

Chapter 7. Instantaneous plate kinematics in an imaginary 2-plate world

© 1988 – 2015 by Vincent S. Cronin; revised 28 February 2017

7.1 Introduction

We have been developing a set of tools that we will use to model the instantaneous and finite motion of plates. Instantaneous motion occurs over short time intervals (effectively sometime less than ~3 million years) and involves small displacements. Finite motion occurs over longer time intervals and involves displacements of many kilometers.

In this chapter, we are going to work with an imaginary planet that has just two plates. We will import an external datafile containing the geographic coordinates of a set of points around the boundary between the two plates. We will model the motion of the smaller plate as observed from a frame of reference that is fixed to the bigger plate. We will start by illustrating the circular finite motion of a set of points around the perimeter of the smaller plate around a pole of relative motion that is fixed with respect to both plates. Then we will learn how to use a polygon rather than a set of points to represent the plate boundaries. Finally, we will shrink the time interval down to the extent that we are effectively looking at instantaneous motion, and illustrate the tangential velocities using vector graphics.

That sounds like a lot of work. Actually, it's just good nerdy fun. Let's get started.

7.2 Some user-defined functions

We will use some of the user-defined functions developed in previous chapters.

```
In[281]:= convert2Cart[lat_, long_] := {Cos[lat Degree] Cos[long Degree],  
    Cos[lat Degree] Sin[long Degree], Sin[lat Degree]};  
  
In[282]:= unitVect3D[ vect_ ] :=  
    {(vect[[1]] / Norm[vect]), (vect[[2]] / Norm[vect]), (vect[[3]] / Norm[vect])};  
  
In[283]:= zRotation[w_, dT_] := {{Cos[(w dT) Degree], -Sin[(w dT) Degree], 0},  
    {Sin[(w dT) Degree], Cos[(w dT) Degree], 0}, {0, 0, 1}};  
  
In[284]:= circMotion[x_, m1_, w_, dT_] := Module[{m2, m3, answer}, m2 = zRotation[w, dT];  
    m3 = Inverse[m1];  
    answer = m3.m2.m1.x;  
    answer];
```

7.3 Importing the plate-boundary file

The input data that we will use in this chapter is an Excel spreadsheet (.xls file) in which each record (that is, each horizontal row) has 2 values/columns. The first column contains the latitude of a point in degrees (range of latitude is -90° at the south pole to 90° at the north pole), and the second column has the longitude of the point in degrees (range of longitude is -180° to 180° , with west longitudes as negative numbers). An Excel file called otherPlate.xls has been made available to you, or can be downloaded from

<http://croninprojects.org/Vince/PlateKinematics/KinematicsPrimer/otherPlate.xls>

Finding the Excel data file on your computer

Each user will need to ensure that the path to the input data file is correctly specified in the first input line of this notebook in the blue box below. For example, a correct specification for the file "otherPlate.xls" located on Vince Cronin's MacBook Pro computer would look like this:

```
mydata = Import[\"/Users/vincecronin/Desktop/otherPlate.xls\"];
```

In the code immediately following this text, be certain that the path to the input data file is correctly specified.

```
In[285]:= mydata = Import[\"/Users/vince/Desktop/otherPlate.xls\"];
```

Let's look at the input file

```
In[286]:= mydata
```

```
Out[286]= {{{{26.6199, 2.68982}, {21.7168, 3.76375}, {16.8072, 4.76668}, {17.5893, 8.69125},
{18.5038, 12.6027}, {19.5458, 16.5011}, {14.7523, 17.9898}, {9.94967, 19.4141},
{5.14118, 20.7971}, {6.44126, 25.1602}, {7.86389, 29.4963}, {9.40022, 33.8082},
{4.71448, 35.5667}, {0.0243823, 37.3016}, {1.74554, 41.8147},
{3.55745, 46.2964}, {5.4485, 50.7543}, {7.40691, 55.1971}, {9.42076, 59.6347},
{11.478, 64.0782}, {13.5663, 68.5393}, {15.6731, 73.0309}, {20.1566, 70.7042},
{24.6078, 68.2402}, {29.0159, 65.595}, {33.3676, 62.7151}, {37.6459, 59.5342},
{41.8284, 55.9686}, {45.8853, 51.912}, {43.9692, 48.2676}, {42.1131, 44.6778},
{40.3233, 41.1292}, {38.6061, 37.6105}, {36.9674, 34.1124}, {35.413, 30.6272},
{33.9483, 27.1488}, {32.5787, 23.672}, {31.3092, 20.193}, {30.1446, 16.7087},
{29.0894, 13.2172}, {28.1477, 9.71717}, {27.3234, 6.20801}, {26.6199, 2.68982}}}}
```

