SolveVelocityTriProblem.nb

 Written August 1, 2018, by Vince Cronin; revised 4 December 2018

A code to compute the total instantaneous velocity of a point along a mid-ocean ridge in a reference frame whose "north pole" is the pole of instantaneous relative motion between two plates

Introduction

The general background of this analysis is explained in Cronin (1994).

This version contains data that is useful to investigate the North American-Nubian plate system.

Input

Angular velocities of the 2 - plate system

The following data describe the instantaneous motion of individual plates in an NNR reference frame (e.g., Argus, D.F., Gordon, R.G., and DeMets, C., 2011). A code to compute the meridional and zonal components of the instantaneous velocity of a point along a mid-o

NA refers to the North American plate, and NB refers to the Nubian plate.

```
In[469]:= latNA = -4.85; longNA = -80.64; angvelNA = 0.209;
```

```
In[470]:= latNB = 47.68; longNB = -68.44; angvelNB = 0.292;
```

The following data describe the instantaneous motion of one plate as observed from another plate (e.g., DeMets, C., Gordon, R.G., and Argus, D.F., 2010).

Angular velocities describe counter - clockwise (positive) rotation of the first listed plate relative to the plate listed second

```
In[471]:= latNBNA = 79.2; longNBNA = 40.2; angvelNBNA = 0.233;
```

Locations of the endpoints of the transform fault

Kane transform fault

```
In[472]:= faultName = "Kane";
```

In[473]:=
```
latWestEnd = 23.842; longWestEnd = -46.345;
```

In[474]:=
```
latEastEnd = 23.649; longEastEnd = -44.855;
```

Atlantis transform fault

faultName = "Atlantis";

latWestEnd = 30.135; longWestEnd = -42.693;

latEastEnd = 29.985; longEastEnd = -42.018;

Oceanographer transform fault

faultName = "Oceanographer";

latWestEnd = 35.287; longWestEnd = -36.327;

latEastEnd = 35.026; longEastEnd = -34.994;

User - defined functions

In[475]:=
```
convert2cart[lat_, long_] := {Cos[lat Degree] Cos[long Degree],
   Cos[lat Degree] Sin[long Degree], Sin[lat Degree]};
```

In[476]:=
```
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[477]:=
```
unitVect3D[vect_] :=
  {(vect[[1]] / Norm[vect]), (vect[[2]] / Norm[vect]), (vect[[3]] / Norm[vect])};
```

In[478]:=
```
tangentialVelocityComputer[latPole_, longPole_, latPoint_, longPoint_, angVel_] :=
  Module[{a, b, c, d, e, answer}, a = {Cos[latPole Degree] Cos[longPole Degree],
     Cos[latPole Degree] Sin[longPole Degree], Sin[latPole Degree]};
   b = {Cos[latPoint Degree] Cos[longPoint Degree],
     Cos[latPoint Degree] Sin[longPoint Degree], Sin[latPoint Degree]};
   c = VectorAngle[a, b] / Degree;
   d = (angVel * 6371.01 * 2 * π * Sin[c Degree]) / 360;
   e = unitVect3D[Cross[a, b]] * d;
   answer = e;
   answer];
```

In[479]:=
```
findTangentialVector[locVectPole_, locVectPoint_, angVel_] :=
  Module[{a, b, c, answer}, a = VectorAngle[locVectPole, locVectPoint] / Degree;
   b = (angVel * 6371.01 * 2 * π * Sin[a Degree]) / 360;
   c = unitVect3D[Cross[locVectPole, locVectPoint]] * b;
   answer = c;
   answer];
```

We now know the length of each side of a plane triangle (two plane triangles, in fact -- one for each end of the transform fault) and need to know the interior angles. Given a triangle whose sides have lengths a, b, and c (where the angle between sides a and b is C, etc.), the law of cosines holds that

$$c^2 = a^2 + b^2 - 2ab\,\text{Cos}[C]$$

Re-arranging to isolate the unknown quantity, the angle C,

$$C = \text{ArcCos}\left[\frac{a^2+b^2-c^2}{2ab}\right]$$

In[480]:=
```
interiorAngles[a_, b_, c_] :=
  Module[{α, β, χ, result}, α = ArcCos[(b^2 + c^2 - a^2) / (2 * b * c)] / Degree;
   β = ArcCos[(a^2 + c^2 - b^2) / (2 * a * c)] / Degree;
   χ = ArcCos[(a^2 + b^2 - c^2) / (2 * a * b)] / Degree;
   result = {a, b, c, α, β, χ};
   result];
```

Computation

Convert the input data from geographic coordinates (lat, long) to cartesian coordinates

In[481]:=
```
poleNA = convert2cart[latNA, longNA];
```

In[482]:=
```
poleNB = convert2cart[latNB, longNB];
```

In[483]:=
```
poleNBNA = convert2cart[latNBNA, longNBNA];
```

