

# The Shape of a Planet

Sam Watkins <sam@nipl.net>

October 2009

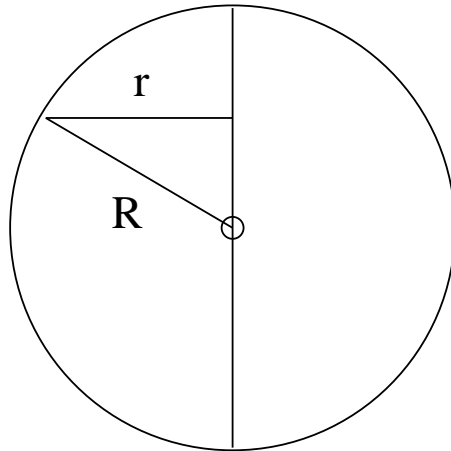
This paper shows how to determine the basic shape of a planet, spinning on its axis, using simple Newtonian physics and calculus. A planet's shape may be determined from its mass  $M$  (kg), its polar radius  $r_0$  (m), and its rate of spin  $\omega$  (radians/s).

Details such as mountains are not considered, we assume that the planet is locally flat. For calculating gravity, we assume that the planet's mass may be modelled by a point-mass at its centre. This model is exact for spherical objects, and practical for planets.

Consider a ball sitting somewhere on the surface of the planet. Two accelerations act upon the ball at that point: gravity  $\vec{g}$ , and normal reaction  $\vec{n}$ . The sum  $\vec{c}$  of these accelerations keeps the ball moving with the planet, in circular motion about the axis.

$$\vec{c} = \vec{g} + \vec{n}$$

The normal reaction at a point on the surface is perpendicular to the surface. We need only consider a cross-section of the planet, passing through the axis:



We know the formulae for the acceleration  $\vec{g}$  due to gravity, and the acceleration  $c$  needed to achieve circular motion:

$$\vec{g} = -G \frac{M}{R^2} \hat{R}$$

$$\vec{c} = -\omega^2 \vec{r}$$

$G = 6.67428 \times 10^{-11} m^3 kg^{-1} s^{-1}$  is the gravitational constant,  $M$  is the mass of the planet,  $\vec{R}$  is the vector from the center of the planet (not used directly),  $R$  is the magnitude of  $\vec{R}$ ,  $\hat{R}$  is the unit vector in the direction of  $\vec{R}$ ,  $\omega$  is the rate at which the planet is spinning (radians per second), and  $\vec{r}$  is the perpendicular vector from the axis of the planet. The gravity vector is directed toward the centre of the planet. The circular motion vector is directed toward the nearest point on the axis of the planet.

We can express the normal reaction in these terms:

$$\vec{n} = \vec{c} - \vec{g}$$

$$\vec{n} = G \frac{M}{R^2} \hat{R} - \omega^2 \vec{r}$$

Consider a point on the surface  $x, y$ , using cartesian coordinates with origin at the center of the planet,  $x$ -axis corresponding to the planet's equator and  $y$ -axis to the planet's axis, where  $x \geq 0$  and  $y \geq 0$ . We split the vectors into their  $x$  and  $y$  components, and consider the magnitude  $R$ .

$$r_x, r_y = x, 0$$

$$R_x, R_y = x, y$$

$$R^2 = x^2 + y^2$$

$$n_x = G \frac{M}{R^2} \frac{x}{R} - \omega^2 x$$

$$n_y = G \frac{M}{R^2} \frac{y}{R}$$

Now, considering the curve of the surface, it is perpendicular to the normal:

$$\frac{dy}{dx} = -\frac{n_x}{n_y}$$

$$\frac{dy}{dx} = \left( \frac{\omega^2 R^3}{GM} - 1 \right) \frac{x}{y}$$

Let  $a = \frac{\omega^2}{GM}$ :

$$\frac{dy}{dx} = (aR^3 - 1) \frac{x}{y}$$

This equation determines the curve of the surface, and can be integrated as follows:

$$ydy = (aR^3 - 1) xdx$$

$$dy^2 = (aR^3 - 1) dx^2$$

Substitute  $X = x^2$ ,  $Y = y^2$ ,  $Z = X + Y = R^2$ :

$$dY = \left( aZ^{\frac{3}{2}} - 1 \right) dX$$

$$d(Z - X) = \left( aZ^{\frac{3}{2}} - 1 \right) dX$$

$$dZ = aZ^{\frac{3}{2}} dX$$

$$\frac{dX}{dZ} = \frac{1}{a} Z^{-\frac{3}{2}}$$

$$X = -\frac{2}{a} Z^{-\frac{1}{2}} + k$$

$$x^2 = -\frac{2}{aR} + k$$

$$r^2 = -\frac{2GM}{\omega^2 R} + k$$

At the pole,  $R = r_0$  and  $r = 0$ , so we can find  $k$ :

$$k = \frac{2GM}{\omega^2 r_0}$$

So we have  $r$  as a function of  $R$ :

$$r^2 = \frac{2GM}{\omega^2} \left( \frac{1}{r_0} - \frac{1}{R} \right)$$

We can express  $x, y$  in terms of  $R$ , giving a parametric equation for the curve of the planet's surface, which we can plot using a computer:

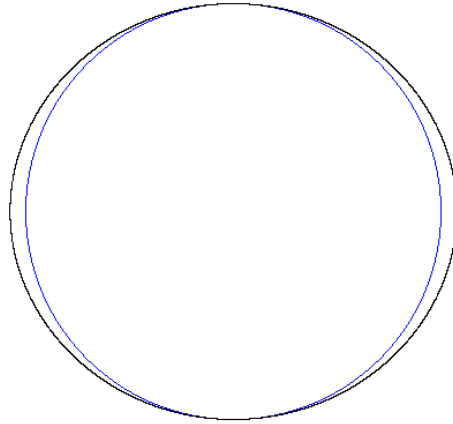
$$x, y = r, \sqrt{R^2 - r^2}$$

At the equator,  $r = R$ , so the equatorial radius  $r_1$  satisfies:

$$r_1^2 = \frac{2GM}{\omega^2} \left( \frac{1}{r_0} - \frac{1}{r_1} \right)$$

This is a cubic in  $r_1$  which can be solved exactly, but I have not done so here.

