Chapter 9. Instantaneous plate kinematics in an *n*-plate world

9.1 Introduction

We have finally made it to an *n*-plate world, like the one we inhabit. OK, not exactly like the one we inhabit, because we are still going to use a spherical model of Earth. But working with *n* plates (where $n > 3$) is going to make our work much more interesting.

In this chapter, we are going to continue to work with instantaneous velocities of plates relative to each other. We will not work with the finite motion of plates, or with the instantaneous motion of plates relative to an external frame of reference as might be provided by hot spots, GPS, or other coordinate systems that are external to the lithospheric plates.

9.2 User-defined functions

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

```
In[1]:= convert2Cart[lat_, long_] := {Cos[lat Degree] Cos[long Degree],
            Cos[lat Degree] Sin[long Degree], Sin[lat Degree]};

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

In[3]:= 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}];
```

```
In[4]:= 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]
```

9.3 Instantaneous kinematics in an *n*-plate system

Permutations

There are three unique angular velocity vectors associated with the instantaneous relative motions among three plates, not counting the angular velocity vectors that are colinear to these three.

$$_A\Omega_B,\ _B\Omega_C,\ _C\Omega_A$$

How many unique angular velocity vectors are associated with 4 plates: A, B, C and D?

$$_A\Omega_B,\ _B\Omega_C,\ _C\Omega_A,\ _C\Omega_D,\ _D\Omega_A,\ _B\Omega_D$$

How many unique angular velocity vectors are associated with 5 plates: A, B, C, D and E?

$$_A\Omega_B,\ _B\Omega_C,\ _C\Omega_A,\ _C\Omega_D,\ _D\Omega_A,\ _B\Omega_D,\ _E\Omega_A,\ _E\Omega_B,\ _E\Omega_C,\ _E\Omega_D$$

There seems to be a pattern here. If I have three plates, each plate is involved in two angular velocity vectors. That suggests we should have 3 x 2 = 6 angular velocity vectors in our list, but since $_i\Omega_j$ is colinear with $_j\Omega_i$ for any plates i and j, the total number of unique angular velocity vectors in a 3-plate system is just 6/2 = 3 as we demonstrated above. If I have 4 plates, each plate is involved in 3 angular velocity vectors. So I should have (4 * 3)/2 = 6 unique angular velocity vectors. If I have 5 plates, each plate is involved in 4 angular velocity vectors. So I should have (5*4)/2 = 10 unique angular velocity vectors. The pattern seems to be as follows:

$$\text{Number of unique (non-colinear) angular velocity vectors given } n \text{ plates} = \frac{(n*(n-1))}{2}$$

Exercise 9-1. (a) Determine the number of unique angular velocity vectors that would be associated with an 11-plate system, as considered by Minster and Jordan's RM2 model (1978).
(b) The table presenting the RM2 rotational poles lists 27 poles and angular velocities. Why do you think they just listed 27 poles?

Exercise 9-2. In chapter 8 we determined that the instantaneous angular velocity vectors for a given 3-plate system ($_A\Omega_B$, $_B\Omega_C$, $_C\Omega_A$) must be coplanar or else the corresponding vector circuit would not close.
After giving sufficient thought to the matter, explain (and, if possible, demonstrate) why you think all of the instantaneous angular velocity vectors for a given 4-plate system (choose one: "must be" or "need not be") coplanar.

A 4-plate system

There are six angular velocity vectors associated with 4 plates: A, B, C and D.

$$_A\Omega_B,\ _B\Omega_C,\ _C\Omega_A,\ _C\Omega_D,\ _D\Omega_A,\ _B\Omega_D$$

Given four plates, there are four sets of 3-plate systems: ABC, ABD, ACD and BCD. Each of these triplets have instantaneous angular velocity vectors that are coplanar, so the rotational poles for a given triplet are located along the same great circle. Each intersection of the great circles is located at the pole of rotation that is common to both 3-plate systems. For example, the ABC great circle includes the AB, BC and CA poles, while the ABD great circle includes the AB, BD and DA poles. The two great circles intersect at the AB pole.

Let's build a graphic to show an example of how the great circles and poles of rotation for a 4-plate system relate to each other. We will take a plate system considered by Minster and Jordan (1978): Pacific, North America, Cocos and Nazca.

