Winding Numbers on Discrete Surfaces (Supplemental Material)

This supplement provides detailed pseudocode (Section A) for the *surface winding number (SWN)* method of Feng *et al.* 2023, and discusses the homological perspective on this method (Section B).

A PSEUDOCODE

Our pseudocode is expressed via a halfedge mesh data structure encoding a triangle mesh $M = (V, E, F)$, and use $\overrightarrow{ij}$ to denote the halfedge from i to j. Subroutines not defined here are described in the list below; many correspond to standard libraries/data structures.

- IsManifold(M, i) — returns true if i is a manifold vertex of M.
- IsBoundary(M, ij) — returns true if ij is a boundary edge of M.
- Orientation($M, \overrightarrow{ij}$) — returns true if the orientation of halfedge $\overrightarrow{ij}$ matches the canonical orientation of edge ij in M, and false otherwise.
- Twin($M, \overrightarrow{ij}$) — returns the twin of halfedge $\overrightarrow{ij}$ in M.
- Prev($M, \overrightarrow{ij}$) — returns the previous halfedge in the face containing halfedge $\overrightarrow{ij}$ of M.
- OppositeVertex($M, \overrightarrow{ij}$) — returns the vertex k opposite $\overrightarrow{ij}$ in face ijk of M.
- CornersOf(M, i) — returns the set of corners $_i^{jk}$ incident on vertex i of M.
- EndpointsOf(M, Γ) — returns the set $V \setminus V^*$ of vertices comprising the interior endpoints of a discrete 1-chain Γ on M.
- InteriorVertices(M, Γ) — returns the set of vertices which are not interior endpoints of the discrete 1-chain Γ on M.
- OutgoingHalfedgeOnCurve(M, i, Γ) — for a vertex i in M, returns an arbitrary halfedge $\overrightarrow{ij}$ whose tail is i, such that $\Gamma_{ij} \neq 0$. If i is a boundary vertex, instead return the most clockwise halfedge.
- SolvePositiveSemidefinite(A, b) — solves a sparse positive semidefinite linear system Ax = b, returning x (and picking an arbitrary shift if A has constants in its null space).
- SolveLinearProgram($M, \ell, \Gamma, \varepsilon, s$) — solves the linear program in Equation 11, for a mesh M with edge lengths ℓ, curve Γ, parameter ε and shifts s.

Algorithm 1 SurfaceWindingNumber($M, \ell, \theta, \Gamma, \varepsilon$)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$, on a mesh $M = (V, E, F)$ with edge lengths ℓ, corner angles θ, and a parameter ε for the linear program.

Output: The winding number function w defined on corners of M (Section 3.5).

$c \leftarrow$ ComputeReducedCoordinates(M, Γ) ▷§2.3.1
$u \leftarrow$ SolveJumpEquation(M, θ, Γ, c) ▷§2.4.3, §3.2
$\omega \leftarrow$ DarbouxDerivative(M, Γ, u) ▷§2.4.2
$\gamma \leftarrow$ HarmonicPart(M, θ, ω) ▷§3.3
$\mathring{v} \leftarrow$ IntegrateLocally(M, γ)
$s \leftarrow$ ComputeRelativeJumps($M, \mathring{v}$)
$\sigma \leftarrow$ SolveLinearProgram($M, \ell, \Gamma, \varepsilon, s$) ▷§3.4
$v \leftarrow$ RecoverSolution($M, \mathring{v}, \sigma$)
$\tilde{c} \leftarrow$ SubtractJumpDerivative(M, Γ, v, c) ▷§2.4.1
$w \leftarrow$ SolveJumpEquation($M, \theta, \Gamma, \tilde{c}$) ▷§3.5
return w

Algorithm 2 ComputeReducedCoordinates(M, Γ)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$ on a mesh $M = (V, E, F)$.

Output: A function $c \in \mathbb{Z}^{|C|}$ expressing values at corners relative to some reference value (Section 2.3.1).

