

SAND REPORT

SAND2005-6272

Unlimited Release

Printed September 2005

Geophysical Subsurface Imaging and Interface Identification

David Day, Pavel Bochev, Kevin Pendley, Chester Weiss, Allen Robinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2005-6272
Unlimited Release
Printed September 2005

Geophysical Subsurface Imaging and Interface Identification

David Day, Pavel Bochev, Kevin Pendley
Computational Mathematics and Algorithms Department
Sandia National Laboratories
Albuquerque, NM 87185-1110
Chester Weiss
Geophysics Department
Albuquerque, NM 87185-0750
Allen Robinson
Computational Physics R&D Department
Albuquerque, NM 87185-0819

Abstract

Efforts to address research issues associated with computational tools for electromagnetic inversion problems are addressed. Advances in linear solvers, edge element discretizations, and sharp interface inversion tools are described. An algebraic multigrid linear solver for Maxwell's equations in the frequency domain, is available in the Trilinos package ML. The software package Ptenos was developed, that implements the De Rham complex of hexahedral finite element meshes, including edge elements for Maxwell's equations. In sharp interface inversion methods, the conductivity is approximated by a piecewise constant in bulk regions. A triangulation represents the two dimensional surface between regions. Methods for condensed curvilinear representations of the interface surface were studied, but certain robustness issues were not resolved. Inversion methods were implemented using existing software packages to manipulate the interface surface mesh. The sharp interface inversion methodology was evaluated on a three-dimensional direct current problem using P1 finite elements.

Acknowledgments

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Geophysical Subsurface Imaging and Interface Identification

Contents

1	Introduction and Summary	6
2	AMG for time harmonic Maxwell's equations	7
2.1	AMG for Maxwell's in the time domain	7
2.2	Frequency domain solvers	8
2.3	Software	10
3	Exact sequences of finite elements on hexahedral and quadrilateral lattices	10
3.1	De Rham complex	13
3.2	The van Welij complex: an exact finite element sequence on hexahedra	14
3.3	Example: solution of eddy current equations using face and edge elements. ...	41
4	Free-form surface design	43
4.1	Adaptive element quality	45
4.2	Local shape reconstruction	46
4.3	Quadratic triangles	49
5	Interface Identification	58
5.1	Creating tetrahedral meshes with TetGen	59
5.2	Combining surface meshes into tetrahedral meshes	59
5.3	Motion of a spherical interface	60
5.4	Refinement and coarsening of surface meshes.....	64
5.5	Implementation of deforming surfaces	64

1 Introduction and Summary

Electromagnetic induction is a classic geophysical exploration method designed for subsurface characterization - in particular, sensing the presence of geologic heterogeneities and fluids such as groundwater and hydrocarbons. Several approaches to the computational problems associated with predicting and interpreting electromagnetic phenomena in and around the earth are addressed herein. Publications resulting from the project include [31].

To obtain accurate and physically meaningful numerical simulations of natural phenomena, computational algorithms should operate in discrete settings that reflect the structure of governing mathematical models. In section 2, the extension of algebraic multigrid methods for the time domain eddy current equations to the frequency domain problem is discussed. Software was developed and is available in Trilinos ML package. In section 3 we consider finite element approximations of De Rham's complex. We describe how to develop a family of finite element spaces that forms an exact sequence on hexahedral grids. The ensuing family of non-affine finite elements is called a van Welij complex, after the work [37] of van Welij who first proposed a general method for developing tangentially and normally continuous vector fields on hexahedral elements. The use of this complex is illustrated for the eddy current equations and a conservation law problem. Software was developed and is available in the Ptenos finite element package.

The more popular methods of geophysical inversion seek solutions to an unconstrained optimization problem by imposing stabilizing constraints in the form of smoothing operators on some enormous set of model parameters (i.e. "over-parametrize and regularize"). In contrast we investigate an alternative approach whereby sharp jumps in material properties are preserved in the solution by choosing as model parameters a modest set of variables which describe an interface between adjacent regions in physical space. While still over-parametrized, this choice of model space contains far fewer parameters than before, thus easing the computational burden, in some cases, of the optimization problem. And most importantly, the associated finite element discretization is aligned with the abrupt changes in material properties associated with lithologic boundaries as well as the interface between buried cultural artifacts and the surrounding Earth.

In section 4, algorithms and tools are described that associate a smooth interface surface to a given triangulation. In particular, the tools support surface refinement and coarsening. Section 5 describes some preliminary results on the application of interface identification methods to some model problems in geophysical inversion. Due to time constraints, the results described here use the GNU Triangulated Surface Library for the manipulation of surface meshes and the TetGen software library for the generation of tetrahedral meshes.

2 AMG for time harmonic Maxwell's equations

An algebraic multigrid (AMG) linear solver for Maxwell's equations in the frequency domain, $\nabla \times \nabla \times + i\omega\mu\sigma$ is available in the Trilinos package ML. The capabilities of the solver have been discussed in [30]. in a presentation at the 2004 Copper Mountain conference on Iterative Methods titled "An AMG solver for the 3D time-harmonic Maxwell equations." Discretization of Maxwell's equations generates a sparse linear system with coefficient matrix \mathbf{A} with real and imaginary parts $\mathbf{A} = \mathbf{S} + i\mathbf{M}$.

We found that the AMG linear solvers developed for time domain simulations of physics governed by the eddy current equations readily extend to frequency domain simulations. ML does not support complex arithmetic. Equivalent real forms of the linear systems are solved, but these methods are not recommended for complex symmetric (and non-Hermitian) linear systems [12]. The emphasis herein is on the development of smoothers for ML in real arithmetic.

As a post script to the work described next, smoothers (e.g. complex symmetric QMR) may now be implemented using complex arithmetic by using the new Teuchos complex arithmetic package in Trilinos.

2.1 AMG for Maxwell's in the time domain

The algorithm for the frequency domain operator is based on AMG methods for the time domain Maxwell's operator $\nabla \times \nabla \times + \mu\sigma dt$. The AMG methods use Distributed Relaxation [20, 22, 4, 5] smoothers. Distributed Relaxation for the frequency domain operator will be derived from the method for the time domain operator. Other code is stored in `Trilinos/packages/ml/examples/Other`.

2.1.1 AMG

In order to apply AMG to \mathbf{S} , denote the coarsest level as $k_0 = 0$. The input is a set of operators on the finest level ($k > 0$). The input includes \mathbf{S}_k , the null space of \mathbf{S}_k , \mathbf{T}_k and the prolongation or interpolation operator, \mathbf{P}_k .

The construction of \mathbf{P}_k , the prolongation operator, addresses several issues. Actually there are two prolongation operators on each level, one that acts on nodal quantities, and another that acts on edge quantities. Furthermore the edge prolongator is "smoothed" $\mathbf{P} := (I - \rho\mathbf{D}^{-1}\mathbf{S})\mathbf{P}$, where \mathbf{D} is the diagonal of \mathbf{S} , and ρ is an upper bound for the spectral radius of $\mathbf{D}^{-1}\mathbf{S}$.

Multigrid is defined recursively. For example, $\mathbf{A}_{k-1} := \mathbf{P}_k^T \mathbf{A}_k \mathbf{P}_k$.

The basic subroutine $\text{MG}(\mathbf{b}_k, \mathbf{u}_k, k)$: takes as input the right-hand side \mathbf{b}_k on the k th level, and returns the approx solution \mathbf{u}_k also on the k th level.

$\text{MG}(\mathbf{b}_k, \mathbf{u}_k, k)$:

- $R_k(\mathbf{A}_k, \mathbf{b}_k, \mathbf{u}_k)$
- If $k \neq 0$
 1. $\mathbf{r}_k = \mathbf{b}_k - \mathbf{A}_k \mathbf{u}_k$
 2. $\mathbf{b}_{k-1} = \mathbf{P}_k^T \mathbf{r}_k$
 3. $\mathbf{u}_{k-1} = \mathbf{0}$
 4. MG($\mathbf{b}_{k-1}, \mathbf{u}_{k-1}, k-1$)
 5. $\mathbf{u}_k := \mathbf{u}_k + \mathbf{P}_k \mathbf{u}_{k-1}$

The operator R_k is called the smoother. On the most coarse level only, $k = 0$, R_0 is an exact solve. The smoother is defined next.

2.1.2 Distributed Relaxation smoothers

Distributed Relaxation smoothers (for time domain problems, [1]) incorporate the null space of \mathbf{S} (think gauge invariance).

1. $A_{proj} := T^T A T$
2. $Ax_e = f$ (e.g. polynomial smoother)
3. $A_{proj} x_n = T^T (f - Ax_e)$
4. $x_e := x_e + T x_n$
5. $Ax_e = f$ (again)

In steps two and three above, x_e and x_n are approximated by applying a few iterations of some iterative linear solver.

2.2 Frequency domain solvers

The software selectively uses equivalent real forms [12] to implement complex arithmetic.

The spectrum of \mathbf{A} is observed to be an L-shaped region. In regions where the mesh is coarse, $i\mathbf{M}$ dominates, and the corresponding high frequency eigenvalues are near to the imaginary axis. In regions where the mesh is fine, \mathbf{S} dominates, and the corresponding high frequency eigenvalues are near to the positive real axis. Moreover, as local mesh refinement proceeds, the legs of the L become increasingly thin (or equivalently long).

The standard equivalent real form of \mathbf{A} is

$$\mathbf{B} = \begin{bmatrix} \mathbf{S} & -\mathbf{M} \\ \mathbf{M} & \mathbf{S} \end{bmatrix}.$$

\mathbf{A} has the attractive property that its spectrum lies entirely in the closed first quadrant of the complex plane. The spectrum of \mathbf{B} is the union of the spectrum of \mathbf{A} with the spectrum of its complex conjugate. The standard equivalent real form is not recommended for matrices, such as \mathbf{A} , with spectrum in the first quadrant.

In ML, a complex polynomial preconditioner is applied to \mathbf{A} using complex arithmetic implicitly. The prolongators and null spaces are determined from

$$P \rightarrow \text{diag}(P, P) \text{ and } T \rightarrow \text{diag}(T, T).$$

In preliminary results for a problem on a cubic domain with a regular mesh of 8000 nodes, 24,000 edges, MG preconditioned GMRES reduces the residual norm by a factor of 10^{-10} in 23 iterations.

Problems with textbook polynomial smoothers arise in examples of highly conductive materials modeled on coarse grids, $\|\mathbf{M}\| > \|\mathbf{S}\|$. In such cases one complex GMRES step with diagonal preconditioning, produces satisfactory results. The residuals satisfy $\mathbf{r}_{n+1} = (I - \alpha_n \mathbf{D}^{-1} \mathbf{A}) \mathbf{r}_n$ where \mathbf{D} is the diagonal of \mathbf{A} , and α_n is the complex number minimizing $\|\mathbf{r}_{n+1}\|_2$. The difficulty in implementing this smoother lies in simulating complex arithmetic within ML.

2.2.1 Reproducing Kernels

An alternative is to use the Bergman kernel to define a polynomial preconditioner of degree p with respect to an inner product that corresponds to the linear system at hand.

First we describe how to associate an inner product with a linear system. It is easy (see [13]) to generate orthogonal polynomials based on a given inner product, e.g.

$$(2.1) \quad \langle f, g \rangle_\rho = \int_a^b f(x) \overline{g(x)} \rho(x) dx.$$

Given a simply connected subdomain \mathcal{D} of the complex plane with Lipschitz boundary, Bergman's kernel polynomials (see [19]) are based on

$$(2.2) \quad \langle f, g \rangle_{\mathcal{D}} = \int_{\mathcal{D}} f(z) \overline{g(z)} dx dy.$$

However, for our purposes, it may suffice to just integrate over two lines, $[a, b]$ on the real axis determined from \mathbf{S} , and $[ic, id]$ on the imaginary axis determined from $i\mathbf{M}$;

$$(2.3) \quad \langle f, g \rangle = \int_a^b f(x) \overline{g(x)} dx + \int_{ic}^{id} f(z) \overline{g(z)} |dz|,$$

Next we show how to use reproducing kernels to construct polynomial preconditioners following [15] §2.5. Let \mathcal{P}_n denote the set of polynomials of degree at most n .

The Krylov subspace $\mathcal{K}_n(\mathbf{A}; r_o)$ is isomorphic to \mathcal{P}_{n-1} . The isomorphism maps the expansion coefficients of the vector with respect to the basis $\{\mathbf{A}^j r_o\}_{j \geq 0}$ to the coefficients of a polynomial (as a sum of the monomials).

This is actually a very familiar idea. Given an inner product on vectors, one may use the isomorphism to implicitly define an inner product on polynomials. In this way, the iterative methods in general purpose software packages are instances of kernel polynomials. Conversely, given an inner product on polynomials, one may use the isomorphism and the associated kernel polynomials to construct special iterative methods.

In a polynomial based iterative method for solving $\mathbf{A}x = f$ from the initial guess x_o , the n th residual $r_n = f - \mathbf{A}x_n$ has the form $r_n = p_n(\mathbf{A})r_o$ for a degree n polynomial $p_n(\cdot)$ such that $p_n(0) = 1$. The minimum residual problem is $\|r_n\| = \min\{\|p(\mathbf{A})r_o\| : p \text{ in } \mathcal{P}_n \text{ and } p_n(0) = 1\}$. If $\{\psi_j\}_{j \geq 0}$ is a family of orthogonal polynomials, then the corresponding n th kernel polynomial is given by $K_n(t; \xi) = \sum \psi_j(t)\overline{\psi_j(\xi)}$. The family of polynomials that solve the minimal residual problem are $\{K_j(t; 0)/K_j(0; 0)\}_{j \geq 0}$ (see [15] Theorem 2.5.1).

2.3 Software

The algorithm is implemented in ML, Sandia's algebraic multilevel software package. The setup of the multigrid method leverages an existing setup method for real-valued systems from time-domain Maxwell problems. The block matrix \mathbf{B} is applied implicitly to a vector via wrappers in Aztec 2.1. (Aztec does not have complex-arithmetic capabilities). An existing Distributed Relaxation method for real systems was adapted specifically to operate on block systems like \mathbf{B} . The code currently has explicit dependencies on Sandia's finite difference software package, EM3D, which generates \mathbf{T} strictly for a Yee grid.

However, generalizing it (to edge elements) would be fairly straight-forward.

3 Exact sequences of finite elements on hexahedral and quadrilateral lattices

The operators gradient, curl and divergence are fundamental to the differential equations that model physical phenomena such as electromagnetic waves, diffusion, etc. These operators along with their domains¹ $H(\Omega, \mathbf{grad})$, $H(\Omega, \mathbf{curl})$ and $H(\Omega, \mathbf{div})$ form a mathematical structure called *De Rham* complex. A fundamental property of the De Rham complex is the exactness of the sequence

$$(3.1) \quad H(\Omega, \mathbf{grad}) \xrightarrow{\nabla} H(\Omega, \mathbf{curl}) \xrightarrow{\nabla \times} H(\Omega, \mathbf{div}) \xrightarrow{\nabla \cdot} L^2(\Omega).$$

¹We choose this notation favored by Bossavit [8], instead of the commonly used one $H^1(\Omega)$ in order to emphasize the connection between the spaces and the differential operators

Exactness means that each differential operator maps the space to its left into the kernel of the next differential operator. In Section 3.1 we discuss this property in more detail.

Maxwell's equations are one important example of a mathematical model that fits the structure of (3.1). When these equations are solved by finite element methods, it is advantageous to use a family of spaces that mimics the exactness property of De Rham's complex. This will result in approximate solutions that adhere better to the principles governing the physics of electromagnetic fields. Utility of such finite elements is by no means limited to the Maxwell's equations. The spaces $H(\Omega, \mathbf{grad})$, $H(\Omega, \mathbf{div})$ and $H(\Omega, \mathbf{curl})$ are important to other physical models and they also appear in various mixed methods; see [10]. This is, perhaps, why the traditional focus in finite element research has been on conforming approximations of the individual spaces rather than on the whole De Rham complex. The standard C^0 nodal finite elements are a classical example of $H(\Omega, \mathbf{grad})$ conforming spaces, while $H(\Omega, \mathbf{div})$ and $H(\Omega, \mathbf{curl})$ conforming finite elements have been proposed as early as 1977 by Raviart and Thomas [34]. However, it was Bossavit who first pointed out the importance of approximating the complete De Rham complex and recognized the principal role of differential forms in this process.

The Whitney complex; see [8] and [7], is the first example of a complete exact sequence of finite element spaces. Whitney elements are defined on simplicial triangulations and provide conforming approximation of $H(\Omega, \mathbf{grad})$, $H(\Omega, \mathbf{div})$, $H(\Omega, \mathbf{curl})$, and $L^2(\Omega)$. In three dimensions the Whitney complex contains the spaces \mathbf{W}^i , $i = 0, \dots, 3$. Each space \mathbf{W}^i has degrees of freedom defined on i -simplices. For $i = 0, 1, 2$ and 3 these are the nodes, edges, faces and tetrahedrons of the simplicial triangulation. Thus, \mathbf{W}^0 is the familiar scalar C^0 -finite element space, while \mathbf{W}^3 is a piecewise constant space on tetrahedrons. The functions in \mathbf{W}^1 are vector fields which are also called "edge elements" because their degrees of freedom are circulations along the edges. Likewise, functions in \mathbf{W}^2 are vector fields called "face elements" because their degrees of freedom are fluxes across a 2-simplex, that is, a face. The Whitney complex was originally conceived as an approximation tool for differential forms so the exactness property

$$(3.2) \quad \mathbf{W}^0 \xrightarrow{\nabla} \mathbf{W}^1 \xrightarrow{\nabla \times} \mathbf{W}^2 \xrightarrow{\nabla \cdot} \mathbf{W}^3.$$

was an intrinsic characteristic of the ensuing spaces. This property allowed Bossavit to justify their use in computational electromagnetism in his seminal paper [7]. Examples of finite element spaces that form exact sequences exist on other meshes as well. For bricks, prisms and rectangles finite elements that are part of a discrete De Rham complex were developed by Nedelec, Brezzi, Douglas, Fortin and Marini, among others; see [10], [14], [28] and [29].

All these spaces, including the Whitney complex, are examples of affine families of finite element spaces. Affine families are distinguished by the existence of an affine mapping between each element \mathcal{K} in the triangulation and a canonical *reference*² element $\hat{\mathcal{K}}$; see [9, p.72]

²In three dimensions the reference element can be either a tetrahedron (3-simplex) or a cube; in two dimen-

or [11]. As a result, such finite element spaces contain affine images of polynomials defined locally on the reference element. A general methodology for developing exact sequences of affine finite element spaces was suggested in [21] by Hiptmair. His method extends application of differential forms ideas beyond the original example of Whitney spaces.

Popularity of affine finite element spaces is largely owed to the analytical and computational convenience afforded by the simplicity of affine maps. However, because the class of transformations between $\hat{\mathcal{K}}$ and \mathcal{K} is restricted to affine maps, the type of computational elements is limited to tetrahedrons, triangles, parallelepipeds or parallelograms, and does not include quadrilaterals or hexahedrals; see [9, p.114]. Indeed, geometry of hexahedrals (or quadrilaterals) implies that the mapping between a reference element $\hat{\mathcal{K}}$ and an *isoparametric* element \mathcal{K} is not affine (unless \mathcal{K} belongs to the exceptional cases mentioned above). Consequently, the finite element space on \mathcal{K} ceases to be a polynomial space, even if the reference space on $\hat{\mathcal{K}}$ is such. Instead it contains functions that are (nonlinear) images of polynomials. This does not pose a serious problem when the set of degrees of freedom is limited to the nodal values. In fact, nodal isoparametric elements are routinely used along with their affine counterparts.

However, exact sequences of finite elements necessarily involve vector valued functions and degrees of freedom that depend on normal and tangential directions on the isoparametric element. In such cases development of the relevant vector bases becomes rather unintuitive, especially if one attempts to carry it on a reference element. Since the use of unstructured hexahedral/quadrilateral grids in engineering computations, including electromagnetism, is widespread, it is desirable to develop a standard procedure that leads to a discrete De Rham complex on hexahedral grids.

An intuitive method for developing edge and face elements on arbitrary hexahedra (isoparametric bricks) was first given by van Welij [37]. The van Welij elements are defined directly in the computational domain using the coordinate functions of the inverse mapping between a reference and computational elements. In this report we show that the edge and face elements proposed in [37] form a part of a discrete De Rham complex on hexahedra. We develop a systematic procedure for deriving each space from this complex based on representations of nodes, edges and faces in terms of local coordinate maps. While it is clear that this process can be interpreted and cast in the language of differential forms we choose instead to emphasize geometrical constructs that relate directly to the implementation of the finite elements. The first link in our complex is the standard isoparametric space $Q_1(\mathcal{K})$ which plays the role of \mathbf{W}^0 from the Whitney complex. The second and third links are analogues of the edge and face elements, while the last space can be identified with a single density function. For parallelepipeds or parallelograms we recover the well-known examples of [29], or [14]. However, for general hexahedrals the edge, face and density spaces are strikingly different - they are images of rational functions defined on the reference element. In other words, to define exact

sions triangles and squares are used.

sequences on hexahedrals requires more than polynomial reference spaces.

We also pay special attention to finding the proper restrictions of the finite element sequence to two space dimensions. While the resulting sequence cannot be exact in the same sense as its three dimensional counterpart, we identify two exactness properties related to the two curl operators in \mathbb{R}^2

3.1 De Rham complex

Differential equations models are mathematical expressions of physical phenomena. They operate in an abstract framework described by differential operators and function spaces representing their domains and ranges. Three fundamental operators are the gradient, curl and divergence. These operators and their domains form a mathematical structure called a De Rham complex. Our presentation will follow closely the treatment of Bossavit in [6]-[7]. To discuss further De Rham's complex let Ω denote a bounded region in \mathbb{R}^n with boundary $\partial\Omega$. We assume that $\partial\Omega$ has two disjoint pieces denoted by Γ and Γ^* . As usual, $L^2(\Omega)$ and $\mathbf{L}^2(\Omega)$ will denote the spaces of all square integrable scalar and vector functions on Ω . We introduce the spaces

$$(3.3) \quad H(\Omega, \mathbf{grad}) = \{\phi \in L^2(\Omega) | \nabla\phi \in \mathbf{L}^2(\Omega)\},$$

$$(3.4) \quad H(\Omega, \mathbf{curl}) = \{\mathbf{u} \in \mathbf{L}^2(\Omega) | \nabla \times \mathbf{u} \in \mathbf{L}^2(\Omega)\},$$

$$(3.5) \quad H(\Omega, \mathbf{div}) = \{\mathbf{u} \in \mathbf{L}^2(\Omega) | \nabla \cdot \mathbf{u} \in L^2(\Omega)\}.$$

Clearly, $H(\Omega, \mathbf{grad})$ is a closed subspace of $L^2(\Omega)$ and (3.4)-(3.5) are closed subspaces of $\mathbf{L}^2(\Omega)$. To use these spaces as domains for the gradient, curl and the divergence, they must be augmented with suitable boundary conditions. Consider first Γ and the subspaces of (3.3)-(3.5)

$$(3.6) \quad H_0(\Omega, \mathbf{grad}) = \{\phi \in H(\Omega, \mathbf{grad}) | \phi = 0 \text{ on } \Gamma\},$$

$$(3.7) \quad H_0(\Omega, \mathbf{curl}) = \{\mathbf{u} \in H(\Omega, \mathbf{curl}) | \mathbf{u} \times \mathbf{n} = 0 \text{ on } \Gamma\},$$

$$(3.8) \quad H_0(\Omega, \mathbf{div}) = \{\mathbf{u} \in H(\Omega, \mathbf{div}) | \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \Gamma\}.$$

The four spaces, $H_0(\Omega, \mathbf{grad})$, $H_0(\Omega, \mathbf{curl})$, $H_0(\Omega, \mathbf{div})$, $L^2(\Omega)$, and the three operators, ∇ , $\nabla \times$ and $\nabla \cdot$, form a *De Rham complex relative to Γ* .