FOR Plate	**Moving Plate**	**Lat (°N)**	**Long (°E)**	$_{\text{FOR}}\omega_{\text{Moving}}$ **(°/Myr)**
PCFC	NOAM	48.77	−73.91	0.852
PCFC	COCO	38.72	−107.39	2.208
PCFC	NAZC	56.64	−87.88	1.539
NOAM	COCO	29.80	−121.28	1.489
NAZC	COCO	5.63	−124.40	0.972
NOAM	NAZC	−−	−−	−−

Notice that the North American-Nazca pole is not included in the published data, because the two plates do not have a common boundary. So one of our tasks (just to prove we can do it) will be to fill-in the missing data for this plate pair.

We start by defining the unit location vectors to the various poles of rotation.

```
In[5]:= pcfcPnoam = convert2Cart[48.77, -73.91];
        noamPpcfc = -1 * pcfcPnoam; pcfcωnoam = 0.852;
```

```
In[6]:= pcfcPcoco = convert2Cart[38.72, -107.93];
        cocoPpcfc = -1 * pcfcPcoco;
        pcfcωcoco = 2.208;
```

```
In[8]:= pcfcPnazc = convert2Cart[56.64, -87.88];
        nazcPpcfc = -1 * pcfcPnazc;
        pcfcωnazc = 1.539;
```

```
In[10]:= noamPcoco = convert2Cart[29.80, -121.28];
         cocoPnoam = -1 * noamPcoco;
         noamωcoco = 1.489;
```

```
In[12]:= nazcPcoco = convert2Cart[5.63, -124.40];
         cocoPnazc = -1 * nazcPcoco;
         nazcωcoco = 0.972;
```

Filling-in missing data

We will find the missing data, knowing that

$$_{\text{NAZC}}\Omega_{\text{NOAM}} = {}_{\text{NAZC}}\Omega_{\text{PCFC}} + {}_{\text{PCFC}}\Omega_{\text{NOAM}}$$

and

$$_{\text{NAZC}}\Omega_{\text{NOAM}} = {}_{\text{NAZC}}\Omega_{\text{COCO}} + {}_{\text{COCO}}\Omega_{\text{NOAM}}$$

based on our discussion of closed vector circuits in chapter 8.

```
In[14]:= nazcΩnoam1 = (pcfcωnazc * nazcPpcfc) + (pcfcωnoam * pcfcPnoam);
         nazcPnoam1 = unitVect3D[nazcΩnoam1];
         nazcωnoam1 = Norm[nazcΩnoam1];

In[16]:= nazcΩnoam2 = (nazcωcoco * nazcPcoco) + (noamωcoco * cocoPnoam);
         nazcPnoam2 = unitVect3D[nazcΩnoam2];
         nazcωnoam2 = Norm[nazcΩnoam2];
```

Let’s take a look at the output values to see how they compare.

```
In[18]:= {nazcΩnoam1, nazcΩnoam2}

Out[18]= {{0.124322, 0.306171, -0.644658},
          {0.124388, 0.306141, -0.644637}}

In[19]:= {nazcPnoam1, nazcPnoam2}

Out[19]= {{0.171617, 0.422645, -0.889898},
          {0.171712, 0.422615, -0.889894}}

In[20]:= {nazcωnoam1, nazcωnoam2}

Out[20]= {0.724418, 0.724398}
```

To my (shamefully lazy) eye, these values look to be close enough so that I am inclined to take the mean of each value to fill-in our missing data. Hence, the unit location vector to the Nazca-North American pole is

```
In[21]:= nazcPnoam = unitVect3D[((nazcPnoam1 + nazcPnoam2) / 2)]

Out[21]= {0.171664, 0.42263, -0.889896}

In[22]:= noamPnazc = -1 * nazcPnoam

Out[22]= {-0.171664, -0.42263, 0.889896}

In[23]:= nazcωnoam = ((nazcωnoam1 + nazcωnoam2) / 2)

Out[23]= 0.724408
```

And if you want to fill-in the blanks in the table above, the geographic coordinates of the pole around which Nazca rotates counter-clockwise relative to North America are given by

```
In[24]:= findGeogCoord[noamPnazc]

Out[24]= {62.8602, -112.106}
```

Illustrating the intersecting great circles and rotational poles

Sometime between a faculty meeting, tennis practice, and a meeting at my son’s high school, I wrote a bit of code called **`makeGreatCircle`** used in plotting a great circle whose orientation is defined by the coordinates of a vector that is normal to the great circle. You’re welcome. An explanation of this function is available in the appendix. Let's prepare some nails to hit with this new hammer.

