

剛体の回轉直立と形の関係

2006年2月16日

福井大学 工学部 物理工学科
13年度入学 14番 木澤 隆之

目次

序章

ゆで卵のような prolate 形や碁石のような oblate 形の剛体を机上において、高速で回転させると立ち上がる。これは、鉛直軸の角速度を減少させる摩擦力の働きによって、重力に逆らって重心が上がり剛体が立つと言う現象です。この現象は 100 年以上前から知られており、特に、1952 年に Braams, Hugenholtz 両氏によって偏芯球について完全な厳密解があることがわかっている。さらに、2002 年に Moffatt, 下村両氏によって任意の軸対称形において、近似的な保存量があるとする理論が発表された。

この理論によると、回転する剛体の運動方程式 (オイラー方程式) に対して I_3 を対称軸, I_1 をそれ以外の軸まわりの慣性モーメント, Ω, n をそれぞれ鉛直軸, 対称軸方向の角速度, θ を鉛直軸と対称軸のなす角度, 重心の高さ h としたとき $I_3 n = I_1 \Omega \cos \theta$ という近似式が得られます。この近似をジャイロスピック近似とよび、剛体の運動方程式を積分する際に用いることによって、ジェレット定数 $J = A \Omega h$ を導出することができます。この定数から鉛直軸の角速度 Ω を減少させる摩擦力の働きによって、 h すなわち重心を上げ、剛体が立ち上がることが分かる。

本卒業研究では、任意の軸対称形にとらわれずさまざまな剛体の形において、いかに直立するか検証して見ることにした。2005 年大森氏のプログラムをもとに、剛体の並進・回転運動を表す微分方程式を 4 次のルンゲクッタ法を用いて解き、上記の近似を用いずに軸対称・非軸対称物体の回転直立現象を再現し、その回転直立時間が軸比にどう依存しているか、数値実験によって調べた。さらにその結果をすでに与えられている数式と比較し、Moffatt-下村理論の検証にあたる。

注)「回転直立」という言葉は『科学』戸田盛和「回転する卵はなぜ直立する」(2002)のタイトルから、我々が造語したものである。

第1章 軸対称物体の回転直立

—Moffatt下村理論の紹介—

この章では文献 [?, ?, ?] をもとに Moffatt と下村による軸対称物体の回転直立の理論の詳細な解説を行う。

1.1 座標系と接触点

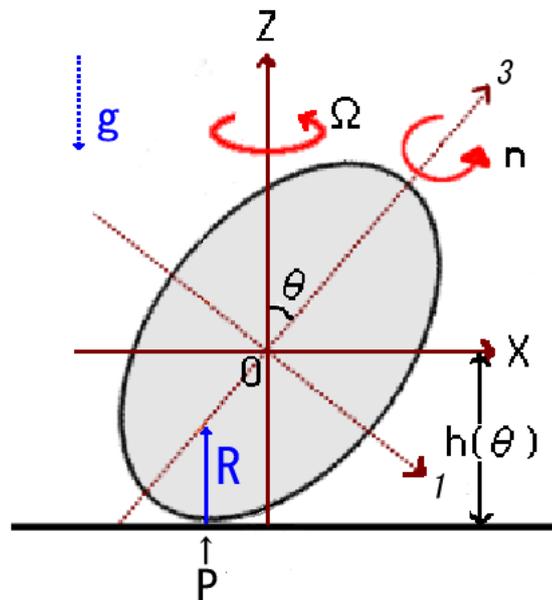


図 1.1: 回転する軸対称物体

点 O を重心とする任意の軸対称物体が、水平面上を滑りながら点 O のまわりに回転運動をしているとする。考えている瞬間における物体と水平面との接触点を P とする。点 P において摩擦力が働かなければ、運動方程式の解として定常的な歳差運動が存在することは、よく知られている [4,5]。そこで、さしあたり定常的な歳差運動が実現していて、したがって、点 O の速度はゼロと見なして、以下の議論を展開する。その上で、非常に弱い摩擦力が働いた場合に、その効果により運動にどのような変化が起きるか、具体的には点 O が上昇するかどうか、を考察することにする。

まず座標軸の取り方を説明する。 X, Y, Z 系は点 O を原点として水平面内で回転する座標系である。 Z 軸は鉛直上向きであり、 X 軸は水平面内に、点 P が XZ 平面にのるようにとる。 Y 軸は X 軸と Z 軸とに垂直で、 X, Y, Z 軸が右手系をなす向きにとる。 X, Y 軸は点 P の運動に追隨して水平面内で回転するが、考えている瞬間におけるその角速度を Ω とする。 $1, 2, 3$ 系は物体に固定された座標系（剛体系）であり、点 O を原点とする。 3 軸は物体の対称軸方向を、 $1, 2$ 軸はそれに垂直な 2 方向を向くようにとる。

1, 2, 3 軸も右手系をなす。形状が軸対称であるため, 1, 2 軸の取り方に任意性はない。ξ, η, ζ 軸は, 物体の角速度ベクトル ω で回転している。X_p = \vec{OP} は O を原点とした接点 P の位置ベクトルとする。このとき物体が受ける力は

- 重力 $Mg = (0, 0, -Mg)$
- 垂直抗力 R
- Y 方向への摩擦力 F

である。摩擦力が Y 方向を向いていることは, 物体の対称軸 (3) 軸の傾き (図中の θ) の変化が, 回転角速度に比べて無視できるほど小さいという仮定の帰結である。

物体の角速度ベクトル ω の各成分は

$$\begin{aligned}\omega_1 &= -\Omega \sin \theta \\ \omega_2 &= \dot{\theta} \\ \omega_3 &= n\end{aligned}\tag{1.1}$$

となる。

また水平面内回転系での各成分は

$$\begin{aligned}\omega_X &= -\omega_1 \cos \theta + \omega_3 \sin \theta \\ &= -\Omega \sin \theta \cos \theta + n \sin \theta \\ &= (n - \Omega \cos \theta) \sin \theta \\ \omega_Y &= \omega_2 = \dot{\theta} \\ \omega_Z &= \omega_3 \cos \theta + \omega_1 \sin \theta \\ &= n \cos \theta + \Omega \sin^2 \theta\end{aligned}\tag{1.2}$$

ただし n は ω の 3 軸成分で

$$n = \dot{\psi} + \Omega \cos \theta$$

である。ただし $\dot{\psi}$ は図 (1.2) に示すような関係にある。

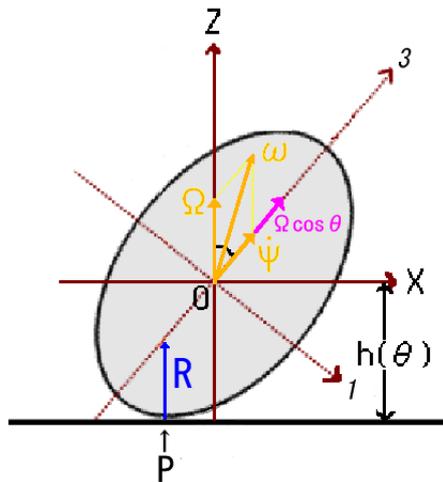


図 1.2: 2つの系での ω と n

1軸2軸の主慣性モーメントを I_1 , 3軸の主慣性モーメントを I_3 とすると, 角運動量 L の各成分は

$$\begin{aligned} L_1 &= -I_1\Omega \sin \theta \\ L_2 &= I_1\dot{\theta} \\ L_3 &= I_3n \end{aligned} \quad (1.3)$$

となる。また

$$\begin{aligned} L_X &= L_1 \cos \theta + L_3 \sin \theta \\ L_Y &= L_2 \\ L_Z &= -L_3 \sin \theta + L_3 \cos \theta \end{aligned} \quad (1.4)$$

(2.3) 式を (2.4) 式に代入して水平面内回転系にすると L は

$$\mathbf{L} = ((I_3n - I_1\Omega \cos \theta) \sin \theta, I_1\dot{\theta}, I_1\Omega \sin^2 \theta + I_3n \cos \theta) \quad (1.5)$$

となる。

この L の時間に対する変化は, オイラー方程式

$$\frac{\partial \mathbf{L}}{\partial t} + (\boldsymbol{\omega} \times \mathbf{L}) = \mathbf{X}_P \times \mathbf{R} + \mathbf{F} \quad (1.6)$$

で与えられる。ここで F は点 P にはたらく摩擦力である。

図 (2.3) は角度 θ が微小角 $\Delta\theta$ 変化したときを描いてある。 X', Z' は変化後の水平面内回転系の座標軸で h' は変化後の重心の高さである。図 (2.3) の点 P 付近を拡大したものが図 (2.4) である。ここで線分 LN に着目すると

$$LN \simeq -X_P \Delta\theta \quad (1.7)$$

であり, また

$$\begin{aligned} LN &= OL - ON \\ &= h - \frac{h'}{\cos(\Delta\theta)} \end{aligned}$$

である。 $\Delta\theta \simeq 0$ のとき, $\cos(\Delta\theta) = 1$ となることから

$$LN \simeq h - h' = -\Delta h \quad (1.8)$$

ゆえに

$$X_P = \frac{dh}{d\theta} \quad (1.9)$$

となる。これを用いると位置ベクトル $\mathbf{X}_P = (X_P, 0, Z_P)$ の成分は

$$X_P = \frac{dh}{d\theta}, \quad Z_P = -h(\theta) \quad (1.10)$$

となる。

また, $\mathbf{R} + \mathbf{F} = (0, F, R)$ であるから

$$\mathbf{X}_P \times (\mathbf{R} + \mathbf{F}) = (-Z_P F, -R X_P, F X_P) \quad (1.11)$$

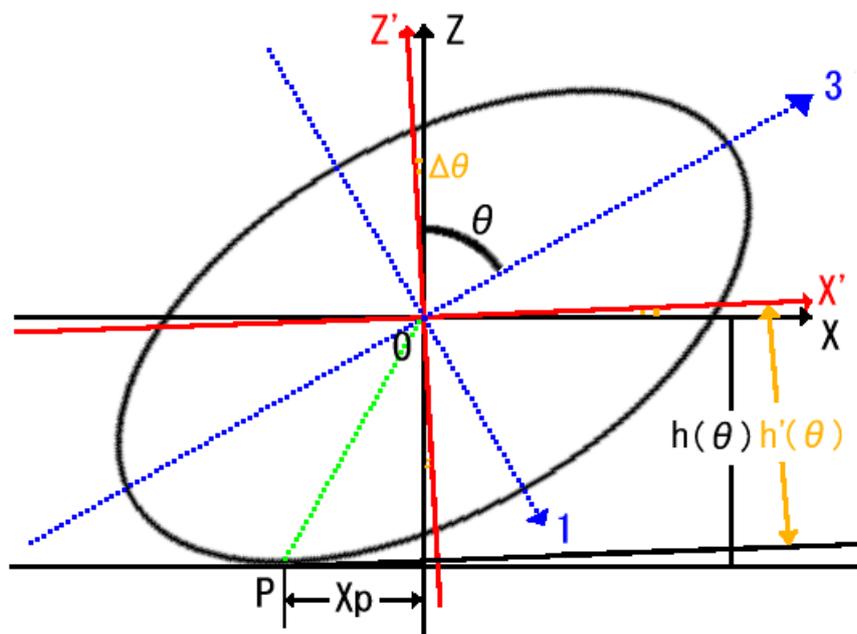


图 1.3: 微小な角度変化 1

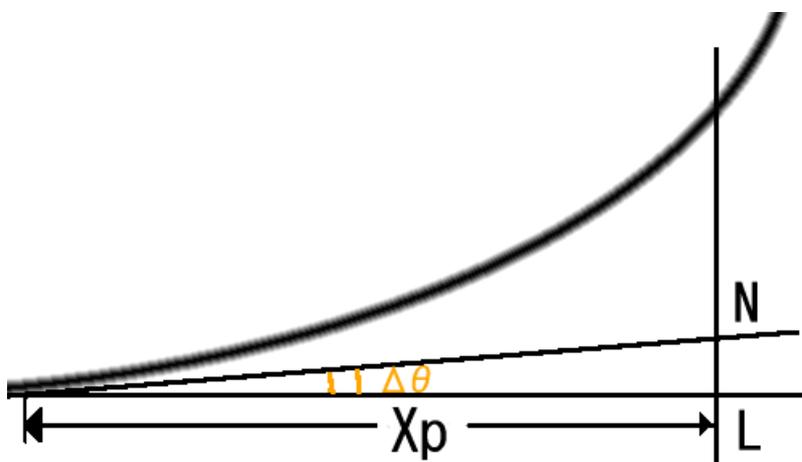


图 1.4: 微小な角度変化 2

となり (2.6) 式を用いるとオイラー方程式の各成分は

$$\frac{d}{dt}[(I_3 n - I_1 \Omega \cos \theta) \sin \theta] - I_1 \Omega \dot{\theta} = -Z_P F \quad (1.12)$$

$$A \ddot{\theta} + \Omega(I_3 n - I_1 \Omega \cos \theta) \sin \theta = -R X_P \quad (1.13)$$

$$A \dot{\Omega} + \frac{d}{dt}[(I_3 n - I_1 \Omega \cos \theta) \cos \theta] = F X_P \quad (1.14)$$