The dual complex can be introduced by using the adjoint differential operators ∇^* , $(\nabla \times)^*$ and $(\nabla \cdot)^*$. The adjoint operators are defined in the usual manner by virtue of the integral identities

$$\begin{aligned} \int_{\Omega} \nabla\phi \cdot \mathbf{v} dx &= \int_{\Omega} \phi \nabla^* \mathbf{v} dx + \int_{\Gamma^*} uv \cdot \mathbf{n} d\Gamma; \\ \int_{\Omega} \nabla \times \mathbf{u} \cdot \mathbf{v} dx &= \int_{\Omega} \mathbf{u} \cdot (\nabla \times)^* \mathbf{v} dx + \int_{\Gamma^*} \mathbf{u} \cdot \mathbf{v} \times \mathbf{n} d\Gamma; \end{aligned}$$

and

$$\int_{\Omega} v \nabla \cdot \mathbf{u} dx = \int_{\Omega} \mathbf{u} (\nabla \cdot)^* \psi dx + \int_{\Gamma^*} \psi \mathbf{u} \cdot \mathbf{n} d\Gamma,$$

that is

$$\nabla^* = -\nabla \cdot, \quad (\nabla \times)^* = \nabla \times \quad \text{and} \quad (\nabla \cdot)^* = -\nabla$$

with domains

$$(3.9) \quad H_0^*(\Omega, \text{div}) = \{\mathbf{v} \in H(\Omega, \text{div}) \mid \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma^*\},$$

$$(3.10) \quad H_0^*(\Omega, \text{curl}) = \{\mathbf{v} \in H(\Omega, \text{curl}) \mid \mathbf{v} \times \mathbf{n} = 0 \text{ on } \Gamma^*\},$$

$$(3.11) \quad H_0^*(\Omega, \text{grad}) = \{\psi \in H(\Omega, \text{grad}) \mid \psi = 0 \text{ on } \Gamma^*\}.$$

The spaces (3.9)-(3.11), and $L^2(\Omega)$, and the adjoint operators ∇^* , $(\nabla \times)^*$, $(\nabla \cdot)^*$ form the dual De Rham complex. Note that these four spaces and three operators define a complex relative to Γ^* , whose dual is the original De Rham complex relative to Γ .

3.2 The van Welij complex: an exact finite element sequence on hexahedra

To avoid confusion with the well-established notation for the Whitney complex we use \mathcal{W}^i to denote the members of the van Welij complex. Of course, the index i now does not refer to an i -simplex, but to the position of the finite element space in the exact sequence. Thus, \mathcal{W}^0 is the usual C^0 isoparametric finite element space on hexahedra; \mathcal{W}^1 is the edge element space; \mathcal{W}^2 is the face element space, and \mathcal{W}^3 is the last member of the discrete complex. As in the case of Whitney elements, the degrees of freedom for \mathcal{W}^0 , \mathcal{W}^1 and \mathcal{W}^2 are scalar nodal values, circulations and fluxes of vector fields. Likewise, \mathcal{W}^3 is associated with a single degree of freedom per element.

The parallels between the Whitney and the van Welij complexes do not end here. The finite element bases that form the Whitney complex allow simple and elegant description in terms of the *barycentric* coordinates λ_i associated with each n -simplex of the triangulation; see [7] and [8]. The particular forms of these basis functions involve combinations of λ_i and their gradients, and correlate with expressions that describe nodes, lengths of edges, areas of faces and volumes of simplices. While hexahedra in \mathbb{R}^3 do not possess barycentric coordinates, we show that the bases of van Welij elements admit virtually the same descriptions, but relative to trilinear local basis functions defined on each hexahedral element. Combinations of these local functions and their gradients describe the nodes, lengths of edges, areas of faces and volumes of hexahedra in the same manner as barycentrics do for n -simplices. These similarities are hardly coincidental - conceptually, both families of spaces are rooted in approximations of differential forms. For more details about this connections we refer to [21].

We begin with developing the finite element functions for one element. Then we show how to combine functions defined on individual elements into finite element spaces on hexahedral or quadrilateral triangulation of the computational domain. For the first and the last members of the complex \mathcal{W}^i this process is standard. This is not so for \mathcal{W}^1 and \mathcal{W}^2 , where we have to deal with vector fields defined on elements sharing an edge or a face. To combine these fields into a single piecewise polynomial field with the desired tangential or normal continuity, it is necessary to introduce edge and face orientation for the triangulation.

3.2.1 The van Welij complex on a hexahedral

This section describes in detail how the van Welij complex may be developed in two and three space dimensions for one element and shows the exactness property of the resulting family of finite element spaces. Three-dimensions are the natural setting for edge and face elements and is considered first. The two-dimensional complex is then derived by embedding a quadrilateral element into a virtual hexahedral and consistent restriction of the ensuing 3D complex to the plane.

3.2.2 Parametrizations of hexahedra

This section introduces and studies mappings between convex hexahedra in \mathbf{R}^3 . Such mappings and their properties are fundamental to the development and understanding of finite element analogues of exact sequences of spaces. We recall for further reference the space $Q_k(\boldsymbol{\xi})$ of all polynomial functions whose degree in each coordinate direction does not exceed k .

To this end, we endow \mathbf{R}^3 with two distinct reference frames: a *physical* space with coordinates $(x_1, x_2, x_3) \equiv \mathbf{x}$, and a *reference* or *parameter* space with coordinates $(\xi_1, \xi_2, \xi_3) \equiv \boldsymbol{\xi}$. In the parameter space we consider the open cube $\hat{\mathcal{K}} = (-1, 1)^d$, $d = 3$ with vertices $\boldsymbol{\xi}^{\alpha\beta\gamma} = (\alpha, \beta, \gamma)$, $\alpha, \beta, \gamma = \pm 1$. In the physical space we consider a convex, open hexahedron \mathcal{K} with vertices $\mathbf{x}^{\alpha\beta\gamma} = (x_1^\alpha, x_2^\beta, x_3^\gamma)$; see Fig. 3.1. The hexahedron \mathcal{K} will serve as a prototype finite element, that is, the basic building block for the finite element spaces. The cube $\hat{\mathcal{K}}$ will be its *reference* element.

In addition to one-to-one mappings between \mathcal{K} and $\hat{\mathcal{K}}$, we will also consider mappings between the surfaces and the edges of the two elements. Such *face* and *edge* parametrizations are useful when one wants to compute surface and line integrals on \mathcal{K} . The need to compute such integrals rarely occurs in nodal finite elements where degrees of freedom are located at the vertices. However, here we are concerned with finite element spaces where degrees of freedom are fluxes and circulations of vector fields which naturally leads to evaluation of surface and line integrals.

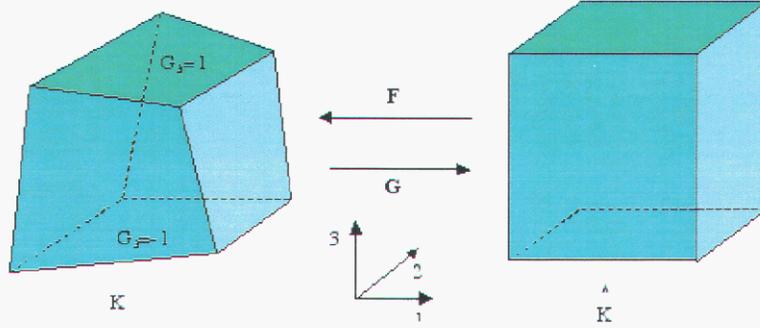


Figure 3.1. A hexahedron and its reference element

Directional and nodal bases on $\hat{\mathcal{K}}$. Central to all further developments will be the set of directional basis functions defined on $\hat{\mathcal{K}}$ by

$$(3.12) \quad \hat{\phi}_i^\alpha(\boldsymbol{\xi}) = \frac{1}{2} \left(1 + \alpha \xi_i \right); \quad i = 1, \dots, 3; \quad \alpha = \pm 1.$$

Since the upper index of these functions is either 1 or -1 , for simplicity we write ϕ_i^+ or ϕ_i^- whenever its value is fixed. Each $\hat{\phi}_i^\alpha$ is linear polynomial in ξ_i and is constant on the plane $\xi_i = \text{const}$. This plane is perpendicular to the i -th coordinate direction in the reference space. In particular, $\hat{\phi}_i^\alpha = 1$ on $\xi_i = \text{sign}(\alpha)$ and $\hat{\phi}_i^\alpha = 0$ on $\xi_i = -\text{sign}(\alpha)$. Therefore, directional basis functions do not form a nodal basis. However, their products can be used to describe the nodes $\hat{\mathcal{N}}$, edges $\hat{\mathcal{E}}$, faces $\hat{\mathcal{F}}$ and the cube $\hat{\mathcal{K}}$ itself as follows³:

$$(3.13) \quad \hat{\mathcal{N}} = \{ \boldsymbol{\xi} \in \hat{\mathcal{K}} \mid \hat{\phi}_i^\alpha \hat{\phi}_j^\beta \hat{\phi}_k^\gamma = 1; \alpha, \beta, \gamma = \pm 1; i, j = 1, 2, 3; i < j < k. \}$$

$$(3.14) \quad \hat{\mathcal{E}} = \{ \boldsymbol{\xi} \in \hat{\mathcal{K}} \mid \hat{\phi}_i^\alpha \hat{\phi}_j^\beta = 1; \alpha, \beta = \pm 1; i, j = 1, 2, 3; i < j \}$$

$$(3.15) \quad \hat{\mathcal{F}} = \{ \boldsymbol{\xi} \in \hat{\mathcal{K}} \mid \hat{\phi}_i^\alpha = 1; \alpha = \pm 1; i = 1, 2, 3 \}$$

$$(3.16) \quad \hat{\mathcal{K}} = \{ \boldsymbol{\xi} \in \mathbb{R}^3 \mid 0 < \hat{\phi}_i^\alpha \hat{\phi}_j^\beta \hat{\phi}_k^\gamma < 1 \}$$

Individual edges and faces will be denoted by $\hat{\mathcal{E}}_{ij}^{\alpha\beta}$ and $\hat{\mathcal{F}}_i^\alpha$, respectively. We will not change our notation for the nodes to $\boldsymbol{\xi}_{ijk}^{\alpha\beta\gamma}$ because there's only one permutation of 1,2 and 3 in (3.13). Intersection of two faces determines an edge according to

$$(3.17) \quad \hat{\mathcal{E}}_{ij}^{\alpha\beta} = \hat{\mathcal{F}}_i^\alpha \cap \hat{\mathcal{F}}_j^\beta.$$

and intersection of three faces determines a node:

$$(3.18) \quad \boldsymbol{\xi}^{\alpha\beta\gamma} = \hat{\mathcal{F}}_i^\alpha \cap \hat{\mathcal{F}}_j^\beta \cap \hat{\mathcal{F}}_k^\gamma.$$

³The barycentrics λ_i have similar property: the set $\lambda_i = 0$ describes a face of a tetrahedron, $\lambda_i \lambda_j = 0$ gives an edge and $\lambda_i \lambda_j \lambda_k = 0$ is a vertex. The tetrahedron itself is described by $0 < \lambda_i \lambda_j \lambda_k < 1$.

Taking intersections of two faces at a time in (3.18) gives the three edges

$$\hat{\mathcal{E}}_{ij}^{\alpha\beta} = \hat{\mathcal{F}}_i^\alpha \cap \hat{\mathcal{F}}_j^\beta; \quad \hat{\mathcal{E}}_{jk}^{\beta\gamma} = \hat{\mathcal{F}}_j^\beta \cap \hat{\mathcal{F}}_k^\gamma; \quad \text{and} \quad \hat{\mathcal{E}}_{ki}^{\gamma\alpha} = \hat{\mathcal{F}}_k^\gamma \cap \hat{\mathcal{F}}_i^\alpha,$$

that share $\xi^{\alpha\beta\gamma}$. Using the directional reference basis (3.12) we define eight other functions as follows

$$(3.19) \quad \hat{N}^{\alpha\beta\gamma}(\xi) = \hat{\phi}_1^\alpha(\xi) \hat{\phi}_2^\beta(\xi) \hat{\phi}_3^\gamma(\xi).$$

These functions belong to the space $Q_1(\xi)$ of all trilinear polynomials. If $\xi^{\kappa\mu\nu}$ is one of the vertices of $\hat{\mathcal{K}}$ it is easy to see that

$$(3.20) \quad \hat{N}^{\alpha\beta\gamma}(\xi^{\kappa\mu\nu}) = \delta_{\alpha\kappa} \delta_{\beta\mu} \delta_{\gamma\nu},$$

i.e., $\hat{N}^{\alpha\beta\gamma}$ vanishes on all but one of the vertices of the reference cube. Thus, (3.19) form a local *nodal* basis on $\hat{\mathcal{K}}$.

Parametrizations of \mathcal{K} . Parametrization of \mathcal{K} is a one-to-one mapping $F : \hat{\mathcal{K}} \mapsto \mathcal{K}$. Such a mapping can be constructed in the following manner: Given a hexahedral \mathcal{K} with vertices $\mathbf{x}^{\alpha\beta\gamma}$ we proceed to define the function

$$(3.21) \quad F_{\mathcal{K}}(\xi) = \sum_{\alpha,\beta,\gamma} \mathbf{x}^{\alpha\beta\gamma} \hat{N}^{\alpha\beta\gamma}(\xi).$$

This function maps the vertex $\xi^{\alpha\beta\gamma}$ of $\hat{\mathcal{K}}$ into the vertex $\mathbf{x}^{\alpha\beta\gamma}$ of \mathcal{K} . Moreover, it can be shown that $F_{\mathcal{K}}$ is the unique invertible mapping $\hat{\mathcal{K}} \mapsto \mathcal{K}$ with an inverse $G_{\mathcal{K}}$, such that

$$(3.22) \quad F_{\mathcal{K}}(\xi^{\alpha\beta\gamma}) = \mathbf{x}^{\alpha\beta\gamma} \quad \text{and} \quad G_{\mathcal{K}}(\mathbf{x}^{\alpha\beta\gamma}) = \xi^{\alpha\beta\gamma}.$$

As a result, $F_{\mathcal{K}}$ is the desired parametrization of \mathcal{K} .

In what follows, the subscript \mathcal{K} will be dropped and we will simply write F and G . The coordinate functions of these mappings will be denoted by G_j and F_i , respectively. From (3.21) it is not hard to see that F belongs to $Q_1(\xi) \times Q_1(\xi) \times Q_1(\xi)$ so that each coordinate function F_i is a trilinear polynomial. The explicit form of these polynomials is given by

$$(3.23) \quad \begin{aligned} F_i(\xi) &= \sum_{u,s,t=0,1} C_i^{ust} \xi_1^u \xi_2^s \xi_3^t \\ &= C_i^{111} \xi_1 \xi_2 \xi_3 + C_i^{110} \xi_1 \xi_2 + C_i^{101} \xi_1 \xi_3 + C_i^{011} \xi_2 \xi_3 + \\ &\quad C_i^{100} \xi_1 + C_i^{010} \xi_2 + C_i^{001} \xi_3 + C_i^{000}. \end{aligned}$$

The coefficients of each F_i can be expressed in terms of the vertex coordinates $\mathbf{x}^{\alpha\beta\gamma}$ of \mathcal{K} . Collecting like terms in (3.21) we find that

$$C_i^{ust} = \sum_{\alpha,\beta,\gamma=\pm 1} \alpha^u \beta^s \gamma^t x_i^{\alpha\beta\gamma}.$$

From (3.23) it is clear that unless \mathcal{K} is a parallelogram, F is not an affine mapping and G is not a polynomial function.

Directional and nodal bases on \mathcal{K} . A composition of G with the directional basis (3.12) gives rise to directional basis functions

$$(3.24) \quad \phi_i^\alpha(\mathbf{x}) = (\hat{\phi}_i^\alpha \circ G)(\mathbf{x}) = \frac{1}{2} \left(1 + \alpha G_i(\mathbf{x}) \right); \quad i = 1, \dots, 3$$

defined on \mathcal{K} . We also note for further use the identity

$$\nabla \phi_i^\alpha = \frac{\alpha}{2} \nabla G_i.$$

The directional basis (3.24) gives a systematic description of the nodes \mathcal{N} , edges \mathcal{E} , and faces \mathcal{F} of \mathcal{K} , and \mathcal{K} itself:

$$(3.25) \quad \mathcal{N} = \{ \mathbf{x} \in \mathcal{K} \mid \phi_i^\alpha \phi_j^\beta \phi_k^\gamma = 1; \alpha, \beta, \gamma = \pm 1; i, j = 1, 2, 3; i < j < k. \}$$

$$(3.26) \quad \mathcal{E} = \{ \mathbf{x} \in \mathcal{K} \mid \phi_i^\alpha \phi_j^\beta = 1; \alpha, \beta = \pm 1; i, j = 1, 2, 3; i < j \}$$

$$(3.27) \quad \mathcal{F} = \{ \mathbf{x} \in \mathcal{K} \mid \phi_i^\alpha = 1; \alpha = \pm 1; i = 1, 2, 3 \}$$

$$(3.28) \quad \mathcal{K} = \{ \mathbf{x} \in \mathbb{R}^3 \mid 0 < \phi_i^\alpha \phi_j^\beta \phi_k^\gamma < 1 \}.$$

Individual edges and faces will be denoted by $\mathcal{E}_{ij}^{\alpha\beta}$ and \mathcal{F}_i^α respectively. For the nodes, edges and faces on \mathcal{K} we have relations identical to (3.17) and (3.18).

A nodal basis on \mathcal{K} is derived in the same manner as the directional basis in (3.24) but now the composition is between G and the functions (3.19):

$$(3.29) \quad N^{\alpha\beta\gamma}(\mathbf{x}) = (\hat{N}^{\alpha\beta\gamma} \circ G)(\mathbf{x}).$$

Parametrization of faces and edges. Consider first parametrization of the surfaces in \mathcal{K} . Since F maps the reference face $\hat{\mathcal{F}}_i^\alpha$ into the face \mathcal{F}_i^α of \mathcal{K} , parametrization of \mathcal{F}_i^α is simply the restriction of F to the reference face. The ensuing parametrization function is a $\mathbb{R}^2 \mapsto \mathbb{R}^3$ mapping given by

$$(3.30) \quad \Phi_i^\alpha(\boldsymbol{\xi}) = F(\boldsymbol{\xi} \mid \xi_i = \alpha).$$

The coordinate functions of this mapping are the polynomials (3.23) with the ξ_i th reference variable set to +1 or -1. Therefore, face parametrization is by *bilinear* polynomials in ξ_j and ξ_k . If \mathcal{K} is a parallelepiped, these polynomials simplify to an affine mapping.

Parametrization of edges can be obtained in the same manner. First we note that the F maps the reference edge $\hat{\mathcal{E}}_{ij}^{\alpha\beta}$ to the edge $\mathcal{E}_{ij}^{\alpha\beta}$ of \mathcal{K} . As a result, parametrization of this edge is available by restriction of F to $\hat{\mathcal{E}}_{ij}^{\alpha\beta}$ which gives the $\mathbb{R} \mapsto \mathbb{R}^3$ mapping

$$(3.31) \quad E_{ij}^{\alpha\beta}(\boldsymbol{\xi}) = F(\boldsymbol{\xi} | \xi_i = \alpha; \xi_j = \beta).$$

The coordinate functions of the edge parametrization are the polynomials (3.23) in which ξ_i th and ξ_j th reference variables are set to +1 or -1. These functions are linear polynomials in ξ_k .

Algebraic properties. We recall the standard definitions of the Jacobian matrices

$$J_F = \begin{pmatrix} \frac{\partial F_1}{\partial \xi_1} & \frac{\partial F_1}{\partial \xi_2} & \frac{\partial F_1}{\partial \xi_3} \\ \frac{\partial F_2}{\partial \xi_1} & \frac{\partial F_2}{\partial \xi_2} & \frac{\partial F_2}{\partial \xi_3} \\ \frac{\partial F_3}{\partial \xi_1} & \frac{\partial F_3}{\partial \xi_2} & \frac{\partial F_3}{\partial \xi_3} \end{pmatrix} \quad \text{and} \quad J_G = \begin{pmatrix} \frac{\partial G_1}{\partial x_1} & \frac{\partial G_1}{\partial x_2} & \frac{\partial G_1}{\partial x_3} \\ \frac{\partial G_2}{\partial x_1} & \frac{\partial G_2}{\partial x_2} & \frac{\partial G_2}{\partial x_3} \\ \frac{\partial G_3}{\partial x_1} & \frac{\partial G_3}{\partial x_2} & \frac{\partial G_3}{\partial x_3} \end{pmatrix}$$

for F and G , respectively. $J_F(\boldsymbol{\xi})$ and $J_G(\mathbf{x})$ are linear maps in \mathbb{R}^3 which are invertible for every $\boldsymbol{\xi} \in \hat{\mathcal{K}}$ and $\mathbf{x} \in \mathcal{K}$. The transpose gradients of the coordinate mappings F_i and G_j form the rows of the two Jacobians. The columns of J_F will play special role in all further developments so we denote them by V_i .

Differentiation of the equation $(G \circ F)(\boldsymbol{\xi}) = \boldsymbol{\xi}$ gives the identity $J_G J_F = I$. In terms of dot products between the rows of J_G and the columns of J_F this identity is equivalent to

$$(3.32) \quad \nabla G_j \cdot V_i = \delta_{ij}, \quad i, j = 1, 2, 3.$$

Thus, $\{V_i\}$ and $\{\nabla G_i\}$ form a base and reciprocal sets of vectors. Using Cramer's rule, the identities

$$\det J_F = (V_i \times V_j) \cdot V_k, \quad \det J_G = (\nabla G_i \times \nabla G_j) \cdot \nabla G_k = (\det J_F)^{-1}$$

and (3.32), we obtain the following relations:

$$(3.33) \quad \nabla G_i = \frac{1}{\det J_F} (V_j \times V_k) \quad \text{and} \quad V_i = \det J_F (\nabla G_j \times \nabla G_k);$$

where (ijk) is a cyclic permutation of $(1, 2, 3)$. Another useful identity that follows from (3.33) is

$$(3.34) \quad \nabla G_i \times \nabla G_j = \frac{1}{\det J_F} V_k$$

where (ijk) is again an even permutation of the indices 1, 2, 3.

Considered as $\mathbb{R}^3 \mapsto \mathbb{R}^3$ functions, the edge and face parametrizations Φ_i^α and $E_{ij}^{\alpha\beta}$ are degenerate mappings. Their Jacobians can be expressed in terms of the vectors V_i that form the columns of J_F . For example,

$$J_F = \begin{pmatrix} V_1 & V_2 & V_3 \end{pmatrix}; J_{\Phi_3^\alpha} = \begin{pmatrix} V_1 & V_2 & 0 \end{pmatrix}; J_{E_{23}^{\alpha\beta}} = \begin{pmatrix} V_1 & 0 & 0 \end{pmatrix}.$$

Volume, surface and line integrals on \mathcal{K} . Let $f(\mathbf{x})$ denote a real valued, continuous scalar function defined on \mathcal{K} . We will need to evaluate integrals of such functions on \mathcal{K} as well as on its faces and edges. These integrals can be computed on the reference element $\hat{\mathcal{K}}$ using the appropriate parametrization. For \mathcal{K} the parametrization is given by F in (3.21) and

$$\begin{aligned} \int_{\mathcal{K}} f(\mathbf{x}) d\mathbf{x} &= \int_{\hat{\mathcal{K}}} (f \circ F)(\boldsymbol{\xi}) |\det J_F| d\boldsymbol{\xi} \\ (3.35) \qquad \qquad &= \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (f \circ F)(\boldsymbol{\xi}) |V_i \cdot (V_j \times V_k)| d\xi_1 d\xi_2 d\xi_3. \end{aligned}$$

Parametrization of a face \mathcal{F}_i^α is provided by the function Φ_i^α . In view of (3.30), the standard definition of a surface integral; see [27, p.322], specializes to

$$\begin{aligned} \int_{\mathcal{F}_i^\alpha} f(\mathbf{x}) dS &= \int_{\hat{\mathcal{F}}_i^\alpha} (f \circ \Phi_i^\alpha)(\boldsymbol{\xi}) \|V_j \times V_k\| d\xi_j d\xi_k \\ (3.36) \qquad \qquad &= \int_{-1}^1 \int_{-1}^1 \left[(f \circ F)(\boldsymbol{\xi}) \|V_j \times V_k\| \right]_{\xi_i=\alpha} d\xi_j d\xi_k. \end{aligned}$$

We can also specialize the standard definition of a path integral; see [27, p.282] to integrals along the edges of \mathcal{K} . Parametrization of an edge $\mathcal{E}_{ij}^{\alpha\beta}$ is given by the mapping $E_{ij}^{\alpha\beta}(\boldsymbol{\xi})$, and in view of (3.31)

$$\begin{aligned} \int_{\mathcal{E}_{ij}^{\alpha\beta}} f(\mathbf{x}) dl &= \int_{\hat{\mathcal{E}}_{ij}^{\alpha\beta}} (f \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) \|V_k\| d\xi_k \\ (3.37) \qquad \qquad &= \int_{-1}^1 \left[(f \circ F)(\boldsymbol{\xi}) \|V_k\| \right]_{\xi_i=\alpha, \xi_j=\beta} d\xi_k. \end{aligned}$$

In addition to the above integrals, we will also need to compute the integrals

$$\int_{\mathcal{F}_i^\alpha} \mathbf{f}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) dS \quad \text{and} \quad \int_{\mathcal{E}_{ij}^{\alpha\beta}} \mathbf{f}(\mathbf{x}) \cdot \mathbf{t}(\mathbf{x}) dl$$

giving the flux and circulation of a vector field across faces and along edges of \mathcal{K} , respectively.