$c \leftarrow 0^{|C|}$
for $i \in$ InteriorVertices(M, Γ) **do**
  if IsManifold(M, i) = False **then continue**
  $\overrightarrow{ij_0} \leftarrow$ OutgoingHalfedgeOnCurve(M, i, Γ)
  $\overrightarrow{ij} \leftarrow \overrightarrow{ij_0}$
  sum $\leftarrow 0$
  do
    if IsBoundary(M, ij) = False **then**
      $k \leftarrow$ OppositeVertex($M, \overrightarrow{ij}$)
      jump $\leftarrow$ Orientation($M, \overrightarrow{ij}$) ? $\Gamma_{ij} : -\Gamma_{ij}$
      sum += jump
      $c_i^{jk} \leftarrow$ sum
    $\overrightarrow{ij} \leftarrow$ Twin(M, Prev($M, \overrightarrow{ij}$)) ▷next outgoing halfedge
  while $\overrightarrow{ij} \neq \overrightarrow{ij_0}$
return c

Algorithm 3 SolveJumpEquation(M, θ, Γ, c)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$ on a mesh $M = (V, E, F)$ with corner angles θ, and reduced coordinates $c \in \mathbb{R}^{|C|}$.

Output: A function $u \in \mathbb{R}^{|C|}$ defined on corners of M, where u solves Equation 10. Values at corners adjacent to endpoints of Γ are left undefined, to be interpolated using Equation 4.

$L \leftarrow$ BuildLaplacian(M, θ, Γ)
$b \leftarrow$ BuildJumpLaplaceRHS(M, θ, Γ, c)
$u_0 \leftarrow$ SolvePositiveSemidefinite(L, b)
$u \leftarrow 0 \in \mathbb{R}^{|C|}$ ▷Apply shifts to recover u (Section 3.2).
for $_i^{jk} \in C$ **do** $u_i^{jk} \leftarrow u_0 + c_i^{jk}$
return u

Algorithm 4 DarbouxDerivative(M, Γ, u)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$, and a function $u \in \mathbb{R}^{|C|}$ with integer jumps across edges of a mesh $M = (V, E, F)$.

Output: The Darboux derivative $\omega \in \mathbb{R}^{|E|}$ of u, as a discrete 1-form on edges of M (Section 2.4.2).

$\omega \leftarrow 0 \in \mathbb{R}^{|E|}$
for $ij \in E$ **do**
  if $i \in$ EndpointsOf(M, Γ) or $j \in$ EndpointsOf(M, Γ) **then**
    continue
  $k \leftarrow$ OppositeVertex($M, \overrightarrow{ij}$)
  $\omega_{ij} \leftarrow u_j^{ki} - u_i^{jk}$
return ω

Algorithm 5 BuildLaplacian(M, θ, Γ)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$ on a mesh $M = (V, E, F)$ with corner angles θ.

Output: The operator $L \in \mathbb{R}^{|V^*|\times|V^*|}$ of Equation 10.

$L \leftarrow 0 \in \mathbb{R}^{|V^*|\times|V^*|}$ ▷*initialize empty sparse matrix*
for $pqr \in F$ **do**
  for $ijk \in C(pqr)$ **do** ▷*C: circular shifts*
    if $i \in$ EndpointsOf(M, Γ) or $j \in$ EndpointsOf(M, Γ) **then**
      continue
    $L_{ii},\ L_{jj}$ += $\frac{1}{2}\cot(\theta_k^{ij})$
    $L_{ij},\ L_{ji}$ −= $\frac{1}{2}\cot(\theta_k^{ij})$
return L

Algorithm 6 BuildJumpLaplaceRHS(M, θ, Γ, c)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$ on a mesh $M = (V, E, F)$ with corner angles θ, and reduced coordinates $c \in \mathbb{R}^{|C|}$ (Section 2.3.1).
Output: The vector $b \in \mathbb{R}^{|V^*|}$ in Equation 10.