と表される。

1.2 ジェレット定数

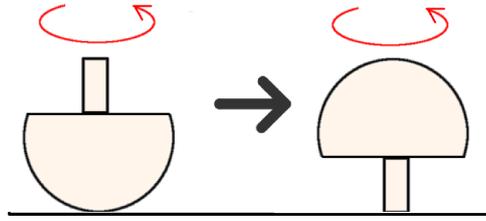


図 1.5: 逆立ちごま

軸対称物体の回転直立の現象に関係のあるものとして、逆立ちごまがある。逆立ちごまとは球の上部を切り取って、棒を付けたもので、偏心球となっている。これを回転させると重心が上昇し、上下逆になり心棒でまわり続ける (図 2.5)。これについては 1952 年に Braams と Hugenholtz によって理論的に解明されている。それによると逆立ちするためには

- ・ こまと接触面との摩擦が不可欠である。
- ・ 接触点がすべて力学的エネルギーが減少しなければならない。

ということが分かっている。このようにエネルギーが保存しない系であるにもかかわらず、ひとつの運動定数が存在する。その運動定数は、角運動量ベクトル L と接触点の位置ベクトル X_P との内積として

$$J = -L \cdot X_P \quad (1.15)$$

で定義される量である。

任意の軸対称物体に対するジェレット定数の時間微分を計算すると

$$\begin{aligned} \dot{J} &= -\dot{L} \cdot X_P - L \cdot \dot{X}_P \\ &= (I_3 n - I_1 \Omega \cos \theta) X_P^2 \frac{d \sin \theta}{dt} \frac{d}{X_P} \end{aligned} \quad (1.16)$$

となる。図 2.6 のように軸対称物体の形が半径 r の球であるとき重心が球の中心から a だけずれている

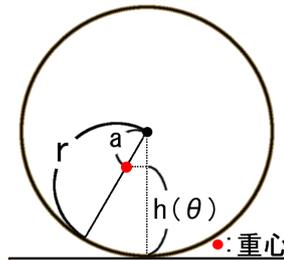


図 1.6: 球の場合の $h(\theta)$

とすると

$$h(\theta) = r - d \cos \theta \quad (1.17)$$

となるので

$$X_P = \frac{dh}{d\theta} = a \sin \theta \quad (1.18)$$

となることから、 $\dot{J} = 0$ となり J は運動定数となる。

しかし、一般の軸対称物体の場合は J は運動定数とはならない。

剛体の回転直立と形の関係

2006年2月16日

福井大学 工学部 物理工学科
13年度入学 14番 木澤 隆之

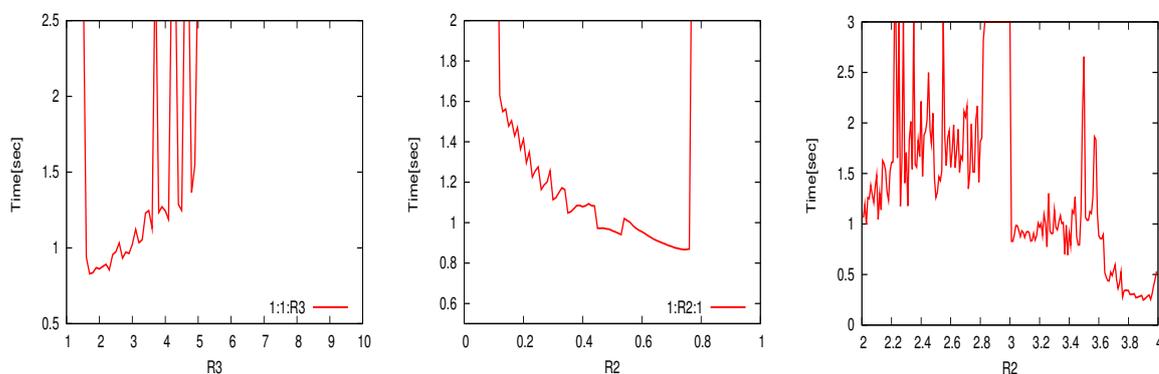
ゆで卵のような prolate 形や碁石のような oblate 形の剛体を机上において、高速で回転させると立ち上がる。これは、鉛直軸の角速度を減少させる摩擦力の働きによって、重力に逆らって重心が上がり剛体が立つと言う現象です。この現象は100年以上前から知られており、特に、1952年に Braams, Hugenholtz 両氏によって偏芯球について完全な厳密解があることがわかっている。さらに、2002年に Moffatt, 下村両氏によって任意の軸対称形において、近似的な保存量があるとする理論が発表された。

この理論によると、回転する剛体の運動方程式(オイラー方程式)に対して I_3 を対称軸, I_1 をそれ以外の軸まわりの慣性モーメント, Ω, n をそれぞれ鉛直軸, 対称軸方向の角速度, θ を鉛直軸と対称軸のなす角度, 重心の高さ h としたとき $I_3 n = I_1 \Omega \cos \theta$ という近似式が得られます。この近似をジャイロスコピック近似とよび、剛体の運動方程式を積分する際に用いることによって、ジェレット定数 $J = A \Omega h$ を導出することができます。この定数から鉛直軸の角速度 Ω を減少させる摩擦力の働きによって、 h すなわち重心を上げ、剛体が立ち上がることが分かる。

本卒業研究では、任意の軸対称形にとらわれずさまざまな剛体の形において、いかに直立するか検証して見ることにした。2005年大森氏のプログラムをもとに、剛体の並進・回転運動を表す微分方程式を4次のルンゲクッタ法を用いて解き、上記の近似を用いずに軸対称・非軸対称物体の回転直立現象を再現し、その回転直立時間が軸比にどう依存しているか、数値実験によって調べた。さらにその結果をすでに与えられている数式と比較し、Moffatt-下村理論の検証にあたる。

数値計算によるシミュレーションを行い、剛体の回転直立と形がどう関係しているかについて調べてみた。その結果、下図のように球に近い剛体は立ち上がるのが遅く、ある軸比で立ち上がる時間が最小となり、またさらに変形すると立ち上がるのが遅くなった。

また、3軸不等の剛体でも立ち上がることが分かった。そして、軸対称形・非軸対象形に問わず、必ず最長軸が回転軸になることが分かった。



1.3 ジャイロスコピック近似

軸対称物体の立ち上がり現象を考えると、回転が遅いと立ち上がらないので回転が速い場合について考える。オイラー方程式の Y 成分は

$$I_1 \ddot{\theta} - I_1 \Omega^2 \cos \theta \sin \theta + I_3 n \Omega \sin \theta = -R X_P \quad (1.19)$$

である。 Ω^2 が大きいとすると、 Ω が含まれている左辺の第 2 項と第 3 項は右辺の $-R X_P$ よりも十分に大きいとすることができる。次に時間スケールに着目すると軸対称物体が立ち上がる時間は回転で決まる短いものではなく摩擦で決まる長いものであるので $|\dot{\theta}| \ll \Omega$ とすることができる。よって (1.19) 式は近似的に

$$I_3 n - I_1 \Omega \cos \theta = \Omega \sin \theta \quad (1.20)$$

とでき、立ち上がるまでの過程すなわち、 $\sin \theta \neq 0$ では

$$I_3 n = I_1 \Omega \cos \theta \quad (1.21)$$

となる。これをジャイロスコピック近似とよぶ。

このジャイロスコピック近似を取り入れると (1.16) 式は $\dot{J} = 0$ となり J は運動定数となる。また (1.5) 式で与えられる角運動量 L は

$$\mathbf{L} = (0, I_1 \dot{\theta} I_1 \Omega) \quad (1.22)$$

となる。したがって (1.15) 式から

$$\begin{aligned} J &= -(-h(\theta) \cdot I_1 \Omega) \\ &= A \Omega h \end{aligned} \quad (1.23)$$

となる。

また、オイラー方程式の X 成分、(1.12) 式は

$$I_1 \Omega \dot{\theta} = F Z_P \quad (1.24)$$

となる。そして Z 成分、(1.14) 式は

$$I_1 \dot{\Omega} = F X_P \quad (1.25)$$

となり (1.24)、(1.25) 式から

$$\frac{\dot{\Omega}}{\Omega} = \frac{X_P}{Z_P} \dot{\theta} = -\frac{\dot{h}}{h} \quad (1.26)$$

が導かれる。この (1.26) 式を積分すると $\Omega h = \text{一定}$ となる。また (1.8) 式、(1.23) 式、(1.24) 式より任意の軸対称物体の θ に対するジャイロスコピック近似を用いた方程式は

$$J \dot{\theta} = -F h^2(\theta) \quad (1.27)$$

という簡単な 1 階の微分方程式となる。

ここで軸対称物体と床との接触点 P での摩擦係数を μ 、すべり速度を V_P としてクーロン摩擦力を仮定すると、摩擦力 F は

$$F = -\mu M g \frac{V_P}{|V_P|} \quad (1.28)$$

で与えられる。

V_P は剛体の角運動量 ω と接触点 P の位置ベクトル X_P との外積で表すことができる。すなわち

$$\begin{aligned} V_P &= \omega \times X_P \\ &= (\Omega \sin^2 \theta + n \cos \theta) \frac{dh}{d\theta} + (n - \Omega \cos \theta) h(\theta) \sin \theta \end{aligned} \quad (1.29)$$

である。 $\beta(\theta) = (\sin^2 \theta + (I_1/I_3) \cos^2 \theta)^{-1/2}$ として (1.21) 式, (1.23) 式をもちいて (1.29) 式の V_P を書き換えると次のようになる。

$$V_P = \frac{J}{I_1} \beta^{-3} h^{-1} \frac{d(\beta h)}{d\theta} \quad (1.30)$$

F と V_P の関係は分かっているものとし, $h(\theta)$ も幾何学的な考察から分かっているとすると (2.27) 式を積分することで θ は t の関数として求まる。

1.4 回転楕円体の場合

ここでは密度が一樣な回転楕円体の場合に (1.25) 式を積分してみる。回転楕円体の対称軸方向の半径を a , それと垂直な方向の半径を b , 質量を M , 密度を ρ とする。この回転楕円体は

$$a^2(x^2 + y^2) + b^2z^2 = a^2b^2$$

と表すことができる。このときの対称軸 (ζ 軸) まわりの慣性モーメントは

$$\begin{aligned} I_3 &= \int_V \rho(\xi^2 + \eta^2) d\xi d\eta d\zeta \\ &= 2\rho \int d\eta \int d\zeta \int \xi^2 d\xi \\ &= 2\rho \int_0^b r^2 dr \int_0^\pi \sin\theta d\theta \int_0^{2\pi} d\phi (r \sin\theta \cos\phi)^2 \\ &= 2\rho \int_0^b r^4 dr \int_0^\pi \sin^3\theta d\theta \int_0^{2\pi} \cos^2\phi d\phi \\ &= 2Mb^2/5 \end{aligned} \tag{1.31}$$

