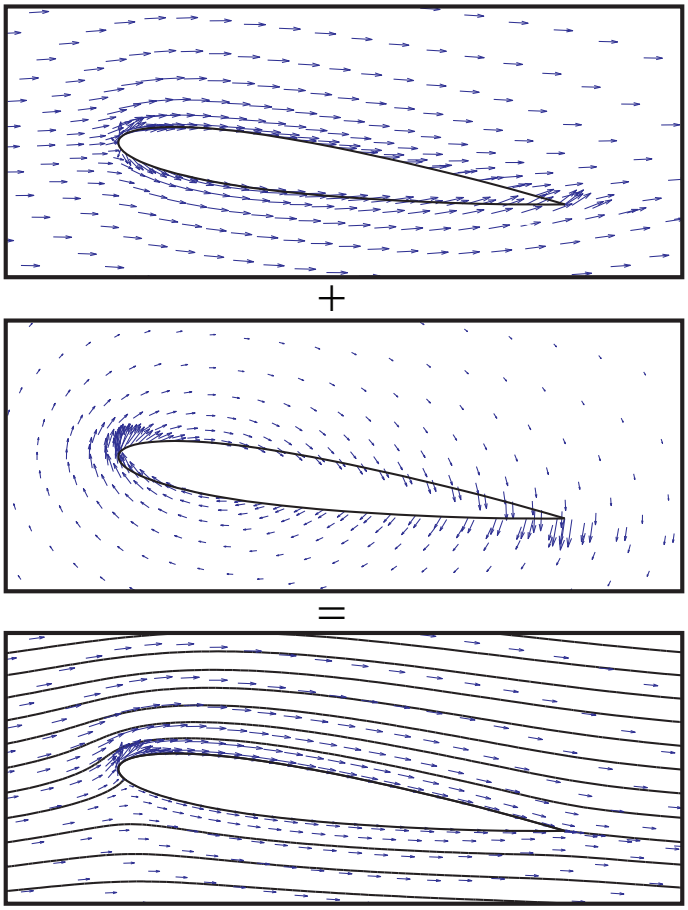
Master of Science Thesis

IsoGeometric Mimetic Methods

Applied geometry in CFD

René Hiemstra

June 11, 2011



Faculty of Marine Technology

Delft University of Technology

IsoGeometric Mimetic Methods

Applied geometry in CFD

Master of Science Thesis

For obtaining the degree of Master of Science in Marine Technology at
Delft University of Technology

René Hiemstra

June 11, 2011

Faculty of Marine Technology · Delft University of Technology

Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF MARINE TECHNOLOGY

The undersigned hereby certify that they have read and recommend to the Faculty of Marine Technology for acceptance the thesis entitled **"IsoGeometric Mimetic Methods"** by **René Hiemstra** in fulfillment of the requirements for the degree of **Master of Science**.

Dated: June 11, 2011

Supervisors:

Prof.Dr.Ir. R.H.M. Huijsmans

Dr.Ir. M.I. Gerritsma

Dr.Ir. H.J. de Koning Gans

Dr. S.J. Hulshoff

Dr.Ir. G. Vaz

"Let no man ignorant of geometry enter here"

(Plato. ca 429-347 BC.)

Preface

Reaching the summer of 2009 I finished CFD1, a first years master course given by Dr. Ir. Marc Gerritsma at the TU Delft faculty of Aerospace Engineering. This was my first real acquaintance with the field of Computational Fluid Dynamics. Being familiar with splines from the context of Computer Aided Design, I asked Marc, whether splines were also considered as a basis for finite element analysis in CFD. I thought, why use C^0 finite elements, when you can use splines which have continuous derivatives resulting in a continuous representation of field quantities and derivatives.

Marc introduced me to the recent developments of IsoGeometric Analysis, a novel concept employing CAD technology in finite element analysis, with as main goal the seamless integration of the two fields. I was directly convinced that this new technology would change Analysis for good.

Some nine months later, when I returned from my study abroad at the University of Southampton, I showed Marc a few pages I wrote down on a finite volume type of discretization technique which employed B-splines as a computational basis. In this, Marc recognized that B-splines are also a possible basis for the mimetic discretization methods, he and his students were working on. And so, we decided to combine IsoGeometric Analysis with Mimetic Methods as a topic for my master thesis.

This thesis represents the last milestone in becoming a maritime engineer. I would like to express my gratitude to those who helped to reach the fulfillment of my thesis. Special thanks goes to Marc, who was a guide throughout this thesis work, and always supported my ideas. To Dennis Ernens, with whom I had fruitful discussions concerning IsoGeometric Analysis. Most gratitude goes towards my family. Especially my mother and father, who always supported me without having any idea what I was actually doing. And Lastly, to my girlfriend, Ece, who was always there to distract me with her smile and with whom I shared the good and bad moments during this final year. She is probably even more thrilled that I am finally finishing, then I am myself.

Summary

In this thesis I present a novel discretization procedure which combines two relatively new technologies for solving partial differential equations (PDE's):

IsoGeometric Analysis (IGA) is a new paradigm which provides an exact geometry description and tight integration of Computer Aided Design (CAD) and Finite Element Analysis (FEA) by using the same basis for representation of the unknown field variables as is used for describing the geometry in CAD.

Mimetic Discretization Methods on the other hand combine concepts from the Finite Element Method (FEM) and the Finite Volume Method (FVM) and provide a unified and straightforward approach to model any physical field problem. Mimetic Methods aim at preserving as much as possible the structure of a PDE by 'mimicking' at the discrete level, important properties of the continuous realm, such that symmetries and conserved quantities are preserved. Central in this framework is the relation between physics and geometry.

The Mimetic Discretization approach developed in this thesis is based upon B-splines[1] for representing the unknown field variables. Besides inheriting all advantages from the IsoGeometric Analysis framework, B-splines appear as a natural basis for Mimetic Discretization Methods. They can be seen as higher order Whitney forms and provide vector spaces which are discretely conservative by construction. The resulting discretization approach resembles a Finite Volume Method on a staggered grid for the representation of the conservation laws and a Finite Element Method for the representation of the constitutive equations. In short, the scheme features the following advantages,

- exact geometry description and tight integration with CAD;
- fundamentally a higher order approach, featuring spectral like convergence. In practice though, IGA is confined to low or medium order due to bad conditioning of inner product mass matrices as a function of the polynomial order;

[1] All modern CAD technologies are derived from B-splines

- increased continuity resulting in continuous representations of field variables and derivatives;
- a strong indication exists that these methods automatically meet the inf-sup conditions, leading to naturally stable discretizations of any physical problem;
- local conservation of primal variables (strong) and secondary variables (weak);
- in contrast to FVM and FEM which describe variables only locally, the Mimetic discretization is induced with a global topology which makes it possible to make useful decompositions of field variables.

We have made use of this final property to perform numerical calculations of irrotational incompressible flows that include lift. Although these problems seem elementary, conventional methods, like the FEM and FVM are unable to solve these kind of problems without resorting to an engineering approach. We presented a novel procedure, which provides a discrete analogue to the harmonic function, which can directly be related to the lift. Although this procedure has only been applied to flows around a single object, it can readily be applied to lifting flows around more than one object. The results show that the lift can accurately be approximated, already on quite coarse meshes.

Table of Contents

Preface vii

Summary ix

List of Figures xvii

List of Tables xxvii

1 Introduction 1

1.1 IsoGeometric Analysis . . . 1

1.1.1 Practical application of IGA . . . 5

1.2 Mimetic Discretization Methods . . . 9

1.2.1 Preliminary classification of field quantities . . . 10

2 Foundations of Mimetic Discretization Methods 13

2.1 Structure of physical problems . . . 13

2.1.1 Physical quantities, geometry and orientation . . . 14

2.1.2 Local verses global quantities . . . 15

2.1.3 Topological relations . . . 15

2.1.4 Constitutive relations . . . 17

2.1.5 Classification of physical quantities . . . 18

2.2 Continuous modeling - concepts from Differential Geometry . . . 19
2.2.1 Manifolds . . . 19
2.2.2 The exterior product between forms . . . 24
2.2.3 Exterior Differential Forms . . . 25
2.2.4 The exterior derivative d . . . 28
2.2.5 The Hodge star operator $*$ . . . 30
2.2.6 Pulling back forms . . . 33
2.2.7 Integrating forms . . . 36
2.3 Discrete modeling - concepts from Algebraic Topology . . . 36
2.3.1 Cell complexes . . . 37
2.3.2 Chains and the boundary operator ∂ . . . 40
2.3.3 Cochains and the co-boundary operator δ . . . 42
2.3.4 Incidence matrices . . . 45
2.3.5 Discrete De Rahm sequence . . . 48
2.4 Differential Geometry and Algebraic Topology connected . . . 48
2.4.1 Interpolation of differential forms / Reduction to cochains . . . 48

3 B-splines and NURBS as a basis for geometry and mesh generation 51

3.1 Things to know about CAD before starting IsoGeometric Analysis . . . 51
3.2 B-splines . . . 55
3.2.1 B-spline basis functions . . . 55
3.2.2 The knot vector . . . 58
3.2.3 B-spline curves . . . 60
3.2.4 B-spline surfaces . . . 62
3.2.5 Derivatives of B-splines . . . 64
3.2.6 Global interpolation in curved coordinates . . . 65
3.3 Refinement . . . 66

3.3.1 Knot insertion - h-refinement . . . 66
3.3.2 Degree elevation - p-refinement . . . 68
3.3.3 Increased order and continuity - k-refinement . . . 70
3.4 NURBS - Non-Uniform Rational B-splines . . . 72
3.4.1 NURBS curves . . . 72
3.4.2 NURBS surfaces . . . 73
3.4.3 Derivatives of NURBS . . . 75
3.5 Current limitations of NURBS as a basis for geometry and mesh generation . . . 76

4 IsoGeometric Mimetic Discretization Methods 79
4.1 Introduction . . . 79
4.2 Spline differential forms in 1D . . . 80
4.2.1 Reconstruction of 0-forms . . . 81
4.2.2 Reconstruction of 1-forms . . . 83
4.2.3 Reduction of 0-forms to B-spline control points . . . 85
4.2.4 Reduction of 1-forms to B-spline control edges . . . 89
4.3 Spline differential forms in 2D . . . 91
4.3.1 Primal cell complex . . . 91
4.3.2 Dual cell complex . . . 95
4.4 Application of B-splines to the topological relations . . . 98
4.4.1 The gradient operator . . . 98
4.4.2 The curl operator . . . 99
4.4.3 The divergence operator . . . 100
4.5 Application of B-splines to the Metric dependent relations . . . 101
4.5.1 Discrete Hodge star . . . 102
4.5.2 Boundary conditions . . . 107
4.5.3 Discrete DeRahm sequence revisited . . . 109

4.5.4 The discrete co-derivative . . . 110
4.5.5 Discrete scalar and vector Laplacian . . . 110
4.6 Numerical experiments - 2D discretization of the poisson equation . . . 113

5 Practical Applications in 2D 119

5.1 Discrete representation of the harmonic form . . . 121
5.1.1 The harmonic cochain under mesh refinement . . . 125
5.2 Kutta condition . . . 125
5.3 Discrete equations for potential flows that include lift . . . 127
5.4 Determination of the lift . . . 127
5.5 Lifting flow over a cylinder . . . 128
5.5.1 Problem setup . . . 129
5.5.2 NURBS geometry and mesh generation . . . 130
5.5.3 Results for uniform flow around the cylinder . . . 133
5.5.4 Results for lifting flow over the cylinder with $P_{stag} = (1, -1/4\pi)$ . . . 135
5.5.5 Results for lifting flow over the cylinder with $P_{stag} = (1, -1/2\pi)$ . . . 137
5.5.6 Results for lifting flow over the cylinder with $P_{stag} = (2, -1/2\pi)$ . . . 139
5.5.7 Discussion . . . 141
5.6 Lifting potential flow over the NACA0012 airfoil . . . 142
5.6.1 Problem setup . . . 142
5.6.2 NURBS geometry and mesh generation . . . 142
5.6.3 Results uniform flow around the NACA0012 . . . 145
5.6.4 Results lifting flow over the NACA0012 with $2\,\mathrm{deg}$ of incidence . . . 147
5.6.5 Results lifting flow over the NACA0012 with $4\,\mathrm{deg}$ of incidence . . . 149
5.6.6 Results lifting flow over the NACA0012 with $6\,\mathrm{deg}$ of incidence . . . 151
5.6.7 Results lifting flow over the NACA0012 with $8\,\mathrm{deg}$ of incidence . . . 153
5.6.8 Discussion . . . 155

Conclusion **157**

Recommendations **159**

Bibliography **161**

A Continuous field reconstruction **167**

A.1 Edge functions 169

A.2 Application of the edge functions to the grad, curl and div 170

List of Figures

1.1 The two-dimensional Boussinesq equations, modeling shallow water free surface flows. The x-component of velocity obtained using 552 triangles with fifth order polynomials on each triangle. (a) The spurious oscillations in the solution are due to the use of straight-sided elements for the geometric approximation. (b) The cylinder is approximated by elements with curved edges, and the oscillations are eliminated. From Eskilsson and Sherwin [31] 2

1.2 (a) CAD geometric model of a horseshoe. (b) coarse mesh straight from CAD system. (c) and (d) model is refined for analysis without affecting geometry. From Hughes et al. [42]. 3

1.3 (a) Vertical stress component after analysis. (b) 'Spectral' like convergence. From Hughes et al. [42]. 3

1.4 Comparison of C^0 finite elements and C^{p-1} Isogeometric Analysis employing B-splines. From Cottrell et al. [22]. 4

1.5 Convection-Diffusion in two dimensions. Problem features sharp boundary layers skew to the mesh is $45\,\mathrm{deg}$. The mesh is uniform and spans 20 by 20 elements. The number of dofs is $(20+p)^2$ where p is the degree of the basis. Obtained results with increasing degree. From Hughes et al. [42]. 5

1.6 Every geometric feature of the NASA Aluminum testbed cylinder (ATC) is accounted for, Figure 1.6(a). The mesh consisted of 228.936 rational quadratic elements and 2.219.184 degrees of freedom. Figure 1.6(b) compares numerical results with experiments illustrating that IsoGeometric Analysis can be applied to large complex problems. From Cottrell et al. [23]). 6

1.7 (a) Problem setup and (b) construction of analysis suitable geometry of one third of the flow domain consisting of approximately 1.5 million degrees of freedom. From Bazilevs et al. [12]. (c) Pressure contours at several blade cross-sections viewed from the back of the blade plotted on the deformed configuration. The large negative pressure at the suction side of the airfoil creates a favorable aerodynamic torque. (d) Iso-surfaces of air speed. The flow exhibits complex behavior. The vortical feature generated at the blade tip is convected downstream of the rotor with very little decay. (e) Iso-contours of relative wind speed at a 30m radial cut at different time instances superposed on a moving blade. The flow stays fully attached on the pressure side and separates on the suction side. Note the deflection of the blades. From Bazilevs et al. [13] 7

1.8 Problem setup dam break with obstacle problem . From Akkerman et al. [4] . . . 8

1.9 Comparison of the solution produced by the coarse and fine meshes.. From Akkerman et al. [4] . . . 8

1.10 Dam break with obstacle. Pressure time history at four points on the object-comparison between experimental data (labeled MARIN) and computational results. The main trend is captured on both meshes. From Akkerman et al. [4] . . 9

1.11 Free surface flow around a Wigley hull at Fr $= 0.25$. From Akkerman et al. [4] . 9

1.12 Preliminary classification of physical quantities in a theory . . . 11

1.13 Hodge decomposition of an inviscid flow around a lifting airfoil. (Top) Contribution due to gradient of potential. (Middle) Contribution due to harmonic function. (Bottom) Resulting flow. . . . 12

2.1 Physical field quantities related to inner oriented geometric objects in $\mathbb{R}^3$ . . . 14

2.2 Physical field quantities related to outer oriented geometric objects in $\mathbb{R}^3$ . . . 15

2.3 Fundamental theorem of calculus, Figure 2.3(a), relating global quantities associated with lines to global quantities at the boundary points. The Stokes theorem, Figure 2.3(b), relating global quantities associated with surfaces to global quantities along its boundary curves. The divergence theorem, Figure 2.3(c), relating global quantities associated with a volume to global quantities at the bounding surfaces. Observe that a change in geometry doesn't affect the discrete relations. 16

2.4 The topological relations form the horizontal connections in the diagram. These relations thus involve only objects of the same orientation. . . . 17

2.5 The constitutive relations appear as the vertical links between dual geometric objects 18

2.6 The Tonti classification diagram of physical quantities in three dimensional space. Each slot represents a physical quantity referred to a geometric object; that is scalar quantities living in points $1P$, vector valued quantities related to lines $3L$, vector valued quantities related to surfaces $3S$ and scalar valued quantities associated with volumes $1V$. The topological relations appear as the horizontal links between quantities associated with geometric objects of the same orientation, and have an intrinsically discrete nature. The constitutive relations appear as the vertical connections between quantities of opposed orientation. It is in general here where the discrete approximation comes in. Diagonal connections are also possible. These processes are, however, irreversible. From Mattiussi [52] . . . 18

2.7 The earth (surface of a sphere) is a two-dimensional manifold in $\mathbb{R}^3$ since it can be represented by a collection of two-dimensional charts. . . . 20

2.8 The tangent space of a 1-manifold in $\mathbb{R}^3$ . . . 20

2.9 The tangent space of a 2-manifold in $\mathbb{R}^3$ . . . 21

2.10 The same vector $\boldsymbol{\alpha}$ in two different coordinate systems . . . 22

2.11 Vector $\boldsymbol{\alpha}$ in terms of the primal and dual basis 23

2.12 2.12(a) The exterior product of two 1-forms $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ yields the 2-form $\boldsymbol{\alpha} \wedge \boldsymbol{\beta}$. 2.12(b) Multiplying by yet another 1-form $\boldsymbol{\gamma}$ give the 3-form $\boldsymbol{\alpha} \wedge \boldsymbol{\beta} \wedge \boldsymbol{\gamma}$ 25

2.13 Inner and outer oriented spaces for exterior differential forms in $\mathbb{R}^3$ 28

2.14 Double De Rahm sequence for respectively the inner oriented and outer oriented differential forms in $\mathbb{R}^3$ 30

2.15 The Hodge star operator $*$ establishes the vertical links between dual differential forms of opposite orientation. 31

2.16 Pulling back 1-form α from physical space Ω to the reference domain Ω'. 34

2.17 Topologically equivalent grids 37

2.18 Faces and cofaces in $\mathbb{R}^2$. Figure (a) depicts a 1-cell and a 2-cell and their accompanying faces, the 0-cells and 1-cells at their boundary. Figure (b) illustrates the inverse relation. The co-faces of a 1-cell are its neighboring 2-cells and the co-faces of a 0-cell are its neighboring 1-cells. 38

2.19 The two dimensional mesh in Figure 2.17 can for example be divided into 0-,1- and 2-cells. The collection of all these k-cells forms the cell complex, which in practical terms is simply our primal grid in topological space. 38

2.20 Inner and outer oriented geometric objects in $\mathbb{R}^2$, showing a '+' sign for compatible directions and '-' sign for incompatible directions between a k-cell and its boundary $(k-1)$-cell. 39

2.21 A numbered, outer oriented 2-dimensional primal cell complex. In the numbering of the different cells we make the distinction between 0-cells assigned by $\tilde{P}$, 1-cells by $\tilde{L}$ and 2-cells by $\tilde{S}$, where the tilde refers to the outer orientation. The numbering is arbitrary 39

2.22 The dual cell complex in $\mathbb{R}^2$. Inner oriented 0-cells P_i are associated with outer oriented 2-cells $\tilde{S}_i$ on the primal complex; inner oriented 1-cells L_i act dual to outer oriented 1-cells $\tilde{L}_i$ on the primal cell complex; and finally inner oriented 2-cells S_i on the dual complex act dual to inner oriented 0-cells $\tilde{P}_i$ on the primal complex. 40

2.23 A numbered and positively oriented two dimensional cell complex 41

2.24 The boundary of the boundary is empty. Observe that the third figure shows opposed orientation of the 1-cells that comprise the edges of the volume. They consequently cancel each other out. 41

2.25 Action of the co-boundary operator on an outer oriented $\tilde{0}$-cochain, relating for example discrete values of the streamfunction $\bar{\psi}^{(\tilde{0})}$ with discrete fluxes $\bar{q}^{(\tilde{1})}$. . . . 43

2.26 Action of the co-boundary operator on an outer oriented $\tilde{1}$-cochain, relating for example discrete fluxes $\bar{q}^{(\tilde{1})}$ with discrete mass-densities $\bar{m}^{(\tilde{2})}$. 43

2.27 Action of the co-boundary operator on a 0-cochain, relating for example discrete potentials $\bar{\varphi}^{(0)}$ and discrete velocities $\bar{u}^{(1)}$. 43

2.28 Action of the co-boundary operator on a 1-cochain, relating for example discrete velocities $\bar{u}^{(1)}$ with discrete vorticity $\bar{\omega}^{(2)}$ 44

2.29 The vector identities $\mathrm{curl} \cdot \mathrm{grad}$ and $\mathrm{div} \cdot \mathrm{curl}$ lead to the associated null space. A double action of the co-boundary operator to a k-cochain leads to an empty $(k+2)$-cochain. 44

2.30 Incidence matrices on the primal cell complex K in $\mathbb{R}^2$ 45

2.31 Incidence matrices on the dual complex in $\mathbb{R}^2$. Note that the orientation of the dual cell complex is opposite to that of the primal complex. 47

3.1 Ducks holding the spline in place on the drawing board 52

3.2 A cubic B-spline curve in two-dimensional space defined as $\mathbf{C}(x) = \sum_{i=0}^{6} \bar{\mathbf{P}}_i \; N_{i,3}(x), \qquad 0 \leq x \leq 4$. The curve consists out of 4 polynomial pieces joined together with curvature continuity. 53

3.3 In classical finite element analysis (3.3(a)), the parameter space is local to elements. Each element has its own mapping from the reference element to physical space. In NURBS based IsoGeometric Analysis on the other hand (3.3(b)), the B-spline parameter space is local to the entire patch. Internal knots partition the patch into elements. A single B-spline maps parameter space to physical space. From Cottrell et al. [22] 54

3.4 Recursive generation of a cubic B-spline basis using the De Boor formula and a knot vector of $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$ 56

3.5 The cubic B-spline basis function $N_{3,3}(x)$ is composed out of 4 different polynomial pieces. 57

3.6 The variation decreases for increasing degree. Note also that the B-spline is confined in the convex hull of its control polygon. 58

3.7 Figure 3.7(a) shows a uniform quadratic B-spline curve. The repeated knot in the knot vector $\mathbf{X}$ in figure 3.7(b) causes a kink in the curve, by introducing a new basis function and changing the existing basis functions only in span $x \in [3, 5]$. . 59

3.8 A quadratic B-spline curve is mapped from parameter space to physical space. The B-spline basis is defined by knot vector $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$. 61

3.9 Recursive generation of quadratic b-spline basis, defined by knot vector $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$, in knot span $x \in [2, 3]$ 62

3.10 Some examples of bi-cubic two-dimensional B-spline basis functions constructed by tensor products of cubic uni-variate basis functions. The one-dimensional bases are defined by the knot vectors $\mathbf{X}^1 = \mathbf{X}^2 = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$. The black selection denotes the support of the respective basis function 63

3.11 Bi-quadratic tensor product B-spline surface constructed from knot vectors $\mathbf{X}^1 = \{0, 0, 0, 2, 2, 2\}$ and $\mathbf{X}^2 = \{0, 0, 0, 1.5, 3, 3, 3\}$ 64

3.12 Knot insertion for a cubic degree B-spline curve built with knot vector $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$. Inserting a new knot, $\mathbf{X}_{ref} = \{0, 0, 0, 0, 1, 2, 2.5, 3, 4, 4, 4, 4\}$, changes the basis locally, see Figure 3.12(b) and 3.12(c). The new control points are calculated such that the geometry does not change, Figure 3.12(a) 68

3.13 Degree elevation of B-spline curve. Note the knot multiplicities in the knot vector. This is necessary in order to maintain the degree of continuity in the derivatives of the curve. 69

3.14 The original basis of degree 1, Figure 3.14(a), is refined using h-refinement, Figure 3.14(b), and p-refinement in Figure 3.14(c). Figure 3.14(d) shows the standard FEA approach of hp-refinement, where the basis is obtained after h-refinement and subsequent p-refinement. First elevating the degree in the original basis to $p = 2$ (3.14(c)) and subsequent h-refinement leads to the concept of k-refinement, Figure 3.14(e). Compare Figure 3.14(d) with 3.14(e) and observe that the increased smoothness of the basis leads to fewer basis functions. 71

3.15 A NURBS quarter circle constructed from quadratic basis functions and knot vector $\mathbf{X} = \{0, 0, 0, 1, 1, 1\}$. The dotted line denotes the quadratic curve with weights all equal to one. 73

3.16 A NURBS surface is mapped from the parametric domain, to the circular plane in physical space. The bi-quadratic surface is constructed using the knot vectors $\mathbf{X}^1 = \mathbf{X}^2 = \{0, 0, 0, 1, 1, 1\}$ and control points and corresponding weights shown. 74

3.17 Figure 3.17(a) shows the coarse mesh of a rectangular domain with circular hole modeled using four patches in a CAD program. The NURBS geometry is bi-quadratic and build from knot vectors $\mathbf{X^1} = \{0, 0, 0, 1, 1, 1\}$ and $\mathbf{X^2} = \{0, 0, 0, 0.5, 1, 1, 1\}$ and control points and weights from Figure 3.17(b). Note that the two control points, $\bar{\mathbf{P}}_{01}$ and $\bar{\mathbf{P}}_{02}$, have the same location and create the artificial upper left corner. Although this causes a singular point at this location, this poses no problem in IGA [42]. 74

3.18 The coarse mesh of the domain with circular hole is globally refined for analysis. Meshes produced using global h-refinement, where element size is halved each iteration. 75

3.19 Mesh 3 in Figure 3.18 is more locally refined near the boundary of the hole. . . . 75

3.20 A CAD model of a ships propeller can be quite a complicated geometry to model. The surface description in Figure 3.20(a), consists out of 72 patches. Continuity constraints between patches make sure that the overall surface is smooth, see Figure 3.20(b). 76

4.1 The NURBS control mesh can be seen as a primal cell complex consisting of control points P_i, control edges L_i and control faces S_i. The control points, edges and faces have been given the inner orientation. 80

4.2 Given a zero cochain $\bar{\varphi}$ we can reconstruct a continuous representation of a 0-form $\phi_h^0(x) \in \mathcal{S}^0_{p,\mathbf{X}}$ by taking a linear combination of B-spline control points $\bar{\varphi}_i$ and B-spline basis functions $N_{i,p}(x)$. 82

4.3 This figure continues upon Figure 4.2. The coboundary process transfers quantities from the control points to control edges, Figure 4.3(a), and provides us a discrete relation of the gradient, $\bar{u}_i = \bar{\varphi}_i - \bar{\varphi}_{i-1}$. A continuous representation of the gradient $u_h^1(x) \in \mathcal{S}^1_{p-1,\mathbf{X}}$ can subsequently be obtained by reconstruction of $\bar{u} \in \mathcal{C}^1$ using the Curry Schoenberg B-spline basis functions $M_{i,p-1}$ which are associated with control edges. These edge type of basis functions are simply differently normalized B-splines, see Figure 4.3(b). 84

4.4 The B-spline coefficients $\bar{f}_i$, obtained by reduction of the continuous function $f(x) = \cos(\frac{\pi}{2}x)$, are interpolated by the B-spline basis functions $N_{i,p}(x)$, depicted below in Figure 4.4. 88

4.5 The top figure shows the absolute error between the exact function and B-spline approximation in Figure 4.4. Note that the approximation does not interpolate the exact function at the collocation points t_j. The error reduces however with the expected $p+1$ order of convergence in the L^2 norm under mesh refinement, Figure 4.5 below. 89

4.6 Primal cell complex build using $\mathbf{X}_1 = \{0,0,0,1,1,1\}$. We can choose to assign either inner (4.6(a)) or outer (4.6(b)) orientation to the control points, edges and faces. 92

4.7 Some modes of inner oriented B-spline differential forms on the primal cell complex. The knot vector which defines the B-spline spaces is $\mathbf{X}_1 = \mathbf{X}_2 = \{0,0,0,1,1,1\}$ 94

4.8 A B-spline dual grid is obtained by subtracting excess control points $\mathbf{o}$ and edges $- - -$ at the boundary. These can be eliminated by incorporating boundary conditions in strong form 95

4.9 Some modes of outer oriented B-spline differential forms. 97

4.10 The co boundary process gives a discrete equation of the gradient 99

4.11 The co boundary process provides a discrete equation of the curl 100

4.12 The co boundary process provides a discrete equation of the divergence 101

4.13 Bi-quadratic tensor product B-spline surface constructed from knot vectors $\mathbf{X}^1 = \{0,0,0,2,2,2\}$ and $\mathbf{X}^2 = \{0,0,0,1.5,3,3,3\}$ 103

4.14 The normal flux $u_n \, \mathrm{d}\Gamma$ at the east boundary needs to be taken into account. . . 108

4.15 Meshes for *Curved0*, produced with global h-refinement 113

4.16 Meshes for *Curved1*, produced with global h-refinement 114

4.17 Meshes for *Curved2*, produced with global h-refinement 114

4.18 Meshes for *Curved3*, produced with global h-refinement 114

4.19 Convergence results for $\varphi_h^0(\mathbf{y})$ with respect to step size. IGA MIM stands for the IsoGeometric Mimetic discretization approach proposed here, and IGA FEA stands for the Galerkin approach. Almost identical results are obtained for both methods. Observe that convergence rates start sub-optimal on the curved geometries. After sufficient refinement, however, the convergence rates are above optimal. 115

4.20 Contour plot of $\varphi_h^0(\mathbf{y})$ for the different geometries. Although *Curved3* is self overlapping, the obtained results on the fine mesh seem quite good. 116

4.21 Condition number of system matrix with respect to the polynomial degree. . . . 117

4.22 Condition number of system matrix with respect to the mesh size. 118

5.1 Hodge decomposition of an inviscid flow around a lifting airfoil. (Top) Contribution due to gradient of potential. (Middle) Contribution due to harmonic function. (Bottom) Resulting flow. 120

5.2 Non-contractible cell complex 122

5.3 Oscillatory flow around the hole in the non-contractible domain of figure 5.2 . . . 124

5.4 Kutta condition for the cylinder. We will enforce a stagnation point at P_{stag}. Since the velocity in the direction of $\mathbf{n}$ is already zero due to Neumann BC's, we only need to pose zero velocity in the direction of $\mathbf{t}$. 125

5.5 Kutta condition for the NACA0012. To enforce that the flow leaves the trailing edge smoothly we impose that the velocity in the direction of $\mathbf{t}$ is zero. In this case the flow leaves the trailing edge at TE in the direction of $\mathbf{n}$. $\mathbf{n}$ is thus tangent to the streamline at TE. 126

5.6 The circulation can be calculated by summing the harmonic cochains on the boundary, multiplied by $\bar{c}$. 128

5.7 Lifting flow over cylinder. 129

5.8 NURBS 2D geometric domain around a cylinder of radius $R = 1$. The geometry is modeled in a CAD program using four patches. The size of the domain is 5 by 5. The meshes are obtained by global h-refinement with increased resolution near the cylinder wall. 131

5.9 Each patch is refined for analysis. Calculations are performed on the above meshes. 131

5.10 Uniform flow around a cylinder. The blue circle denotes the stagnation point, which is chosen at $R = 1$ and $\theta = 0\,\mathrm{deg}$. The results have been obtained for bi-degree 6 mesh 3. 133

5.11 Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (1, 0)$. Figures (a) depicts the $L_2(\Omega)$ error in the potential. Optimal convergence results are obtained. Figure (b) depict the numerical results for the drag coefficients, which are close to machine precision. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ 134

5.12 Potential flow around a lifting cylinder. The stagnation point is chosen at $R = 1$ and $\theta = -1/4\pi$. The results have been obtained for mesh 3, bi-degree 6. 135

5.13 Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (1, -1/4\pi)$. Figures (a) and (b) depict the numerical results for the lift and drag coefficients. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ. 136

5.14 Potential flow around a lifting cylinder. The stagnation point is chosen at $R = 1$ and $\theta = -1/2\pi$. The results have been obtained for mesh 3, bi-degree 6. 137

5.15 Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (1, -1/2\pi)$. Figures (a) and (b) depict the numerical results for the lift and drag coefficients. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ. 138

5.16 Potential flow around a lifting cylinder. The stagnation point is chosen at $R = 2$ and $\theta = -1/2\pi$. The results have been obtained for mesh 3, bi-degree 6. 139

5.17 Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (2, -1/2\pi)$. Figures (a) and (b) depict the numerical results for the lift and drag coefficients. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ. 140

5.18 Problem setup, flow around the NACA0012 142

5.19 Coarse model of the flow domain, straight from CAD. Note that the corners have been artificially created by placing two control points at the same location. . . . 143

5.20 Meshes obtained using global h-refinement in the radial direction. 143

5.21 A closer look at the airfoil reveals the increased resolution we provided near the boundary 144

5.22 Potential flow around a lifting NACA0012 airfoil at 0 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6. 145

5.23 Uniform flow around the NACA0012. Figure (a) depicts the numerical drag coefficients. These are no longer close to machine precision, as for the cylinder. Figures (b) to (f) depict the pressure coefficient C_p over the airfoil and compares with the results obtained by a 2nd order panel method using 1000 panels. 146

5.24 Potential flow around a lifting NACA0012 airfoil at 2 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6. 147

5.25 Lifting flow over the NACA0012 with 2 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a small numerical drag component. Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels 148

5.26 Potential flow around a lifting NACA0012 airfoil at 4 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6. 149

5.27 Lifting flow over the NACA0012 with 4 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a small numerical drag component. Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels. 150

5.28 Potential flow around a lifting NACA0012 airfoil at 6 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6. 151

5.29 Lifting flow over the NACA0012 with 6 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a minor numerical drag component. Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels. 152

5.30 Potential flow around a lifting NACA0012 airfoil at 8 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6. 153

5.31 Lifting flow over the NACA0012 with 8 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a minor numerical drag component Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels. 154

A.1 1D expansion of a 0-form in terms of nodal basis functions 167

A.2 Relation between 168

A.3 A.14 is exact, coordinate free and invariant under C^1 transformations 171

A.4 A.16 is exact, coordinate free and invariant under C^1 transformations 172

A.5 A.21 is exact, coordinate free and invariant under C^1 transformations 173

List of Tables

3.1 Local knot vector associated with basis function $N_{i,2}$ 60

5.1 Lifting flow around the cylinder. Number of degrees of freedom for mesh 1 to 4 and polynomial degree 2 to 6. 132

5.2 Lifting flow over the NACA0012. Number of degrees of freedom for mesh 1 to for and polynomial degree 2 to 6. 144

Chapter 1

Introduction

In this thesis we combine the strengths of IsoGeometric Analysis and Mimetic Discretization methods. IsoGeometric Analysis provides an exact description of the geometry and seamless integration with CAD. This is achieved by using CAD technology in the discretization process.

Mimetic methods combine concepts from the Finite Element Method (FEM) and the Finite Volume Method (FVM) and provide a unified and straightforward approach to model any physical field problem. Mimetic Methods aim at preserving as much as possible the structure of a PDE by 'mimicking' at the discrete level, important properties of the continuous realm, such that fundamental symmetries and quantities are conserved. Central in this framework is the relation between physics and geometry.

1.1 IsoGeometric Analysis

Traditionally, geometry has been represented differently in the fields of Computer Aided Design (CAD) and Finite Element Analysis (FEA). This means that CAD geometry, which can be seen as exact, must be translated into an analysis suitable geometry (ASG) for meshing and input into a FEA program. This process is highly labor intensive, often costing more time than the actual analysis, and is accompanied by a loss of geometric precision.

A precise geometry description is important in many practical applications. Fluid structure interaction (FSI) requires a precise description of the fluid structure interface; non-linear phenomena like transition to turbulence and shell buckling analysis are extremely sensitive to small deviations in the geometry. An example where spurious oscillations arise due to crude geometric approximation with straight sided elements at the boundary is shown in Figure 1.1.

The trend towards increasingly complex and larger problems also demands higher order approximations and automatic adaptive mesh refinement. Efficient numerical procedures that

converge to a satisfactory result are in most cases impossible for large scale complex problems without accurate geometry description, higher order approximations and mesh adaptivity. The last one has not been widely adopted in the industry since mesh refinement requires access to the exact geometry, and therefore requires direct communication with the CAD-representation.

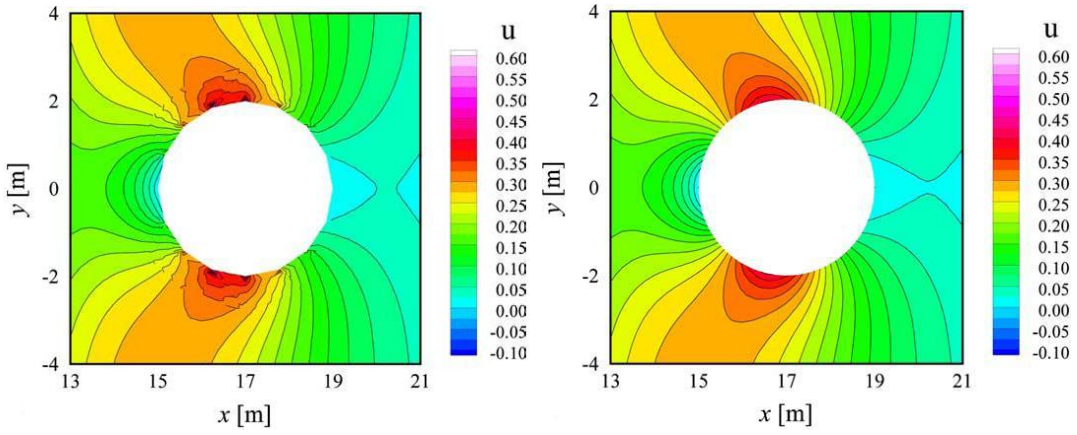


Figure 1.1: The two-dimensional Boussinesq equations, modeling shallow water free surface flows. The x-component of velocity obtained using 552 triangles with fifth order polynomials on each triangle. (a) The spurious oscillations in the solution are due to the use of straight-sided elements for the geometric approximation. (b) The cylinder is approximated by elements with curved edges, and the oscillations are eliminated. From Eskilsson and Sherwin [31]

IsoGeometric Analysis (IGA), introduced by Hughes et al. [42], addresses these issues, by directly integrating FEA with CAD. The IsoGeometric Analysis concept unifies the two fields of engineering design and analysis by expanding the unknown field variables in the same basis as that of the geometry description from CAD.

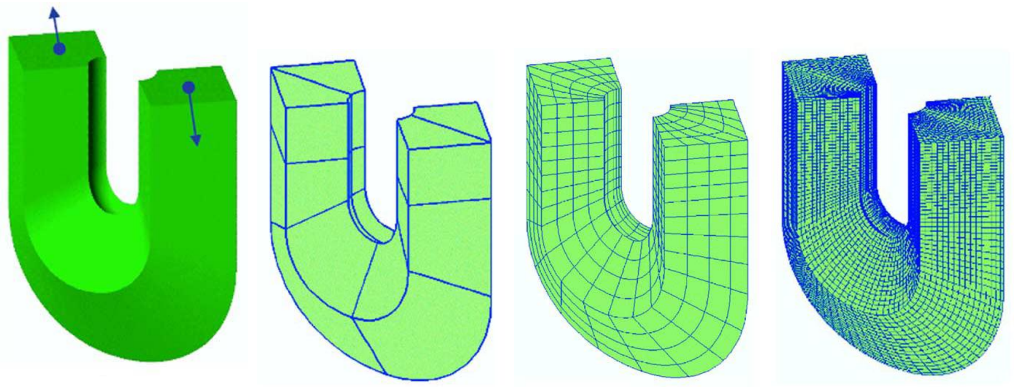
There are several CAD technologies available to the IsoGeometric Analysis framework, all of which are derived from B-splines. NURBS, short for Non Uniform Rational B-splines, are the standard technology employed in CAD systems. NURBS generalize B-splines and consequently inherit all their favorable properties for use in free form design. NURBS, however, extend B-splines by allowing the exact representation of conic sections like circles and ellipsoids. Furthermore, many efficient and numerically stable algorithms exist to generate NURBS and their derivatives [22, 54]. B-splines and NURBS also posses some mathematical properties desirable in Analysis:

- They feature linear independence, compact support and form a partition of unity;
- They are not polluted by spurious oscillations, such as the Runge phenomena;
- Polynomial order of accuracy is merely an input parameter allowing 'spectral like' convergence. in practice though, IGA, is confined to low or medium order due to the bad conditioning of mass matrices with respect to the polynomial order.

- Variation diminishing property, which causes B-splines to respond less to a change in the coefficients for increasing degree.

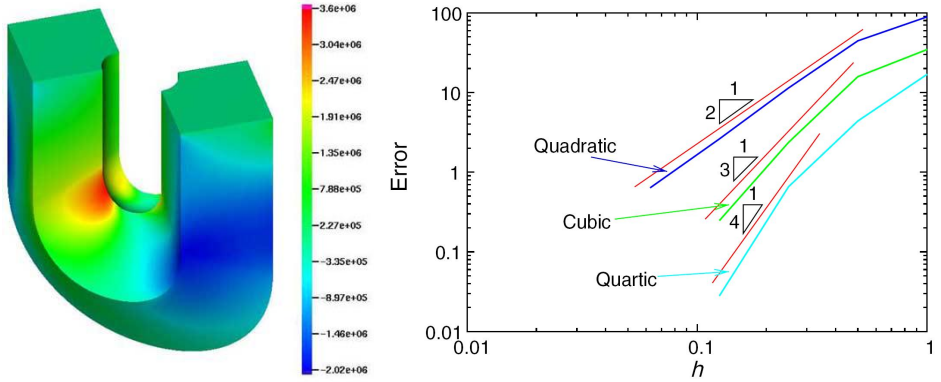
- Splines of degree p have up to $p-1$ continuous derivatives at element interfaces.

IsoGeometric Analysis based on B-splines have refinement procedures analogous to the h- and p-refinement in FEA, which are knot insertion and degree elevation, respectively. Interaction with the CAD-representation is not necessary because the geometric model from a CAD program is directly used as a coarse mesh within the analysis program. Despite the geometry being fixed at the coarsest level of discretization, the mesh and the corresponding basis, can then be refined and order elevated while maintaining the exact original geometry without interaction with the CAD system. An example that displays these properties for a linear elasticity problem using solid elements is shown in figure 1.2(a) to 1.2(d) with optimal convergence results shown in 1.3(b).



(a) (b) (c) (d)

Figure 1.2: (a) CAD geometric model of a horseshoe. (b) coarse mesh straight from CAD system. (c) and (d) model is refined for analysis without affecting geometry. From Hughes et al. [42].

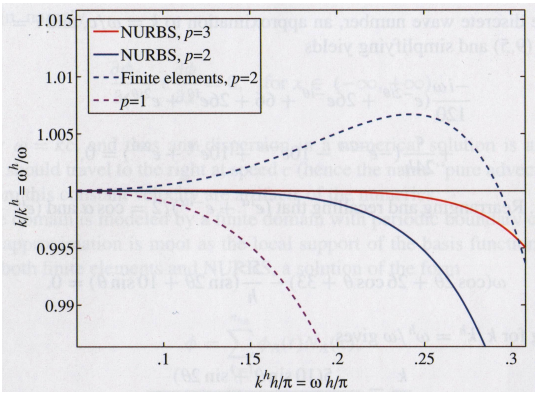


(a) (b)

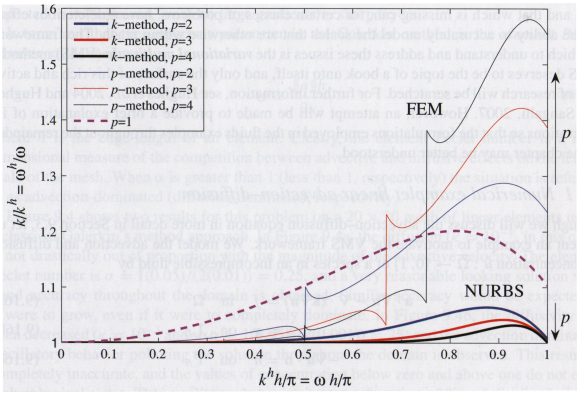
Figure 1.3: (a) Vertical stress component after analysis. (b) 'Spectral' like convergence. From Hughes et al. [42].

The property of splines of having a high level of derivative continuity at knot (element) interfaces gives rise to a new type of refinement, called k-refinement, where the degree is elevated together with the continuity at knots. This new type of refinement has proven to be potentially more powerful than p-refinement where C^0 continuity is maintained across knot interfaces while the degree is elevated.

Besides all practical advantages provided by the IGA framework, B-splines and consequently NURBS have proven to be outstanding basis functions for analysis. The increased continuity of the B-spline basis has resulted in a dramatic improvement of the spectral properties, as compared to C^0 finite elements. This is perhaps best illustrated in the one-dimensional test problems of pure convection and pure diffusion, where geometry does not play any role. The case of pure convection, the wave equation, is shown in Figure 1.4(a). We note that linear B-splines give identical results as linear FEM. Unfortunately the authors don't show any FEM results of higher order then quadratic. It is however clear that the B-spline basis performs much better than C^0 finite elements and converges with increasing order. Figure 1.4(b) depicts the comparison between C^0 FEM and C^{p-1} continuous B-splines for the heat equation, the case of pure diffusion. Observe that the results obtained using B-splines are more accurate throughout the entire spectrum and that the entire NURBS spectrum converges for all modes with increasing order using the new concept of k-refinement, whereas increasing the order of the finite element approximation with p-refinement results in divergence of the higher frequencies.



(a) Pure convection in the first order wave equation. Phase errors versus non-dimensional wave numbers. Comparison of linear and quadratic finite elements, C^1 quadratic and C^2 cubic B-splines.



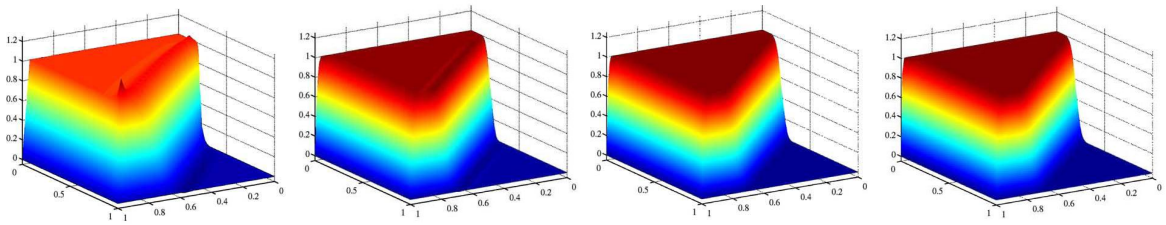
(b) Pure diffusion in the heat equation. Phase errors versus non-dimensional wave numbers. Comparison of classical C^0 -continuous finite elements and B-splines for $p = 1$ to 4.

Figure 1.4: Comparison of C^0 finite elements and C^{p-1} Isogeometric Analysis employing B-splines. From Cottrell et al. [22].

The superior spectral properties of the B-spline basis, as compared to C^0 finite elements, suggests that B-splines and NURBS might provide more accurate approximations to general fluid flow problems, which show a combination of convective and diffusive behavior. This might especially be apparent in applications such as wave propagation and turbulence where the entire discrete spectrum may contribute significantly in the solution [22]. This presumption

has been verified by Akkerman et al. [3] who have studied the role of continuity in turbulent flows.

A combination of convective and diffusive behavior is frequently studied using the linear convection-diffusion equation. This is a good starting point for more complicated non-linear systems like the Navier-Stokes equations, because many difficulties also arise in this simpler linear setting. When the ratio of convection/diffusion is large, sharp boundary layers arise which can be difficult to capture. The conventional Galerkin FEM is not able to resolve these sharp boundary layers and spurious oscillations may result. These oscillations are frequently suppressed by adding some artificial diffusion while attempting to minimize the negative impact on the solution. Generally these oscillations become more pronounced when the polynomial order is increased. However the variation diminishing property plus the concept of k-refinement illustrate the robustness of B-splines and NURBS based IsoGeometric Analysis when order is increased. Increased robustness is also confirmed by several researchers [14, 48]



(a) p=1 (b) p=3 (c) p=5 (d) p=8

Figure 1.5: Convection-Diffusion in two dimensions. Problem features sharp boundary layers skew to the mesh is $45\,\mathrm{deg}$. The mesh is uniform and spans 20 by 20 elements. The number of dofs is $(20+p)^2$ where p is the degree of the basis. Obtained results with increasing degree. From Hughes et al. [42].

when it comes to mesh distortion of IsoGeometric elements as compared to standard finite elements. In the study of Lipton et al. [48] it is further noted that robustness increases with order. This robustness makes them potentially attractive for many large deformation problems like free surface deformation and fluid structure interaction problems.

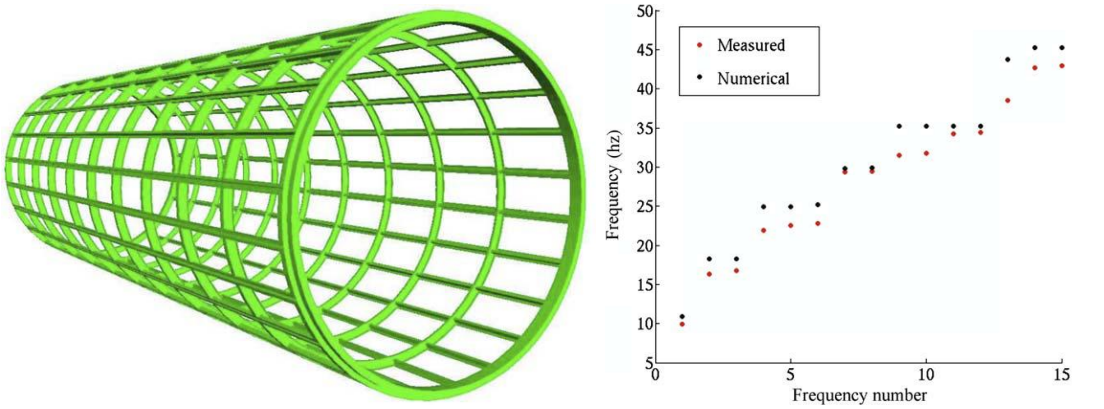
1.1.1 Practical application of IGA

Since its introduction, IGA has successfully been applied to a wide variety of problems in several fields of physics and engineering [42, 48, 14, 45, 23, 43, 3, 7, 8, 10, 11, 29, 30, 66, 18, 20, 19, 17, 21]. Development is ongoing and IsoGeometric Analysis has over the years reached a certain level of maturity. In the following we give a glimpse at some applications of IGA to real world problems in fluid dynamics and structural mechanics.

Structural vibrations of the NASA testbed cylinder

In Cottrell et al. [23] an example is depicted of IsoGeometric analysis applied to structural vibrations of a 'real world' geometry found in the aerospace industry: the NASA aluminum

testbed cylinder (ATC). This example demonstrates that the IsoGeometric concept can be applied on large-scale problems. Figure 1.6(a) shows the frame assembly of the ATC, where every geometric feature of the real geometry is exactly represented in the model. The mesh consisted of 228.936 rational quadratic elements and 2.219.184 degrees of freedom. As seen in Figure 1.6(b) quite satisfactory frequency results have been obtained compared to the experimental data.



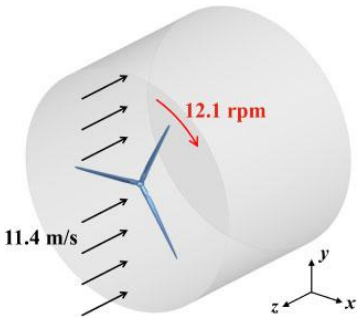
(a) IsoGeometric model of frame assembly.

(b) Comparison of numerical and experimental frequency results for the frame assembly

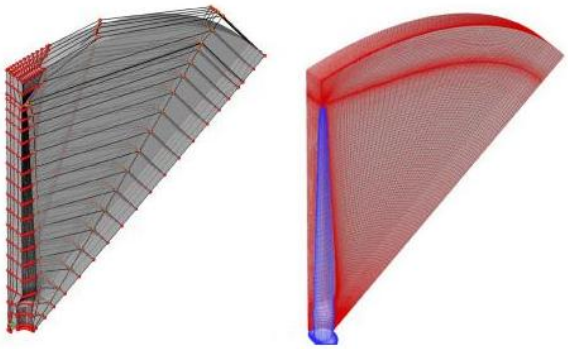
Figure 1.6: Every geometric feature of the NASA Aluminum testbed cylinder (ATC) is accounted for, Figure 1.6(a). The mesh consisted of 228.936 rational quadratic elements and 2.219.184 degrees of freedom. Figure 1.6(b) compares numerical results with experiments illustrating that IsoGeometric Analysis can be applied to large complex problems. From Cottrell et al. [23]).

Fluid structure interaction of a 5 MW wind turbine at full scale

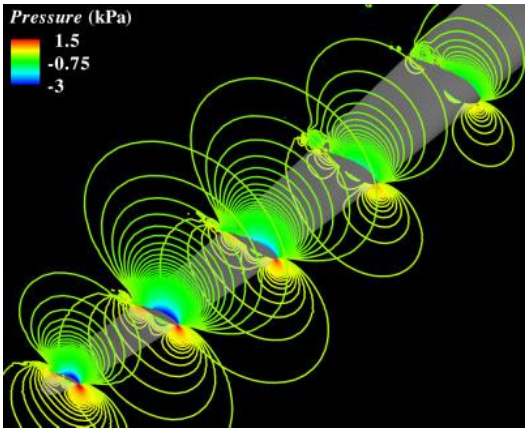
In Bazilevs et al. [12] and Bazilevs et al. [13] the interaction between airflow and structure is simulated around 5MW wind turbine blades at full scale. Every structural feature of the composite blades has been exactly accounted for in the fluid structure simulation. Advantage is taken from symmetry, such that only one third of the model needs to be taken into account. The structure and fluid domain is modeled using quadratic NURBS elements and the total number of degrees of freedom of the flow domain is approximately 1.5 million. The fluid formulation makes use of the residual-based variational multi scale method [41, 9] which is currently the most sophisticated Large Eddy Simulation (LES).



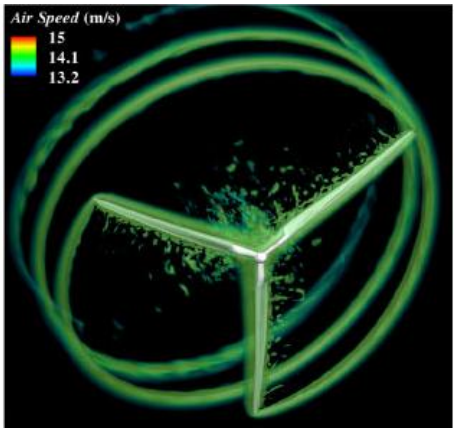
(a) Problem setup



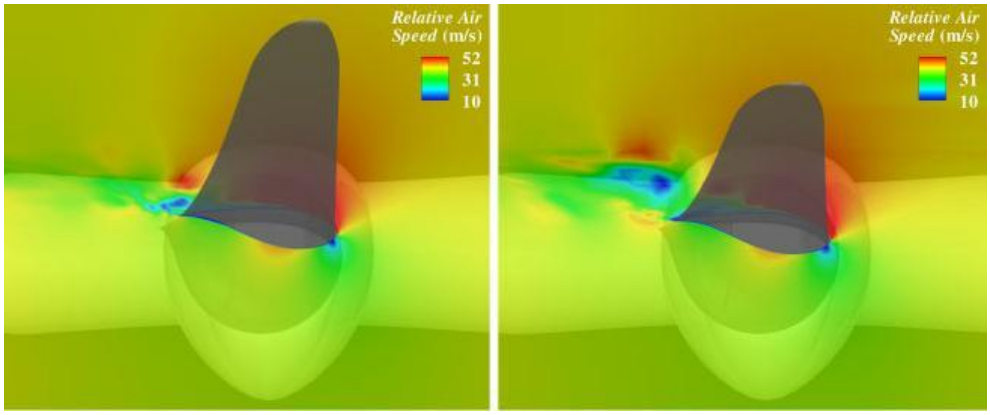
(b) NURBS modeling of flow domain



(c) Pressure contours at several blade cross sections



(d) Isosurfaces of air speed



(e) Isocontours of relative wind speed at a 30m radial cut at different time instances superposed on a moving blade.

Figure 1.7: (a) Problem setup and (b) construction of analysis suitable geometry of one third of the flow domain consisting of approximately 1.5 million degrees of freedom. From Bazilevs et al. [12]. (c) Pressure contours at several blade cross-sections viewed from the back of the blade plotted on the deformed configuration. The large negative pressure at the suction side of the airfoil creates a favorable aerodynamic torque. (d) Iso-surfaces of air speed. The flow exhibits complex behavior. The vortical feature generated at the blade tip is convected downstream of the rotor with very little decay. (e) Iso-contours of relative wind speed at a 30m radial cut at different time instances superposed on a moving blade. The flow stays fully attached on the pressure side and separates on the suction side. Note the deflection of the blades. From Bazilevs et al. [13]

Free surface phenomena for the dam break problem and the Wigley hull

In Akkerman et al. [4] two-phase air fluid simulations are performed using the level set approach. Level set methods represent the free surface interface implicitly through the introduction of a scalar field [44]. Both the air and fluid are modeled using the incompressible Navier-Stokes equations in the variational multi scale framework [41, 9].

A typical test case which can asses the performance for complex free surface phenomena is the dam brake problem. The problem setup is shown in Figure 1.8. Experiments for this test case were performed at the Maritime Research Institute Netherlands (MARIN), and the data is often used to validate free-surface software for marine engineering applications. The simulation is performed using two quadratic NURBS meshes, consisting of 32 x 16 x 16 (coarse) and 64 x 32 x 32 elements for the problem domain, and 2 x 6 x 3 and 4 x 12 x 6 elements for the object.

Figure 1.9 shows two snapshots for the two different meshes. In Figure 1.10 results for the pressure at two different points are compared with experiments from MARIN. The main trend is captured on both meshes. However, the fine mesh does not seem to perform much better than the coarse mesh, compared to the experimental results.

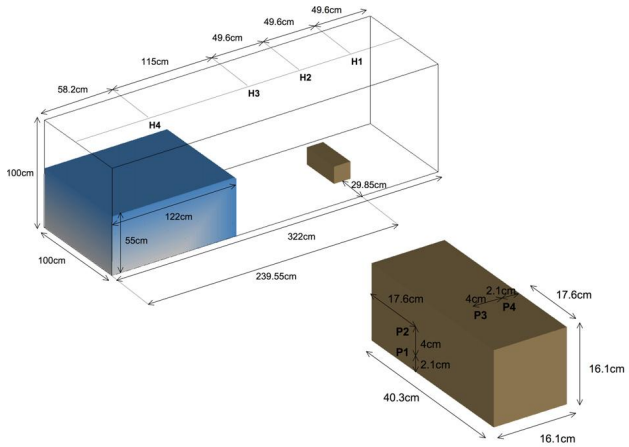


Figure 1.8: Problem setup dam break with obstacle problem . From Akkerman et al. [4]

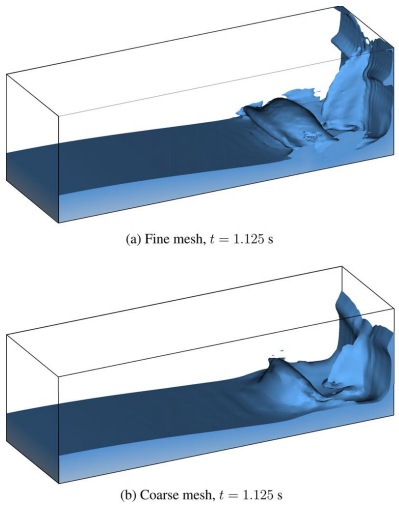


Figure 1.9: Comparison of the solution produced by the coarse and fine meshes.. From Akkerman et al. [4]

A typical test case to asses the performance of free surface calculations around ships is the quasi static flow around the Wigley Hull. The geometry is exactly modeled using quadratic NURBS elements on one half of the domain with symmetry boundary conditions. The problem mesh consists of 96 x 48 x 32 quadratic NURBS elements. Reasonable results are obtained, compared to the ITTC experiments, see Figure 1.11.

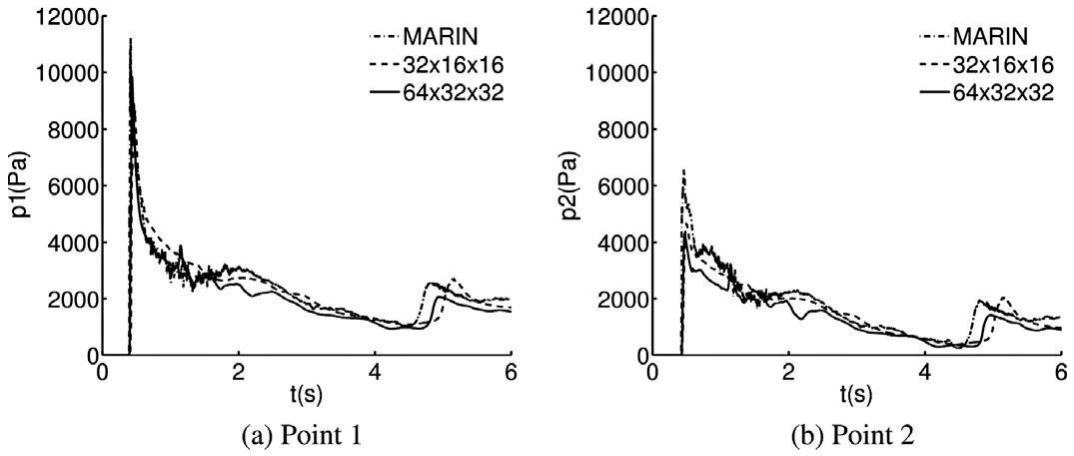
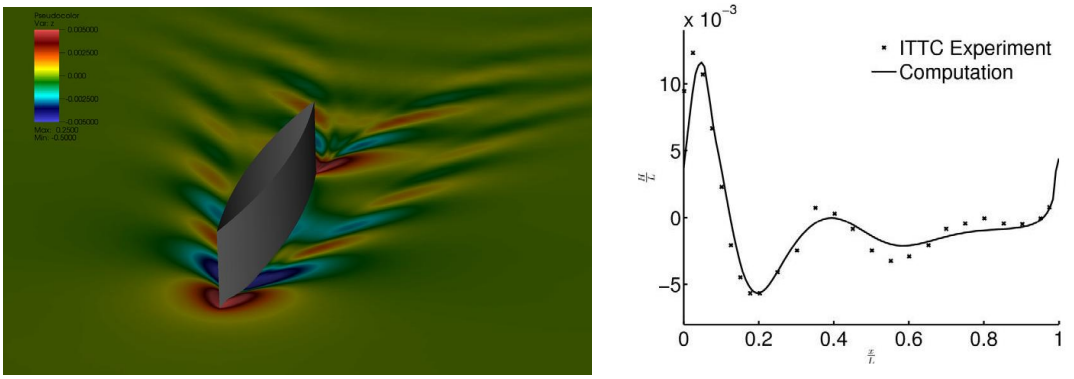


Figure 1.10: Dam break with obstacle. Pressure time history at four points on the object-comparison between experimental data (labeled MARIN) and computational results. The main trend is captured on both meshes. From Akkerman et al. [4]



(a) Wave pattern around the Wigley hull at Fr = 0.25.

(b) Free surface level along the length of the Wigley hull at Fr = 0.25. Comparison of numerical solution with ITTC experimental results

Figure 1.11: Free surface flow around a Wigley hull at Fr = 0.25. From Akkerman et al. [4]

1.2 Mimetic Discretization Methods

IsoGeometric Analysis seems to be an exciting new technology with a lot of potential in many fields of engineering. To summarize, it features many advantages over classical FEM: it provides seamless integration of CAD and analysis, making the cumbersome meshing-process redundant; exact geometry representation; increased robustness and accuracy; and finally making refining strategies become practically applicable. Important is to realize that these advantages are simply the result of the choice of basis functions!!

The discretization procedure of current IsoGeometric analysis methods, however, is based upon the finite element method and therefore also inherit its drawbacks. These drawbacks

are especially apparent in our main field of interest, being fluid dynamics. The most severe issues which IGA consequently shares with FEM in the field of fluid dynamics are:

- The finite element basis is incapable of generating a divergence free velocity field for incompressible flow.
- Stable basis for the velocity and pressure is only obtained under some stringent requirements called the inf/sup conditions.

These issues are unwanted artifacts which emerge during discretization. It appears that in the discretization process we lost the connection between physical quantities and the underlying geometry. In the derivation of physical laws of a physical theory, for instance, we usually start with analysis on geometrical objects such as a control volume and its bounding surfaces. These geometric features are lost however, when the resulting mathematical model is represented in the form as set of partial differential equations (PDE's). Popular methods, like the finite element method, take such a PDE as a starting point and consequently lose the connection between physical quantities and their related geometric elements, leading to instabilities and the inability to preserve conserved quantities.

In this thesis I present a new discretization procedure which displays all advantages of the IsoGeometric approach, yet naturally leads to stable consistent discretizations with strong conservation of the primal variables. The proposed method belongs to the class of Mimetic Discretization Methods, also known as Compatible Discretizations or Discrete Exterior Calculus (DEC). Mimetic Methods attempt to preserve as much as possible the structure of the physical problem at hand - i.e. to adequately mimic the properties of the continuous physical world in the discrete setting - such that fundamental symmetries and conserved quantities are preserved. A strong presumption exists that these methods naturally lead to consistent and stable approximations.

1.2.1 Preliminary classification of field quantities

If we take a closer look at the different physical quantities in a theory, we can observe that it's possible to structure them in roughly two distinct groups. In mechanical terms we have:

1. *configuration variables* which describe the state of a mechanical system, such as position, displacement, velocity, and acceleration
2. *source variables* which drive the state of the mechanical system, such as force, pressure, stress and electric charge

The relation among different configuration variables, or source variables respectively, is given by the physical laws. These relations are topological and do not contain any material relations. The connection between the configuration variables and sources variables is where the metric

dependent part comes in. These relations are given by the so called constitutive equations, which contain the material specific relations.

A preliminary classification of the different physical quantities in a physical theory can look as follows

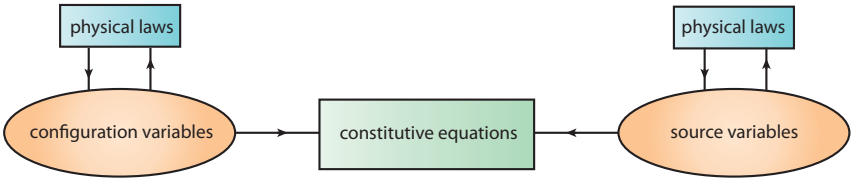


Figure 1.12: Preliminary classification of physical quantities in a theory

We shall see in Chapter 2, Section 2.3), that we can exactly represent the physical laws in discrete terms, by introducing the concepts of *chain* and *cochain* from the mathematical field of Algebraic Topology. The discretization of these relations is finite volume like, so we stay on familiar ground. Because the physical laws do not contain any metric they have the same form and value on topologically equivalent grids. The discretization of these relations is the same on a nice uniform Cartesian mesh and on a highly curved grid! Secondly these relations don't change on moving meshes, as long as the topology stays the same.

The constitutive equations - the metric and material dependent relations - is where the discrete approximation comes in, and requires a continuous representation of the field variables. First the mathematical model in the form of a PDE is reformulated using concepts from *Differential Geometry* - introduced in Chapter 2, Section 2.2- expressing the variables in terms of *differential forms*. This ensures that the connection of quantities with their related geometric entities is maintained in the continuous formulation.

In Chapter 4 the continuous and the discrete are connected by introducing B-spline basis functions which can reconstruct finite dimensional continuous representations of differential forms from discrete cochains. As we shall see B-splines provide a natural basis for Mimetic Methods.

Before we construct B-spline vector spaces of differential forms, we introduce them in the context of geometry and mesh generation, in Chapter 3. Here we will thoroughly examine the properties of B-splines and NURBS that make them attractive as a pre-analysis tool. During our discussion we will frequently make the comparison with FEA.

In the final chapter, we apply the theory developed in Chapter 2, 3 and 4 to the incompressible irrotational flow over lifting bodies. At first glance, these problems seem elementary. Conventional approaches like the FEM and FVM, are however not able solve these problems without resorting to an engineering approach. We use the so called Hodge decomposition and explicitly calculate the component of the flow which is responsible for a net circulation and associated lift force, see Figure 1.13. This procedure can only be performed using global relations. The FEM and FVM will consequently not be able to follow the same approach,

since they describe relations only locally.

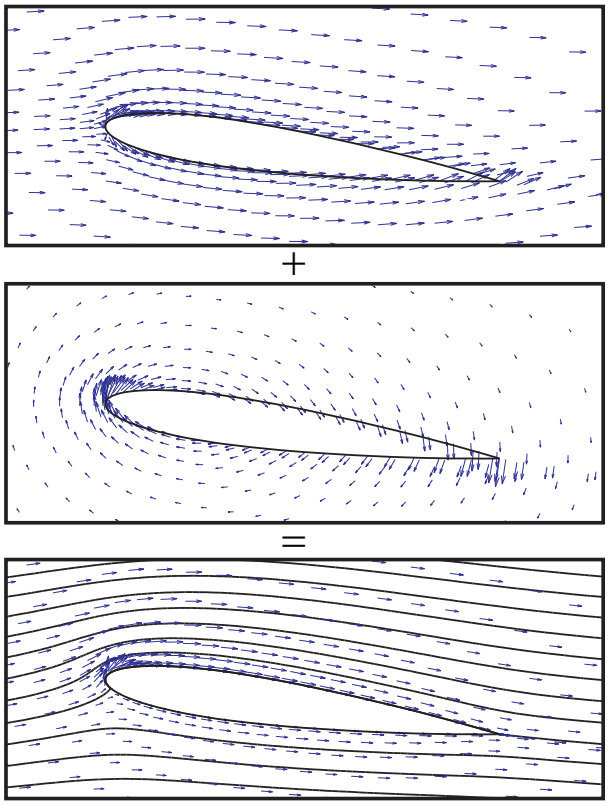


Figure 1.13: Hodge decomposition of an inviscid flow around a lifting airfoil. (Top) Contribution due to gradient of potential. (Middle) Contribution due to harmonic function. (Bottom) Resulting flow.

Chapter 2

Foundations of Mimetic Discretization Methods

Discretization is the process of transforming a continuous field problem - usually represented as a set of continuous partial differential equations (PDE's) - in to a set of discrete equations which can be handled by a computer. This process is always accompanied by a loss of information, since an infinite dimensional problem is reduced to a finite number of discrete equations.

Mimetic Discretization Methods, however, aim to preserve as much as possible the structure of the continuous field problem at hand. They attempt to adequately 'mimic' the properties of the continuous realm in the discrete setting, such that important symmetries and conserved quantities are preserved discretely.

