Reverse-Engineering Salvagnini STREAMBEND CAM Logic: A Comprehensive Mathematical and Topological Framework for Automated Bend Classification

1. Executive Summary

This report serves as the foundational technical specification for Phase 2 of the reverse-engineering initiative targeting the Salvagnini STREAMBEND CAM logic. The central objective is the development of a robust algorithmic framework capable of autonomously classifying sheet metal bends as either **BEND_UP (Positive)** or **BEND_DOWN (Negative)**. This classification is the primary determinant for the kinematic instructions sent to Salvagnini Panel Benders (e.g., P2, P4), machines that distinguish themselves from traditional press brakes through their ability to manipulate independent upper and lower bending blades without reorienting the workpiece.[1]

The research establishes that the distinction between "Up" and "Down" is not merely a geometric attribute but a kinematic relationship between the workpiece's current orientation—specifically the face secured by the automatic blankholder (Base Face)—and the target flange.[3] To automate this classification within a Computer-Aided Manufacturing (CAM) kernel, we leverage **Vector Algebra** for spatial reasoning and **OpenCASCADE Technology (OCCT)** for topological interrogation.

Key findings indicate that relying solely on local edge convexity is insufficient due to topological inconsistencies in imported CAD data (e.g., STEP/IGES format variations). Instead, a global **Signed Distance Function (SDF)** approach, which projects the centroid of the target flange onto the normal vector of the base face, provides the highest fidelity classification.[5] This method is robust against geometric noise and aligns directly with the physical operation of the machine: a flange residing in the positive half-space of the base sheet requires an upward stroke (Lower Blade), while a flange in the negative half-space requires a downward stroke (Upper Blade).[1]

Crucially, the report addresses the "Orientation Problem" inherent in Boundary Representation (B-Rep) modeling. The geometric normal of a surface in OpenCASCADE does not always align with the material exterior. We derive a rigorous method for calculating the "Material Normal" by synthesizing the parametric surface derivatives with the

TopAbs_Orientation flag.[8] Furthermore, we incorporate **Rodrigues' Rotation Formula** to mathematically align arbitrary base faces with the global Z-axis, simplifying the classification logic to a 1-dimensional coordinate comparison.[10]

This document details the mathematical derivations, algorithmic logic, OpenCASCADE implementation strategies, and edge-case handling required to produce a production-grade CAM solution.

---

2. The Salvagnini Kinematic Paradigm and Manufacturing Context

To accurately reverse-engineer the logic driving a Salvagnini Panel Bender, it is essential to first deconstruct the physical reality that the software attempts to model. The mathematical definitions of "Up" and "Down" are derived directly from the machine's constraints and capabilities.

2.1 The Evolution from Press Brake to Panel Bender

Traditional sheet metal forming relies predominantly on the Press Brake. In this paradigm, a ram (punch) descends into a die. The physics of the press brake dictate that the bend is always formed by forcing the material *into* the die. To create a reverse bend (e.g., a Z-profile), the operator or a robotic arm must physically flip the sheet 180 degrees. The CAM logic for press brakes focuses on "back gauge positioning" and "ram depth."

The Salvagnini Panel Bender (P2, P4, P1 variants) fundamentally alters this kinematic relationship. The sheet is placed horizontally on a worktable and clamped by an automatic blankholder. A manipulator moves the sheet in the X-Y plane to position the bend line.[3] The bending unit consists of two independent blades:

1. **Upper Blade:** Actuates downwards to fold the material down.
2. **Lower Blade:** Actuates upwards to fold the material up.

This architecture eliminates the need for flipping the part. Consequently, the CAM software must parse the 3D geometry to determine not just the *angle* of the bend, but the *direction* of the fold relative to the clamped face. If the CAM software misclassifies a "Down" bend as "Up," the machine will attempt to fold the flange through the blankholder, resulting in a catastrophic collision.

2.2 Semantic Definitions of Bend Polarity

The terms "Positive" and "Negative" are standard nomenclature in Salvagnini programming.[2] These terms map directly to the Cartesian coordinate system of the machine's bending unit, but they must be relative to the *part's* current orientation.