同様にして

$$I_1 = I_2 = M(a+b)^2/5 \tag{1.32}$$

となる。楕円体上の点 $(x_0, 0, z_0)$ において

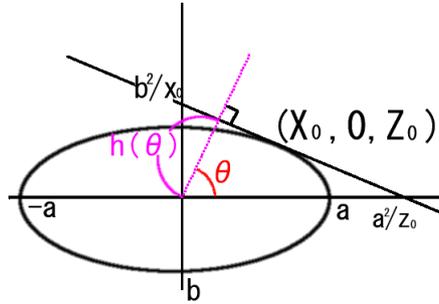


図 1.7: 重心の高さ $h(\theta)$

$$\left(\frac{z_0}{a}\right)^2 + \left(\frac{x_0}{b}\right)^2 = 1 \tag{1.33}$$

であるが、両辺を微分すると

$$\frac{z}{a^2} dz + \frac{x}{b^2} dx = 0 \tag{1.34}$$

となるから点 $(x_0, 0, z_0)$ における接線の式は

$$\begin{aligned} \frac{z_0}{a^2}(z - z_0) + \frac{x_0}{b^2}(x - x_0) &= 0 \\ \Rightarrow \frac{z_0}{a^2}z + \frac{x_0}{b^2}x &= \left(\frac{z_0}{a}\right)^2 + \left(\frac{x_0}{b}\right)^2 = 1 \end{aligned} \tag{1.35}$$

であり、この接線と原点 O との距離が h であるから図 1.7 からわかるように

$$h(\theta) = \frac{a^2}{z_0} \cos\theta = \frac{b^2}{x_0} \sin\theta \tag{1.36}$$

である。(2.33) 式と (2.36) 式から

$$h(\theta) = \sqrt{a^2 \cos^2 \theta + b^2 \sin^2 \theta} \quad (1.37)$$

となる。このときの P 点すべり速度は

$$V_P = - \left(\frac{J}{4Ah^2} (a^2 - b^2) \sin 2\theta \right) \quad (1.38)$$

と簡単になり，クーロン摩擦力を仮定する式を積分すると

$$\tan \theta = (a/b) \tan \mu q(t - t_0) \quad (1.39)$$

となる。ただし $q = Mgab(a - b)/|a - b||J|$ であり， t_0 は積分定数である。

1.5 Moffatt - 下村理論のまとめと更なる課題

Moffatt-下村理論から任意の軸対称形において回転直立することは理解できる。しかし、Moffatt-下村理論の仮定するジャイロスコピック近似は必ずしも良い精度で成り立つ保証はない。そこで本研究では、直立に要する時間が軸比にどう依存するか、またそれが卵形のような prolate 形と碁石のような oblate 形でどのように違うかを数値シミュレーションで調べ、その結果を (1.39) 式と比較することで Moffatt-下村理論の検証をしたいと考える。さらに、軸対称性を破ると何が起こるかについても非常に興味をもつところである。以上の点について更なる研究が必要でないかと感じた。

第2章 数値シミュレーションによる剛体の回転 直立と形の関係

2.1 剛体の形状

剛体の形状は楕円体とする。慣性主軸系での座標を (ξ, η, ζ) とすると表面方程式は

$$\left(\frac{\xi}{R_1}\right)^2 + \left(\frac{\eta}{R_2}\right)^2 + \left(\frac{\zeta}{R_3}\right)^2 = 1 \quad (2.1)$$

である。

体積 $V = \frac{4\pi}{3} R_a R_b R_c$ が半径 R_{00} の球の体積に等しくなるように R_a, R_b, R_c を決めることにする。

このようにして主軸方向の半径 R_0 を決めるとすると

$$R_0 = \frac{R_{00}}{(R'_1 R'_2 R'_3)^{1/3}} \quad (2.2)$$

$$\begin{cases} R_1 = R_0 R'_1 \\ R_2 = R_0 R'_2 \\ R_3 = R_0 R'_3 \end{cases} \quad (2.3)$$

と決めることで実現される。

実際このとき体積は

$$V = \frac{4\pi}{3} R_1 R_2 R_3 = \frac{4\pi}{3} R_0^3 R'_1 R'_2 R'_3 = \frac{4\pi}{3} R_{00}^3 \quad (2.4)$$

となる。

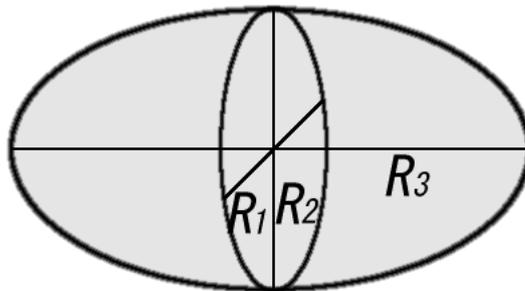


図 2.1: $R_a : R_b : R_c$ の剛体

2.2 初期条件

図 2.2 の様に, x, y, z 成分の L 系と 1, 2, 3 成分の B 系の座標軸をとり, 3 軸と xy 面のなす角を θ とおくと, B 系の単位方向ベクトル e_1, e_2, e_3 は L 系で表すと,

$$(\vec{e}_1)_x = 1.0 \quad (\vec{e}_1)_y = 0.0 \quad (\vec{e}_1)_z = 0.0 \quad (2.5)$$

$$(\vec{e}_2)_x = 0.0 \quad (\vec{e}_2)_y = \cos \theta \quad (\vec{e}_2)_z = -\sin \theta \quad (2.6)$$

$$(\vec{e}_3)_x = 0.0 \quad (\vec{e}_3)_y = \sin \theta \quad (\vec{e}_3)_z = \cos \theta \quad (2.7)$$

となる。

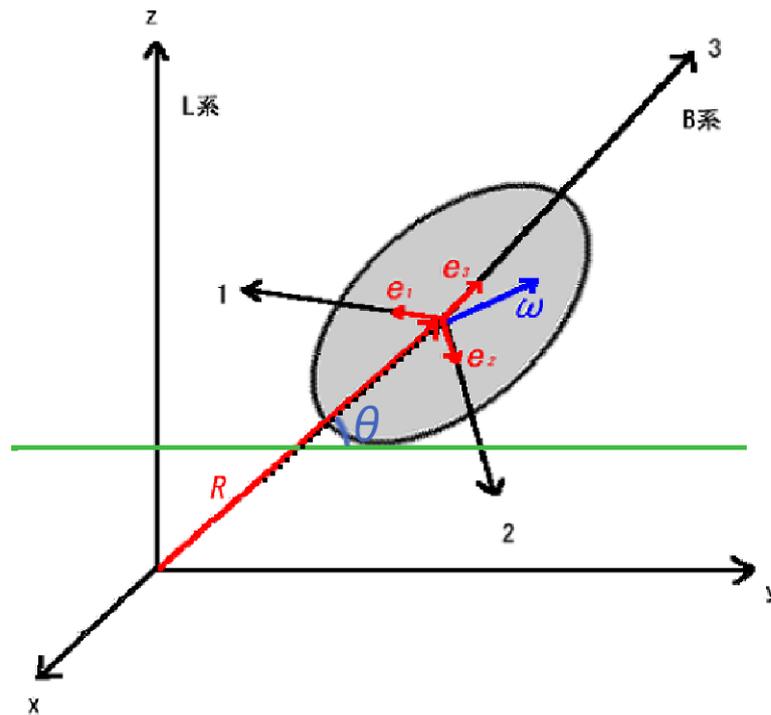


図 2.2: e_1, e_2, e_3 の剛体

2.3 prolate 形の直立時間の軸比依存性

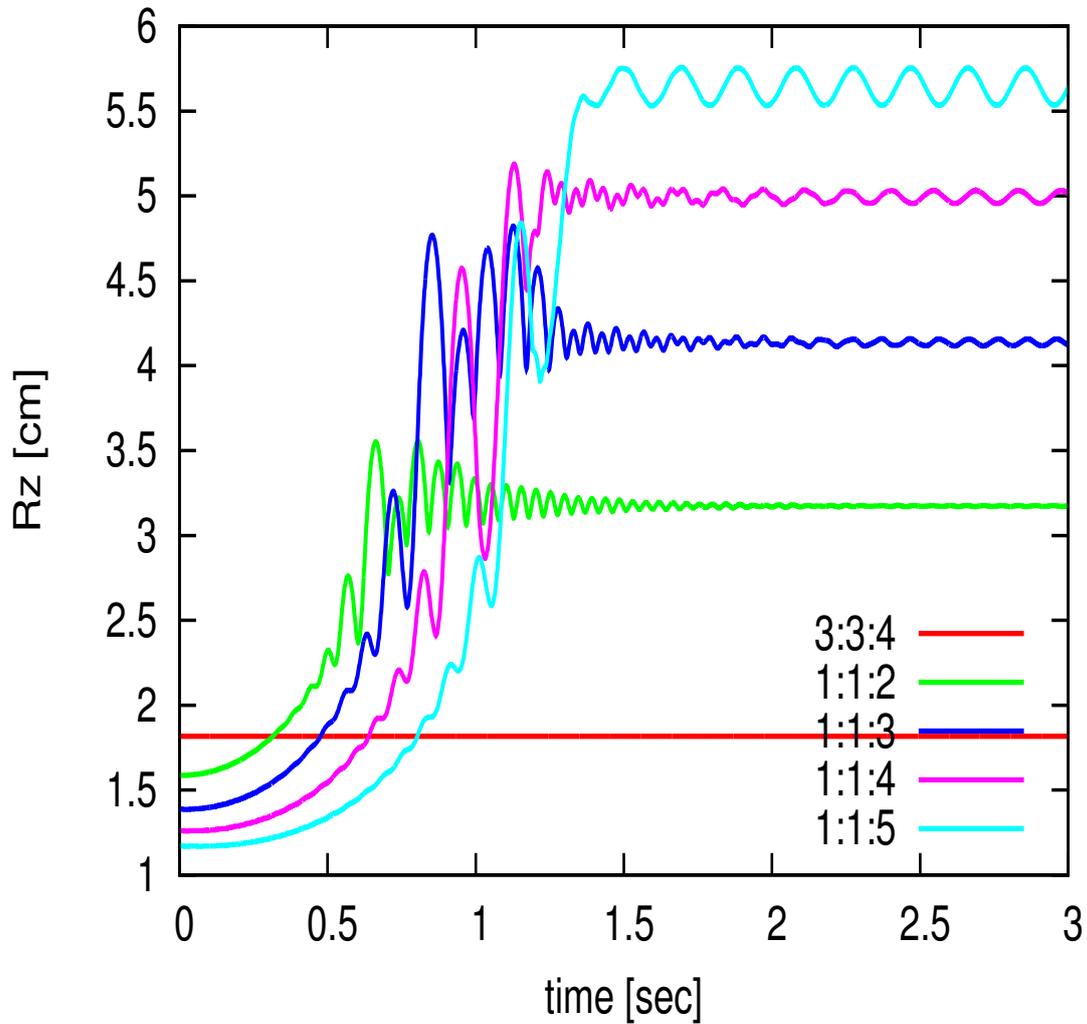


図 2.3: 軸比による重心の高さの時間変化 (1)

