6
An Algorithm to Make Cuts for Magnetic Scalar Potentials

6A. Introduction and Outline

In this chapter we consider a general finite element-based algorithm to make cuts for magnetic scalar potentials and investigate how the topological complexity of the three-dimensional region, which constitutes the domain of computation, affects the computational complexity of the algorithm. The algorithm is based on standard finite element theory with an added computation required to deal with topological constraints brought on by a scalar potential in a multiply connected region. The process of assembling the finite element matrices is also modified in the sense described at length in the previous chapter.

Regardless of the topology of the region, an algorithm can be implemented with $O(m_0^3)$ time complexity and $O(m_0^2)$ storage where $m_0$ denotes the number of vertices in the finite element discretization. However, in practice this is not useful since for large meshes the cost of finding cuts would become the dominant factor in the magnetic field computation. In order to make cuts worthwhile for problems such as nonlinear or time-varying magnetostatics, or in cases of complicated topology such as braided, knotted, or linked conductor configurations, an implementation of $O(m_0^3)$ time complexity and $O(m_0)$ storage is regarded as ideal. The obstruction to ideal complexity is related to the structure of the fundamental group. This chapter describes an algorithm that can be implemented with $O(m_0^2)$ time complexity and $O(m_0^{4/3})$ storage complexity given no more topological data than that contained in the finite element connection matrix.

Electromagnetic Context and Numerical Motivation. The proper context for the algorithm of this chapter is in variational principles, the finite element
6. CUTS ALGORITHM

method, and their connection to the topology of the domain of computation. Before seeing how topology enters the picture when considering a scalar potential, recall that magnetoquasistics refers to the class of electromagnetics problems where the magnetic field is described by the following limiting case of Maxwell’s equations:

\[ \int_{\partial V} \mathbf{B} \cdot dS = 0, \]

\[ \int_{\partial S} \mathbf{H} \cdot dl = \int_{S} \mathbf{J} \cdot \mathbf{n} \, ds, \]

\[ \int_{\partial S'} \mathbf{E} \cdot dl = -\frac{d}{dt} \int_{S'} \mathbf{B} \cdot \mathbf{n} \, ds. \]

Here \( \partial V \) refers to the boundary of a region \( V \) and \( \partial S \) is the boundary of a surface \( S \). The displacement current \( \partial \mathbf{D} / \partial t \) in Ampère’s law is neglected and the current density vector \( \mathbf{J} \) is assumed to be solenoidal. The magnetic flux density vector \( \mathbf{B} \) is related to the magnetic field intensity \( \mathbf{H} \) by \( \mathbf{B} = \mathbf{B}(\mathbf{H}) \), or for linear isotropic media, \( \mathbf{B} = \mu \mathbf{H} \). The current density \( \mathbf{J} \) in conductors is related to the electric field intensity vector \( \mathbf{E} \) by Ohm’s law: \( \mathbf{J} = \sigma \mathbf{E} \). Let \( R \) denote a region which is the intersection of the region where it is desired to compute the magnetic field and where the current density \( \mathbf{J} \) is zero. In \( R \),

\[ \text{curl} \mathbf{H} = 0, \]

so that in terms of a scalar potential \( \psi \),

\[ \mathbf{H} = \text{grad} \psi \]

in any contractible subset of \( R \) but in general \( \psi \) is globally multivalued as seen via Ampère’s Law. This was illustrated in Figure 1.8 for a current-carrying trefoil knot \( c \) with current \( I \) and cut \( S \). In that case, if \( \mathbf{H} = \text{grad} \psi \), Ampère’s Law implies that a scalar potential \( \psi \) is multivalued as illustrated with loop \( c_1 \) where \( \int_{c_1} \mathbf{H} \cdot dl = I \) implies that \( \psi \) is multivalued when there is no cut. On the other hand, \( \int_{c_2} \mathbf{H} \cdot dl = I - I = 0 \), even though \( c_2 \) is not contractible to a point implying that \( c_2 \) gives no information about \( \psi \). With the cut in place as shown in the figure and a discontinuity \( I \) imposed on \( \psi \) from one side of the cut to the other, \( \psi \) is made to be single-valued on the cut complement. Note that Ampère’s law does not require that \( c_2 \) intersect the cut.

It is common practice to sidestep the multivalued scalar potential by expressing the magnetic field as \( \mathbf{H} = \mathbf{H}_p + \text{grad} \, \tilde{\psi} \), where \( \mathbf{H}_p \) is a particular solution for the field obtained, say, from the Biot–Savart Law. However, in this case, if \( \mathbf{B} = \mu \mathbf{H} \) and \( \mu \to \infty \), then \( \mathbf{H} \to 0 \) so that \( \mathbf{H}_p = -\text{grad} \, \psi \), leading to significant cancellation error for computation in regions where \( \mathbf{H} \approx 0 \) while a “total” scalar potential as in (6–1) does not suffer from cancellation error. In addition, integration of the Biot–Savart integral destroys the sparsity of any discretization. The particular solution, \( \mathbf{H}_p \), can be set to zero in a multiply connected region when the notion of a cut which makes the scalar potential single-valued is introduce. In practice use of cuts is viable if the software to generate cuts does not require
input from the user. These reasons are an incentive to developing an algorithm for automatic computation of cuts.

Outline. With the preceding motivation in mind, we outline the main sections of this chapter. Section 6B introduces the essential pieces of (co)homology theory and differential forms required for defining the notion of cuts and for finding an algorithm to compute them. Section 6C presents a variational formulation which can be used in the context of the finite element method. Section 6D fills in the piece missing from Section 6C, describing an algorithm for finding the topological constraints on the variational problem, and gives an analysis of the computational complexity of finding cuts. The overall process of computing cuts is summarized in algorithm 6.1 and the algorithm of Section 6E is summarized in algorithm 6.2. Two example problems are considered in order to illustrate the obstruction to $O(n)$ complexity. Section 6F concludes with a summary of the main results, a review of geometric insights used to reduce the complexity of the algorithm, and suggestions for future work.

6B. Topological and Variational Context

Preceding chapters have generously set the context for the algebraic structures and duality theorems needed in order to establish a general cuts algorithm. However, the following points regarding the relevance of these tools add further motivation to the purpose of this chapter. One advantage of formulating cuts in terms of cohomology groups is that when a constructive proof of the existence of cuts is phrased in terms of the formalism of a certain homology theory, the proof gives way to a variational formulation for a cuts algorithm. The proof is sketched here while some more details are given in Section MA-I. Another advantage is that various homology theories give several ways to view cuts. In particular, when a finite element mesh is viewed as a simplicial complex as in Chapter 4, simplicial homology theory provides a framework for implementing an algorithm to make cuts and determines appropriate data structures. Finally, since the homology groups can be computed with coefficients in $\mathbb{Z}$, an implementation of the algorithm uses only integer arithmetic, thus avoiding introduction of rounding errors associated with real arithmetic. This implies that rounding error analysis is required only for the well-understood parts of the algorithm which depend on standard finite element theory.

Before considering the algorithm, we summarize the relevant groups from earlier chapters. Let $R$ be the nonconducting region with boundary $\partial R$. The complement of $R$ relative to $\mathbb{R}^3$, denoted by $R^c$, is the union of the problem “exterior” and the conducting region. Recall that Ampère’s law is a statement about closed loops in $R$ which link nonzero current. In terms of homology, $H_1(R; \mathbb{Z})$ can be viewed as the group of equivalence classes of closed loops in $R$ which link closed paths in $R^c$ which may be current paths. Two closed loops in $R$ lie in the same equivalence class if together they comprise the boundary of a surface in $R$. As noted, the $\mathbb{Z}$ in $H_1(R; \mathbb{Z})$ expresses the fact that in this case there is no loss of information if one builds $H_1$ by taking only integer linear combinations of closed loops. As discussed in Section 3D, the homology groups
for $R$ are torsion-free so that integer coefficients are sufficient. The rank of $H_1(R)$, denoted by $\beta_1(R)$, is a characteristic parameter of $R$, significant because it describes the number of independent closed loops in $R$ which link nonzero current. As such it will characterize the number of variational problems to be solved in the cuts algorithm.

