Chapter 12. Cycloid finite motion between plates

12.1 Introduction

My impression is that most people don't give any thought whatsoever to the question of where a given point on one plate goes, relative to an observer across the boundary on an adjacent plate. Most people don't know or care that plates exist. Of that small group of people who have given any thought to the matter, my impression is that most think that the point moves in a circular trajectory around something called an Euler pole that somehow stays fixed to the observer's plate and to the moving point. With all due respect, that idea cannot be correct given the observed data on the relative motion of plates.

It is curious to note that the fact that finite relative motion cannot be circular has been well known since the early 1970s if not before. The trajectory of a point on one plate relative to another plate is almost always more complex than a circle, because the angular velocity vectors related to a given plate are almost always non-coaxial. And a plate cannot rotate in a circular finite trajectory around more than one axis to which it is fixed at a time.

What, then, is the simplest model for the finite relative motion of plates that is permissible given the observed instantaneous plate-to-plate motion data? The answer (or at least an answer) follows from the observation that the observed instantaneous motion of one plate relative to another might be considered the sum of the motion of one plate and the motion of the other plate, both discerned from a frame of reference that is external to the plates. If ${}_{E}\Omega_{A}$ is constant and ${}_{E}\Omega_{B}$ is constant, ${}_{A}\Omega_{B}$ will be constant over the finite time interval. During a finite time interval, plate A can rotate around the ${}_{E}\Omega_{A}$ axis at a constant angular speed, plate B can rotate around the ${}_{E}\Omega_{B}$ axis at a constant angular speed, and at any instant during that time interval the angular speed of any point on plate B observed from plate A will be ${}_{A}\omega_{B}$ directed around the ${}_{A}\Omega_{B}$ axis.

If you are sitting on plate A watching a point on plate B during this time interval, the point will trace a regular geometric figure that is simple, but not circular.

12.2 User-defined functions

We will use the following user-defined function developed in a previous chapter.

```
convert2Cart[lat_, long_] := {Cos[lat Degree] Cos[long Degree],
   Cos[lat Degree] Sin[long Degree], Sin[lat Degree]};

unitVect3D[vect_] :=
  {(vect[[1]] / Norm[vect]), (vect[[2]] / Norm[vect]), (vect[[3]] / Norm[vect])};

findGeogCoord[vect_] := Module[{lat, long, a, b, c, d, e, f}, a = ArcSin[vect[[3]]];
   b = {vect[[1]], vect[[2]], 0};
   c = If[((Abs[vect[[1]]] < (1 × 10^-14)) && (Abs[vect[[2]]] < (1 × 10^-14))),
     {1, 1, 0}, {vect[[1]] / Norm[b], vect[[2]] / Norm[b], 0}];
   d = {1, 0, 0};
   e = VectorAngle[c, d];
   f = If[(vect[[2]] < 0), (-e), (e)];
   lat = a (180 / π);
   long = If[
     ((Abs[vect[[1]]] < (1 × 10^-14)) && (Abs[vect[[2]]] < (1 × 10^-14))), 0, (f (180 / π))];
   {lat, long}];
```

```
makeGreatCircle[normal_] := Module[{a, x1, y1, z1, x2, y2, z2, j1, j2, j3, θ, b},
  a = Table[{Cos[i Degree], Sin[i Degree], 0}, {i, 0, 360, 5}];
  x1 = {1, 0, 0};
  y1 = {0, 1, 0};
  z1 = {0, 0, 1};
  z2 = normal;
  x2 = unitVect3D[Cross[z2, z1]];
  y2 = unitVect3D[Cross[z2, x2]];
  j1 = {{x1.x2, y1.x2, z1.x2}, {x1.y2, y1.y2, z1.y2}, {x1.z2, y1.z2, z1.z2}};
  θ = VectorAngle[z1, z2];
  j2 = {{1, 0, 0}, {0, Cos[θ], Sin[θ]}, {0, -Sin[θ], Cos[θ]}};
  j3 = Inverse[j1];
  b = Table[j3.j2.j1.a[[i]], {i, 1, Length[a]}];
  b]
```

```
circMotion[x_, angVelVect_, dT_] :=
  Module[{north, rotPole, w, xPole, yPole, m1, m2, m3, answer}, north = {0, 0, 1};
   rotPole = unitVect3D[angVelVect];
   w = Norm[angVelVect];
   xPole = unitVect3D[rotPole × north]; yPole = unitVect3D[rotPole × xPole];
   m1 = {{xPole[[1]], xPole[[2]], xPole[[3]]}, {yPole[[1]], yPole[[2]], yPole[[3]]},
     {rotPole[[1]], rotPole[[2]], rotPole[[3]]}};
   m2 = {{Cos[(w dT) Degree], -Sin[(w dT) Degree], 0},
     {Sin[(w dT) Degree], Cos[(w dT) Degree], 0}, {0, 0, 1}};
   m3 = Inverse[m1];
   answer = m3.m2.m1.x;
   answer];
```