We need to use the built-in *Mathematica* function **Flatten** to remove one of the dimensions of this array -- that is, to get rid of one set of curly brackets that enclose the dataset. We will call this new and improved datafile **inData**.

```
In[287]:= inData = Flatten[mydata, 1];
```

Now, we need to convert the data of **inData** to Cartesian geographic coordinates, using the user-defined function **convert2Cart** that we developed in an earlier chapter, and the built-in *Mathematica* function **Table**. We specify the latitude of the record in row **i** using **inData[[i,1]]**, and the longitude is **inData[[i,2]]**. The total number of rows or records in the datafile is given by **Length[inData]**.

```
In[288]:= inDataCart =
Table[convert2Cart[inData[[i, 1]], inData[[i, 2]]], {i, 1, Length[inData]}];
```

If we were now to plot these points on the surface of a unit sphere, they would look like the red dots in Figure 7-1.

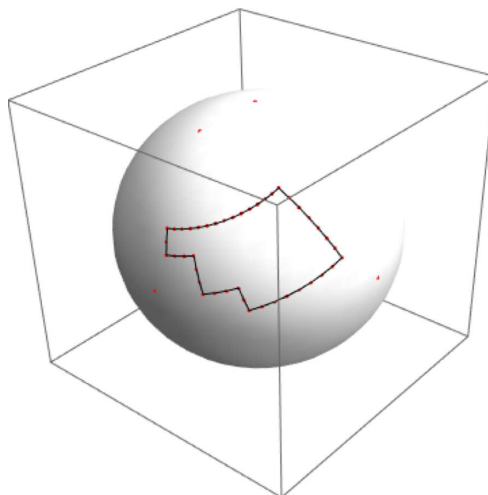


Figure 7-1. The graphic displayed above is a static image. The red dots are plotted where the positive X, Y and Z axes intersect the sphere, at the pole of rotation located at latitude 65°N longitude 15°W, and at points along the initial boundary between the plates. The plate boundary lines are either concentric with the pole or radial to the pole.

7.4 Circular motion of points around the edge of one plate as observed from the other plate

The data that we just imported are the locations of points around an imaginary plate that was carefully constructed so that its boundaries are either radial to or concentric with a point located at 65°N latitude and 15°W longitude. This focal point will be used as the pole of relative motion between the small plate and the other plate, which comprises the entire rest of the planet's surface.

The unit vectors along the geographic Cartesian axes are

```
In[289]:= zGeog = {0, 0, 1}; xGeog = {1, 0, 0}; yGeog = {0, 1, 0};
```

We will choose a positive pole located at latitude 65° (in the northern hemisphere) and longitude -15° (in the western hemisphere). In the Cartesian geographic coordinate system, the pole's unit location vector is given by

```
In[290]:= poleLat = 65; poleLong = -15;
```

```
In[291]:= pole = convert2Cart[poleLat, poleLong];
```

```
In[292]:= xRot = unitVect3D[Cross[pole, zGeog]];
```

```
In[293]:= yRot = unitVect3D[Cross[pole, xRot]];
```

```
In[294]:= zRot = pole;
```