- **Base Face (F_{base}):** The planar face of the sheet metal part that is currently gripped by the blankholder. In the digital model, this serves as the local reference plane ($z = 0$).
- **BEND_UP (Positive):** A folding operation where the flange moves in the $+Z$ direction relative to the Base Face. Geometrically, the flange resides in the half-space defined by the positive normal of the Base Face. This requires the **Lower Blade**.
- **BEND_DOWN (Negative):** A folding operation where the flange moves in the $-Z$ direction relative to the Base Face. Geometrically, the flange resides in the half-space opposite the positive normal. This requires the **Upper Blade**.

This distinction highlights the necessity of a rigorous vector-based approach. Visual inspection is insufficient; the algorithm must mathematically verify which side of the plane the material volume occupies.

---

3. Mathematical Foundations of Computational Geometry

The core of the classification logic relies on Vector Algebra and Analytic Geometry. This section establishes the fundamental theorems and derivations used to construct the classification algorithm.

3.1 Vectors, Normals, and Coordinate Systems

A vector v in $\mathbb{R}^3$ is defined by its components $\langle v_x, v_y, v_z \rangle$. For our purposes, the most critical vector is the **Surface Normal**.

3.1.1 The Surface Normal

For a planar face, the normal vector $\hat{n}$ is a unit vector ($||\hat{n}|| = 1$) that is perpendicular to the surface at every point.[13] The normal defines the orientation of the plane. If a plane is defined by the equation $Ax + By + Cz + D = 0$, the normal vector is $n = \langle A, B, C \rangle$.

3.1.2 The Dot Product and Scalar Projection

The **Dot Product** (or scalar product) of two vectors a and b is defined as:

$$a \cdot b = ||a|| \, ||b|| \cos \theta$$

where θ is the angle between the vectors.[14]

Algebraically, it is the sum of the products of their components:

$$a \cdot b = a_x b_x + a_y b_y + a_z b_z$$

Significance for Bend Classification:

The dot product provides a mechanism to measure "alignment." If $\hat{n}$ is a unit normal vector, the dot product $v \cdot \hat{n}$ represents the **Scalar Projection** of v onto the direction of $\hat{n}$.

- If $v \cdot \hat{n} > 0$: v points generally in the same direction as $\hat{n}$ (Angle $\theta < 90^\circ$).
- If $v \cdot \hat{n} < 0$: v points generally in the opposite direction (Angle $\theta > 90^\circ$).
- If $v \cdot \hat{n} = 0$: The vectors are orthogonal (Angle $\theta = 90^\circ$).[16]

This property is the mathematical engine behind the **Signed Distance Function** used in our primary classification algorithm.

3.2 Defining the Plane and Half-Spaces

A plane divides 3D space into two regions: the **Positive Half-Space** (the side the normal points to) and the **Negative Half-Space**.

Let the Base Face be represented by a plane Π passing through a point P_0 (the centroid) with normal $\hat{n}$. The signed distance D of any arbitrary point P_{test} from this plane is derived from the scalar projection of the vector $r = P_{test} - P_0$ onto $\hat{n}$ [5]:

$$D = (P_{test} - P_0) \cdot \hat{n}$$

- **Interpretation of D:**
  - D: The perpendicular distance from the point to the plane.
  - **Sign of D:** Determines the side. Positive is "Above/Up", Negative is "Below/Down".[18]

This mathematical model aligns perfectly with the Salvagnini kinematics. If we set P_{test} to be the centroid of the target flange, the sign of D explicitly tells us whether to BEND_UP or BEND_DOWN.

3.3 Rotational Alignment via Rodrigues' Formula

In many CAM applications, it is beneficial to transform the local coordinate system of the part such that the Base Face aligns with the global XY plane (Normal aligns with Z). This simplifies visualization and subsequent calculations (where Z coordinate directly equals height).

Rodrigues' Rotation Formula provides an efficient method to compute the rotation matrix that aligns a source vector $\hat{a}$ (Base Normal) with a destination vector $\hat{b}$ (Global Z $\langle 0, 0, 1 \rangle$).[10]

Derivation:

1. **Axis of Rotation (k):** The vector perpendicular to both $\hat{a}$ and $\hat{b}$ is found via the cross product:

$$k = \hat{a} \times \hat{b}$$

   Normalize to unit vector $\hat{u} = \frac{k}{||k||}$.