I wrote a computer program to draw planet shapes based on these formulae, and to calculate the amount of bulge at the equator. Here is the picture generated of Saturn, which has a visible bulge compared to the reference circle:



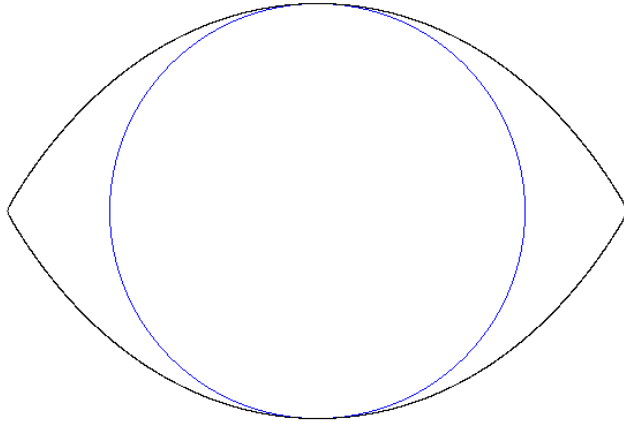
The values calculated for the bulges (equatorial diameter minus polar diameter) of Earth and Saturn are a bit different from what has been observed in real life:

Earth: computed bulge: 21.89 km, actual bulge: 42.72 km.

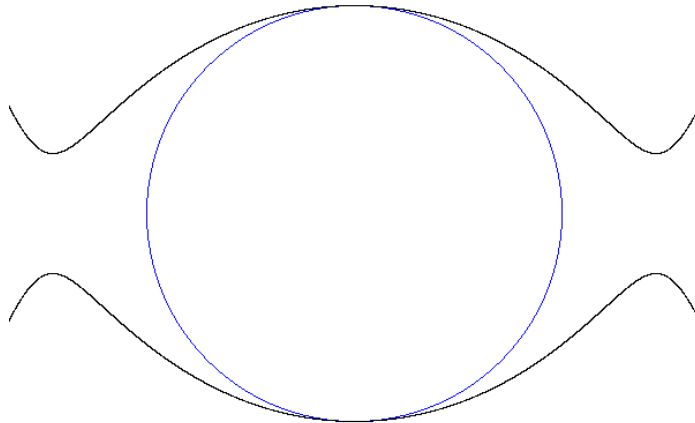
Saturn: computed bulge: 8239.7 km, actual bulge: 11808 km.

The calculated values are not close enough to the actual values, so I think I have made some mistake, or did not consider some significant factor. Nevertheless, I think my method is the right approach.

Here is a picture of the shape Saturn would have, if it were spinning 56% faster than it really does, at the critical point.



Here is a picture of the shape Saturn would have, if it were spinning 59% faster than it really does. The shape is no longer closed. Saturn could not spin that fast and remain in one piece.



Here is the program I used, it is written in brace, my dialect of C:

```
#!/lang/b
use b

cstr usage[] =
    "m          r0          p          [steps=10000]",
    "mass      polar-radius spin-period ",
    "5.9736e24  6356800    86164.09  # Earth",
    "5.6869e26  54364000    37050.56  # Saturn",
    NULL
```

```

num G = 6.67428e-11 # m^3/kg/s

Main()
  num M, r0, p, s, a, b, R2, r2, r, x, y, yp, xp, R, dR, r1, steps
  steps = 10000
  Getargs(num, M, r0, p)
  getargs(num, steps)
  warn("M=%g r0=%f p=%f ", M, r0, p)
  s = 2*pi/p

  paper(600, 400)
  int max_wh = imax(w, h)
  zoom((max_wh * 0.6) / (r0*2))
  blue()
  circle(0, 0, r0)

  black()
  a = 2*G*M/(s*s)
  b = 1/r0
  x = 0, y = r0, yp = y, xp = 0
  dR = r0 / steps
  for R=r0; R<r0*2; R+=dR:
again    R2 = R*R
         r2 = a*(b - 1/R)
         if r2 < -tinynum || r2 > R2 + tinynum:
           break
         r2 = clamp(r2, 0, R2)
         r = sqrt(r2)
         x = r
         y = sqrt(R2 - r2)
         if sd(x) > w_2:
           break
         point(x, y)
         point(-x, y)
         point(x, -y)
         point(-x, -y)
         Paint()
         if xp && hypot(yp-y, xp-x) > r0 / h / 2:
           R-=dR ; dR /= 2 ; R+=dR ; again
         eif xp && hypot(yp-y, xp-x) < r0 / h / 8:
           R-=dR ; dR *= 2 ; R+=dR ; again
         yp = y ; xp = x

  r1 = hypot(x, y)
  warn("r1 ~=%f ", r1)
  warn("equatorial bulge = 2*(r1-r0) = %f km", 2*(r1-r0)/1000)

```