2.1 Structure of physical problems

Understanding the structure of the physics behind the equations is in my opinion of vital importance if one wishes to successfully discretize a continuous field problem. Fortunately, many different physical phenomena, like for example electromagnetism, elasticity and fluid flow, all share a common structure. This can be explained by the common 'geometric background upon which the physics is build' [52]. Tonti [65] and later Mattiussi [52] studied the structure of physical phenomena extensively and they emphasize the following very important observations

A In every physical theory there are physical quantities which are naturally related to the most basic geometric and chronometric objects.

B The need to consider two kinds of orientation of the associated geometric objects: inner

or outer orientation.

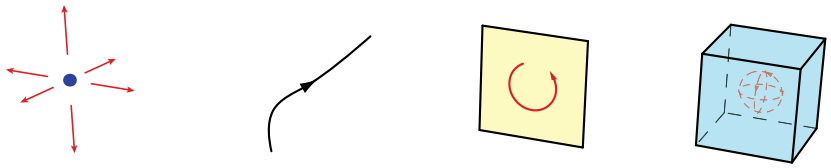
C Distinction between local (continuous) quantities and global (discrete) quantities, which are related by integration of field variables over a geometric object.

D The existence of a basic set of discrete physical conservation laws within each theory which state that a global physical quantity referring to a geometric object, is equal to another global physical quantity referring to its boundary.

2.1.1 Physical quantities, geometry and orientation

All physical quantities are naturally related to the most basic geometrical and chronometrical objects, like points, lines, surfaces, volumes, time instants and time intervals. In order to perform operations like summation and integration, the associated orientation of the geometric objects needs to be taken into account. We can distinguish between inner and outer oriented geometric objects. We shall discuss some examples, many of which are open for interpretation.

Charge of a particle, pressure, temperature, velocity potential and electric potential can be considered as scalar functions defined *in points* at a certain *time instant*; velocity of a fluid element is measured as the displacement *along a line segment* during a *time interval*; strain of a fiber is measured as the stretching *along a line* during a *time interval*; vorticity can be measured as the amount of rotation *in a plane* during a *time interval*.

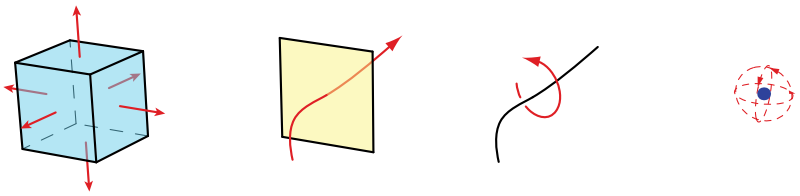


(a) in a point (b) along line (c) in a plane (d) inside a volume

Figure 2.1: Physical field quantities related to inner oriented geometric objects in $\mathbb{R}^3$

Similarly, there are field quantities which are naturally related to outer oriented geometrical objects (Figure 2.2): mass density, electric charge or energy can flow *out of a volume* during a *time interval*; energy, momentum and velocity fluxes are measured as the transfer *through a surface* during some *time interval*; angular momentum of a body is measured *around an axis of rotation* at a specific *time instant*; a vortex potential can be defined as the rotation *around a point* at a particular *time instant*.

We emphasize that most examples are open for discussion, for example vorticity could also be considered around an axis of rotation during a time interval (Biot-Savart).



(a) out of volume (b) through a plane (c) around a line (d) around a point

Figure 2.2: Physical field quantities related to outer oriented geometric objects in $\mathbb{R}^3$

2.1.2 Local verses global quantities

In the mathematical representation of physical quantities we make a distinction between local and global quantities. Local quantities represent a continuous distribution of some physical quantity. These are typically the mathematical variables used to describe physical phenomena in the form of a PDE.

The local variables, or field variables, are mathematical abstractions that can only be approximated by means of global, discrete variables. This approximation process is done by measuring. How we measure a local quantity in terms of global quantities is usually clear from its units. For example, velocity is measured as the distance in meters per second; density is measured as mass per unit of volume and pressure is measured as force per unit square meters. The operation that connects these local (continuous) quantities with the global (discrete) quantities is integration. Some examples are, integration of velocity v in time yields distance $\bar{s}$; integration of density ρ over a volume yields mass $\bar{m}$ and integration of the pressure p over a finite surface results in a discrete value of force $\bar{F}$.

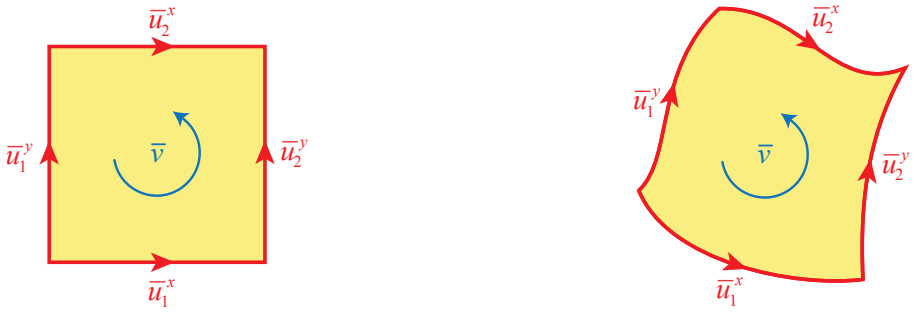
$$\bar{s} = \int_t v \, \mathrm{d}t, \qquad \bar{m} = \int_V \rho \, \mathrm{d}V \qquad \text{and} \qquad \bar{F} = \int_S p \, \mathrm{d}S,$$

where global quantities are denoted with a bar. Note that a local quantity is associated with an infinitesimal geometric object, while global quantities are related to finite dimensional analogues.

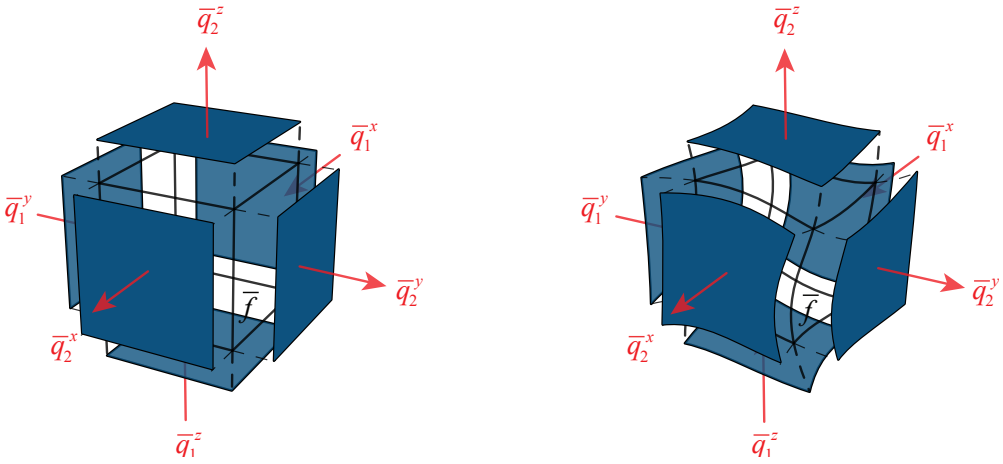
2.1.3 Topological relations

In every physical theory there are basic physical conservation laws which state that a global physical quantity referring to a geometric object is equal to another global quantity associated with its boundary. In $\mathbb{R}^3$ these equations are given by the well known *fundamental theorem of calculus*, relating the change of a quantity over a line segment as the difference between the boundary points, Figure 2.3(a); *Stokes theorem*, stating that the amount of rotation in a plane is equal to the circulation around its boundary, Figure 2.3(b); and finally the *divergence theorem*, relating the change of a global physical quantity in a volume as the difference between the in- and out-going fluxes, Figure 2.3(c).

(a) Fundamental theorem of calculus: $\bar{u} = \int_C \text{grad}\phi \; d\mathbf{c} = \bar{\phi}(b) - \bar{\phi}(a)$



(b) The Stokes theorem: $\bar{v} = \iint_S \text{curl}\, \mathbf{u} \cdot d\mathbf{S} = \int_C \mathbf{u} \; d\mathbf{c} = \bar{u}_1^x + \bar{u}_2^y - \bar{u}_2^x - \bar{u}_1^y$



(c) The divergence theorem: $\bar{f} = \iiint_V \text{div}\, \mathbf{q} \; dV = \iint_S \mathbf{q} \; d\mathbf{S} = \bar{q}_2^x - \bar{q}_1^x + \bar{q}_2^y - \bar{q}_1^y + \bar{q}_2^z - \bar{q}_1^z$

Figure 2.3: Fundamental theorem of calculus, Figure 2.3(a), relating global quantities associated with lines to global quantities at the boundary points. The Stokes theorem, Figure 2.3(b), relating global quantities associated with surfaces to global quantities along its boundary curves. The divergence theorem, Figure 2.3(c), relating global quantities associated with a volume to global quantities at the bounding surfaces. Observe that a change in geometry doesn't affect the discrete relations.

The physical laws are the same on topological equivalent grids, i.e. are the same on a highly curved grid as on a square Cartesian grid; that's why we refer to them as *topological relations*. Topological relations have an intrinsically discrete nature and can thus exactly be represented in terms of global quantities.

The variables in these equations are only related by summation, differentiation and integration. They do not involve any physical or material parameters. Figure 2.4 illustrates how the topological relations appear as the horizontal links between quantities associated with geometric objects of the same orientation.

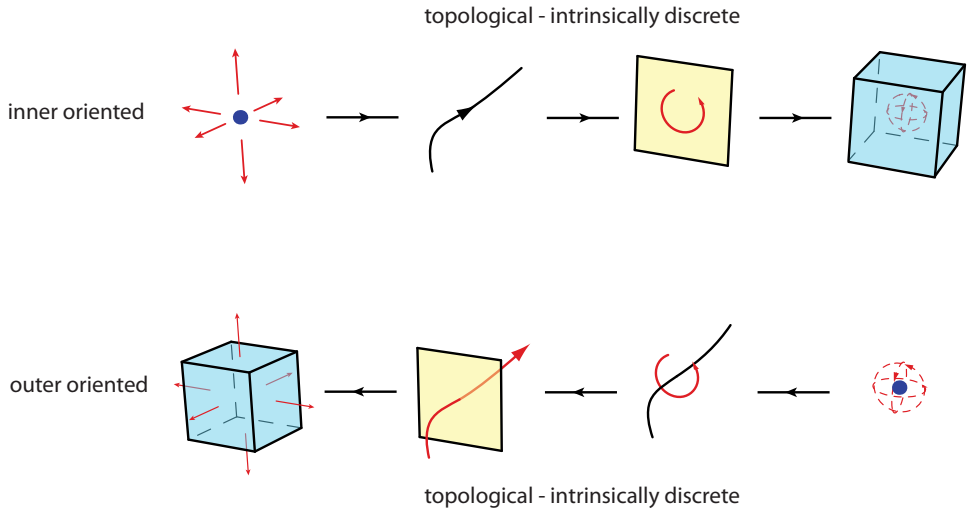


Figure 2.4: The topological relations form the horizontal connections in the diagram. These relations thus involve only objects of the same orientation.

2.1.4 Constitutive relations

The constitutive relations is where the metric (length, angle, etc) and material parameters start to play a role. Constitutive relations invoke a connection between dual geometric objects, see Figure 2.5. This operation can only be performed in continuous space and it is generally here where the discrete approximation comes in.

An example of a constitutive relation from the field of linear elasticity, linking stress σ_{ij} and strain ϵ_{kl} by the material stiffness tensor c_{ijkl}, is Hooke's law in three dimensional space

$$\sigma_{ij} = c_{ijkl} \cdot \epsilon_{kl}$$

The strain, the stretching of a fiber, is clearly associated with inner oriented line segments and the stress is a physical quantity related to outer oriented surfaces.

To model the constitutive relations, we require a continuous representation in terms of field variables. This is achieved using B-spline basis functions which reconstruct continuous representations from global discrete quantities (Chapter 4).

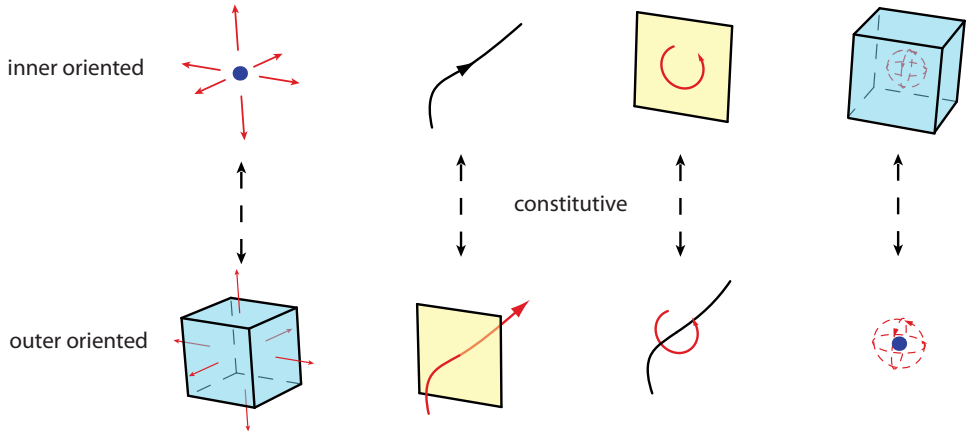


Figure 2.5: The constitutive relations appear as the vertical links between dual geometric objects

2.1.5 Classification of physical quantities

The association of physical quantities with geometric objects in space and time naturally leads to the classification of physical quantities within a physical theory, Figure 2.6.

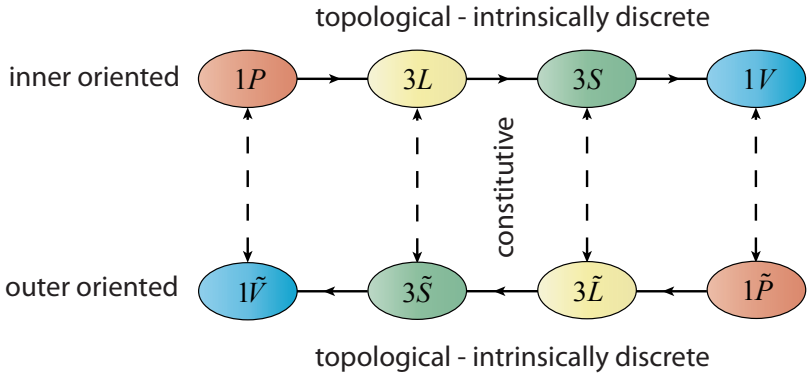


Figure 2.6: The Tonti classification diagram of physical quantities in three dimensional space. Each slot represents a physical quantity referred to a geometric object; that is scalar quantities living in points $1P$, vector valued quantities related to lines $3L$, vector valued quantities related to surfaces $3S$ and scalar valued quantities associated with volumes $1V$. The topological relations appear as the horizontal links between quantities associated with geometric objects of the same orientation, and have an intrinsically discrete nature. The constitutive relations appear as the vertical connections between quantities of opposed orientation. It is in general here where the discrete approximation comes in. Diagonal connections are also possible. These processes are, however, irreversible. From Mattiussi [52]

In this chapter we introduce the necessary mathematical tools upon which the mimetic framework is built. Section 2.2 starts with an introduction to *Differential Geometry*, which provides

us with a continuous mathematical representation of physical quantities in terms of exterior differential forms and the mathematical tools to represent the topological relations and the constitutive equations in continuous space. Differential forms allow us to rewrite a partial differential equation in a way that the connection with geometry is maintained in the mathematical formulation.

Section 2.3 introduces important concepts from Algebraic Topology, the discrete analogue of Differential Geometry. We shall see that we can rephrase the topological equations in discrete form which is exact and metric free.

Finally, in Section 2.4, we discuss how these two branches of mathematics are connected, by introducing the two basic operations of reduction and reconstruction. Local quantities can be reduced to yield global quantities and vice versa, global quantities can be used to reconstruct local quantities.

2.2 Continuous modeling - concepts from Differential Geometry

This section introduces geometry and related physical quantities in terms of the theory of exterior differential forms. For an extensive introduction into Differential Geometry the reader is referred to Flanders [33] and Frankel [34] .

2.2.1 Manifolds

In order to describe geometry and related physical quantities we need to take a closer look at the space in which we represent them. In the general case this space is curved and under some requirements such a space is called a *manifold*.

Definition 1 An n-dimensional manifold $\mathcal{M}^n$ in $\mathbb{R}^m$ is a parametric space which locally looks like $\mathbb{R}^n$. It is covered by a family of curvilinear coordinate patches $V_i \subset \mathbb{R}^m$,

$$\mathcal{M}^n = V_1 \cup V_2 \cup \ldots,$$

where each coordinate patch is the image of an embedding $U_i \subset \mathbb{R}^n$,

$$\psi_i(\mathbf{x}) \; : \; U_i(\mathbf{x}) \mapsto V_i(\mathbf{y})$$

The map ψ maps a point in the parameter space $\mathbf{x} \; \epsilon \; U$ to a point $\mathbf{y}$ on the manifold. It is required that ψ is a smooth one-to one map, meaning that the inverse map ψ^{-1} exist, and is continuously differentiable[1]. We further require that the parameters of overlapping coordinate patches $V_i \cap V_j$, can be written in terms of one-another

$$\mathbf{x}_{V_i} = F(\mathbf{x}_{V_j}) \quad \text{and vice versa} \quad \mathbf{x}_{V_j} = F^{-1}(\mathbf{x}_{V_i})$$

[1] In Differential Geometry this is called a diffeomorphism

The inverse map ψ^{-1} is called a chart, analogue to charts of the earth's surface. In fact, you can think of a manifold as being an n-dimensional world which is represented by an atlas consisting of a set of charts. Each chart maps a part of the world to a planar surface. From an observer on earth the earth's surface is indistinguishable from euclidean space $\mathbb{R}^2$, however the sphere differs from the plane on a larger scale.

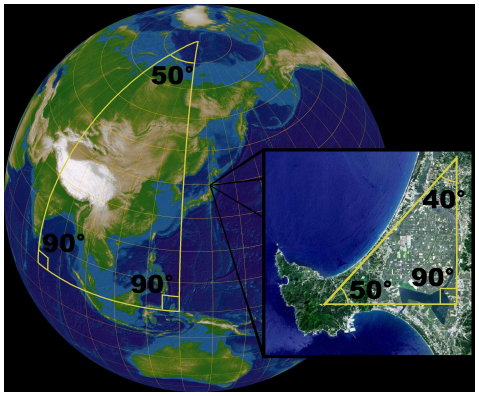


Figure 2.7: The earth (surface of a sphere) is a two-dimensional manifold in $\mathbb{R}^3$ since it can be represented by a collection of two-dimensional charts.

The straightforward examples of manifolds which we will encounter in $\mathbb{R}^3$ are smooth curves - Figure 2.8, smooth surfaces - Figure 2.9 - and smooth volumes. Manifolds, however, also arise in more abstract ways as solutions to differential equations or configuration spaces.

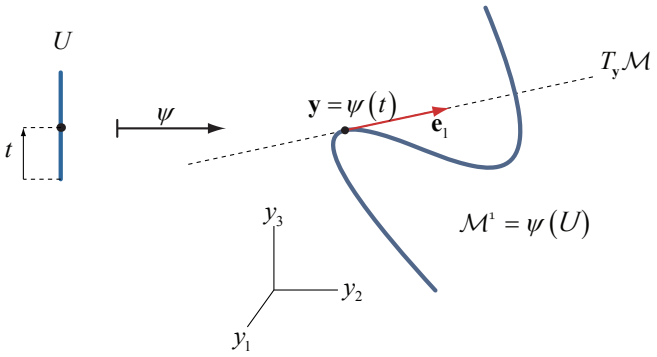


Figure 2.8: The tangent space of a 1-manifold in $\mathbb{R}^3$

An important consequence of the definition of a manifold is that it has a well-defined tangent space $T_{\mathbf{y}}\mathcal{M}^n$, which is a local copy of $\mathbb{R}^n$ containing all tangent vectors at a point $\mathbf{y} \in \mathcal{M}^n$. This fact allows us to apply the methods of differential and integral calculus and linear algebra to the study of manifolds.

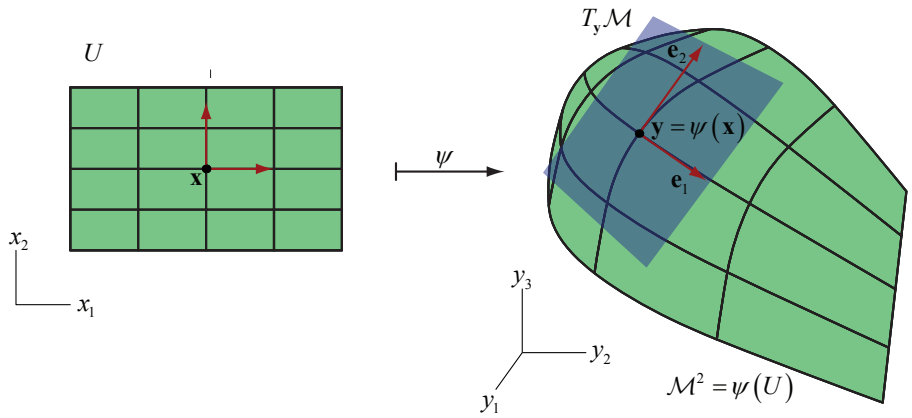


Figure 2.9: The tangent space of a 2-manifold in $\mathbb{R}^3$

The Tangent Space

A basis in which to represent the tangent vectors is easily obtained as follows. Consider the coordinate map from Figure 2.9. $\boldsymbol{\psi} : U(\mathbf{x}) \mapsto V(\mathbf{y})$, maps a point in the parameter space $\mathbf{x} \in U \subset \mathbb{R}^n$ to a point on the manifold $\mathbf{y} \in \mathcal{M}^n \subset \mathbb{R}^m$; thus maps m coordinates from n parameters,

$$\mathbf{y} = \boldsymbol{\psi}(\mathbf{x}) = \begin{cases} y^1 &= \psi^1\left(x^1, ..., x^n\right) \\ &\vdots \quad\quad \vdots \\ y^m &= \psi^m\left(x^1, ..., x^n\right) \end{cases}$$

where $n \leq m$. Observe that the vectors tangent to the coordinate lines $\mathbf{e}_i$ (obtained by keeping all but one parameter x^i constant), form a natural basis in which to represent tangent vectors. This basis, the collection $\mathbf{E} = (\mathbf{e}_1, ..., \mathbf{e}_n)$, is called the *primal basis* and can easily be obtained by use of the chain rule

$$\mathbf{e}_i = \frac{\partial}{\partial x^i} = \frac{\partial}{\partial y^j}\frac{\partial y^j}{\partial x^i} = \begin{pmatrix} 1 & & \oslash \\ & \ddots & \\ \oslash & & 1 \end{pmatrix} \begin{pmatrix} \frac{\partial \psi^1}{\partial x^i} \\ \vdots \\ \frac{\partial \psi^m}{\partial x^i} \end{pmatrix} = \frac{\partial \boldsymbol{\psi}}{\partial x^i} \tag{2.1}$$

Observe that the tangent space at $\mathbf{y}$ is the linear space spanned by the collection of the tangent basis vectors, i.e. the columns of the Jacobian matrix $D\boldsymbol{\psi}$, leading to the following definition

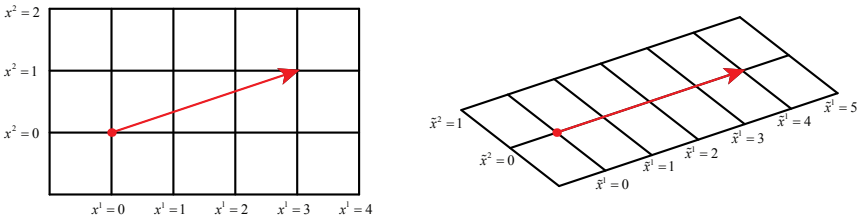
Definition 2 If $\mathcal{M}^n$ is an n-dimensional manifold, and $\mathbf{y} \in \mathcal{M}^n$, then the tangent space at $\mathbf{y}$ is the set $T_{\mathbf{y}}\mathcal{M}^n$, the span of all tangent vectors at $\mathbf{y}$ and is defined by

$$T_{\mathbf{y}}\mathcal{M}^n = \operatorname{col}\{D\boldsymbol{\psi}(\mathbf{x})\} \qquad \text{where} \quad D\boldsymbol{\psi}(\mathbf{x}) = (\mathbf{e}_1 \ \cdots \ \mathbf{e}_n) = \begin{pmatrix} \frac{\partial \psi^1}{\partial x^1} & \cdots & \frac{\partial \psi^1}{\partial x^n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \psi^m}{\partial x^1} & \cdots & \frac{\partial \psi^m}{\partial x^n} \end{pmatrix}$$

Connection between vectors and the primal basis

It is apparent that mathematical objects like points and vectors do not depend on the coordinate system in which we represent them; they are so called *invariant* under coordinate transformations.

EXAMPLE 2.2.1 We may write a vector in any coordinate system, where it is represented by a different set of coefficients and basis vectors, yet still describes the same object.



(a) $\boldsymbol{\alpha} = 3 \cdot \mathbf{e}_1 + 1 \cdot \mathbf{e}_2$ (b) $\boldsymbol{\alpha} = 4 \cdot \tilde{\mathbf{e}}_1 + 0 \cdot \tilde{\mathbf{e}}_2$

Figure 2.10: The same vector $\boldsymbol{\alpha}$ in two different coordinate systems

Consider the vector $\boldsymbol{\alpha}$ described in the primal basis[2]

$$\boldsymbol{\alpha} = \sum_{i=1}^{m} a^i \mathbf{e}_i = a^i \mathbf{e}_i$$

In order to see how the coefficients and basis change, we define new coordinates $\tilde{x}^j$ expressed in terms of the old coordinates x^i

$$\tilde{x}^j = \tilde{x}^j\left(x^1, \ldots, x^n\right) \quad \text{and corresponding basis} \quad \tilde{\mathbf{e}}_j = \frac{\partial}{\partial \tilde{x}^j} = \frac{\partial x^i}{\partial \tilde{x}^j}\frac{\partial}{\partial x^i}$$

Then we can write for the vector $\boldsymbol{\alpha}$

$$\boldsymbol{\alpha} \quad = \quad a^i \mathbf{e}_i \quad = \quad a^i \frac{\partial}{\partial x^i} \quad = \quad a^i \left(\frac{\partial}{\partial \tilde{x}^j}\frac{\partial \tilde{x}^j}{\partial x^i}\right) \quad = \quad \left(a^i \frac{\partial \tilde{x}^j}{\partial x^i}\right)\frac{\partial}{\partial \tilde{x}^j} \quad = \quad \tilde{a}^j \tilde{\mathbf{e}}_j$$

We can further deduce that the coefficients and basis vectors are related in the following way

$$\boldsymbol{\alpha} = \left(a^i \frac{\partial \tilde{x}^j}{\partial x^i}\right)\left(\frac{\partial x^k}{\partial \tilde{x}^j}\mathbf{e}_k\right) = a^i \left(\frac{\partial x^k}{\partial x^i}\right)\mathbf{e}_k = a^i \delta_i^k \mathbf{e}_k = a^i \mathbf{e}_i$$

We can conclude that a change of coordinates transforms the coefficients a^i in exactly the opposite way as the basis $\mathbf{e}_k$. The coefficients are called *contra variant* (denoted with superscript) and the primal basis is called *covariant* (denoted with a subscript).

[2]Note that in the latter expression we used the Einstein summation convention which implies that whenever an index appears twice, once as a subscript and once as a superscript, a summation needs to be performed over this index.

Connection between co-vectors and the dual basis

We might as well switch the roles of the coefficients and basis to obtain co-variant coefficients and a contra variant basis.

$$\boldsymbol{\alpha} = \left(a_k \frac{\partial x^k}{\partial \tilde{x}^j} \right) \left(\frac{\partial \tilde{x}^j}{\partial x^i} \mathbf{e}^i \right) = a_i \mathbf{e}^i$$

This mathematical object, similar to a tangent vector, is called a *co-vector* or *one-form*. Co-vectors live in the *co-tangent space*, usually called *dual space*. The contra variant basis is called the *dual basis*.

Definition 3 The dual basis $\tilde{\mathbf{E}}$ is the collection $\left(\mathbf{e}^1, ..., \mathbf{e}^n\right)$ and is defined by

$$\mathbf{e}_i \cdot \mathbf{e}^j = \delta_i^j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Figure 2.11 shows vector $\boldsymbol{\alpha}$ in terms of the primal basis $\boldsymbol{\alpha} = a^1\mathbf{e}_1 + a^2\mathbf{e}_2$ and in terms of the dual basis $\boldsymbol{\alpha} = a_1\mathbf{e}^1 + a_2\mathbf{e}^2$. The figure also illustrates the relation between the primal and dual basis vectors $\mathbf{e}_i \cdot \mathbf{e}^j = \delta_i^j$

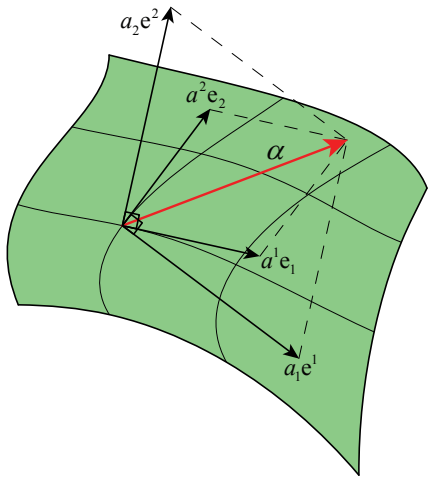


Figure 2.11: Vector $\boldsymbol{\alpha}$ in terms of the primal and dual basis

So far we know that the dual basis acts dual to the primal basis, but what do the dual basis vectors actually look like? The definition of the dual basis leads to the fact that $\mathbf{e}^j = dx^j$

$$\mathbf{e}_i \cdot \mathbf{e}^j = \delta_i^j \quad \Rightarrow \quad \frac{\partial}{\partial x^i} dx^j = \frac{\partial x^j}{\partial x^i} = \delta_i^j$$

The dx^j provides a basis for the co-tangent space and a co-tangent vector $\boldsymbol{\beta}$ can consequently be written as

$$\boldsymbol{\beta} = b_j \mathbf{e}^j = b_j \, dx^j$$

This is the more familiar notation of a 1-form. As can be concluded from the mathematical representation, 1-forms are the natural entities which 'live beneath a line integral' [33]. They can represent the same quantities as ordinary vectors, like velocity and acceleration, however, because of their mathematical representation can be readily integrated over line segments to yield global quantities. This is the main reason why to use differential forms instead of vectors.

2.2.2 The exterior product between forms

Thus far we are able to describe 0-forms - point wise defined scalar functions - and 1-forms - vector valued functions defined along line segments. To define their higher dimensional relatives, we introduce the exterior, or wedge product $\wedge$.

Definition 4 The exterior product $\wedge$ is defined by the following basic properties [33]

1. Antisymmetry: $dx^i \wedge dx^j = -dx^j \wedge dx^i$
2. Linear independence: $dx^i \wedge dx^i = 0$
3. Linearity: $\left(c \cdot dx^i\right) \wedge dx^j = dx^i \wedge \left(c \cdot dx^j\right) = c\left(dx^i \wedge dx^j\right)$ and $\left(dx^k + dx^l\right) \wedge dx^m = dx^k \wedge dx^m + dx^l \wedge dx^m$
4. Associativity: $\left(dx^k \wedge dx^l\right) \wedge dx^m = dx^k \wedge \left(dx^l \wedge dx^m\right)$

Property 1 induces an orientation to every k-form. Property 2 shows that when any two differential forms are linearly dependent, are in the same linear space, then their exterior product is zero. The third property illustrates that the exterior product only acts upon the basis, not the coefficients and it is linear in its arguments. Associativity, property number 4, means that the order in which the operations are performed does not matter as long as the order of the operands is not changed.

The exterior product between forms in $\mathbb{R}^3$

Consider the following three 1-forms $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, depicted in Figure 2.12(a) and 2.12(b)

$$\boldsymbol{\alpha} = a_1 dx^1 + a_2 dx^2 + a_3 dx^3, \quad \boldsymbol{\beta} = b_1 dx^1 + b_2 dx^2 + b_3 dx^3 \quad \text{and} \quad \boldsymbol{\gamma} = c_1 dx^1 + c_2 dx^2 + c_3 dx^3$$

The exterior product of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is the 2-form $\boldsymbol{\alpha} \wedge \boldsymbol{\beta}$, see Figure 2.12(a) and can be represented in terms of basis 2-co-vectors as

$$\begin{aligned}\boldsymbol{\alpha} \wedge \boldsymbol{\beta} &= \left(a_1 dx^1 + a_2 dx^2 + a_3 dx^3\right) \wedge \left(b_1 dx^1 + b_2 dx^2 + b_3 dx^3\right) \\ &= \left(a_2 b_3 - a_3 b_2\right) dx^2 \wedge dx^3 + \left(-a_1 b_3 + a_3 b_1\right) dx^3 \wedge dx^1 + \left(a_1 b_2 - a_2 b_1\right) dx^1 \wedge dx^2\end{aligned}$$

Note that the coefficients are those which would have appeared in the usual vector cross product of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The difference between the vector cross product and the exterior product

is that the exterior product doesn't require the linear space to possess a metric, in contrast to the vector cross product which naturally assumes a metric, since it is defined as the vector orthogonal to another vector, and orthogonality is a metric concept. Furthermore the wedge product is defined in $\mathbb{R}^n$ for all n, whereas the cross product only for $\mathbb{R}^3$.

(a) (b)

Figure 2.12: 2.12(a) The exterior product of two 1-forms $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ yields the 2-form $\boldsymbol{\alpha} \wedge \boldsymbol{\beta}$. 2.12(b) Multiplying by yet another 1-form $\boldsymbol{\gamma}$ give the 3-form $\boldsymbol{\alpha} \wedge \boldsymbol{\beta} \wedge \boldsymbol{\gamma}$

The exterior product between $\boldsymbol{\alpha} \wedge \boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ is the 3-form $\boldsymbol{\alpha} \wedge \boldsymbol{\beta} \wedge \boldsymbol{\gamma}$, Figure 2.12(b), and can be expanded in terms of the basis 3-form $dx^1 \wedge dx^2 \wedge dx^3$

$$\begin{aligned}\boldsymbol{\alpha} \wedge \boldsymbol{\beta} \wedge \boldsymbol{\gamma} &= \left(a_1 dx^1 + a_2 dx^2 + a^3 dx^3\right) \wedge \left(b_1 dx^1 + b_2 dx^2 + b_3 dx^3\right) \wedge \left(c_1 dx^1 + c_2 dx^2 + c_3 dx^3\right) \\ &= \left(a_1 b_2 c_3 - a_3 b_2 c_1 + a_2 b_3 c_1 + a_3 b_1 c_2 - a_2 b_3 c_2 - a_2 b_1 c_3\right) dx^1 \wedge dx^2 \wedge dx^3\end{aligned}$$

The single coefficient of a 3-form in $\mathbb{R}^3$ is the determinant of the coefficients of the original three 1-forms. In vector calculus this would be $(\mathbf{a} \times \mathbf{b}, \mathbf{c})$, representing the volume of the parallelepiped, spanned by the vectors **a** **b** and **c**.

The exterior product between forms in $\mathbb{R}^n$

In general the action of the wedge product is given by a single operation between a p-form and a q-form

$$\alpha^p \wedge \beta^q = (-1)^{pq}\, \beta^q \wedge \alpha^p \tag{2.2}$$

In particular for odd degree forms $\alpha^{2r+1} \wedge \alpha^{2r+1} = 0$. So

$$dx^i \wedge dx^j = -dx^j \wedge dx^i \qquad \text{and} \qquad dx^i \wedge dx^i = 0$$

which are the first two basic rules we introduced in definition 4.

2.2.3 Exterior Differential Forms

Exterior differential forms occur implicitly in all aspects of physics and engineering because they are the natural objects appearing as integrands of line, surface, and volume integrals.

The exterior product allows us to define a k-dimensional linear independent basis for k-forms. Let E^* be an n-dimensional vector space for 1-forms on an n-dimensional manifold Ω

$$E^* = \left\{dx^1, dx^2, ..., dx^n\right\} \tag{2.3}$$

Then a space for k-forms is obtained by exterior multiplication

$$\bigwedge^k(\Omega) = E^* \wedge E^* \wedge ... \wedge E^*, \quad \dim \bigwedge^k(\Omega) = \binom{n}{k} \tag{2.4}$$

The dimension of the space for k-forms is equal to the binomial coefficient which gives the number of ways, disregarding order, in which k objects can be chosen among n objects. An object from this space, a k-form, can generally be represented as

$$\alpha^{(k)} = \sum_I f_I(\mathbf{x}) dx^I \qquad \text{where} \qquad dx^I = dx^{i_1} \wedge dx^{i_2} \wedge ..... \wedge dx^{i_k} \tag{2.5}$$

Here the coefficients of α, $f_I(\mathbf{x})$, are smooth functions, which denote the spatial distribution of some quantity, and dx^I refers to the geometric object it is associated with. As can be expected from the notation, k-forms can be readily integrated over a k-dimensional manifold.

Differential forms in $\mathbb{R}^3$

In three dimensional space, $\mathbb{R}^3$, we have 0-forms which are related to points, 1-forms to line segments, 2-forms to surfaces and 3-forms which are associated with volumes. To define these mathematical quantities we distinguish the following continuous spaces and corresponding basis

$\bigwedge^0(\Omega)$ - H_P - the space of 0-forms,

$\bigwedge^1(\Omega)$ - H_L - the space of 1-forms, spanned by the basis 1-forms dx^1, dx^2 and dx^3.

$\bigwedge^2(\Omega)$ - H_S - the space of 2-forms, spanned by the basis 2-forms $dx^2 \wedge dx^3$, $dx^3 \wedge dx^1$ and $dx^1 \wedge dx^2$

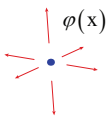
$\bigwedge^3(\Omega)$ - H_V - the space of 3-forms, spanned by the basis 3-form $dx^1 \wedge dx^2 \wedge dx^3$

Some examples of differential forms in $\mathbb{R}^3$ are

0-form

A 0-form is simply a scalar function related to points - e.g. the velocity potential of a perfect fluid.

$$\varphi^0 = \varphi(\mathbf{x})$$

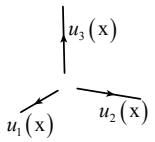


where $\mathbf{x} = \left(x^1, x^2, x^3\right)$ are general curved coordinates.

1-form

A 1-form is a smooth function that lives on line-segments - e.g. the velocity of a fluid element along a line segment.

$$u^1 = u_1(\mathbf{x})dx^1 + u_2(\mathbf{x})dx^2 + u_3(\mathbf{x})dx^3$$

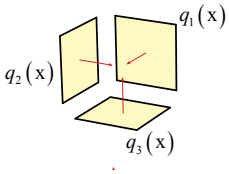


You can think of the dx^i as infinitesimal increment in the general curved coordinate x^i.

2-form

A 2-form dwells on surfaces - e.g. fluid flux through a surface.

$$q^2 = q_1(\mathbf{x})dx^2 \wedge dx^3 + q_2(\mathbf{x})dx^3 \wedge dx^1 + q_3(\mathbf{x})dx^1 \wedge dx^2$$

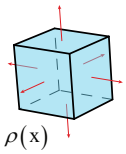


You can think of the $dx^i \wedge dx^j$ as a surface element with sides dx^i and dx^j.

3-form

Finally we have a 3-form, a smooth function related to a volume element - e.g. the specific mass (density) of a fluid-element.

$$\rho^3 = \rho(\mathbf{x})\ dx^1 \wedge dx^2 \wedge dx^3$$



$dx^1 \wedge dx^2 \wedge dx^3$ can be regarded as an infinitesimal, possibly curved, volume element with sides in the three basis directions.

Primal and dual complex of differential forms in $\mathbb{R}^3$

Following the reasoning from Section 2.1 we can distinguish between inner and outer oriented differential forms. Figure 2.13 illustrates the different spaces for differential forms in $\mathbb{R}^3$, where the outer oriented spaces are denoted with a tilde.

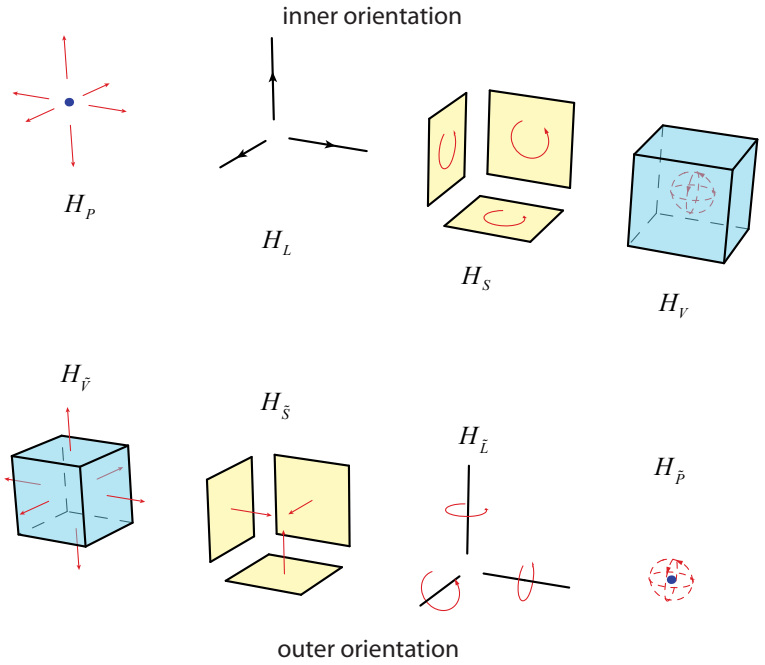


Figure 2.13: Inner and outer oriented spaces for exterior differential forms in $\mathbb{R}^3$

We shall see that the different spaces are horizontally connected by the topological relations, for which we shall introduce a single operator in Section 2.2.4, the exterior derivative d. The vertical links are given by the constitutive equations, which are modeled by the Hodge- (dual) star $*$. This operator contains all the metric properties and is treated in Section 2.2.5.

2.2.4 The exterior derivative d

Just like ordinary continuous smooth functions, differential forms can be differentiated. When a k-form is differentiated, a $(k+1)$-form is obtained. It is not more difficult to compute d in a curvilinear coordinate system then in a Cartesian one.

Differential of a 0-form

Consider the 0-form $\varphi^0 = \varphi(\mathbf{x})$, then its differential is a 1-form, and is given in $\mathbb{R}^3$ by:

$$\mathrm{d}\varphi^0 = \left(\frac{\partial\varphi}{\partial x^1}\right) dx^1 + \left(\frac{\partial\varphi}{\partial x^2}\right) dx^2 + \left(\frac{\partial\varphi}{\partial x^3}\right) dx^3$$

Observe that in Cartesian coordinates these coefficients are the components of the gradient of $\varphi(\mathbf{x})$

Exterior derivative of a 1-form

A 1-form in general coordinates is given by:

$$u^1 = u_1(\mathbf{x})dx^1 + u_2(\mathbf{x})dx^2 + u_3(\mathbf{x})dx^3$$

then applying the exterior derivative d yields

$$\begin{aligned}
\mathrm{d}u^1 =& \mathrm{d}u_1(\mathbf{x}) \wedge dx^1 + \mathrm{d}u_2(\mathbf{x}) \wedge dx^2 + \mathrm{d}u_3(\mathbf{x}) \wedge dx^3 \\
=& \left(\left(\frac{\partial u_1}{\partial x^1} \right) dx^1 + \left(\frac{\partial u_1}{\partial x^2} \right) dx^2 + \left(\frac{\partial u_1}{\partial x^3} \right) dx^3 \right) \wedge dx^1 \\
&+ \left(\left(\frac{\partial u_2}{\partial x^1} \right) dx^1 + \left(\frac{\partial a_2}{\partial x^2} \right) dx^2 + \left(\frac{\partial u_2}{\partial x^3} \right) dx^3 \right) \wedge dx^2 \\
&+ \left(\left(\frac{\partial a_3}{\partial x^1} \right) dx^1 + \left(\frac{\partial a_3}{\partial x^2} \right) dx^2 + \left(\frac{\partial u_3}{\partial x^3} \right) dx^3 \right) \wedge dx^3 \\
=& \left(\frac{\partial u_3}{\partial x^2} - \frac{\partial u_2}{\partial x^3} \right) dx^2 \wedge dx^3 + \left(\frac{\partial u_1}{\partial x^3} - \frac{\partial u_3}{\partial x^1} \right) dx^3 \wedge dx^1 + \left(\frac{\partial u_2}{\partial x^1} - \frac{\partial u_1}{\partial x^2} \right) dx^1 \wedge dx^2
\end{aligned}$$

In Cartesian coordinates these coefficients are the components of the curl.

Exterior derivative of a 2-form

Finally, for a 2-form $q^2 = q_1(\mathbf{x})dx^2 \wedge dx^3 + q_2(\mathbf{x})dx^3 \wedge dx^1 + q_3(\mathbf{x})dx^1 \wedge dx^2$ we obtain after differentiation:

$$\begin{aligned}
\mathrm{d}q^2 &= \mathrm{d}q_1(\mathbf{x}) \wedge dx^2 \wedge dx^3 + \mathrm{d}q_2(\mathbf{x}) \wedge dx^3 \wedge dx^1 + \mathrm{d}q_3(\mathbf{x}) \wedge dx^1 \wedge dx^2 \\
&= \left(\frac{\partial q_1}{\partial x^1} + \frac{\partial q_2}{\partial x^2} + \frac{\partial q_3}{\partial x^3} \right) dx^1 \wedge dx^2 \wedge dx^3
\end{aligned}$$

whose single component is given in Cartesian coordinates by the divergence.

Important is to realize that we have a single operator, the exterior derivative d, which depending on the k-form on which it acts, encodes the gradient, the curl or the divergence.

De Rahm complex

The examples above show that the exterior derivative d is a generalization of the grad, curl and div operators and that

$$\text{grad} \ : \ H_P \to H_L, \qquad \text{curl} \ : \ H_L \to H_S \qquad \text{and} \qquad \text{div} \ : \ H_S \to H_V, \tag{2.6}$$

The $(k+1)$-form obtained after applying the exterior derivative is called an exact form. Applying the exterior derivative to an exact form $\mathrm{d}\alpha^k$ leads to the associated null space

$$\mathrm{d}(\mathrm{d}\alpha^k) = \mathrm{d}^2\alpha^k = 0^{k+2}, \tag{2.7}$$

which in terms of the differential operators of vector calculus is written as

$$\text{curl} \cdot \text{grad} = 0 \qquad \text{and} \qquad \text{div} \cdot \text{curl} = 0 \tag{2.8}$$

This gives rise an exact sequence, called De Rahm complex, which can be set up for both the complex of inner oriented differential forms and the complex of outer oriented differential forms

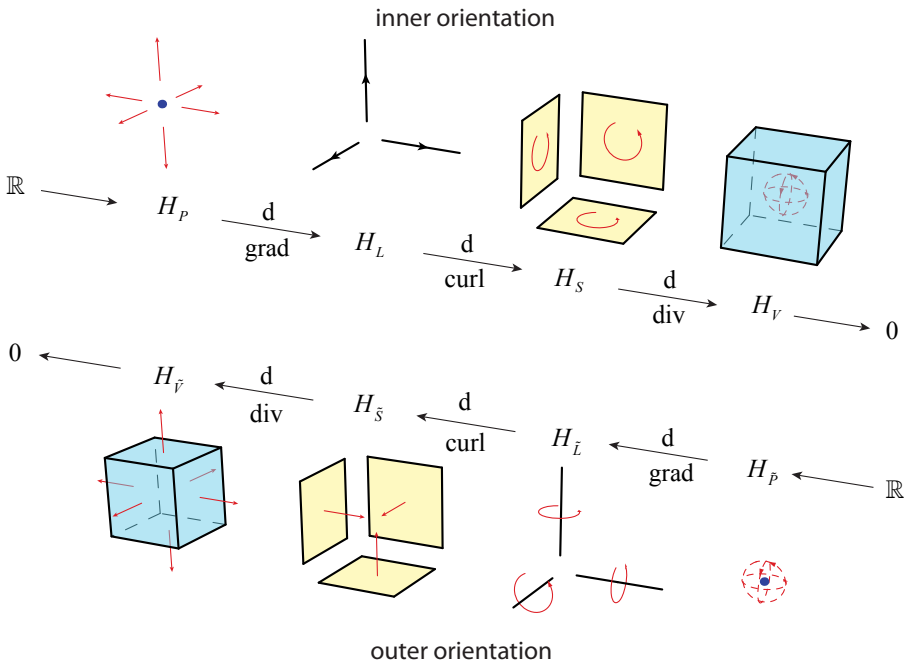


Figure 2.14: Double De Rahm sequence for respectively the inner oriented and outer oriented differential forms in $\mathbb{R}^3$

2.2.5 The Hodge star operator $*$

Since the gradient maps 0-forms to 1-forms and the divergence maps 2-forms to 3-forms, we can observe that it's not directly possible to apply the Laplacian $\Delta = \text{div grad}$ to a 0-form.

$$\text{grad} \ : \ H_P \to H_L \qquad \text{and} \qquad \text{div} \ : \ H_S \to H_V,$$

To establish a link between the gradient and the divergence operator, we introduce the Hodge star operator - $*$ - which assigns quantities to new 'dual' geometric objects. In fact, the Hodge maps k-forms to (n-k)-forms of opposite orientation. Some examples of the action of the Hodge in $\mathbb{R}^3$ are

$$*\,1 = dy^1 \wedge dy^2 \wedge dy^3 \qquad \text{and} \qquad *\,dy^1 \wedge dy^2 \wedge dy^3 = 1$$

$$* dy^1 = dy^2 \wedge dy^3, \qquad * dy^2 = dy^3 \wedge dy^1 \qquad \text{and} \qquad * dy^1 = dy^2 \wedge dy^3$$

The Hodge star operator establishes a connection between the primal complex and the dual complex. This mathematical operator encodes the metric dependency as part of the constitutive relations in a physical theory.

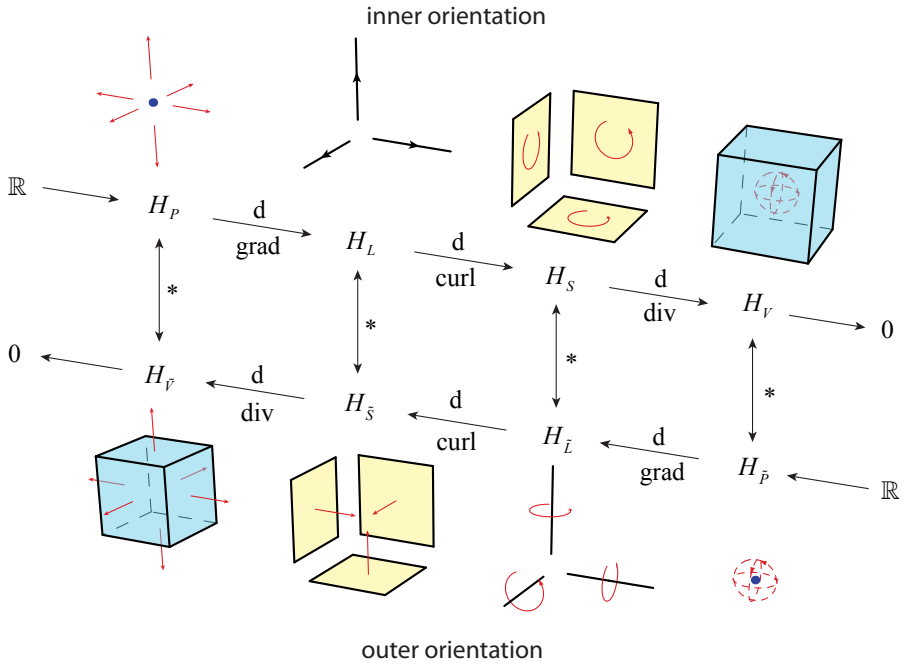


Figure 2.15: The Hodge star operator $*$ establishes the vertical links between dual differential forms of opposite orientation.

The metric is contained in the following inner product, which defines the Hodge

$$\alpha^k \wedge *\beta^k = \left\langle \alpha^k, \beta^k \right\rangle \mathrm{d}\Omega \tag{2.9}$$

The inner product between 0-forms is simply the inner product between two scalar functions. More interesting is the inner product between 1-forms. Consider the following 1-forms u^1 and w^1 in $\mathbb{R}^2$,

$$u^1(\mathbf{x}) = u_1\, dx^1 + u_2\, dx^2 \qquad \text{and} \qquad w^1(\mathbf{x}) = w_1\, dx^1 + w_2\, dx^2 \tag{2.10}$$

written in terms of it's vector components in local coordinates x^i. The Hodge star inner product, in equation 4.51, can consequently be written in terms of local coordinates as,

$$w^1 \wedge *u^1 = \begin{pmatrix} w_1 & w_2 \end{pmatrix} \begin{pmatrix} \langle dx^1, dx^1 \rangle & \langle dx^1, dx^2 \rangle \\ \langle dx^2, dx^1 \rangle & \langle dx^2, dx^2 \rangle \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \mathrm{d}\Omega \tag{2.11}$$

The matrix containing the inner products $\langle dx^i, dx^j \rangle$ is the covariant version of the metric tensor, and is often denoted by g^{ij}. We can calculate these components, given a parametric map $\mathbf{y} = \Phi(\mathbf{x})$, stating physical coordinates y^i in terms of local coordinates x^j. We will illustrate how to do this in Chapter 4, when we apply B-spline basis functions in mimetic discretizations.

The co-derivative d^*

While the exterior derivative maps a k-form to a $(k+1)$-form, the co-derivative maps a k-form to a $(k-1)$-form

$$\mathrm{d} : \bigwedge^{(k)}(\Omega) \mapsto \bigwedge^{(k+1)}(\Omega) \qquad \text{and} \qquad \mathrm{d}^* : \bigwedge^{(k)}(\Omega) \mapsto \bigwedge^{(k-1)}(\Omega) \tag{2.12}$$

The exterior derivative is purely topological. The co-derivative, however, contains a metric part and can be written in terms of the Hodge and the exterior derivative as

$$\mathrm{d}^* = (-1)^{n(k+1)+1} * \mathrm{d} * \tag{2.13}$$

Note that from $\mathrm{dd} = 0$ and $** = \pm 1$ we have

$$\mathrm{d}^*\mathrm{d}^* = \pm(*\mathrm{d}*)(*\mathrm{d}*) = 0 \tag{2.14}$$

The metric is given by an inner product, which illustrates that the co-derivative d^* is the Hilbert adjoint operator of the exterior derivative d

$$\left\langle \mathrm{d}\varphi^{(0)}, u^{(1)} \right\rangle = \left\langle \varphi^{(0)}, \mathrm{d}^* u^{(1)} \right\rangle \tag{2.15}$$

The Laplacian Δ

The Laplacian is a very important operator in physics and also within this thesis. The Laplacian acting on a differential k-form can be written in terms of the exterior derivative d and the co-derivative d^*. Since $\mathrm{d}^*\mathrm{d}^* = \mathrm{d}\,\mathrm{d} = 0$, we can write Δ as

$$\Delta = (\mathrm{d}\,\mathrm{d}^* + \mathrm{d}^*\mathrm{d})^2 = \mathrm{d}^*\mathrm{d} + \mathrm{d}\,\mathrm{d}^* \tag{2.16}$$

In many physical problems the Laplacian is applied to a 0-form. Because $\mathrm{d}^*\varphi^{(0)} = 0$ (check this in the De Rahm sequence in Figure 2.15), the Laplacian acting on a 0-form can be written as

$$\Delta\, \varphi^{(0)} = (\mathrm{d}^*\mathrm{d} + \mathrm{d}\,\mathrm{d}^*)\, \varphi^{(0)} = \mathrm{d}^*\mathrm{d}\, \varphi^{(0)} \tag{2.17}$$

2.2.6 Pulling back forms

This section shows that the integration and differentiation of differential forms are operations that can be entirely done without any notion of metric. By the use of a parametric map we can differentiate and integrate differential forms in a rectangular domain and subsequentally make a transformation to the real curvilinear domain. This makes calculation using differential forms in curved domains as easy as calculation in rectangular domains.

Suppose α^k is a k-form defined on an m-dimensional sub-manifold Ω of $\mathbb{R}^n$, where the coordinates are denoted by $\mathbf{Y}\left(y^1, y^2, ..., y^n\right)$:

$$\alpha^{(k)} = \sum_I f_I(\mathbf{y}) dy^I$$

We can substitute for 'new' coordinates $\mathbf{X}\left(x^1, x^2, ..., x^m\right)$ by restating the old variables in terms of the new ones by functions:

$$\begin{aligned} y^1 &= \phi_1\left(x^1, x^2, ..., x^m\right) \\ &\vdots \\ y^n &= \phi_n\left(x^1, x^2, ..., x^m\right) \end{aligned}$$

In short notation we can write $\mathbf{y} = \phi(\mathbf{x})$, where $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \; ... \; \phi_n(\mathbf{x})]^T$. The functions ϕ_i are assumed to be smooth and defined on a common domain Ω', an open subset of $\mathbb{R}^n$ (diffeomorphism).

Applying the pull back ϕ^* to α^k basically comes down to applying a change of variable. We substitute the 'old' coordinate variables $\mathbf{y}$ with $\phi(\mathbf{x})$ and subsequently use the 'new' variables $\mathbf{x}$ as a linear basis. This means that

$$\phi^*\alpha^k = \sum_I \left(\phi^* f_I(\mathbf{y})\right)\left(\phi^* dy^I\right) = f_I\left(\phi(\mathbf{x})\right) d\phi^I \tag{2.18}$$

Observe that the pullback operation turns k-forms on the target domain Ω into k-forms on the parametric domain. Thus while $\phi \; : \; \Omega' \mapsto \Omega$,

$$\phi^* \; : \bigwedge^k(\Omega) \mapsto \bigwedge^k(\Omega') \tag{2.19}$$

The pullback operation is nicely compatible with exterior differentiation and integration of forms. This is because the pull back operator ϕ^* commutes with respect to the exterior derivative d and the wedge product $\wedge$, i.e.

$$\phi^* \mathrm{d}\alpha^k = \mathrm{d}\phi^*\alpha^k \qquad \text{and} \qquad \phi^*(\alpha^k \wedge \beta^l) = (\phi^*\alpha^k) \wedge (\phi^*\beta^l) \tag{2.20}$$

The proof in case of a 0-form $f^{(0)}$ looks as follows

$$\phi^* \mathrm{d}f^{(0)} = \phi^* \left(\frac{\partial f^{(0)}}{\partial y^i} dy^i \right) = \frac{\partial \phi^* f^{(0)}}{\partial \phi^i} d\phi^i = \frac{\partial \phi^* f^{(0)}}{\partial \phi^i} \frac{\partial \phi^i}{\partial x^j} dx^j = \frac{\partial \phi^* f^{(0)}}{\partial x^j} dx^j = \mathrm{d}\phi^* f^{(0)}$$

The fact that the pull back commutes with the exterior derivative and the wedge product is why integration of differential forms in curved space is no more difficult as in Cartesian space. Consider the pull back of the 1-form in example 2.2.2

EXAMPLE 2.2.2 Suppose $\alpha = a_1(\mathbf{y})dy^1 + a_2(\mathbf{y})dy^2$ is a 1-form on the two-dimensional manifold Ω in $\mathbb{R}^2$, where the euclidean coordinates are denoted by $\mathbf{Y}\left(y^1, y^2\right)$. Substituting for 'new' coordinates $\mathbf{X}\left(x^1, x^2\right)$

$$\begin{aligned} y^1 &= \Phi_1\left(x^1, x^2\right) \\ y^2 &= \Phi_2\left(x^1, x^2\right) \end{aligned}$$

$\mathbf{y} = \Phi(\mathbf{x})$ can be regarded as a parametric map from domain Ω' to Ω

$$\Phi \ : \ \Omega' \mapsto \Omega$$

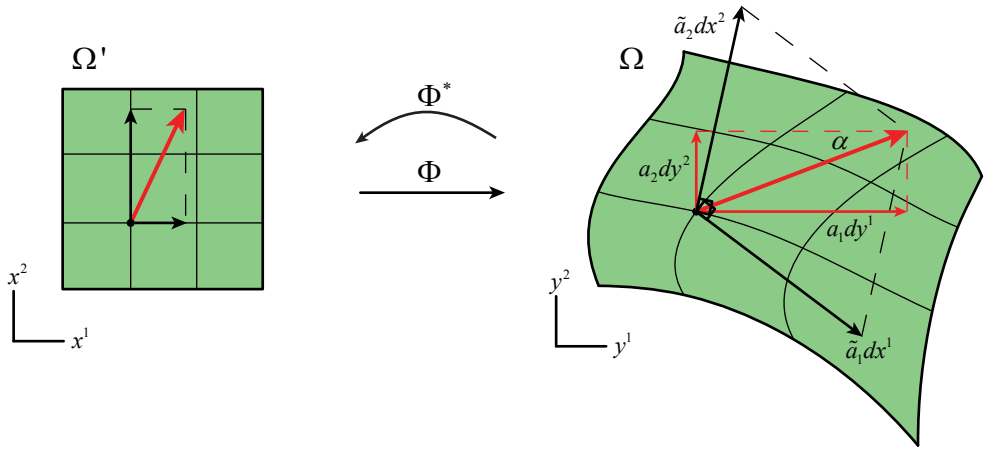


Figure 2.16: Pulling back 1-form α from physical space Ω to the reference domain Ω'.

The basis co-vectors change in the following sense

$$dy^1 = \frac{\partial y^1}{\partial x^1} dx^1 + \frac{\partial y^1}{\partial x^2} dx^2 \qquad \text{and} \qquad dy^2 = \frac{\partial y^2}{\partial x^1} dx^1 + \frac{\partial y^2}{\partial x^2} dx^2$$

In terms of the parametric map Φ this transformation looks as follows

$$\begin{pmatrix} dy^1 \\ dy^2 \end{pmatrix} = \begin{pmatrix} \frac{\partial \Phi^1}{\partial x^1} & \frac{\partial \Phi^1}{\partial x^2} \\ \frac{\partial \Phi^2}{\partial x^1} & \frac{\partial \Phi^2}{\partial x^2} \end{pmatrix} \begin{pmatrix} dx^1 \\ dx^2 \end{pmatrix}$$

The 1-form α can be expressed in the new coordinate system as

$$\alpha = (a_1(\mathbf{y})\ a_2(\mathbf{y})) \begin{pmatrix} \frac{\partial \Phi^1}{\partial x^1} & \frac{\partial \Phi^1}{\partial x^2} \\ \frac{\partial \Phi^2}{\partial x^1} & \frac{\partial \Phi^2}{\partial x^2} \end{pmatrix} \begin{pmatrix} dx^1 \\ dx^2 \end{pmatrix} = \tilde{a}_1(\mathbf{y})dx^1 + \tilde{a}_2(\mathbf{y})dy^2$$

Since α is now expressed in terms of the dx^1 and dx^2, we can perform topological operations like differentiation and integration in the reference domain.

Now let us try to find an explicit expression for the pullback of forms that we are going to encounter in $\mathbb{R}^3$. Suppose $\alpha^{(k)}$ is a k-form defined on a sub manifold Ω of $\mathbb{R}^3$, where the coordinates are denoted by $\mathbf{Y}\left(y^1, y^2, y^3\right)$. The pullback $\phi^*\alpha^k$ is a k-form in the reference domain defined on an open subset U also in $\mathbb{R}^3$ with coordinates $\mathbf{X}\left(x^1, x^2, x^3\right)$.

Pullback of 0-forms in $\mathbb{R}^3$

The pullback of the 0-form $\alpha^{(0)} = f(\mathbf{y})$ in $\mathbb{R}^3$, the pullback of a single function, has already been shown to be

$$\phi^*\alpha^{(0)} = f(\phi(\mathbf{x})) \tag{2.21}$$

Pullback of 1-forms in $\mathbb{R}^3$

The pullback of the 1-form $\alpha^1 = \sum_{i=1}^{3} f_i(\mathbf{y})dy^i$ in $\mathbb{R}^3$ is

$$\phi^*\alpha^1 = \sum_{j=1}^{3}\sum_{i=1}^{3} f_i(\phi(\mathbf{x}))\frac{\partial \phi^i}{\partial x^j}dx^j \tag{2.22}$$

Pullback of 2-forms in $\mathbb{R}^3$

For a 2-form

$$\alpha^2 = \sum_{1\le i<j\le 3} f_{ij}(\mathbf{y})dy^i \wedge dy^j \tag{2.23}$$

the pullback is given by:

$$\phi^*\alpha^2 = \sum_{1\le k<l\le n}\ \sum_{1\le i<j\le m} f_{ij}\left(\phi(\mathbf{x})\right) \begin{vmatrix} \frac{\partial \phi^i}{\partial x^k} & \frac{\partial \phi^i}{\partial x^l} \\ \frac{\partial \phi^j}{\partial x^k} & \frac{\partial \phi^j}{\partial x^l} \end{vmatrix} dx^k \wedge dx^l \tag{2.24}$$

Here the geometric significance of determinants becomes clear. They act as a kind of volume change factor on the parametric space. Higher degree differential forms act similar when the pullback operator is applied.

Pullback of 3-forms in $\mathbb{R}^3$

The result of the pullback on the 3-form $\alpha^{(3)} = f(\mathbf{y})dy^1 \wedge dy^2 \wedge dy^3$ in 3-dimensional space is:

$$\phi^*\alpha^{(3)} = f(\phi(\mathbf{x}))\begin{vmatrix} \frac{\partial\phi^1}{\partial x^1} & \frac{\partial\phi^1}{\partial x^2} & \frac{\partial\phi^1}{\partial x^3} \\ \frac{\partial\phi^2}{\partial x^1} & \frac{\partial\phi^2}{\partial x^2} & \frac{\partial\phi^2}{\partial x^3} \\ \frac{\partial\phi^3}{\partial x^1} & \frac{\partial\phi^3}{\partial x^2} & \frac{\partial\phi^3}{\partial x^3} \end{vmatrix} dx^1 \wedge dx^2 \wedge dx^3 \tag{2.25}$$

2.2.7 Integrating forms

Exterior differential forms are the natural mathematical objects appearing as integrands over k-dimensional geometric manifolds. Integration is thus straightforward. Taking orientation into account we can pull back a k-form and perform the integration in the reference domain

$$\int_{\Omega} \alpha^{(k)} = \int_{\Omega'} \phi^*\alpha^{(k)} \tag{2.26}$$

The topological relations can be stated in discrete form by the generalized Stokes theorem. It contains as special cases the classical integration definitions of vector calculus, namely the gradient, Gauss and Stokes theorem.

Theorem 2.2.1 Generalized Stokes Theorem. Let α be a $(k-1)$-form on a sub manifold Ω^k of $\mathbb{R}^n$ with boundary $\partial\Omega^k$, then

$$\int_{\Omega^k} d\alpha^{(k-1)} = \int_{\partial\Omega^k} \alpha^{(k-1)} \tag{2.27}$$

Usually the domain under consideration, Ω^k and $\partial\Omega^k$, is partitioned into sub domains (cells), we can collect in a k-chain. Performing integration over this k-chain provides us a collection of discrete global quantities, we can similarly collect in a so called k-cochain. The concept of chain and cochain are fundamental to Algebraic Topology, the discrete counterpart of Differential Geometry. In fact, cochains are the discrete analogue of differential forms. In the next section we shall study some important concepts from Algebraic Topology which provides us with the tools to set up a discrete analogue to the De Rahm sequences for both inner and outer oriented differential forms.

2.3 Discrete modeling - concepts from Algebraic Topology

In this section we introduce discrete geometry by means of *Algebraic Topology*. Topology describes the relations between different geometric objects, however without the notion of distance or measure.

Because topological relations do not contain any metric they have the same form and value on topological equivalent grids. The discretization of these relations is the same on a nice uniform Cartesian mesh and on a highly curved grid, see Figure 2.17! Secondly these relations don't change on moving meshes, as long as the topology doesn't change. In this section we shall work in topological space only; most examples will be given for the 2-dimensional case, since these are more illustrative.

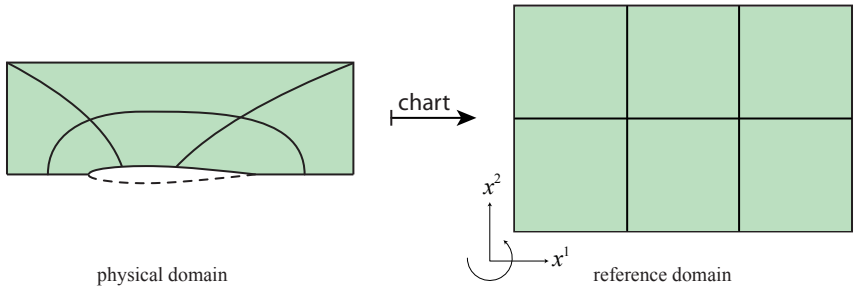


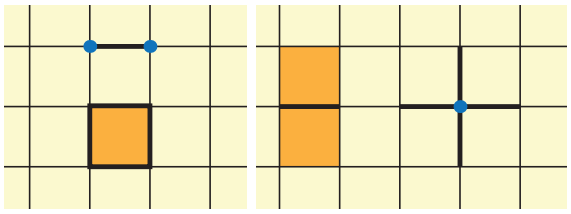
Figure 2.17: Topologically equivalent grids

2.3.1 Cell complexes

The geometric domain under consideration can be subdivided into a series of discrete sub domains, which in the context of Algebraic Topology are called cells. In three dimensional space we make distinction between

- 0-cells - representing vertices (points);
- 1-cells - edges (lines);
- 2-cells - faces (surfaces);
- 3-cells - representing volumes:

0-cells are logically connected to 1-cells, 1-cells to 2-cells and similarly 2-cells are connected to 3-cells. If a k-cell is the boundary of a $(k+1)$-cell, we shall call them the *face* of the $(k+1)$-cell, see Figure 2.18(a). The opposite connection - when a $(k+1)$-cell is connected to a k-cell - the $(k+1)$-cell is called the *coface* of the k-cell, illustrated in Figure 2.18(b).



(a) Faces (b) Cofaces

Figure 2.18: Faces and cofaces in $\mathbb{R}^2$. Figure (a) depicts a 1-cell and a 2-cell and their accompanying faces, the 0-cells and 1-cells at their boundary. Figure (b) illustrates the inverse relation. The co-faces of a 1-cell are its neighboring 2-cells and the co-faces of a 0-cell are its neighboring 1-cells.

Definition 5 In $\mathbb{R}^n$, the collection of all oriented 0, 1,... ,n-cells is called the *cell complex* K

Figure 2.19 depicts a cell complex in two-dimensional space. In practice a cell complex is simply the computational mesh in topological space.