We know that the closure of the Nazca-Pacific-Cocos vector circuit is not perfect

```
In[25]:= (nazcωcoco * nazcPcoco) +
          (pcfcωcoco * cocoPpcfc) + (pcfcωnazc * pcfcPnazc)

Out[25]= {0.0151517, -0.00481283, -0.000358135}
```

If closure was perfect, it would have generated {0, 0, 0} as the previous result. Given perfect data that achieve closure, the vector that is normal to the rotation vectors in the Nazca-Pacific-Cocos system could be defined by taking the cross product of any two non-colinear location vectors to the relevant poles (${}_{NAZC}P_{PCFC}$, ${}_{PCFC}P_{NAZC}$, ${}_{NAZC}P_{COCO}$, ${}_{COCO}P_{NAZC}$, ${}_{COCO}P_{PCFC}$ or ${}_{PCFC}P_{COCO}$). But because we do not have perfect data, we will take the average of three cross products.

```
In[26]:= normal1a = unitVect3D[Cross[nazcPcoco, cocoPpcfc]]

Out[26]= {0.744635, -0.55429, -0.371862}

In[27]:= normal1b = unitVect3D[Cross[nazcPcoco, pcfcPnazc]]

Out[27]= {-0.740734, 0.552799, 0.381742}

In[28]:= normal1c = unitVect3D[Cross[cocoPpcfc, pcfcPnazc]]

Out[28]= {0.729373, -0.563223, -0.388323}

In[29]:= normal1 = unitVect3D[normal1a + normal1b + normal1c]

Out[29]= {0.733343, -0.564767, -0.378479}
```

In a similar manner, we define the following

```
In[30]:= normal2a = unitVect3D[Cross[nazcPnoam, noamPpcfc]];
         normal2b = unitVect3D[Cross[nazcPnoam, pcfcPnazc]];
         normal2c = unitVect3D[Cross[noamPpcfc, pcfcPnazc]];
         normal2 = unitVect3D[normal2a + normal2b + normal2c];

In[31]:= normal3a = unitVect3D[Cross[noamPcoco, cocoPpcfc]];
         normal3b = unitVect3D[Cross[noamPcoco, pcfcPnoam]];
         normal3c = unitVect3D[Cross[cocoPpcfc, pcfcPnoam]];
         normal3 = unitVect3D[normal3a + normal3b + normal3c];

In[32]:= normal4a = unitVect3D[Cross[nazcPcoco, cocoPnoam]];
         normal4b = unitVect3D[Cross[nazcPcoco, noamPnazc]];
         normal4c = unitVect3D[Cross[cocoPnoam, noamPnazc]];
         normal4 = unitVect3D[normal4a + normal4b + normal4c];
```

Next, we make great circles that are perpendicular to each of the four normal vectors computed above.

```
In[33]:= greatCirc1 = makeGreatCircle[normal1];
         greatCirc2 = makeGreatCircle[normal2];
         greatCirc3 = makeGreatCircle[normal3];
         greatCirc4 = makeGreatCircle[normal4];
```

Then we prepare the graphics output files.

```
In[34]:= 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"];

In[35]:= out2 = Graphics3D[Line[{{{1, 0, 0}, {-1, 0, 0}},
                {{0, 1, 0}, {0, -1, 0}}, {{0, 0, 1}, {0, 0, -1}}}]];

In[36]:= out3 = Graphics3D[Line[greatCirc1]];

In[37]:= out4 = Graphics3D[Line[greatCirc2]];

In[38]:= out5 = Graphics3D[Line[greatCirc3]];

In[39]:= out6 = Graphics3D[Line[greatCirc4]];

In[40]:= markers2 = {pcfcPnoam, noamPpcfc, pcfcPcoco,
             cocoPpcfc, pcfcPnazc, nazcPpcfc, noamPcoco, cocoPnoam,
             nazcPcoco, cocoPnazc, nazcPnoam, noamPnazc};

In[41]:= markers1 = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}};

In[42]:= out7 = ListPointPlot3D[markers1,
             AspectRatio → 1, BoxRatios → {1, 1, 1}, PlotStyle → Red,
             PlotRange → All, PlotRangePadding → 0.1];

In[43]:= out8 = ListPointPlot3D[markers2,
             AspectRatio → 1, BoxRatios → {1, 1, 1},
             PlotStyle → Directive[Green, PointSize[Large]],
             PlotRange → All, PlotRangePadding → 0.1];
```