12.3 Data from NNR-MORVEL56

The following data are from the no-net-rotation solution derived by Argus and others (2011) based on the MORVEL velocity model (DeMets and others, 2010) and Peter Bird's descriptions of 56 plates (Bird, 2003). These data constitute a model for the motion of individual plates in a reference frame that is external to the plates. The abbreviations in the first column are co=Cocos, nz=Nazca, na=North America, nb=Nubia (the western part of the African plate of Minster and Jordan, 1978), and pa=Pacific. For a given row/record, the second column contains the latitude of the poles around which the corresponding plate moves in a counter-clockwise manner, the third column has the corresponding polar longitude, and the fourth column has the angular speed.

$$nnr = \begin{pmatrix} co & 26.93 & -124.31 & 1.198 \\ nz & 46.23 & -101.06 & 0.696 \\ na & -4.85 & -80.64 & 0.209 \\ nb & 47.68 & -68.44 & 0.292 \\ pa & -63.58 & 114.70 & 0.651 \end{pmatrix};$$

To access the datum in the first row, fourth column of the **nnr** data matrix defined above, we use the expression **nnr[[1,4]]** as follows

```
nnr[[1, 4]]
```

```
1.198
```

We will now extract motion data for several plates relative to the no-net-rotation external reference frame.

```
eΩco = nnr[[1, 4]] * convert2Cart[nnr[[1, 2]], nnr[[1, 3]]];
```

```
eΩnz = nnr[[2, 4]] * convert2Cart[nnr[[2, 2]], nnr[[2, 3]]];
```

```
eΩna = nnr[[3, 4]] * convert2Cart[nnr[[3, 2]], nnr[[3, 3]]];
```

```
eΩnb = nnr[[4, 4]] * convert2Cart[nnr[[4, 2]], nnr[[4, 3]]];
```

```
eΩpa = nnr[[5, 4]] * convert2Cart[nnr[[5, 2]], nnr[[5, 3]]];
```

12.4 Gratuitous personal story

I was a graduate student with an idea. The idea involved the motion of a point on one plate as viewed from another plate over time. I already knew it could not be a circle. I sought out a professor in the math department who might be able to help. I sat in an uncomfortable hardwood chair across the desk from him, and described what I thought I knew about the problem. I said that an observer is on a plate that is rotating at a constant rate around an axis. The observer is looking at a point on a plate that is rotating at a constant rate around a different axis. The two axes are not moving relative to each other. I asked about the name of the trajectory of that point. I thought that if I knew what the curve was called, I might find a mathematical description of the curve that would help me in my work.

He smiled, closing his eyes as he raised his face as if to look at the sky. After what seemed like an eternity, eyes still closed, he said, "Now let me get this straight. You are sitting in a chair, gazing at a beautiful geisha who is standing in serene stillness on a round dias. The dias is rotating slowly. She is at ease, looking into your eyes as you pass before her. In the geisha's hands and resting on her right shoulder is the bamboo shaft of a parasol. She is slowly twirling the shaft, and the broad paper parasol is rotating behind her head. And on the parasol, off to one side but rotating around the bamboo shaft, is a single red dot. And you want to know the name of the curve traced by that red dot?" He opened his eyes and lowered his gaze to look directly at me. "Why, my poorly educated young friend, that is none other than the famous cycloid!"

A cycloid is a figure of rotation around a moving axis or, in spherical kinematics, around two axes. The reason the math professor thought I should know about it is that the outer planets move in figures that are like cycloids as observed from Earth over the course of years. This odd retrograde motion was the reason why the early celestial models for an Earth-centered universe required such elaborate machinery to produce "epicycles." Copernicus had a partial answer to the problem, involving circular motions of all the planets around the Sun. (The orbital trajectories of planets around the sun are better described as ellipses.) None-the-less, it remains true that the outer planets move in looping cycloidal trajectories as observed from Earth. The reason is that the outer planets revolve around the Sun, which (from Earth) appears to revolve around the Earth. So Jupiter revolves around the Sun that, in turn, revolves around the center of Earth, as observed from a vantage point on Earth's surface.

12.5 Cycloidal finite motion

Let's get back to finite plate kinematics. Imagine a system in which an observer is fixed to plate A. In a coordinate system that is external to the plates, plate A moves in a positive (counter-clockwise) circular trajectory around pole ${}_{E}P_{A}$ at a constant angular speed of ${}_{E}\omega_{A}$. In that same external coordinate system, plate B moves in a positive (counter-clockwise) circular trajectory around pole ${}_{E}P_{B}$ at a constant angular speed of ${}_{E}\omega_{B}$. There is a bright red dot painted on plate B to serve as our reference point. Over the finite time that we are modeling, the two poles remain a constant distance from each other.

How do we model the motion of the red dot? We know the location vector to the red dot as it exists today, at time = 0. As the observer looks at pole ${}_{E}P_{A}$ over time, that pole remains fixed to the plate she is on. When the observer turns her head to look at pole ${}_{E}P_{B}$ over time, that pole rotates around pole ${}_{E}P_{A}$ in a negative (clockwise) direction at a speed of ${}_{E}\omega_{A}$. Simultaneously, plate B is rotating around pole ${}_{E}P_{B}$ in a positive (counter-clockwise) direction, carrying with it the red dot. Where is the red dot at any time other than today, as seen by the observer?

