Orbital Momentum as a Proportion of Degrees Pi to Seconds in a Year

Date: 2024-09-13 ***Author:*** Taylor Metz, *unaffiliated*

Abstract

---

1. Summary:

Performing a series of mathematical evaluations involving constants like π (pi) and φ (phi), and exploring their relationships to angles, time, and speed differences. The calculations focus on how these constants interrelate in certain physical contexts, such as velocities, corrections, and differences.

---

2. Evaluate Expression:

$$\frac{\left(0.002 \times \frac{31,583,458}{\pi} \times \frac{360}{361} \times \frac{3}{2} + (360 - 361)\right)}{1.01}$$

2.1 Compute the Inner Components:

1. First, calculate:

   $$\frac{31,583,458}{\pi} = \frac{31,583,458}{3.141592653} \approx 10,043,243.964$$

2. Multiply by :

   $$0.002 \times 10,043,243.964 \approx 20,086.488$$

3. Multiply by :

   $$20,086.488 \times 0.999444 \approx 20,084.415$$

4. Multiply by :

   $$20,084.415 \times 1.5 \approx 30,126.623$$

5. Add :

   $$30,126.623 - 1 = 30,125.623$$

6. Finally, divide by :

   $$\frac{30,125.623}{1.01} \approx 29,850.52$$

---

3. Evaluate Golden Ratio Expression:

$$1 - \frac{\left(\frac{1 + \sqrt{5}}{\pi} - 1\right)}{2}$$

3.1 Compute Golden Ratio Components:

1. The golden ratio is:

   $$\frac{1 + \sqrt{5}}{2} \approx 1.618033988$$

2. Compute:

   $$\frac{1 + \sqrt{5}}{\pi} = \frac{3.236067977}{3.141592653} \approx 1.031042$$

3. Subtract 1:

   $$1.031042 - 1 = 0.031042$$

4. Divide by 2:

   $$\frac{0.031042}{2} \approx 0.015521$$

5. Finally, subtract from 1:

   $$1 - 0.015521 = 0.984479$$

3.2 Three Space Dimensions and Time of Delta Phi:

1. Start with the golden ratio:

   $$\frac{1 + \sqrt{5}}{2} \approx 1.618033988$$

2. Multiply by 4 and the absolute value of :

   $$4 \times 0.015521 \approx 0.062084$$

3. Add to the golden ratio:

   $$1.618033988 + 0.062084 \approx 1.680118$$

---

4. Function Calculation:

For the function :

$$f(X, Y) = \frac{(X \times Y) + (Y \times X)}{X + Y}$$

First Case Calculation:

1. Calculate:
   $$\frac{(40,223.7 \times 25,362) + (25,362 \times 40,223.7)}{40,223.7 + 25,362} = \frac{2 \times 40,223.7 \times 25,362}{65,585.7}$$

$$\frac{2 \times 1,019,607,019.4}{65,585.7} \approx \frac{2,039,214,038.8}{65,585.7} \approx 31,089.23$$

Second Case Calculation:

1. Compute the components:

   $$1.587 \times 25,362 \approx 40,274.814$$
   $$0.627 \times 40,223.7 \approx 25,207.492$$]

2. Sum and divide:

$$\frac{40,274.814 + 25,207.492}{1.587 + 0.627} = \frac{65,482.306}{2.214} \approx 29,570$$

---

5. Phi Difference:

1. Differences:

   $$29,700 - 29,570 = 130$$
   $$29,850 - 29,700 = 150$$
   $$180 - 150 = 30$$

2. Correction term:

   $$\frac{30}{361 \times 3} + 0.015521 \times 2 \approx 0.08644$$

3. Next, compute:

   $$1 - \frac{130}{150} \approx 1.153846153846 - 1 = 0.153846$$

4. Final calculation:

   $$365.2926 \times \left(1 + \frac{0.153846}{0.08644} - 1.618033988 - \frac{1.618033988}{10}\right)$$

Step-by-Step:

1. Compute the ratio:

   $$\frac{0.153846}{0.08644} \approx 1.78$$

2. Subtract the golden ratio and its fraction:

   $$1.78 - 1.618033988 - 0.1618033988 \approx 0.0001626$$

3. Apply this to the original expression:

   $$365.2425 \times (1 + 0.0001626) \approx 365.2929$$
   $$\frac{365.2929}{1 + 0.0001626} \approx 365.2425$$