In[44]:= `Show[out1, out2, out3, out4, out5, out6, out7, out8]`

Out[44]=

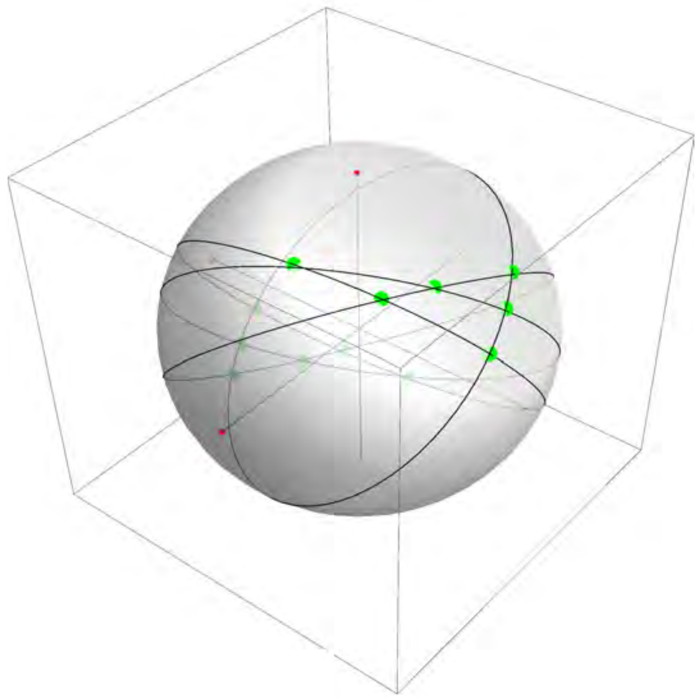

Figure 9-1. The graphic displayed above can be manipulated with the computer's mouse or trackpad. The X, Y and Z axes are the straight lines through the center of the sphere, and the positive ends of those axes are marked by red dots. The green dots mark the poles of rotation. The great circles are each coplanar with three rotational axes, corresponding to three plate pairs. Note that the great circles always intersect at a pole of rotation.

Exercise 9-3. The preceding section emphasized the inter-relatedness of angular velocity vectors in a multi-plate system.
Given a 4-plate system like the one we just considered, would changing the angular speed (the magnitude of the angular velocity vector) have an affect on the direction or magnitude of other angular velocity vectors? Explain your answer.

9.4 The dreaded geohedron

A couple of years after Dan McKenzie and Robert Parker published their paper on plate tectonics in angular-velocity space (1974), I was required to read that paper in an undergraduate course in "Geophysics and Plate Tectonics" taught by Donald McIntyre at Pomona College. McKenzie and Parker's paper introduced the idea of a geohedron, which is fundamentally a collection of vector circuits. McIntyre loved this kind of thing, and at the time I frankly had no idea what he was talking about. Later, when Minster and Jordan's model was published (1978) and included a stereographic image of the geohedron for their RM2 model, one would have thought that Professor McIntyre had died and gone to heaven. (Some of us less enthusiastic students nearly hurt ourselves rolling our eyes.) A few decades later, we will reproduce the Minster and Jordan geohedron in a way that can be manipulated in *Mathematica*.

The plate system studied by Minster and Jordan (1978) had 11 plates, and there were 18 spots on their map where three plates are together. These places are called triple junctions, and these important boundary regions are considered separately in a future chapter. While we could consider all of the possible plate triplets, whether they are in contact with each other or not, we will simplify our work by considering only the plate triplets that share a triple junction. We are using this older study for simplicity, because it involves fewer plates than the current models.

