

# Discrete Torsion of Connection Forms on Simplicial Meshes

## SUPPLEMENTAL MATERIAL – ACM SIGGRAPH 2025

Theo Braune, Mark Gillespie, Yiying Tong, and Mathieu Desbrun

This supplemental material contains pseudocode for our algorithm in [Section 1](#), and provides additional details on the derivation of our analytical reference solution for the convergence test of the discrete Levi-Civita connection in [Section 2](#).

### 1 Pseudocode

Our pseudocode is expressed via a halfedge mesh data structure encoding a triangle mesh  $M = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ . We use  $\vec{ij} \in \mathcal{H}$  to denote the halfedge from  $i$  to  $j$ , and use  $\mathcal{H}$  to denote the set of halfedges which run along interior edges (i.e., the set of edges on which our trivial connection stores meaningful values). We make use of standard halfedge operations such as  $\text{HALFEDGE}(f)$  which returns the first halfedge in face  $f$  according to the mesh data structure's ordering.

We write vectors  $(a, b) \in \mathbb{R}^2$  as complex numbers  $z = a + bi \in \mathbb{C}$ , and use the argument  $\text{ARG}(a + bi) := \text{atan2}(b, a)$ , the scalar cross product  $(a + bi) \times (c + di) := ad - bc$ , and the complex inverse  $(a + bi)^{-1} := (a - bi)/(a^2 + b^2)$ . We write our face frames using complex numbers as well: on each face  $f$ , we let  $F_{\text{re}}$  be the unit vector pointing along  $\text{HALFEDGE}(f)$ , and let  $F_{\text{im}}$  be the unit vector rotated  $90^\circ$  counter-clockwise in the plane of  $f$ . Then  $E_f = a + bi$  denotes a frame with  $x$ -axis  $e_x = aF_{\text{re}} + bF_{\text{im}}$  and  $y$ -axis  $e_y = -bF_{\text{re}} + aF_{\text{im}}$ .

We also use the following standard quantities and subroutines:

- $\theta_i^{jk}$  is the corner angle of triangle  $ijk$  at vertex  $i$ .
- $d_1 \in \mathbb{Z}^{\mathcal{F} \times \mathcal{E}}$  is the 1-form discrete exterior derivative, i.e. the face-edge signed incidence matrix (see e.g. [Crane et al. \[2013, §3.6\]](#)).
- $\text{ANGLESUMS}(M, \ell)$  returns a vector  $\Phi \in \mathbb{R}^{\mathcal{V}}$  containing the sum of corner angles around each vertex  $i$ .
- $\text{ANGLEDFACTS}(M, \ell)$  returns a vector  $K \in \mathbb{R}^{\mathcal{V}}$  containing the angle defect around each vertex  $i$  (Explicitly,  $K_i$  is  $2\pi - \Phi_i$  for interior vertices, and  $\pi - \Phi_i$  for boundary vertices).
- $\text{LAYOUTDIAMOND}(M, \ell, \vec{ij})$  returns positions  $p_i, p_j, p_k, p_l \in \mathbb{C}$  for the vertices of the two adjacent triangles  $ijk, jil$  (see e.g. [Sharp et al. \[2021, Appendix A\]](#)).

#### Algorithm 1 $\text{HALFEDGE DIRECTION IN FACE FRAME}(M, \ell, E, \vec{ij})$

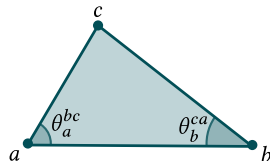
**Input:** A moving frame  $E \in \mathbb{C}^{\mathcal{F}}$  and halfedge  $\vec{ij}$  on a mesh  $M$  with edge lengths  $\ell$ .