To compute the flux of \mathbf{f} across \mathcal{F}_i^α , note that $\phi_i^\alpha = 1$ on this face and, as a result, $\nabla\phi_i^\alpha$ is normal at every point of the surface \mathcal{F}_i^α . Since $\nabla\phi_i^\alpha = (\alpha/2)\nabla G_i$, the vector

$$(3.38) \quad \mathbf{n}(\mathbf{x}) = \frac{\nabla G_i}{\|\nabla G_i\|}.$$

defines a unit normal to \mathcal{F}_i^α . To apply (3.36) we must find the image of this vector on $\hat{\mathcal{K}}$ under the face parametrization Φ_i^α . From (3.33) it follows that

$$(3.39) \quad (\mathbf{n} \circ \Phi_i^\alpha)(\boldsymbol{\xi}) = \left[(V_j \times V_k) / \|V_j \times V_k\| \right]_{\xi_i=\alpha}$$

and so the flux of \mathbf{f} is given by

$$(3.40) \quad \begin{aligned} \int_{\mathcal{F}_i^\alpha} \mathbf{f}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) dS &= \int_{\hat{\mathcal{F}}_i^\alpha} (\mathbf{f} \circ \Phi_i^\alpha)(\boldsymbol{\xi}) \cdot (\mathbf{n} \circ \Phi_i^\alpha)(\boldsymbol{\xi}) \|V_j \times V_k\| d\xi_j \xi_k \\ &= \int_{\hat{\mathcal{F}}_i^\alpha} (\mathbf{f} \circ \Phi_i^\alpha)(\boldsymbol{\xi}) \cdot \frac{V_j \times V_k}{\|V_j \times V_k\|} \|V_j \times V_k\| d\xi_j \xi_k \\ &= \int_{-1}^1 \int_{-1}^1 \left[(\mathbf{f} \circ F)(\boldsymbol{\xi}) \cdot (V_j \times V_k) \right]_{\xi_i=\alpha} d\xi_j \xi_k. \end{aligned}$$

A unit tangential vector for the edge $\mathcal{E}_{ij}^{\alpha\beta}$ can be determined by observing that $\nabla\phi_i^\alpha$ and $\nabla\phi_j^\beta$ are normal to the two faces which intersect at this edge(see (3.17)). As a result, the vector

$$(3.41) \quad \mathbf{t} = \frac{\nabla G_i \times \nabla G_j}{\|\nabla G_i \times \nabla G_j\|}$$

is tangential to $\mathcal{E}_{ij}^{\alpha\beta}$ and has unit length. To compute the circulation of \mathbf{f} according to (3.37) it is necessary to find the image of \mathbf{t} on the reference element. Using (3.33) in (3.41) shows that

$$(3.42) \quad (\mathbf{t} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) = \left[V_k / \|V_k\| \right]_{\xi_i=\alpha, \xi_j=\beta}.$$

As a result, circulation of \mathbf{f} along $\mathcal{E}_{ij}^{\alpha\beta}$ is given by

$$(3.43) \quad \begin{aligned} \int_{\mathcal{E}_{ij}^{\alpha\beta}} \mathbf{f}(\mathbf{x}) \cdot \mathbf{t}(\mathbf{x}) dl &= \int_{\hat{\mathcal{E}}_{ij}^{\alpha\beta}} (\mathbf{f} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) \cdot (\mathbf{t} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) \|V_k\| d\xi_k \\ &= \int_{\hat{\mathcal{E}}_{ij}^{\alpha\beta}} (\mathbf{f} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) \cdot \frac{V_k}{\|V_k\|} \|V_k\| d\xi_k \\ &= \int_{-1}^1 \left[(\mathbf{f} \circ F)(\boldsymbol{\xi}) \cdot V_k \right]_{\xi_i=\alpha, \xi_j=\beta} d\xi_k. \end{aligned}$$

3.2.3 The van Welij complex in \mathbb{R}^3

We now proceed to define the spaces \mathcal{W}^i . These spaces are related to the sets \mathcal{N} , \mathcal{E} and \mathcal{F} of all nodes, edges and faces on \mathcal{K} , and the hexahedral \mathcal{K} itself. As a result, characterizations of these sets in terms of the directional basis functions in (3.25)-(3.28) provide the guidelines for developing bases for each space \mathcal{W}^i .

The space \mathcal{W}^0 . We introduce $\mathcal{W}^0(\mathcal{K})$ as the set of all functions $q(\mathbf{x})$ defined on \mathcal{K} that are images of trilinear polynomials defined on the reference element $\hat{\mathcal{K}}$ under the mapping G . Symbolically, we write this as

$$(3.44) \quad \mathcal{W}^0 = \{q = (\hat{q} \circ G)(\mathbf{x}) \mid \hat{q} \in Q_1(\hat{\mathcal{K}})\}.$$

Unless \mathcal{K} is a parallelepiped, the functions in $\mathcal{W}^0(\mathcal{K})$ are not polynomials. However, each function is still uniquely determined by its nodal values. Therefore, to describe \mathcal{W}^0 it is sufficient to exhibit a set of nodal basis functions on \mathcal{K} . We have already constructed such basis in (3.29) by mapping the reference basis (3.19) to \mathcal{K} . This is in fact the standard method for defining affine families of finite elements and it works equally well for isoparametric nodal spaces; see e.g., [18, p.106].

There is, however, an alternative, direct way of coming up with the same basis without appealing to a local reference basis. It is based on the characterization of \mathcal{N} in (3.25) in terms of the directional functions ϕ_i^α . This representation indicates that the set

$$(3.45) \quad W_{ijk}^{\alpha\beta\gamma}(\mathbf{x}) = \phi_i^\alpha(\mathbf{x})\phi_j^\beta(\mathbf{x})\phi_k^\gamma(\mathbf{x}), \quad \mathbf{x} \in \mathcal{K}.$$

defines a basis on $\mathcal{W}^0(\mathcal{K})$. Therefore, we can circumvent the reference basis and define $\mathcal{W}^0(\mathcal{K})$ directly as

$$(3.46) \quad \mathcal{W}^0(\mathcal{K}) = \text{span} \{W_{ijk}^{\alpha\beta\gamma}(\mathbf{x})\}.$$

In the present case it is easy to see that

$$W_{ijk}^{\alpha\beta\gamma}(\mathbf{x}) = (\hat{N}^{\alpha\beta\gamma} \circ G)(\mathbf{x}) = N^{\alpha\beta\gamma}(\mathbf{x}),$$

which proves that (3.44) and (3.45) are equivalent to each other

A few comments are now in order. We saw that a nodal basis on \mathcal{K} can be defined either by mapping a nodal basis from the reference element or by working from a characterization of the DOF set in terms of the directional basis on \mathcal{K} . It turns out that the second method, namely deriving basis functions directly from representations of nodes, edges and faces in terms of the directional bases, is more intuitive for vector valued isoparametric spaces. Such are the next two spaces we are about to develop. These spaces will have degrees of freedom that

depend on the tangential and normal directions to the edges and faces of \mathcal{K} . If we approach construction of such fields by first trying to develop like fields on the reference element, the correct form of the reference basis vectors would be very unintuitive unless \mathcal{K} is a rectangular brick. This is caused by the fact that tangential and normal vectors on \mathcal{K} are mapped back to the reference element according to (3.42) and (3.39), respectively. The reference element approach also makes it more difficult to find the proper normalizations of the vector bases so as to ensure unit fluxes and circulations. This is caused by the nonlinearity of the mapping F between $\hat{\mathcal{K}}$ and \mathcal{K} .

The space \mathcal{W}^1 . We define $\mathcal{W}^1(\mathcal{K})$ to be the set of all vector fields \mathbf{v} on \mathcal{K} that are uniquely determined by their circulations along the edges of \mathcal{K} and which are images of certain *rational* vector fields $\hat{\mathbf{v}}$ defined on the reference element $\hat{\mathcal{K}}$. Symbolically,

$$(3.47) \quad \mathcal{W}^1(\mathcal{K}) = \{\mathbf{v} = (\hat{\mathbf{v}} \circ G)(\mathbf{x}) \mid \hat{\mathbf{v}} \in R(\hat{\mathcal{K}})\}.$$

We postpone the discussion of the exact form of the prototypes $\hat{\mathbf{v}}$ until after an explicit basis for \mathcal{W}^1 is derived. Since \mathcal{E} is the set of all degrees of freedom (DOF) for $\mathcal{W}^1(\mathcal{K})$, to describe this space we need to exhibit a set of vector fields whose circulations are zero along all but one of the edges. These fields will provide the basis for \mathcal{W}^1 . An intuitive way of developing such basis directly on \mathcal{K} was first suggested by van Welij in [37]. Here we formalize the original method of van Welij in a manner that is consistent with our goal of developing an exact sequence. For this purpose we derive the basis starting from the edge characterization (3.26) in terms of directional bases. To illustrate the main idea consider the edge

$$\mathcal{E}_{23}^{++} = \{\mathbf{x} \in \mathcal{K} \mid \phi_2^+(\mathbf{x})\phi_3^+(\mathbf{x}) = 1\}.$$

We seek a vector field with unit circulation along \mathcal{E}_{23}^{++} and whose tangential component is zero along all other edges. The edge representation in (3.26) suggests to consider

$$W_{23}^{++} = \phi_2^+(\mathbf{x})\phi_3^+(\mathbf{x})\mathbf{v}$$

where \mathbf{v} is a vector field to be determined. The circulation of this function is zero along the three “parallel” edges \mathcal{E}_{23}^{+-} , \mathcal{E}_{23}^{-+} and \mathcal{E}_{23}^{--} where either $\phi_2^+ = 0$ or $\phi_3^+ = 0$. The choice of \mathbf{v} will lead to zero circulations along the remaining eight edges. These edges frame the two faces $\mathcal{F}_1^\pm = \{\mathbf{x} \in \mathcal{K} \mid \phi_1^\pm = 1\}$ pierced by \mathcal{E}_{23}^{++} . On each face $G_1 = \text{const}$ and therefore, the vector ∇G_1 has no tangential component along these edges. However, for nondegenerate \mathcal{K} this vector will have a tangential component along \mathcal{E}_{23}^{++} . The same is true for $-\nabla G_1$ and we can take either one of these two vectors for \mathbf{v} . Since $\nabla \phi_1^\alpha = (\alpha/2)\nabla G_1$, the orientation choice can be accounted for by setting $\mathbf{v} = \nabla \phi_1^\alpha$ so that

$$W_{23}^{++} = \phi_2^+\phi_3^+\nabla \phi_1^\alpha.$$

Using the same argument for the remaining faces we find that

$$(3.48) \quad \mathcal{W}^1(\mathcal{K}) = \text{span} \{W_{ij}^{\alpha\beta}(\mathbf{x})\}, \quad W_{ij}^{\alpha\beta} = \phi_i^\alpha \phi_j^\beta \nabla \phi_k^\gamma,$$

where γ is fixed and represents $+1$ or -1 . The number of distinct basis functions in (3.48) is exactly twelve. The choice of γ determines the orientation of the vector field along a given edge but does not change the number of independent basis functions. The choice of $\nabla \phi_k^\gamma$ as a direction field for $W_{ij}^{\alpha\beta}$ is not incidental. One can argue that a better choice would be the unit tangent defined in (3.41). However, unless \mathcal{K} is a rectangular brick, this field will have a nonzero tangential component along the edges of the faces pierced by $\mathcal{E}_{ij}^{\alpha\beta}$.

By changing variables in $W_{ij}^{\alpha\beta}$ we can find its prototype on the reference element $\hat{\mathcal{K}}$. Using (3.33) in (3.48) gives the twelve vector fields

$$(3.49) \quad \hat{W}_{ij}^{\alpha\beta}(\boldsymbol{\xi}) = (W_{ij}^{\alpha\beta} \circ F)(\boldsymbol{\xi}) = \frac{\gamma}{2\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi}) \hat{\phi}_j^\beta(\boldsymbol{\xi}) (V_i \times V_j)$$

defined on $\hat{\mathcal{K}}$. Their span forms the prototype space $\hat{\mathcal{W}}^1(\hat{\mathcal{K}})$ containing the preimages $\hat{\mathbf{v}}$ of the functions in $\mathcal{W}^1(\mathcal{K})$. It is clear that we could have defined $\mathcal{W}^1(\mathcal{K})$ starting from the basis (3.49) and deriving (3.48) by mapping this basis to \mathcal{K} . We see that there's a striking difference between the types of functions in $\hat{\mathcal{W}}^1(\hat{\mathcal{K}})$ and in $Q_1(\hat{\mathcal{K}})$, which was the prototype for $\mathcal{W}^0(\mathcal{K})$. It is clear that unless \mathcal{K} is a rectangular brick (so that $\det J_F = \text{const}$) the functions in $\hat{\mathcal{W}}^1(\hat{\mathcal{K}})$ are not polynomials but rational functions.

Let us now use (3.49) to prove that $W_{ij}^{\alpha\beta}$ indeed have unit circulation along $\mathcal{E}_{ij}^{\alpha\beta}$ (their circulation along all other edges is zero by virtue of their construction!). By the definition of the path integral (3.37)

$$\int_{\mathcal{E}_{ij}^{\alpha\beta}} W_{ij}^{\alpha\beta}(\mathbf{x}) \cdot \mathbf{t}(\mathbf{x}) dl = \int_{\hat{\mathcal{E}}_{ij}^{\alpha\beta}} (W_{ij}^{\alpha\beta} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) \cdot (\mathbf{t} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) \|V_k\| d\xi_k$$

where \mathbf{t} is the vector defined in (3.41). To complete the change of variables, recall that parametrization of edges in (3.31) is obtained by restriction of two of the reference variables. As a result, restricted to an edge, (3.49) specializes to

$$(W_{ij}^{\alpha\beta} \circ E_{ij}^{\alpha\beta})(\boldsymbol{\xi}) = \left[(W_{ij}^{\alpha\beta} \circ F)(\boldsymbol{\xi}) \right]_{\xi_i=\alpha, \xi_j=\beta} = \frac{\gamma}{2\det J_F} (V_i \times V_j).$$

The last identity holds because $\hat{\phi}_i^\alpha(\boldsymbol{\xi}) \hat{\phi}_j^\beta(\boldsymbol{\xi}) = 1$ along $\hat{\mathcal{E}}_{ij}^{\alpha\beta}$. In combination with (3.42)

$$\begin{aligned} \int_{\mathcal{E}_{ij}^{\alpha\beta}} W_{ij}^{\alpha\beta}(\mathbf{x}) \cdot \mathbf{t}(\mathbf{x}) dl &= \int_{\hat{\mathcal{E}}_{ij}^{\alpha\beta}} \frac{\gamma(V_i \times V_j)}{2\det J_F} \cdot \frac{V_k}{\|V_k\|} \|V_k\| d\xi_k \\ &= \int_{-1}^1 \frac{\gamma(V_i \times V_j) \cdot V_k}{2\det J_F} d\xi_k \\ &= \int_{-1}^1 \frac{\gamma \det J_F}{2\det J_F} d\xi_k = \frac{\gamma}{2} \int_{-1}^1 d\xi_k = \gamma. \end{aligned}$$

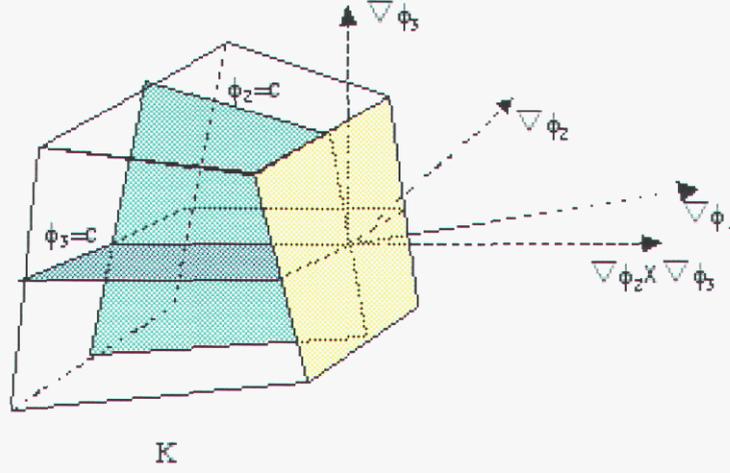


Figure 3.2. Vector fields with nonzero normal component across a face

Equation (3.49) also shows why it is more intuitive to develop the vector bases directly on \mathcal{K} . Indeed, on a rectangular reference element, $\nabla \hat{\phi}_k^\gamma$ is collinear with all edges perpendicular to the face $\hat{\phi}_k^\gamma = 1$. As a result, it is tempting to define the basis function for the edge $\hat{\mathcal{E}}_{ij}^{\alpha\beta}$ as

$$\hat{\phi}_i^\alpha(\boldsymbol{\xi}) \hat{\phi}_j^\beta(\boldsymbol{\xi}) \nabla \hat{\phi}_k^\gamma(\boldsymbol{\xi}).$$

Using relations symmetric to (3.32) and (3.33) shows that mapping this vector field back to \mathcal{K} gives

$$\phi_i^\alpha(\mathbf{x}) \phi_j^\beta(\mathbf{x}) (U_i \times U_j) / \det J_G$$

where U_i denote the columns of J_G . Except for the trivial cases of rectangular bricks this will not give a function with the desired circulation property on \mathcal{K} .

The space \mathcal{W}^2 . We define $\mathcal{W}^2(\mathcal{K})$ to be the set of all vector fields that are uniquely determined by their fluxes across the faces of \mathcal{K} and which are images of certain rational vector fields $\hat{\mathbf{v}}$ defined on $\hat{\mathcal{K}}$, i.e.,

$$(3.50) \quad \mathcal{W}^2(\mathcal{K}) = \{\mathbf{v} = (\hat{\mathbf{v}} \circ G)(\mathbf{x}) \mid \hat{\mathbf{v}} \in R(\hat{\mathcal{K}})\}.$$

As before, we postpone the discussion on the exact nature of the prototype fields until after an explicit basis for $\mathcal{W}^2(\mathcal{K})$ is produced.

The set of all degrees of freedom for $\mathcal{W}^2(\mathcal{K})$ is represented by the set \mathcal{F} of all faces on \mathcal{K} . Therefore, to describe $\mathcal{W}^2(\mathcal{K})$ it is necessary to exhibit basis functions which have unit flux

across one of the faces and zero flux across all other faces. Again, we make no attempts to find and map suitable prototypes from the reference element, instead we use the face characterization (3.27) to develop the desired fields directly on \mathcal{K} . To illustrate this process, consider the face $\mathcal{F}_1^+ = \{\mathbf{x} \in \mathcal{K} | \phi_1^+(\mathbf{x}) = 1\}$ of \mathcal{K} . Here we need a field with a unit flux across this face and whose normal component is zero across all other faces of \mathcal{K} . The face description in terms of the directional basis is again the clue that prompts us to seek this field in the form

$$W_1^+ = \phi_1^+(\mathbf{x})\mathbf{v}.$$

This choice ensures that W_1^+ is zero on the opposite face \mathcal{F}_1^- . To make the flux of this function zero on the remaining four faces \mathcal{F}_2^\pm and \mathcal{F}_3^\pm we seek \mathbf{v} that is tangential to these faces and has a component in the direction of $\nabla\phi_1^+$ (the normal to \mathcal{F}_1^+). To find such a \mathbf{v} , note that ∇G_2 is normal to \mathcal{F}_2^\pm and ∇G_3 - to \mathcal{F}_3^\pm . Therefore, the vector $\nabla G_2 \times \nabla G_3$ is tangential at all points of these four faces; see Fig. 3.2. At the same time, if \mathcal{K} is convex,

$$\det J_G = (\nabla G_2 \times \nabla G_3) \cdot \nabla G_1 \neq 0$$

which means that $\nabla G_2 \times \nabla G_3$ has a nonzero component in the direction of $\nabla\phi_1^+$. We account for the possible orientation choices by setting $\mathbf{v} = \nabla\phi_2^\beta \times \nabla\phi_3^\gamma$. The resulting vector field

$$W_1^+ = \phi_1^+ \left(\nabla\phi_2^\beta \times \nabla\phi_3^\gamma \right),$$

where β and γ are fixed, has the desired property and

$$(3.51) \quad \mathcal{W}^2(\mathcal{K}) = \text{span} \{W_i^\alpha(\mathbf{x})\}, \quad W_i^\alpha = \phi_i^\alpha \left(\nabla\phi_j^\beta \times \nabla\phi_k^\gamma \right).$$

The number of basis functions in (3.51) is six. The two indices β and γ determine the orientation of the vector basis function. It is convenient to choose these so that the vector field is outward pointing.

Changing variables in (3.51) gives the six fields

$$(3.52) \quad \hat{W}_i^\alpha(\boldsymbol{\xi}) = (W_i^\alpha \circ F)(\boldsymbol{\xi}) = \frac{\beta\gamma}{4\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi}) V_i$$

defined on the reference element. These fields are the prototypes of the basis functions (3.51). Their span forms the space $\hat{\mathcal{W}}^2(\hat{\mathcal{K}})$ which contains preimages of all functions in $\mathcal{W}^2(\mathcal{K})$. It is also obvious that $\hat{\mathcal{W}}^2(\hat{\mathcal{K}})$ contains rational functions unless \mathcal{K} is a rectangular brick.

By construction, the basis function W_i^α has zero flux on all faces except \mathcal{F}_i^α . Using (3.52) we can show that the flux of W_i^α across this face is one. By the definition of the surface integral (3.36)

$$\int_{\mathcal{F}_i^\alpha} W_i^\alpha(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) dS = \int_{\hat{\mathcal{F}}_i^\alpha} (W_i^\alpha \circ \Phi_i^\alpha)(\boldsymbol{\xi}) \cdot (\mathbf{n} \circ \Phi_i^\alpha)(\boldsymbol{\xi}) \|V_j \times V_k\| d\xi_j \xi_k,$$

where \mathbf{n} is the unit normal defined in (3.38). The parametrization Φ_i^α of this face is obtained from F by setting $\xi_i = \alpha$. Since $\phi_i^\alpha = 1$ on \mathcal{F}_i^α formula (3.52) specializes to

$$(W_i^\alpha \circ \Phi_i^\alpha)(\boldsymbol{\xi}) = \left[(W_i^\alpha \circ F)(\boldsymbol{\xi}) \right]_{\xi_i=\alpha} = \frac{\beta\gamma}{4\det J_F} V_i.$$

From (3.39) and this expression we see that

$$\begin{aligned} \int_{\mathcal{F}_i^\alpha} W_i^\alpha(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) dS &= \int_{\hat{\mathcal{F}}_i^\alpha} \frac{\beta\gamma V_i \cdot (V_j \times V_k)}{4\det J_F \|V_j \times V_k\|} \|V_j \times V_k\| d\xi_j \xi_k \\ &= \frac{\beta\gamma}{4} \int_{-1}^1 \int_{-1}^1 d\xi_j \xi_k = \beta\gamma. \end{aligned}$$

The space \mathcal{W}^3 . We define $\mathcal{W}^3(\mathcal{K})$ as the space of all scalar functions on \mathcal{K} that are uniquely determined by their total mass on \mathcal{K} and which are images of a suitable class of functions defined on $\hat{\mathcal{K}}$. We can interpret such functions as a density distribution on \mathcal{K} .

Functions in $\mathcal{W}^3(\mathcal{K})$ have a single degree of freedom identified with \mathcal{K} . To describe all such functions we need a basis function which has unit total mass on \mathcal{K} . This basis function must be related to the volume deformation under the mapping G and so we set

$$(3.53) \quad \mathcal{W}^3(\mathcal{K}) = \text{span} \{W_{\mathcal{K}}\}; \quad W_{\mathcal{K}} = \left(\nabla \phi_i^\alpha \times \nabla \phi_j^\beta \right) \cdot \nabla \phi_k^\gamma$$

where α, β and γ are fixed. The choice of basis in (3.53) is motivated by the formula

$$\det J_G = (\nabla G_i \times \nabla G_j) \cdot \nabla G_k$$

which implies that

$$|W_{\mathcal{K}}| = |\nabla \phi_i^\alpha \cdot (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma)| = \frac{|\alpha\beta\gamma|}{8} \det J_G = \det J_G / 8.$$

In a moment we will show that this is the proper normalization to obtain unit total mass.

To find the prototype of $W_{\mathcal{K}}$ on a reference element we change variables in (3.53):

$$(3.54) \quad \hat{W}_{\hat{\mathcal{K}}}(\boldsymbol{\xi}) = (W_{\mathcal{K}} \circ F)(\boldsymbol{\xi}) = \frac{\alpha\beta\gamma}{8\det J_F}.$$

This prototype is a rational function which can be interpreted as a density distribution on $\hat{\mathcal{K}}$. If \mathcal{K} is a rectangular brick, (3.54) corresponds to a constant density in which case we recover the standard space Q_0 of zero degree polynomials. If \mathcal{K} is not too distorted, Q_0 is a good approximation to \mathcal{W}^3 .

