

Inertia Tensor for MORVEL Tectonic Plates (Postprint)

Authors: Li Chunxiao

Date: 2017-10-20T00:00:00+00:00

Abstract

The NNR (No-Net-Rotation)-MORVEL56 (Mid-Ocean Ridge VELOCITY) is a set of angular velocities describing the motions of 56 plates relative to a No-Net-Rotation reference frame. These plates can be adjusted in terms of non-overlapping polygonal regions, separated by plate boundaries on a unit sphere. During the calculation on the kinematic parameters for these 56 plates in a NNR reference frame using the International Terrestrial Reference Frame (ITRF) velocity field, the geometric parameters of tectonic plates play a significant role in establishing an absolute plate motion model based on space geodesy results. The computational method for these geometric parameters implemented as a FORTRAN90 program is described in this paper, allowing an evaluation of the area and the inertia tensor of a polygonal region on a unit sphere. This program is mainly built on a triangulation algorithm and the adaptive Simpson's double integral method for spherical polygons, which produces highly reliable results for all 56 modern plates.

Full Text

Inertia Tensor for MORVEL Tectonic Plates

Li Chunxiao

1. Yunnan Observatories, Chinese Academy of Sciences, Kunming 650011, China
2. University of Chinese Academy of Sciences, Beijing 100049, China Email: lcx@ynao.ac.cn

Abstract

The NNR-MORVEL56 model comprises a set of angular velocities describing the motions of 56 tectonic plates relative to a No-Net-Rotation (NNR) reference frame. These plates can be represented as non-overlapping polygonal re-

gions on a unit sphere, separated by plate boundaries. When calculating the kinematic parameters for these 56 plates in an NNR reference frame using the International Terrestrial Reference Frame (ITRF) velocity field, the geometric parameters of tectonic plates play a significant role in establishing an absolute plate motion model based on space geodesy results. This paper describes a computational method for these geometric parameters, implemented as a FORTRAN90 program, that enables evaluation of the area and inertia tensor of a polygonal region on a unit sphere. The program is built primarily on a triangulation algorithm and the adaptive Simpson's double integral method for spherical polygons, which produces highly reliable results for all 56 modern plates.

Keywords: Tectonic plate; Spherical polygon; Inertia tensor; NNR-MORVEL56

Introduction

Most of Earth's major geological features can be understood through the interactions between tectonic plates, which move independently—separating from, colliding with, and sliding past one another. Until the mid-1960s, a unified theory was developed to explain Earth's dynamics. Several decades after the inception of plate tectonics theory, plate dynamic models constructed using geological and geophysical data remained dominant until long-term geodetic observations were gathered to estimate contemporary plate kinematic parameters [?]. As one of the most representative geological plate motion models, NUVEL-1A has been a mainstream model regarding plate dynamics and kinematics. With the increased quantity and quality of geological and geodetic data in recent years, MORVEL refined the precision and accuracy of geometric and kinematic parameters for 56 plates, partly derived from an updated digital model of plate boundaries by Bird [?]. Relative to NUVEL-1A, MORVEL incorporates more than twice as many plates and covers more of Earth's surface, and nearly all NUVEL-1A angular velocities differ significantly from their MORVEL counterparts. To derive an absolute motion model in an NNR reference frame, however, inertia tensors are always considered indispensable attributes of these plates. Despite various established methods for calculating plate inertia tensors corresponding to the NUVEL-1A model presented in many papers [?], it is necessary to recalculate a new set for the NNR-MORVEL56 model, given the considerable discrepancy between NUVEL-1A and MORVEL.

When a polygon on the unit sphere is employed for the representation of a tectonic plate, a simplified analysis of the plate inertia tensor can be performed through a numerical method carried out over all 56 plates. This paper illustrates the method for calculating all nine components of the inertia tensor, which requires precise knowledge of the plate boundaries. The boundary file contains a two-column sequence of latitude-longitude plate boundary coordinates that fully enclose the plate in the counterclockwise direction. The first section introduces concepts regarding the no-net-rotation conditions and indicates the calculation of the Euler vector in an absolute motion model. The following section describes

the detailed mathematical models to estimate the area and inertia tensor of spherical polygons. The final section is dedicated to presenting the results for 56 modern plates, and the appendix provides the original FORTRAN90 program for obtaining the aforementioned results.