**Output:** The vector  $z \in \mathbb{C}$  indicating the direction in which  $\vec{ij}$  points when written in frame  $E$ .

```

1:  $abc \leftarrow \text{FACE}(\vec{ij})$ 
2: if  $\vec{ij} == \vec{ab}$  then
3:   return  $E_f^{-1}$ 
4: else if  $\vec{ij} == \vec{bc}$  then
5:   return  $-\exp(-i\theta_b^{ca})E_f^{-1}$ 
6: else  $\# \vec{ij} == \vec{ca}$ 
7:   return  $-\exp(i\theta_a^{bc})E_f^{-1}$ 

```



#### Algorithm 2 $\text{COMPUTE HINGE CONNECTION}(M, \ell, E)$

**Input:** A moving frame  $E \in \mathbb{C}^{\mathcal{F}}$  on a mesh  $M$  with edge lengths  $\ell$ .

**Output:** The hinge connection  $\eta \in \mathbb{R}^{\mathcal{H}}$ , written in frame  $E$ .

```

1: for each  $\vec{ij} \in \mathcal{H}$  do
2:    $z_{ij} \leftarrow \text{HALFEDGE DIRECTION IN FACE FRAME}(M, \ell, E, \vec{ij})$ 
3:    $z_{ji} \leftarrow \text{HALFEDGE DIRECTION IN FACE FRAME}(M, \ell, E, \vec{ji})$ 
4:    $\eta_{ij} \leftarrow \text{ARG}(-z_{ij}/z_{ji})$ 
5: return  $\eta$ 

```

#### Algorithm 3 $\text{COMPUTE DISCRETE LEVI CIVITA OFFSET}(M, \ell, b)$

**Input:** A mesh  $M$  with edge lengths  $\ell$ , and a matrix  $b \in \mathbb{R}^{3 \times \mathcal{F}}$  giving the location of the dual vertex of each face  $ijk$  written in barycentric coordinates on  $ijk$ . We use  $b_i^{jk}$  to denote the barycentric coordinate associated to vertex  $i$  within face  $ijk$ .

**Output:** A discrete 1-form  $\lambda \in \mathbb{R}^{\mathcal{E}}$  giving the difference between the discrete Levi-Civita connection and the hinge connection.

```

1:  $\Phi, K \leftarrow \text{ANGLESUMS}(M, \ell), \text{ANGLEDFACTS}(M, \ell)$ 
2:  $A \leftarrow 0 \in \mathbb{R}^{\mathcal{V}}$  # zero-initialize dual areas
3: for each  $\vec{ij} \in \mathcal{H}$  do
4:    $p_i, p_j, p_k, p_l \leftarrow \text{LAYOUTDIAMOND}(M, \ell, \vec{ij})$ 
5:    $p_{ijk} \leftarrow b_i^{jk}p_i + b_j^{ki}p_j + b_k^{ij}p_k$  # location in planar layout
6:    $p_{jil} \leftarrow b_j^{il}p_j + b_i^{lj}p_i + b_l^{ji}p_l$ 
7:    $\varphi_{*ij} \leftarrow \text{ARG}[(p_{ijk} - p_i)/(p_{jil} - p_i)]$  # dual corner angle
8:    $A_{*ij} \leftarrow \frac{1}{2}(p_{jil} - p_i) \times (p_{ijk} - p_i)$  # dual triangle area
9:    $A_i \leftarrow A_i + A_{*ij}$  # add triangle area to vertex dual area
10: for each  $ij \in \mathcal{E}$  do # Equation 8 of the main paper
11:    $\lambda_{ij} \leftarrow \frac{1}{A_{*ij} + A_{*ji}} \left( K_i A_{*ij} \left( \frac{A_{*ij}}{A_i} - \frac{\varphi_{*ij}}{\Phi_i} \right) - K_j A_{*ji} \left( \frac{A_{*ji}}{A_j} - \frac{\varphi_{*ji}}{\Phi_j} \right) \right)$ 
12: return  $\lambda$ 

```

#### Algorithm 4 $\text{COMPUTE TORSION CONNECTION OFFSET}(M, \ell, b, p)$

**Input:** A mesh  $M = (\mathcal{V}, \mathcal{E}, \mathcal{F})$  with given edge lengths  $\ell$ , a matrix  $b \in \mathbb{R}^{3 \times \mathcal{F}}$  giving the locations of the dual vertices, and a scalar torsion potential  $p \in \mathbb{R}^{\mathcal{F}}$ .