Table 3.1. Discrete De Rham complex on a hexahedral \mathcal{K}

SPACE	DOF LOCATION	BASIS	DIMENSION
\mathcal{W}^0	$\phi_i^\alpha \phi_j^\beta \phi_k^\gamma = 1$	$\phi_i^\alpha \phi_j^\beta \phi_k^\gamma$	$2^d \binom{d}{d} = 8$
\mathcal{W}^1	$\phi_i^\alpha \phi_j^\beta = 1$	$\phi_i^\alpha \phi_j^\beta \nabla \phi_k^\gamma$	$2^{d-1} \binom{d}{d-1} = 12$
\mathcal{W}^2	$\phi_i^\alpha = 1$	$\phi_i^\alpha (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma)$	$2^{d-2} \binom{d}{d-2} = 6$
\mathcal{W}^3	\mathcal{K}	$\nabla \phi_i^\alpha (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma)$	$2^{d-3} \binom{d}{d-3} = 1$

It remains to show that $\mathcal{W}_{\mathcal{K}}$ has unit mass. For this purpose we use (3.35) and (3.54) to find that

$$\begin{aligned} \int_{\mathcal{K}} \mathcal{W}_{\mathcal{K}}(\mathbf{x}) d\mathbf{x} &= \int_{\hat{\mathcal{K}}} (\mathcal{W}_{\mathcal{K}} \circ F)(\boldsymbol{\xi}) |\det J_F| d\boldsymbol{\xi} = \int_{\hat{\mathcal{K}}} \frac{\alpha\beta\gamma}{8 \det J_F} |\det J_F| d\boldsymbol{\xi} \\ &= \frac{\alpha\beta\gamma}{8} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 d\xi_1 \xi_2 \xi_3 = \alpha\beta\gamma \end{aligned}$$

The finite element complex on hexahedra is summarized in Table 3.1.

Exactness of the complex \mathcal{W}^i . Let us now show that $\mathcal{W}^i(\mathcal{K})$ form an exact sequence. Recall that for vectors fields \mathbf{U} , \mathbf{V} and a scalar function u

$$(3.55) \quad \nabla \times (u\mathbf{V}) = u\nabla \times \mathbf{V} + \nabla u \times \mathbf{V},$$

$$(3.56) \quad \nabla \cdot (u\mathbf{V}) = \nabla u \cdot \mathbf{V} + u\nabla \cdot \mathbf{V},$$

and

$$(3.57) \quad \nabla \cdot (\mathbf{U} \times \mathbf{V}) = \mathbf{V} \cdot (\nabla \times \mathbf{U}) - \mathbf{U} \cdot (\nabla \times \mathbf{V}).$$

From (3.57) also follows that

$$(3.58) \quad \nabla \cdot (\nabla f \times \nabla g) = 0.$$

Consider first the gradients of $\mathcal{W}^0(\mathcal{K})$ functions. Using the chain rule and (3.45)

$$(3.59) \quad \nabla W_{ijk}^{\alpha\beta\gamma} = \sum_{(ijk)_+} \phi_i^\alpha \phi_j^\beta \nabla \phi_k^\gamma = \sum_{(ijk)_+} \sigma_k^\gamma W_{ij}^{\alpha\beta},$$

where summation is over cyclic permutations of (1, 2, 3) and $\sigma_k^\gamma = \pm 1$ depending on the orientation choice of $W_{ij}^{\alpha\beta}$. This establishes the inclusion

$$(3.60) \quad \nabla(\mathcal{W}^0(\mathcal{K})) \subset \mathcal{W}^1(\mathcal{K}).$$

Consider next curls of $\mathcal{W}^1(\mathcal{K})$ functions. Using (3.55) for $W_{ij}^{\alpha\beta}$ gives

$$(3.61) \quad \begin{aligned} \nabla \times W_{ij}^{\alpha\beta} &= \nabla \times (\phi_i^\alpha \phi_j^\beta \nabla \phi_k^\gamma) \\ &= \phi_i^\alpha (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma) + \phi_j^\beta (\nabla \phi_i^\alpha \times \nabla \phi_k^\gamma) \\ &= \sigma_{jk}^{\beta\gamma} W_i^\alpha - \sigma_{ik}^{\alpha\gamma} W_j^\beta. \end{aligned}$$

The minus sign comes from the permutation (jik) while $\sigma_{jk}^{\beta\gamma}$ and $\sigma_{ik}^{\alpha\gamma}$ equal ± 1 depending on the orientation of W_i^α and W_j^β , respectively. This establishes the inclusion

$$(3.62) \quad \nabla \times (\mathcal{W}^1(\mathcal{K})) \subset \mathcal{W}^2(\mathcal{K}).$$

Finally, consider divergences of $\mathcal{W}^2(\mathcal{K})$ functions. From (3.56)

$$\begin{aligned} \nabla \cdot W_i^\alpha &= \nabla \cdot ((\phi_i^\alpha \nabla \phi_j^\beta \times \nabla \phi_k^\gamma)) \\ &= \nabla \phi_i^\alpha (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma) + \phi_i^\alpha \nabla \cdot (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma) \end{aligned}$$

From (3.24) $\nabla \phi_i^\alpha = \alpha/2 \nabla G_i$. Since $\nabla G_k \cdot (\nabla G_i \times \nabla G_j) = \det J_G$ the first term above equals

$$\frac{\alpha\beta\gamma}{8} \det J_G.$$

The second term vanishes thanks to (3.58) and we find that

$$(3.63) \quad \nabla \cdot W_i^\alpha = \frac{\alpha\beta\gamma}{8} \det J_G = \alpha\beta\gamma W_K.$$

This establishes the inclusion

$$(3.64) \quad \nabla \cdot (\mathcal{W}^2(\mathcal{K})) \subset \mathcal{W}^3(\mathcal{K}).$$

Change of variables. Assembly of finite element matrices requires computation of integrals involving both the basis functions and their derivatives. These integrals are computed on the reference element using standard quadrature rules which necessitates a change of variables

from \mathcal{K} to $\hat{\mathcal{K}}$. Equations (3.49), (3.52) and (3.54) show the local reference bases resulting from such a change. For convenience these formulas are collected below.

$$\begin{aligned}
(W_{ijk}^{\alpha\beta\gamma} \circ F)(\boldsymbol{\xi}) &= \hat{N}^{\alpha\beta\gamma}(\boldsymbol{\xi}) = \hat{\phi}_i^\alpha(\boldsymbol{\xi})\hat{\phi}_j^\beta(\boldsymbol{\xi})\hat{\phi}_k^\gamma(\boldsymbol{\xi}) \\
(W_{ij}^{\alpha\beta} \circ F)(\boldsymbol{\xi}) &= \hat{W}_{ij}^{\alpha\beta}(\boldsymbol{\xi}) = \frac{\gamma}{2\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi})\hat{\phi}_j^\beta(\boldsymbol{\xi})(V_i \times V_j) \\
(W_i^\alpha \circ F)(\boldsymbol{\xi}) &= \hat{W}_i^\alpha(\boldsymbol{\xi}) = \frac{\beta\gamma}{4\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi})V_i \\
(W_K \circ F)(\boldsymbol{\xi}) &= \hat{W}_K(\boldsymbol{\xi}) = \frac{\alpha\beta\gamma}{8\det J_F}.
\end{aligned}$$

To change variables in expressions involving derivatives of the basis functions, note that the relevant differentiation operators are gradients for \mathcal{W}^0 , curls for \mathcal{W}^1 and divergence for \mathcal{W}^2 . The functions from \mathcal{W}^3 provide an approximation for $L^2(\Omega)$ and do not possess square integrable derivatives. To change variables in $\nabla W_{ijk}^{\alpha\beta\gamma}(\mathbf{x})$ we express this gradient as a sum of \mathcal{W}^1 functions (see (3.59)) and then use (3.49) to obtain

$$(3.65) \quad \left((\nabla W_{ijk}^{\alpha\beta\gamma}) \circ F \right)(\boldsymbol{\xi}) = \sum_{(ijk)_+} \sigma_k^\gamma \frac{\gamma}{2\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi})\hat{\phi}_j^\beta(\boldsymbol{\xi})(V_i \times V_j).$$

The curl of $W_{ij}^{\alpha\beta}$ is a linear combination of \mathcal{W}^2 functions. To change the variables we use (3.61) and (3.52) to find that

$$(3.66) \quad \left((\nabla \times W_{ij}^{\alpha\beta}) \circ F \right)(\boldsymbol{\xi}) = \frac{1}{4\det J_F} \left(\sigma_{jk}^{\beta\gamma} \left(\beta\gamma \hat{\phi}_i^\alpha(\boldsymbol{\xi})V_i \right) - \sigma_{ik}^{\alpha\gamma} \left(\alpha\gamma \hat{\phi}_j^\beta(\boldsymbol{\xi})V_j \right) \right).$$

Lastly, for the divergence of W_i^α on $\hat{\mathcal{K}}$ the change of variables gives

$$(3.67) \quad \left((\nabla \cdot W_i^\alpha) \circ F \right)(\boldsymbol{\xi}) = \frac{\alpha\beta\gamma}{8\det J_F}.$$

3.2.4 The van Welij complex in \mathbb{R}^2

In this section we develop the proper restriction of van Welij spaces to quadrilateral elements. We show that a relation similar to the exactness in three-dimensions still exists, albeit in a form that involves different combinations of spaces. The differences are caused by the existence of two curl operators in 2D and the fact that ‘‘faces’’ and ‘‘edges’’ of a quadrilateral coincide. The first two-dimensional curl is defined by embedding a scalar function into a 3D vector function and taking the curl of the former:

$$(3.68) \quad \nabla \times \phi := \nabla \times (\phi \mathbf{k}) = \nabla \phi \times \mathbf{k} = \phi_y \mathbf{i} - \phi_x \mathbf{j} + 0\mathbf{k}.$$

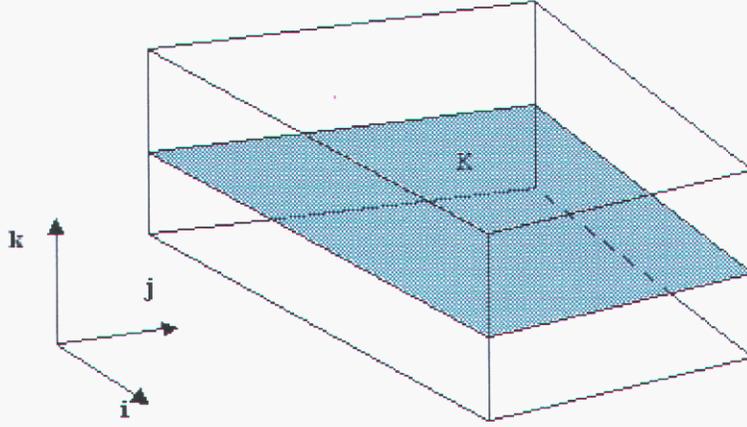


Figure 3.3. Quadrilateral \mathcal{K} and its virtual hexahedral.

The second curl operates on two-dimensional vector fields and is defined by

$$(3.69) \quad \nabla \times \mathbf{u} := \nabla \times (u_1 \mathbf{i} + u_2 \mathbf{j} + 0 \mathbf{k}) = (u_{2x} - u_{1y}) \mathbf{k}.$$

Dropping the last term in (3.68) and identifying the result in (3.69) with the coefficient of \mathbf{k} gives the definitions of the two curls:

$$\nabla \times \phi := \begin{pmatrix} \phi_y \\ -\phi_x \end{pmatrix} \quad \text{and} \quad \nabla \times \mathbf{u} := u_{2x} - u_{1y}.$$

A vector product between two-dimensional vectors can be defined along the same lines as

$$(3.70) \quad \mathbf{u} \times \mathbf{v} := (u_1 \mathbf{i} + u_2 \mathbf{j}) \times (v_1 \mathbf{i} + v_2 \mathbf{j}) = \det \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \\ \mathbf{v}_1 & \mathbf{v}_2 \end{pmatrix} \mathbf{k}$$

Given a quadrilateral element \mathcal{K} and a reference element $\hat{\mathcal{K}} = [-1, 1]^2$, we proceed to embed \mathcal{K} into the three-dimensional *virtual* hexahedral

$$\tilde{\mathcal{K}} = \{\mathbf{x} | (x_1, x_2) \in \mathcal{K}, -1 \leq x_3 \leq 1\},$$

(see Fig. 3.3). The mapping F between the virtual hexahedral $\tilde{\mathcal{K}}$ and the reference element in 3D has a third coordinate function given by $x_3 = F_3(\boldsymbol{\xi}) = \xi_3$ so that $G_3(\mathbf{x}) = x_3$. The vertical edges of $\tilde{\mathcal{K}}$ are described by $\phi_i^\alpha \phi_j^\beta = 1$, $i, j = 1, 2$. We will call these edges *virtual*. In terms of notations introduced earlier in §3.2.3 and definition (3.70)

$$V_3 = \nabla G_3 = \mathbf{k}, \quad \det J_F = V_1 \times V_2.$$

As a result, (3.33) specialize to

$$(3.71) \quad \nabla G_1 = (V_2 \times \mathbf{k}) / \det J_F \quad V_1 = (\nabla G_2 \times \mathbf{k}) \det J_F$$

$$(3.72) \quad \nabla G_2 = (\mathbf{k} \times V_1) / \det J_F \quad V_2 = (\mathbf{k} \times \nabla G_1) \det J_F.$$

Alternatively, we can write (3.71)-(3.72) in compact form as

$$(3.73) \quad \nabla G_i = (-1)^j V_j \times \mathbf{k} \quad \text{and} \quad V_i = (-1)^j \nabla G_j \times \mathbf{k}.$$

A nodal basis on the reference quadrilateral can be defined by restricting the argument of $\hat{N}^{\alpha\beta\gamma}$ in (3.19) to $\xi(1)$. Then $\hat{\phi}_3^2(\xi_1, \xi_2, 1) = 1$ and

$$(3.74) \quad \hat{N}^{\alpha\beta*}(\xi_1, \xi_2) := \hat{N}^{\alpha\beta\gamma}(\xi_1, \xi_2, 1) = \hat{\phi}_1^\alpha(\xi_1, \xi_2) \hat{\phi}_2^\beta(\xi_1, \xi_2).$$

The third coordinate can be dropped because the two directional functions in (3.74) are independent of ξ_3 . In what follows we will use ξ and \mathbf{x} to denote points in both two and three-dimensions, with the actual dimension being dependent on the context.

We now proceed to develop the proper restrictions of the van Welij complex in two-dimensions. This can be accomplished by considering either the top or the bottom faces of the virtual element. Here we choose to work with the top face $\phi_3^+ = 1$.

The nodal space \mathcal{W}^0 in 2D. Restriction of the nodal space $\mathcal{W}^0(\mathcal{K})$ is straightforward and is accomplished by setting $\mathbf{x} = (x_1, x_2, 1)$ in (3.45). The two dimensional basis set is

$$(3.75) \quad \begin{aligned} W_{ij*}^{\alpha\beta*}(\mathbf{x}) &:= W_{ijk}^{\alpha\beta\gamma}((x_1, x_2, 1)) \\ &= \phi_i^\alpha((x_1, x_2, 1)) \phi_j^\beta((x_1, x_2, 1)) = \phi_i^\alpha(\mathbf{x}) \phi_j^\beta(\mathbf{x}); \end{aligned}$$

for $i, j = 1, 2$ and $\mathbf{x} \in K$.

The edge space \mathcal{W}^1 in 2D. To define the restriction of the edge element space $\mathcal{W}^1(\mathcal{K})$ we consider the four edges of the top face $\phi_3^+ = 1$. Let us fix the edge \mathcal{E}_{23}^{++} . Note that this edge is not virtual and is parallel to one of the sides of the quadrilateral \mathcal{K} . The basis function for \mathcal{E}_{23}^{++} is

$$W_{23}^{++} = \phi_2^+ \phi_3^+ \nabla \phi_1^\alpha.$$

Choosing $\mathbf{x} = (x_1, x_2, 1)$ sets ϕ_3^+ equal to 1 and gives the restriction of W_{23}^{++} :

$$W_{2*}^{++}(\mathbf{x}) := W_{23}^{++}(x_1, x_2, 1) = \phi_2^+(\mathbf{x}) \nabla \phi_1^\alpha(\mathbf{x}) \quad \mathbf{x} \in \mathcal{K}.$$

Thus, in two-dimensions there are four ‘‘edge’’ basis functions and

$$(3.76) \quad \mathcal{W}^1(\mathcal{K}) = \text{span} \{W_{i*}^{\alpha*}(\mathbf{x})\}, \quad W_{i*}^{\alpha*} = \phi_i^\alpha \nabla \phi_j^\beta = \frac{\beta}{2} \phi_i^\alpha \nabla G_j; \quad i \neq j.$$

The face space \mathcal{W}^2 in 2D. To find the proper restriction of \mathcal{W}^2 consider the four faces perpendicular to the plane $x_3 = 0$ where the original quadrilateral is located. One of these faces is \mathcal{F}_1^+ and its associated basis function is

$$W_1^+ = \phi_1^+ \left(\nabla \phi_2^\beta \times \nabla \phi_3^+ \right) = (-1)^2 \phi_1^+ \left(\nabla \phi_2^\beta \times \mathbf{k}/2 \right)$$

The basis function for the adjacent face \mathcal{F}_2^+ is

$$W_2^+ = \phi_2^+ \left(\nabla \phi_3^+ \times \nabla \phi_1^\alpha \right) = (-1)^1 \phi_2^+ \left(\nabla \phi_1^\alpha \times \mathbf{k}/2 \right)$$

Setting $\mathbf{x} = (x_1, x_2, 1)$ restricts W_1^+ and W_2^+ to planar vector fields. However, both fields are normalized by area, while the proper normalization for a planar restriction would be by the length of the quadrilateral side. Thus, we multiply the planar vector fields by 2 (the length of the virtual edge) and use the fact that $\phi_3^+ = \mathbf{k}/2$ to find the proper restriction:

$$\mathcal{W}^2(\mathcal{K}) = \text{span} \{W_i^\alpha\},$$

$$(3.77) \quad W_i^\alpha = (-1)^j \phi_i^\alpha \left(\nabla \phi_j^\beta \times \mathbf{k} \right) = \frac{(-1)^j \beta}{2} \phi_i^\alpha \left(\nabla G_j \times \mathbf{k} \right); \quad i \neq j.$$

The density space \mathcal{W}^3 . To find the restriction of the basis function in (3.53) we set $\mathbf{x} = (x_1, x_2, 1)$ and use the fact that $\nabla G_3 = \mathbf{k}$. This gives

$$W_{\mathcal{K}}(x_1, x_2, 1) = \left(\nabla \phi_1^\alpha(x_1, x_2, 1) \times \nabla \phi_2^\beta(x_1, x_2, 1) \right) \cdot \mathbf{k}/2.$$

The right hand side in this equation is precisely the definition of the two-dimensional vector product (3.70). After normalizing by the length of the virtual edge,

$$(3.78) \quad W_{\mathcal{K}}(\mathbf{x}) = \nabla \phi_1^\alpha(\mathbf{x}) \times \nabla \phi_2^\beta(\mathbf{x}) = \frac{\alpha\beta}{4} \nabla G_1 \times \nabla G_2 \quad \mathbf{x} \in \mathcal{K}.$$

Since the vector product (3.70) is identified with a constant, this function is consistent with its three-dimensional prototype. Using (3.71)-(3.72) we find that

$$(W_{\mathcal{K}} \circ F)(\boldsymbol{\xi}) = \frac{\alpha\beta(V_2 \times \mathbf{k}) \times (\mathbf{k} \times V_1)}{4\det J_F^2} = \frac{\alpha\beta(V_1 \times V_2)}{4\det J_F^2} = \frac{\alpha\beta}{4\det J_F}.$$

The last identity follows from $\det J_F = V_1 \times V_2$, and gives the local reference element prototype of $W_{\mathcal{K}}$.

Local reference bases. To find the prototypes of the two-dimensional restrictions on the reference square $\hat{\mathcal{K}}$, we can develop formulas analogous to (3.49)-(3.52). Using (3.71)-(3.72), or (3.73) to change the variables gives

$$(3.79) \quad (W_{ij^*}^{\alpha\beta*} \circ F)(\boldsymbol{\xi}) = \hat{W}_{ij^*}^{\alpha\beta*}(\boldsymbol{\xi}) = \hat{\phi}_i^\alpha(\boldsymbol{\xi})\hat{\phi}_j^\beta(\boldsymbol{\xi})$$

$$(3.80) \quad (W_{i^*}^{\alpha*} \circ F)(\boldsymbol{\xi}) = \hat{W}_{i^*}^{\alpha*}(\boldsymbol{\xi}) = \frac{(-1)^{i\beta}}{2\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi})(V_i \times \mathbf{k})$$

$$(3.81) \quad (W_i^\alpha \circ F)(\boldsymbol{\xi}) = \hat{W}_i^\alpha(\boldsymbol{\xi}) = \frac{\beta}{2\det J_F} \hat{\phi}_i^\alpha(\boldsymbol{\xi})V_i$$

$$(3.82) \quad (W_{\mathcal{K}} \circ F)(\boldsymbol{\xi}) = \hat{W}_{\mathcal{K}}(\boldsymbol{\xi}) = \frac{\alpha\beta}{4\det J_F}.$$

Exactness in two-dimensions. The spaces defined in this section manage so far to recover the essential properties of their three-dimensional prototypes. For example, W^0 , W^1 and W^2 are $H(\Omega, \mathbf{grad})$, $H(\Omega, \mathbf{curl})$ and $H(\Omega, \mathbf{div})$ conforming spaces, respectively. However, at first glance it seems that the critical exactness property has been lost. We still have that

$$\nabla \cdot (\phi_i^\alpha(\mathbf{x})\phi_j^\beta(\mathbf{x})) = \phi_i^\alpha(\mathbf{x})\nabla \cdot \phi_j^\beta(\mathbf{x}) + \nabla \phi_i^\alpha(\mathbf{x})\phi_j^\beta(\mathbf{x})$$

and

$$\nabla \cdot \left(\frac{(-1)^{j\beta}}{2} \phi_i^\alpha(\nabla G_j \times \mathbf{k}) \right) = \frac{(-1)^{j\alpha\beta}}{4} \nabla G_i \times \nabla G_j$$

which verify the inclusions

$$\nabla W^0 \subset W^1 \quad \text{and} \quad \nabla \cdot W^2 \subset W^3$$

However, neither one of the two curl operators can possibly verify the inclusion $\nabla \times W^1 \subset W^2$. The curl in (3.69) gives a scalar function, while the curl in (3.68) cannot even be applied to functions in W^1 .

