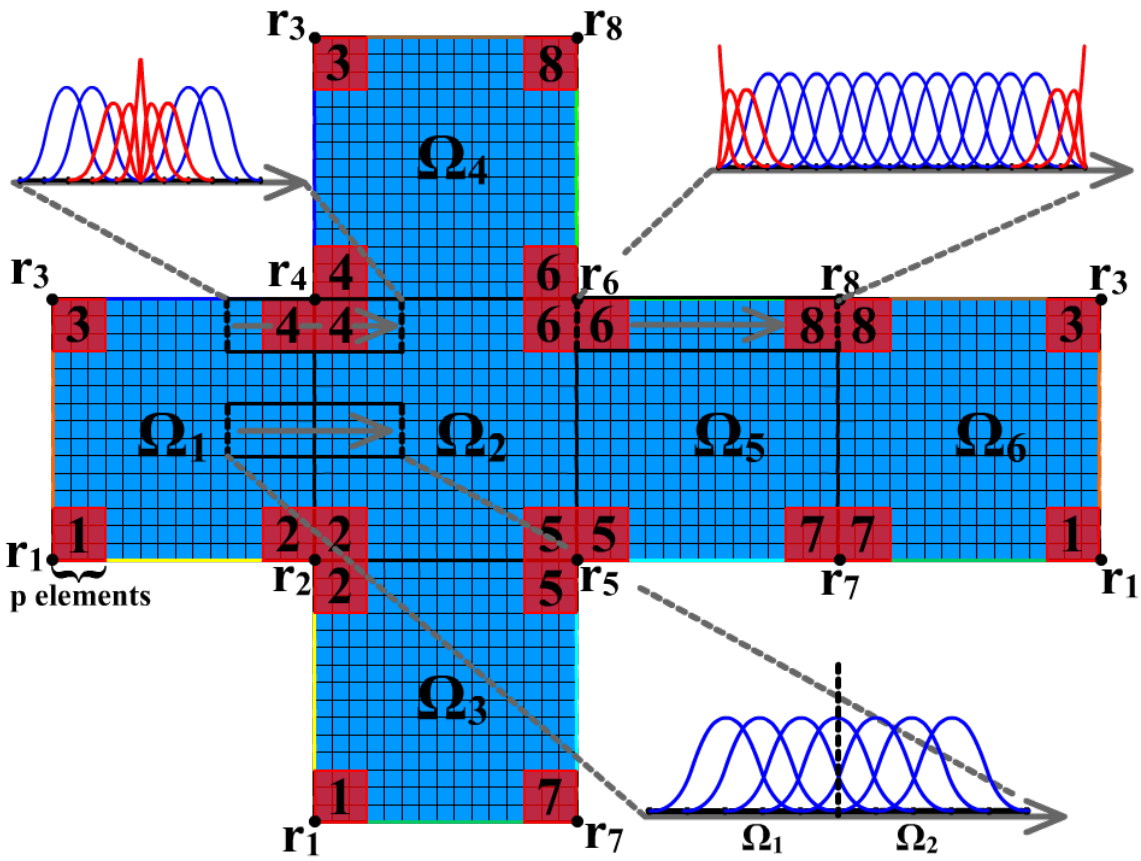


Figure 9.7: As an example of the incompatibility of basis functions near the triple points, we consider the tensor-product function resulting from the two univariate functions plotted in green. The function is supported by segment a) and to achieve smoothness in the vertical direction, we have to couple with the function supported by b). Finally, to achieve smoothness in the horizontal direction we have no choice but to couple b) with c) but then c) is not compatible with a).

10

Refinement Strategies

In section 4.6, we have introduced the concept of hierarchical refinement. We presented algorithms (1) and (2) as a basic tool for refinement. Both algorithms take a set of elements ϵ as input, so far, we have not discussed possible strategies for element selection.
There are two main strategies that come to mind: refinement based on cell size and refinement based on curvature. In this chapter, we will discuss possible strategies for element selection based on these two properties and possible refinement criteria.

10.1. Refinement Based on Cell Size

Let u^* be the u-component of the exact solution $\mathbf{u}^*$ of (7.1). One can measure the quality of the approximate solution at time-instance t^k utilizing, for example, the $L_2(\mathcal{M}_k)$- norm

$$r_u^k = \left(\int_\Omega \left(u^*\big|_{t=t^k} - u^k \right)^2 \sqrt{g_k} \mathrm{d}\boldsymbol{\xi} \right)^{1/2}, \tag{10.1}$$

or any other norm with measure $\sqrt{g_k}$. Even though u^* will in most cases not be known, we can see from (10.1) that deviations of u from u^* on segments of Ω where $\sqrt{g_k}$ is large have to be considered more critical in the assessment of numerical quality. It therefore makes sense to refine elements on which $\sqrt{g_k}$ is large with respect to some reference value μ_{cell}.
A possibility is to base this reference value on the initial condition. Assuming that the initial grid resolution has been appropriately chosen with respect to the initial geometry, the reference value can be based on the average value of $\sqrt{g_0}$ on each element. For reasons mentioned in section 4.6, however, it is not recommended to refine single cells. It is more favorable to craft a refinement criterion based on function support, this way it is ensured that at least one coarse function is replaced by finer ones upon refinement. Thus, let $\Sigma = \{w_1, \dots, w_N\}$ be the isogeometric basis at time instance t^k, we define

$$\mu_0^i = \frac{\int\limits_{\operatorname{supp} w_i} \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} 1 \mathrm{d}\boldsymbol{\xi}}, \quad \forall w_i \in \Sigma. \tag{10.2}$$

As a next step, we define the reference value as follows

$$\mu_{\text{cell}} = \frac{1}{N} \sum_{i=1}^{N} \mu_0^i. \tag{10.3}$$

After the time-stepping procedure commences, the geometry will deform and the average value of $\sqrt{g_k}$ will most likely increase on each element which is why it should be compared to the reference value after each iteration. To this purpose, we introduce

$$\mu_k^i = \frac{\int\limits_{\operatorname{supp} w_i} \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} 1 \mathrm{d}\boldsymbol{\xi}} \tag{10.4}$$

as the average value of $\sqrt{g_k}$ over supp w_i. Here, we will assume that each cell is refined only once, i.e. $\forall \epsilon_i \in \mathscr{A}: \quad \epsilon_i \in \mathscr{T}_1$ or $\epsilon_i \in \mathscr{T}_2$. With these considerations in mind, in words, a criterion to add an element $\epsilon_i \in \mathscr{A}$ to the set of to-be-refined elements ϵ during the k^{th} iteration, could be as follows:

$$\begin{aligned} &\text{For all } \mu_k^i, \text{ if } \mu_k^i > k_{\text{cell}} \mu_{\text{cell}}, \text{ add all elements that support } w_i \\ &\text{and are elements from } \mathscr{T}_1 \text{ to } \epsilon. \end{aligned} \tag{10.5}$$

For some $k_{\text{cell}} > 1$.
The collection of to-be-refined elements ϵ can then be extended by elements that should be refined due to curvature (see section 10.2), after which either algorithm (1) or (2) is called with argument ϵ.
A further improvement of above principles is to compute (10.2) and (10.4) with an additional w_i-measure, i.e.

$$\mu_0^i = \frac{\int\limits_{\operatorname{supp} w_i} w_i \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} w_i \mathrm{d}\boldsymbol{\xi}} \tag{10.6}$$

and

$$\mu_k^i = \frac{\int\limits_{\operatorname{supp} w_i} w_i \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} w_i \mathrm{d}\boldsymbol{\xi}}. \tag{10.7}$$

The idea behind this change is that, in deciding whether a function should be refined or not, one might argue that deviations from the reference value μ_0^i inside the *'bulk'* of the function (where the function value is large) should be considered more critical than deviations on points outside of the bulk. This extension of the measure is valid since $w_i > 0$ on supp w_i. Since we only refine functions from $\mathscr{B}_1$, in order to save computational resources, it makes sense to only compute μ_k^i for $\Sigma \ni w_i \in \mathscr{B}_1$ during each iteration.

10.2. Refinement Based on Curvature

At all time-instances t^k, $\mathbf{s}^k$ is an element from span Σ. Thus, it is important that the resolution of the basis suffices to capture all important geometrical details. In section 10.1 we introduced to concept of refinement via cell size. Another refinement criterion that comes to mind is the refinement based on curvature. Intuitively, segments of $\mathscr{M}_k$ that are strongly curved, are the surface-equivalent to large gradients in functions.
In section 2.4.1, we introduced the concept of *principal curvature* and in subsection 2.4.4, we presented a cheap way to compute the two principal curvatures. We will now deduce a feasible refinement strategy based on curvature. As in section 10.1, it makes sense to introduce a reference value of the curvature μ_{curve} based on the initial condition, keeping in mind that the sign of the curvature

should not matter for the refinement. With (10.6) in mind, this suggests a function-support based average of the form

$$\kappa_0^i = \frac{\int\limits_{\operatorname{supp} w_i} (|\kappa_1(\boldsymbol{\xi})| + |\kappa_2(\boldsymbol{\xi})|)\, w_i \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} w_i \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}, \tag{10.8}$$

where κ_1 and κ_2 are the two principal curvatures, extracted from the shape-operator $[S]$ through equation (2.44). The problem with (10.8), however, is that it is not compatible with quadrature schemes whenever either of the κ_i (or both) change sign inside one function-support (since the integrand is not a C^∞-continuous functions anymore in that case). A workaround for this issue is to introduce the initial support-based averages via

$$\kappa_0^i = \frac{\int\limits_{\operatorname{supp} w_i} (\kappa_1^2(\boldsymbol{\xi}) + \kappa_2^2(\boldsymbol{\xi}))\, w_i \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} w_i \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}, \tag{10.9}$$

and define the reference-value by

$$\mu_{\text{curve}} = \frac{1}{N} \sum_{i=1}^{N} \kappa_0^i. \tag{10.10}$$

Analogous to (10.7), we introduce

$$\kappa_k^i = \frac{\int\limits_{\operatorname{supp} w_i} (\kappa_1^2(\boldsymbol{\xi}) + \kappa_2^2(\boldsymbol{\xi}))\, w_i \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} w_i \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}}, \tag{10.11}$$

as a measure for curvature within one function-support after the k^{th} iteration. A criterion to add an element $\epsilon_i \in \mathscr{A}$ to the set of to-be-refined elements ϵ, could be

$$\begin{aligned} &\text{For all } \kappa_k^i, \text{ if } \kappa_k^i > k_{\text{curve}} \mu_{\text{curve}}, \text{ add all elements that support } w_i \\ &\text{and are elements from } \mathscr{T}_1 \text{ to } \epsilon. \end{aligned} \tag{10.12}$$

For some $k_{\text{curve}} > 1$.
After (10.5) and (10.12) have been carried out, if ϵ is not empty, algorithm (1) or (2) is called with ϵ as argument. The corresponding pseudocode can be found in algorithm (5).

Remark. *Both the average value of $\sqrt{g_k}$ and the average curvature do not change after refinement. This is why finer functions might be refined again after the next iteration. We circumvented this issue by requiring that cells from $\mathscr{T}_2$ are not eligible for refinement (refining only once is sufficient in this problem). In settings where it makes sense to refine twice or more, this issue could be solved by introducing reference values that are multiples of μ_{cell} and μ_{curve} for functions from $\mathscr{B}_2$ and their finer counterparts.*

Algorithm 5 Refinement Based on Cell Size and Curvature

$\epsilon \leftarrow \emptyset$
for all $\Sigma \ni w_i \in \mathscr{B}_1$ **do**
  if $\mu_k^i > k_{\text{cell}}\mu_{\text{cell}}$ **then**
    $\lambda \leftarrow \{\epsilon_i \in \operatorname{supp} w_i \mid \epsilon_i \in \mathscr{T}_1\}$
    $\epsilon \leftarrow \epsilon \cup \lambda$
  end if
  if $\kappa_k^i > k_{\text{curve}}\mu_{\text{curve}}$ **then**
    $\lambda \leftarrow \{\epsilon_i \in \operatorname{supp} w_i \mid \epsilon_i \in \mathscr{T}_1\}$
    $\epsilon \leftarrow \epsilon \cup \lambda$
  end if
end for
call Refine $\mathscr{A}$ (ϵ)

11

Results

In this chapter we will present the results of both an implementation on a torus, as well as a (gaming) sphere. The implementation generally follows the principles from chapter 7, with IgA bases and domains from chapter 8 and 9, respectively. The PID-parameters (see section 7.7) have been set to $tol = 0.01$, $h_{\min} = 0.1$ and $h_{\max} = 15$. We did not limit the growth and reduction of the time-step (i.e. $m = 0$ and $M = \infty$). The matrix and vector assemblies are carried out with Gauss-schemes (see chapter 5) of order six and the linear systems are solved with an iterative CG-solver. The refinement strategies from chapter 10 are present in the implementation with the parameters $k_{\text{cell}} = 2.1$ and $k_{\text{curve}} = 120$ for the torus and $k_{\text{curve}} = 160$ for the sphere. The implementation has been realized in the Python-package *Nutils* [3].
As in [25], the algorithm was manually terminated in order to avoid geometric intersection at later stages.

11.1. Implementation on the Torus

We have implemented the numerical scheme on the torus utilizing the basis from section 8.2.2 with the initial basis $\mathscr{B}_1$ constructed using the parameters $p = 3$ and $n = 50$ (making for an initial basis of cardinality 2500). The outer and inner radii have been set to $R_2 = 40$ and $R_1 = 20$, respectively. The function *'initial'* is given by four three-dimensional Gauss-functions $G_{\mathbf{x_0}}$

$$G_{\mathbf{x}_0}(\mathbf{x}) = \exp\left(-\sum_{i=1}^{3}\left(\frac{x_i - x_i^0}{20}\right)^2\right). \tag{11.1}$$

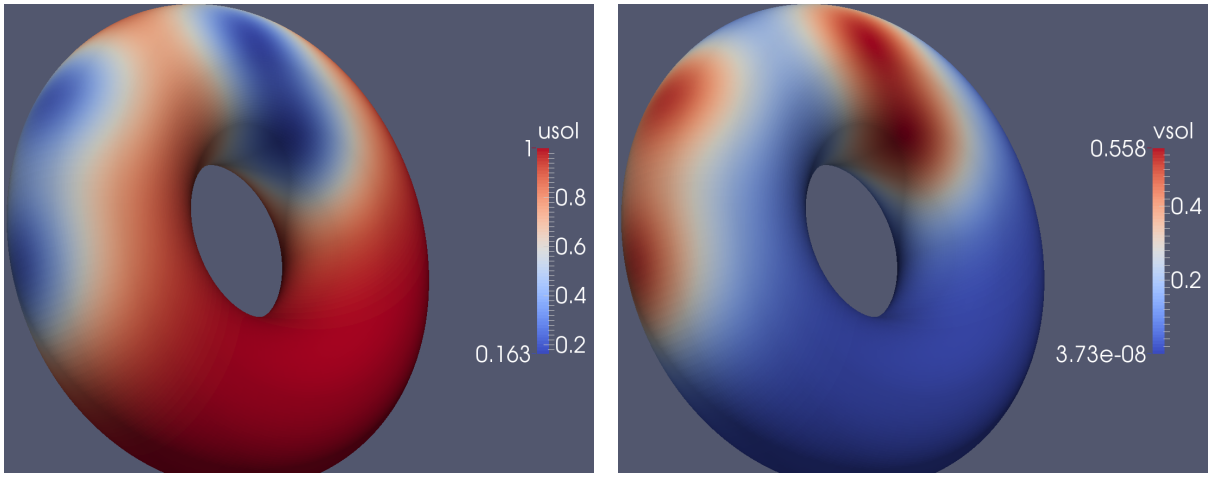
The $G_{\mathbf{x}_0}$ are centered at

$$\mathbf{x}_0 = \mathbf{i_s}(\xi_i, \eta_i), \tag{11.2}$$

with $\mathbf{i_s}$ as in (8.1). The (ξ_i, η_i) are given by $(\xi_1, \eta_1) = (1/2, 1/2)$, $(\xi_2, \eta_2) = (0.6, 0.55)$, $(\xi_3, \eta_3) = (0.75, 0.7)$ and $(\xi_4, \eta_4) = (0.85, 0.9)$. The initial conditions satisfy

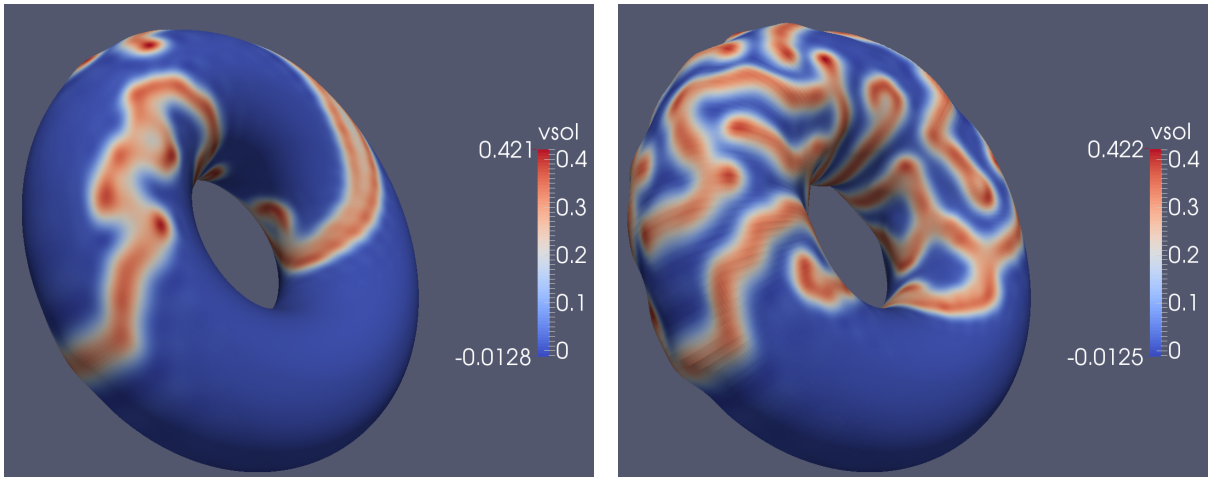
$$\begin{aligned} U(t=0) &= 1 - 0.75 \times \text{initial} \\ V(t=0) &= 0.5 \times \text{initial}, \end{aligned} \tag{11.3}$$

and are projected onto Σ before the first iteration, as well as the initial geometry-parameterization $\mathbf{i}_s$ (see figure 11.1, in the plots *'usol'* corresponds to U^k and *'vsol'* to V^k at $t = t^k$). We used the parameters $F = 0.04$, $H = 0.06$, $K = 0.001$, $d_1 = 0.2$ and $d_2 = 0.1$. Since $K \ll 1$, the geometry has been disregarded in the time-step selection.



(a) (b)

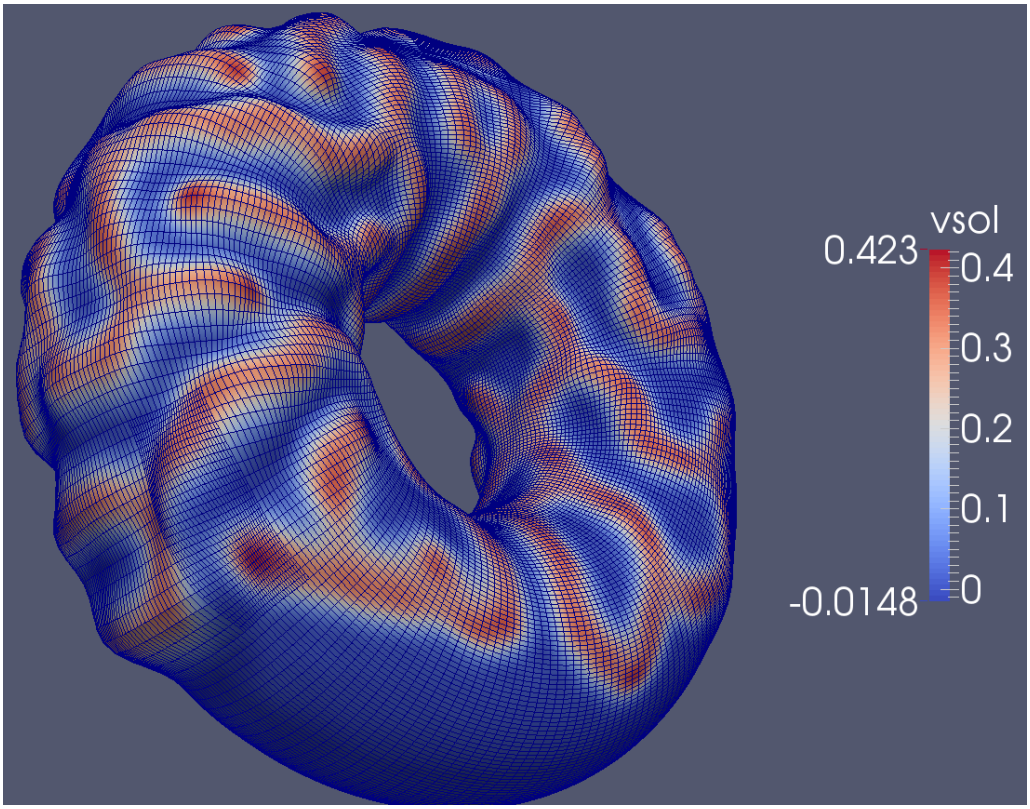
Figure 11.1: Initial condition of U (a) and V (b) plotted on $\mathscr{M}_0$. The local counterparts of the functions shown form the first iterants of the time-stepping scheme.



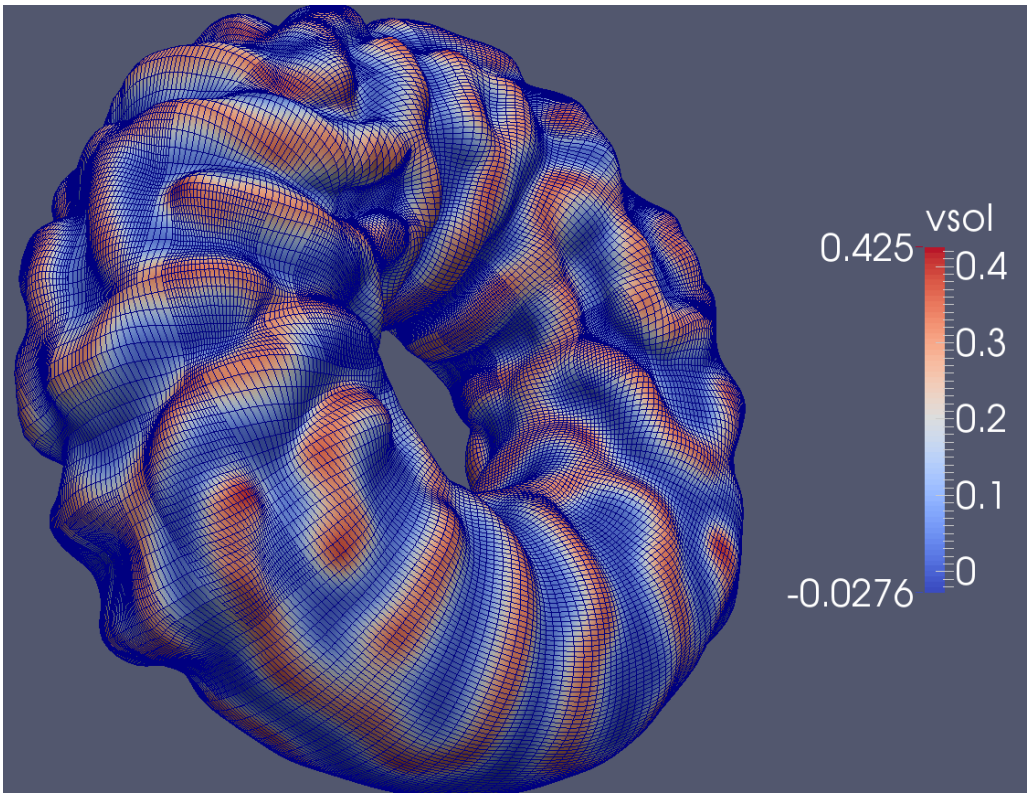
(a) (b)

Figure 11.2: Substrate V depicted on the geometry at (a) $t^{400} = 2000$ and (b) $t^{800} = 6800$.

Figures 11.2 and 11.3 show the state in which V^k and the geometry $\mathscr{M}_k$ find themselves for various t^k. In figure 11.2 (a) we see first signs of pattern formation that intensify in (b). In figure 11.3 (a) the patterns have spread out further and we see the first clear signs of geometric deformation. In figure 11.3 (b), the patterns have spread over the entire geometry and deformation has intensified. The grid has been enabled in order to indicate the segments of $\mathscr{M}_k$ that have been refined. In 11.3 (b), we can see that the solver has refined a large portion of the inner side of the torus, most likely due to curvature. Figures 11.4 to 11.6 show the geometry after the last iteration from various angles.



(a)



(b)

Figure 11.3: Substrate V depicted on the geometry at $t^{1200} = 13000$ (a) and $t^{1600} = 19000$ (b). The grid has been enabled in order to indicate where the solver refined.

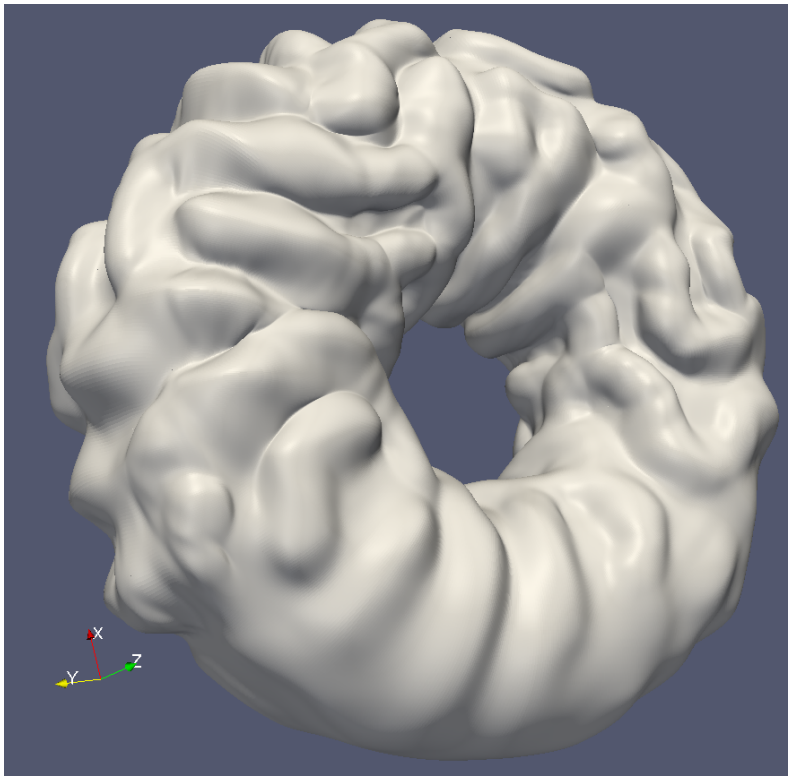


Figure 11.4: The final geometry at $t^{1740} = 2.2 \times 10^4$.