まず、さまざまな軸比における重心座標の Z 成分を調べた。図 2.3 の横軸は時刻 [sec], 縦軸は重心の高さ [cm] の値である。

この時の諸条件は以下のとおりである。

- ・ 剛体の形状 楕円体
- ・ 半径 $R_{00} = 2.0[\text{cm}]$ とし $R_1 : R_2 : R_3$ を与え (2.2) 式, (2.3) 式をみたす値
- ・ 楕円体の質量 $m = V\rho[\text{g}], \quad \rho = 1[\text{g}/\text{cm}^3]$
- ・ 初期回転角速度 $100\pi [\text{rad}/\text{s}]$
- ・ 摩擦係数 $\mu = 0.5 [\text{cm}/\text{s}^2]$
- ・ ステップ幅 $\Delta t = 0.0001[\text{sec}]$
- ・ 3 軸の水平方向とのなす角度の初期値 $0.0122[\text{rad}]$

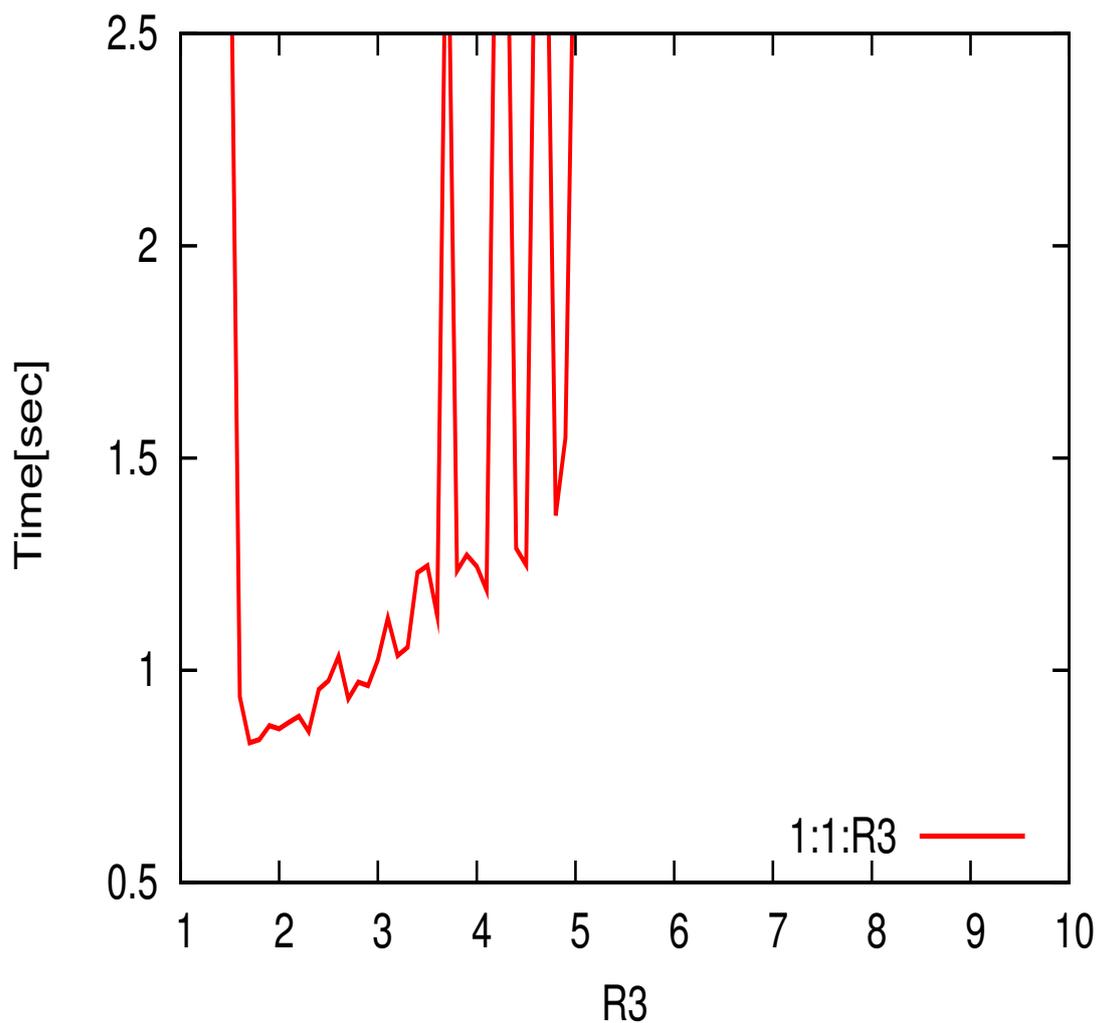


図 2.4: 剛体の立ち上がりと軸比の関係 (1)

まず、prolate 形では、すべてにおいて最長軸である 3 軸が回転軸となった。図 2.4 は、軸比の変化による立ち上がりの時間を示したものである。ただし、 $\theta = 5^\circ$ のときに立ち上がったとした。横軸は 3 軸の軸比 $1:1:R_3$ であり、縦軸は時刻 [sec] の値である。まず、この図から、剛体が立ち上がるまでの時間は、球に近い形では立ちあがるまでに時間がかかり、 $1:1:1.6$ あたりで立ち上がるまでの時間が最小になり、以後徐々に遅くなることが分かった。グラフが直線的にでないのは、剛体がジャンプすることが原因だと思われる。この結果を (1.39) 式と比較することは、今後の課題として残しておく。

2.4 oblate 形の直立時間の軸比の依存性

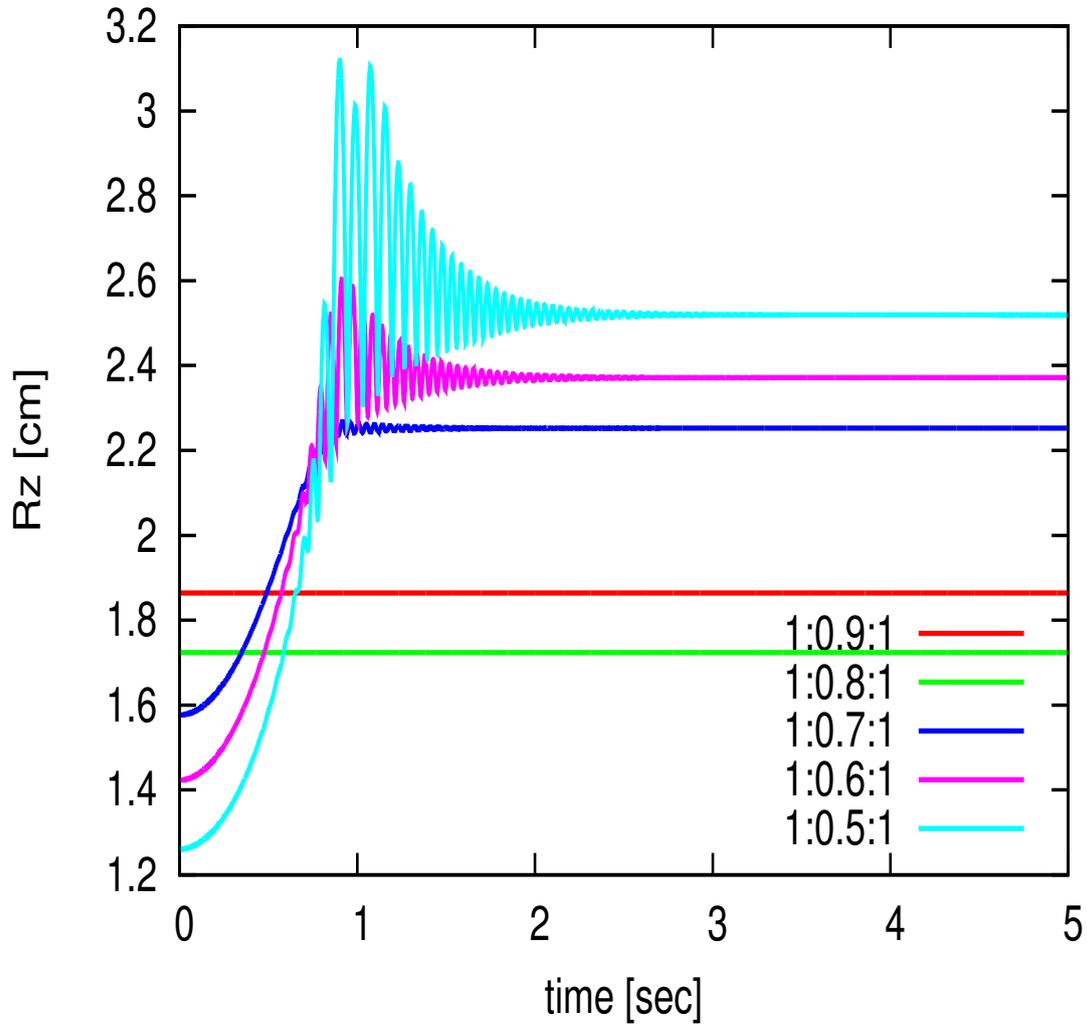


図 2.5: 軸比による重心の高さの時間変化 (2)

次に、さまざまな軸比における重心座標の Z 成分を調べた。図 2.5 の横軸は時刻 [sec], 縦軸は重心の高さ [cm] の値である。

この時の諸条件は以下のとおりである。

- ・ 剛体の形状 楕円体
- ・ 半径 $R_{00} = 2.0[\text{cm}]$ とし $R_1 : R_2 : R_3$ を与え (2.2) 式, (2.3) 式をみたす値
- ・ 楕円体の質量 $m = V\rho[\text{g}], \quad \rho = 1[\text{g}/\text{cm}^3]$
- ・ 初期回転角速度 $100\pi [\text{rad}/\text{s}]$
- ・ 摩擦係数 $\mu = 0.5 [\text{cm}/\text{s}^2]$
- ・ ステップ幅 $\Delta t = 0.0001[\text{sec}]$
- ・ 3 軸の水平方向とのなす角度の初期値 $0.0122[\text{rad}]$

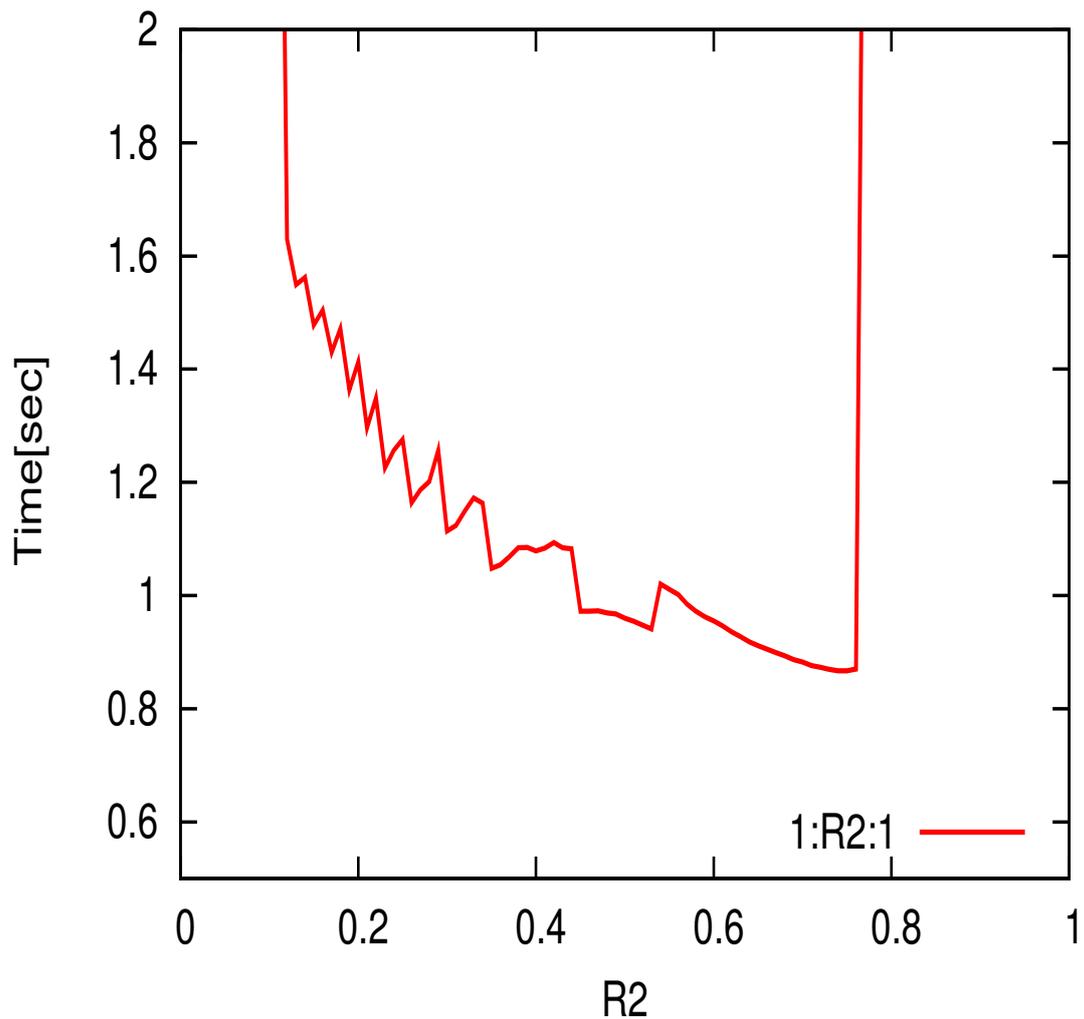


図 2.6: 剛体の立ち上がりと軸比の関係 (2)

