

# 巻貝殻形態の理論形態学的解析パッケージ Spiral Shell Form

野下浩司<sup>1</sup>

## Spiral Shell Form: A computer software package for theoretical morphological analysis of spiral shell form

Koji NOSHITA<sup>1</sup>

**Abstract** The Spiral Shell Form was developed as an original computer software package designed for theoretical morphological analysis of gastropod shell forms using the Raup's model. It is composed of the following three programs. Spiral Shell Measure is a software for capturing coordinate data of the points of interest on computer image files of a specimen. Raup's Parameter Calculator is a Mathematica program for calculation of Raup's parameters from coordinate data that were collected using the Spiral Shell Measure. Spiral Shell Properties is a Mathematica program for computing physical properties of a theoretical model such as shell volume, area of the aperture, center of gravity, soft-body ratio and length of trajectory of centroid of aperture.

**Key words:** theoretical morphology, morphometrics, Raup's model

### はじめに

貝類の殻形態の代表的な理論形態モデルとしてRaupモデルがある(Raup, 1962, 1966; Raup & Michelson, 1965)。Raupモデルとは殻口の形状を円または橢円で近似した母曲線で表し、その母曲線を巻軸を中心拡大・回転・変位させ、その軌跡によって殻形態を描くモデルである。巻貝の殻形態の記述だけを目的とすればより洗練されたモデルが存在するが(Bayer, 1978; McGhee, 1978; Løvtrup & Løvtrup, 1988; Okamoto, 1988a; Ackerly, 1989; Illert, 1989; Stone, 1995; Rice, 1998; Hammor & Bucher, 2005)，それらのモデルでは形態パラメータを実際の標本から測定・算出することが困難であり、また可能であったとしても非常に面倒である。そのため多数の標本に基づく理論形態空間解析を行うには、Raupモデルのように形態パラメータの取得が容易なモデルの方が有効である。

本稿ではMicrosoft Visual Basic 6.0で開発され画像から計測行うこととするSpiral Shell Measureと、Wolfram Research Mathematica 7.0で開発され、Raupパラメータを測定値から算出することを目的とするRaup's Parameter Calculator、及びRaupモデルに基づく各機能

パラメータの算出を目的とするSpiral Shell Properties からなるソフトウェアパッケージを紹介し、その概要と使用法を示す。

### Raupモデルについて

元々のRaupモデルでは、母曲線の拡大率( $W$ )、巻軸に沿った母曲線の転移率( $T$ )、巻軸に対する相対的な距離( $D$ )、橢円近似した母曲線の縦横比( $S$ )の四つのパラメータによって殻形態が描かれる。本稿では、これらのうち $S$ を除くパラメータ( $W$ ,  $T$ ,  $D$ )に加え、母曲線の傾きに関するパラメータ( $\Delta$ ,  $\Gamma$ )の計五つのパラメータを考える。 $\Delta$ は上下方向、 $\Gamma$ は左右方向の傾きを表す(図1)。これらRaupパラメータは巻軸から外側殻口端までの距離( $d_i$ )、螺旋の高さ( $f_i$ )、巻軸からカルスの内側殻口端までの距離( $c$ )から以下のように計算される。

$$W = e^{2 \frac{n \sum_{i=1}^n d_i \ln d_i - \sum_{i=1}^n i \sum_{i=1}^n \ln d_i}{n \sum_{i=1}^n i^2 - (\sum_{i=1}^n i)^2}},$$

<sup>1</sup> 静岡大学理学部地球科学科、422-8529 静岡市駿河区大谷836

<sup>1</sup> Institute of Geosciences, Shizuoka University, 836 Oya, Suruga-ku, Shizuoka 422-8529, Japan

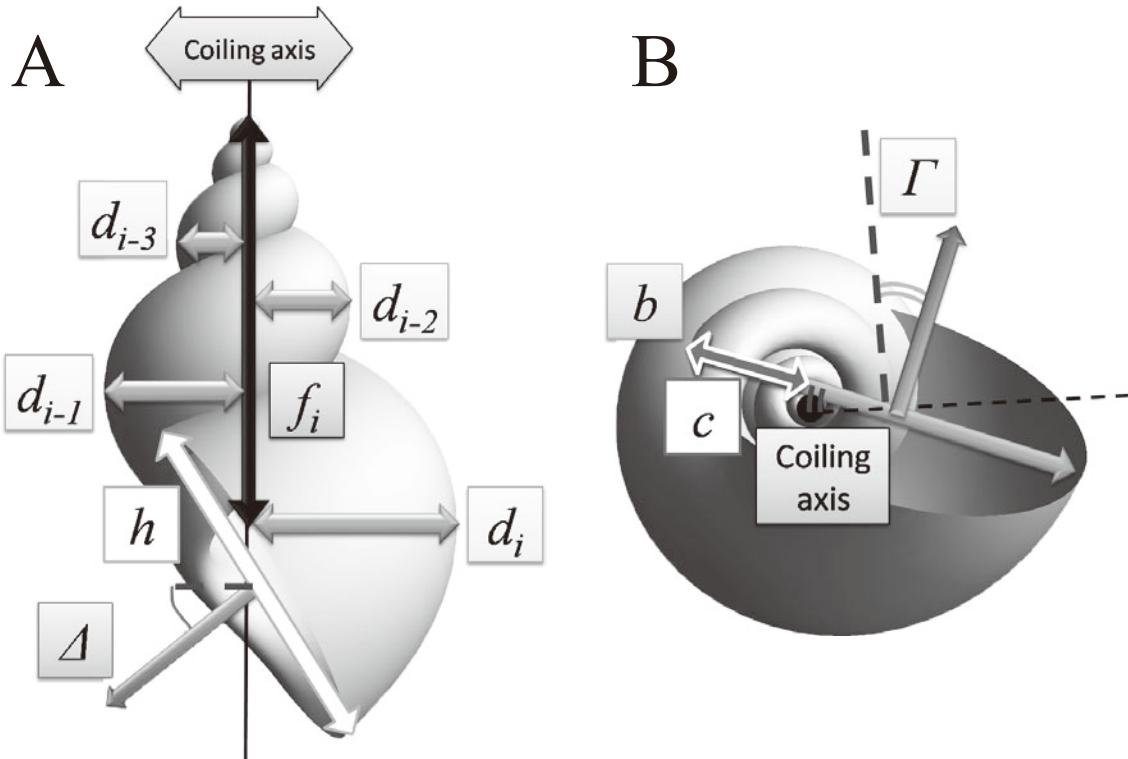


図1 Raup パラメータを求めるための測定値。A, 側方, B, 脣側。

Fig. 1 Measurement of Raup's parameter. (A) Lateral side. (B) Umbilical side.