2. Net Lithosphere Rotation

A no-net-rotation model for the lithosphere assumes that the integral of $\mathbf{v} \times \mathbf{r}$ over the Earth's surface equals zero, i.e., $\int_{\text{Earth}} \mathbf{v} \times \mathbf{r} dS = 0$, where \mathbf{r} is the radial vector of the surface element on a unit sphere and \mathbf{v} corresponds to the horizontal velocity at that position. The angular velocity of net rotation ω_{net} was computed as the total angular momentum of all plates divided by the moment of inertia of the entire lithosphere, using the equation [?]:

$$\omega_{\text{net}} = \frac{\int_{\text{Earth}} \mathbf{r} \times \mathbf{v} dS}{\int_{\text{Earth}} r^2 dm}$$

Then it is convenient to convert the equation into the following form:

$$\int_{\text{Earth}} \mathbf{v} \times \mathbf{r} dS = 8\pi \int_{\text{Earth}} \omega_i Q_i$$

where ω_i is the Euler vector describing the motion of plate i , Q_i is the inertia tensor of plate i , and i goes from 1 to N relative to an inertial reference frame such as ITRF2008. The angular velocities of the plates relative to the NNR reference frame were then found by vector subtraction, namely $\omega_{\text{NNR}i} = \omega_i - \omega_{\text{net}}$.

For those tectonic plates where angular velocities are not available in the geodetic model such as ITRF2000-PMM due to insufficient data, Altamimi tested four cases to perfect the incomplete geodetic model. The fourth case described a method for estimating the missing angular velocity. Here we employ it in this paper using a simple equation written as:

$$\omega_{\text{ITRF}} = \omega_{\text{MORVEL}} + \omega_{\text{ITRF}}^{\text{missing}}$$

where $\omega_{\text{ITRF}}^{\text{missing}}$ is the undetermined rotation vector of the plate in the ITRF model, and ω_{MORVEL} is the MORVEL rotation vector for the plate adjacent to the missing plate.

3. Area and Inertia Tensor of Plates

3.1 Evaluation of Plate Area

Spherical polygons are defined by great circle arcs connecting points on the sphere, whose positions are given by latitudes and longitudes. An algorithm for determining the area of a spherical polygon of

arbitrary shape has been presented by Bevis and Cambaresi [?], where the kernel idea is to compute the interior angle at each vertex of the spherical polygon. In this paper, however, we employ a somewhat similar method to that of Miller [?], determining the area of a spherical polygon by summing the signed areas of component triangles. For a spherical polygon of n sides, the spherical excess is generalized as $E = \sum_{i=1}^n \theta_i - (n - 2) \times 180^\circ$, where θ_i are the interior angles of the polygon.

Considering a spherical polygon $ABCD$ as shown in Figure 1: see original paper, the north pole combined with any two adjacent vertices of the polygon can constitute a spherical triangle, such as NAB . The two sides of the triangle are known from the latitudes of their vertices, i.e., $a = 90^\circ - \text{latitude}(A)$ and $b = 90^\circ - \text{latitude}(B)$. Taking the previously obtained two sides and the included angle specified by the difference between longitudes of vertices A and B , the opposite side c can be calculated via the haversine formula:

$$\text{hav}(c) = \text{hav}(b - a) + \sin(a) \sin(b) \text{hav}(\Delta\lambda)$$

where the haversine function is defined as $\text{hav}(\theta) = (1 - \cos\theta)/2$ with θ in radians. Having obtained all three sides of the spherical triangle, we can use L' Huillier' s formula to obtain its excess:

$$\tan\left(\frac{E}{4}\right) = \sqrt{\tan\left(\frac{s}{2}\right) \tan\left(\frac{s-a}{2}\right) \tan\left(\frac{s-b}{2}\right) \tan\left(\frac{s-c}{2}\right)}$$

where $s = (a + b + c)/2$.