```
In[484]:= pointWestEnd = convert2cart[latWestEnd, longWestEnd];
          pointEastEnd = convert2cart[latEastEnd, longEastEnd];
```

Combine unit location vector coordinates with the corresponding angular speed

```
In[485]:= angVelVectorNA = {poleNA[[1]], poleNA[[2]], poleNA[[3]], angvelNA};

In[486]:= angVelVectorNB = {poleNB[[1]], poleNB[[2]], poleNB[[3]], angvelNB};

In[487]:= angVelVectorNBNA = {poleNBNA[[1]], poleNBNA[[2]], poleNBNA[[3]], angvelNBNA};
```

Determine the tangential velocity vectors at both ends of the

```
In[488]:= tangVelVectWNA = findTangentialVector[poleNA, pointWestEnd, angVelVectorNA[[4]]];

In[489]:= tangSpeedWNA = Norm[%];

In[490]:= tangVelVectENA = findTangentialVector[poleNA, pointEastEnd, angVelVectorNA[[4]]];

In[491]:= tangSpeedENA = Norm[%];

In[492]:= tangVelVectWNB = findTangentialVector[poleNB, pointWestEnd, angVelVectorNB[[4]]];

In[493]:= tangSpeedWNB = Norm[%];

In[494]:= tangVelVectENB = findTangentialVector[poleNB, pointEastEnd, angVelVectorNB[[4]]];

In[495]:= tangSpeedENB = Norm[%];

In[496]:= tangRelVelVectWNBNA =
            findTangentialVector[poleNBNA, pointWestEnd, angVelVectorNBNA[[4]]];

In[497]:= tangSpeedWNBNA = Norm[%];

In[498]:= tangRelVelVectENBNA =
            findTangentialVector[poleNBNA, pointEastEnd, angVelVectorNBNA[[4]]];

In[499]:= tangSpeedENBNA = Norm[%];
```

Initial check for closure of vector triangle for the west end of the transform fault

```
In[500]:= firstStepW = interiorAngles[tangSpeedWNA, tangSpeedWNB, tangSpeedWNBNA];

In[501]:= test = firstStepW[[4]] + firstStepW[[5]] + firstStepW[[6]];

In[502]:= check1 = If[(Abs[test - 180] < (1 × 10^-10)), "OK", "We have a problem here"];
```

Instantaneous velocities of west end of the transform

The meridional velocity (mm/yr) is the velocity toward or away from the specified pole of instantaneous relative plate motion.

```
In[503]:= meridionalVelW1 = Abs[tangSpeedWNA Sin[firstStepW[[5]] Degree]];
```

```
In[504]:= meridionalVelW2 = Abs[tangSpeedWNB Sin[firstStepW[[4]] Degree]];

In[505]:= check2 = If[(Abs[Abs[meridionalVelW1] - Abs[meridionalVelW2]] < (1 × 10^-10)),
              "OK", "We have a problem here"];
```

The zonal velocity (mm/yr) is the velocity around the specified pole of instantaneous relative plate motion. A positive zonal velocity involves a counter - clockwise (positive) rotation around that pole.

```
In[506]:= zonalVelW1 = (tangSpeedWNA Cos[firstStepW[[5]] Degree]) - tangSpeedWNBNA/2;

In[507]:= zonalVelW2 = (tangSpeedWNB Cos[firstStepW[[4]] Degree]) - tangSpeedWNBNA/2;

In[508]:= check3 = If[(Abs[Abs[zonalVelW1] - Abs[zonalVelW2]] < (1 × 10^-10)),
              "OK", "We have a problem here"];

In[509]:= totalSpeedWestEnd = Sqrt[meridionalVelW1^2 + zonalVelW1^2]

Out[509]= 10.9179
```

Tangential velocity triangle: a, b, c, α, β, χ

```
In[510]:= secondStepW =
            interiorAngles[totalSpeedWestEnd, Abs[meridionalVelW1], Abs[zonalVelW1]];
```

Initial check for closure of vector triangle for the east end of the transform fault

```
In[511]:= firstStepE = interiorAngles[tangSpeedENA, tangSpeedENB, tangSpeedENBNA];

In[512]:= test = firstStepE[[4]] + firstStepE[[5]] + firstStepE[[6]]

Out[512]= 180.
```

Instantaneous velocities of east end of the transform

The meridional velocity (mm/yr) is the velocity toward or away from the specified pole of instantaneous relative plate motion.

```
In[513]:= meridionalVelE1 = tangSpeedENA Sin[firstStepE[[5]] Degree];

In[514]:= meridionalVelE2 = tangSpeedENB Sin[firstStepE[[4]] Degree];

In[515]:= check4 = If[(Abs[Abs[meridionalVelE1] - Abs[meridionalVelE2]] < (1 × 10^-10)),
              "OK", "We have a problem here"];
```

The zonal velocity (mm/yr) is the velocity around the specified pole of instantaneous relative plate motion.