$$T = \sqrt{\frac{\sum_{i=1}^n f_i^2 - \frac{1}{n} (\sum_{i=1}^n f_i)^2}{\sum_{i=1}^n d_i^2 - \frac{1}{n} (\sum_{i=1}^n d_i)^2}},$$

$$D = \frac{c}{b+c}.$$

### Spiral Shell Measure

Spiral Shell Measure (付録1) は上記の Raup パラメータを得るために、まず画像から  $d_i$ ,  $f_i$ ,  $c$ ,  $\Delta$ ,  $\Gamma$  を計測するためのプログラムである。

Spiral Shell Measure (以下 SSM) を起動すると図2のようなインターフェースが現れる。ドライブリストボックス (図2A), ディレクトリリストボックス (図2B), ファイルリストボックス (図2C) を用い計測を行いたい画像ファイルを選択する。“Aspect Ratio” フレーム (図2D) に選択した画像ファイルの縦横比を入力し, “Load” ボタン (図2E) をクリックすることでピクチャーボックス (図2F) に画像が表示される。SSMには三つの測定モードがあり (“Lateral side with  $\Delta$ ”, “Lateral side without  $\Delta$ ”, “Umbilical side”), 裂口の側方と巻軸を中心で90°回転させた側方及び臍側の三方向から撮影した画像に対応している。表示した画像に対応する測定モードを選択し (図2G, H), “measurement value” フレーム (図2I) 内の項目を上から順に測定していく。“Scale” は画像のスケーリングのための項目で、テキストボックスに入力し

た値幅に応じて画像内のスケール上の二点をクリックすることでスケーリングを行う。“Coiling axis” は巻軸を決めるための項目であり, “Lateral side” ではクリックした2点を結ぶ直線で, “Umbilical side” ではクリックした1点で巻軸を定義する。“d” は測定値 “ $d_i$ ” と “ $f_i$ ” を測定するための項目で、巻き始めの方から順に外側殻口端を半周毎に図1Aに示される矢印の端点をクリックすることで測定を行う。“h” は “Lateral side with  $\Delta$ ” では殻口の高さ “ $h$ ” 及び “ $\Delta$ ” を, “Lateral side without  $\Delta$ ” では “ $h$ ” だけを測定するための項目で、 $h$ を定める線分 (図1A) の両端をクリックすることで  $h$ の値を得る。加えて, “Lateral side with  $\Delta$ ” の場合には螺環の成長方向を,  $h$ を定める線分を含む直線によって分けられる領域のいずれかをクリックすることで定め,  $\Delta$ を求める。“b” は測定値 “ $b$ ” 及び “ $\Gamma$ ” を測定するための項目で、 $b$ を定める線分 (図1B) の両端をクリックすることで  $b$ の値を得る。また、螺環の成長方向を,  $b$ を定める線分を含む直線によって分けられる領域のいずれかをクリックすることで定め,  $\Gamma$ を求める。各測定モードでの測定が終了したら、標本名または標本番号入力のためのテキストボックス (図2J) にいずれかを入力し、必要があれば情報入力用テキストボックス (図2K) に情報を入力する。最後に “Output” ボタン (図2L) をクリックし、データ保管用のExcelファイルに出力する。出力はそれぞれ図3のように行われる。

本パッケージでは Raup's Parameter Calculator で Raup パラメータを算出することを想定しているため、Lateral

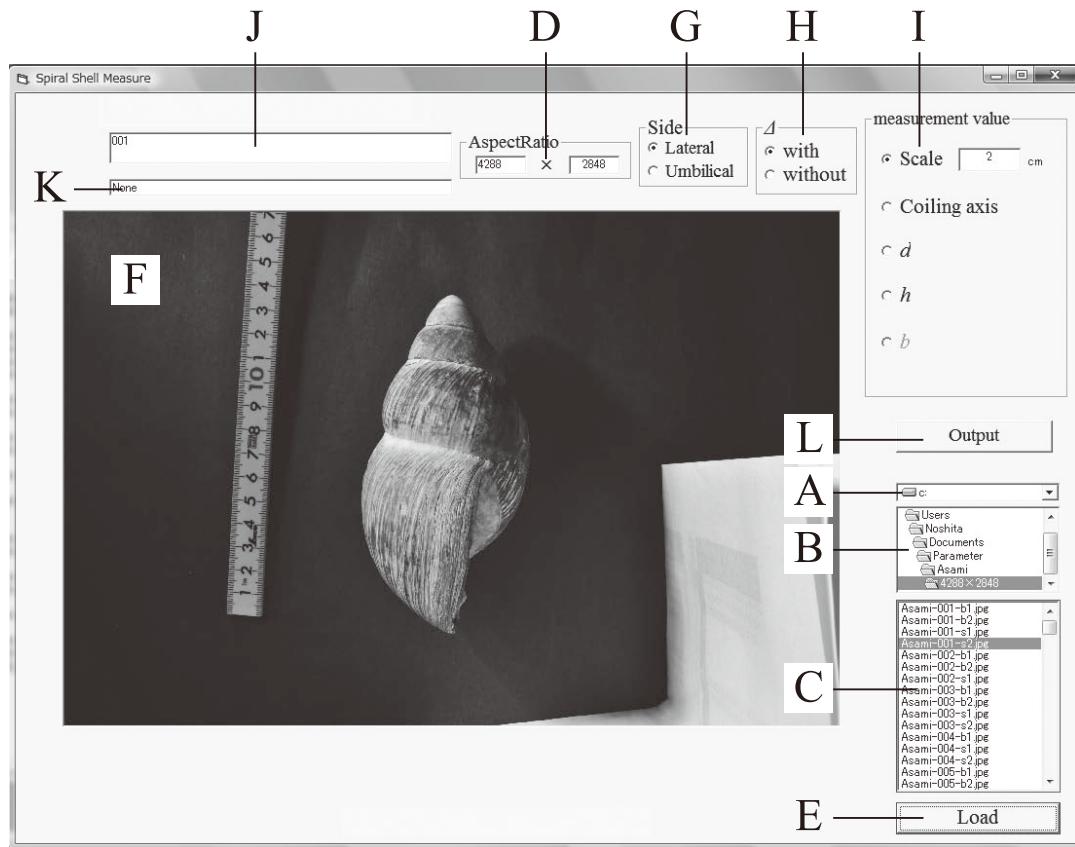


図2 Spiral Shell Measure のインターフェース。詳細は本文参照。

Fig. 2 Graphical user interface of Spiral Shell Measure. See text in deal.