To find the area of a spherical polygon, one may use successive vertices in pairs to form a spherical triangle. Each spherical triangle employs the north pole as a common vertex to make calculations convenient. When calculating the areas of individual triangles, we adopt a convention that the sign of the triangle area (which has the same value as the spherical excess for a polygon on a unit sphere) is identical to the sign of the difference between the longitudes of a pair of adjacent vertices. If the longitude of the first vertex is less than the second one, then the sign of this triangle area is defined as positive for a set of points arranged in the anticlockwise way and vice versa. Therefore, the area of the spherical polygon is regarded as the absolute value of the sum of the signed spherical excesses for each of the spherical triangles.

Taking the facility of calculation for the upcoming inertia tensor into account, a provision is crucial that the vertex points traversing the polygon must be enumerated in the counterclockwise direction.

[Figure 1: see original paper] illustrates the spherical polygon and triangles used for calculating area and inertia tensor: (a) N triangles constructed from a counterclockwise spherical polygon of n sides; (b) Spherical triangle encompassing the South Pole and polygon sides traversing the 180th meridian.

One should note two special cases when estimating geometric parameters of spherical polygons. The first case occurs when polygon sides cross the International Date Line: as illustrated in Figure 1: see original paper, 2π must be added to the longitude to ensure continuity of calculation. A similar situation applies when a vertex leaves the 180th meridian—for vertex 2, 2π must be subtracted from its longitude. The second case occurs when a spherical polygon has an area larger than a hemisphere. Under the implicit assumption that any polygon area is less than 2π , for extensive polygons such as triangles surrounding the north pole N , the exact area is calculated as the complement: $\text{Area}_{\text{exact}} = 4\pi - \text{Area}_{\text{complement}}$, where the complement includes the north pole (e.g., $\text{Area}_{ABC} = 4\pi - \text{Area}_{\text{complement}}$). The original FORTRAN90 program in the appendix has taken into account these two singular cases.

3.2 Estimation of Plate Inertia Tensor The components of the symmetric inertia tensor can be calculated for a region using the following formula:

$$I_{\mu\nu} = \int_P (\delta_{\mu\nu} - x_\mu x_\nu) dS$$

where $\mu, \nu = 1, 2, 3$ are the Cartesian coordinates, $\delta_{\mu\nu}$ are the elements of the identity matrix, and the integration is carried out over the surface of a plate. These inertia tensors are based on the hypothesis that the surface density of the plate is unity and entirely describe the plate geometry. For instance, the plate area is easily calculated by taking the trace: $\text{Area} = \frac{1}{2} \sum_\mu I_{\mu\mu}$. This implies invariance of the trace under coordinate rotations, and the sum of diagonal components is always double the area of the polygon. Generally speaking, non-diagonal components indicate the asymmetry of the polygon with respect to the Cartesian axes, and all diagonal components have positive values, which is useful for verification tests together with the trace property.

In this paper we propose a somewhat different method from Schettino [?] for constructing the spherical triangle. The integral on the right-hand side of the inertia tensor equation is easily calculated for spherical triangles. The components of the total tensor are therefore given by:

$$I_{\mu\nu} = \sum_{i=1}^N \int_{T_i} (\delta_{\mu\nu} - x_\mu x_\nu) dS$$

where $\sum_{i=1}^N T_i$ is the total polygon area, which has been illustrated in the previous section.

Let a point on the sphere be given in spherical coordinates (ϕ, λ) , where ϕ is latitude and λ is longitude, so its Cartesian coordinates are given by:

$$x = \cos \phi \cos \lambda, \quad y = \cos \phi \sin \lambda, \quad z = \sin \phi$$

Then the area element dS at this position is equal to $\cos \phi d\phi d\lambda$. The components of the inertia tensor for a triangle are therefore written in spherical coordinates as:

$$I_{\mu\nu} = \int_{\lambda_1}^{\lambda_2} \int_{\phi_{\text{lower}}(\lambda)}^{\pi/2} f_{\mu\nu}(\phi, \lambda) d\phi d\lambda$$

where λ_1, λ_2 are the longitudes of vertices and the function $f_{\mu\nu}$ is given by the appropriate Cartesian product.