The matrix that allows us to transform from the Cartesian geographic coordinate system (**xGeog**, **yGeog**, **zGeog**) to the rotating coordinate system (**xRot**, **yRot**, **zRot**) is

```
In[295]:= j1 = 
$$\begin{pmatrix} \text{Dot}[x\text{Geog}, x\text{Rot}] & \text{Dot}[y\text{Geog}, x\text{Rot}] & \text{Dot}[z\text{Geog}, x\text{Rot}] \\ \text{Dot}[x\text{Geog}, y\text{Rot}] & \text{Dot}[y\text{Geog}, y\text{Rot}] & \text{Dot}[z\text{Geog}, y\text{Rot}] \\ \text{Dot}[x\text{Geog}, z\text{Rot}] & \text{Dot}[y\text{Geog}, z\text{Rot}] & \text{Dot}[z\text{Geog}, z\text{Rot}] \end{pmatrix};$$

```

and the matrix that allows us to transform from the rotating coordinate system (**xRot**, **yRot**, **zRot**) back to the Cartesian geographic coordinate system (**xGeog**, **yGeog**, **zGeog**) is

```
In[296]:= j3 = Inverse[j1];
```

Current angular velocities of one plate as observed from another plate range from ~0.133°/Myr between the Lwandle and Antarctic plates to 4.966°/Myr between the Rivera and Pacific plates (DeMets and others, 2010). We will use a geologically reasonable (and mathematically appealing) angular velocity of 1°/Myr in this section, and look at displacements at 4 Myr intervals from the present out to 20 Myr in the future.

```
In[297]:=  $\omega = 1;$ 
```

```
In[298]:= minT = 4; maxT = 20; stepT = 4;
```

As we did in the previous chapter, we use the function **circMotion** to compute locations of our reference points at the specified times, and call the resulting data file **prelimResults**.

```
In[299]:= prelimResults = Table[circMotion[inDataCart[[i]], j1,  $\omega$ , deltaT],
  {i, 1, Length[inDataCart]}, {deltaT, minT, maxT, stepT}];
```

We reduce the dimension of the **prelimResults** matrix so that it is a simple n -row, 3-column matrix where n is the number of time steps multiplied by the number of reference points.

```
In[300]:= outResults = Flatten[prelimResults, 1];
```

Then we prepare various graphics files that will be combined to illustrate the results in a 3D graphic.

```
In[301]:= graphics1 = Graphics3D[Sphere[{0, 0, 0}, 1],
  AspectRatio  $\rightarrow$  1, BoxRatios  $\rightarrow$  {1, 1, 1}, PlotRange  $\rightarrow$  All,
  PlotRangePadding  $\rightarrow$  0.1, ColorOutput  $\rightarrow$  GrayLevel, Lighting  $\rightarrow$  "Neutral"];

```

```
In[302]:= graphics2 = ListPointPlot3D[outResults, AspectRatio  $\rightarrow$  1, BoxRatios  $\rightarrow$  {1, 1, 1},
  PlotStyle  $\rightarrow$  Green, PlotRange  $\rightarrow$  All, PlotRangePadding  $\rightarrow$  0.1];
```

```

In[303]:= graphics3 = ListPointPlot3D[inDataCart, AspectRatio → 1,
    BoxRatios → {1, 1, 1}, PlotStyle → Red, PlotRange → All, PlotRangePadding → 0.1];
In[304]:= markers2 = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {pole[[1]], pole[[2]], pole[[3]]}};
In[305]:= graphics4 = ListPointPlot3D[markers2, AspectRatio → 1, BoxRatios → {1, 1, 1},
    PlotStyle → Red, PlotRange → All, PlotRangePadding → 0.1];
In[306]:= Show[graphics1, graphics2, graphics3, graphics4]