A	B	F	G	D	E
001	8	1.386593	2.085536	2.555923	3.519169
None		7.755007	-0.22631	End	
001	8	6.596321	-0.18296	-0.40699	End
002		2.358979	2.826354		
None		2.1461278	1.75864		
002		2.46354	-1.05924	End	
003		2.716043	0.857953	-0.12503	End
		0.260568	0.456349	0.633105	1.030754
					1.415792
					2.005524

図3 Spiral Shell Measure 及びRaup's Parameter Calculatorで用いるデータファイルの形式。ダッシュの枠内が一標本に用いる範囲。A, 標本番号。B,  $d_i$ 及び $f_i$ の個数。C, 情報欄。D,  $d_i$ の行。E,  $f_i$ の行。F,  $h$ のセル。G,  $\Delta$ のセル。H,  $b$ のセル。I,  $c$ のセル。J,  $\Gamma$ のセル。Fig. 3 An example of data file for Spiral Shell Measure and Raup's Parameter Calculator. (A) Serial number of specimen. (B) Number of  $d_i$  and  $f_i$ . (C) Memorandum of specimen. (D)–(J) Values of  $d_i$ ,  $f_i$ ,  $h$ ,  $\Delta$ ,  $b$ ,  $c$  and  $\Gamma$ .

side with  $\Delta$ , Umbilical side の順に測定を行い、一つの標本につき計4行の出力を行う（図3）。しかし、Raup's Parameter Calculatorを用いずにRaupパラメータを算出する場合はこの限りではない。

### Raup's Parameter Calculator

本プログラムはRaupパラメータをSSMでの測定値から算出するプログラムである（付録2）。インターフェースはMathematicaノートブックを用い、そこにSSMで測定したデータを保存しているExcelファイルを組み込みシンボルImportで取り込み、本体であるMathematicaパッケージ（付録2）を組み込むことで計算を実行する。出力はMathematica上でシンボルW, T, Dd,  $\Delta$ ,  $\Gamma$ にそれぞれ格納される。

### Spiral Shell Properties

本プログラムはRaupモデルに基づき描かれる巻貝の殻形態の機能パラメータ（姿勢の安定性、殻の形成効率、外敵に対する防御効率）を見積もるための各パラメータ（二通りの姿勢における重心位置、体積、殻口面積、軟体部率、殻口中心の軌跡の長さ）を見積もることを目的に開発された（付録3）。Mathematicaノートブックをインターフェースとし、RaupパラメータW, T, D,  $\Delta$ ,  $\Gamma$

の組み合わせを入力し、本体であるMathematicaパッケージ（付録3）を読み込むことで実行する。出力はMathematica上でシンボルGkak（殻口に垂直な方向に軟体部が伸びると仮定した姿勢における重心位置）、Geok（殻口中心の軌跡の接線方向に軟体部が伸びると仮定した姿勢における重心位置）、V（全体の体積）、S（殻口面積）、BR（軟体部率）、Acok（殻口中心の軌跡の長さ）にそれぞれ格納される。

### 重心・体積

巻貝の重心を求める方法は、平巻きの巻貝についてはRaup & Chamberlain (1967) が、異常巻きのアンモノイドについてはOkamoto (1988b) でそれぞれ考案されているが、いずれも“欠損部”を持つ巻貝の重心を求めることには向いていない。欠損部とは螺環が一周前の螺環に重複するために“欠けて”しまう部分のことである（図4）。Raup and Graus (1972) は一般的な巻貝の体積を見積もる式を示しており、またStone (1997) は螺環の重複を考慮し不等成長する巻貝でもより高い精度で体積を求めることが出来るwhorl-by-whorl法を提出している。しかし、Stone (1997) の方法は非常に面倒で多数の標本の解析を行うのには向いていない。そこで本プログラムではOkamoto (1988b) で示されたものと同様な曲線近傍の体積を求める方法を採用する。

まず、殻口を円で近似した母曲線を十分に分割する。

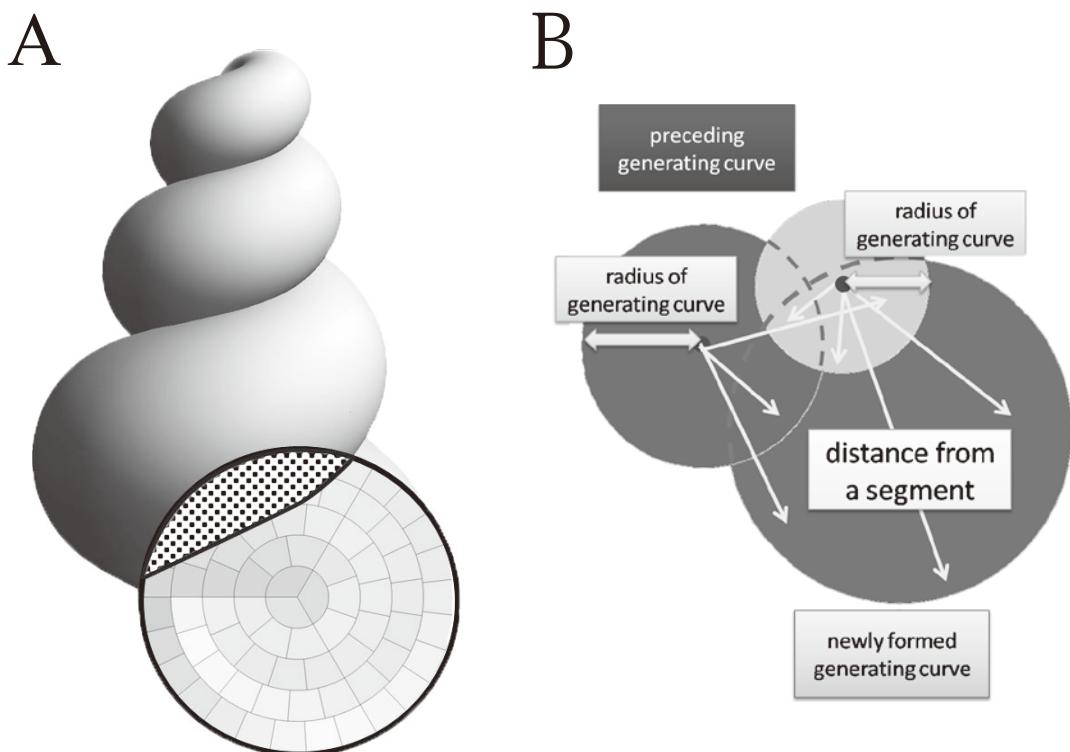


図4 “欠損部”。A、側方からみた場合。ドットの打ってある範囲が欠損部にあたる。B、巻軸を含む断面。以前の螺環により最も新たに作られた螺環は“欠ける”。

Fig. 4 “Hollow section”. (A) Lateral side. Dotted area is hollow section. (B) Cross section include coiling axis.