The first cohomology group of $R$, denoted by $H^1(R; \mathbb{Z})$ can be regarded as the group of curl-free vector fields in $R$ modulo vector fields which are the gradient of some function. As in the case of $H_1$, it is enough to take “forms with integer periods” meaning that integrating the field about a generator of $H_1$ gives an integer. The rank of $H^1(R; \mathbb{Z})$ is also $\beta_1(R)$. In three dimensions, $H_1(R)$ and $H^1(R)$ formalize Ampère’s law in the sense that, respectively, they are algebraic structures describing linking of current and irrotational fields in $R$ due to currents in $R^c$.

The second relative homology group $H_2(R, \partial R; \mathbb{Z})$, is the group of equivalence classes of surfaces in $R$ with boundary in $\partial R$. Classes in $H_2(R, \partial R; \mathbb{Z})$ are surfaces with boundary in $\partial R$ but which are not themselves boundaries of a volume in $R$. Its rank is the second Betti number, $\beta_2(R, \partial R)$. $H_2(R, \partial R; \mathbb{Z})$ is the quotient space of surfaces which are 2-cycles up to boundary in $\partial R$ modulo surfaces which are boundaries (Figure 6.1). In three dimensions these turn out to be surfaces used for flux linkage calculations. This will be shown precisely, but we need to start with a definition to get to the algorithm.

The reader should note that an essential requirement for a cut is that it must be a barrier to every loop which links a current in the sense of Ampère’s law. For this to occur, the boundaries of the cuts must be on the boundary $\partial R$ of the region in question. Thus we expect to have at least $\beta_1(R)$ cuts, one cut corresponding to each independent current. In fact there are $\beta_1(R)$ families of cuts where each family is an equivalence class of cuts associated with each current. This is a geometric and intuitive way of understanding what cuts are, but does not establish their existence or give a way of computing them. The duality theorems considered in Chapter 3 and the isomorphism described below provide the formalism required for showing existence and constructing an algorithm.

**The Isomorphism** $H^1(R; \mathbb{Z}) \simeq [R, S^1]$. The isomorphism discussed here allows us to restrict our attention to cuts which are embedded manifolds in $R$, and at the same time gives a way of computing these cuts through a variational problem.
The isomorphism establishes a relationship between $H^1(R; \mathbb{Z})$ and maps from $R$ to the circle $S^1$, where $S^1$ is regarded as the unit circle in the complex plane. Let $[R, S^1]$ denote the space of maps $f : R \rightarrow S^1$ up to the equivalence relation given by homotopy. It is the case that

$$H^1(R; \mathbb{Z}) \simeq [R, S^1],$$

which says that the group of cohomology classes of $R$ with integer periods is isomorphic to the group of homotopy classes of maps from $R$ to $S^1$ [Kot87].

The reason for introducing (6–2) is twofold: First, there is no guarantee that homology classes can be represented by manifolds (surfaces). As discussed below, Equation (6–2) provides this guarantee. Second, since it associates each class in $H^1$ to a class of maps from $R$ to $S^1$, a suitable choice of energy functional on $[R, S^1]$ will give a variational problem and an algorithm for computing cuts. Hence, this isomorphism and Poincaré–Lefschetz duality give way to an algorithm for cuts.

Concretely, choosing $\mu = d\theta/2\pi$, a closed, nonexact 1-form on $S^1$, then $f^*(\mu)$ is the “pullback” of $\mu$ via $f$ so that one can regard $f^*(\mu)$ as monitoring the change in $\theta$ as one covers a family of cuts in $R$. Poincaré–Lefschetz duality and the preimage theorem [GP74] say that for a regular point $p$ of $f$ on $S^1$,

$$\int_R \omega \wedge f^* \left( \frac{d\theta}{2\pi} \right) = \int_{f^{-1}(p)} \omega$$

for any closed 2-form $\omega$. Thus, given $f : R \rightarrow S^1$ where $f$ winds around $S^1$, the pullback $f^*(\mu)$ is the Poincaré–Lefschetz dual to $f^{-1}(p)$ [Kot87].

In terms of vector calculus, given a map $f : R \rightarrow S^1$, Equation (6–3) can be re-written as

$$\frac{1}{2\pi i} \int_R G \cdot \text{grad}(\ln f) \, dV = \int_{f^{-1}(p)} G \cdot n \, dS$$

for any $G \in \mathcal{G}$. When $G = \mu H$ and $H = 1/(2\pi i) \, \text{grad}(\ln f)$, each side of Equation (6–4) can be regarded as an expression for the energy of a system of (unit) currents in $\mathbb{R}^3 - R$. In particular, note that the right hand side is the integral over a cut of the magnetic flux due to a unit current.

**Variational Aspects of an Algorithm for Cuts.** This section considers a variational formulation for a cuts algorithm, the associated the Euler–Lagrange equation, and discusses the exact solutions of this variational problem while hinting at its topological flavor. Though not necessary for continuity to following sections, we employ the framework described above to find an explicit solution to the nonconvex variational problem and verify that the resulting function is single-valued.

The cuts algorithm consists of finding a solution to the variational problem of minimizing

$$F(f) = \int_V \text{grad} \tilde{f} \cdot \text{grad} f \, dV,$$
subject to
\begin{equation}
\tilde{f} f = 1 \quad \text{in } V,
\end{equation}
and for \(1 \leq j \leq \beta_1(V)\), the \(j\)th cut requires
\begin{equation}
\frac{1}{2\pi i} \oint_{c_k} \text{grad}(\ln f) \cdot dl = \delta_{jk}
\end{equation}
for \(1 \leq k \leq \beta_1(V)\). Here \(f\) is a map from \(V\) to \(\mathbb{C}\), and the \(c_l\), for \(1 \leq l \leq \beta_1(V)\), are curves representing a basis for \(H_1(V; \mathbb{Z})\). Equation (6–6) shows that the solution to the above problem defines a map to the unit circle in the complex plane
\begin{equation}
f : V \rightarrow S^1.
\end{equation}
Taking the inverse image of a regular value on \(S^1\) (i.e., a point \(p\) on \(S^1\) such that the gradient of \(f\) is nonzero at every point in the preimage), we end up with a surface whose boundary lies in \(\partial V\). This “cut surface” represents a relative homology class in \(H_2(V, \partial V; \mathbb{Z})\) which is dual to the constraint represented by (6–7). This comes about because \(S^1\) is an Eilenberg–MacLane space \(K(\mathbb{Z}, 1)\) as discussed in detail in Section MA-I.

Our immediate objective is to make a distinction between how the variational problem (6–5)–(6–7) is handled numerically and analytically. First we note that one can handle the constraint set forth in (6–6) by letting
\begin{equation}
f = e^{2\pi i \psi}
\end{equation}
where \(\psi\) is a real differentiable function which, by (6–7), must be multivalued. Substituting (6–9) into (6–5) gives
\begin{equation}
F(e^{2\pi i \psi}) = 4\pi^2 \int_V \text{grad } \psi \cdot \text{grad } \psi \, dV.
\end{equation}

The starting point for an algorithm is the observation that the Euler–Lagrange equation corresponding to (6–10) is just Laplace’s equation. Hence, in principle, an algorithm to find cuts is easily implemented once one can modify existing finite element code for solving Laplace’s equation. Two subtleties which must be addressed are, first, interelement continuity conditions must be modified in order to respect (6–7) and second, from Equation (6–8), the inverse image of \(f\) can be obtained by considering the equipotentials of \(\psi\) modulo integers. Addressing these two subtleties, an algorithm to find cuts in any region can be implemented provided a “reasonable” finite element mesh exists, that is, a mesh on which Laplace’s equation can be solved.