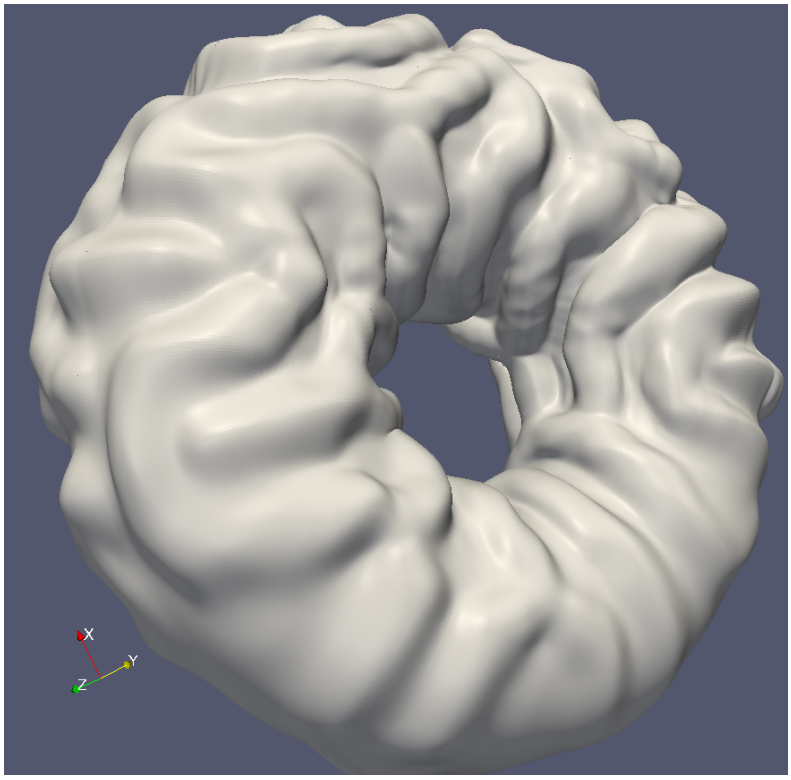


Figure 11.5: The geometry after the last iteration.

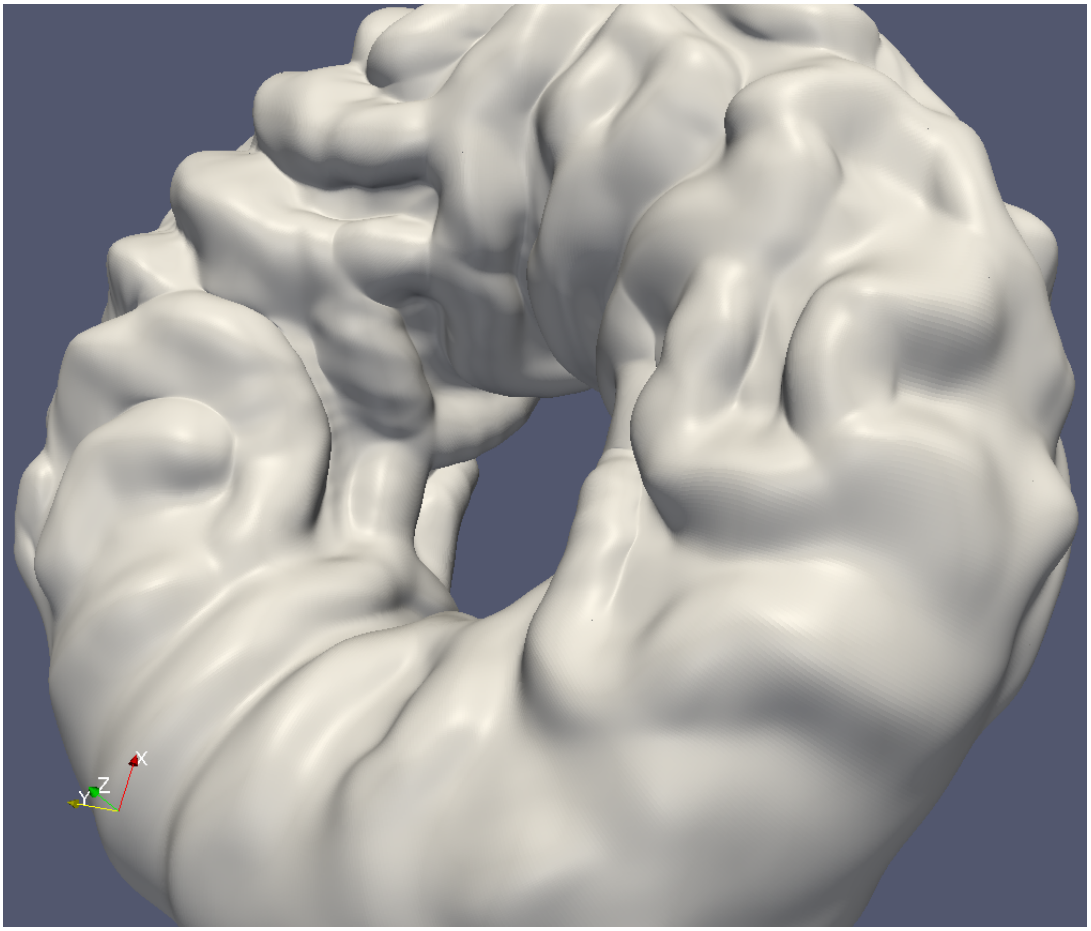


Figure 11.6: The geometry after the last iteration.

11.2. Implementation on the Gaming Sphere

In this section, we will present results corresponding to an implementation of the numerical scheme on the gaming sphere (see section 9.2), using the basis from section 9.3 with parameters $n = 28$ and $p = 3$. The radius of the initial sphere is given by $R = 40$. We present two simulations, one simulation with the same parameters as in section 11.1 and another simulation with F changed to $F = 0.0285$. This time, the function *'initial'* is given by four three-dimensional Gauss-functions

$$G_{\mathbf{x}_0}(\mathbf{x}) = \exp\left(-\left(\frac{x_i - x_i^0}{20}\right)^2 - \left(\frac{x_i - x_i^0}{15}\right)^2 - \left(\frac{x_i - x_i^0}{15}\right)^2\right). \tag{11.4}$$

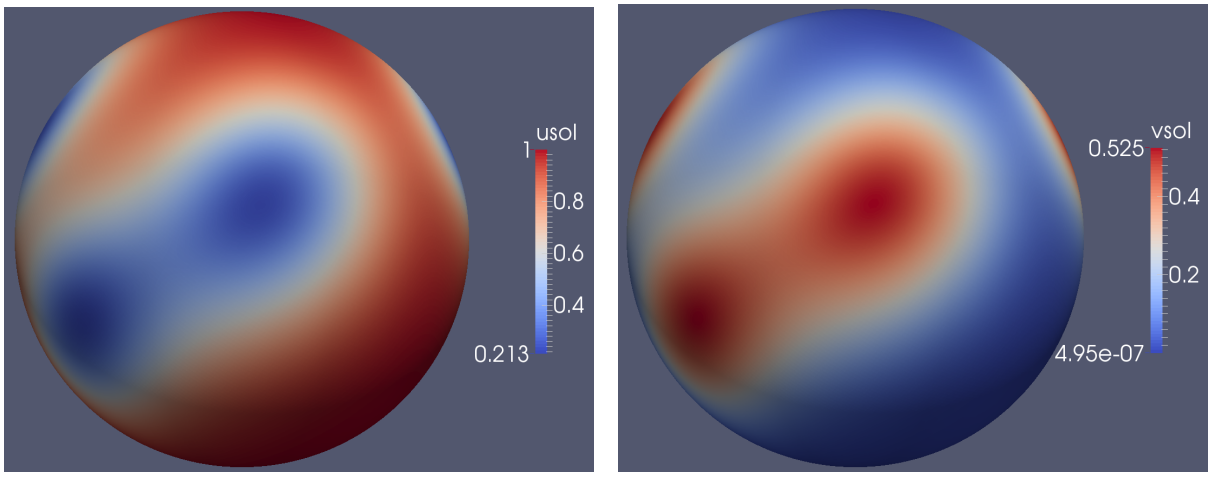
centered at

$$\mathbf{x}_0(\xi_i, \eta_i) = R\left[\sin(\xi_i)\cos(\eta_i), \sin(\xi_i)\sin(\eta_i), \cos(\xi_i)\right], \tag{11.5}$$

with $(\xi_1, \eta_1) = (0,0)$, $(\xi_2, \eta_2) = (0.3\pi, 0.4\pi)$, $(\xi_3, \eta_3) = (0.4\pi, 0.7\pi)$ and $(\xi_4, \eta_4) = (0.65\pi, \pi)$. As before, the initial concentrations satisfy

$$\begin{aligned} U(t=0) &= 1 - 0.75 \times \text{initial} \\ V(t=0) &= 0.5 \times \text{initial}, \end{aligned} \tag{11.6}$$

see figure 11.7.

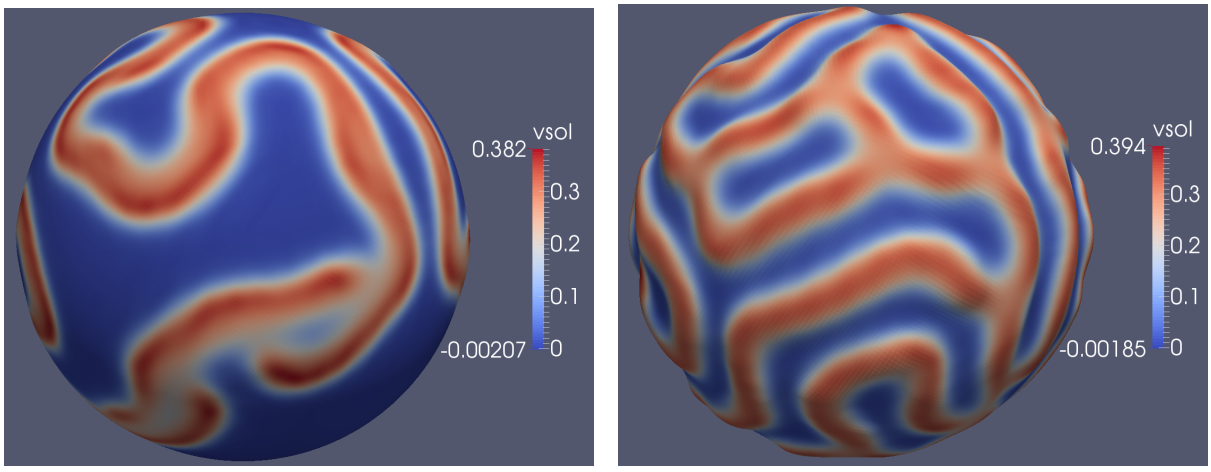


(a) (b)

Figure 11.7: Initial condition of U (a) and V (b) plotted on $\mathscr{M}_0$. The local counterparts of the functions shown form the first iterants of the time-stepping scheme.

11.2.1. $F = 0.04$

For a simulation with $F = 0.04$, figures 11.8, 11.9 and 11.10 show the state in which V^k and the geometry $\mathscr{M}_k$ find themselves for various t. In figure 11.8 (a), we see that the four Gaussians have formed several narrow bands of nonzero concentration and in (b) we see strong pattern formation as well as the first signs of geometrical deformations. These deformations intensify in figure 11.9 and we can see that the solver has performed the first local refinements, most likely due to curvature. In figure 11.10, the deformation intensify further and we can see that the solver has refined a large portion of the grid. Figures 11.11 to 11.13 show the geometry after the last iteration from various angles. The simulation has been terminated after 1650 iterations.



(a) (b)

Figure 11.8: Substrate V and the geometry for $F = 0.04$ at (a) $t^{400} = 1650$ and (b) $t^{800} = 6980$.

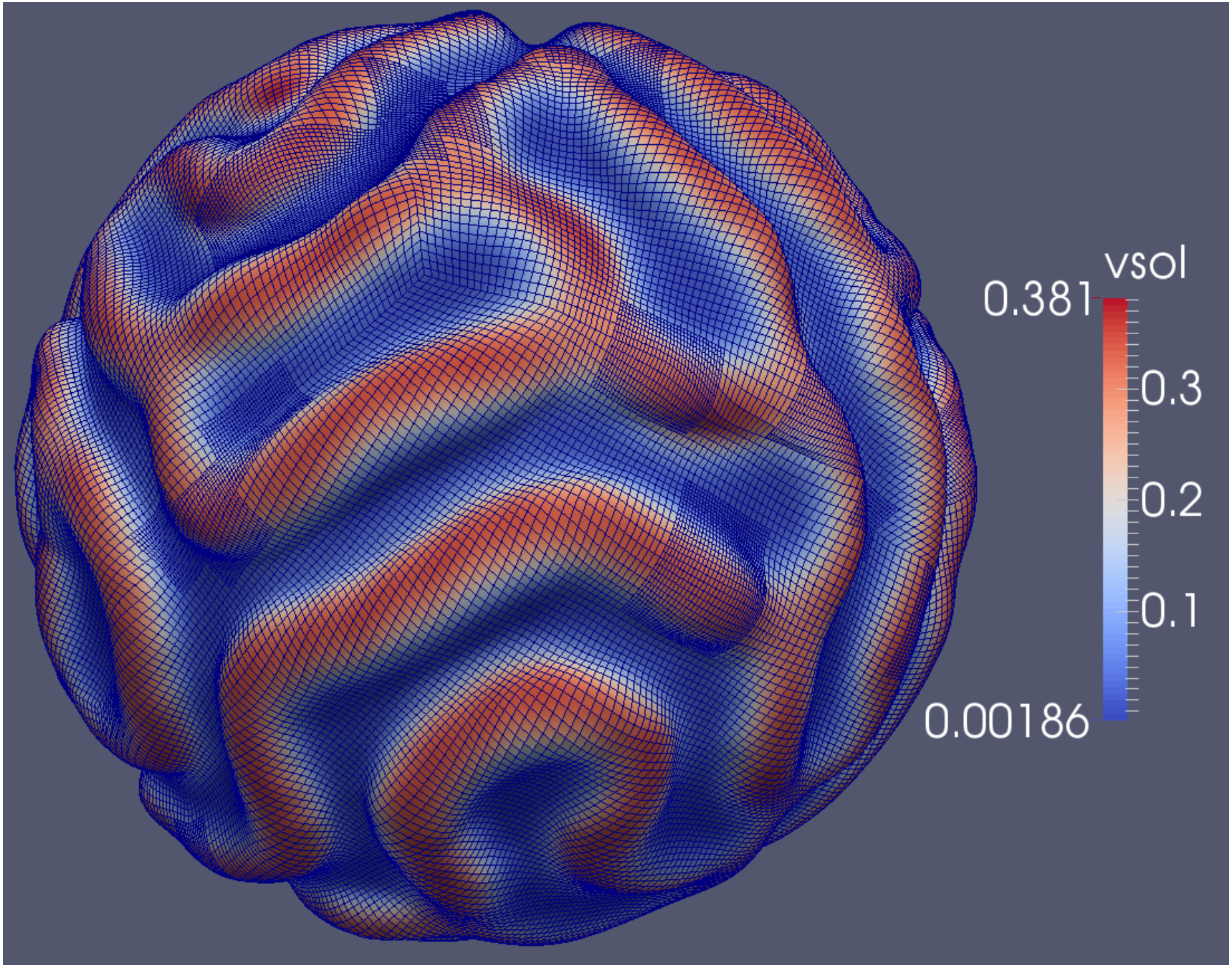


Figure 11.9: Substrate V and the geometry for $F = 0.04$ at $t^{1200} = 1.3 \times 10^4$.

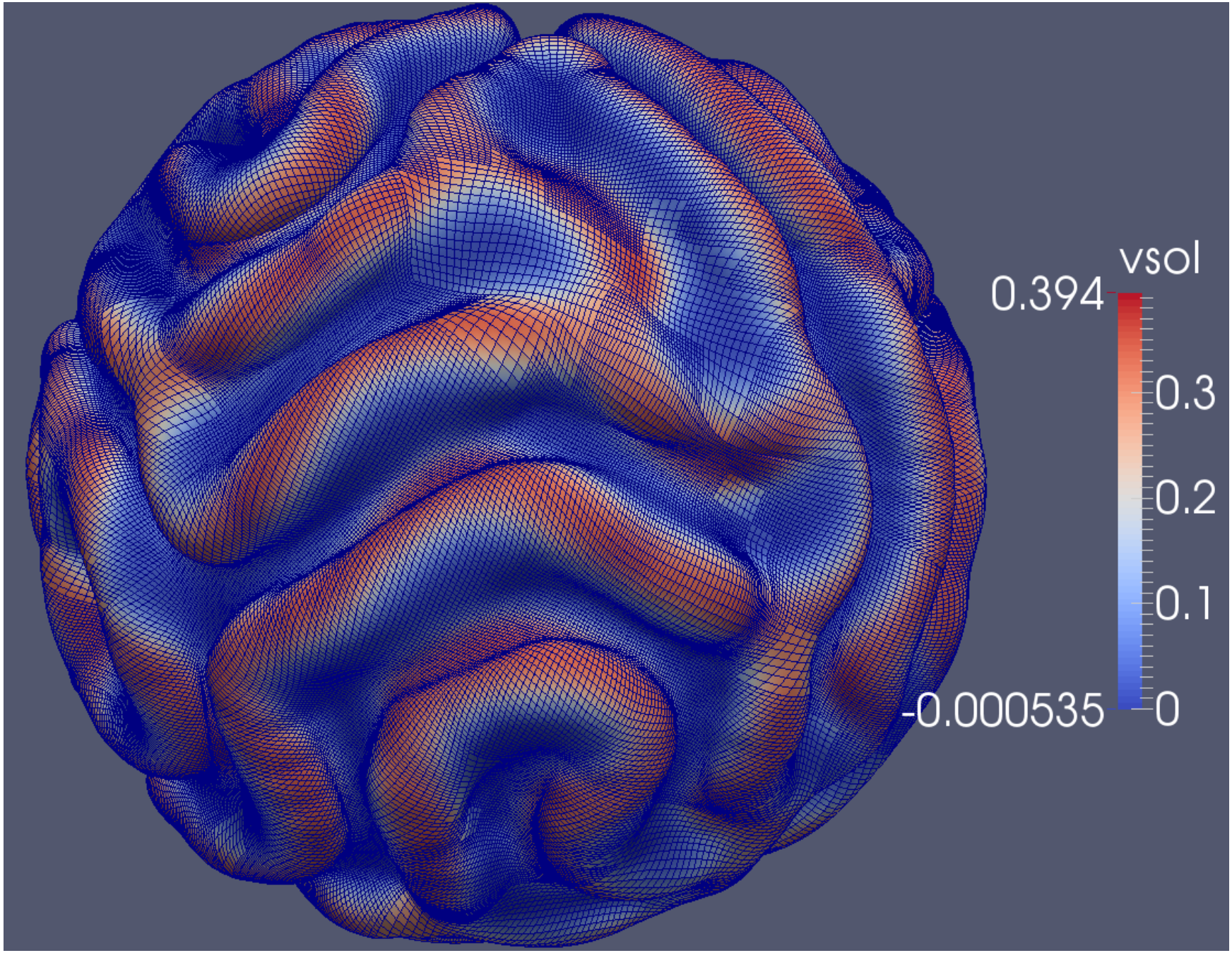


Figure 11.10: Substrate V and the geometry for $F = 0.04$ at $t^{1600} = 1.9 \times 10^4$.

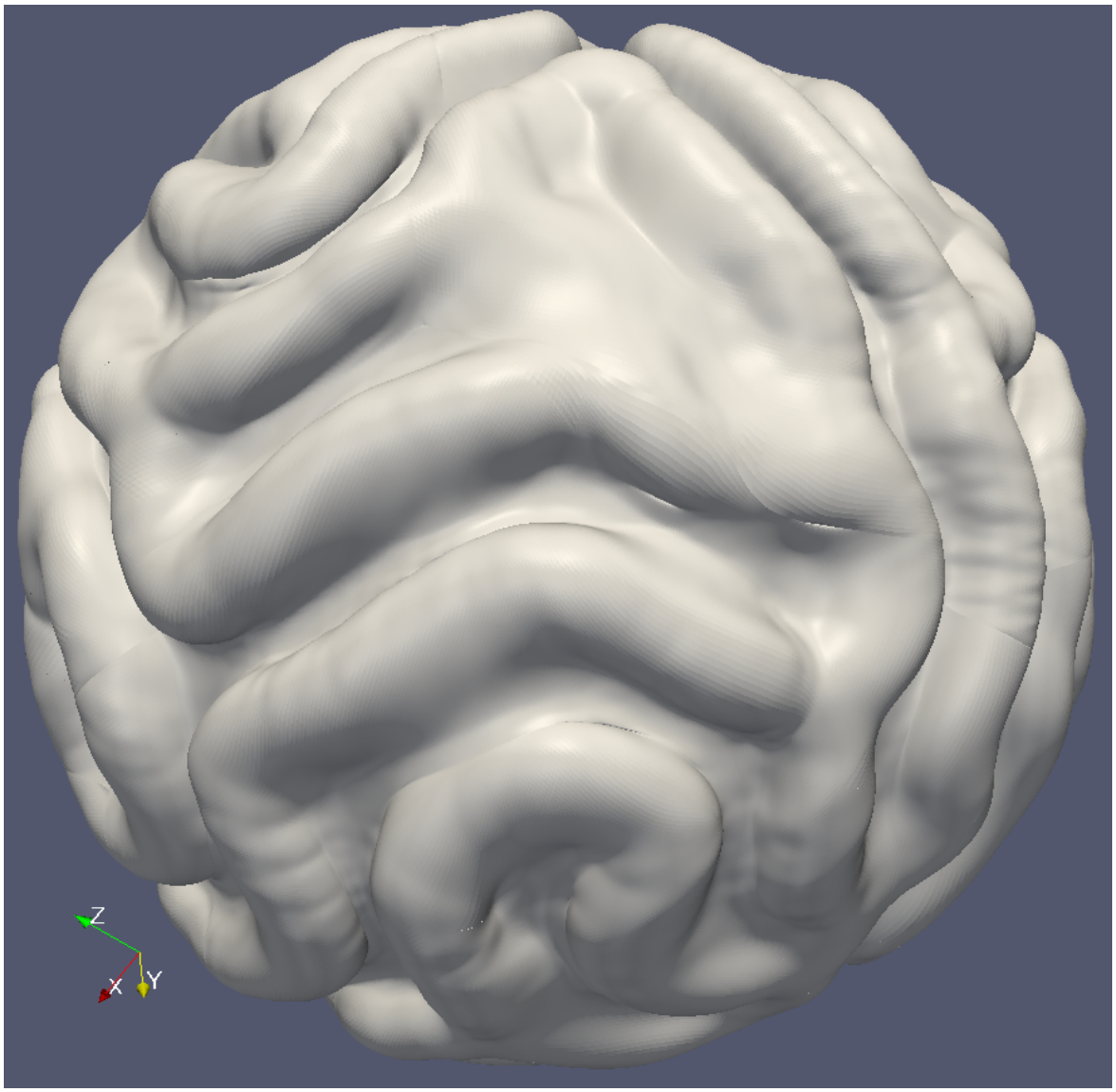


Figure 11.11: The final geometry for $F = 0.04$ at $t^{1650} \simeq 2 \times 10^4$.

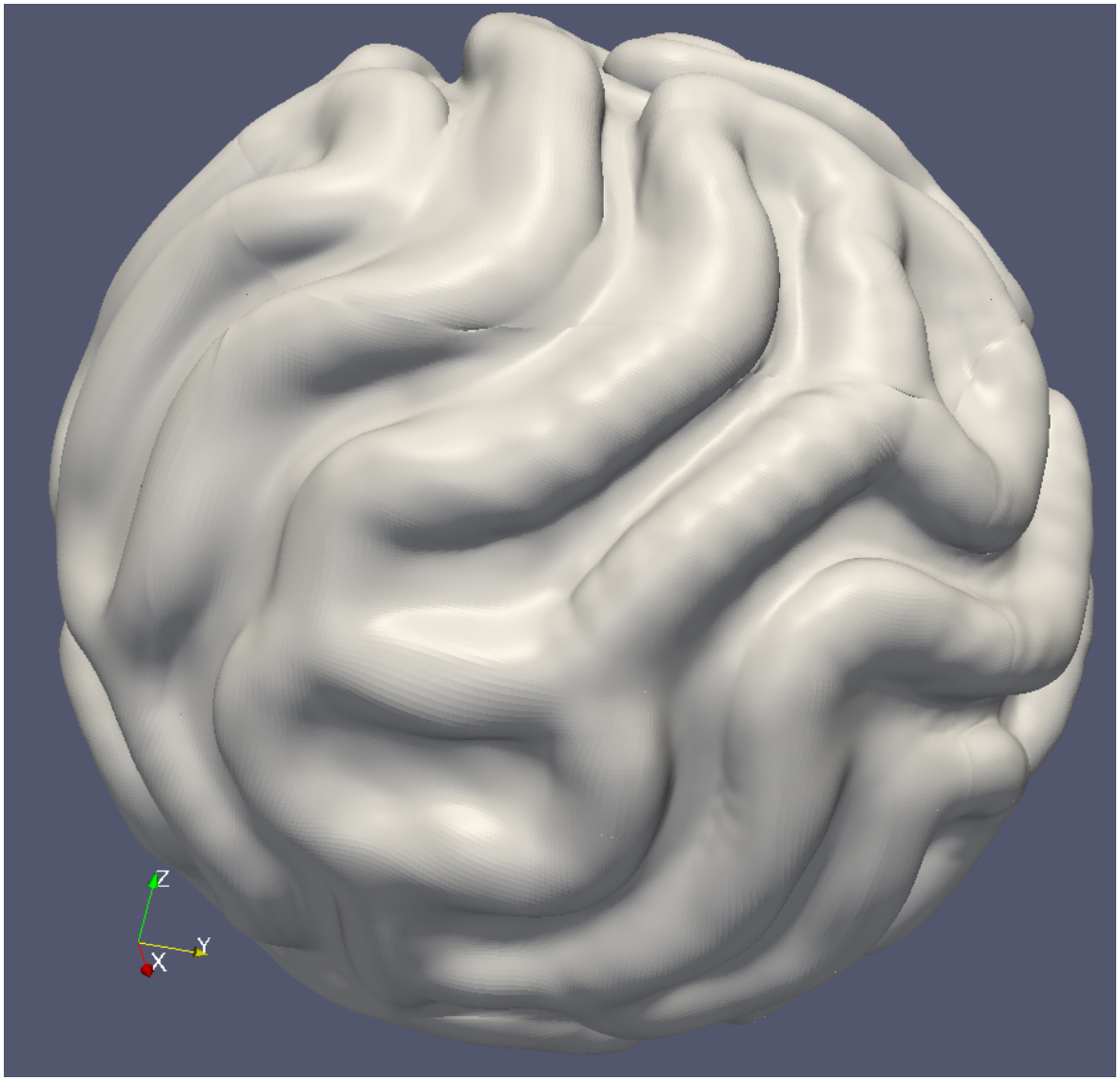


Figure 11.12: The geometry after the last iteration for hte simulation with $F = 0.04$.