$b \leftarrow 0 \in \mathbb{R}^{|V^*|}$
for $i \in$ InteriorVertices(M, Γ) **do**
  for ${}_i^{jk} \in$ CornersOf(M, i) and $j, k \notin$ EndpointsOf(M, Γ) **do**
    b_i −= $\frac{1}{2}\cot(\theta_k^{ij}) \cdot c_i^{jk}$
    b_j += $\frac{1}{2}\cot(\theta_k^{ij}) \cdot c_i^{jk}$
    b_i −= $\frac{1}{2}\cot(\theta_j^{ki}) \cdot c_i^{jk}$
    b_k += $\frac{1}{2}\cot(\theta_j^{ki}) \cdot c_i^{jk}$
return b

Algorithm 7 HarmonicComponent(M, θ, ω)

Input: A co-closed 1-form $\omega \in \mathbb{R}^{|E|}$ on a mesh $M = (V, E, F)$ with corner angles θ.
Output: A harmonic 1-form $\gamma \in \mathbb{R}^{|E|}$.

$d_1 \leftarrow$ BuildOneFormExteriorDerivative(M)
$*_1 \leftarrow$ BuildOneFormHodgeStar(M, θ)
$\widetilde{\beta} \leftarrow$ SolvePositiveSemidefinite$(d_1 *_1^{-1} d_1^T, d_1\omega)$
$\delta\beta \leftarrow *_1^{-1} d_1^T \widetilde{\beta}$
$\gamma \leftarrow \omega - \delta\beta$
return γ

Algorithm 8 SubtractJumpDerivative(M, Γ, v, c)

Input: A 1-chain $\Gamma \in \mathbb{Z}^{|E|}$ on a mesh $M = (V, E, F)$, residual function $v \in \mathbb{R}^{|C|}$, and reduced coordinates $c \in \mathbb{R}^{|C|}$ associated with Γ.
Output: Updated reduced coordinates $\tilde{c}$ encoding new jump constraints for the jump Laplace equation (Section 3.5).

for $i \in$ InteriorVertices(M, Γ) **do**
  if IsManifold(M, i) = False **then** continue
  for ${}_i^{jk} \in$ CornersOf(M, i) and $j, k \notin$ EndpointsOf(M, Γ) **do**
    if IsBoundary(M, ij) **then** continue
    $\ell \leftarrow$ OppositeVertex$(M, \text{Twin}(M, \overrightarrow{ij}))$
    $\tilde{c}_i^{jk} = c_i^{jk} - (v_i^{jk} - v_i^{\ell j})$
return $\tilde{c}$

Algorithm 9 BuildOneFormExteriorDerivative(M)

Input: A mesh $M = (V, E, F)$.
Output: A sparse matrix $d_1 \in \mathbb{Z}^{|F|\times|E|}$ representing the discrete exterior derivative on 1-forms.

$d_1 \leftarrow 0 \in \mathbb{Z}^{|F|\times|E|}$ ▷*initialize empty sparse matrix*
for $pqr \in F$ **do**
  for $ijk \in C(pqr)$ **do** ▷*C: circular shifts*
    $(d_1)_{pqr,ij} \leftarrow$ Orientation$(M, \overrightarrow{ij})$? 1 : −1
return d_1

Algorithm 10 BuildOneFormHodgeStar(M, θ)

Input: A mesh $M = (V, E, F)$ with corner angles θ.
Output: A sparse diagonal matrix $*_1 \in \mathbb{Z}^{|F|\times|E|}$ representing the Hodge star acting on discrete 1-forms.

$*_1 \leftarrow 0 \in \mathbb{Z}^{|E|\times|E|}$ ▷*initialize empty sparse matrix*
for $pqr \in F$ **do**
  for $ijk \in C(pqr)$ **do** ▷*C: circular shifts*
    $(*_1)_{ij,ij}$ += $\frac{1}{2}\cot\theta_k^{ij}$
return $*_1$

Algorithm 11 IntegrateLocally(M, γ)

Input: A harmonic 1-form $\gamma \in \mathbb{R}^{|E|}$ on a mesh $M = (V, E, F)$.
Output: Corner values $\mathring{v}_i^{jk}$ integrating γ in each triangle of M.