For a deeper understanding of situations where a complete set of cuts \(\{S_i\}\), \(1 \leq i \leq \beta_1(V)\), enable one to use a single-valued scalar potential in
\[\tilde{V} = V - \bigcup_{i=1}^{\beta_1(V)} S_i,\]
but \(\tilde{V}\) is not simply connected, we need a better understanding of the solution of the variational problem (6–5)–(6–7). To this end, we will now handle the constraint (6–6) by a Lagrange multiplier which can be eliminated to obtain a
“harmonic map equation” for $f$. When confronted with this nonlinear partial differential equation we will use what we have established regarding magnetic scalar potentials, the Biot–Savart law, and linking numbers, to produce an explicit solution.

If we append to the functional (6–5) a Lagrange multiplier term corresponding to the constraint (6–6), we end up with a variational problem for the functional

$$(6–11) \quad \tilde{F}(f, \lambda) = \int_V \text{grad} \tilde{f} \cdot \text{grad} f + \lambda(\tilde{f} f - 1) \, dV,$$

subject to the constraint (6–7). When the first variation of this functional with respect to $f$ is set to zero we obtain the weak Galerkin form

$$0 = \int_V ((\text{grad} \delta f) \cdot (\text{grad} \tilde{f}) + (\text{grad} \delta \tilde{f}) \cdot (\text{grad} f) + \lambda(f \delta \tilde{f} + \tilde{f} \delta f)) \, dV.$$

If we eliminate the derivatives of the variation of $f$ through integration by parts, we obtain

$$0 = \int_V (\delta f (-\text{div grad} \tilde{f} + f) + \delta \tilde{f} (-\text{div grad} f + \lambda f)) \, dV + \int_{\partial V} (\delta f \frac{\partial \tilde{f}}{\partial n} + \delta \tilde{f} \frac{\partial f}{\partial n}) \, dS.$$

Writing $f = f_r + i f_i$, where $f_r$ and $f_i$ are real functions that can be varied independently, one finds that the vanishing of the above expression for all admissible $\delta f$ implies

$$(6–12) \quad \text{div grad} f = \lambda f \quad \text{in } V,$$

$$\frac{\partial f}{\partial n} = 0 \quad \text{on } \partial V.$$

When the variation of the functional (6–11) with respect to $\lambda$ is set to zero, we recover the constraint (6–6). We begin to eliminate the Lagrange multiplier from Equation (6–12) by first taking the Laplacian of the constraint (6–6) to obtain

$$0 = \text{div grad}(\tilde{f} f) = (\text{div grad} \tilde{f}) f + 2 \text{ grad } \tilde{f} \cdot \text{grad } f + \tilde{f} \text{ div grad } f$$

or

$$(6–13) \quad \Re(\tilde{f} \text{ div grad } f) = -|\text{grad } f|^2,$$

where $\Re(\cdot)$ denotes the real part of $\cdot$. Multiplying Equation (6–12) by $\tilde{f}$ and using Equations (6–6) and (6–13) we can solve for $\lambda$:

$$-|\text{grad } f|^2 = \Re(\tilde{f} \text{ div grad } f) = \lambda \tilde{f} f = \lambda$$

, and rephrase (6–12) as

$$(6–14) \quad \text{div grad } f = -|\text{grad } f|^2 f \quad \text{in } V,$$

$$\frac{\partial f}{\partial n} = 0 \quad \text{on } \partial V.$$

Equations (6–14) and (6–7) provide a set of equations for the single-valued function defined in the discussion leading to Equation (6–8). At first sight, the
solution to these equations in the exterior of current carrying wires is not obvious. If we perform the substitution given by Equation (6–9) then (6–14) reduces to a boundary value problem involving Laplace’s equation and a multivalued function. In the algorithm for computing cuts it was necessary to deal with this multivaluedness in the context of interelement constraints — appealing to physical intuition would have lead to a circular argument where the magnetic scalar potential would be needed to define the cuts for the magnetic scalar potential! In the present case we want to develop our intuition about cuts and seek explicit expressions for the cuts. Hence we are free to use the equipotentials of the multivalued scalar potential tied to any easy-to-use constitutive law as equivalent cuts.

In order to find the function $f$ satisfying (6–6) and (6–14) we first find a set of integer-valued currents in the exterior of $V$ which insure that the corresponding scalar potential satisfies (6–7). Suppose this is accomplished by imposing $\beta_1(V)$ integer valued currents $\{n_i\}$ on $\beta_1(V)$ closed curves $\{c_i\}$ in the exterior of $V$. The scalar potential is given by the line integral of the magnetic field found through the use of the Biot–Savart Law:

\begin{equation}
\psi(p) - \psi(p_0) = \frac{1}{4\pi} \int_{p_0}^p \sum_{i=1}^{\beta_1(V)} n_i \oint_{c_i} \frac{[\mathbf{r} - \mathbf{r}'] \times d\mathbf{r}}{|\mathbf{r} - \mathbf{r}'|^3}.
\end{equation}

Note that we can modify this $\psi$ so that it satisfies a Neumann boundary condition on $\partial V$ by adding a single valued harmonic function which vanishes at $p_0$. From Equation (6–9), the desired $f$ is easily seen to be

\begin{equation}
f(r) = f_0 \exp\left(\frac{i}{2} \int_{p_0}^{p} \sum_{i=1}^{\beta_1(V)} n_i \oint_{c_i} \frac{[\mathbf{r} - \mathbf{r}'] \times d\mathbf{r}}{|\mathbf{r} - \mathbf{r}'|^3}\right),
\end{equation}

where the multiplicative factor

\begin{equation}f_0 = e^{2\pi i \psi(p_0)}\end{equation}

is an arbitrary complex number of unit length. If we go from $p_0$ to $p_0$ along a closed loop $c$ the expression for the linking number (3–13) shows us that the expression for $f$ changes by

\[\exp\left(2\pi i \sum_{i=1}^{\beta_1(V)} n_i \text{Link}(c, c_i')\right),\]

which is equal to 1. This shows that a concrete understanding of the linking number makes the single-valuedness of $f$ manifest and finally, up to boundary conditions, Equation (6–16) is indeed the solution to the variational problem defined by (6–5)–(6–7) and the boundary value problem defined by (6–14) and (6–7). Lefschetz duality and the theory underlying the algorithm ensure that cuts take care of current linkage, but nowhere has anything been done to make $V - \bigcup S_i$ simply connected.
**Computational Overhead Required for Cuts.** In order to evaluate the utility of having cuts, one must also compare computational overhead with the expected acceleration in solution time. Given a finite element mesh it is possible to compare the solution time complexities of the associated finite element matrix equations for scalar potential and vector methods. Since semidirect matrix solvers are commonly used for large problems, we argue in terms of a conjugate gradient (CG) solver iteration for the finite element matrix equations. Let \( F_{s0} \) denote the number of floating point operations (FLOPs) per conjugate gradient solver iteration for nodal interpolation of a scalar potential on a tetrahedral mesh. Similarly, let \( F_0 \) and \( F_1 \) respectively denote the number of FLOPs per CG iteration for node- and edge-based vector interpolation [Kot91]. Finally, let \( X_s \), \( X_0 \), and \( X_1 \) denote the number of nonzero entries in the stiffness matrix for nodal interpolation of a scalar potential, and node- and edge-based vector interpolation, respectively. Assuming similar distributions of eigenvalues in the matrices, the convergence of CG for each matrix is the same and the ratio of the complexity of the CG iterations is a reflection of the ratio of computer run times.

Table 6.1 summarizes how vector formulations compare to a scalar potential formulation based on analysis of simplices in a mesh discussed in Section 4C. The top line of the table is derived from Equations (4-17), (4-21), and (4-23), then using (4-18), forming ratios, and letting \( m_0 \to \infty \) in the ratios.

If a scalar potential can be computed, it provides a substantial speed-up, but the overhead is that of computing cuts. Hence, cuts may be most useful in the context of time-varying or nonlinear problems where cuts are computed only once but iterative solutions are required for the field.