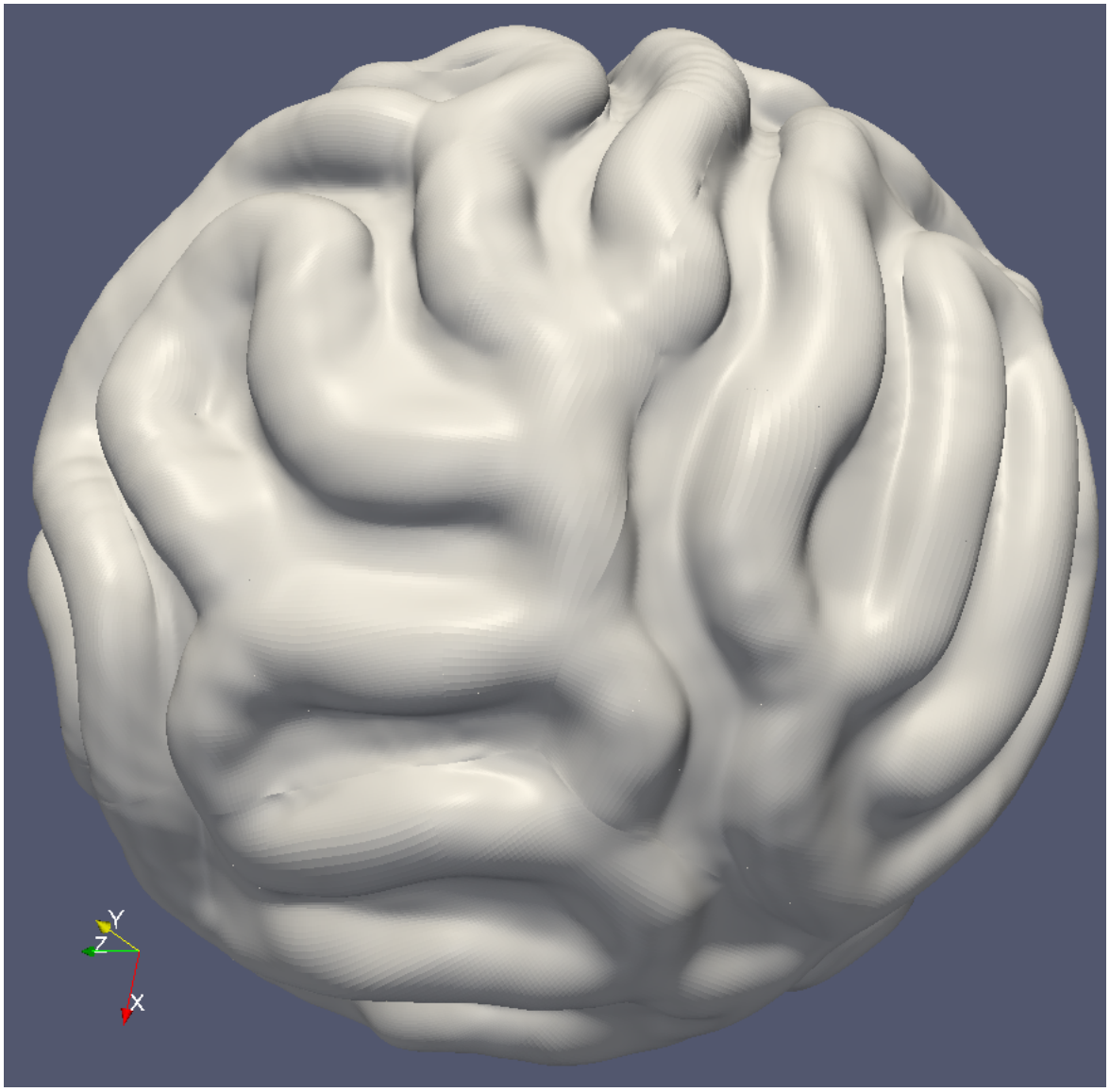
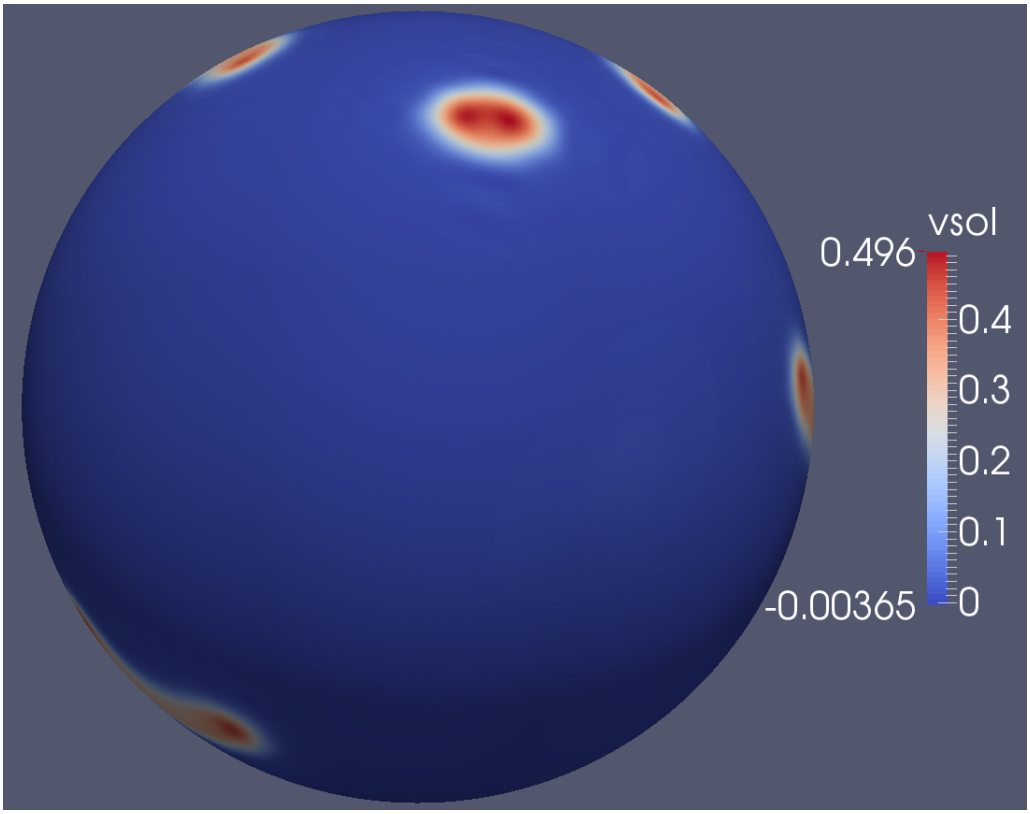


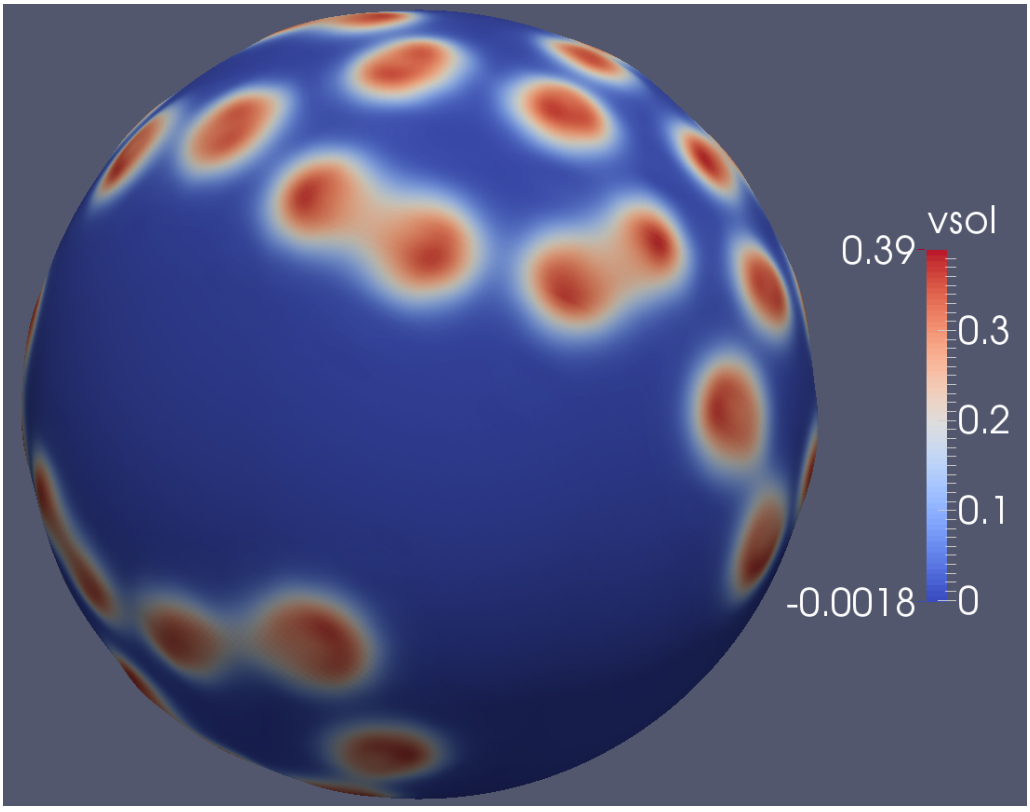
Figure 11.13: The geometry after the last iteration for the simulation with $F = 0.04$.

11.2.2. $F = 0.0285$

For a simulation with $F = 0.0285$, figures 11.14, 11.15 and 11.16 show the state in which V^k and the geometry $\mathcal{M}_k$ find themselves for various t^k. As opposed to figure 11.8 (a), figure 11.14 (a) does not show a connected band of high concentration but only a number small dots. In (b) the amount of dots increases. The amount of dots increases futher in 11.15 and we also see one little narrow strip. We see the first signs of geometric deformation. In figure 11.16 the deformations intensify and we see that the solver has refined many of the dots, due to curvature or cell size. Figures 11.17 to 11.19 show the geometry after the last iteration from various angles. The simulation has been terminated after 1930 iterations.



(a)



(b)

Figure 11.14: Substrate V and the geometry for $F = 0.0285$ at (a) $t^{400} = 224$ and (b) $t^{800} = 2050$.

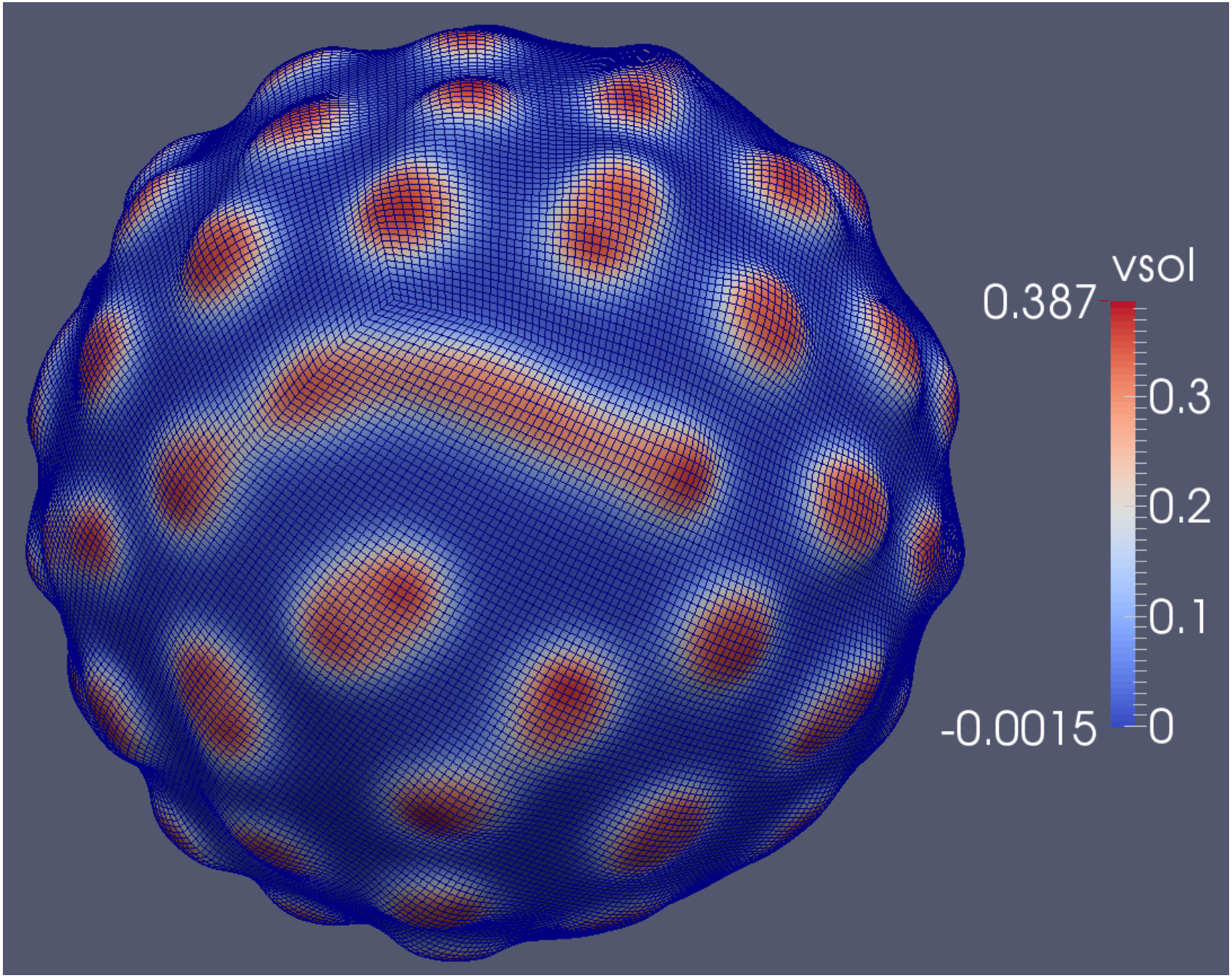


Figure 11.15: Substrate V and the geometry for $F = 0.0285$ at $t^{1200} = 7.6 \times 10^3$.

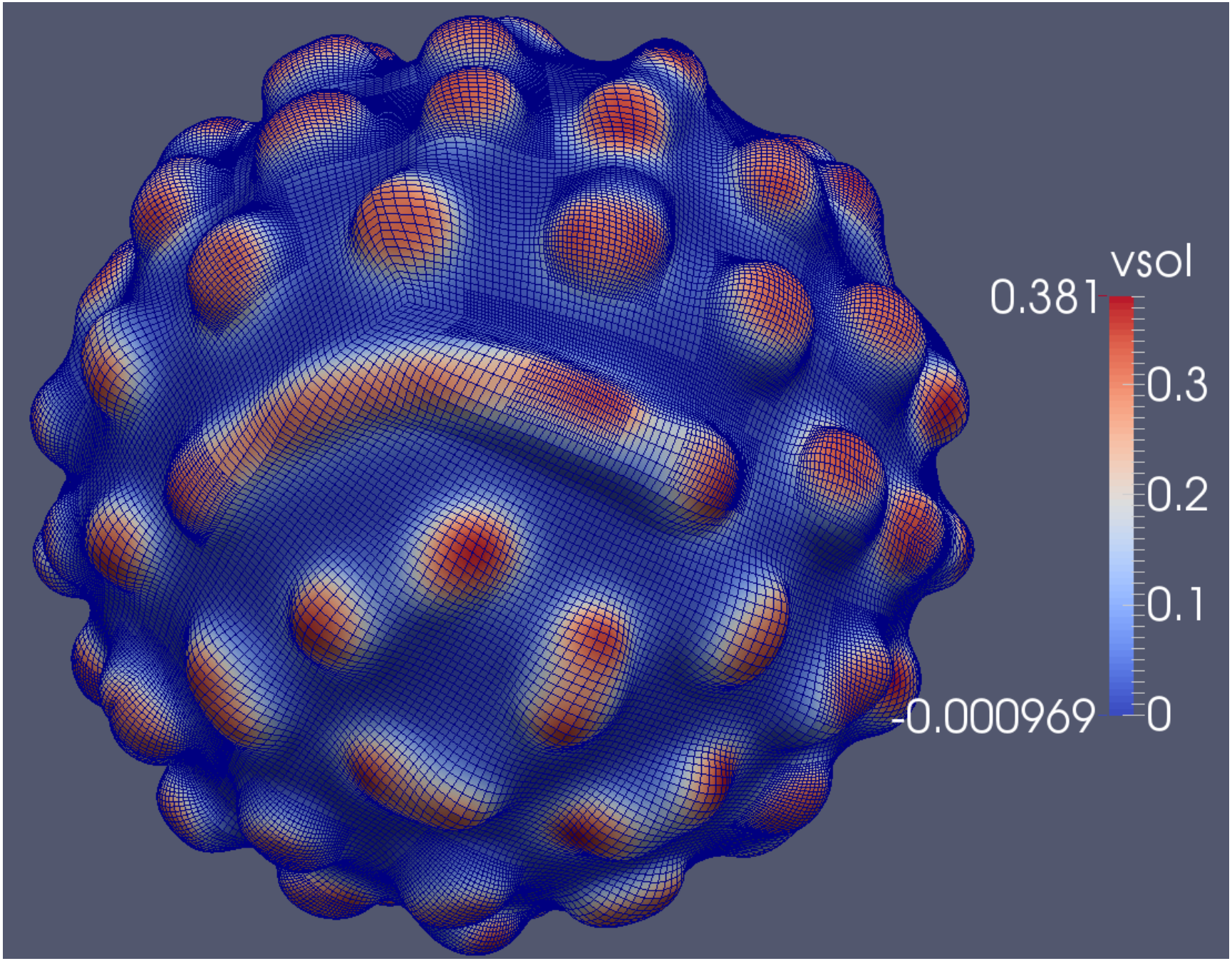


Figure 11.16: Substrate V and the geometry for $F = 0.0285$ at $t^{1600} = 1.54 \times 10^4$.

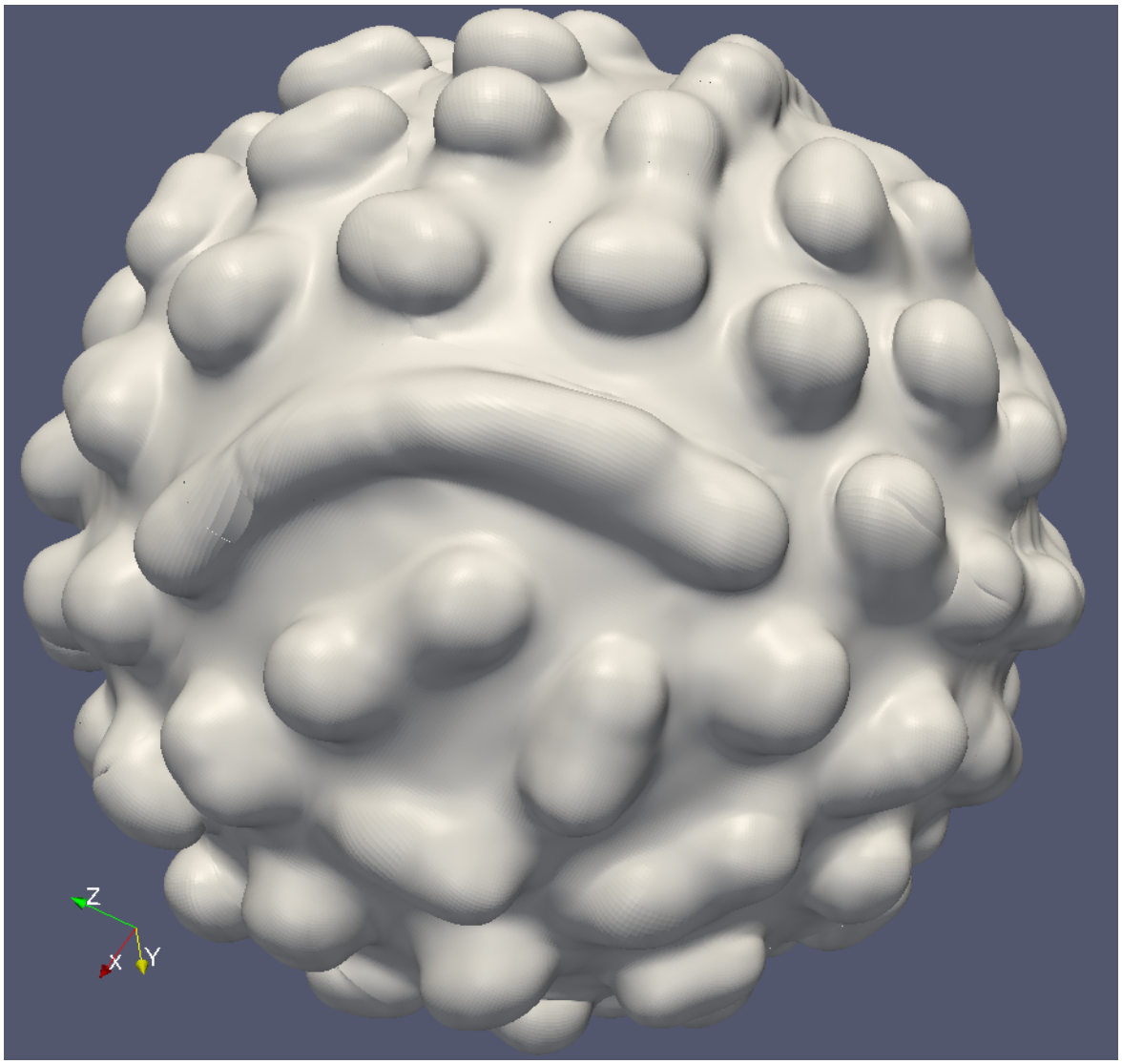


Figure 11.17: The final geometry for $F = 0.0285$ at $t^{1930} \simeq 2 \times 10^4$.

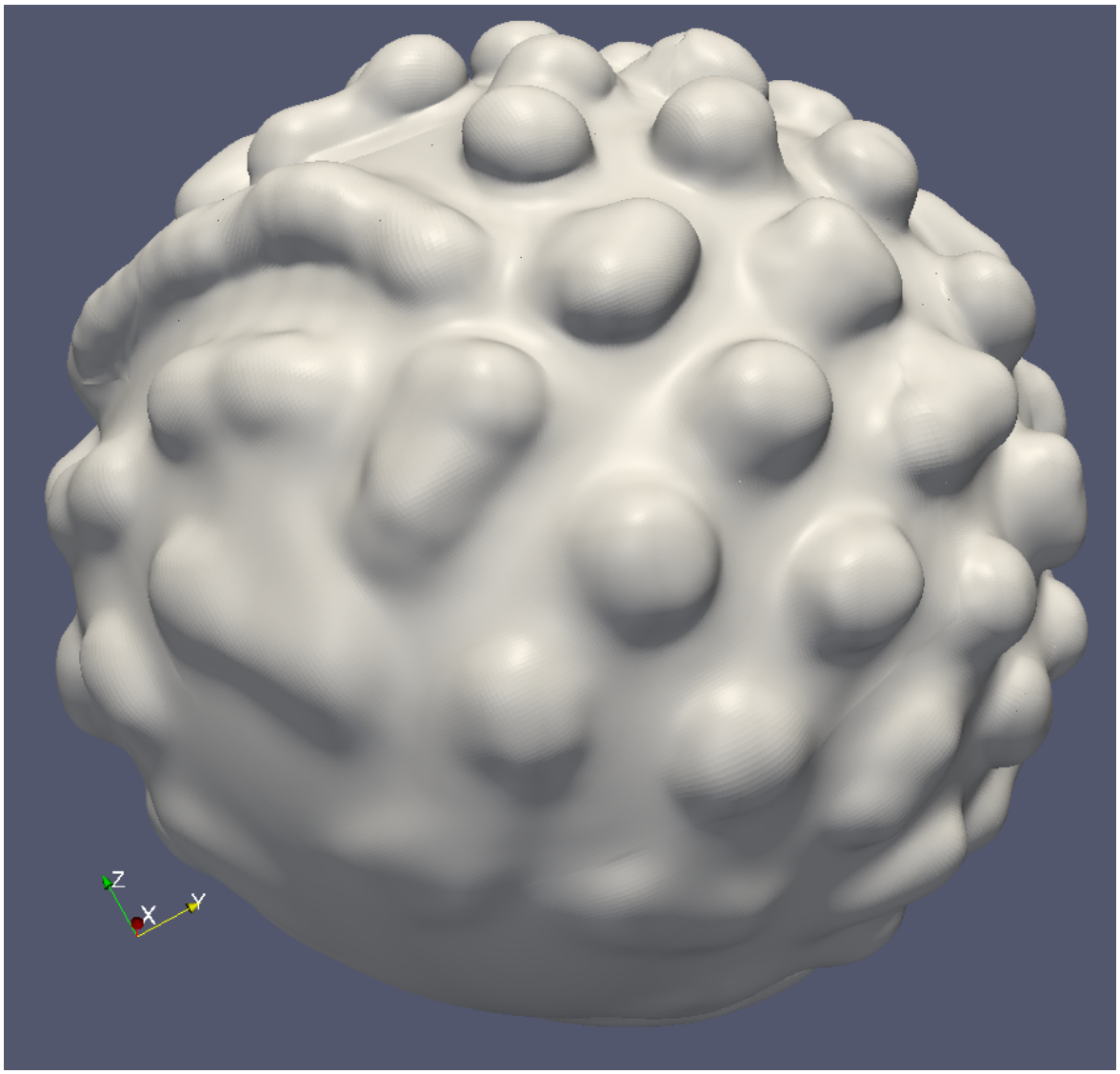


Figure 11.18: The geometry after the last iteration for the simulation with $F = 0.0285$.