Next, as a result of the symmetry of the inertia tensor, the six independent components are expressed in the following way. The upper limit of the inner integral about latitude is always set to $\pi/2$ because the North Pole is considered the common vertex of each spherical triangle. In contrast, with this simple upper limit, the lower limit function $\phi_{\text{lower}}(\lambda)$ can be obtained from a series of derivations, whose concrete form is formulated as $\arctan\left(\frac{C_1 \cos \lambda + C_2 \sin \lambda + C_3}{\text{something}}\right)$, where C_1, C_2, C_3 represent three constants.

Once the integrated triangles are determined by one side of the polygon, such as AB , we can write their expression in the following form:

$$I_{\mu\nu} = \int_{\lambda_A}^{\lambda_B} \int_{\phi_{\text{lower}}(\lambda)}^{\pi/2} f_{\mu\nu}(\phi, \lambda) \cos \phi d\phi d\lambda$$

where (ϕ_A, λ_A) and (ϕ_B, λ_B) are the latitude and longitude of vertices A and B , respectively.

Unlike the area estimation process, evaluation of the inertia tensor is associated with the integral order. Hence, a counterclockwise direction must be adopted in the procedure. All special situations have been considered in the Fortran program, including the 180th meridian case and the case of encompassing the South Pole. In addition to the two special cases in area estimation, the principal moments in tensor calculation need to be deduced from the whole inertia tensor of the spherical surface when involving polygons containing the south pole, i.e., $\text{Moment}_{\text{exact}} = \frac{8\pi}{3} - \text{Moment}_{\text{principal}}$.

4. Results and Analysis

The NNR-MORVEL56 model contains 56 tectonic plates around the Earth, and the software OSXStereonet developed by Cardozo and Allmendinger [?] was applied to plot the global plate distribution map, as illustrated in [Figure 2: see original paper]. Utilizing the Fortran program, we estimated geometry parameters of all 56 modern plates with the accuracy of the inertia tensor better than 10^{-6} . lists the area and inertia tensor of all MORVEL plates, which provide essential material for calculating plate kinematic parameters in the NNR reference frame.

The sum of the areas of all plates equals 12.566340 steradians, which is slightly less than the surface area of the unit sphere, namely 4π (12.566371), with a relative error of 0.00023%. Our results indicate that the relative errors for the six components of the total tensor are 0.00017%, 0.00029%, 0.00026%, 0.0010%, 0.00016%, and 0.00010%, respectively. It is shown that the inertia tensor of the entire spherical surface is $\frac{8\pi}{3}\mathbf{I}$, where \mathbf{I} takes the identity matrix. The discrepancy probably arises from either the imperfection of plate coverage over the entire sphere or the unavoidable rounding errors in floating-point arithmetic.

[Figure 2: see original paper] Plate boundaries and geometries employed for MORVEL: (a) View direction at 35° south, 0° east; (b) View direction at 20° north, 180° east.

Geometric parameters of 56 modern plates included in NNR-MORVEL56. (Table content preserved exactly as in original)

5. Conclusion

The method for computing the areas and inertia tensors of tectonic plates has been presented. This method is based upon the triangulation algorithm and the adaptive Simpson's double integral procedure, which can be applied to spherical polygons representing such tectonic plates. Results for the NNR-MORVEL56 tectonic plates show that highly reliable data can be produced, as long as one starts from the precise definition of the plate boundaries. In addition, a FORTRAN90 program has been attached to the end of this paper, which is expected to be valuable for future studies of the kinematics and dynamics associated with the motions of tectonic plates.

References

- [1] Hamblin W K, Christiansen E H. Earth's dynamic systems [M]. 10th ed. State of New Jersey: Prentice Hall, 2003.
- [2] Larson K M, Freymueller J T, Philipson S. Global plate velocities from the global positioning system [J]. *Journal of Geophysical Research Solid Earth*, 1997, 102(B5): 9961-9981.
- [3] Sella G F, Dixon T H, Mao A. Revel: a model for recent plate velocities from space geodesy [J]. *Journal of Geophysical Research Solid Earth*, 2002, 107(B4): ETG 11-1- ETG 11-30.
- [4] Corne K, Holt W E, John H A. An integrated global model of present-day plate motions and plate boundary deformation [J]. *Geophysical Journal International*, 2003, 154(1): 8-34.
- [5] Altamimi Z, Metivier L, Collilieux X. ITRF 2008 plate motion model [J]. *Journal of Geophysical Research Solid Earth*, 2012, 117(B7): 47-56.
- [6] Peter B. An updated digital model of plate boundaries [J]. *Geochemistry Geophysics Geosystems*, 2003, 4(3): 101-112.