An Excel spreadsheet called RM2.xls is available to you at

http://bearspace.baylor.edu/Vince_Cronin/www/GradStruct/RM2.xls

That file has 54 records (rows) and 5 columns, presenting information derived from Table 2 of Minster and Jordan's 1978 paper. Let's import the Excel file, as explained in chapter 7

```
In[45]:= mydata = Import["/Users/vincecronin/Desktop/RM2.xls"];
```

flatten it by 1 dimension so that it behaves itself in a convenient manner

```
In[46]:= inputData = Flatten[mydata, 1];
```

and look at the first four records of the resulting file

```
In[47]:= Take[inputData, 4]

Out[47]= {{pcfc, noam, 0.155629, -0.539542, 0.640764},
          {noam, pcfc, -0.155629, 0.539542, -0.640764},
          {pcfc, coco, -0.530344, -1.63904, 1.38114},
          {coco, pcfc, 0.530344, 1.63904, -1.38114}}
```

We could have used the Take function to look at the last 2 elements of the file

```
In[48]:= Take[inputData, -2];
```

or elements 9 through 11 of the file

```
In[49]:= Take[inputData, {9, 11}];
```

so we can look at any part of the file we choose without having to look at the entire dataset. The first column indicates the plate that is the frame of reference, and the second column is the plate that is moving in a counter-clockwise direction around the pole of rotation. The third, fourth and fifth columns give the X, Y, and Z components of the instantaneous angular velocity vector for that plate pair. The length of that vector is the angular speed, in degrees per million years. The first record has the first line from Table 2 (Minster and Jordan, 1978), and the second record has the antipole corresponding to the first line of Table 2, and so on down the line.

A geohedron is a construction made in angular velocity space using angular velocity vectors to connect verticies that represent the plates. The line segments between any two points are parallel to the rotational axis between those two plates, and the length of the line segment is proportional to the angular speed.

Making the geohedron begins by choosing an origin for the plot, and it is often easiest to choose the plate that has the largest number of adjacent plates as the origin: the Pacific plate. The Pacific plate then has coordinates of {0, 0, 0}. The North American plate is located at the end of the angular velocity vector ${}_{\text{PCFC}}\omega_{\text{NOAM}}$, at coordinates {0.155629, -0.539542, 0.640764} according to the Excel spreadsheet. Hence, we can read the coordinates of Cocos, Nazca, Eurasia, India and Antarctica (along with North America) right off the Excel spreadsheet.

In[50]:=
```
pcfc = {0, 0, 0};
noam1 =
  {inputData[[1, 3]], inputData[[1, 4]], inputData[[1, 5]]};
coco1 = {inputData[[3, 3]], inputData[[3, 4]], inputData[[3, 5]]};
nazc1 = {inputData[[5, 3]], inputData[[5, 4]], inputData[[5, 5]]};
eura1 = {inputData[[7, 3]], inputData[[7, 4]], inputData[[7, 5]]};
indi1 = {inputData[[9, 3]], inputData[[9, 4]], inputData[[9, 5]]};
anta1 =
  {inputData[[11, 3]], inputData[[11, 4]], inputData[[11, 5]]};
```

We can now get a second estimate of where some of these plates are located, by vector addition. For example, the coordinates of North America in angular-velocity space can be found by the vector additions ${}_{\mathrm{PCFC}}\Omega_{\mathrm{COCO}} + {}_{\mathrm{COCO}}\Omega_{\mathrm{NOAM}}$ or ${}_{\mathrm{PCFC}}\Omega_{\mathrm{EURA}} + {}_{\mathrm{EURA}}\Omega_{\mathrm{NOAM}}$. If this was a perfectly consistent model in which all of the vector circuits closed perfectly, the two sums would yield the same coordinates as the original estimate of North America's position. For the moment, we will keep the results separate.

In[57]:=
```
Take[inputData, {13, 14}]
```

Out[57]=
```
{{noam, coco, -0.670887, -1.10428, 0.739994},
 {coco, noam, 0.670887, 1.10428, -0.739994}}
```

In[58]:=
```
noam2 = coco1 +
   {inputData[[14, 3]], inputData[[14, 4]], inputData[[14, 5]]};
```

In[59]:=
```
Take[inputData, {17, 18}]
```

Out[59]=
```
{{noam, eura, -0.0637759, 0.0697457, 0.210782},
 {eura, noam, 0.0637759, -0.0697457, -0.210782}}
```

In[60]:=
```
noam3 = eura1 +
   {inputData[[18, 3]], inputData[[18, 4]], inputData[[18, 5]]};
```

Let's print these three location estimates for the point representing North America in angular-velocity space as a list so that we can compare them.