```

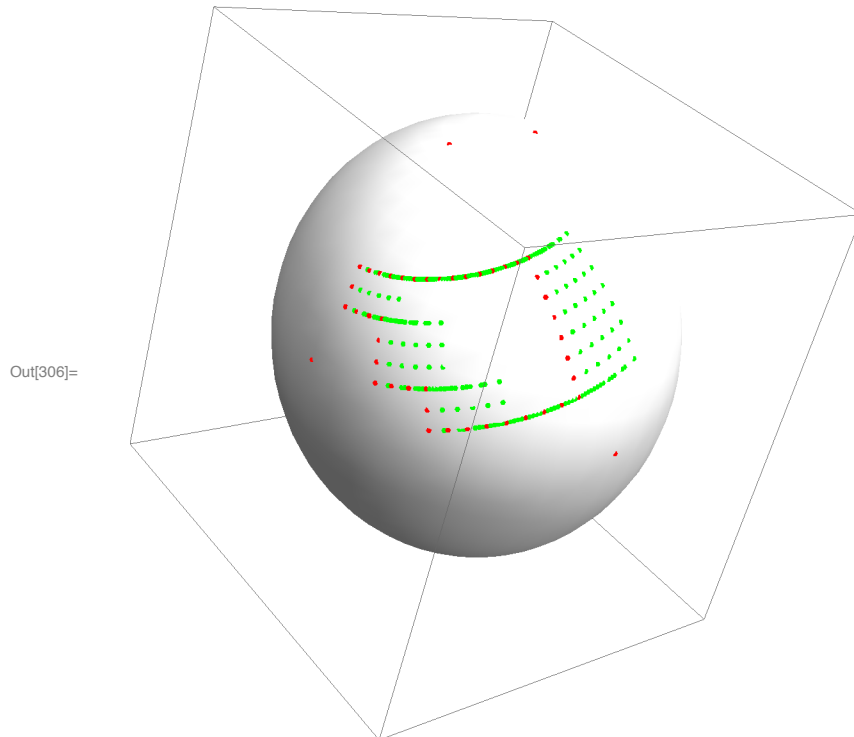


Figure 7-2. The graphic displayed above can be manipulated with the computer's mouse or trackpad. The red dots are plotted where the positive X, Y and Z axes intersect the sphere, at the pole of rotation located at latitude 65°N longitude 15°W, and at points along the initial boundary between the plates. The green dots are the locations of points along the boundary of the small plate that is moving as observed from the large plate at equal time intervals.

We can use the built-in *Mathematica* function **Line** to draw a line around the boundaries between the two plates at the initial time.

```

In[307]:= graphics5 = Graphics3D[Line[inDataCart]];

```

```
In[308]:= Show[graphics1, graphics2, graphics3, graphics4, graphics5]
```

```
Out[308]=
```

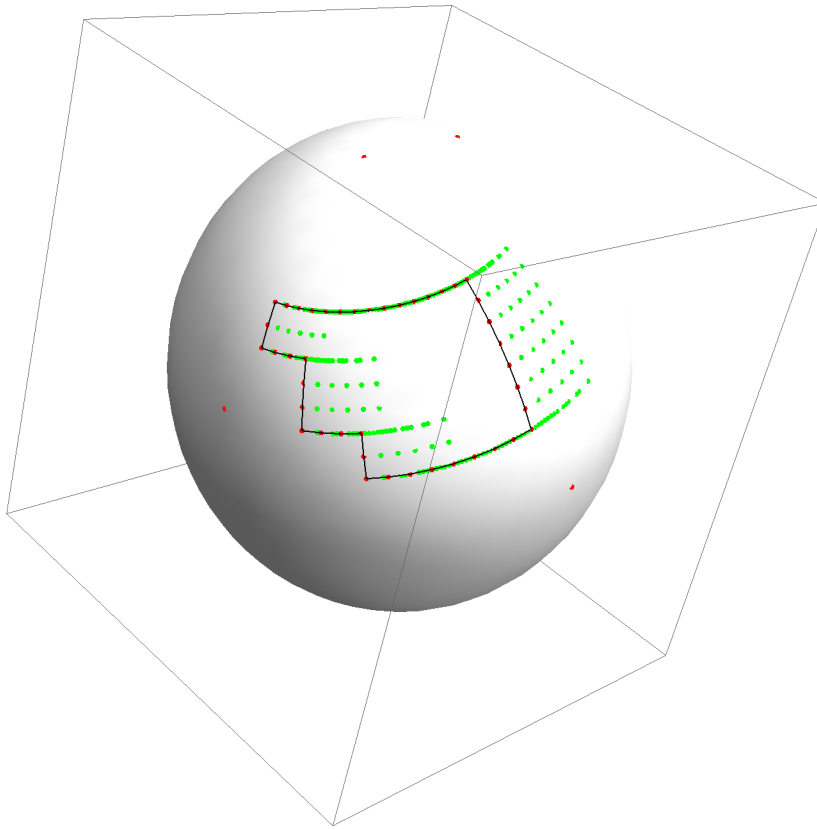


Figure 7-3. The black polygon is a set of lines through the red dots that mark the initial location of the plate boundary. The other red dots are plotted where the positive X, Y and Z axes intersect the sphere, and at the pole of rotation located at latitude 65°N longitude 15°W. The green dots are the locations of points along the boundary of the small plate that is moving as observed from the large plate at equal time intervals.