Building the cycloid relative-motion model

We start with the location vector to our reference point (**locVectRP**), as defined in the geographic coordinate system {**geogX**, **geogY**, **geogZ**}.

```
latRP = 34.05; longRP = -118.24;
```

```
startRP = convert2Cart[latRP, longRP];
```

```
geogX = {1, 0, 0}; geogY = {0, 1, 0}; geogZ = {0, 0, 1};
```

We will use the Pacific plate as the observer's plate and the Cocos plate as the moving plate, just so we can play with real data. From the input data, we need to specify the locations of the observer's plate pole (**obsZ**), the moving plate's pole (**movZ**) and their respective angular speeds (**obsω** and **movω**).


```
obsZ = convert2Cart[nnr[[5, 2]], nnr[[5, 3]]];
movZ = convert2Cart[nnr[[1, 2]], nnr[[1, 3]]];
obsω = nnr[[5, 4]];
movω = nnr[[1, 4]];
```

Second, we need to define the model time (**modT**) at which the reference point's position relative to the observer will be computed.

```
modT = 1;
```

Third, we transform the geographic coordinates of the reference point so that we know the location vector to the reference point in one of the two coordinate systems that are fixed to the two externally-defined poles of rotation (${}_{E}P_{pa}$ and ${}_{E}P_{co}$). Specifically, we want to know the coordinates of the reference point at **modT** = 0 in the coordinate system in which the pole ${}_{E}P_{co}$ is the Z_{co} axis, the unit vector that coincides with the vector cross product ${}_{E}P_{pa} \times {}_{E}P_{co}$ is the Y_{co} axis, and the X_{co} axis is found from $Y_{co} \times Z_{co}$.

```
movY = unitVect3D[obsZ × movZ]; movX = unitVect3D[movY × movZ];
```

$$\text{j1temp} = \begin{pmatrix} \text{movX.geogX} & \text{movX.geogY} & \text{movX.geogZ} \\ \text{movY.geogX} & \text{movY.geogY} & \text{movY.geogZ} \\ \text{movZ.geogX} & \text{movZ.geogY} & \text{movZ.geogZ} \end{pmatrix};$$

In this case, the preceding **j1temp** matrix can be simplified to the following matrix because the basis vectors of the positive geographic axes all have components of either 0 or 1.

$$\text{j1} = \begin{pmatrix} \text{movX[[1]]} & \text{movX[[2]]} & \text{movX[[3]]} \\ \text{movY[[1]]} & \text{movY[[2]]} & \text{movY[[3]]} \\ \text{movZ[[1]]} & \text{movZ[[2]]} & \text{movZ[[3]]} \end{pmatrix};$$

Let's prove that **j1temp** and **j1** are the same matrices. First, we will display the **j1temp** matrix created in the classic manner using dot products between basis vectors.

```
MatrixForm[j1temp]
```

$$\begin{pmatrix} 0.621325 & 0.0566243 & 0.781504 \\ -0.601167 & 0.674141 & 0.429106 \\ -0.502546 & -0.736429 & 0.452902 \end{pmatrix}$$

Now, we will display the **j1** matrix that does not include dot products, and so will execute faster.

```
MatrixForm[j1]
```

$$\begin{pmatrix} 0.621325 & 0.0566243 & 0.781504 \\ -0.601167 & 0.674141 & 0.429106 \\ -0.502546 & -0.736429 & 0.452902 \end{pmatrix}$$

They are the same, so we will use the **j1** matrix because it is simpler and faster.

Fourth, we construct a matrix we can use to rotate the reference point by an angle (${}_{E}\omega_{co}$ * **modT**).

$$\text{j2} = \begin{pmatrix} \text{Cos[(-mov}\omega\text{ Degree) modT]} & \text{Sin[(-mov}\omega\text{ Degree) modT]} & 0 \\ \text{-Sin[(-mov}\omega\text{ Degree) modT]} & \text{Cos[(-mov}\omega\text{ Degree) modT]} & 0 \\ 0 & 0 & 1 \end{pmatrix};$$

Fifth, we transform the new coordinates of the rotated reference point to a coordinate system in which the pole ${}_{E}P_{pa}$ is the Z_{pa} axis, the unit vector that coincides with the vector cross product ${}_{E}P_{pa} \times {}_{E}P_{co}$ is the Y_{pa} axis, and the X_{pa} axis is found from $Y_{pa} \times Z_{pa}$.

```
obsY = unitVect3D[obsZ × movZ]; obsX = unitVect3D[obsY × obsZ];
```

Specifying this coordinate system, and hence the transformation matrix, takes a bit of work. We determine the value and sign of the angle γ between the poles to the moving and observer's plates. The range of γ is between −180° and 180°. The angle γ is measured in the direction from the pole of the moving plate to the pole of the observer's plate. An anti-clockwise rotation around the vector result of the cross product ${}_{E}\Omega_{slower} \times {}_{E}\Omega_{faster}$ (where ${}_{E}\omega_{slower} \leq {}_{E}\omega_{faster}$) yields a positive value of γ. In our case, the angular speed of the observer's plate (Pacific) is slower than that of the moving plate (Cocos).