**Output:** A discrete 1-form  $\mu \in \mathbb{R}^{\mathcal{E}}$  giving the difference between the discrete Levi-Civita connection and the connection with the desired torsion  $dp$ .

```

1: return  $\text{COMPUTE DISCRETE LEVI CIVITA OFFSET}(M, \ell, b) + d_1^T p$ 

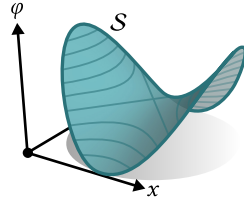
```

*Note.* When using the simple construction of the hinge map  $\eta$  in [Algorithm 2](#), the differential  $(d\eta)_i$  at a vertex  $i$  may differ from the angle defect  $K_i$  by a multiple of  $2\pi$ . When running trivial connections, one should use  $K_i$  as the initial connection curvature to ensure that Gauss-Bonnet is satisfied. Similarly, when starting from another connection  $\alpha = \eta + \lambda$ , using  $K + d\lambda$  as the initial connection curvature ensures that Gauss-Bonnet is satisfied.

## 2 Connections on Quadratic Surfaces

Let  $A \in \mathbb{R}^{2 \times 2}$  be a symmetric matrix. Then  $A$  gives rise to a quadratic form  $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}$ , where  $\varphi : p \mapsto \frac{1}{2}p^T A p$ , which defines a surface

$$\mathcal{S} = \{(x, y, \varphi(x, y))\} \subset \mathbb{R}^3. \quad (1)$$



The surface  $\mathcal{S}$  has a natural parameterization  $f(p) = (p, \varphi(p))$ .

*Metric.* The metric of  $\mathcal{S}$  at a point  $p \in \mathbb{R}^2$  in our parameterization is  $g_p = df_p^T df_p$ . The differential of  $f$  is

$$df_p = \begin{pmatrix} \mathbb{I} \\ (Ap)^T \end{pmatrix}, \quad (2)$$

so the metric is given explicitly by

$$g_p = df_p^T df_p = \mathbb{I} + (Ap)(Ap)^T. \quad (3)$$

*Normals.* We can write  $\mathcal{S}$  as a level set of  $\psi(x, y, z) = \varphi(x, y) - z$ , so the normal of  $\mathcal{S}$  points along  $\nabla\psi$ . The unit normal is thus

$$\hat{n}_p = \frac{1}{\sqrt{1 + p^T A^2 p}} \begin{pmatrix} Ap \\ -1 \end{pmatrix}. \quad (4)$$

*Covariant Derivative.* To compute the covariant derivative operator, we start with a vector field  $V(p)$  in the plane and define a lifted field tangent to the surface  $\mathcal{S}$ :

$$f_* V = \begin{pmatrix} V(p) \\ \varphi_* V(p) \end{pmatrix} = \begin{pmatrix} V(p) \\ p^T A V(p) \end{pmatrix} \quad (5)$$

In order to evaluate the covariant derivative  $\nabla_W^{\mathcal{S}} V$  in direction  $W \in \mathbb{R}^2$ , we start by evaluating the  $\mathbb{R}^3$  directional derivative

$$\nabla_W^{\mathbb{R}^3} (f_* V) = \begin{pmatrix} \nabla_W^{\mathbb{R}^3} V \\ p^T A \nabla_W^{\mathbb{R}^3} V + W^T A V \end{pmatrix}. \quad (6)$$