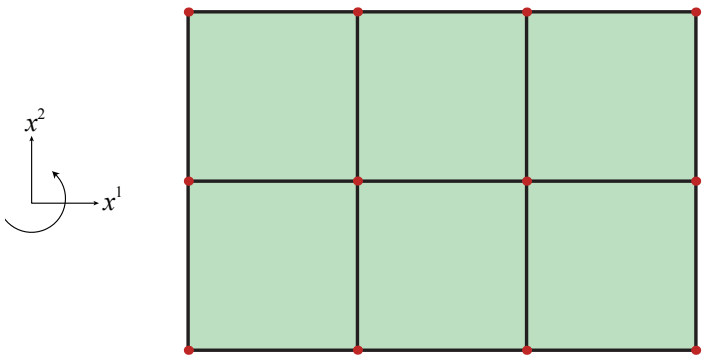
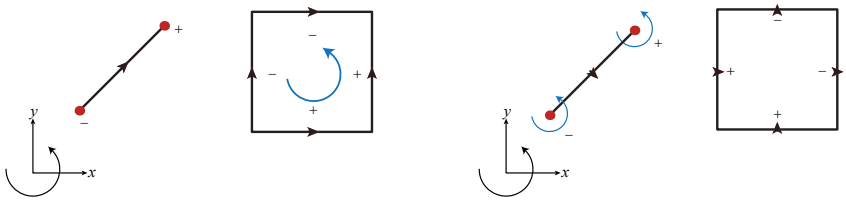


Figure 2.19: The two dimensional mesh in Figure 2.17 can for example be divided into 0-,1- and 2-cells. The collection of all these k-cells forms the cell complex, which in practical terms is simply our primal grid in topological space.

In order to assign quantities to the different k-cells in a cell-complex and to perform operations on them, we need to number and orient all cells inside the complex. The numbering can be performed in any arbitrary fashion, but the orientation needs to be chosen. Analogous to continuous space where we made distinction between differential forms associated to inner or outer oriented infinitesimal geometric objects, discrete k-cells are induced with either an inner or outer orientation. Figure 2.20 illustrates the two types of orientations for geometric objects in 2-dimensional space.



(a) Inner orientation (b) Outer orientation

Figure 2.20: Inner and outer oriented geometric objects in $\mathbb{R}^2$, showing a '+' sign for compatible directions and '-' sign for incompatible directions between a k-cell and its boundary $(k-1)$-cell.

Since we make the distinction between inner and outer oriented geometric objects, we can choose to place all inner oriented objects on a primal complex and all outer oriented objects on a dual complex, or vice versa.

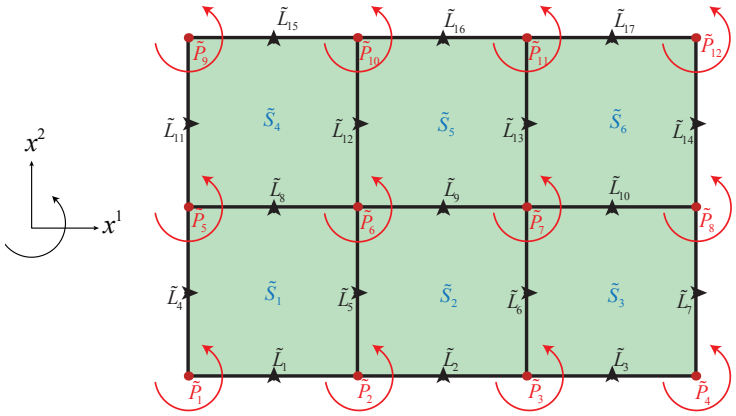


Figure 2.21: A numbered, outer oriented 2-dimensional primal cell complex. In the numbering of the different cells we make the distinction between 0-cells assigned by $\tilde{P}$, 1-cells by $\tilde{L}$ and 2-cells by $\tilde{S}$, where the tilde refers to the outer orientation. The numbering is arbitrary

If the outer oriented objects are placed on the primal cell complex, the inner oriented objects should be placed on the dual cell complex. The dual cell complex associates to every k-cell on the primal complex an (n-k)-cell of opposite orientation. The number assigned to a (n-k)-cell on the dual cell-complex is chosen the same as the connecting k-cell on the primal complex.

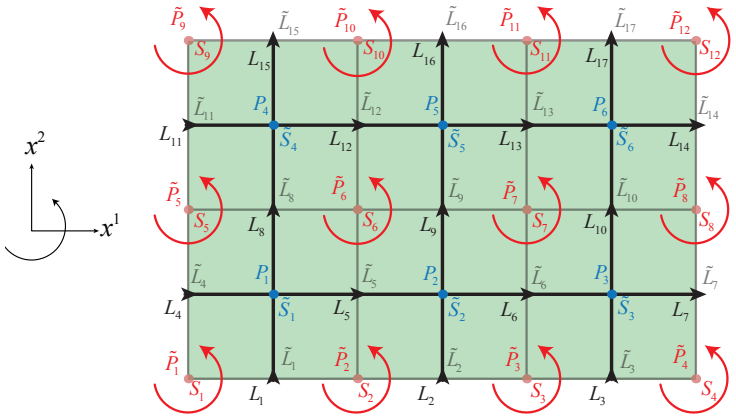


Figure 2.22: The dual cell complex in $\mathbb{R}^2$. Inner oriented 0-cells P_i are associated with outer oriented 2-cells $\tilde{S}_i$ on the primal complex; inner oriented 1-cells L_i act dual to outer oriented 1-cells $\tilde{L}_i$ on the primal cell complex; and finally inner oriented 2-cells S_i on the dual complex act dual to inner oriented 0-cells $\tilde{P}_i$ on the primal complex.

2.3.2 Chains and the boundary operator ∂

Following the numbering, we can collect all k-cells inside the cell complex K to form a k-chain, denoted by $c^{(k)}$. A k-chain is formal linear combination of k-cells,

$$c^{(k)} = \left\{ s_1 c_1^{(k)}, s_2 c_2^{(k)}, \ldots, s_{p-1} c_{p-1}^{(k)}, s_p c_p^{(k)} \right\} \tag{2.28}$$

where s_i is either $+/-1$, denoting whether the k-cell is induced with a positive or negative sense of direction with respect to the coordinate axes, the rules from Figure 2.20.

The different k-chains inside the cell complex are connected by means of the boundary operator, denoted as ∂. Applying the boundary operator to a chain returns its boundary $(k-1)$-chain,

$$\partial \quad : \quad c^p \mapsto c^{p-1} \tag{2.29}$$

So for a two dimensional cell complex we have $c^{(2)} \xrightarrow{\partial} c^{(1)} \xrightarrow{\partial} c^{(0)}$. Applying the boundary operator twice leads to the null space; i.e. the boundary of the boundary is empty,

$$\partial \partial c^p = 0 \tag{2.30}$$

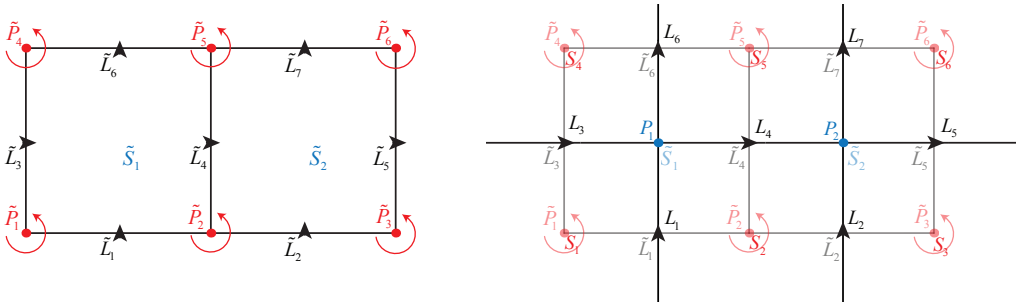
This is illustrated in Example 2.3.1 and Figure 2.24.

EXAMPLE 2.3.1 Consider the numbered and oriented primal and dual cell complexes in Figure 2.23. We can collect all outer oriented 0-,1- and 2-chains on the primal complex in corresponding $\tilde{k}$-chains. All $\tilde{k}$-cells have been given positive direction with respect to the coordinate axes, so

$$c^{(\tilde{0})} = \left\{\tilde{P}_1, \tilde{P}_2, \tilde{P}_3, \tilde{P}_4, \tilde{P}_5, \tilde{P}_6\right\}, \quad c^{(\tilde{1})} = \left\{\tilde{L}_1, \tilde{L}_2, \tilde{L}_3, \tilde{L}_4, \tilde{L}_5, \tilde{L}_6, \tilde{L}_7\right\} \quad \text{and} \quad c^{(\tilde{2})} = \left\{\tilde{S}_1, \tilde{S}_2\right\}$$

Similarly we can collect all inner oriented k-cells from the dual complex in k-chains

$$c^{(0)} = \{P_1, P_1\} \quad c^{(1)} = \{L_1, L_2, L_3, L_4, L_5, L_6, L_7\} \quad \text{and} \quad c^{(2)} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$$

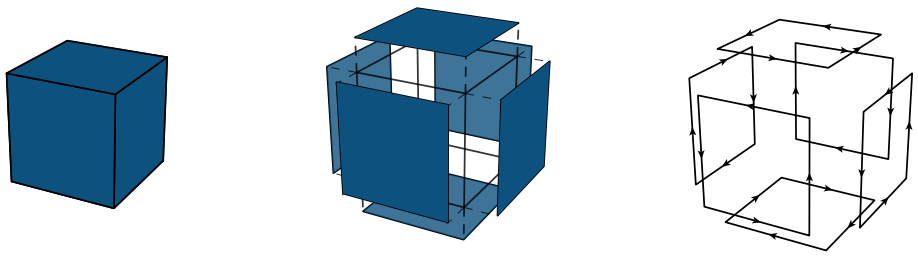


(a) Primal cell complex (b) Dual cell complex

Figure 2.23: A numbered and positively oriented two dimensional cell complex

Taking the boundary of the outer oriented 2-chain twice leads to an empty 0-chain.

$$\begin{aligned}
c^{(\tilde{2})} &= \tilde{S}_1 + \tilde{S}_2 \\
\partial c^{(\tilde{2})} &= \partial\tilde{S}_1 + \partial\tilde{S}_2 = \left\{\tilde{L}_1 + \tilde{L}_3 - \tilde{L}_4 - \tilde{L}_6\right\} + \left\{\tilde{L}_2 + \tilde{L}_4 - \tilde{L}_5 - \tilde{L}_7\right\} \\
\partial\partial c^{(\tilde{2})} &= \partial\tilde{L}_1 + \partial\tilde{L}_3 - \partial\tilde{L}_6 + \partial\tilde{L}_2 - \partial\tilde{L}_5 - \partial\tilde{L}_7 \\
&= \left\{\tilde{P}_1 - \tilde{P}_2\right\} + \left\{\tilde{P}_2 - \tilde{P}_3\right\} + \left\{-\tilde{P}_1 + \tilde{P}_4\right\} - \left\{-\tilde{P}_3 + \tilde{P}_6\right\} - \left\{\tilde{P}_4 - \tilde{P}_5\right\} - \left\{\tilde{P}_5 - \tilde{P}_6\right\} \\
&= 0
\end{aligned}$$



(a) $c^{(3)}$ (b) $\partial c^{(3)}$ (c) $\partial\partial c^{(3)}$

Figure 2.24: The boundary of the boundary is empty. Observe that the third figure shows opposed orientation of the 1-cells that comprise the edges of the volume. They consequently cancel each other out.

2.3.3 Cochains and the co-boundary operator δ

We have seen in the previous section that exterior differential k-forms are the natural mathematical objects appearing as integrands over k-dimensional manifolds. After the pull back is applied, integration can be performed in topological space, and we can integrate ω^k over a k-chain $c^{(k)} = \left\{ s_1 c_1^{(k)}, s_2 c_2^{(k)}, \ldots, s_{p-1} c_{p-1}^{(k)}, s_p c_p^{(k)} \right\}$

$$\int_{c^{(k)}} \omega^k = \sum_{i=1}^{p} s_i \int_{c_i^{(k)}} \omega^k = \sum_{i=1}^{p} s_i \cdot \bar{\omega}_i^{(k)} = \bar{\omega}^{(k)} \tag{2.31}$$

$\bar{\omega}^{(k)}$ is called a k-cochain. A cochain is in fact, simply a set of ordered discrete numbers, which are associated with k-chains of the same orientation and numbering. For example, in the dual cell complex of Figure 2.23, we can associate discrete values of the potential $\bar{\varphi}_i$ to all inner oriented 0-cells in the cell complex and collect these in a 0-cochain; assign discrete values of the velocity $\bar{u}_i$ to all inner oriented 1-cells, collected in a 1-cochain; and associate discrete values of the vorticity $\bar{\omega}_i$ with all inner oriented 2-cells, collected in a 2-cochain:

$$\bar{\varphi}^{(0)} = \{\bar{\varphi}_1, \bar{\varphi}_2, \ldots, \bar{\varphi}_5, \bar{\varphi}_6\}, \quad \bar{u}^{(1)} = \{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_6, \bar{u}_7, \} \quad \text{and} \quad \bar{\omega}^{(2)} = \{\bar{\omega}_1, \bar{\omega}_2\}$$

Similarly we can associate outer oriented discrete quantities to all $\tilde{k}$-cells of the outer oriented primal cell complex in Figure 2.23 and collect these in $\tilde{k}$-cochains,

$$\bar{\psi}^{(\tilde{0})} = \left\{\bar{\psi}_1, \bar{\psi}_2\right\}, \quad \bar{q}^{(\tilde{1})} = \{\bar{q}_1, \bar{q}_2, \ldots, \bar{q}_6, \bar{q}_7, \} \quad \text{and} \quad \bar{m}^{(\tilde{2})} = \{\bar{m}_1, \bar{m}_2, \ldots, \bar{m}_5, \bar{m}_6, \}$$

Here $\bar{\psi}_i$ can represent for example discrete values of the stream function; $\bar{q}_i$ discrete values of fluxes and $\bar{m}$ discrete mases.

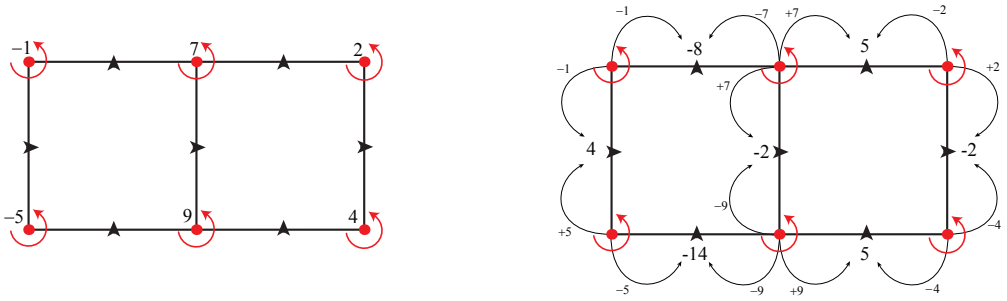
Next we introduce a very important operation, the co-boundary operator δ. It allows us to transfer quantities from k-cells to $(k+1)$-cells, and is the formal adjoint of the boundary operator ∂. The co-boundary operator can be introduced by means of the generalized Stokes theorem in terms of chains and cochains,

$$\delta \bar{\omega}^{(k-1)} \left(c^{(k)} \right) = \bar{\omega}^{(k-1)} \left(\partial c^{(k)} \right) \tag{2.32}$$

Compare this with the generalized Stokes theorem in terms of exterior differential forms,

$$\int_{c^{(k)}} \mathrm{d}\omega^{(k-1)} = \int_{\partial c^{(k)}} \omega^{(k-1)} \tag{2.33}$$

The co-boundary operator δ thus performs the same task on cochains as the exterior derivative d on forms and can consequently be considered as the discrete analogue of the exterior derivative. It allows us to state the topological relations in purely discrete form by use of the generalized Stokes theorem, stated in terms of chains and cochains. As we have seen, the

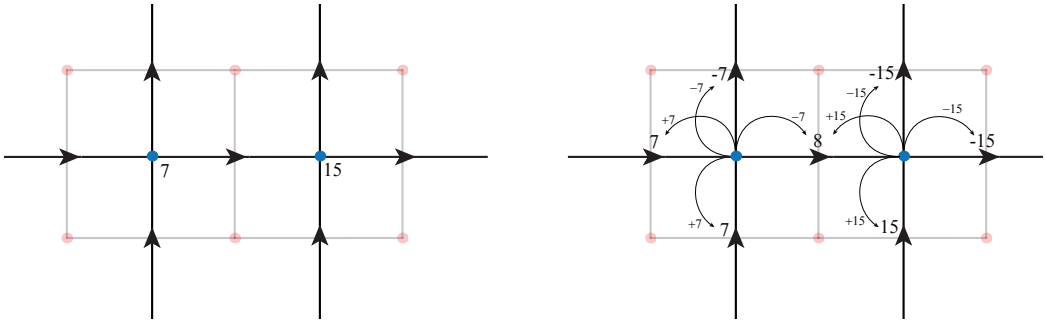


(a) $\bar{\psi}^{(\tilde{0})} = \{-5, 9, 4, -1, 7, 2\}$ (b) $\delta\bar{\psi}^{(\tilde{0})} = \bar{q}^{(\tilde{1})} = \{-14, 5, 4, -2, -2, -8, 5\}$

Figure 2.25: Action of the co-boundary operator on an outer oriented $\tilde{0}$-cochain, relating for example discrete values of the streamfunction $\bar{\psi}^{(\tilde{0})}$ with discrete fluxes $\bar{q}^{(\tilde{1})}$.

(a) $\bar{q}^{(\tilde{1})} = \{-1, -2, 2, 7, 4, 9, -5\}$ (b) $\delta\bar{q}^{(\tilde{1})} = \bar{m}^{(\tilde{2})} = \{-15, 6\}$

Figure 2.26: Action of the co-boundary operator on an outer oriented $\tilde{1}$-cochain, relating for example discrete fluxes $\bar{q}^{(\tilde{1})}$ with discrete mass-densities $\bar{m}^{(\tilde{2})}$.

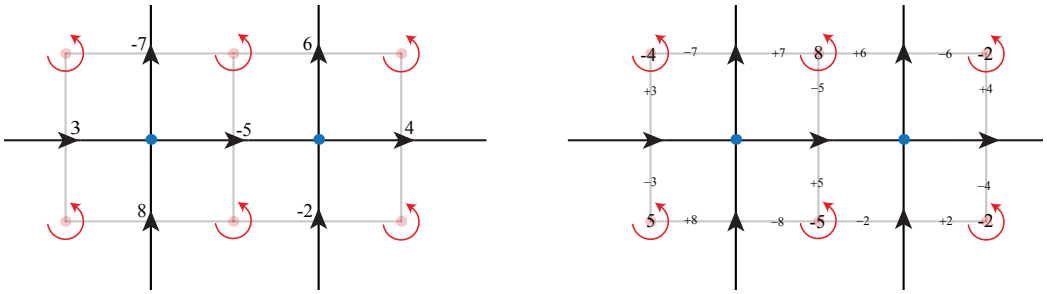


(a) $\bar{\varphi}^{(0)} = \{7, 15\}$ (b) $\delta\bar{\varphi}^{(0)} = \bar{u}^{(1)} = \{7, 15, 7, 8, -15, -7, -15\}$

Figure 2.27: Action of the co-boundary operator on a 0-cochain, relating for example discrete potentials $\bar{\varphi}^{(0)}$ and discrete velocities $\bar{u}^{(1)}$.

exterior derivative d behaves like the gradient, curl and divergence, The co-boundary operator is the discrete version of the exterior derivative and can consequently be associated with the gradient curl and divergence in the same way. Figures 2.25, 2.26, 2.27 and 2.28 illustrate the action of the co-boundary.

Similar to a double action of the exterior derivative, a double action of the co-boundary



(a) $\bar{u}^{(1)} = \{8, -2, 3, -5, 4, -7, 6\}$ (b) $\delta\bar{u}^{(1)} = \bar{\omega}^{(2)} = \{5, -5, -2, -4, 8, -2\}$

Figure 2.28: Action of the co-boundary operator on a 1-cochain, relating for example discrete velocities $\bar{u}^{(1)}$ with discrete vorticity $\bar{\omega}^{(2)}$

operator is zero

$$\delta\delta = 0 \tag{2.34}$$

This property is illustrated for a two-dimensional cell complex in Figure 2.29.

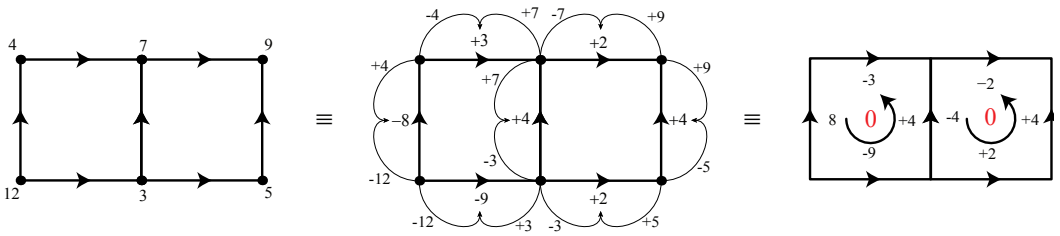


Figure 2.29: The vector identities $\text{curl} \cdot \text{grad}$ and $\text{div} \cdot \text{curl}$ lead to the associated null space. A double action of the co-boundary operator to a k-cochain leads to an empty $(k+2)$-cochain.

Remember the exact sequence, or De Rahm complex

$$\mathbb{R} \;\rightarrow\; H_P \;\overset{\text{grad}}{\rightarrow}\; H_L \;\overset{\text{curl}}{\rightarrow}\; H_S \;\overset{\text{div}}{\rightarrow}\; H_V \;\rightarrow\; 0 \tag{2.35}$$

with the familiar vector identities grad , curl and div ; the gradient mapping values from points to lines, the curl mapping from lines to surfaces and the divergence mapping values from surfaces to volumes.

The co-boundary operator is the discrete version of the div, grad and curl operators, allowing us to formulate (2.35) as an exact discrete sequence

$$\mathbb{R} \;\rightarrow\; \mathcal{C}^{(0)} \;\overset{\delta}{\rightarrow}\; \mathcal{C}^{(1)} \;\overset{\delta}{\rightarrow}\; \mathcal{C}^{(2)} \;\overset{\delta}{\rightarrow}\; \mathcal{C}^{(3)} \;\rightarrow\; 0 \tag{2.36}$$

where $C^{(k)}$ represents the space of k-cochains. In such a way an exact discrete sequence can be set up for both the primal and dual cell complex

2.3.4 Incidence matrices

A numbered and oriented cell complex possesses a matrix representation of the coboundary operator. These so called *incidence matrices* thus allow metric free, discrete representations of the grad, curl and div operators.

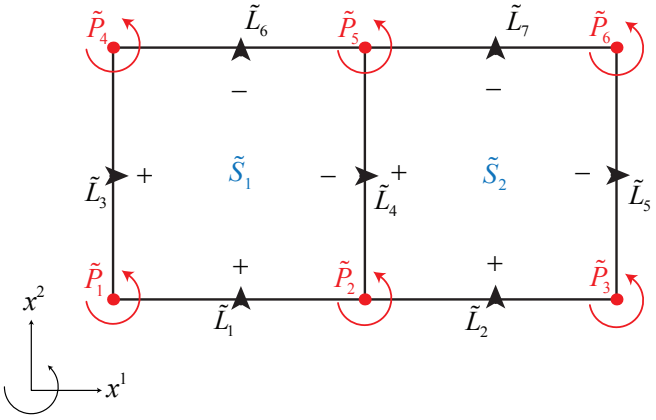
Definition 6 Given a cell complex consisting of a collection of k-cells and its cofaces, a set of $(k+1)$-cells. If all k-cells and $(k+1)$-cells are numbered and oriented, then we can set up an incidence matrix, defined by

$$\mathbb{D}_{ij}^{(k+1,k)} = \begin{cases} 0 & \text{if the j-th } k\text{-cell is not a face of the i-th } (k+1)\text{-cell} \\ 1 & \text{if the j-th } k\text{-cell is a face of the i-th } (k+1)\text{-cell with the same orientation} \\ -1 & \text{if the j-th } k\text{-cell is a face of the i-th } (k+1)\text{-cell with opposite orientation} \end{cases} \tag{2.37}$$

In n dimensions we have n incidence matrices. For instance in three-dimensional space we have the three matrices

$$\mathbb{D}^{(3,2)}, \quad \mathbb{D}^{(2,1)} \quad \text{and} \quad \mathbb{D}^{(1,0)}$$

Consider the numbered and oriented cell complex in Figure 2.30. $\mathbb{D}^{(\tilde{1},\tilde{0})}$ connects outer oriented points with outer oriented lines and $\mathbb{D}^{(\tilde{2},\tilde{1})}$ connects outer oriented lines with outer oriented surfaces.



(a) Primal cell complex K

$\mathbb{D}^{(\tilde{1},\tilde{0})}$	Points 1	2	3	4	5	6
Lines 1	+1	-1	0	0	0	0
2	0	+1	-1	0	0	0
3	-1	0	0	+1	0	0
4	0	-1	0	0	+1	0
5	0	0	-1	0	0	+1
6	0	0	0	+1	-1	0
7	0	0	0	0	+1	-1

(b) Incidence matrix from points to lines

$\mathbb{D}^{(\tilde{2},\tilde{1})}$	Lines 1	2	3	4	5	6	7
Surfaces 1	+1	0	+1	-1	0	-1	0
2	0	+1	0	+1	-1	0	-1

(c) Incidence matrix from lines to surfaces

Figure 2.30: Incidence matrices on the primal cell complex K in $\mathbb{R}^2$

One can readily check that the incidence matrices are a matrix representation of the coboundary-operator.

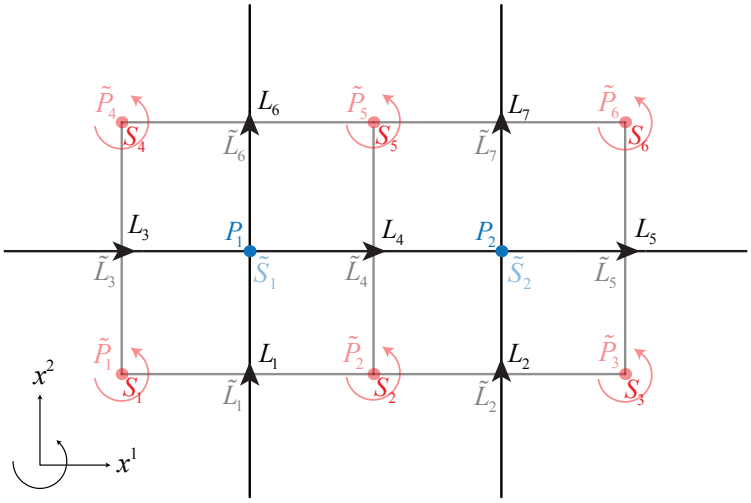
EXAMPLE 2.3.2 Take $\bar{\psi}^{(\tilde{0})} = \{-5, 9, 4, -1, 7, 2, \}$ and $\delta\bar{\psi}^{(\tilde{0})} = \bar{q}^{(\tilde{1})} = \{-14, 5, 4, -2, -2, -8, 5\}$ from Figure 2.25. The action of the coboundary δ on $\bar{\psi}^{(\tilde{0})}$ in terms of the incidence matrix $\mathbb{D}^{(\tilde{1},\tilde{0})}$ is

$$\delta\bar{\psi}^{(\tilde{0})} = \bar{q}^{(\tilde{1})} \qquad \Longrightarrow \qquad \mathbb{D}^{(\tilde{1},\tilde{0})}\,\bar{\psi}^{(\tilde{0})} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} -5 \\ 9 \\ 4 \\ -1 \\ 7 \\ 2 \end{pmatrix} = \begin{pmatrix} -14 \\ 5 \\ 4 \\ -2 \\ -2 \\ -8 \\ 5 \end{pmatrix}$$

EXAMPLE 2.3.3 In example 2.25 we had the $\tilde{1}$-cochain $\bar{q}^{(\tilde{1})} = \{-1, -2, 2, 7, 4, 9, -5\}$ and $\tilde{2}$-cochain $\bar{m}^{(\tilde{2})} = \{-15, 6\}$. The action of the coboundary operator on $\bar{q}^{(\tilde{1})}$ in terms of the incidence matrix $\tilde{\mathbb{D}}^{(2,1)}$ is,

$$\delta\bar{q}^{(\tilde{1})} = \mathbb{D}^{(\tilde{2},\tilde{1})} \cdot \bar{q}^{(\tilde{1})} = \bar{m}^{(\tilde{2})} \qquad \Longrightarrow \qquad \begin{pmatrix} 1 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -2 \\ 2 \\ 7 \\ 4 \\ 9 \\ -5 \end{pmatrix} = \begin{pmatrix} -15 \\ 6 \end{pmatrix}$$

Incidence matrices can also be set up for the dual cell complex. Consider the numbered and oriented dual cell complex in Figure 2.31. $\mathbb{D}^{(1,0)}$ connects points with lines on the dual complex. Similarly $\mathbb{D}^{(2,1)}$ connects lines with surfaces on the dual complex.



(a) Dual cell complex $\tilde{K}$

$\mathbb{D}^{(1,0)}$

Lines \ Points	1	2
1	+1	0
2	0	+1
3	+1	0
4	-1	+1
5	0	-1
6	-1	0
7	0	-1

(b) Incidence matrix from points to lines

$\mathbb{D}^{(2,1)}$

Surfaces \ Lines	1	2	3	4	5	6	7
1	+1	0	-1	0	0	0	0
2	-1	+1	0	-1	0	0	0
3	0	-1	0	0	-1	0	0
4	0	0	+1	0	0	+1	0
5	0	0	0	+1	0	-1	+1
6	0	0	0	0	+1	0	-1

(c) Incidence matrix from lines to surfaces

Figure 2.31: Incidence matrices on the dual complex in $\mathbb{R}^2$. Note that the orientation of the dual cell complex is opposite to that of the primal complex.

The incidence matrices 'mimic' some very important properties of the continuous vector identities, the div, grad and the curl. First of all,

$$\mathbb{D}^{(\tilde{2},\tilde{1})} \cdot \mathbb{D}^{(\tilde{1},\tilde{0})} = 0 \qquad \text{and} \qquad \mathbb{D}^{(2,1)} \cdot \mathbb{D}^{(1,0)} = 0$$

In general we have that

$$\mathbb{D}^{(k+2,k+1)} \cdot \mathbb{D}^{(k+1,k)} = 0$$

These discrete relations are analogue to the relations curl $\cdot$ grad $= 0$ and div $\cdot$ curl $= 0$ in continuous space. Furthermore, observe in Figures 2.30 and 2.31 the following important symmetries between the incidence matrices from the primal complex and the incidence matrices on the dual complex:

$$\mathbb{D}^{(1,0)} = (\mathbb{D}^{(\tilde{2},\tilde{1})})^T \quad \text{and} \quad \mathbb{D}^{(2,1)} = (\mathbb{D}^{(\tilde{1},\tilde{0})})^T$$

In general we have in n-dimensional space:

$$\mathbb{D}^{(n-k+1,n-k)} = (\mathbb{D}^{(\tilde{k+1},\tilde{k})})^T \tag{2.38}$$

This corresponds in $\mathbb{R}^3$ to the symmetry relations that div $= -\text{grad}^T$ and curl $= -\text{curl}^T$. Note the absence of the minus sign. These are included in the orientation.

2.3.5 Discrete De Rahm sequence

Similar as (2.35), we can set up an exact discrete sequence for the dual cell complex. This sequence is set up in reverse order such that discrete quantities associated to p-cells on the primal complex can be related to quantities associated with (n-p)-cells on the dual complex.

$$\begin{array}{ccccccccccc} \mathbb{R} & \rightarrow & \mathcal{C}^{(0)} & \stackrel{\delta}{\rightarrow} & \mathcal{C}^{(1)} & \stackrel{\delta}{\rightarrow} & \mathcal{C}^{(2)} & \stackrel{\delta}{\rightarrow} & \mathcal{C}^{(3)} & \rightarrow & 0 \\ 0 & \leftarrow & \tilde{\mathcal{C}}^{(3)} & \stackrel{\delta}{\leftarrow} & \tilde{\mathcal{C}}^{(2)} & \stackrel{\delta}{\leftarrow} & \tilde{\mathcal{C}}^{(1)} & \stackrel{\delta}{\leftarrow} & \tilde{\mathcal{C}}^{(0)} & \leftarrow & \mathbb{R} \end{array} \tag{2.39}$$

Compare the above discrete De Rahm sequences with Figure 2.14. To summarize, these two sequences are intrinsically discrete and have a purely topological character. They are vertically connected by the constitutive equations, which contain the metric and material parameters. Since the constitutive equations can only be represented in continuous space, we need a continuous representation constructed from the discrete quantities on the primal and dual cell complex. In the following section we introduce two basic operations which provide the connection between the continuous and the discrete.

2.4 Differential Geometry and Algebraic Topology connected

In the previous section we have seen that the topological relations can be exactly represented by discrete topological structures: the incidence matrices. The metric dependent relations, however, - the hodge star, the co-differential etc - can only be explained in terms of the continuous concepts introduced from the mathematical field of Differential Geometry. We thus need to connect the fields of Differential Geometry and Algebraic Topology.

2.4.1 Interpolation of differential forms / Reduction to cochains

We follow the work of Bochev and Hyman [16, 15] and introduce two separate operators - $\mathcal{R}_k$ and $\mathcal{I}_k$ - which define all discrete structures in our framework:

$$\mathcal{R}_k \quad : \quad \Lambda^k \mapsto \mathcal{C}^{(k)} \qquad \text{and} \qquad \mathcal{I}_k \quad : \quad \mathcal{C}^{(k)} \mapsto \Lambda^k_h \tag{2.40}$$

$\mathcal{R}_k$ is called the reduction map, reducing a continuous differential form $f^k \in \Lambda^k(\Omega)$ to a discrete cochain $\bar{f} \in \mathcal{C}^{(k)}$; and $\mathcal{I}_k$ the interpolation map, reconstructing the cochain $\bar{f}$ back to

a finite dimensional representation of a continuous differential form $f_h^k \in \Lambda_h^k(\Omega)$. f_h^k is thus a finite dimensional approximation of f^k.

The map $\mathcal{R}$ should satisfy the important property that $\mathcal{R}\mathrm{d} = \mathbb{D}\mathcal{R}$, i.e. the following is a commutative diagram:

$$\begin{array}{ccc}
\Lambda^k(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^{k+1}(\Omega) \\
\downarrow{\scriptstyle \mathcal{R}_k} & & \downarrow{\scriptstyle \mathcal{R}_{k+1}} \\
\mathcal{C}^{(k)} & \xrightarrow{\mathbb{D}^{(k+1,k)}} & \mathcal{C}^{(k+1)} \\
\downarrow{\scriptstyle \mathcal{I}_k} \uparrow{\scriptstyle \mathcal{R}_k} & & \downarrow{\scriptstyle \mathcal{I}_{k+1}} \uparrow{\scriptstyle \mathcal{R}_{k+1}} \\
\Lambda_h^k(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda_h^{k+1}(\Omega)
\end{array}$$

Differential forms and cochains are naturally related by integration. The straightforward choice for the reduction operation is therefore the DeRahm map

$$\left\langle \mathcal{R}\omega, c^{(k)} \right\rangle = \int_{c^{(k)}} \omega \tag{2.41}$$

The reduction of a 0-form would consequently provide a set of discrete points on the function; reduction of a 1-form leads to a set of line integrals; similarly the reduction of a 2-form by the DeRahm map would give a collection of surface integrals, etc.

The DeRahm map follows the commuting diagram $\mathcal{R}\mathrm{d} = \mathbb{D}\mathcal{R}$ by construction. Using the generalized Stokes theorem and the fact that the boundary operator is the adjoint of the co-boundary provides the proof,

$$\left\langle \mathcal{R}\mathrm{d}\omega, c^{(k+1)} \right\rangle = \int_{c^{(k+1)}} \mathrm{d}\omega = \int_{\partial c^{(k+1)}} \omega = \left\langle \mathcal{R}\omega, \partial c^{(k+1)} \right\rangle = \left\langle \delta\mathcal{R}\omega, c^{(k+1)} \right\rangle \tag{2.42}$$

The DeRahm map is however not the only possible reduction operator, as we shall see in Chapter 4, when we develop a reduction operator for B-splines.

The choice of interpolation operator is quite flexible, since there are a multitude of ways in which global quantities can be used to reconstruct local approximations. For example, the Finite Difference Method reconstructs quantities in the form of a Taylor series expansion. The Finite Element Method uses an expansion in terms of basis functions. However, to obtain consistent discrete approximations, $\mathcal{I}$ should satisfy two conditions. First of all, $\mathcal{I}$ should be the right inverse of $\mathcal{R}$,

$$\mathcal{R}\mathcal{I}\ \mathcal{C}^{(k)} = \mathcal{C}^{(k)} \qquad \Longrightarrow \qquad \mathcal{R}\mathcal{I} = id \tag{2.43}$$

Secondly, the interpolation map $\mathcal{I}$ should be an approximate left inverse of $\mathcal{R}$

$$\mathcal{I}\mathcal{R}\ \Lambda^k = \Lambda_h^k = \Lambda^k + O\left(h^{p+1}\right) \qquad \Longrightarrow \qquad \mathcal{I}\mathcal{R} = id + O\left(h^{p+1}\right) \tag{2.44}$$

where h and $p+1$ denote respectively the partitioning size and the approximation order. Of course the approximation order also depends on the smoothness of the exact solution.

Appendix A illustrates interpolation of cochains by a general class of basis functions.

Before we apply B-splines to connect the two fields of Differential Geometry and Algebraic Topology in Chapter 4, we introduce them in the context of geometry and mesh generation in the following chapter.

Chapter 3

B-splines and NURBS as a basis for geometry and mesh generation

In this chapter we introduce B-splines and Non-Uniform Rational B-splines (NURBS) in the context of geometry and mesh generation. We will see that the properties that make B-splines and NURBS attractive for use in CAD, also make them attractive for use in analysis. We will first introduce some of the common CAD vocabulary, and relate with FEA. Subsequently we introduce B-splines in one and two dimensions and study its interesting refinement properties. Finally, we discuss the construction of NURBS from B-splines and review the state of the art concerning geometry and mesh generation for IsoGeometric Analysis.

Knowledge of the construction of B-splines, thoroughly explained in this chapter, will provide a solid background for Chapter 4, where we construct B-spline spaces of differential forms for use in analysis.

3.1 Things to know about CAD before starting IsoGeometric Analysis

Computer Aided Geometric Design (CAGD) is concerned with the generation of smooth curves and surfaces, which generally have to satisfy a large number of constraints. To achieve this using a single polynomial requires a high degree, since a p-degree polynomial can satisfy $p+1$ constraints. High degree polynomials are inefficient to process, can become unstable and have the disadvantage that changes are global, while local control is demanded [54].

These issues can obviously be overcome by using more than one polynomial element. Successive polynomial elements should join with some level of continuity and this continuity should be maintained during editing of the piecewise polynomial. A function constructed from polynomial elements joined together with some prescribed level of continuity between element

interfaces, is called a spline.

The word spline, finds its origin in shipbuilding, where it refers to a bendable stroke of wood, along which smooth curves could be drawn for the lines plan of a ship. Metal weights, called ducks, were placed such that the spline had its preferred shape on the drawing board, see Figure 3.1. Basic theory of structures explains that the bending moment M is a continuous

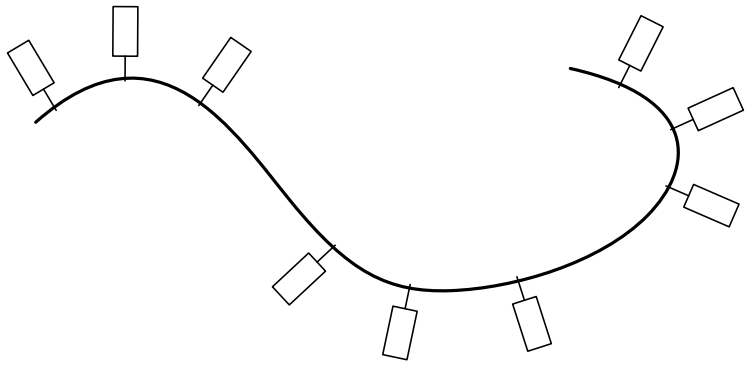


Figure 3.1: Ducks holding the spline in place on the drawing board

function along the spline, except at a duck, where M is generally only C^0 continuous. Since the curvature of the spline is proportional to M, $\kappa = \frac{M}{EI}$, the spline is curvature continuous everywhere. Curvature continuity is an important requirement in design, since it guarantees the smooth change of reflections. A spline of degree three (4th order) has a maximum continuity of C^2, and is thus curvature continuous. Cubic degree splines are therefore most commonly used in CAD.

For a polynomial basis to be useful in a CAD environment, the required continuity should be build directly into the basis. Otherwise, inter-element smoothness would be cumbersome to maintain. This is exactly what made B-splines such a success in CAGD. B-splines, where the B stands for basis, are the natural basis functions in which to define splines. Smoothness is directly built into the B-spline basis and can be arbitrarily chosen from C^0, to maximum C^{p-1} continuity, between elements.

While B-splines are convenient for free form modeling, they are not capable of exactly representing some simple conic sections like circles and ellipsoids, making them less suitable for CAD. This why today, the facto standard technology in CAD is a generalization of B-splines called NURBS. NURBS stands for Non-Uniform Rational B-splines and are rational functions of B-splines. NURBS inherit all their favorable properties for use in free form design, like the build-in inter-element continuity and the local control. NURBS, however, extend B-splines by allowing the exact representation of conic sections.

In the context of IsoGeometric Analysis, CAD is not merely a tool to create the geometric model. It simultaneously delivers a coarse mesh, which is ready for use in analysis. Because this coarse mesh represents the exact geometry, the mesh can be refined and order elevated, without communication with the CAD program. This means a giant leap forward compared to traditional FEA, where CAD geometry needs to be translated into a format which is suitable

for finite element analysis. This process is highly labor intensive, often costing more time than the actual analysis, and besides is accompanied by a loss in geometric precision [42]. Moreover, at every refinement step, communication is required with a CAD -program.

Just as with standard polynomials applied in iso-parametric FEM, we define NURBS and B-spline geometry using a parametric map which consist of a linear combination of a set of basis functions and associated degrees of freedom. In the curve case we have

$$\mathbf{C}(x) = \sum_{i=0}^{n} \bar{\mathbf{P}}_i \; N_{i,p}(x), \qquad a \leq x \leq b,$$

where the counter n is either bigger or equal to the degree p. (Remember that for standard polynomials, n is always equal to p).

Figure 3.2 illustrates such a parametric map $\mathbf{y} = \mathbf{C}(x)$, from parameter space to physical space, for a B-spline curve of degree 3. The curve consist out of four polynomial pieces which join smoothly (C^2 continuous) at element interfaces. These locations are called the *knots* or *breakpoints*. In the following we shall only refer to them as knots. The degrees of freedom are

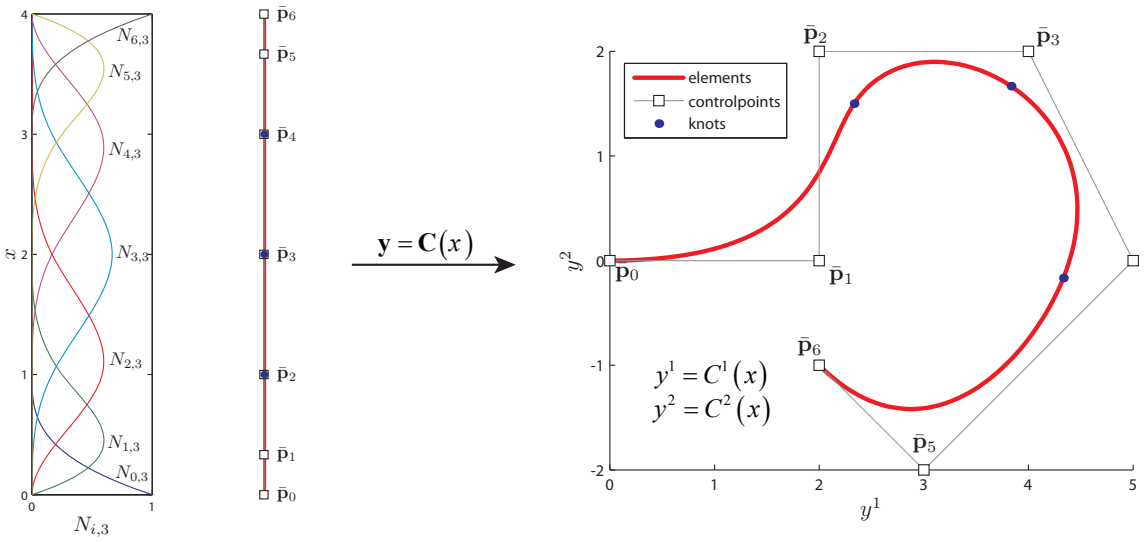
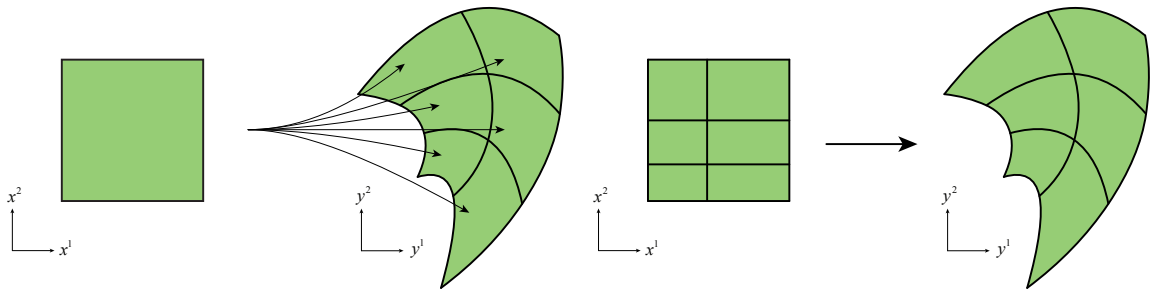


Figure 3.2: A cubic B-spline curve in two-dimensional space defined as $\mathbf{C}(x) = \sum_{i=0}^{6} \bar{\mathbf{P}}_i \; N_{i,3}(x), \qquad 0 \leq x \leq 4$. The curve consists out of 4 polynomial pieces joined together with curvature continuity.

called *control points*. Their collection forms a control polygon, generally called the *control mesh*. At the start and end points, the spline is clamped in. This allows to accurately define the start/end points and the tangent, see figure 3.2. The control mesh is a first order approximation of the curve. In fact, the spline is always confined in the convex hull of the control mesh. This is called the convex hull property, which in practice means that the spline cannot wiggle more than the control mesh does. This is exactly why splines are not polluted

by Runge's phenomena. The control mesh allows intuitive editing of the B-spline or NURBS curve. This is yet another reason why B-splines and NURBS are so convenient in geometric design.

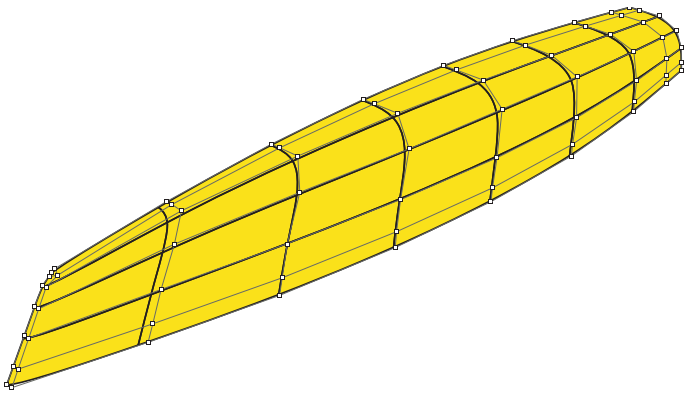
Unlike standard finite element basis functions, B-splines and NURBS basis functions are local to patches, rather than elements, see Figure 3.3. A patch is a rectangular sub domain which is partitioned into elements. Patches can therefore, in the context of finite element analysis, be seen as macro- or super-elements. As illustrated in Figure 3.3(b), geometry is defined patch wise by way of a single parametric map from parameter space to the physical space. In contrast, standard finite elements each have their own mapping from the parent element to physical space, see Figure 3.3(a).



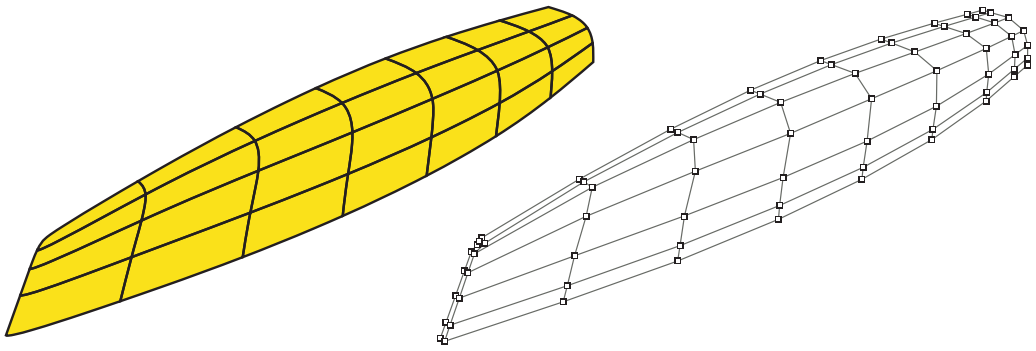
(a) FEA (b) IGA

Figure 3.3: In classical finite element analysis (3.3(a)), the parameter space is local to elements. Each element has its own mapping from the reference element to physical space. In NURBS based IsoGeometric Analysis on the other hand (3.3(b)), the B-spline parameter space is local to the entire patch. Internal knots partition the patch into elements. A single B-spline maps parameter space to physical space. From Cottrell et al. [22]

In contrast to standard polynomial bases, B-spline and NURBS geometry allow two definitions of a mesh. As in standard FEM there is the physical mesh denoting the geometry partitioned into elements. Secondly there is the notion of the control mesh which controls the geometry. An example illustrating the two definitions of the mesh, is given in Figure 3.4. This example shows a CAD-model of a sailing yacht, where only half of the hull is modeled because of symmetry. This geometry is modeled using one patch only.



(a) CAD half model of a sailing yacht



(b) Physical mesh (c) Control mesh

Figure 3.4: A CAD half model of a sailing yacht, modeled using one patch only. NURBS are characterized by two definitions of a mesh: a physical mesh of the geometry, Figure (b); and a control mesh, which controls the geometry, Figure (c).

3.2 B-splines

Now we are familiar with the CAD vocabulary, we can discuss the generation of NURBS and their derivatives. Because NURBS are constructed from B-splines, it is straightforward to start discussing B-splines.

3.2.1 B-spline basis functions

There are several ways to define B-spline basis functions and to prove their important properties. Originally, B-splines were invented by Schoenberg [58, 59], whom is most often called the father of splines. He derived B-splines using divided differences [26], showing that B-splines have in fact much in common with finite differencing. From this definition on, Cox [24] and de Boor [27] independently deduced a stable algorithm which recursively computes the B-spline basis of certain degree and continuity, known as the Cox-DeBoor recursive formula. This algorithm paved the way of application of B-splines in computer graphics and computer-aided geometric design.

A totally different way of looking at B-spline theory, are Polar Forms [55, 56, 63]. This elegant classical mathematical tool, also called blossoming, is based on the principle that polynomials are equivalent to multi-affine maps. Polar forms can simplify the construction of polynomial and piecewise polynomial curves and surfaces [63] and have led to new surface representations and algorithms. Take for example, smooth piecewise polynomial surfaces over arbitrary triangulations [25], which are otherwise very hard to construct.

Since this is merely an introduction into B-spline theory, we shall introduce B-splines and deduce some of their important properties directly from the Cox-DeBoor recursive formula.

The recursion starts with piecewise constants and builds higher order basis functions by the use of the so called knot vector.

Definition 7 Let $\mathbf{X} = \{x_0, ..., x_m\}$ be a non-decreasing sequence of real numbers, i.e. $x_i \leq x_{i+1}$, $i = 0, ..., m-1$. The x_i are called *knots*, and $\mathbf{X}$ is the *knot vector*. The ith B-spline basis function of degree p, denoted by $N_{i,p}(x)$, is defined as

$$N_{i,0}(x) = \begin{cases} 1 & \text{if } x_i \leq x \leq x_{i+1}, \\ 0 & \text{otherwise }, \end{cases}$$

For p =1,2,3,..., they are defined by

$$N_{i,p}(x) = \frac{x - x_i}{x_{i+p} - x_i} N_{i,p-1}(x) + \frac{x_{i+p+1} - x}{x_{i+p+1} - x_{i+1}} N_{i+1,p-1}(x). \tag{3.1}$$

(By convention $\frac{0}{0}$ is defined as 0)

Figure 3.4 illustrates the recursive generation of a cubic degree B-spline basis, by the Cox-DeBoor formula.

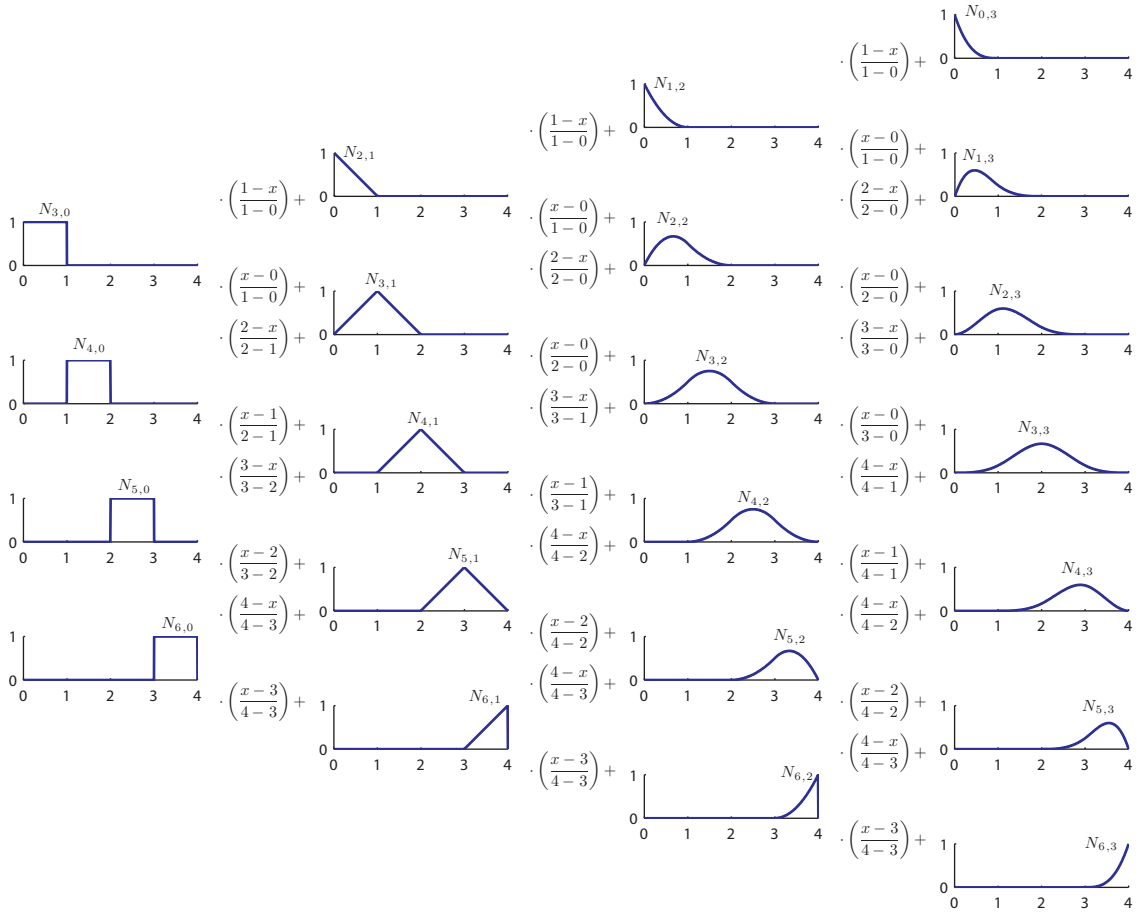


Figure 3.4: Recursive generation of a cubic B-spline basis using the De Boor formula and a knot vector of $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$

The recursion marches through the knot vector $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$ (This will be made clear in Section 3.2.2) and takes linear combinations of two $(p-1)$-degree basis functions to generate a p-degree basis function. Observe also that for increasing degree, the continuity increases together with its support. The support of a p-degree B-spline is $p+1$ knot spans. Note that the recursion always generates a triangular table.

In Figure 3.5 we zoom into the cubic basis function $N_{3,3}(x)$ generated in Figure 3.4. This figure illustrates how B-spline basis functions are composed out of different polynomial pieces. The basis function is nonzero only from $x = 0$ to 4 and has internal knots at $x = 1$, 2 and 3.

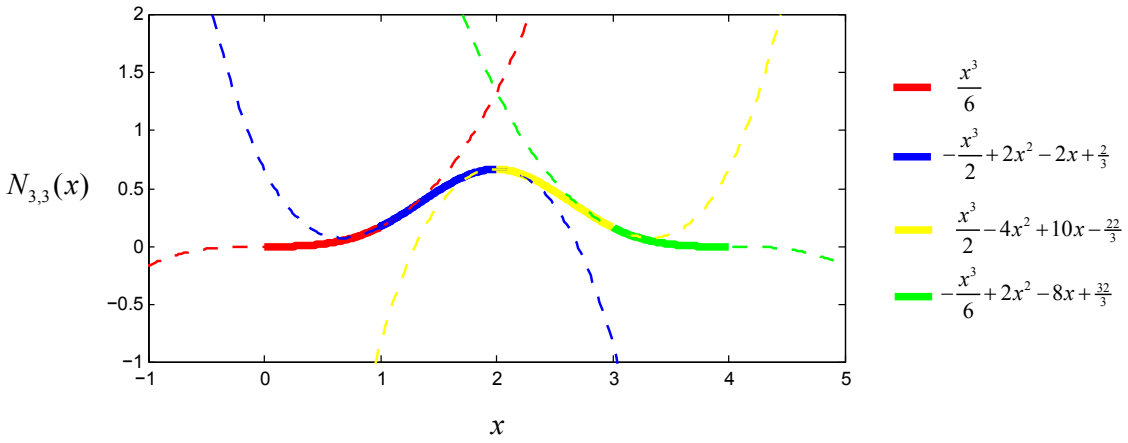


Figure 3.5: The cubic B-spline basis function $N_{3,3}(x)$ is composed out of 4 different polynomial pieces.

B-splines posses several important properties for use in CAD as well as analysis, which can directly be derived from the recursive formula. Some of these properties can immediately be obtained from figure 3.4.

1. The basis constitutes a partition of unity,

$$\sum_{i=0}^{n} N_{i,p}(x) = 1 \tag{3.2}$$

2. The basis functions always form a linear independent basis. This is particularly important in analysis.

3. B-splines feature local support, i.e. they are non-zero in a finite part of the domain. In fact, the support is minimal since it spans $p+1$ knots. Local support makes sure that changes have a local effect only, and leads to sparsely banded matrices.

4. Continuity between polynomial elements can be prescribed resulting up to $p-1$ continuous derivatives. This is possibly the most striking feature of B-splines compared to standard polynomial bases applied in FEA.

5. The basis functions are strictly positive, as can be seen in Figure 3.4. This is why a B-spline is always confined in the convex hull of its control polygon. We already mentioned that the convex hull property makes B-splines well behaved, since a B-spline cannot wiggle more than the control polygon does.

6. Variation of a B-spline curve decreases with increasing degree. This is called the variation diminishing property of B-splines, see Figure 3.6. B-splines respond less to a change in the coefficients (control points) with increasing degree. This can be good or bad. Good in the sense that small disturbances in the coefficients do not lead to excessive differences in the final representation of the polynomial curve, when the degree goes up (standard polynomial bases are notorious for this, known as Runge's phenomenon). Bad, because high order B-splines respond less to a change in the coefficients leading to ill conditioned interpolation or mass matrices, already for medium to high order.

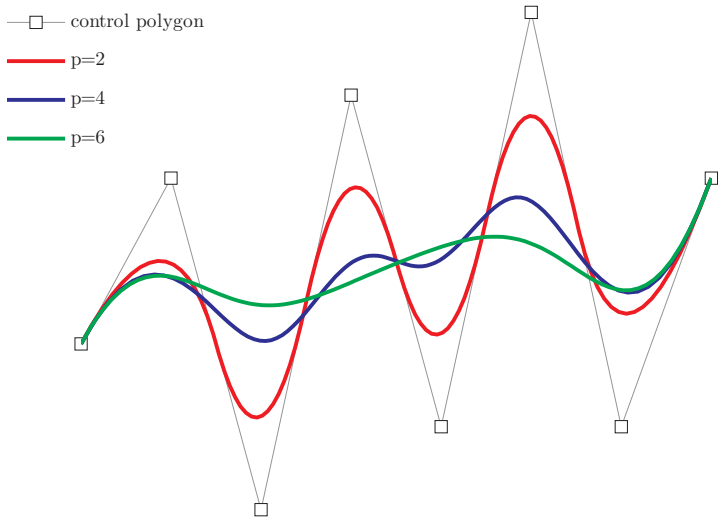


Figure 3.6: The variation decreases for increasing degree. Note also that the B-spline is confined in the convex hull of its control polygon.

3.2.2 The knot vector

The p-degree B-spline basis is uniquely defined by the knot vector. This is why it is very important to understand exactly how the knot vector influences the basis functions. We already know that the knot vector is an increasing set of numbers which partitions a patch into elements. The general representation of the knot vector looks like

$$\mathbf{X} = \{ \underbrace{a, ..., a}_{p+1}, x_{p+1}, ..., x_i, x_{i+1}, ..., x_{m-p-1}, \underbrace{b, ..., b}_{p+1} \}, \qquad x_{i+1} \geq x_i \tag{3.3}$$

This knot vector contains $m+1$ knots and in general we choose $a = 0$ and $b = 1$.

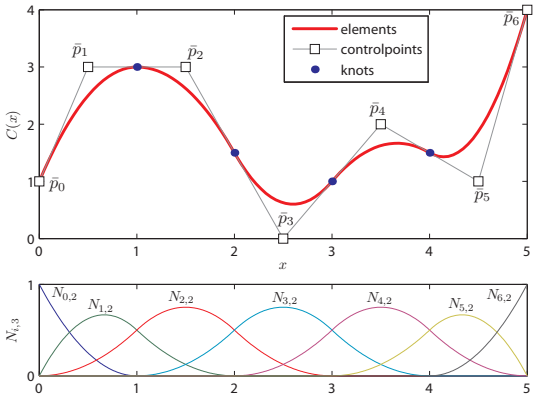
Two special cases arise from (3.3). First of all if $h = x_{i+1} - x_i$ is constant, then the knot vector is uniform. The corresponding B-splines are called cardinal. The second special case occurs when no internal knots are present (3.4), in which case the B-spline consist of one polynomial element and reduces to a Bezier curve.

$$\mathbf{X} = \{ \underbrace{a, ..., a}_{p+1}, \underbrace{b, ..., b}_{p+1} \} \tag{3.4}$$

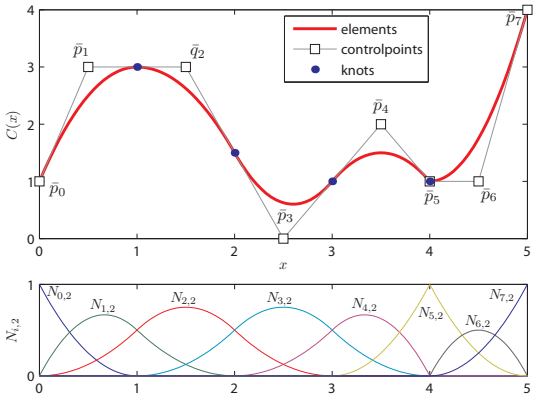
The definition of the knot vector in (3.3) provides two ways in which we can manipulate the p-degree B-spline basis

1. Choose a non-uniform distribution of the knots.
2. Introduce repeated knots.

A non-uniform distribution of the knots allows more resolution in a certain part of the domain. Repeated knots allow local control over the continuity of the basis. Every multiple knot decreases the continuity by one. The knot vector in (3.3) starts and ends with $p+1$ multiple knots, hence the continuity is locally decreased from C^{p-1} to C^0 continuity. This means that the begin and end control points of the B-spline will be a nodal interpolation. Such a knot vector is called clamped. Unclamped B-splines are not common in CAD, and we will not discuss these type of B-spline basis here.



(a) $p = 2$, $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ and $\bar{P} = (1, 3, 3, 0, 2, 1, 4)$



(b) $p = 2$, $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$ and $\bar{P} = (1, 3, 3, 0, 2, 1, 1, 4)$

Figure 3.7: Figure 3.7(a) shows a uniform quadratic B-spline curve. The repeated knot in the knot vector $\mathbf{X}$ in figure 3.7(b) causes a kink in the curve, by introducing a new basis function and changing the existing basis functions only in span $x \in [3, 5]$.

Figure 3.7 illustrates the effect of multiple knots. Both curves are clamped at the begin and end-point, because the knot vectors start and end with $p+1$ repeated knots. The knot vector which defines the B-spline basis in Figure 3.7(b) contains 1 multiple knot at $x = 4$. This

repeated knot only affects the basis functions in span $x \in [3, 5]$. An extra basis function is created and the continuity is locally decreased from C^1 to C^0 at $x = 4$.

Although knot vectors constitute a global partitioning of the parametric domain, they have a local character as far as the basis functions are concerned. Table 3.1 depicts how the global knot vector $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$, used to construct the basis in Figure 3.7(b), can be decomposed into a local knot vector relevant to basis function $N_{i,2}$. In general, the basis

Table 3.1: Local knot vector associated with basis function $N_{i,2}$

$\mathbf{X}$:	0	0	0	1	2	3	4	4	5	5	5
$N_{0,2}$:	0	0	0	1							
$N_{1,2}$:		0	0	1	2						
$N_{2,2}$:			0	1	2	3					
$N_{3,2}$:				1	2	3	4				
$N_{4,2}$:					2	3	4	4			
$N_{5,2}$:						3	4	4	5		
$N_{6,2}$:							4	4	5	5	
$N_{7,2}$:								4	5	5	5

function $N_{i,p}(x)$ is defined by the $p + 1$ knots x_i to x_{i+p+1}. The begin and end knots give us the support of the basis function, i.e. where the basis function is non-zero. This support is minimal and spans $p + 1$ knots. Observe that at every parametric point $x \in [x_i, x_{i+1}]$, $p + 1$ basis functions are non-zero. Take for example $x = 2.5 \in [2, 3]$. In Table 3.1 we see that the three non-zero basis functions are $N_{2,2}$, $N_{3,2}$ and $N_{4,2}$.

3.2.3 B-spline curves

A B-spline curve of degree p is defined as a linear combination of control points $\bar{\mathbf{P}}_i$ and p-degree B-spline basis functions $N_{i,p}(x)$,

$$\mathbf{C}(x) = \sum_{i=0}^{n} \bar{\mathbf{P}}_i \, N_{i,p}(x), \qquad a \leq x \leq b, \tag{3.5}$$

where the basis functions are defined on the non-uniform knot vector as defined in (3.3).

An example of a parametric B-spline curve in two-dimensions is shown in Figure 3.8

$\mathbf{y} = \mathbf{C}(x)$ can be seen as a parametric map, mapping a point x in parameter space to each of the physical coordinates y^j.

$$y^1 = C^1(x) = \sum_{i=0}^{7} \bar{P}_i^1 \, N_{i,2}(x) \qquad \text{and} \qquad y^2 = C^2(x) = \sum_{i=0}^{7} \bar{P}_i^2 \, N_{i,2}(x).$$

In order to calculate a point on a B-spline curve three steps are necessary

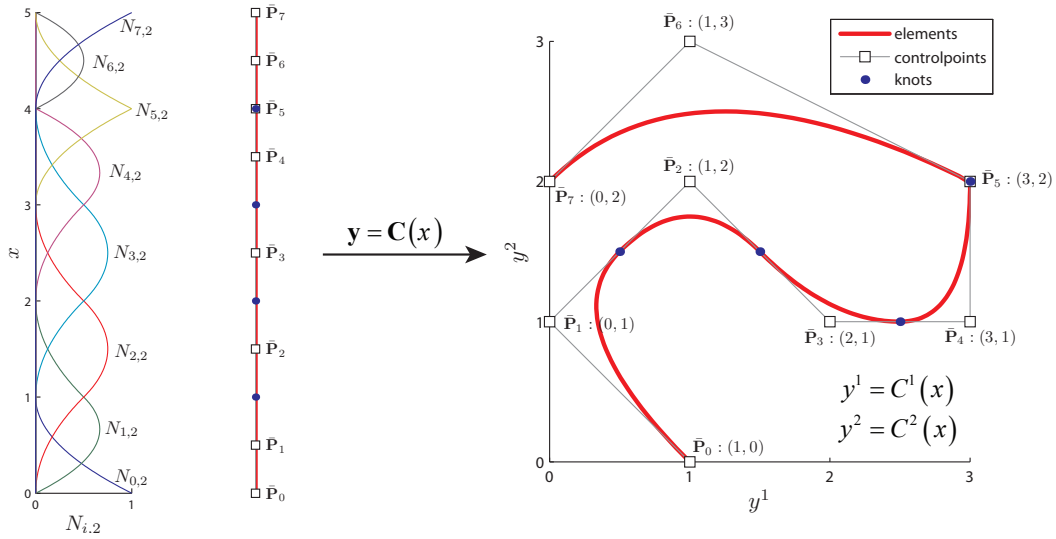


Figure 3.8: A quadratic B-spline curve is mapped from parameter space to physical space. The B-spline basis is defined by knot vector $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

1. Determine local knot span in which the point lies;

2. Calculate the $p + 1$ non-vanishing basis functions;

3. Multiplying the nonzero basis functions with the corresponding control points;

Lets say we want to compute the value on the quadratic degree B-spline curve, shown in Figure 3.8, at parameter value $x = 2.5$. The local knot span in which x lies is obviously $2 - 3$. On the left hand side of Figure 3.8 and in Table 3.1 we can see that the nonzero basis functions are $N_{2,2}$, $N_{3,2}$ and $N_{4,2}$. We can now calculate the values of the non-zero basis functions at $x = 2.5$, by applying the De Boor algorithm (Definition 7) as in Figure 3.9.

$$
\begin{aligned}
N_{2,2}(2.5) &= \left(\frac{3-2.5}{3-2}\right) \cdot \left(\frac{3-2.5}{3-1}\right) = 1/8 \\
N_{3,2}(2.5) &= \left(\frac{3-2.5}{3-2}\right) \cdot \left(\frac{2.5-1}{3-1}\right) + \left(\frac{2.5-2}{3-2}\right) \cdot \left(\frac{4-2.5}{4-2}\right) = 6/8 \\
N_{4,2}(2.5) &= \left(\frac{2.5-2}{3-2}\right) \cdot \left(\frac{2.5-2}{4-2}\right) = 1/8
\end{aligned}
$$

Subsequentally we multiply the non-zero basis functions by their corresponding control points,

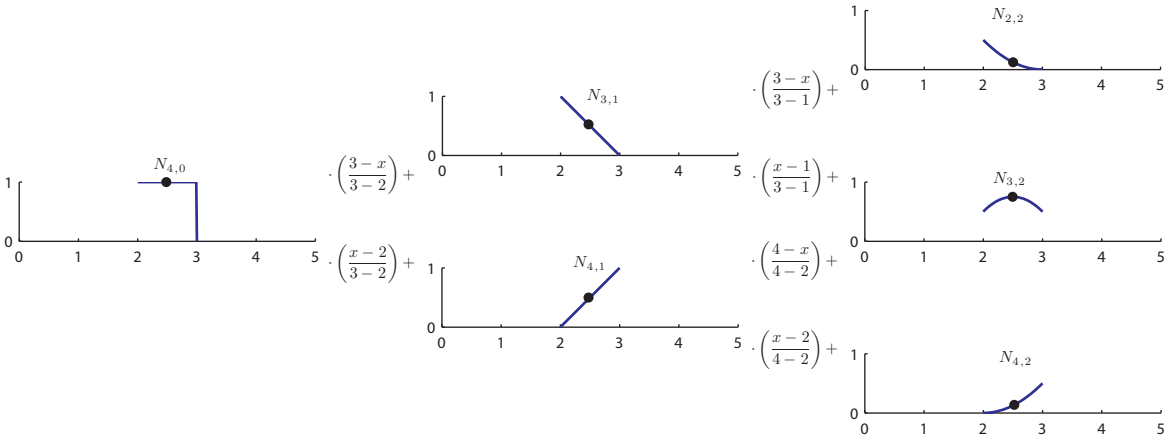


Figure 3.9: Recursive generation of quadratic b-spline basis, defined by knot vector $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$, in knot span $x \in [2, 3]$

see Figure 3.8.

$$C(2.5) = \sum_{i=2}^{4} \bar{\mathbf{P}}_i \, N_{i,2}(2.5) = 1/8 \cdot \bar{\mathbf{P}}_2 + 6/8 \cdot \bar{\mathbf{P}}_3 + 1/8 \cdot \bar{\mathbf{p}}_4$$
$$= 1/8 \cdot \begin{pmatrix} 1 & 2 \end{pmatrix} + 6/8 \cdot \begin{pmatrix} 2 & 1 \end{pmatrix} + 1/8 \cdot \begin{pmatrix} 3 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 9/8 \end{pmatrix}$$

So $y^1 = 2$ and $y^2 = 9/8$.