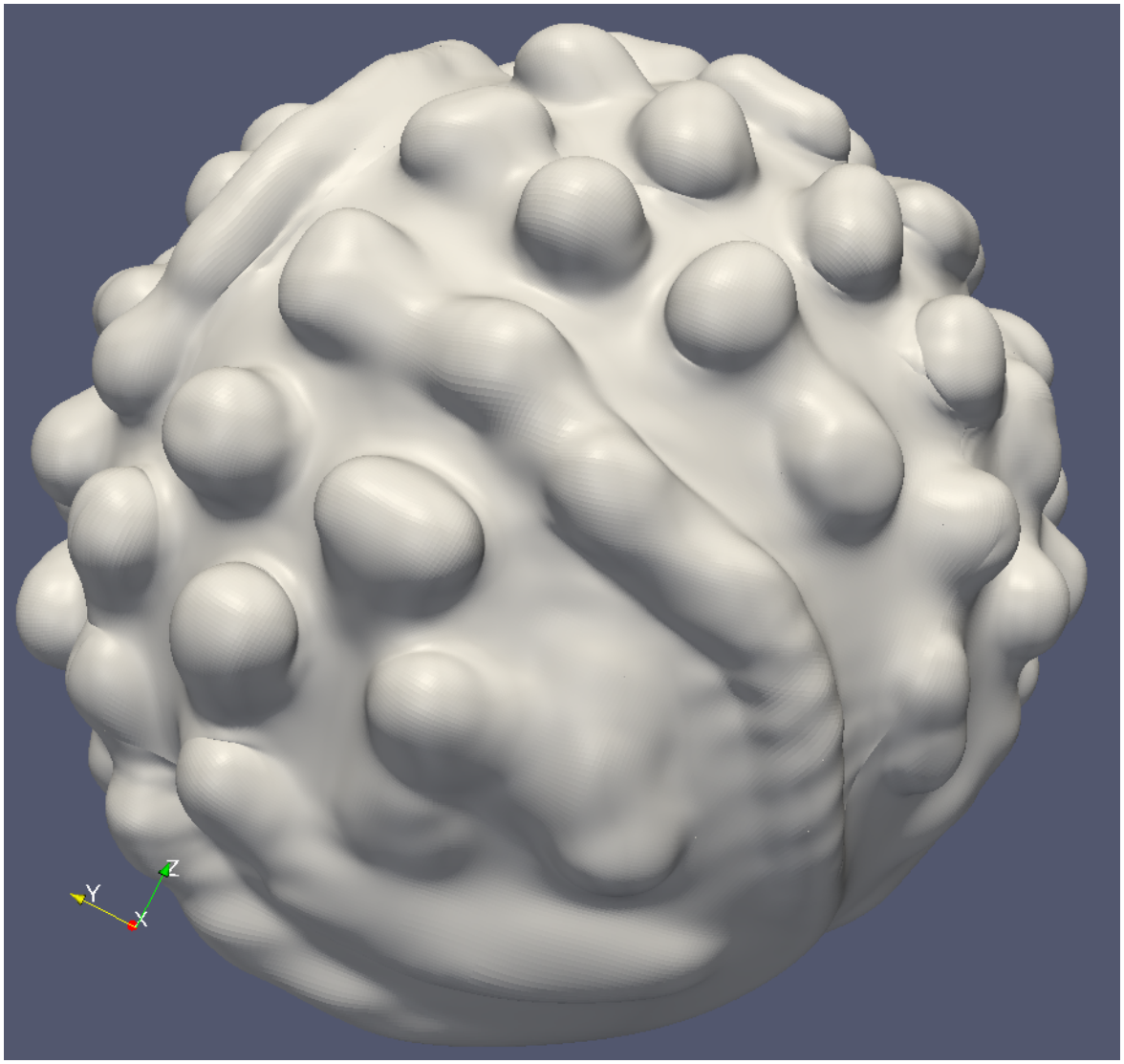
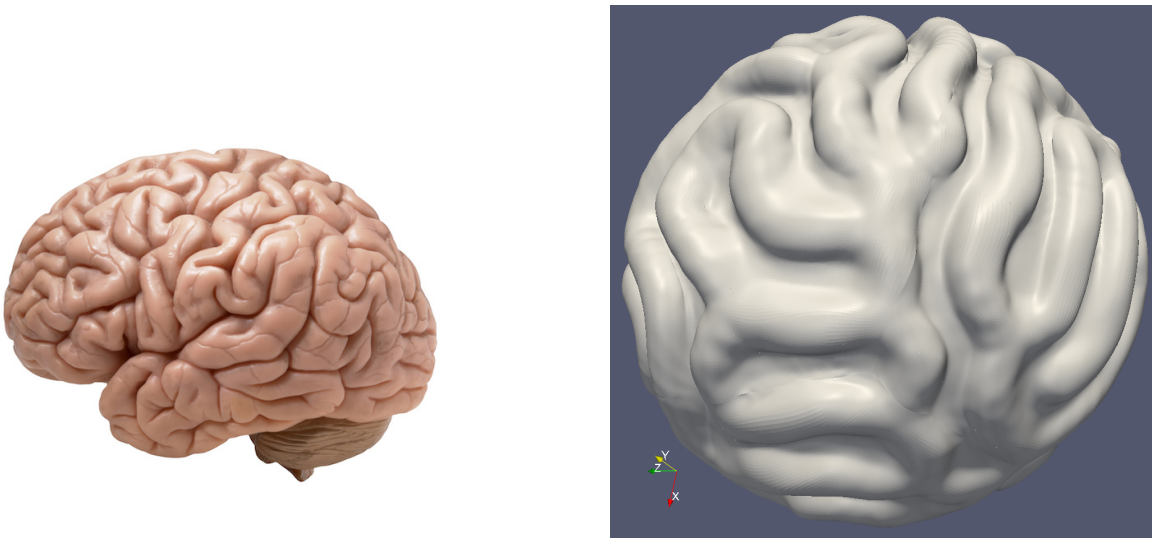


Figure 11.19: The geometry after the last iteration for the simulation with $F = 0.0285$.

11.3. Discussion

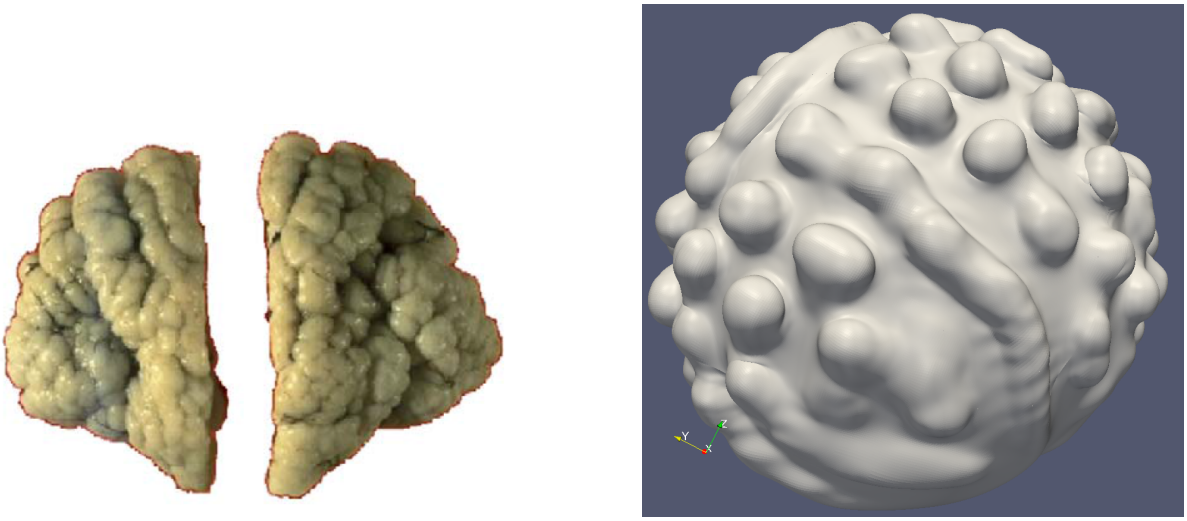
The numerical scheme from chapter 7 has been succesfully implemented on both the torus and the sphere and the results meet the expectations. Pattern formation of the cencentrations is present on both geometries and the patterns manifest themselves in surface deformations. The deformations show a high degree of resemblence to typical brain patterns found in healthy adult individuals for $F = 0.04$ on both the torus and the (gaming) sphere (see figures 11.20 and 11.4).



(a) (b)

Figure 11.20: Typical neural patterns in a healthy brain (a) and the results from the simulation with $F = 0.04$ on the gaming sphere (b).

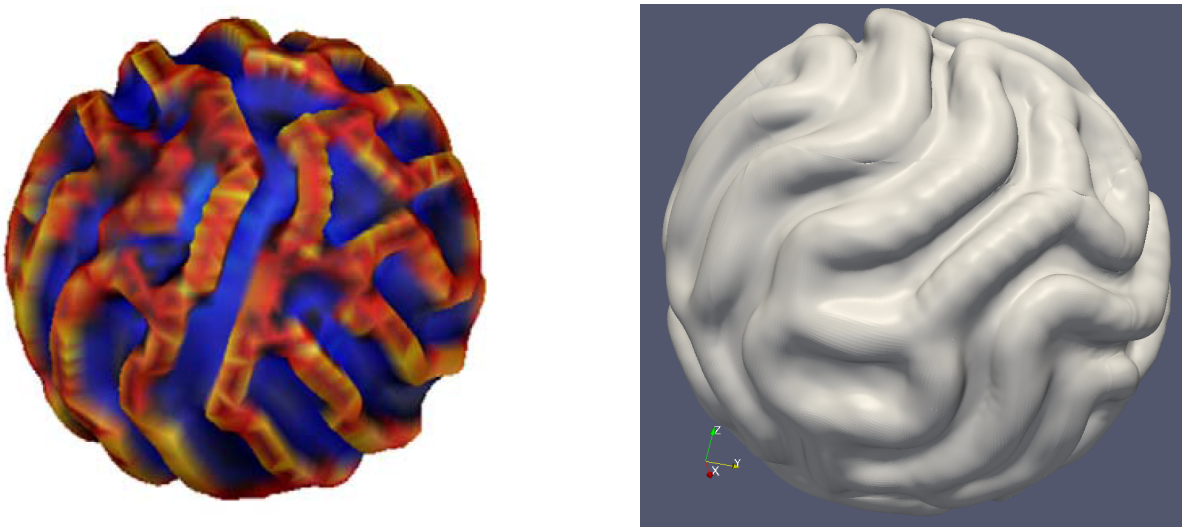
The implementation with $F = 0.0285$ shows mild resemblence with the neuropathology *polymicrogyria* (see figure 11.21), which suggests that within the framework of this model, neuropathologies can be explained by deviations of the reaction rate F due to genetical anomalies or other extrinsic influences.



(a) (b)

Figure 11.21: Polymicrogyria (a) and the results from the simulation with $F = 0.0285$ on the gaming sphere (b).

With above simulations we were largely able to confirm the findings presented in [25]. Note that we have used a growth factor $K = 0.001$ as opposed to $K = 0.0005$ in [25]. Overall the results from the IgA-scheme, not surprisingly, exhibit improved smoothness when compared to their FEM-counterpart from [25] (see figure 11.22) and the smoothness greatly contributes to the overall visual appeal of the results. It is noteworthy that the time-scale necessary to achieve similarly-sized folds as in [25] is about a factor five larger even though the growth factor is doubled. A possible explanation is that we used a different initial condition.



(a) (b)

Figure 11.22: The resulting geometry of a simulation with the classical FEM-approach from [25] (a) and the geometry resulting from the IgA-scheme introduced in chapter 7 (b).

The effect of the patch boundaries is visible in figures 11.11 to 11.13 and manifests itself in 'kinks' at the transition between pairs of patches due to the locally reduced smoothness of the IgA basis from 9.3. This kink can lead to oppositely directed normal vectors which facilitates unphysical self-intersections of the geometry and other defects whenever the simulation is not terminated in time. Figure 11.23 shows the geometry resulting from a simulation using $F = 0.04$ and a different initial condition that was terminated after 1800 instead of 1650 iterations. Figure 11.23 shows stronger folding but a large amount of geometrical intersections and strong kinks occuring predominantly by the patch boundaries.
This suggest that the results can be further improved by a range of cosmetic interventions like local smoothing of either the geometry or the normal vector by the patch boundaries. The effect of the patch boundaries can be further reduced by the utilization of the basis from section 9.4, possibly in conjunction with smoothing by the security layers. That the absense of patch boundaries is likely to reduce the amount of defects is further substantiated by the fact that they are not visible in the results from the torus. The improved smoothness will most likely lead to greater visual appeal of the results but also to stronger folds as it may be possible to perform a larger amount of iterations without geometric clashing. Of course, the possibility of self-intersection may also be regarded as a shortcoming of the model. The effect of the patch boundaries on the concentrations will be studied in detail in chapter 12.

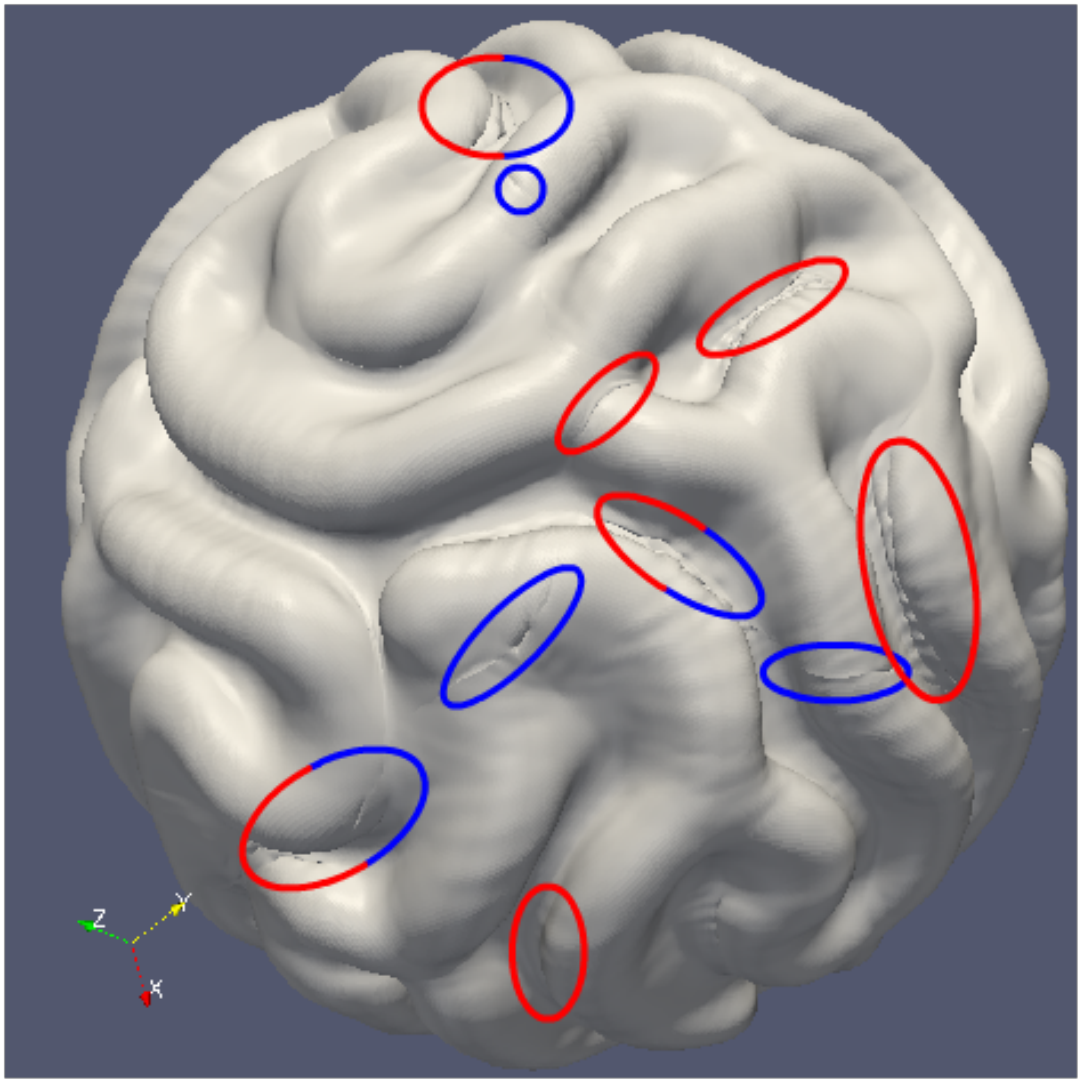


Figure 11.23: Typical defects that arise whenever the simulation is not terminated in time. The red circles show the spots at which the geometry intersects with itself and the blue circles show places with noticable kinks. The circles in red and blue show places geometric clashing that occurs by the patch boundaries.

12

Numerical Experiments

In this chapter, we present numerical experiments with which we aim to test the resolution offered by the time-stepping scheme from section 7.2 with corresponding spatial discretization from section 7.3 on an initial geometry given by a sphere. Again, we use the basis introduced in section 9.3 and the PID-controller from section 7.7. Apart from that, the main motivation for these experiments is to examine the effects of the patch boundaries and compare results of an implementation on the ordinary sphere and the gaming sphere. Unfortunately, the differential equation (7.1) is too complicated to derive an exact solution that could serve as a meaningful benchmark for the numerical scheme. This is why we will present qualitative results based on the following observation: since the differential equation, in the global sense, does not treat any direction with favoritism, in the presence of a spherically symmetric initial condition, the solution is expected to stay spherically symmetric (see figure 12.1).

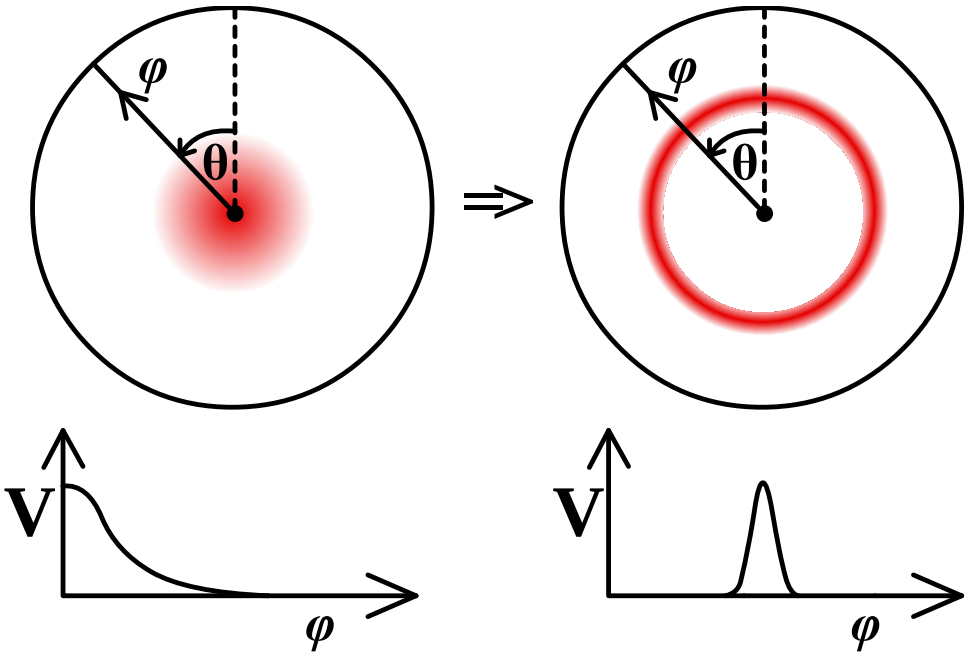


Figure 12.1: Given a spherically symmetrical initial condition (in both U and V), the solution should stay spherically symmetrical. Assuming there is a coordinate system with the z-axis intersecting the radial center of the initial condition, for fixed t, we can expect the exact solution to be a function of the azimuthal angle ϕ only

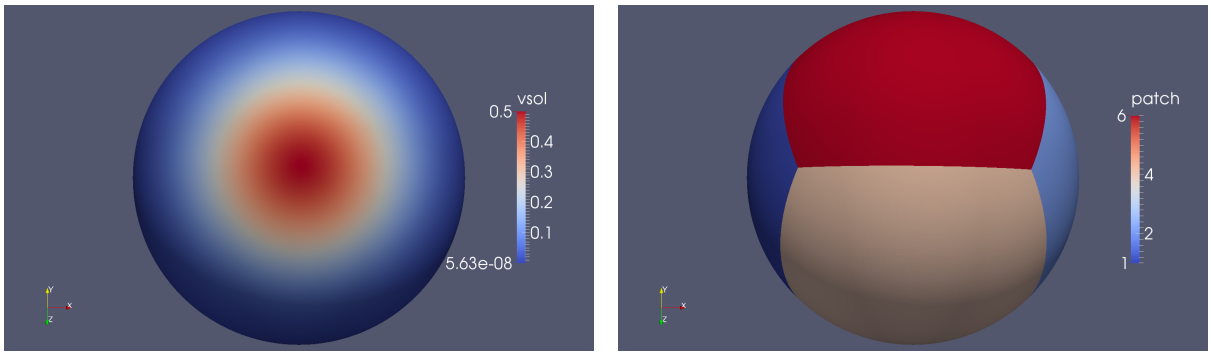
We adopt the same parameters as in section 11.2 with $F = 0.04$.
We will present numerical experiments for $p = 2$ and $p = 3$ with $n = 12$, $n = 22$, $n = 32$ and $n = 42$ (see section 9.3). Plots will be shown of the initial condition V^0 ($k = 0$), V^{100}, V^{200} and V^{300}. U and V are initialized to the Gaussians (in a global sense)

$$U(t=0) = 1 - 0.75 \exp\left(-\sum_{i=1}^{3}\left(\frac{x_i - x_i^0}{20}\right)^2\right)$$
$$V(t=0) = 0.5 \exp\left(-\sum_{i=1}^{3}\left(\frac{x_i - x_i^0}{20}\right)^2\right), \tag{12.1}$$

where

$$\mathbf{x}^0 = R\left(0, \sin\frac{\pi}{4}, \cos\frac{\pi}{4}\right)^T. \tag{12.2}$$

The center of the Gaussian(s) from (12.1) is through (12.2) located exactly on the border where two patches meet (see figure 12.2).



(a) Initial condition in V

(b) Various patches highlighted in different colors

Figure 12.2: Position of the initial condition

Before the time-stepping procedure commences, the first iterands u^0, v^0 and $\mathbf{s}^0$ are constructed from a projection of (12.1) onto Σ and by a projection of either $\mathbf{s}_2 \circ \mathbf{s}_1$ (ordinary sphere) or $\mathbf{s}_2^* \circ \mathbf{s}_1$ (gaming sphere, see section 9.2) onto the same basis Σ, initialized to $\Sigma \leftarrow \mathscr{B}_1$. Since substrate V is the driving force behind surface derformation, we will focus our attention on V and all figures will depict the state in which V finds itself.

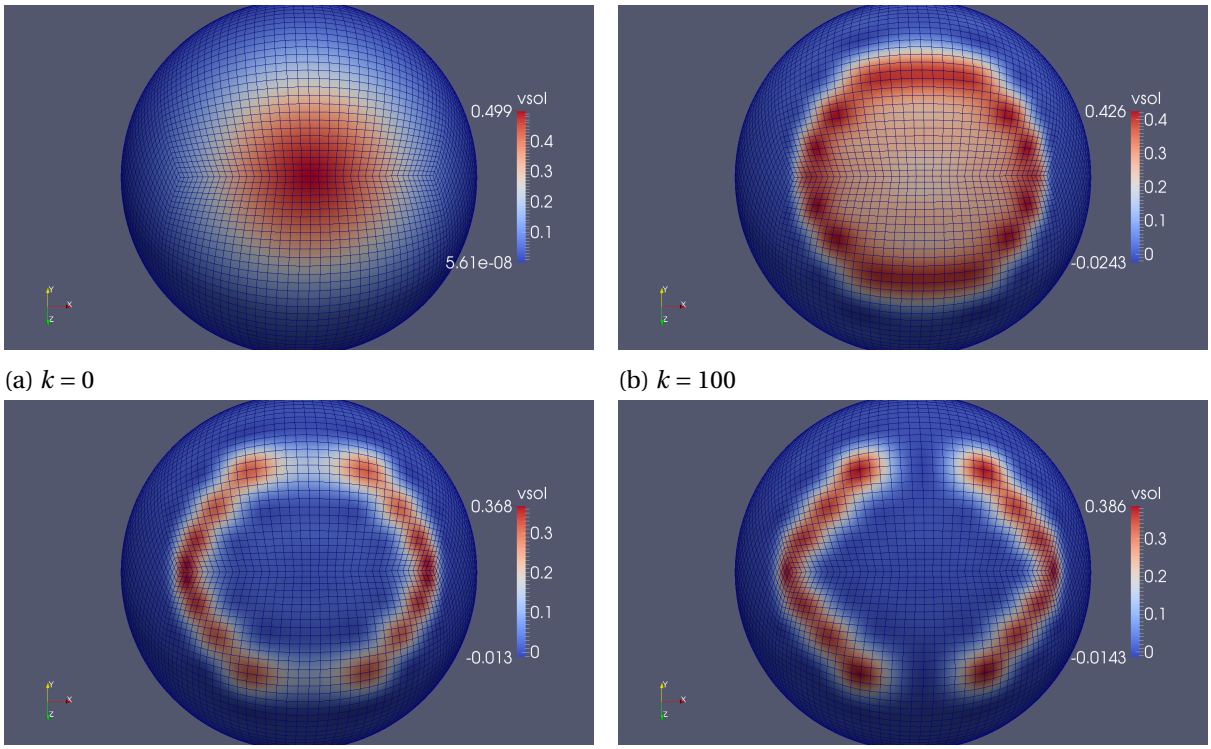
Remark. *Since the PID-controller from section 7.7 selects step-sizes based on the behavior of u^k and v^k an equal amount of iterations does not necessarily correspond to an equal amount of elapsed time between different simulations. We will bluntly ignore this issue for the sake of convenience. Heuristically, the time frames after* 300 *iterations fall within* 20% *deviation of one another.*

A color-coded legend is depicted on the right of each picture. It shows the minimum and the maximum value (over all evaluation points) that V^k assumes on $\mathscr{M}_k$.