The covariant derivative projects out the normal component of this directional derivative:

$$\nabla_W^{\mathbb{R}^3} f_* V - n_p n_p^T \nabla_W^{\mathbb{R}^3} f_* V = \nabla_W^{\mathbb{R}^3} f_* V + \begin{pmatrix} Ap \\ -1 \end{pmatrix} \frac{W^T A V}{1 + p^T A^2 p}. \quad (7)$$

Pulling this  $\mathbb{R}^3$ -valued expression back to the  $xy$ -plane by taking the first two components yields the intrinsic expression:

$$\nabla_W^{\mathcal{S}} V = \nabla_W^{\mathbb{R}^2} V + \frac{W^T A V}{1 + p^T A^2 p} Ap. \quad (8)$$

(The third component of Equation 7 is the lift  $\varphi_* \nabla_W^{\mathcal{S}} V = p^T A \nabla_W^{\mathcal{S}} V$ .)

*Parallel Transport.* Now, consider the radial curve  $\gamma(t) = tq$  to a vector  $q \in \mathbb{R}^2$ . In general  $\gamma$  is not geodesic, but we can still parallel transport vectors  $V$  along  $\gamma$  using the parallel transport equation

$$\nabla_{\dot{\gamma}}^{\mathcal{S}} V = 0. \quad (9)$$

If we consider  $V(t)$  as a function of time along  $\gamma$  and expand out the covariant derivative expression from Equation 8, we obtain

$$\dot{V} = -\frac{\dot{\gamma}^T A V}{1 + \gamma^T A^2 \gamma} A \gamma = -\frac{t}{1 + t^2 q^T A^2 q} [(Aq)(Aq)^T] V. \quad (10)$$

When  $V(0)$  is orthogonal to  $Aq$  (with respect to the standard inner product on  $\mathbb{R}^2$ ), then the parallel field  $V(t) = V(0)$  is constant. When  $V(0)$  points along  $Aq$ , then  $V(t)$  always points in the same direction—making the ansatz  $V(t) = \lambda(t)Aq$ , yields the equation

$$\dot{\lambda}(t)Aq = -\frac{tq^T A^2 q}{1 + t^2 q^T A^2 q} \lambda(t)Aq. \quad (11)$$

One can check that the solution  $\lambda(t)$  with  $\lambda(0) = 1$  is given by

$$\lambda(t) = \frac{1}{\sqrt{1 + t^2 q^T A^2 q}}. \quad (12)$$

Finally, we convert back to the standard basis, using the fact that  $(Aq \quad \mathbb{J}Aq)$  is orthogonal so its inverse is its transpose over  $\|Aq\|_{\mathbb{R}^2}^2$ :

$$V(t) = (Aq \quad \mathbb{J}Aq) \begin{pmatrix} \frac{1}{\sqrt{1 + t^2 q^T A^2 q}} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Aq & \mathbb{J}Aq \end{pmatrix}^T V_0 \quad (13)$$

$$= \left( \frac{1}{\sqrt{1 + t^2 q^T A^2 q}} \frac{(Aq)(Aq)^T}{q^T A^2 q} + \frac{(\mathbb{J}Aq)(\mathbb{J}Aq)^T}{q^T A^2 q} \right) V_0. \quad (14)$$

*Parallel-Propagated Frame.* We find a parallel-propagated frame and dual coframe by transporting the standard basis from the origin:

$$E_p = V(1) = \frac{1}{\sqrt{1 + p^T A^2 p}} \frac{(Ap)(Ap)^T}{p^T A^2 p} + \frac{(\mathbb{J}Ap)(\mathbb{J}Ap)^T}{p^T A^2 p}, \quad (15)$$

$$\theta_p = E_p^T g_p = \sqrt{1 + p^T A^2 p} \frac{(Ap)(Ap)^T}{p^T A^2 p} + \frac{(\mathbb{J}Ap)(\mathbb{J}Ap)^T}{p^T A^2 p}. \quad (16)$$

