

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4 import matplotlib.gridspec as gridspec
5 import warnings
6 warnings.filterwarnings('ignore')
7
8 plt.rcParams['text.usetex'] = False
9 plt.rcParams['font.family'] = 'DejaVu Sans'
10 plt.rcParams['font.weight'] = 'bold'
11 plt.rcParams['axes.labelweight'] = 'bold'
12 plt.rcParams['axes.titleweight'] = 'bold'
13 plt.rcParams['figure.dpi'] = 150
14 plt.rcParams['savefig.dpi'] = 300
15 plt.rcParams['text.antialiased'] = True
16 plt.rcParams['axes.edgecolor'] = 'black'
17 plt.rcParams['axes.linewidth'] = 1.5
18
19 plt.style.use('default')
20
21 def create_god_equation_visualization():
22     fig = plt.figure(figsize=(20, 12))
23     fig.patch.set_facecolor('white')
24     fig.suptitle('The God Equations: Fundamental Laws of the Universe',
25                 fontsize=26, fontweight='bold', y=0.995, color='black')
26
27     gs = gridspec.GridSpec(3, 3, figure=fig, hspace=0.4, wspace=0.3)
28
29     ax1 = fig.add_subplot(gs[0, :])
30     ax1.axis('off')
31     einstein_eq = r'$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}$'
32     ax1.text(0.5, 0.55, einstein_eq, fontsize=26, ha='center', va='center', color='black',
33             weight='bold',
34             bbox=dict(boxstyle='round,pad=1', facecolor='lightblue', alpha=0.9, edgecolor='darkblue', linewidth=3))
35     ax1.text(0.5, 0.15, 'Einstein Field Equations\n(Geometry of Spacetime = Energy-Matter)',
36             fontsize=18, ha='center', va='center', style='italic', color='darkblue', weight='bold')
37     ax1.set_xlim(0, 1)
38     ax1.set_ylim(0, 1)
39
40     ax2 = fig.add_subplot(gs[1, 0], projection='3d')
41     visualize_spacetime_curvature(ax2)
42
43     ax3 = fig.add_subplot(gs[1, 1])
44     ax3.axis('off')
45     schrodinger_eq = r'$i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi$'
46     ax3.text(0.5, 0.7, schrodinger_eq, fontsize=28, ha='center', va='center', color='black', weight='bold',
47             bbox=dict(boxstyle='round,pad=0.8', facecolor='lightgreen', alpha=0.9, edgecolor='darkgreen', linewidth=3))
48     ax3.text(0.5, 0.3, 'Schrodinger Equation\n(Quantum Mechanics)',
49             fontsize=14, ha='center', va='center', style='italic', color='darkgreen', weight='bold')
50     ax3.set_xlim(0, 1)
51     ax3.set_ylim(0, 1)
52
53     ax4 = fig.add_subplot(gs[1, 2])
54     ax4.axis('off')
55     maxwell_eqs = [
56         r'$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}$',
57         r'$\nabla \cdot \mathbf{B} = 0$',
58         r'$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$',
59         r'$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}$'

```