Table of Seconds in 365.2425 versus 361° or 24/99 years as leap years

---

1. **Constants and Definition of Variables/Scaling**

Constants							
n	1.000		Phi	1.618033989		Days	365.242424242
degree = d	360.000		dPhi	0.015036215		Hours	8,765.818181818
d + n = dn	361.000		1-dPhi	98.496378520%		Minutes	525,949.09090909
dn / d = delta n	1.002777778		1-dPhi^2	99.977391224%		Seconds = s	31,556,945.454546
pi	3.141592654		1-dPhi^3	99.999660050%		Delta to time (leap year difference)	364.935566077

Constants							
time	31,583,458.000		1-dPhi^4	99.999994888%		6.443392256%	
Earth momentum = mom	29,761.29					^– *365	23.51838173
						^– /24	97.993257225%

2. **360° year + (1° * pi) of 360° year**

361pi/360	
npi = d+n * pi	1,134.114947946
npi/d	3.150319300
pi * delta n = piDn	3.150319300
piDn - (npi/d)	100.000000000%

3. **A’, B’, C’ dimensions of t Time 365.2425 Orbital Momentum**

t Time Seconds in 365.2425 Day Year	
t	31,583,458.000
10^-7	0.000000100
t x 10^-7 = td	3.158345800
td / delta n	3.149596920
td / pi = piDt	100.533269213%
piDt / delta n = A`	100.254783702%
piDt - delta n = B`	0.255491435%
A- B	99.999292267%
(A+ B)/2+(1/2) = C`	100.255137569%
C` - 1	0.255137569%
C- B	99.999646134%

4. **A’, B’, C’ dimensions of s Seconds 365 and 24/99 leap year Orbital Momentum**

s / Y seconds per year	
s	31,556,945.454546
10^-7	0.000000100
s x 10^-7 = sd	3.155694545
sd / delta n	3.146953009
sd / pi = piDs	100.448877159%

s / Y seconds per year	
piDs / delta n = A`	100.170625422%
piDs - delta n = B`	0.171099382%
A- B	99.999526040%
(A+ B)/2+(1/2) = C`	100.170862402%
C` - 1	0.170862402%
C- B	99.999763020%

5. **Variance t Time to s Second 365》366 days**

Outputs			
A	B	C	D
s - t	t-s	s-t * Delta to Time	t-s * Delta to Time
-26,512.545455	26,512.545455	-28,220.852755	28,220.852755
0.000000000	0.000000000	0.000000000	0.000000000
-0.002651255	0.002651255	-0.002822085	0.002822085
-0.002643910	0.002643910	-0.002814268	0.002814268
-0.084392053%	0.084392053%	-0.089829764%	0.089829764%
-0.084158280%	0.084158280%	-0.089580928%	0.089580928%
-0.084392053%	0.084392053%	-0.089829764%	0.089829764%
0.000233773%	-0.000233773%	0.000248836%	-0.000248836%
-0.084275167%	0.084275167%	-0.089705346%	0.089705346%
-0.084275167%	0.084275167%	-0.089705346%	0.089705346%
0.000116887%	-0.000116887%	0.000124418%	-0.000124418%
	A+D	B+C	
K	1,708.307300656	-1,708.307300656	
d(ST) = [S|T]-/+(K*2)	-29,929.160055856	29,929.160055856	
mom	29,761.29	29,761.29	
d(ST) +/- mom	(167.8700558558)	(167.8700558558)	

Code And Draft Genralization

Time as a Golden Ratio?

The proportional difference of 1 degree of momentum to space time to the golden ratio proportional to Pi in a novel methodology

Abstract

Under the following observations:

```
1. When a year is represented as 365.2424 days in 400 years, the amount of seconds required for the

2. (Pi times 361 degrees radian) divided 360 degrees radian is approximately 3.158345810, since 361/

3. 3.15031664 / 3.15834581 = 0.997457792; 3.15834581 / 3.15031664 = 1.002548687; (1.002548687 + 0.99

4. □=( 1+sqrt(5) ) / 2 ~=1.618033989.

5. ((((1+5)÷π)−1)÷2) = 0.015036215, which as 1-x^n for x=0.015036215 and n of:
```

x^N		
N	X^N|1-(X_N)	
1	0.015036215	0.984963785
2	0.000226088	0.999773912
3	0.000003400	0.999996600
4	0.000000051	0.999999949
5	0.000000001	0.999999999
6	0.000000000	1.000000000
7	0.000000000	1.000000000

Then it can be assumed that there is a proportional difference than can be represented within these series.