12.1. Numerical Experiments for $p = 2$

In this section, we present numerical experiments for bases of order two (in either direction). We start off by presenting results for $n = 12$, $n = 22$, $n = 32$ and $n = 42$ on the ordinary sphere and continue with results from the same parameters on the gaming sphere.

12.1.1. Ordinary Sphere



(a) $k = 0$

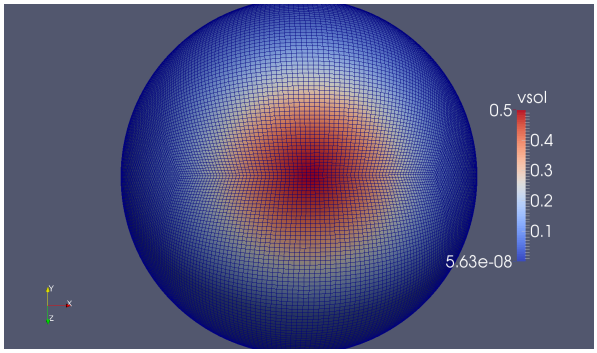
(b) $k = 100$

(c) $k = 200$

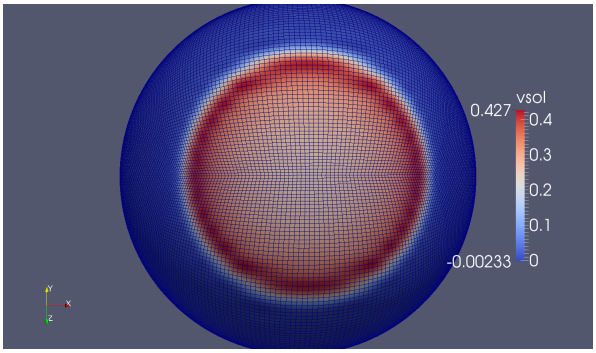
(d) $k = 300$

Figure 12.3: V^k for $n = 12$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.

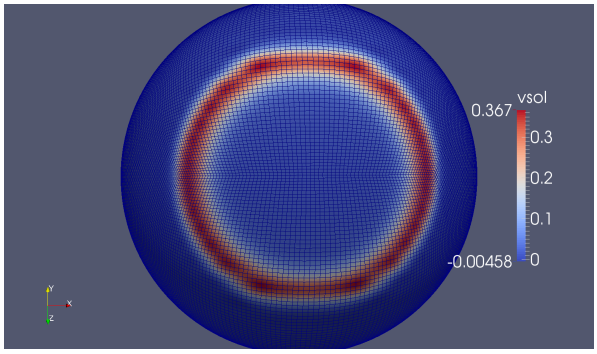
In figure 12.3, we see strong blurring at the wavefront of V^{100}, it is also seen that spherical symmetry already starts to suffer. The presence of individual basis functions is clearly noticable. In V^{200}, symmetry is clearly broken as the wavefront separates into two pieces which intensifies after 300 iterations. Apart from the low resolution, figure 12.3 shows a noticable fringing-effect along the patch boundary of the first iterand. That the separation of V^k aligns with this patch boundary might be an after-effect of this fringing.
In figure 12.4, as a result of the higher resolution, we see less fringing of the first iterant. V^{100} shows effects similar to 12.3 (b) but significantly reduced. In (c), spherical symmetry starts to break as the wavefront starts to form a shape that more closely resembles an ellipse with its major axis aligned with the patch boundary. This effect intensifies in V^{300}.
Figure 12.5 (a) exhibits no fringing effects. Spherical symmetry is still present after 100 and 200 iterations along with relatively sharp wavefronts. Symmetry, however, breaks after 300 iterations in a way similar to figure 12.4 (d).
Just as figure 12.5, figure 12.6 shows unnoticable initial fringing. Strong symmetry is present in (b) as well as (c), along with evident sharpness of the wavefront. Even (d) still clearly shows spherical symmetry along with sharpness at both the inner and outer radius of the wavefront.



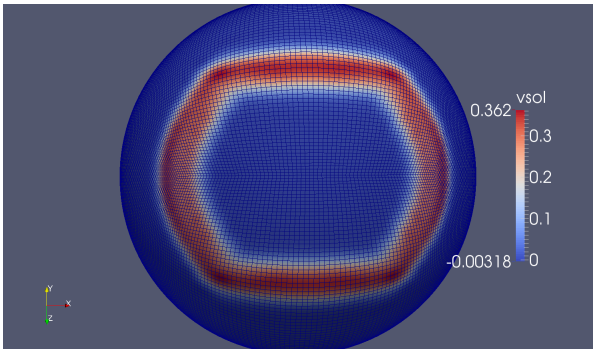
(a) $k = 0$



(b) $k = 100$

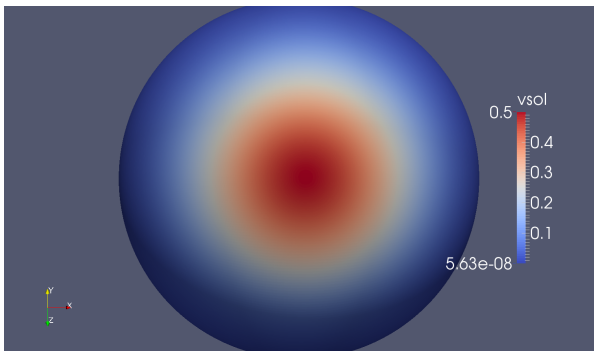


(c) $k = 200$

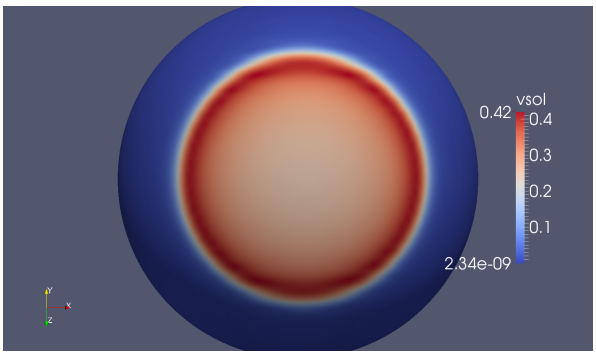


(d) $k = 300$

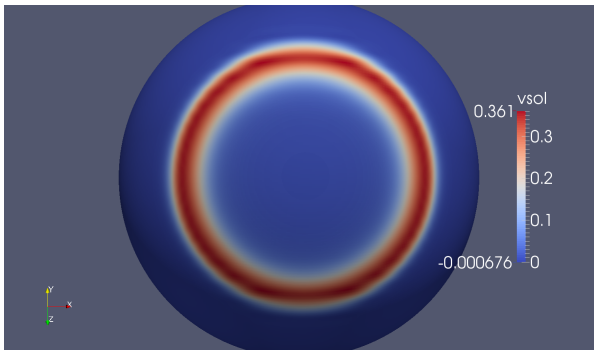
Figure 12.4: Results for $n = 22$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.



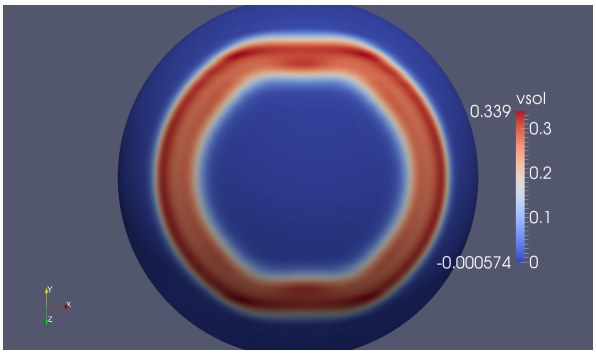
(a) $k = 0$



(b) $k = 100$

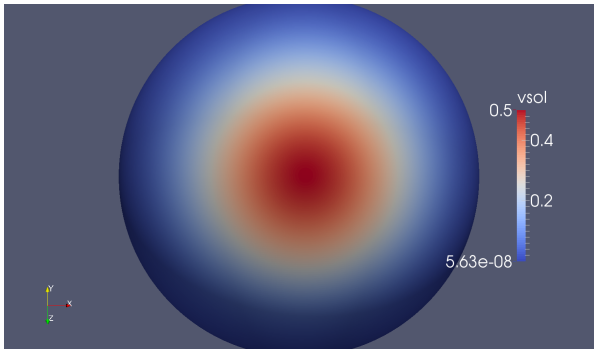


(c) $k = 200$

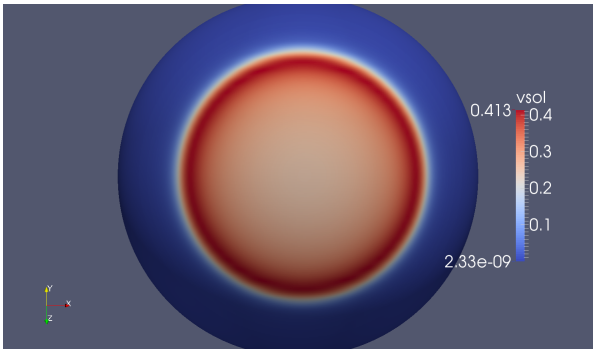


(d) $k = 300$

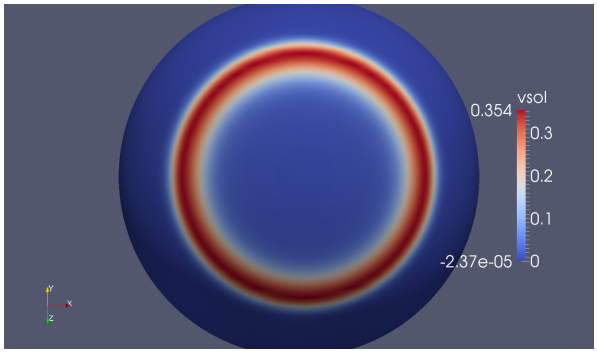
Figure 12.5: Results for $n = 32$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.



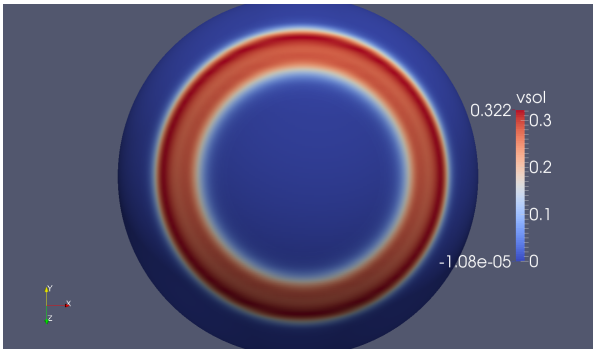
(a) $k = 0$



(b) $k = 100$



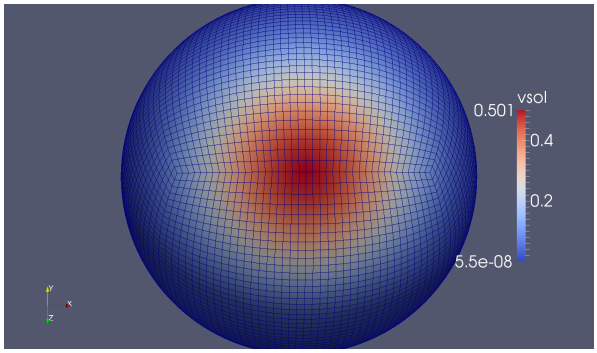
(c) $k = 200$



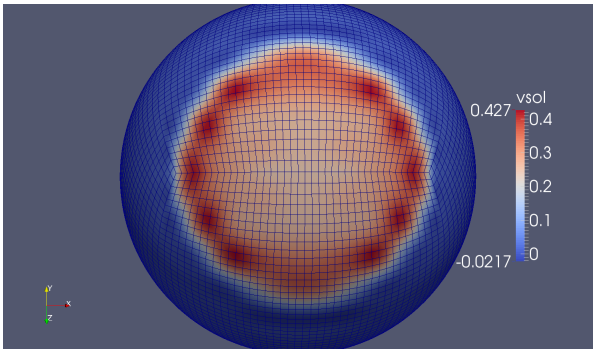
(d) $k = 300$

Figure 12.6: Results for $n = 42$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.

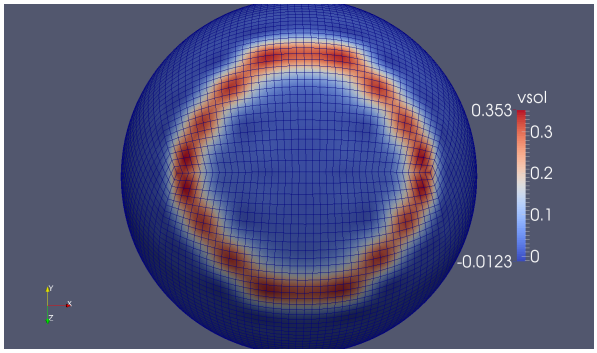
12.1.2. Gaming Sphere



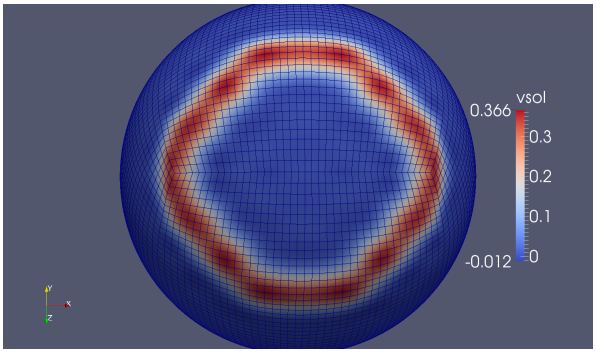
(a) $k = 0$



(b) $k = 100$



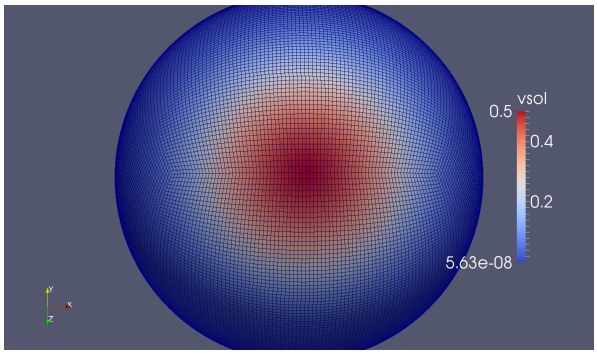
(c) $k = 200$



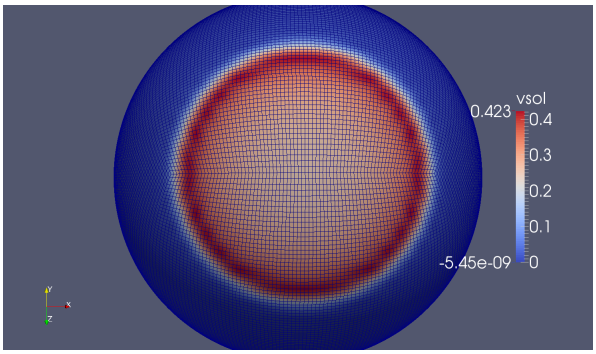
(d) $k = 300$

Figure 12.7: Results for $n = 12$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.

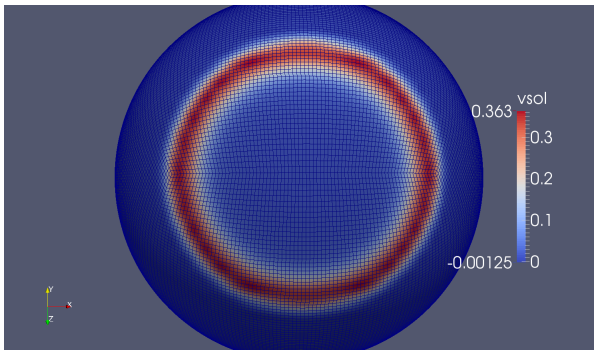
In figure 12.7, the fringing effects are somewhat reduced when compared to figure 12.3 (a). In V^{100}, blurring at the wavefront is comparable to figure 12.3 (b), spherical symmetry, however, seems to be slightly improved. In V^{200}, symmetry is clearly broken as the wavefront forms a rounded sphere and the effect intensifies in (c). Note, however, that unlike in figure 12.3, separation does not occur. This might support the hypothesis that after-effects of the initial fringing are suppressed on the gaming sphere.
Figure 12.8 shows little fringing in the first iterant. Spherical symmetry as well as a resonably sharp wavefront are still present in (b). Figure 12.8 (c) still exhibits decent symmetry and sharpness, in (d) however, symmetry is broken in a fashion similar to figure 12.4 (d). Note however, that the effect is significantly reduced when compared to its counterpart from the ordinary sphere.
Figure 12.9 (a) (b) and (c) show strong symmetry as well as sharpness. In (d), however, symmetry is slightly broken at the inner circle of the wavefront at the places where the patches meet. Overall, figure 12.9 shows better results than figure 12.5.
Figure 12.10 shows symmetry as well as sharpness in all of its subplots. Even the inner part of the wavefront in (d) is still noticably symmetrical. Overall the results are comparable to figure 12.6.



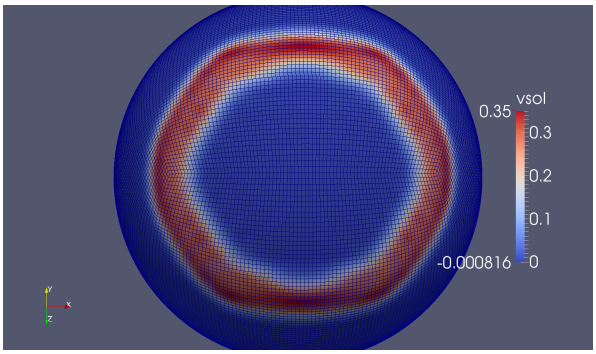
(a) $k = 0$



(b) $k = 100$

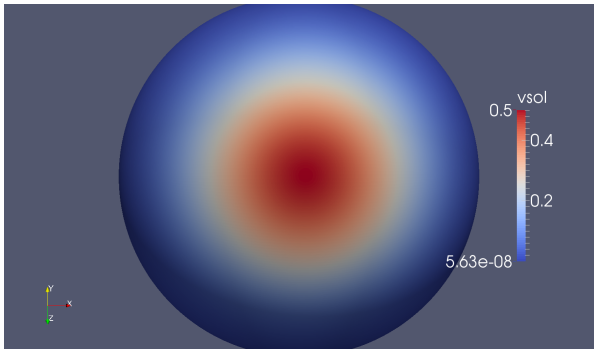


(c) $k = 200$

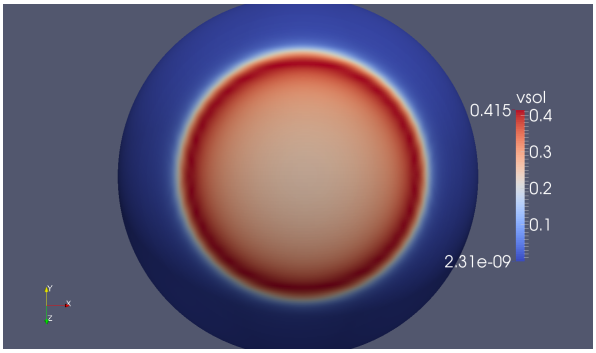


(d) $k = 300$

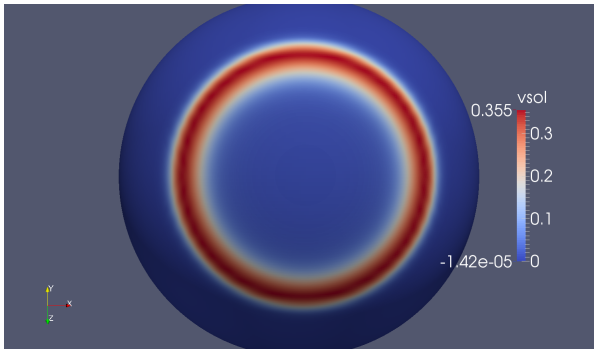
Figure 12.8: Results for $n = 22$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.



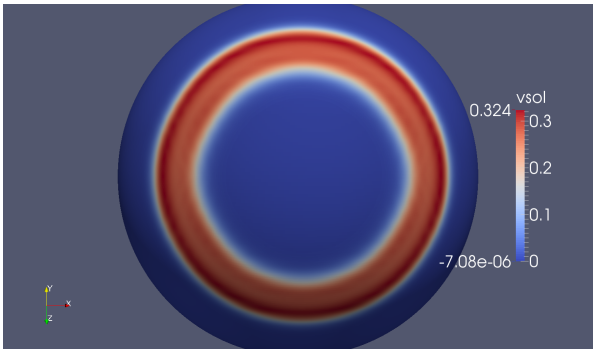
(a) $k = 0$



(b) $k = 100$

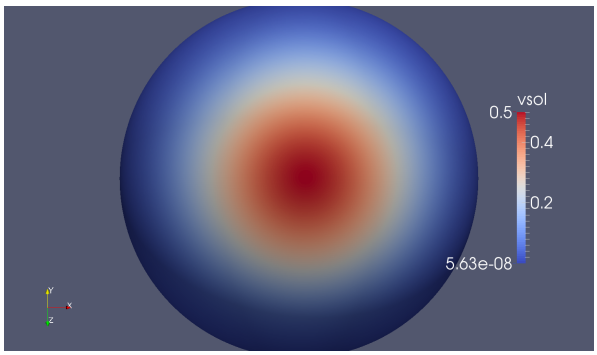


(c) $k = 200$

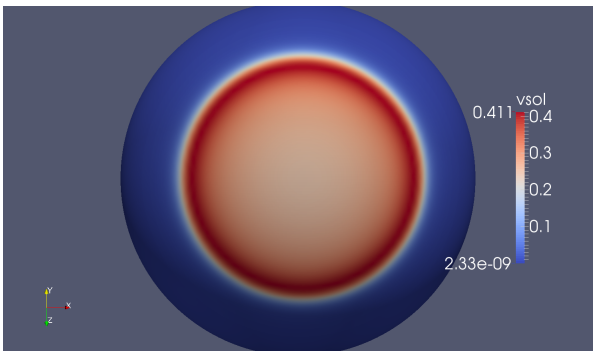


(d) $k = 300$

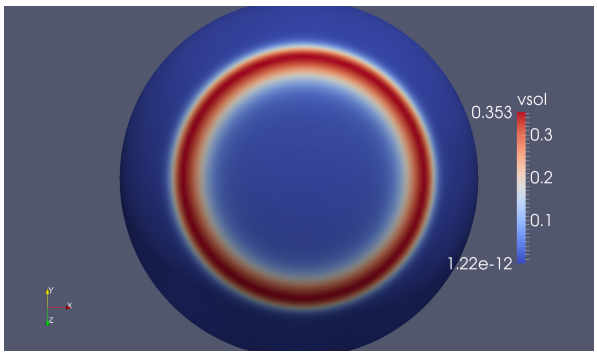
Figure 12.9: Results for $n = 32$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.



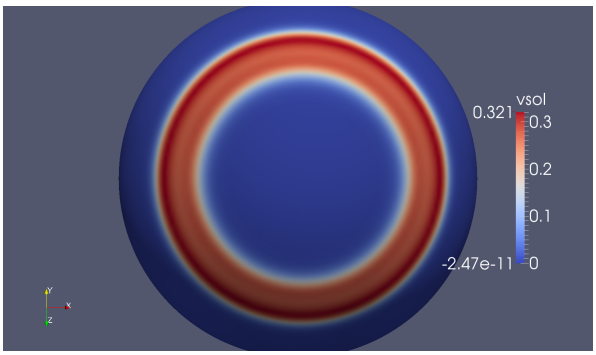
(a) $k = 0$



(b) $k = 100$



(c) $k = 200$



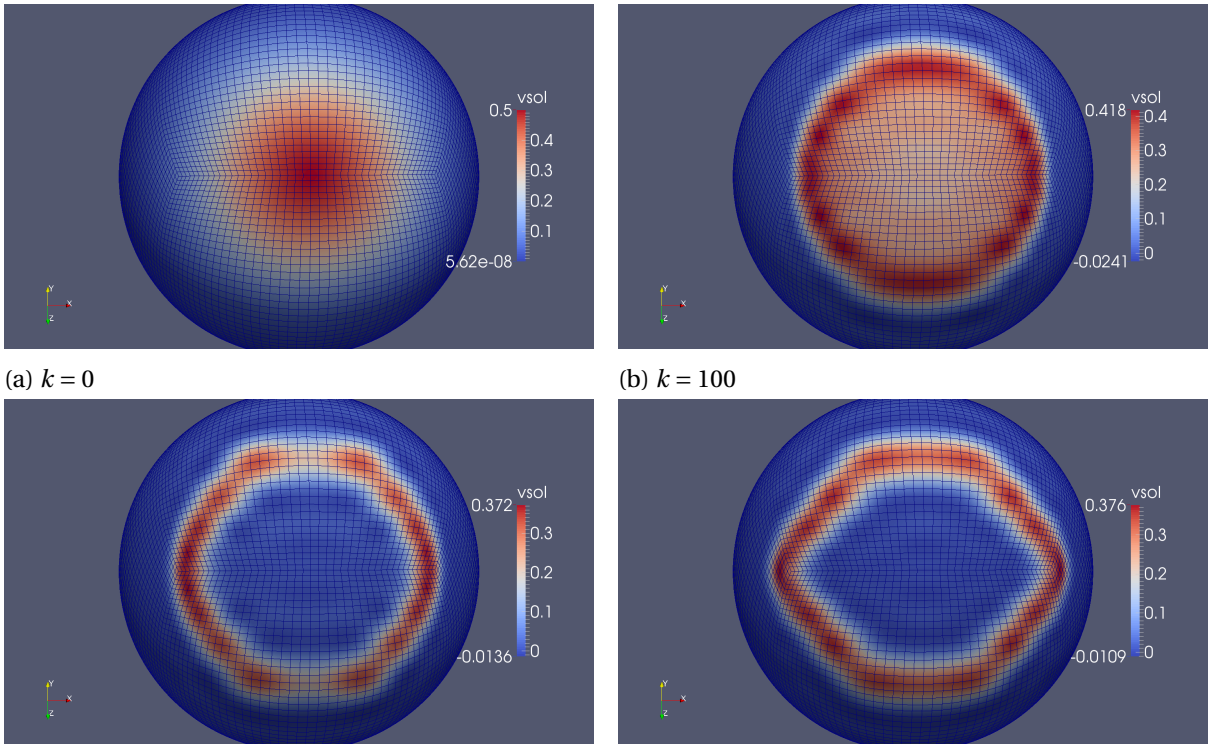
(d) $k = 300$

Figure 12.10: Results for $n = 42$, $p = 2$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.