Exercise 7-1. Copy Figure 7-3 into a separate document by clicking on the image, choosing **Save Selection As...** from the **File** menu, and saving it as a PDF or other file. Make a hard-copy print of the figure, and do the following. (a) Label each segment of the boundary outlined by the black-line polygon in Figure 7.3 as either a convergent, divergent or transform boundary. (b) How would you describe the displacement along the transform faults between these two plates? (c) Assuming that sea-floor spreading occurs along the divergent boundaries and subduction occurs along convergent boundaries, is the smaller plate getting bigger or smaller over time? (d) What can you say about the rate at which the divergent boundary moves relative to the larger plate? (e) What shape would the oceanic fracture zones have, and what spatial relationships would they have with respect to the rotational pole or with each other?

Assignment. Read Morgan (1968).

A brief sidetrip

Now let's introduce a little bit of mayhem into the system by using the same plate system but changing the pole of rotation. Remember, the plate-shape data was optimized for a pole of rotation located at 65°N latitude and 15°W longitude. Now we are going to see what will happen if we take this plate and move it around a different pole, located at 55°N latitude, 25°W longitude.

```
In[309]:= Clear[poleLat, poleLong, pole, xRot, yRot, zRot, j1, j3, prelimResults,
             outResults, markers2, graphics1, graphics2, graphics3, graphics4, graphics5];
```

```
In[310]:= poleLat = 55; poleLong = -25;
```

```

In[311]:= pole = convert2Cart[poleLat, poleLong];
          zRot = pole;
          xRot = unitVect3D[Cross[pole, zGeog]];
          yRot = unitVect3D[Cross[pole, xRot]];

In[312]:= j1 =  $\begin{pmatrix} \text{Dot}[x\text{Geog}, x\text{Rot}] & \text{Dot}[y\text{Geog}, x\text{Rot}] & \text{Dot}[z\text{Geog}, x\text{Rot}] \\ \text{Dot}[x\text{Geog}, y\text{Rot}] & \text{Dot}[y\text{Geog}, y\text{Rot}] & \text{Dot}[z\text{Geog}, y\text{Rot}] \\ \text{Dot}[x\text{Geog}, z\text{Rot}] & \text{Dot}[y\text{Geog}, z\text{Rot}] & \text{Dot}[z\text{Geog}, z\text{Rot}] \end{pmatrix}$ ;

In[313]:= j3 = Inverse[j1];

In[314]:=  $\omega = 1$ ;

In[315]:= minT = 4; maxT = 20; stepT = 4;

In[316]:= prelimResults = Table[circMotion[inDataCart[[i]], j1,  $\omega$ , deltaT],
                                {i, 1, Length[inDataCart]}, {deltaT, minT, maxT, stepT}];

In[317]:= outResults = Flatten[prelimResults, 1];

In[318]:= graphics1 = Graphics3D[Sphere[{0, 0, 0}, 1],
                                AspectRatio  $\rightarrow$  1, BoxRatios  $\rightarrow$  {1, 1, 1}, PlotRange  $\rightarrow$  All,
                                PlotRangePadding  $\rightarrow$  0.1, ColorOutput  $\rightarrow$  GrayLevel, Lighting  $\rightarrow$  "Neutral"];

In[319]:= graphics2 = ListPointPlot3D[outResults, AspectRatio  $\rightarrow$  1, BoxRatios  $\rightarrow$  {1, 1, 1},
                                PlotStyle  $\rightarrow$  Green, PlotRange  $\rightarrow$  All, PlotRangePadding  $\rightarrow$  0.1];

In[320]:= graphics3 = ListPointPlot3D[inDataCart, AspectRatio  $\rightarrow$  1,
                                BoxRatios  $\rightarrow$  {1, 1, 1}, PlotStyle  $\rightarrow$  Red, PlotRange  $\rightarrow$  All, PlotRangePadding  $\rightarrow$  0.1];

In[321]:= markers2 = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {pole[[1]], pole[[2]], pole[[3]]}};

In[322]:= graphics4 = ListPointPlot3D[markers2, AspectRatio  $\rightarrow$  1, BoxRatios  $\rightarrow$  {1, 1, 1},
                                PlotStyle  $\rightarrow$  Red, PlotRange  $\rightarrow$  All, PlotRangePadding  $\rightarrow$  0.1];

In[323]:= graphics5 = Graphics3D[Line[inDataCart]];