In our case, the angular speed of the observer's plate (Pacific) is slower than that of the moving plate (Cocos). Hence, the vector cross product ${}_{E}\Omega_{pa} \times {}_{E}\Omega_{co}$ defines the axis of rotation, and this axis is coincident with **movY**. (Would it have been coincident with **movY** if the observer was on the Cocos plate rather than on the Pacific plate?)

What is an efficient way to determine the sign of the angle γ? If the **obsZ** axis is found by a positive rotation around **movY** from **movZ**, then **obsZ** will be within 90° of the **movX** axis. If the **obsZ** axis is a negative rotation around **movY** from **movZ**, then **obsZ** will be more than 90° away from the **movX** axis. So to define the sign of the rotation angle, we measure the angle between **obsZ** and **movX**.

```
a = VectorAngle[obsZ, movX];
```

```
If[a < (π / 2), (s = 1), (s = (-1))];
```

```
γ = (s) (VectorAngle[movZ, obsZ]);
```

The matrix **j3** reflects a rotation of coordinate axes around a common Y axis from a system in which the pole to the moving plate lies along the Z axis to a system in which the pole to the observer's plate lies along the Z axis.

$$j3 = \begin{pmatrix} \text{Cos}[\gamma] & 0 & -\text{Sin}[\gamma] \\ 0 & 1 & 0 \\ \text{Sin}[\gamma] & 0 & \text{Cos}[\gamma] \end{pmatrix};$$

Sixth, we find the coordinates of the rotated reference point in a coordinate system that is rotated around ${}_{E}P_{pa}$ by an angle ($-\ {}_{E}\omega_{pa}$ * **modT**).

$$j4 = \begin{pmatrix} \text{Cos}[(\text{obs}\omega\ \text{Degree})\ \text{modT}] & \text{Sin}[(\text{obs}\omega\ \text{Degree})\ \text{modT}] & 0 \\ -\text{Sin}[(\text{obs}\omega\ \text{Degree})\ \text{modT}] & \text{Cos}[(\text{obs}\omega\ \text{Degree})\ \text{modT}] & 0 \\ 0 & 0 & 1 \end{pmatrix};$$

Seventh, we define a transformation matrix to return us to the Cartesian geographic coordinate system. We start with the traditional dot-product method for defining the transformation matrix

$$j5temp = \begin{pmatrix} \text{obsX.geogX} & \text{obsY.geogX} & \text{obsZ.geogX} \\ \text{obsX.geogY} & \text{obsY.geogY} & \text{obsZ.geogY} \\ \text{obsX.geogZ} & \text{obsY.geogZ} & \text{obsZ.geogZ} \end{pmatrix};$$

```
MatrixForm[j5temp]
```

$$\begin{pmatrix} -0.777193 & -0.601167 & -0.185929 \\ -0.618163 & 0.674141 & 0.404239 \\ -0.117673 & 0.429106 & -0.895556 \end{pmatrix}$$

then use the "cut-to-the-chase" method (**j5**) recognizing that the **geogX**, **geogY** and **geogZ** basis vectors have components that are all either 0 or 1

$$j5 = \begin{pmatrix} \text{obsX[[1]]} & \text{obsY[[1]]} & \text{obsZ[[1]]} \\ \text{obsX[[2]]} & \text{obsY[[2]]} & \text{obsZ[[2]]} \\ \text{obsX[[3]]} & \text{obsY[[3]]} & \text{obsZ[[3]]} \end{pmatrix};$$

```
MatrixForm[j5]
```

$$\begin{pmatrix} -0.777193 & -0.601167 & -0.185929 \\ -0.618163 & 0.674141 & 0.404239 \\ -0.117673 & 0.429106 & -0.895556 \end{pmatrix}$$

As we saw before, these two matrices are the same, so we will use the simpler formulation (**j5**) to avoid having to compute those 9 dot products. The result will be the location vector (**endRP**) to the red dot after **modT** million years in a coordinate system fixed to the observer.

```
endRP = j5.j4.j3.j2.j1.startRP;
```

We finish by converting the location vector to standard geographic coordinates. Actually, they are not really standard geographic coordinates, but rather they are geographic coordinates as they might have been etched onto the observer's plate today and carried with the plate during finite motion. We do this because the initial question was "where will the point on the Cocos plate be relative to the Pacific plate in 1 million years." It is simply convenient to have a map of the fixed Pacific plate with today's geographic coordinate system represented, and then locate the original and final positions of the reference point relative to that map.

```
result = findGeogCoord[endRP]
```

```
{33.9303, -117.952}
```



```
findGeogCoord[startRP]
```

```
{34.05, -118.24}
```