12.2. Numerical Experiments for $p = 3$

In this section, analogous to section 12.1, we present numerical experiments for bases of order three (in either direction) for $n = 12$, $n = 22$, $n = 32$ and $n = 42$ on the ordinary sphere as well as the gaming sphere.

12.2.1. Ordinary Sphere



(a) $k = 0$

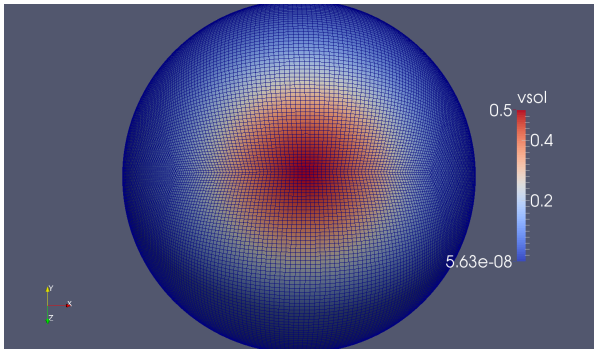
(b) $k = 100$

(c) $k = 200$

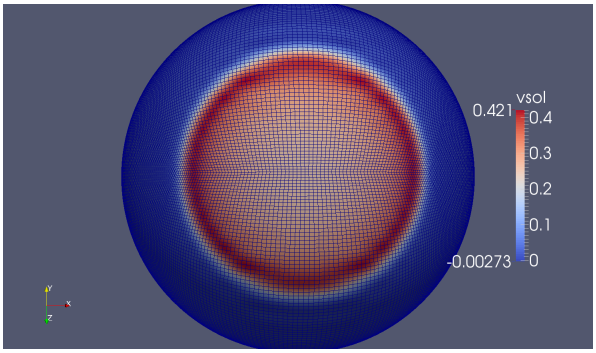
(d) $k = 300$

Figure 12.11: Results for $n = 12$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.

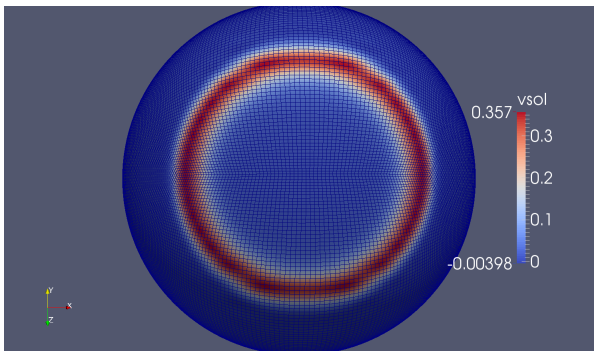
As before, figure 12.11 (a) shows strong fringing effects. The wavefront in (b) is unsharp but symmetry is still decent. After $k = 200$ the wavefront seems to separate, in (d), however, the separation has been reversed. Overall, the results are slightly better than in figure 12.3, even though symmetry is clearly compromised in (d).
Figure 12.12 shows results that are comparable to figure 12.8. (b) and (c) both show decent symmetry and sharpness which is compromised in (d).
Figure 12.13 b) and (c) show strong symmetry as well as sharpness but (d) shows slight symmetry breaking, especially at the inner circle of the wavefront. Overall the results are better than those from figure 12.5 but slightly worse than figure 12.9.
Finally, figure 12.14 shows sharpness as well as symmetry in (a) through (d). Overall the results are comparable to figures 12.6 and 12.10.



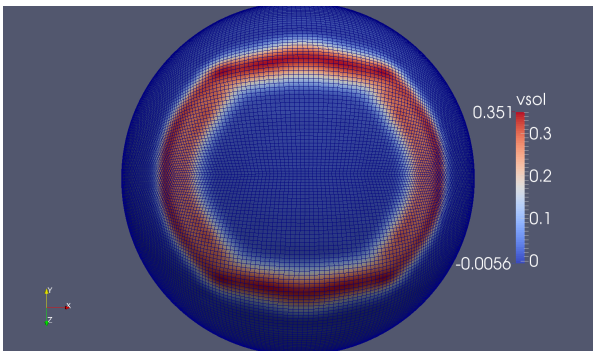
(a) $k = 0$



(b) $k = 100$

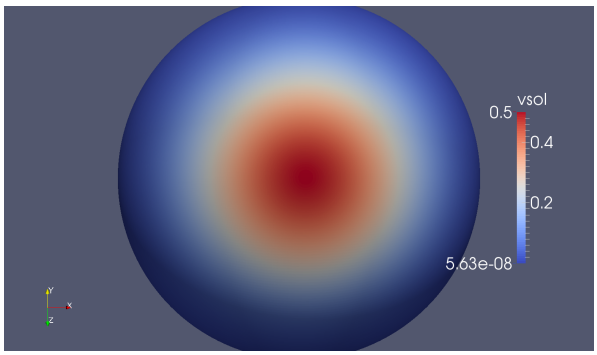


(c) $k = 200$

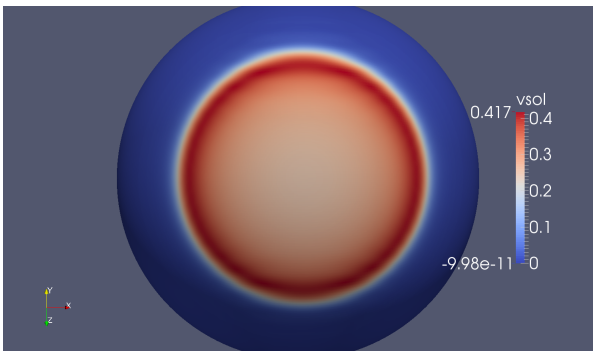


(d) $k = 300$

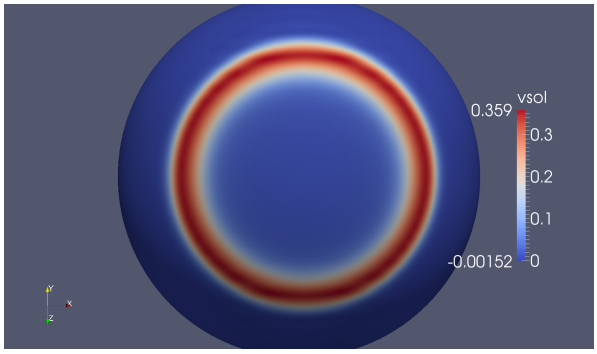
Figure 12.12: Results for $n = 22$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.



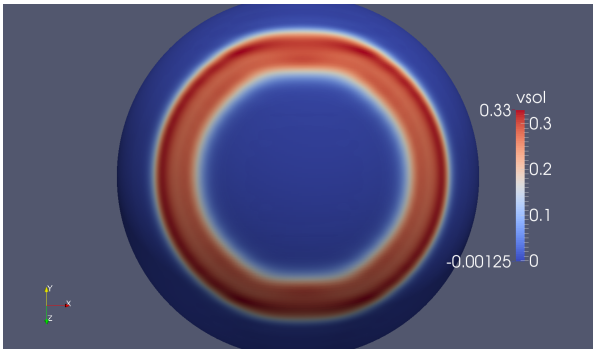
(a) $k = 0$



(b) $k = 100$

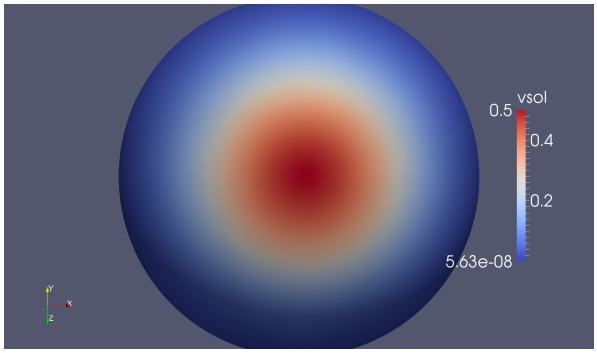


(c) $k = 200$

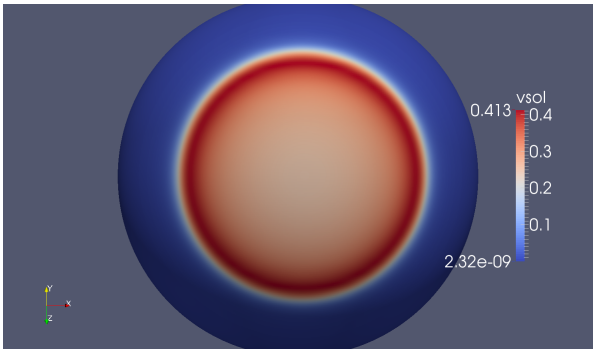


(d) $k = 300$

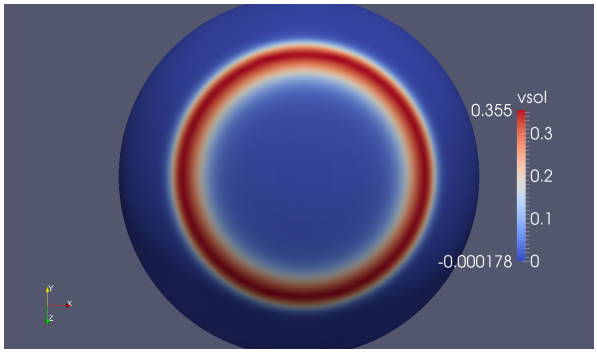
Figure 12.13: Results for $n = 32$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.



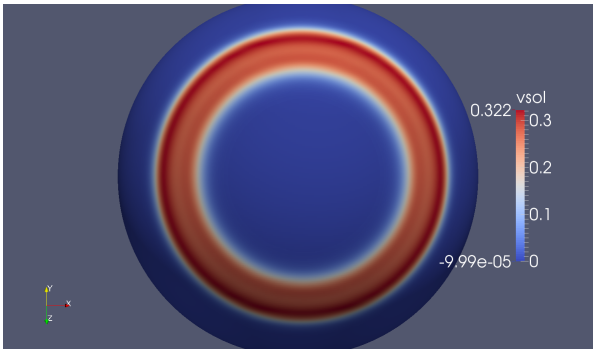
(a) $k = 0$



(b) $k = 100$



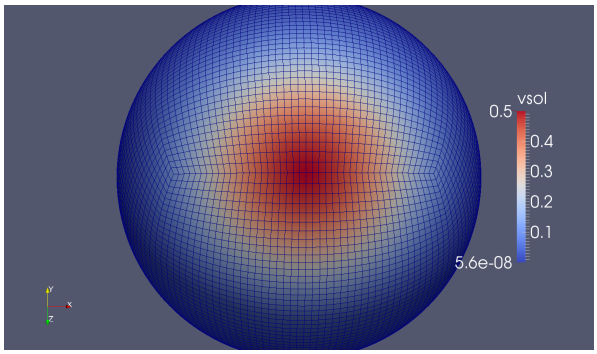
(c) $k = 200$



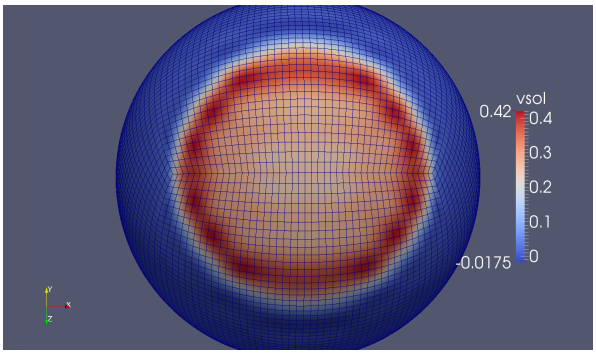
(d) $k = 300$

Figure 12.14: Results for $n = 42$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the ordinary sphere.

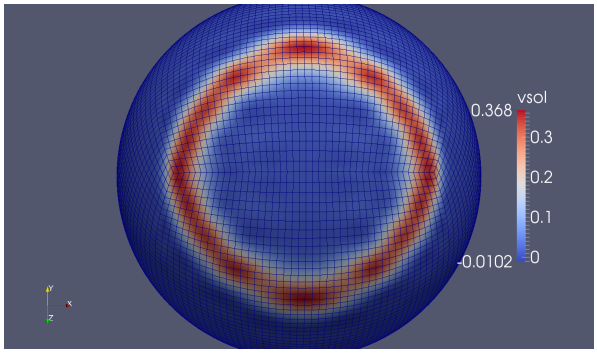
12.2.2. Gaming Sphere



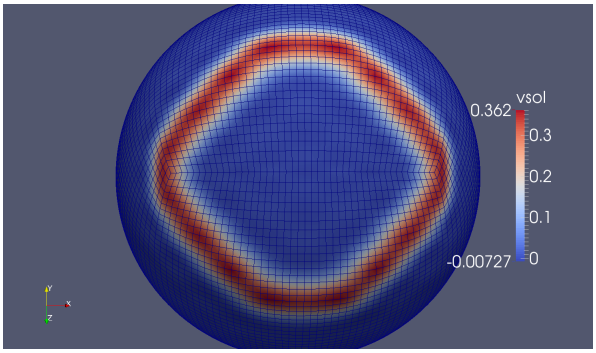
(a) $k = 0$



(b) $k = 100$



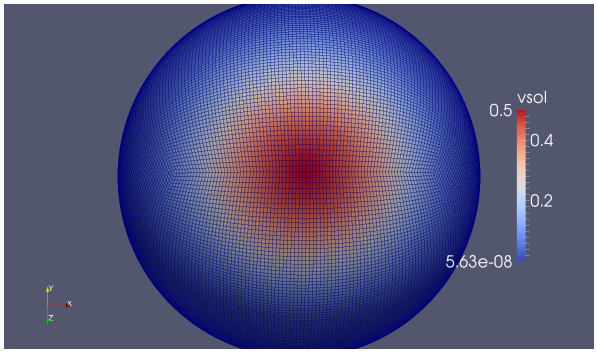
(c) $k = 200$



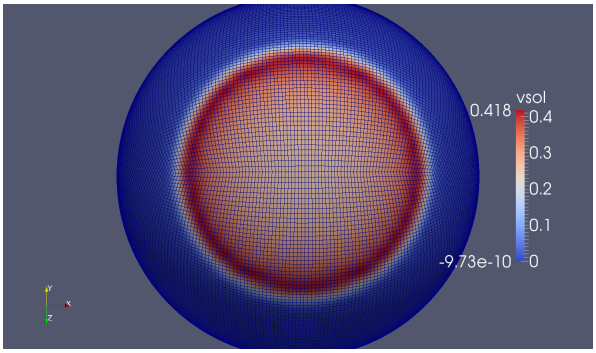
(d) $k = 300$

Figure 12.15: Results for $n = 12$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.

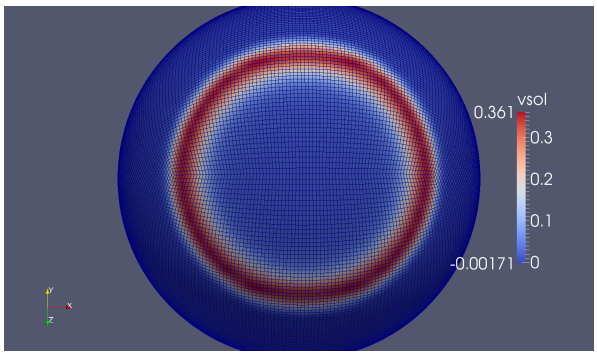
Figure 12.15 shows strong blurring at the wavefront in (b) and (c). Symmetry is broken as well, which is clearly seen in (d). Overall we see slightly less blurring than in figure 12.7 but similar results when it comes to symmetry.
Figure 12.16 shows strong symmetry as well as sharpness in (a), (b) and (c) which is, however, broken in (d). Again, the major axis of the elliptical shape in (d) aligns with the patch boundary. Overall the results are comparable, yet slightly worse than those from figure 12.5. In direct comparison to figure 12.12, the results from figure 12.5 are clearly better.
Figure 12.17 shows strong radial symmetry as well as sharpness in (a) through (c). In (d) symmetry is still decent but some minor fringing appears at the patch boundaries. Overall, the results are significantly better than in figure 12.13 and slightly better than in figure 12.9.
Finally, figure 12.18 shows strong symmetry in (a) through (d), as well as sharpness. The results are comparable to figures 12.6, 12.10 and 12.14.



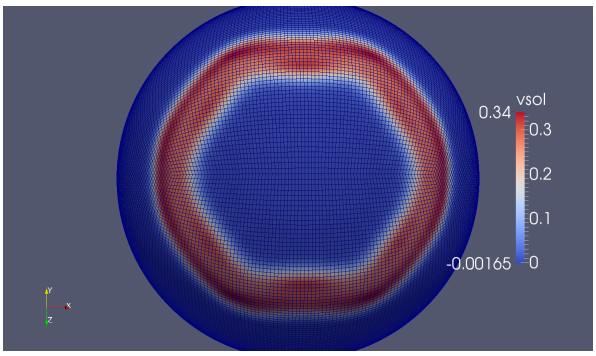
(a) $k = 0$



(b) $k = 100$

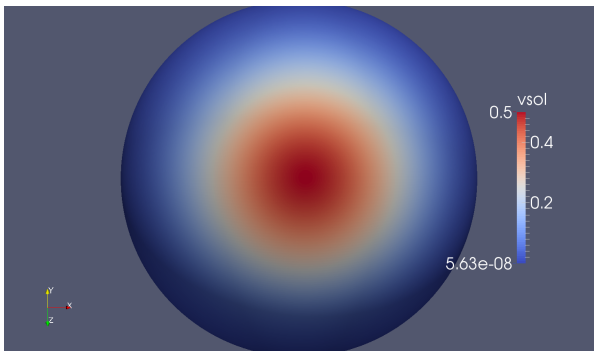


(c) $k = 200$

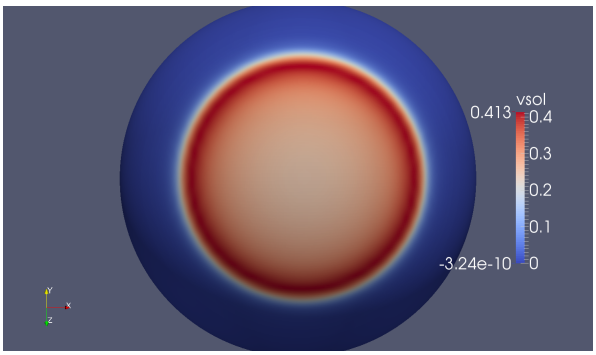


(d) $k = 300$

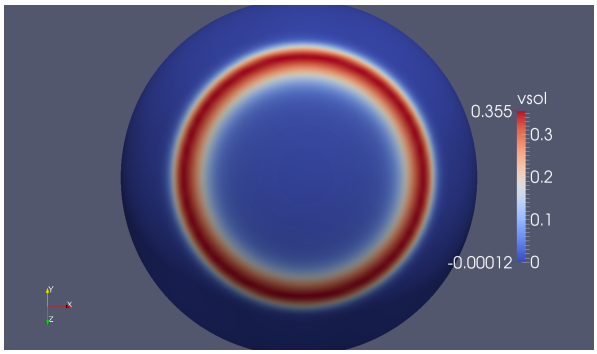
Figure 12.16: Results for $n = 22$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.



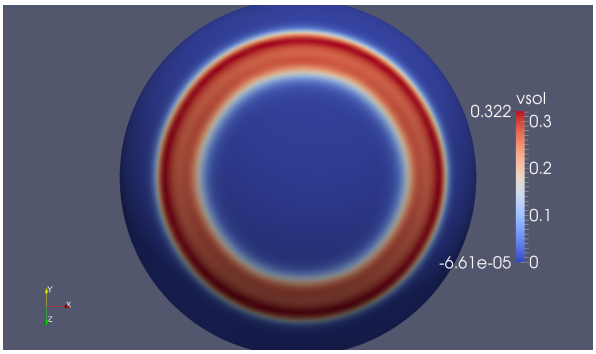
(a) $k = 0$



(b) $k = 100$

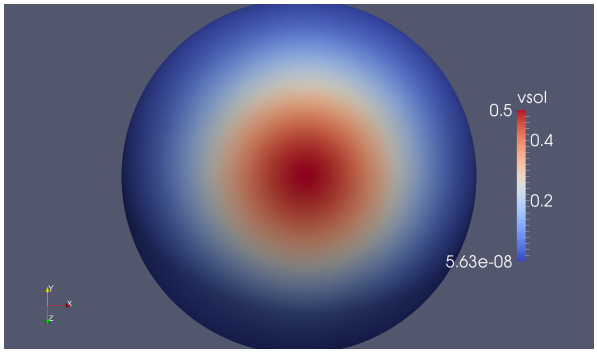


(c) $k = 200$

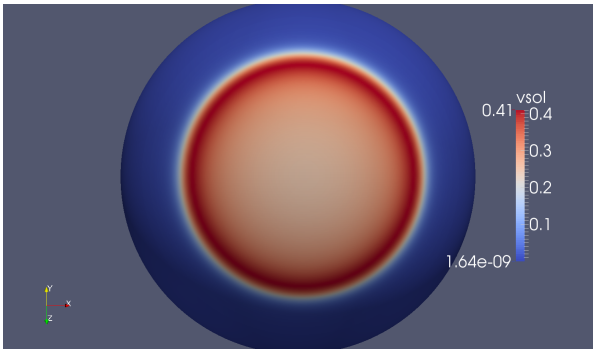


(d) $k = 300$

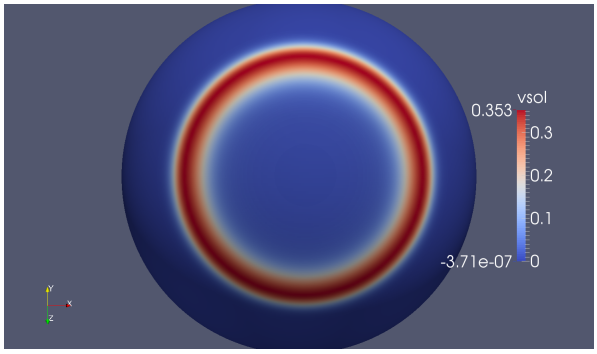
Figure 12.17: Results for $n = 32$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.



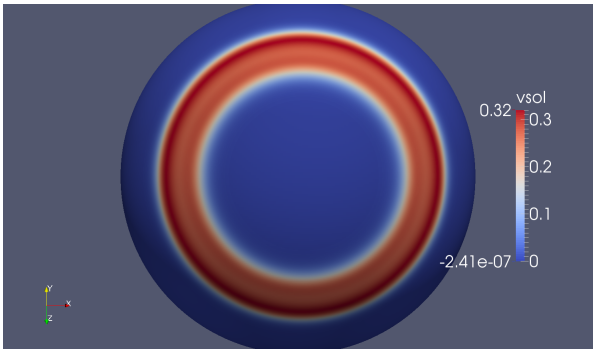
(a) $k = 0$



(b) $k = 100$



(c) $k = 200$



(d) $k = 300$

Figure 12.18: Results for $n = 42$, $p = 3$ after (a) 0, (b) 100, (c) 200 and (d) 300 iterations on the gaming sphere.

12.3. Discussion

In sections 12.1 and 12.2, we presented numerical experiments, testing the overall performance of the numerical scheme introduced in chapter 7 for various basis resolutions on both the ordinary and the gaming sphere. In a direct comparison, the gaming sphere emerges as clear winner over the ordinary sphere. This might be a result of the reduced angle that cells make with one another at the patch boundaries and the reduced fringing effects that result. Another reason might be the fact that the ratio between the largest cells (in the center of patches) and the smallest cells (by the patch boundaries) is significantly larger on the ordinary sphere. This manifests itself in a reduced resolution nearby the center of patches which results in increased blurring of the wavefront which can, for example, be seen by comparing figures 12.13 and 12.17. On the other hand, the increased resolution by the patch boundaries is largely compromised by the fringing effects resulting from the steep angle that cells make with one another there.

Symmetry is usually broken in a way that reveals the location of the patch boundary as for example in figures 12.4, 12.7 and 12.8. The alignment of wavefronts with the patch boundary after symmetry breaking is likely a result of the different (lower) continuity properties of nearby basis functions. Another reason could be the coupling of DOFs by the boundaries that locally decreases the resolution. A possible conclusion is the need to refine by the boundaries before the time-stepping procedure commences.

Overall, it is seen that a resolution of $n = 42$ is necessary to warrant symmetry over a time-frame that corresponds to 300 iterations, as can be seen in figures 12.6, 12.10, 12.14 and 12.18. The only simulation with $n \neq 42$ that performs decently for all k, is the one shown in figure 12.17.

In all of the simulations, it is seen that bases with $p = 3$ slightly outperform their counterparts with $p = 2$. One might be tempted to conclude that higher-order bases should be favored over lower-order

ones, this conclusion, however, might be premature. Since the basis introduced in section (9.3) has exactly $(p+1)^2$ nonzero basis functions on each element, the bandwidth of the system matrix will grow with p. This will manifest itself mainly through increased assembly costs since more nonzero entries have to be computed. Another drawback of higher-order bases is their increased support, which makes the refinement strategies from chapter 10 less efficient, due to the reduced locality. It is thus important to make trade-offs between computational costs and performance. Quantitative numerical experiments are needed to determine how the computational costs scale with increasing p and how this compares to the increased costs resulting from the increase in n needed to achieve quantitatively similar results for lower-order bases.
Finally it should be mentioned that in almost all figures from sections 12.1 and 12.2, we can see unphysical negative concentrations. These negative concentrations are most likely a result of wiggle formation in the vicinity of steep gradients. A possible remedy is the inclusion of a flux-limiter in the spatial discretization of the numerical scheme of chapter 7. Note, however, that this would constitute a *'cosmetic'* intervention rather than a substantial mathematical improvement of the scheme (in fact, the $L_2(\mathscr{M}_k)$-error is expected to increase). For this reason, we have chosen to ignore unphysical oscillations and it is seen that wiggle-formation is negligible for bases with $n = 32$ or higher (greatly facilitated by the smoothness of the basis functions).