- [7] DeMets C, Gordon R G, Argus D F. Geologically current plate motions [J]. *Geophysical Journal International*, 2010, 181(1): 1-80.
- [8] Argus D F, Gordon R G. No-net-rotation model of current plate velocities incorporating plate motion model nuvel-1 [J]. *Geophysical Research Letters*, 1991, 18(11): 2039-2042.
- [9] Corne K, Holt W E. A no-net-rotation model of present-day surface motions [J]. *Geophysical Research Letters*, 2001, 28(23): 4407-4410.
- [10] Argus D F, Gordon R G, DeMets C. Geologically current motion of 56 plates relative to the no-net-rotation reference frame [J]. *Geochemistry Geophysics Geosystems*, 2011, 12(11): 75-87.
- [11] Klemann V, Martinec Z, Ivins E R. Glacial isostasy and plate motion [J]. *Journal of Geodynamics*, 2008, 46(3-5): 95-103.
- [12] Solomon S C, Sleep N H. Some simple physical models for absolute plate motions [J]. *Journal of Geophysical Research*, 1974, 79(17): 2557-2567.
- [13] Torsvik T H, Steinberger B, Gurnise M, et al. Plate tectonics and net lithosphere rotation over the past 150 my [J]. *Earth & Planetary Science Letters*, 2010, 291(1): 106-112.
- [14] Zuheir A, Patrick S, Claude B. The impact of a no-net-rotation condition on ITRF2000 [J]. *Geophysical Research Letters*, 2003, 30(2): 36-1-36-4.
- [15] Bevis M, Cambaresi G. Computing the area of a spherical polygon of arbitrary shape [J]. *Mathematical Geology*, 1987, 19(4): 335-346.
- [16] Heckbert P S. *Graphics Gems IV* [M]. Boston: Academic Press Professional, 1994: 132-137.
- [17] Schettino A. Computational methods for calculating geometric parameters of tectonic plates [J]. *Computers & Geosciences*, 1999, 25(8): 897-907.
- [18] Cardozo N, Allmendinger R W. Spherical projections with OSXStereonet [J]. *Computers & Geosciences*, 2013, 51: 193-205.

Appendix: The FORTRAN90 Program

```
MODULE CALCULATE_SPHPOLAERA_INERTEN
! THIS IS A FORTRAN90 MODULE, WHICH INCLUDES A SET OF FUNCTIONS AND SUBROUTINES
! USED FOR CALCULATING THE AREAS AND INERTIA TENSORS OF AMOUNTS SPHERICAL POLYGONS.
! *****
! DISCUSSION:
! VERTEXES OF THE SPHERICAL POLYGON MUST BE SEQUENCED IN A COUNTERCLOCKWISE DIRECTION.
! THE FIRST POINT IN BOUNDARY FILES SHOULD ALWAYS BE IDENTICAL TO THE LAST ONE
! IN ORDER TO CONSTRUCT A CLOSED POLYGON OF N SIDES WITH N+1 DATA POINTS.
! MODIFIED: 26 DECEMBER 2014
! AUTHOR: CHUNXIAO LI
```

IMPLICIT NONE

PARAMETER :: PI = 3.1415926535897932D0

PARAMETER :: HALFPI = PI/2.0D0

PARAMETER :: DEGREE = 180.0D0/PI !DEGREES PER RADIAN

*! SPITC RETURNS THE AREA AND SIX COMPONENTS OF THE INERTIA TENSOR OF A N-SIDED
! SPHERICAL POLYGON.*

*! THE LAT AND LON SHOULD TAKE DEGREE AS THEIR UNIT, RANGING FROM -90 TO 90 AND
! FROM -180 TO 180, RESPECTIVELY.*

! THE N+1 DATA POINTS PORTRAY A POLYGON WITH N VERTEXES.