*Levi-Civita-Connection.* To find the Levi-Civita connection, we write  $\theta_p$  in Cartesian coordinates with components  $\lambda_0, \lambda_1, \mu_0, \mu_1$ :

$$\theta_p = \begin{pmatrix} \lambda_0 dx + \mu_0 dy \\ \lambda_1 dx + \mu_1 dy \end{pmatrix}, \quad (17)$$

In the parallel-propagated frame, the Levi-Civita connection is  $\omega = \mathbb{J}\alpha = \mathbb{J}(\alpha_0 dx + \alpha_1 dy)$ . The Cartan structure equations are thus

$$0 = d\theta_p + \omega_p \wedge \theta_p \quad (18)$$

$$= \begin{pmatrix} \frac{\partial \mu_0}{\partial x} - \frac{\partial \lambda_0}{\partial y} & 0 \\ \frac{\partial \mu_1}{\partial x} - \frac{\partial \lambda_1}{\partial y} \end{pmatrix} dx \wedge dy + \begin{pmatrix} 0 & -\alpha \\ \alpha & 0 \end{pmatrix} \wedge \begin{pmatrix} \lambda_0 dx + \mu_0 dy \\ \lambda_1 dx + \mu_1 dy \end{pmatrix} \quad (19)$$

$$= \left( \begin{pmatrix} \frac{\partial \mu_0}{\partial x} - \frac{\partial \lambda_0}{\partial y} \\ \frac{\partial \mu_1}{\partial x} - \frac{\partial \lambda_1}{\partial y} \end{pmatrix} + \begin{pmatrix} -\mu_1 & \lambda_1 \\ \mu_0 & -\lambda_1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \right) dx \wedge dy \quad (20)$$

One can check that the following  $\alpha_p$  solves Equation 20, yielding the desired connection 1-form in the parallel-propagated frame:

$$\alpha_p = \frac{\det(A)}{1 + p^T A^2 p + \sqrt{1 + p^T A^2 p}} (y dx - x dy). \quad (21)$$

For convenience, we include a Mathematica notebook to compute  $\alpha_p$  directly. At the origin,  $\det(A)$  is the Gaussian curvature  $\kappa$ , so:

$$\alpha_{\text{origin}} = \frac{1}{2} \det(A) (y dx - x dy) = \frac{1}{2} \kappa (y dx - x dy). \quad (22)$$

Around any point  $p$ , we can approximate  $\mathcal{S}$  up to second order by a paraboloid centered at  $p$  with the same Gaussian curvature as  $\mathcal{S}$  at  $p$ . Thus, in the parallel-propagated frame at  $p$  we have a second order accurate approximation  $\alpha_p \approx \frac{1}{2} \kappa_p (y dx - x dy)$ . We use this expression for the convergence tests in Sec. 4.3 of the main paper.

## References

- Keenan Crane, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. 2013. *Digital Geometry Processing with Discrete Exterior Calculus*. In *ACM SIGGRAPH 2013 Courses*.  
 Nicholas Sharp, Mark Gillespie, and Keenan Crane. 2021. *Geometry Processing with Intrinsic Triangulations*. In *ACM SIGGRAPH 2021 Courses*.

This notebook demonstrates the calculation of the Levi-Civita connection of a quadratic surface expressed in cartesian coordinates in the parallel propagated frame based at the origin.

Given a symmetric matrix  $A$ , we can define a height field

$$c: \mathbb{R}^2 \rightarrow \mathbb{R}^3: p = (x, y) \mapsto \left( x, y, \frac{1}{2} p^T A p \right)$$

and a quadratic surface  $M = c(\mathbb{R}^2) \subset \mathbb{R}^3$  as the image

In[573]:=

```
(*Define the matrix that will form the quadratic form*)
A = {{a, d}, {d, b}};
J = {{0, -1}, {1, 0}};
```

In[510]:=

```
(*Define the surface in cartesian coordinates*)
c[x_, y_] := {
  x,
  y,
  ({x, y}.A.{x, y}) / 2
};
```

In[511]:=

```
DcDx = D[c[x, y], x] // Simplify;
DcDy = D[c[x, y], y] // Simplify;
```

Now, for this frame we calculate the first fundamental form.