This confusing situation can be avoided if the curl relation is sought by restriction of the existing relations in three-dimensions, rather than directly. Since the curl seems to be the culprit here we take as a starting point the edge element space W^1 on the virtual hexahedral $\tilde{\mathcal{K}}$. The edges on this element can be divided into two sets: the four edges on the top face that are parallel with the edges of the quadrilateral \mathcal{K} :

$$\tilde{\mathcal{F}}_{13}^{-+}, \quad \tilde{\mathcal{F}}_{23}^{-+}, \quad \tilde{\mathcal{F}}_{13}^{++}, \quad \tilde{\mathcal{F}}_{23}^{++}.$$

and the four virtual edges passing through the vertices of \mathcal{K} :

$$\tilde{\mathcal{F}}_{12}^{--}, \quad \tilde{\mathcal{F}}_{12}^{-+}, \quad \tilde{\mathcal{F}}_{12}^{+-}, \quad \tilde{\mathcal{F}}_{12}^{++};$$

The basis functions for the first set are

$$W_{23}^{\alpha+} = \phi_i^\alpha \phi_3^+ \nabla \phi_j^\beta, \quad i, j = 1 \quad \text{or} \quad 2.$$

These functions were used to define the two-dimensional edge element. The basis functions for the four virtual edges are

$$W_{ij}^{\alpha\beta} = \left(\phi_i^\alpha \phi_j^\beta \right) \nabla \phi_3^+, \quad i, j = 1 \quad \text{or} \quad 2.$$

These functions were not used in the definition of $\mathcal{W}^1(\mathcal{K})$, however they will be handy now. Let us compute their curls. Since $\nabla \phi_3^+ = \nabla G_3/2 = \mathbf{k}/2$,

$$W_{ij}^{\alpha\beta} = \frac{1}{2} \left(\phi_i^\alpha \phi_j^\beta \right) \mathbf{k} = \frac{1}{2} (W_{ij^*}^{\alpha\beta^*}) \mathbf{k}, \quad i, j = 1, 2$$

where $W_{ij^*}^{\alpha\beta^*}$ are the 2D nodal basis functions from (3.75). As a result,

$$\begin{aligned} \nabla \times W_{ij}^{\alpha\beta} &= \frac{1}{2} \nabla \times \left(W_{ij^*}^{\alpha\beta^*} \mathbf{k} \right) \\ &= \frac{1}{2} \left(\phi_i^\alpha \left(\nabla \phi_j^\beta \times \mathbf{k} \right) + \phi_j^\beta \left(\nabla \phi_i^\alpha \times \mathbf{k} \right) \right) \end{aligned}$$

To interpret this relation in two dimensions, note that $\nabla \times (W_{ij^*}^{\alpha\beta^*} \mathbf{k})$ coincides with the definition (3.68) of the first curl operator in 2D applied to the scalar function $W_{ij^*}^{\alpha\beta^*} \mathbf{k}$, while from (3.77) it is clear that

$$\phi_i^\alpha \left(\nabla \phi_j^\beta \times \mathbf{k} \right) = (-1)^j W_i^\alpha \quad \text{and} \quad \phi_j^\beta \left(\nabla \phi_i^\alpha \times \mathbf{k} \right) = (-1)^i W_j^\beta.$$

Therefore, the curls of virtual edge basis lead to a relation between the nodal space \mathcal{W}^0 and the face space \mathcal{W}^2 :

$$(3.83) \quad \nabla \times W_{ij^*}^{\alpha\beta^*} = \pm \frac{1}{2} (W_i^\alpha - W_j^\beta); \quad i \neq j.$$

Let us now compute the curls of the vector fields representing the basis functions of the first set restricted to the top face. On this face $\phi_3^+ = 1$ so that (3.61) gives

$$\nabla \times \left[W_{23}^{\alpha+} \right]_{\phi_3^+=1} = \nabla \times \left(\phi_i^\alpha \nabla \phi_j^\beta \right) = \nabla \phi_i^\alpha \times \nabla \phi_j^\beta.$$

The term on the right hand side is the definition of the 2D density function (3.78). As far as $\nabla \times \left(\phi_i^\alpha \nabla \phi_j^\beta \right)$ is concerned, from (3.69) and (3.76) we can conclude that this expression

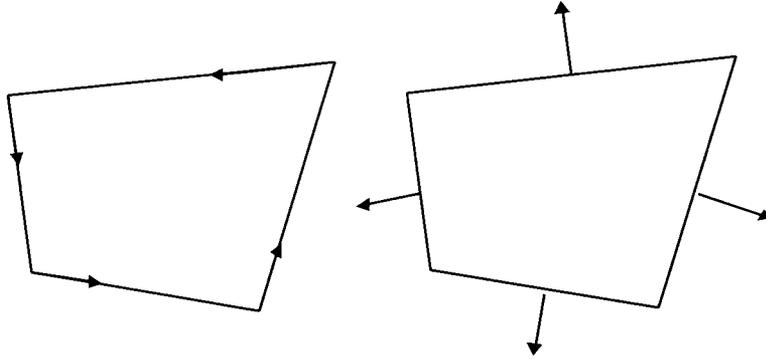


Figure 3.4. Orientation choice for edge and face elements on \mathcal{K} .

represents curl of an edge basis function in 2D. As a result, curls of the parallel edges lead to a relation between the edge space \mathcal{W}^1 and the density space \mathcal{W}^3 according to:

$$(3.84) \quad \nabla \times W_{i*}^{\alpha*} = W_{\mathcal{K}}.$$

Therefore, in 2D the exactness still exists, but in a modified form with two separate curl relations which “skip” a space. This phenomena can be explained by realizing that the space $\mathcal{W}^0(\mathcal{K})$ in actuality represents restrictions of two different three-dimensional spaces. First, it is a restriction of the nodal space $\mathcal{W}^0(\tilde{\mathcal{K}})$ which leads to the relation $\nabla \mathcal{W}^0 \subset \mathcal{W}^1$. However, we can also consider $\mathcal{W}^0(\mathcal{K})$ as representing restrictions of the edge basis functions for the four virtual edges. This leads to the first of the two curl relations in two dimensions!

Orientation of $\mathcal{W}^1(\mathcal{K})$ and $\mathcal{W}^2(\mathcal{K})$. The vector fields defined by (3.76) and (3.77) involve an implicit choice of orientation by virtue of the index β . Orientation is important when several elements are combined together and one has to match the vector fields on the shared edges and faces. For the edge elements (3.77) we adopt an orientation in which the four edges \mathcal{E}_i^α are traversed in a counterclockwise direction, and for the face elements (3.77) we adopt orientation in the direction of the outward normal to the face \mathcal{F}_i^α ; see Fig. 3.4. The sets of oriented edge and face basis functions on \mathcal{K} is summarized in Table 3.2. In what follows we will refer to this choice as the *standard edge and face element orientation* on a quadrilateral.

3.2.5 The van Welij complex on hexahedral and quadrilateral lattices

In this section we show how edge and face elements on individual elements can be combined to form finite element spaces on hexahedral or rectangular triangulations. We also show that the resulting edge element vector field is tangentially continuous, while the face elements lead to a normally continuous field. The case of quadrilateral lattices is considered first to outline

Table 3.2. Oriented edge and face elements on tetrahedron.

Edge/face	Edge vector field	Face vector field
bottom	$W_{\mathcal{E}}^b = -\frac{1}{2}\phi_2^- \nabla G_1$	$W_{\mathcal{F}}^b = \frac{1}{2}\phi_2^- \nabla G_1 \times \mathbf{k}$
right	$W_{\mathcal{E}}^r = \frac{1}{2}\phi_1^+ \nabla G_2$	$W_{\mathcal{F}}^r = \frac{1}{2}\phi_1^+ \nabla G_2 \times \mathbf{k}$
top	$W_{\mathcal{E}}^t = \frac{1}{2}\phi_2^+ \nabla G_1$	$W_{\mathcal{F}}^t = -\frac{1}{2}\phi_2^+ \nabla G_1 \times \mathbf{k}$
left	$W_{\mathcal{E}}^l = \frac{1}{2}\phi_1^- \nabla G_2$	$W_{\mathcal{F}}^l = -\frac{1}{2}\phi_1^- \nabla G_2 \times \mathbf{k}$

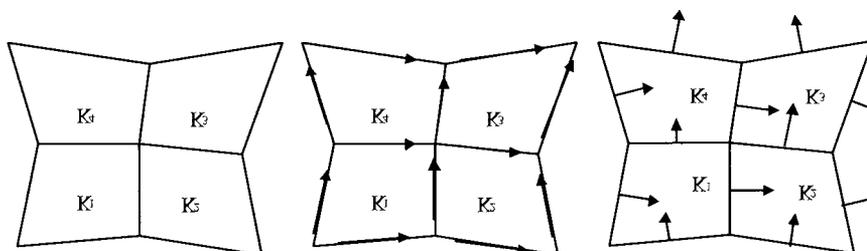


Figure 3.5. Standard \mathcal{T}_h and the sets $\vec{\mathcal{E}}, \vec{\mathcal{F}}$ for an oriented triangulation $\vec{\mathcal{T}}_h$.

the basic ideas in a relatively simpler setting. Then we proceed to define the finite elements on hexahedral lattices. For the nodal space \mathcal{W}^0 and the space \mathcal{W}^3 forming the finite element spaces is standard and here we do not take time to discuss their construction.

The complex on quadrilateral lattice To illustrate the process of joining together edge and face vector fields on several elements it suffices to consider a simple triangulation \mathcal{T}_h consisting of four quadrilaterals $\mathcal{K}_i, i = 1, \dots, 4$; see Fig. 3.5. On each quadrilateral there are the edge vector fields $\mathcal{W}^1(\mathcal{K}_i)$ and the face vector fields $\mathcal{W}^2(\mathcal{K}_i)$ which are assumed to have the standard orientation; see Fig. 3.4. To match the vector fields defined on elements \mathcal{K}_i it is necessary to introduce orientation for the edges \mathcal{E} and the faces \mathcal{F} of \mathcal{T}_h . In two-dimensions edges and faces coincide with the sides of the quadrilaterals. Therefore, edge orientation is understood as choosing an ordering of the side's endpoints, i.e., a direction in which to traverse the edge. Face orientation, in contrast, is understood as choosing a normal direction to the side of the quadrilateral. This gives rise to an *oriented triangulation* $\vec{\mathcal{T}}_h$ with edges and faces denoted by $\vec{\mathcal{E}}$ and $\vec{\mathcal{F}}$, respectively. An example of $\vec{\mathcal{T}}_h$ and a standard triangulation \mathcal{T}_h

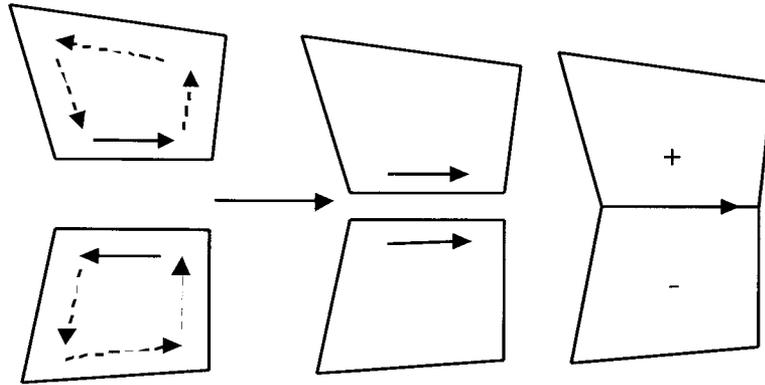


Figure 3.6. Formation of an edge basis function from local vector fields.

is shown on Fig. 3.5. The orientation on $\vec{\mathcal{T}}_h$ is *global* and should not be confused with the standard element orientation of Fig.3.4 which is defined *locally*.

The edge element space. To define $\mathcal{W}^1(\mathcal{T}_h)$ consider the set $\vec{\mathcal{E}}$ of all edges in $\vec{\mathcal{T}}_h$. Each edge is associated with a degree of freedom⁴ and a basis function $W_{\vec{\mathcal{E}}_i}$ such that

$$(3.85) \quad \int_{\vec{\mathcal{E}}_j} W_{\vec{\mathcal{E}}_i} dl = \delta_{ij}.$$

It is clear that the support of $W_{\vec{\mathcal{E}}_i}$ consists of all elements that share the edge $\vec{\mathcal{E}}_i$; in two dimensions there are at most two such quadrilaterals. Let us exhibit a basis function for the edge $\vec{\mathcal{E}}_s = \mathcal{K}_1 \cap \mathcal{K}_4$ shared by \mathcal{K}_1 and \mathcal{K}_4 . The locally oriented vector field on \mathcal{K}_4 associated with this edge is $W_{\vec{\mathcal{E}}_s}^b$; for \mathcal{K}_1 this field is represented by $W_{\vec{\mathcal{E}}_s}^t$. Because these fields follow the local orientation, along $\vec{\mathcal{E}}_s$ they run in opposite directions; see Fig. 3.6. To combine the two local fields into one, tangentially continuous field defined on $\mathcal{K}_1 \cup \mathcal{K}_4$, they must be oriented according to the orientation on $\vec{\mathcal{E}}_s$. The vector field on \mathcal{K}_4 is already aligned with the edge. To align the field on \mathcal{K}_1 the appropriate local definition of the vector function is multiplied by -1 . This process is illustrated on Fig. 3.6. The resulting edge basis function is

$$(3.86) \quad W_{\vec{\mathcal{E}}_s} = \begin{cases} W_{\vec{\mathcal{E}}_s}^b & \text{for } \mathbf{x} \in \mathcal{K}_4 \\ -W_{\vec{\mathcal{E}}_s}^t & \text{for } \mathbf{x} \in \mathcal{K}_1 \end{cases}.$$

⁴When boundary conditions are added, some degrees of freedom will be removed; we discuss boundary conditions later.

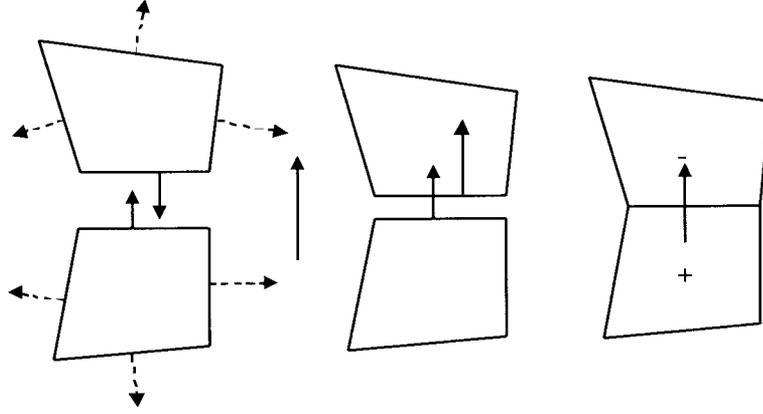


Figure 3.7. Formation of a face basis function from local vector fields.

The edge element space $W^1(\vec{\mathcal{T}}_h)$ is now completely defined by

$$(3.87) \quad W^1(\vec{\mathcal{T}}_h) = \text{span} \{W_{\vec{\mathcal{E}}_s} \mid \vec{\mathcal{E}}_s \in \vec{\mathcal{E}}\}.$$

Clearly, $\dim W^1(\vec{\mathcal{T}}_h) = \dim \vec{\mathcal{E}}$ and

$$(3.88) \quad W^1(\vec{\mathcal{T}}_h) \subset H(\Omega, \text{curl}).$$

The face element space. To define $W^2(\vec{\mathcal{F}})$ we proceed in a similar fashion and consider the set $\vec{\mathcal{F}}$ of all oriented faces in $\vec{\mathcal{T}}_h$. Each face has a basis function $W_{\vec{\mathcal{F}}_i}$ such that

$$\int_{\vec{\mathcal{F}}_j} W_{\vec{\mathcal{F}}_i} \cdot \mathbf{n} dS = \delta_{ij}.$$

Support of the basis function contains all elements that share the face; in two dimensions there are at most two such quadrilaterals.

Let us develop a basis function for the face $\vec{\mathcal{F}}_s = \mathcal{K}_1 \cap \mathcal{K}_4$ shared by \mathcal{K}_1 and \mathcal{K}_4 . The locally oriented vector fields on \mathcal{K}_1 and \mathcal{K}_4 associated with this edge are $W_{\vec{\mathcal{F}}}^t$ and $W_{\vec{\mathcal{F}}}^b$, respectively. The normal components of these fields across $\vec{\mathcal{F}}_s$ run in opposite directions; see Fig. 3.7. The field on \mathcal{K}_1 is aligned with the orientation on the face $\vec{\mathcal{E}}_s$; thus now it is necessary to multiply by -1 the field on \mathcal{K}_4 ; see Fig. 3.7. The resulting basis function on $\mathcal{K}_1 \cup \mathcal{K}_4$ is

$$(3.89) \quad W_{\vec{\mathcal{F}}_s} = \begin{cases} -W_{\vec{\mathcal{F}}}^b & \text{for } \mathbf{x} \in \mathcal{K}_4 \\ W_{\vec{\mathcal{F}}}^t & \text{for } \mathbf{x} \in \mathcal{K}_1 \end{cases}.$$

Basis functions for all remaining faces can be defined in a similar manner. The face element space $W^2(\vec{\mathcal{T}}_h)$ is now completely determined by

$$(3.90) \quad W^2(\vec{\mathcal{T}}_h) = \text{span} \{W_{\vec{\mathcal{F}}_s}^- \mid \vec{\mathcal{F}}_s \in \vec{\mathcal{F}}\}.$$

Dimension of $W^2(\vec{\mathcal{F}})$ equals the number of faces and

$$(3.91) \quad W^2(\vec{\mathcal{T}}_h) \subset H(\Omega, \text{div}).$$

Exactness on tetrahedral lattices. For one element, the exactness relation (3.83) follows by a direct differentiation of the nodal basis function. To develop analogous relation for a general tetrahedral lattice we assume that the edges and the faces of the lattice are endowed with orientation and form an oriented triangulation $\vec{\mathcal{T}}_h$. Let \mathcal{N}_s denote a node in $\vec{\mathcal{T}}_h$. For simplicity, assume that \mathcal{N}_s is shared by the four elements K_i as in Fig. 3.5. These four elements form the support of the nodal basis function $W_{\mathcal{N}_s}$ associated with \mathcal{N}_s . The four faces emanating from \mathcal{N}_s are

$$\begin{aligned} \vec{\mathcal{F}}_S &= \mathcal{K}_1 \cap \mathcal{K}_2 \\ \vec{\mathcal{F}}_E &= \mathcal{K}_2 \cap \mathcal{K}_3 \\ \vec{\mathcal{F}}_N &= \mathcal{K}_3 \cap \mathcal{K}_4 \\ \vec{\mathcal{F}}_W &= \mathcal{K}_4 \cap \mathcal{K}_1 \end{aligned}$$

Using the same arguments as for (3.89) the basis functions for each face can be expressed in terms of the local vector fields as

$$\begin{aligned} W_{\vec{\mathcal{F}}_S}^- &= \begin{cases} W_{\vec{\mathcal{F}}}^r & \text{for } \mathbf{x} \in \mathcal{K}_1 \\ -W_{\vec{\mathcal{F}}}^l & \text{for } \mathbf{x} \in \mathcal{K}_2 \end{cases} \\ W_{\vec{\mathcal{F}}_E}^- &= \begin{cases} W_{\vec{\mathcal{F}}}^b & \text{for } \mathbf{x} \in \mathcal{K}_2 \\ -W_{\vec{\mathcal{F}}}^t & \text{for } \mathbf{x} \in \mathcal{K}_3 \end{cases} \\ W_{\vec{\mathcal{F}}_N}^- &= \begin{cases} W_{\vec{\mathcal{F}}}^r & \text{for } \mathbf{x} \in \mathcal{K}_4 \\ -W_{\vec{\mathcal{F}}}^l & \text{for } \mathbf{x} \in \mathcal{K}_3 \end{cases} \\ W_{\vec{\mathcal{F}}_W}^- &= \begin{cases} W_{\vec{\mathcal{F}}}^t & \text{for } \mathbf{x} \in \mathcal{K}_1 \\ -W_{\vec{\mathcal{F}}}^b & \text{for } \mathbf{x} \in \mathcal{K}_4 \end{cases} \end{aligned}$$

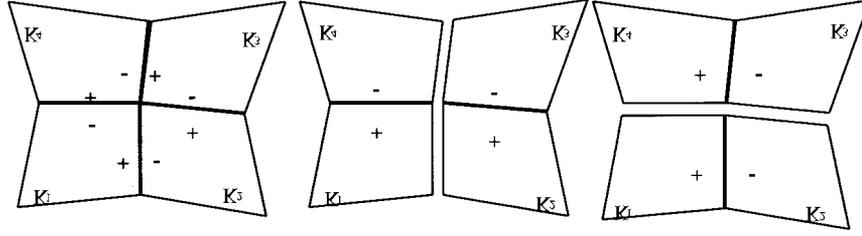


Figure 3.8. Local representations of $\nabla \times W_{\mathcal{N}_s}$ and face basis functions.

Using (3.83) on each quadrilateral gives four local relations between the local face element vector field and the curl of the local nodal basis function:

$$\begin{aligned}\nabla \times W_{\mathcal{N}_s} &= W_{\mathcal{F}}^r - W_{\mathcal{F}}^t && \text{on } \mathcal{K}_1 \\ \nabla \times W_{\mathcal{N}_s} &= W_{\mathcal{F}}^t - W_{\mathcal{F}}^l && \text{on } \mathcal{K}_2 \\ \nabla \times W_{\mathcal{N}_s} &= W_{\mathcal{F}}^l - W_{\mathcal{F}}^b && \text{on } \mathcal{K}_3 \\ \nabla \times W_{\mathcal{N}_s} &= W_{\mathcal{F}}^b - W_{\mathcal{F}}^r && \text{on } \mathcal{K}_4\end{aligned}$$

Comparing the local expressions for the curl of $W_{\mathcal{N}_s}$ with the local expressions for the face element basis functions we obtain the exactness relation (see Fig. 3.8)

$$(3.92) \quad \nabla \times W_{\mathcal{N}_s} = W_{\mathcal{F}_S}^{\rightarrow} - W_{\mathcal{F}_N}^{\rightarrow} + W_{\mathcal{F}_E}^{\rightarrow} - W_{\mathcal{F}_W}^{\rightarrow}$$

3.3 Example: solution of eddy current equations using face and edge elements.

We consider the region from Section 3.1 with boundary $\partial\Omega = \Gamma \cup \Gamma^*$. The governing equations for the electromagnetic field in Ω are given by

$$(3.93) \quad \nabla \times \mathbf{H} = \mathbf{J} \quad \text{in } \Omega$$

$$(3.94) \quad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad \text{in } \Omega$$

$$(3.95) \quad \nabla \cdot \mathbf{B} = 0 \quad \text{in } \Omega$$

$$(3.96) \quad \nabla \cdot \mathbf{J} = 0 \quad \text{in } \Omega$$

$$(3.97) \quad \mathbf{B} = \mu \mathbf{H}$$

$$(3.98) \quad \mathbf{J} = \sigma \mathbf{E}$$

Initial values of the magnetic flux density \mathbf{B} are required to satisfy 3.95. System (3.93)-(3.98) must be closed by choosing appropriate boundary conditions. Here we consider

$$(3.99) \quad \mathbf{n} \times \mathbf{H}_b = 0 \quad \text{and} \quad \mathbf{n} \cdot \mathbf{E}_b = 0 \quad \text{on } \Gamma$$

$$(3.100) \quad \mathbf{n} \cdot \mathbf{B}_b = 0 \quad \text{and} \quad \mathbf{n} \times \mathbf{E}_b = 0 \quad \text{on } \Gamma^*$$

To make the best use of the van Welij complex we must choose wisely the two primary fields that will be approximated. This choice is governed by the exactness property of the finite element complex, i.e., the inclusion $\nabla \times (\mathcal{W}^1) \subset \mathcal{W}^2$. This inclusion and (3.94) suggest the use of edge elements to approximate the electric field \mathbf{E} and face elements to represent the magnetic flux density \mathbf{B} . The other two variables can be eliminated from (3.93) with the help of (3.97)-(3.98). The resulting system in terms of \mathbf{E} and \mathbf{B} is

$$(3.101) \quad \nabla \times \frac{1}{\mu} \mathbf{B} = \sigma \mathbf{E} \quad \text{in } \Omega$$

$$(3.102) \quad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad \text{in } \Omega$$

$$(3.103)$$

To discretize (3.101)-(3.102) we assume the representations

$$(3.104) \quad \mathbf{B}_h = \sum_{\vec{f}} \Phi_{\vec{f}} W_{\vec{f}};$$

$$(3.105) \quad \mathbf{E}_h = \sum_{\vec{\varepsilon}} C_{\vec{\varepsilon}} W_{\vec{\varepsilon}},$$

This choice makes it possible to satisfy exactly (after the time derivative is approximated by a finite difference) Eq. (3.102). However, the finite element representations (3.104)-(3.105) cannot be substituted directly into the first equation because face elements do not have well-defined curls. Thus, we proceed to rewrite (3.101) in weak form by integrating by parts the identity

$$\int_{\Omega} \hat{\mathbf{E}} \cdot \nabla \times \frac{\mathbf{B}}{\mu} d\Omega = \int_{\Omega} \sigma \mathbf{E} \cdot \hat{\mathbf{E}} d\Omega \quad \forall \hat{\mathbf{E}} \in H(\Omega, \text{curl}),$$

The ensuing variational equation

$$(3.106) \quad \int_{\Omega} \frac{1}{\mu} \mathbf{B} \cdot \nabla \times \hat{\mathbf{E}} d\Omega - \int_{\Gamma} (\mathbf{H}_b \times \mathbf{n}) \cdot \hat{\mathbf{E}} d\Gamma = \int_{\Omega} \sigma \mathbf{E} \cdot \hat{\mathbf{E}} d\Omega \quad \forall \hat{\mathbf{E}},$$

where \mathbf{H}_b is the (given) magnetic field on the boundary, can now be used in conjunction with the representations (3.104)-(3.105).

Let us now develop the fully discrete approximation of the Maxwell's equations (3.101)-(3.102). We replace the time derivative in (3.102) by a difference and insert the finite element approximations of \mathbf{E} and \mathbf{B} into the resulting equation and (3.106). This gives the algebraic system

$$(3.107) \int_{\Omega} \frac{1}{\mu} \mathbf{B}_h^{n+1} \cdot \nabla \times \hat{\mathbf{E}}_h d\Omega - \int_{\Gamma} (\mathbf{H}_b \times \mathbf{n}) \cdot \hat{\mathbf{E}}_h d\Gamma = \int_{\Omega} \sigma \mathbf{E}_h^{n+1} \cdot \hat{\mathbf{E}}_h d\Omega \quad \forall \hat{\mathbf{E}}_h \in \mathcal{W}^1$$

$$(3.108) \quad -\frac{\mathbf{B}_h^{n+1} - \mathbf{B}_h^n}{\Delta t} = \nabla \times \mathbf{E}_h^{n+1}$$

Because the curl of \mathbf{E}_h^{n+1} belongs to \mathcal{W}^2 the second equation can be solved exactly. Substituting

$$\mathbf{B}_h^{n+1} = \mathbf{B}_h^n - \Delta t \nabla \times \mathbf{E}_h^{n+1}.$$

into (3.107) gives an algebraic equation only in terms of \mathbf{E}_h^{n+1} :

$$(3.109) \quad \int_{\Omega} \sigma \mathbf{E}_h^{n+1} \cdot \hat{\mathbf{E}}_h + \frac{\Delta t}{\mu} (\nabla \times \mathbf{E}_h^{n+1}) \cdot (\nabla \times \hat{\mathbf{E}}_h) d\Omega = \int_{\Omega} \frac{1}{\mu} \mathbf{B}_h^n \cdot (\nabla \times \hat{\mathbf{E}}_h) d\Omega - \int_{\Gamma} (\mathbf{H}_b \times \mathbf{n}) \cdot \hat{\mathbf{E}}_h d\Gamma \quad \forall \hat{\mathbf{E}}_h \in \mathcal{W}^1.$$

The resulting scheme has some very attractive computational properties. It ensures that the approximate magnetic flux density is divergenceless provided $\nabla \cdot \mathbf{B}_h^0 = 0$. This can be accomplished, for example, by defining $\mathbf{B}_h^0 = \nabla \times \mathbf{A}_h^0$, where \mathbf{A}_h^0 is a vector potential representation in the edge element space \mathcal{W}^1 .