Introduction

Given the combinatorics of integer and degrees rotations observed in the seemingly random observations of the abstract, Python code was created to examine how these would look with respect to Earths momentum around the Sun, and how it would compare to the Speed of Light moving as a vector from the sun as Earth revolves moving ever slightly away from that light both in our orbit and rotation of a day. Under these, a direct proportional representation was found that when compared to traditional methods resulted in the following outcomes: **Outputs:** - Using circumference / seconds as Linear Velocity: 29761.29 m/s - Angular Momentum: 2.66e+40 kg·m²/s - Linear Velocity as fraction of speed of light: 0.992704876 - Using circumference * geometric ratio * 10^-8: 29611.88 m/s - Angular Momentum: 2.65e+40 kg·m² - Adjusted Linear Velocity as fraction of speed of light: 0.987721272

Which, when comparing the adjusted speed of light with the new adjusted method of degree rotations results in a comparison of 0.987721272 ~= 0.984963785; 1 - (0.987721272 - 0.984963785) = 0.992704876. Since 0.992704876 is equal to the relative speed of light under classical frameworks, the methodology must stand to reason that the proportional difference to 1 degree rotation compares to 1 second rotation is generalized as: ***(ϕ÷π)-(361π÷360)***.

Concept

What We Need to Do:

1. **Extract the actual mathematical formulas** from the Python code.
2. **Generalize them** into a clear equation that can be directly input into a calculator or Excel as a math formula.
3. **Incorporate the golden ratio or other adjustments** (if needed) for comparison.

Key Formula in the Code

Linear Velocity Formula:

The Python code uses this equation for linear velocity:

[V_{\text{linear}} = \frac{0.002 \times T}{\pi \times \frac{360}{361} \times \frac{3}{2}} + \frac{360 - 361}{1.01}]

where:

$$

- (T = 31,583,458) seconds is the time for one orbit.
- (\pi = 3.14159).
- The geometric adjustment factor is (\frac{361}{360}).
- There's a correction with the term (\frac{360 - 361}{1.01}). $$

This can be entered directly into Excel. For example, in Excel:

```
V_linear = ((0.002 * 31583458 / PI() * 360 / 361 * 3 / 2) + ((360 - 361) / 1.01))
```

This gives the linear velocity in meters per second.

Angular Momentum Formula:

In the code, angular momentum is calculated using the formula:

$$ [L = M \times V_{\text{linear}} \times R] where:

- (M = 5.972 \times 10^{24} , \text{kg}) is the mass of the Earth.
- (R = 1.496 \times 10^{11} , \text{m}) is the radius of Earth's orbit. $$

To write this as a formula in Excel:

```
Angular_Momentum = (5.972E24 * V_linear * 1.496E11)
```

Adjusted Velocity:

The adjusted velocity in the code uses the geometric ratio:

[V_{\text{adjusted}} = \text{circumference} \times \frac{361 \times \pi}{360} \times 10^{-8}]
This is a modification of the standard velocity by a geometric factor. For Excel, it translates to:

```
V_adjusted = (2 * PI() * 1.496E11) * (361 * PI() / 360) * 10^-8
```

Golden Ratio Incorporation:

If you want to add a **golden ratio proportional adjustment**, you can multiply the velocity by:

[\text{Golden Ratio} = 1 - \left(\frac{(1 + \sqrt{5})}{\pi} - 1 \right) \div 2]
This can be calculated in Excel as:

```
Golden_Ratio = 1 - (((1 + SQRT(5)) / PI()) - 1) / 2
```

Then, you can apply it to the velocity as:

```
V_final = V_linear * Golden_Ratio
```

Full Generalized Equations for Excel:

1. Linear Velocity:

```
V_linear = ((0.002 * 31583458 / PI() * 360 / 361 * 3 / 2) + ((360 - 361) /
1.01))
```

2. Angular Momentum:

```
Angular_Momentum = (5.972E24 * V_linear * 1.496E11)
```

3. Adjusted Velocity:

```
V_adjusted = (2 * PI() * 1.496E11) * (361 * PI() / 360) * 10^-8
```

4. Golden Ratio:

```
Golden_Ratio = 1 - (((1 + SQRT(5)) / PI()) - 1) / 2
V_final = V_linear * Golden_Ratio
```

Python Code

```
import math
import matplotlib.pyplot as plt
import numpy as np

def calculate_velocities_and_angular_momentum():
    """
    Calculate and print the linear and adjusted orbital velocities of Earth
and its angular momentum.
    """
```

```
    # Constants
    radius_of_earth_orbit = 1.496e11  # meters
    seconds_per_orbit = 31583458  # seconds in one orbit
    geometric_ratio = 361 * math.pi / 360  # Geometric adjustment factor
    mass_of_earth = 5.972e24  # kg

    # Calculate linear velocity
    circumference = 2 * math.pi * radius_of_earth_orbit
    linear_velocity = ((0.002*seconds_per_orbit/math.pi*360/361 /
geometric_ratio / 359 )*(3/2))+((360-361)))/1.01

    # Adjust the linear velocity with the geometric ratio
    adjusted_linear_velocity = circumference * geometric_ratio * 10**-8

    # Calculate angular momentum
    angular_momentum = mass_of_earth * linear_velocity * radius_of_earth_orbit
    adjusted_angular_momentum = mass_of_earth * adjusted_linear_velocity *
radius_of_earth_orbit

    # Output the results
    print(f"Using circumference / seconds as Linear Velocity:
{linear_velocity:.2f} m/s")
    print(f"Angular Momentum: {angular_momentum:.2e} kg·m²/s")
    print(f"Using circumference * geometric ratio * 10^-8:
{adjusted_linear_velocity:.2f} m/s")
    print(f"Angular Momentum: {adjusted_angular_momentum:.2e} kg·m²")

    return linear_velocity, adjusted_linear_velocity, radius_of_earth_orbit

def plot_orbits(radius_of_earth_orbit, adjusted_radius_of_earth_orbit):
    """
    Plot Earth's orbit around the Sun, showing the actual and adjusted orbits.
    """
    # Define the angles for the orbits
    angles = np.linspace(0, 2 * np.pi, 1000)

    # Calculate the x and y coordinates for the orbits
    x_actual = radius_of_earth_orbit * np.cos(angles)
    y_actual = radius_of_earth_orbit * np.sin(angles)
    x_adjusted = adjusted_radius_of_earth_orbit * np.cos(angles)
    y_adjusted = adjusted_radius_of_earth_orbit * np.sin(angles)

    # Create the plot
    plt.figure(figsize=(3,3))
    plt.plot(x_actual, y_actual, 'r-', label='Actual Orbit (Red)')
    plt.plot(x_adjusted, y_adjusted, 'b-', label='Adjusted Orbit (Blue)')
```

```
    plt.xlabel('X (meters)')
    plt.ylabel('Y (meters)')
    plt.title('Earth\'s Orbit Around the Sun')
    plt.legend()
    plt.grid(True)
    plt.axis('equal')
    plt.show()

# Calculate velocities and angular momentum
linear_velocity, adjusted_linear_velocity, radius_of_earth_orbit =
calculate_velocities_and_angular_momentum()

# Plot orbits with the geometric ratio affecting the radius
geometric_ratio = 361 * math.pi / 360  # Geometric adjustment factor
adjusted_radius_of_earth_orbit = radius_of_earth_orbit * geometric_ratio *
10**-8

def compare_with_light_speed():
    """
    Compare the linear and adjusted orbital velocities of Earth with the speed
of light.
    """
    # Constants
    speed_of_light = 2.998e8  # meters/second
    radius_of_earth_orbit = 1.496e11  # meters
    seconds_per_orbit = 31583458  # seconds in one orbit
    geometric_ratio = 361 * math.pi / 360  # Geometric adjustment factor
    mass_of_earth = 5.972e24  # kg
    # Calculate linear velocity
    circumference = 2 * math.pi * radius_of_earth_orbit
    # Calculate velocities
    linear_velocity = circumference / seconds_per_orbit
    adjusted_linear_velocity = circumference * geometric_ratio * 10**-8
    # Compare with speed of light
    linear_velocity_ratio = linear_velocity / speed_of_light*10**4
    adjusted_linear_velocity_ratio = adjusted_linear_velocity /
speed_of_light*10**4
    # Output the results
    print(f"Linear Velocity as fraction of speed of light:
{linear_velocity_ratio:.10f}")
    print(f"Adjusted Linear Velocity as fraction of speed of light:
{adjusted_linear_velocity_ratio:.10f}")
# Call the function to compare velocities
compare_with_light_speed()

#plot_orbits(radius_of_earth_orbit, adjusted_radius_of_earth_orbit) #
```

```
"""
Outputs:

Using circumference / seconds as Linear Velocity: 29761.29 m/s
Angular Momentum: 2.66e+40 kg·m²/s

Using circumference * geometric ratio * 10^-8: 29611.88 m/s

Angular Momentum: 2.65e+40 kg·m²

Linear Velocity as fraction of speed of light: 0.9927048763

Adjusted Linear Velocity as fraction of speed of light: 0.9877212724
"""
```

Golden Ratio Proportional Algebraically:

$1-(((1+\sqrt{5})\div\pi)-1)\div 2$

Angular Momentum Algebraically:

$(((0.002\times 31583458\div\pi\times 360\div 361)\times(3\div 2))+((360-361)))\div 1.01$

Summary:

- These are the **generalized math formulas** based on the Python code, ready to be entered into Excel.
- You can **use these formulas** to plug in inputs and get the desired outputs (e.g., velocity, angular momentum, etc.).