In[513]:=

```
g00 = FullSimplify[DcDx.DcDx];
g01 = FullSimplify[DcDx.DcDy];
g11 = FullSimplify[DcDy.DcDy];
```

In[519]:=

```
FirstFundamentalForm = {{g00, g01}, {g01, g11}};
```

Based on the expression in the supplemental material to our paper “Discrete Torsion of Connection Forms on Simplicial Meshes” we calculate the parallel propagated frame based in the origin. We parallel propagate the standard frame along radial lines leaving the origin.

In[524]:=

```
q = {x, y};
Aq = A.q;
JAq = J.Aq;
qA2q = q.A.A.q;
```

We showed that given a vector  $V_0$  in the tangent space at the origin, and a direction vector  $q \in \mathbb{R}^2$ , the parallel transported vector along the curve  $\gamma(t) = tq$  is given by

$$V(t) = \left( \frac{1}{\sqrt{1 + t^2 q^T A^2 q}} \frac{(Aq)(Aq)^T}{q^T A^2 q} + \frac{(JAq)(JAq)^T}{q^T A^2 q} \right) V_0,$$

In[528]:=

```
PPFMatrix =
  (1 / Sqrt[1 + qA2q]) * (Outer[Times, Aq, Aq] / qA2q) + (Outer[Times, JAq, JAq] / qA2q);
```

Therefore, evaluated at a point  $p = (x, y)$ , the parallel propagated standard frame is given by

In[529]:=

```
PPFX = PPFMatrix.{1, 0};
PPFY = PPFMatrix.{0, 1};
```

Now, we can do a sanity check to ensure the orthonormality of the PPF with respect to the first fundamental form.

In[588]:=

```
OrthoCheck00 = FullSimplify[PPFX.(FirstFundamentalForm.PPFX)];
OrthoCheck10 = FullSimplify[PPFY.(FirstFundamentalForm.PPFX)];
OrthoCheck11 = FullSimplify[PPFY.(FirstFundamentalForm.PPFY)];
{{OrthoCheck00, OrthoCheck10}, {OrthoCheck10, OrthoCheck11}} // MatrixForm
```

Out[591]//MatrixForm=

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

We can express the PPF in cartesian coordinates via

$$e_{1\text{PPF}} = \alpha_1 \partial_x + \beta_1 \partial_y \text{ and } e_{2\text{PPF}} = \alpha_2 \partial_x + \beta_2 \partial_y$$

We are now aiming to find the dual Parallel-Propagated frame

$e_{\text{PPF}}^1$  and  $e_{\text{PPF}}^2$ . We will stick to cartesian coordinates, i.e we define scalar functions  $\lambda_1, \lambda_2, \mu_1, \mu_2$ , such that  $e_{\text{PPF}}^1 = \lambda_1 dx + \mu_1 dy$  and  $e_{\text{PPF}}^2 = \lambda_2 dx + \mu_2 dy$  is the dual frame.

The coefficients for the dual frame are given by

$$\theta = \frac{1}{p^T A^2 p} \left( \sqrt{1 + p^T A^2 p} (A p) (A p)^T + (J A p) (J A p)^T \right):$$

In[535]:=

```
{{λ1, μ1}, {λ2, μ2}} =
  Sqrt[1 + qA2q] (Outer[Times, Aq, Aq] / qA2q) + (Outer[Times, JAq, JAq] / qA2q);
```

Now, we can do a sanity check to see ensure that  $e_{\text{PPF}}^i(e_{j\text{PPF}}) = \delta_{ij}$