Testing the cycloid relative-motion model

If our model is consistent, the distance from the reference point to the pole ${}_{E}P_{co}$ should remain constant for all model times. We need to be able to track the ${}_{E}P_{co}$ pole, which is moving in a circular trajectory around the ${}_{E}P_{pa}$ pole. Let's build a model for the motion of pole ${}_{E}P_{co}$ (a.k.a. **movZ**) relative to the Pacific plate. Come to think of it, that is exactly what we did in chapter 5, section 4, so the general solution should be familiar. What we will do here is notice that if we set **movω** to zero, matrix **J2** becomes an identity matrix and essentially drops out of the cycloid equation.

```
startPole = movZ;
```

```
endPole = j5.j4.j3.j1.startPole;
```

```
VectorAngle[endPole, endRP]
```

```
0.154119
```

```
VectorAngle[startPole, startRP]
```

```
0.154119
```

So the angular distance between the pole and the reference point remains the same. What about the angular distance between the two poles?

```
VectorAngle[startPole, obsZ]
```

```
2.22667
```

```
VectorAngle[endPole, obsZ]
```

```
2.22667
```

As we might have suspected, the angular distance between pole ${}_{E}P_{pa}$ and pole ${}_{E}P_{co}$ remains the same during finite displacement. It seems that we have created an internally consistent model of simple cycloidal relative motion. *Bueno.*

12.6 The cycloid module

To make our lives simpler, and to leave us time to play with the dog, pet the cat, nag the kids/students, and love (or appease) our significant others, we construct a module to compute the cycloid solution. The input data are the full angular velocity vector for the observer's plate (**obsΩ_**), the full angular velocity vector for the moving plate that the observer is looking at (**movΩ_**), the location vector to the reference point (**refPt_**), and the model time (**t_**).


```
cycloid[obsΩ_, movΩ_, refPt_, t_] := Module[
   {a, b, c, d, e, f, g, h, i, j, k, l, m1, m2, m3, m4, m5, n, p, answer}, a = {1, 0, 0};
   b = {0, 1, 0};
   c = {0, 0, 1};
   d = unitVect3D[obsΩ];
   e = Norm[obsΩ];
   f = unitVect3D[movΩ];
   g = Norm[movΩ];
   h = unitVect3D[d × f];
   i = h;
   j = unitVect3D[h × d];
   k = unitVect3D[i × f];
   m1 = {{k[[1]], k[[2]], k[[3]]}, {i[[1]], i[[2]], i[[3]]}, {f[[1]], f[[2]], f[[3]]}};
   m2 = {{Cos[(-g Degree) t], Sin[(-g Degree) t], 0},
     {-Sin[(-g Degree) t], Cos[(-g Degree) t], 0}, {0, 0, 1}};
   l = VectorAngle[d, k];
   If[l < (π / 2), (n = 1), (n = (-1))];
   p = (n) (VectorAngle[f, d]);
   m3 = {{Cos[p], 0, -Sin[p]}, {0, 1, 0}, {Sin[p], 0, Cos[p]}};
   m4 = {{Cos[(e Degree) t], Sin[(e Degree) t], 0},
     {-Sin[(e Degree) t], Cos[(e Degree) t], 0}, {0, 0, 1}};
   m5 = {{j[[1]], h[[1]], d[[1]]}, {j[[2]], h[[2]], d[[2]]}, {j[[3]], h[[3]], d[[3]]}};
   answer = m5.m4.m3.m2.m1.refPt;
   answer];
```

Let's see if it gives us the same answers as we derived above, which were called **`result`**.

```
result
```

```
{33.9303, -117.952}
```

```
findGeogCoord[cycloid[eΩpa, eΩco, startRP, modT]]
```

```
{33.9303, -117.952}
```

The cycloid module is a good tool.

Exercise 12-1. The Pacific plate moves slowly to the northwest relative to someone on the North American plate. City Hall in Los Angeles is built on the Pacific plate, sort of. Actually, it is built atop a deep basin created in the borderland between the Pacific and North American plates. But it is located west of the San Andreas fault, and we shall simplify reality just like most textbooks and assume that THE SAN ANDREAS FAULT IS THE PLATE BOUNDARY between the Pacific and North American plates. (That's not really true. The San Andreas fault is one very important part of a complex boundary zone, but...) Relative to the North American plate, where will LA City Hall be in 1 million years if its current location is 34.05°N, 118.24°W? Copy the cycloid module into a new *Mathematica* notebook and use the data from NNR-MORVEL56 to compute the new location.

Exercise 12-2. City Hall in San Francisco is located at 37.78°N, 122.41°W, on the North American plate, sort of. Play along with the fantasy... Use the *Mathematica* notebook you developed in the previous problem to estimate, to the nearest half-million years, when City Hall in Los Angeles will pass most closely by City Hall in San Francisco. You might want to add a bit of code to determine the distance between SF City Hall and the output of your previous model at different input times. When the city halls experience their closest pass, what general direction will LA City Hall be relative to SF City Hall? ...and why would that be interesting?