In[61]:=
```
{noam1, noam2, noam3}
```

Out[61]=
```
{{0.155629, -0.539542, 0.640764},
 {0.140543, -0.534759, 0.641143},
 {0.155833, -0.539835, 0.640728}}
```

These three vectors are quite similar, so let's simplify our lives by taking their average as the position of the point that represents North America in angular-velocity space.

In[62]:=
```
noam = ((noam1 + noam2 + noam3) / 3);
```

Now we will repeat this process for other plates that touch the Pacific plate.

In[63]:=
```
coco2 = noam1 +
   {inputData[[13, 3]], inputData[[13, 4]], inputData[[13, 5]]};
coco3 = nazc1 + {inputData[[25, 3]],
   inputData[[25, 4]], inputData[[25, 5]]};
coco = ((coco1 + coco2 + coco3) / 3); nazc2 = coco1 +
  {inputData[[26, 3]], inputData[[26, 4]], inputData[[26, 5]]};
nazc = ((nazc1 + nazc2) / 2); anta2 = indi1 +
  {inputData[[54, 3]], inputData[[54, 4]], inputData[[54, 5]]};
anta = ((anta1 + anta2) / 2); indi2 = anta1 +
  {inputData[[53, 3]], inputData[[53, 4]], inputData[[53, 5]]};
indi3 = eura1 + {inputData[[43, 3]],
   inputData[[43, 4]], inputData[[43, 5]]};
indi = ((indi1 + indi2 + indi3) / 3); eura2 = indi1 +
  {inputData[[44, 3]], inputData[[44, 4]], inputData[[44, 5]]};
eura3 = noam1 + {inputData[[17, 3]], inputData[[17, 4]],
   inputData[[17, 5]]}; eura = ((eura1 + eura2 + eura3) / 3);
```

Plates in the RM2 model that do not touch the Pacific plate (African, Arabian, Caribbean, South American) are found through vector addition. For example, the Arabian plate can be found either by the sum ${}_{PCFC}\Omega_{INDI}$ + ${}_{INDI}\Omega_{ARAB}$ or by the sum ${}_{PCFC}\Omega_{EURA}$ + ${}_{EURA}\Omega_{ARAB}$.

In[65]:=
```
arab1 = eura1 +
  {inputData[[45, 3]], inputData[[45, 4]], inputData[[45, 5]]};
arab2 = indi1 + {inputData[[48, 3]],
   inputData[[48, 4]], inputData[[48, 5]]};
arab = ((arab1 + arab2) / 2); carb1 = noam1 +
  {inputData[[20, 3]], inputData[[20, 4]], inputData[[20, 5]]};
carb2 = nazc1 + {inputData[[24, 3]],
   inputData[[24, 4]], inputData[[24, 5]]};
carb = ((carb1 + carb2) / 2); soam1 = nazc1 +
  {inputData[[32, 3]], inputData[[32, 4]], inputData[[32, 5]]};
soam2 = noam1 + {inputData[[28, 3]], inputData[[28, 4]],
   inputData[[28, 5]]}; soam3 = anta1 +
  {inputData[[36, 3]], inputData[[36, 4]], inputData[[36, 5]]};
soam = ((soam1 + soam2 + soam3) / 3); afrc1 = noam1 +
  {inputData[[15, 3]], inputData[[15, 4]], inputData[[15, 5]]};
afrc2 = eura1 + {inputData[[41, 3]],
   inputData[[41, 4]], inputData[[41, 5]]};
afrc3 = indi1 + {inputData[[38, 3]], inputData[[38, 4]],
   inputData[[38, 5]]}; afrc4 = anta1 +
  {inputData[[51, 3]], inputData[[51, 4]], inputData[[51, 5]]};
afrc = ((afrc1 + afrc2 + afrc3 + afrc4) / 4);
```

If RM2 was a perfectly consistent model, different routes to finding the coordinates of a given plate would yield the same results. Neither RM2 or any other model is perfectly consistent because of the uncertainties in the input data. To produce a coherent geohedron, we average multiple estimates of a given plate's location in angular-velocity space to yield a final position.