```

```
In[324]:= Show[graphics1, graphics2, graphics3, graphics4, graphics5]
```

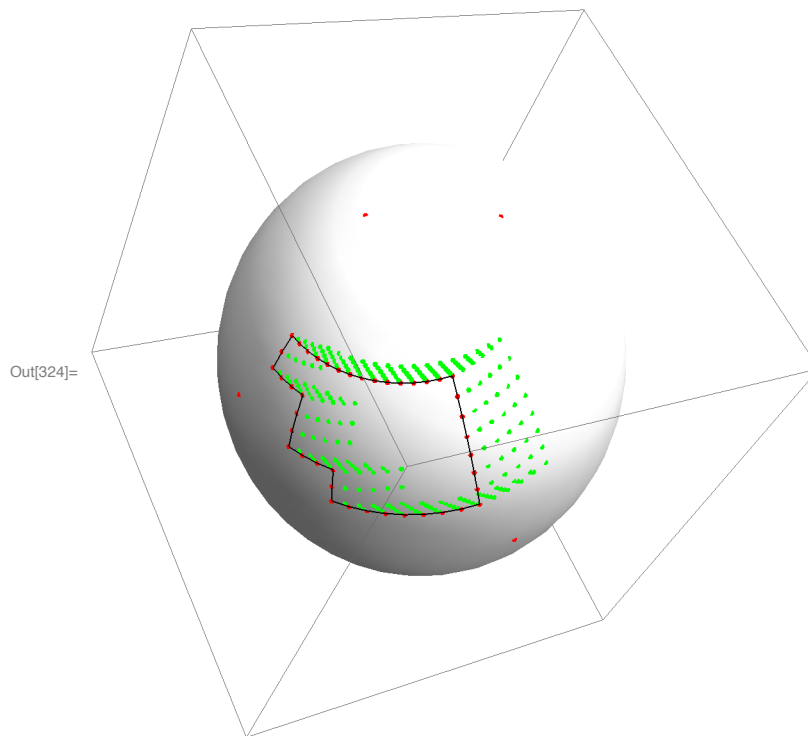


Figure 7-4. The black polygon is a set of lines through the red dots that mark the initial location of the plate boundary. The other red dots are plotted where the positive X, Y and Z axes intersect the sphere, and at the pole of rotation located at latitude 55°N longitude 25°W. The green dots are the locations of points along the boundary of the small plate that is moving as observed from the large plate at equal time intervals.

Exercise 7-2. What happens when the smaller plate does not rotate around the original pole of rotation, around which its transform faults and other boundaries were optimized?

7.5 References and relevant texts

DeMets, C., Gordon, R.G., and Argus, D.F., 2010, Geologically current plate motions: *Geophysical Journal International*, doi: 10.1111/j.1365-246X.2009.04491.x

McKenzie, D. & Slater, J. G., 1971 The evolution of the Indian Ocean since the late Cretaceous. *Geophys. J. R. astr. Soc.* 25, 437-528.

Morgan, W.J., 1968, Rises, trenches, great faults, and crustal blocks: *Journal of Geophysical Research*, v. 73, no. 6, p. 1959-1982.

The URL of the interactive map that shows where all of the Plate Boundary Observatory sites are located is <http://www.unavco.org/instrumentation/networks/status/pbo/>. Clicking on any of the circles that mark the location of individual PBO sites opens a dialog box containing some information about that particular site as well as a link to the site's overview page.