```

59     ]
60     y_pos = 0.85
61     for eq in maxwell_eqs:
62         ax4.text(0.5, y_pos, eq, fontsize=13, ha='center', va='center', color='black',
weight='bold')
63         y_pos -= 0.2
64         ax4.text(0.5, 0.05, "Maxwell's Equations\n(Electromagnetism)", fontsize=14, ha=
'center', va='center',
65                 style='italic', color='darkorange', weight='bold',
66                 bbox=dict(boxstyle='round,pad=0.5', facecolor='lightyellow', alpha=0.9,
edgecolor='darkorange', linewidth=3))
67         ax4.set_xlim(0, 1)
68         ax4.set_ylim(0, 1)
69
70     ax5 = fig.add_subplot(gs[2, 0])
71     ax5.axis('off')
72     ax5.text(0.5, 0.7, r'$E = mc^2$', fontsize=32, ha='center', va='center', color=
'black', weight='bold',
73             bbox=dict(boxstyle='round,pad=0.8', facecolor='peachpuff', alpha=0.9,
edgecolor='darkorange', linewidth=3))
74     ax5.text(0.5, 0.3, 'Mass-Energy Equivalence\n(Special Relativity)',
75             fontsize=14, ha='center', va='center', style='italic', color='darkorange',
weight='bold')
76     ax5.set_xlim(0, 1)
77     ax5.set_ylim(0, 1)
78
79     ax6 = fig.add_subplot(gs[2, 1])
80     ax6.axis('off')
81     dirac_eq = r'$(i\gamma^\mu\partial_\mu - m)\psi = 0$'
82     ax6.text(0.5, 0.7, dirac_eq, fontsize=24, ha='center', va='center', color='black',
weight='bold',
83             bbox=dict(boxstyle='round,pad=0.8', facecolor='plum', alpha=0.9, edgecolor=
'purple', linewidth=3))
84     ax6.text(0.5, 0.3, 'Dirac Equation\n(Relativistic Quantum Mechanics)',
85             fontsize=14, ha='center', va='center', style='italic', color='purple',
weight='bold')
86     ax6.set_xlim(0, 1)
87     ax6.set_ylim(0, 1)
88
89     ax7 = fig.add_subplot(gs[2, 2])
90     ax7.axis('off')
91     sm_lagrangian = r'$\mathcal{L}_{SM} = \mathcal{L}_{gauge} + \mathcal{L}_{fermion} + \mathcal{L}_{Higgs} + \mathcal{L}_{Yukawa}$'
92     ax7.text(0.5, 0.7, sm_lagrangian, fontsize=12, ha='center', va='center', color=
'black', weight='bold',
93             bbox=dict(boxstyle='round,pad=0.8', facecolor='lightcyan', alpha=0.9,
edgecolor='teal', linewidth=3))
94     ax7.text(0.5, 0.3, 'Standard Model Lagrangian\n(Particle Physics)',
95             fontsize=14, ha='center', va='center', style='italic', color='teal', weight=
'bold')
96     ax7.set_xlim(0, 1)
97     ax7.set_ylim(0, 1)
98
99     return fig
100
101 def visualize_spacetime_curvature(ax):
102     u = np.linspace(-2, 2, 50)
103     v = np.linspace(-2, 2, 50)
104     U, V = np.meshgrid(u, v)
105     R = np.sqrt(U**2 + V**2)
106     Z = -2 / (R + 0.5) + 2
107     ax.plot_surface(U, V, Z, cmap='viridis', alpha=0.8, edgecolor='none', antialiased=
True)
108     ax.set_xlabel('X', fontsize=12, fontweight='bold', color='black')
109     ax.set_ylabel('Y', fontsize=12, fontweight='bold', color='black')
110     ax.set_zlabel('Spacetime Curvature', fontsize=12, fontweight='bold', color='black')
111     ax.set_title('Spacetime Curvature\n(Mass warps spacetime)', fontsize=12, fontweight=
'bold', color='black')

```