In[556]:=

```
check00 = FullSimplify[{λ1, μ1}.PPFX];
check10 = FullSimplify[{λ2, μ2}.PPFX];
check01 = FullSimplify[{λ1, μ1}.PPFY];
check11 = FullSimplify[{λ2, μ2}.PPFY];
{{check00, check10}, {check01, check11}} // MatrixForm
```

Out[560]//MatrixForm=

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Now, we are aiming to solve the Cartan structure equations for the Levi-Civita connection. This means, we are searching for a 1-form  $\alpha = \omega_1 dx + \omega_2 dy$  such that

$$d \begin{pmatrix} e_{\text{PPF}}^1 \\ e_{\text{PPF}}^2 \end{pmatrix} + \begin{pmatrix} 0 & -\alpha \\ \alpha & 0 \end{pmatrix} \wedge \begin{pmatrix} e_{\text{PPF}}^1 \\ e_{\text{PPF}}^2 \end{pmatrix} = 0$$

Here,  $d$  denotes the ordinary exterior derivative on 1-forms. It holds

$$d \begin{pmatrix} e_{\text{PPF}}^1 \\ e_{\text{PPF}}^2 \end{pmatrix} = \begin{pmatrix} \frac{\partial \mu_1}{\partial x} & -\frac{\partial \lambda_1}{\partial y} \\ \frac{\partial \mu_2}{\partial x} & -\frac{\partial \lambda_2}{\partial y} \end{pmatrix} dx \wedge dy$$

In[561]:=

```
DLambda1Dy = FullSimplify[D[λ1, y]];
DLambda2Dy = FullSimplify[D[λ2, y]];
DMu1Dx = FullSimplify[D[μ1, x]];
DMu2Dx = FullSimplify[D[μ2, x]];
```

In[565]:=

```
dPPFX = FullSimplify[DMu1Dx - DLambda1Dy];
dPPFY = FullSimplify[DMu2Dx - DLambda2Dy];
dPPF = {dPPFX, dPPFY};
```

It holds for the wedge product

$$\begin{pmatrix} 0 & -\alpha \\ \alpha & 0 \end{pmatrix} \wedge \begin{pmatrix} e_{\text{PPF}}^1 \\ e_{\text{PPF}}^2 \end{pmatrix} = \begin{pmatrix} -\alpha \wedge e_{\text{PPF}}^1 \\ \alpha \wedge e_{\text{PPF}}^2 \end{pmatrix} =$$

$$\begin{pmatrix} -\omega_1 \mu_2 + \omega_2 \lambda_2 \\ \omega_1 \mu_1 - \omega_2 \lambda_1 \end{pmatrix} dx \wedge dy = \begin{pmatrix} -\mu_2 & \lambda_2 \\ \mu_1 & -\lambda_1 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} dx \wedge dy$$

In[568]:=

```
BforCartanStructure = FullSimplify[{{-μ2, λ2}, {μ1, -λ1}}];
```

We can therefore now solve for the coefficients of the Levi-Civita connection

In[596]:=

```
{ω1, ω2} = FullSimplify[LinearSolve[BforCartanStructure, -dPPF]]
```

Out[596]=

$$\left\{ \left( (a b - d^2) y \left( 1 + (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2 - \sqrt{1 + (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2} \right) \right) / \right.$$

$$\left( (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2 \right) \left( 1 + (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2 \right),$$

$$- \left( (a b - d^2) x \left( 1 + (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2 - \sqrt{1 + (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2} \right) / \left( (a^2 + d^2) x^2 + \right. \right.$$

$$\left. \left. 2 (a + b) d x y + (b^2 + d^2) y^2 \right) \left( 1 + (a^2 + d^2) x^2 + 2 (a + b) d x y + (b^2 + d^2) y^2 \right) \right) \}$$

Finally, we verify the simpler form of the solution given in the supplemental material:

In[576]:=

```
{ω1, ω2} = Det[A] / (1 + qA2q + Sqrt[1 + qA2q]) (-J.q) // FullSimplify
```

Out[576]=

```
{0, 0}
```