12.7 Finite motion of ${}_{A}\Omega_{B}$ relative to plate A

Where would the pole ${}_{pa}P_{co}$ be after our brief model displacement of 1 million years? Relative to the observer on the Pacific plate, the ${}_{E}\Omega_{pa}$ angular velocity vector is unchanged, but the ${}_{E}\Omega_{co}$ vector is displaced by a rotation around the ${}_{E}\Omega_{pa}$ axis. The new ${}_{E}\Omega_{co}$ vector is given by

```
endΩco = movω endPole
```

```
{-0.595518, -0.889476, 0.537954}
```

and because ${}_{pa}\Omega_{co} = {}_{E}\Omega_{co} - {}_{E}\Omega_{pa}$, we can compute the new location of pole ${}_{pa}P_{co}$. The initial angular velocity vector ${}_{pa}\Omega_{co}$ at model time = 0 is given by

```
paΩco0 = eΩco - eΩpa
```

```
{-0.48101, -1.1454, 1.12558}
```

and the angular velocity vector at model time = 1 is given by

```
paΩco1 = endΩco - eΩpa
```

```
{-0.474478, -1.15264, 1.12096}
```

The initial Pacific-Cocos pole location was

```
findGeogCoord[unitVect3D[paΩco0]]
```

```
{42.178, -112.78}
```

and the final location after 1 million years, relative to the fixed Pacific plate, is

```
findGeogCoord[unitVect3D[paΩco1]]
```

```
{41.9652, -112.374}
```

The reference point started out an angular distance of

```
VectorAngle[paΩco0, startRP] / Degree
```

```
9.18837
```

degrees from the instantaneous pole of relative motion between the Pacific and Cocos plates, but it ended up an angular distance of

```
VectorAngle[paΩco1, endRP] / Degree
```

```
9.15475
```

degrees from the instantaneous pole of relative motion between the Pacific and Cocos plates after 1 million years. The angular distance from the reference point to the pole decreased, so the tangential velocity of the reference point relative to the Pacific plate has decreased. It has decelerated.

Exercise 12-3. Between today and the time when San Francisco and Los Angeles converge, does the model indicate that the tangential velocity of Los Angeles will accelerate, decelerate or remain constant? Support your answer with appropriate computation.

12.8 Cycloidal trajectories

We all know what a circle looks like. What does a cycloid on a sphere look like? To provide a couple of answers to that question, we will use a triangle to represent the moving plate, and look at its motion from -500 Myr to +500 Myr. First, we will model the triangle as the moving Pacific plate relative to North America, then we will look at the moving North America relative to the Pacific Plate. We assume that ${}_{E}\Omega_{na}$ and ${}_{E}\Omega_{pa}$ (and hence ${}_{na}\Omega_{pa}$) are constant during that finite time interval.

(In the traditional model of circular finite motion, it would be unnecessary to construct two models -- one each from the perspective of the Pacific and North American plates. The trajectories would look the same in both cases. That is not true of cycloidal finite relative motion.)

First, we define the apices of a triangle (**tri**) that we will use to represent the Pacific plate,

```
tri = {convert2Cart[34.05, -118.24], convert2Cart[28.25, -120.45],
   convert2Cart[32.5, -125.25], convert2Cart[34.05, -118.24]};
```

and then use the cycloid module to find the triangle's position relative to the North American plate at 50 Myr intervals from -500 Myr to +500 Myr.

```
triangles = Table[cycloid[eΩna, eΩpa, tri[[i]], j], {j, -500, 500, 50}, {i, 1, 4, 1}];
```

Second, we will use the cycloid module to define the trajectory of each of the apices from -500 Myr to +500 Myr, using a finer step interval of 5 Myr.

```
curve1 = Table[cycloid[eΩna, eΩpa, tri[[1]], i], {i, -500, 500, 5}];
curve2 = Table[cycloid[eΩna, eΩpa, tri[[2]], i], {i, -500, 500, 5}];
curve3 = Table[cycloid[eΩna, eΩpa, tri[[3]], i], {i, -500, 500, 5}];
```

Third, we find the locations of the moving plate's pole and the pole of relative motion between the two plates at 50 Myr intervals.

```
naΩpa = eΩpa - eΩna;

naPpa = unitVect3D[naΩpa];

paPna = -naPpa;

ePpa = unitVect3D[eΩpa];

ePna = unitVect3D[eΩna];

relativePoles = Table[circMotion[naPpa, -eΩna, i], {i, -500, 500, 50}];
movPlatePoles = Table[circMotion[ePpa, -eΩna, i], {i, -500, 500, 50}];
```