*! THE OUTPUT PARAMETER, SPHERICALPOLYGONAREA, REPRESENTS THE AREA OF THE SPHERICAL POLYGON
! IN STERADIANS FOR A UNIT SPHERE.*

! THE REMAMENENT SIX PARAMETERS INDICATE SIX COMPONENTS OF THE INERTIA TENSOR

SUBROUTINE SPITC(LAT, LON, N, SPHERICALPOLYGONAREA, &
SUM11, SUM22, SUM33, SUM12, SUM13, SUM23)

IMPLICIT NONE

! EPS: THE TOLERANCE INVOLVED ON COMPUTATION IS SPECIFIED BY EPS.

INTEGER :: N

DOUBLE PRECISION :: LAT(N), LON(N)

DOUBLE PRECISION :: SPHERICALPOLYGONAREA

DOUBLE PRECISION :: SUM11, SUM22, SUM33, SUM12, SUM13, SUM23

DOUBLE PRECISION :: LON1, LON2, LAT1, LAT2

DOUBLE PRECISION :: HAVB, DLON, PDLON

DOUBLE PRECISION :: T, A, B, C, S, SUM, EXCESS

DOUBLE PRECISION :: C1, C2, C3

COMMON /GROUP1/ C1, C2, C3

DOUBLE PRECISION :: S11, S22, S33, S12, S13, S23

DOUBLE PRECISION :: EPS

COMMON /GROUP2/ EPS

SUM = 0.0D0

SUM11 = 0.0D0

SUM22 = 0.0D0

SUM33 = 0.0D0

SUM12 = 0.0D0

SUM13 = 0.0D0

SUM23 = 0.0D0

DO J = 1, N-1

LON1 = LON(J)/DEGREE

LAT1 = LAT(J)/DEGREE

LON2 = LON(J+1)/DEGREE

LAT2 = LAT(J+1)/DEGREE

CALL COEFFICIENT(LAT1, LON1, LAT2, LON2)