Algorithm design must begin with the choice of an algebraic framework for the computation, and for reasons of computability, this is the most critical choice. The often-made assumption that cuts must render the region simply connected forces one to work with a structure called the first homotopy group for which basic questions related to this group are not known to be algorithmically decidable. In practical terms, homotopy-based algorithms are limited to problems reducible to a planar problem. Thus, their success depends on the fact that 2-d surfaces are completely classified (up to homeomorphism) by their Euler characteristic and numbers of connected components and boundary components. The (co)homology arguments presented here lead to a general definition of cuts and an algorithm for computing them by linear algebra techniques, but when using sparsity of the matrices to make the computation efficient, homotopy emerges as an important and useful tool.

<table>
<thead>
<tr>
<th></th>
<th>Node ((X_0 &amp; F_0))</th>
<th>Edge ((X_1 &amp; F_1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X/X_s)</td>
<td>9</td>
<td>7.5</td>
</tr>
<tr>
<td>(F/F_{s0})</td>
<td>7.5–7.6</td>
<td>9.7–10.7</td>
</tr>
</tbody>
</table>

Table 6.1. Relative number \((X)\) of nonzero entries in the stiffness matrix and number of floating point operations \((F)\) per CG iteration for node- and edge-based vector interpolation compared to scalar potential \((X_s\) and \(F_{s0}\)\) on large tetrahedral meshes.
6C. Variational Formulation of the Cuts Problem

On the basis of the tools introduced above, the computation of cuts can be formulated as a novel use of finite elements subject to two constraints imposed by the topology of \( R \). The idea is to come up with a variational problem for finding minimum energy maps \( f \) from classes in \([R, S^1]\). Hence, a principle for finding cuts is to compute a collection of maps, \( f_q : R \rightarrow S^1 \), \( 1 \leq q \leq \beta_1(R) \) which correspond to a basis of the first cohomology group with integer coefficients \( H^1(R; \mathbb{Z}) \) and by duality, to a basis for \( H_2(R, \partial R; \mathbb{Z}) \). Any map in the homotopy class can be used, but picking harmonic maps reduces the problem to quadratic functionals tractable by the finite element method. Furthermore, the level surfaces of these maps are nicer than in the generic case.

As a variational problem, finding cuts can be rephrased in the following manner. “Computing maps” means finding the minima to \( \beta_1(R) \) “energy” functionals

\[
F(f_q) = \int_R \nabla f_q \cdot \nabla f_q \, dR, \quad 1 \leq q \leq \beta_1(R),
\]

subject to two constraints:

\[
f_q f_q = 1 \quad \text{in } R
\]

and, for the \( j \)th cut, \( 1 \leq j \leq \beta_1(R) \),

\[
\frac{1}{2\pi i} \oint_{z_k} \nabla \ln f_j \cdot dl = P_{jk},
\]

where \( \{z_q\} \), \( 1 \leq q \leq \beta_1(R) \), defines a set of generators of \( H_1(R) \) and \( P_{jk} \) is the period of the \( j \)th 1-form on \( z_k \) is an entry of a nonsingular period matrix \( P \). Intuitively, one might require the period matrix to be the identity matrix, but this is overly restrictive for a practical implementation of the algorithm. In fact, as discussed in Section 6E, direct computation of a basis for \( H_1(R) \) is impractical but an equivalent criterion can be used to satisfy the constraint imposed by \( H_1(R) \).

The solution to each map in the variational problem is unique since the “angle” of each \( f_q \) is a (multivalued) harmonic function which is uniquely specified by Equation (6–20) [GK95]. When the functionals are minimized, a set of cuts is computed by the formula

\[
S_q = f_q^{-1}(p_q),
\]

where \( p_q \) is any regular value of \( f_q \), \( 1 \leq q \leq \beta_1(R) \). Note that \( S_q \) is the Poincaré–Lefschetz dual to \( d \ln f/2\pi i \), as seen in equation 6–3 [Kot87].

On any contractible subset of \( R \), constraint (6–19) is satisfied by letting

\[
f_q = \exp(2\pi i \varphi), \quad 1 \leq q \leq \beta_1,
\]

where \( \varphi \) is some real, locally single-valued, but globally multivalued, differentiable function. Choosing \( f_q \) this way, the Euler–Lagrange equation of (6–18) is Laplace’s equation [GK95].

Equation (6–22) is satisfied on open, contractible subsets \( U_i \) and their intersections. When the \( U_i \) form a cover of \( R \), the global problem is assembled by considering the combinatorics of intersections \( \bigcap U_i \) as noted in [BT82, §13]
and also described in [GR65] in the context of the Cousin problem in complex analysis. For a computer implementation using standard data structures, it is sufficient to take a (tetrahedral) discretization of $R$ and consider what happens across faces of the tetrahedral elements. Using the normalized angle of $f_q$ [Kot89a],

\begin{equation}
\theta_q = (\ln f_q / 2\pi i) \mod 1,
\end{equation}

on an element and interpolating $\theta_q$ linearly over each element, we must consider that (6–20) prevents $\theta_q$ from being globally well-defined.

Section 6E addresses how constraints (6–20) are handled without making explicit reference to a set of curves representing a basis for $H_1(R)$. However, we begin by considering constraint (6–19) and the finite element-based part of the algorithm in the next section. Since each of the $\beta_1$ problems is treated in the same way, in the next section we drop the subscript denoting the $q$th variational problem in order to simplify notation.

6D. The Connection Between Finite Elements and Cuts

Here we consider the variational problems for (any) one of the $\beta_1(R)$ maps $f_i$ representing a class in $H_2(R, \partial R)$ (by duality) and how the variational problem is handled via the finite element method. While the principles behind computing cuts are not dependent on the type of discretization used, this section is set in the context of first-order nodal variables on a tetrahedral finite element mesh as the data structures are simple.

**The Role of Finite Elements in a Cuts Algorithm.** Consider a tetrahedral discretization of $R$, denoted by $K$, with $m_3$ elements, $m_0$ nodes. The $i$th tetrahedron in $K$ is denoted by $\sigma^3_i$. Recalling Equations (6–22) and (6–23), let $\theta$ be discretized on each element by the set $\theta^i_j$, $1 \leq i \leq m_3$, $1 \leq j \leq 4$ for each one of the $\beta_1(R)$ variational problems. Here the subscript refers to the $j$th node of the $i$th tetrahedron and the $\theta^i_j$ on individual elements are defined modulo integers since we seek a map into the circle. Furthermore, constraints (6–20) require that there be discontinuities in $\theta^i_j$ between pairs of adjacent elements. This is not a problem since the finite element connection process is modified accordingly as described below. To make a bridge to the finite element method, we also let $u_k$, $1 \leq k \leq m_0$ denote a potential discretized on nodes of the mesh [Kot89a].

The usual finite element connection matrix is defined as

$$C_{ij} = \begin{cases} 
1 & \text{if global node } k \text{ is the } j \text{th local node in } \sigma^3_i, \\
0 & \text{otherwise.} 
\end{cases}$$

The modified connection process is the marriage, via the finite element connection matrix, of variables defined locally on the nodes of individual elements ($\theta^i_j$) to variables defined on the global node set of the mesh ($u_k$). Consequently, variables defined element-by-element are said to be on the unassembled mesh while those defined on the global nodes are on the assembled mesh.

The global constraints (6–20) are handled via nodal discontinuities on the unassembled mesh, $J^i_j$, $1 \leq i \leq m_3$, $1 \leq j \leq 4$ (Figure 6.2). The jumps are
speciﬁed so that for a given global node, \( J^i_j \) is an integer-valued jump relative to a \( J^k_l \) which is set to zero. From the perspective of a global node, there is a set of \( J^i_j \) associated with the global node (one \( J^i_j \) per element in which the global node is a local node), and one \( J^i_j \) per set can be set to zero. Then \( \theta^i_j \), which is defined on the unassembled mesh, is written

\[
\theta^i_j = \sum_{k=1}^{m_0} C^i_j k u_k + J^i_j, \quad 1 \leq i \leq m_3, \quad 1 \leq j \leq 4.
\]

For each of the \( \beta_1(R) \) variational problems, there is one set of variables \( J^i_j \).