for $ijk \in F$ **do**
  $g_{\overrightarrow{ij}} \leftarrow$ Orientation$(M, \overrightarrow{ij})$? $\gamma_{ij} : -\gamma_{ij}$
  $g_{\overrightarrow{jk}} \leftarrow$ Orientation$(M, \overrightarrow{jk})$? $\gamma_{jk} : -\gamma_{jk}$
  $\mathring{v}_i^{jk} \leftarrow 0$
  $\mathring{v}_j^{ki} \leftarrow g_{\overrightarrow{ij}}$
  $\mathring{v}_k^{ij} \leftarrow g_{\overrightarrow{ij}} + g_{\overrightarrow{jk}}$
return $\mathring{v}$

Algorithm 12 ComputeRelativeJumps$(M, \mathring{v})$

Input: A value $\mathring{v}_i^{jk}$ per corner of a mesh $M = (V, E, F)$.
Output: Values $s \in \mathbb{R}^{|E|}$ that give the jump between locally integrated values across each edge of M.

$s \leftarrow 0 \in \mathbb{R}^{|E|}$ ▷*initialize zero vector*
for $ij \in E$ and IsBoundary(M, ij) = False **do**
  $s_{ij} \leftarrow \mathring{v}_i^{jk} - \mathring{v}_i^{lj}$
return s

Algorithm 13 RecoverSolution$(M, \mathring{v}, \sigma)$

Input: A value $\mathring{v}_i^{jk}$ per corner of a mesh $M = (V, E, F)$, and per-triangle shifts $\sigma \in \mathbb{R}^{|F|}$.
Output: A value v_i^{jk} per corner describing the residual function.

for ${}_i^{jk} \in C$ **do** $v_i^{jk} \leftarrow \mathring{v}_i^{jk} + \sigma_{ijk}$
return v

B HOMOLOGICAL PERSPECTIVE

Here we discuss the homological perspective on SWN, starting with the case of closed, oriented surfaces (B.1) before proceeding to surfaces with boundary (B.2) and nonorientable surfaces (B.3). The basic tools are the first homology group $H_1(M)$ and cohomology group $H^1(M)$ of the surface M, which provide dual descriptions of its topology. Throughout we assume that M is manifold: while we find that SWN works on nonmanifold meshes in practice, the duality theorems formally apply only to manifolds.

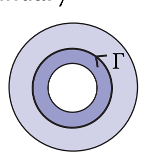
B.1 Overview of the Homological Picture

Homology is the theory of boundaries. A closed curve Γ on a surface M is said to be *nullhomologous* if Γ is the boundary of a region. Conversely, the homology group $H_1(M)$ describes loops which are not region boundaries. Munkres [1984, Chapters 1 & 5] gives a detailed introduction to homology and the dual theory of cohomology.

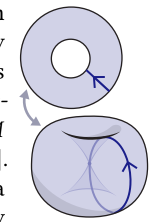
The connection to SWN is simplest when Γ is a closed curve on a closed, oriented surface M. In this case, the 1-form $\gamma = \mathcal{D}u$ computed in Section 3.3—known as the *Poincaré dual* of Γ—encodes Γ's homology class. Formally, Poincaré duality provides a canonical isomorphism $\varphi : H_1(M) \longrightarrow H^1(M)$ [Munkres 1984, §65]. Concretely, this map provides a harmonic 1-form $\varphi(\Gamma)$ such that for any loop η, the integral $\int_\eta \gamma$ counts the signed number of intersections between η and Γ [Griffiths and Harris 2014, p.56]. Two closed curves Γ_1 and Γ_2 map to the same harmonic 1-form if and only if the curves are homologous. Hence, any jump harmonic function integrating $\gamma = \varphi(\Gamma)$—*e.g.* the function v in Section 3.4—must jump across a chain homologous to Γ. Consequently, the linear program used to compute v minimizes the ℓ^1 norm of the jump $g = \mathcal{J}v$ subject to the constraint that g is homologous to Γ. In this case, one could avoid cohomology and directly solve an optimal homologous chain problem *à la* Dey et al. [2010]. However, harmonic 1-forms are essential in our generalization to broken curves.

When Γ is broken, it lacks a well-defined homology class. Consequently, the 1-form $\mathcal{D}u$ is no longer harmonic for broken curves. Nonetheless, we can take the harmonic component γ of $\mathcal{D}u$, which we interpret as an "approximate homology class" for Γ. SWN then searches for the optimal nonbounding loop $g = Jv$ within this homology class. Among other things, the homology class constraint ensures that g is always a closed loop, even when Γ is broken.