```
In[516]:= zonalVelE1 = (tangSpeedENA Cos[firstStepE[[5]] Degree]) - tangSpeedENBNA/2;

In[517]:= zonalVelE2 = (tangSpeedENB Cos[firstStepE[[4]] Degree]) - tangSpeedENBNA/2;
```

In[518]:=
```
check5 = If[(Abs[Abs[zonalVelE1] - Abs[zonalVelE2]] < (1 × 10^-10)),
    "OK", "We have a problem here"];
```

In[519]:=
```
totalSpeedEastEnd = Sqrt[meridionalVelE1^2 + zonalVelE1^2]
```

Out[519]= 11.45

Tangential velocity triangle: a, b, c, α, β, χ

In[520]:=
```
secondStepE =
  interiorAngles[totalSpeedEastEnd, Abs[meridionalVelE1], Abs[zonalVelE1]];
```

In[521]:=
```
consolidatedChecks = {check1, check2, check3, check4, check5};
```

Output

Did the code generate expected results at five places in the process?

In[522]:=
```
consolidatedChecks
```

Out[522]= {OK, OK, OK, OK, OK}

In[523]:=
```
faultName
```

Out[523]= Kane

The meridional velocity (mm/yr) is the velocity toward or away from the specified pole of instantaneous relative plate motion.

The meridional velocity of the west end of the fault is ...

In[524]:=
```
Abs[meridionalVelW1]
```

Out[524]= 10.9177

The meridional velocity of the east end of the fault is ...

In[525]:=
```
Abs[meridionalVelE1]
```

Out[525]= 11.45

The zonal velocity (mm/yr) is the velocity around the specified pole of instantaneous relative plate motion.

The zonal velocity of the west end of the fault is ...

In[526]:=
```
Abs[zonalVelW1]
```

Out[526]= 0.0803753

The zonal velocity of the east end of the fault is ...

In[527]:= `Abs[zonalVelE1]`

Out[527]= `0.0157565`

The total speed of the west boundary point in the NNR reference frame is ...

In[528]:= `totalSpeedWestEnd`

Out[528]= `10.9179`

Tangential velocity triangle: a, b, c, α, β, χ

In[529]:= `secondStepW`

Out[529]= `{10.9179, 10.9177, 0.0803753, 90., 89.5782, 0.421802}`

The total speed of the east boundary point in the NNR reference frame is ...

In[530]:= `totalSpeedEastEnd`

Out[530]= `11.45`

Tangential velocity triangle: a, b, c, α, β, χ

In[531]:= `secondStepE`

Out[531]= `{11.45, 11.45, 0.0157565, 90., 89.9212, 0.0788453}`

Other results

In[532]:= `tangVelVectWNA`

Out[532]= `{-10.5359, -2.76295, 11.9342}`

In[533]:= `tangSpeedWNA`

Out[533]= `16.1575`

In[534]:= `tangVelVectENA = findTangentialVector[poleNA, pointEastEnd, angVelVectorNA[[4]]]`

Out[534]= `{-10.4346, -2.78662, 12.4035}`

In[535]:= `tangSpeedENA`

Out[535]= `16.4467`

In[536]:= `tangVelVectWNB = findTangentialVector[poleNB, pointWestEnd, angVelVectorNB[[4]]]`

Out[536]= `{7.66927, 11.9113, 7.52096}`

In[537]:= `tangSpeedWNB`

Out[537]= `16.0394`

In[538]:= `tangVelVectENB = findTangentialVector[poleNB, pointEastEnd, angVelVectorNB[[4]]]`

Out[538]= `{7.3554, 12.3671, 8.01201}`

```
In[539]:= tangSpeedENB
Out[539]= 16.4694

In[540]:= tangRelVelVectWNBNA =
           findTangentialVector[poleNBNA, pointWestEnd, angVelVectorNBNA[[4]]]
Out[540]= {18.1083, 14.5701, -4.4324}

In[541]:= tangSpeedWNBNA
Out[541]= 23.6611

In[542]:= tangRelVelVectENBNA =
           findTangentialVector[poleNBNA, pointEastEnd, angVelVectorNBNA[[4]]]
Out[542]= {17.6995, 15.0385, -4.43051}

In[543]:= tangSpeedENBNA
Out[543]= 23.6444
```

Tangential velocity triangle: a, b, c, α, β, χ

```
In[544]:= firstStepW
Out[544]= {16.1575, 16.0394, 23.6611, 42.8967, 42.5087, 94.5947}

In[545]:= firstStepE
Out[545]= {16.4467, 16.4694, 23.6444, 44.0456, 44.1219, 91.8325}
```

References

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(11), Q1101, doi:10.1029/2011GC003751.

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. 181, p. 1-80, doi:10.1111/j.1365-246X.2010.04491.x.