次に oblate 形においては, すべてにおいて最長軸である 1 軸, もしくは 3 軸が回転軸となった。図 2.6 は軸比の変化による立ち上がりの時間を示したものである。ただし, $\theta = 5^\circ$ のときに立ち上がったとした。横軸は 2 軸の軸比 $1:R_2:1$ であり, 縦軸は時刻 [sec] の値である。この図から, 剛体が立ち上がるまでの時間は, 球に近い形では立ちあがるまでに時間がかかり, $1:0.75:1$ 付近で立ち上がるまでの時間が最小となり, 以後徐々に遅くなることが分かった。これもグラフが直線的にでないのは, 剛体がジャンプすることが原因だと思われる。この結果を (1.39) 式と比較することは, 今後の課題として残しておく。

2.5 非軸対称形の直立時間の軸比依存性

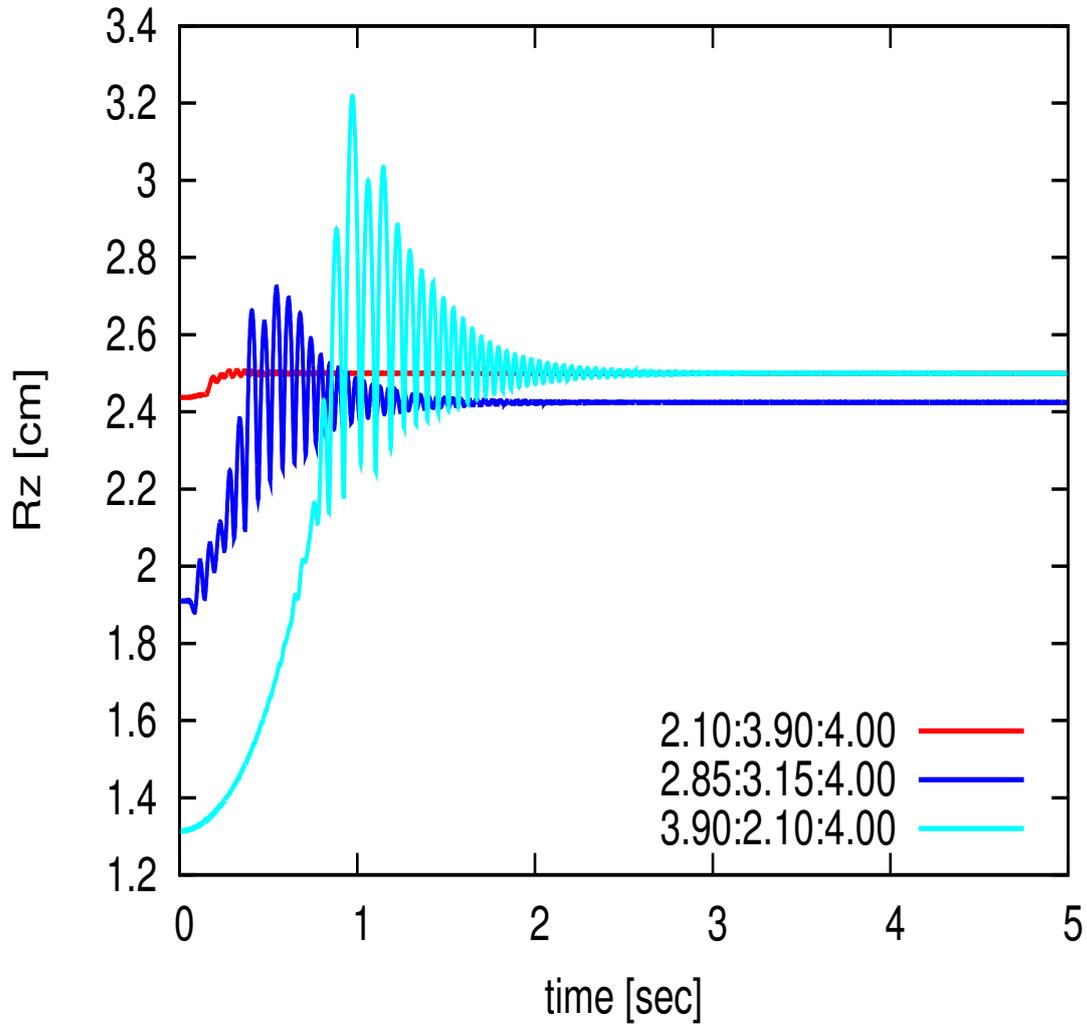


図 2.7: 軸比による重心の高さの時間変化 (3)

次に、三軸不等の軸比における重心座標の Z 成分を調べた。図 2.7 の横軸は時刻 [sec]、縦軸は重心の高さ [cm] の値である。

この時の諸条件は以下のとおりである。

- ・ 剛体の形状 三軸不等の剛体
- ・ 半径 $R_{00} = 2.0[\text{cm}]$ とし $R_1 : R_2 : R_3$ を与え (2.2) 式, (2.3) 式をみたす値
- ・ 剛体の質量 $m = V\rho[\text{g}]$, $\rho = 1[\text{g}/\text{cm}^3]$
- ・ 初期回転角速度 100π [rad/s]
- ・ 摩擦係数 $\mu = 0.5$ [cm/s²]
- ・ ステップ幅 $\Delta t = 0.0001[\text{sec}]$
- ・ 3 軸の水平方向とのなす角度の初期値 $0.0122[\text{rad}]$

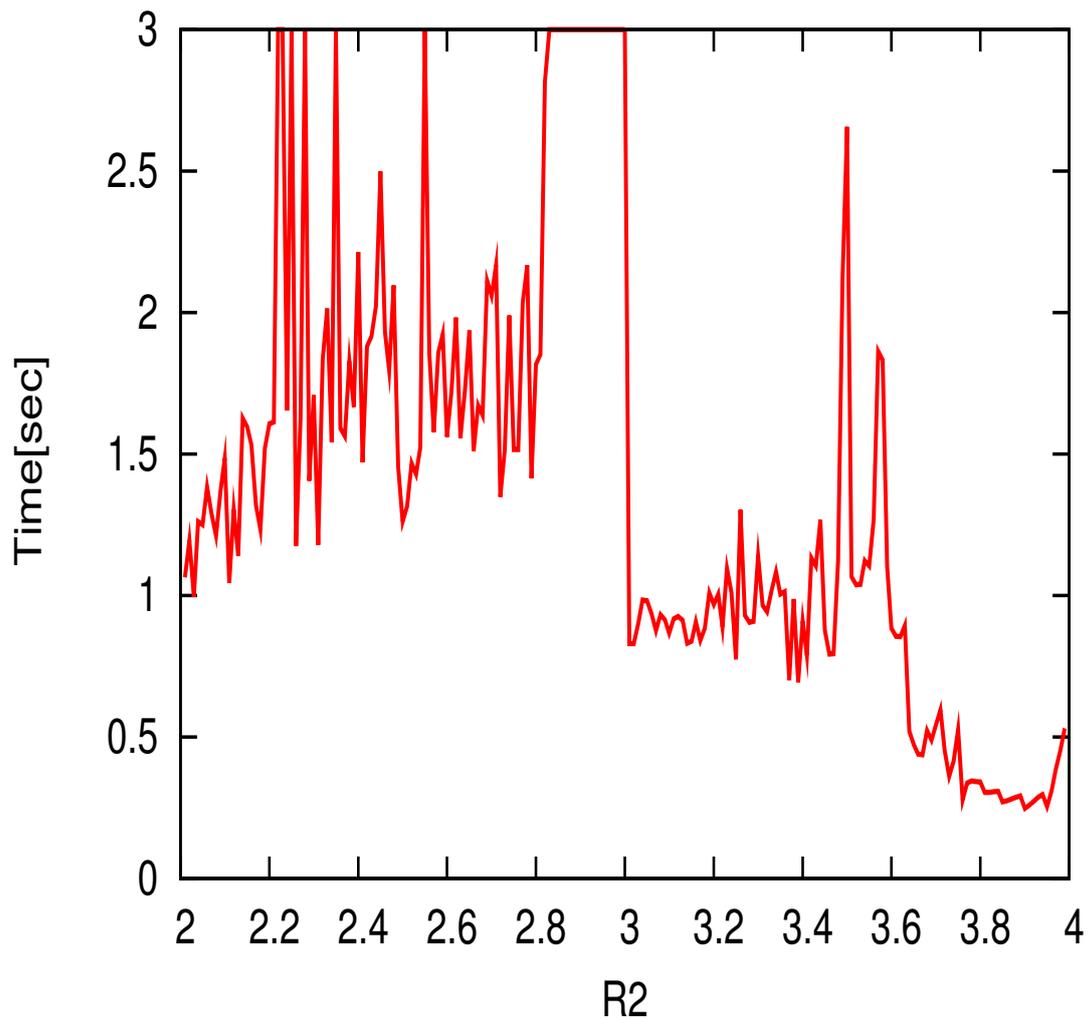


図 2.8: 剛体の立ち上がり時間と軸比の関係 (3)

三軸不等においても、すべてにおいて最長軸である3軸が回転軸となった。図 2.8 軸比の変化による立ち上がり時間を示したものである。ただし、 $\theta = 5^\circ$ のときに立ち上がったとした。横軸は軸比を $6-R_2:R_2:4$ としたときの R_2 、縦軸は時刻 [sec] の値である。この図から、剛体が立ち上がるまでの時間は、球に近い形では立ちあがるまでに時間がかかり、2軸の軸比が大きいほうが立ち上がるまでの時間が短いことが分かる。これもグラフが直線的にでないのは、剛体がジャンプすることが原因だと思われる。この結果を (1.39) 式と比較することは、今後の課題として残しておく。

第3章 結論

数値計算によるシミュレーションを行い、剛体の回転直立と形がどう関係しているかについて調べてみた。その結果、球に近い剛体は立ち上がるのが遅く、ある軸比で立ち上がる時間が最小となり、またさらに変形すると立ち上がるのが遅くなった。

また、3軸不等の剛体でも立ち上がるのが分かった。そして、軸対称形・非軸対称形に問わず、必ず最長軸が回転軸になることが分かった。

参考文献

- [1] H.K.Moffatt and Y.Shimomura: *Spinning eggs - a paradox resolved*,
Nature **416**,385-386(2002).
- [2] 下村裕:「立ち上がる回転ゆで卵の解」パリテイ,
18,52-56(2003).
- [3] 戸田盛和:「回転する卵はなぜ直立する」科学(岩波書店),
72,932-939(2002).
- [4] 小井出昭一郎:「力学」岩波書店,
pp.90-165(1987).
- [5] 戸田盛和:「力学」岩波書店,
pp.144-188(1982).
- [6] 村井興平:「卵を回すとなぜ立つか」,
福井大学工学部 物理工学科 卒業研究(2004).
- [7] 大森英胤:「卵を回すとなぜ立つか2」,
福井大学工学部 物理工学科 卒業研究(2005).