```

112     ax.view_init(elev=25, azim=45)
113     ax.set_facecolor('white')
114     ax.tick_params(colors='black')
115
116 def create_detailed_einstein_visualization():
117     fig, axes = plt.subplots(2, 2, figsize=(16, 12))
118     fig.patch.set_facecolor('white')
119     fig.suptitle('Einstein Field Equations: Detailed Breakdown', fontsize=22, fontweight=
'bold', color='black', y=0.98)
120
121     axes[0, 0].axis('off')
122     full_eq = r'$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi}{c^4}T_{\mu\nu}$'
123     axes[0, 0].text(0.5, 0.7, full_eq, fontsize=26, ha='center', va='center', color=
'black', weight='bold',
124                     bbox=dict(boxstyle='round,pad=1', facecolor='lightblue', alpha=0.9,
edgecolor='darkblue', linewidth=3))
125     axes[0, 0].text(0.5, 0.2, 'Complete Einstein Field Equations',
126                     fontsize=16, ha='center', va='center', style='italic', color=
'darkblue', weight='bold')
127     axes[0, 0].set_xlim(0, 1)
128     axes[0, 0].set_ylim(0, 1)
129
130     axes[0, 1].axis('off')
131     axes[0, 1].text(0.5, 0.9, 'LEFT SIDE: Geometry of Spacetime',
132                     fontsize=16, ha='center', fontweight='bold', color='darkblue')
133
134     terms_data = [
135         (r'$R_{\mu\nu}$', 'Ricci Curvature Tensor\n(How spacetime curves)'),
136         (r'$-\frac{1}{2}Rg_{\mu\nu}$', 'Scalar Curvature Term\n(Average curvature)'),
137         (r'$\Lambda g_{\mu\nu}$', 'Cosmological Constant\n(Dark energy)')
138     ]
139
140     y_pos = 0.7
141     for term, desc in terms_data:
142         axes[0, 1].text(0.5, y_pos, term, fontsize=20, ha='center', va='center', color=
'black', weight='bold',
143                         bbox=dict(boxstyle='round,pad=0.5', facecolor='lightblue', alpha=
0.9, edgecolor='darkblue', linewidth=2))
144         axes[0, 1].text(0.5, y_pos-0.1, desc, fontsize=11, ha='center', va='center',
145                         style='italic', color='darkblue', weight='bold')
146         y_pos -= 0.25
147     axes[0, 1].set_xlim(0, 1)
148     axes[0, 1].set_ylim(0, 1)
149
150     axes[1, 0].axis('off')
151     axes[1, 0].text(0.5, 0.9, 'RIGHT SIDE: Energy and Matter',
152                     fontsize=16, ha='center', fontweight='bold', color='darkorange')
153     stress_energy = r'$\frac{8\pi}{c^4}T_{\mu\nu}$'
154     axes[1, 0].text(0.5, 0.6, stress_energy, fontsize=22, ha='center', va='center', color
='black', weight='bold',
155                     bbox=dict(boxstyle='round,pad=0.5', facecolor='peachpuff', alpha=0.9,
edgecolor='darkorange', linewidth=2))
156     axes[1, 0].text(0.5, 0.35, r'$T_{\mu\nu}$: Stress-Energy Tensor',
157                     fontsize=13, ha='center', va='center', color='black', weight='bold')
158     axes[1, 0].text(0.5, 0.25, '(Energy, momentum, pressure, stress)',
159                     fontsize=11, ha='center', va='center', style='italic', color=
'darkorange', weight='bold')
160     axes[1, 0].text(0.5, 0.1, r'$G$: Gravitational constant',
161                     fontsize=11, ha='center', va='center', color='black', weight='bold')
162     axes[1, 0].text(0.5, 0.02, r'$c$: Speed of light',
163                     fontsize=11, ha='center', va='center', color='black', weight='bold')
164     axes[1, 0].set_xlim(0, 1)
165     axes[1, 0].set_ylim(0, 1)
166
167     axes[1, 1].axis('off')
168     axes[1, 1].text(0.5, 0.9, 'Physical Meaning', fontsize=16, ha='center', fontweight=

```