For the edge elements considered here, tangential \mathbf{E} are essential boundary conditions and tangential \mathbf{H} are natural boundary conditions. The approach listed above has been described in the time domain electromagnetics context in [38]. A description of an hp-adaptive approach using $H(\Omega, \text{curl})$ conforming elements is given in [33].

4 Free-form surface design

Sharp interface inversion uses a reduced order model. The conductivity is approximated by a piecewise constant in bulk regions. A triangulation represents the two dimensional surface between regions. The objective of the inversion is to optimize the location of the interface. The optimization uses shape derivatives. The computational complexity is sensitive to the number of vertices in the interface. The problem is addressed in Computer Graphics by associated a curvilinear surfaces to a coarse triangulation, a technique called Free-form surface design

(FFSD) [35, 36]. Spline surfaces of order less than five are impossible for general unstructured surface triangulations.

The papers of FFSD are a decade old, due to the fact that the original developers left academia for the private sector. Our implementation is incomplete. We have not yet developed robust and efficient algorithms to generate local second order (discussed in section 4.3). Instead, we moved on (in section 5) to inversion methods based directly on the triangulation.

FFSD is an r -adaptive method; the vertices move to fair the associated surface. The objective includes surface the triangulation quality, measured in terms of the angles between element normals. By adding terms to the cost functional minimized during surfaced fairing, other tasks may be accomplished simultaneously, such as improving the mesh quality. In section 4.1 we propose that certain numerically stable algorithms be applied in the mesh optimization.

The sharp interface identification methods discussed in the next section are implemented using the GNU Triangulated Surface Library, GTS. This section describes the tools that would need to be developed to make the sharp interface tools, described in the next section, independent of GTS. That task would be simplified by using a package such as Sandia's UTILIB.

FFSD methods locally reconstruct the interface surface. The original applications of FFSD constrain the overhead cost per image much more severely than in sharp interface inversion. Improvement algorithms are presented in section 4.2.

Local shape reconstruction tools associate a tangent space and normal space to the surface, and also support some decimation (surface coarsening) [36]. The surface is invariant under deformations of the nodes along the tangent space. The tools also provide other information needed to associate a surface to the triangulation. Local and global interpolation are supported. Local interpolation determines the quadratic Bézier surface over an element. The data for the local problem is, at each vertex, a normal vector and an angle. The global interpolation tool is intended for use by the customer in visualizing this interface, and determines the global C^1 surface [3].

In section 4.2.2, an improved algorithm for local coordinate fitting is presented. Furthermore the local shape reconstruction tools support surface fairing — the motion of the nodes to reduce curvature as described in section 4.2.3. The local shape reconstruction tools support surface decimation based on small angles. A second, more powerful surface decimation tool based on pair (generalization of edge) contraction is reviewed in section 4.2.5.

Ideas from linear algebra are emphasized, along with some special notation. The symbol $M(n, m)$ denotes the set of n by m matrices with real coefficients. The subset of orthogonal matrices in $M(n, n)$ is denoted by $O(n)$, and the subset of matrices in $O(n)$ with determinant one is denoted by $SO(n)$.

4.1 Adaptive element quality

A model of element quality is developed for addition to the fairing objective function with improved numerical stability properties.

The quality of a simplicial finite element is twice the ratio of the in radius to the circumradius [11, 32]. The measure is popular in the analysis of finite element methods, but less popular in the meshing community. Formulas for element quality as a function of edge lengths have superior numerical stability properties, compared to formulas depending on vertex locations. As we shall now show, the nontrivial step is to compute the triangle surface area, S .



Figure 4.1. Annotated illustrations of poor quality obtuse (left) and acute (right) triangular finite elements are shown.

Note that the triangle circumradius is given by $\frac{L_1 L_2 L_3}{4S}$, where L_i is the i th edge length and S is the surface area (page seven of [17]). The half perimeter $p = \frac{L_1 + L_2 + L_3}{2}$ and the triangle in radius is $\frac{S}{p}$. Given these quantities, mesh quality is defined as a function of each element, $q(e)$. One may define a flow on the nodal positions that maximizes quality using spring forces to $dz = F(z)$ [32]. Another choice is $F = \nabla q(e)$. Suppose that edge k is shared by elements i and j , and has node l as an endpoint. On element i , global edge k is local edge m , and on element j , global edge k is local edge n . Then add to component l of dz

$$\frac{e_m^T \nabla q_i}{q_i} \frac{1}{2} + \frac{e_n^T \nabla q_j}{q_j} \frac{1}{2}.$$

A factor of $\frac{1}{2}$ is used to avoid unnecessary node drift. Saddle points may be escaped by adding small random perturbations. From now on, we view the element quality as a function of element edge lengths.

The gradient flow is optimal, but to understand how it improves upon the spring forces, consider ∇q in the two cases of asymptotically bad triangles: a severely obtuse triangle and a triangle that is not severely obtuse but is severely acute, (see Figure 4.1). The edges are labeled so that the edge lengths are non-increasing, $a \geq b \geq c$.

Formulas for element quality as a function of edge lengths have superior numerical stability properties, compared to formulas depending on vertex locations. As we shall now show, the nontrivial step is to compute the triangle surface area, S . The implementation involves Heron's formula for the area of a triangle in terms of its edge lengths, or more precisely in term of the facial differences, $q = ABC/(abc)$,

$$A = -a + b + c, \quad B = a - b + c, \quad C = a + b - c.$$

The numerically stable formula for computing facial differences is (see [24],

$$\begin{aligned} A &= (\max(b, c) - a) + \min(b, c), \\ B &= (\max(a, c) - b) + \min(a, c), \\ C &= (\max(a, b) - c) + \min(a, b). \end{aligned}$$

In the case of ordered edge lengths, $C \geq B \geq c \geq A > 0$.

In the obtuse case the result of the perturbation analysis is that to first order, $\nabla q^T = (-1, 1, 1)4/a$. The spring forces are purely repulsive, but The force $\partial_a q$ is contractive, and contractive forces are not used in [32]. In the acute case, to first order

$$\nabla q^T = (\cos \alpha, -\cos \alpha, (1 + \cos^2 \alpha)/2)4/a.$$

The gradient flow improves on the spring forces by, in addition, moving the element towards a right triangle, $\alpha \rightarrow \pi/2$.

Similar comments extend to tetrahedral mesh refinement. If a, b, c products of lengths of pairs of opposite edges, then the tetrahedron circumradius is

$$\frac{\sqrt{(a+b+c)(a+b-c)(a-b+c)(-a+b+c)}}{24V}.$$

(page 10 of [17]). And the tetrahedron in radius is

$$\frac{3V}{S_1 + S_2 + S_3 + S_4},$$

(page 11 of [17]), where S_i is the surface area of the i th face. Numerically stable formulas for S and V in terms of edge lengths are given in [24] and [25] respectively.

4.2 Local shape reconstruction

The FFSD algorithm of [36] that associates a smooth surface to the triangulation that minimizes the principal curvature. Once a fair surface is obtained, a C^1 piecewise polynomial interpolant is used. Although Nielson's implementation of a Clough-Tocher element is suggested in [36], we prefer the Bézier form representation of the Bell triangle [3]. Re-triangulation is discussed in the following subsections.

4.2.1 Vertex neighborhood parametrizations

In order to understand FFSD algorithms precisely enough to implement them, certain details must be spelled out in detail. For example, in order to define the surface through a vertex and its neighbors, an algorithm is needed to order the neighbors. Given an interior node with

coordinates \mathbf{p}_0 and n neighbors, order the neighbors as follows: The first neighbor, \mathbf{p}_1 , is arbitrary. For $j > 1$, choose \mathbf{p}_j such that the triplet $(\mathbf{p}_0, \mathbf{p}_{j-1}, \mathbf{p}_j)$ are the vertices of a triangle, until $j = n$.

Given a boundary node \mathbf{p}_0 , choose \mathbf{p}_1 that is also a boundary node. If $1 < j < n$, check that \mathbf{p}_j is not a boundary node.

In the following, $\mathbf{p}_{n+1} = \mathbf{p}_1$. For $1 \leq j \leq n$, define θ_j by $0 < \theta_j < \pi$ and $\cos(\theta_j) = (\mathbf{p}_j - \mathbf{p}_0)^T(\mathbf{p}_{j+1} - \mathbf{p}_0)$.

At an interior node, $\alpha = 2\pi / \sum \theta_j$, and at a boundary node $\alpha = \pi / \sum \theta_j$. A boundary node with only two neighbors is a vertex. Vertices are treated differently, by interpolation of the boundary curves.

Define $\theta_0 = 0$, $\mathbf{u}_0 = \mathbf{0}$, and for $1 \leq j \leq n$, and let $\phi_j = \sum_0^j \alpha \theta_i$ and

$$\mathbf{u}_j = \|\mathbf{p}_j - \mathbf{p}_0\|(\cos(\phi_j), \sin(\phi_j)).$$

4.2.2 Local coordinate fitting

An algebraic function $s(\mathbf{u})$ such that $s(\mathbf{u}_0) = \mathbf{p}_0$, is fit to the data in the least squares sense; for $1 \leq j \leq n$, $s(\mathbf{u}_j) \approx \mathbf{p}_j$. In [36], five different parametrizations are used to try to minimize the condition number of the fitting. Furthermore, each vertex is assumed to have at least five neighbors, $n \geq 5$. The two components of \mathbf{u} are denoted $\mathbf{u}^T = [u, v]$. Then, an improved fitting algorithm is presented.

The treatments for interior and boundary nodes differ. At an interior node, a fit of the form $\mathbf{b}(\mathbf{u})^T = [1, u, v, u^2/2, uv, v^2/2]$ is attempted. Define \mathbf{B} in $M(5, n)$ by

$$[\mathbf{b}(\mathbf{u}_1), \dots, \mathbf{b}(\mathbf{u}_n)] = \begin{bmatrix} \mathbf{e}^T \\ \mathbf{B} \end{bmatrix}, \quad \mathbf{e}^T = [1, \dots, 1].$$

Due to $\mathbf{b}(\mathbf{0}) = \mathbf{e}_1$, there exists a matrix \mathbf{C} in $M(3, 5)$ (to be determined next) such that $s(\mathbf{u}) = [\mathbf{p}_0, \mathbf{C}]\mathbf{b}(\mathbf{u})$. The pseudo-inverse of \mathbf{B} , namely $\mathbf{B}^+ = \mathbf{V}\Sigma^{-1}\mathbf{U}^T$, is defined in terms of the singular value decomposition $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^T$.

As long as \mathbf{B} has no small singular values, for $\mathbf{P} = [\mathbf{p}_1 - \mathbf{p}_0, \dots, \mathbf{p}_n - \mathbf{p}_0]$, the least squares solution is $\mathbf{C} = \mathbf{B}^+\mathbf{P}$. Otherwise, $\mathbf{b}(\mathbf{u})^T = [1, u, v, (u^2 + v^2)/2]$ is used to define \mathbf{B} in $M(3, 3)$ and the similar construction determines \mathbf{C} in $M(3, 3)$. Note that $n \geq 3$ at an interior node for any surface mesh. An improved algorithm is presented below.

At a non-vertex boundary node ($n \geq 3$), a fit to $\mathbf{b}(\mathbf{u})^T = [1, u, v, u^2/2]$ is attempted. Otherwise, the surface is fit to $\mathbf{b}(\mathbf{u})^T = [1, u, v]$.

Scaling the u 's so that the largest radius is one is recommended due to the following observations. While scaling the points (u 's) so that minimum radius is unity often significantly increases the condition number, we observe that scaling the points so that maximum radius is unity never significantly increases the condition number, and sometimes significantly decreases the condition number.

A more accurate and expensive way to compute the reduced basis involves a QR factorization of a submatrix of matrix of left singular vectors, \mathbf{U} . If one (or two) singular values are relatively small, construct \mathbf{Q} such that $[\mathbf{U}(3 : 5, 5), \mathbf{Q}]$ (or $[\mathbf{U}(3 : 5, 4 : 5), \mathbf{Q}]$) is in $O(3)$. For

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^T \end{bmatrix},$$

in $M(4, 5)$ (or in $M(3, 5)$), replace \mathbf{B} with \mathbf{PB} .

4.2.3 Surface fairing

Surface fairing refers to moving the nodes to minimize the curvature. A differential motion of the nodes, dp , in the tangent plane, $dp \cdot n = 0$, does not change the surface. Fairing a surface changes the surface, $dp \cdot n \neq 0$. Node motions in the tangent plane are useful for optimizing element quality. In surface fairing some nodes are held fixed, such as vertices. A nonlinear solver that acts as a smoother on the remaining nodes (e.g. ten nonlinear conjugate gradient iterations) to locally move the nodes to minimize the curvature.

4.2.4 Objective function

Points are *contravariant* tensors with coordinates u^i , and the columns of the 3 by 2 Jacobian matrix, \mathbf{J} are *covariant* tensors $\mathbf{x}_i = \frac{\partial \mathbf{x}}{\partial u^i}$, and $\mathbf{x}_{ij} = \frac{\partial^2 \mathbf{x}}{\partial u^i \partial u^j}$. The metric tensor has components $[g_{ij}] = [\mathbf{x}_i^T \mathbf{x}_j]$.

A given surface and the curves within it have the following properties. The normal to the tangent plane is given by $\mathbf{N} = \mathbf{x}_1 \times \mathbf{x}_2 / |\mathbf{x}_1 \times \mathbf{x}_2|$. The second fundamental form has components $b_{ij} = \mathbf{x}_{ij} \cdot \mathbf{N}$. Lastly, we observe that the principal curvatures (κ_1, κ_2) are given by the eigenvalues of the generalized eigenvalue problem for $([b_{ij}]_{1 \leq i, j \leq 2}, [g_{ij}]_{1 \leq i, j \leq 2})$.

The objective function is the Frobenius norm of the second fundamental form over the surface [36]. If the columns of the Jacobian are orthonormal, then the objective function depends on the principal curvatures. In practice the average of the values Really a weighted average of the values of the second fundamental form at the nodes is weighted.

4.2.5 Surface coarsening

The algorithm of [16] as implemented in GTS is used to coarsen the surfaces. In this section the coarsening algorithm is described.

Consider a surface subdivision (triangulation) (V, E, K) (with graph (V, E)) of vertices, edges and faces (triangles). Suppose that a target number of faces, $m \ll n = |K|$, is given along with a tolerance τ (e.g. $\text{diam}(V) / \sqrt{m}$).

Defintion 1. *The pair of vertices (v_i, v_j) is contractible in the initial subdivision, $(v_i, v_j) \in C$, if either $(v_i, v_j) \in E$ or $\|v_i - v_j\| < \tau$.*

Edge contraction algorithms have a property, similar to transitivity, that the edges of a contracted node are the edges of the parents. The pair contraction algorithm has the same property, by definition.

Defintion 2. A pair contraction merges $(v_i, v_j) \in C$ into $\hat{v} \in V$. The edges (faces) connected to \hat{v} are the union of the edges (faces) connected to v_i and v_j . The sets E, C, K are updated by merging adjacency sets.

Each face adjacent to a vertex has a normal \mathbf{n} and tangent plane $\{\mathbf{x} : \mathbf{n}^T \mathbf{x} + \delta = 0\}$. Associate to each vertex v the error matrix

$$\mathbf{M} = \sum_{\text{adjfaces}} \mathbf{p}\mathbf{p}^T, \mathbf{p}^T = (\mathbf{n}^T, \delta), \mathbf{M} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = 0.$$

At a boundary vertex with the normal plane to the boundary edge defined by $\mathbf{q}^T = (\mathbf{n}^T, \delta)$, add $10\mathbf{q}\mathbf{q}^T$ to \mathbf{M} .

Note that if $(v_i, v_j) \in C - E$, and \mathbf{M}^i denotes the error matrix for \mathbf{v}_i then contraction merges \mathbf{M}^i and \mathbf{M}^j to $\mathbf{M}^i + \mathbf{M}^j$. The authors apply the rule in general.

Defintion 3. The contracting merges $(v_i, v_j) \in C$ into $\hat{v} \in V$ is

$$\kappa = \min\{\mathbf{u}^T(\mathbf{M}^i + \mathbf{M}^j)\mathbf{u} : \mathbf{u}^T = (\mathbf{w}^T, 1)\}.$$

The nodes are contracted to \hat{v} such that the argmin is $\hat{\mathbf{u}}^T = (\hat{v}^T, 1)$.

For

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{m}_{12} \\ \mathbf{m}_{12}^T & \mu \end{bmatrix}, \mathbf{M}_{11}\hat{v} = -\mathbf{m}_{12}.$$

If for $\bar{v} = (v_i + v_j)/2$, there holds $\|\hat{v} - \bar{v}\| > \tau$, then use a least squares solution of the form $\bar{v} - \mathbf{M}_{11}^+(\mathbf{m}_{12} + \mathbf{M}_{11}\bar{v})$.

In summary, the decimation by pairs algorithm to coarsen the surface is implemented as follows. Order the contractible pairs C by increasing cost κ . As long as the number of faces, $|K|$, is greater than m , do the following. Attempt to contract the most contractible pair. Allow the contraction if no normal of an adjacent face flips. If the contraction is not allowed, move on to the next contractible pair. Lastly, update the surface triangulation, C , the error matrices, and re-order the new costs in C .

4.3 Quadratic triangles

FFSD packages are built on hierarchical B-spline surface packages. In FFSD, locally a surface fits given vertices and vertex normals. The simplest possible type of surface is a quadratic Bézier triangle. There are three extra unknowns, that one might hope to use to match angles

at the vertices. These unknowns correspond to the limited freedom in choosing the control points as discussed in section 4.3.2. Further orientation constraints on the control points are worked out in section 4.3.3. The Jacobian may have critical points, and we figured out how to avoid them. The Jacobian may also have nearly critical points corresponding to folds in the surface however, and these we are not guaranteed to avoid. Several approaches were tried, and the results are documented here. As in section 4.1, gradient flows are used to solve optimization problems. We will use this idea extensively in section 4.3.4. The perturbation theory for Jacobian singular values is discussed in section 4.3.1.

As the surface mesh decimation tools will remove nodes interpolated in a linear triangle, refinement by triangulation in free form surface design requires curved triangles. These notes begin to describe how quadratic surfaces might be used. The problem of finding a good surface may be posed as a multivariate polynomial equation. The degree of the system is smaller than in the generic case, leading to singularities that the author does not yet well understand.

A quadratic surface is defined (or constructed) locally as follows.

The deCasteljau Algorithm for Quadratic Surfaces

```

for  $i = 0 : n$ 
  for  $j = 0 : n - i$ 
     $u = [i; j; n - i - j]/n$ 
     $c_{100} = [b_{200}, b_{110}, b_{101}]u$ 
     $c_{010} = [b_{110}, b_{020}, b_{011}]u$ 
     $c_{001} = [b_{101}, b_{011}, b_{002}]u$ 
     $f(u) = [c_{100}, c_{010}, c_{001}]u$ 
  end
end
end

```

The vertices $[b_{200}, b_{020}, b_{002}]$ together with three control points $[b_{101}, b_{110}, b_{011}]$ define the surface. The indexing of vertices and control points is as follows.

		020	
	110		011
200	101		002

The specification of a surface is by vertices and the surface unit normals at the vertices, and some additional information.

Differentials are independent of the choice of two dimensional coordinates. The surface is defined at a barycentric point, u such that $e^T u = 1$ and $u \geq 0$. The differential of a barycentric point is a barycentric vector, d such that $e^T d = 0$. Some algebraic manipulations yield that

$$f(u + d) = f(u) + 2[c_{100}, c_{010}, c_{001}]d + \mathcal{O}(\|d\|^2),$$

for c_{100} , c_{010} and c_{001}] defined in the deCasteljau algorithm. The tangent spaces at the vertices are determined as follows.

$$(4.1) \quad \begin{aligned} f(e_1 + d) - f(e_1) &= 2[b_{200}, b_{110}, b_{101}]d \\ f(e_2 + d) - f(e_2) &= 2[b_{110}, b_{020}, b_{011}]d \\ f(e_3 + d) - f(e_3) &= 2[b_{101}, b_{011}, b_{002}]d \end{aligned}$$

The tangent space at vertex b_{200} is in the range of $[b_{110} - b_{200}, b_{101} - b_{200}]$. The tangent space at vertex b_{020} is in the range of $[b_{110} - b_{020}, b_{011} - b_{020}]$ and so on. Each control point lies along the intersection of the corresponding tangent planes. A three dimensional space of control points determines surfaces with specified normals. In section 4.3.2, the space is parametrized, and the control point closest to the corresponding edge is determined. Such control points (nearly) minimize the second fundamental form, but, as curvature is the ratio of the second to the first fundamental forms, critical points may be present. Such surfaces may also have folds.

We attempt to uniquely determine the control points, and the surface, by specifying at each vertex the angle between the tangents to the boundary curve. The control points are moved iteratively to obtain the desired angles. An initial surface that is regular is required. Moving the control points to obtain regular surfaces is not trivial.

A two dimensional parametrization of the surface is required at times. Here the first two coordinates of u , $r(x, y) = f(x, y, 1 - x - y)$. The Jacobian matrix is

$$(4.2) \quad J = [r_{,1}, r_{,2}] = 2[c_{100} - c_{001}, c_{010} - c_{001}].$$

The second derivatives are constant,

$$\begin{aligned} r_{,11} &= 2(b_{200} - 2b_{101} + b_{002}), \\ r_{,12} &= 2(b_{002} + b_{110} - b_{011} - b_{101}), \\ r_{,22} &= 2(b_{020} - 2b_{011} + b_{002}). \end{aligned}$$

In the plane domain (x, y) the triangular subdomain with vertices $(0, 0)$, $(1, 0)$ and $(0, 1)$ determines the surface.

The critical points of r correspond to an eigenvector u of

$$(4.3) \quad (B_{100} - B_{001})u = (B_{010} - B_{001})u\xi,$$

where $c_k(u) = B_k u$. Being able to write down the critical points is one of the biggest advantages of using with quadratic surfaces. The eigenvectors is normalized as a (possibly infinite) barycentric point. The rank of J at a critical point is one unless $\xi = 1$, in which case $J = 0$. The coordinates of a critical point are the first two components of the eigenvector. If ξ is real, then $Jv = 0$ for $v^T = [1, \xi](1 + \xi^2)^{-1}$.

Generically, the critical points of f are isolated,

$$J(x + dx)(v + dv) = J(x)v + J(x)dv + Hdx + \dots$$

for

$$H = \left[\frac{\partial r_i v_j}{\partial x_j \partial x_k} \right] = [[r_{,11}, r_{,12}]v, [r_{,21}, r_{,22}]v].$$

On the other hand, the Jacobian matrix singular values generically have one dimensional level sets.

Nearly critical values in the triangular domain are detected in two steps. Evaluation over a regular net, using 720 singular value decompositions (SVDs), is reliable. First, any critical points inside the (open) domain are determined. Complex critical points are projected into the real plane, so as to detect a pair of nearly real critical points. Second, the smaller Jacobian singular value is minimized along each (closed) edge. The line minimization is by brute force. There are two levels, and on the fine level, we first attempt to use a Newton method using the Hessian. But if this produces large steps, brute force is applied once again. Presently, detection costs something like 120 SVDs.

4.3.1 Perturbation of SVDs

The derivatives of the singular value decomposition (SVD) of the matrix A in $M(3, 2)$ due to a one dimensional family of perturbations in the direction of the matrix B are reviewed. For $A(t) = A + Bt = U(t)\Sigma(t)V(t)^T$ with U in $O(3)$, Σ in $M(3, 2)$ and V in $O(2)$, analytic expressions for $\dot{\Sigma}$, \dot{V} , \dot{U} and $\dot{\Sigma}$ are derived, in that order.

Differentiate $A(t)$ once with respect to t ,

$$B = \dot{U}\Sigma V^T + U\dot{\Sigma}V^T + U\Sigma\dot{V}^T.$$

The differentials of the matrices of singular vectors have the form

$$(4.4) \quad \dot{U} = US, \dot{V} = VK$$

for S in $o(3)$ and K in $o(2)$ to be determined. The derivatives of the singular values are the diagonal elements of $C := U^T B V$.