これら各部位を  $Sg_{ij}$  とし、その位置は図心により代表される。 $i$  と  $j$  は分割された一つの部位を示すインデックスであり、本プログラムでは 1,200 の部位に分割した各部位それを  $i$  と  $j$  の組で一意に示す。各  $Sg_{ij}$  について欠損部かどうかを、最も新しい殻口上に存在する  $Sg_{ij}$  から以前の母曲線の中心への各距離を各母曲線の半径と比較することで判定する（図4B）。各母曲線の半径より  $Sg_{ij}$  から以前の各母曲線への距離が短ければ、その部位を欠損部とする。この判定を 1,200 すべての部位に関して行うことで殻口上の特定の領域が欠損部となる（図4）。また、体積も  $Sg_{ij}$  毎に求める。始めに  $Sg_{ij}$  の図心の軌跡を考え、その軌跡での位置を  $s$  であらわす。体積の過大評価を防ぐためその接線方向に垂直な面に  $Sg_{ij}$  を射影し  $Sg_{ij}'$  とする。次に  $Sg_{ij}'$  と図心を共有し、かつ同じ面積を持つ円  $c_{ij}$  を考える。 $s$  で  $c_{ij}$  の面積を積分すると以下のように  $c_{ij}$  の軌跡の体積  $v_{ij}$  が計算される。

$$v_{ij} = \int \pi r(s)^2 ds$$

$r(s)$  は  $c_{ij}$  の  $s$  での半径である。ただし、 $r(s)$  は  $s$  での曲率半径より小さくなくてはならない。こうして求まった  $v_{ij}$  を欠損部を除き積算することで巻貝全体の体積を求める。またこれと同様に、重心も以下のように求められる。

$$\begin{aligned} p_{ij} &= \frac{\int_0^l \pi r(s)^2 X(s) ds}{\int_0^l \pi r(s)^2 ds} \\ &= \frac{\int_0^l r(s)^2 X(s) ds}{\int_0^l r(s)^2 ds} \end{aligned}$$

$X(s)$  は  $s$  における  $c_{ij}$  の図心の座標である。こうして求まった  $p_{ij}$  を欠損部を除き、質量で重みづけしてから積算すれば巻貝全体の重心が求まる。

こうして求まった巻貝の重心を基に、二通りの姿勢をとった場合、すなわち 1) 殻口に垂直な方向に軟体部が伸びると仮定した場合と 2) 殻口中心の軌跡の接線方向に軟体部が伸びると仮定した場合の重心位置をそれぞれ求める。

### 軟体部率

ここでは殻の厚さを一定としたときの巻貝全体の体積に対する軟体部（殻を除いた領域）の体積の比率を軟体部率と呼ぶ。つまり、

$$E = \frac{Vs}{V}.$$

$E$  は軟体部率、 $Vs$  は軟体部の体積、 $V$  は巻貝全体の体積である。本プログラムでは殻の厚さは螺環半径の 1/20 としている。

### 殻口中心の軌跡の長さ

殻口中心の軌跡の長さは以下のように求められる。

$$A = \frac{(-1 + W^{\frac{\theta_{\max}}{2\pi}})\sqrt{(4\pi^2(x^2+y^2)+(x^2+y^2+z^2)\log W^2}}}{\log W}.$$

$W$  は Raup パラメータの螺環の拡大率、 $\theta_{\max}$  は巻数に  $2\pi$  をかけたもの、 $\{x, y, z\}$  は殻口中心の座標である。本プログラムでは  $\theta_{\max}$  は  $16\pi$  としてある。

### おわりに

Spiral Shell Form は Raup モデルに基づき設計されているが、Spiral Shell Propertiesにおいて用いた曲線近傍の体積を求める方法は Okamoto (1988b) でも採用されている通り、Raup モデルに限らず適用することが出来る。そのため、軟体部率や殻口中心の軌跡の長さについても基本的に同様の方法で他の巻貝形態の理論形態モデルにおいて見積もることが可能である。

静岡大学理学部地球科学教室の生形貴男博士には本パッケージの開発において様々な助言を頂いた。ここに記して謝意を表する。

### 引用文献

- Ackerly S. C. (1989), Kinematics of accretionary shell growth, with examples from brachiopods and mollusks. *Paleobiology*, **15**, 147–164.
- Bayer U. (1978), Morphogenetic programs, instabilities, and evolution—a theoretical study. *Neues Jahrbuch für Geologie und Paläontologie, Abhandlungen*, **156**, 226–261.
- Hammer Ø. & Bucher H. (2005), Models for the morphogenesis of the molluscan shell. *Lethaia*, **38**, 111–122.
- Illert C. (1989), Formulation and solution of the classical seashell problem. II.—Tubular three-dimensional seashell surfaces. *Il Nuovo Cimento*, **D11**, 761–780.
- Løvtrup S. & Løvtrup M. (1988), The morphogenesis of molluscan shells: a mathematical account using biological parameters. *Journal of Morphology*, **197**, 53–62.
- McGhee G. R. Jr. (1978), Analysis of the shell torsion phenomenon in the Bivalvia. *Lethaia*, **11**, 315–329.
- Okamoto T. (1988a), Analysis of heteromorphy ammonoids by differential geometry. *Palaeontology*, **31**, 35–52.
- Okamoto T. (1988b), Changes in life orientation during the ontogeny of some heteromorphy ammonoids. *Palaeontology*, **31**, 281–294.
- Raup D. M. (1962), Computer as aid in describing form in gastropod shells. *Science*, **138**, 150–152.
- Raup D. M. (1966), Geometric analysis of shell coiling: general problems. *Journal of Paleontology*, **44**, 1178–1190.
- Raup D. M. & Chamberlain A. Jr. (1967), Equations for volume and center of gravity in ammonoid shells. *Journal of Paleontology*, **41**, 566–574.
- Raup D. M. & Michelson A. (1965), Theoretical morphology of the coiled shell. *Science*, **147**, 1294–1295.
- Raup D. M. & Graus R. R. (1972), General equations for volume and surface area of a longarithmically coiled

- shell. *Mathematical Geology*, **4**, 307–316.

Rice S. H. (1998), The bio-geometry of mollusc shells. *Paleobiology*, **24**, 133–149.

Stone J. R. (1995), CerioShell: a computer program designed to simulate variation in shell form. *Paleobiology*, **21**, 509–519.

Stone J. R. (1997), Mathematical determination of coiled shell volumes and surface areas. *Lethaia*, **30**, 213–219.