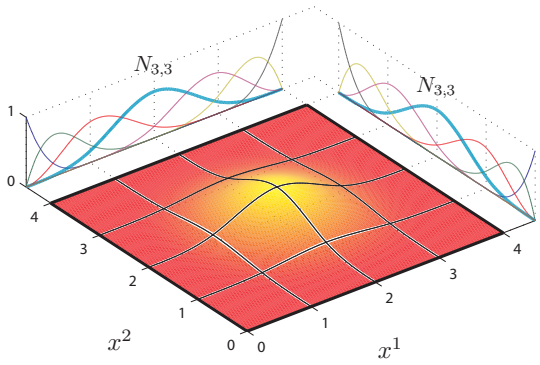
3.2.4 B-spline surfaces

There are roughly two ways in which the uni-variate B-splines can be extended to higher dimensions. By the theory of polar forms, B-splines have been generalized to arbitrary triangulations. These triangular B-splines [25, 53] share many interesting properties with the uni-variate case, they are, however, difficult to construct. Most other methods to construct B-spline bases in higher dimensions rely on tensor products of the uni-variate basis functions. This is how we shall extend B-splines to higher dimensions.

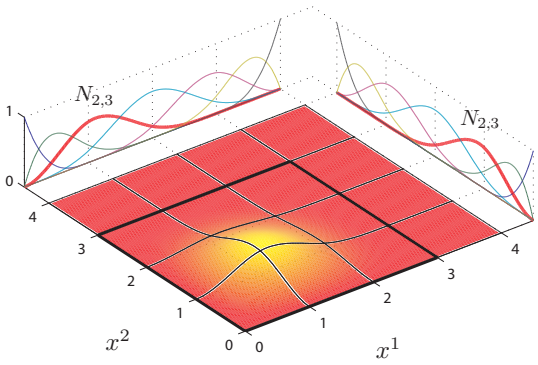
Similar to the curve case, a B-spline surface is defined as a linear combination of its control points $\mathbf{P}_{ij}$ and corresponding tensor product basis functions $N_{i,p}(x^1) \cdot N_{j,q}(x^2)$.

$$\mathbf{S}(x^1, x^2) = \sum_{i=0}^{n} \sum_{j=0}^{m} \bar{\mathbf{P}}_{ij} \, N_{i,p}(x^1) \, N_{j,q}(x^2). \tag{3.6}$$

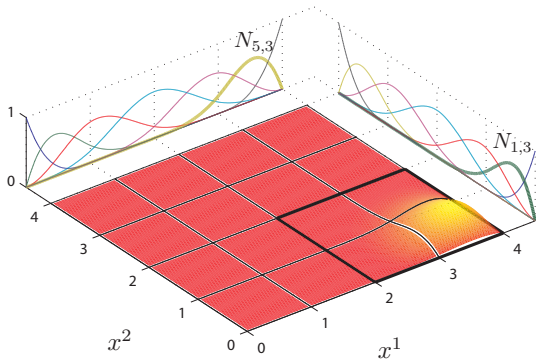
Note that the tensor product basis functions $N_{i,p}(x^1) \cdot N_{j,q}(x^2)$ are simply the product of univariate B-spline basis functions, each parameterized in a different coordinate x^j, defined on a separate knot vector $\mathbf{X}^j$ and possibly of a different degree. Some examples of bi-cubic tensor product B-spline basis functions are presented in Figure 3.10.



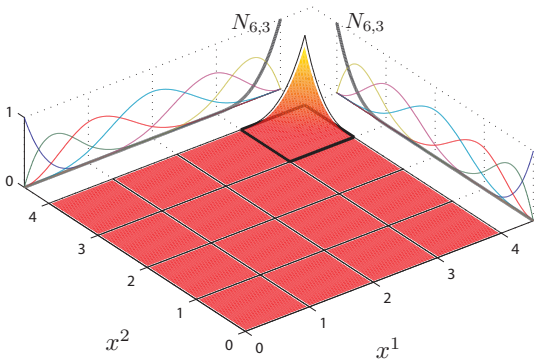
(a) $N_{3,3}(x^1) \cdot N_{3,3}(x^2)$



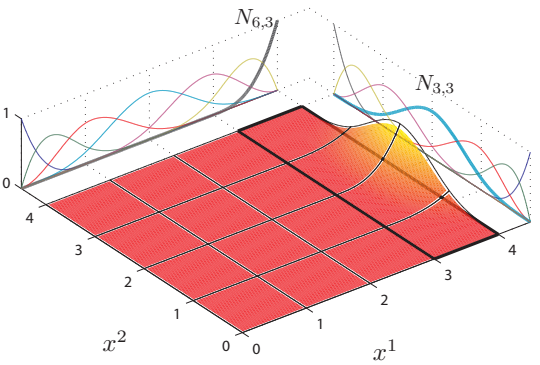
(b) $N_{2,3}(x^1) \cdot N_{2,3}(x^2)$



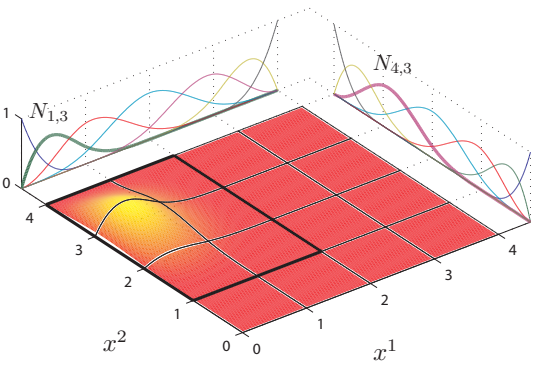
(c) $N_{5,3}(x^1) \cdot N_{1,3}(x^2)$



(d) $N_{6,3}(x^1) \cdot N_{6,3}(x^2)$



(e) $N_{6,3}(x^1) \cdot N_{3,3}(x^2)$



(f) $N_{1,3}(x^1) \cdot N_{4,3}(x^2)$

Figure 3.10: Some examples of bi-cubic two-dimensional B-spline basis functions constructed by tensor products of cubic uni-variate basis functions. The one-dimensional bases are defined by the knot vectors $\mathbf{X}^1 = \mathbf{X}^2 = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$. The black selection denotes the support of the respective basis function

$\mathbf{y} = \mathbf{S}(\mathbf{x})$, in 3.6, can be seen as a parametric map, mapping local coordinates x^1 and x^2 to the physical coordinates y^i. An example of a bi-quadratic surface patch spanning two elements, is shown in Figure 4.13. Its bases are defined on knot vectors $\mathbf{X}^1 = \{0, 0, 0, 2, 2, 2\}$ and $\mathbf{X}^2 = \{0, 0, 0, 1.5, 3, 3, 3\}$ and the physical coordinates are calculated using

$$y^1 = S^1(x^1, x^2) = \sum_{i=0}^{2}\sum_{j=0}^{3} \bar{P}^1_{ij}\, N_{i,2}(x^1)N_{j,2}(x^2)$$

$$y^2 = S^1(x^1, x^2) = \sum_{i=0}^{2}\sum_{j=0}^{3} \bar{P}^2_{ij}\, N_{i,2}(x^1)N_{j,2}(x^2)$$

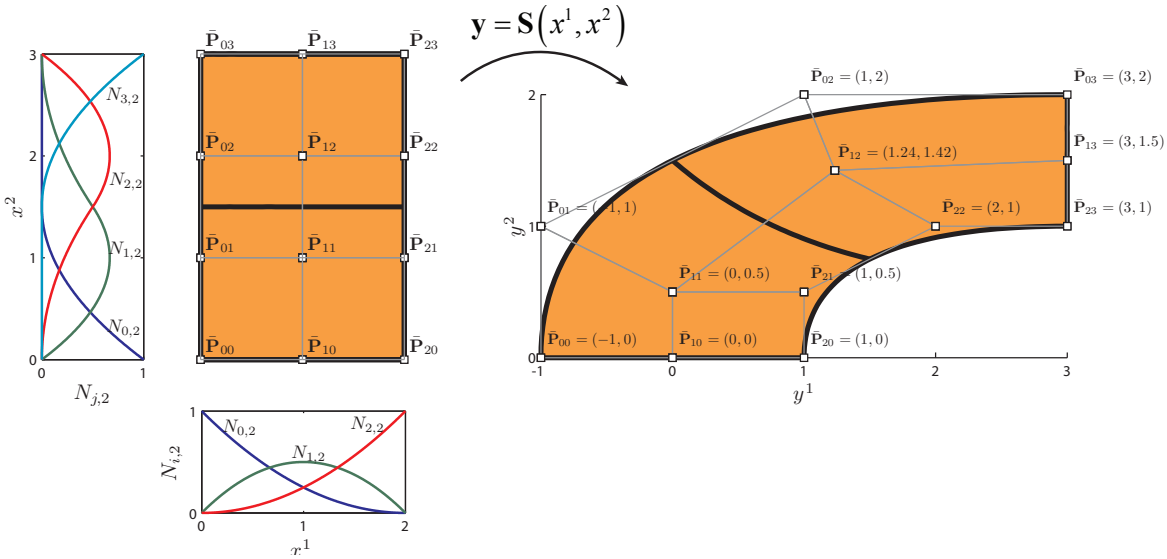


Figure 3.11: Bi-quadratic tensor product B-spline surface constructed from knot vectors $\mathbf{X}^1 = \{0, 0, 0, 2, 2, 2\}$ and $\mathbf{X}^2 = \{0, 0, 0, 1.5, 3, 3, 3\}$

3.2.5 Derivatives of B-splines

The recurrence relation (Definition 7) leads to an interesting result for derivatives of B-spline basis functions (proof see Piegl and Tiller [54])

$$\frac{\partial}{\partial x} N_{i,p}(x) = \frac{p}{x_{i+p} - x_i}\, N_{i,p-1}(x) - \frac{p}{x_{i+p+1} - x_{i+1}}\, N_{i+1,p-1}(x). \tag{3.7}$$

We can directly use (3.7) to define the derivative of a B-spline curve

$$\mathbf{C}'(x) = \sum_{i=0}^{n} \bar{\mathbf{P}}_i\, \frac{\partial}{\partial x} N_{i,p}(x), \tag{3.8}$$

or we can insert (3.7) into (3.8) to obtain the derivative of the curve in terms of a B-spline basis of one degree lower

$$\begin{aligned}\mathbf{C}'(x) &= \sum_{i=0}^{n} \bar{\mathbf{P}}_i \left(\frac{p}{x_{i+p} - x_i} N_{i,p-1}(x) - \frac{p}{x_{i+p+1} - x_{i+1}} N_{i+1,p-1}(x) \right) \\ &= \sum_{i=0}^{n} \bar{\mathbf{P}}_i \left(p \frac{N_{i,p-1}(x)}{x_{i+p} - x_i} \right) - \sum_{i=1}^{n+1} \bar{\mathbf{P}}_{i-1} \left(p \frac{N_{i,p-1}(x)}{x_{i+p} - x_i} \right) \\ &= \bar{\mathbf{P}}_0 \left(p \frac{N_{0,p-1}(x)}{x_p - x_0} \right) + \sum_{i=1}^{n} \left(\bar{\mathbf{P}}_i - \bar{\mathbf{P}}_{i-1} \right) \left(p \frac{N_{i,p-1}(x)}{x_{i+p} - x_i} \right) - \bar{\mathbf{P}}_n \left(p \frac{N_{n+1,p-1}(x)}{x_{n+p+1} - x_{n+1}} \right)\end{aligned}$$

The first and last term evaluate to $\frac{0}{0}$, which is 0 by definition, so

$$\mathbf{C}'(x) = \sum_{i=1}^{n} \left(\bar{\mathbf{P}}_i - \bar{\mathbf{P}}_{i-1} \right) \left(p \frac{N_{i,p-1}(x)}{x_{i+p} - x_i} \right).$$

Now let $\mathbf{X}'$ be the knot vector obtained by dropping the first and the last knots from $\mathbf{X}$, i.e. $\mathbf{X}' = \{ \underbrace{a, ..., a}_{p}, x_1, ..., x_{m-p-2}, \underbrace{b, ..., b}_{p} \}$. Then it is easy to check that the function $N_{i+1,p-1}(x)$, computed on $\mathbf{X}$, is equal to $N_{i,p-1}(x)$ computed on $\mathbf{X}'$. We can thus write the derivative of a B-spline curve as

$$\mathbf{C}'(x) = \sum_{i=0}^{n-1} \left(\bar{\mathbf{P}}_{i+1} - \bar{\mathbf{P}}_i \right) \left(p \frac{N_{i,p-1}(x)}{x_{i+p} - x_i} \right). \tag{3.9}$$

The recursive definition of B-splines makes it possible to write the derivative of a B-spline as a B-spline of one degree lower and new degrees of freedom $\bar{\mathbf{Q}}_i = \bar{\mathbf{P}}_{i+1} - \bar{\mathbf{P}}_i$. We have thus obtained an equation of the derivative in purely discrete terms. As we shall see relation (3.9) is the foundation of our Mimetic Discretization Method.

Note that the Cox-DeBoor recursive formula not only provides the p-degree B-spline basis, but simultaneously the polynomial bases of the $p-1$ continuous derivatives.

3.2.6 Global interpolation in curved coordinates

We shall frequently need to interpolate a given function $\phi(\mathbf{y})$ in curved coordinates x^j, for example to obtain Dirichlet boundary conditions. Because B-splines are not interpolary basis functions, we need to interpolate Dirichlet boundary conditions by solving a system of equations. In the curve case, for a given knot vector $\mathbf{X}$, we need to determine $n+1$ control points. To obtain a solvable system, we need to choose $n+1$ corresponding interpolation points, each in the support of the associated basis function. Several choices are available. For example,

every point is chosen to lie in the middle of the support of the basis functions

$$x^*_{j,p} = \frac{1}{2} \cdot (x_j + x_{j+p+1}) \,. \tag{3.10}$$

The standard interpolation points, however, are the so called Greville abscissa or Schoenberg points [26]. These sites represent the location of the control points in parametric space, see for example figure 3.2, 3.8 and 4.13. The Greville abscissa are given by

$$x^*_{j,p} = \frac{1}{p} \cdot \sum_{k=j+1}^{j+p} x_k. \tag{3.11}$$

Let $\mathbf{y} = \mathbf{C}(x)$ be a parametric map from local coordinate x to global coordinates $\mathbf{y}$, for example a B-spline or NURBS curve. Then the function to be interpolated at the Greville abscissa is $\phi(\mathbf{y}) = \phi(\mathbf{C}(x^*_{j,p}))$. Since we have $n+1$ interpolation points ($j = 0, 1, \ldots, n$) for $n+1$ unknown control points $\bar{\boldsymbol{\phi}}_i$, we have $n+1$ equations for $n+1$ unknowns

$$\sum_{i=0}^{n} \bar{\boldsymbol{\phi}}_i \, N_{i,p}(x^*_{j,p}) = \phi(\mathbf{C}(x^*_{j,p})) \qquad \text{for} \qquad j = 0, 1 \ldots n, \tag{3.12}$$

which is always a square, invertible system, see de Boor [26]. In the case the boundary is a surface, as in three dimensional problems, we could use repeated curve interpolation, see Piegl and Tiller [54].

3.3 Refinement

One of the most powerful aspects of B-splines is the multitude of ways in which the polynomial bases can be enriched, while the underlying geometry description stays untouched.

The fundamental refinement procedures of B-splines are called knot insertion and degree elevation, which are analogous to the h- and p-refinement in FEA respectively. Combination of these two has led to the so called hp-refinement in FEA. In IGA, besides control over the element size and polynomial order of the basis, we can also manipulate the continuity of the basis. This leads to a whole new refinement procedure, referred to as k-refinement.

Because the geometry is exactly represented at the coarsest level of discretization, the mesh and corresponding basis can be refined and order elevated without interaction with the CAD program.

3.3.1 Knot insertion - h-refinement

The most fundamental way of refining the polynomial basis is by means of knot insertion, which provides control over element size. Knot insertion is similar to h-refinement in FEA,

however it is much more general since continuity in between elements can be regulated. As the term *knot insertion* implies, h-refinement is achieved by inserting new knots into the existing knot vector. Inserting a single knot, locally changes the bases, and calculation of the new degrees of freedom, such that the original geometry is maintained is easy. To insert a knot t, we first determine the knot span $\{x_k, x_{k+1}\}$ that contains this new knot. Once k is determined, the p new control points $\bar{\mathbf{Q}}_{k-p+1}$ to $\bar{\mathbf{Q}}_k$ are calculated as a function of the old control points $\bar{\mathbf{P}}_i$ [54]

$$\bar{\mathbf{Q}}_i = (1 - a_i)\,\bar{\mathbf{P}}_{i-1} + a_i \cdot \bar{\mathbf{P}}_i, \tag{3.13}$$

where the ratio a_i is defined as

$$a_i = \frac{t - x_i}{x_{i+p} - x_i} \qquad \text{for} \qquad k - p + 1 \leq i \leq k.$$

Figure 3.12 illustrates the insertion of a single knot t at parameter value $x = 2.5$. This cubic degree curve is constructed using the knot vector $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$, so t lies in the local knot span $\{x_5 = 2, x_6 = 3\}$. This means we need to recalculate new control points $\mathbf{Q}_3$ to $\mathbf{Q}_5$. The ratio's a_i are determined as

$$a_3 = \frac{t - x_3}{x_6 - x_3} = \frac{2.5 - 0}{3 - 0} = 5/6, \quad a_4 = \frac{t - x_4}{x_7 - x_4} = \frac{2.5 - 1}{4 - 1} = 1/2, \quad a_5 = \frac{t - x_5}{x_8 - x_5} = \frac{2.5 - 2}{4 - 2} = 1/4$$

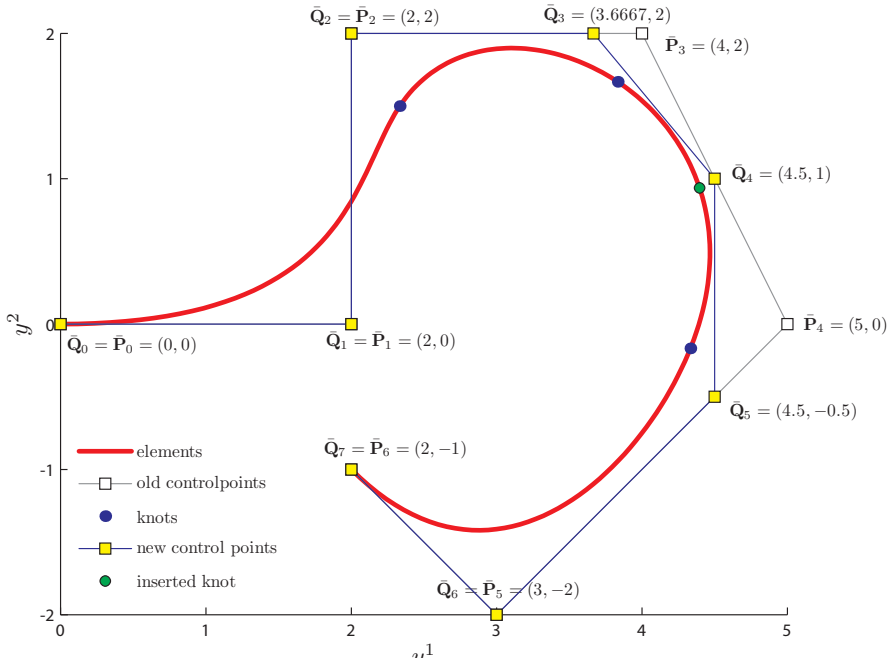
The new control points $\bar{\mathbf{Q}}_i$ can subsequentally be calculated using 3.13

$$\begin{aligned}
&\bar{\mathbf{Q}}_0 = \bar{\mathbf{P}}_0, \quad \bar{\mathbf{Q}}_1 = \bar{\mathbf{P}}_1, \quad \bar{\mathbf{Q}}_2 = \bar{\mathbf{P}}_2, \\
&\bar{\mathbf{Q}}_3 = (1 - 5/6)\,\bar{\mathbf{P}}_2 + 5/6 \cdot \bar{\mathbf{P}}_3 = 1/6 \cdot \begin{pmatrix} 2 & 2 \end{pmatrix} + 5/6 \cdot \begin{pmatrix} 4 & 2 \end{pmatrix} = \begin{pmatrix} 3\frac{2}{3} & 2 \end{pmatrix} \\
&\bar{\mathbf{Q}}_4 = (1 - 1/2)\,\bar{\mathbf{P}}_3 + 1/2 \cdot \bar{\mathbf{P}}_4 = 1/2 \cdot \begin{pmatrix} 4 & 2 \end{pmatrix} + 1/2 \cdot \begin{pmatrix} 5 & 0 \end{pmatrix} = \begin{pmatrix} 4\frac{1}{2} & 1 \end{pmatrix} \\
&\bar{\mathbf{Q}}_5 = (1 - 1/4)\,\bar{\mathbf{P}}_4 + 1/4 \cdot \bar{\mathbf{P}}_5 = 3/4 \cdot \begin{pmatrix} 5 & 0 \end{pmatrix} + 1/4 \cdot \begin{pmatrix} 3 & -2 \end{pmatrix} = \begin{pmatrix} 4\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \\
&\bar{\mathbf{Q}}_6 = \bar{\mathbf{P}}_5, \quad \bar{\mathbf{Q}}_7 = \bar{\mathbf{P}}_6
\end{aligned}$$

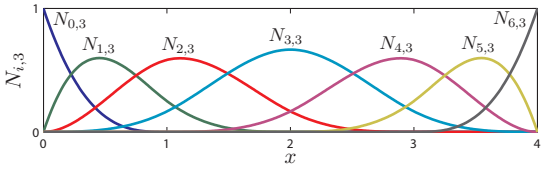
Insertion of the single knot t at $x = 2.5$ has resulted in 1 more element and one extra basis function and associated degree of freedom. Although splitting up the third element in two has resulted in a new and richer basis, the new control points are such that the curve is geometrically identical to the original.

The above process may be repeated to obtain a refined basis with control over the element size, by manipulating the spacing between knots, and direct control over the continuity, by adding existing knots. Knot insertion is thus much more general than h-refinement in FEA.

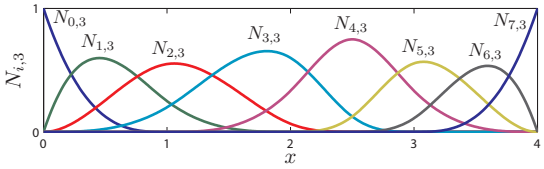
Interesting to note is that besides providing a simple and elegant way of refinement, knot insertion is ideal in geometric multi grid solvers, see for example Hollig et al. [40] and Hollig [39].



(a) Curve with old and new control polygon



(b) Original Basis



(c) Refined Basis

Figure 3.12: Knot insertion for a cubic degree B-spline curve built with knot vector $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$. Inserting a new knot, $\mathbf{X}_{ref} = \{0, 0, 0, 0, 1, 2, 2.5, 3, 4, 4, 4, 4\}$, changes the basis locally, see Figure 3.12(b) and 3.12(c). The new control points are calculated such that the geometry does not change, Figure 3.12(a)

3.3.2 Degree elevation - p-refinement

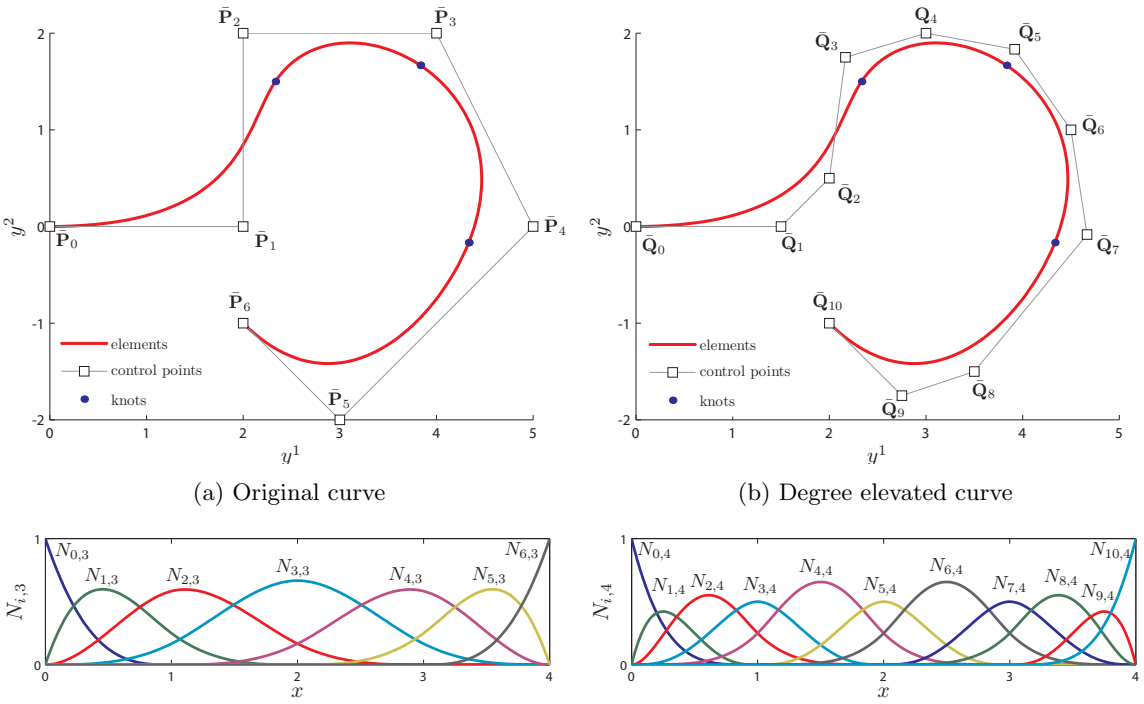
Degree elevation increases the degree of the polynomial basis, without changing the geometry. In order to maintain the degree of continuity in the derivatives of the B-spline, the continuity must stay the same while raising the degree. To achieve this, every unique knot in the original knot vector has to be repeated. For example

$$\mathbf{X}_{original} = \{0, 0, 0, 1, 2, 3, 3, 4, 4, 4\} \overset{\text{degree elevate}}{\Longrightarrow} \mathbf{X}_{new} = \{0, 0, 0, 0, 1, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4\}$$

$\mathbf{X}_{new}$ is the knot vector obtained after degree elevating the basis associated with $X_{original}$ from degree two to three.

Figure 3.13 shows an example where the cubic B-spline curve from the previous section is

degree elevated to degree four. This results into four extra fourth-degree basis functions and associated control points, while the partitioning in elements remains unaltered. Clearly, degree



(a) Original curve

(b) Degree elevated curve

(c) Original basis: $\mathbf{X} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$

(d) Degree elevated basis: $\mathbf{X}_{new} = \{0, 0, 0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4, 4, 4\}$

Figure 3.13: Degree elevation of B-spline curve. Note the knot multiplicities in the knot vector. This is necessary in order to maintain the degree of continuity in the derivatives of the curve.

elevation is similar to conventional p-refinement. However, where p-refinement only raises the degree of C^0 continuous finite elements, degree elevation works for arbitrary continuity of the original basis, C^0 to C^{p-1}. Degree elevation is thus a generalization of p-refinement in FEA.

Degree elevation is more involved than knot insertion. Most algorithms start by decomposing a B-spline into its Bezier segments by inserting existing knots (a Bezier curve is equivalent to a one element B-spline curve). Subsequently, the degree of each polynomial element is increased. Finally, all polynomial elements are sewed together by removing excess knots, in order to produce a single order elevated B-spline curve.

Unlike knot insertion, degree elevation is a global process. Recently multi-degree B-spline schemes have been developed [61, 64], which allow a B-spline to consist of polynomial elements of different degree. In such a B-spline representation, degree elevation could be used as a local procedure.

3.3.3 Increased order and continuity - k-refinement

In FEA, the order in which h- and p-refinement are combined, does not lead to differences in the refined polynomial basis. The order in which degree elevation and knot insertion are combined, however, does lead to different results. Degree elevation and knot insertion do not commute. This leads to a whole new type of refinement, referred to as k-refinement [42, 22], of which no analogue exist in FEA. In k-refinement both order and continuity of the basis is increased. This is achieved by first order elevating the coarse mesh and subsequentally performing knot insertion. Because knot insertion is performed after the degree elevation process, the refined basis consist of basis functions with both increased order and continuity, see Figure 3.14.

k-refinement is potentially more powerful than both knot insertion and degree elevation, because the order can be increased without an excessive increase in degrees of freedom. Let $\mathcal{P}$ be a basis for the 1D vector space of all polynomials with given continuity constraints in between a partitioning of m elements. In classical FEA, C^0 continuity is imposed between polynomial elements and the dimensionality of the vector space $\mathcal{P}$ is

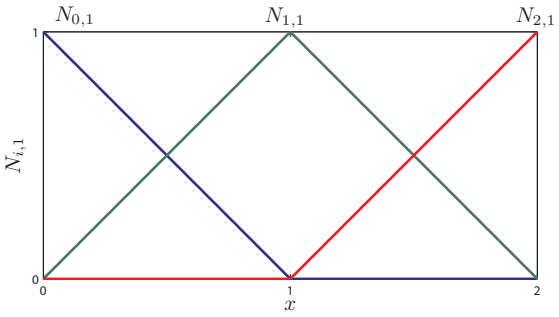
$$\dim(\mathcal{P}) = m(p+1) - (m-1)\cdot 1 = m\cdot p + 1, \tag{3.14}$$

where p stands for the polynomial degree. Raising the order of the basis by one degree results in an increase in m degrees of freedom. On the other hand if we impose maximum continuity between elements, then the dimensionality of $\mathcal{P}$ is given by

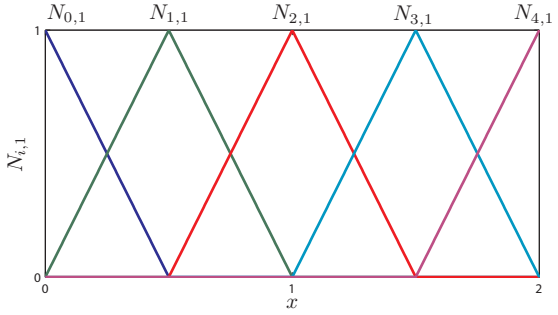
$$\dim(\mathcal{P}) = m(p+1) - (m-1)\cdot p = m + p \tag{3.15}$$

If we increase the polynomial order of the basis now we obtain one extra degree of freedom. In higher dimensional space the differences become even more pronounced. The latter is pure k-refinement. In practice pure k-refinement is never achieved since the continuity at the knots in the original knot vector can not be raised since this would alter the geometry. Nevertheless, the effect of k-refinement might still be significant since the initial coarse mesh from CAD is generally much coarser than the refined mesh required in IsoGeometric Analysis. Figure 3.14 shows the relative differences between hp-refinement in FEA and the novel k-refinement in IGA.

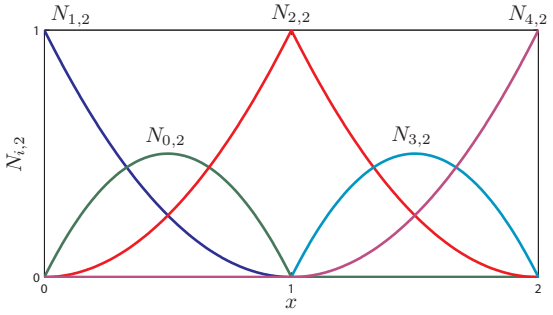
For a more elaborate discussion on B-spline knot insertion and degree elevation, we refer the reader to Piegl and Tiller [54]. k-refinement is explained in detail in Hughes et al. [42] and Cottrell et al. [22]. Currently the most efficient, yet simple, algorithm, which also allows for simultaneous degree elevation and knot insertion, has been developed by Hua [1].



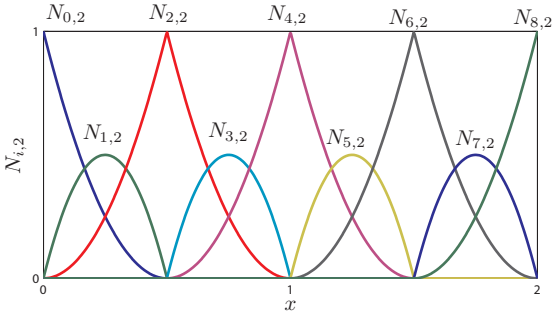
(a) original basis: $\mathbf{X} = \{0, 0, 1, 2, 2\}$



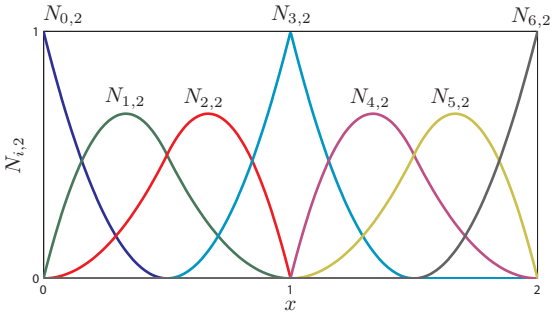
(b) h-ref: $\mathbf{X} = \{0, 0, 0.5, 1, 1.5, 2, 2\}$



(c) p-ref: $\mathbf{X} = \{0, 0, 0, 1, 1, 2, 2, 2\}$



(d) hp-ref: $\mathbf{X} = \{0, 0, 0, 0.5, 0.5, 1, 1, 1.5, 1.5, 2, 2, 2\}$



(e) k-ref: $\mathbf{X} = \{0, 0, 0, 0.5, 1, 1, 1.5, 2, 2, 2\}$

Figure 3.14: The original basis of degree 1, Figure 3.14(a), is refined using h-refinement, Figure 3.14(b), and p-refinement in Figure 3.14(c). Figure 3.14(d) shows the standard FEA approach of hp-refinement, where the basis is obtained after h-refinement and subsequent p-refinement. First elevating the degree in the original basis to $p = 2$ (3.14(c)) and subsequent h-refinement leads to the concept of k-refinement, Figure 3.14(e). Compare Figure 3.14(d) with 3.14(e) and observe that the increased smoothness of the basis leads to fewer basis functions.

3.4 NURBS - Non-Uniform Rational B-splines

Fundamental geometric objects like the circle, ellipse, cylinder and sphere are frequently encountered in engineering applications and other fields. It is thus of vital importance that the CAD-description can exactly represent these type of geometries. Polynomials fail in this respect. NURBS, which are rational functions of B-splines, can however exactly represent these conic sections. Moreover, they generalize B-splines and consequently inherit all their favorable properties for use in free form geometric design.

3.4.1 NURBS curves

A NURBS curve, can be seen as a weighted B-spline curve and is generated as follows

$$\mathbf{C}(x) = \frac{\mathbf{C}^w(x)}{W(x)} = \frac{\sum_{i=0}^{n} \bar{\mathbf{P}}_i \; \bar{w}_i \; N_{i,p}(x)}{\sum_{i=0}^{n} \bar{w}_i \; N_{i,p}(x)}. \tag{3.16}$$

The $\bar{w}_i$ are called the weights. If these are set to 1, the NURBS curve reduces to a B-spline. Note that $\mathbf{C}^w(x)$ is a vector valued and $W(x)$ a scalar valued B-spline function.

Equivalent to (3.16), we could write the NURBS curve as a linear combination of control points $\bar{\mathbf{P}}_i$ and NURBS basis functions $N^r_{i,p}$

$$\mathbf{C}(x) = \sum_{i=0}^{n} \bar{\mathbf{P}}_i \; N^r_{i,p}(x) \qquad \text{where} \qquad N^r_{i,p}(x) = \frac{\bar{w}_i \; N_{i,p}(x)}{W(x)} \tag{3.17}$$

The basis functions $N^r_{i,p}$ are not polynomial, but rational functions. They inherit the continuity across knots, the local support, and the positiveness of B-spline basis functions. Furthermore, they always form a partition of unity, since

$$\sum_{i=0}^{n} N^r_{i,p}(x) = \frac{\sum_{i=0}^{n} \bar{w}_i \; N_{i,p}(x)}{\sum_{i=0}^{n} \bar{w}_i \; N_{i,p}(x)} = 1$$

Figure 3.15 illustrates the effect of the weights. The curve is a quarter circle constructed from quadratic NURBS, where the weight of the middle control point is set to half the square root of two. Observe that the middle control point pulls the curve less strong as in the case of a B-spline curve (the dotted line).

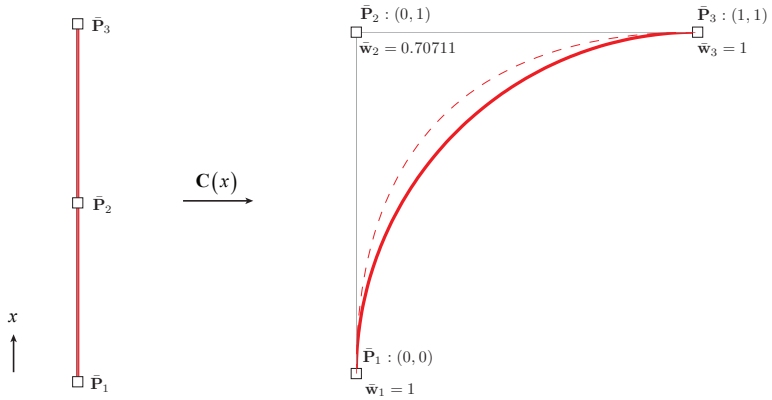


Figure 3.15: A NURBS quarter circle constructed from quadratic basis functions and knot vector $\mathbf{X} = \{0,0,0,1,1,1\}$. The dotted line denotes the quadratic curve with weights all equal to one.

3.4.2 NURBS surfaces

Similar to (3.16), a NURBS surface can be expanded in terms of control points, B-spline basis functions and weights as

$$\mathbf{S}(x^1,x^2) = \frac{\mathbf{S}^w(x^1,x^2)}{W(x^1,x^2)} = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} \bar{\mathbf{P}}_{ij}\ \bar{w}_{ij}\ N_{i,p}(x^1)N_{j,q}(x^2)}{\sum_{i=0}^{n}\sum_{j=0}^{m} \bar{w}_{ij}\ N_{i,p}(x^1)N_{j,q}(x^2)}. \tag{3.18}$$

Where $\mathbf{S}^w(x^1,x^2)$ is a vector valued B-spline and $W(x^1,x^2)$ a scalar valued B-spline function.

Figure 3.16 illustrates how a NURBS surface is mapped from parameter space to the circular domain in physical space. The control points and weights have been set such that each side of the patch maps to a quarter circle in physical space. Hence the circular domain

Next we shall discuss a brief example of a multi patch geometry that we will use in analysis in a subsequent chapter. The geometry is given by the rectangular domain containing a circular hole, depicted in figure 3.17(a). The coarse mesh is modeled in a CAD program using four bi-quadratic NURBS patches. Each patch is mapped from parameter space to physical space using

$$\mathbf{S}(x^1,x^2) = \frac{\mathbf{S}^w(x^1,x^2)}{W(x^1,x^2)} = \frac{\sum_{i=0}^{2}\sum_{j=0}^{2} \bar{\mathbf{P}}_{ij}\ \bar{w}_{ij}\ N_{i,2}(x^1)N_{j,2}(x^2)}{\sum_{i=0}^{2}\sum_{j=0}^{2} \bar{w}_{ij}\ N_{i,2}(x^1)N_{j,2}(x^2)}.$$

Figure 3.17(b) shows the control mesh and corresponding weights of the upper left patch.

In order to obtain an Analysis Suitable Geometry (ASG) we need to refine and order elevate the basis of each patch. The basis of the NURBS patch $\mathbf{S}(\mathbf{x})$ can be refined by refining the

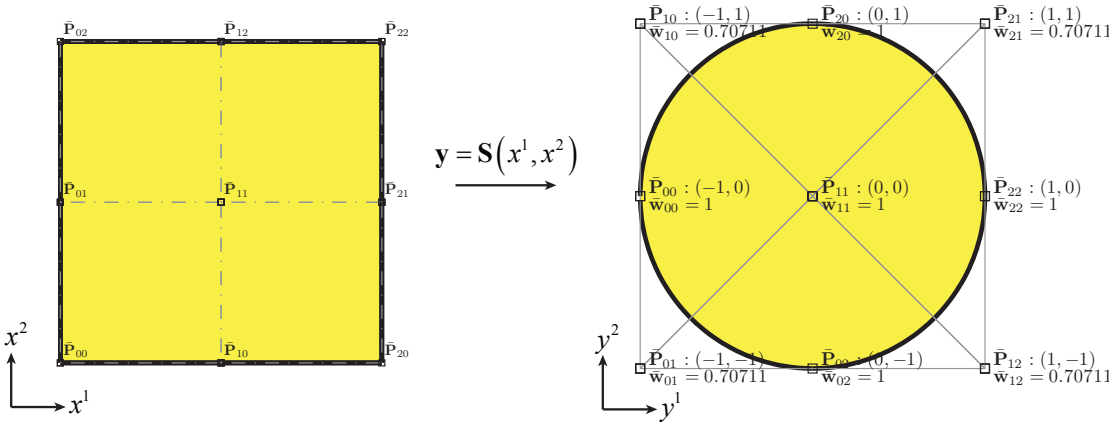
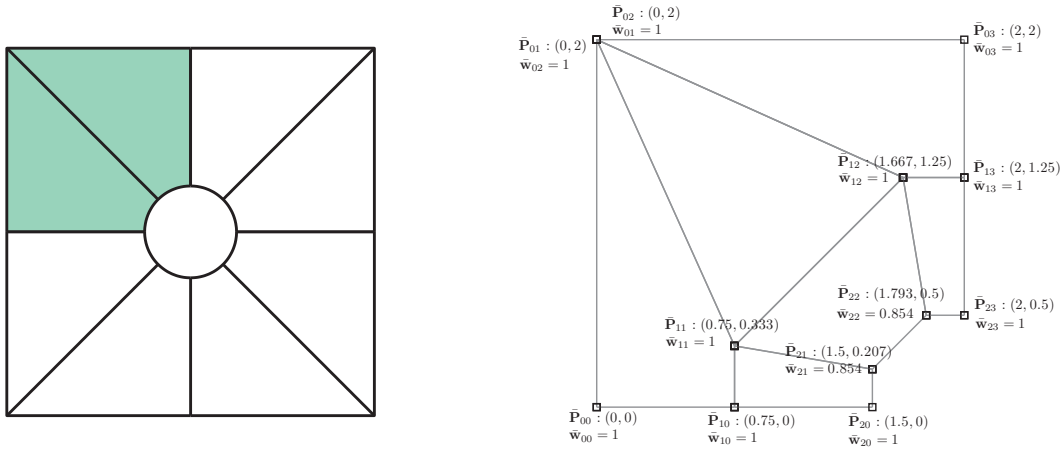


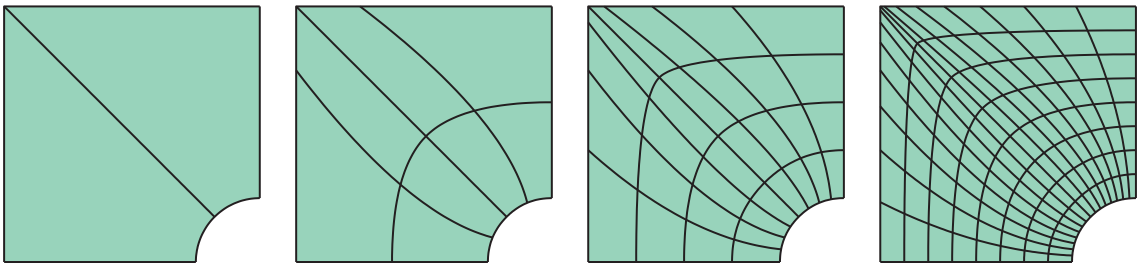
Figure 3.16: A NURBS surface is mapped from the parametric domain, to the circular plane in physical space. The bi-quadratic surface is constructed using the knot vectors $\mathbf{X}^1 = \mathbf{X}^2 = \{0, 0, 0, 1, 1, 1\}$ and control points and corresponding weights shown.



(a) coarse mesh (b) control mesh

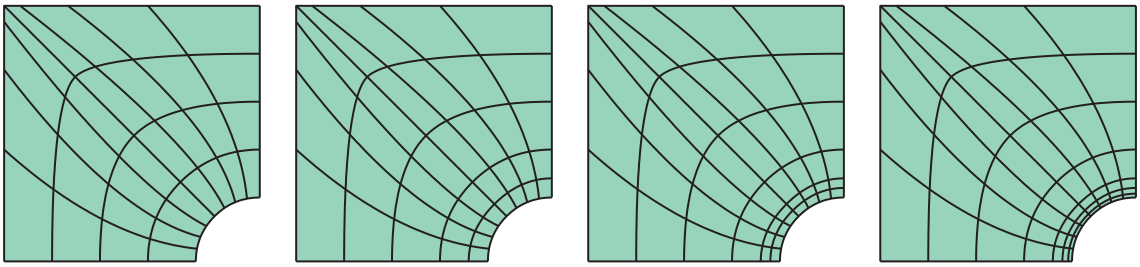
Figure 3.17: Figure 3.17(a) shows the coarse mesh of a rectangular domain with circular hole modeled using four patches in a CAD program. The NURBS geometry is bi-quadratic and build from knot vectors $\mathbf{X^1} = \{0, 0, 0, 1, 1, 1\}$ and $\mathbf{X^2} = \{0, 0, 0, 0.5, 1, 1, 1\}$ and control points and weights from Figure 3.17(b). Note that the two control points, $\bar{\mathbf{P}}_{01}$ and $\bar{\mathbf{P}}_{02}$, have the same location and create the artificial upper left corner. Although this causes a singular point at this location, this poses no problem in IGA [42].

respective B-spline functions $\mathbf{S}^w(\mathbf{x})$ and $W(\mathbf{x})$ as we have learned in the previous sections. Note that in this process both $\mathbf{S}^w(\mathbf{x})$ and $W(\mathbf{x})$ do not change geometrically. An example of global refinement is shown in Figure 3.18 and an example of single knot insertion near the boundary is shown in Figure 3.19. Note that knot insertion splits up a hole row or column of elements. Knot insertion can not be achieved locally due to the tensor product structure of NURBS.



(a) mesh 1 (b) mesh 2 (c) mesh 3 (d) mesh 4

Figure 3.18: The coarse mesh of the domain with circular hole is globally refined for analysis. Meshes produced using global h-refinement, where element size is halved each iteration.



(a) mesh 3 (b) x = 0.90 (c) x = 0.95 (d) x = 0.98

Figure 3.19: Mesh 3 in Figure 3.18 is more locally refined near the boundary of the hole.

Besides the algebraic definition of NURBS as presented here, NURBS also allow an intuitive geometrical interpretation. NURBS can be seen as B-splines in one extra dimension, projected onto a plane. Farin [32] discusses the geometric nature of NURBS extensively. We note that although NURBS is currently the standard technology applied in CAD, other technologies like trigonometric B-splines [49] would similarly provide an excellent basis.

3.4.3 Derivatives of NURBS

Because NURBS are rational functions of B-splines, derivatives of NURBS will clearly depend on the B-splines and their derivatives. NURBS derivatives are easily calculated by use of the the quotient rule. If $\mathbf{C}(x) = \frac{\mathbf{C}^w(x)}{W(x)}$ is a NURBS curve, then it's derivative is given by

$$\frac{\partial \mathbf{C}(x)}{\partial x} = \frac{1}{(W(x))^2}\left(W(x)\cdot\frac{\partial \mathbf{C}^w(x)}{\partial x} - \frac{\partial W(x)}{\partial x}\cdot\mathbf{C}^w(x)\right). \tag{3.19}$$

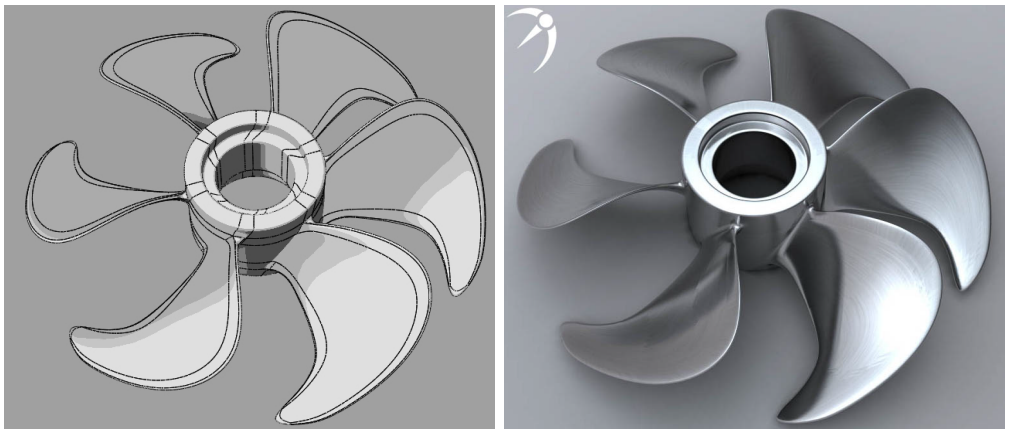
A similar expression is obtained for the partial derivatives of a NURBS surface $\mathbf{S}(x^1,x^2) = \frac{\mathbf{S}^w(x^1,x^2)}{W(x^1,x^2)}$

$$\frac{\partial \mathbf{S}(x^1,x^2)}{\partial x^j} = \frac{1}{(W(x^1,x^2))^2}\left(W(x^1,x^2)\cdot\frac{\partial \mathbf{S}^w(x^1,x^2)}{\partial x^j} - \frac{\partial W(x^1,x^2)}{\partial x^j}\cdot\mathbf{S}^w(x^1,x^2)\right). \tag{3.20}$$

These expressions might seem complex, they consist however out of B-splines and their derivatives, which are easy to compute. Efficient algorithms for evaluating NURBS and their kth order derivatives can be found in Piegl and Tiller [54].

3.5 Current limitations of NURBS as a basis for geometry and mesh generation

Since B-spline patches are topologically square, they allow for modeling of simple geometries only. So it can take a significant amount of patches to construct a complicated geometry. Figure 3.20(a) shows an example of the CAD-model of a ships propeller, modeled using 72 patches. Although continuity within each patch is automatically guaranteed due to the build-in continuity of B-spline and NURBS bases, maintaining continuity between patches is still an important issue. Constraints need to be set, in order for the geometry to have some level of smoothness across patch interfaces. Figure 3.20(b) shows a computer render of the CAD model of the ships propeller, where these continuity constraints have been set up correctly. The surface is curvature continuous across patch interfaces. Recently new surface schemes like



(a) (b)

Figure 3.20: A CAD model of a ships propeller can be quite a complicated geometry to model. The surface description in Figure 3.20(a), consists out of 72 patches. Continuity constraints between patches make sure that the overall surface is smooth, see Figure 3.20(b).

manifold T-splines [60, 37] and manifold triangular B-splines [25, 36] have been developed, which allow the modeling of any complex shape, using just a single patch. Continuity is thus automatically guaranteed, alleviating the cumbersome process of imposing continuity constraints at patch interfaces. These surface schemes are likely to have a major influence not only on CAD, but also on IsoGeometric Analysis.

The second major problem concerning NURBS as a basis for geometric design follows from its tensor product structure. NURBS do not allow local refinement. Refinement is achieved by knot insertion which introduces a whole new row or column of control points. This has been one of the major drawbacks of NURBS, both as a basis for CAD and FEA. More recently

T-splines have been introduced by Sederberg et al. [60] to alleviate this problem. Knots are allowed to terminate at T-sections. T-splines are however tailor made for geometric design and when applied in an analysis environment several problems arise. T-splines can loose their linear independence and refinement is not that local as it seems [11, 30]. Hierarchical B-splines [46, 47] and LR-splines [29] seem to be more suitable in analysis. Linear independence of the basis is automatically guaranteed and knot insertion can be done locally, as in T-splines, and leads to the minimum number of inserted control points.

The major drawback of CAD-based IsoGeometric Analysis, at this moment however, is volumetric mesh generation [22]. Currently, CAD programs allow modeling of parametric curves and surfaces only. Volumes are modeled by definition of their bounding surfaces. Since IsoGeometric Analysis in three dimensional space requires the parameterization of a volume, we are faced with a problem. The coarse mesh cannot be supplied by a CAD program. This is the single most important problem, hampering the application of volumetric IsoGeometric Analysis, at this time. Generation of a computational mesh in 3D can be as much work as in conventional approaches. Possible solutions to this problem are available. The T-spline sewing algorithm, in Sederberg et al. [62], could be generalized to 3D. Triangular B-splines, introduced by Dahmen et al. [25] and more recently Neamtu [53] would form a convenient basis, since there exist efficient algorithms in parameterizing volumes using tetrahedra.

Chapter 4

IsoGeometric Mimetic Discretization Methods

In this chapter we combine IsoGeometric Analysis with Mimetic discretization methods. The resulting discretization approach resembles a finite volume type of discretization for the conservation laws on a staggered grid, and a finite element type of discretization for the constitutive equations. Besides inheriting all the advantages of the IsoGeometric Analysis concept, this scheme is locally conservative, naturally stable and of arbitrary order.

4.1 Introduction

Every discretization method starts with reconstructing the physical field variables into a finite dimensional continuous representation. The Finite Difference Method reconstruct quantities in the form of a Taylor series expansion; the Finite Volume Method uses a Taylor series expansion in combination with an integral approach; and the Finite Element Method uses an expansion in terms of basis functions. The latter more easily deals with curved geometries, can evaluate a given quantity at arbitrary parameter values and is easily generalized to higher order.

In this thesis, B-splines are applied to connect the continuous world of Differential Geometry with its discrete counterpart Algebraic Topology. B-splines and NURBS have been introduced in a practical setting in the context of geometry and mesh generation in chapter 3. We have seen that there are two different definitions of the NURBS mesh. The physical mesh, which represents the geometry, and the control mesh, which controls the underlying geometry. The control mesh, accompanied by the NURBS geometry from a CAD program, gives us an initial representation of a primal cell complex, see Figure 4.1. The NURBS *control points* P_i can be regarded as 0-cells. These are logically connected by *control edges* L_i (1-cells). We can further distinguish discrete *control faces* S_i (2-cells) enclosed by four edges. In three-

dimensional space the collection of six boundary control faces would form a discrete *control volume* V_i(3-cell).

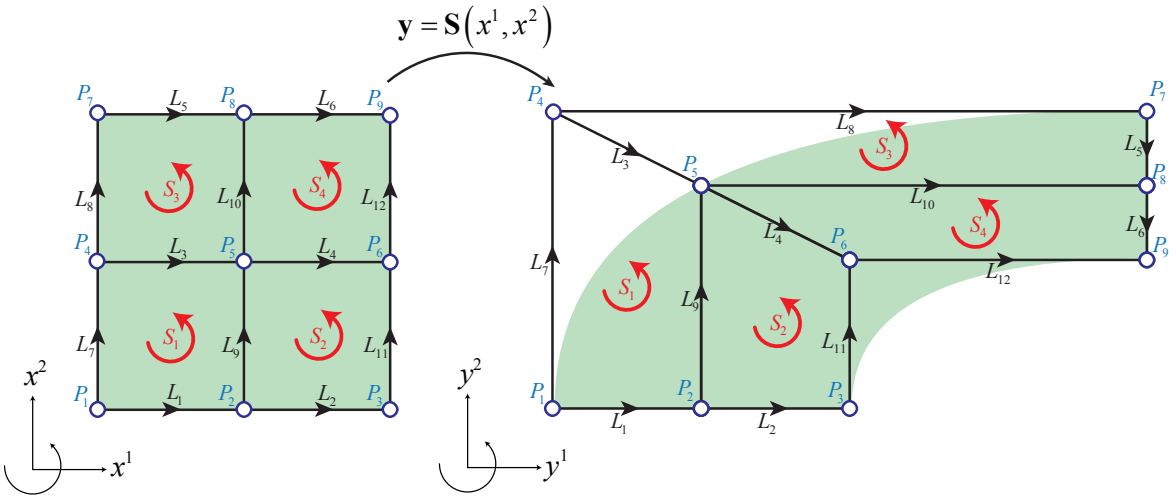


Figure 4.1: The NURBS control mesh can be seen as a primal cell complex consisting of control points P_i, control edges L_i and control faces S_i. The control points, edges and faces have been given the inner orientation.

Since B-spline basis functions are naturally associated with point values - the control points - they appear as an appropriate basis in which to represent 0-forms. From B-splines we shall deduce basis functions which are naturally related with control edges, providing a basis for 1-forms. These edge basis functions appear simply to be scaled B-splines, known as the Curry Schoenberg B-splines. By application of the wedge product we can easily construct basis functions which are naturally associated with control faces (2-forms) and in 3D with control volumes (3-forms). These spaces are equivalent to those developed by [20, 19, 17, 57, 6]. In Buffa et al. [19] they show that these spaces are inf-sup-stable for Stokes Flow.

As we shall see, B-spline differential forms follow a discrete DeRahm sequence both in reduced discrete as in reconstructed continuous form and hence provide a natural basis for Mimetic Discretization Methods. A strong indication exists that these methods naturally lead to stable consistent approximations to any physical problem.

4.2 Spline differential forms in 1D

In this section we cover the fundamental properties of the B-spline basis which make them suitable in Mimetic Methods. The main recipe is that B-splines follow the DeRahm sequences both in reduced discrete form as well as in reconstructed continuous form. The theory will be developed in the one dimensional setting. Generalization to higher dimensional space is

subsequently straightforward. In short, the aim of this section is to define the spaces and operators in the following commuting diagram,

$$\begin{array}{ccccccccc}
\mathbb{R} & \longrightarrow & \Lambda^0(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^1(\Omega) & \longrightarrow & 0 \\
 & & \downarrow \mathcal{R}_0 & & \downarrow \mathcal{R}_1 & & \\
\mathbb{R} & \longrightarrow & \mathcal{C}^{(0)} & \xrightarrow{\delta} & \mathcal{C}^{(1)} & \longrightarrow & 0 \\
 & & \downarrow \mathcal{I}_0 \;\uparrow \mathcal{R}_0 & & \downarrow \mathcal{I}_1 \;\uparrow \mathcal{R}_1 & & \\
\mathbb{R} & \longrightarrow & \mathcal{S}^0_{p,\mathbf{X}} & \xrightarrow{\mathrm{d}} & \mathcal{S}^1_{p-1,\mathbf{X}} & \longrightarrow & 0
\end{array} \tag{4.1}$$

The spaces to be constructed should thus follow

$$\delta\mathcal{R}_0 = \mathcal{R}_1\mathrm{d} \qquad \text{and} \qquad \mathrm{d}\mathcal{I}_0 = \mathcal{I}_1\delta \tag{4.2}$$

As explained in Chapter 2, Section 2.4, we further require that

$$\mathcal{R}_k\mathcal{I}_k\mathcal{C}^{(k)} = \mathcal{C}^{(k)} \qquad \text{and} \qquad \mathcal{I}_k\mathcal{R}_k\Lambda^k(\Omega) = \mathcal{S}^k_{d,\mathbf{X}} = \Lambda^k(\Omega) + O(h^{d+1}) \tag{4.3}$$

where h is a measure of the mesh size and $d+1$ the polynomial order of accuracy. Note that (4.3) also depends on the function to be approximated.

As is customary in IGA, we use the same partitioning for the definition of the geometry as for the reconstructed spaces of differential forms. The domain partitioning is thus provided by the coarse p-degree NURBS geometry and shall be denoted as knot vector $\mathbf{X}$

$$\mathbf{X} = \{ \underbrace{a, ..., a}_{p+1}, x_{p+1}, ..., x_i, x_{i+1}, ..., x_{m-p-1}, \underbrace{b, ..., b}_{p+1} \}, \qquad x_{i+1} \geq x_i \tag{4.4}$$

$\mathbf{X}$ shall be used to define the univariate spline spaces of differential 0 and 1-forms, $\mathcal{S}^0_{p,\mathbf{X}}$ and $\mathcal{S}^1_{p-1,\mathbf{X}}$ respectively. These spaces can subsequentally be order elevated and refined for analysis purposes as desired.

4.2.1 Reconstruction of 0-forms

Since B-splines are naturally associated with point values - the control points - they appear as a convenient basis in which to represent 0-forms. Given a 0-cochain $\bar{\varphi} \in \mathcal{C}^{(0)}$ - containing a set of B-spline control points $\bar{\varphi}_i$- we can reconstruct a continuous representation of a 0-form

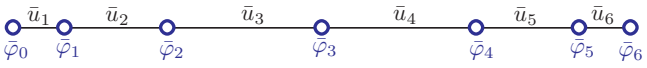
$\phi_h^0(x) \in \mathcal{S}^0_{p,\mathbf{X}}$ by taking a linear combination of control points $\bar{\varphi}_i$ and B-spline basis functions $N_{i,p}(x)$,

$$\phi_h^0(x) = \mathcal{I}_0 \bar{\phi} = \sum_{i=0}^{n} \bar{\varphi}_i \; N_{i,p}(x) \tag{4.5}$$

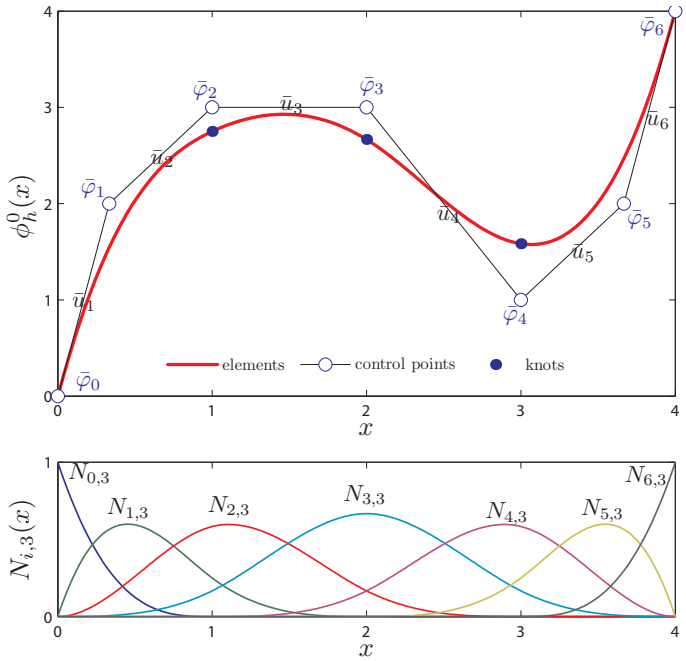
The B-spline basis functions are defined on knot vector $\mathbf{X}$. The univariate space of reconstructed 0-forms $\mathcal{S}^0_{p,\mathbf{X}} = \mathcal{I}_0 \mathcal{R}_0 \Lambda^0(\Omega)$ can be defined as the span of all p-degree B-spline basis functions $N_{i,p}(x)$,

$$\mathcal{S}^0_{p,\mathbf{X}} := \text{span}\,\{N_{i,p}(x)\}_{i=0}^{n} \tag{4.6}$$

Figure 4.2 depicts a B-spline control mesh which can be thought of as a 1D primal cell complex K. The knot vector $\mathbf{X} = \{0,0,0,0,1,2,3,4,4,4,4\}$ accompanying the NURBS geometry is used to define the cubic degree B-spline basis for zero-forms. Using this basis, a 0-form is reconstructed for a given 0-cochain on the primal cell complex K.



(a) 1D primal cell complex K, associated with the cubic degree B-spline geometry partitioned by knot vector $\mathbf{X} = \{0,0,0,0,1,2,3,4,4,4,4\}$



(b) Reconstructing a 0-form: $\phi_h^0(x) = \mathcal{I}_0 \bar{\varphi} = \sum_{i=0}^{6} \bar{\varphi}_i \; N_{i,3}(x)$

Figure 4.2: Given a zero cochain $\bar{\varphi}$ we can reconstruct a continuous representation of a 0-form $\phi_h^0(x) \in \mathcal{S}^0_{p,\mathbf{X}}$ by taking a linear combination of B-spline control points $\bar{\varphi}_i$ and B-spline basis functions $N_{i,p}(x)$.

4.2.2 Reconstruction of 1-forms

For the reconstruction of 1-forms we seek a linear combination of coefficients and basis functions which are naturally associated with edges. This reconstruction is required to follow the commutative diagram,

$$\begin{array}{ccc} \bar{\varphi} \in \mathcal{C}^{(0)} & \xrightarrow{\quad\delta\quad} & \bar{u} \in \mathcal{C}^{(1)} \\ \Big\downarrow \mathcal{I}_0 & & \Big\downarrow \mathcal{I}_1 \\ \phi_h^0(x) \in \mathcal{S}^0_{p,\mathbf{X}} & \xrightarrow{\quad\mathrm{d}\quad} & u_h^1(x) \in \mathcal{S}^1_{p-1,\mathbf{X}} \end{array} \tag{4.7}$$

for a 1-cochain $\bar{u} \in \mathcal{C}^{(1)}$ which is the co-boundary of $\bar{\varphi}$ we obtain the following relation for the coefficients $\bar{u}_i$

$$\bar{u} = \delta\bar{\varphi} \qquad \Longrightarrow \qquad \bar{u}_i = \bar{\varphi}_i - \bar{\varphi}_{i-1} \qquad \text{for} \qquad i = 1, \dots, n$$

The co-boundary process is illustrated in Figure 4.3(a). Observe that the coefficients $\bar{u}_i$ can be associated with edges, because $\bar{\varphi}_i - \bar{\varphi}_{i-1}$ represent the difference between subsequent control points $\bar{\varphi}_i$. The 1-form $u_h^0(x)$ which is the exterior derivative of the 0-form $\phi_h^0(x)$ should thus be an expression of the form,

$$u_h^1(x) = \mathcal{I}_1\bar{u} = \sum_{i=1}^{n} \bar{u}_i \; M_{i,p-1}(x)\mathrm{d}x \qquad \text{and} \qquad \bar{u}_i = \bar{\varphi}_i - \bar{\varphi}_{i-1}$$

Where the $M_{i,p-1}(x)$ are new basis functions of degree $p-1$ defined along edges. Using the commutative property (4.7) and the fact that the derivative of a B-spline can be written as a B-spline of one order lower (See Chapter 3, page 65) we obtain an expression of the edge basis functions $M_{i,p-1}(x)$ in terms of the B-spline basis functions $N_{i,p-1}(x)$,

$$\begin{aligned} \mathcal{I}_1\delta\bar{\varphi} = \mathrm{d}\mathcal{I}_0\bar{\varphi} = \mathrm{d}\sum_{i=0}^{n} \bar{\varphi}_i \; N_{i,p}(x) \\ \sum_{i=1}^{n} (\bar{\varphi}_i - \bar{\varphi}_{i-1}) \; M_{i,p-1}(x)\mathrm{d}x = \sum_{i=1}^{n} (\bar{\varphi}_i - \bar{\varphi}_{i-1}) \left(p\frac{N_{i,p-1}(x)}{x_{i+p} - x_i} \right) \mathrm{d}x \end{aligned} \tag{4.8}$$

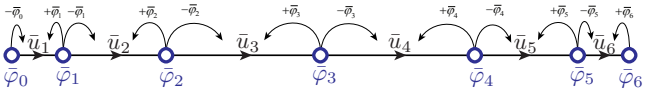
The new edge type of basis function of degree $p-1$ are thus defined as

$$M_{i,p-1}(x) := c_i \cdot N_{i,p-1}(x) \qquad \text{where} \qquad c_i = \frac{p}{x_{i+p} - x_i} \tag{4.9}$$

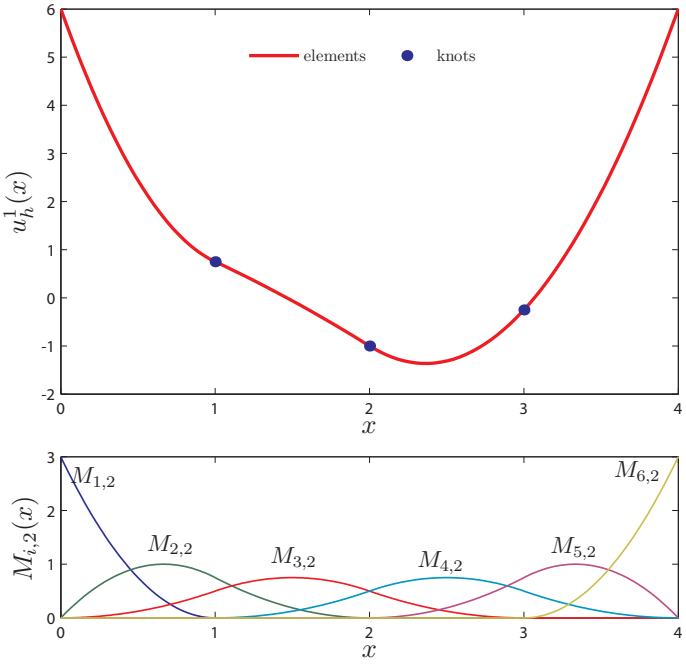
where the B-spline basis functions $N_{i,p-1}(x)$ of degree $p-1$ are defined on knot vector $\mathbf{X}$. Observe that the edge type of basis functions $M_{i,p-1}$ are simply differently normalized B-splines, see also Figure 4.3(b). This scaling of the B-splines was the original representation,

derived by Schoenberg [58, 59], and is known as the Curry Schoenberg B-spline. It was later that B-splines were scaled to form a partition of unity.