2. **Angle of Rotation (θ):** The angle between them is found via the dot product:

$$\theta = \arccos(\hat{a} \cdot \hat{b})$$

3. **Rotation Matrix ($\mathbf{R}$):**

$$\mathbf{R} = \mathbf{I} + (\sin\theta)[\hat{u}]_\times + (1 - \cos\theta)[\hat{u}]_\times^2$$

   Where $[\hat{u}]_\times$ is the skew-symmetric cross-product matrix of $\hat{u}$:

$$[\hat{u}]_\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

Application: By multiplying the centroid coordinates of all faces by $\mathbf{R}$, we effectively "flatten" the base face to the horizontal. The Z-coordinate of the flange centroid in this transformed space becomes a direct proxy for the bend direction.[11]

4. OpenCASCADE Topological Architecture

To implement the mathematical theories described above, one must navigate the specific data structures of the OpenCASCADE Technology (OCCT) library. OCCT is the industry standard for open-source B-Rep modeling, but it introduces specific complexities regarding orientation and topology traversal that must be handled correctly.

4.1 The B-Rep Hierarchy

OCCT uses a boundary representation hierarchy defined in the TopoDS package. The relevant classes for this project are:

- **TopoDS_Shape:** The base class for all topological objects.
- **TopoDS_Solid:** Represents the entire volume of the sheet metal part.
- **TopoDS_Shell:** The collection of faces bounding the solid.
- **TopoDS_Face:** A topological surface. This is the primary unit of analysis for identifying the "Base" and "Flange".[22]
- **TopoDS_Edge:** A curve bounded by vertices. Used to identify connectivity between faces.[24]
- **TopoDS_Wire:** A loop of edges defining the boundary of a face.

4.2 The Orientation Problem: Geometric vs. Topological Normals

One of the most critical aspects of using OCCT for manufacturing logic is understanding the TopAbs_Orientation flag. A TopoDS_Face is composed of a reference to a Geom_Surface (Geometry) and an orientation (Topology).

- **Geometric Normal (N_{geom}):** Intrinsic to the parametric surface $S(u, v)$. Calculated as $\frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}$.
- **Topological Orientation:** A flag that can be TopAbs_FORWARD or TopAbs_REVERSED.
  - **FORWARD:** The material of the solid lies on the side of the surface opposite to the geometric normal. (i.e., The normal points "Out").
  - **REVERSED:** The material lies on the same side as the geometric normal. (i.e., The normal points "In").

Crucial Insight: If a face has TopAbs_REVERSED orientation, the geometric normal vector retrieved from GeomLProp_SLProps or BRepGProp will point *into* the part. Using this vector for the "Up/Down" calculation will invert the result, causing a collision on the machine.

The Correct Material Normal (N_{mat}):

The algorithm must always compute the effective material normal:

$$N_{mat} = \begin{cases} N_{geom} & \text{if Orientation} == \text{FORWARD} \\ -N_{geom} & \text{if Orientation} == \text{REVERSED} \end{cases}$$

Note: In some OCCT versions/contexts, the convention is that FORWARD means the natural normal points *out*. The definitive check is to ensure the vector points *away* from the solid volume.[8]

4.3 Surface Properties and Traversal

To extract the data needed for vector algebra, OCCT provides specific tools:

- **BRepGProp:** Computes global properties like Volume, Area, and Center of Mass. We use this to find the centroid (P_0 and P_{flange}).[26]
- **BRepAdaptor_Surface:** Provides a unified interface to access surface geometry (Plane, Cylinder, NURBS). Used to validate that the Base Face is indeed planar.
- **TopExp_Explorer:** The standard iterator for traversing topology (e.g., "Find all Faces in this Solid" or "Find all Edges in this Face").
- **TopTools_IndexedDataMapOfShapeListOfShape:** Essential for building an adjacency graph (finding which faces share an edge).