## 付録1 Spiral Shell Measure のソースコード.

## **Appendix 1 Computer programs that compose the Spiral Shell Measure.**



```

Dim delx2 As Single, dely2 As Single
Dim vecDelX1 As Single, vecDelY1 As Single
Dim vecDelX2 As Single, vecDelY2 As Single
Dim vecCoSX As Single, vecCoSY As Single
Dim innerdel1 As Single, innerdel2 As Single
Dim delx As Single, dely As Single
Dim deldel1 As Single, deldel2 As Single
Dim gamx1 As Single, gamy1 As Single
Dim gamx2 As Single, gamy2 As Single
Dim vecGamX1 As Single, vecGamY1 As Single
Dim vecGamX2 As Single, vecGamY2 As Single
Dim innergam1 As Single, innergam2 As Single
Dim gamx As Single, gamy As Single
Dim gamgam1 As Single, gamgam2 As Single
'Digitizer*****
picSample.Circle (X, Y), 30, RGB(255, 0, 0)
j = j + 1
ccx(j) = X
ccy(j) = Y
If opScale.Value = True Then
    i = i + 1
    cx(i) = X
    cy(i) = Y
    If i = 2 Then
        startX = cx(1)
        startY = cy(1)
        goalX = cx(2)
        goalY = cy(2)
        Scl = (((goalX - startX) ^ 2 + (goalY - startY) ^ 2) ^ (1 / 2)) _
            / txtScale.Text
        picSample.Line (startX, startY)-(goalX, goalY)
        i = 0
    End If
End If
'Lateral side-----
If opLateral.Value = True Then
    If opCoS.Value = True Then
        i = i + 1
        cx(i) = X
        cy(i) = Y
        If i = 2 Then
            startX = cx(1)
            startY = cy(1)
            goalX = cx(2)
            goalY = cy(2)
            If startX = goalX Then
                goalX = goalX + 0.01
            End If
            CoSA = -((goalY - startY) / (-goalX + startX))
            CoSB = -((-goalY * startX) + goalX * startY) / (-goalX + startX)
            PerpA = (-goalX + startX) / (goalY - startY)
            picSample.Line (startX, startY)-(goalX, goalY)
            i = 0
        End If
    End If

```

```

End If
If opd.Value = True Then
  k = k + 1
  ddx(k) = X
  ddy(k) = Y
  PerpB(k) = -PerpA * X + Y
  CrossX(k) = -((CoSB - PerpB(k)) / (CoSA - PerpA))
  CrossY(k) = -((CoSB * PerpA - CoSA * PerpB(k)) / (CoSA - PerpA))
  picSample.Circle (CrossX(k), CrossY(k)), 30, RGB(255, 0, 0)
  picSample.Line (CrossX(k), CrossY(k))-ddx(k), ddy(k))
  d(k) = (((X - CrossX(k)) ^ 2 + (Y - CrossY(k)) ^ 2) ^ (1 / 2)) / Scl
  f(k) = (((CrossX(k) - startX) ^ 2 -
            + (CrossY(k) - startY) ^ 2) ^ (1 / 2)) / Scl
  nN = k
End If
If oph.Value = True Then
  i = i + 1
  cx(i) = X
  cy(i) = Y
  If i = 2 Then
    hstartX = cx(1)
    hstartY = cy(1)
    hgoalX = cx(2)
    hgoalY = cy(2)
    picSample.Line (hstartX, hstartY)-(hgoalX, hgoalY)
    h = (((hgoalX - hstartX) ^ 2 -
           + (hgoalY - hstartY) ^ 2) ^ (1 / 2)) / Scl
  If opCdel0.Value = True Then
    i = 0
  End If
End If
If opCdel1.Value = True Then
  If i = 3 Then
    delx = cx(3)
    dely = cy(3)
    vechX = hgoalX - hstartX
    vechY = hgoalY - hstartY
    vecDelX1 = -(vechY / Sqr(vechX ^ 2 + vechY ^ 2))
    vecDelY1 = vechX / Sqr(vechX ^ 2 + vechY ^ 2)
    vecDelX2 = vechY / Sqr(vechX ^ 2 + vechY ^ 2)
    vecDelY2 = -(vechX / Sqr(vechX ^ 2 + vechY ^ 2))
    vecCoSX = goalX - startX
    vecCoSY = goalY - startY
    innerdel1 = vecDelX1 * vecCoSX + vecDelY1 * vecCoSY
    del1 = ArcCos(innerdel1 / ((vecCoSX ^ 2 + vecCoSY ^ 2) -
                                ^ (1 / 2)))
    delx1 = ((hstartX + hgoalX) / 2) + 1000 * vecDelX1
    dely1 = ((hstartY + hgoalY) / 2) + 1000 * vecDelY1
    deldel1 = ((delx - delx1) ^ 2 + (dely - dely1) ^ 2) ^ (1 / 2)
    innerdel2 = vecDelX2 * vecCoSX + vecDelY2 * vecCoSY
    del2 = ArcCos(innerdel2 / ((vecCoSX ^ 2 + vecCoSY ^ 2) -
                                ^ (1 / 2)))
    delx2 = ((hstartX + hgoalX) / 2) + 1000 * vecDelX2
    dely2 = ((hstartY + hgoalY) / 2) + 1000 * vecDelY2
  End If
End If

```

```

deldel2 = ((delx - delx2) ^ 2 + (dely - dely2) ^ 2) ^ (1 / 2)
If deldel1 < deldel2 Then
    del = del1
    picSample.Line ((hstartX + hgoalX) / 2,_
                    (hstartY + hgoalY) / 2)_-
                    -(delx1, dely1)
Else
    del = del2
    picSample.Line ((hstartX + hgoalX) / 2,_
                    (hstartY + hgoalY) / 2)-(delx2, dely2)
End If
i = 0
End If
End If
End If
End If
'Umbilical side-----
If opUmbilical.Value = True Then
    If opCoS.Value = True Then
        CosX = X
        CosY = Y
    End If
    If opb.Value = True Then
        k = k + 1
        cx(k) = X
        cy(k) = Y
        If k = 2 Then
            bstartX = cx(1)
            bstartY = cy(1)
            bgoalX = cx(2)
            bgoalY = cy(2)
            picSample.Line (bstartX, bstartY)-(bgoalX, bgoalY)
            CrossX(1) = (bstartX + bgoalX) / 2
            CrossY(1) = (bstartY + bgoalY) / 2
            picSample.Circle (CrossX(1), CrossY(1)), 30, RGB(255, 0, 0)
            b = (((bgoalX - bstartX) ^ 2 + (bgoalY - bstartY) ^ 2) ^ (1 / 2)) / Scl
            c = (((CrossX(1) - CosX) ^ 2 + (CrossY(1) - CosY) ^ 2) _-
                  ^ (1 / 2)) / Scl - b / 2
        End If
        If k = 3 Then
            gamx = cx(3)
            gamy = cy(3)
            vecbX = bgoalX - bstartX
            vecbY = bgoalY - bstartY
            vecGamX1 = -(vecbY / Sqr(vecbX ^ 2 + vecbY ^ 2))
            vecGamY1 = vecbX / Sqr(vecbX ^ 2 + vecbY ^ 2)
            vecGamX2 = vecbY / Sqr(vecbX ^ 2 + vecbY ^ 2)
            vecGamY2 = -(vecbX / Sqr(vecbX ^ 2 + vecbY ^ 2))
            vecCoSX = CrossX(1) - CosX
            vecCoSY = CrossY(1) - CosY
            innergam1 = vecGamX1 * vecCoSX + vecGamY1 * vecCoSY
            gam1 = ArcCos(innergam1 / ((vecCoSX ^ 2 + vecCoSY ^ 2) _-
                ^ (1 / 2)))
            gamx1 = CrossX(1) + 1000 * vecGamX1
        End If
    End If
End If