謝辞

本論文を作成するにあたり、田嶋直樹先生には終始御厚いご指導を賜りましたことを感謝し、御礼申し上げます。また鈴木敏男先生、林明久先生にも、多方面にわたり御指導して頂いたことを併せて御礼申し上げます。

本研究に対してご意見を頂いた、多くの物理工学科先生方ならびに支えてくださったみなさまにはお詫びとともに、感謝の気持ちを捧げ、謝辞の言葉とさせていただきます。

付録 Program List

プログラムは昨年度の大森英胤氏の卒業研究 [7] に使用したものをもとに以下の改良を加えた。図の部分は、軸比を変化させるために (1)prolate 形 (2)oblate 形 (3) 比軸対称 として付け加えた。図は剛体の立ち上がった状態を仮定するために (1)prolate 形 (2)oblate 形 (3) 比軸対称 として改良した。

図の部分の説明

- (1)prolate 形については R_3 軸の比が 1.1 ~ 10 まで値が変化できるように改良した。
- (2)oblate 形については R_2 軸の比が 0.01 ~ 0.99 まで値が変化できるように改良した。
- (3) 比軸対称形については R_1 軸の比が 3.01 ~ 3.99 R_2 軸の比が 2.01 ~ 2.99 まで値が変化できるように改良した。

図の部分の説明

実験室系 (x, y, z) から見た重心系 (a, b, c) の単位ベクトルを次のように定義すると

$$\begin{aligned}ea &= (1, 0, 0) \\eb &= (0, \cos \theta, -\sin \theta) \\ec &= (0, \sin \theta, \cos \theta)\end{aligned}$$

- (1)prolate 形については $ec_z = \cos \theta = 0.9962$ つまり $\theta = 5^\circ$ で立ち上がると仮定した。
- (2)oblate 形については $eb_z = -\sin \theta = -0.0872$ つまり $\theta = 5^\circ$ で立ち上がると仮定した。
- (3) 非軸対称形については $ec_z = \cos \theta = -0.9962$ つまり $\theta = 5^\circ$ で立ち上がると仮定した。

```
/*
kaiten7(c&d&e).c
*/

#include <stdio.h>
#include <math.h>

/*method = 0 : Euler method, 1 : 4th-order Runge-Kutta method */
#define METHOD 1

typedef struct { double x; double y; double z; } Lvec; /* vector in L-frame */
typedef struct { double a; double b; double c; } Bvec; /* vector in B-frame */
typedef struct { double x; double y; double z; double a; double b; double c; }
    LBvec; /* vector whose components both in L- and B-frames are necessary */

int forces(Lvec *rc, Lvec *vc, Bvec *omg, Lvec *ea, Lvec *eb, Lvec *ec,
           Lvec *frc, LBvec *trq);
int normal_point(Bvec *n, Bvec *tp);
/*void init1(Lvec *ea, Lvec *eb, Lvec *ec, Bvec *omg0, Lvec *rc, Lvec *vc);*/
void init2(Lvec *ea, Lvec *eb, Lvec *ec, Bvec *omg0, Lvec *rc, Lvec *vc);
const double pi = 3.141592653589793;

main(){
    kaiten5();
}

Bvec R; /* radius in the principal axes */
double mass; /* total mass of the rigid body */
```

```

double volume; /* volume of the rigid body */
const double g_gravity = 980.0; /* gravitational acceleration [cm/s^2] */

int kaiten5(){

    Bvec radius_ratio ; /* relative lengths of principal axes */
    double R00=2.0 ;      /* radius for shperical shape [cm] */
    double density=1;     /* density [g/cm^3] */

    Lvec ea /*= { 0.0, 0.0,-1.0}*/ ; /* unit vector for a-axis of B-frame */
    Lvec eb /*= {-1.0, 0.0, 0.0}*/ ; /* unit vector for b-axis of B-frame */
    Lvec ec /*= { 0.0, 1.0, 0.0}*/ ; /* unit vector for c-axis of B-frame */
    double dt;           /* time step size [sec] */
    /*double max_t=30.0;*/
    double max_itime=300000; /* the number of time steps */
    int mprint=100;     /* period to print the mechanical state */

    Bvec moi;          /* moment of inertia */
    Bvec omg0;        /* initial angular velocity vector, B-frame */
    Bvec omg;         /* angular velocity vector, B-frame */
    double omgsize;
    double omg_gosa;
    Bvec omgm;        /* for Runge Kutta */
    double R0 ;
    double t,fct;
    double minimum_height ;
    double eng_tot, eng_rot, eng_tra, eng_gra;
    Lvec rc ; /* center of mass coordinate, =(0,0,0) in B-frame */
    Lvec vc ; /* velocity of center of mass, =(0,0,0) in B-frame */
    Bvec tp ; /* tangential point */
    Lvec dtp ; /* displacement from center of mass to tangential point */
    Lvec vtp ; /* velocity of tangential point */
    Lvec evtp ; /* unit vector parallel to velocity of tangential point */
    Lvec frc ; /* total force */
    LBvec trq ; /* Toruque as for center of mass */
    double d_eng_tot;
    double d_omgsize;
    double f1,f2,f3,fn1,fn2,fn3;
    Lvec eam, ebm, ecm; /* for Runge Kutta */
    Lvec rcm,vcm; /* for Runge Kutta */
    double e1dx,e1dy,e1dz, e2dx,e2dy,e2dz, e3dx,e3dy,e3dz;
    double kx,ky,kz, kax,kay,kaz, kbx,kby,kbz, kcx,kcy,kcz;
    double krx,kry,krz, kvx,kvy,kvz;
    double k1x ,k1y ,k1z , k2x ,k2y ,k2z , k3x ,k3y ,k3z , k4x ,k4y ,k4z ;
    double k1ax,k1ay,k1az, k2ax,k2ay,k2az, k3ax,k3ay,k3az, k4ax,k4ay,k4az;
    double k1bx,k1by,k1bz, k2bx,k2by,k2bz, k3bx,k3by,k3bz, k4bx,k4by,k4bz;
    double k1cx,k1cy,k1cz, k2cx,k2cy,k2cz, k3cx,k3cy,k3cz, k4cx,k4cy,k4cz;
    double k1rx,k1ry,k1rz, k2rx,k2ry,k2rz, k3rx,k3ry,k3rz, k4rx,k4ry,k4rz;
    double k1vx,k1vy,k1vz, k2vx,k2vy,k2vz, k3vx,k3vy,k3vz, k4vx,k4vy,k4vz;
    int itime; /* counter of time step, taking 0..max_itime */
    double Jellet; /* Jellet constant */
    Bvec mez;
    FILE *log_res1; /* output for detailed graphs */
    FILE *log_res2; /* output for detailed graphs */
    FILE *log_chk1; /* output for check */
    FILE *log_res3;
    FILE *log_res4;

```

```

FILE *log_res5;
FILE *log_res6;
FILE *log_res7;
Lvec angmom; /* angular momentum in L-frame */
double angmomsize;
double F1,F2,F3;
int k;
/* log_res1 = NULL;*/ log_res1=fopen("kaiten.g1","w");
/* log_res2 = NULL;*/ log_res2=fopen("kaiten.g2","w");
    log_chk1 = NULL; /*log_chk1=fopen("kaiten.chk","w");*/
/* log_res3 = NULL; */ log_res3=fopen("kaiten.g3","w");
/* log_res4 = NULL; */ log_res4=fopen("kaiten.g4","w");
/* log_res7 = NULL */ log_res7=fopen("kaiten.g7","w");

```

```

fprintf(stderr,"check : pi = %20.16f\n",pi);

```

```

(1) for(k=1;k<=90;k++){
    radius_ratio.c=1+k*0.1;
    radius_ratio.a=1.0;
    radius_ratio.b=1.0;

(2) for(k=1;k<=99;k++){
    radius_ratio.b=k*0.01;
    radius_ratio.a=1.0;
    radius_ratio.c=1.0;

(3) for(k=1;k<=99;k++){
    radius_ratio.a=3.0+(0.01*k);
    radius_ratio.b=3.0-(0.01*k);
    radius_ratio.c=4.0;

```

A

```

fprintf(stderr,"radius ratio = 1:1:?" ,radius_ratio.a,radius_ratio.b,radius_ratio.c);

```

```

R0=R00/pow(radius_ratio.a*radius_ratio.b*radius_ratio.c,1.0/3.0);
R.a=R0*radius_ratio.a; R.b=R0*radius_ratio.b; R.c=R0*radius_ratio.c;
if(R.a < R.b) minimum_height=R.a; else minimum_height = R.b;
if(R.c < minimum_height) minimum_height = R.c;
fprintf(stderr,"R=(%f %f %f) min=%f\n",R.a,R.b,R.c,minimum_height);

```

```

volume=4*pi*R.a*R.b*R.c/3; /* volume of the rigid body [cm^3] */
mass=density*volume;
fprintf(stderr,"volume=%f density=%f mass=%f\n",volume,density,mass);
moi.a=mass*(R.b*R.b+R.c*R.c)/5; /* moment inertia [g cm^2] */
moi.b=mass*(R.c*R.c+R.a*R.a)/5;
moi.c=mass*(R.a*R.a+R.b*R.b)/5;
fprintf(stderr,"MOI=(%f %f %f)\n",moi.a,moi.b,moi.c);

```

```

f1=(moi.b-moi.c)/moi.a; f2=(moi.c-moi.a)/moi.b; f3=(moi.a-moi.b)/moi.c;
fn1=1/moi.a; fn2=1/moi.b; fn3=1/moi.c;

dt=1.0e-6;
// fprintf(stderr,"input dt=");
// scanf("%lf",&dt);
fprintf(stderr,"check dt=%e\n",dt);

max_itime=3/dt;
mprint=0.001/dt; if(mprint<1) mprint++;
fprintf(stderr,"max_itime=%f mprint=%d\n",max_itime,mprint);

/*init1(&ea, &eb, &ec, &omg0, &rc ,&vc);*/
init2(&ea, &eb, &ec, &omg0, &rc ,&vc);
fprintf(stderr,"rc.z=%f\n",rc.z);

fprintf(stderr,"%s\n ec - (ea x eb) =(e %e %e)\n"
, "check of right-handedness of vectors {ea,eb,ec}:"
,ec.x - (ea.y*eb.z-ea.z*eb.y)
,ec.y - (ea.z*eb.x-ea.x*eb.z)
,ec.z - (ea.x*eb.y-ea.y*eb.x));

omg.a=omg0.a; omg.b=omg0.b; omg.c=omg0.c;

for(itime=0;;itime++){ t=itime*dt;

angmom.x=moi.a*omg.a*ea.x + moi.b*omg.b*eb.x + moi.c*omg.c*ec.x ;
angmom.y=moi.a*omg.a*ea.y + moi.b*omg.b*eb.y + moi.c*omg.c*ec.y ;
angmom.z=moi.a*omg.a*ea.z + moi.b*omg.b*eb.z + moi.c*omg.c*ec.z ;
angmomsize=sqrt(angmom.x*angmom.x + angmom.y*angmom.y + angmom.z*angmom.z);
omgsize = sqrt(omg.a*omg.a + omg.b*omg.b + omg.c*omg.c);

eng_rot = (moi.a*omg.a*omg.a + moi.b*omg.b*omg.b + moi.c*omg.c*omg.c)*0.5 ;
eng_tra = (0.5*mass)*(vc.x*vc.x + vc.y*vc.y + vc.z*vc.z) ;
eng_gra = (mass*g_gravity)*(rc.z-minimum_height);
eng_tot = eng_rot + eng_tra + eng_gra;
if(itime==0) fprintf(stderr,"eng_tot=%f\n",eng_tot);

if(log_res1 != NULL && itime % mprint == 0)
fprintf(log_res1,"%f %f %f\n"
,t,omg.a,omg.b,omg.c,ea.x,ea.y,ea.z,eb.x,eb.y,eb.z,ec.x,ec.y,ec.z
,rc.x,rc.y,rc.z,vc.x,vc.y,vc.z);

d_eng_tot=eng_tot-2841751.291223551612347364;