4.4 Feature Recognition: Adjacency Graphs

Before classifying a bend, we must identify the "Bend Feature" cluster. A bend is not a single entity in B-Rep; it is a topological relationship:

$$\text{Face}_{Base} \xleftrightarrow{\text{Edge}_1} \text{Face}_{Bend} \xleftrightarrow{\text{Edge}_2} \text{Face}_{Flange}$$

Typically, Face_{Base} and Face_{Flange} are planar, while Face_{Bend} is cylindrical (or a b-spline approximation of a cylinder). The algorithm must traverse this chain to pair the Base with its corresponding Flange.[8]

---

5. Algorithmic Framework for Bend Classification

This section synthesizes the mathematical and topological concepts into a coherent algorithmic workflow. We present three potential methods, evaluating their strengths and weaknesses, and recommending the **Signed Distance Function (SDF)** as the primary solution.

5.1 Method A: The Signed Distance Function (SDF)

This method is the direct implementation of the "Half-Space" test derived in Section 3.2. It is the most robust method for determining "Up" vs "Down" because it relies on the global spatial position of the flange mass, which is less sensitive to local edge noise than derivative-based methods.[5]

5.1.1 Algorithm Steps

1. **Input:** TopoDS_Face Base, TopoDS_Face Flange.
2. **Base Analysis:**
   - Calculate Centroid P_{base} using BRepGProp::SurfaceProperties.
   - Calculate Geometric Normal N_{geom} at the centroid using BRepLProp_SLProps.
   - Check Base.Orientation(). If REVERSED, invert N_{geom} to get N_{mat}.
3. **Flange Analysis:**
   - Calculate Centroid P_{flange} using BRepGProp::SurfaceProperties.
4. **Vector Construction:**
   - Construct the difference vector: $V_{disp} = P_{flange} - P_{base}$.
5. **Projection (The Classifier):**
   - Calculate Signed Distance $D = V_{disp} \cdot N_{mat}$.
6. **Decision Logic:**
   - **If $D > \epsilon$:** The flange is in the positive half-space. **Result: BEND_UP**.
   - **If $D < -\epsilon$:** The flange is in the negative half-space. **Result: BEND_DOWN**.
   - **If $|D| \leq \epsilon$:** The flange is coplanar (Hem or Flat).

5.1.2 Numerical Example

- **Base:** Center $(0,0,0)$, Normal $(0,0,1)$ [Up].
- **Flange (Up Bend):** Center $(10,0,10)$.
  - $V_{disp} = \langle 10,0,10 \rangle$.
  - $D = \langle 10,0,10 \rangle \cdot \langle 0,0,1 \rangle = 10$.
  - $10 > 0 \rightarrow$ **BEND_UP**.
- **Flange (Down Bend):** Center $(10,0,-10)$.
  - $V_{disp} = \langle 10,0,-10 \rangle$.
  - $D = \langle 10,0,-10 \rangle \cdot \langle 0,0,1 \rangle = -10$.
  - $-10 < 0 \rightarrow$ **BEND_DOWN**.

This method holds true even for acute (30°) or obtuse (150°) bends. As long as the flange has not wrapped fully around (360°) or remained flat (0°), the centroid will lie in the respective half-space.[29]

5.2 Method B: Dihedral Angle and Edge Convexity

This method analyzes the local transition between the Base Face and the Bend Face (the cylinder). It determines if the edge is "Convex" or "Concave."

5.2.1 Mathematical Logic

The convexity of an edge between two faces can be determined by the cross product of their normals compared to the edge direction vector.[25]

$$C = N_{Base} \times N_{Bend}$$

If C aligns with the edge vector E (Dot product positive), the turn is "Left" (or Up, depending on chirality). If anti-parallel, it is "Right" (Down).

5.2.2 Limitations

While mathematically sound for mesh geometry, this is risky in B-Rep for two reasons:

1. **Edge Orientation Ambiguity:** In OCCT, a TopoDS_Edge can be oriented FORWARD or REVERSED within the TopoDS_Wire of the face. Extracting the "true" flow of the edge requires parsing the wire loop, which adds computational overhead and complexity.
2. **Tangency Noise:** At the precise junction of a plane and a tangent cylinder, the normals are technically parallel. The cross product $N \times N$ is zero vector 0. One must sample slightly *away* from the edge, introducing arbitrary parameters (e.g., "sample at $u + 0.1$").