Figure 4.3 continues upon Figure 4.2 and illustrates the co-boundary process and the reconstruction of the 1-form which is the exterior derivative of the 0-form in Figure 4.2.



(a) The coboundary process $\bar{u}_i = \bar{\varphi}_i - \bar{\varphi}_{i-1}$



(b) reconstruction: $u_h^1(x) = \mathcal{I}_1 \bar{u} = \sum_{i=1}^{6} \bar{u}_i \, M_{i,2}(x)\mathrm{dx}$, where $\bar{u}_i = \bar{\varphi}_i - \bar{\varphi}_{i-1}$

Figure 4.3: This figure continues upon Figure 4.2. The coboundary process transfers quantities from the control points to control edges, Figure 4.3(a), and provides us a discrete relation of the gradient, $\bar{u}_i = \bar{\varphi}_i - \bar{\varphi}_{i-1}$. A continuous representation of the gradient $u_h^1(x) \in \mathcal{S}^1_{p-1,\mathbf{X}}$ can subsequently be obtained by reconstruction of $\bar{u} \in \mathcal{C}^1$ using the Curry Schoenberg B-spline basis functions $M_{i,p-1}$ which are associated with control edges. These edge type of basis functions are simply differently normalized B-splines, see Figure 4.3(b).

The Curry Schoenberg B-spline basis functions $M_{i,p-1}(x)$ have the following interesting property [26],

$$\int_{\mathbb{R}} M_{i,p-1}(x)\mathrm{dx} = 1 \tag{4.10}$$

We shall make use of this property in Chapter 5, where we use it to calculate the circulation around a lifting object by integrating velocity - a one form - over a path enclosing the object.

The new basis functions for 1-forms allow us to define the reconstructed space of spline 1-forms $\mathcal{S}^1_{p-1,\mathbf{X}}$ as

$$\mathcal{S}^1_{p-1,\mathbf{X}} := \operatorname{span}\{M_{i,p-1}(x)\mathrm{dx}\}_{i=1}^n \tag{4.11}$$

4.2.3 Reduction of 0-forms to B-spline control points

The reconstruction process is simply defined as evaluation of a point on the spline, given a set of control points or control edges and associated basis functions. The reduction operation $\mathcal{R}$ requires the inverse relation: given a set of points on the spline and/or its derivatives, calculate the unique set of B-spline coefficients. Since B-spline basis functions are not nodal - the control points do not interpolate the spline function - we can not use the De Rahm map as a reduction operator. Remember that the De Rahm map acting on a zero form provides a collection of points on the function (Chapter 2, page 48). In order to use these function values directly as discrete degrees of freedom, the reconstruction map $\mathcal{I}$ should be a nodal interpolation. The choice of B-splines as the reconstruction map, implies that we need to define a new reduction operator.

Consider a spline function $f_h(x)$, an element of $\mathcal{S}^0_{p,\mathbf{X}}$. Each polynomial piece $I_j = (x_j, x_{j+1})$, of non-zero length, can be expanded into a Taylor polynomial of order $p+1$ around a point $t \in I_j$

$$\begin{aligned} f_h(x) &= f_h(t) + f_h'(t)(x-t) + \frac{1}{2}f_h''(t)(x-t)^2 + \ldots + \frac{(x-t)^p}{p!}D^p f_h(t) \\ &= \sum_{d=0}^{p} \frac{(x-t)^d}{d!}\, D^d f_h(t) \end{aligned} \tag{4.12}$$

We will need an important identity first proved by Marsden [51], which simplifies many dealings with splines

Theorem 4.2.1 Marsden's Identity. For any $t \in \mathbb{R}$

$$(t-x)^p = \sum_{j=0}^{n} \psi_{j,p}(t)\; N_{j,p}(x) \qquad \text{where} \qquad \psi_{j,p}(t) = (t-x_{j+1})\ldots(t-x_{j+p}) \tag{4.13}$$

Differentiating Marsden's Identity $p-d$ times with respect to t we obtain,

$$D^{p-d}(t-x)^p = \frac{p!(t-x)^d}{d!} = \sum_{j=0}^{n} D^{p-d}\psi_{j,p}(t)\; N_{j,p}(x)$$

Rearranging,

$$\frac{(x-t)^d}{d!} = \frac{(-1)^d}{p!} \sum_{j=0}^{n} D^{p-d}\psi_{j,p}(t)\; N_{j,p}(x)$$

Inserting this result in the Taylor expansion of $f_h(x)$ from equation 4.12 yields

$$\begin{aligned} f_h(x) &= \sum_{d=0}^{p} \left(\sum_{j=0}^{n} \frac{(-1)^d}{p!} \, D^{p-d}\psi_{j,p}(t) \, N_{j,p}(x) \right) D^d f_h(t) \\ &= \sum_{j=0}^{n} \left(\sum_{d=0}^{p} \frac{(-1)^d}{p!} \, D^{p-d}\psi_{j,p}(t) \, D^d f_h(t) \right) N_{j,p}(x) \\ &= \sum_{j=0}^{n} \lambda_{j,p} f_h \, N_{j,p}(x) \end{aligned} \tag{4.14}$$

Since $f_h(x) \in \mathcal{S}^0_{p,\mathbf{X}}$ allows an expansion in terms of the linearly independent B-spline basis functions $N_{i,p}(x)$, the coefficients $\lambda_{j,p} f_h$ in (4.14) are uniquely determined discrete real numbers. Hence they must be independent of the parameter t, leading to the following theorem

Theorem 4.2.2 Dual functionals [28]. For any $f_h(x) \in \mathcal{S}^0_{p,\mathbf{X}}$ and $t_j \in (x_j, x_{j+p+1})$

$$f_h(x) = \sum_{j=0}^{n} \lambda_{j,p} f_h \, N_{j,p}(x) \qquad \text{where} \qquad \lambda_{j,p} f_h = \sum_{d=0}^{p} \frac{(-1)^d}{p!} \, D^{p-d}\psi_{j,p}(t_j) \, D^d f_h(t_j) \tag{4.15}$$

The positioning of the t_j is free, however only makes sense if they are chosen to lie in the support of the respective basis function $N_{j,p}(x)$, i.e. $t_j \in (x_j, x_{j+p+1})$. The Greville abscissa are a likely candidate for the positioning of these points. The $\lambda_{j,p} f_h$ are known as the B-spline dual functionals. Functionals use functions as input and give a real number as output.

The dual functionals have the following important property, see de Boor [26]

$$\lambda_{i,p} N_{j,p} = \delta_{ij} \qquad \text{and hence} \qquad \lambda_{i,p} \left(\sum_{j=0}^{n} \bar{f}_j N_{j,p}(x) \right) = \bar{f}_i \tag{4.16}$$

Equation (4.16) provides a closed form relation between a point on the spline function $f_h(x)$ and the B-spline coefficients $\bar{f}_i$. The reduction operator $\mathcal{R}_0$ acting on zero forms is thus given by the B-spline dual functionals

$$\mathcal{R}_0 := \lambda_{i,p}, \quad \text{for} \quad i = 0, 1, \ldots, n \tag{4.17}$$

From (4.16) it follows that reducing a spline to a set of B-spline coefficients and subsequent reconstruction can be performed without loss in information. The reduction thus preserves splines,

$$\mathcal{R}_0 \mathcal{I}_0 \bar{f} = \mathcal{R}_0 \left(\sum_{j=0}^{n} \bar{f}_j N_{j,p}(x) \right) = \bar{f} \tag{4.18}$$

Furthermore, the spline approximation $f_h(x) = \mathcal{I}_0\mathcal{R}_0 f^0(x)$ to a smooth function $f(x)$, approximates the function $f(x)$ and simultaneously its $p-1$ continuous derivatives up to optimal order of accuracy, at least for regular knot partitions [26],

$$\left\| D^j f(x) - D^j \mathcal{I}\mathcal{R} f(x) \right\|_{L^\infty(\Omega)} = O(h^{p+1-j}) \tag{4.19}$$

Here the error is measured in the supremum norm, h is a measure of the local mesh size and $p+1-j$ the order of convergence. For sharp error estimates see de Boor and Fix [28] and Lyche and Schumaker [50].

From (4.18) and (4.19) it follows that the above derived reduction and reconstruction operators $\mathcal{R}_0$ and $\mathcal{I}_0$ acting on 0-forms follow the requirements in (4.2) and (4.3)

$$\mathcal{R}_0\mathcal{I}_0 = id \qquad \text{and} \qquad \mathcal{I}_0\mathcal{R}_0 = id + O(h^{p+1})$$

We will discuss two examples illustrating the reduction and interpolation process.

EXAMPLE 4.2.1 Lets reduce a continuous function $f(x)$ to a collection of B-spline coefficients $\mathcal{R}f(x) = \bar{f}$, in the case the B-spline coefficients $\bar{f}_i$ are associated with quadratic B-spline basis functions. Choosing the collocation points t_j as the Greville Abscissa $t_j^* = \frac{x_{j+1}+\ldots+x_{j+p}}{p}$ (Chapter 3, page 65) we obtain for the first term $\frac{1}{p!}\, D^{p-d}\psi_{j,p}(t)$

$$\frac{1}{2!}\, D^0\psi_{j,2}(t) = \frac{1}{2}\,(t - x_{j+1})\,(t - x_{j+2})\,, \qquad \frac{1}{2!}\, D^1\psi_{j,2}(t) = t - \frac{x_{j+1}+x_{j+2}}{2} = t - t_j^*$$
$$\text{and} \frac{1}{2!}\, D^2\psi_{j,2}(t) = 1$$

Filling this result into the DeBoor-Fix theorem (4.2.2), at parameter value $t_j = t_j^*$, we obtain an explicit relation for the B-spline coefficients $\bar{f}_i$ in terms of the continuous function $f(x)$ and its derivatives.

$$\bar{f}_j = f(t_j^*) - 1 \cdot (t_j^* - t_j^*) D^1 f(t_j^*) + \frac{1}{2}\left(t_j^* - x_{j+1}\right)\left(t_j^* - x_{j+2}\right) D^2 f(t_j^*)$$

EXAMPLE 4.2.2 We use the result from Example 4.2.1 to reduce and subsequently reconstruct a smooth function $f^0(x) = \cos(0.5\pi x)$ on the interval $x \in (0, 4)$ with quadratic degree B-splines. We choose the knot partition $\mathbf{X} = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$. The interpolation sites $t_j \in (x_j, x_{j+p+1})$ are chosen as the Greville abscissa t_j^*, which are calculated as $\mathbf{t}^* = \{0.0, 0.5, 1.5, 2.5, 3.5, 4.0\}$.

The function and its derivatives at the collocation points t_j are

$$f(t_j^*) = \cos(0.5\pi t_j^*), \qquad D^1 f(t_j^*) = -0.5 * \pi \sin(\pi t_j^*) \qquad \text{and} \qquad D^2 f(t_j^*) = -0.25 * \pi^2 \cos(\pi t_j^*)$$

The reduction $\mathcal{R}f(x) = \bar{f}$ results in

$$\bar{f}_0 = 1 \cdot \cos(0.00\pi) - \frac{\pi^2}{8}(0.0 - 0.0)(0.0 - 0.0) \cdot \cos(0.00\pi) = \cos(0.00) = 1$$
$$\bar{f}_1 = 1 \cdot \cos(0.25\pi) - \frac{\pi^2}{8}(0.5 - 0.0)(0.5 - 1.0) \cdot \cos(0.25\pi) = \cos(0.25\pi)\left(1 + \frac{\pi^2}{32}\right) \approx 0.9252$$
$$\bar{f}_2 = 1 \cdot \cos(0.75\pi) - \frac{\pi^2}{8}(1.5 - 1.0)(1.5 - 2.0) \cdot \cos(0.75\pi) = \cos(0.75\pi)\left(1 + \frac{\pi^2}{32}\right) \approx -0.9252$$
$$\bar{f}_3 = 1 \cdot \cos(1.25\pi) - \frac{\pi^2}{8}(2.5 - 2.0)(2.5 - 3.0) \cdot \cos(1.25\pi) = \cos(1.25\pi)\left(1 + \frac{\pi^2}{32}\right) \approx -0.9252$$
$$\bar{f}_4 = 1 \cdot \cos(1.75\pi) - \frac{\pi^2}{8}(3.5 - 3.0)(3.5 - 4.0) \cdot \cos(1.75\pi) = \cos(1.75\pi)\left(1 + \frac{\pi^2}{32}\right) \approx 0.9252$$
$$\bar{f}_5 = 1 \cdot \cos(2.00\pi) - \frac{\pi^2}{8}(4.0 - 4.0)(4.0 - 4.0) \cdot \cos(2.00\pi) = \cos(2.00\pi) = 1$$

The reconstructed spline function $f_h(x) = \mathcal{I}\bar{f} = \sum_{j=0}^{n} \bar{f}_j \; N_{i,2}(x)$ is depicted in the top Figure 4.4; the B-spline basis functions $N_{i,2}(x)$ are shown in Figure 4.4 below; and an analysis of the error is depicted in Figure 4.5. We observe the expected rate of convergence $O(h^3)$under mesh refinement.

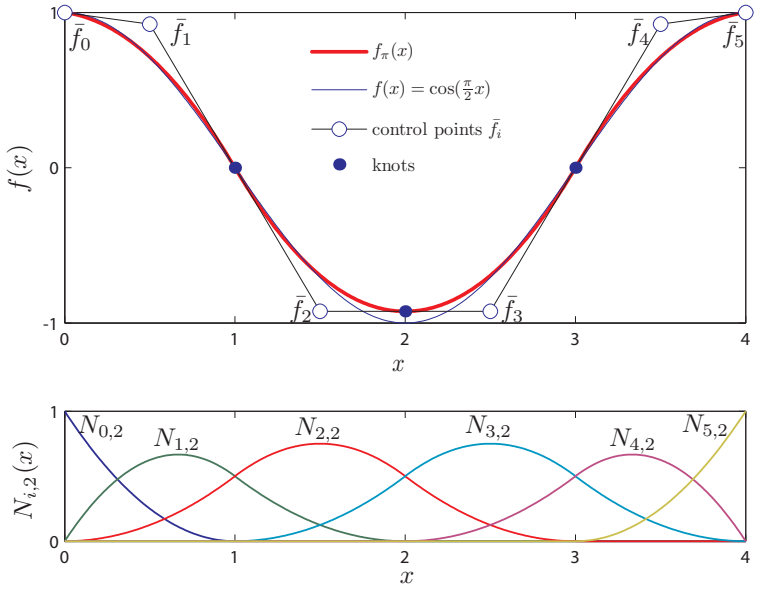


Figure 4.4: The B-spline coefficients $\bar{f}_i$, obtained by reduction of the continuous function $f(x) = \cos(\frac{\pi}{2}x)$, are interpolated by the B-spline basis functions $N_{i,p}(x)$, depicted below in Figure 4.4.

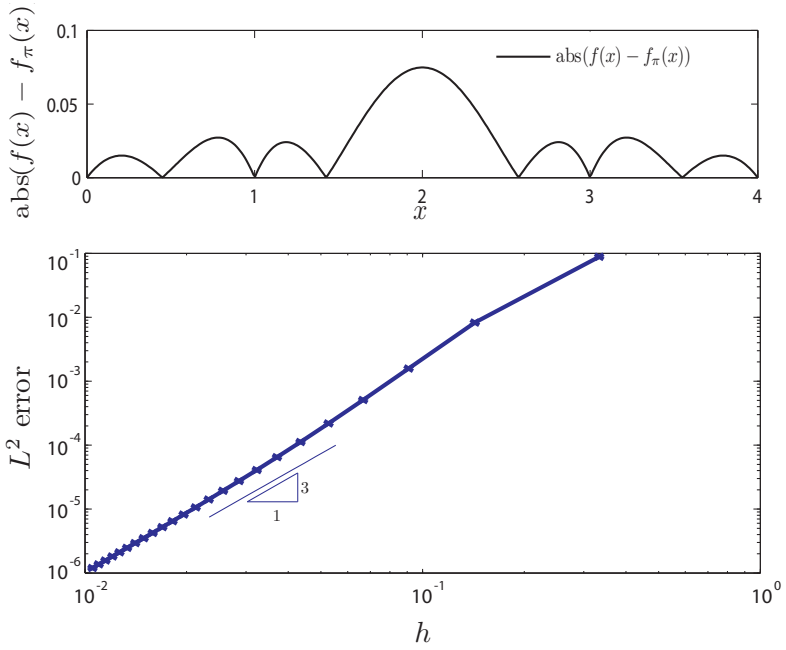


Figure 4.5: The top figure shows the absolute error between the exact function and B-spline approximation in Figure 4.4. Note that the approximation does not interpolate the exact function at the collocation points t_j. The error reduces however with the expected $p+1$ order of convergence in the L^2 norm under mesh refinement, Figure 4.5 below.

4.2.4 Reduction of 1-forms to B-spline control edges

Now that the reduction operation of zero forms to B-spline control points is clear, we can easily derive a reduction operator which reduces 1-forms to control edges. The duality property (equation 4.16) between B-spline basis functions and the dual functionals implies that the reduction operator for 1-forms should be given by

$$\mathcal{R}_1 := \frac{1}{c_i}\lambda_{i,p-1}\frac{\partial}{\partial \mathrm{x}} \qquad \text{for} \qquad i = 1, \ldots, n \tag{4.20}$$

The proof is straightforward, applying the scaled dual functional to the 1-form $u_h^1(x) \in \mathcal{S}^1_{p-1,\mathbf{X}}$,

$$\frac{1}{c_i}\lambda_{i,p-1}\frac{\partial}{\partial \mathrm{x}}u_h^1(x) = \frac{1}{c_i}\lambda_{i,p-1}\frac{\partial}{\partial \mathrm{x}}\left(\sum_{j=1}^{n} \bar{u}_j\ c_j N_{j,p-1}(x)\mathrm{dx}\right) = \bar{u}_i \tag{4.21}$$

Here we have used the fact that $\lambda_{i,p-1}N_{j,p-1}(x) = \delta_{ij}$ and $\frac{\partial}{\partial \mathrm{x}}\mathrm{dx} = 1$. The reconstruction operator $\mathcal{R}_1$ is thus a tangent vector that takes a 1-form and gives a discrete real number (For a discussion about tangent vectors, the reader is referred to the discussion in Chapter 2, page 22).

From (4.21) it follows that the reduction operator for 1-forms preserves spline 1-forms, i.e. $\mathcal{R}_1\mathcal{I}_1 = id$. It also follows from (4.19) that the reduction and reconstruction approximate a

1-form to optimal order of accuracy, i.e. $\mathcal{I}_1\mathcal{R}_1 = id + O(h^p)$. It can further be shown that the reduction operator follows the commuting diagram,

$$\begin{array}{ccc} \bar{\varphi} \in \mathcal{C}^{(0)} & \xrightarrow{\quad\delta\quad} & \bar{u} \in \mathcal{C}^{(1)} \\ \uparrow \mathcal{R}_0 & & \uparrow \mathcal{R}_1 \\ \phi_h^0(x) \in \mathcal{S}^0_{p,\mathbf{X}} & \xrightarrow{\quad\mathrm{d}\quad} & u_h^1(x) \in \mathcal{S}^1_{p-1,\mathbf{X}} \end{array} \tag{4.22}$$

The commuting property in terms of the dual functionals looks as follows

$$\delta\mathcal{R}_0\phi_h^0(x) = \mathcal{R}_1\mathrm{d}\phi_h^0(x) \qquad \Longrightarrow \qquad (\lambda_{j,p} - \lambda_{j-1,p})\,\phi_h^0(x) = \frac{x_{j+p} - x_j}{p}\lambda_{j,p-1}\mathrm{d}\phi_h^0(x)$$

Using the expression of the dual functionals in terms of the free parameter t, following from (4.14), we can write the left hand side as

$$(\lambda_{j,p} - \lambda_{j-1,p})\,\phi_h^0(x) = \sum_{d=0}^{p} \frac{(-1)^d}{p!}\, D^{p-d}\left(\psi_{j,p}(t) - \psi_{j-1,p}(t)\right)\, D^d\phi_h^0(t) \tag{4.23}$$

The Newton polynomials $\psi_{j-1,p}(t)$ and $\psi_{j,p}(t)$ can be rewritten as in terms of a the Newton polynomial $\psi_{j,p-1}(t)$ of one degree lower.

$$\begin{aligned} \psi_{j,p}(t) &= (t - x_{j+1})\ldots(t - x_{j+p}) = (t - x_{j+p})\,\psi_{j,p-1}(t) \\ \psi_{j-1,p}(t) &= (t - x_j)\ldots(t - x_{j+p-1}) = (t - x_j)\,\psi_{j,p-1}(t) \end{aligned}$$

Subtracting the former from the latter yields,

$$\psi_{j,p}(t) - \psi_{j-1,p}(t) = -\psi_{j,p-1}(t)\,(x_{j+p} - x_j)$$

Filling the above result into equation 4.23, completes the proof

$$\begin{aligned} (\lambda_{j,p} - \lambda_{j-1,p})\,\phi_h^0(x) &= \frac{x_{j+p} - x_j}{p}\sum_{d=0}^{p}\frac{(-1)^{d+1}}{(p-1)!}\, D^{p-d}\psi_{j,p-1}(t)\, D^d\phi_h^0(t) \\ &= \frac{x_{j+p} - x_j}{p}\sum_{d=1}^{p}\frac{(-1)^{d+1}}{(p-1)!}\, D^{p-d}\psi_{j,p-1}(t)\, D^d\phi_h^0(t) \\ &= \frac{x_{j+p} - x_j}{p}\sum_{d=0}^{p-1}\frac{(-1)^{d}}{(p-1)!}\, D^{p-1-d}\psi_{j,p-1}(t)\, D^dD\phi_h^0(t) \\ &= \frac{x_{j+p} - x_j}{p}\lambda_{j,p-1}\mathrm{d}\phi_h^0(t) \end{aligned}$$

In line two we have used $D^p\psi_{j,p-1}(t) = 0$. In line four the fact that the exterior derivative acting on a function is simply the derivative in the classical sense: $\mathrm{d} = D$.

4.3 Spline differential forms in 2D

In the previous section we introduced a basis for 0- and 1-forms in terms of B-spline basis functions. In this section we make the extension to the two-dimensional setting. Applying the wedge product between the B-spline basis functions $N_{i,p}(x)$ and Curry Schoenberg B-splines $M_{i,p-1}(x)$ we can reconstruct the different spaces of inner and outer oriented 0-, 1- and 2-forms in our cell complex. We shall see that the resulting reconstructed spaces of differential forms follow an exact De Rahm sequence and hence are conservative by construction. B-splines can thus be seen as generalizing Whitney forms to arbitrary order. In this section we will define the spaces and operators in the following commutative diagram which can be set up for both the primal and dual cell complex,

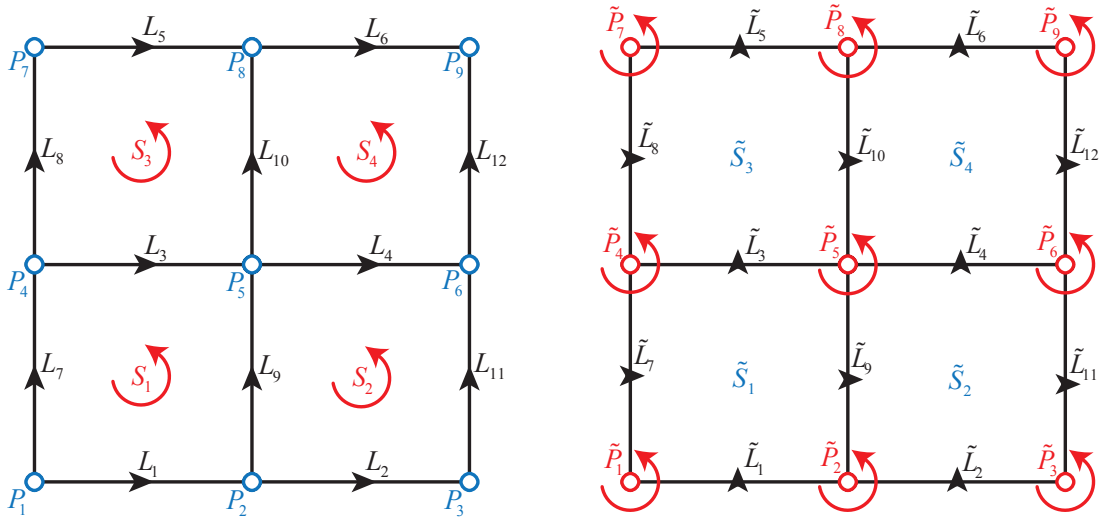
$$\begin{array}{ccccccccc}
\mathbb{R} & \longrightarrow & \Lambda^0(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^1(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^2(\Omega) & \longrightarrow & 0 \\
 & & \downarrow{\scriptstyle \mathcal{R}_0} & & \downarrow{\scriptstyle \mathcal{R}_1} & & \downarrow{\scriptstyle \mathcal{R}_2} & & \\
\mathbb{R} & \longrightarrow & \mathcal{C}^{(0)} & \xrightarrow{\mathbb{D}^{(1,0)}} & \mathcal{C}^{(1)} & \xrightarrow{\mathbb{D}^{(2,1)}} & \mathcal{C}^{(2)} & \longrightarrow & 0 \\
 & & \downarrow{\scriptstyle \mathcal{I}_0}\ \uparrow{\scriptstyle \mathcal{R}_0} & & \downarrow{\scriptstyle \mathcal{I}_1}\ \uparrow{\scriptstyle \mathcal{R}_1} & & \downarrow{\scriptstyle \mathcal{I}_2}\ \uparrow{\scriptstyle \mathcal{R}_2} & & \\
\mathbb{R} & \longrightarrow & \Lambda^0_h(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^1_h(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^2_h(\Omega) & \longrightarrow & 0
\end{array} \tag{4.24}$$

4.3.1 Primal cell complex

We assume that the geometric domain under consideration is given by a NURBS map, see Figure 4.1. The NURBS control mesh provides an initial representation of a primal cell complex. The accompanying knot vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ partition the p_1 by p_2 degree geometric domain under consideration into elements.

$$\mathbf{X}_d = \{ \underbrace{a^d, ..., a^d}_{p_d+1}, x^d_{p+1}, ..., x^d_i, x^d_{i+1}, ..., x^d_{m_d-p_d-1}, \underbrace{b^d, ..., b^d}_{p_d+1} \}, \qquad x^d_{i+1} \geq x^d_i \tag{4.25}$$

Applying the theory developed in Chapter 2, Section 2.3, we can assign either the inner - Figure 4.6(a) - or outer orientation -Figure 4.6(b)- to the discrete control points, edges and faces in the primal cell complex. We can for example associate discrete potentials with the inner oriented control points; discrete velocities with inner oriented control edges and discrete vorticity with inner oriented control faces. Similarly we can for example associate discrete values of the stream function with outer oriented control points; discrete fluxes with outer oriented control edges; and discrete mass densities with outer oriented control faces. By means of a linear combination between these discrete co-chains and associated B-spline basis functions, we can reconstruct continuous representations of differential forms.



(a) Inner oriented objects on the primal B-spline grid (b) Outer oriented objects on the dual B-spline grid

Figure 4.6: Primal cell complex build using $\mathbf{X}_1 = \{0, 0, 0, 1, 1, 1\}$. We can choose to assign either inner (4.6(a)) or outer (4.6(b)) orientation to the control points, edges and faces.

Reduction / Reconstruction of inner oriented differential forms in 2D

In the previous section we defined the univariate spaces of differential 0- and 1-forms

$$\mathcal{S}^0_{p_d,\mathbf{X}_d} := \operatorname{span}\{N_{i,p_d}(x)\}_{i=0}^{n_d} \qquad \text{and} \qquad \mathcal{S}^1_{p_d-1,\mathbf{X}_d} := \operatorname{span}\{M_{i,p_d-1}(x)\mathrm{dx}\}_{i=1}^{n_d} \tag{4.26}$$

where $M_{i,p_d-1}(x) = c_{i,p_d-1} \cdot N_{i,p_d-1}(x)$ and $c_{i,p_d-1} = \frac{p_d}{x_{i+p_d}-x_i}$.

Suppose we discretize the inner oriented differential forms on the primal cell complex. Applying the wedge product between these univariate spaces of differential forms, we can readily construct two-dimensional spaces $\Lambda^k_h(\Omega)$ of B-spline differential k-forms,

$$\begin{aligned}
\Lambda^0_h(\Omega) &:= \mathcal{S}^0_{p_1,\mathbf{X}_1} \wedge \mathcal{S}^0_{p_2,\mathbf{X}_2} \\
\Lambda^1_h(\Omega) &:= \mathcal{S}^1_{p_1-1,\mathbf{X}_1} \wedge \mathcal{S}^0_{p_2,\mathbf{X}_2} + \mathcal{S}^0_{p_1,\mathbf{X}_1} \wedge \mathcal{S}^1_{p_2-1,\mathbf{X}_2} \\
\Lambda^2_h(\Omega) &:= \mathcal{S}^1_{p_1-1,\mathbf{X}_1} \wedge \mathcal{S}^1_{p_2-1,\mathbf{X}_2}
\end{aligned} \tag{4.27}$$

In the previous section we observed that the reduction process is given in terms of the dual functionals, which act dual to the B-spline basis functions. Again applying the wedge product between the reduction operands for 1D space we obtain the reduction operators in 2D space. The reduction acting on 0-forms is given by,

$$\mathcal{R}_0 := \lambda_{i,p_1}\lambda_{j,p_2} \qquad \text{for} \quad i = 0, \ldots, n_1 \quad \text{and} \quad j = 0, \ldots, n_2 \tag{4.28}$$

The reduction operator for 1-forms is given by the following tangent vector,

$$\begin{aligned}
\mathcal{R}_1^{(1)} &:= \frac{\lambda_{i,p_1-1}\lambda_{j,p_2}}{c_i^1}\frac{\partial}{\partial \mathrm{x}_1} \qquad \text{for} \quad i=1,\ldots,n_1 \quad \text{and} \quad j=0,\ldots,n_2 \\
\mathcal{R}_1^{(2)} &:= \frac{\lambda_{i,p_1}\lambda_{j,p_2-1}}{c_j^2}\frac{\partial}{\partial \mathrm{x}_2} \qquad \text{for} \quad i=0,\ldots,n_1 \quad \text{and} \quad j=1,\ldots,n_2 \\
\mathcal{R}_1 &= \mathcal{R}_1^{(1)} + \mathcal{R}_1^{(2)}
\end{aligned} \tag{4.29}$$

Similarly the reduction operator for 2-forms is given by the tangent 2-vector,

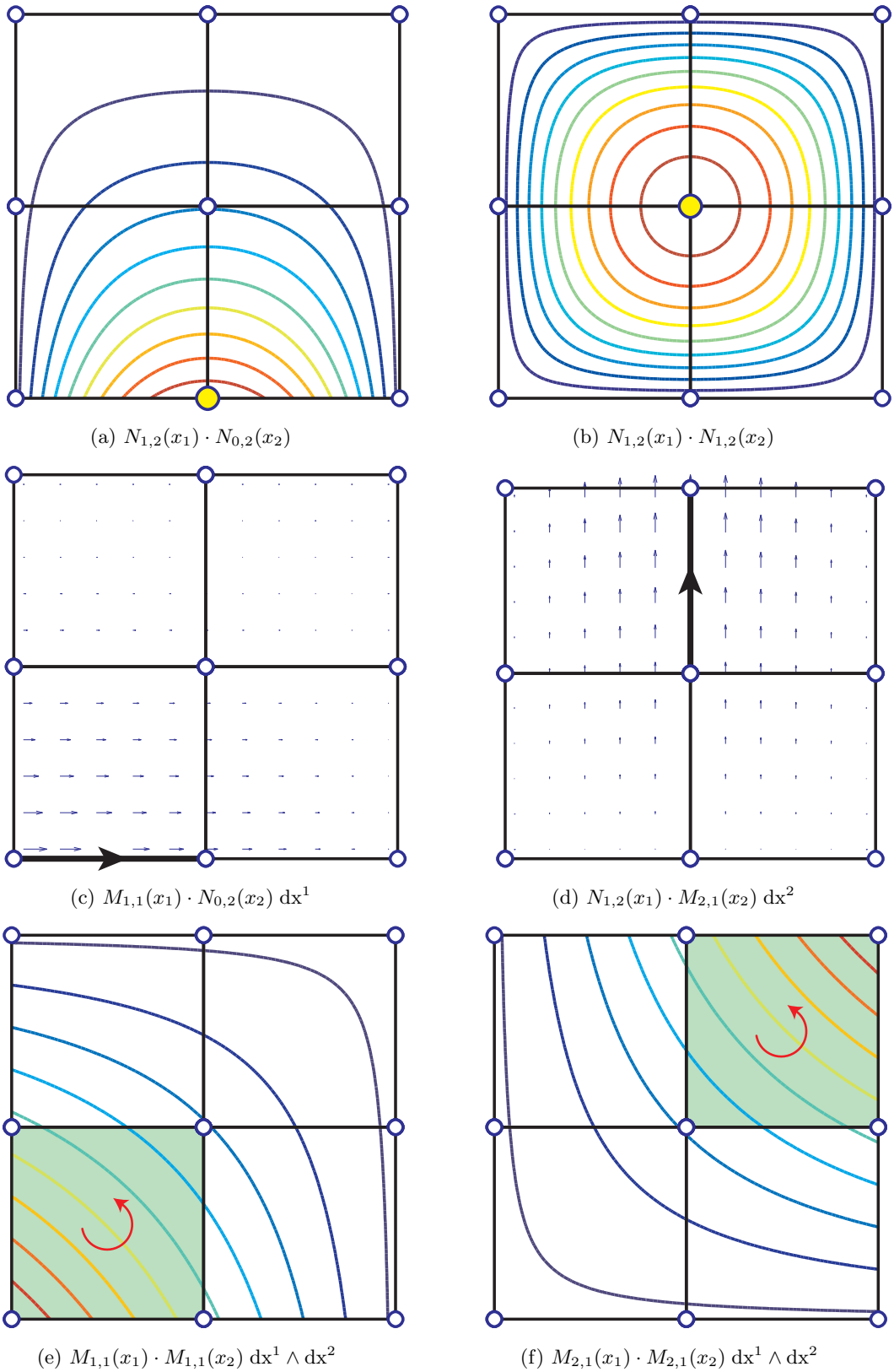
$$\mathcal{R}_2 := \frac{\lambda_{i,p_1-1}\lambda_{j,p_2-1}}{c_i^1 \cdot c_j^2}\frac{\partial}{\partial \mathrm{x}_1} \wedge \frac{\partial}{\partial \mathrm{x}_2} \qquad \text{for} \quad i=1,\ldots,n_1 \quad \text{and} \quad j=1,\ldots,n_2 \tag{4.30}$$

After the reduction stage we can reconstruct continuous representations of differential forms using the 2D spaces defined in (4.27). Examples of inner oriented differential forms are reconstruction of the potential $\phi^0(x)$, the velocity field $u^1(\mathbf{x})$ and the vorticity $\omega^2(\mathbf{x})$,

$$\begin{aligned}
\phi_h^0(\mathbf{x}) &= \mathcal{I}_0\bar{\varphi} = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \bar{\varphi}_{ij}\ N_{i,p_1}(x_1)N_{j,p_2}(x_2), \\
u_h^1(\mathbf{x}) &= \mathcal{I}_1\bar{u} = \sum_{i=1}^{n_1}\sum_{j=0}^{n_2} \bar{u}_{ij}^1\ M_{i,p_1-1}(x_1)N_{j,p_2}(x_2)\mathrm{dx}^1 + \sum_{i=0}^{n_1}\sum_{j=1}^{n_2} \bar{u}_{ij}^2\ N_{i,p_1}(x_1)M_{j,p_2-1}(x_2)\mathrm{dx}^2, \\
\omega_h^2(\mathbf{x}) &= \mathcal{I}_2\bar{\omega} = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \bar{\omega}_{ij}\ M_{i,p_1-1}(x_1)M_{j,p_2-1}(x_2)\mathrm{dx}^1 \wedge \mathrm{dx}^2,
\end{aligned} \tag{4.31}$$

where $\bar{\varphi} \in \mathcal{C}^{(0)}$, $\bar{u} \in \mathcal{C}^{(1)}$ and $\bar{\omega} \in \mathcal{C}^{(2)}$.

Figure 4.7 illustrates some of the B-spline basis functions for inner oriented 0-, 1- and 2-forms on the primal cell complex, which are applied in the reconstruction phase. In these figures, the close relation ship between the local field quantities and discrete points, lines and edges becomes clear. In contrast, in conventional FEA, all local quantities are reconstructed from point quantities. They consequently loose the ability to conserve some important quantities like mass and rotation.



(a) $N_{1,2}(x_1) \cdot N_{0,2}(x_2)$

(b) $N_{1,2}(x_1) \cdot N_{1,2}(x_2)$

(c) $M_{1,1}(x_1) \cdot N_{0,2}(x_2)\ \mathrm{dx}^1$

(d) $N_{1,2}(x_1) \cdot M_{2,1}(x_2)\ \mathrm{dx}^2$

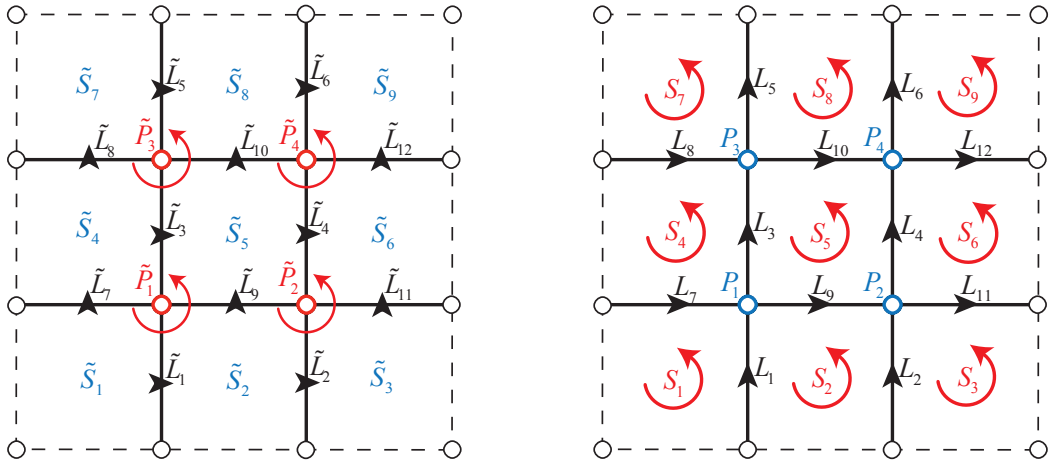
(e) $M_{1,1}(x_1) \cdot M_{1,1}(x_2)\ \mathrm{dx}^1 \wedge \mathrm{dx}^2$

(f) $M_{2,1}(x_1) \cdot M_{2,1}(x_2)\ \mathrm{dx}^1 \wedge \mathrm{dx}^2$

Figure 4.7: Some modes of inner oriented B-spline differential forms on the primal cell complex. The knot vector which defines the B-spline spaces is $\mathbf{X}_1 = \mathbf{X}_2 = \{0, 0, 0, 1, 1, 1\}$

4.3.2 Dual cell complex

While the primal cell complex is defined by the degree p_d and knot vector $\mathbf{X}_d$ (for d=1,2) of the NURBS geometry, setting up a dual grid using B-splines in less straight forward. The B-spline control mesh is a primal grid by definition since it is enclosed by a boundary. For any B-spline primal grid of dimension n_1 by n_2 control points, we can construct a dual grid using a B-spline control mesh of $n_1 + 1$ by $n_2 + 1$ control points and subsequently removing the boundary control points and edges. There is some freedom in choosing the degree and knot vector for the dual grid; an obvious choice however is the B-spline control mesh of degree p_1+1 by p_2+1 with the same interior knots as the primal grid. The excess control points and edges along the boundary should be removed by incorporating Dirichlet and/or Neumann boundary conditions in strong form. Imposing Dirichlet boundary conditions strongly is straight forward. Imposing Neumann boundary conditions in strong form, is however more difficult, since in general the local coordinate system is not orthogonal. Figures 4.8(a) and 4.8(b) illustrate such a constructed dual of the primal grids depicted respectively in Figures 4.6(a) and 4.6(b).



(a) Outer oriented objects on the dual B-spline grid (b) Inner oriented objects on the dual B-spline grid

Figure 4.8: A B-spline dual grid is obtained by subtracting excess control points o and edges – – – at the boundary. These can be eliminated by incorporating boundary conditions in strong form

Reduction / Reconstruction of outer oriented differential forms in 2D

The choice of placing the inner oriented differential forms on the primal cell complex, means we have to place the outer oriented differential forms on a dual cell complex. Suppose for the moment we explicitly define a dual cell complex in terms of B-spline basis functions and associated degrees of freedom, as described above then we can set up the following spaces of

outer oriented B-spline differential forms in 2D

$$\begin{aligned}
\Lambda_h^{\tilde{0}}(\Omega) &:= \mathcal{S}^0_{p_1+1,\mathbf{X}_1} \wedge \mathcal{S}^0_{p_2+1,\mathbf{X}_2} \\
\Lambda_h^{\tilde{1}}(\Omega) &:= \mathcal{S}^0_{p_1+1,\mathbf{X}_1} \wedge \mathcal{S}^1_{p_2,\mathbf{X}_2} + \mathcal{S}^1_{p_1,\mathbf{X}_1} \wedge \mathcal{S}^0_{p_2+1,\mathbf{X}_2} \\
\Lambda_h^{\tilde{2}}(\Omega) &:= \mathcal{S}^1_{p_1,\mathbf{X}_1} \wedge \mathcal{S}^1_{p_2,\mathbf{X}_2}
\end{aligned} \tag{4.32}$$

The reduction of outer oriented differential forms can be defined in a similar way as for the inner oriented ones. The only difference is that the 1- and 2- direction for outer oriented 1-forms are reversed. We have the following reduction operator for outer oriented 0-forms,

$$\mathcal{R}_{\tilde{0}} := \lambda_{i,p_1+1}\lambda_{j,p_2+1} \qquad \text{for} \quad i = -1,\dots,n_1 \quad \text{and} \quad j = -1,\dots,n_2 \tag{4.33}$$

The reduction operator for outer oriented 1-forms is given by the following tangent vector,

$$\begin{aligned}
\mathcal{R}^{(1)}_{\tilde{1}} &:= \frac{\lambda_{i,p_1+1}\lambda_{j,p_2}}{c^2_{j,p_2}} \frac{\partial}{\partial \mathrm{x}_2} \qquad \text{for} \quad i = -1,\dots,n_1 \quad \text{and} \quad j = 0,\dots,n_2 \\
\mathcal{R}^{(2)}_{\tilde{1}} &:= \frac{\lambda_{i,p_1}\lambda_{j,p_2+1}}{c^1_{i,p_1}} \frac{\partial}{\partial \mathrm{x}_1} \qquad \text{for} \quad i = 0,\dots,n_1 \quad \text{and} \quad j = -1,\dots,n_2 \\
\mathcal{R}_{\tilde{1}} &= \mathcal{R}^{(1)}_{\tilde{1}} + \mathcal{R}^{(2)}_{\tilde{1}}
\end{aligned} \tag{4.34}$$

Similarly the reduction operator for outer oriented 2-forms is given by the tangent 2-vector,

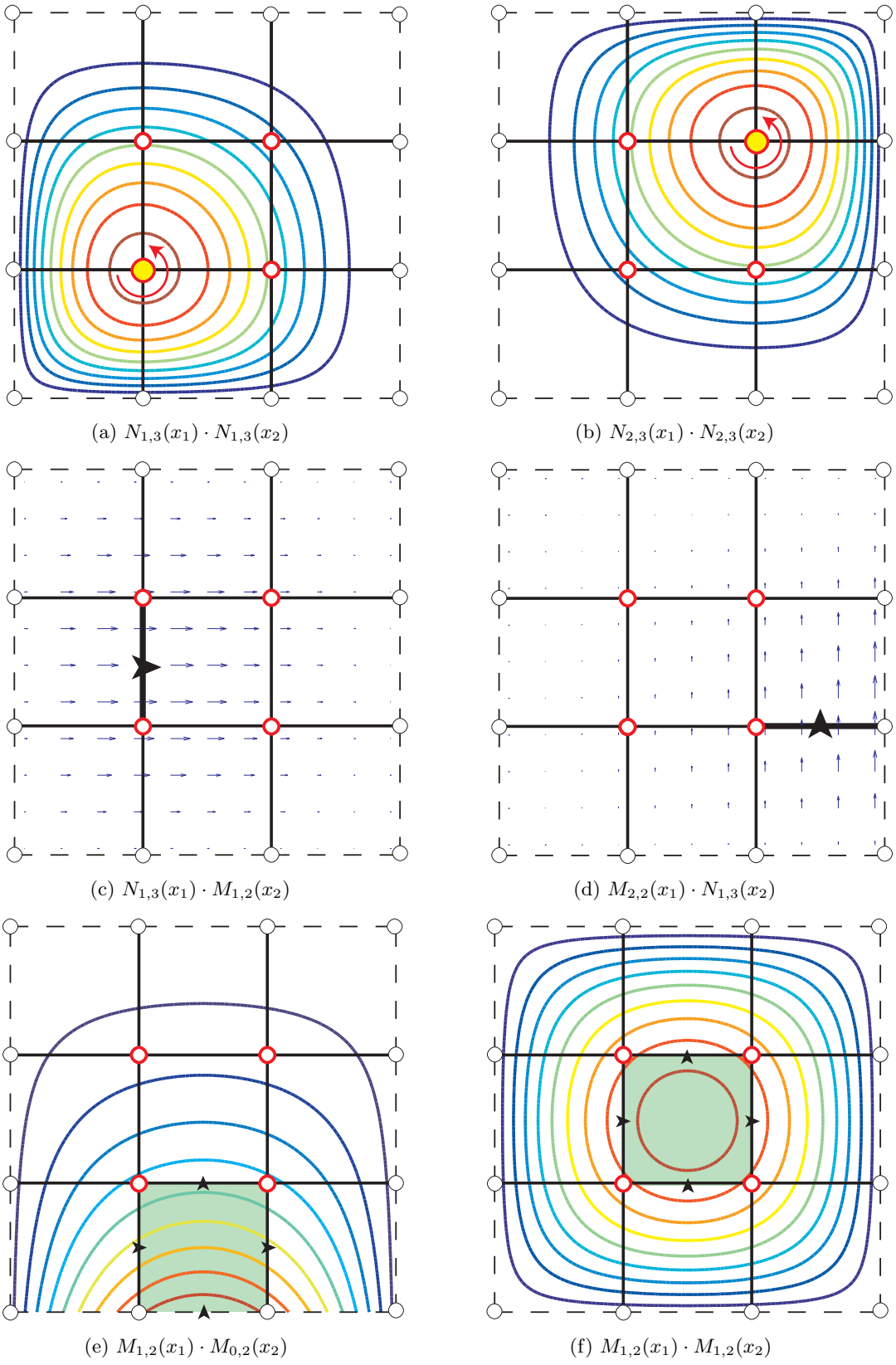
$$\mathcal{R}_{\tilde{2}} := \frac{\lambda_{i,p_1}\lambda_{j,p_2}}{c^1_{i,p_1} \cdot c^2_{j,p_2}} \frac{\partial}{\partial \mathrm{x}_1} \wedge \frac{\partial}{\partial \mathrm{x}_2} \qquad \text{for} \quad i = 0,\dots,n_1 \quad \text{and} \quad j = 0,\dots,n_2 \tag{4.35}$$

Continuous representations of outer oriented differential forms can subsequently be reconstructed using the 2D spaces defined in 4.32. Examples are reconstruction the streamfunction $\psi^{\tilde{0}}(x)$, the flux $q^{\tilde{1}}(\mathbf{x})$ and the mass density $\rho^{\tilde{2}}(\mathbf{x})$,

$$\begin{aligned}
\psi_h^{\tilde{0}}(\mathbf{x}) &= \mathcal{I}_0\bar{\psi} = \sum_{i=-1}^{n_1}\sum_{j=-1}^{n_2} \bar{\psi}_{ij}\ N_{i,p_1+1}(x_1)N_{j,p_2+1}(x_2), \\
q_h^{\tilde{1}}(\mathbf{x}) &= \mathcal{I}_1\bar{q} = \sum_{i=-1}^{n_1}\sum_{j=0}^{n_2} \bar{q}^1_{ij}\ N_{i,p_1+1}(x_1)M_{j,p_2}(x_2)\mathrm{dx}^1 + \sum_{i=0}^{n_1}\sum_{j=-1}^{n_2} \bar{q}^2_{ij}\ M_{i,p_1}(x_1)N_{j,p_2+1}(x_2)\mathrm{dx}^2, \\
\rho_h^{\tilde{2}}(\mathbf{x}) &= \mathcal{I}_2\bar{\rho} = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \bar{\rho}_{ij}\ M_{i,p_1}(x_1)M_{j,p_2}(x_2)\mathrm{dx}^1 \wedge \mathrm{dx}^2,
\end{aligned} \tag{4.36}$$

where $\bar{\psi} \in \mathcal{C}^{(\tilde{0})}$, $\bar{q} \in \mathcal{C}^{(\tilde{1})}$ and $\bar{\rho} \in \mathcal{C}^{(\tilde{2})}$.

Figure 4.9 illustrates some of the B-spline basis functions for outer oriented 0-, 1- and 2-forms on the dual cell complex, which are applied in the reconstruction phase.



(a) $N_{1,3}(x_1) \cdot N_{1,3}(x_2)$

(b) $N_{2,3}(x_1) \cdot N_{2,3}(x_2)$

(c) $N_{1,3}(x_1) \cdot M_{1,2}(x_2)$

(d) $M_{2,2}(x_1) \cdot N_{1,3}(x_2)$

(e) $M_{1,2}(x_1) \cdot M_{0,2}(x_2)$

(f) $M_{1,2}(x_1) \cdot M_{1,2}(x_2)$

Figure 4.9: Some modes of outer oriented B-spline differential forms.

The construction of a dual grid is by no means necessary. In Buffa et al. [20, 19, 17] and [57] they circumvent using a dual grid by discretizing all quantities, whether inner or outer oriented, on one primal grid. Although they are able to set up exact sequences for both the inner and outer oriented differential forms in the cell complex, they are not able to connect them using the Hodge star operator. In stead they use the weak formulation. This resembles a finite volume type of discretization for the conservation laws on a single grid and a finite element type of discretization for the constitutive equations. Although their discretization approach features incidence matrices that mimic the exterior derivative, they lose the symmetry property $\mathbb{D}^{(k+1,k)} = \left(\mathbb{D}^{(n-k+1,n-k)}\right)^T$. One Consequence is that the discrete Laplacian is not symmetric. They also have problems incorporating boundary conditions.

Also in this thesis we shall not explicitly define a dual grid in terms of basis functions and associated degrees of freedom. We shall implicitly define a dual grid by making use of the symmetry property $\mathbb{D}^{(k+1,k)} = \left(\mathbb{D}^{(n-k+1,n-k)}\right)^T$ to follow the exact sequence on the dual grid. We shall make the connection between the primal grid and implicit dual grid by means of the Hodge star inner product between k-forms on the primal cell complex. The advantage is that all symmetries are preserved, while no dual grid needs to be explicitly defined. Furthermore Neumann boundary conditions can be easily applied in weak form.

4.4 Application of B-splines to the topological relations

B-splines can be seen as higher order Whitney forms [20] and thus provide vector spaces which are discretely conservative by construction. Approximation by these type of basis functions will result in a reconstructed flow field which locally conserves some important quantities like mass and rotation. In the following we show that $\mathcal{I}_{k+1}\delta = \mathrm{d}\mathcal{I}_k$ and as a consequence we can use the incidence matrices as a discrete gradient, discrete curl and discrete divergence in combination with the B-spline reconstruction.

4.4.1 The gradient operator

Let $\phi_h^0(\mathbf{x})$ be reconstructed in terms of B-spline basis functions from the 0-cochain $\bar{\varphi} \in \mathcal{C}^{(0)}$

$$\phi_h^0(\mathbf{x}) = \mathcal{I}_0\bar{\varphi} = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \bar{\varphi}_{ij}\ N_{i,p_1}(x_1)N_{j,p_2}(x_2) \tag{4.37}$$

Consider the case where $u_h^1(\mathbf{x}) = \text{grad}\ \phi_h^0(\mathbf{x})$,

$$\text{grad}\ \phi_h^0(\mathbf{x}) = \frac{\partial}{\partial x_1}\ \phi_h^0(\mathbf{x})\mathrm{dx}^1 + \frac{\partial}{\partial x_2}\phi_h^0(\mathbf{x})\mathrm{dx}^2 \tag{4.38}$$

The first component looks as follows,

$$\begin{aligned}\frac{\partial}{\partial x_1}\ \phi_h^0(\mathbf{x})\mathrm{d}x^1 &= \frac{\partial}{\partial x_1}\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\bar{\varphi}_{ij}\ N_{i,p_1}(x_1)N_{j,p_2}(x_2)\mathrm{d}x^1 \\ &= \sum_{i=1}^{n_1}\sum_{j=0}^{n_2}(\bar{\varphi}_{i,j}-\bar{\varphi}_{i-1,j})\ M_{i,p_1-1}(x_1)N_{j,p_2}(x_2)\mathrm{d}x^1 \end{aligned} \tag{4.39}$$

A similar result is obtained in the 2-direction. Next it can be shown that,

$$u_h^1(\mathbf{x}) = \mathcal{I}_1\bar{u} = \sum_{i=1}^{n_1}\sum_{j=0}^{n_2}\bar{u}_{ij}^1\ M_{i,p_1-1}(x_1)N_{j,p_2}(x_2)\mathrm{d}x^1 + \sum_{i=0}^{n_1}\sum_{j=1}^{n_2}\bar{u}_{ij}^2\ N_{i,p_1}(x_1)M_{j,p_2-1}(x_2)\mathrm{d}x^2,\quad \bar{u}\in\mathcal{C}^{(1)} \tag{4.40}$$

where the 1-cochain $\bar{u}\in\mathcal{C}^{(1)}$ is given by $\bar{u}=\delta\bar{\varphi}$, in particular (see Figure 4.10)

$$\bar{u}_{i,j}^1 = \bar{\varphi}_{i,j}-\bar{\varphi}_{i-1,j} \qquad \text{and} \qquad \bar{u}_{i,j}^2 = \bar{\varphi}_{i,j}-\bar{\varphi}_{i,j-1} \tag{4.41}$$

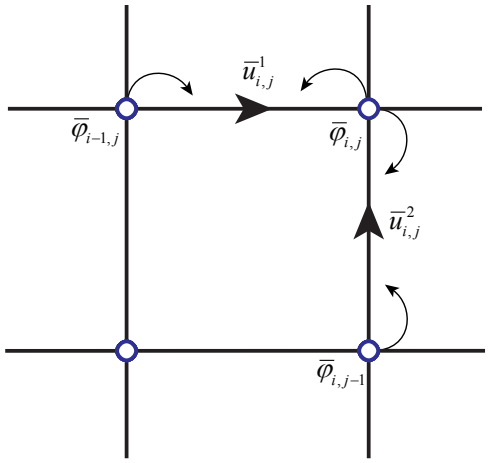


Figure 4.10: The co boundary process gives a discrete equation of the gradient

A.14 provides a discrete equation for the gradient which is exact, coordinate free and invariant under C^1 transformations

4.4.2 The curl operator

Let $u_h^1(\mathbf{x})$ be a 1-form reconstructed from $\bar{u}\in\mathcal{C}^{(1)}$. The 2-form $\omega_h^2(\mathbf{x})$ is obtained by taking the curl of the 1-form $u_h^1(\mathbf{x})$,

$$\omega_h^2(\mathbf{x}) = \text{curl}\ u_h^1(\mathbf{x}) = \left(\frac{\partial}{\partial x_2}u_1(\mathbf{x}) - \frac{\partial}{\partial x_1}u_2(\mathbf{x})\right)\mathrm{d}x^1\wedge\mathrm{d}x^2 \tag{4.42}$$

Next it can be shown that (see figure 4.11)

$$\bar{\omega}_{i,j} = \bar{u}^1_{i,j-1} + \bar{u}^2_{i,j} - \bar{u}^1_{i,j} + \bar{u}^2_{i-1,j} \,, \tag{4.43}$$

with

$$\omega^2_h(\mathbf{x}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \bar{\omega}_{i,j} \; M_{i,p-1}(x_1) M_{j,q-1}(x_2) \tag{4.44}$$

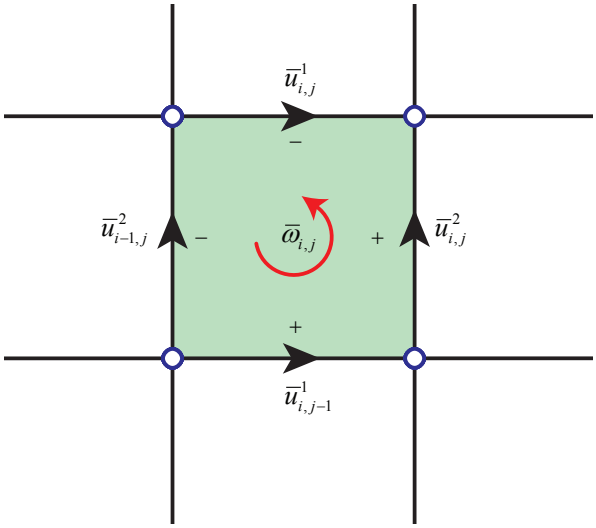


Figure 4.11: The co boundary process provides a discrete equation of the curl

A.16 is discretely exact, coordinate free and invariant under C^1 transformations

4.4.3 The divergence operator

Consider the divergence equation,

$$\rho^{\tilde{2}}_h(\mathbf{x}) = \text{div}\; q^{\tilde{1}}_h(\mathbf{x}) = \left(\frac{\partial}{\partial x_1} q_1(\mathbf{x}) + \frac{\partial}{\partial x_2} q_2(\mathbf{x}) \right) \text{dx}^1 \wedge \text{dx}^2 \tag{4.45}$$

The flux vector $q^{\tilde{1}}_h(\mathbf{x}) = q_1 \text{dx}^2 - q_2 \text{dx}^1$ can be expanded as

$$q_1(\mathbf{x}) = \sum_{i=-1}^{n_1} \sum_{j=0}^{n_2} \bar{q}^1_{i,j} \; N_{i,p_1+1}(x_1) M_{j,p_2}(x_2),$$

$$q_2(\mathbf{x}) = \sum_{i=0}^{n_1} \sum_{j=-1}^{n_2} \bar{q}^2_{i,j} \; M_{i,p_1}(x_1) N_{j,p_2+1}(x_2)$$

If $\rho_h^{\tilde{2}}(\mathbf{x})$ is expanded in terms of volume basis functions

$$\rho_h^{\tilde{2}}(\mathbf{x}) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \bar{\rho}_{i,j}\ M_{i,p_1}(x_1) M_{j,p_2}(x_2)\ , \tag{4.46}$$

the divergence equation reduces to $\bar{\rho} = \delta \bar{q}$, more precisely (see Figure 4.12)

$$\bar{\rho}_{i,j} = \bar{q}^1_{i,j} - \bar{q}^1_{i-1,j} + \bar{q}^2_{i,j} - \bar{q}^2_{i,j-1} \tag{4.47}$$

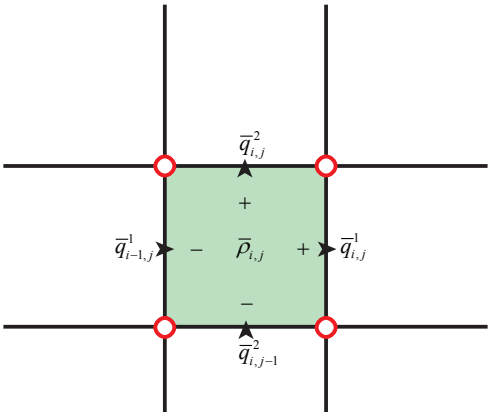


Figure 4.12: The co boundary process provides a discrete equation of the divergence

4.47 is exact, coordinate free and invariant under C^1 transformations

4.5 Application of B-splines to the Metric dependent relations

In this section we shall discuss how relations related to metric are taken into account. With metric we mean the notion of length, scale, measure and angle. Different quantities are also related by material properties which depend on the metric. In elasticity problems for example, the stress is measured as the relative elongation, or strain, times the Young's modulus. The metric combined with material behavior is described in the constitutive equations.

Recall that the constitutive equations are described by the Hodge star $*$ which maps continuous differential k-forms to $(n-k)$-forms of opposed orientation and induces a locally defined inner product (Chapter 2, page 30),

$$\alpha^{(k)} \wedge *\beta^{(k)} = \left\langle \alpha^{(k)}, \beta^{(k)} \right\rangle \mathrm{d}\Omega \tag{4.48}$$

In the discrete setting, the Hodge star should map k-cochains to $(n-k)$-cochains of opposite orientation and should similarly induce a globally defined inner product. There are thus two ways in which we can proceed to develop a discrete Hodge star:

1. Define a globally defined inner product and deduce a discrete Hodge operator.

2. Define a discrete Hodge explicitly and induce an inner product.

The first method leads to a finite element type of discretization and is convenient since it is not necessary to set up a dual grid. The two dimensional discretization method presented in this thesis follows this approach.

The second approach leads to a finite volume type of discretization and it is necessary to set up a basis for both grids. Before we explain the approach we have taken in this thesis, we briefly note that the second approach can lead to a fortunate discretization in terms of the primal and dual B-spline basis defined earlier. Because the Curry Schoenberg B-splines are simply scaled B-spline $M_{i,p_d}(x) = c_{i,d} \cdot N_{i,p_d}(x)$ the following is numerically exact,

$$* \quad : \quad \mathcal{S}^0_{p_d,\mathbf{X}_d} \mapsto \mathcal{S}^1_{p_d,\mathbf{X}_d} \tag{4.49}$$

This means that in 1D we can set up a discrete Hodge which is a diagonal matrix and is exact. In terms of the spaces for differential forms in 2D defined earlier, we can obtain a discretely exact Hodge between 0- and dual n-forms, which is a diagonal matrix. The Hodge between 1- and dual $(n-1)$-forms is in all but one direction discretely exact and needs the following projection step in one dimension

$$\pi \, \mathcal{S}^1_{p_d-1,\mathbf{X}_d} \approx \mathcal{S}^0_{p_d+1,\mathbf{X}_d} \tag{4.50}$$

This even holds in 3D. I think that it is possible to construct a tridiagonal Hodge between 1- and dual (n-1)-forms, irrespective of the polynomial order. The reason why is that the recursive definition of B-splines makes it possible to find $N_{i,p+1}(x)$ from $N_{i,p-1}(x)$, $N_{i+1,p-1}(x)$ and $N_{i+2,p-1}(x)$. This projection should resemble degree elevation process in case of Bezier splines (One element B-spline patch). The dual functionals might come handy in developing such a projection.

The discrete Hodge should satisfy some algebraic constraints , see Hiptmair [38]. Since $\alpha^{(k)} \wedge *\beta^{(k)} = \beta^{(k)} \wedge *\alpha^{(k)}$ in discrete terms will become $\bar{\alpha}^T \mathbb{H}^d \bar{\beta} = \bar{\beta}^T \mathbb{H}^d \bar{\alpha}$, the discrete version of the Hodge $\mathbb{H}^d$ should be square, symmetric, positive definite and preferably a sparse matrix. Since in this thesis we use the inner product definition, to discretize the Hodge, and B-splines feature local support, all these properties are met.

4.5.1 Discrete Hodge star

In order to introduce a sense of measure, the geometric domain under consideration should be fitted with a coordinate system. A NURBS parametric map $\mathbf{y} = \mathbf{S}(\mathbf{x})$ provides exactly what we need. Each coordinate (x_1, x_2) maps to a coordinate (y_1, y_2) in physical space and $\mathbf{y} = \mathbf{S}(\mathbf{x})$ can be regarded as a map from the parameter domain Ω' to the physical domain Ω

$$\mathbf{S} \; : \; \Omega' \mapsto \Omega$$

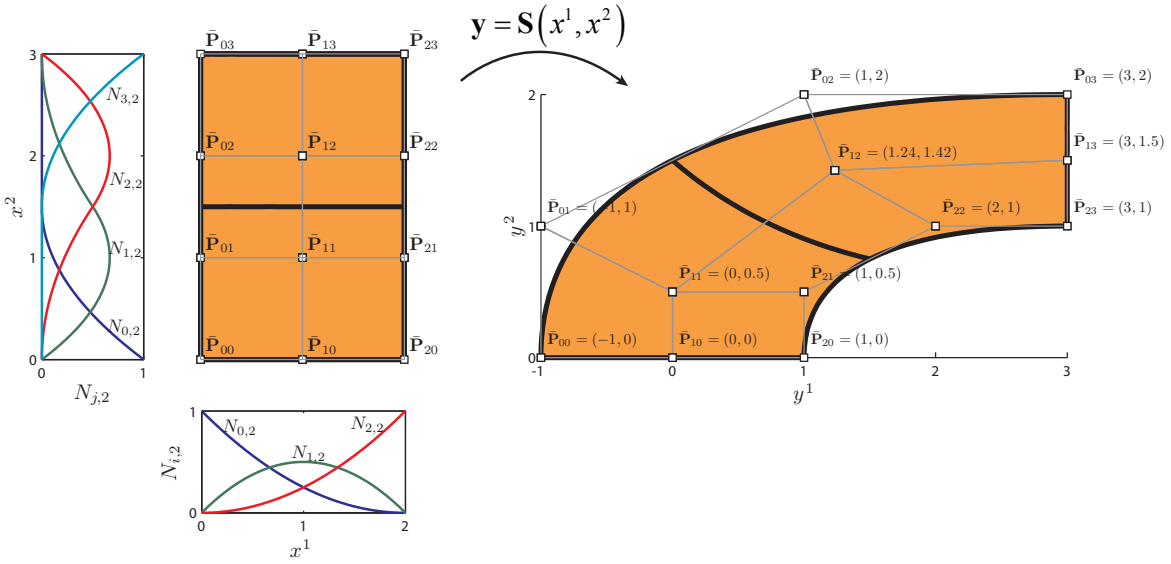


Figure 4.13: Bi-quadratic tensor product B-spline surface constructed from knot vectors $\mathbf{X}^1 = \{0, 0, 0, 2, 2, 2\}$ and $\mathbf{X}^2 = \{0, 0, 0, 1.5, 3, 3, 3\}$

By use of this map we can define an inner product which defines the notion of a metric. This inner product is given by the definition of the Hodge star operator between k-forms.

$$\alpha^{(k)} \wedge *\beta^{(k)} = \left\langle \alpha^{(k)}, \beta^{(k)} \right\rangle \mathrm{d}\Omega \tag{4.51}$$

The Hodge star operator assigns geometric quantities to 'new' dual geometric objects. In this thesis we will encounter this inner product between 0-forms and dual 2-forms and between 1-forms and dual 1-forms. Consider the following 1-forms $u^{(1)}$ and $w^{(1)}$, which can be written in terms of it's vector components in local coordinates x^i as

$$u^{(1)}(\mathbf{x}) = u_1 \, \mathrm{dx}^1 + u_2 \, \mathrm{dx}^2 \qquad \text{and} \qquad w^{(1)}(\mathbf{x}) = w_1 \, \mathrm{dx}^1 + w_2 \, \mathrm{dx}^2 \tag{4.52}$$

The Hodge star inner product, in equation 4.51, can consequently be written in terms of local coordinates as

$$w^{(1)} \wedge *u^{(1)} = \begin{pmatrix} w_1 & w_2 \end{pmatrix} \begin{pmatrix} \langle \mathrm{dx}^1, \mathrm{dx}^1 \rangle & \langle \mathrm{dx}^1, \mathrm{dx}^2 \rangle \\ \langle \mathrm{dx}^2, \mathrm{dx}^1 \rangle & \langle \mathrm{dx}^2, \mathrm{dx}^2 \rangle \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \mathbf{S}^* \mathrm{d}\Omega \tag{4.53}$$

The matrix containing the inner products $\langle \mathrm{dx}^i, \mathrm{dx}^j \rangle$ is the covariant version of the metric tensor, and shall be denoted by g^{ij}. Furthermore $\mathbf{S}^* \mathrm{d}\Omega$ is the pull back of the physical domain to parameter space.

The components of the metric tensor

The inner product components of the metric tensor - $g^{ij} = \langle \mathrm{dx}^i, \mathrm{dx}^j \rangle$ - depend on the metric induced by the parametric map $\mathbf{S}$. To make this clear, the pull back is applied to the global basis co-vectors dy^i

$$\mathbf{S}^* \mathrm{dy}^i = \frac{\partial S^i}{\partial x_j} \mathrm{dx}^j$$

By writing the pullback as a matrix vector equation, we may take the inverse of the Jacobian matrix $\mathbf{J}$ in order to write the dx^j in terms of the parametric map $\mathbf{S}$ and the dy^i components

$$\begin{pmatrix} \mathrm{dy}^1 \\ \mathrm{dy}^2 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial S^1}{\partial x_1} & \frac{\partial S^1}{\partial x_2} \\ \frac{\partial S^2}{\partial x_1} & \frac{\partial S^2}{\partial x_2} \end{pmatrix}}_{\mathbf{J}} \begin{pmatrix} \mathrm{dx}^1 \\ \mathrm{dx}^2 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \mathrm{dx}^1 \\ \mathrm{dx}^2 \end{pmatrix} = \underbrace{\frac{1}{\det \mathbf{J}} \begin{pmatrix} \frac{\partial S^2}{\partial x_2} & -\frac{\partial S^1}{\partial x_2} \\ -\frac{\partial S^2}{\partial x_1} & \frac{\partial S^1}{\partial x_1} \end{pmatrix}}_{\mathbf{J}^{-1}} \begin{pmatrix} \mathrm{dy}^1 \\ \mathrm{dy}^2 \end{pmatrix} \tag{4.54}$$

Filling in the separate terms, we can write the products $g^{ij} = \langle \mathrm{dx}^i, \mathrm{dx}^j \rangle$ in terms of the parametric map $\mathbf{S}$. The result is

$$\begin{aligned} g^{11} &= \langle \mathrm{dx}^1, \mathrm{dx}^1 \rangle = \frac{1}{(\det \mathbf{J})^2} \left(\left(\frac{\partial S^2}{\partial x_2} \right)^2 + \left(\frac{\partial S^1}{\partial x_2} \right)^2 \right) \\ g^{12} = g^{21} &= \langle \mathrm{dx}^1, \mathrm{dx}^2 \rangle = \frac{-1}{(\det \mathbf{J})^2} \left(\frac{\partial S^2}{\partial x_2} \frac{\partial S^2}{\partial x_1} + \frac{\partial S^1}{\partial x_2} \frac{\partial S^1}{\partial x_1} \right) \\ g^{22} &= \langle \mathrm{dx}^2, \mathrm{dx}^2 \rangle = \frac{1}{(\det \mathbf{J})^2} \left(\left(\frac{\partial S^2}{\partial x_1} \right)^2 + \left(\frac{\partial S^1}{\partial x_1} \right)^2 \right) \end{aligned} \tag{4.55}$$

Pullback of the physical domain to parameter space

In order to perform integration of equation 4.53, we need to a apply the pullback $\mathbf{S}^*$ of the volumeform $\mathrm{d}\Omega$

$$\mathbf{S}^* \mathrm{d}\Omega = \mathbf{S}^* \mathrm{dy}^1 \wedge \mathrm{dy}^2 = \begin{vmatrix} \frac{\partial S^1}{\partial x_1} & \frac{\partial S^1}{\partial x_2} \\ \frac{\partial S^2}{\partial x_1} & \frac{\partial S^2}{\partial x_2} \end{vmatrix} \mathrm{dx}^1 \wedge \mathrm{dx}^2 = \det \mathbf{J} \, \mathrm{d}\Omega' \tag{4.56}$$

Both the components of the metric tensor g^{ij} and the pull back of the volume form $\mathbf{S}^* \mathrm{d}\Omega$ consist of derivatives of the parametric map $\mathbf{S}$. Since in our case this map is given by the NURBS geometry, they are easily evaluated using derivatives of NURBS, as explained in Chapter 3, page 75.