```



```

If opUmbilical.Value = True Then
    xlsSheet.Cells(i, 2).Value = b
    xlsSheet.Cells(i, 3).Value = c
    xlsSheet.Cells(i, 4).Value = gam
    xlsSheet.Cells(i, 5).Value = "End"
End If
xlsBook.Close saveChanges:=True
xlsApp.Quit
Set xlsSheet = Nothing
Set xlsBook = Nothing
Set xlsApp = Nothing
End Sub

```

## 付録2 Raup's Parameter Calculator のソースコード.

## **Appendix 2** Computer programs that compose the Raup's Parameter Calculator.

### 付録3 Spiral Shell Properties のソースコード.

### **Appendix 3 Computer programs that compose the Spiral Shell Properties.**

```

(in sideset[[i,j]]+outsideset[[i,j-1]])/2]],{i,1,imax,1},{j,1,6i-3,1}];
sv=Table[N[ss[[i,j,k]]-gcset[[i,j]]],{i,1,imax,1},{j,1,6i-3,1},{k,1,If[i==1,6,8]}];
gv=Table[N[{(Log[W] gcset[[i,j,1]]-2 π gcset[[i,j,2]])}_
/(Sqrt[(Abs[(Log[W] gcset[[i,j,1]]_-
-2 π gcset[[i,j,2]])/Sqrt[4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)])]^2_-
+Abs[(2 π gcset[[i,j,1]]+Log[W] gcset[[i,j,2]])/Sqrt[4 π ^2 (gcset[[i,j,1]]^2_-
+gcset[[i,j,2]]^2)+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)]]^2_-
+Abs[(Log[W] gcset[[i,j,3]])/Sqrt[4 π ^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)]]^2_-
*(4 π ^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-
+Log[W]^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2))]),_-
(2 π gcset[[i,j,1]]+Log[W] gcset[[i,j,2]])/(Sqrt[(Abs[(Log[W] gcset[[i,j,1]]_-
-2 π gcset[[i,j,2]])/Sqrt[4 π ^2 (gcset[[i,j,1]]^2+geset[[i,j,2]]^2)-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)])]^2_-
+Abs[(2 π gcset[[i,j,1]]+Log[W] gcset[[i,j,2]])/Sqrt[4 π ^2 (gcset[[i,j,1]]^2_-
+gcset[[i,j,2]]^2)+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)]]^2_-
+Abs[(Log[W] gcset[[i,j,3]])/Sqrt[4 π ^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-
+Log[W]^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)]]^2_-
*(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2))),_-
(Log[W] geset[[i,j,3]])/(Sqrt[(Abs[(Log[W] gcset[[i,j,1]]-2 π gcset[[i,j,2]])_-
/Sqrt[4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)])]^2_-
+Abs[(2 π gcset[[i,j,1]]+Log[W] gcset[[i,j,2]])/Sqrt[4 π ^2 (gcset[[i,j,1]]^2_-
+gcset[[i,j,2]]^2)+Log[W]^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)]]^2_-
+Abs[(Log[W] gcset[[i,j,3]])/Sqrt[4 π ^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)]]^2_-
*(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2))),_-
{i,1,imax,1},{j,1,6i-3,1}];

'Projection *****
insgvN=Table[{k,N[ArcCos[sv[[i,j,k]].gv[[i,j]]/Norm[sv[[i,j,k]]]]]},_-
{i,1,imax,1},{j,1,6i-3,1},{k,1,If[i==1,6,8],1}];
sortinsgvN=Table[Sort[insgvN[[i,j]],#1[[2]]<#2[[2]]&],{i,1,imax,1},{j,1,6i-3,1}];
θ 12=Table[sortinsgvN[[i,j,k]],[i,1,imax,1},{j,1,6i-3,1},{k,1,2,1}];
unitsv12=Table[N[Normalize[sv[[i,j,θ 12[[i,j,k,1]]]]]],_-
{i,1,imax,1},{j,1,6i-3,1},{k,1,2,1}];
KK[a_,b_,g_]:=-(Abs[a.b a.g-2 b.g] a.b+(-4+(a.b)^2) a.g)_-
/(2 Sqrt[(4-(a.b)^2) ((a.g)^2-a.b a.g b.g+(b.g)^2)]);
LL[a_,b_,g_]:=Sqrt[(a.b a.g-2 b.g)^2]/Sqrt[(4-(a.b)^2) ((a.g)^2-a.b a.g b.g+(b.g)^2)];
area=Table[Normalize[KK[unitsv12[[i,j,1]],unitsv12[[i,j,2]],gv[[i,j]]]*unitsv12[[i,j,1]]_-
+LL[unitsv12[[i,j,1]],unitsv12[[i,j,2]],gv[[i,j]]]*unitsv12[[i,j,2]]],_-
{i,1,imax,1},{j,1,6i-3,1}];
angleareag=Table[ArcCos[N[area[[i,j]].gv[[i,j]]]],{i,1,imax,1},{j,1,6i-3,1}];
angleareagCorrect=angleareag;
angleareagCorrect[[1]]=Table[If[angleareag[[1,j]]>Pi/2,Pi-angleareag[[1,j]],_-
angleareag[[1,j]]],{j,1,3,1}];
aS=Table[ π /(3 imax^2) Cos[Pi/2-angleareagCorrect[[i,j]]],{i,1,imax,1},{j,1,6i-3,1}];
'Estimate volume *****
cr=Table[Sqrt[aS[[i,j]]/Pi],[i,1,imax,1},{j,1,6i-3,1}];
volS=Table[1/3 π cr[[i,j]]^2 A[[i,j]] (3+Log[W] A[[i,j]] ((Log[W] A[[i,j]])_-
/(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)+Log[W]^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2_-
+gcset[[i,j,3]]^2))+3/(Sqrt[(4 π ^2 (geset[[i,j,1]]^2+gcset[[i,j,2]]^2)_-