Therefore, Method B is recommended only as a secondary check if Method A returns an inconclusive result (e.g., $|D| \approx 0$).

5.3 Method C: Normal Vector Alignment (Angle Check)

This method calculates the angle between the Base Normal and the Flange Normal.

$$\alpha = \arccos(N_{base} \cdot N_{flange})$$

- **Utility:** This is excellent for calculating the **Bend Angle** (e.g., is it 90° or 45° ?).
- **Deficiency:** It is ambiguous regarding direction for 90° bends.
  - Up Bend Normal: $\langle 1, 0, 0 \rangle$ (assuming X-bend).
  - Down Bend Normal: $\langle -1, 0, 0 \rangle$ (or $\langle 1, 0, 0 \rangle$ depending on if we look at the "top" or "bottom" surface of the flange). Without tracking which surface of the flange corresponds to the "top" surface of the base, the normal vector direction alone is insufficient to distinguish Up from Down reliably.[25]

---

6. Advanced Vector Algebra and Edge Case Handling

The idealized geometric world often breaks down when dealing with real-world CAD files. This section addresses advanced scenarios and the "fuzzy" nature of imported geometry.

6.1 The "Return Bend" / Z-Bend Scenario

Consider a Z-profile where Bend 1 is UP and Bend 2 is DOWN.

- **Step 1:** The algorithm identifies Face A as Base. It classifies Bend 1 (Face B) as **UP**.
- **Step 2:** To classify Bend 2 (Face C), the algorithm must conceptually move the "Base" definition. Face B becomes the **Temporary Base Face**.
- **Logic:** The classification is always *relative* to the face immediately preceding the bend in the kinematic chain.
- **Salvagnini Logic:** The machine clamps Face A. It bends Face B up. Then, to bend Face C, the manipulator might shift. However, usually, Z-bends are performed in sequence. The "Direction" is always relative to the *current* orientation of the material held by the blankholder. If the blankholder moves to Face B, then Face B defines the $Z = 0$ plane, and Face C's centroid is compared to Face B's normal.[32]

6.2 Hems and Flattening

A hem involves bending to an acute angle (e.g., 135° Up) and then flattening it.

- **Stage 1 (Pre-bend):** The flange is clearly in the Positive Half-Space. $D > 0$. Classification: **BEND_UP**.
- **Stage 2 (Flattened):** The flange is now parallel to the base, resting on top of it.
  - Centroid Distance $D \approx \text{Thickness}$.
  - Since Thickness > 0, it is still technically in the Positive Half-Space.
  - **Distinction:** The normal of the hemmed flange will be **Anti-Parallel** (

$N_{hem} \approx -N_{base}$) to the base normal (faces looking at each other). This signature (Parallel Normals + Small Positive Offset) identifies a Hem.

6.3 Geometric Fidelity: K-Factors and Bend Allowances

While the classification logic relies on centroids, the *position* of that centroid is determined by the unfolding logic used to create the 3D model.

- **Bend Allowance (BA):** The arc length of the neutral axis. $BA = \frac{\pi}{180} \times A \times (R + K \times T)$. [34]
- **Relevance:** If the CAD model was exported with an incorrect K-Factor, the flange length might be slightly off. However, this linear error does not affect the *sign* of the displacement vector V_{disp} . The robustness of the SDF method lies in its insensitivity to these micro-dimensional errors, unlike methods relying on exact surface tangency.

6.4 Handling Curved/Conical Bends

If the bend radius varies (Conical Bend), the "Bend Face" is not a cylinder.

- **Impact:** The axis of the bend is not a single line but a cone vector.
- **Solution:** The SDF method remains valid. The centroid of the flange still dictates the mass distribution. The "Up/Down" classification is a binary kinematic decision; the machine cannot bend "diagonally up." It approximates the cone via step-bending or specialized tooling. For the CAM logic, if the mass is > 0, it is an Up command.

---

7. Implementation Strategy and Architecture

This section provides a blueprint for the C++ implementation within the OpenCASCADE environment.

7.1 Data Structures

We define a BendFeature struct to encapsulate the topological analysis.

