Abstract

This work applies the concept of time dilation from special relativity [1] to the legendary reign lengths of the Sumerian Kings of Kish [2]. For each king, the velocity (as a fraction of the speed of light) required for their experienced lifetime to match a proper time of 40 years is inferred. The resulting velocities, experienced times, and total distances traveled are calculated, with discussion of both individual and collective relativistic scenarios. The Python code used for these calculations is shown in Listing 1 and the resulting table is presented in Table 1.

Relativistic Time Dilation Applied to the Reigns of the Kings of Kish

Babak Makkinejad*

November 23, 2025

1 Introduction

The Sumerian King List attributes extremely long reigns to early rulers of Kish [2]. By interpreting these reigns through the lens of special relativity [1], we can compute the velocity a king would hypothetically need to travel in order to experience only 40 years (t_0) while their reign appears much longer in the Earth's reference frame.

2 Methodology

For each king with a reign length T (in Earth years), the velocity v (as a fraction of the speed of light, c) is determined using the standard time dilation formula [1]:

$$t_0 = T\sqrt{1 - \frac{v^2}{c^2}} \tag{1}$$

Rearranged to solve for v:

$$v = c\sqrt{1 - \left(\frac{t_0}{T}\right)^2}. \tag{2}$$

If the proper lifetime t_0 exceeds the legendary reign T, the velocity is set to a small fraction of the speed of light ($v = 0.001c$).

*Email: babak_makkinejad@hotmail.com

3 Python Implementation

The Python code used to compute the inferred velocities, experienced times, and distances for each king is shown in Listing 1 [3, 4].

Listing 1: Python code to compute velocities, experienced times, and distances for the Kings of Kish.

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

king_names = [
    # Antediluvian (mythical) kings
    "Jushur","Kullassina-bel","Nangishlishma","En-tarah-ana","Babum",
    "Puannum","Kalibum","Zuqaqip","Atab","Mashda","Arwium","Etana",
    "Balih","En-men-lu-ana","Dumuzid, the Shepherd","Ensipazi-anna",
    "Enmengal-ana","Dumuzid, the Fisherman",
    # Postdiluvian kings
    "Jushur","Kullassina-bel","Nangishlishma","En-tarah-ana","Babum",
    "Puannum","Kalibum","Zuqaqip","Atab","Mashda","Arwium","Etana",
    "Balih","En-me-barage-si","Aga"
]

reigns = np.array([
    1200,960,670,420,300,840,960,900,600,840,720,1560,
    400,1200,1000,700,670,1000,
    1200,960,670,420,300,840,960,900,600,840,720,1560,
    400,900,625
])  # Earth years

t0 = 40  # Proper time in years
c = 1    # Speed of light units

# Compute velocities with special handling if t0 >= reign
velocities = np.where(
    reigns > t0,
    c * np.sqrt(1 - (t0 / reigns)**2),
    0.001 * c
)

# Experienced times for each king
experienced_times = reigns * np.sqrt(1 - velocities**2 / c**2)

# Distances traveled in light-years
```

```
distances = velocities * reigns
total_distance = np.sum(distances)

# Print table of results
print(f"{'King':<25} {'Reign':>6} {'v/c':>8} {'Experienced':>12} {'Distance':>10}")
print("-"*65)
for name, reign, v, exp, d in zip(king_names, reigns, velocities, experienced_times, distances):
    print(f"{name:<25} {reign:>6} {v:>8.4f} {exp:>12.2f} {d:>10.2f}")

print(f"\nTotal distance traveled: {total_distance:.2f} light-years")

# Plot experienced times
plt.figure(figsize=(12,5))
plt.bar(king_names, experienced_times, color='steelblue')
plt.ylabel('Experienced Time (years)')
plt.xticks(rotation=45, ha='right')
plt.title(f'Experienced Time of Sumerian Kings (t0={t0} years)')
plt.tight_layout()
plt.show()
```

4 Results

The results are summarized in Table 1, which lists each king's legendary reign, inferred velocity (v/c), experienced time, and distance traveled. The table clearly distinguishes antediluvian (shaded) and postdiluvian kings.

5 Discussion

The velocities required for legendary reigns far exceeding t_0 approach the speed of light [1]. If all kings were on the same spaceship, a single velocity would be computed using the longest reign. Distances traveled under this hypothetical model are enormous [2]. The Python libraries used for the computations and plotting are cited in Listings 1 [3, 4].

6 Conclusion

This framework demonstrates an amusing, physically motivated reinterpretation of the Sumerian King List using special relativity. It allows the con-

Table 1: Inferred velocities, experienced times, and distances for the Kings of Kish assuming $t_0 = 40$ years. Shaded rows represent antediluvian (mythical) kings; unshaded rows are postdiluvian kings.

King	Reign (yr)	v/c	Experienced Time (yr)	Distance (ly)
Jushur	1200	0.9986	40.00	1198.31
Kullassina-bel	960	0.9982	40.00	958.31
Nangishlishma	670	0.9978	40.00	667.84
En-tarah-ana	420	0.9971	40.00	418.88
Babum	300	0.9960	40.00	298.80
Puannum	840	0.9987	40.00	839.00
Kalibum	960	0.9982	40.00	958.31
Zuqaqip	900	0.9980	40.00	898.20
Atab	600	0.9967	40.00	598.02
Mashda	840	0.9987	40.00	839.00
Arwium	720	0.9981	40.00	718.61
Etana	1560	0.9987	40.00	1559.49
Balih	400	0.9950	40.00	399.00
En-men-lu-ana	1200	0.9986	40.00	1198.31
Dumuzid, the Shepherd	1000	0.9980	40.00	998.00
Ensipazi-anna	700	0.9979	40.00	697.57
Enmengal-ana	670	0.9978	40.00	667.84
Dumuzid, the Fisherman	1000	0.9980	40.00	998.00
Jushur	1200	0.9986	40.00	1198.31
Kullassina-bel	960	0.9982	40.00	958.31
Nangishlishma	670	0.9978	40.00	667.84
En-tarah-ana	420	0.9971	40.00	418.88
Babum	300	0.9960	40.00	298.80
Puannum	840	0.9987	40.00	839.00
Kalibum	960	0.9982	40.00	958.31
Zuqaqip	900	0.9980	40.00	898.20
Atab	600	0.9967	40.00	598.02
Mashda	840	0.9987	40.00	839.00
Arwium	720	0.9981	40.00	718.61
Etana	1560	0.9987	40.00	1559.49
Balih	400	0.9950	40.00	399.00
En-me-barage-si	900	0.9980	40.00	898.20
Aga	625	0.9969	40.00	623.81

version of mythical reign lengths into hypothetical relativistic velocities, experienced lifetimes, and interstellar distances.

References

References

[1] A. Einstein, "Zur Elektrodynamik bewegter Körper," *Annalen der Physik*, vol. 17, pp. 891–921, 1905.

[2] Thorkild Jacobsen, *The Sumerian King List*, Chicago: University of Chicago Press, 1906.

[3] J. D. Hunter et al., *Matplotlib: Visualization with Python*, 3rd edition, 2023. `https://matplotlib.org`

[4] C. R. Harris et al., *Array programming with NumPy*, Nature 585, 357–362, 2023. `https://numpy.org`