Next, notation is introduced to keep track of the decomposition $M(3, 2) = M(2, 2) \oplus M(1, 2)$. For Σ_2 , S_2 and C_2 are in $M(2, 2)$,

$$\Sigma = \begin{bmatrix} \Sigma_2 \\ 0 \end{bmatrix}, S = \begin{bmatrix} S_2 & -s_3 \\ s_3^T & 0 \end{bmatrix}, C = \begin{bmatrix} C_2 \\ c_3 \end{bmatrix}.$$

$$C_2 = S_2 \Sigma_2 - \Sigma_2 K + \dot{\Sigma}_2, c_3 = s_3^T \Sigma_2.$$

Next we determine S and K . $C_2 + C_2^T = (S_2 + K)\Sigma_2 - \Sigma_2(S_2 + K) + 2\dot{\Sigma}_2$, and in particular $c_{21} + c_{12} = g_{21}(\sigma_1 - \sigma_2)$, $G = S_2 + K$, where $C = [c_{ij}]$ and $G = [g_{ij}]$.

Note in the following that $s_{21} = -s_{12}$.

$S_2 = Rs_{21}$ and $K = Rk_{21}$ for

$$R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

Next we determine s_{21} and k_{21} . For

$$M = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and

$$D = \begin{bmatrix} \sigma_1 - \sigma_2 & 0 \\ 0 & \sigma_1 + \sigma_2 \end{bmatrix},$$

there holds

$$M \begin{bmatrix} c_{12} \\ c_{21} \end{bmatrix} = DM \begin{bmatrix} s_{12} \\ k_{12} \end{bmatrix}.$$

Noting that $M^{-1} = M/2$, one may show that

$$(4.5) \quad \begin{bmatrix} s_{12} \\ k_{12} \end{bmatrix} = \mu \begin{bmatrix} \sigma_2 & \sigma_1 \\ \sigma_1 & \sigma_2 \end{bmatrix} \begin{bmatrix} c_{12} \\ c_{21} \end{bmatrix},$$

for $\mu = 1/((\sigma_2 - \sigma_1)(\sigma_2 + \sigma_1))$.

The most important case is the qualitative properties in a neighborhood of a regular critical point, $\sigma_1 > \sigma_2 = 0$. Note that $s_{31} = c_{31}/\sigma_1$ and as $\sigma_2 \rightarrow 0$, $[k_{21}, s_{21}] \rightarrow [c_{12}, c_{21}]/\sigma_1$. There is a pole singularity in $s_{32} = c_{32}/\sigma_2$.

Lastly we determine the second derivative of σ_2 in the direction B . Differentiate again,

$$0 = \ddot{U}\Sigma V^T + 2\dot{U}\dot{\Sigma}V^T + \\ U\ddot{\Sigma}V^T + 2\dot{U}\Sigma\dot{V}^T + \\ U\Sigma\ddot{V}^T + 2U\dot{\Sigma}\dot{V}^T.$$

Substitute $\ddot{U} = U(S^2 + \dot{S})$, $\ddot{V} = V(K^2 + \dot{K})$, to find

$$0 = (S^2 + \dot{S})\Sigma + 2S\dot{\Sigma} \\ + \ddot{\Sigma} - 2S\Sigma K + \\ U\Sigma(K^2 + \dot{K}) - 2\dot{\Sigma}K.$$

Along the diagonal, there holds

$$0 = ((S^2)_{ii} + (K^2)_{ii})\sigma_i + \ddot{\sigma}_i - 2(S\Sigma K)_{ii}.$$

Therefore, $S_2\Sigma_2K = s_{21}k_{21}R\Sigma_2R = -s_{21}k_{21}\text{diag}(\sigma_2, \sigma_1)$. Furthermore,

$$S\Sigma K = \begin{bmatrix} -s_{21}k_{21}\text{diag}(\sigma_2, \sigma_1) \\ -s_3^T\Sigma_2K \end{bmatrix}$$

Because S is skew symmetric, $x^T S^2 x = -\|Sx\|_2^2$, and we have

$$(4.6) \quad \ddot{\sigma}_2 = (s_{21}^2 + s_{32}^2 + k_{21}^2)\sigma_2 - s_{21}k_{21}\sigma_1.$$

4.3.2 Control points with prescribed normals

Specification of the vertices and normals defines a three dimensional space of control points. The formulas stated here are independent of the choice of surface parametrization (Jacobians). Given vertices and surface vertex normals, $\{v_i, n_i\}_{i=1}^3$, the corresponding matrix of control points $[b_{101}, b_{110}, b_{011}]$ is a function of an a in R^3 . By Equation (4.1), the control points are along the intersection of pairs of tangent spaces.

Control points with specified vertices and vertex normals have the form

$$(4.7) \quad \begin{aligned} b_{101} &= v_1 + T_1 G_1(:, 1)/2, \\ b_{110} &= v_2 + T_2 G_2(:, 1)/2, \\ b_{011} &= v_3 + T_3 G_3(:, 1)/2, \end{aligned}$$

where for $1 \leq i \leq 3$ and $a_4 = a_1$, each

$$(4.8) \quad G_i(a) = G_i(0) + P_i[a_i, 0; 0, a_{i+1}].$$

A dual relationship between the second columns of $G_i(a)$ and the control points exists.

The matrices \mathbf{G}_i and \mathbf{P}_i may be determined as follows. A function that returns a matrix $SO(3)$ with specified first column is used to transform $\{n_i\}_{1 \leq i \leq 3}$ to $\{\mathbf{T}_i\}_{1 \leq i \leq 3}$. Form

$$\mathbf{L}_i = [\mathbf{T}_i, \mathbf{T}_{i+1}]$$

in $M(3, 4)$, with $\mathbf{T}_4 = \mathbf{T}_1$. Compute the singular value decompositions, $\mathbf{L}_i = \mathbf{U}_i S_i \mathbf{V}_i^T$ where S_i is in $M(3, 4)$. Form

$$f_i = \mathbf{V}_i(:, 1:3) S_i(:, 1:3)^{-1} \mathbf{U}_i^T(v_{i+1} - v_i),$$

with $v_4 = v_1$. Then for $f_0 = f_3$,

$$\mathbf{G}_i = 2[-f_{i-1}(3:4), f_i(1:2)].$$

\mathbf{P}_i are computed so that $\mathbf{T}_i \mathbf{P}_i(:, 2) = \mathbf{T}_{i+1} \mathbf{P}_{i+1}(:, 1)$. For $\mathbf{V}_0 = \mathbf{V}_3$,

$$\mathbf{P}_i = [-\mathbf{V}_{i-1}(3:4, 4) \mathbf{V}_i(1:2, 4)].$$

The Jacobian J is a function of a as well as x , $J = J(a, x)$. To first order, $J(a + da, x + dx)v = J(a, x)v + Hdx + Ada$. By Equations (4.7) and (4.8),

$$(4.9) \quad \begin{aligned} D_a b_{101} &= \frac{1}{2}[T_1 P_1(:, 1), 0, 0], \\ D_a b_{110} &= \frac{1}{2}[0, T_2 P_2(:, 1), 0], \\ D_a b_{011} &= \frac{1}{2}[0, 0, T_3 P_3(:, 1)]. \end{aligned}$$

At the barycentric point u , $A = [T_1 P_1(:, 1), T_2 P_2(:, 1), T_3 P_3(:, 1)] \otimes (Ev)$ for $E = [u_3 - u_1, -u_1; u_2, u_1; -u_2, u_3 - u_2]$

4.3.3 Orientation

Given vertices and vertex normals, we have determined a three dimensional set of control points, parametrized by $(G_i(a))_{1 \leq i \leq 3}$, with a a real dimensional vector. As we will now show, the geometric constraint that the surface is positively oriented corresponds to the algebraic constraint $\det(G_i(a)) > 0$.

One may associate to the control points the 2×2 matrices $(\mathbf{T}_i)_{i=1}^3$ and $(\mathbf{G}_i)_{i=1}^3$ as follows:

$$\begin{aligned} 2[b_{101} - b_{200}, b_{110} - b_{200}] &= T_1 G_1(a) \\ 2[b_{110} - b_{020}, b_{011} - b_{020}] &= T_2 G_2(a) \\ 2[b_{011} - b_{002}, b_{101} - b_{002}] &= T_3 G_3(a) \end{aligned}$$

If Q is in $SO(3)$, then $Qa \times Qb = Q(a \times b)$. The orientations of the tangent planes at the vertices are related to the determinant of the G matrices. Consider the columns of the G matrices. If $G_1 = [g_1, g_2]$, then

$$\begin{aligned} 2(b_{101} - b_{200}) \times 2(b_{110} - b_{200}) &= T_1 g_1 \times T_1 g_2 = \\ &= [T_1, n_1][g_1; 0] \times [T_1, n_1][g_2; 0] = [T_1, n_1]([g_1; 0] \times [g_2; 0]) = \\ &= [T_1, n_1]([g_1; 0] \times [g_2; 0]) = n_1 \det(G_1). \end{aligned}$$

Equation (4.8) implies that

$$P_1^{-1} G_1(a) = P_1^{-1} G_1(0) + \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}.$$

We define x, y, z and r by

$$P_1^{-1} G_1(0) = \begin{bmatrix} y_1 & z_1/r_1 \\ r_1 & x_1 \end{bmatrix}.$$

This implies that $\det(G_1) = ((\alpha_1 + y_1)(\alpha_2 + x_1) - z_1) \det(P_1)$. And furthermore, $\det(G_1(a)) \det(P_1) = ((\alpha_1 + x_1)(\alpha_2 + y_1) - z_1) \det(P_1)$. Finally,

$$\alpha_1 \alpha_2 + \alpha_1 x_1 + \alpha_2 y_1 = (\det(G_1(a)) - \det(G_1(0))) \det(P_1).$$

A similar construction applies to G_2 and G_3 . These equations may be solved given $\det(G_i)$, say $\det(G_i) = 1$ (but this is not at all obvious). This approach to selecting feasible control points is not discussed further.

Note that if $G_i(a)$ is singular, then the surface has a cusp singularity at the vertex. This is problematic for gradient based optimization.

4.3.4 Optimization of the control points

If tentative control points have been chosen, and the corresponding surface has a critical point or a fold, then the location of the control points must be changed or optimized. Three different objectives apply, depending on the nature of the singularity. The most common is a fold. A second case concerns a (nearly) critical point on the boundary of the domain. The objective is to maximize the perturbed (nearly) critical value. The third case is a critical point in the (open triangular) domain. The objective is to move the critical point nearer to the boundary.

The strategy is to select da to control the flow of the argmin, x , along some desirable vector field dx . A locally optimal da is applied to the global regularization problem. The Jacobian matrix has singular value decomposition $J = U_j \Sigma_j V_j^T$ for $V_j = [v_1, v_2]$ and $Jv_i = u_i \sigma_i$, $1 \leq i \leq 2$. The derivative of the (right) singular vectors in the direction v_2 is given in Equation (4.4). Let $\kappa = K_{21}$. To first order, $J(a + da, x + dx)(v_2 + dv_2) = Jv_2 + J dv_2 + H dx + A da = u_2 \sigma_2 - u_1 \sigma_1 d\kappa + H dx + A da$.

4.3.5 Interior Critical Points

For a regular critical point (i.e. $\sigma_1 > \sigma_2 = 0$), consider the example $r(s, t) = [s, st, s^2 + t^2]^T$. The unique critical point is $[s, t] = [0, 0]$. The surface intersects itself along the curve $r(0, \pm t)$. The example is the normal form (in the sense of catastrophe theory) of a regular critical point.

The critical point is an end point of the curve of self-intersection. The simplest case is a critical point of f , $\sigma_2 = 0$, in the (open) triangular domain. The differentials of the control points, da , and the critical point, dx satisfy

$$(4.10) \quad A da + H dx - u_1 \sigma_1 d\kappa = 0.$$

Given dx , if A is (nearly) singular, perturb dx slightly to reduce the norm of da , subject to the constraint that the direction of dx does not change significantly.

The control points move in the direction da chosen so that the critical points move in the same direction dx as the direction of the shorted path out of the triangular domain. $d\hat{x}$. As usual, the starting point is a singular value decomposition; $[A, H, -u_1 \sigma_1] = U_l \Sigma_l V_l^T$ for V_l in $M(6, 3)$. The matrix of right singular vectors, $V_l = [V_1, V_2]$, for V_1 and V_2 both in $M(6, 3)$. For any da , the descent direction always exactly solves Equation (4.10). The equation that says this is that for $F = -[I_2, 0]V_2^{-T}V_1$, there holds $dx = F da$ (assuming that V_2 has full column rank). Next compute another singular value decomposition, $F = U_f \Sigma_f V_f^T$ for V_f in $M(3, 2)$. The search directions are determined from a regularization problem in standard form,

$$\min \|F da - d\hat{x}\|^2 + \lambda^2 \|da\|^2.$$

The parameter λ is chosen to minimize the angle between $dx = F da$ and $d\hat{x}$ by a line search in the direction of da . The eigenvalue problem, Equation (4.3), determines the nearby critical point $x(\tau)$.

4.3.6 Exterior Critical Points

Next we discuss a partial solution in the case of a nearly critical point in the (open) triangular domain, $\sigma_2 > 0$.

Consider the entire $\{J(x) : x \in R^2\}$. Along the curve $u_2^T H dx = 0$, there holds $\sigma_2(x + dx\tau) = \sigma_2(x) + \mathcal{O}(\tau)^2$. Moreover along $u_2^T H dx = 0$, the right singular vector u_2 changes slowly, and so H also changes slowly. In a one parameter family of Jacobian matrices, in a neighborhood of a regular critical point (i.e. $\sigma_1 > \sigma_2 = 0$), by Equation (4.6) there holds $\ddot{\sigma}_2 = (c_{32})^2/\sigma_2 + \mathcal{O}(1)$. This “force” keeps σ_2 from changing sign. This explains why choosing dx so that σ_2 decreases, namely $dx = -H^T u_2$, decreases σ_2 insignificantly: the second derivative $\ddot{\sigma}_2$ becomes even larger and positive, and after a brief negligible decrease, σ_2 actually increases.

Due to $dc_{32} = u_3^T H dx$, variation of x in the curve $u_3^T H dx = 0$ is insensitive to small values of σ_2 , $\ddot{\sigma}_2 = \mathcal{O}(1)$. The curve $u_3^T H dx = 0$ through a critical point traces a fissure in the $\sigma_2(x)$ surface. Such fissures on the $\sigma_2(x)$ surface correspond to folds in the triangular surface. A very common situation is that no critical points are in the triangular domain, and the smallest values on two edges are on two nested level sets of $\sigma_2(x)$ that both enclose a critical point of $\sigma_2(x)$. If the two minima along two edges are in the level set that encloses the same critical point, simply optimize the smaller argmin.

We see that if the triangular domain is free of critical points, then the minimum over the boundary of the Jacobian singular values is a very good estimate for the minimum singular value of the Jacobian over the whole triangle. Recall that at a local minimum, \hat{x} , along an edge the gradient of σ_2 , $u_2^T H$, points in the direction of the inward pointing normal. Suppose that the vector t is a unit tangent vector to the edge. For x near to \hat{x} , the differential du_2 changes extremely quickly from $u_2(\hat{x})$. $du_2 = u_3 u_3^T H dx / \sigma_2 +$ lower order terms. For $\delta = \epsilon u_3^T H t / \sigma_2$, $\nabla \sigma_2(\hat{x} + t\epsilon) = (u_2 + u_3 \delta)^T H / (1 + \delta^2) +$ lower order terms. The estimate is accurate for $\epsilon < 1$ and equivalently for $\delta < 1/\sigma_2$, which is much larger than one. If x varies along a curve that transversely intersects the fissure, then $u_2(\hat{x} + t\mu) \approx u_3(\hat{x}) \text{ sign}(\mu)$, for appropriate t .

In order to use a local theory to maximize the minimum σ_2 , the derivative of $\hat{x}(a)$ with respect to a must be negligible. In general, instead of $da = A^T u_2$, one may use $g = A^T u_2 - A^T u_3 \xi$, for the θ such that $u_3^T A da = 0$. A line search will determine a suitable value of σ_2 just as at a boundary point. If θ is the angle between $A^T u_2$ and $A^T u_3$, then $u_2^T A g = \|A^T u_2\|^2 \sin^2(\theta)$, which is always positive, so g is always a descent direction. But the rate of descent can be very slow, and can change very quickly. This is not robust in practice. Furthermore, we have not ensured that normals have a positive orientation.

4.3.7 Formulation of a Polynomial Equation

The problem of finding the control points with specified angles is equivalent to a multivariate polynomial equation. Suppose that at the i th local vertex, γ_i is the cosine of the desired angle.

The trivariate quartic polynomial equation $\mathbf{f}(\mathbf{a}) = \mathbf{0}$ for

$$\mathbf{a} = [\alpha_1, \alpha_2, \alpha_3]^T,$$

may be solved using resultants. The problem statement is as follows. We are given scalars

$$\mathbf{c} = [\gamma_1, \gamma_2, \gamma_3]^T,$$

and matrix-valued functions with linear coefficients

$$(4.11) \quad \mathbf{G}^i(\mathbf{a}) = \mathbf{G}^i + \mathbf{P}^i \begin{bmatrix} \alpha_i & 0 \\ 0 & \alpha_{i+1} \end{bmatrix}$$

where $\alpha_4 = \alpha_1$. We form the normal equations

$$\mathbf{W}^i(\mathbf{a}) = (\mathbf{G}^i)^T(\mathbf{a})\mathbf{G}^i(\mathbf{a}).$$

The diagonal elements of $\mathbf{W}^i(\mathbf{a})$ are univariate quadratic polynomials, and the off diagonal elements are bivariate quadratic polynomials. The three bivariate quartic equations

$$\mathbf{f}^i(\mathbf{a}) = \gamma_i^2 \mathbf{W}_{1,1}^i \mathbf{W}_{2,2}^i - \mathbf{W}_{1,2}^i \mathbf{W}_{2,1}^i$$

together form the trivariate quartic system.

We can use this to build the 3×35 coefficient matrix \mathbf{F} in a way that depends only on the input data. Here 75% of the entries in \mathbf{F} vanish. The solutions include the control points we want, as well as control points in which the cosine of the i th vertex angle is $-\gamma_i$, and also control points such that the surface is not positively oriented. We will not go into the details to the construction of resultants here. However the technique to extract rows from the resultant to determine an eigenvalue problem does not work in this case.

5 Interface Identification

Interface identification is a reduced basis method for inverse problems. Assumptions are made about the data topology. Linear perturbation methods are used to characterize the interior scatter boundary (see section 5.7-5.9 in [23]). Piecewise real-analytic conductivities may be determined from static boundary measurements [26]. The inverse conductivity problem for small inclusions and Maxwell's equations may be addressed using asymptotics as discussed, for example, in [2]. Here we assume that the inclusion is not small. A P1 finite element method is used. Mesh generation is discussed in section 5.1.

The sharp interface inverse algorithm involves perturbing the triangulated interface surface mesh in a low dimensional space of smooth deformations. The determination of such a space (that is independent of mesh coordinates) follows.

Allowing each vertex to move in an arbitrary real direction has the advantage that any required surface (of the same genus) is ‘reachable’ from the original surface. However if two adjacent vertices move towards each other, then the surface will become *knotted* (folded or looped). The treatment of a search direction (surface deformation) that adds knots to the surface is not trivial.

A surface is invariant under motion in the tangent space. One may constrain the surface deformation to be along the normals. On the other hand, normal deformations poorly approximate rigid motions. One may include rigid motion of the surface, $y(x) = Qx + b$, Q in $SO(3)$. If the initial surface is a sphere, translations suffice. A two phase method is used: a rigid motion phase followed by a normal deformation phase. Phase one is discussed in section 5.3 and the problem of a general deforming interface is discussed in section 5.5.

Results on the use of the sharp interface inversion tools in the FDM3D finite difference for the time-harmonic Maxwell’s equations were not available in time to include in this report.

5.1 Creating tetrahedral meshes with TetGen

We investigate interface adaptivity based on the solution to elliptic partial differential equations solved by the P1 finite element method. The underlying discretisation of the model domain is composed of a Delaunay tetrahedral mesh generated by the OpenSource meshing package TetGen v.1.3.4 (see <http://tetgen.berlios.de/>). In this section, we review how to use TetGen.

TetGen will accept Geomview format files as input, and GTS can output Geomview format files, `*.off`. For the first experiments with TetGen, GTS created a small sphere surface mesh and saved it in this Geomview format. Invoking TetGen with `tetgen -pqe0 outfile.off` created the first tet-mesh, comprising multiple files.

TetGen’s `*.smesh` file format is similar to GTS’s `*.gts` file format, except TetGen’s facet list refers to the list of vertices (nodes) and GTS’s refers to an edge list that refers to the vertices.

To create one surface inside of another, the icosahedron example surface (`icosa.gts`) was used as an inner surface. The outer surface was this same icosahedron model with the vertices farther from the origin. The vertices and facets from these two models were assembled into an `*.smesh` file by hand. The command `tetgen -pqz concenicosa` meshed the regions.

A few other `*.smesh` files were assembled to explore region and node attribute features and to characterize the numerical requirements on the meshes.

5.2 Combining surface meshes into tetrahedral meshes

We wrote a program, called `Enmesh`, to use TetGen and GTS together. The `Enmesh` program transforms two input surface meshes (in GTS format) and a region file (which tells TetGen

that the inner surface divides the domain into one interior and one exterior subdomain). It is also possible to input additional points (as is needed for mesh refinement) into a tetrahedral mesh. Three output files define the mesh: the vertices coordinates, `*.node`; the indices of the vertices comprising each tetrahedron, `*.ele`; and the indices of the vertices comprising these face in each input surface, `*.face`.

The TetGen data structure is called TetGenIO (see the TetGen User Manual). Enmesh parses the surface files using GTS, and then populates the TetGenIO data structure.

The optional `regionfile.txt` is used to label the tetrahedral elements inside and outside of the inner surface. The first line of `regionfile.txt` consists of three floating point values for the coordinates of some point inside the inner sphere and another value that will serve as a label for all tetrahedra in this region. The second line of `regionfile.txt` consists of three floating point values for the coordinates of some point outside the inner sphere but still inside the outer sphere and another value to label the tetrahedra in this region. Note that these two lines may be reversed. e.g.:

```
0.0 0.0 0.0 4
-2.0 -2.0 -2.0 3
```

means that the area from $(0, 0, 0)$ to the inner surface will be labeled with a 4, and the region from the point $(-2, -2, -2)$ to either surface will be labeled with a 3. (Assuming the inner surfaces origin is at $(0, 0, 0)$ and its radius is less than 2.)

Element quality is important for calculation speed, stability, and convergence. There is no single, easily calculated measure of element quality. TetGen generates Delaunay meshes of elements (tetrahedra) with radius-edge ratio bounded by a user supplied parameter (say two). A tetrahedron's radius-edge ratio is the radius of its circumsphere divided by its shortest edge length. A low radius-edge ratio is a necessary, but not a sufficient, condition for element quality. A tetrahedron's aspect ratio is its longest edge length divided by the diameter of its inscribed sphere. The higher ratios correspond to 'slivers', badly shaped tetrahedra that cause problems in the finite element calculations. The aspect ratios for the tetrahedra in the `syndata4` mesh are shown in Figure 5.1 (solid lines). The aspect ratios here are satisfactory, but must be monitored.

5.3 Motion of a spherical interface

An example with the simplest initial surface, a sphere, is considered next. In this case our reduced Jacobian is 3×3 , corresponding to translations. The domain is a sphere of radius ten. Here the (potential) field is the solution of Laplace's equation with diffusion constant one in the domain outside the inclusion, and the diffusion constant is ten within the inclusion. Physically, this corresponds to a direct current problem.

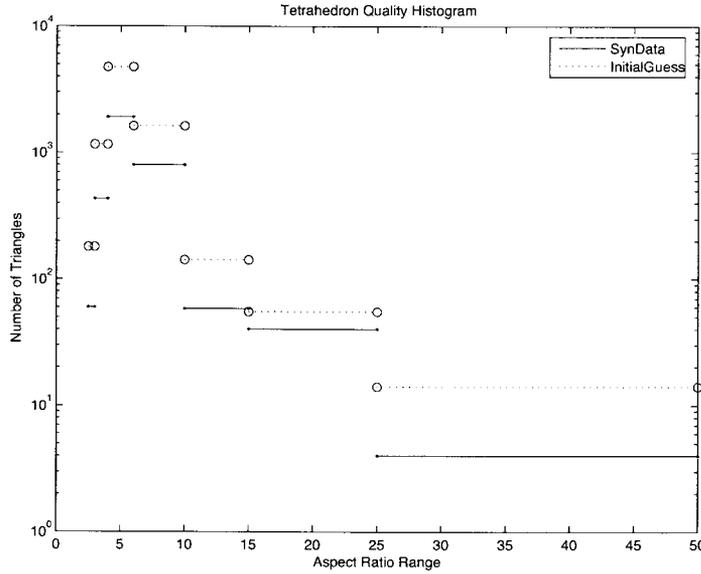


Figure 5.1. Histogram of tetrahedral element quality for two representative meshes. The `syndata4` mesh has 3303 tetrahedra and 920 vertices. The `InitialGuess` mesh has 7910 tetrahedra and 1736 vertices. Delaunay meshes contain some slivers with low element quality.