PDLON = LON2 - LON1

```
DLON = ABS(PDLON)

IF (DLON .GT. 1.0E-6) THEN
  IF (DLON .GT. PI) DLON = 2.0D0*PI - DLON
  IF (LON2 .LT. LON1 .AND. PDLON .LT. -PI) LON2 = LON2 + 2.0D0*PI
  IF (LON2 .GT. LON1 .AND. PDLON .GT. PI) LON2 = LON2 - 2.0D0*PI

  HAVB = HAV(LAT2-LAT1) + COS(LAT1)*COS(LAT2)*HAV(DLON)
  B = 2.0D0*ASIN(SQRT(HAVB))
  A = HALFPI - LAT1
  C = HALFPI - LAT2
  S = 0.5D0*(A + B + C)

  EXCESS = 4.0D0*ATAN(SQRT(TAN(S/2.0D0)*TAN((S-A)/2.0D0)* &
    TAN((S-B)/2.0D0)*TAN((S-C)/2.0D0)))

  IF (LON2 .LT. LON1) EXCESS = -EXCESS

  SUM = SUM + EXCESS

  CALL FSIM2(LON1, LON2, FS_LOWERLIMIT, FS_UPPERLIMIT, F11, EPS, S11)
  CALL FSIM2(LON1, LON2, FS_LOWERLIMIT, FS_UPPERLIMIT, F22, EPS, S22)
  CALL FSIM2(LON1, LON2, FS_LOWERLIMIT, FS_UPPERLIMIT, F33, EPS, S33)
  CALL FSIM2(LON1, LON2, FS_LOWERLIMIT, FS_UPPERLIMIT, F12, EPS, S12)
  CALL FSIM2(LON1, LON2, FS_LOWERLIMIT, FS_UPPERLIMIT, F13, EPS, S13)
  CALL FSIM2(LON1, LON2, FS_LOWERLIMIT, FS_UPPERLIMIT, F23, EPS, S23)

  SUM11 = SUM11 + S11
  SUM22 = SUM22 + S22
  SUM33 = SUM33 + S33
  SUM12 = SUM12 + S12
  SUM13 = SUM13 + S13
  SUM23 = SUM23 + S23
END IF
END DO

IF (SUM .LT. 0.0D0) THEN
  SPHERICALPOLYGONAREA = SUM + 4.0D0*PI
  SUM11 = 8.0D0/3.0D0*PI - SUM11
  SUM22 = 8.0D0/3.0D0*PI - SUM22
  SUM33 = 8.0D0/3.0D0*PI - SUM33
ELSE
  SPHERICALPOLYGONAREA = SUM
  SUM11 = SPHERICALPOLYGONAREA - SUM11
  SUM22 = SPHERICALPOLYGONAREA - SUM22
  SUM33 = SPHERICALPOLYGONAREA - SUM33
```

```
END IF

SUM12 = -SUM12
SUM13 = -SUM13
SUM23 = -SUM23

RETURN
END SUBROUTINE

!*****
! GIVEN THE DOMAIN OF INTEGRAL, FSIM2 RETURNS THE INTEGRAL OF F.
SUBROUTINE FSIM2(A, B, FS_LOWERLIMIT, FS_UPPERLIMIT, F, EPS, S)
  IMPLICIT NONE
  EXTERNAL FS_LOWERLIMIT, FS_UPPERLIMIT, F

  DOUBLE PRECISION :: A, B, EPS, S
  DOUBLE PRECISION :: H, S1, S2, TS1, TS2, X, G, S0, C
  INTEGER :: N, J

  N = 1
  H = 0.5D0*(B - A)
  C = (B - A)*1.0E-06

  CALL SIMP1(A, FS_LOWERLIMIT, FS_UPPERLIMIT, F, EPS, S1)
  CALL SIMP1(B, FS_LOWERLIMIT, FS_UPPERLIMIT, F, EPS, S2)
  TS1 = H*(S1 + S2)

  DO WHILE (H .GE. C)
    X = A - H
    TS2 = 0.5D0*TS1
    DO J = 1, N
      X = X + 2.0D0*H
      CALL SIMP1(X, FS_LOWERLIMIT, FS_UPPERLIMIT, F, EPS, G)
      TS2 = TS2 + H*G
    END DO

    S = (4.0D0*TS2 - TS1)/3.0D0
    N = N + N

    IF (N .GE. 16 .AND. ABS(S - S0) .LE. EPS*(ABS(S) + 1.0D0)) THEN
      RETURN
    END IF

    S0 = S
    TS1 = TS2
    H = 0.5D0*H
  END DO
```

```
END DO

RETURN
END SUBROUTINE

SUBROUTINE SIMP1(X, FS_LOWERLIMIT, FS_UPPERLIMIT, F, EPS, G)
  IMPLICIT NONE
  EXTERNAL FS_LOWERLIMIT, FS_UPPERLIMIT, F

  DOUBLE PRECISION :: X, EPS, G
  DOUBLE PRECISION :: Y1, Y2, H, C, TS1, TS2, Y, GO
  INTEGER :: N, I

  N = 1
  Y1 = FS_LOWERLIMIT(X)
  Y2 = FS_UPPERLIMIT(X)
  H = 0.5D0*(Y2 - Y1)
  C = (Y2 - Y1)*1.0E-06
  TS1 = H*(F(X, Y1) + F(X, Y2))

  DO WHILE (H .GE. C)
    Y = Y1 - H
    TS2 = 0.5D0*TS1
    DO I = 1, N
      Y = Y + 2.0D0*H
      TS2 = TS2 + H*F(X, Y)
    END DO