12.4. Existing Shortcomings and Possible Remedies

As a result of the necessity to reassemble the system matrix after each iteration, it is of utmost importance to find proficient refinement criteria. In chapter 10, we have introduced two strategies that are based on the local properties of the geometry. Possibly the biggest shortcoming of the current implementation is the absense of refinement based on the behavior of the iterands u^k and v^k on $\mathscr{M}_k$. The benchmark problem from sections 12.1 and 12.2 is characterized by three stages:

1. rapid radially outward diffusion and formation of a wavefront with high concentration and inner layer with moderate concentration (transition from (a) to (b) in the figures),
2. depletion of the inner layer and formation of a narrow ring with steep gradients (transition from (b) to (c)),
3. sluggish radially outward as well as inward diffusion (transition from (c) to (d)).

It is seen that even moderate resolutions (those with $n = 22$) perform decently in the first two of these stages since symmetry tends to be still present in (c) but absent in (d). A possible reason for this behavior is the formation of the narrow ring in (c), which might be too narrow or contain too steep gradients to be accurately represented by an element from the linear span of a low-resolution basis. This suggests the introduction of an adaptive shockwave-refinement in conjunction with coarsening, based on average function value and / or gradient that continuously refines and coarsens based on the position of the wavefront.
The results from an experimental implementation based on these principles is shown in figure 12.19. The solver operates as follows: after the k^{th} iteration, the set of function supports of $\mathscr{B}_1 = \{w_1, \dots, w_N\}$ that violate either of the criteria

$$\frac{\int\limits_{\operatorname{supp} w_i} v^k \sqrt{g_k}\,\mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} \sqrt{g_k}\,\mathrm{d}\boldsymbol{\xi}} < 0.95\mu_{\text{val}} \tag{12.3}$$

or

$$\frac{\int\limits_{\operatorname{supp} w_i} \left\langle \nabla_k v^k, \nabla_k v^k \right\rangle_{g_k} \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}} < 2.5 \mu_{\mathrm{grad}}, \tag{12.4}$$

is determined and the set ϵ of all elements contained in this subset of Ω is stored in memory. Here (12.3) considers the average function value of V^k and (12.4) the average norm of the gradient of V^k as a criterion for refinement.
Analogous to chapter 10, the values of μ_{val} and μ_{grad} given by

$$\mu_{\mathrm{val}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\int\limits_{\operatorname{supp} w_i} v \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}$$

$$\mu_{\mathrm{grad}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\int\limits_{\operatorname{supp} w_i} \langle \nabla_0 v, \nabla_0 v \rangle_{g_0} \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}{\int\limits_{\operatorname{supp} w_i} \sqrt{g_0} \mathrm{d}\boldsymbol{\xi}}, \tag{12.5}$$

where v is the local counterpart of the V-component of the *exact* initial condition from (12.1) (note that the exact initial geometry has already been projected onto Σ). Before each iteration the set of functions that violate (12.3) or (12.4) is determined and the set elements ϵ of the corresponding basis functions is stored in memory.
At the beginning of each iteration, Σ is reset to default: $\Sigma \leftarrow \mathscr{B}_1$ and the integrals from (12.3) and (12.4) are computed utilizing the current set of elements. As a next step the set of elements is set to default as well $\mathscr{A} \leftarrow \mathscr{T}_1$ and ϵ is passed to algorithm (2) in order to acquire a basis refined on the segments of Ω that violate (12.3) or (12.4). Given Σ^k as the basis during the k^{th} iteration that results from this procedure, algorithm (2) ensures that each component of the solution vector at t_k is projected onto the newly formed basis Σ^{k+1}.

Remark. *Note that this projection might not be possible without projection errors since $\Sigma^k \not\subset \Sigma^{k+1}$ whenever the set of to be refined elements during the k^{th} iteration is unequal to that from the $k+1^{th}$ iteration.*
There are more subtleties involved here when it comes to projection and matrix assembly. The mass matrix as well as the right hand side vector $\mathbf{F}$ can not be built by looping over the set of elements $\mathscr{A}^{k+1}$ during the $k+1^{th}$ iteration. This is a result of the fact that for the assembly of both functions containing $\sqrt{g_k}$ have to be integrated. $\sqrt{g_k}$ might contain functions from $\mathscr{B}_2$ that are supported by elements from $\mathscr{T}_1$ in $\mathscr{A}^{k+1}$ (leading to quadrature errors, see chapter 5). Each component of the right-hand side vector $\mathbf{F}$ of the projection is of the form

$$F_i = \int_\Omega w_i f \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}, \tag{12.6}$$

and the mass matrix

$$[A]_{i,j} = \int_\Omega w_i w_j \sqrt{g_k} \mathrm{d}\boldsymbol{\xi}, \tag{12.7}$$

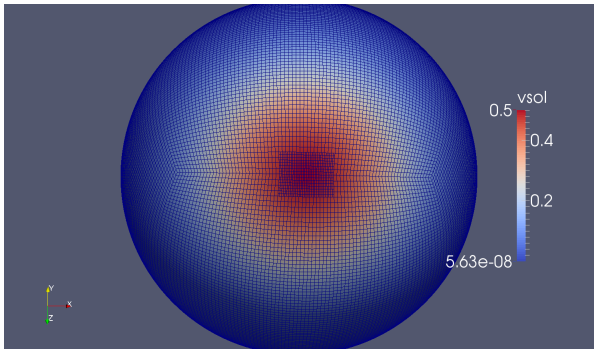
thus, the assembly should be carried out over a grid $\mathscr{C}^{k+1}$ that contains all elements from $\mathscr{T}_2$ in

$\mathscr{A}^{k+1}$ and $\mathscr{A}^k$ (or simply over $\mathscr{T}_2$ itself, which is, however, more expensive).

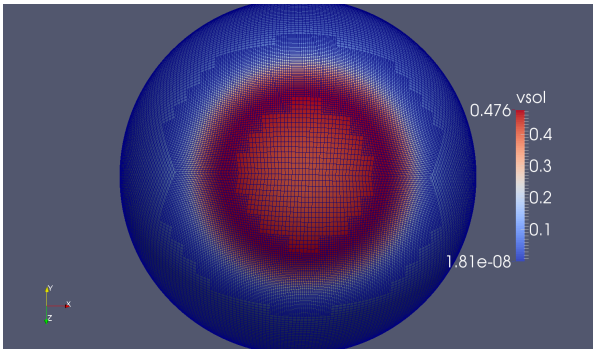
After the projection is completed, a time-step is carried out with basis Σ^{k+1}.

Remark. *Since [B] (see section 7.3) contains $\sqrt{g_k}$, as well as $\sqrt{g_{k-1}}$, its assembly should be carried out by looping over $\mathscr{T}_2$.*
We use the exact expression of v in (12.5) to compute the reference values in order to ensure that the first iterant v^0 is already an element from the span of a basis compliant with (12.4) and (12.5). In chapter 10, we were not confronted with the problem of incompatible grids since $\operatorname{span}\Sigma^k \subset \operatorname{span}\Sigma^{k+1}$ *for all k. Not allowing for coarsening when refining based on the properties of the geometry is reasonable since manifestations of surface deformation through the v^k are largely irreversible.*

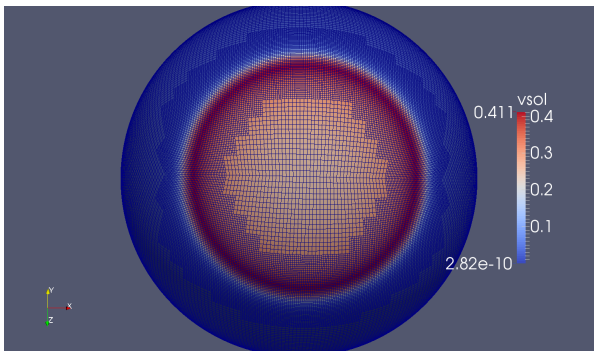
Figure 12.19 shows very strong radial symmetry as well as sharpness at the wavefronts. Overall, the results are comparable to those that result from a high resolution scheme as in figure 12.10, while the computational costs are only slightly larger than in the simulation from 12.8.
Combining the refinement technique presented in this chapter with the ones from chapter 10 is a nontrivial task since the techniques presented there do not allow for coarsening. Furthermore, coarsening can have a negative effect on the quality of the geometry since shockwave refinement does not take into account the geometrical properties like curvature and cell size. A possible remedy is to only coarsen when the corresponding projection error is sufficiently small.
Here we have chosen an initial basis $\mathscr{B}_1$ whose basis functions are supported by large segments of Ω (and $\mathscr{M}_k$) with respect to the characteristic dimensions of the wavefront. When finer bases are utilized it is recommended to not only refine the function supports that violate(12.4) or (12.5) but also the supports of the n nearest neighbours (where the value of n depends on the resolution of the basis).



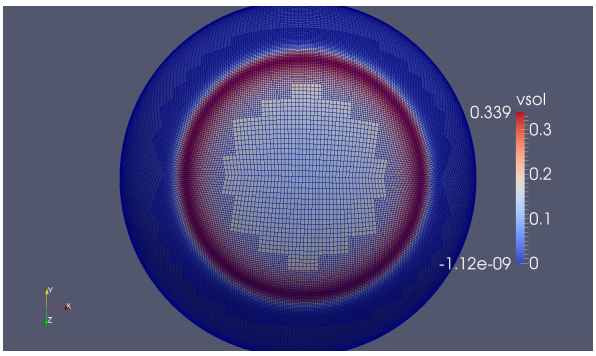
(a) $k = 0$



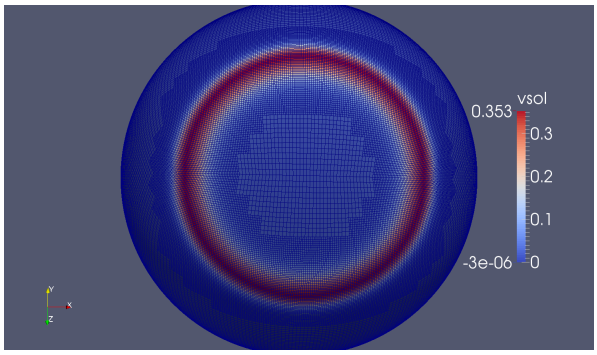
(b) $k = 50$



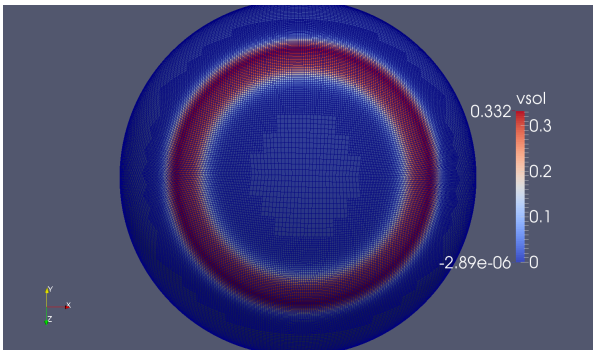
(c) $k = 100$



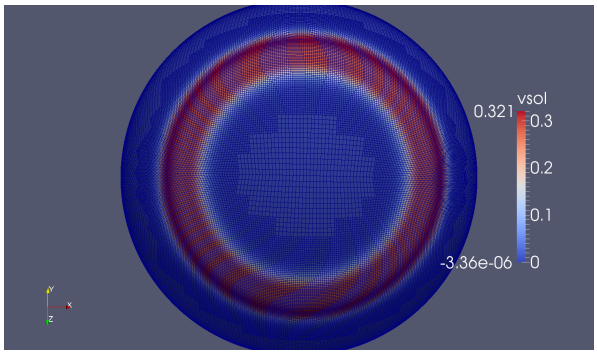
(d) $k = 150$



(e) $k = 200$



(f) $k = 250$



(g) $k = 300$

Figure 12.19: Results with shockwave-refinement for $n = 22$, $p = 2$ after (a) 0, (b) 50, (c) 100, (d) 150, (e) 200, (f) 250 and (g) 300 iterations on the gaming sphere. The grid has been turned on in order to see where the solver refines. The refinement regions are characterized by the increased density of grid points.

13

Space-Time Galerkin

In chapter 7, we introduced a general discretized scheme to tackle (7.1) utilizing a finite-difference discretization in time. In this chapter, we will conceptually present a possible way to discretize based on the principles from FEA in both the spatial as well as the temporal component. None of the following has been practically realized and we only present a starting point that may serve as inspiration for the interested reader.
The advantage of a FEM-discretization in the temporal component is the possibility to locally refine and coarsen in time. This might save computational ressources on segments of Ω where the chemical species u and v are relatively static.
A disadvantage, however, is the fact that a complete FEM-discretization will lead to a non-linear system which demands an iterative approach utilizing Picard- or Newton-iteration. Since in the scheme from chapter 7 the matrices have to be rebuilt after each iteration, the difference in computational costs may be less severe and the possibility to locally refine in time (and space) may save computational ressources. This chapter provides possible inspiration for improvements of the current implementation.
We start off by presenting the discretization after which we will introduce a Newton-scheme to tackle it. We end this chapter on possible temporal refinement strategies.

13.1. Discretization

Introducing the operator

$$L\begin{pmatrix} u \\ v \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} d_1\Delta_t u + F(1-u) - u\partial_t \ln\sqrt{g_t} - uv^2 - \partial_t u \\ d_2\Delta_t v - (F+k)v - v\partial_t \ln\sqrt{g_t} + uv^2 - \partial_t v \\ K v\mathbf{n}(\mathbf{s}) - \partial_t \mathbf{s} \end{pmatrix}, \tag{13.1}$$

we can write (7.1) in the form

$$L\left(\mathbf{u}^*\right) = 0, \quad \forall(\boldsymbol{\xi}, t) \in \Omega \times [t^k, t^{k+1}] \tag{13.2}$$

where $\mathbf{u}^*$ is the exact solution of (7.1) over the spatial domain Ω and temporal interval $[t^k, t^{k+1}]$ with some initial condition

$$\mathbf{u}^*\big|_{t=t^k} = \mathbf{f}^k. \tag{13.3}$$

The $\mathbf{s}$-component of $\mathbf{u}^*$ then parameterizes $\mathscr{M}_t$ over the temporal interval $[t^k, t^{k+1}]$.
It is convenient to have a domain of the form $\Omega \times [0,1]$ with Ω as in the previous chapters and $\Omega \times [0,1]$

representing its extension in the temporal direction. This is accomplished by replacing $[t^k, t^{k+1}] \to [0,1]$, $\mathrm{d}t \to h_k \mathrm{d}\tau$ and $\partial_t \to \partial_\tau = \frac{1}{h_k}\hat{\partial}_\tau$, where as before $h_k = t^{k+1} - t^k$ and

$$\hat{\partial}_\tau = \frac{\partial}{\partial \tau}. \tag{13.4}$$

Henceforth $\mathbf{u}$ will refer to the (approximate) solution over $\Omega \times [0,1]$.
As a first step, we determine the weak form of (13.2) by multiplying by a test function $w(\boldsymbol{\xi}, t)$ and integrating over $\Omega \times [0,1]$ with measure $\sqrt{g_t} h_k$

$$\int_0^1 \int_\Omega w L(\mathbf{u}) \sqrt{g_\tau} h_k \mathrm{d}\boldsymbol{\xi} \mathrm{d}\tau = 0, \tag{13.5}$$

thus

$$\int_0^1 \int_\Omega \begin{pmatrix} d_1 \langle \nabla_\tau w, \nabla_\tau u \rangle_{g_\tau} + wF(1-u) - wu\partial_\tau \ln \sqrt{g_\tau} - wuv^2 - w\partial_\tau u \\ d_2 \langle \nabla_\tau w, \nabla_\tau v \rangle_{g_\tau} - w(F+k)v - wv\partial_\tau \ln \sqrt{g_\tau} + wuv^2 - w\partial_\tau v \\ wK v\mathbf{n} - w\partial_\tau \mathbf{s} \end{pmatrix} \sqrt{g_t}\, h_k \mathrm{d}\boldsymbol{\xi} \mathrm{d}\tau = 0, \tag{13.6}$$

where ∇_τ refers to the Laplace-Beltrami operator of $\mathscr{M}_t$ with $t = h_k \tau$.
As a next step we construct a basis $\Sigma = \{w_1, \ldots, w_N\}$ that is compliant with (7.30). Restricting our attention to a spherically shaped initial geometry, we can simply use the basis from 9.3. As a next step we extend it in the temporal direction in a tensor-product way. Hence, let

$$\Xi^\tau_{m,p} = \left\{ \underbrace{0, \ldots, 0}_{p+1 \text{ times}}, \underbrace{\frac{1}{m-p}, \ldots, \frac{m-p-1}{m-p}}_{m-p-1 \text{ terms}}, \underbrace{1, \ldots, 1}_{p+1 \text{ times}} \right\}, \tag{13.7}$$

be the clamped knot vector utilized to construct

$$\sigma^t_{n,p} = \left\{ N_{1,p}(\tau), \ldots, N_{m,p}(\tau) \right\}. \tag{13.8}$$

Given a collection $\{w_1, \ldots, w_N\}$ of bivariate basis functions from section 9.3, the trivariate basis can be constructed via

$$w_{i,j}(\xi, \eta, \tau) = w_i(\xi, \eta) N_{j,p}(\tau). \tag{13.9}$$

The tensor-product index i, j is then replaced by a single global index such that we acquire the trivariate basis $\bar{\Sigma} = \{w_1, \ldots, w_M\}$.
An additional constraint we have to impose on the w_i is $w_i(\xi, \eta, 0) = 0$. This is easiest achieved by disregarding $N_{1,p}(\tau)$ in the tensor product from (13.9) . Thus, let us introduce the bases

$$\Sigma = \left\{ w_i \in \bar{\Sigma} \mid w_i|_{\tau=0} = 0 \right\} \tag{13.10}$$

and

$$\Sigma^1 = \left\{ w_i \in \bar{\Sigma} \mid w_i|_{\tau=0} \neq 0 \right\}. \tag{13.11}$$

With the tensor-product structure of the basis functions in mind, we see that Σ has cardinality $N(m-1)$ and Σ^1 has cardinality N. In order to be conform with the previous chapters, we replace $N(m-1) \to$

N and $N \to N/(m-1)$ for the global function indices.
We can approximate u, v and $\mathbf{s}$ as follows

$$\begin{aligned} u &= \sum_j c_j^u w_j + u_0 \\ v &= \sum_j c_j^v w_j + v_0 \\ \mathbf{s} &= \sum_j \mathbf{c}_j^{\mathbf{s}} w_j + \mathbf{s}_0, \end{aligned} \tag{13.12}$$

where

$$\begin{aligned} u_0|_{\tau=0} &= f_u^k \\ v_0|_{\tau=0} &= f_v^k \\ \mathbf{s}_0|_{\tau=0} &= \mathbf{f}_{\mathbf{s}}^k. \end{aligned} \tag{13.13}$$

Here f_u^k, f_v^k and $\mathbf{f}_{\mathbf{s}}^k$ refer to the individual components of the temporal boundary condition $\mathbf{f}^k$.
The $\mathbf{c}_j^{\mathbf{s}}$ are vectors in $\mathbb{R}^3$ and their components are

$$\mathbf{c}_j^{\mathbf{s}} = \begin{pmatrix} c_j^1 \\ c_j^2 \\ c_j^3 \end{pmatrix}. \tag{13.14}$$

As a next step, we assign some global ordering to the $c_j^u, c_j^v, c_j^1, c_j^2, c_j^3$ and define the vector $\mathbf{c}$ containing these weights in the corresponding order. One possible choice is

$$\mathbf{c} = \left(c_1^u, c_1^v, c_1^1, c_1^2, c_1^3, \ldots, c_N^u, c_N^v, c_N^1, c_N^2, c_N^3\right)^T. \tag{13.15}$$

Note that with the definition from (13.12): $u = u(\mathbf{c}), v = v(\mathbf{c}), \mathbf{s} = \mathbf{s}(\mathbf{c})$.
In (13.2), we successively replace w by each of the $w_i \in \Sigma$. By that, we receive a system of $5N$ equations for $5N$ unknowns. Let us define

$$\begin{aligned} F_u^i(\mathbf{c}) &\equiv \int_0^1 \int_\Omega \left[d_1 \langle \nabla_\tau w_i, \nabla_\tau u \rangle_{g_\tau} + w_i \left(F(1-u) - u\partial_\tau \ln \sqrt{g_\tau} - uv^2 - \partial_\tau u \right) \right] \sqrt{g_\tau} h_k \mathrm{d}\boldsymbol{\xi} \mathrm{d}\tau \\ F_v^i(\mathbf{c}) &\equiv \int_0^1 \int_\Omega \left[d_2 \langle \nabla_\tau w_i, \nabla_\tau v \rangle_{g_\tau} - w_i \left((F+k)v - v\partial_\tau \ln \sqrt{g_\tau} + uv^2 - \partial_\tau v \right) \right] \sqrt{g_\tau} h_k \mathrm{d}\boldsymbol{\xi} \mathrm{d}\tau \\ F_{\mathbf{s}_l}^i(\mathbf{c}) &\equiv \int_0^1 \int_\Omega w_i \left[Kvn_l - (\partial_\tau \mathbf{s})_l \right] \sqrt{g_\tau} h_k \mathrm{d}\boldsymbol{\xi} \mathrm{d}\tau. \end{aligned} \tag{13.16}$$

Note that $\sqrt{g_\tau}$ is a function of $\mathbf{c}$, as well.
As a next step, we define the vector-valued function $\mathbf{F}(\mathbf{c})$ by choosing some global ordering of the $F_u^i(\mathbf{c})$, $F_v^i(\mathbf{c})$, $F_{\mathbf{s}_l}^i(\mathbf{c})$, for example the ordering corresponding to $\mathbf{c}$

$$\mathbf{F}(\mathbf{c}) = \left(F_u^1, F_v^1, F_{\mathbf{s}_1}^1, F_{\mathbf{s}_2}^1, F_{\mathbf{s}_3}^1, \ldots, F_u^N, F_v^N, F_{\mathbf{s}_1}^N, F_{\mathbf{s}_2}^N, F_{\mathbf{s}_3}^N\right)^T, \tag{13.17}$$

We have to solve the system $\mathbf{F}(\mathbf{c}) = 0$, this however cannot be done directly since $\mathbf{F}(\mathbf{c})$ is non-linear. We choose for an iterative scheme based on the *Newton-Raphson-method* [6]. To this end, we define the Jacobian matrix of $\mathbf{F}$ with respect to $\mathbf{c}$ by

$$\frac{\partial \mathbf{F}}{\partial \mathbf{c}} = \begin{bmatrix} \frac{\partial F_1}{\partial c_1} & \frac{\partial F_1}{\partial c_2} & \cdots & \frac{\partial F_1}{\partial c_{5N}} \\ \frac{\partial F_2}{\partial c_1} & \frac{\partial F_2}{\partial c_2} & \cdots & \frac{\partial F_2}{\partial c_{5N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_{5N}}{\partial c_1} & \frac{\partial F_{5N}}{\partial c_2} & \cdots & \frac{\partial F_{5N}}{\partial c_{5N}} \end{bmatrix}, \tag{13.18}$$

the iteration is carried out as follows: we start with an initial guess $\mathbf{c}^0$ and update the l-th iterand $\mathbf{c}^l$ after each iteration via $\mathbf{c}^{l+1} = \mathbf{c}^l + \Delta\mathbf{c}$, where

$$\frac{\partial \mathbf{F}}{\partial \mathbf{c}}(\mathbf{c}^l)\Delta\mathbf{c} = -\mathbf{F}(\mathbf{c}^l). \tag{13.19}$$

The iteration is terminated once some termination criterion has been reached.
For a collection of derivatives that can be utilized to construct (13.19), see appendix A.5.