Discrete Hodge between 1- and dual $(n-1)$-forms

In order to condense the notation to some extend, we shall employ vector notation for the co-chains and basis functions. Vectors will be written in boldface type. Co-chains are denoted

with a bar on top to show they are a vector containing discrete numbers (degrees of freedom). Vectors containing functions will be made clear by their input. Furthermore, two-dimensional basis functions are developed by the use of tensor products of the one-dimensional B-splines $N_p(x_i)$ and $M_{p-1}(x_i)$ in the following way

$$\begin{aligned}
&R_k(\mathbf{x}) = N_{i,p}(x_1) \cdot N_{j,q}(x_2) && k = i + j \cdot n \\
&R_k^{(1)}(\mathbf{x})\ \mathrm{dx}^1 = M_{i,p-1}(x_1)\mathrm{dx}^1 \cdot N_{j,q}(x_2) && k = i + j \cdot (n-1) \\
&R_k^{(2)}(\mathbf{x})\ \mathrm{dx}^2 = N_{i,p}(x_1) \cdot M_{j,q-1}(x_2)\mathrm{dx}^2 && k = j + i \cdot (m-1) \\
&R_k^{(1,2)}(\mathbf{x})\ \mathrm{dx}^1 \wedge \mathrm{dx}^2 = M_{i,p-1}(x_1)\mathrm{dx}^1 \cdot M_{j,q-1}(x_2)\mathrm{dx}^2 && k = i + j \cdot (n-1)
\end{aligned} \tag{4.57}$$

The following notations of the expansion of a zero form are thus equivalent

$$\varphi^0(\mathbf{x}) = \sum_{i=0}^{n}\sum_{j=0}^{m} \bar{\varphi}_{ij}\ N_{i,p}(x_1) \cdot N_{j,q}(x_2) = \sum_k \bar{\varphi}_k\ R_k(\mathbf{x}) = \bar{\boldsymbol{\varphi}}^T \cdot \mathbf{R}(\mathbf{x})$$

Here we shall use the latter vector notation. We further assume that the dimensions of these vectors are obvious from the context.The one-forms $w^1(\mathbf{x})$ and $u^1(\mathbf{x})$ from equation 4.53 can thus be expanded in vector notation as

$$\begin{aligned}
w^1(\mathbf{x}) &= \bar{\mathbf{w}}_1^T \cdot \mathbf{R}^{(1)}(\mathbf{x})\ \mathrm{dx}^1 + \bar{\mathbf{w}}_2^T \cdot \mathbf{R}^{(2)}(\mathbf{x})\ \mathrm{dx}^2 \\
u^1(\mathbf{x}) &= \bar{\mathbf{u}}_1^T \cdot \mathbf{R}^{(1)}(\mathbf{x})\ \mathrm{dx}^1 + \bar{\mathbf{u}}_2^T \cdot \mathbf{R}^{(2)}(\mathbf{x})\ \mathrm{dx}^2
\end{aligned} \tag{4.58}$$

Substituting the quantities in terms of co-chains and basis functions into equation 4.53, we obtain the following relation

$$w^{(1)} \wedge *u^{(1)} = \begin{pmatrix} \bar{\mathbf{w}}_1^T \cdot \mathbf{R}^{(1)}(\mathbf{x}) & \bar{\mathbf{w}}_2^T \cdot \mathbf{R}^{(2)}(\mathbf{x}) \end{pmatrix} \begin{pmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{u}}_1^T \cdot \mathbf{R}^{(1)}(\mathbf{x}) \\ \bar{\mathbf{u}}_2^T \cdot \mathbf{R}^{(2)}(\mathbf{x}) \end{pmatrix} \det \mathbf{J}\ \mathrm{d}\Omega' \tag{4.59}$$

Equation (4.59) contains four components which we need to compute. Using indices i and j to address each of the four components and the fact that $g^{12} = g^{21}$, we can rearrange as follows

$$\begin{aligned}
w^{(1)} \wedge *u^{(1)} &= \sum_{i=1}^{2}\sum_{j=1}^{2} \left(\bar{\mathbf{w}}_i^T \cdot \mathbf{R}^{(i)}(\mathbf{x})\right) \left(\mathbf{R}^{(j)}(\mathbf{x})^T \cdot \bar{\mathbf{u}}_j\right) g^{ij}\ \det \mathbf{J}\ \mathrm{d}\Omega' \\
&= \sum_{i=1}^{2}\sum_{j=1}^{2} \left(\bar{\mathbf{w}}_i^T \cdot \left(\mathbf{R}^{(i)}(\mathbf{x}) \otimes \mathbf{R}^{(j)}(\mathbf{x})^T\right) \cdot \bar{\mathbf{u}}_j\right) g^{ij}\ \det \mathbf{J}\ \mathrm{d}\Omega'
\end{aligned} \tag{4.60}$$

Note that by changing the order of multiplication we have taken the outer product of the two vectors containing the basis functions. The inner product is still locally defined. In order to obtain a global representation of the Hodge star operator, we need to perform integration.

Since the relation is already pulled back to the parameter space, we can perform integration here.

$$\int_{\Omega} w^{(1)} \wedge *u^{(1)} = \sum_{i=1}^{2}\sum_{j=1}^{2} \bar{\mathbf{w}}_i^T \cdot \left\{ \int_{\Omega'} \left(\mathbf{R}^{(i)}(\mathbf{x}) \otimes \mathbf{R}^{(j)}(\mathbf{x})^T \right) \cdot g^{ij} \ \det \mathbf{J} \ \mathrm{d}\Omega' \right\} \cdot \bar{\mathbf{u}}_j \tag{4.61}$$

The discrete representation of the Hodge is subsequently given by

$$\mathbb{H}^{(\tilde{1},1)} = \begin{pmatrix} \int_{\Omega'} \left(\mathbf{R}^{(1)}(\mathbf{x}) \otimes \mathbf{R}^{(1)}(\mathbf{x})^T \right) \cdot g^{11} \ \det \mathbf{J} \ \mathrm{d}\Omega' & \int_{\Omega'} \left(\mathbf{R}^{(1)}(\mathbf{x}) \otimes \mathbf{R}^{(2)}(\mathbf{x})^T \right) \cdot g^{12} \ \det \mathbf{J} \ \mathrm{d}\Omega' \\ \int_{\Omega'} \left(\mathbf{R}^{(2)}(\mathbf{x}) \otimes \mathbf{R}^{(1)}(\mathbf{x})^T \right) \cdot g^{21} \ \det \mathbf{J} \ \mathrm{d}\Omega' & \int_{\Omega'} \left(\mathbf{R}^{(2)}(\mathbf{x}) \otimes \mathbf{R}^{(2)}(\mathbf{x})^T \right) \cdot g^{22} \ \det \mathbf{J} \ \mathrm{d}\Omega' \end{pmatrix} \tag{4.62}$$

The integrals in 4.62 are in practice evaluated using Gauss numerical quadrature. This integration is performed per element in a similar way as for conventional polynomial bases. For more efficient quadrature see Hug [2]. This Hodge automatically fulfills the requirements we set up earlier, on page 102. The Hodge between 1- and dual 1-forms is always square, symmetric, positive definite and sparse.

Discrete Hodge between 0- and volume forms

Given the expansion of a weighting function $w^0(\mathbf{x})$ and a 0-form $f^0(\mathbf{x})$ in terms of B-spline basis functions

$$w^0(\mathbf{x}) = \bar{\mathbf{w}}^T \cdot \mathbf{R}(\mathbf{x}) \qquad \text{and} \qquad f^0(\mathbf{x}) = \bar{\mathbf{f}}^T \cdot \mathbf{R}(\mathbf{x}) \tag{4.63}$$

we can define an inner product between 0-forms, which defines the Hodge between 0- and volume forms,

$$\begin{aligned} w^{(0)} \wedge *f^{(0)} &= \left\langle \bar{\mathbf{w}}^T \ \mathbf{R}(\mathbf{x}) \ , \ \mathbf{R}(\mathbf{x})^T \ \bar{\mathbf{f}} \right\rangle \mathrm{d}\Omega \\ &= \bar{\mathbf{w}}^T \left(\mathbf{R}(\mathbf{x}) \otimes \mathbf{R}(\mathbf{x})^T \right) \bar{\mathbf{f}} \ \mathrm{d}\Omega \end{aligned} \tag{4.64}$$

To obtain a global representation of the Hodge we can pull the result back to parameter space and perform integration

$$\int_{\Omega'} w^{(0)} \wedge *f^{(0)} = \bar{\mathbf{w}}^T \ \left\{ \int_{\Omega'} \left(\mathbf{R}(\mathbf{x}) \otimes \mathbf{R}(\mathbf{x})^T \right) \ \det \mathbf{J} \ \mathrm{d}\Omega' \right\} \bar{\mathbf{f}} \tag{4.65}$$

The Hodge inner product matrix between 0- and volume forms in 2d is thus given by

$$\mathbb{H}^{(\tilde{2},0)} = \int_{\Omega'} \left(\mathbf{R}(\mathbf{x}) \otimes \mathbf{R}(\mathbf{x})^T \right) \ \det \mathbf{J} \ \mathrm{d}\Omega' \tag{4.66}$$

Which is always a square, symmetric, positive definite and sparse matrix.

Discrete Hodge between volume- and 0-forms

The discrete Hodge between volume- and 0- forms on the dual, is a similar expression. Given the two form $w^2(\mathbf{x})$ and $\omega^2(\mathbf{x})$ expanded in terms of the B-spline basis for two forms

$$w^2(\mathbf{x}) = \bar{\mathbf{w}}^T \cdot \mathbf{R}^{(1,2)}(\mathbf{x})\ \mathrm{dx}^1 \wedge \mathrm{dx}^2 \qquad \text{and} \qquad \omega^2(\mathbf{x}) = \bar{\boldsymbol{\omega}}^T \cdot \mathbf{R}^{(1,2)}(\mathbf{x})\ \mathrm{dx}^1 \wedge \mathrm{dx}^2 \tag{4.67}$$

the Hodge star inner product definition between 2-forms and 0-forms is given as

$$\begin{aligned} w^2 \wedge *\omega^2 &= \left\langle w^2, \omega^2 \right\rangle \mathrm{d}\Omega \\ &= \left\langle \bar{\mathbf{w}}^T \cdot \mathbf{R}^{(1,2)}(\mathbf{x})\ \mathrm{dx}^1 \wedge \mathrm{dx}^2, \bar{\boldsymbol{\omega}}^T \cdot \mathbf{R}^{(1,2)}(\mathbf{x})\ \mathrm{dx}^1 \wedge \mathrm{dx}^2 \right\rangle \mathrm{d}\Omega \\ &= \bar{\mathbf{w}}^T \ \left(\mathbf{R}^{(1,2)}(\mathbf{x}) \otimes \mathbf{R}^{(1,2)}(\mathbf{x}) \right) \bar{\boldsymbol{\omega}}\ \left\langle \mathrm{dx}^1 \wedge \mathrm{dx}^2, \mathrm{dx}^1 \wedge \mathrm{dx}^2 \right\rangle \det \mathbf{J}\ \mathrm{d}\Omega' \end{aligned} \tag{4.68}$$

The metric is contained in the inner product term $\left\langle \mathrm{dx}^1 \wedge \mathrm{dx}^2, \mathrm{dx}^1 \wedge \mathrm{dx}^2 \right\rangle$ which can be rewritten using the pullback of the volume form (4.56) $\mathrm{d}\Omega = \mathrm{dy}^1 \wedge \mathrm{dy}^2 = \det \mathbf{J} \mathrm{dx}^1 \wedge \mathrm{dx}^2$ as follows

$$\left\langle \mathrm{dx}^1 \wedge \mathrm{dx}^2, \mathrm{dx}^1 \wedge \mathrm{dx}^2 \right\rangle = \left\langle \frac{1}{\det \mathbf{J}} \mathrm{dy}^1 \wedge \mathrm{dy}^2, \frac{1}{\det \mathbf{J}} \mathrm{dy}^1 \wedge \mathrm{dy}^2 \right\rangle = \frac{1}{(\det \mathbf{J})^2} \tag{4.69}$$

since $\left\langle \mathrm{dy}^1 \wedge \mathrm{dy}^2, \mathrm{dy}^1 \wedge \mathrm{dy}^2 \right\rangle = 1$. The inner product definition of the Hodge star can subsequently be written as

$$w^2 \wedge *\omega^2 = \bar{\mathbf{w}}^T \ \left(\mathbf{R}^{(1,2)}(\mathbf{x}) \otimes \mathbf{R}^{(1,2)}(\mathbf{x}) \right) \ \bar{\boldsymbol{\omega}}\ \frac{1}{\det \mathbf{J}}\ \mathrm{d}\Omega' \tag{4.70}$$

Integrating this relation will provide us a discrete representation of the Hodge star between volume- and dual 0-forms,

$$\mathbb{H}^{(\tilde{0},2)} = \int_{\Omega'} \left(\mathbf{R}^{(1,2)}(\mathbf{x}) \otimes \mathbf{R}^{(1,2)}(\mathbf{x})^T \right)\ \frac{1}{\det \mathbf{J}}\ \mathrm{d}\Omega' \tag{4.71}$$

Again, this Hodge is square, symmetric, positive definite and sparse.

4.5.2 Boundary conditions

Dirichlet boundary conditions are easy to apply. Once the system of equations has been set up, the Dirichlet boundary conditions can be enforced strongly by multiplying the boundary conditions with the associated columns of the system matrix and subsequently subtracting these from the right hand side vector.

Neumann boundary conditions require a little more effort to impose. These are taken into account by the following boundary integral,

$$\int_{\partial\Omega} w^{(0)} \wedge *u^{(1)} \tag{4.72}$$

Here $w^{(0)}$ can be regarded as a test function and $*u^{(1)}$ is the flux. Every differential k-form at the boundary of a manifold Ω can be decomposed into a component tangent and a component normal to the boundary. For a 1-form $u^1(\mathbf{x})$ we have

$$u^1(\mathbf{x}) = u_t \mathrm{d}\Gamma + u_n \mathrm{dx}^n \tag{4.73}$$

where $d\Gamma$ is tangent and dx^n is normal to the boundary, as illustrated in Figure 4.14.

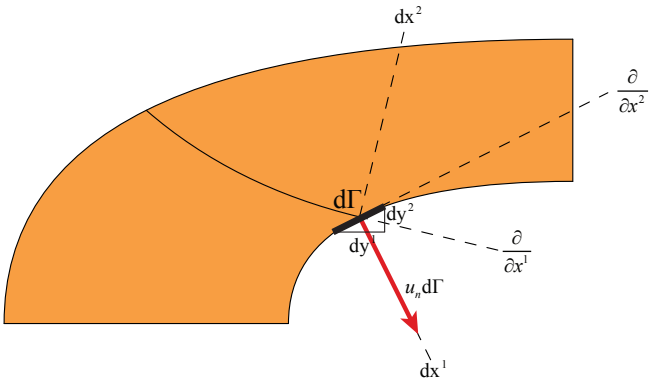


Figure 4.14: The normal flux u_n dΓ at the east boundary needs to be taken into account.

The flux term $q^{\tilde{1}} = *u^1(\mathbf{x})$ is then given by,

$$q^{\tilde{1}}(\mathbf{x}) = *u^1(\mathbf{x}) = u_t(\mathbf{y})\mathrm{dx}^n + u_n(\mathbf{y})\mathrm{d}\Gamma \tag{4.74}$$

The normal flux component, $u_n(\mathbf{y})\mathrm{d}\Gamma$, is the boundary condition we wish to implement, and is thus a known function. The boundary integral in (4.72) can then be stated in terms of this normal flux as,

$$\int_{\partial\Omega} w^{(0)} \wedge u_n(\mathbf{y})\mathrm{d}\Gamma \tag{4.75}$$

The normal flux $u_n(\mathbf{y})\mathrm{d}\Gamma$, in (4.75), is still defined in physical space, so we need to apply the pull back. Say, we wish to impose Neumann boundary conditions at the east boundary in Figure 4.14. Then, since $\frac{\partial}{\partial x^1} = 0$ at the east boundary, the pull back of dΓ is given by,

$$\mathrm{d}\Gamma = \mathrm{dy}^1 + \mathrm{dy}^2 \qquad \text{where} \qquad \mathrm{dy}^1 = \frac{\partial S^1}{\partial x^2}\mathrm{dx}^2 \qquad \text{and} \qquad \mathrm{dy}^2 = \frac{\partial S^2}{\partial x^2}\mathrm{dx}^2, \tag{4.76}$$

and the pull back of $u_n(\mathbf{y})\mathrm{d}\Gamma$ becomes,

$$\mathbf{S}^* \left(u_n(\mathbf{y})\mathrm{d}\Gamma\right) = u_n\left(\mathbf{S}(\mathbf{x})\right)\left(\frac{\partial S^1}{\partial x^2} + \frac{\partial S^2}{\partial x^2}\right)\mathrm{dx}^2. \tag{4.77}$$

Furthermore, the weighting function $w^0(\mathbf{x})$ is expanded in terms of one-dimensional B-spline basis functions as a 0-form

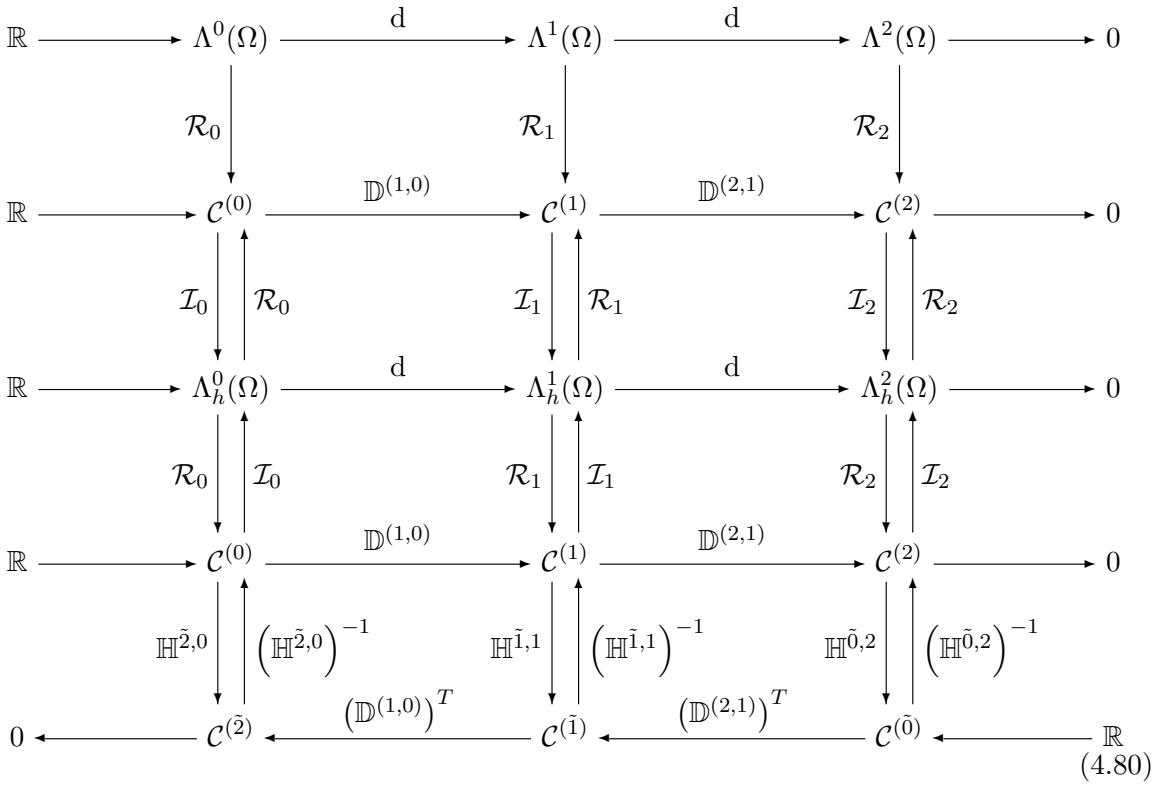
$$w^0(x^i) = \bar{\mathbf{w}}^T \cdot \mathbf{N}(x^i) \tag{4.78}$$

Inserting (4.77) and (4.78) into (4.75) provides us the integral we need to evaluate at the east boundary,

$$\int_{\partial\Omega} w^{(0)} \wedge *u^{(1)} = \bar{\mathbf{w}}^T \cdot \int_{\partial\Omega} \mathbf{N}_q(x^2) \cdot q_n(\mathbf{S}(1, x^2)) \left(\frac{\partial S^1}{\partial x^2} + \frac{\partial S^2}{\partial x^2} \right) \mathrm{dx}^2 \tag{4.79}$$

4.5.3 Discrete DeRahm sequence revisited

At this moment we have developed the basic tools to discretize a range of physical problems in space. We introduced the incidence matrices in Chapter 2, Section 2.3, which mimic the exterior derivative; and in this chapter we developed a discrete Hodge operator using B-splines. Furthermore, we introduced a reduction $\mathcal{R}$ and reconstruction operation $\mathcal{I}$ employing B-splines which connect the continuous with the discrete and commute with the exterior derivative. The commuting diagram in (4.80) summarizes all operators of the proposed discretization approach.



(4.80)

It is very important to understand the operators in this diagram, and their commuting properties. The first line represents the spaces $\Lambda^k(\Omega)$ hosting the local quantities we want to

approximate. Using the reduction operation $\mathcal{R}_k$ we go from local to global quantities and reduce these infinite dimensional continuous spaces to finite dimensional discrete spaces $\mathcal{C}^{(k)}$. This reduction operation commutes with the exterior derivative, so $\mathbb{D}^{k+1,k}\ \mathcal{R}_k = \mathcal{R}_{k+1}\ \mathrm{d}$. Subsequentally we can reconstruct finite dimensional continuous spaces $\Lambda_h^k(\Omega)$ which approximate $\Lambda^k(\Omega)$ to a certain order. Again the reconstruction operation commutes with the exterior derivative, so $\mathrm{d}\ \mathcal{I}_k = \mathcal{R}_{k+1}\ \mathbb{D}^{k+1,k}$.

The fourth line represents our discrete primal cell complex. Since we have that $\mathcal{R}_k\ \mathcal{I}_k =$ *identity* the fourth line is exactly the same as the second. We connect the primal cell complex with a dual cell complex (line 5) by means of the discrete Hodge star developed in this chapter. Note that since we have implicitly defined the dual cell complex, we cannot reconstruct cochains to differential forms on the dual grid.

4.5.4 The discrete co-derivative

Since we have constructed discrete analogues of the exterior derivative and Hodge operators, we are able to represent a discrete variant of the co-derivative. Analogous to continuous space (see Chapter 2, Section 2.2.5), the discrete co-derivative is a map from a k-cochain to a $(k-1)$-cochain,

$$\mathbb{D}^{*(k-1,k)} \quad : \quad \mathcal{C}^{(k)} \mapsto \mathcal{C}^{(k-1)}$$

In 2D space we have two of these operators. Analogous to continuous space, where the co-derivative is defined as $\mathrm{d}^* = (-1)^{2(k+1)+1} * \mathrm{d} *$ (Chapter 2, Section 2.2.5), we write these operators in terms of the discrete Hodge and the discrete exterior derivative as,

$$\mathbb{D}^{*(0,1)} = -\left(\mathbb{H}^{(\tilde{2},0)}\right)^{-1}\ \left(\mathbb{D}^{(1,0)}\right)^T\ \mathbb{H}^{(\tilde{1},1)} \tag{4.81}$$

$$\mathbb{D}^{*(1,2)} = -\left(\mathbb{H}^{(\tilde{1},2)}\right)^{-1}\ \left(\mathbb{D}^{(2,1)}\right)^T\ \mathbb{H}^{(\tilde{0},2)} \tag{4.82}$$

Note that the discrete versions of the co-derivative require the inverse of a discrete Hodge. These are expensive to calculate and are full matrices. In practice we will always attempt to circumvent such operations.

4.5.5 Discrete scalar and vector Laplacian

The Laplacian is an important operator in many physical problems. Its building blocks are the exterior derivative and the co-derivative (Chapter 2, Section 2.2.5). In 2D space we have three different types of the Laplacian. A scalar Laplacian acting on a 0-form; the vector Laplacian acting on a 1-form; and yet another scalar Laplacian acting on a 2-form.

Analogous to continuous space (Chapter 2, Section 2.2.5), the discrete Laplacian acting on a k-form can be constructed as follows,

$$\Delta_d^{(k)} = \mathbb{D}^{(k,k-1)}\mathbb{D}^{*(k-1,k)} + \mathbb{D}^{*(k,k+1)}\mathbb{D}^{(k+1,k)} \tag{4.83}$$

Scalar Laplacian acting on a 0-form

The scalar Laplacian is very common operator in physics. It occurs in many physical phenomena such the diffusion in heat and fluid flow; wave propagation; electric and gravitational potentials, etc. Also in the Navier-Stokes equation a scalar Poisson problem frequently needs to be solved for the pressure.

Exactly as in the continuous case, $\mathbb{D}^*\bar{\varphi}^{(0)} = 0$ (follow the discrete De Rahm sequence in 4.80), and consequently, the scalar Laplacian acting on a 0-form $\bar{\varphi}^{(0)}$ is given by,

$$\Delta_d^{(0)}\ \bar{\varphi}^{(0)} = \mathbb{D}^{*(0,1)}\mathbb{D}^{(1,0)}\ \bar{\varphi}^{(0)} \tag{4.84}$$

The Poisson equation, written in terms of the discrete Hodge and the Discrete exterior derivative, can be stated as,

$$= -\left(\mathbb{H}^{(\tilde{2},0)}\right)^{-1}\ \left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)}\mathbb{D}^{(1,0)}\ \bar{\varphi}^{(0)} = \bar{f}^{(0)} \tag{4.85}$$

We can bring the left Hodge to the right hand, such that we do not need to invert a matrix.

$$\left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)}\mathbb{D}^{(1,0)}\ \bar{\varphi}^{(0)} = -\mathbb{H}^{(\tilde{2},0)}\ \bar{f}^{(0)} \tag{4.86}$$

Furthermore, instead of determining $\bar{f}^{(0)} = \mathcal{R}_0 f^0$ and calculating $\mathbb{H}^{(\tilde{2},0)}$, we can use the known function f^0 directly in the inner product definition of the Hodge (4.64),

$$w^0 \wedge *f^0 = \left\langle w^0, f^0 \right\rangle \mathrm{d}\Omega \tag{4.87}$$

The right hand side then requires the evaluation of the following integral,

$$\int_{\Omega'} w^0 \wedge *f^0 = \bar{\mathbf{w}}^T \int_{\Omega'} \mathbf{R}(\mathbf{x}) \cdot f^0\ \det \mathbf{J}\ \mathrm{d}\Omega' \tag{4.88}$$

where f^0 is the known function.

To illustrate that the mimetic discretization approach to Poisson's problem involving a 0-form is equal to the weak formulation (continuous Galerkin), we multiply by a test function, a 0-form w^0, and perform integration over Ω (4.89). In line (4.90) and (4.91) we applied the

generalized Stokes theorem (Chapter 2, theorem 2.2.1, page 36).

$$\int_\Omega w^0 \wedge \left(\mathrm{d} * \mathrm{d}\varphi^0\right) = \int_\Omega w^0 \wedge *f^0 \tag{4.89}$$

$$\int_\Omega \mathrm{d}\left(w^0 \wedge *\mathrm{d}\varphi^0\right) - \int_\Omega \mathrm{d}w^0 \wedge *\mathrm{d}\varphi^0 = \int_\Omega w^0 \wedge *f^0 \tag{4.90}$$

$$\int_\Omega \mathrm{d}w^0 \wedge *\mathrm{d}\varphi^0 = -\int_\Omega w^0 \wedge *f^0 + \int_{\partial\Omega} w^0 \wedge *\mathrm{d}\varphi^0 \tag{4.91}$$

Finally, using the inner product definition of the Hodge star (4.51) and the result we obtained earlier in (4.75) regarding Neumann boundary conditions, we can write (4.91) as follows,

$$\int_\Omega \left\langle \mathrm{d}w^0, \mathrm{d}\varphi^0 \right\rangle \mathrm{d}\Omega = -\int_\Omega \left\langle w^0, f^0 \right\rangle \mathrm{d}\Omega + \int_{\partial\Omega} w^0 \wedge u_n \mathrm{d}\Gamma \tag{4.92}$$

This is exactly the weak formulation for the Poisson problem.

Scalar Laplacian acting on a 2-form

Using the weak formulation (4.92) hides the fact that the Laplacian can be applied to other than point related quantities (0-forms). Application of (4.83) to a 2-form yields,

$$\Delta_d^{(2)} \bar{\psi}^{(2)} = \mathbb{D}^{(2,1)} \mathbb{D}^{*(1,2)} \bar{\psi}^{(2)} = -\mathbb{D}^{(2,1)} \left(\mathbb{H}^{(\tilde{1},2)}\right)^{-1} \left(\mathbb{D}^{(2,1)}\right)^T \mathbb{H}^{(\tilde{0},2)} \bar{\psi}^{(2)} \tag{4.93}$$

Here we used that $\mathbb{D}\bar{\psi}^{(2)} = 0$ (follow the discrete DeRahm sequence in (4.80)). In this case we do need to invert a Hodge matrix.

Vector Laplacian acting on a 1-form

Besides the well known Laplacian acting on scalar functions, there are also cases where the Laplacian operator acts on vector valued functions. An example is the diffusion term in the Navier-Stokes equations. The Laplacian acting on the flux, an outer oriented 1-form, is given by

$$\Delta_d^{(\tilde{1})}\ \bar{q}^{(\tilde{1})} = \left(\mathbb{D}^{(\tilde{1},\tilde{0})} \mathbb{D}^{*(\tilde{0},\tilde{1})} + \mathbb{D}^{*(\tilde{1},\tilde{2})} \mathbb{D}^{(\tilde{2},\tilde{1})}\right) \bar{q}^{(\tilde{1})} \tag{4.94}$$

In case of incompressible flow, $\mathbb{D}^{(\tilde{2},\tilde{1})} \bar{q}^{(\tilde{1})} = 0$, and the vector Laplacian simplifies to

$$\Delta_d^{(\tilde{1})}\ \bar{q}^{(\tilde{1})} = \mathbb{D}^{(\tilde{1},\tilde{0})} \mathbb{D}^{*(\tilde{0},\tilde{1})}\ \bar{q}^{(\tilde{1})} \tag{4.95}$$

4.6 Numerical experiments - 2D discretization of the poisson equation

In this section we shall perform numerical experiments for some simple test cases of the scalar Poisson equation acting on a 0-form, which we can compare with an exact solution. We also compare the results obtained with our IsoGeometric Mimetic approach with those obtained from Galerkin based IsoGeometric Analysis.

Consider the scalar Poisson equation in the 2-dimensional interval $\Omega = [0,1]^2$,

$$\Delta\varphi^0 = f^0 \tag{4.96}$$

of which the exact solution and accompanying right hand side function are given by

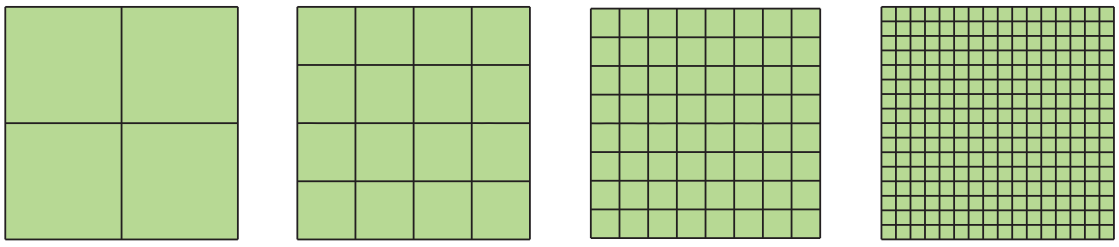
$$\begin{aligned} \varphi^0(\mathbf{y}) &= \sin(i\pi y_1) \cdot \sin(i\pi y_2) \\ f^0(\mathbf{y}) &= \frac{\partial^2 \varphi}{\partial y_1^2} + \frac{\partial^2 \varphi}{\partial y_1^2} \\ &= -(i\pi)^2 \sin(i\pi y_1) \cdot \sin(i \cdot \pi y_2) \end{aligned} \tag{4.97}$$

where φ^0 is and f^0 are both zero forms. Furthermore, in our experiments we use $i = 2$. We can discretely represent (4.96) as

$$\left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)} \mathbb{D}^{(1,0)} \, \bar{\varphi}^{(0)} = \bar{f}^{(\tilde{2})} \tag{4.98}$$

The Dirichlet boundary conditions for this problem are homogeneous and are incorporated strongly.

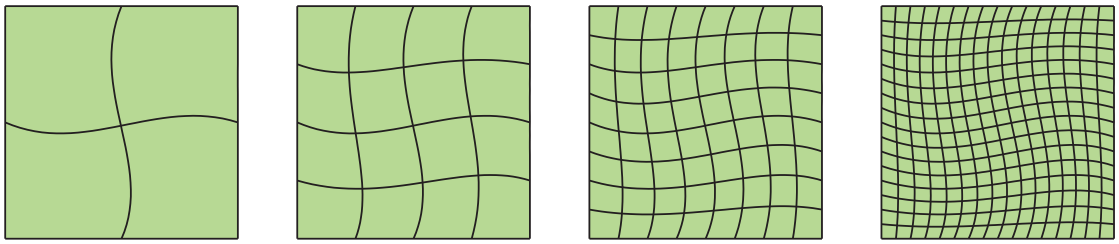
We perform numerical experiments on the geometries and meshes depicted in the Figures 4.15, 4.16, 4.17 and 4.18 below. The geometry in Figure 4.18 is self overlapping, which will illustrate the robustness of the numerical method.



(a) mesh 1 (b) mesh 2 (c) mesh 3 (d) mesh 4

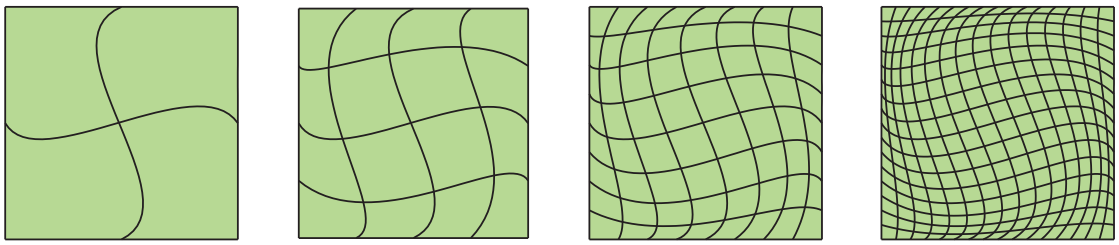
Figure 4.15: Meshes for *Curved0*, produced with global h-refinement

Figure 4.19 shows the difference between the exact solution and the numerical approximation, in the $L_2(\Omega)$ norm. The maximum step width h_{max} is determined as the largest diagonal



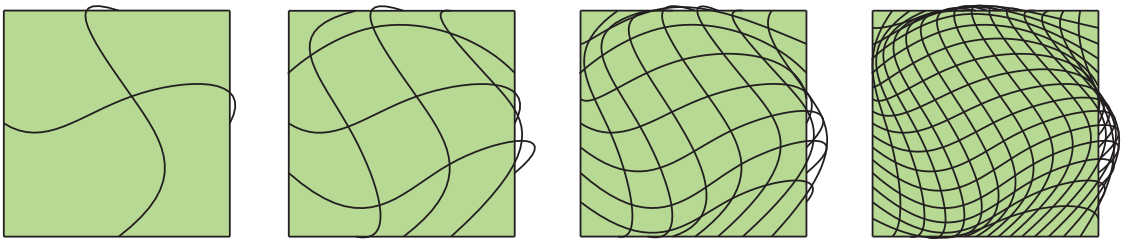
(a) mesh 1 (b) mesh 2 (c) mesh 3 (d) mesh 4

Figure 4.16: Meshes for *Curved1*, produced with global h-refinement



(a) mesh 1 (b) mesh 2 (c) mesh 3 (d) mesh 4

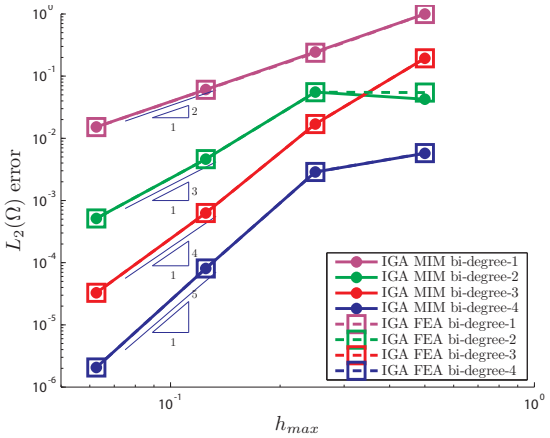
Figure 4.17: Meshes for *Curved2*, produced with global h-refinement



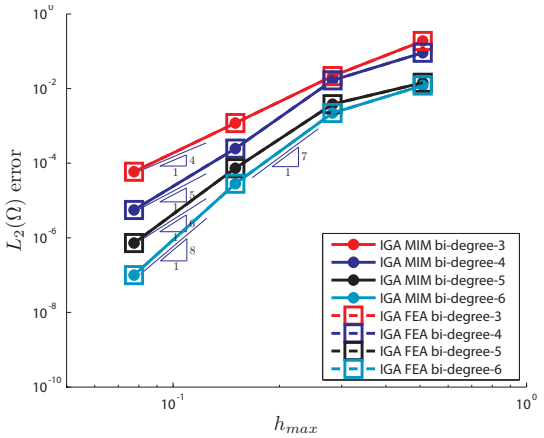
(a) mesh 1 (b) mesh 2 (c) mesh 3 (d) mesh 4

Figure 4.18: Meshes for *Curved3*, produced with global h-refinement

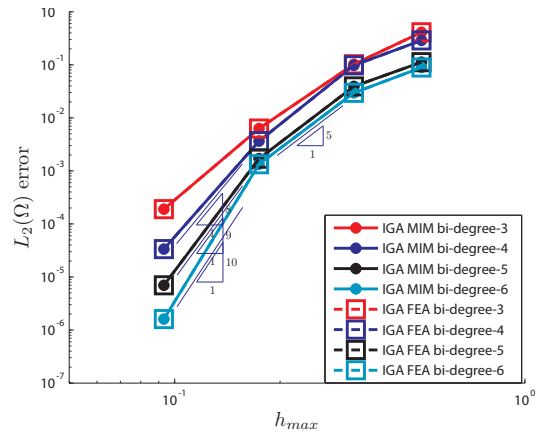
of all elements in the mesh, divided by the square root of two. We compare a Galerkin based IsoGeometric Analysis approach (denoted by IGA FEA) with our mimetic discretization (denoted by IGA MIM). Almost identical results are obtained. This is not unexpected since in case of the scalar Poisson equation for a 0-form, the Mimetic discretization proposed here is equal to the weak form (see (4.89) to (4.89)). There are however two differences in discretization. First of all, derivatives are represented differently. Opposed to the conventional approach, we have the advantage that derivatives can be stated in either continuous or in reduced discrete form, resulting in strong conservation properties. Secondly, we calculate the inner product in curved coordinates, while a conventional FEA approach applies the push forward and calculates inner products in physical coordinates.



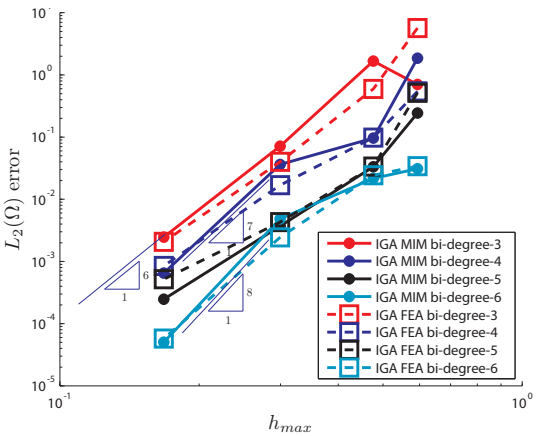
(a) Convergence of $\varphi^0(\mathbf{y})$ in the L_2 norm on *Curved0*



(b) Convergence of $\varphi^0(\mathbf{y})$ in the L_2 norm on *Curved1*



(c) Convergence of $\varphi^0(\mathbf{y})$ in the L_2 norm on *Curved2*



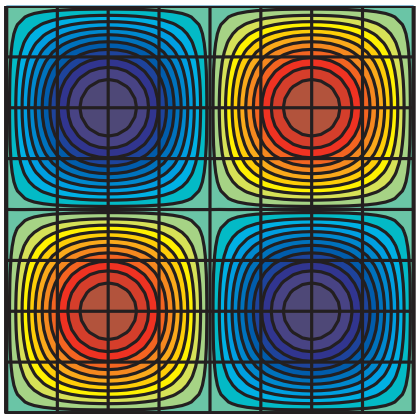
(d) Convergence of $\varphi^0(\mathbf{y})$ in the L_2 norm on *Curved3*

Figure 4.19: Convergence results for $\varphi_h^0(\mathbf{y})$ with respect to step size. IGA MIM stands for the IsoGeometric Mimetic discretization approach proposed here, and IGA FEA stands for the Galerkin approach. Almost identical results are obtained for both methods. Observe that convergence rates start sub-optimal on the curved geometries. After sufficient refinement, however, the convergence rates are above optimal.

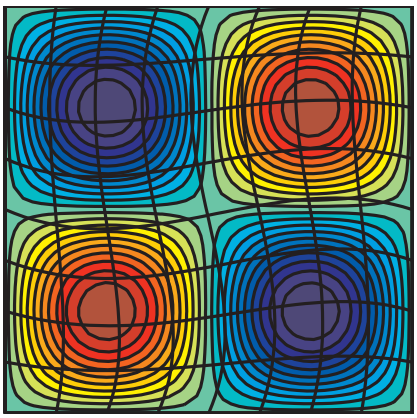
For the linear geometry of *Curved0* we obtain the expected convergence rates $O(h^{p+1})$. The convergence rates for the non-linear geometries start with suboptimal. This is not unexpected, since a non-linear geometry map may deteriorate the approximation power. It is however interesting to see that, once the mesh has been sufficiently refined, the convergence is above optimal.

Figure 4.20 shows a contour plot of the numerical solution of φ^0, for the different geometries. Quite good results are obtained even for the self overlapping geometry. Sufficient refinement is however needed. We note that a Galerkin based IsoGeometric Analysis approach will give identical results. The fact that good results are obtained, even on these ridiculous geometries,

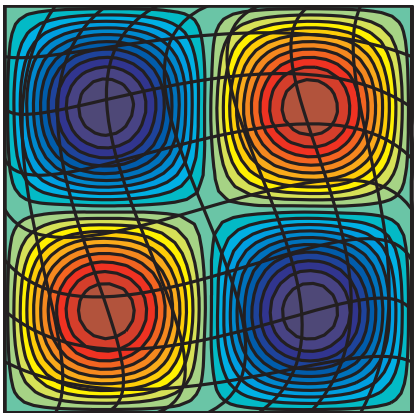
illustrate the robustness of B-spline basis functions.



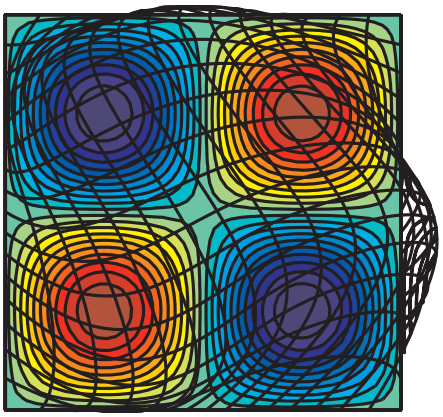
(a) Results on *Curved0*, mesh 3, bi-degree 3



(b) Results on *Curved1*, mesh 3, bi-degree 3



(c) Results on *Curved2*, mesh 3, bi-degree 3



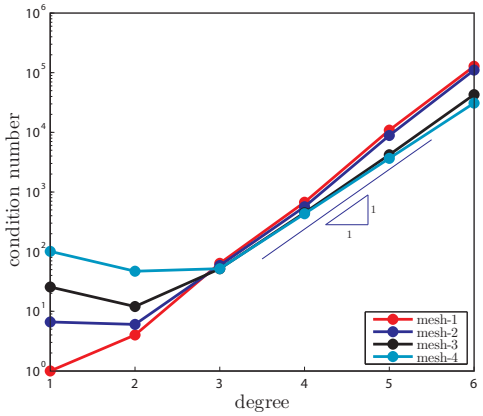
(d) Results on *Curved3*, mesh 4, bi-degree 3

Figure 4.20: Contour plot of $\varphi_h^0(\mathbf{y})$ for the different geometries. Although *Curved3* is self overlapping, the obtained results on the fine mesh seem quite good.

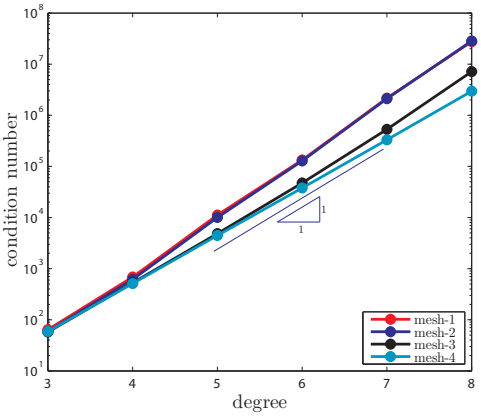
Figure 4.21 illustrates the conditioning of the system matrix as a function of the polynomial degree p. These numbers seem quite disturbing. The condition number increases exponentially, with slope p. The bad conditioning can be found in the mass matrices (inner product matrices - the Hodge) and is solely due to the choice of basis functions: B-splines. Galerkin based IsoGeometric Analysis features the same bad conditioning. This is probably the biggest disadvantage of using B-splines or NURBS as basis functions.

Although the conditioning with respect to the polynomial order is very bad, it does not seem to influence accuracy. The approximation keeps converging towards the exact solution with good rates of convergence, see Figures 4.19 (a) to (d). This is quite a remarkable observation. The explanation can be found in the variation diminishing property of B-splines, see Chapter 3, page 58. For high degree B-splines, a change in the coefficients leads to a much smaller

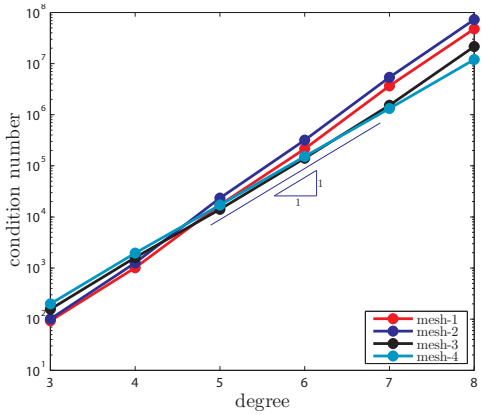
change in the solution. This is opposite to the general understanding of bad conditioning where a small change in the coefficients leads to excessive changes in the solution.



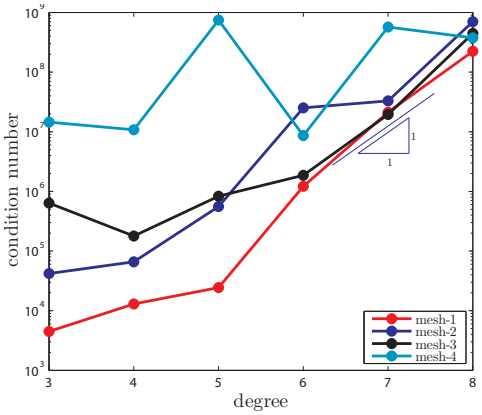
(a) Results on *Curved0*



(b) Results on *Curved1*

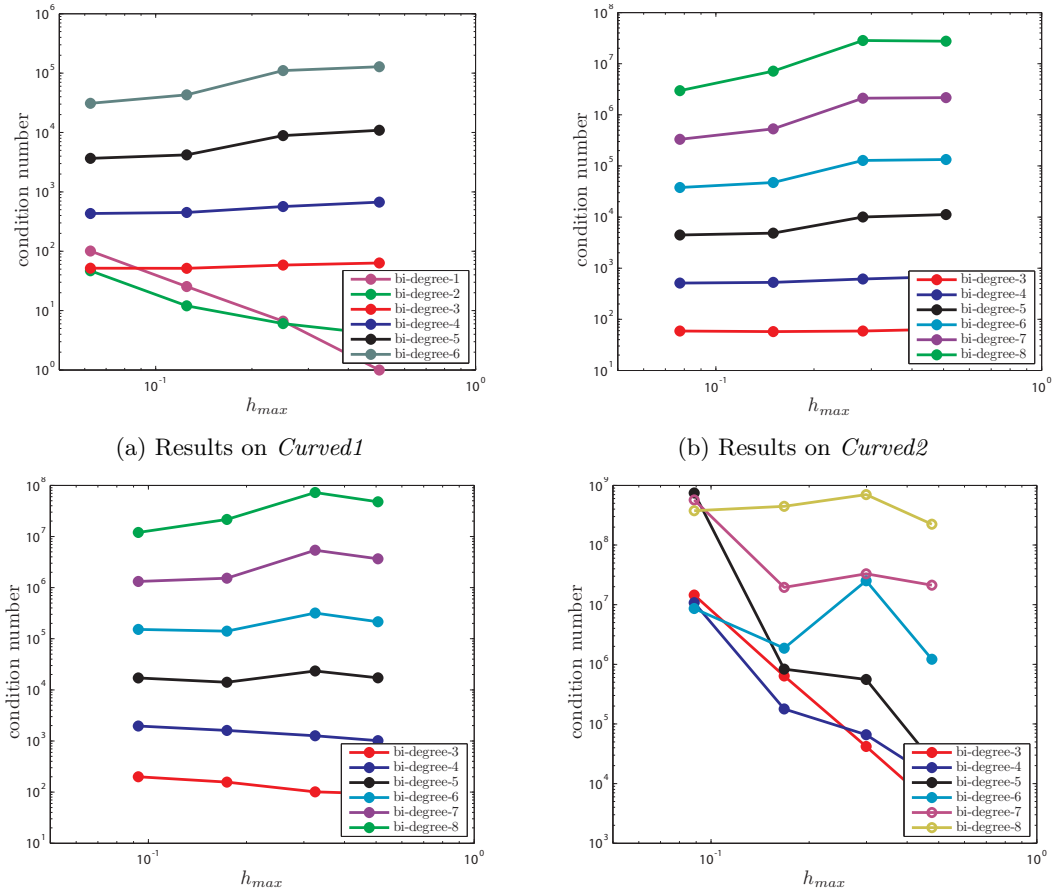


(c) Results on *Curved3*



(d) Results on *Curved4*

Figure 4.21: Condition number of system matrix with respect to the polynomial degree.



(a) Results on *Curved1*

(b) Results on *Curved2*

(c) Results on *Curved3*

(d) Results on *Curved4*

Figure 4.22: Condition number of system matrix with respect to the mesh size.

While the conditioning with respect to the polynomial order is lousy, the conditioning with respect to mesh-size is excellent, see Figure 4.22. The conditioning of the system stays more or less constant under h-refinement.

Although the bad conditioning with respect to the polynomial order, doesn't seem to directly effect the accuracy, it is still an important factor for iterative solvers. IsoGeometric discretizations based on B-splines or NURBS, employing inner product mass matrices, will thus in practice be confined to low to medium order.

In the next chapter we shall apply the IsoGeometric Mimetic discretization method to some practical problems involving incompressible irrotational flows over a lifting cylinder and NACA0012 airfoil.

Chapter 5

Practical Applications in 2D

In this chapter we apply the mimetic discretization method, based on B-spline basis functions, on potential flow problems that include lift. Of course these kind of problems can easily be tackled by conventional boundary element methods; the motivation of this exercise, however, is to study the behavior of the so called harmonic function.

According to the classical Helmholtz Hodge decomposition, every quantity can be decomposed into three basic components: a curl free, a divergence free and a harmonic part.

$$u = \text{grad}\varphi + \text{curl}\psi + h \tag{5.1}$$

Since curl $\cdot$ grad and div $\cdot$ curl is zero by definition, gradφ and curlψ are respectively the curl and divergence vanishing components. The final part, h, is called the harmonic function and is both mass and rotationally conservative. These components are geometrically intuitive as well as useful in practice.

Examining 5.1 in case u represents velocity, then φ is the velocity potential, ψ is the stream-function and h is the harmonic component of velocity. The harmonic function is responsible for a net global circulation in a non-contactable domain, a domain enclosing a hole. An example could be an airfoil under an angle of attack, where a net amount of global circulation around the airfoil causes a net amount of lift. From the Kutta-Joukowsky theorem, 5.2, we know that the amount of lift L produced by an airfoil or any other lift producing object, is proportional to the circulation Γ around the object.

$$L = \rho_\infty V_\infty \Gamma \tag{5.2}$$

where ρ_∞ is the density of the medium and V_∞ the freestreem velocity.

Switching to the notation we use in our Differential Geometry framework, we can write Helmholtz-Hodge decomposition of a 1-form as

$$u^{(1)} = \mathrm{d}\varphi^0 + \mathrm{d}^*\psi^2 + h^1 \tag{5.3}$$

When the flow is considered to be irrotational, meaning $\mathrm{d}u^{(1)} = 0$, the hodge decomposition simplifies to:

$$u^{(1)} = \mathrm{d}\varphi^{(0)} + h^{(1)} \tag{5.4}$$

In this case all of $u^{(1)} = 0$ is harmonic. Potential flow can thus be divided into a non-circulatory flow component and a circulatory flow component. An example for potential flow around a lifting airfoil is shown in Figure 5.1.

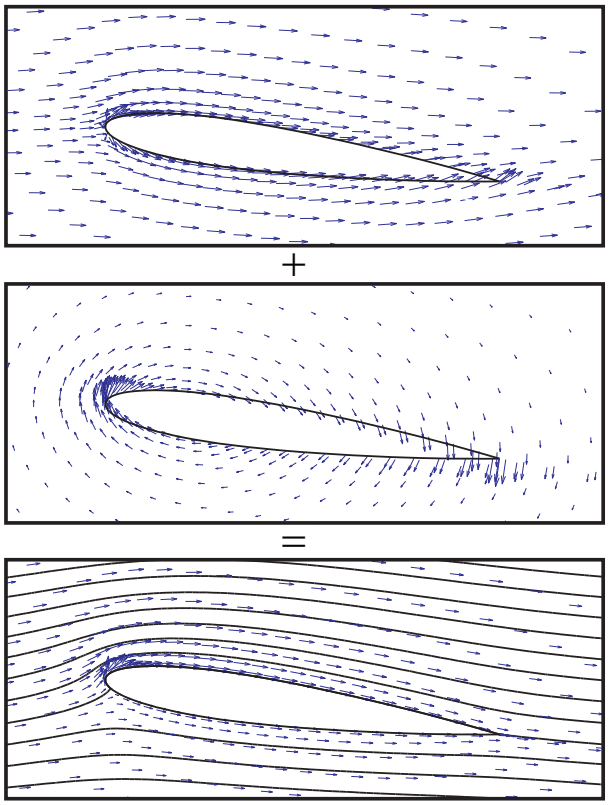


Figure 5.1: Hodge decomposition of an inviscid flow around a lifting airfoil. (Top) Contribution due to gradient of potential. (Middle) Contribution due to harmonic function. (Bottom) Resulting flow.

The circulation can be calculated by integrating the velocity along any closed path enclosing the airfoil. Then since $a = b$ for any closed path

$$\oint_a^b u^{(1)} dr = \oint_a^b \left(\mathrm{d}\varphi^{(0)} + h^{(1)}\right) dr = \varphi_b - \varphi_a + \oint_a^b h^{(1)} dr = \oint_a^b h^{(1)} dr \tag{5.5}$$

Thus if we know the harmonic function, we know the circulation and we can determine the lift according to 5.2.

Before we dwell further into the derivation of the harmonic function, let's take a look at how a conventional FEM would tackle a potential flow problem that includes lift. Suppose the

harmonic function could be written as the differential of a potential, then

$$\oint_a^b h^{(1)} dr = \oint_a^b \mathrm{d}\varphi^{(0)} dr = \varphi^{(0)}(b) - \varphi^{(0)}(a) \tag{5.6}$$

This can only be nonzero for any closed path if a discontinuity of the potential exist at $a = b$. Conventional approaches make use of this by making an artificial cut in the domain, from the trailing edge to the boundary. This will allow a discontinuity in the potential to exist and thus the simulation of lift. This engineering approach might still work well in case the domain under consideration features one hole. In the case of multiple holes, this approach becomes difficult if not impossible.

In this chapter we will explicitly calculate the harmonic function up to a constant and subsequently use the result in our discretization. The model problems we use contain one hole only. Multiple holes, for example an airfoil with flap, will not pose any problems using our approach.

5.1 Discrete representation of the harmonic form

We seek a finite dimensional representation of the continuous harmonic form in terms of discrete degrees of freedom - a co-chain - and associated basis functions. Obviously the basis functions have been chosen and what rests is to find the harmonic cochain. In this section we shall discuss how the harmonic cochain can be derived, up to a constant. This constant is proportional to the circulation.

The discrete analogue to (5.4) is

$$\bar{u}^{(1)} = \mathbb{D}^{(1,0)} \bar{\varphi}^{(0)} + \bar{h}^{(1)} \cdot \bar{c}. \tag{5.7}$$

where $\bar{u}^{(1)} \in \mathcal{C}^{(1)}$. From (5.7) we can observe that the discrete space of 1-cochains $\mathcal{C}^{(1)}$ is spanned by the columns of $\mathbb{D}^{(1,0)}$ and the harmonic co-chain $\bar{h}^{(1)}$. Hence, we can use the columns of $\mathbb{D}^{(1,0)}$ and $\bar{h}^{(1)}$ as a linear independent basis for $\mathcal{C}^{(1)}$. The number of column vectors of $\bar{h}^{(1)}$ is equal to the number of holes in the mesh. So for the numerical simulation of an airfoil with flap the harmonic cochain has two columns.

The velocity 1-form $\bar{u}^{(1)}$ should satisfy the two conditions:

$$\text{Irrotational flow:} \quad \mathbb{D}^{(2,1)} \bar{u}^{(1)} = \mathbb{D}^{(2,1)} \bar{h}^{(1)} \cdot \bar{c} = 0 \tag{5.8}$$

$$\text{Mass conservation:} \quad \left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)} \bar{u}^{(1)} = \left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)} \left(\mathbb{D}^{(1,0)} \bar{\varphi}^{(0)} + \bar{h}^{(1)} \cdot \bar{c}\right) = 0 \tag{5.9}$$

In 5.8 we used that $\mathbb{D}^{(2,1)}\mathbb{D}^{(1,0)}$ is zero by construction. So strictly speaking, $\mathbb{D}^{(1,0)}\bar{\varphi}^{(0)}$ nor $\bar{h}^{(1)} \cdot \bar{c}$ need to conserve mass, as long as the sum of the terms is mass conservative. Obviously, in this case $\bar{h}^{(1)}$ is no longer harmonic. It will however still cause a global circulation in a non-contractible domain.

There are now two separate ways in which we can find $\bar{h}^{(1)}$ such that together with the columns of $\mathbb{D}^{(1,0)}$, it spans the whole space of $\mathcal{C}^{(1)}$. The first possibility is to construct $\bar{h}^{(1)}$ such that it conserves mass. In this case we require the metric and the two conditions in 5.8 and 5.9 can be combined to set up the following system of equations:

$$\begin{pmatrix} \mathbb{D}^{(2,1)} \\ \left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)} \end{pmatrix} \bar{h}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.10}$$

As we will shortly see, the dimension of this system is N by $N - 1 + a$; where a denotes the number of holes in the mesh. If $a = 1$, we obtain a square system and we can simply solve for $\bar{h}^{(1)}$. If $a > 1$, the a columns of $\bar{h}^{(1)}$ can be found using singular value decomposition.

In the second approach we relax the condition of mass conservation for $\bar{h}^{(1)}$, and construct its columns such that they are orthogonal to the columns of $\mathbb{D}^{(1,0)}$. In this case the columns of $\mathbb{D}^{(1,0)}$ and $\bar{h}^{(1)}$ also span the whole space of $\mathcal{C}^{(1)}$. Constructing $h^{(1)}$ orthogonal to the columns of $\mathbb{D}^{(1,0)}$ amounts to stating that $\left(\mathbb{D}^{(1,0)}\right)^T \bar{h}^{(1)} = 0$. We can thus find a basis for $\bar{h}^{(1)}$ by solving the following problem,

$$\begin{pmatrix} \mathbb{D}^{(2,1)} \\ \left(\mathbb{D}^{(1,0)}\right)^T \end{pmatrix} \bar{h}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.11}$$

Note that (5.11) seems quite similar to (5.10). However, while (5.10) requires the metric (Hodge star), (5.11) is metric free. The second approach yields a form of mass conservation, where the Hodge is the identity matrix. We shall illustrate the second approach in an example.

Example 1 Consider a non-contractible cell complex which has been numbered and oriented, see Figure 5.2. This domain is topologically equivalent to the flow domain around for example an airfoil.

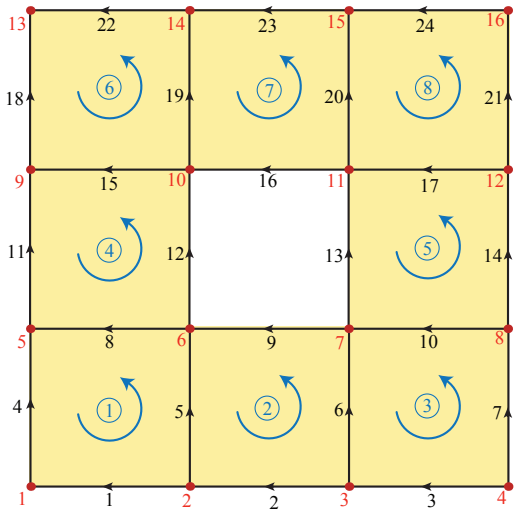


Figure 5.2: Non-contractible cell complex

The topological relations are discretely represented by the incidence matrices. $\mathbb{D}^{(1,0)}$ gives the connectivity between 0-chains and 1-chains (points and lines) and represents the discrete grad.

$$\mathbb{D}^{(1,0)} = \begin{pmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{pmatrix} \tag{5.12}$$

$\mathbb{D}^{(2,1)}$ shows the connectivity between 1-chains and 2-chains (from lines to surfaces) representing the discrete curl

$$\mathbb{D}^{(2,1)} = \begin{pmatrix}
-1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1
\end{pmatrix} \tag{5.13}$$

Constructing the harmonic cochain $\bar{h}^{(1)}$ such that it obeys the condition of irrotational flow and is orthogonal to the columns of $\mathbb{D}^{(1,0)}$, amounts to setting

$$\mathbb{D}^{(2,1)}\ \bar{h}^{(1)} = 0 \qquad\qquad \left(\mathbb{D}^{(1,0)}\right)^T\ \bar{h}^{(1)} = 0 \tag{5.14}$$

In the case of the cell complex shown in figure 5.2, we have 16 0-chains, 24 1-chains and 8 2-chains. This means that the first set of equations in 5.14 is of dimension (8 by 24) and the second set of equations has dimension (16 by 24). The harmonic 1-cochain can subsequentally be found by solving a linear system of equations which is of dimension (24 by 24).

$$\begin{pmatrix} \mathbb{D}^{(2,1)} \\ \left(\mathbb{D}^{(2,1)}\right)^T \end{pmatrix} \begin{pmatrix} \bar{h}^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.15}$$

More generally, if we consider the topology to be Cartesian, we have N by M 0-chains, $M\,(N-1) + N\,(M-1)$ 1-chains and $(M-1)\,(N-1) - a$ number of 2-chains, where a is the number of holes present in the geometry. The final system we have to solve has the following number of rows and columns

- rows: $N \cdot M + (M-1)\,(N-1) - a = 2 \cdot N \cdot M - N - M - a + 1$

- columns: $M\,(N-1)+N\,(M-1)=2\cdot N\cdot M-N-M$

Thus in the case of 1 hole, we have a square system, which we can solve up to a constant. In the case of multiple holes in the domain, the harmonic function is a linear combination of a basis vectors, which can be found using singular value decomposition (SVD). The harmonic co-chain belonging to the cell complex of figure 5.2 is found by solving the system of equations in 5.15. The result is given by

$$\bar{h}^{(1)} = (\ 1\quad 2\quad 1\quad 1\quad 1\quad -1\quad -1\quad 1\quad 4\quad 1\quad 2\quad 4\quad -4\quad -2\quad -1\quad -4\quad -1\quad 1\quad 1\quad -1\quad -1\quad -1\quad -2\quad -1\ )^T \tag{5.16}$$

One can readily check that both $\mathbb{D}^{(2,1)}\ \bar{h}^{(1)}=0$ and $\left(\mathbb{D}^{(1,0)}\right)^T\ \bar{h}^{(1)}=0$ by substituting.

Interpolation of the harmonic 1-cochain in (5.16), using the edge type of B-spline basis functions reveals that the motion is oscillatory around the hole, see Figure 5.3. Important is to note that in curved space, the flow is able to cross the boundary, since we did not set up any boundary conditions. The potential φ^0 will however be calculated such that u^1 satisfies the boundary conditions.

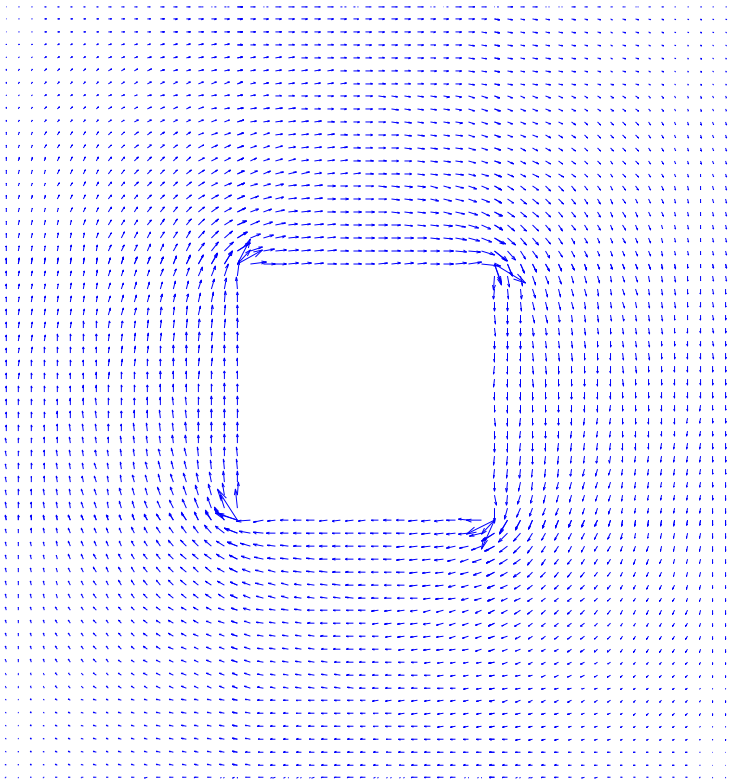


Figure 5.3: Oscillatory flow around the hole in the non-contractible domain of figure 5.2

As is illustrated above, it is quite straightforward to find the harmonic cochain using our discrete framework. We emphasize however that the harmonic co-chain can only be found by using global relations. A conventional FEM or FVM will consequently not be able to find the harmonic function directly, since discrete relations are only locally defined.

5.1.1 The harmonic cochain under mesh refinement

In order to obtain the harmonic cochain we need to solve an additional system of linear equations, as described above. For a refined mesh this can become very expensive, especially when multiple holes exist and we need to use SVD. It is however possible to calculate the harmonic cochain on the initial coarse mesh and subsequently refine it, to find a basis on the fine mesh.

This is possible because we can determine $\bar{h}^{(1)}$ using only topological structures (5.11). Since topology doesn't change under mesh refinement, the columns of the refined harmonic cochain $\bar{h}^{(1)}_{ref}$ will stay orthogonal to the refined columns of $\mathbb{D}^{(1,0)}_{ref}$; secondly $\bar{h}^{(1)}_{ref}$ will stay irrotational. We used the transformation rules of B-splines to refine the harmonic cochain. The difference in normalization between ordinary B-splines and the Curry Schoenberg B-splines should be taken into account.

5.2 Kutta condition

Since we need to solve for one additional degree of freedom - the constant $\bar{c}$ associated with the harmonic cochain $h^{(1)}$ - we need to impose an extra condition on the flow. This is the so called Kutta condition which selects how the flow leaves the object under consideration. In case of lifting potential flow around a blunt body, the Kutta condition is given by a stagnation point on the object. Consider the flow around a cylinder, Figure 5.4, where we require a stagnation point at P_{stag}. Since the velocity in the direction of $\mathbf{n}$ is already zero, because no flow is allowed to cross the boundary, we need only to impose zero velocity in the direction of $\mathbf{t}$.

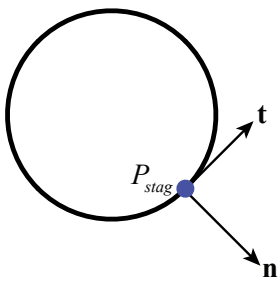


Figure 5.4: Kutta condition for the cylinder. We will enforce a stagnation point at P_{stag}. Since the velocity in the direction of $\mathbf{n}$ is already zero due to Neumann BC's, we only need to pose zero velocity in the direction of $\mathbf{t}$.

In case of an airfoil with a finite angle trailing edge, the Kutta condition is not given by a stagnation point. Consider Figure 5.5, which depicts the NACA0012 at an incidence of 10 deg. We can not place a stagnation point at the trailing edge TE of the airfoil, since in this case we need to impose two conditions: (1) velocity in the direction of $\mathbf{t}$ is zero; and (2) the velocity in the direction of $\mathbf{n}$ is zero.

Exactly the same as for the cylinder, we impose the velocity in the direction of $\mathbf{t}$ to be zero. This will enforce the flow to leave the trailing edge smoothly in the direction of $\mathbf{n}$.