```

```

(1)  if(fabs(ec.z) > 0.8 || itime >= max_itime) {
      fprintf(log_res7,"%f %f\n", radius_ratio.c,t);
      if(fabs(ec.z) > 0.8 ) {printf("OK ");}
      else {printf("NG ");}
      fprintf(stderr,"%f %f\n", radius_ratio.c,t);

```

```

(2)  if(fabs(eb.z) < 0.1 || itime >= max_itime) {
      fprintf(log_res7,"%f %f\n", radius_ratio.b,t);
      if(fabs(eb.z) < 0.1 ) {printf("OK ");}
      else {printf("NG ");}
      fprintf(stderr,"%f %f\n", radius_ratio.b,t);

```

```

(3)  if(fabs(ec.z) < (-0.8) || itime >= max_itime) {
      fprintf(log_res7,"%f %f\n", radius_ratio.b,t);
      if(fabs(ec.z) < (-0.8) ) {printf("OK ");}
      else {printf("NG ");}
      fprintf(stderr,"%f %f\n", radius_ratio.b,t);

```

B

```

break;
}

```

```

/*if(t==0.2) {
    fprintf(stderr,"eng=%30.18f,omega=%f\n",eng_tot,omgsize);
    fprintf(log_res3,"%30.18f %20.18f\n",dt,fabs(d_eng_tot));
    fprintf(log_res4,"%30.18f %30.18f\n",dt,fabs(d_omgsize));
}*/

```

```

/*jellelt constant*/

```

```

mez.a=-ea.z; mez.b=-eb.z; mez.c=-ec.z;
/*fprintf(stderr,"mez(%f %f %f)\n",mez.a,mez.b,mez.c);*/
normal_point(&mez,&tp);
Jellelt=-(moi.a*omg.a*tp.a + moi.b*omg.b*tp.b + moi.c*omg.c*tp.c);
if(itime%mprint==0) fprintf(log_res3,"%f %f\n",t,Jellelt);

```

```

/*Gyroscopic balance*/

```

```

F1= moi.c * omg.c;
F2 = moi.a * ec.z*(ec.x*(omg.b*ea.y-omg.a*eb.y)-ec.y*(omg.b*ea.x-omg.a*eb.x))/(ec.x*ec.x+ec.y*ec.y );
F3=F1-F2;

```

```

if (itime%mprint==0) fprintf(log_res4,"%f %f %f\n",t,F1,F2);

/*if (itime%mprint==0) fprintf(log_res4,"%f %f\n",t,fabs(F3));*/
/*if(itime%mprint==0) fprintf(log_res4,"%f %f\n",t,fabs((2*F3)/(F1+F2)));*/

/*Omega Size*/
if(t==0) fprintf(stderr,"init OMG=%f\n",omgsize);
/*if (t>2.4228) {fprintf(stderr,"omg=%f\n",omgsize); break;}*/

if(log_chk1 != NULL && itime % mprint == 0){
/*
fprintf(log_chk1,"%f %e %e %e %e %e %e\n",t,
ea.x*ea.x + ea.y*ea.y + ea.z*ea.z -1.0,
eb.x*eb.x + eb.y*eb.y + eb.z*eb.z -1.0,
ec.x*ec.x + ec.y*ec.y + ec.z*ec.z -1.0,
ea.x*eb.x + ea.y*eb.y + ea.z*eb.z
,
eb.x*ec.x + eb.y*ec.y + eb.z*ec.z
,
ec.x*ea.x + ec.y*ea.y + ec.z*ea.z
);
*/
fprintf(log_chk1,"%f %e\n",t,
fabs(ea.x*eb.x + ea.y*eb.y + ea.z*eb.z)
+fabs(eb.x*ec.x + eb.y*ec.y + eb.z*ec.z)
+fabs(ec.x*ea.x + ec.y*ea.y + ec.z*ea.z)
);
}

if(itime >= max_itime) break;

forces(&rc, &vc, &omg, &ea, &eb, &ec, &frc, &trq);

if(log_res2 != NULL && itime % mprint == 0)
fprintf(log_res2,"%f %f %f\n"
,t,angmom.x,angmom.y,angmom.z,angmomsize,omgsize
,eng_rot,eng_tra,eng_gra,eng_tot
,frc.x,frc.y,frc.z, trq.x,trq.y,trq.z, trq.a,trq.b,trq.c);

k1x=dt*(f1*omg.b*omg.c+fn1*trq.a);
k1y=dt*(f2*omg.c*omg.a+fn2*trq.b);
k1z=dt*(f3*omg.a*omg.b+fn3*trq.c);

k1ax=dt*(omg.c*eb.x-omg.b*ec.x);
k1ay=dt*(omg.c*eb.y-omg.b*ec.y);
k1az=dt*(omg.c*eb.z-omg.b*ec.z);

k1bx=dt*(omg.a*ec.x-omg.c*ea.x);
k1by=dt*(omg.a*ec.y-omg.c*ea.y);
k1bz=dt*(omg.a*ec.z-omg.c*ea.z);

k1cx=dt*(omg.b*ea.x-omg.a*eb.x);
k1cy=dt*(omg.b*ea.y-omg.a*eb.y);
k1cz=dt*(omg.b*ea.z-omg.a*eb.z);

k1rx=dt*vc.x;
k1ry=dt*vc.y;
k1rz=dt*vc.z;

k1vx=(dt/mass)*frc.x;
k1vy=(dt/mass)*frc.y;
k1vz=(dt/mass)*frc.z;

```

```

omgm.a=omg.a+k1x/2; omgm.b=omg.b+k1y/2; omgm.c=omg.c+k1z/2;
eam.x=ea.x+k1ax/2; eam.y=ea.y+k1ay/2; eam.z=ea.z+k1az/2;
ebm.x=eb.x+k1bx/2; ebm.y=eb.y+k1by/2; ebm.z=eb.z+k1bz/2;
ecm.x=ec.x+k1cx/2; ecm.y=ec.y+k1cy/2; ecm.z=ec.z+k1cz/2;
rcm.x=rc.x+k1rx/2; rcm.y=rc.y+k1ry/2; rcm.z=rc.z+k1rz/2;
vcm.x=vc.x+k1vx/2; vcm.y=vc.y+k1vy/2; vcm.z=vc.z+k1vz/2;

/* printf("%f %f %f %f\n",vc.z,rcm.z,k1vz,vcm.z);*/

forces(&rcm, &vcm, &omgm, &eam, &ebm, &ecm, &frc, &trq);

k2x=dt*(f1*omgm.b*omgm.c+fn1*trq.a);
k2y=dt*(f2*omgm.c*omgm.a+fn2*trq.b);
k2z=dt*(f3*omgm.a*omgm.b+fn3*trq.c);

k2ax=dt*(omgm.c*ebm.x-omgm.b*ecm.x);
k2ay=dt*(omgm.c*ebm.y-omgm.b*ecm.y);
k2az=dt*(omgm.c*ebm.z-omgm.b*ecm.z);

k2bx=dt*(omgm.a*ecm.x-omgm.c*eam.x);
k2by=dt*(omgm.a*ecm.y-omgm.c*eam.y);
k2bz=dt*(omgm.a*ecm.z-omgm.c*eam.z);

k2cx=dt*(omgm.b*eam.x-omgm.a*ebm.x);
k2cy=dt*(omgm.b*eam.y-omgm.a*ebm.y);
k2cz=dt*(omgm.b*eam.z-omgm.a*ebm.z);

k2rx=dt*vcm.x;
k2ry=dt*vcm.y;
k2rz=dt*vcm.z;

k2vx=(dt/mass)*frc.x;
k2vy=(dt/mass)*frc.y;
k2vz=(dt/mass)*frc.z;

omgm.a=omg.a+k2x/2; omgm.b=omg.b+k2y/2; omgm.c=omg.c+k2z/2;
eam.x=ea.x+k2ax/2; eam.y=ea.y+k2ay/2; eam.z=ea.z+k2az/2;
ebm.x=eb.x+k2bx/2; ebm.y=eb.y+k2by/2; ebm.z=eb.z+k2bz/2;
ecm.x=ec.x+k2cx/2; ecm.y=ec.y+k2cy/2; ecm.z=ec.z+k2cz/2;
rcm.x=rc.x+k2rx/2; rcm.y=rc.y+k2ry/2; rcm.z=rc.z+k2rz/2;
vcm.x=vc.x+k2vx/2; vcm.y=vc.y+k2vy/2; vcm.z=vc.z+k2vz/2;

/* printf("%f %f %f %f %f %f\n",rcm.z,k2rz,vc.z,frc.z,k2vz,vcm.z);*/

forces(&rcm, &vcm, &omgm, &eam, &ebm, &ecm, &frc, &trq);

k3x=dt*(f1*omgm.b*omgm.c+fn1*trq.a);
k3y=dt*(f2*omgm.c*omgm.a+fn2*trq.b);
k3z=dt*(f3*omgm.a*omgm.b+fn3*trq.c);

k3ax=dt*(omgm.c*ebm.x-omgm.b*ecm.x);
k3ay=dt*(omgm.c*ebm.y-omgm.b*ecm.y);
k3az=dt*(omgm.c*ebm.z-omgm.b*ecm.z);

k3bx=dt*(omgm.a*ecm.x-omgm.c*eam.x);
k3by=dt*(omgm.a*ecm.y-omgm.c*eam.y);
k3bz=dt*(omgm.a*ecm.z-omgm.c*eam.z);

k3cx=dt*(omgm.b*eam.x-omgm.a*ebm.x);
k3cy=dt*(omgm.b*eam.y-omgm.a*ebm.y);