```

169     'bold', color='darkgreen')
170     meaning_text = """"Matter and energy tell spacetime how to curve,
171 and curved spacetime tells matter how to move.
172
173 • Mass/Energy warps the fabric of spacetime
174 • This warping is what we experience as gravity
175 • Planets orbit because they follow curved paths
176 • Time runs slower in stronger gravitational fields
177 • Light bends around massive objects""""
178     axes[1, 1].text(0.5, 0.45, meaning_text, fontsize=12, ha='center', va='center', color
='black', weight='bold',
179     bbox=dict(boxstyle='round,pad=1', facecolor='lightgreen', alpha=0.9,
180     edgecolor='darkgreen', linewidth=3))
181     axes[1, 1].set_xlim(0, 1)
182     axes[1, 1].set_ylim(0, 1)
183
184     plt.tight_layout()
185     return fig
186
187 def create_quantum_gravity_visualization():
188     fig, axes = plt.subplots(1, 2, figsize=(16, 6))
189     fig.patch.set_facecolor('white')
190     fig.suptitle('The Quest for Quantum Gravity: Unifying the God Equations',
191     fontsize=20, fontweight='bold', color='black', y=0.98)
192
193     axes[0].axis('off')
194     axes[0].text(0.5, 0.9, 'GENERAL RELATIVITY', fontsize=18, ha='center', fontweight=
'bold', color='darkblue')
195     gr_eq = r'$G_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}$'
196     axes[0].text(0.5, 0.7, gr_eq, fontsize=20, ha='center', va='center', color='black',
197     weight='bold', bbox=dict(boxstyle=
'round,pad=0.8', facecolor='lightblue', alpha=
0.9, edgecolor='darkblue', linewidth=3))
198     axes[0].text(0.5, 0.5, 'Describes gravity\nLarge scales\nContinuous
199 spacetime\nDeterministic',
200     fontsize=13, ha='center', va='center', color='darkblue', weight='bold')
201     axes[0].text(0.5, 0.05, '?', fontsize=40, ha='center', va='center', color='red',
202     fontweight='bold')
203     axes[0].set_xlim(0, 1)
204     axes[0].set_ylim(0, 1)
205
206     axes[1].axis('off')
207     axes[1].text(0.5, 0.9, 'QUANTUM MECHANICS', fontsize=18, ha='center', fontweight=
'bold', color='darkgreen')
208     qm_eq = r'$i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi$'
209     axes[1].text(0.5, 0.7, qm_eq, fontsize=20, ha='center', va='center', color='black',
210     weight='bold',
211     bbox=dict(boxstyle='round,pad=0.8', facecolor='lightgreen', alpha=0.9,
212     edgecolor='darkgreen', linewidth=3))
213     axes[1].text(0.5, 0.5, 'Describes quantum world\nSmall
214 scales\nDiscrete/probabilistic\nUncertainty principle',
215     fontsize=13, ha='center', va='center', color='darkgreen', weight='bold')
216     axes[1].text(0.5, 0.05, '?', fontsize=40, ha='center', va='center', color='red',
217     fontweight='bold')
218     axes[1].set_xlim(0, 1)
219     axes[1].set_ylim(0, 1)
220
221     plt.tight_layout()
222     return fig
223
224 if __name__ == "__main__":
225     print("Creating God Equation visualizations...")
226     print("=" * 50)
227
228     print("\n1. Creating main God Equations visualization...")
229     fig1 = create_god_equation_visualization()
230     plt.show()
231
232

```

```

223     print("\n2. Creating detailed Einstein breakdown...")
224     fig2 = create_detailed_einstein_visualization()
225     plt.show()
226
227     print("\n3. Creating quantum gravity visualization...")
228     fig3 = create_quantum_gravity_visualization()
229     plt.show()
230
231     print("\n" + "=" * 50)
232     print("All visualizations complete!")
233     print("Close each window to see the next visualization.")
234 if __name__ == "__main__":
235     print("Creating God Equation visualizations...")
236     print("=" * 50)
237
238     print("\n1. Creating main God Equations visualization...")
239     fig1 = create_god_equation_visualization()
240     fig1.savefig('god_equations_main.png', dpi=300, bbox_inches='tight', facecolor=
'white')
241     print("    Saved as: god_equations_main.png")
242     plt.show()
243
244     print("\n2. Creating detailed Einstein breakdown...")
245     fig2 = create_detailed_einstein_visualization()
246     fig2.savefig('einstein_detailed.png', dpi=300, bbox_inches='tight', facecolor='white'
)
247     print("    Saved as: einstein_detailed.png")
248     plt.show()
249
250     print("\n3. Creating quantum gravity visualization...")
251     fig3 = create_quantum_gravity_visualization()
252     fig3.savefig('quantum_gravity.png', dpi=300, bbox_inches='tight', facecolor='white')
253     print("    Saved as: quantum_gravity.png")
254     plt.show()
255
256     print("\n" + "=" * 50)
257     print("All visualizations complete!")
258     print("Images saved in the same folder as your script.")

```