C++

```
struct BendFeature {
    TopoDS_Face BaseFace;
    TopoDS_Face BendFace;   // The cylindrical face
```

```
    TopoDS_Face FlangeFace; // The target planar face

    gp_Pnt BaseCentroid;
    gp_Dir BaseNormal;      // Corrected for Orientation
    gp_Pnt FlangeCentroid;

    double SignedDistance;
    double BendAngle;       // Calculated via normals

    enum Type {
        BEND_UP,
        BEND_DOWN,
        HEM,
        UNKNOWN
    } Classification;
};
```

7.2 The Material Normal Utility Function

This critical helper function resolves the orientation ambiguity discussed in Section 4.2.

C++

```cpp
// Function to get the reliable "Material Outward" normal
gp_Dir GetMaterialNormal(const TopoDS_Face& face) {
    // 1. Get Geometric Properties
    BRepAdaptor_Surface surface(face);

    // 2. Evaluate at parametric center (U_mid, V_mid)
    double u = (surface.FirstUParameter() + surface.LastUParameter()) / 2.0;
    double v = (surface.FirstVParameter() + surface.LastVParameter()) / 2.0;

    gp_Pnt p;
    gp_Vec d1u, d1v;
    surface.D1(u, v, p, d1u, d1v);

    // 3. Compute Parametric Normal (Cross Product)
    gp_Vec normGeom = d1u.Crossed(d1v);

    // 4. Normalize
```

```
    if (normGeom.Magnitude() < Precision::Confusion()) {
        // Handle singularity (rare for planes)
        return gp::DZ();
    }
    gp_Dir unitNorm(normGeom);

    // 5. Apply Topological Orientation Correction
    // If the face is REVERSED, the material is on the "Geometric Normal" side,
    // meaning the "Outward" normal is opposite to geometry.
    if (face.Orientation() == TopAbs_REVERSED) {
        unitNorm.Reverse();
    }

    return unitNorm;
}
```

7.3 The Classification Routine (SDF Method)

C++

```cpp
BendFeature::Type ClassifyBend(const TopoDS_Face& base, const TopoDS_Face& flange) {
    // 1. Get Base Data
    gp_Dir nBase = GetMaterialNormal(base);

    GProp_GProps baseProps;
    BRepGProp::SurfaceProperties(base, baseProps);
    gp_Pnt pBase = baseProps.CentreOfMass();

    // 2. Get Flange Data
    GProp_GProps flangeProps;
    BRepGProp::SurfaceProperties(flange, flangeProps);
    gp_Pnt pFlange = flangeProps.CentreOfMass();

    // 3. Vector Algebra: Signed Distance
    // Vector from Base Center to Flange Center
    gp_Vec vDisp(pBase, pFlange);

    // Project onto Base Normal
    double signedDist = vDisp.Dot(gp_Vec(nBase));
```

```
  // 4. Thresholding (e.g., 0.1mm to account for meshing noise)
  if (signedDist > 0.1) {
      return BendFeature::BEND_UP;
  } else if (signedDist < -0.1) {
      return BendFeature::BEND_DOWN;
  } else {
      // Distance is near zero. Check Normals for Hem.
      gp_Dir nFlange = GetMaterialNormal(flange);
      double align = nBase.Dot(nFlange);

      // If normals are opposite (Dot ~ -1), it's a Hem folded back.
      // Hems are technically "Up" bends flattened, but classification
      // depends on the specific machine cycle desired.
      if (align < -0.9) return BendFeature::HEM;

      return BendFeature::UNKNOWN; // Likely coplanar flat sheet
  }
}
```

7.4 Table: Comparison of Classification Methods

The following table summarizes the trade-offs between the explored methodologies, justifying the selection of SDF as the primary solution.