```

```

+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2)))))_  
,{i,1,imax,1},{j,1,6i-3,1}];  
shellV=Sum[e[[i,j]]*volS[[i,j]],{i,(1-SR)*imax,imax,1},{j,1,6i-3,1}];  
V=Sum[e[[i,j]]*volS[[i,j]],{i,1,imax,1},{j,1,6i-3,1}];  
cogS=Table[{(3 NIntegrate[(cr[[i,j]]^2 π ( α Log[W]  
+Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2]))]^3  
*(gcset[[i,j,1]] Cos[1/Log[W] 2 π Log[1+( α Log[W])_  
/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2))]]]  
-gcset[[i,j,2]] Sin[1/Log[W] 2 π Log[1+( α Log[W])_  
/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2))]]])]  
/(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2  
+gcset[[i,j,3]]^2) Log[W]^2)^{(3/2),{ α ,0,A[[i,j]]},AccuracyGoal->16}]_  
/( π cr[[i,j]]^2 A[[i,j]] (3+Log[W] A[[i,j]] ((Log[W] A[[i,j]]])  
/(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2))_  
+3/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2))])])]  
(3 NIntegrate[(cr[[i,j]]^2 π ( α Log[W]+Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2))]^3 _  
*(gcset[[i,j,2]] Cos[1/Log[W] 2 π Log[1+( α Log[W])_  
/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2))]]]  
+gcset[[i,j,1]] Sin[1/Log[W] 2 π Log[1+( α Log[W])_  
/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2))]]])]  
/(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)+(gcset[[i,j,1]]^2  
+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2)^{(3/2),{ α ,0,A[[i,j]]},AccuracyGoal->16}]_  
/( π cr[[i,j]]^2 A[[i,j]] (3+Log[W] A[[i,j]] ((Log[W] A[[i,j]])/(4 π ^2 (gcset[[i,j,1]]^2  
+gcset[[i,j,2]]^2)+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2+gcset[[i,j,3]]^2))+3_  
/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2)  
+Log[W]^2 (gcset[[i,j,1]]^2+gcset[[i,j,2]]^2_  
+gcset[[i,j,3]]^2))])),(3 NIntegrate[(cr[[i,j]]^2 π geset[[i,j,3]] ( α Log[W]  
+Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+geset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+geset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2))]^3 _  
/(4 π ^2 (gcset[[i,j,1]]^2+geset[[i,j,2]]^2)  
+(gcset[[i,j,1]]^2+geset[[i,j,2]]^2+gcset[[i,j,3]]^2) Log[W]^2)^{(3/2),  
{ α ,0,A[[i,j]]},AccuracyGoal->16})/( π cr[[i,j]]^2 A[[i,j]] (3+Log[W] A[[i,j]]_  
*((Log[W] A[[i,j]])/(4 π ^2 (gcset[[i,j,1]]^2+geset[[i,j,2]]^2)  
+Log[W]^2 (gcset[[i,j,1]]^2+geset[[i,j,2]]^2+gcset[[i,j,3]]^2))_  
+3/(Sqrt[(4 π ^2 (gcset[[i,j,1]]^2+geset[[i,j,2]]^2)+Log[W]^2 (gcset[[i,j,1]]^2  
+geset[[i,j,2]]^2+gcset[[i,j,3]]^2))]))},{i,1,imax,1},{j,1,6i-3,1}];  
G=Sum[ ρ [[i,j]]*e[[i,j]]*volS[[i,j]]*cogS[[i,j]],{i,1,imax,1},{j,1,6i-3,1}]_  
/Sum[ ρ [[i,j]]*e[[i,j]]*volS[[i,j]],{i,1,imax,1},{j,1,6i-3,1}];  
'Body ratio'*****  
BR=(V-shellV)/V;  
'Area of apearture'*****  
aSlast=Map[2^( θ max/ π ) #^2 π &,cr,{2}];  
'center of gravity in each posture'*****  
kakkou=Table[N[{W^( θ max/(2 π )) (Cos[ θ max] gcset[[i,j,1]]-gcset[[i,j,2]] Sin[ θ max]),_  
W^( θ max/(2 π )) (Cos[ θ max] gcset[[i,j,2]]+gcset[[i,j,1]] Sin[ θ max]),_  
W^( θ max/(2 π )) gcset[[i,j,3]]}],{i,1,imax,1},{j,1,6i-3,1}];
```

```

cok=Sum[ τ [[i,j]]*e[[i,j]]*kakkou[[i,j]],[i,1,imax,1],[j,1,6i-3,1]]/Sum[e[[i,j]],_
{i,1,imax,1},{j,1,6i-3,1}];  

S=Sum[e[[i,j]]* τ [[i,j]]*aSlast[[i,j]],[i,1,imax,1],[j,1,6i-3,1]];  

cok0s=Sum[ τ [[i,j]]*e[[i,j]]*gcset[[i,j]],[i,1,imax,1],[j,1,6i-3,1]]/Sum[e[[i,j]],_
{i,1,imax,1},{j,1,6i-3,1}];  

Acok=1/Log[W] (-1+W^( θ max/(2 π ))) Sqrt[(4 π ^2 (cok0s[[1]]^2+cok0s[[2]]^2)_
+Log[W]^2 (cok0s[[1]]^2+cok0s[[2]]^2+cok0s[[3]]^2))];  

veckak=Normalize[Cross[kakkou[[imax,2]],kakkou[[imax,1]]]];  

inkak=veckak.{0,0,-1};  

vecRotkak=Normalize[Cross[veckak,{0,0,-1}]];  

rotVeckak={{inkak-(-1+inkak) vecRotkak[[1]]^2,_
(1-inkak) vecRotkak[[1]] vecRotkak[[2]]-Sqrt[1-inkak^2] vecRotkak[[3]],_
Sqrt[1-inkak^2] vecRotkak[[2]]-(-1+inkak) vecRotkak[[1]] vecRotkak[[3]]],_
{(1-inkak) vecRotkak[[1]] vecRotkak[[2]]+Sqrt[1-inkak^2] vecRotkak[[3]],_
inkak-(-1+inkak) vecRotkak[[2]]^2,-Sqrt[1-inkak^2] vecRotkak[[1]]_-
(-1+inkak) vecRotkak[[2]] vecRotkak[[3]],{-Sqrt[1-inkak^2] vecRotkak[[2]]_-
(-1+inkak) vecRotkak[[1]] vecRotkak[[3]],Sqrt[1-inkak^2] vecRotkak[[1]]_-
(-1+inkak) vecRotkak[[2]] vecRotkak[[3]],inkak-(-1+inkak) vecRotkak[[3]]^2}}];  

uukak=rotVeckak.(G-cok);  

zzkak=rotVeckak.veckak;  

veccok={{Cos[(2 π Log[1+(Acok Log[W])]/Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)_
+Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2))]/Log[W]] (Log[W] cok[[1]]_-
2 π cok[[2]])-(2 π cok[[1]]+Log[W] cok[[2]]) Sin[(2 π Log[1+(Acok Log[W])_/
Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)+Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2))_/
Log[W]])/(Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)+Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2))],(Cos[(2 π Log[1+(Acok Log[W])]/Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)_
+Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2))]/Log[W]] (2 π cok[[1]]_+
Log[W] cok[[2]])+(Log[W] cok[[1]]-2 π cok[[2]]) Sin[(2 π Log[1+(Acok Log[W])_/
Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)+Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2))_/
Log[W]])/(Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)_+
Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2))],_
(Log[W] cok[[3]])/Sqrt[4 π ^2 (cok[[1]]^2+cok[[2]]^2)]_+
Log[W]^2 (cok[[1]]^2+cok[[2]]^2+cok[[3]]^2));  

inveccok=veccok.{0,0,-1};  

vecRotveccok=Normalize[Cross[veccok,{0,0,-1}]];  

rotVecveccok={{inveccok-(-1+inveccok) vecRotveccok[[1]]^2,_
(1-inveccok) vecRotveccok[[1]] vecRotveccok[[2]]-Sqrt[1-inveccok^2]_*
vecRotveccok[[3]],Sqrt[1-inveccok^2] vecRotveccok[[2]]_-
(-1+inveccok) vecRotveccok[[1]] vecRotveccok[[3]],_
{(1-inveccok) vecRotveccok[[1]] vecRotveccok[[2]]+Sqrt[1-inveccok^2] _*
vecRotveccok[[3]],inveccok-(-1+inveccok) vecRotveccok[[2]]^2,_-
Sqrt[1-inveccok^2] vecRotveccok[[1]]-(-1+inveccok) vecRotveccok[[2]] _*
vecRotveccok[[3]],{-Sqrt[1-inveccok^2] vecRotveccok[[2]]_-
(-1+inveccok) vecRotveccok[[1]] vecRotveccok[[3]],Sqrt[1-inveccok^2] _*
vecRotveccok[[1]]-(-1+inveccok) vecRotveccok[[2]] vecRotveccok[[3]],_
inveccok-(-1+inveccok) vecRotveccok[[3]]^2}}];  

uucok=rotVecveccok.(G-cok);  

zzcok=rotVecveccok.veccok;  

roty0[uu_, φ_]:= {Cos[ φ ],-Sin[ φ ],0},{Sin[ φ ],Cos[ φ ],0},{0,0,1}].uu  

φ y0[{ux_,uy_,uz_}]:=-Sign[uy]*ArcCos[ux/Sqrt[Abs[ux]^2+Abs[uy]^2]]  

φ kak= φ y0[uukak];  

φ cok= φ y0[uucok];  

cont00=Table[{W^( θ /(2 π )) (Cos[ θ ] (1-(2 Dd)/(-1+Dd)+Cos[ Γ ] Cos[ φ ])_
+Sin[ θ ] (-Cos[ Δ ] Cos[ φ ] Sin[ Γ ]+Sin[ Δ ] Sin[ φ ])),_