Remark. *One might choose to replace the scheme from (13.19) by quasi-Newton approaches nearly quadratic convergence without the need to assemble the Jacobian matrix analytically. Schemes based on the secant method may be more suitable, further information can be found in [8] and [22].*

13.2. Time-Slabbing

The scheme from (13.19) together with the initial condition from (13.13) forms the basis of the algorithm. Assuming that the initial temporal interval corresponds to $[0, t^1]$ with some initial time-step t^1, the functions $u_0, v_0, \mathbf{s}_0$ can be constructed, for example, by a tensor-product of the exact initial condition(s) $\mathbf{i}_u, \mathbf{i}_v, \mathbf{i}_\mathbf{s}$ with the temporal function $N_{1,p}(t)$. In an IgA-setting, however, it is costumary to project the initial condition onto the function space before the iteration commences. Thus, let

$$\Sigma_0^1 = \left\{ w_i|_{\tau=0} \;\middle|\; w_i \in \Sigma^1 \right\}, \tag{13.20}$$

the initial condition can (approximately) be incorporated by selecting $u_0, v_0, \mathbf{s}_0$ as the L_2-projections of $\mathbf{i}_u, \mathbf{i}_v, \mathbf{i}_\mathbf{s}$ onto Σ_0^1 after which the Newton-scheme commences.
Let $\mathbf{u}^{k+1}$ denote the solution resulting form the iterative scheme (13.19) over the temporal interval $[t^k, t^{k+1}]$. The basic idea is to utilize $\mathbf{u}^k\big|_{\tau=1}$ as initial condition of the scheme over the temporal interval $[t^k, t^{k+1}]$. Assuming that iterations $k-1$ and k utilize the same bivariate spatial basis in their tensor-product structure, this is accomplished by selecting the weights of $\mathbf{u}_i^k, i \in [u, v, \mathbf{s}]$ that correspond to the functions of the form

$$w_i = f_i(\xi, \eta) N_{m,p}(\tau) \tag{13.21}$$

during the k^{th} iteration and assigning them to the functions

$$\Sigma^1 \ni w_i = f_i(\xi, \eta) N_{1,p}(\tau), \tag{13.22}$$

during the $k+1^{\text{th}}$ iteration to construct u_0, v_0, and $\mathbf{s}_0$.
A reasonable choice for the initial guess of (13.19) is the vector $\mathbf{c}^0$ that corresponds to the function

$$\boldsymbol{\beta}^k \quad \text{with} \quad \boldsymbol{\beta}^k(\xi, \eta, \tau) = \begin{bmatrix} u_0(\xi, \eta, 0) \\ v_0(\xi, \eta, 0) \\ \mathbf{s}_0(\xi, \eta, 0) \end{bmatrix}, \tag{13.23}$$

i.e. the function that equals the initial condition over the entire temporal interval $[0, 1]$. Again, this is easily accomplished by assigning the weight of

$$\Sigma^1 \ni w_i = f_i(\xi, \eta) N_{1,p}(\tau) \tag{13.24}$$

in each of the u_0, v_0, $\mathbf{s}_0$ to the entries in $\mathbf{c}^0$ that correspond to the weights of functions

$$\Sigma \ni w_{i+(j-1)\frac{N}{m}} = f_i(\xi, \eta) N_{j,p}(\tau), \quad j \in \{2, \dots, m\} \tag{13.25}$$

of the individual components of $\boldsymbol{\beta}^k$, $\boldsymbol{\beta}_u^k, \boldsymbol{\beta}_v^k$ and $\boldsymbol{\beta}_{\mathbf{s}}^k$. If the Newton-scheme should fail to converge, a possible remedy is to decrease the value of t^{k+1} and repeat above steps. More sophisticated ways of choosing the initial guess can be found in [31]. Above scheme forms the basic algorithm.
As mentioned in the beginning of this chapter, discretizing in space and time utilizing an IgA approach yields the advantage of the possibility to refine locally in all parametric coordinates. Since the soltion of (7.1) exhibits many different characteristics, like shockwaves (see section 12.4) and geometric deformations but also relative idleness (after pattern formation has completed), approaching refinement with more sophisticated measures may be a reasonable alternative to the method presented in this thesis. We shall not present concrete refinement strategies but refer to the publications by Meitner et al. [27] and Rannacher et al. [5], for possible inspiration.

A

Appendix

A.1. Appendix Calculus on Geometric Objects

Lemma A.1.1. *Given some sufficiently smooth* $\mathbf{U} : \mathcal{M} \to T_P\mathcal{M}$*, the operator* $\nabla\cdot$ *that satisfies*

$$\int_{\mathcal{M}} W \nabla \cdot \mathbf{U} \mathrm{d}\mathbf{x} = -\int_{\mathcal{M}} \nabla W \cdot \mathbf{U} \mathrm{d}\mathbf{x}, \tag{A.1}$$

for all functions W *that vanish on* $\partial\mathcal{M}$*, is, in local coordinates, given by*

$$\nabla \cdot \mathbf{u} = \frac{1}{\sqrt{g}} \sum_{i=1}^{n} \frac{\partial}{\partial \xi_i} \left(\sqrt{g} u_i\right). \tag{A.2}$$

Proof. From (A.1), we can deduce an expression in local coordinates.

$$\begin{aligned} \int_\Omega w \nabla \cdot \mathbf{u} \sqrt{g} \mathrm{d}\boldsymbol{\xi} &= -\int_\Omega \langle \nabla w, \mathbf{u} \rangle_g \sqrt{g} \mathrm{d}\boldsymbol{\xi} \\ &= -\int_\Omega \sum_{i=1}^{n} \frac{\partial w}{\partial \xi_i} u_i \sqrt{g} \mathrm{d}\boldsymbol{\xi} \\ &\overset{w|_{\partial\Omega}=0}{=} \int_\Omega w \sum_{i=1}^{n} \frac{\partial}{\partial \xi_i} \left(\sqrt{g} u_i\right) \mathrm{d}\boldsymbol{\xi}. \end{aligned} \tag{A.3}$$

Comparing the left and right hand sides, we have

$$\nabla \cdot \mathbf{u} = \frac{1}{\sqrt{g}} \sum_{i=1}^{n} \frac{\partial}{\partial \xi_i} \left(\sqrt{g} u_i\right). \tag{A.4}$$

□

Lemma A.1.2. *On geometries without boundary, we have*

$$\int_{\mathcal{M}} W \Delta U \mathrm{d}\mathbf{x} = -\int_{\mathcal{M}} \nabla W \cdot \nabla U \mathrm{d}\mathbf{x}, \tag{A.5}$$

or in local coordinates

$$\int_\Omega w\Delta u\sqrt{g}\mathrm{d}\boldsymbol{\xi} = -\int_\Omega \langle\nabla w,\nabla u\rangle_g\sqrt{g}\mathrm{d}\boldsymbol{\xi}. \tag{A.6}$$

Proof. Let $W:\mathscr{M}\to\mathbb{R}$ and $\mathbf{U}:\mathscr{M}\to T_P\mathscr{M}$, utilizing equation (2.21), in local coordinates, we can deduce

$$\begin{aligned}\nabla\cdot(w\mathbf{u}) &= \frac{1}{\sqrt{g}}\sum_{i=1}^{n}\frac{\partial}{\partial\xi_i}\left(\sqrt{g}wu_i\right)\\ &= \frac{1}{\sqrt{g}}\sum_{i=1}^{n}\left[w\frac{\partial}{\partial\xi_i}\left(\sqrt{g}u_i\right)+\frac{\partial w}{\partial\xi_i}\sqrt{g}u_i\right]\\ &= \frac{1}{\sqrt{g}}\sum_{i=1}^{n}w\frac{\partial}{\partial\xi_i}\left(\sqrt{g}u_i\right)+\sum_{i=1}^{n}\frac{\partial w}{\partial\xi_i}u_i\\ &= w\nabla\cdot\mathbf{u}+\langle\nabla w,\mathbf{u}\rangle_g.\end{aligned} \tag{A.7}$$

Translating above identity to global coordinates and rearranging yields

$$W\nabla\cdot\mathbf{U} = \nabla\cdot(W\mathbf{U})-\nabla W\cdot\mathbf{U}. \tag{A.8}$$

Replacing $\mathbf{U}\to\nabla U$, integrating over $\mathscr{M}$ and utilizing lemma 1, we find

$$\int_{\mathscr{M}} W\Delta U\mathrm{d}\mathbf{x} = \int_{\partial\mathscr{M}} W\nabla U\cdot\mathbf{N}_{\partial\mathscr{M}}\mathrm{d}l-\int_{\mathscr{M}}\nabla W\cdot\nabla U\mathrm{d}\mathbf{x}. \tag{A.9}$$

For geometries without boundary, the integral over $\partial\mathscr{M}$ vanishes and the lemma follows. □

Proposition A.1.1. *Let $[A]$ be symmetric and let $[B]$ be symmetric positive definite (SPD). Then $[B][A]$ is diagonalizable.*

Proof. As $[B]$ is SPD, it possesses an SPD (and thus invertible) matrix root $[B]^{1/2}$ with $[B]=[B]^{1/2}[B]^{1/2}$. Now we have

$$[B]^{-1/2}[B][A][B]^{1/2} = [B]^{1/2}[A][B]^{1/2}, \tag{A.10}$$

where the matrix on the right-hand side is symmetric.
Hence, $[B][A]$ is similar to a symmetric matrix and thus diagonalizable. □

A.2. Appendix Finite-Element Analysis

Lemma A.2.1. *Let $(\cdot,\cdot)_L$ be an inner product, i.e. let L satisfy*

$$\begin{aligned}(U,V)_L &= (V,U)_L\\ (\alpha U,V)_L &= \alpha(U,V)_L\\ (U+V,W)_L &= (U,W)_L+(V,W)_L\\ (U,U)_L &\geq 0 \text{ and } (U,U)_L = 0 \implies U=0.\end{aligned} \tag{A.11}$$

Furthermore, let $\Sigma = \{W_1, \dots, W_N\}$ be a (linearly independent) basis over $\mathscr{M}$. Then the matrix

$$[M] = \begin{bmatrix} (W_1, W_1)_L & \dots & (W_1, W_N)_L \\ \vdots & & \vdots \\ (W_N, W_1)_L & \dots & (W_N, W_N)_L \end{bmatrix} \tag{A.12}$$

is SPD (and thus non-singular).

Proof. From the linear independence of Σ, we have $0 = \sum_j c_j W_j$ if and only if $c_j = 0, \forall j \in \{1, \dots, N\}$. Now take any $0 \neq \sum_j c_j W_j \in \operatorname{span}\Sigma$. We have $(U, U)_L > 0, \forall U \neq 0$, so, in particular

$$\begin{aligned} \left(\sum_j c_i W_j, \sum_j c_j W_j \right)_L &= \sum_i \sum_j c_i c_j (W_i, W_j)_L \\ &= \mathbf{c}^T [M] \mathbf{c} > 0. \end{aligned} \tag{A.13}$$

Since the choice of the c_j was arbitrary, it follows that $[M]$ is SPD. □

A.3. Appendix Isogeometric Analysis

Proposition A.3.1. *Let the B-spline basis $\bar{\Sigma}$ satisfy $1 \in \operatorname{span}\Sigma$, then the L_2-projection is* mass conserving.

Proof. Suppose the function $F : \mathscr{M} \to \mathbb{R}$ is projected onto the basis Σ via an L_2-projection. We call the projection *mass conserving* whenever the projection U satisfies

$$\int_{\mathscr{M}} U \mathrm{d}\mathbf{x} = \int_{\mathscr{M}} F \mathrm{d}\mathbf{x}. \tag{A.14}$$

Now let $c_1^1, \dots c_n^1$ be the weights that satisfy $\sum_i c_i^1 w_i = 1$.
In an L_2-projection, we craft U in such a way that it satisfies

$$\forall i \in \{1, \dots, m\} : \quad \int_{\mathscr{M}} W_i \, (U - F) \, \mathrm{d}\mathbf{x} = 0. \tag{A.15}$$

Performing a weighed sum over all i, we find

$$\begin{aligned} & \int_{\mathscr{M}} \sum_i c_i^1 W_i \, (U - F) \, \mathrm{d}\mathbf{x} = 0 \\ \implies & \int_{\mathscr{M}} (U - F) \, \mathrm{d}\mathbf{x} = 0 \\ \implies & \int_{\mathscr{M}} U \mathrm{d}\mathbf{x} = \int_{\mathscr{M}} F \mathrm{d}\mathbf{x}. \end{aligned} \tag{A.16}$$

□

A.4. Appendix Isogeometric Implementation

We start this section with the following lemma

Lemma A.4.1. *Given a differentiable function $f(t)$ on $[a,b]$, let $h = b-a$ and*

$$I_h = f(a)h. \tag{A.17}$$

Then

$$\left| \int_a^b f(t)\mathrm{d}t - I_h \right| \leq \frac{1}{2} M h^2, \tag{A.18}$$

where

$$M = \max_{t\in[a,b]} f'(t). \tag{A.19}$$

Proof. See [41, p. 49]. □

Consider the differential equation in integral form

$$u^{k+1} = u^k + \int_{t^k}^{t^k+1} f(u)\mathrm{d}t. \tag{A.20}$$

Let

$$I_k = \int_{t^k}^{t^{k+1}} f(t)\mathrm{d}t, \tag{A.21}$$

and let I_k^h denote the approximation of (A.21) utilized in the time-stepping scheme. We define the *local truncation error* as follows

$$\tau_k = \frac{I_k - I_k^h}{h^k}. \tag{A.22}$$

With (7.1) in mind, for fixed ξ, η, we have from lemma (A.4.1)

$$\int_{t^k}^{t^{k+1}} \mathbf{f}_u(t)\mathrm{d}t = \left[-u^k \partial_t \left(\ln \sqrt{g_k} \right) + d_1 \Delta_k u^k - F u^k - u^k \left(v^k \right)^2 + F \right] h^k + \mathcal{O}(h_k^2). \tag{A.23}$$

Expanding $u^k = u^{k+1} + \mathcal{O}(h^k)$, we have

$$\int_{t^k}^{t^{k+1}} \mathbf{f}_u(t)\mathrm{d}t = \left[-u^{k+1} \partial_t \left(\ln \sqrt{g_k} \right) + d_1 \Delta_k u^{k+1} - F u^{k+1} - u^k \left(v^k \right)^2 + F \right] h^k + \mathcal{O}(h_k^2). \tag{A.24}$$

and replacing

$$\partial_t \left(\ln \sqrt{g_k} \right) \to \frac{\ln \sqrt{g_k} - \ln \sqrt{g_{k-1}}}{h_{k-1}} \tag{A.25}$$

yields an additional $\mathcal{O}(h_{k-1})$-error. Therefore, we have

$$\tau_k(\mathbf{f}_u) = \mathcal{O}(h_k) + \mathcal{O}(h_{k-1}). \tag{A.26}$$

Similarly, we find

$$\tau_k(\mathbf{f}_v) = \mathcal{O}(h_k) + \mathcal{O}(h_{k-1}), \tag{A.27}$$

and

$$\tau_k(\mathbf{f_s}) = \mathcal{O}(h_k), \tag{A.28}$$

where $\tau_k(\mathbf{f}_u)$, $\tau_k(\mathbf{f}_v)$ and $\tau_k(\mathbf{f_s})$ constitute the local truncation errors of the time-discretization

$$\begin{aligned}
\int_{t^k}^{t^{k+1}} \mathbf{f}_u \mathrm{d}t &\simeq h^k \left[-u^{k+1}\partial_t^h \left(\ln\sqrt{g_k}\right) + d_1 \Delta_k u^{k+1} - F u^{k+1} - u^k \left(v^k\right)^2 + F \right] \\
\int_{t^k}^{t^{k+1}} \mathbf{f}_v \mathrm{d}t &\simeq h^k \left[-v^{k+1}\partial_t^h \left(\ln\sqrt{g_k}\right) + d_1 \Delta_k v^{k+1} - (F+H) v^{k+1} + u^k \left(v^k\right)^2 \right] \\
\int_{t^k}^{t^{k+1}} \mathbf{f_s} \mathrm{d}t &\simeq h^k K \mathbf{n}^k v^{k+1},
\end{aligned} \tag{A.29}$$

introduced in section 7.2.

A.5. Appendix Space-Time Galerkin

Here we list a collection of derivatives that can be used in chapter 13.

$$\begin{aligned}
\frac{\partial F_u^i}{\partial c_j^u} &= \int_0 \int_\Omega \left[d_1 \langle \nabla_\tau w_i, \nabla_\tau w_j \rangle_{g_\tau} - w_i w_j (1+v^2) - w_i \partial_\tau w_j \right] \sqrt{g_\tau} - w_i w_j \partial_\tau \left(\sqrt{g_\tau}\right) h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \\
\frac{\partial F_u^i}{\partial c_j^v} &= \int_0^1 \int_\Omega \left[-2uv w_i w_j \right] \sqrt{g_\tau} h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \\
\frac{\partial F_u^i}{\partial c_j^l} &= \int_0^1 \int_\Omega w_i \left[F(1-u) - u - uv^2 - \partial_\tau u \right] \frac{\partial \sqrt{g_\tau}}{\partial c_j^l} + w_i d_1 \frac{\partial}{\partial c_j^l} \left(\langle \nabla_\tau w_i, \nabla_\tau u \rangle_{g_\tau} \sqrt{g_\tau} \right) h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \\
\frac{\partial F_v^i}{\partial c_j^u} &= \int_0^1 \int_\Omega w_i w_j v^2 \sqrt{g_\tau} h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}t \\
\frac{\partial F_v^i}{\partial c_j^v} &= \int_0^1 \int_\Omega \left[d_2 \langle \nabla_\tau w_i, \nabla_\tau w_j \rangle_{g_\tau} - w_i w_j (2uv - F - k) - w_i \partial_\tau w_j \right] \sqrt{g_\tau} - w_i w_j \partial_\tau \left(\sqrt{g_\tau}\right) h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \\
\frac{\partial F_v^i}{\partial c_j^l} &= \int_0^1 \int_\Omega w_i \left[-v(1+F+k) + uv^2 - \partial_\tau v \right] \frac{\partial \sqrt{g_\tau}}{\partial c_j^l} + w_i d_2 \frac{\partial}{\partial c_j^l} \left(\langle \nabla_\tau w_i, \nabla_\tau u \rangle_{g_\tau} \sqrt{g_\tau} \right) h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \\
\frac{\partial F_{\mathbf{s}_l}^i}{\partial c_j^u} &= 0 \\
\frac{\partial F_{\mathbf{s}_l}^i}{\partial c_j^v} &= \int_0^1 \int_\Omega w_i \left[K w_j n_l - \partial_\tau w_j \right] \sqrt{g_\tau} \mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \\
\frac{\partial F_{\mathbf{s}_l}^i}{\partial c_j^r} &= \int_0^1 \int_\Omega w_i \left[K v \frac{\partial n_l}{\partial c_j^r} - \frac{\partial}{\partial c_j^r} \left(\partial_\tau \mathbf{s}\right)_l \right] \sqrt{g_\tau} + w_i \left[K v n_l - \left(\partial_\tau \mathbf{s}\right)_l \right] \frac{\partial \sqrt{g_\tau}}{\partial c_j^r} h_k \mathrm{d}\boldsymbol{\xi}\mathrm{d}t,
\end{aligned} \tag{A.30}$$

where

$$\begin{aligned}
\partial_\tau \sqrt{g_\tau} &= \frac{1}{2\sqrt{g_\tau}} \frac{\partial}{\partial t} \left\| \frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right\|^2 \\
&= \frac{1}{\sqrt{g_\tau}} \left(\frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right) \cdot \frac{\partial}{\partial t} \left(\frac{\partial \mathbf{s}}{\partial \xi} \times \frac{\partial \mathbf{s}}{\partial \eta} \right) \\
&= \mathbf{n} \cdot \left[\left(\frac{\partial^2 \mathbf{s}}{\partial \xi \partial t} \times \frac{\partial \mathbf{s}}{\partial \eta} \right) + \left(\frac{\partial^2 \mathbf{s}}{\partial \eta \partial t} \times \frac{\partial \mathbf{s}}{\partial \xi} \right) \right]
\end{aligned} \tag{A.31}$$

and

$$\begin{aligned}\frac{\partial\sqrt{g_\tau}}{\partial c_j^r} &= \frac{1}{2\sqrt{g_\tau}}\frac{\partial}{\partial c_j^r}\left\|\frac{\partial \mathbf{s}}{\partial \xi}\times\frac{\partial \mathbf{s}}{\partial \eta}\right\|^2 \\ &= \frac{1}{\sqrt{g_\tau}}\left(\frac{\partial \mathbf{s}}{\partial \xi}\times\frac{\partial \mathbf{s}}{\partial \eta}\right)\cdot\frac{\partial}{\partial c_j^r}\left(\frac{\partial \mathbf{s}}{\partial \xi}\times\frac{\partial \mathbf{s}}{\partial \eta}\right) \\ &= \mathbf{n}\cdot\left[\frac{\partial w_j}{\partial \xi}\left(\mathbf{e}_r\times\frac{\partial \mathbf{s}}{\partial \eta}\right)+\left(\frac{\partial \mathbf{s}}{\partial \xi}\times\mathbf{e}_r\right)\frac{\partial w_j}{\partial \eta}\right],\end{aligned} \tag{A.32}$$

where $\mathbf{e}_i$ is the i-th canonical unit vector in $\mathbb{R}^3$.
Furthermore, we use

$$\begin{aligned}\frac{\partial}{\partial c_j^l}\left(\langle\nabla_\tau w_i,\nabla_\tau u\rangle_{g_\tau}\sqrt{g_\tau}\right) &= -\langle\nabla_\tau w_i,\nabla_\tau u\rangle_{g_\tau}\frac{\partial\sqrt{g_\tau}}{\partial c_j^r} \\ &+\frac{1}{\sqrt{g_\tau}}\left(\left[\frac{\partial}{\partial c_j^r}\tilde{J}\right]\nabla w_i\right)\cdot\left([\tilde{J}]\,\nabla u\right)+\left([\tilde{J}]\,\nabla w_i\right)\cdot\left(\left[\frac{\partial}{\partial c_j^r}\tilde{J}\right]\nabla u\right),\end{aligned} \tag{A.33}$$

where

$$[\tilde{J}] = \begin{pmatrix}\frac{\partial \mathbf{s}}{\partial \eta} & -\frac{\partial \mathbf{s}}{\partial \xi}\end{pmatrix}, \tag{A.34}$$

and

$$\left[\frac{\partial}{\partial c_j^r}\tilde{J}\right] = \begin{pmatrix}\frac{\partial w_j}{\partial \eta}\mathbf{e}_r & -\frac{\partial w_j}{\partial \xi}\mathbf{e}_r\end{pmatrix}. \tag{A.35}$$

Bibliography

[1] Gaussian quadrature weights and abscissae. `http://pomax.github.io/bezierinfo/legendre-gauss.html`. Accessed: 29-06-2015.

[2] Gray-scott patterns for various f and h. `http://mrob.com/pub/comp/xmorphia/`. Accessed: 12-01-2016.

[3] Nutils. `http://www.nutils.org/`. Accessed: 12-01-2016.

[4] P Antolin, A Buffa, F Calabro, M Martinelli, and G Sangalli. Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. *Computer Methods in Applied Mechanics and Engineering*, 285:817–828, 2015.

[5] Roland Becker and Rolf Rannacher. *A feed-back approach to error control in finite element methods: Basic analysis and examples*. Citeseer, 1996.

[6] Adi Ben-Israel. A newton-raphson method for the solution of systems of equations. *Journal of Mathematical analysis and applications*, 15(2):243–252, 1966.

[7] A Biondi, H Nogueira, D Dormont, M Duyme, D Hasboun, A Zouaoui, M Chantome, and C Marsault. Are the brains of monozygotic twins similar? a three-dimensional mr study. *American journal of neuroradiology*, 19(7):1361–1367, 1998.

[8] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, pages 577–593, 1965.

[9] Florian Buchegger, Bert Jüttler, and Angelos Mantzaflaris. Adaptively refined multi-patch b-splines with enhanced smoothness. *Applied Mathematics and Computation*, 272:159–172, 2016.

[10] Jean Céa. Approximation variationnelle des problèmes aux limites. In *Annales de l'institut Fourier*, volume 14, pages 345–444, 1964.

[11] Annabelle Collin, Giancarlo Sangalli, and Thomas Takacs. Approximation properties of multi-patch c^1 isogeometric spaces. *arXiv preprint arXiv:1509.07619*, 2015.

[12] J Austin Cottrell, Thomas JR Hughes, and Yuri Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[13] Carl De Boor. *A Practical Guide to Splines*, volume 10.

[14] Carl De Boor. A practical guide to splines. *Mathematics of Computation*, 1978.

[15] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. Thb-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[16] Magnus Rudolph Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. 1952.

[17] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39):4135–4195, 2005.

[18] Tijana T Ivancevic. *Applied differential geometry: a modern introduction*. World Scientific, 2014.

[19] David Kahaner, Cleve Moler, and Stephen Nash. Numerical methods and software. *Englewood Cliffs: Prentice Hall, 1989*, 1, 1989.

[20] E. Kreyszig. *Differential Geometry*. Differential Geometry. Dover Publications, 1991. ISBN 9780486667218. URL `https://books.google.nl/books?id=P73DrhE9F0QC`.

[21] Dimitri Kuzmin, Rainald Löhner, and Stefan Turek. *Flux-Corrected Transport: Principles, Algorithms, and Applications. Scientific Computation*. Springer, 2005.

[22] Eric Kvaalen. A faster broyden method. *BIT Numerical Mathematics*, 31(2):369–372, 1991.

[23] John M Lee. *Introduction to Smooth manifolds*. Springer Verlag, New York, 2001.

[24] Kyoung J Lee, WD McCormick, Qi Ouyang, and Harry L Swinney. Pattern formation by interacting chemical fronts. *Science*, 261(5118):192–194, 1993.

[25] Julien Lefèvre and Jean-François Mangin. A reaction-diffusion model of the human brain development. In *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pages 77–80. IEEE, 2010.

[26] Andrei Ludu. *Nonlinear waves and solitons on contours and closed surfaces*. Springer Science & Business Media, 2012.

[27] Dominik Meidner, Rolf Rannacher, and Jevgeni Vihharev. Goal-oriented error control of the iterative solution of finite element equations. *Journal of numerical mathematics*, 17(2):143–172, 2009.

[28] JD Murray. On pattern formation mechanisms for lepidopteran wing patterns and mammalian coat markings. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 295(1078): 473–496, 1981.

[29] BN Nagorcka and JR Mooney. The role of a reaction-diffusion system in the initiation of primary hair follicles. *Journal of theoretical biology*, 114(2):243–272, 1985.

[30] Barrett O'neill. *Elementary differential geometry*. Academic press, 2006.

[31] Roger P Pawlowski, John N Shadid, Joseph P Simonis, and Homer F Walker. Globalization techniques for newton-krylov methods and applications to the fully coupled solution of the navier-stokes equations. *SIAM review*, 48(4):700–721, 2006.

[32] John E Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.

[33] Ramón G Plaza, Faustino Sanchez-Garduno, Pablo Padilla, Rafael A Barrio, and Philip K Maini. The effect of growth and curvature on pattern formation. *Journal of Dynamics and Differential Equations*, 16(4):1093–1121, 2004.

[34] Steven Rosenberg. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. Number 31. Cambridge University Press, 1997.

[35] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[36] T Sachs. Patterned differentiation in plants. *Differentiation*, 11(1):65–73, 1978.

[37] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.

[38] AMP Valli, GF Carey, and ALGA Coutinho. Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Communications in Numerical Methods in Engineering*, 18(2):131–139, 2002.

[39] Henk A Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2): 631–644, 1992.

[40] Jos van Kan, A Segal, and Fred Vermolen. *Numerical methods in scientific computing*. VSSD, 2005.

[41] C. Vuik, P Van Beek, F Vermolen, and J Van Kan. *Numerical Methods for Ordinary differential equations*. VSSD, 2007.

[42] Anh-Vu Vuong. *Adaptive Hierarchical Isogeometric Finite Element Methods*. Springer Science & Business Media, 2012.

[43] Julius Weingarten. Ueber eine klasse auf einander abwickelbarer flächen. *Journal für die reine und angewandte Mathematik*, 59:382–393, 1861.