To finish specifying \( \theta^i_j \), the relationship between sets \( J^i_j \) and the homology class of the corresponding cut is needed. Recall that the sets of local nodal jumps \( J^i_j \) are defined on the unassembled ﬁnite elements problem. Since the discontinuity in \( \theta \) must be consistent across the face of a tetrahedron, we introduce a set of discontinuities across faces, \( \zeta^i_l \), \( 1 \leq l \leq m_2 \) where \( m_2 \) is the number of faces in the mesh. These are illustrated in Figure 6.2. Since there are \( \beta_1 \) variational problems there is a set \( \{ \zeta^i_{l_1}, \ldots, \zeta^i_{l_k} \} \), and for the \( k \)th cut, given \( \zeta^i_k \), the remaining \( J^i_j \) can be found by the back substitution

\[
J^i_m - J^i_n = \zeta^i_{l_{ij}},
\]

when elements \( i \) and \( j \) share face \( l_{ij} \). The “topological computation” relating face jumps \( \zeta_k \) to the relative homology class of the \( k \)th cut is discussed in Section 6E. At this point, if one is interested in using a scalar potential, but not in making the cuts visible to the user as a diagnostic tool, the set of face jumps \( \{ \zeta_1, \ldots, \zeta_{\beta_1} \} \) is a set of cuts. However, note that these cuts are not embedded manifolds, but nonetheless represent a basis for \( H_2(R, \partial R; \mathbb{Z}) \). We will see in Section 6E that the face jumps can be identiﬁed with a class in the simplicial homology group related to \( H_2(R, \partial R) \).

On the basis of the variational problem deﬁned in equations (6–18)–(6–20), “ﬁnite element analysis” can be used to solve for each of \( \beta_1(R) \) potentials, \( u_k \). On each connected component of \( R \), begin by setting one (arbitrary) variable
in \( u_k \) to zero. Then define barycentric coordinates \( \lambda_i^l, 1 \leq i \leq 4 \), on the \( l \)th tetrahedron, \( \sigma_i^l \), and build the element stiffness matrix \( \mathcal{K}_{mn}^l \):

\[
\mathcal{K}_{mn}^l = \int_{\sigma_i^l} \lambda_m^l \cdot \lambda_n^l \, dV.
\]

Discretizing the normalized angle \( \theta \) \((6-23)\) on \( \sigma_i^3 \) by

\[
\theta^l = \sum_{i=1}^{4} \lambda_i^l \theta_i^l
\]

and substituting into the functional \((6-18)\) gives

\[
(6-27) \quad F(\theta) = 4 \pi^2 \sum_{l=1}^{m_3} \sum_{m=1}^{4} \sum_{n=1}^{4} \theta_l^m \theta_l^n \mathcal{K}_{mn}^l.
\]

Using Equation \((6-24)\) to “assemble the mesh” gives a quadratic form in \( u_k \). The minimum of the quadratic form is obtained by differentiating with respect to the \( u_k \), resulting in the matrix equation

\[
(6-28) \quad \sum_{i=1}^{m_0} \mathcal{K}_{ij} u_i = -f_j.
\]

Here \( \{ \mathcal{K}_{ij} \} \) forms the usual stiffness matrix,

\[
(6-29) \quad \mathcal{K}_{ij} = \left( \sum_{l=1}^{m_3} \sum_{m=1}^{4} \sum_{n=1}^{4} C_{nj}^l \mathcal{K}_{mn}^l C_{mi}^l \right),
\]

and, by Equation \((6-25)\), the source term

\[
(6-30) \quad f_j = \left( \sum_{l=1}^{m_3} \sum_{m=1}^{4} \sum_{n=1}^{4} \mathcal{J}_n^l \mathcal{K}_{mn}^l C_{mj}^l \right)
\]

is related to the homology class of a relative cycle in \( H_2(R, \partial R) \) by means of \( \{ \zeta_j \} \). Thus, with the exception of computing \( \{ \zeta_j \} \) and forming source term \((6-30)\), the algorithm is readily implemented in any finite element analysis program. This gives the maps from \( R \) to \( S^1 \). To find the cut, recall Poincaré–Lefschetz duality and Equation \((6-21)\). For each connected component of \( R \),

\[
(6-31) \quad (\theta')^l_j = C_{jk}^l (u_k + c) \ \text{mod} \ 1,
\]

where \( c \) is a constant so that \( \theta' = 0 \) is a regular value of \( f \). After solving \( \beta_1(R) \) variational problems, we proceed element by element to find and plot \( f_{\psi q}^{-1}(\theta' = 0), \ 1 \leq q \leq \beta_1(R) \), to obtain a set of cuts. This is done in an unambiguous way if the mesh is fine enough to ensure that, over an element, \( \theta' \) does not go more than one third of the way around the circle.

In order to use the cut for a scalar potential computation, the cut must be specified in terms of internal faces of the mesh, much as sets \( \zeta_i \) are defined. For this we define \( \beta_1 \) sets \( S_i \) of faces obtained by perturbing a level set of the harmonic map onto internal faces of the finite element mesh. On a tetrahedral mesh this is done by simply choosing the element face which is on the side of the
level set indicated by the normal (gradient direction) when the level set passes through the element. This is illustrated in Figure 6.3.

![Figure 6.3. Level set of harmonic map is perturbed in gradient direction onto a face of the tetrahedral element. The face is selected merely by virtue of the fact that the potentials $u(v_i \geq s)$ where $v_i$ is the $i$th vertex. In other words, the direction of $\nabla u$ defines a normal to the cut, but it is unnecessary to actually compute the gradient. Cases in which the level set is perturbed onto an edge or vertex of the tetrahedron can be ignored since the level set will perturb onto the face of a neighboring tetrahedron.](image)

Besides computing the $\{\zeta_1, \ldots, \zeta_{\beta_1}\}$ and incorporating the $S_{\alpha}$ as subcomplexes of the mesh, the computation of cuts makes use of standard software found in finite element software packages. Although the sign of a floating point calculation is required at one step, the “topological” part of the code is otherwise implemented with integer arithmetic and is therefore immune to rounding errors.

The following “algorithm” shows how cuts software fits into the typical finite element analysis process. The second and third steps are not “standard” finite element software, but are implemented with integer arithmetic and thus avoid introducing rounding error.

**Algorithm 6.1. Finite elements and cuts**
1. **Tetrahedral mesh generation and refinement**
2. **Extraction of simplicial complex:** Employ the data structures from Chapter 4 and generate the data needed for computing interelement constraints $\zeta_i$.
3. **Topological processing:** Compute interelement constraints $\{\zeta_1, \ldots, \zeta_{\beta_1}\}$ defined in Equation (6–25) and described in section 6E and algorithm 6.2.
4. **Finite element solution:** Use (6–29) and (6–30) to form (6–28) for each of $\beta_1$ variational problems, and solve them by the finite element method.
5. **Obtain cuts:** Level sets of the harmonic maps computed in the last step are cuts. When perturbed onto the 2-skeleton of the mesh to define $\beta_1$ surfaces $S_{\alpha}$, they are the data needed to do a magnetic scalar potential calculation.

**6E. Computation of 1-Cocycle Basis**

Now we consider the computation of the interelement constraints which come about as a result of constraint (6–20) on the variational problem and were defined
in Equation (6-25). The computation must be in terms of the finite element discretization \( K \), and as indicated at the beginning of Section 6B, simplicial (co)homology is the algebraic framework. Here we use the results of companion chapter [GK01a] which describes the simplicial complex, and how it generates the finite element data structures required for computing cuts when taking the view that \( K \) and its simplicial complex are the same thing.

Chapter 4 discusses the duality between the boundary homomorphism on a simplicial complex \( K \) and its dual complex \( \overline{K} \) and shows that the duality appears naturally in the data structures describing \( K \). This duality is a discrete version of Poincaré–Lefschetz duality introduced in Chapter 3, but stated at the level of the simplicial chain complex (the algebraic structure which is most suited to describing finite element meshes). The dual complex is also the most appropriate structure for describing the computation of the topological constraints for the variational problem. Below we give the basic definitions needed to formulate the computation of the topological constraints.