Now let's see what this all looks like in 3D.

In[66]:=
```
geohedCircuits =
  {arab, indi, eura, arab, afrc, anta, indi, pcfc, eura, afrc,
   noam, soam, carb, noam, coco, nazc, pcfc, noam, eura, pcfc,
   anta, pcfc, coco, nazc, carb, soam, nazc, soam, afrc};
```

In[67]:=
```
out1 = Graphics3D[
   Line[{{{0.3, 0, 0}, {0, 0, 0}}, {{0, 0.3, 0}, {0, 0, 0}},
     {{0, 0, 0.3}, {0, 0, 0}}}], Boxed → False];
```

In[68]:=
```
out2 = Graphics3D[Line[geohedCircuits], Boxed → False];
```

In[69]:=
```
markers1 = {{0.3, 0, 0}, {0, 0.3, 0}, {0, 0, 0.3}};
```

In[70]:=
```
out3 = ListPointPlot3D[markers1, AspectRatio → 1,
   PlotStyle → Red, PlotRange → All, Boxed → False];
```

In[71]:=
```
out4 = ListPointPlot3D[geohedCircuits, AspectRatio → 1,
   PlotStyle → Directive[Green, PointSize[Large]],
   PlotRange → All, Boxed → False];
```

In[72]:=
```
Show[out1, out2, out3, out4]
```

Out[72]=

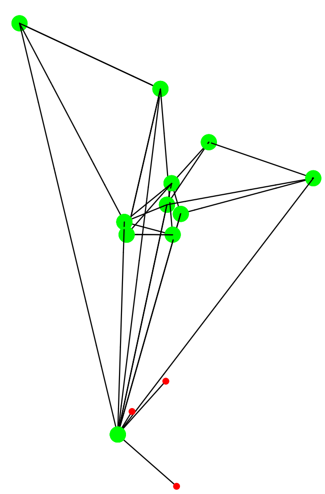

Figure 9-2. The graphic displayed above can be manipulated with the computer's mouse or trackpad. The coordinate axes have lengths of 0.3°/Myr.

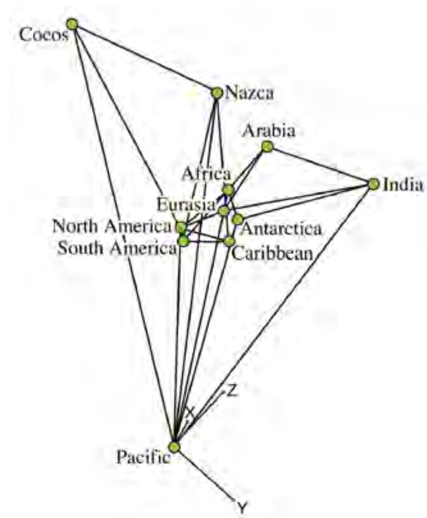

Figure 9-3. Static image of the geohedron, with the vertices (green dots) and coordinate axes in angular-velocity space labeled. The coordinate axes have lengths of 0.3°/Myr.

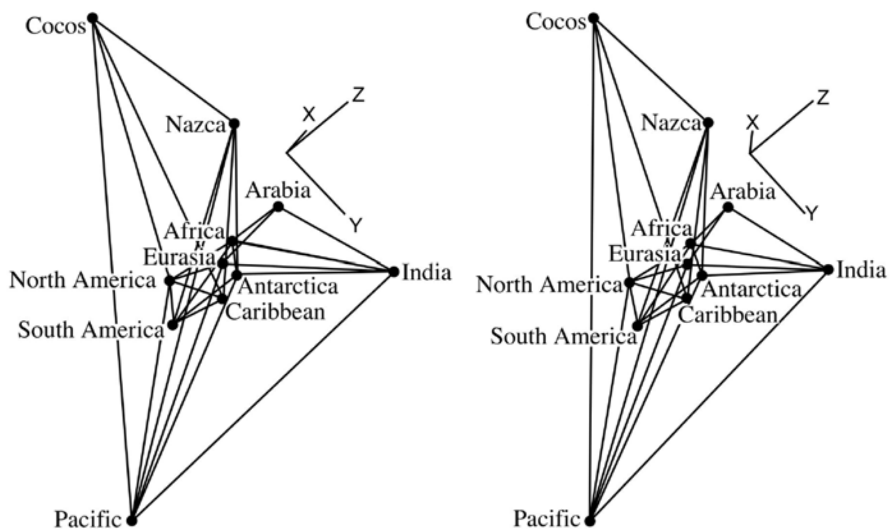

Figure 9-4. Stereo pair of the RM2 geohedron, adapted from Figure 2 of Minster and Jordan (1978). The origin of their geohedron was not the same as in Figure 9-3 above, but that does not affect the geometry of the geohedron. The coordinate axes have lengths of 0.3°/Myr.