Feature	**Method A: SDF (Recommended)**	**Method B: Edge Convexity**	**Method C: Normal Angle**
Primary Metric	Global Centroid Position	Local Cross Product	Normal Alignment
Mathematics	Dot Product (Scalar Projection)	Cross Product (Vector Orthogonality)	Arccosine (Dot Product)
Robustness	**High** (Averages out noise)	**Low** (Sensitive to edge topology)	**Medium** (Ambiguous direction)
Orientation Sensitivity	Handles REVERSED natively	Requires complex Wire parsing	Ambiguous

Computational Cost	Low (Global Properties)	Medium (Edge Iteration)	Low (Surface Properties)
Handling Hems	Detects offset (Thickness)	Fails (Tangent is continuous)	Detects anti-parallelism
Best Use Case	**Up/Down Determination**	Convex/Concave Feature Detection	Bend Angle Calculation

---

8. Conclusion

The reverse-engineering of Salvagnini STREAMBEND CAM logic requires a shift from visual geometric inspection to rigorous topological interrogation. By implementing the **Signed Distance Function (SDF)** method, we successfully translate the kinematic reality of the machine—actuating blades into positive or negative half-spaces—into a deterministic vector algebra calculation.

This approach effectively bypasses the common pitfalls of CAD data exchange, such as inconsistent edge orientations and variable geometric definitions, by anchoring the logic in the physics of the mass distribution. The flange's centroid simply cannot be in the positive half-space if the bend is Down.

8.1 Future Work and Phase 3 Outlook

With the "Up/Down" classification solved, Phase 3 should focus on:

1. **Bend Sequencing:** Using the classification to determine collision-free orders (e.g., Outside-In strategy).
2. **Tool Selection:** Mapping the bend length and clearance to the ABA (Automatic Blankholder Adjustment) and CLA (Auxiliary Blades) configurations.[37]
3. **Kinematic Simulation:** Visualizing the P4 manipulator movement using the generated Rodrigues rotation matrices to simulate the exact part path.

This mathematical framework provides the solid kernel upon which these advanced CAM features can be built.

Works cited

1. Salvagnini Panel Bending Services - Unifabs, accessed February 1, 2026, https://unifabs.eu/manufacturing-services/salvagnini-panel-bender/
2. The flexible bending solution. - Sp-Tech, s.r.o., accessed February 1, 2026, https://www.sp-tech.cz/static/soubory/stranka-71/prospekt-salvagnini-p2-152.pdf

3. Achieving bending efficiencies with panel benders - The Fabricator, accessed February 1, 2026, https://www.thefabricator.com/thefabricator/article/bending/achieving-bending-efficiencies-with-panel-benders
4. The Panel Bender: Your Ultimate Guide - Italian Machinery Association, accessed February 1, 2026, https://www.italianmachines.eu/en/publications/437/the-panel-bender-your-ultimate-guide
5. Distance from point to plane - Math Insight, accessed February 1, 2026, https://mathinsight.org/distance_point_plane
6. Distance from a point to a plane - Wikipedia, accessed February 1, 2026, https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_plane
7. Panel Bender -P4 - MachineryHost, accessed February 1, 2026, https://f.machineryhost.com/71b2fab0316bfa7fa24ce0a0e5460bad/Brochure%20-%20Panel%20Bender%20-P4.pdf
8. How to get the dihedral angle of adjoin faces - Open CASCADE Technology, accessed February 1, 2026, https://dev.opencascade.org/content/how-get-dihedral-angle-adjoin-faces
9. Correct face normal calculation - Forum Open Cascade Technology, accessed February 1, 2026, https://dev.opencascade.org/content/correct-face-normal-calculation
10. Rotation of a vector distribution to align with a normal vector - Math Stack Exchange, accessed February 1, 2026, https://math.stackexchange.com/questions/61547/rotation-of-a-vector-distribution-to-align-with-a-normal-vector
11. General Rotation Matrix in 3D | Rodrigues' Formula Derivation & SO(3) - YouTube, accessed February 1, 2026, https://www.youtube.com/watch?v=LLHfCGkaQrA
12. Universal Sheet Metal Panel Bending Machine - Salvagnini, accessed February 1, 2026, https://www.salvagninigroup.com/en-US/products/panel-benders
13. Normal (geometry) - Wikipedia, accessed February 1, 2026, https://en.wikipedia.org/wiki/Normal_(geometry)
14. 12.3: The Dot Product - Mathematics LibreTexts, accessed February 1, 2026, https://math.libretexts.org/Bookshelves/Calculus/Calculus_(OpenStax)/12%3A_Vectors_in_Space/12.03%3A_The_Dot_Product
15. Dot product - Wikipedia, accessed February 1, 2026, https://en.wikipedia.org/wiki/Dot_product
16. Dot products (article) - Khan Academy, accessed February 1, 2026, https://www.khanacademy.org/math/multivariable-calculus/thinking-about-multivariable-function/x786f2022:vectors-and-matrices/a/dot-products-mvc
17. Finding The Angle Between Two Vectors - Calculus 3 - YouTube, accessed February 1, 2026, https://www.youtube.com/watch?v=dYPRYO8QhxU
18. Point-Plane Distance -- from Wolfram MathWorld, accessed February 1, 2026, https://mathworld.wolfram.com/Point-PlaneDistance.html
19. signed distance between plane and point - Stack Overflow, accessed February 1, 2026,