Definitions. For a tetrahedral mesh \( K \), denote the nodes, edges, faces, and tetrahedra as \( 0-, 1-, 2-, \) and \( 3\)-simplexes, respectively, though in general it is possible to have an \( n \)-dimensional simplicial mesh. The total number of \( p \)-simplexes in a mesh is denoted by \( m_p \). Each set of \( p \)-simplexes forms a linear space with (for the present purpose) coefficients in \( \mathbb{Z} \), \( C_p(K) \), called the \( p \)-chain group.

There is a boundary homomorphism \( \partial_p : C_p(K) \to C_{p-1}(K) \) which takes \( p \)-simplexes to \((p-1)\)-simplexes such that the composition of two successive transformations is zero:

\[
\partial_i \partial_{i+1} = 0, \quad 1 \leq i \leq n.
\]

In other words, \( \text{im} \partial_{i+1} \subseteq \ker \partial_i \), and the sequences in (6–32) are summarized in the simplicial chain complex:

\[
0 \to C_n(K) \xrightarrow{\partial_n} \cdots \xrightarrow{\partial_p} C_{p-1}(K) \xrightarrow{\partial_{p-1}} \cdots \xrightarrow{\partial_1} C_1(K) \xrightarrow{\partial_1} C_0(K) \to 0.
\]

As in the continuum case, this allows us to define homology (and cohomology) groups.

The adjoint operator of the boundary homomorphism is the coboundary operator \( \partial^T_{p+1} : C^p(K) \to C^{p+1}(K) \) where \( C^p(K) \) is the simplicial cochain group of functionals on \( p \)-chains; formally, \( C^p(K) = \text{hom}(C_p(K), \mathbb{Z}) \). The cochain \( c^p \in C^p \) satisfies the relation

\[
\langle c^p, \partial^T_{p+1} c^p+1 \rangle = \langle \partial^T_{p+1} c^p, c^p+1 \rangle
\]

for any \( c^p+1 \in C^p+1 \). This is a discrete version of Stokes’ theorem (compare Equations (3–2) and (3–3)) and serves as a definition of \( \partial^T_{p+1} \). \( \partial^T_i \partial^T_{i+1} = 0 \) so that there is a cochain complex:

\[
0 \to C^n(K) \xrightarrow{\partial^T_n} \cdots \xrightarrow{\partial^T_p} C^{p-1}(K) \xrightarrow{\partial^T_{p-1}} \cdots \xleftarrow{\partial^T_1} C^1(K) \xleftarrow{\partial^T} C^0(K) \to 0.
\]

Simplicial homology groups are quotient groups \( H_p(K) = \ker \partial_p / \text{im} \partial_{p+1} \) and the \( p \)th Betti number \( \beta_p \) is the rank of \( H_p \): the simplicial cohomology groups are
defined by \( H^p(K) = \ker \partial^T_{p+1} / \im \partial^T_p \). An element of \( H^p(K) \) is the coset
\[
\zeta + B^p,
\]
where \( B^p(K) = \im \partial^T_p \) is the \( p \)-coboundary subgroup of \( C^p(K) \) and \( \zeta \in \ker \partial^T_{p+1} \) is a \( p \)-cocycle. The ranks of the homology and cohomology groups are related:
\[
\text{Rank}(H_p) = \text{Rank}(H^p).
\]

By identifying nonboundary \( p \)-simplexes in \( K \) with \((n-p)\)-cells (so that when \( n = 3 \), 3-simplexes become nodes, 2-simplexes become edges, etc.), a formal dual of \( K \), called the dual complex \( DK \), can be formed directly from the connection matrix describing \( K \). (The boundary is excluded since there are no 3-simplexes in \( \partial K \).) The dual \( DK \) is not simplicial but cellular, and the number of \( p \)-cells of \( DK \) is denoted by \( m_p \). \( DK \) is a cellular complex in the sense of (6–33). Poincaré duality amounts to saying that the coboundary operators of the simplicial complex are dual to the boundary homomorphisms of the cellular complex, denoted by \( \partial_p \) in the sense that
\[
\partial^T_{p+1} = \partial_n \]
\[\text{GK01a}\]. Thus, for a 3-dimensional complex,
\[
0 \longrightarrow C_3(DK) \xrightarrow{\partial_3} C_2(DK) \xrightarrow{\partial_2} C_1(DK) \xrightarrow{\partial_1} C_0(DK) \longrightarrow 0
\]
and
\[
0 \longrightarrow C^3(DK) \xleftarrow{\partial^*_3} C^2(DK) \xleftarrow{\partial^*_2} C^1(DK) \xleftarrow{\partial^*_1} C^0(DK) \longleftarrow 0.
\]
The (co)homology of \( DK \) is isomorphic to the (co)homology of \( K \) in complementary dimensions. In other words, Poincaré duality on the (co)chain level gives us the Poincaré duality of Section 6B.

**Formulation of a 1-co cycle generator set.** The duality between boundary and coboundary operators as set forth above is useful for formulating the outstanding problem of computing \( \{\zeta_1, \ldots, \zeta_\beta \} \) introduced in equation (6–25). These variables were introduced in order to handle interelement topological constraints (6–20) of the variational problem, but Equation (6–20) cannot be applied directly since a set of generators for \( H_1(R) \) is generally not known beforehand and is hard to compute. On the other hand, (6–20) simply gives the periods of 1-cocycles integrated on a homology basis, so that it is enough to know a basis for the nontrivial (i.e. noncoboundary) 1-cocycles. Here, the \( \zeta_i \) are described by \( \beta_1(R) \) 1-cocycles which are generators for classes in \( H^1(DK; \mathbb{Z}) \), and by duality represent sets of faces having nonzero jumps in backsubstitution Equation (6–25). The advantage of formulating the problem in terms of \( H^1(DK; \mathbb{Z}) \) is that it immediately yields a matrix equation, and the 1-cocycles form \( \beta_1(R) \) source terms for the right-hand side of Equation (6–28).

In general, equivalence classes in \( H^1(DK; \mathbb{Z}) \), the first cohomology group of the dual complex with integer coefficients, can be represented by integer-valued 1-forms which are functionals on 1-cells of the dual mesh. However, it is only possible (and necessary) to compute, for each equivalence class, a generating 1-cocycle defined by two properties described below.
First, to obtain a basis for noncobounding 1-cocycles a basis for the 1-coboundary subgroup $B^1(DK; \mathbb{Z})$ must be fixed by constructing the image of a 0-coboundary map $\partial^T_1 : C^0 \rightarrow C^1$. This not only fixes how $B^1(DK; \mathbb{Z})$ is represented, but allows the computation of 1-cocycles which represent closed, nonexact 1-forms.