B.2 Relative Homology for Surfaces with Boundary

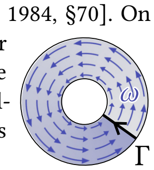


To make sense of our algorithm on surfaces with boundary—and in particular to justify Equation 12—we need to extend the discussion of homology to include *relative homology*. When M has no boundary, nullhomologous curves are precisely the curves enclosing regions, and nonbounding loops are characterized by the usual *absolute homology group* $H_1(M)$. However, the situation is more complicated if M has a boundary. For instance, an annulus has a single homology generator: a loop Γ wrapping around the middle. Though Γ separates the annulus into two components, it is not itself the boundary of any region since each component's boundary also includes a circle from the annulus' boundary.



Instead, nonbounding loops on a surface with boundary are described by the relative homology group $H_1(M, \partial M)$. On an annulus, *e.g.*, this group is generated by a curve connecting the boundary circles. Formally, it is the first homology group of M after collapsing ∂M to a point [Munkres 1984, §9]. *E.g.* collapsing the boundary of the annulus yields a sphere with two points identified, whose homology generator corresponds to the nonbounding curve on the annulus.

Relative Cohomology. Similarly, a surface with boundary has both absolute and relative cohomology groups. The absolute group $H^1(M)$ consists of harmonic 1-forms tangent to the boundary, while the relative group $H^1(M, \partial M)$ consists of harmonic 1-forms normal to the boundary [Poelke and Polthier 2016]. *Lefschetz duality* provides a map between $H_1(M, \partial M)$ and $H^1(M)$ [Munkres 1984, §70]. On an annulus, *e.g.*, the relative homology generator maps to a 1-form circulating around the center. Since nonbounding loops correspond to the relative homology group, our dual harmonic 1-forms are members of $H^1(M)$ and must thus lie tangent to ∂M.



Hodge Decomposition. On manifolds with boundary, one can decompose a k-form ω using the Hodge-Friedrichs-Morrey decomposition [Schwarz 2006, Corollary 2.4.9]:

$$\Omega^k = d\Omega_D^{k-1} \oplus \delta\Omega_N^{k+1} \oplus \left(\mathcal{H}^k \cap d\Omega^{k-1}\right) \oplus \mathcal{H}_N^k \tag{12}$$

$$= d\Omega_D^{k-1} \oplus \delta\Omega_N^{k+1} \oplus \left(\mathcal{H}^k \cap \delta\Omega^{k+1}\right) \oplus \mathcal{H}_D^k \tag{13}$$

Here a subscript D (for Dirichlet) denotes a space of forms with zero tangential component on ∂M, a subscript N (for Neumann) denotes a space of forms with zero normal component on ∂M, and $\mathcal{H}$ denotes the space of harmonic fields, (*i.e.* k-forms satisfying $d\gamma = \delta\gamma = 0$).

To extract a tangential harmonic 1-form, we apply the first decomposition (Equation 12). Multiplying both sides by d and δ yields a pair of equations determining the tangential harmonic component of a 1-form ω. A short calculation shows that these equations are the standard equations solved to perform Hodge decomposition on closed surfaces with zero-Neumann conditions on the boundary.

B.3 Local Coefficients for Nonorientable Surfaces

As discussed in Section 3.7, our algorithm also extends to nonorientable surfaces so long as one explicitly provides curve normals which specify which direction the surface winding number should jump across the curve. Such a choice of normals makes Γ into an element of the first homology group with local coefficients in the sense of Hatcher [2002, Section 3.H], which is Poincaré dual to the ordinary first cohomology group [Hatcher 2002, Theorem 3H.6].

Hodge Decomposition. Our discussion of Hodge decomposition used the codifferential $\delta := *d*$, which may look ill-defined on nonorientable surfaces: the definition uses $*$ which depends on the orientation of M. However, reversing orientation multiplies $*$ by -1, so because δ uses $*$ twice the signs cancel and δ remains well-defined on nonorientable surfaces. Hence, Hodge decomposition still works via the usual linear systems.