    G = (4.0D0*TS2 - TS1)/3.0D0
    N = N + N

    IF (N .GE. 16 .AND. ABS(G - GO) .LE. EPS*(ABS(G) + 1.0D0)) THEN
      RETURN
    END IF

    GO = G
    TS1 = TS2
    H = 0.5D0*H
  END DO

  RETURN
END SUBROUTINE
```

```
!*****
! FS_LOWERLIMIT RETURNS THE LOWER LIMIT CONDUCTING AS A FUNCTION OF
! LONGITUDE; AND FS_UPPERLIMIT RETURNS A CONSTANT UPPER LIMIT, PI/2.
```

```
FUNCTION FS_LOWERLIMIT(LON)
  IMPLICIT NONE
  DOUBLE PRECISION :: FS_LOWERLIMIT
  DOUBLE PRECISION :: LON
  DOUBLE PRECISION :: C1, C2, C3
  COMMON /GROUP1/ C1, C2, C3

  FS_LOWERLIMIT = -ATAN((C1*COS(LON) + C2*SIN(LON))/C3)
  RETURN
END FUNCTION
```

```
FUNCTION FS_UPPERLIMIT(LON)
  IMPLICIT NONE
  DOUBLE PRECISION :: FS_UPPERLIMIT
  DOUBLE PRECISION :: LON

  FS_UPPERLIMIT = HALFPI
  RETURN
END FUNCTION
```

```
!*****
! COEFFICIENT RETURNS THREE COEFFICIENTS RELATING THE SPHERICAL COORDINATES OF ANY
! TWO ADJACENT VERTEXES OF THE POLYGON.
```

```
SUBROUTINE COEFFICIENT(LAT1, LON1, LAT2, LON2)
  IMPLICIT NONE
  DOUBLE PRECISION :: LAT1, LON1, LAT2, LON2
  DOUBLE PRECISION :: C1, C2, C3
  COMMON /GROUP1/ C1, C2, C3

  C1 = SIN(LAT1)*COS(LON1)*COS(LAT2)*SIN(LON2) - &
      COS(LAT1)*SIN(LON1)*SIN(LAT2)*COS(LON2)
  C2 = SIN(LAT1)*SIN(LON1)*COS(LAT2)*COS(LON2) - &
      COS(LAT1)*COS(LON1)*SIN(LAT2)*SIN(LON2)
  C3 = COS(LAT1)*COS(LAT2)*SIN(LON2 - LON1)

  RETURN
END SUBROUTINE
```

```
!*****
! THE FOLLOWING INTEGRANDS CORRESPOND TO SIX COMPONENTS OF THE INERTIA TENSOR.
```

```
FUNCTION F11(LON, LAT)
  IMPLICIT NONE
  DOUBLE PRECISION :: F11
  DOUBLE PRECISION :: LAT, LON

  F11 = (COS(LAT))**3 * SIN(LAT) * (COS(LON))**2
```

```
RETURN  
END FUNCTION
```

```
FUNCTION F22(LON, LAT)  
  IMPLICIT NONE  
  DOUBLE PRECISION :: F22  
  DOUBLE PRECISION :: LAT, LON  
  
  F22 = (COS(LAT))**3 * SIN(LAT) * (SIN(LON))**2  
  RETURN  
END FUNCTION
```

```
FUNCTION F33(LON, LAT)  
  IMPLICIT NONE  
  DOUBLE PRECISION :: F33  
  DOUBLE PRECISION :: LAT, LON  
  
  F33 = (COS(LAT))**2 * SIN(LAT) * ((COS(LON))**2 + (SIN(LON))**2)  
  RETURN  
END FUNCTION
```

```
FUNCTION F12(LON, LAT)  
  IMPLICIT NONE  
  DOUBLE PRECISION :: F12  
  DOUBLE PRECISION :: LAT, LON  
  
  F12 = (COS(LAT))**3 * SIN(LAT) * COS(LON) * SIN(LON)  
  RETURN  
END FUNCTION
```

```
FUNCTION F13(LON, LAT)  
  IMPLICIT NONE  
  DOUBLE PRECISION :: F13  
  DOUBLE PRECISION :: LAT, LON  
  
  F13 = (COS(LAT))**2 * SIN(LAT) * COS(LAT) * COS(LON)  
  RETURN  
END FUNCTION
```

```
FUNCTION F23(LON, LAT)  
  IMPLICIT NONE  
  DOUBLE PRECISION :: F23  
  DOUBLE PRECISION :: LAT, LON  
  
  F23 = (COS(LAT))**2 * SIN(LAT) * COS(LAT) * SIN(LON)  
  RETURN
```

```
END FUNCTION
```

```
!*****  
! HAVERSINE FUNCTION  
FUNCTION HAV(X)  
  IMPLICIT NONE  
  DOUBLE PRECISION :: HAV  
  DOUBLE PRECISION :: X  
  
  HAV = (1.0D0 - COS(X))/2.0D0  
  RETURN  
END FUNCTION
```

```
END MODULE
```

```
!*****  
! END OF MODULE
```

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.