```
In[160]:= site3NormAzNSpeed =
  azimuthNSpeed[(inData[[3, 4]] - (1000 * translationVector[[1]])),
   (inData[[3, 5]] - (1000 * translationVector[[2]]))];
```

Distance between the first site listed in the input table (first row of input data) and the second site (km) before displacement

```
In[161]:= sideLength12km =
  √((m1[[1, 1]] - m1[[2, 1]])^2 + (m1[[1, 2]] - m1[[2, 2]])^2) / 1000;
```

Distance between the second site and the third site (km)

```
In[162]:= sideLength23km =
  √((m1[[2, 1]] - m1[[3, 1]])^2 + (m1[[2, 2]] - m1[[3, 2]])^2) / 1000;
```

Distance between the first site and the third site (km)

```
In[163]:= sideLength13km =
  √((m1[[1, 1]] - m1[[3, 1]])^2 + (m1[[1, 2]] - m1[[3, 2]])^2) / 1000;
```

End Interlude

Step 10: Find the value of the rotational velocity, Ω, in matrix m5: the value in $m5_{31}$

```
In[164]:= angularVelocityRad = m5[[3, 1]];

In[165]:= angVelNanoRadYr = angularVelocityRad / 10^-9;
```

Step 11: Define the 2-D Lagrangian strain tensor (ε_{ij}) and call it matrix m6

```
In[166]:= m6 = ( m5[[4, 1]]  m5[[5, 1]]
                m5[[5, 1]]  m5[[6, 1]] );
```

Step 12: Determine the orientations of the principal infinitesimal strain axes by finding the eigenvectors of the symmetrical Lagrangian infinitesimal strain tensor

```
In[167]:= eValues = Eigenvalues[m6];

In[168]:= e1 = If[eValues[[1]] > eValues[[2]], eValues[[1]], eValues[[2]]];

In[169]:= e2 = If[eValues[[1]] > eValues[[2]], eValues[[2]], eValues[[1]]];

In[170]:= e1 - e2;

In[171]:= eVectors = Eigenvectors[m6];
```

```
In[172]:= s1V = If[eValues[[1]] > eValues[[2]], eVectors[[1]], eVectors[[2]]];

In[173]:= s1UV = {s1V[[1]]/Norm[s1V], s1V[[2]]/Norm[s1V]};

In[174]:= s2V = If[eValues[[1]] > eValues[[2]], eVectors[[2]], eVectors[[1]]];

In[175]:= s2UV = {s2V[[1]]/Norm[s2V], s2V[[2]]/Norm[s2V]};

In[176]:= angleS1HAxis = (ArcCos[unitNorthVector.s1UV] (180/π)) + centroidGridConvergence;

In[177]:= azimuthS1HAxis1 = If[(s1UV[[1]] < 0), (360 - angleS1HAxis), angleS1HAxis];

In[178]:= azimuthS1HAxis2 =
   If[azimuthS1HAxis1 > 180, azimuthS1HAxis1 - 180, azimuthS1HAxis1 + 180];

In[179]:= angleS2HAxis = (ArcCos[unitNorthVector.s2UV] (180/π)) + centroidGridConvergence;

In[180]:= azimuthS2HAxis1 = If[s2UV[[1]] < 0, 360 - angleS2HAxis, angleS2HAxis];

In[181]:= azimuthS2HAxis2 =
   If[azimuthS2HAxis1 > 180, azimuthS2HAxis1 - 180, azimuthS2HAxis1 + 180];
```

Step 13: Determine the magnitude of the maximum infinitesimal shear strain (γ_{max}) at 45° to the maximum principal strain axis

```
In[182]:= maxShearStrain = 2 Sqrt[(((1/2) (m6[[1, 1]] - m6[[2, 2]]))^2 + (m6[[1, 2]])^2)];
```

Step 14: Determine the areal strain between the GPS sites

```
In[183]:= circleArea = π;

In[184]:= s1 = 1 + e1;

In[185]:= s2 = 1 + e2;

In[186]:= ellipseArea = π × (s1) × (s2);

In[187]:= areaStrain = (ellipseArea - circleArea)/circleArea;

In[188]:= m6[[1, 1]] + m6[[2, 2]];
```

Step 15: Identify/compute the second and third invariants of the strain tensor

```
In[189]:= firstInvariant = (m6[[1, 1]] + m6[[2, 2]]);
```

In[190]:=
```
secondInvariant = ((m6[[1, 1]]) (m6[[2, 2]])) - (m6[[1, 2]])^2;
```

In[191]:=
```
thirdInvariant = Det[m6];
```

Step 16: Compute the uncertainties

In[192]:= m7 =

$$\begin{pmatrix} \frac{1}{inData[[1,6]]^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{inData[[1,7]]^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{inData[[2,6]]^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{inData[[2,7]]^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{inData[[3,6]]^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{inData[[3,7]]^2} \end{pmatrix};$$

In[193]:=
```
m8 = Transpose[m2];
```

We use a Moore - Penrose matrix inverse (PseudoInverse) to compute M9, because the matrix resulting from m8.m7.m2 can be a poorly conditioned matrix.

In[194]:=
```
m9 = PseudoInverse[m8.m7.m2];
```

In[195]:= uncertTransE = $\sqrt{m9[[1, 1]]}$;

In[196]:= uncertTransN = $\sqrt{m9[[2, 2]]}$;

In[197]:= uncertRot = $\sqrt{m9[[3, 3]]}$;

In[198]:= uncertExx = $\sqrt{m9[[4, 4]]}$;

In[199]:= uncertExy = $\sqrt{m9[[5, 5]]}$;

In[200]:= uncertEyy = $\sqrt{m9[[6, 6]]}$;

In[201]:= summaryPlot = StrainEllipse[e1 * 10^9, e2 * 10^9, azimuthS1HAxis1, azimuthS2HAxis1];

Output

Distances between PBO sites before displacement

Distance between the first site and the second site (km)

In[202]:=
```
{{site1, site2}, sideLength12km}
```

Out[202]=
```
{{P574, EWPP}, 22.5701}
```

Distance between the second site and the third site (km)

In[203]:= {{site2, site3}, sideLength23km}

Out[203]= {{EWPP, P577}, 29.2641}

Distance between the first site and the third site (km)

In[204]:= {{site1, site3}, sideLength13km}

Out[204]= {{P574, P577}, 29.0547}

Translation vector in the North American reference frame

Translation vector {east component, north component} where negative values are west and south, respectively. Expressed in (mm/yr)

In[205]:= N[translationVector * 1000]

Out[205]= {-18.6194, 19.3472}

Uncertainties in translation vector {east component, north component} expressed in (m/yr)

In[206]:= N[{uncertTransE, uncertTransN}]

Out[206]= {0.30536, 0.0738617}

Azimuth of translation vector (degrees)

In[207]:= N[azTranslationVect]

Out[207]= 316.375

Translation speed (mm/yr)

In[208]:= N[translationSpeedMM]

Out[208]= 26.8514

Total velocities in the North American reference frame (mm/yr)

Site 1 azimuth and velocity in the North American reference frame (mm/yr)

In[209]:= {site1, (site1AzimuthNSpeed[[1]] + site1GridConvergence),
 site1AzimuthNSpeed[[2]]}

Out[209]= {P574, 315.203, 27.4653}

Site 2 azimuth and velocity in the North American reference frame (mm/yr)

In[210]:= {site2, (site2AzimuthNSpeed[[1]] + site2GridConvergence),
 site2AzimuthNSpeed[[2]]}

Out[210]= {EWPP, 315.647, 31.3927}

Site 3 azimuth and velocity in the North American reference frame (mm/yr)