```

```

W^( θ /(2 π )) (Cos[ Δ ] Cos[ θ ] Cos[ φ ] Sin[ Γ ]+(1-(2 Dd)_
/(-1+Dd)+Cos[ Γ ] Cos[ φ ]) Sin[ θ ]-Cos[ θ ] Sin[ Δ ] Sin[ φ ]),_
W^( θ /(2 π )) (-((2 T)/(-1+Dd))+Cos[ φ ] Sin[ Γ ] Sin[ Δ ]+Cos[ Δ ] Sin[ φ ]),_
{ θ , θ max-4Pi, θ max,Pi/6},{ φ ,0,2Pi,Pi/6}],

cont01=Map[ #-cok&,cont00,{2}];

contkak01=Map[rotY0[rotVeckak. #,φ kak]&,cont01,{2}];

contcok01=Map[rotY0[rotVecveccok. #,φ cok]&,cont01,{2}];

hz[lis_]:=Table[{t,p,lis[[t,p,3]]},{t,1,25,1},{p,1,12,1}];

zsort[ls_]:=Sort[Flatten[ls,1],#1[[3]]<#2[[3]]&];

contkak02=Table[{W^( θ /(2 π )) (Cos[ θ ] (1-(2 Dd)/(-1+Dd)+Cos[ Γ ] Cos[ φ ])__
+Sin[ θ ] (-Cos[ Δ ] Cos[ φ ] Sin[ Γ ]+Sin[ Δ ] Sin[ φ ])),_
W^( θ /(2 π )) (Cos[ Δ ] Cos[ θ ] Cos[ φ ] Sin[ Γ ]+(1-(2 Dd)/(-1+Dd)_
+Cos[ Γ ] Cos[ φ ]) Sin[ θ ]-Cos[ θ ] Sin[ Δ ] Sin[ φ ]),_
W^( θ /(2 π )) (-((2 T)/(-1+Dd))+Cos[ φ ] Sin[ Γ ] Sin[ Δ ]+Cos[ Δ ] Sin[ φ ]),_
{ θ ,( θ max-4Pi)+(zsort[hz[contkak01]][[1,1]]-1)*Pi/6-Pi/12,( θ max-4Pi)_
+(zsort[hz[contkak01]][[1,1]]-1)*Pi/6+Pi/12,Pi/180},_
{ φ ,(zsort[hz[contkak01]][[1,2]]-1)*Pi/6-Pi/12,(zsort[hz[contkak01]][[1,2]]-1)*Pi/6_
+Pi/12,Pi/180}];

contkak03=Map[rotY0[rotVeckak.( #-cok ),φ kak]&,contkak02,{2}];

zhkak12=zsort[contkak03];

Gkak=rotY0[uukak,φ kak]+{0,0,-zhkak12[[1,3]]};

contcok02=contkak02=Table[{W^( θ /(2 π )) (Cos[ θ ] (1-(2 Dd)/(-1+Dd)+Cos[ Γ ] Cos[ φ ])__
+Sin[ θ ] (-Cos[ Δ ] Cos[ φ ] Sin[ Γ ]+Sin[ Δ ] Sin[ φ ])),_
W^( θ /(2 π )) (Cos[ Δ ] Cos[ θ ] Cos[ φ ] Sin[ Γ ]+(1-(2 Dd)_
/(-1+Dd)+Cos[ Γ ] Cos[ φ ]) Sin[ θ ]-Cos[ θ ] Sin[ Δ ] Sin[ φ ]),W^( θ /(2 π ))_
*((-((2 T)/(-1+Dd))+Cos[ φ ] Sin[ Γ ] Sin[ Δ ]+Cos[ Δ ] Sin[ φ ])),_
{ θ ,( θ max-4Pi)+(zsort[hz[contcok01]][[1,1]]-1)*Pi/6-Pi/12,( θ max-4Pi)_
+(zsort[hz[contcok01]][[1,1]]-1)*Pi/6+Pi/12,Pi/180},_
{ φ ,(zsort[hz[contcok01]][[1,2]]-1)*Pi/6-Pi/12,_
(zsort[hz[contcok01]][[1,2]]-1)*Pi/6+Pi/12,Pi/180}];

contcok03=Map[rotY0[rotVecveccok.( #-cok ),φ cok]&,contcok02,{2}];

zhcok12=zsort[contcok03];

Gcok=rotY0[uucok,φ cok]+{0,0,-zhcok12[[1,3]]};

```