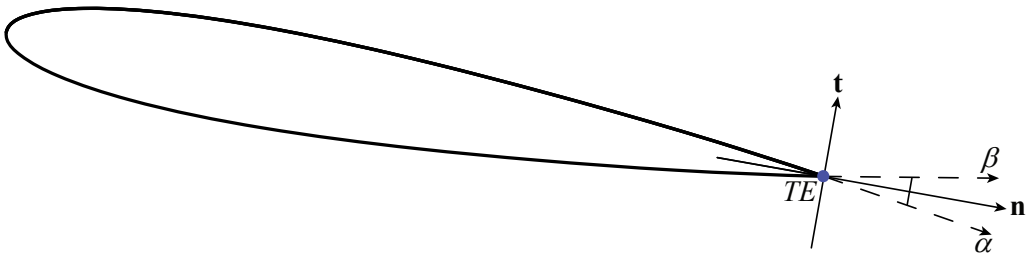


Figure 5.5: Kutta condition for the NACA0012. To enforce that the flow leaves the trailing edge smoothly we impose that the velocity in the direction of $\mathbf{t}$ is zero. In this case the flow leaves the trailing edge at TE in the direction of $\mathbf{n}$. $\mathbf{n}$ is thus tangent to the streamline at TE.

Given the geometry in terms of a parametric NURBS map $\mathbf{y} = \mathbf{S}(\mathbf{x})$, stating physical coordinates $\mathbf{y}$ in terms of local coordinates $\mathbf{x}$, we can determine the vector $\mathbf{t}$ at P_{stag} and TE, in Figure 5.4 and 5.5 respectively. In case of the cylinder this is a trivial exercise. As for the airfoil, we first need to determine the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, see Figure 5.5. Once these vectors have been determined, we can calculate the bi-sector $\mathbf{n}$ and subsequently $\mathbf{t}$,

$$\mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} = \frac{1}{2\,\|\boldsymbol{\alpha}\|} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} + \frac{1}{2\,\|\boldsymbol{\beta}\|} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad \Longrightarrow \quad \mathbf{t} = \begin{pmatrix} n_2 \\ -n_1 \end{pmatrix} = n_2 \frac{\partial}{\partial y_1} - n_1 \frac{\partial}{\partial y_2} \tag{5.17}$$

Let us denote the position where the Kutta condition is imposed, whether it is for the cylinder or for the airfoil, by the local coordinate $\mathbf{x}_K$. Then we can calculate the velocity at this point by reconstructing the 1-cochain $\bar{u}^{(1)}$,

$$u_h^1(\mathbf{x}_K) = \mathcal{I}_1\, \bar{u}^{(1)} = \bar{u}_1^T\, \mathbf{R}^{(1)}(\mathbf{x}_K)\mathrm{dx}^1 + \bar{u}_2^T\, \mathbf{R}^{(2)}(\mathbf{x}_K)\mathrm{dx}^2 \tag{5.18}$$

Here $\bar{u}^{(1)} \in \mathcal{C}^{(1)}$ is given by 5.7. The reconstructed velocity $u_h^1(\mathbf{x}_K)$ is formulated in terms of local coordinates. To determine the component of velocity in the direction of $\mathbf{t}$ we need to reformulate to physical coordinates $\mathbf{y}$. This means we need to apply the push forward $\mathbf{S}_*$. By writing the pull back $\mathbf{S}^*$ of a one form as a matrix vector equation, we may take the inverse of the Jacobian matrix $\mathbf{J}$, such that we obtain an equation for the push forward. We already showed this in the previous chapter; for the sake of completeness we repeat it here.

$$\begin{pmatrix} \mathrm{dy}^1 \\ \mathrm{dy}^2 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial S^1}{\partial x_1} & \frac{\partial S^1}{\partial x_2} \\ \frac{\partial S^2}{\partial x_1} & \frac{\partial S^2}{\partial x_2} \end{pmatrix}}_{\mathbf{J}} \begin{pmatrix} \mathrm{dx}^1 \\ \mathrm{dx}^2 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \mathrm{dx}^1 \\ \mathrm{dx}^2 \end{pmatrix} = \underbrace{\frac{1}{\det \mathbf{J}} \begin{pmatrix} \frac{\partial S^2}{\partial x_2} & -\frac{\partial S^1}{\partial x_2} \\ -\frac{\partial S^2}{\partial x_1} & \frac{\partial S^1}{\partial x_1} \end{pmatrix}}_{\mathbf{J}^{-1}} \begin{pmatrix} \mathrm{dy}^1 \\ \mathrm{dy}^2 \end{pmatrix}$$

The velocity in the direction of $\mathbf{t}$ can then be written as,

$$\begin{aligned}
\mathbf{S}_* \left(u_h^1(\mathbf{x}_K)\right)^T \cdot \mathbf{t} &= \mathbf{S}_* \begin{pmatrix} \left(\bar{u}_1^T \, \mathbf{R}^{(1)}(\mathbf{x}_K) \quad \bar{u}_2^T \, \mathbf{R}^{(2)}(\mathbf{x}_K)\right) \begin{pmatrix} \mathrm{dx}^1 \\ \mathrm{dx}^2 \end{pmatrix} \end{pmatrix} \left(n_2 \frac{\partial}{\partial y_1} - n_1 \frac{\partial}{\partial y_2} \right) \\
&= \frac{1}{\det \mathbf{J}} \left(\bar{u}_1^T \, \mathbf{R}^{(1)}(\mathbf{x}_K) \quad \bar{u}_2^T \, \mathbf{R}^{(2)}(\mathbf{x}_K)\right) \begin{pmatrix} \frac{\partial S^2}{\partial x_2} & -\frac{\partial S^1}{\partial x_2} \\ -\frac{\partial S^2}{\partial x_1} & \frac{\partial S^1}{\partial x_1} \end{pmatrix} \begin{pmatrix} \mathrm{dy}^1 \\ \mathrm{dy}^2 \end{pmatrix} \left(n_2 \frac{\partial}{\partial y_1} - n_1 \frac{\partial}{\partial y_2} \right) \\
&= \frac{1}{\det \mathbf{J}} \left(\bar{u}_1^T \, \mathbf{R}^{(1)}(\mathbf{x}_K) \quad \bar{u}_2^T \, \mathbf{R}^{(2)}(\mathbf{x}_K)\right) \begin{pmatrix} \frac{\partial S^2}{\partial x_2} & -\frac{\partial S^1}{\partial x_2} \\ -\frac{\partial S^2}{\partial x_1} & \frac{\partial S^1}{\partial x_1} \end{pmatrix} \begin{pmatrix} n_2 \\ -n_1 \end{pmatrix} \\
&= \frac{1}{\det \mathbf{J}} \left(\left(\frac{\partial S^2}{\partial x_2} n_2 + \frac{\partial S^1}{\partial x_2} n_1 \right) \left(\mathbf{R}^{(1)}(\mathbf{x}_K) \right)^T \quad \left(-\frac{\partial S^2}{\partial x_1} n_2 - \frac{\partial S^1}{\partial x_1} n_1 \right) \left(\mathbf{R}^{(2)}(\mathbf{x}_K) \right)^T \right) \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix} \\
&= \mathbf{K}^T \, \bar{u}^{(1)}
\end{aligned} \tag{5.19}$$

Here $\mathbf{K}^T$ is a row vector. Using 5.7 we can state the Kutta condition in terms of this row vector as,

$$^*\mathbf{S} \left(u_h^1(\mathbf{x}_K)\right)^T \cdot \mathbf{t} = \mathbf{K}^T \, \bar{u}^{(1)} = \mathbf{K}^T \begin{pmatrix} \mathbb{D}^{(1,0)} & h^{(1)} \end{pmatrix} \begin{pmatrix} \bar{\varphi}^{(0)} \\ \bar{c} \end{pmatrix} = 0. \tag{5.20}$$

5.3 Discrete equations for potential flows that include lift

Since we now have an equation for the additional degree of freedom $\bar{c}$ associated with the harmonic cochain, we are able to set up a system of equations, to solve for the unknown potentials $\bar{\varphi}^{(0)}$ and the unknown constant $\bar{c}$. Since the formulation automatically obeys the condition of irrotational flow, we only need to impose mass conservation (5.9) and the Kutta condition (5.20). The final set of equations becomes,

$$\begin{pmatrix} \left(\mathbb{D}^{(1,0)}\right)^T \mathbb{H}^{(\tilde{1},1)} \\ \mathbf{K}^T \end{pmatrix} \begin{pmatrix} \mathbb{D}^{(1,0)} & h^{(1)} \end{pmatrix} \begin{pmatrix} \bar{\varphi}^{(0)} \\ \bar{c} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.21}$$

We note that boundary conditions still need to be set.

5.4 Determination of the lift

In order to determine the lift we integrate the harmonic part of velocity over a closed contour (5.5). We take the boundary of the object under consideration as the contour of integration. Assume that the object is described by a 2D NURBS map $\mathbf{y} = \mathbf{S}(x_1, 0)$. By keeping the x_2 direction constant, the map describes the boundary curve.

The harmonic part of velocity at the boundary is then only a function of x_1 and can be

reconstructed as,

$$\mathcal{I}_1\ \bar{c}\cdot\bar{h}^{(1)} = \bar{c}\cdot\sum_{i=1}^{n_1}\bar{h}_i^1\ M_i(x_1)\mathrm{dx}^1 + \bar{c}\cdot\sum_{i=0}^{n_1} c_0\cdot\bar{h}_i^2\ N_i(x_1)\mathrm{dx}^2. \tag{5.22}$$

Since $\frac{\partial}{\partial x_1}$ is a vector tangent to the boundary, the co-vector dx^2 is normal to the boundary (Chapter 2, definition of the dual space (3) , page 23). We can consequently neglect the second term in (5.22) and the integral of (5.22) is given by,

$$\Gamma = \oint_{x_1}\bar{c}\cdot\sum_{i=1}^{n_1}\bar{h}_i^1\ M_i(x_1)\mathrm{dx}^1 = \bar{c}\cdot\sum_{i=1}^{n_1}\bar{h}_i^1\ \oint_{x_1} M_i(x_1)\mathrm{dx}^1 = \bar{c}\cdot\sum_{i=1}^{n_1}\bar{h}_i^1 \tag{5.23}$$

Here we used the useful property of the Curry Schoenberg B-splines that the integral under every basis function is 1 (property (4.10) introduced in chapter 4, page 84).

EXAMPLE 5.4.1 Figure 5.6 depicts a cylinder and its associated primal cell complex near the boundary. The harmonic cochain $\bar{h}_i$ is associated with control edges. The circulation around the cylinder in Figure 5.6 can be calculated as

$$\Gamma = \bar{c}\cdot\sum_{k=1}^{8}\bar{h}_i. \tag{5.24}$$

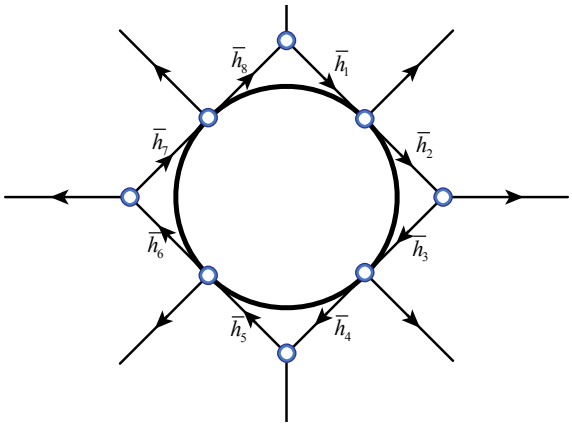


Figure 5.6: The circulation can be calculated by summing the harmonic cochains on the boundary, multiplied by $\bar{c}$.

Since we know the circulation Γ, we can determine the lift according to the Kutta-Joukowski theorem (5.2).

5.5 Lifting flow over a cylinder

In this section we apply the IsoGeometric Mimetic discretization scheme developed in this thesis to the incompressible irrotational flow around a cylinder with lift. We will choose

several different locations of the stagnation point P_{stag} (see Figure 5.7), in order to generate a certain amount of lift, and compare with the analytical solution (see for example Anderson Jr. [5]). We will do so for several refinement stages of the mesh and for several polynomial orders of the B-spline basis.

5.5.1 Problem setup

Figure 5.7 shows the problem setup in polar coordinates (r, θ). We choose the radius of the cylinder as $R = 1$. The free stream variables are chosen as $\rho_\infty = 1$ and $V_\infty = 1$. Furthermore we perform numerical experiments for four different cases of the stagnation point: (1) $P_{stag} = (1, 0)$; (2) $P_{stag} = (1, -1/4\pi)$; (3) $P_{stag} = (1, -1/2\pi)$; and (4) $P_{stag} = (2, -1/2\pi)$;

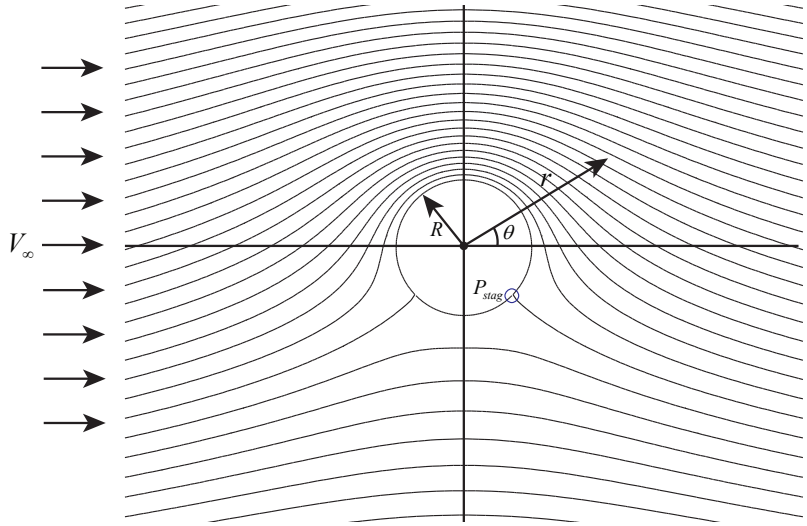


Figure 5.7: Lifting flow over cylinder.

We shall make the comparison with the analytical solution in a number of ways. We will compare the lift coefficient C_L; The drag coefficient C_D, which should obviously be zero; the pressure coefficient C_p at the cylinder wall; and for the uniform flow around the cylinder we calculate the error between the analytical and numerical solution of the potential in the $L_2(\Omega)$ norm as a function of the element size h_{max}. h_{max} is calculated as the maximum diagonal length of all elements in the mesh.

The lift is determined as explained in the previous section. The drag is calculated by integrating the momentum over the outer boundary using high order quadrature. Below you will find the analytical solution to some of the variables we compare with.

Analytical solution to the lift

By the Kutta-Joukowski theorem (5.2), lift L is proportional to the vortex strength Γ. The analytical value of the vortex strength Γ depends on the choice of the stagnation point $P_{stag} =$

(r, θ) by [5],

$$\Gamma\left(P_{stag}(r,\theta)\right) = -4\pi r V_\infty sin(\theta) \tag{5.25}$$

Analytical solution to the velocity potential

For uniform flow around the cylinder we shall compare with the potential in th $L_2(\Omega)$ norm. The analytical solution to the potential of the lifting flow around a circular cylinder can be stated by superimposing a uniform flow, a doublet and a vortex [5]. The result is

$$\Phi(r,\theta) = V_\infty \cdot r\cos(\theta)\left(1+\frac{R^2}{r^2}\right) - \frac{\Gamma}{2\pi}\theta \tag{5.26}$$

Analytical solution to the pressure coefficient at the cylinder wall

We will further compare the numerical value of the pressure coefficient at the boundary of the cylinder with the analytical one. The analytical solution is given by Anderson Jr. [5],

$$C_p(1,\theta) = 1 - 4\sin^2(\theta) + \frac{2\Gamma\sin(\theta)}{\pi R V_\infty} + \left(\frac{\Gamma}{2\pi R V_\infty}\right)^2 \tag{5.27}$$

Boundary conditions

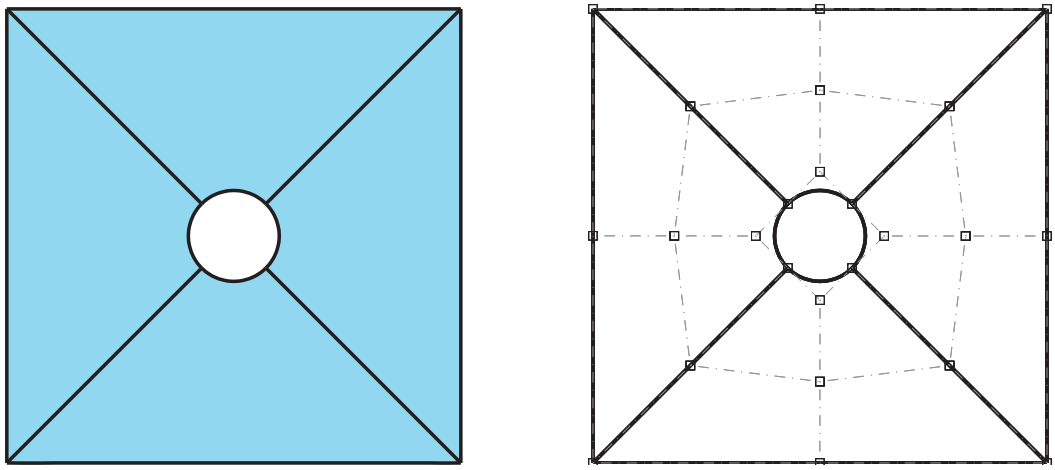
We impose Neumann boundary conditions weakly, both at the outer boundary and at the cylinder wall, by means of a boundary integral containing the analytical solution of the velocity. The analytical solution to the velocity is obtained by differentiating (5.26),

$$\begin{aligned} V_r(r,\theta) &= \left(1-\frac{R^2}{r^2}\right) V_\infty\cos(\theta) \\ V_\theta(r,\theta) &= -\left(1+\frac{R^2}{r^2}\right) V_\infty\sin(\theta) - \frac{\Gamma}{2\pi r} \end{aligned} \tag{5.28}$$

In chapter 4, Section 4.5.2, page 107, it is explained how to implement these boundary conditions.

5.5.2 NURBS geometry and mesh generation

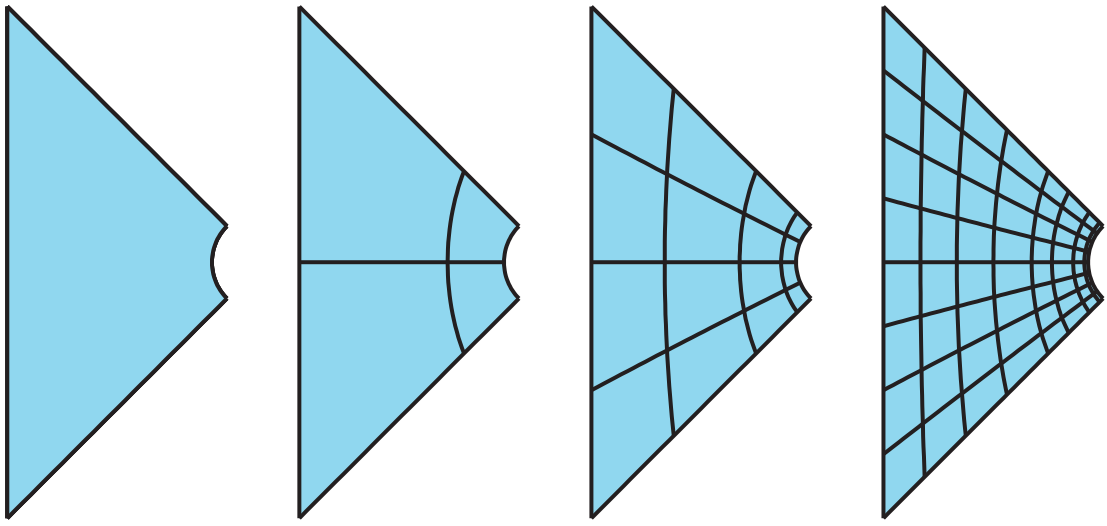
The flow domain around the circular cylinder is modeled in a CAD environment using four bi-quadratic NURBS patches, see Figure 5.8. Chapter 3, Section 3.4 explains how such a geometry can be constructed. The cylinder with radius $R = 1$ is exactly described at the coarsest level of discretization. The flow domain is of size 5 by 5.



(a) Coarse mesh (b) NURBS control mesh

Figure 5.8: NURBS 2D geometric domain around a cylinder of radius $R = 1$. The geometry is modeled in a CAD program using four patches. The size of the domain is 5 by 5. The meshes are obtained by global h-refinement with increased resolution near the cylinder wall.

Numerical calculations are performed using the initial coarse mesh and for three stages of refinement. The meshes are obtained using global refinement, in such a way that additional resolution is provided near the cylinder wall, see Figure 5.9



(a) mesh 1 (b) mesh 2 (c) mesh 3 (d) mesh 4

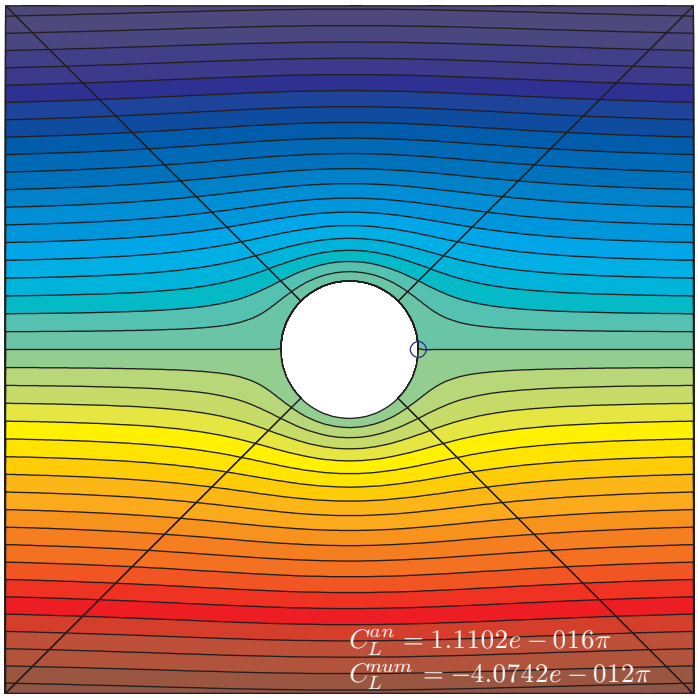
Figure 5.9: Each patch is refined for analysis. Calculations are performed on the above meshes.

Table 5.1 states the number of degrees of freedom, for the different meshes and polynomial orders we have used.

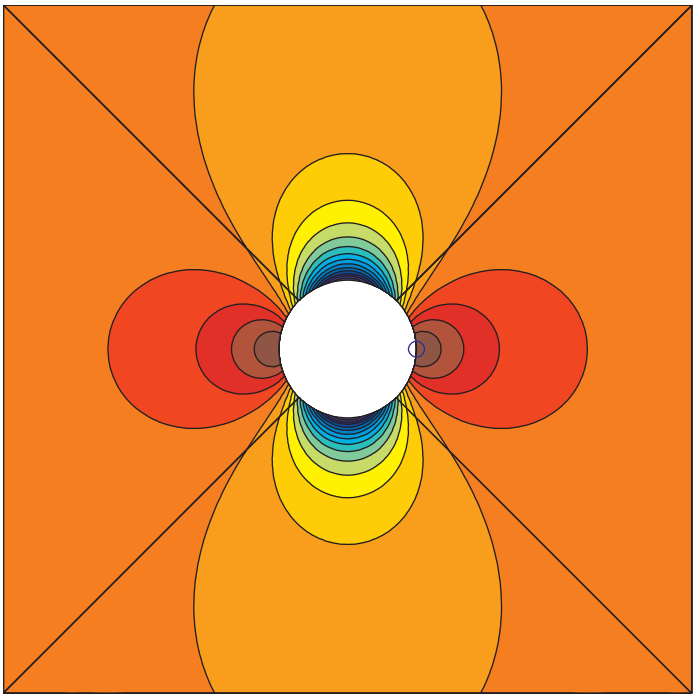
Table 5.1: Lifting flow around the cylinder. Number of degrees of freedom for mesh 1 to 4 and polynomial degree 2 to 6.

degree	mesh 1	mesh2	mesh 3	mesh 4
2	25	49	121	361
3	49	81	169	441
4	81	121	225	529
5	121	169	289	625
6	169	225	361	729

5.5.3 Results for uniform flow around the cylinder

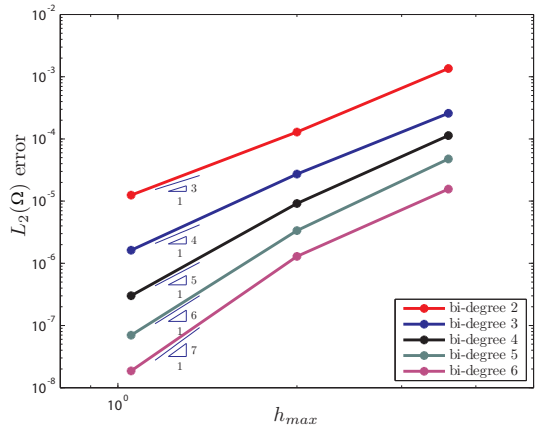


(a) streamlines of the flow

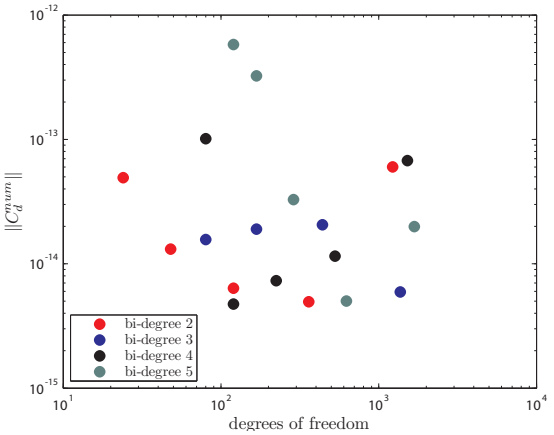


(b) iso-contours of pressure

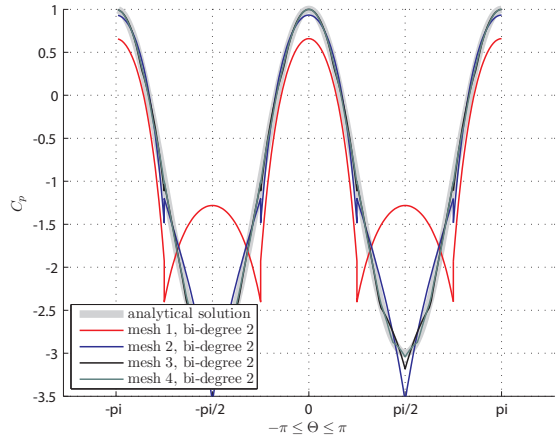
Figure 5.10: Uniform flow around a cylinder. The blue circle denotes the stagnation point, which is chosen at $R = 1$ and $\theta = 0\,\mathrm{deg}$. The results have been obtained for bi-degree 6 mesh 3.



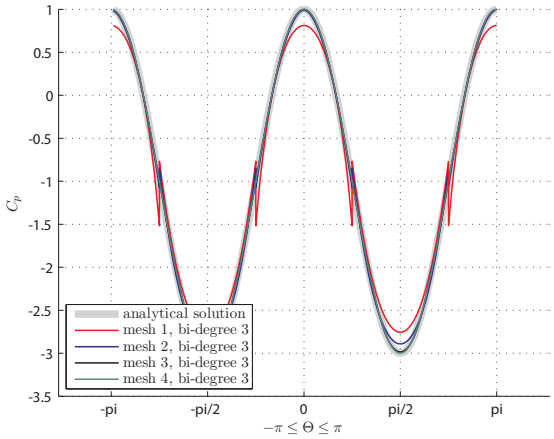
(a) $L_2(\Omega)$ error in the potential



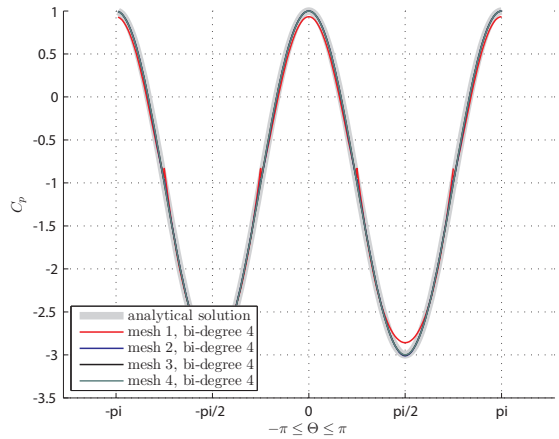
(b) drag coefficients



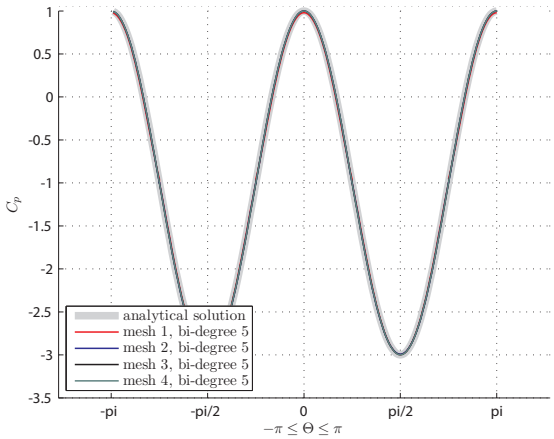
(c) C_p plotted verses θ



(d) C_p plotted verses θ



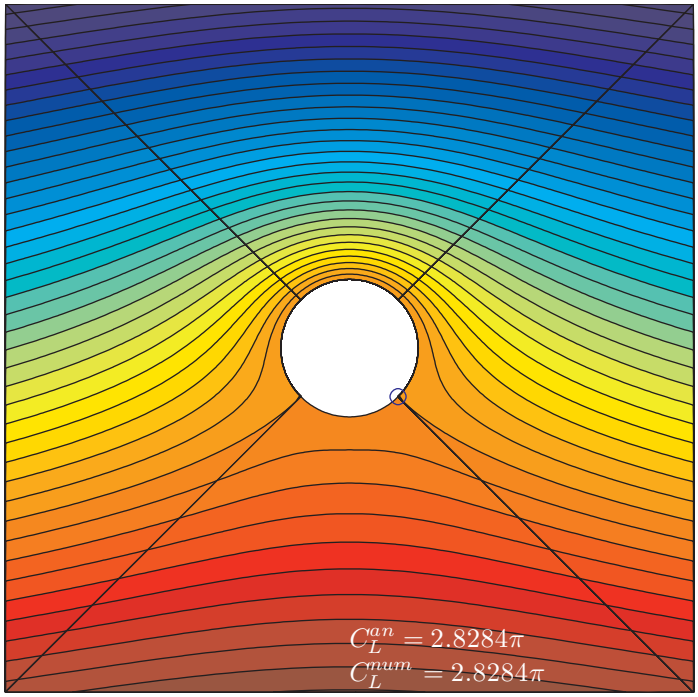
(e) C_p plotted verses θ



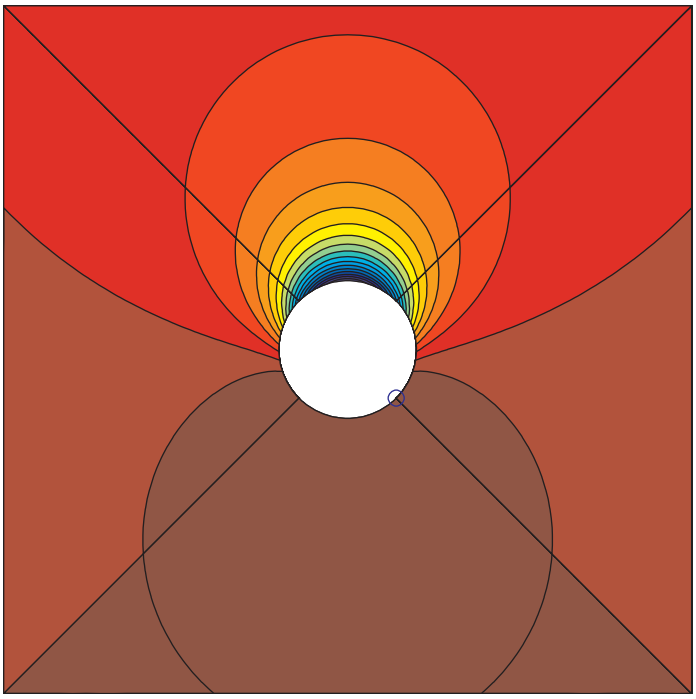
(f) C_p plotted verses θ

Figure 5.11: Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (1, 0)$. Figures (a) depicts the $L_2(\Omega)$ error in the potential. Optimal convergence results are obtained. Figure (b) depict the numerical results for the drag coefficients, which are close to machine precision. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ

5.5.4 Results for lifting flow over the cylinder with $P_{stag} = (1, -1/4\pi)$

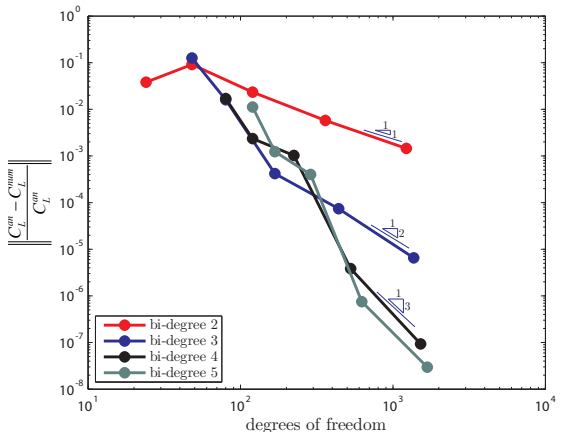


(a) streamlines of the flow

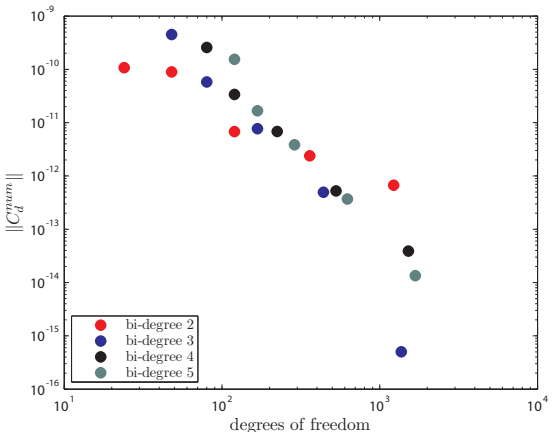


(b) iso-contours of pressure

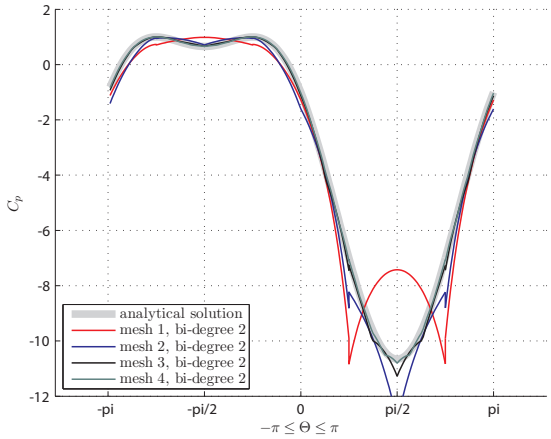
Figure 5.12: Potential flow around a lifting cylinder. The stagnation point is chosen at $R = 1$ and $\theta = -1/4\pi$. The results have been obtained for mesh 3, bi-degree 6.



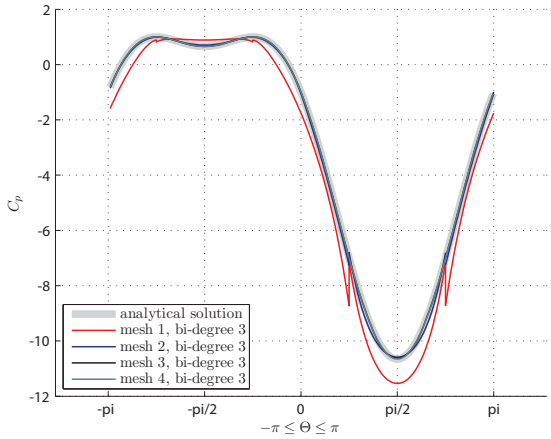
(a) lift coefficients



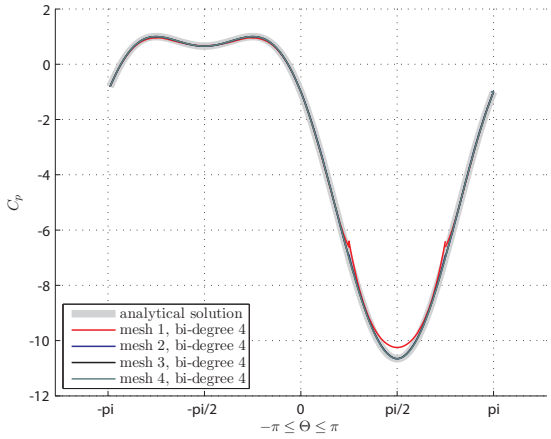
(b) drag coefficients



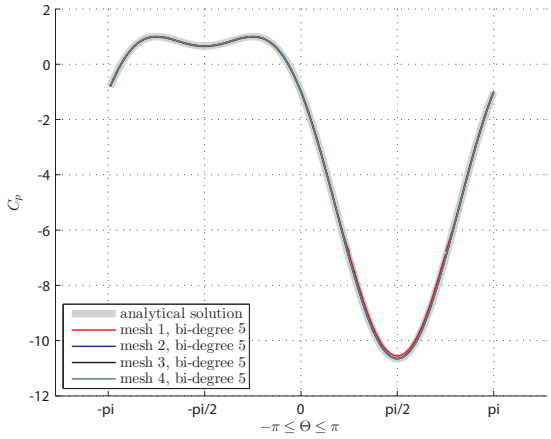
(c) C_p plotted verses θ



(d) C_p plotted verses θ



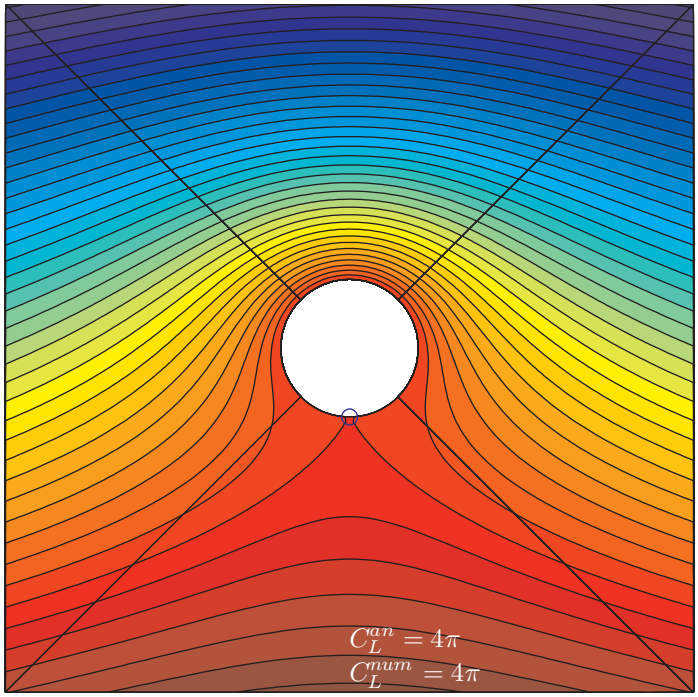
(e) C_p plotted verses θ



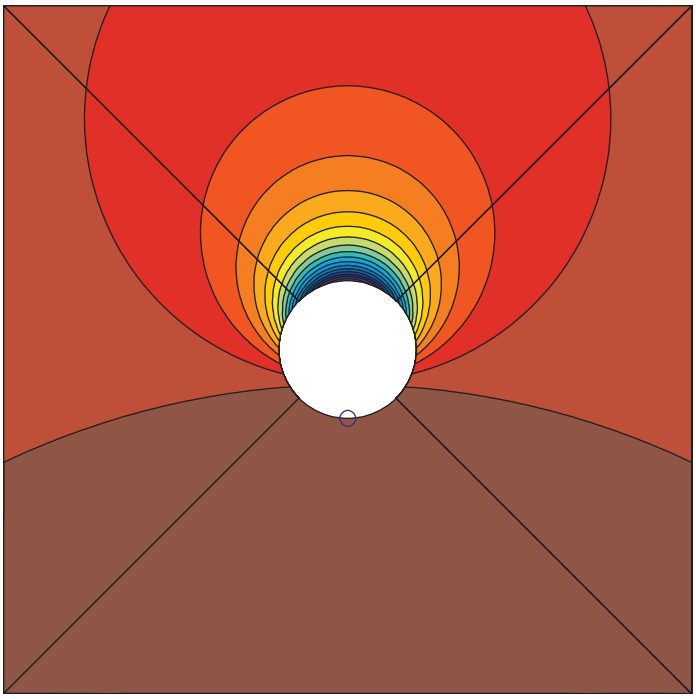
(f) C_p plotted verses θ

Figure 5.13: Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (1, -1/4\pi)$. Figures (a) and (b) depict the numerical results for the lift and drag coefficients. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ.

5.5.5 Results for lifting flow over the cylinder with $P_{stag} = (1, -1/2\pi)$

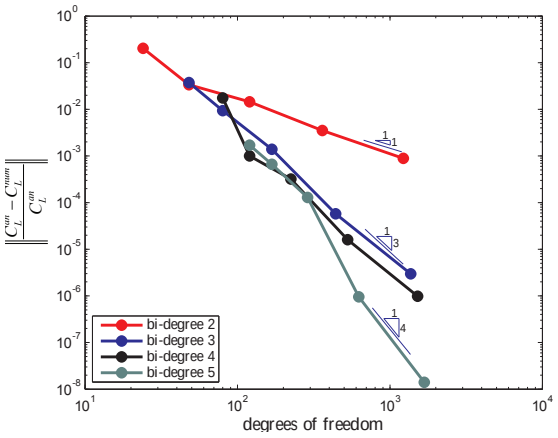


(a) streamlines of the flow

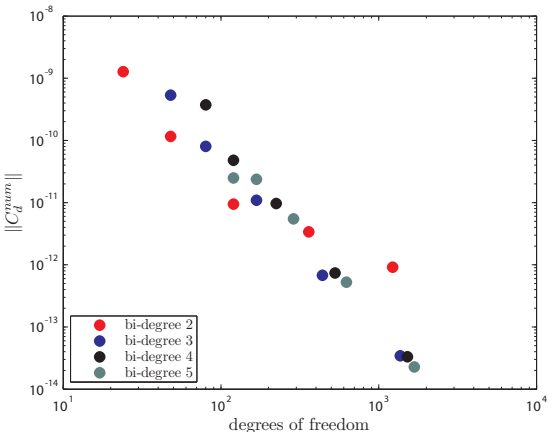


(b) iso-contours of pressure

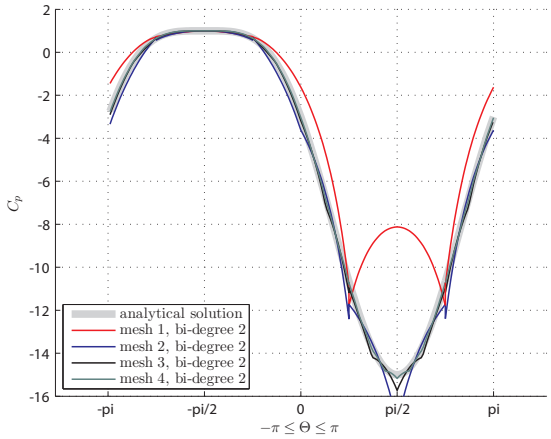
Figure 5.14: Potential flow around a lifting cylinder. The stagnation point is chosen at $R = 1$ and $\theta = -1/2\pi$. The results have been obtained for mesh 3, bi-degree 6.



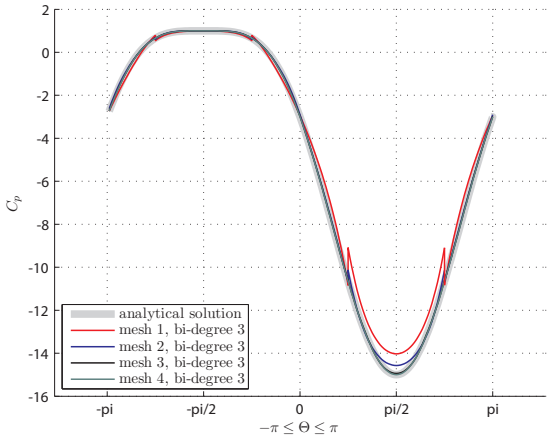
(a) lift coefficients



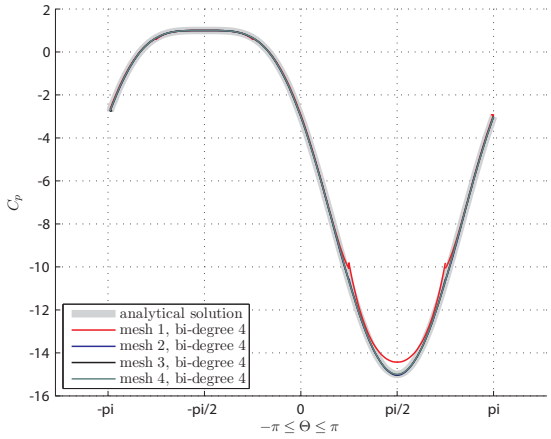
(b) drag coefficients



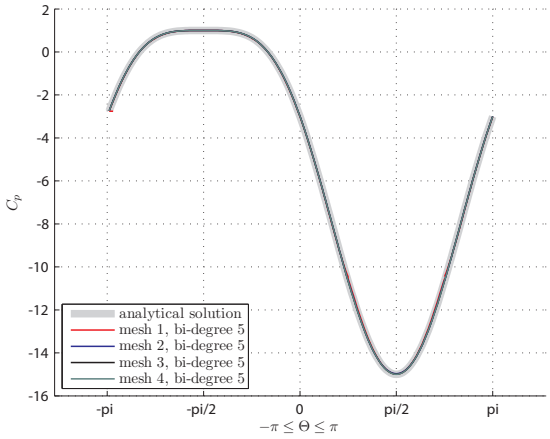
(c) C_p plotted verses θ



(d) C_p plotted verses θ



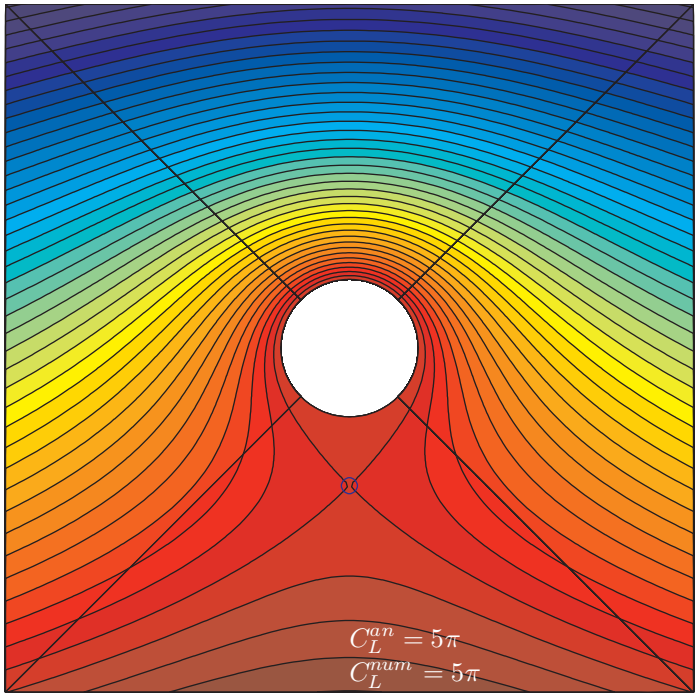
(e) C_p plotted verses θ



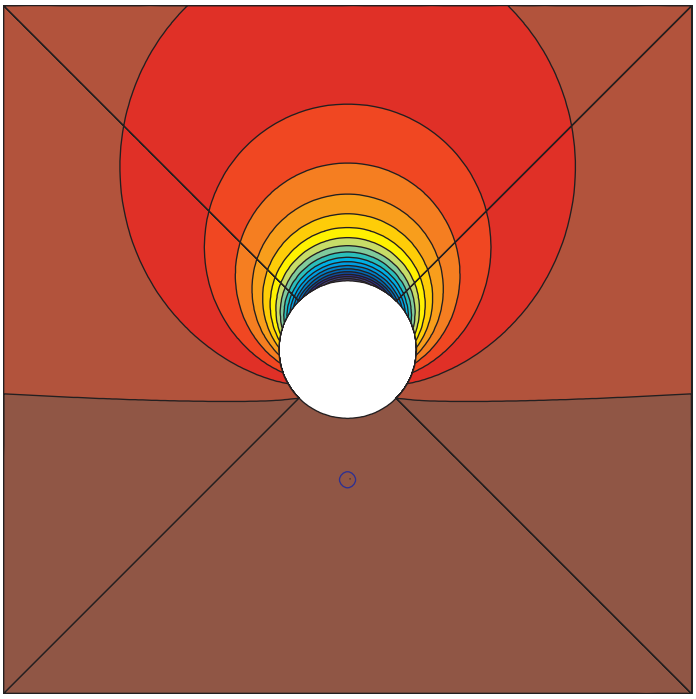
(f) C_p plotted verses θ

Figure 5.15: Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (1, -1/2\pi)$. Figures (a) and (b) depict the numerical results for the lift and drag coefficients. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ.

5.5.6 Results for lifting flow over the cylinder with $P_{stag} = (2, -1/2\pi)$

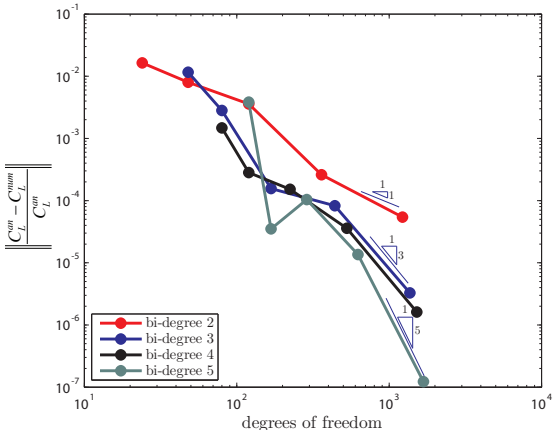


(a) streamlines of the flow

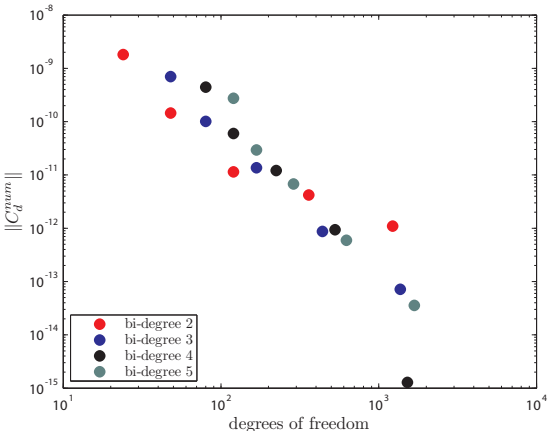


(b) iso-contours of pressure

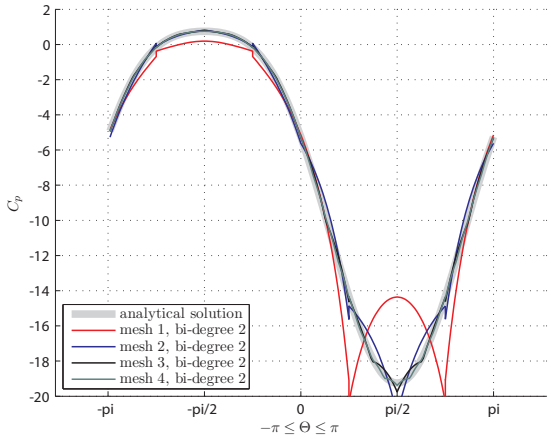
Figure 5.16: Potential flow around a lifting cylinder. The stagnation point is chosen at $R = 2$ and $\theta = -1/2\pi$. The results have been obtained for mesh 3, bi-degree 6.



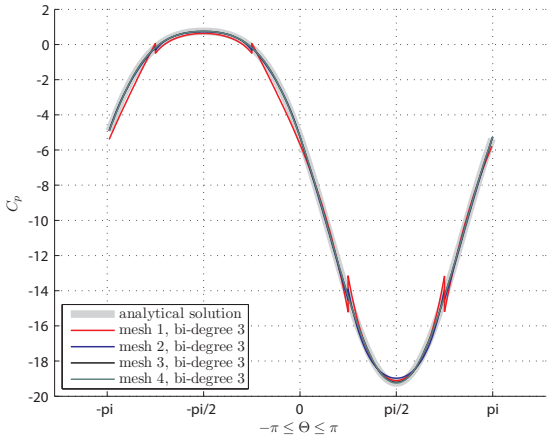
(a) lift coefficients



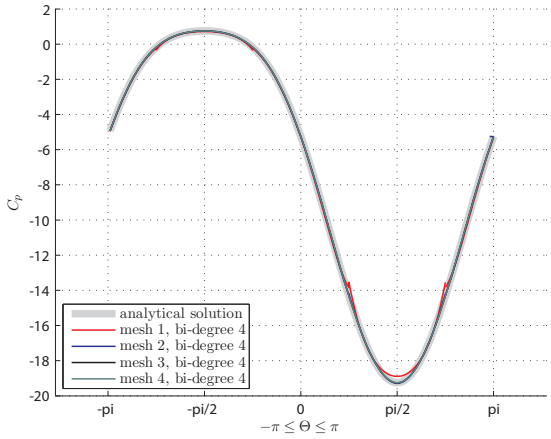
(b) drag coefficients



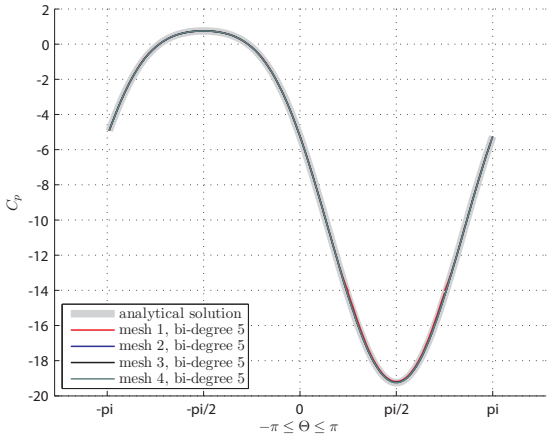
(c) C_p plotted verses θ



(d) C_p plotted verses θ



(e) C_p plotted verses θ



(f) C_p plotted verses θ

Figure 5.17: Convergence results for lifting flow over a cylinder with stagnation point $P_{stag} = (2, -1/2\pi)$. Figures (a) and (b) depict the numerical results for the lift and drag coefficients. Figures (c) to (f) compare the numerical results for the pressure coefficient at the cylinder wall against the analytical solution for the different meshes and order of the basis. The pressure coefficient C_p is plotted in polar coordinates against angle θ.

5.5.7 Discussion

Good results have been obtained for the lifting flow around the cylinder. In this case we can compare with an exact solution, which gives us a clear image of how well our numerical method performs for these kind of problems. The results for the lifting flow over the cylinder show similar trends for all choices of the stagnation point.

We can observe in 5.13(a), 5.15 (a) and 5.17(a) that we are able to accurately determine the lift. The relative error in the lift on the coarse mesh is in the order of 10^{-2}, which is already quite small. The lift seems to converge in most cases at a slope of p.

We further studied the drag, which in the analytical case is zero by d'Alembert's paradox. The numerical drag coefficients are in all cases reasonably close to zero. The absolute value of the drag coefficient seems to be dependent on the total number of degrees of freedom and not on the polynomial order of approximation.

The increased continuity of the B-spline basis clearly contributes to capturing the qualitative behavior of the pressure. The Quadratic B-spline expansion of the potential φ_h^0, is C^1 continuous between elements. The velocity - the spatial derivative of the potential - is consequently piecewise linear with C^0 continuity between elements. The pressure, which is proportional to the square of velocity, is then also C^0 continuous. This low order continuity gives the numerical solution the freedom to behave totally different then the analytical solution. This is clearly visible in Figures 5.11(c), 5.13(c), 5.15 (c) and 5.17(c), which depict the pressure coefficient at the cylinder wall. While the analytical solution shows convex behavior, the numerical solution is concave. Refining the mesh shows that the numerical solution converges, however, this opposite behavior does not disappear.

For the higher order C^{p-1} B-spline basis, see Figures 5.11(d-f), 5.13(d-f), 5.15 (d-f) and 5.17(d-f), the pressure at the cylinder wall is at least tangent continuous. In these cases the numerical solution directly captures the qualitative behavior of the pressure. This illustrates the strength of C^{p-1} continuous B-splines as compared to C^0 finite elements.

At patch interfaces, which are situated at $\theta = 1/4\pi \pm 1/2\pi$, the potential is C^0 continuous in all cases. This means that the velocity and thus the pressure are discontinuous at these points. This discontinuity in the pressure coefficient is only visible on coarse meshes and low order approximations.

5.6 Lifting potential flow over the NACA0012 airfoil

Next, we apply the new discretization approach to the lifting flow over a NACA0012 airfoil. Results are obtained for 0,2,4,6 and 8 deg. We compare our results with those obtained from a 2nd order panel method using 1000 panels. We compare the pressure coefficient over the surface of the airfoil and the lift as a function of degrees of freedom. As we did for the cylinder, we calculate the drag.

5.6.1 Problem setup

Figure 5.18 depict the problem setup for the flow around the NACA0012. The flow domain is of size 22 by 20. The airfoil chord is set to $c = 1$. The in- and out flow velocity is set to $V = 1$. The boundary conditions are weakly enforced.

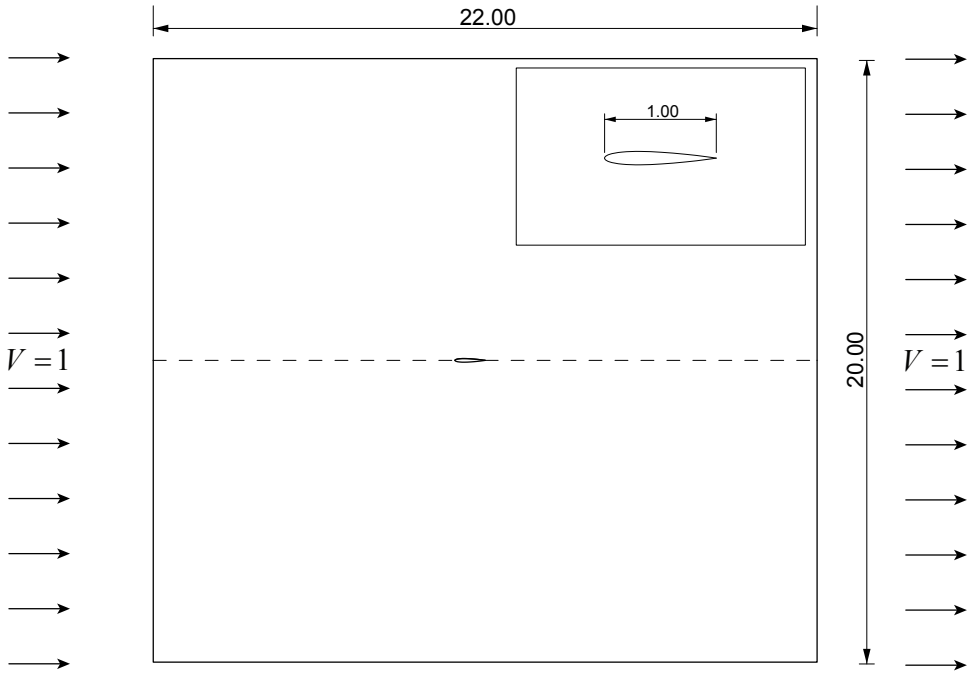
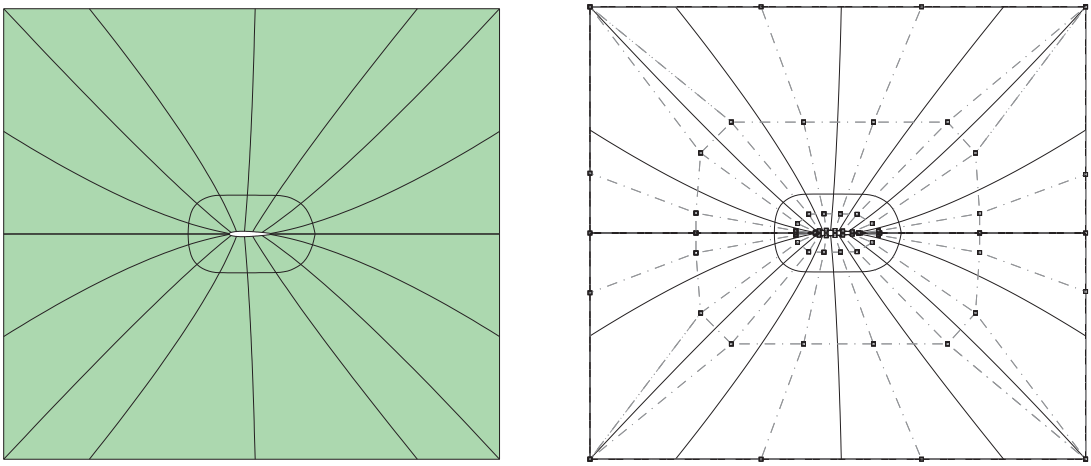


Figure 5.18: Problem setup, flow around the NACA0012

5.6.2 NURBS geometry and mesh generation

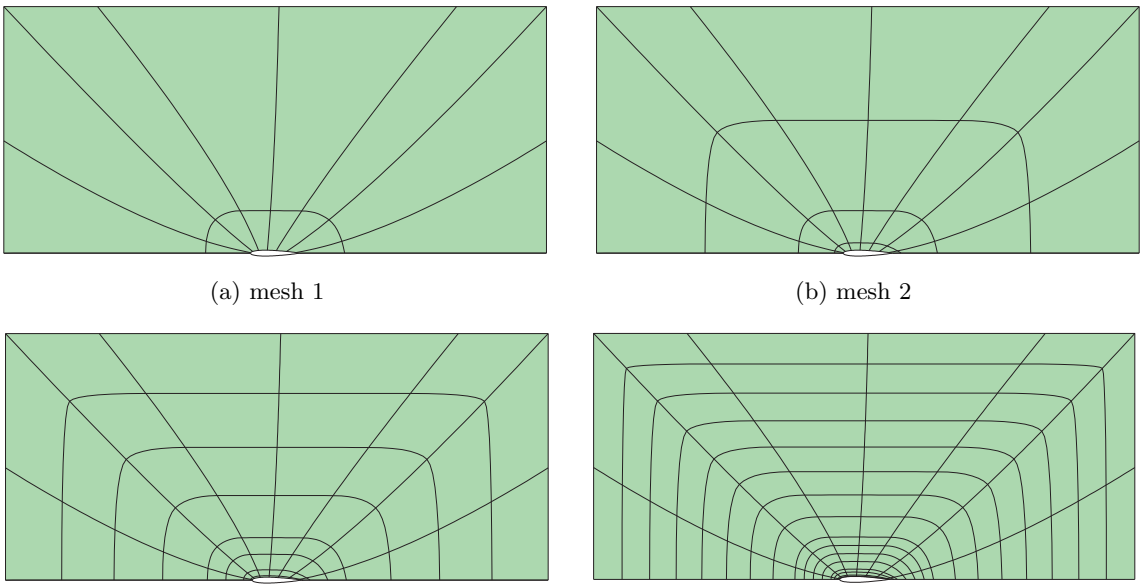
The flow domain, depicted in Figure 5.19, is modeled in a CAD environment using two bi-quadratic patches. The shape of the airfoil has been approximated from a point cloud generated by the analytical formula of the NACA0012. The angle of attack is controlled by rotating the boundary control points around the leading edge.



(a) physical mesh (b) control mesh

Figure 5.19: Coarse model of the flow domain, straight from CAD. Note that the corners have been artificially created by placing two control points at the same location.

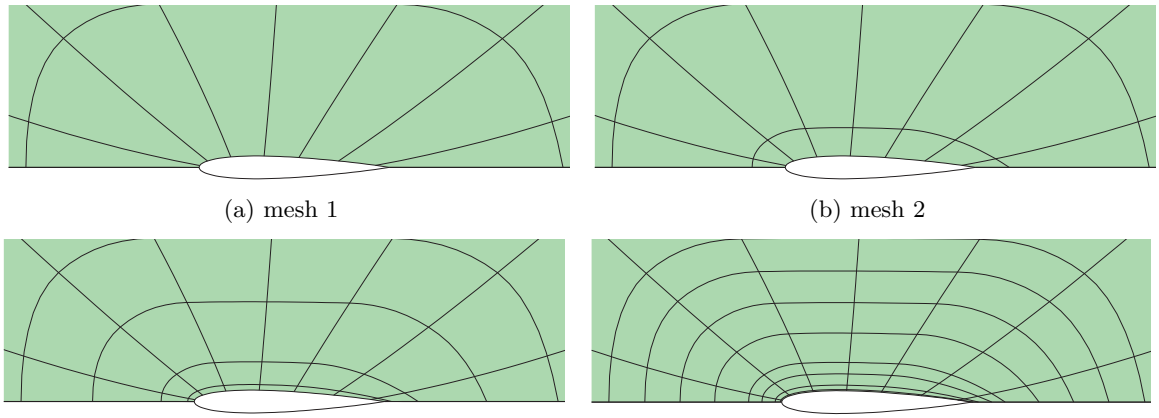
We have performed calculations on the meshes depicted in Figure 5.20. These have been obtained by several stages of global refinement of the coarse mesh. Refinement is only performed in the radial direction, because the tangential direction is considered fine enough from the coarsest discretization on. Increased resolution is provided near the airfoil boundary, see Figure 5.21.



(a) mesh 1 (b) mesh 2

(c) mesh 3 (d) mesh 4

Figure 5.20: Meshes obtained using global h-refinement in the radial direction.



(a) mesh 1 (b) mesh 2

(c) mesh 3 (d) mesh 4

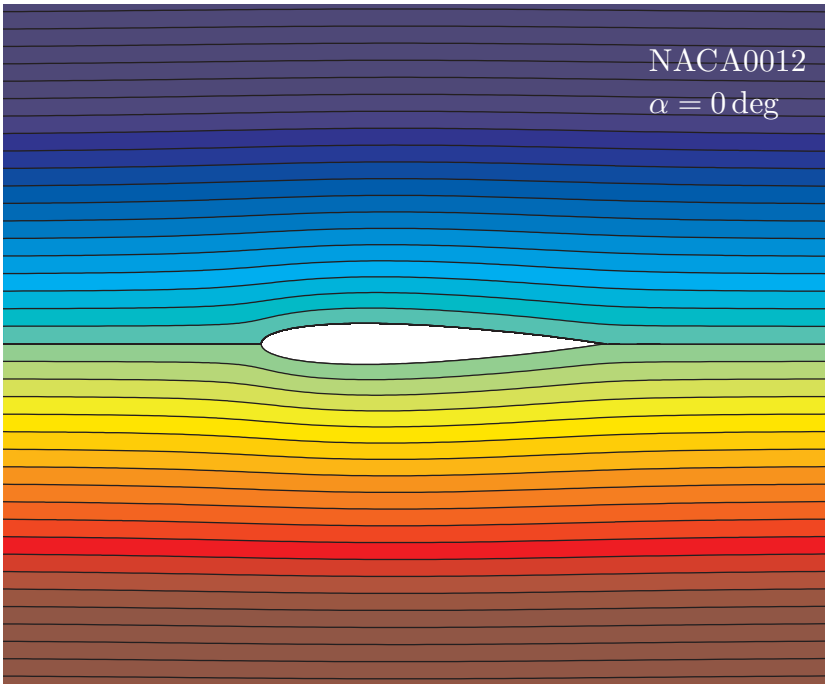
Figure 5.21: A closer look at the airfoil reveals the increased resolution we provided near the boundary

The number of degrees of freedom for the different meshes and different polynomial degrees are depicted in table 5.2. The meshes in 5.20 are used to perform the calculations for an angle of attack of zero degrees. For the other angles of attack, similar meshes have been obtained.

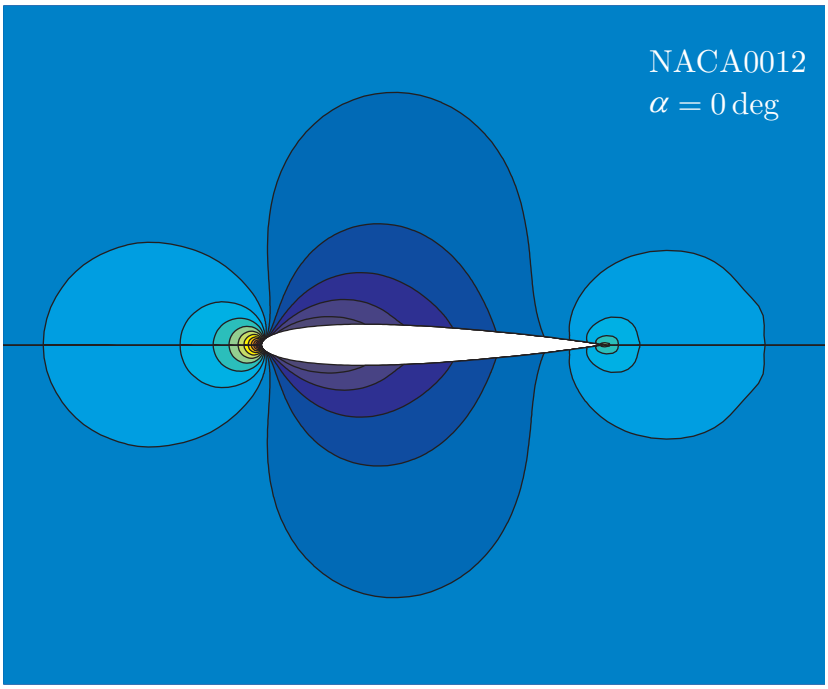
Table 5.2: Lifting flow over the NACA0012. Number of degrees of freedom for mesh 1 to for and polynomial degree 2 to 6.

degree	mesh 1	mesh2	mesh 3	mesh 4
2	73	109	181	325
3	205	273	409	681
4	401	501	701	1101
5	661	793	1057	1585
6	985	1149	1477	2133

5.6.3 Results uniform flow around the NACA0012

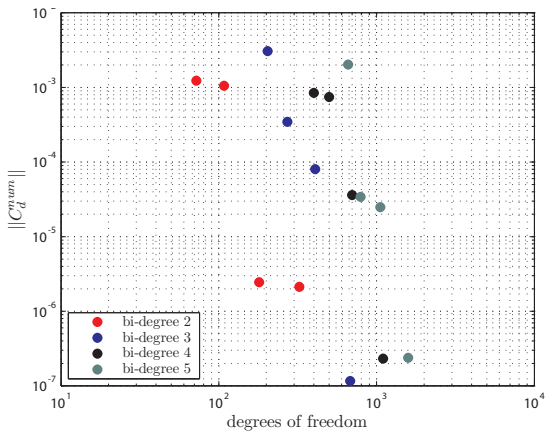


(a) streamlines of the flow

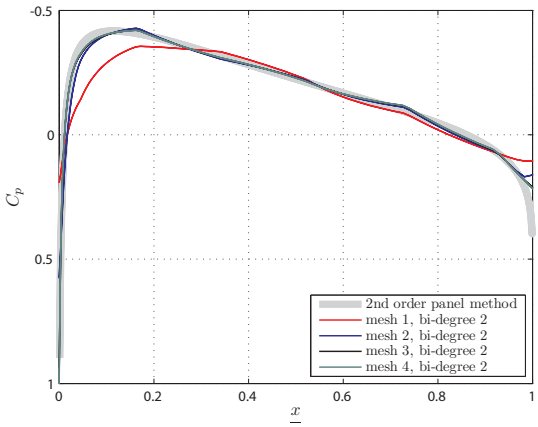


(b) iso-contours of pressure

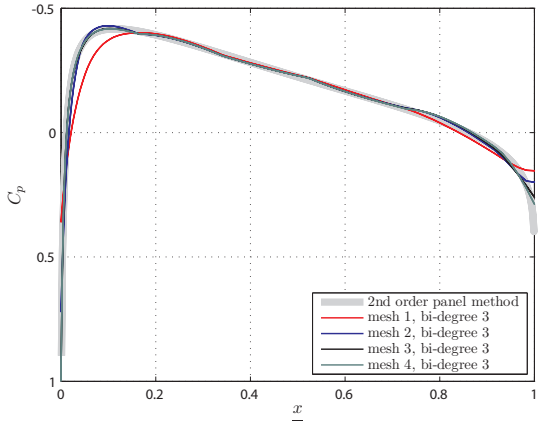
Figure 5.22: Potential flow around a lifting NACA0012 airfoil at 0 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6.