Fourth, we prepare the output graphics files, and show the resulting graphic.

```
out1 = Graphics3D[{Opacity[0.75], Sphere[{0, 0, 0}, 1]},
   AspectRatio → 1, BoxRatios → {1, 1, 1}, PlotRange → All,
   PlotRangePadding → 0.1, ColorOutput → GrayLevel, Lighting → "Neutral"];

out2 = Graphics3D[Table[Line[triangles[[i]]], {i, 1, Length[triangles]}]];

out3 = Graphics3D[Line[Table[curve1[[i]], {i, 1, Length[curve1]}]]];

out4 = Graphics3D[Line[Table[curve2[[i]], {i, 1, Length[curve2]}]]];

out5 = Graphics3D[Line[Table[curve3[[i]], {i, 1, Length[curve3]}]]];

markers1 = tri;

out6 = ListPointPlot3D[markers1, AspectRatio → 1, BoxRatios → {1, 1, 1}, PlotStyle →
    Directive[Red, PointSize[Large]], PlotRange → All, PlotRangePadding → 0.1];

out7 = ListPointPlot3D[relativePoles, AspectRatio → 1, BoxRatios → {1, 1, 1},
   PlotStyle → Green, PlotRange → All, PlotRangePadding → 0.1];

out8 = ListPointPlot3D[movPlatePoles, AspectRatio → 1, BoxRatios → {1, 1, 1},
   PlotStyle → Blue, PlotRange → All, PlotRangePadding → 0.1];

markers2 = {ePna, -ePna};

out9 = ListPointPlot3D[markers2, AspectRatio → 1,
   BoxRatios → {1, 1, 1}, PlotStyle → Directive[Brown, PointSize[Large]],
   PlotRange → All, PlotRangePadding → 0.1];
```

```
Show[out1, out2, out3, out4, out5, out6, out7, out8, out9]
```

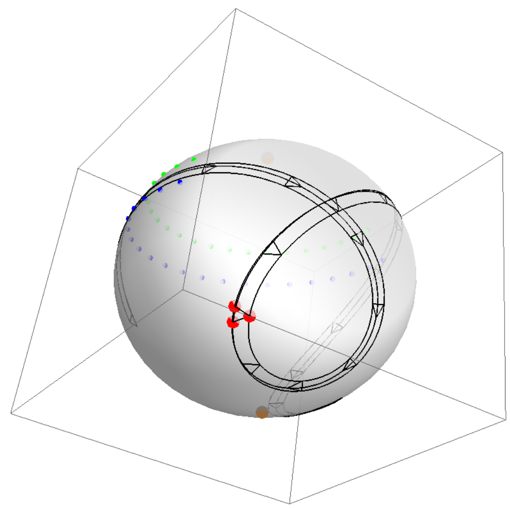

Figure 12-1. Modeled cycloid trajectory of the Pacific plate (triangle) relative to the North American plate from -500 Myr to +500 Myr at 50 Myr intervals, with corresponding flow lines. Red dots are on the apices of the reference triangle at model time = 0 (today). Blue dots mark the pole of the Pacific plate in the NNR reference frame, brown dots mark the North American pole, and green dots mark the pole of relative motion between the two plates -- all plotted at 50 Myr intervals

There are a number of important features of the output to notice. First, notice that the triangles are not evenly spaced, but that their spacing seems to vary in some sort of periodic manner. They are closer together in the looping cusp of the curve, and are farther apart away from the cusp. That indicates a change in tangential velocity of the moving plate relative to the observer's plate. Second, notice that the flow lines of each of the vertices of the triangle are a slightly different shape. Each point on the moving plate would trace a different trajectory from every other point to which it is rigidly coupled. As a consequence of this, there is no shape of plate boundary fault along which there could be purely strike slip without convergence or divergence. Third, notice that the pole of the Pacific plate (blue dots) and the pole of relative motion between the two plates (green dots) both move in concentric circles around the pole of the North American plate (brown dots).

Now, let's reverse the experiment and model how the North American plate would move relative to the Pacific plate during this same time interval. We will use the same triangle to model this relative motion.

```
triangles = Table[cycloid[eΩpa, eΩna, tri[[i]], j], {j, -500, 500, 50}, {i, 1, 4, 1}];
curve4 = Table[cycloid[eΩpa, eΩna, tri[[1]], i], {i, -500, 500, 5}];
curve5 = Table[cycloid[eΩpa, eΩna, tri[[2]], i], {i, -500, 500, 5}];
curve6 = Table[cycloid[eΩpa, eΩna, tri[[3]], i], {i, -500, 500, 5}];
```

```
relativePoles = Table[circMotion[paPna, -eΩpa, i], {i, -500, 500, 50}];
movPlatePoles = Table[circMotion[ePna, -eΩpa, i], {i, -500, 500, 50}];
```

```
out10 = Graphics3D[Table[Line[triangles[[i]]], {i, 1, Length[triangles]}]];
```

```
out11 = Graphics3D[Line[Table[curve4[[i]], {i, 1, Length[curve1]}]]];
```

```
out12 = Graphics3D[Line[Table[curve5[[i]], {i, 1, Length[curve2]}]]];
```

```
out13 = Graphics3D[Line[Table[curve6[[i]], {i, 1, Length[curve3]}]]];
```

```
markers3 = tri;
```

```
out14 = ListPointPlot3D[markers3, AspectRatio → 1, BoxRatios → {1, 1, 1}, PlotStyle →
    Directive[Red, PointSize[Large]], PlotRange → All, PlotRangePadding → 0.1];
```



```
out15 = ListPointPlot3D[relativePoles, AspectRatio → 1, BoxRatios → {1, 1, 1},
   PlotStyle → Green, PlotRange → All, PlotRangePadding → 0.1];
out16 = ListPointPlot3D[movPlatePoles, AspectRatio → 1, BoxRatios → {1, 1, 1},
   PlotStyle → Brown, PlotRange → All, PlotRangePadding → 0.1];
markers4 = {ePpa, -ePpa};
out17 = ListPointPlot3D[markers4, AspectRatio → 1,
   BoxRatios → {1, 1, 1}, PlotStyle → Directive[Blue, PointSize[Large]],
   PlotRange → All, PlotRangePadding → 0.1];
Show[out1, out10, out11, out12, out13, out14, out15, out16, out17]
```