Second, by definition, the 1-cocycle must also satisfy the condition that on the boundary of each 2-cell in $DK$, $\partial_2(2\text{-cell}) = \varepsilon_1 e_1 + \varepsilon_2 e_2 + \cdots + \varepsilon_n e_n$,

\begin{equation}
\langle \zeta_j, \partial_2(2\text{-cell}) \rangle = \sum_{i=1}^{n} \varepsilon_i \zeta_j^i (e_i) = 0,
\end{equation}

where $\varepsilon_i = \pm 1$ denotes the orientation of the $i$th 1-cell on the boundary of the 2-cell (Figure 6.4). The condition must be satisfied on any simply connected subset of the mesh, but the data readily available from the finite element connection matrix relates to 2-cells. Since Poincaré duality for complexes $K$ and $DK$ says that $\partial_2$ and $\partial^T_2$ are identified, the coboundary operator $\partial^T_2$ is the incidence matrix of 2- and 1-cells in $DK$ and contains the data for the 1-cocycle conditions over all of $DK$. Equations (6–34) and (6–39) together say that, for any 1-cocycle $\zeta$,

\begin{equation}
\partial^T_2 \zeta = 0
\end{equation}

(on $DK$). Once a basis for the 1-coboundary subgroup has been fixed, a set of nontrivial 1-cocycles is found by computing a basis of the nullspace of the operator $\partial^T_2$. Equation (6–40) is an exceedingly underdetermined system, but as shown below, fixing a basis for $B^1$ induces a block partition of the matrix, and reduces the computation to a block whose nullspace rank is precisely $\beta_1(R)$.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{diagram.png}
\caption{1-cells on maximal spanning tree (solid) and 2-cells in $DK$. On the right, the 2-cell cocycle condition has four free variables (those not on the tree) while on the left, the condition can be satisfied (trivially) in terms of variables on the tree.}
\end{figure}

In summary there are two conditions which must be satisfied in order to find a set of 1-cocycles which are not coboundaries:

1. A basis for $B^1$ must be fixed by considering the image of a map

\[ \partial^T_1 : C^0 \rightarrow C^1. \]
(2) The 1-cocycles must independently satisfy the 1-cocycle condition on each 2-cell of the mesh.

**Structure of the Matrix Equation for Computing the 1-Cocycle Generators.** The strategy outlined above amounts to constructing bases for $\text{im} \, \partial_T^1$ and $\text{ker} \, \partial_T^2$ (subject to $\text{im} \, \partial_T^1 = 0$) in the complex (6–38). When $\text{im} \, \partial_T^1$ is annihilated, the surviving piece of $\text{ker} \, \partial_T^2$ gives a basis for 1-cohomology generators. In this section we give a method for constructing the required bases while retaining the sparsity of $\partial_T^2$ and show how the construction yields a natural block partition of the matrix.

The arguments of this section have the same motivation as techniques of electrical circuit analysis. The rank argument of this section is a formalization of a familiar equation which relates the number of free variables $n_{\text{free}}$ to the numbers of Kirchhoff current law (node) and Kirchhoff voltage law (loop) equations ($n_{\text{KCL}}$ and $n_{\text{KVL}}$):

$$n_{\text{KCL}} + n_{\text{KVL}} - \beta_0 = n_{\text{free}},$$

where $\beta_0 = \text{Rank}(H_0)$ is the number of connected components of the mesh or circuit.

To fix $B^1(DK)$, we construct a map $\partial_T^1$ satisfying the Stokes Equation (6–34), which for this case says $(c^0, \partial c^0) = (\partial^T c^0, e)$, where $e \in C_1(DK)$, $c^0 \in C^0(DK)$. Defining $c^0$ on vertices of $DK$ and building a maximal tree on the 1-skeleton of $DK$ fixes a basis for $B^1 = \text{im}(\partial_T^1)$ up to a constant on a single vertex on each connected component of $DK$ by associating each vertex (functional) with an edge (functional) on the 1-skeleton. There are $m_0 - \beta_0$ 1-cells on the maximal tree, the same as the rank of $\text{im} \, \partial_T^1$. Since the coboundary subgroup is annihilated in the equivalence relation for cohomology, the variables of $\zeta_i$ corresponding to edges on the maximal tree can be set to zero. Below we see that this reduces the number of free variables enough to permit computation of an appropriate set of independent nullvectors of $\partial_T^2$.

The reduction of free variables for each 1-cocycle solution $\zeta_i$ obtained by the maximal tree induces the following partition on $\partial_T^2$:

$$\partial_T^2 \zeta_i = \begin{pmatrix} \begin{array}{c|c} T & U \\ m_0 - \beta_0 & m_1 - m_0 + \beta_0 \end{array} \end{pmatrix} \begin{pmatrix} 0_r \\ \zeta_U \end{pmatrix} = 0,$$

where columns of block $U$ correspond to 1-cells not on the tree while columns of block $T$ correspond to 1-cells on the tree. Variables in $\zeta_i$ which correspond to 1-cells on the tree are zero, so that there are $\bar{m}_1 - \bar{m}_0 + \beta_0(DK)$ free variables remaining for any nontrivial 1-cocycle (or nullspace) solution to the matrix equation. The following shows that the dimension of the nullspace of block $U$ is $\beta_1(R)$.

The rank of $\partial_T^2$ can be found by a standard argument which considers the ranks of the kernel and image of the boundary homomorphism in the cellular complex (6–37) and ranks of the corresponding homology groups. Since $\partial_2$ is a linear map,

$$\dim \text{im} \partial_2 = \dim C_2 - \dim \ker \partial_2 = \bar{m}_2 - \dim \ker \partial_2.$$
In terms of the rank of the second homology group,

\[(6-43) \quad \dim \ker \tilde{\partial}_2 = \beta_2 + \dim \im \tilde{\partial}_3 = \beta_2 + \tilde{m}_3 - \beta_3,\]

where \(\dim \tilde{\partial}_3 = \tilde{m}_3 - \beta_3\) follows from (6.37). In this case, since \(K\) is the triangulation of a connected 3-manifold with boundary, \(\beta_3 = 0\). In any case, (6-43) and (6-42) give the rank of \(\partial_T^2\):

\[
\dim \im \tilde{\partial}_2 = \tilde{m}_2 - \tilde{m}_3 - \beta_2 + \beta_3.
\]

In terms of cocycle conditions, this result can be interpreted as counting the number of linearly independent cocycle conditions in rows of \(\tilde{\partial}_2\). Considering the set of cocycle conditions on a 3-cell, there is one linearly dependent cocycle condition, giving \(m_3\) extra cocycle conditions in \(\partial_T^2\). There is one linearly dependent equation among each set of cocycle conditions describing “cavities” of the region, giving another \(2\beta_2\) linearly dependent equations.

Consequently, the dimension of the nullspace of block \(U, \mathcal{N}(U)\), in the partition of Equation (6-41) is

\[
\dim \mathcal{N}(U) = (\tilde{m}_1 - \tilde{m}_0 + \beta_0) - (\tilde{m}_2 - \tilde{m}_3 - \beta_2 + \beta_3) = -\chi(DK) + \beta_2 + \beta_0 - \beta_3 = \beta_1,
\]

since the Euler characteristic satisfies

\[
\chi(DK) = \sum_{i=0}^{n} (-1)^i \beta_i = \sum_{i=0}^{n} (-1)^i \tilde{m}_i.
\]

Accounting for \(\tilde{m}_3 + \beta_2 - \beta_3\) linearly dependent cocycle conditions, the following partition of \(U\) into blocks of linearly independent \((U_i)\) and linearly dependent \((U_d)\) equations is a useful picture to keep in mind for the rank argument:

\[
U = \begin{pmatrix}
U_i \\
\tilde{m}_2 - \tilde{m}_3 - \beta_2 + \beta_3 \\
U_d \\
\tilde{m}_3 + \beta_2 - \beta_3
\end{pmatrix}.
\]

In practice, the linear dependence of rows in \(U\) can be exploited when finding a diagonalization of \(U\) so that the nullspace basis \(\{\zeta_1, \ldots, \zeta_{\beta_1}\}\) is relatively sparse.

**Sparsity of \(\partial_T^2\) and \(U\).** Recall that nonboundary 2-simplexes in \(K\) are mapped to 1-cells in \(DK\) and nonboundary 1-simplexes in \(K\) are mapped to 2-cells in \(DK\). In \(K\), the boundary of every 2-simplex has three 1-simplexes so that in \(DK\) each 1-cell is in at most three 2-cells. The inequality comes about because \(\partial K\) does not enter into the contraction of \(DK\); in particular, a 2-simplex with some of its boundary in \(\partial K\) corresponds to a 1-cell which is an edge in fewer than three 2-cells. Consequently, columns of \(\partial_T^2\) have at most three nonzero entries, or \(3\tilde{m}_1\) is an upper bound on the number of nonzero entries in \(\partial_T^2\).