https://stackoverflow.com/questions/3860206/signed-distance-between-plane-and-point
20. how to align any vector with a specified axis through rotations - MATLAB Answers, accessed February 1, 2026, https://www.mathworks.com/matlabcentral/answers/1728115-how-to-align-any-vector-with-a-specified-axis-through-rotations
21. Rodrigues' rotation formula - Wikipedia, accessed February 1, 2026, https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula
22. TopoDS_Face Class Reference - Open CASCADE Technology, accessed February 1, 2026, https://old.opencascade.com/doc/occt-7.1.0/refman/html/class_topo_d_s___face.html
23. TopoDS_Face Class Reference - Open CASCADE Technology, accessed February 1, 2026, https://dev.opencascade.org/doc/refman/html/class_topo_d_s___face.html
24. TopoDS_Edge Class Reference - Open CASCADE Technology, accessed February 1, 2026, https://dev.opencascade.org/doc/refman/html/class_topo_d_s___edge.html
25. Calculating the attribute of a edge - Forum Open Cascade Technology, accessed February 1, 2026, https://dev.opencascade.org/content/calculating-attribute-edge
26. c++ - How to get the coordinates of the center of mass of a TopoDS_Face? - Stack Overflow, accessed February 1, 2026, https://stackoverflow.com/questions/54901561/how-to-get-the-coordinates-of-the-center-of-mass-of-a-topods-face
27. How to get the coordinates of the center of mass of a TopoDS_Face? - Forum Open Cascade Technology, accessed February 1, 2026, https://dev.opencascade.org/content/how-get-coordinates-center-mass-topods-face
28. Finding connected faces of a CAD model - Forum Open Cascade Technology, accessed February 1, 2026, https://dev.opencascade.org/content/finding-connected-faces-cad-model
29. Determining Bending Angles for a BREP - Grasshopper - McNeel Forum, accessed February 1, 2026, https://discourse.mcneel.com/t/determining-bending-angles-for-a-brep/197295
30. Signed angle between two 3D vectors with same origin within the same plane, accessed February 1, 2026, https://stackoverflow.com/questions/5188561/signed-angle-between-two-3d-vectors-with-same-origin-within-the-same-plane
31. Finding/Calculating the bend "direction" of a Sheet metal CAD model - Math Stack Exchange, accessed February 1, 2026, https://math.stackexchange.com/questions/3580414/finding-calculating-the-bend-direction-of-a-sheet-metal-cad-model
32. Bolt Pattern Force Distribution - MechaniCalc, accessed February 1, 2026, https://mechanicalc.com/reference/bolt-pattern-force-distribution
33. Bending Of Beams With Non-Symmetrical Cross Section, accessed February 1,

2026, https://www.aeromech.usyd.edu.au/structures/acs1-p9.html
34. Sheet metal bending calculation basics - The Fabricator, accessed February 1, 2026, https://www.thefabricator.com/thefabricator/article/bending/sheet-metal-bending-calculation-basics
35. Calculating Bend Allowance & Bend Deduction (+ Formulas) - SendCutSend, accessed February 1, 2026, https://sendcutsend.com/blog/guide-to-calculating-bend-allowance-and-bend-deduction/
36. Top 5 Sheet Metal Design Formulas for Bending (2026), accessed February 1, 2026, https://www.approvedsheetmetal.com/blog/5-sheet-metal-design-formulas
37. p4-benders.pdf - Empire Machinery, accessed February 1, 2026, https://www.empire-machinery.com/wp-content/uploads/2021/06/p4-benders.pdf