A dipole source is used along the line segment from $x_l = (1, 0, 1)$ to $x_r = (1, 0, -1)$, with strength

$$\frac{1}{|x - x_l|^2 + .1} - \frac{1}{|x - x_r|^2 + .1}.$$

Because only one dipole is used, the solution is not unique.

The target synthetic data is a tetrahedral mesh called `syndata4`. The mesh has a 'puck' shaped interior surface,

$$\{(x, y, z) : x^2 + y^2 = 1, |z| \leq \frac{1}{2}\} \cup \{(x, y, z) : x^2 + y^2 < 1, |z| = \frac{1}{2}\}.$$

The tetrahedral mesh consists of 3303 tetrahedra and 920 vertices. Figures 5.2 and Figure 5.3 show cut-away views of the mesh from the top and side respectively.

The mesh `initialguess5` is named to reflect its status as the first guess used for solving the inversion problem. Once again the domain is a sphere of radius ten. The inner region, the *seeker*, is a sphere of radius one centered at $(2.5, 2.5, -1)$. The tetrahedral mesh consists

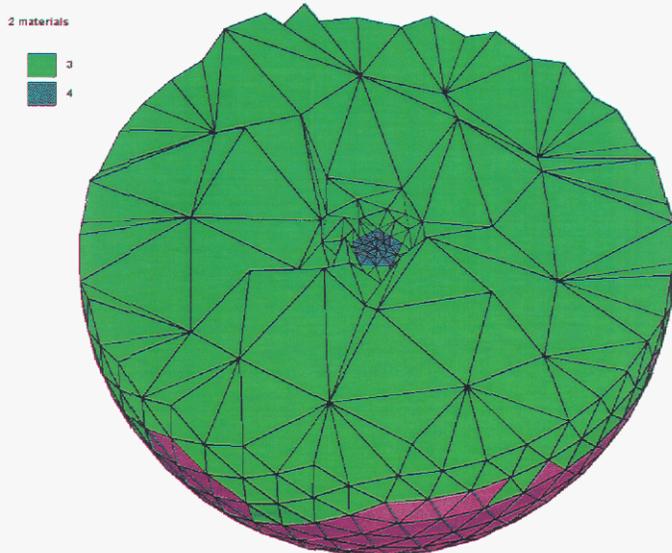


Figure 5.2. A cut-away view of the puck-shaped inclusion in the syndata synthetic data tetrahedral mesh is shown from the top.

of 7910 tetrahedra and 1736 vertices. Figures 5.4 and 5.5 show cut-away views of the seeker surface.

In a meshed domain $\Omega = \Omega^e \cup S \cup \Omega^i$ conforming to the spherical interface $S = (\mathbf{x} : |\mathbf{x} - \mathbf{c}| < r)$, keeping track of the motion of the interface is not trivial.

All the elements with vertices on S define a tubular neighborhood of S , henceforth the tube. For the purpose of computing sensitivities, the interface motion is restricted to the tube. For each element in the tube, the new interface position is determined.

The step-size for determining the numerical shape derivatives was determined empirically. The step-size must not be too small.

5.3.1 Assembly of the perturbed stiffness matrix

As the center moves to $\mathbf{c}(\tau) = \mathbf{c} + \mathbf{d}\tau$, The surface moves to $S_\tau = (\mathbf{x} : |\mathbf{x} - \mathbf{c}(\tau)| < r)$. The interface has moved across an element if some of the elements vertices are inside and some vertices are outside.

During the movement of the interface, the intersection of a large sphere and a small tetrahedron is approximated as an intersection of a plane and a tetrahedron. Only cases in which a plane divides a solid into two subdomains of positive measure are considered in detail. A

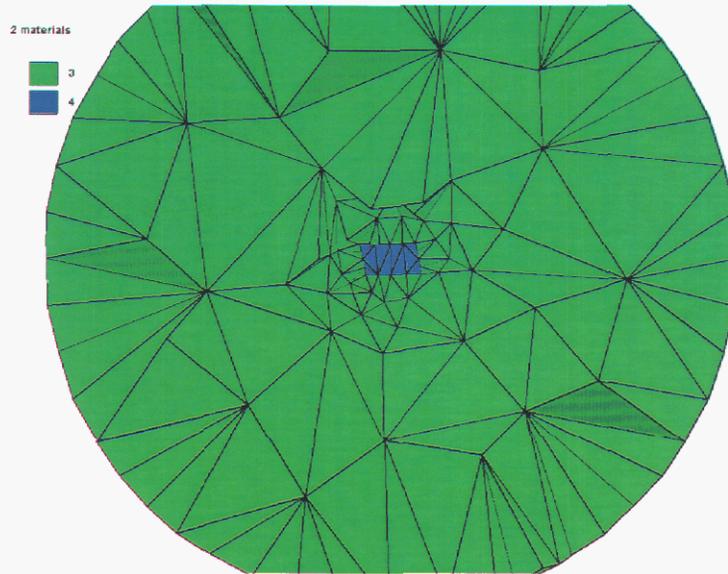


Figure 5.3. The *syndat* tetrahedral mesh is shown in cut-away view from the side.

plane that intersects an edge in two points contains the edge.

The plane may also intersect one, three or four faces. An intersection of one plane does not produce any subdomains. An intersection of three faces produces two subdomains, one a tetrahedron whose apex is the single interior or exterior vertex, and the other a truncated tetrahedron whose volume is the difference between the original volume and the volume of the other subdomain. If a plane intersects four faces, the two subregions are both unions of two tetrahedra. The plane may intersect one or three edges. An intersection on one edge does not produce any subdomains. An intersection on three edges is an intersection on three faces. The plane may intersect one, two, or three vertices. An intersection with one vertex produces no subdomains. If a plane intersects two vertices, then the plane contains one of the edges. If the plane intersects three vertices, the plane contains three edges and therefore is an intersection on one face, producing no subdomains.

Computing the intersection of the whole interface and each element in the mesh, one at a time, is inefficient. Efficient shape derivative calculation is possible if the deformation is constrained to lie in a tubular neighborhood of the surface. By keeping within a one element thick tube neighborhood of the surface, element adjacency is not needed. In order to use larger step sizes, the element adjacency graph will be needed.

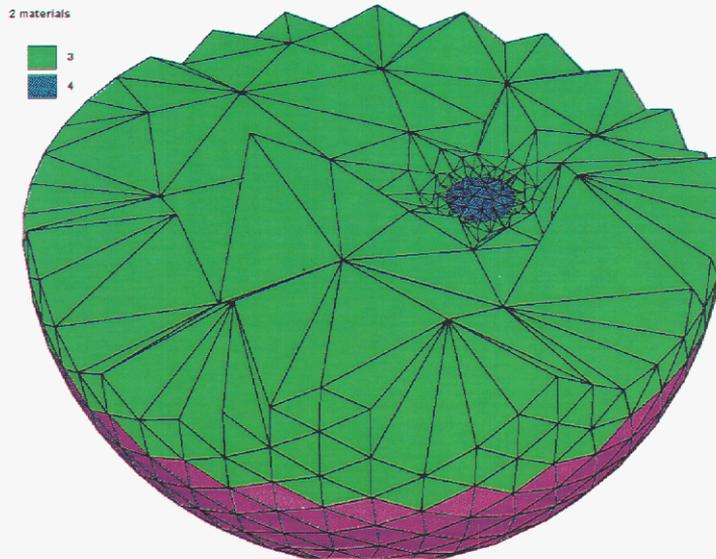


Figure 5.4. The tetrahedral mesh of domain with an the initial guess of the inclusion is shown in cut-away view.

5.4 Refinement and coarsening of surface meshes

As preparation for this discussion of the problem of deforming interfaces, we discuss next the refinement and coarsening of surface meshes. The GTS package is used to coarsen or refine the surface meshes (remove or add vertices) while maintaining the Delaunay property. GTS includes two coarsening algorithms, a global volume preserving algorithm and a default algorithm based on edge lengths. Both work very well. A deformed sphere mesh of 642 vertices is shown in Figure 5.7.

Figure 5.8 shows the result of applying a surface mesh coarsening algorithm from GTS.

5.5 Implementation of deforming surfaces

To change from a seeker sphere to a deforming surface, algorithms are needed for several tasks. In section 5.5.1, an algorithm is presented that defines a low dimensional set of deformations. In section 5.5.2, the step length for numerical gradient with respect to each deformation is determined.

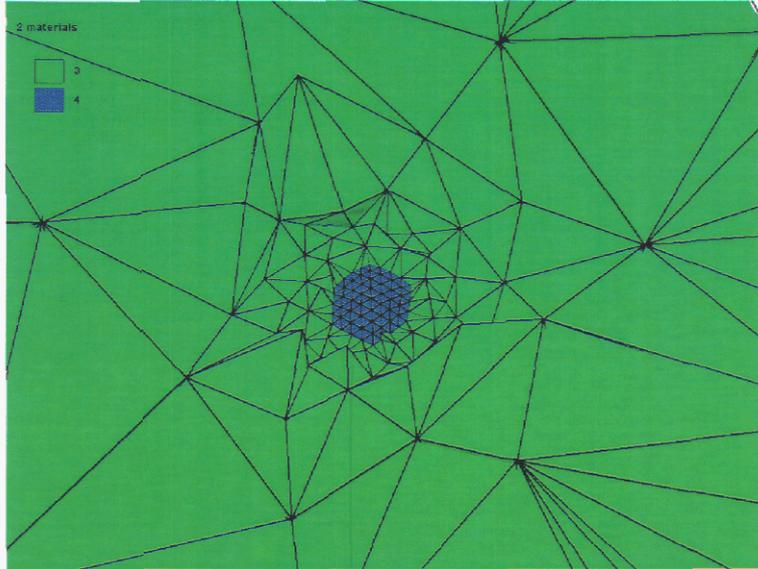


Figure 5.5. The tetrahedral mesh for the initial guess is displayed. The guess of the inclusion location is shown in detail, revealing small tetrahedra near the interface.

5.5.1 Space of Deformations

A low dimensional space of deformations is used. The deformation corresponds to motion along the vertex normals. Spherical harmonics are natural for spheres. For general (genus zero) surfaces we use the fundamental eigenfunctions for the smoothing operator on the connectivity graph.

A method for generating smooth deformations of surfaces (analogous to spherical harmonics) is presented. An averaging matrix A is derived from the Boolean adjacency matrix B for the vertex to vertex connectivity in the surface. In contrast to the graph Laplacian $\text{diag}(B1) - B$, the averaging matrix $A = D(B + I)D$ for the positive definite diagonal D such that $\text{diag}(D^2) = (B + I)1$. The averaging matrix is symmetric and indefinite. An eigenvector of A is interpreted as the surface velocity at each vertex normal. An advantage of the averaging matrix, compared to the Laplacian, is that it is the greatest (easy to compute) eigenvalues of A whose eigenvectors correspond to smooth surface deformations.

These deformations are used to test the GTS coarsening algorithm in a previous section.

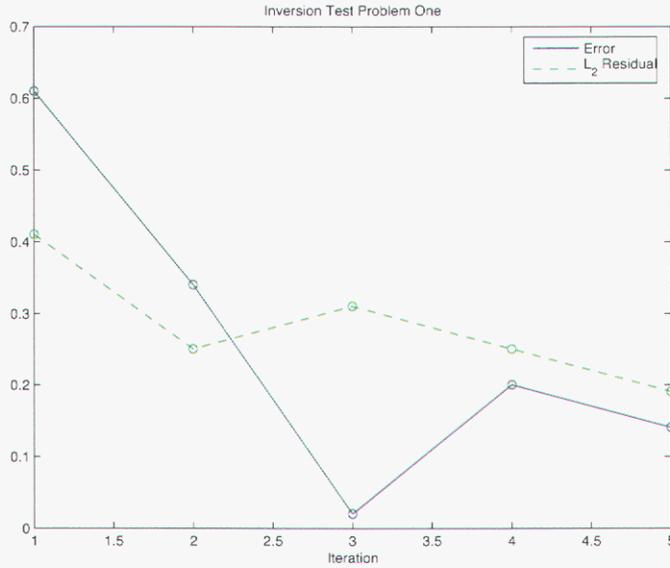


Figure 5.6. Results for first test problem. The solution is not unique, only one data point used. The error is the distance from the center of the seeker sphere to the set of solutions. The L_2 residual here is the objective function, the L_2 error in the gradient along the dipole segment.

5.5.2 Step Lengths

The deformed surface remains within the tube; it passes through vertices or edges of the tube elements. A prerequisite for computing the step lengths and volume fractions is to define the interface motion.

The deformation is a sum of its positive and negative parts, $d = (d + |d|)/2 + (d - |d|)/2$. The positive (negative) part of the deformation moves the surface in the exterior (interior) tube region. For each element in the exterior (interior) tube region, the interface nodes move some non-negative (non-positive) distance along the vertex outward normals. The images of the vertex nodes under the deformations define cutting planes for the tube elements.

Next, we determine step length so that the deformation remains in the tube. A multiple of the positive (negative) part of the deformation will be applied to elements outside (inside) the surface. For each tube element, determine the multiplier such that the cutting plane is on the element boundary. The step length is the minimum over the elements of these multipliers. The vertices will be denoted $\{\mathbf{x}_l, \mathbf{x}_r, \mathbf{x}_i, \mathbf{x}_j\}$ if they split into two pairs, and $\{\mathbf{x}_o, \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$

if they split into a triple and one other. The corresponding normal vectors and deformation components inherit the subscript of the vertex. If one vertex is on the tube,

$$\min_k \frac{\mathbf{n}_o^T(\mathbf{x}_k - \mathbf{x}_o)}{\lambda_o}.$$

If two vertices are on the tube, the exact step length is overly complicated. Instead, the tangent planes are raised at each vertex until one intersects one of the other two vertices,

$$\min \left(\min_k \frac{\mathbf{n}_l^T(\mathbf{x}_l - \mathbf{x}_k)}{\lambda_l}, \min_k \frac{\mathbf{n}_r^T(\mathbf{x}_r - \mathbf{x}_k)}{\lambda_r} \right).$$

Suppose that three vertices are on the tube, and form the 3×3 matrices

$$\mathbf{X} = [\mathbf{x}_i - \mathbf{x}_o, \mathbf{x}_j - \mathbf{x}_o, \mathbf{x}_k - \mathbf{x}_o], \quad \mathbf{N} = [\mathbf{n}_i \lambda_i, \mathbf{n}_j \lambda_j, \mathbf{n}_k \lambda_k].$$

The step length is the real eigenvalue (\mathbf{X}, \mathbf{N}) of appropriate sign and minimal magnitude whose eigenvector coefficients do not sum to zero.

Next compute the volume fractions for the deformation and step length as in the case of the spherical seeker. If a deformation is outside of a tube element, the volume fractions are

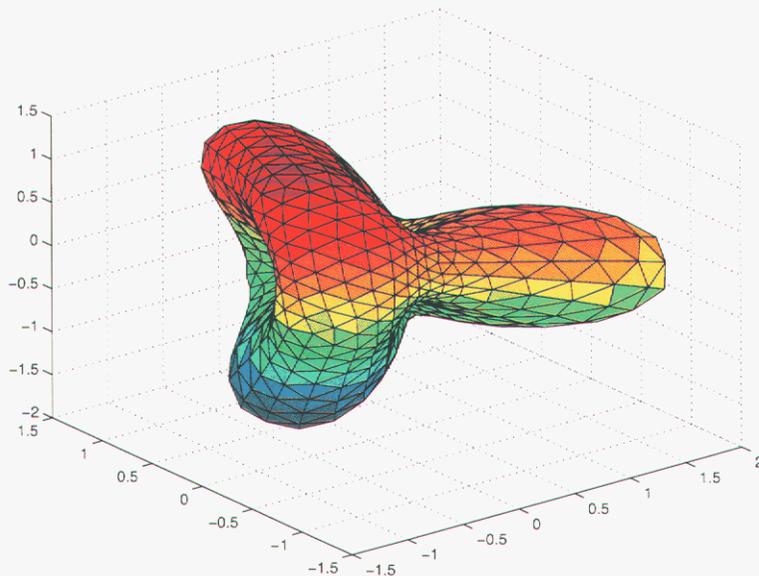


Figure 5.7. A surface mesh with 1280 faces, 1920 edges, and 642 vertices is shown.

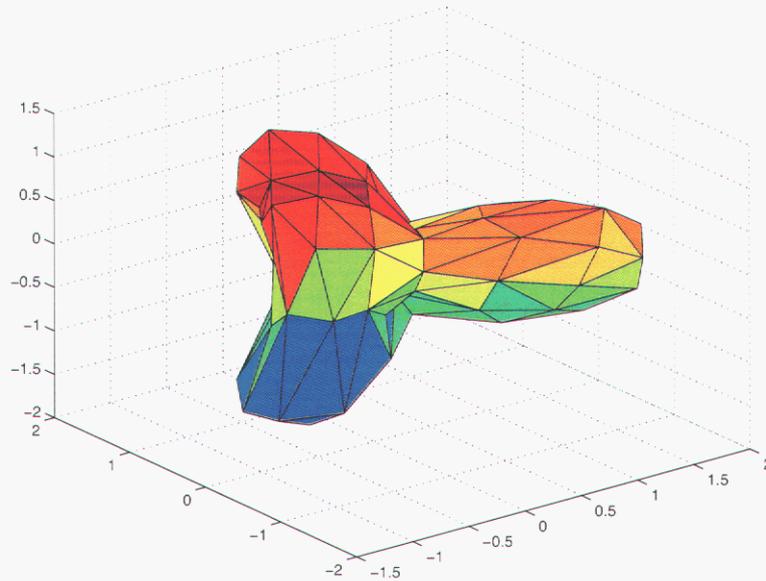


Figure 5.8. The surface mesh shown results from coarsening the mesh of Figure 1. The coarse mesh has 160 faces, 240 edges, and 82 vertices.

trivial but one must still check for correctness. A tube element has one, two or three nodes on the surface. If one node is on the surface, then the cutting plane is perpendicular to the vertex normal. If three nodes are on the surface, then the cutting plane contains the images of the three points. If two points are on the surface, then the cutting plane contains the images of both points, and is perpendicular to an averaged normal. If the two image points are \mathbf{x}_l and \mathbf{x}_r and the corresponding normals are \mathbf{n}_l and \mathbf{n}_r and then use the mean normal is

$$\mathbf{n}_o = \frac{\mathbf{n}_l + \mathbf{n}_r}{|\mathbf{n}_l + \mathbf{n}_r|},$$

and

$$\mathbf{u} = \frac{\mathbf{x}_l - \mathbf{x}_r}{|\mathbf{x}_l - \mathbf{x}_r|}$$

to determine the average normal

$$\mathbf{n} = \frac{(I - \mathbf{u}\mathbf{u}^T)\mathbf{n}_o}{|(I - \mathbf{u}\mathbf{u}^T)\mathbf{n}_o|}.$$

References

- [1] M. Adams, M. Brezina, J. Hu, and R. Tuminaro. Parallel multigrid smoothing: polynomial versus Gauss-Seidel. *J. Comp. Phys.*, 188(2):593–610, 2003.
- [2] H. Ammari, M.S. Vogelius, and D. Volkov. Asymptotic formulas for perturbations in the electromagnetic fields due to the presence of inhomogeneities of small diameter ii. the full maxwell equations. *J. Math. Pures Appl. IX*, 80:769–814, 2001.
- [3] R.E. Barnhill and G. Farin. C1 quintic interpolation over triangles - two explicit representations. *Int. J. Numer. Methods. Engrg.*, 17(12), 1981.
- [4] P. Bochev, C. Garasi, J. Hu, A. Robinson, and R. Tuminaro. An improved algebraic multigrid method for solving Maxwell’s equations. *SIAM J. Sci. Comput.*, 25(2):623–42, 2003.
- [5] P. Bochev, J Hu, A. Robinson, and R. Tuminaro. Towards robust 3D Z-pinch simulations: Discretization and fast solvers for magnetic diffusion in heterogeneous conductors. *Elect. Trans. on Num. Anal.*, 15:186–210, 2003.
- [6] A. Bossavit. A rationale for “edge-elements” in 3-D fields computations. *IEEE Transactions on Magnetism*, 24(1):74–79, 1988.
- [7] A. Bossavit. Whitney forms: a class of finite elements for three dimensional computations in electromagnetism. *IEE Proceedings*, 135(8):493–500, 1988.
- [8] A. Bossavit. *Computational electromagnetism*. Academic Press, 1998.
- [9] D. Braess. *Finite elements. Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, Cambridge, 1997.
- [10] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991.
- [11] P. Ciarlet. *The finite element method for elliptic problems*. North Holland, Amsterdam, 1978.
- [12] D. Day and M. Heroux. Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, 23(2):480–498, 2001.
- [13] D. Day and L. Romero. Roots of polynomials expressed in terms of orthogonal polynomials. *SIAM J. Numer. Anal.*, 2005.

- [14] M. Fortin F. Brezzi, J. Douglas and D. Marini. Efficient rectangular mixed finite elements in two and three space variables. *M²AN Math. Model. Numer. Anal.*, 21:581–604, 1987.
- [15] B. Fischer. *Polynomial based iteration methods for symmetric linear systems*. Advances in Numerical Mathematics. Wiley-Teubner, 1996.
- [16] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 1997.
- [17] P.L. George and H. Borouchaki. *Delaunay triangulation and meshing: Application to finite elements*. Hermes, Paris, 1998.
- [18] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer, Berlin, 1986.
- [19] E. Hille. *Analytic Function Theory*, volume II. Ginn and Company, Boston, 1962.
- [20] R. Hiptmair. Multigrid method for Maxwell’s equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1998.
- [21] R. Hiptmair. Canonical construction of finite elements. *Math. Comp.*, 68:1325–1346, 1999.
- [22] J. Hu, R. Tuminaro, P. Bochev, C. Garasi, and A. Robinson. Toward an h-independent algebraic multigrid method for solving Maxwells equations. *SIAM J. Sci. Comput.*, 00(0):000, 2005.
- [23] V. Isakov. *Inverse Problems for partial differential equations*. Springer, Santa Clara, 1998.
- [24] W. Kahan. Mathematics written in sand. In American Statistical Association, editor, *Proceedings of the Statistical Computing Section*, pages 12–26, 805 15th St NW Washington DC, 20005, 1983. American Statistical Association.
- [25] W. Kahan. What has the volume of a tetrahedron to do with computer programming languages? <http://www.cs.berkeley.edu/wkahan/>, April 2001.
- [26] R.V. Kohn and M. Vogelius. Determining conductivity by boundary measurements II. interior results. *Comm. Pure Appl. Math.*, pages 643–667, 1985.
- [27] J. Marsden and A. Tromba. *Vector calculus*. W.H. Freeman and Co., 1976.
- [28] J.C. Nedelec. Mixed finite elements in \mathbb{R}^3 . *Numerische Mathematik*, 35:315–341, 1980.

- [29] J.C. Nedelec. A new family of finite element methods in \mathbf{R}^3 . *Numerische Mathematik*, 50:57–81, 1986.
- [30] G. A. Newman, J. Hu, R. Tuminaro, and S. Margolis. An amg solver for the 3D time-harmonic maxwell equations. In *Copper Mountain Conference on Iterative Methods*, March 2004.
- [31] G.A. Newman and P. T. Boggs. Solution accelerators for large scale three-dimensional electromagnetic inverse problems. *Inverse Problems*, 20(6):S151 – S171, 2004.
- [32] P. O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM Rev.*, 46(2), 2004.
- [33] W. Rachowicz and L. Demkowicz. An *hp*-adaptive finite element method for electromagnetics part 1: Data structure and constrained approximation. *Computer Methods in Appl. Mech. and Engin.*, 187:307–335, 2000.
- [34] P.A. Raviart and J.M. Thomas. A mixed finite element method for second order elliptic problems. *Lecture Notes in Mathematics*, vol. 606, 1977.
- [35] W. Welch and A. Witkin. Variational surface modeling. *Computer Graphics*, 26, 1992.
- [36] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In *Proceedings of the 21st International SIGGRAPH Conference*, 1994.
- [37] J. S. Van Welij. Calculation of eddy currents in terms of H on hexahedra. *IEEE Transactions on Magnetics*, MAG-21(6):2239–2241, 1985.
- [38] Man-Fai Wong, O. Picon, and V. F. Hanna. A finite element method based on whitney forms to solve maxwell equations in the time domain. *IEEE Transactions on Magnetics*, MAG-31(3):1618–1621, 1995.

Distribution:

1	MS-0123	D. Chavez LDRD Office 1011
2	MS-0899	Technical Library 4536
2	MS-9018	Central Technical Files 8945-1
10	MS-0750	C. Weiss, Geophysics Dept 06116, MS-0750
10	MS-1110	D. Day, Computational Math Dept 01414