**Appendix A: Vector-Based Astronomical Alignment Analysis**

**A.1 Astronomical Parameters and Modelling Conditions**

Celestial alignments were modelled using the following parameters:

- Target star: Alnitak (ζ Orionis), the easternmost star of Orion’s Belt

- Epochs: 2500 ± 30 BCE and 4400 ± 200 BCE

- Event: Vernal equinox heliacal rising

- Observation time: 04:00 local time

- Coordinates: Giza Plateau, 29.9792°N, 31.1342°E (WGS84)

- Corrections: Precession, nutation, ΔT (Skyfield + VSOP87 ephemerides; Morrison & Stephenson, 2004)

- Terrain correction: Horizon slope adjustment using local DEM

- Atmospheric model: Clear-sky Holocene conditions (Butzer, 1976)

- Azimuthal uncertainty bounds:

- Survey tolerance: ±0.5°

- Terrain slope estimation: ±1.0°

- Refraction variability: ±0.2°

- Ephemeris uncertainty (ΔT): ±0.3°

**A.2 S-Value Metric and Vector Formulation**

To assess angular similarity between structures and Alnitak’s rising azimuth, we define the S-value as the 2D Euclidean distance between their corresponding unit vectors on the unit circle:

Here:

- θ*\_p: Azimuth of the structure*

*- θ\_*s: Rising azimuth of Alnitak

- (x*\_p, y\_*p), (x*\_s, y\_*s): Unit vectors of structure and star respectively

This vector-based metric provides a rotationally invariant measure of alignment on a circular domain.

Classification thresholds:

- S < 0.02: Strong alignment

- 0.02 ≤ S < 0.1: Moderate alignment

- S ≥ 0.1: Weak or no alignment

These thresholds were derived from null model simulations (see Section A.6).

**A.3 Python-Based Analytical Reproducibility**

The following Python code calculates S-values using structure azimuths and Alnitak’s positions:

```python

import numpy as np

import pandas as pd

def azimuth\_to\_unit\_vector(az\_deg):

az\_rad = np.deg2rad(az\_deg)

return np.cos(az\_rad), np.sin(az\_rad)

def compute\_s\_value(az\_structure, az\_star):

x\_p, y\_p = azimuth\_to\_unit\_vector(az\_structure)

x\_s, y\_s = azimuth\_to\_unit\_vector(az\_star)

return np.sqrt((x\_p - x\_s)\*\*2 + (y\_p - y\_s)\*\*2)

data = [

{"Structure": "Khufu", "Azimuth": 90.9},

{"Structure": "Khafre Valley Temple", "Azimuth": 90.6},

{"Structure": "Menkaure", "Azimuth": 91.1},

{"Structure": "Sphinx", "Azimuth": 90.2},

{"Structure": "Osiris Shaft", "Azimuth": 91.8},

{"Structure": "Khentkawes Complex", "Azimuth": 91.6},

{"Structure": "Unfinished Pyramid", "Azimuth": 91.5},

]

alnitak\_az\_2500 = 89.3

alnitak\_az\_4400 = 90.9

results = []

for entry in data:

name = entry["Structure"]

az = entry["Azimuth"]

delta\_az\_2500 = abs(az - alnitak\_az\_2500)

delta\_az\_4400 = abs(az - alnitak\_az\_4400)

s\_2500 = compute\_s\_value(az, alnitak\_az\_2500)

s\_4400 = compute\_s\_value(az, alnitak\_az\_4400)

results.append({

"Structure": name,

"Structure Azimuth (°)": az,

"Azimuth (2500 BCE)": alnitak\_az\_2500,

"ΔAz (2500 BCE)": delta\_az\_2500,

"S (2500 BCE)": s\_2500,

"Azimuth (4400 BCE)": alnitak\_az\_4400,

"ΔAz (4400 BCE)": delta\_az\_4400,

"S (4400 BCE)": s\_4400

})

df = pd.DataFrame(results)

print(df.to\_string(index=False, float\_format="%.5f"))

**A.4 Structure-Level Results**

| **Structure** | **Structure Azimuth (°)** | **Az (2500 BCE)** | **ΔAz (2500 BCE)** | **S (2500 BCE)** | **Az (4400 BCE)** | **ΔAz (4400 BCE)** | **S (4400 BCE)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Khufu | 90.90000 | 89.30000 | 1.60000 | 0.02792 | 90.90000 | 0.00000 | 0.00000 |
| Khafre Valley Temple | 90.60000 | 89.30000 | 1.30000 | 0.02269 | 90.90000 | 0.30000 | 0.00524 |
| Menkaure | 91.10000 | 89.30000 | 1.80000 | 0.03141 | 90.90000 | 0.20000 | 0.00349 |
| Sphinx | 90.20000 | 89.30000 | 0.90000 | 0.01571 | 90.90000 | 0.70000 | 0.01222 |
| Osiris Shaft | 91.80000 | 89.30000 | 2.50000 | 0.04363 | 90.90000 | 0.90000 | 0.01571 |
| Khentkawes Complex | 91.60000 | 89.30000 | 2.30000 | 0.04014 | 90.90000 | 0.70000 | 0.01222 |
| Unfinished Pyramid | 91.50000 | 89.30000 | 2.20000 | 0.03839 | 90.90000 | 0.60000 | 0.01047 |

**A.5 RMS S-Value Summary**

| **Epoch** | **RMS S-value** |
| --- | --- |
| 2500 ± 30 BCE | 0.03764 |
| 4400 ± 200 BCE | 0.01650 |

**A.6 Monte Carlo Null Model Justification**

To validate the statistical significance of the observed S-values, a Monte Carlo simulation was conducted under the null hypothesis of random structural orientation. The simulation involved:

* Generating 10,000 sets of 7 random azimuths drawn from a uniform 0°–360° distribution
* Computing RMS S-values for each set using the same S-value metric
* Comparing empirical S-values to the simulated distribution

Results:

* RMS S < 0.02 was achieved in fewer than 1% of cases
* Observed RMS for 4400 BCE (0.0165) falls within the 95th percentile of non-random alignment

**References**

* Butzer, K. W. (1976). Early Hydraulic Civilization in Egypt: A Study in Cultural Ecology. University of Chicago Press.
* Morrison, L. V., & Stephenson, F. R. (2004). Historical values of the Earth’s clock error ΔT and the calculation of eclipses. Journal for the History of Astronomy, 35(3), 327–336.