A lower (upper) bound on the difference between \(3\tilde{m}_1\) and the number of nonzero entries is given by \((b - 2)n_1\) where \(n_1\) is the number of 1-simplexes in \(\partial K\) and \(b\) is an upper (lower) bound on the number of 2-simplexes which meet at a boundary 1-simplex. In the estimate we take two less than \(b\) since the two faces meeting at a boundary edge do not have entries in \(U\).
Block Partition and Sparsity of the Matrix Equation. At this point we are free to choose any method for finding a basis for the nullspace of $U$. Typical methods for matrices with integer coefficients are the Smith and Hermite normal form algorithms [Coh93]. Since $\partial_2^T$ is an incidence matrix with nonzero entries $\pm 1$, problems such as pivot selection can be avoided, but they also destroy the sparsity of $\partial_2^T$ and their time complexity is $O(m_0^3)$. This indicates that the combinatorial structure of the matrix is more important than its numerical structure. The literature on computing sparse nullspace bases of real matrices is applicable here [CEG86, PF90].

$U$ can be block partitioned into a form which preserves most of its sparsity. The partition is based on the observation that a 2-cell Equation (6–39) which has only one free variable after fixing im $\partial_2^T$ is satisfied trivially — variables for such 1-cells do not contribute to the 1-cocycles and can be set to zero in $\zeta_i$. In terms of the maximal tree, this case corresponds to Figure 6.4. In matrix $\partial_2^T$, this elimination amounts to forward substitution of variables on the tree, forming a lower triangular block in $U$ and eliminating variables which are not essential to the description of the null basis while avoiding zero fill-in. When the process of forward substitution halts (as it must if the null space basis we seek is nontrivial), the remaining free variables and cocycle conditions contain a full description of the complex on a substantially smaller set of generators and relations. This results in the following block partition of the matrix equation, where $U_{11}$ is the lower triangular block resulting from the forward substitution:

$$\partial_2^T = \begin{pmatrix} T & U_{11} & 0 \\ U_{21} & U_{22} \end{pmatrix}.$$ 

Block $T$ corresponds to a maximal tree on the 1-cells of $DK$ and, variables associated with $T$ are zero in the nullspace basis. Forward substitution of nullspace basis variables on $T$ gives the lower triangular block $U_{11}$ so that the nullspace basis vectors have the form

$$\zeta_i = \begin{pmatrix} 0_T \\ 0_{U_{11}} \\ \zeta_{U_{22},i} \end{pmatrix}.$$ 

As with $\partial_2^T$, block $U_{22}$ has at most three nonzero entries per column since no operations involve zero fill-in. Figure 6.6 shows examples of $U_{22}$ for two interesting cases. The first example is the Borromean rings and the second example is the trefoil knot, both shown in Figure 6.5.

At this point it is best to admit that there are two ideas from topology which have strong ties to the present construction. One of them is Poincaré’s algorithm for computing the generators and relations of the fundamental group of a complex [Sti93]. This construction is similar with the added constraint of preserving sparsity of the equations and reduction of the Poincaré data into block $U_{22}$ of $\partial_2^T$. Another relevant notion is that of the spine of a manifold [Thu97].

**Algorithm 6.2.** Algorithm for 1-cocycle generator set

1. **Initialize:** Set $\{\zeta_1, \ldots, \zeta_{\delta_1}\}$ to be zero.
Figure 6.5. Borromean rings and trefoil knot. The Borromean rings are three rings which pairwise have zero linking number but are inseparable.

(2) **Maximal Tree**: Construct maximal spanning tree $T$ on 1-skeleton of $DK$.
(3) **Partition**: Set $\zeta_i|_T = 0$ and partition $\partial_2^T$ as in (6-41).
(4) **Forward Substitution**: Forward substitute variables $\zeta_i|_T$ (for all $\zeta_i$ these are the same variables) through $U$ iteratively until the process halts.
(5) **$U_{22}$ Nullbasis**: Compute nullspace basis of $U_{22}$ by a sparse null basis technique or by computing the Smith normal form.

The size of $U_{22}$. The process of partitioning $\partial_2^T$ effectively retracts all the information about the topology of the mesh onto a 2-subcomplex of the mesh. The tree gives a retraction onto the 2-skeleton of $K - \partial K$, and the reduction by forward substitution is a retraction onto a 2-subcomplex $\tilde{K}$ represented by $U_{22}$. In the dual mesh, the retraction, $DK$, has the same “homotopy type” as $DK$ and hence the same (co)homology groups. For a sufficiently good maximal spanning tree (one that is, in some sense, short and fat), the number of 1-cells in $U_{22}$ is the number of faces (of $K$) on $S'$ a set of cuts plus additional surfaces which make any noncontractible loop on a cut contractible. Let $N$ be some measure of the number of degrees of freedom per unit length in $DK$ so that $m_0$ is $O(N^3)$. Note that $\tilde{m}_1$ is linearly related to $m_0$. Let $k$ be the number of 1-cells in $U_{22}$, namely the number of free variables remaining in the reduced matrix. As the mesh is refined, $k$ is on the order of the area of $S'$, that is $O(N^2)$ or $O(m_0^{2/3})$. The complexity of an algorithm to compute the nullspace basis is $O(m_0^2) + O(k^3)$ in time and $O(m_0) + O(k^2)$ in storage, where $k$ is $O(m_0^{2/3})$, so the time complexity becomes $O(m_0^2)$ and space complexity is $O(m_0^{4/3})$. The overall time requirement for computing cuts is that of finding $\{\zeta_1, \ldots, \zeta_{\beta_1}\}$ for each cut and $\beta_1$ solutions of Laplace’s equation to find the nodal potential described in Section 6C.
6. CUTS ALGORITHM

<table>
<thead>
<tr>
<th>a. BORROMEOAN RINGS</th>
<th>b. TREFOIL KNOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0(R)$</td>
<td>1</td>
</tr>
<tr>
<td>$\beta_1(R)$</td>
<td>3</td>
</tr>
<tr>
<td>$m_0$</td>
<td>9665</td>
</tr>
<tr>
<td>$m_3 (= \bar{m}_0)$</td>
<td>48463</td>
</tr>
<tr>
<td>$\bar{m}_1 (= m_2 - n_2)$</td>
<td>93877</td>
</tr>
<tr>
<td>$m_2 (= m_1 - n_1)$</td>
<td>52029</td>
</tr>
<tr>
<td>$m_3 (= m_0 - n_0)$</td>
<td>6614</td>
</tr>
<tr>
<td>$U_{22}$</td>
<td>$4008 \times 2888$</td>
</tr>
<tr>
<td>$nz(U_{22})$</td>
<td>8156</td>
</tr>
</tbody>
</table>

**Figure 6.6.** $U_{22}$ for two cases of interesting topology: (a) complement of Borromean rings (three unlinked but inseparable rings) and (b) complement of a trefoil knot.

6F. Summary and Conclusions

While Ampère’s law gives intuition about the role and nature of cuts, it sheds no light on their construction and computation. On the other hand, the algebraic structures of (co)homology theory are adequate for formulation of an algorithm for finding cuts on finite element meshes which are orientable, embedded submanifolds of the nonconducting region. The algorithm fits naturally into finite element theory.
Starting with the connection matrix, cuts can always be found in $O(m_0^3)$ time and $O(m_0^5)$ storage. However, complexity can be improved to $O(m_0^5)$ time and $O(m_0^{4/3})$ storage by the algorithm outlined above. Moreover, the algorithm discussed in Section 6E preserves sparsity in the finite element matrices and thus does not adversely affect complexity in subsequent computation of a scalar potential with cuts. The speed of the algorithm can be further improved if one starts with a coarse mesh or information about the fundamental group $\pi_1$ and its commutators [Sti93]. It is clear that in the context of adaptive mesh refinement, cuts should be computed on a coarse mesh and then refined with the mesh since even the most coarse mesh contains all the information required for the topological computation. On the other hand, since the topological computation involves only integer arithmetic, computation on a fine mesh does not introduce rounding error.

"The program itself is the only complete description of what the program will do."

P. J. Davis