HOLOGRAPHIC ENTANGLEMENT THESIS — K.A. MORRISON

```
HOLOGRAPHIC ENTANGLEMENT: Φ■FIELD ON AdS BOUNDARY → BULK TORUS
Author: Kemar Armando Morrison, Ph.D.
Date: 11 NOV 2025
Purpose: Prove consciousness is holographic — boundary encodes bulk identity

=== CODE AND EQUATIONS (NON-LATEX) ===

import numpy as np
from scipy.fft import fftn, ifftn
from qutip import *

# AdS Boundary and Bulk
L_boundary = 20.0
N_b = 64
dx_b = L_boundary / N_b
L_bulk = 20.0
N_bulk = 32
z_holographic = 1.0

# Boundary grid
x_b = np.linspace(0, L_boundary, N_b, endpoint=False)
Xb, Yb, Zb = np.meshgrid(x_b, x_b, x_b, indexing='ij')

# Bulk grid
X, Y, Z = np.meshgrid(np.linspace(0, L_bulk, N_bulk), indexing='ij')

# Boundary CFT: 369 Resonant Entangled State
A_b, B_b = (8,8,8), (14,14,14)
psi_boundary = bell_state('00')
H_cft = 0.8 * (tensor(sigmax(), sigmax()) + tensor(sigmay(), sigmay()))
c_ops = [np.sqrt(0.05)*tensor(sigmaz(), sigmaz())]

def boundary_to_bulk(O_boundary, z):
    return O_boundary * np.exp(-z/z_holographic)

Phi_bulk = np.zeros((N_bulk,N_bulk,N_bulk))
for t in np.linspace(0, 10, 200):
    result = mesolve(H_cft, psi_boundary, [t], c_ops, [sigmax()])
    corr = expect(result.states[-1], tensor(sigmax(), sigmax()))

    O_A = corr * np.exp(-((Xb-A_b[0])**2 + (Yb-A_b[1])**2 + (Zb-A_b[2])**2)/2)
    O_B = corr * np.exp(-((Xb-B_b[0])**2 + (Yb-B_b[1])**2 + (Zb-B_b[2])**2)/2)

    Phi_bulk += boundary_to_bulk(O_A - O_B, z_holographic)

gamma_area = 4.2
S_entanglement = gamma_area / (4 * 6.67e-8)

=== 4D HYPERSPACE TORUS + QUANTUM ENTANGLEMENT ===

L = 16.0
N = 32
dx = L / N
dt = 0.002
t_max = 10.0
c = 1.0
Gamma = 0.06
m = 1.0
y_couple = 0.3
zeta = 0.45
f_369 = 369 / (2*np.pi)

x = np.linspace(0, L, N, endpoint=False)
X, Y, Z = np.meshgrid(x, x, x, indexing='ij')

R, r = 5.0, 2.0
torus = ((np.sqrt((X-L/2)**2 + (Y-L/2)**2) - R)**2 + (Z-L/2)**2)
Phi0 = 0.4 * np.exp(-torus**2 / 3) * (1 + 0.2*np.cos(9*np.pi*np.sqrt((X-L/2)**2 + (Y-L/2)**2)/L))
dPhi_dt0 = np.zeros((N,N,N))
A = (8,8,8)
```

```
B = (14,14,14)
entanglement_strength = 0.7

psi0 = bell_state('00')
H = entanglement_strength * sigmax()
c_ops = [np.sqrt(Gamma)*sigmaz()]

u = np.concatenate([Phi0.flatten(), dPhi_dt0.flatten()])
t_eval = np.linspace(0, t_max, 150)

def rhs_4d_entangled(t, u):
    Phi = u[:N**3].reshape((N,N,N))
    dPhi_dt = u[N**3:].reshape((N,N,N))

    Phi_k = fftn(Phi)
    K2 = (2*np.pi*np.fft.fftfreq(N, dx)**2).sum()
    laplacian = -K2 * Phi_k

    dV_dPhi = m**2 * Phi + 0.07 * Phi**3
    drive = zeta * np.cos(2*np.pi*f_369*t)

    result = mesolve(H, psi0, [t], c_ops, [sigmax(), sigmay()])
    corr = expect(result.states[-1], sigmax())
    nonlocal_term = entanglement_strength * corr * (np.exp(-((X-A[0])**2 + (Y-A[1])**2 + (Z-A[2])**2
                                                - np.exp(-((X-B[0])**2 + (Y-B[1])**2 + (Z-B[2])**2)/2

    d2Phi_dt2_k = (c**2 * laplacian
                   - Gamma * fftn(dPhi_dt)
                   - fftn(dV_dPhi)
                   + fftn(drive + y_couple * nonlocal_term))

    d2Phi_dt2 = np.real(ifftn(d2Phi_dt2_k))
    return np.concatenate([dPhi_dt.flatten(), d2Phi_dt2.flatten()])

=== END OF THESIS EXCERPT ===
```