```

```

k3cz=dt*(omgm.b*eam.z-omgm.a*ebm.z);

k3rx=dt*vcm.x;
k3ry=dt*vcm.y;
k3rz=dt*vcm.z;

k3vx=(dt/mass)*frc.x;
k3vy=(dt/mass)*frc.y;
k3vz=(dt/mass)*frc.z;

omgm.a=omg.a+k3x; omgm.b=omg.b+k3y; omgm.c=omg.c+k3z;
eam.x=ea.x+k3ax; eam.y=ea.y+k3ay; eam.z=ea.z+k3az;
ebm.x=eb.x+k3bx; ebm.y=eb.y+k3by; ebm.z=eb.z+k3bz;
ecm.x=ec.x+k3cx; ecm.y=ec.y+k3cy; ecm.z=ec.z+k3cz;
rcm.x=rc.x+k3rx; rcm.y=rc.y+k3ry; rcm.z=rc.z+k3rz;
vcm.x=vc.x+k3vx; vcm.y=vc.y+k3vy; vcm.z=vc.z+k3vz;

/*   printf("%f %f %f %f %f %f\n",rcm.z,k3rz,vc.z,frc.z,k3vz,vcm.z);*/

forces(&rcm, &vcm, &omgm, &eam, &ebm, &ecm, &frc, &trq);

k4x=dt*(f1*omgm.b*omgm.c+fn1*trq.a);
k4y=dt*(f2*omgm.c*omgm.a+fn2*trq.b);
k4z=dt*(f3*omgm.a*omgm.b+fn3*trq.c);

k4ax=dt*(omgm.c*ebm.x-omgm.b*ecm.x);
k4ay=dt*(omgm.c*ebm.y-omgm.b*ecm.y);
k4az=dt*(omgm.c*ebm.z-omgm.b*ecm.z);

k4bx=dt*(omgm.a*ecm.x-omgm.c*eam.x);
k4by=dt*(omgm.a*ecm.y-omgm.c*eam.y);
k4bz=dt*(omgm.a*ecm.z-omgm.c*eam.z);

k4cx=dt*(omgm.b*eam.x-omgm.a*ebm.x);
k4cy=dt*(omgm.b*eam.y-omgm.a*ebm.y);
k4cz=dt*(omgm.b*eam.z-omgm.a*ebm.z);

k4rx=dt*vcm.x;
k4ry=dt*vcm.y;
k4rz=dt*vcm.z;

k4vx=(dt/mass)*frc.x;
k4vy=(dt/mass)*frc.y;
k4vz=(dt/mass)*frc.z;

kx=(k1x+2*(k2x+k3x)+k4x)*(1.0/6.0);
ky=(k1y+2*(k2y+k3y)+k4y)*(1.0/6.0);
kz=(k1z+2*(k2z+k3z)+k4z)*(1.0/6.0);

kax=(k1ax+2*(k2ax+k3ax)+k4ax)*(1.0/6.0);
kay=(k1ay+2*(k2ay+k3ay)+k4ay)*(1.0/6.0);
kaz=(k1az+2*(k2az+k3az)+k4az)*(1.0/6.0);

kbx=(k1bx+2*(k2bx+k3bx)+k4bx)*(1.0/6.0);
kby=(k1by+2*(k2by+k3by)+k4by)*(1.0/6.0);
kbz=(k1bz+2*(k2bz+k3bz)+k4bz)*(1.0/6.0);

kcx=(k1cx+2*(k2cx+k3cx)+k4cx)*(1.0/6.0);
kcy=(k1cy+2*(k2cy+k3cy)+k4cy)*(1.0/6.0);
kcz=(k1cz+2*(k2cz+k3cz)+k4cz)*(1.0/6.0);

```

```

krx=(k1rx+2*(k2rx+k3rx)+k4rx)*(1.0/6.0);
kry=(k1ry+2*(k2ry+k3ry)+k4ry)*(1.0/6.0);
krz=(k1rz+2*(k2rz+k3rz)+k4rz)*(1.0/6.0);

kvx=(k1vx+2*(k2vx+k3vx)+k4vx)*(1.0/6.0);
kvy=(k1vy+2*(k2vy+k3vy)+k4vy)*(1.0/6.0);
kvz=(k1vz+2*(k2vz+k3vz)+k4vz)*(1.0/6.0);

/*   printf("%f %f %f %f\n",krz,frc.z,k4vz,kvz);*/

omg.a=omg.a+kx;
omg.b=omg.b+ky;
omg.c=omg.c+kz;

ea.x=ea.x+kax;
ea.y=ea.y+kay;
ea.z=ea.z+kaz;

eb.x=eb.x+kbx;
eb.y=eb.y+kby;
eb.z=eb.z+kbz;

ec.x=ec.x+kcx;
ec.y=ec.y+kcy;
ec.z=ec.z+kcz;

rc.x=rc.x+krx;
rc.y=rc.y+kry;
rc.z=rc.z+krz;

vc.x=vc.x+kvx;
vc.y=vc.y+kvy;
vc.z=vc.z+kvz;

fct=1/sqrt(ea.x*ea.x+ea.y*ea.y+ea.z*ea.z);
ea.x=fct*ea.x;ea.y=fct*ea.y;ea.z=fct*ea.z;
fct=1/sqrt(eb.x*eb.x+eb.y*eb.y+eb.z*eb.z);
eb.x=fct*eb.x;eb.y=fct*eb.y;eb.z=fct*eb.z;
fct=1/sqrt(ec.x*ec.x+ec.y*ec.y+ec.z*ec.z);
ec.x=fct*ec.x;ec.y=fct*ec.y;ec.z=fct*ec.z;
}
}
if (log_res1 != NULL) fclose(log_res1);
if (log_res2 != NULL) fclose(log_res2);
if (log_chk1 != NULL) fclose(log_chk1);
}

int forces(Lvec *rc, Lvec *vc, Bvec *omg, Lvec *ea, Lvec *eb, Lvec *ec,
          Lvec *frc, LBvec *trq) {
/*
uses global variables : mass, g_gravity,
*/
const double mu_friction = 0.5 ;
const double tspeed0 = 1.0e-3 ;
const double d_fr = 0.05; /* [cm] */

Lvec omgL ; /* angular velocity vector in L-frame */
Bvec tp ; /* tangent point in B-frame */
Bvec mez ; /* unit vector parallel to the gravity */

```

```

Lvec dtp ; /* displacement from center of rotor to tangent point */
Lvec vtp ; /* velocity of tangent point */
Lvec frctp ; /* force operating at the tangent point */
double htp ; /* height of tangent point */
double fr ; /* normal reaction force */
double ff ; /* tangential friction force */
double tspeed ; /* tangential speed */
double fct,t1 ;
double mufrict ; /* friction coefficient mu */

omgL.x = omg->a*ea->x + omg->b*eb->x + omg->c*ec->x ;
omgL.y = omg->a*ea->y + omg->b*eb->y + omg->c*ec->y ;
omgL.z = omg->a*ea->z + omg->b*eb->z + omg->c*ec->z ;
/*fprintf(stderr,"omgL.z=%f\n",omgL.z);*/

mez.a = - ea->z ; mez.b = - eb->z ; mez.c = - ec->z ;
normal_point(&mez, &tp);

dtp.x = tp.a*ea->x + tp.b*eb->x + tp.c*ec->x ;
dtp.y = tp.a*ea->y + tp.b*eb->y + tp.c*ec->y ;
dtp.z = tp.a*ea->z + tp.b*eb->z + tp.c*ec->z ;

vtp.x = vc->x + omgL.y * dtp.z - omgL.z * dtp.y ;
vtp.y = vc->y + omgL.z * dtp.x - omgL.x * dtp.z ;
vtp.z = vc->z + omgL.x * dtp.y - omgL.y * dtp.x ;
/*
printf("%f %f %f %f %f %f\n",vc->x,vc->y,vc->z,vtp.x,vtp.y,vtp.z);
*/
htp = rc->z + dtp.z ;
t1=htp*(1.0/d_fr); if(t1 < -5.0) t1=-5.0;
fr = (mass*g_gravity)*exp(-t1*(1.0+0.1*t1))
    * (1- 0.1*tanh(vc->z * (1.0/15.0)));

/*
t1=htp*(1.0/d_fr); if(t1 < -4.6) t1=-4.6;
fr = (mass*g_gravity)*exp(-t1);
*/
/*
printf("%f %f %f %f %f\n",rc->z, dtp.z, htp, t1, fr); if(1) exit(1);
*/

tspeed = sqrt(vtp.x * vtp.x + vtp.y * vtp.y);
mufrict = mu_friction * tanh(tspeed * (1.0/tspeed0));
ff = fr * mufrict ;

if(tspeed > 1.0e-32) fct = - ff / tspeed ; else fct = 0.0;
frctp.x = fct* vtp.x ;
frctp.y = fct* vtp.y ;
frctp.z = fr ;

frc->x = frctp.x ;
frc->y = frctp.y ;
frc->z = frctp.z - mass * g_gravity ;

trq->x = dtp.y * frctp.z - dtp.z * frctp.y ;
trq->y = dtp.z * frctp.x - dtp.x * frctp.z ;
trq->z = dtp.x * frctp.y - dtp.y * frctp.x ;

trq->a = trq->x * ea->x + trq->y * ea->y + trq->z * ea->z ;
trq->b = trq->x * eb->x + trq->y * eb->y + trq->z * eb->z ;

```

```

    trq->c = trq->x * ec->x + trq->y * ec->y + trq->z * ec->z ;

    /* printf("%f %f %f %f %f %f F\n",rc->z,dtp.z,htp,vc->z,fr,frc->z);*/

/*
    printf("%f %f %f %f %f %f %f\n",mufrict,fr,ff,frctp.x,frctp.y,frctp.z,frc->z);
    if(1) exit(1);
*/

    return 0;
}

int normal_point(Bvec *n, Bvec *tp){
/*
    For ellipsoid.
    uses global variables : Bvec R
*/
    double fct;
    tp->a=R.a*R.a*n->a; tp->b=R.b*R.b*n->b; tp->c=R.c*R.c*n->c;
    fct = 1.0/sqrt(tp->a*n->a + tp->b*n->b + tp->c*n->c) ;
    tp->a = tp->a*fct;
    tp->b = tp->b*fct;
    tp->c = tp->c*fct;

/*
    printf("%f %f %f\n",R.a,R.b,R.c);
    printf("%f %f %f\n",n->a,n->b,n->c);
    printf("%f %f %f\n",tp->a,tp->b,tp->c);
    if(1) exit(1);
*/

    return 0;
}

/*
    If call the init1 , rc and vc are necessary to establish the initial value
*/
/*
void init1(Lvec *ea, Lvec *eb, Lvec *ec, Bvec *omg0, Lvec *rc, Lvec *vc){

    double b;
    Bvec angular_velocity_ratio = {-1.0, 0.0, -0.01};
    double angular_velocity = 100*pi;

    ea->x=0.0; ea->y=0.0; ea->z=-1.0;
    eb->x=-1.0; eb->y=0.0; eb->z=0.0;
    ec->x=0.0; ec->y=1.0; ec->z=0.0;

    b=1/sqrt(angular_velocity_ratio.a*angular_velocity_ratio.a
            +angular_velocity_ratio.b*angular_velocity_ratio.b
            +angular_velocity_ratio.c*angular_velocity_ratio.c);

    omg0->a=angular_velocity*angular_velocity_ratio.a*b;
    omg0->b=angular_velocity*angular_velocity_ratio.b*b;
    omg0->c=angular_velocity*angular_velocity_ratio.c*b;

    rc->x=0; rc->y=0; rc->z=R.a;
    vc->x=0; vc->y=0; vc->z=0;
}

```

```

*/

void init2(Lvec *ea, Lvec *eb, Lvec *ec, Bvec *omg0, Lvec *rc, Lvec *vc){

    double b;
    double theta=(89.3/90.0)*0.5*pi;
    Bvec angular_velocity_ratio = {0.0, -sin(theta), cos(theta)};
        /*relative size of body-frame components of initial angular velocity */

    Bvec n ={0.0, sin(theta), -cos(theta)};
    Bvec tp;
    double angular_velocity = 100*pi; /*size fo initial ang. vel.[radian/sec]*/

    ea->x=1.0; ea->y=0.0; ea->z=0.0;
    eb->x=0.0; eb->y=cos(theta); eb->z=-sin(theta);
    ec->x=0.0; ec->y=sin(theta); ec->z=cos(theta);

    b=1/sqrt(angular_velocity_ratio.a*angular_velocity_ratio.a
            +angular_velocity_ratio.b*angular_velocity_ratio.b
            +angular_velocity_ratio.c*angular_velocity_ratio.c);

    omg0->a=angular_velocity*angular_velocity_ratio.a*b;
    omg0->b=angular_velocity*angular_velocity_ratio.b*b;
    omg0->c=angular_velocity*angular_velocity_ratio.c*b;

    normal_point(&n, &tp);

    rc->x=0; rc->y=0; rc->z=n.a*tp.a + n.b*tp.b + n.c*tp.c;
    vc->x=0; vc->y=0; vc->z=0;
    fprintf(stderr, "%f,%f,%f, %f,%f,%f\n", tp.a, tp.b, tp.c, omg0->a, omg0->b, omg0->c);

}

```