Exercise 9-4. The vertices of the geohedron represent an individual plate in angular-velocity space. If an individual plate's velocities relative to all other plates changed, the position of the vertex that corresponds to that plate would change.

Given the 11-plate system like the one we just considered, would changing the position of an individual vertex in the geohedron necessarily cause a change in the position of other vertices? Explain your answer.

9.4 Models of instantaneous relative plate motion

The development of quantitative models of instantaneous relative plate motion since the late 1960s follows the growth of kinematic information about motion across plate boundaries. As more data were collected, more plates were recognized and could be added to the models. Minster and Jordan (1978) considered 11 plates. Chuck DeMets and his coworkers included 16 plates in their Nuvel-1 and Nuvel-1a models. More recently, the Morvel model of DeMets and others tracks 25 plates, and Don Argus and others have subsequently added 21 plates to Morvel (after Bird, 2003) for a total of 56 plates. Whether we work with 11 plates or five times as many, the fundamental vector relationships we have outlined in this chapter remain the same.

It is reasonable to ask "which model should I use?" My sense is that each successive model has been compiled with knowledge of all major previous models, and that a typical component of most of these papers is a detailed discussion comparing their new model with the prior models. This gives me cause for confidence that, in general, the newer models are more useful and better constrained than the older models. As I write this text in February of 2012, I use the kinematic data from MORVEL (DeMets and others, 2010) for most purposes involving the instantaneous motion of approximately rigid plates, with the additional data from Argus and others (2010) available as necessary. You will notice the names DeMets, Gordon, Argus and Gripp among the authors of many of these models. Chuck DeMets, Richard Gordon, Don Argus and Alice Gripp worked together at Northwestern University in the 1980s and early '90s, and have since dispersed to other universities. In my opinion, their work is carefully done and has proven to be consistently reliable.

9.5 References and relevant texts

Argus, D.F., Gordon, R.G., Heflin, M.B., Ma, C., Eanes, R.J, Willis, P., Peltier, W.R., and Owen, S.E., 2010, The angular velocities of the plates and the velocity of Earth's centre from space geodesy: Geophysical Journal International, v. 180, p. 913-960, doi: 10.1111/j.1365-246X.2009.04463.x.

Argus, D.F., Gordon, R.G., and DeMets, C., 2010, Geologically current motion of 56 plates relative to the no-net-rotation reference frame: Geochemistry Geophysics Geosystems, v. 12, Q 1101, 13 p., doi: 10.1029/2011GC003751.

Bird, P., 2003, An updated digital model of plate boundaries: Geochemistry Geophysics Geosystems, v. 4, no. 3, p. 1027, doi:10.1029/2001GC000252. Available online via http://peterbird.name/publications/2003_PB2002/2003_PB2002.htm

DeMets, C., Gordon, R.G., Argus, D.F., and Stein, S., 1990, Current plate motions: Geophysical Journal International, v. 101, p. 425-478.

DeMets, C., Gordon, R.G., Argus, D.F., and Stein, S., 1994, Effect of recent revisions to the geomagnetic refersal time scale on estimates of current plate motions: Geophysical Research Letters, v. 21, p. 2191-2194.

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

Kreemer, C., Holt, W.E., and Haines, A.J., 2003, An integrated global model of present-day plate motions and plate boundary deformation: Geophysical Journal International, v. 154, p. 8-34.

Minster, J.B., Jordan, T.H., Molnar, P., and Haines, E., 1974, Numerical modelling of instantaneous plate tectonics: Geophysical Journal, Royal Astronomical Society, v. 36, p. 541-576.

Minster, J.B., and Jordan, T.H., 1978, Present-day plate motions: Journal of Geophysical Research, v. 83, no. B11, p. 5331-5354.

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