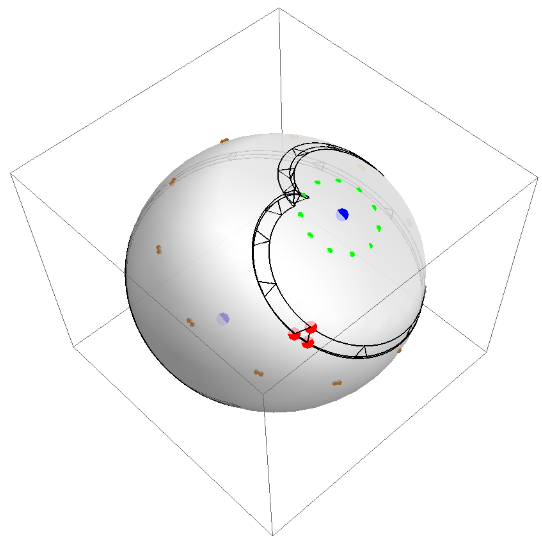

Figure 12-2. Modeled cycloid trajectory of the North American plate (triangle) relative to the Pacific plate from -500 Myr to +500 Myr at 50 Myr intervals, with corresponding flow lines. Red dots are on the apices of the reference triangle at model time = 0 (today). Blue dots mark the pole of the Pacific plate, brown dots mark the North American pole, and green dots mark the pole of relative motion between the two plates -- all plotted at 50 Myr intervals

The uneven spacing of the triangles representing the North American plate is even more dramatic in this figure than in the previous figure, illustrating that the tangential velocity of the North American plate relative to the Pacific plate is not constant. The change in the triangle's orientation in the cusp is dramatic.

12.9 Afterword

I am sometimes asked to compare the results of modeling based cycloidal finite motion with modeling based on circular finite motion. While I understand that people would like to understand how an unfamiliar model might differ from a familiar model, this comparison is problematic for me. In a world where the instantaneous plate-to-plate angular velocity vectors are not coaxial, circular finite motion between the plates is not an admissible model. Comparing an admissible model with an inadmissible model seems like a waste of time to me.

The cycloid finite motion model as presented above is the simplest finite motion model that can be used to extend from instantaneous motion data to finite motion in cases where the plate-to-plate angular velocity vectors are not coaxial. Does that mean that the motion of a point on one plate relative to another plate is necessarily cycloidal? No, it does not. Given additional data as might be provided by marine magnetic anomalies or other finite motion indicators preserved in the geologic record, the full finite motion of selected plates might be discerned. We will see how this can be done in future chapters.

12.10 References and some relevant texts

Argus, D.F., Gordon, R.G., and DeMets, C., 2011, Geologically current motion of 56 plates relative to the no-net-rotation reference frame: Geochemistry, Geophysics, Geosystems, v. 12, no. 11, Q11001 doi:10.1029/2011GC003751.

Cronin, V.S., 1987, Cycloid kinematics of relative plate motion: Geology, v. 15, p. 1006-1009.

Cronin, V.S., 1988, Cycloid tectonics -- a kinematic model of finite relative plate motion: College Station, Texas A&M University, Department of Geology, dissertation, 118 p.

Cronin, V.S., 1991, The cycloid relative-motion model and the kinematics of transform faulting, *in* Hilde, T.W.C., and Carlson, R.L., [editors], Proceedings of the 1987 Geodynamics Symposium: Tectonophysics, v. 187, p. 215-249; corrections to publisher errors, 1991, Tectonophysics, v. 192, no. 3/4, p. 401.

Cronin, V.S., 1992, Types and kinematic stability of triple junctions: Tectonophysics, v. 207, p. 287-301.

Cronin, V.S., 1994, Instantaneous velocity of mid-ocean ridges: Tectonophysics, v. 230, p. 151-159.

DeMets, C., Gordon, R.G., and Argus, D.F., 2010, Geologically current plate motions: Geophysical Journal International, v. 121, p. x, doi: 10.1111/j.1365-246X.2009.04491.x http://www.geology.wisc.edu /~chuck/MORVEL/

Gripp, A.E., and Gordon, R.G., 2002, Young tracks of hotspots and current plate velocities: Geophysical Journal International, v. 150, p. 321-361.

McKenzie, D.P., and Morgan, W.J., 1969, The evolution of triple junctions: Nature, v. 224, p. 125-133.