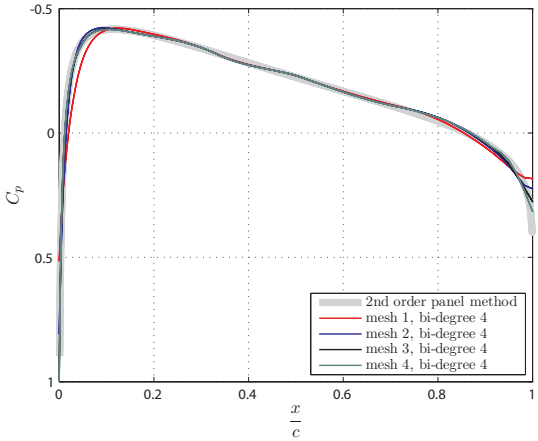
(a) drag coefficients



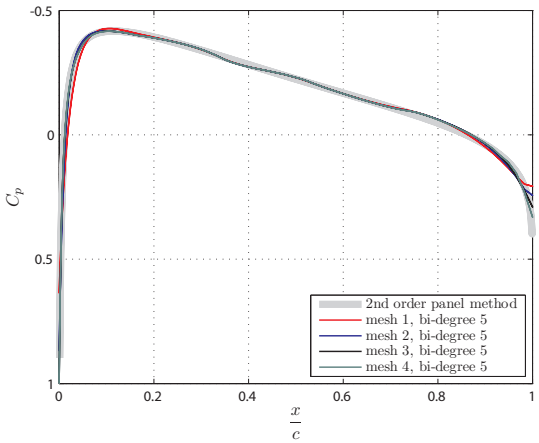
(b) C_p plotted verses $\frac{x}{c}$



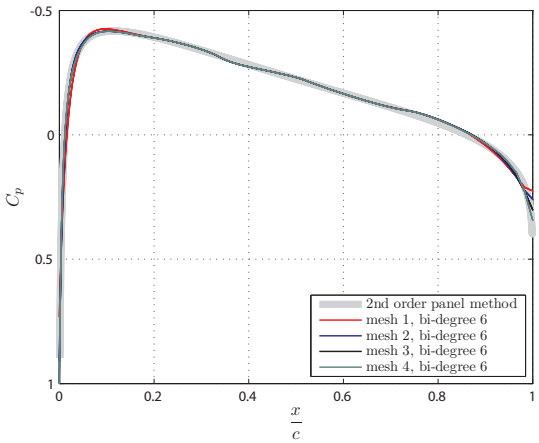
(c) C_p plotted verses $\frac{x}{c}$



(d) C_p plotted verses $\frac{x}{c}$



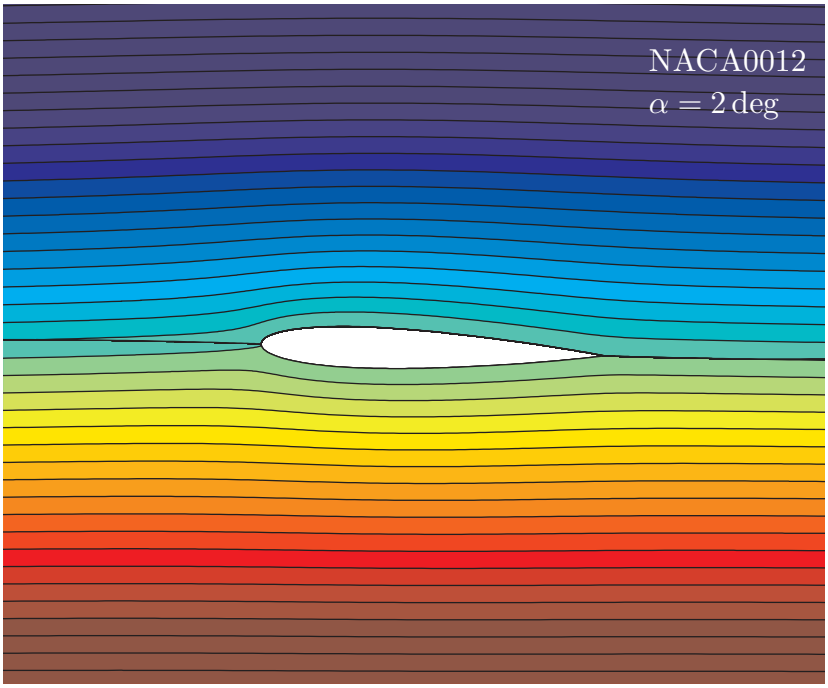
(e) C_p plotted verses $\frac{x}{c}$



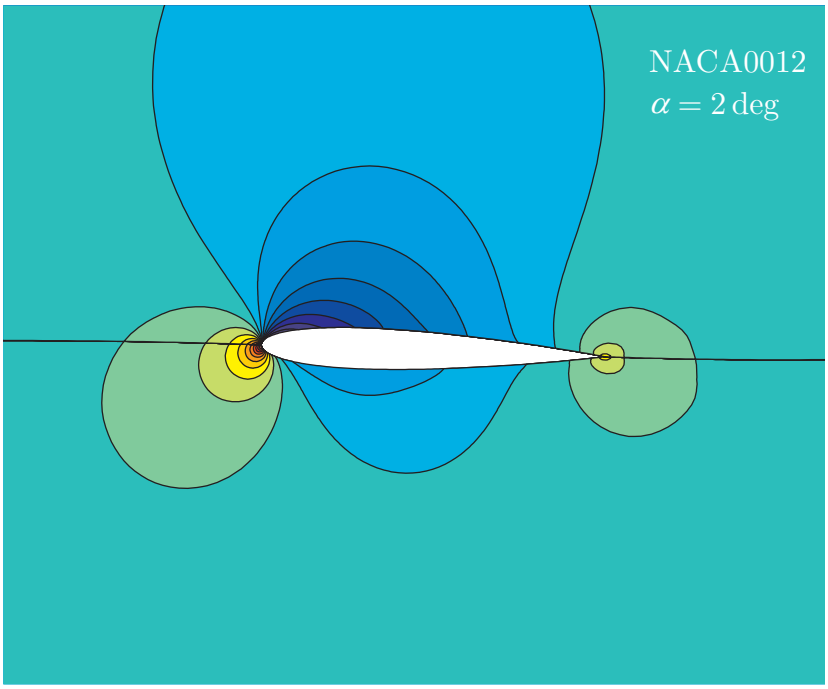
(f) C_p plotted verses $\frac{x}{c}$

Figure 5.23: Uniform flow around the NACA0012. Figure (a) depicts the numerical drag coefficients. These are no longer close to machine precision, as for the cylinder. Figures (b) to (f) depict the pressure coefficient C_p over the airfoil and compares with the results obtained by a 2nd order panel method using 1000 panels.

5.6.4 Results lifting flow over the NACA0012 with $2\,\mathrm{deg}$ of incidence

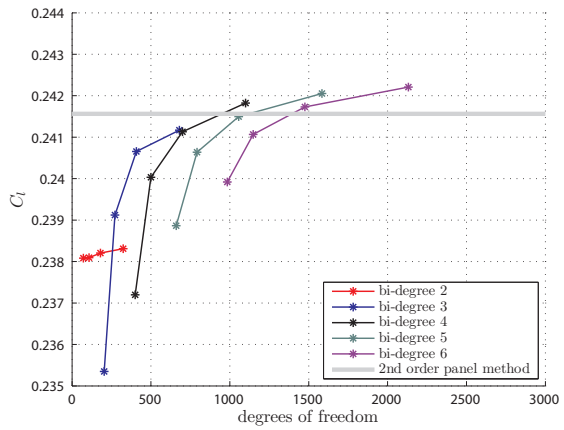


(a) streamlines of the flow

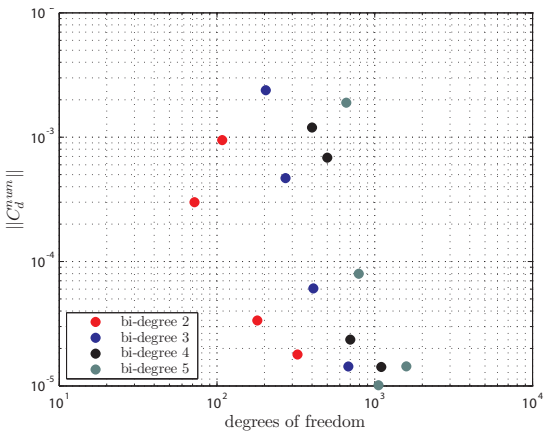


(b) iso-contours of pressure

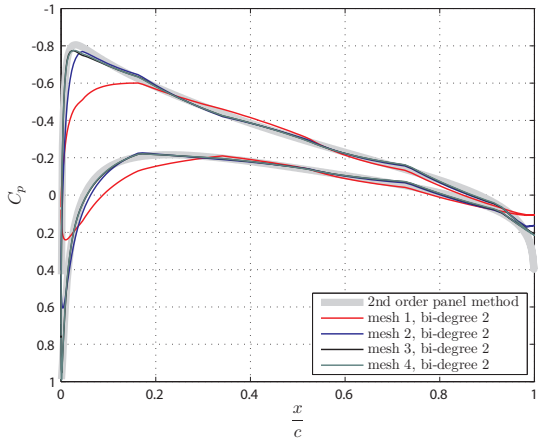
Figure 5.24: Potential flow around a lifting NACA0012 airfoil at 2 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6.



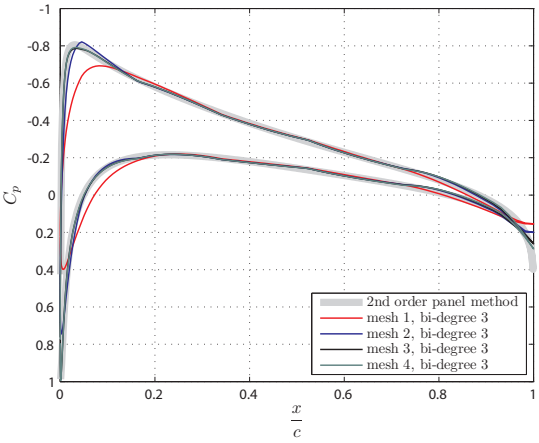
(a) lift coefficients



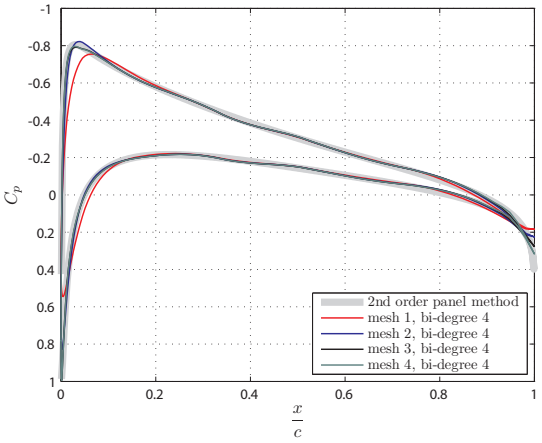
(b) drag coefficients



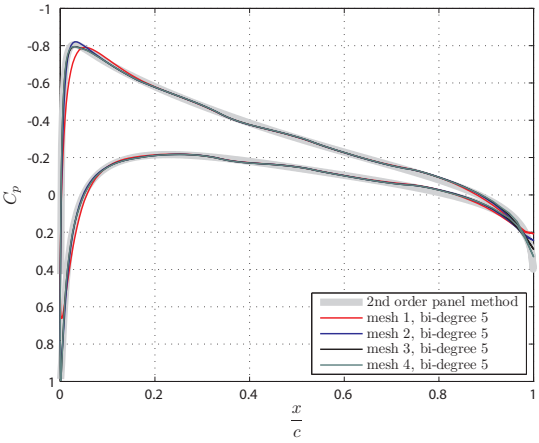
(c) C_p plotted verses $\frac{x}{c}$



(d) C_p plotted verses $\frac{x}{c}$



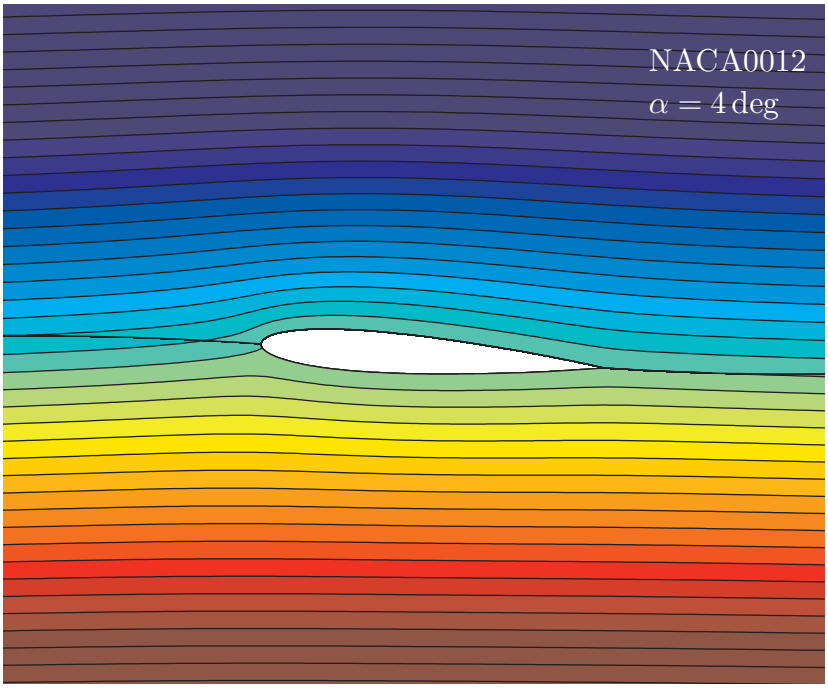
(e) C_p plotted verses $\frac{x}{c}$



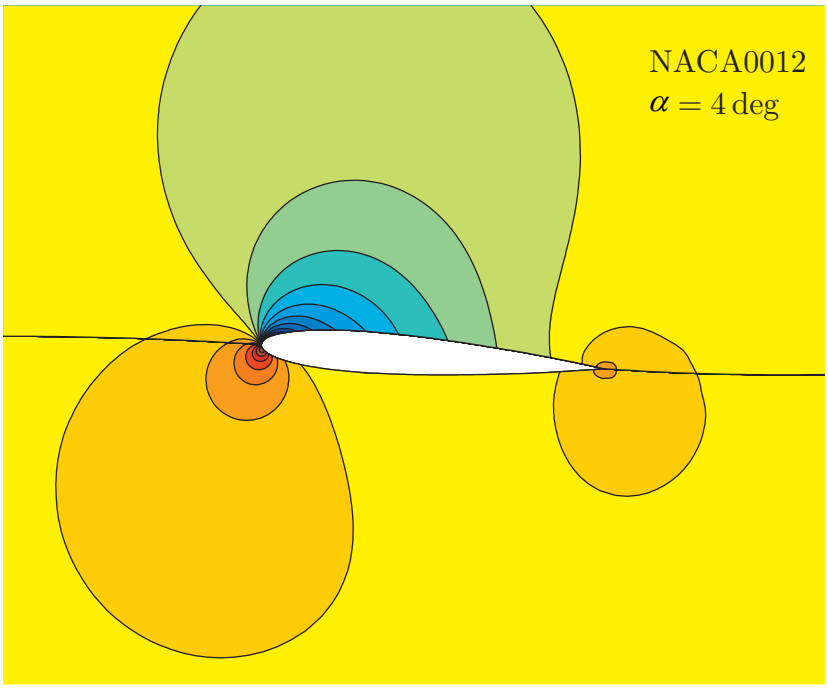
(f) C_p plotted verses $\frac{x}{c}$

Figure 5.25: Lifting flow over the NACA0012 with 2 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a small numerical drag component. Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels .

5.6.5 Results lifting flow over the NACA0012 with 4 deg of incidence

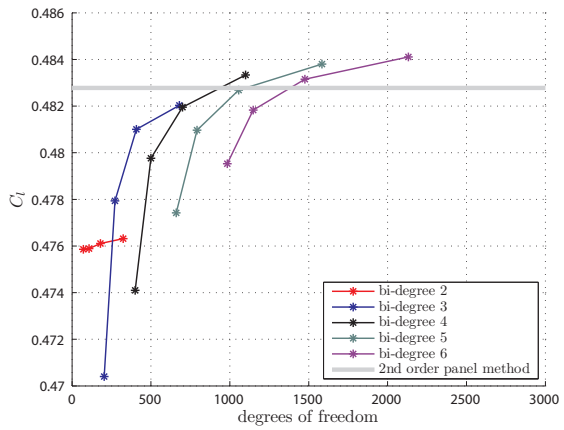


(a) streamlines of the flow

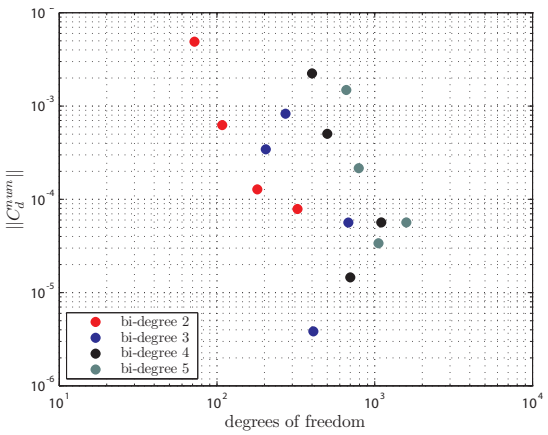


(b) iso-contours of pressure

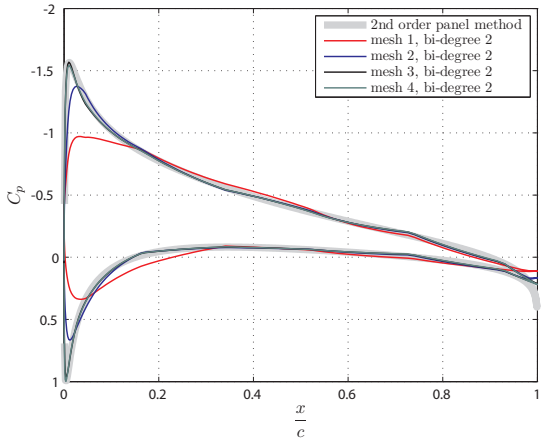
Figure 5.26: Potential flow around a lifting NACA0012 airfoil at 4 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6.



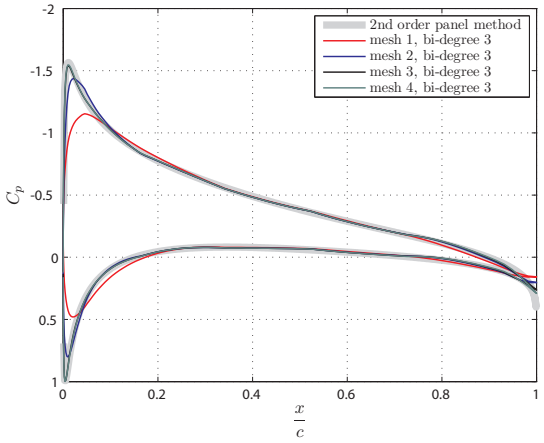
(a) lift coefficients



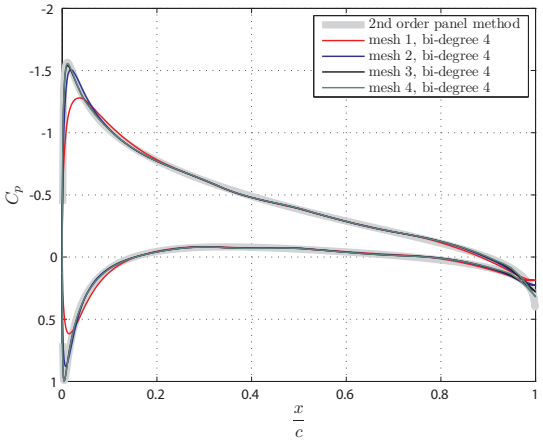
(b) drag coefficients



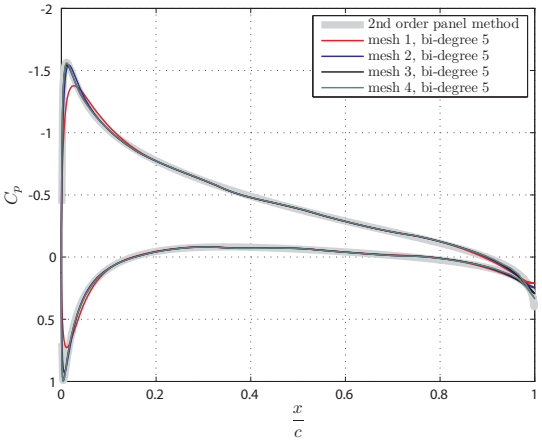
(c) C_p plotted verses $\frac{x}{c}$



(d) C_p plotted verses $\frac{x}{c}$



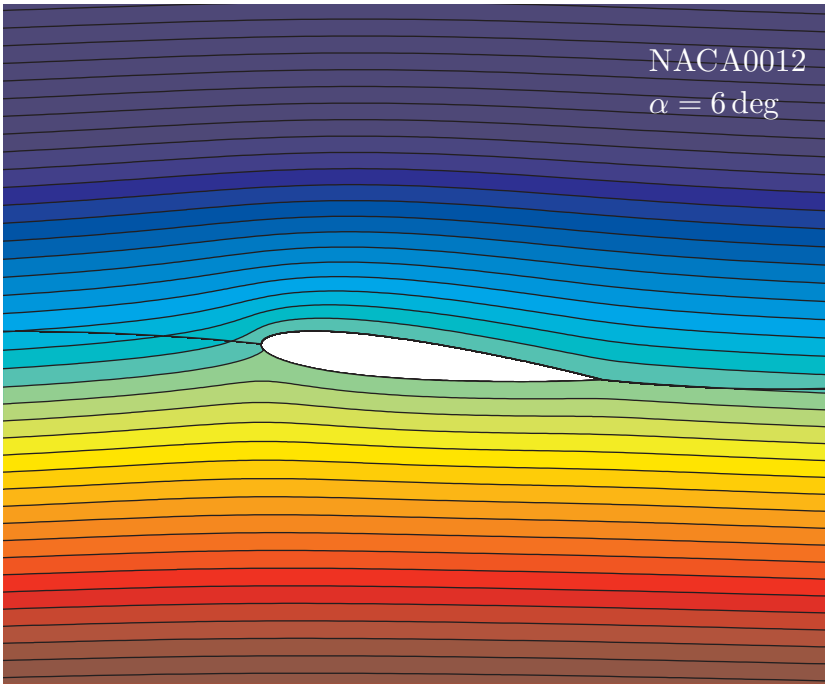
(e) C_p plotted verses $\frac{x}{c}$



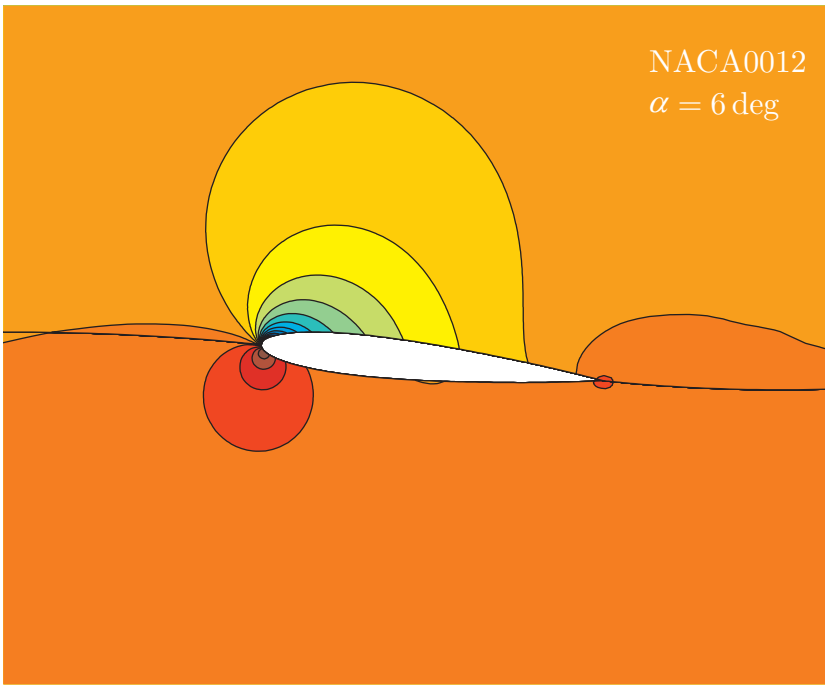
(f) C_p plotted verses $\frac{x}{c}$

Figure 5.27: Lifting flow over the NACA0012 with 4 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a small numerical drag component. Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels.

5.6.6 Results lifting flow over the NACA0012 with 6 deg of incidence

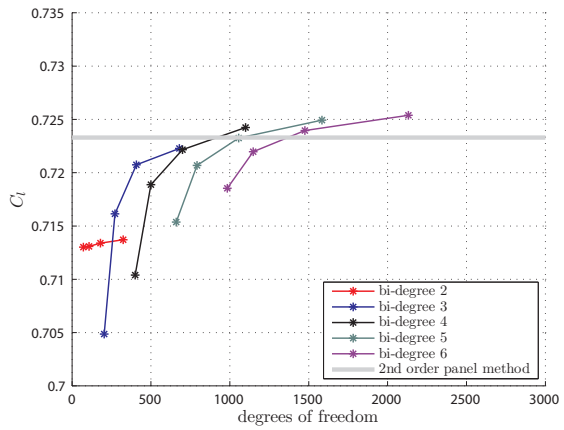


(a) streamlines of the flow

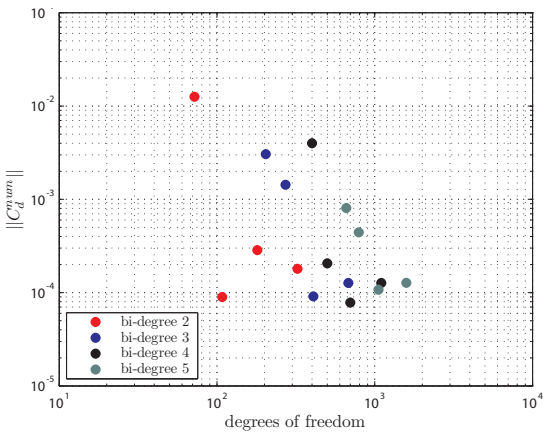


(b) iso-contours of pressure

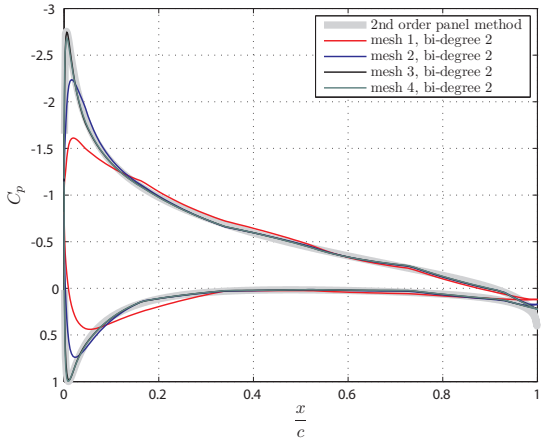
Figure 5.28: Potential flow around a lifting NACA0012 airfoil at 6 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6.



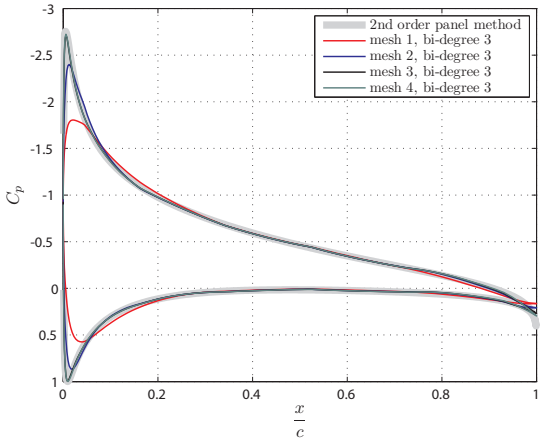
(a) lift coefficients



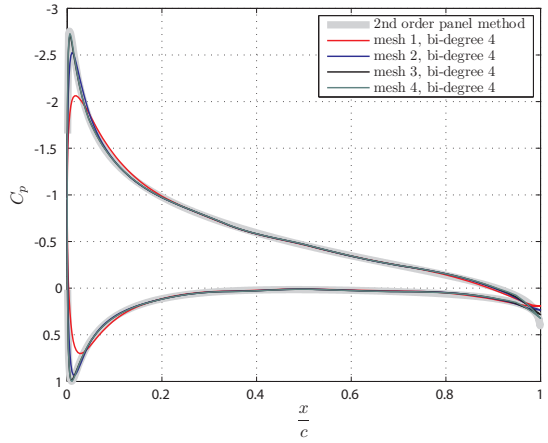
(b) drag coefficients



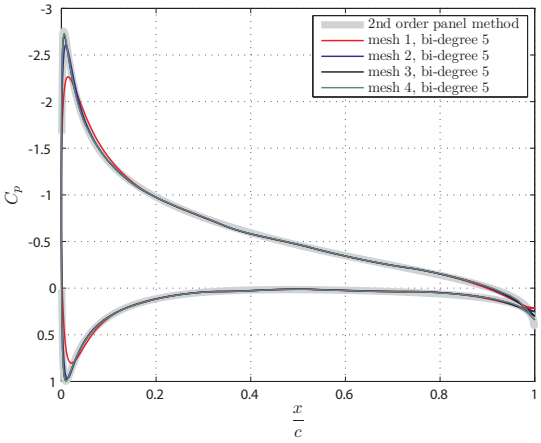
(c) C_p plotted verses $\frac{x}{c}$



(d) C_p plotted verses $\frac{x}{c}$



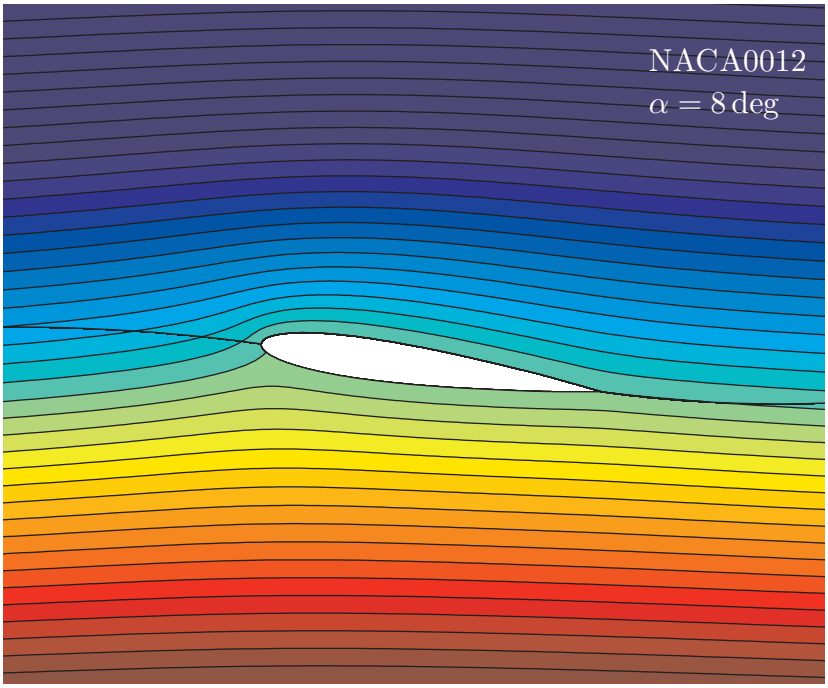
(e) C_p plotted verses $\frac{x}{c}$



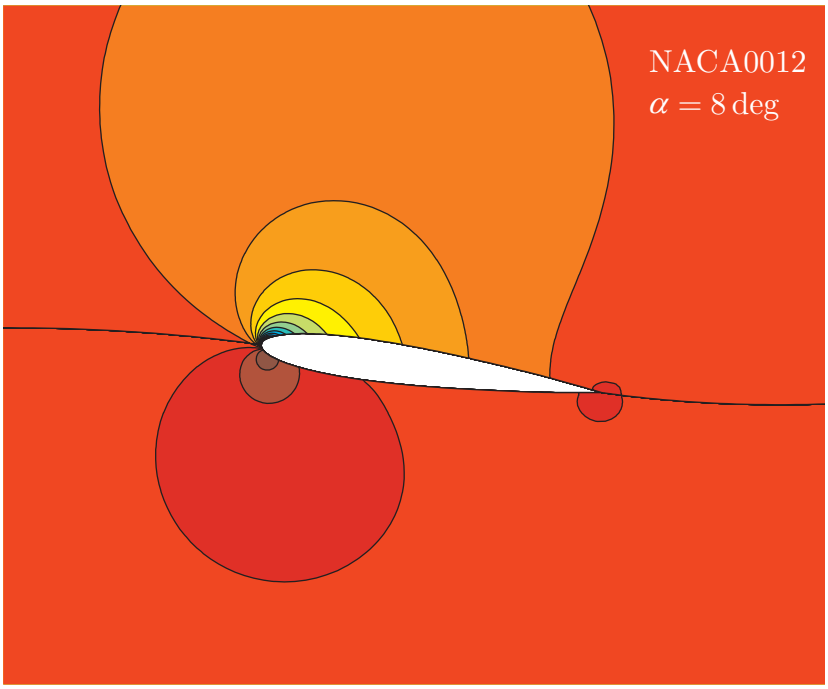
(f) C_p plotted verses $\frac{x}{c}$

Figure 5.29: Lifting flow over the NACA0012 with 6 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a minor numerical drag component. Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels.

5.6.7 Results lifting flow over the NACA0012 with $8\,\mathrm{deg}$ of incidence

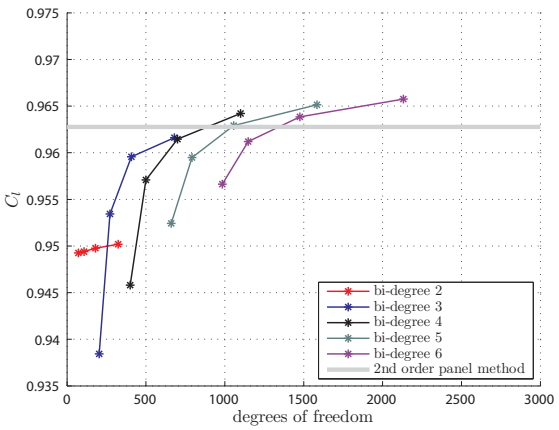


(a) streamlines of the flow

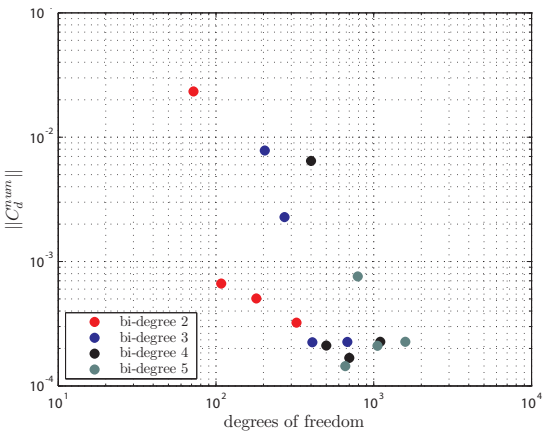


(b) iso-contours of pressure

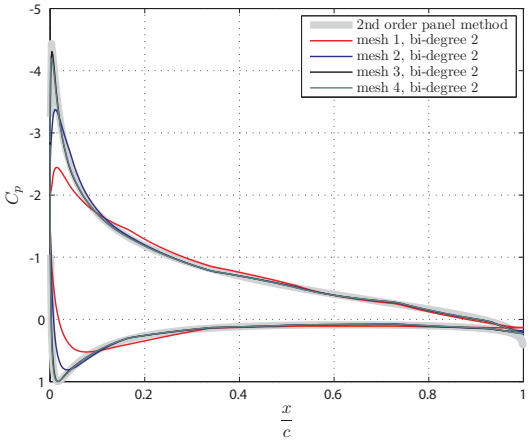
Figure 5.30: Potential flow around a lifting NACA0012 airfoil at 8 degrees of incidence to the flow. The results have been obtained for mesh 3, bi-degree 6.



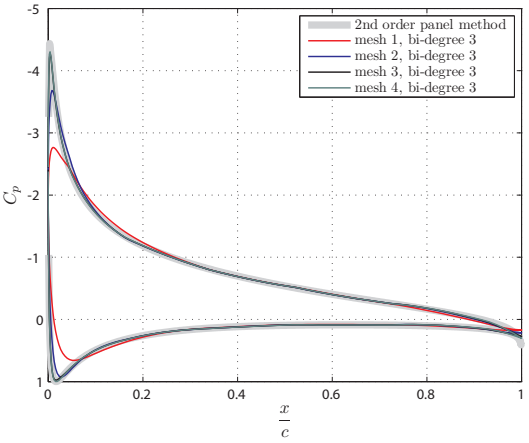
(a) lift coefficients



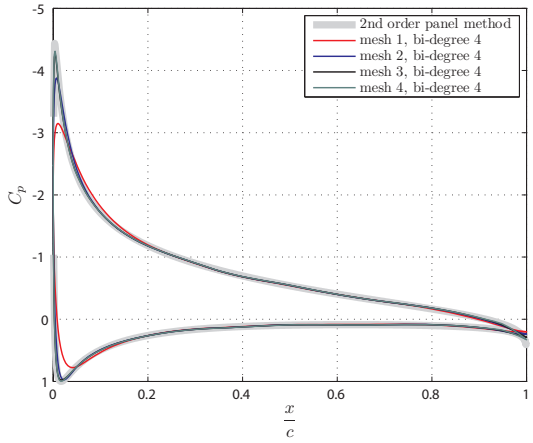
(b) drag coefficients



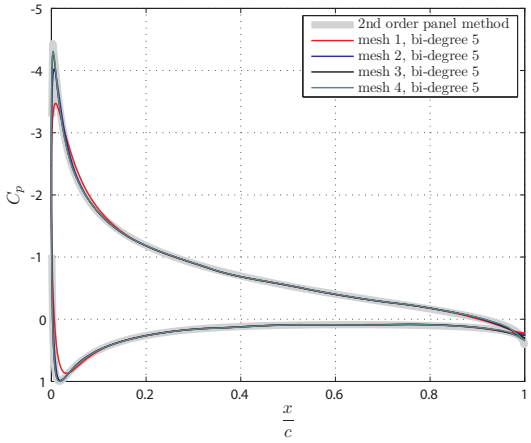
(c) C_p plotted verses $\frac{x}{c}$



(d) C_p plotted verses $\frac{x}{c}$



(e) C_p plotted verses $\frac{x}{c}$



(f) C_p plotted verses $\frac{x}{c}$

Figure 5.31: Lifting flow over the NACA0012 with 8 degrees of incidence to the flow. Figure (a) illustrates the convergence behavior of the lift for the different polynomial orders, as a function of the degrees of freedom. The lift is slightly overestimated, as compared to the panel method. Figure (b) shows a minor numerical drag component Figures (c) to (f) depict the pressure coefficient C_p over the airfoil as compared to a 2nd order panel method using 1000 panels.

5.6.8 Discussion

In this section we calculated the incompressible, irrotational flow around a lifting airfoil at several angles of attack. As a reference solution we used the results obtained by a 2nd order panel method employing 1000 panels. This can give us a qualitative idea how well our numerical scheme performs.

Reasonable agreement is found between our results and those obtained from the panel method. Studying Figures 5.25(a), 5.27(a), 5.29(a) and 5.31(a), we can observe a similar trend in the approximation of the lift, for all angles of attack. The lift we calculate is in all cases, slightly higher than the lift obtained using the panel method. These differences are in the order of 10^{-3}. Note that the approximation on the coarse mesh is already in the order 10^{-2}.

The effect of increased continuity is visible, although not as pronounced as for the flow around the cylinder. The convergence behavior of the quadratic degree B-spline approximation to the lift is quite different than those obtained using the higher degree and continuity approximations.

The numerical scheme is bothered by a small drag component, see Figures 5.23(a), 5.25(b), 5.27(b), 5.29(b) and 5.31(b). This minor drag component is on average in the order of 10^{-4}. Such values of numerical drag in potential flows are not uncommon in numerical calculations. The 2nd order panel method that we compare with obtained similar values of numerical drag. The source of the numerical drag component is not exactly clear. However, in the contour plots of the pressure field, Figures 5.22, 5.24, 5.26, 5.28 and 5.30 we can observe that the iso-contours of pressure at the trailing edge are not that smooth. The source of the small numerical drag component may originate from the trailing edge.

Reasonable agreement is found between the reference solution and our numerical approximation of the pressure coefficient C_p on the surface of the airfoil. In Figures 5.23(b-f), 5.25(c-f), 5.27(c-f), 5.29(c-f) and 5.31(c-f), we can observe that the IsoGeometric mimetic discretization approach converges to the reference solution. As for the flow around the cylinder, the high order and increased continuity seems to pay off. We do however observe some second order deviations compared to the reference solution. These may be the result of our approximation of the geometry.

Conclusion

The aim of this thesis was to combine the strengths of IsoGeometric Analysis with those of Mimetic Discretization Methods, to result in an extremely powerful new discretization approach. B-splines, from which all modern CAD technologies are derived, have made this combination possible. They have shown to be a natural basis within the mimetic framework, since they can readily be identified with points, lines surfaces and volumes and lead to vector spaces which are discretely conservative by construction.

We have provided a proof of concept illustrating many advantages of the proposed combination: exact description of the geometry; tight integration with CAD; arbitrary order approximations; increased continuity of primal variables and derivatives; local conservation of primal variables (strong) and secondary variables (weak); a globally defined topology which makes it possibility to use the hodge decomposition to make useful decompositions of field variables.

We made use of the last property to model the incompressible irrotational flow over lifting objects. Although these problems seem elementary, conventional methods, like the FEM and FVM are unable to solve these kind of problems without resorting to an engineering approach. We presented a novel procedure, which provides a discrete analogue to the harmonic function, which can directly be related to the lift. Although this procedure has only been applied to flows around a single object, it can readily be applied to lifting flows around more than one object. The results show that the lift can accurately be approximated, already on quite coarse meshes.

It was further observed that the increased continuity of B-splines can have a huge impact on the qualitative behavior of the numerical solution. C^0 continuity of pressure may allow the numerical solution to behave concave, where convex behavior is expected. It was observed that C^1 continuity is already sufficient to capture the qualitative behavior of the pressure.

Another important argument, why to apply Mimetic Methods, is that there is a strong indication that these methods naturally lead to stable consistent approximations. We did not investigate this, since we applied the IsoGeometric Mimetic discretization approach to elliptic

boundary value problems, for which stability is not so much an issue. Buffa et al. [17], however, already showed that these spaces of B-spline differential forms are inf-sup stable for more demanding problems like Stokes flow. This is a topic which should be further investigated in the near future.

Although the method is of arbitrary order, B-spline based IsoGeometric Analysis is in practice confined to low or medium order, since the conditioning of inner product mass matrices with order goes at an exponential rate. This bad conditioning doesn't directly seam to affect the accuracy. This can be explained by the variation diminishing property of B-splines. B-splines respond less to a change in the coefficients for increasing order.

While conditioning with polynomial order appears very bad, the conditioning with decreasing mesh size is more or less constant. This might be the leading factor in large complex problems which require many stages of refinement.

We further confirmed the results of several researchers [14, 48] that B-spline based IsoGeometric Analysis is very robust under mesh distortion. We performed some simple test cases of the scalar Poisson equation. In this case Galerkin based IGA and the proposed discretization give identical results and we observed that calculations on meshes containing self overlapping elements give good rates of convergence; at least after sufficient refinement.

Recommendations

This thesis provides a basis for 2D discretization in space. Differential geometry also provides operators like the Lie-derivative which governs convection in space and time. We need a discrete analogue to this operator. A space-time approach might be very interesting in the case of B-splines since they feature a certain level of continuity and continuity in time is very important.

In the proposed discretization we define an inner product and deduce the Hodge star operator. This approach is convenient because it is not necessary to set up a dual grid. For certain problems this Hodge star needs to be inverted, which is a very expensive operation and leads to full matrices. In Chapter 4 we also discussed a different approach where we explicitly define the Hodge star and induce an inner product. We discussed a probable dual grid which might lead to fortunate discretizations in case of B-splines. This should be further investigated.

I have a suspicion that it may be possible to determine the lift directly on the coarse mesh. The lift is only dependent on the in and out flow conditions and a Kutta condition. All of these are set on the boundary. If we are able to impose them exactly, by enforcing boundary conditions strongly, then all conditions are met. It might be that the lift can then directly be determined. This would be a very strong result.

Bibliography

[1] Fast degree elevation and knot insertion for b-spline curves. *Computer Aided Geometric Design*, 22(2), 2005.

[2] Efficient quadrature for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8), 2010.

[3] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41(3), 2008.

[4] I. Akkerman, Y. Bazilevs, C.E. Kees, and M.W. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230(11), 2011.

[5] J. Anderson Jr. *Fundamentals of Aerodynamics*. McGraw Hill, 2001.

[6] Aurore Back and Eric Sonnendrücker. Spline discrete differential forms. Application to Maxwell' s equations. Technical report.

[7] Y. Bazilevs and T.J.R. Hughes. Nurbs-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43(1), 2008.

[8] Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4-5), 2006.

[9] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4), 2007.

[10] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: Theory, algorithms, and computations. *Computational Mechanics*, 43(1), 2008.

[11] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using t-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):229–263, 2009.

[12] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3d simulation of wind turbine rotors at full scale. part i: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65 (1-3), 2011.

[13] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wchner, and K.-U. Bletzinger. 3d simulation of wind turbine rotors at full scale. part ii: Fluidstructure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65(1-3), 2011.

[14] D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The reissner-mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8), 2010.

[15] P. B. Bochev and J. M. Hyman. Applications of algebraic topology to compatible spatial discretizations. 2005.

[16] Pavel B. Bochev and James M. Hyman. Principles of mimetic discretizations of differential operators. 2005.

[17] A. Buffa, J. Rivas, G. Sangalli, and R. Vazquez. Isogeometric discrete differential forms in three dimensions. *SIAM Journal on Numerical Analysis*, 49(2).

[18] A. Buffa, D. Cho, and G. Sangalli. Linear independence of the t-spline blending functions associated with some particular t-meshes. *Computer Methods in Applied Mechanics and Engineering*, 2009.

[19] A. Buffa, C. de Falco, and G. Sangalli. Isogeometric analysis: new stable elements for the stokes equation. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 2010.

[20] A. Buffa, G. Sangalli, and R. Vzquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 199 (17-20), 2010.

[21] S. Cho and S.-H. Ha. Isogeometric shape design optimization: Exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38(1), 2009.

[22] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric Analysis*. John Wiley & Sons, Ltd, 2009.

[23] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43), 2006.

[24] M.G. Cox. The numerical evaluation of b-splines. Technical report, National Physics Laboratory DNAC 4, 1971.

[25] Wolfgang Dahmen, Charles A. Micchelli, and Hans-Peter Seidel. Blossoming begets b-spline bases built better by b-patches. *Mathematics of Computation*, 59(199), 1992.

[26] C. de Boor. *A practical guide to splines / Carl de Boor*. Springer-Verlag, New York :, 1978.

[27] C. de Boor. On calculation with b-splines. *Journal of Approximation Theory*, 6, 1972.

[28] C. de Boor and G. J. Fix. Spline approximation by quasiinterpolants. *Journal of Approximation Theory*, 8(1), 1973.

[29] T. Dokken and V. Skytt. Locally refined splines, 2010. URL http://www.sintef.no/upload/IKT/9011/geometri/Dokken_Pavia_version_July_2.pdf.

[30] M.R. Dorfel, B. Juttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with t-splines. *Computer Methods in Applied Mechanics and Engineering*, 199, 2010.

[31] Claes Eskilsson and Spencer J. Sherwin. Spectral/hp discontinuous galerkin methods for modelling 2d boussinesq equations. *J. Comput. Phys.*, 212(2), 2006.

[32] Gerald E. Farin. *NURBS: From Projective Geometry to Practical Use*. A. K. Peters, Ltd., 2nd edition, 1999.

[33] H. Flanders. *Differential forms with applications to the physical sciences*. Dover Pubns, 1989.

[34] T. Frankel. *The geometry of physics: an introduction*. Cambridge Univ Pr, 2004.

[35] Marc Gerritsma. Edge functions for spectral element methods. In *Spectral and High Order Methods for Partial Differential Equations*. 2007.

[36] Xianfeng Gu, Ying He, and Hong Qin. Manifold splines. In *Graphical Models*, 2005.

[37] Ying He, Kexiang Wang, Hongyu Wang, Xianfeng Gu, and Hong Qin. Manifold t-spline. In *In Proceedings of Geometric Modeling and Processing*, 2006.

[38] R. Hiptmair. Discrete hodge-operators: an algebraic perspective - abstract. *Journal of Electromagnetic Waves and Applications*, 15, 2001.

[39] Klaus Hollig. *Finite Element Methods with B-Splines*. Society for Industrial and Applied Mathematics, 2003.

[40] Klaus Hollig, Ulrich Reif, and Joachim Wipper. Multigrid methods with web-splines. *Numerische Mathematik*, 91, 2002.

[41] T.J.R. Hughes, V.M. Calo, and G. Scovazzi. Variational and multiscale methods in turbulence. *Mechanics of the 21st Century*, 2005.

[42] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. 194(39-41), 2005.

[43] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method nurbs. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50), 2008.

[44] C.E. Kees, I. Akkerman, M.W. Farthing, and Y. Bazilevs. A conservative level set method suitable for variable-order approximations and unstructured meshes. *Journal of Computational Physics*, 230(12), 2011.

[45] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wchner. Isogeometric shell analysis with kirchhoff-love elements. *Computer Methods in Applied Mechanics and Engineering*, 198 (49-52), 2009.

[46] R. Kraft. Hierarchical b-splines. 1994.

[47] R. Kraft. Adaptive and linearly independent multilevel b-splines. *Surface Fitting and Multiresolution Methods*, 1997.

[48] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, and T.J.R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8), 2010.

[49] T. Lyche and R. Winther. A stable recurrence relation for trigonometric b-splines. *Journal of Approximation Theory*, 25(3), 1979.

[50] Tom Lyche and Larry L. Schumaker. Local spline approximation methods. *Journal of Approximation Theory*, 15(4), 1975.

[51] Martin J. Marsden. An identity for spline functions with applications to variation-diminishing spline approximation. *Journal of Approximation Theory*, 3(1), 1970.

[52] Claudio Mattiussi. A Reference Discretization Strategy for the Numerical Solution of Physical Field Problems. *Advances in Imaging and Electron Physics*, 121, 2002.

[53] Marian Neamtu. *What is the natural generalization of univariate splines to higher dimensions?* 2001.

[54] Les Piegl and Wayne Tiller. *The NURBS book (2nd ed.)*. Springer-Verlag New York, Inc., 1997.

[55] L. Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical report, HP Labs, 1987.

[56] L. Ramshaw. Blossoms are polar forms. *Comput. Aided Geom. Des.*, 6, 1989.

[57] Ahmed Ratnani and Eric Sonnendrücker. Arbitrary High-Order Spline Finite Element Solver for the Time Domain Maxwell equations.

[58] I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. part a: On the problem of smoothing of graduation. a first class of analytic approximation formulae. *Quart. Applied Mathematics*, 4, 1946.

[59] I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. part b: On the problem of osculatory formulae. *Quart. Applied Mathematics*, 4:112–141, 1946.

[60] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. T-splines and t-nurccs. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, 2003.

[61] Thomas W. Sederberg, Jianmin Zheng, and Xiaowen Song. Knot intervals and multi-degree splines. *Comput. Aided Geom. Des.*, 20, 2003.

[62] Thomas W. Sederberg, G. Thomas Finnigan, Xin Li, Hongwei Lin, and Heather Ipson. Watertight trimmed nurbs. *ACM Trans. Graph.*, 27(3), 2008.

[63] Hans-Peter Seidel. An introduction to polar forms. *IEEE Comput. Graph. Appl.*, 13, 1993.

[64] Wanqiang Shen and Guozhao Wang. A basis of multi-degree splines. *Computer Aided Geometric Design*, 27(1), 2010.

[65] Enzo Tonti. On the formal structure of physical theories. *Monograph of the Italian National Research Council*, 1975.

[66] W.A. Wall, M.A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197, 2008.

Appendix A

Continuous field reconstruction

In this chapter we shall reconstruct continuous representations from global discrete quantities by the use of basis functions. In fact, we will show how to reconstruct exterior differential forms from chains and co-chains.

Reconstructing a 0-form in 1D

Consider the zero form $\varphi^{(0)}(x) = \varphi(x)$ in figure A.1, representing for example the potential.

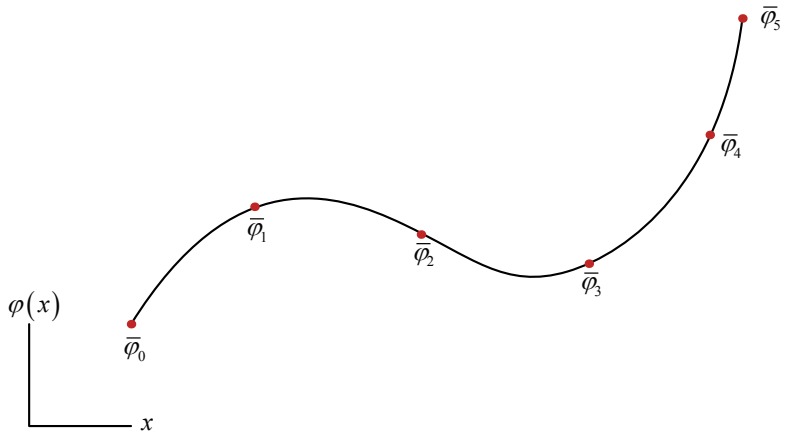


Figure A.1: 1D expansion of a 0-form in terms of nodal basis functions

A 0-form is associated to nodal point values. Consequently $\varphi^{(0)}(x)$ can be expanded in terms of nodal basis functions $h_i(x)$, where the degrees of freedom $\bar{\varphi}_i$ represent points values.

$$\varphi^{(0)}(x) = \sum_{i=0}^{5} \bar{\varphi}_i \, h_i(x)$$

The degrees of freedom can be collected in a 0-chain $\bar{\varphi} = [\bar{\varphi}_0, ..., \bar{\varphi}_5]$.

Reconstructing a 1-form in 1D

The fundamental theorem of calculus tells us how 1-forms and 0-forms are related in terms of discrete quantities

$$\bar{u}_i = \oint_C \mathrm{d}\, \varphi^{(0)} = \varphi_{i+1} - \varphi_i \qquad \text{equal to} \qquad \bar{u} = \delta\bar{\varphi} = [\bar{u}_1, ..., \bar{u}_4] \tag{A.1}$$

This is equal to taking the co-boundary from the 0-chain, depicted in figure A.5

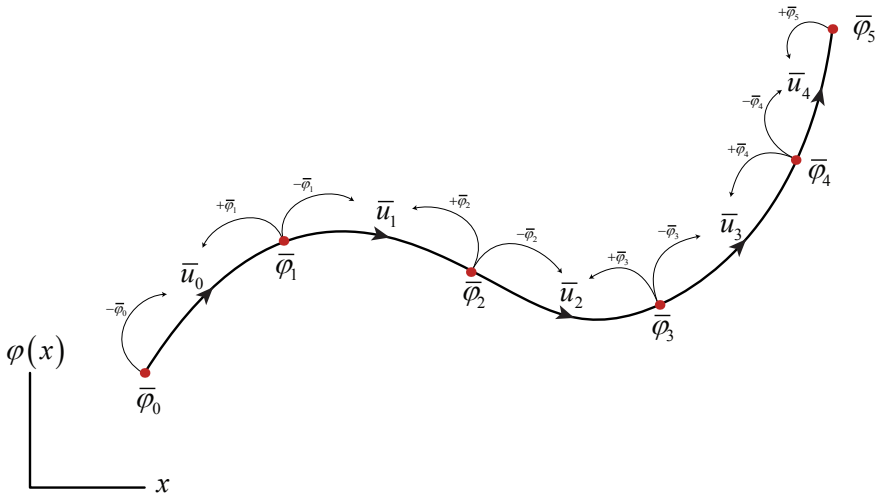


Figure A.2: Relation between

We are thus searching for a representation of a 1-form as a linear combination of the integral quantities $\bar{u}_i$ and new *edge* basis functions $e_i(x)$, defined along edges

$$u^{(1)}(x) = \sum_{i=0}^{4} \bar{u}_i \; e_i(x) \qquad \text{where} \qquad \bar{u}_i = \varphi_{i+1} - \varphi_i \tag{A.2}$$

In this setting we can represent the topological relations exactly, using the concepts of chain and co-chain introduced in the previous chapter, and subsequentally reconstruct continuous field representations of differential forms.

But what do these edge basis functions look like??

A.1 Edge functions

In this section we derive the edge basis functions, which represent the cornerstone of the mimetic approach that we have taken. These basis functions were first derived by Gerritsma [35], and the result holds for an arbitrary basis $h(x)$.

The general expression for the expansion of a 0-form in 1D looks like

$$\varphi^{(0)}(x) = \sum_{i=0}^{n} \bar{\varphi}_i \; h_i(x) \tag{A.3}$$

Taking the exterior derivative of $\varphi^{(0)}(x)$, we obtain the 1-form $u^{(1)}(x) = \mathrm{d}\,\varphi^{(0)}$. In terms of the basis in A.3 we have

$$u^1(x) = \mathrm{d} \sum_{i=0}^{n} \bar{\varphi}_i \; h_i(x) = \sum_{i=0}^{n} \bar{\varphi}_i \; \mathrm{d}h_i(x) \tag{A.4}$$

Partition of unity $\sum_{i=0}^{n} h_i(x) = 1$ implies that $\mathrm{d} \sum_{i=0}^{n} h_i(x) = 0$, so we can add $\bar{\varphi}_k \; \mathrm{d} \sum_{i=0}^{n} h_i(x) = \sum_{i=0}^{n} \bar{\varphi}_k \; \mathrm{d}h_i(x) = 0$ to A.4 to obtain

$$u^1(x) = \sum_{i=0}^{n} (\bar{\varphi}_i - \bar{\varphi}_k) \; \mathrm{d}h_i(x) \tag{A.5}$$

This looks a bit like the expression we are searching for. A.5 should hold for all choices of k, but we can make things easier by choosing $k = 0$ in the expansion

$$\begin{aligned} u^1(x) &= \sum_{i=0}^{n} (\bar{\varphi}_i - \bar{\varphi}_0) \; \mathrm{d}h_i(x) = \sum_{i=0}^{n} \left(\sum_{j=1}^{i} (\bar{\varphi}_j - \bar{\varphi}_{j-1}) \right) \; \mathrm{d}h_i(x) = \sum_{i=0}^{n} \sum_{j=1}^{i} \bar{u}_j \; \mathrm{d}h_i(x) \\ &= \sum_{j=1}^{1} \bar{u}_j \; \mathrm{d}h_1(x) + \sum_{j=1}^{2} \bar{u}_j \; \mathrm{d}h_2(x) + \ldots + \sum_{j=1}^{n} \bar{u}_j \; \mathrm{d}h_n(x) \\ &= \bar{u}_1 \; \sum_{i=1}^{n} \mathrm{d}h_i(x) + \bar{u}_2 \; \sum_{i=2}^{n} \mathrm{d}h_i(x) + \ldots + \bar{u}_n \; \sum_{i=n}^{n} \mathrm{d}h_i(x) \\ &= \sum_{j=1}^{n} \bar{u}_j \; \sum_{i=j}^{n} \mathrm{d}h_i(x) \end{aligned} \tag{A.6}$$

A 1-form can thus be expanded as

$$u^{(1)}(x) = \sum_{i=0}^{n-1} \bar{u}_i \; e_i(x) \tag{A.7}$$

where the degrees of freedom are the co-chains

$$\bar{u}_j = \bar{\varphi}_j - \bar{\varphi}_{j-1} \tag{A.8}$$

and the new edge basis functions are defined as

$$e_j(x) = \sum_{i=j}^{n} \mathrm{d}h_i(x) \tag{A.9}$$

This result holds for all basis functions $h_i(x)$

A.2 Application of the edge functions to the grad, curl and div

Applying tensor products of the nodal basis function $h(x)$ and edge function $e(x)$ we can reconstruct the different spaces of 0-forms, 1-forms, 2-forms and 3-forms in our complex.

The constructed spaces should be able to represent the exact sequence in both the continuous and in the discrete setting

$$\begin{array}{ccccccccc}
\mathbb{R} & \longrightarrow & \mathcal{C}^{(0)} & \xrightarrow{\mathbb{D}^{(1,0)}} & \mathcal{C}^{(1)} & \xrightarrow{\mathbb{D}^{(2,1)}} & \mathcal{C}^{(2)} & \longrightarrow & 0 \\
 & & \mathcal{I}_0 \downarrow \uparrow \mathcal{R}_0 & & \mathcal{I}_1 \downarrow \uparrow \mathcal{R}_1 & & \mathcal{I}_2 \downarrow \uparrow \mathcal{R}_2 & & \\
\mathbb{R} & \longrightarrow & H_P(\Omega) & \xrightarrow{\mathrm{d}} & H_L(\Omega) & \xrightarrow{\mathrm{d}} & H_S(\Omega) & \longrightarrow & 0
\end{array} \tag{A.10}$$

Here $\mathcal{R}$ denotes the reduction and $\mathcal{I}$ the reconstruction operation.

Reconstruction of $H_P \overset{\text{grad}}{\rightarrow} H_L$

Consider

$$u^{(1)} = \text{grad } \varphi^{(0)} \tag{A.11}$$

Let $\phi^{(0)}$ be expanded as a tensor product of basis functions in the coordinates (x_1, x_2, x_3)

$$\varphi(x_1, x_2, x_3) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \varphi_{i,j,k}\ h_i(x_1) h_j(x_2) h_k(x_3) \tag{A.12}$$

Then $u^{(1)}$ can be expanded as a tensor product of edge and nodal basis functions

$$\begin{aligned} u^1(x_1,x_2,x_3) &= \sum_{i=0}^{l-1}\sum_{j=0}^{m}\sum_{k=0}^{n} \bar{u}^1_{i,j,k}\; e_i(x_1)h_j(x_2)h_k(x_3)\;, \\ u^2(x_1,x_2,x_3) &= \sum_{i=0}^{l}\sum_{j=0}^{m-1}\sum_{k=0}^{n} \bar{u}^2_{i,j,k}\; h_i(x_1)e_j(x_2)h_k(x_3)\;, \\ u^3(x_1,x_2,x_3) &= \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n-1} \bar{u}^3_{i,j,k}\; h_i(x_1)h_j(x_2)e_k(x_3)\;. \end{aligned} \tag{A.13}$$

where $\bar{u}^i_{i,j,k}$ are the co-chains

$$\bar{u}^1_{i,j,k} = \bar{\phi}_{i+1,j,k} - \bar{\phi}_{i,j,k}\;, \quad \bar{u}^2_{i,j,k} = \bar{\phi}_{i,j+1,k} - \bar{\phi}_{i,j,k}\;, \quad \text{and} \quad \bar{u}^3_{i,j,k} = \bar{\phi}_{i,j,k+1} - \bar{\phi}_{i,j,k} \tag{A.14}$$

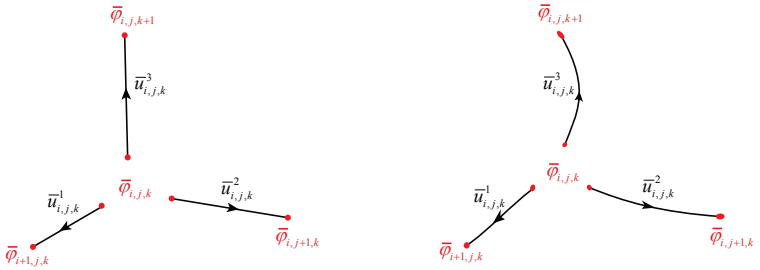


Figure A.3: A.14 is exact, coordinate free and invariant under C^1 transformations

Reconstruction of $H_L \overset{\text{curl}}{\rightarrow} H_S$

If $u^{(1)}$ is a 1-form, then

$$\omega^{(2)} = \text{curl}\; u^{(1)} \tag{A.15}$$

is given by the co-chains

$$\begin{aligned} \bar{\omega}^1_{i,j,k} &= \bar{u}^2_{i,j,k} - \bar{u}^2_{i,j,k+1} - \bar{u}^3_{i,j,k} + \bar{u}^3_{i,j+1,k}\;, \\ \bar{\omega}^2_{i,j,k} &= -\bar{u}^1_{i,j,k} + \bar{u}^1_{i,j,k+1} + \bar{u}^3_{i,j,k} - \bar{u}^3_{i+1,j,k}\;, \\ \bar{\omega}^3_{i,j,k} &= \bar{u}^1_{i,j,k} - \bar{u}^1_{i,j+1,k} - \bar{u}^2_{i,j,k} + \bar{u}^2_{i+1,j,k}\;, \end{aligned} \tag{A.16}$$

which reconstruct

$$\begin{aligned}
\omega^1(x_1,x_2,x_3) &= \sum_{i=0}^{l}\sum_{j=0}^{m-1}\sum_{k=0}^{n-1} \bar{\omega}^1_{i,j,k}\; h_i(x_1)e_j(x_2)e_k(x_3)\;,\\
\omega^2(x_1,x_2,x_3) &= \sum_{i=0}^{l-1}\sum_{j=0}^{m}\sum_{k=0}^{n-1} \bar{\omega}^2_{i,j,k}\; e_i(x_1)h_j(x_2)e_k(x^3)\;,\\
\omega^3(x_1,x_2,x_3) &= \sum_{i=0}^{l-1}\sum_{j=0}^{m-1}\sum_{k=0}^{n} \bar{\omega}^3_{i,j,k}\; e_i(x_1)e_j(x_2)h_k(x^3)\;.
\end{aligned} \tag{A.17}$$

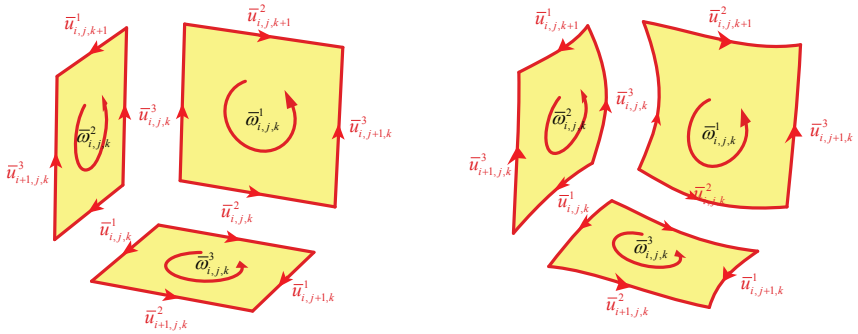


Figure A.4: A.16 is exact, coordinate free and invariant under C^1 transformations

Reconstruction of $H_S \overset{\text{div}}{\rightarrow} H_V$

$$\text{consider} \quad f = \text{div}\; q \tag{A.18}$$

The flux vector can be expanded as

$$\begin{aligned}
q^1(x_1,x_2,x_3) &= \sum_{i=0}^{l}\sum_{j=0}^{m-1}\sum_{k=0}^{n-1} \bar{q}^1_{i,j,k}\; h_i(x_1)e_j(x_2)e_k(x_3)\;,\\
q^2(x_1,x_2,x_3) &= \sum_{i=0}^{l-1}\sum_{j=0}^{m}\sum_{k=0}^{n-1} \bar{q}^2_{i,j,k}\; e_i(x_1)h_j(x_2)e_k(x_3)\;,\\
q^3(x_1,x_2,x_3) &= \sum_{i=0}^{l-1}\sum_{j=0}^{m-1}\sum_{k=0}^{n-1} \bar{q}^3_{i,j,k}\; e_i(x_1)e_j(x_2)h_k(x_3)\;.
\end{aligned} \tag{A.19}$$

If f is expanded in terms of volume basis functions

$$f(x_1, x_2, x_3) = \sum_{i=0}^{l-1} \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} \bar{f}_{i,j,k}\ e_i(x_1) e_j(x_2) e_k(x_3)\ , \tag{A.20}$$

the divergence equation reduces to

$$\bar{f}_{i,j,k} = \bar{q}^1_{i+1,j,k} - \bar{q}^1_{i,j,k} + \bar{q}^2_{i,j+1,k} - \bar{q}^2_{i,j,k} + \bar{q}^3_{i,j,k+1} - \bar{q}^3_{i,j,k} \tag{A.21}$$

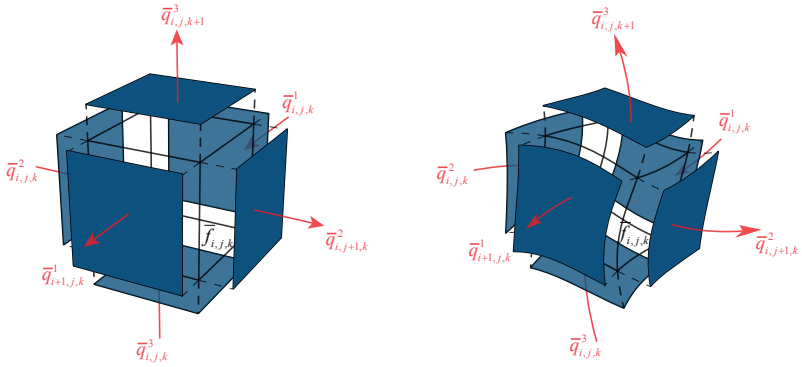


Figure A.5: A.21 is exact, coordinate free and invariant under C^1 transformations