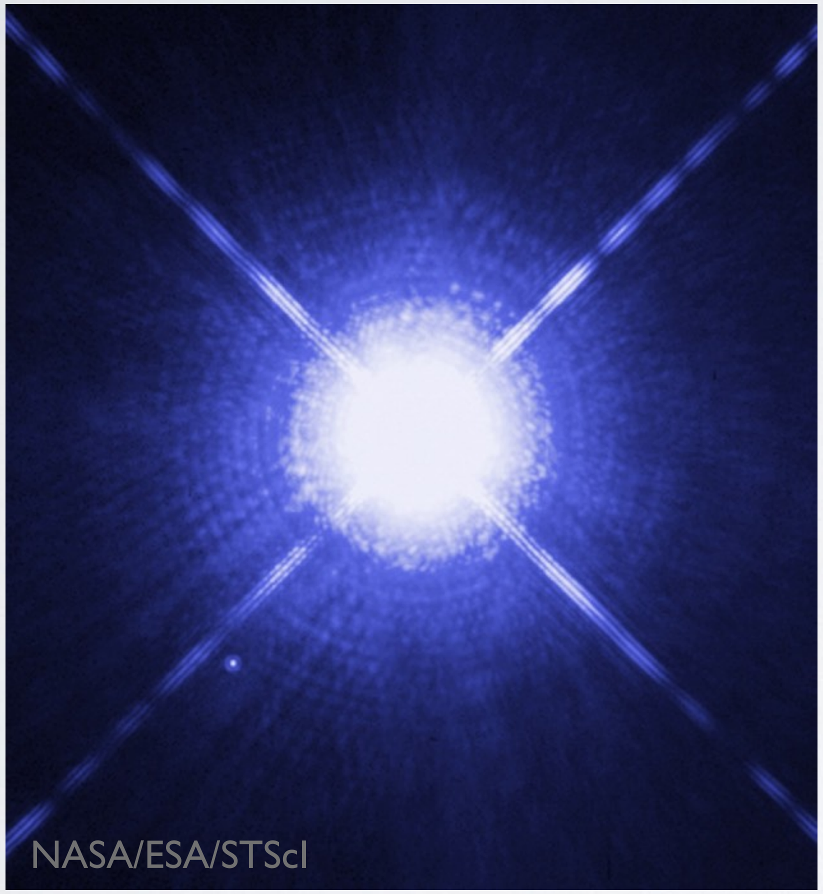
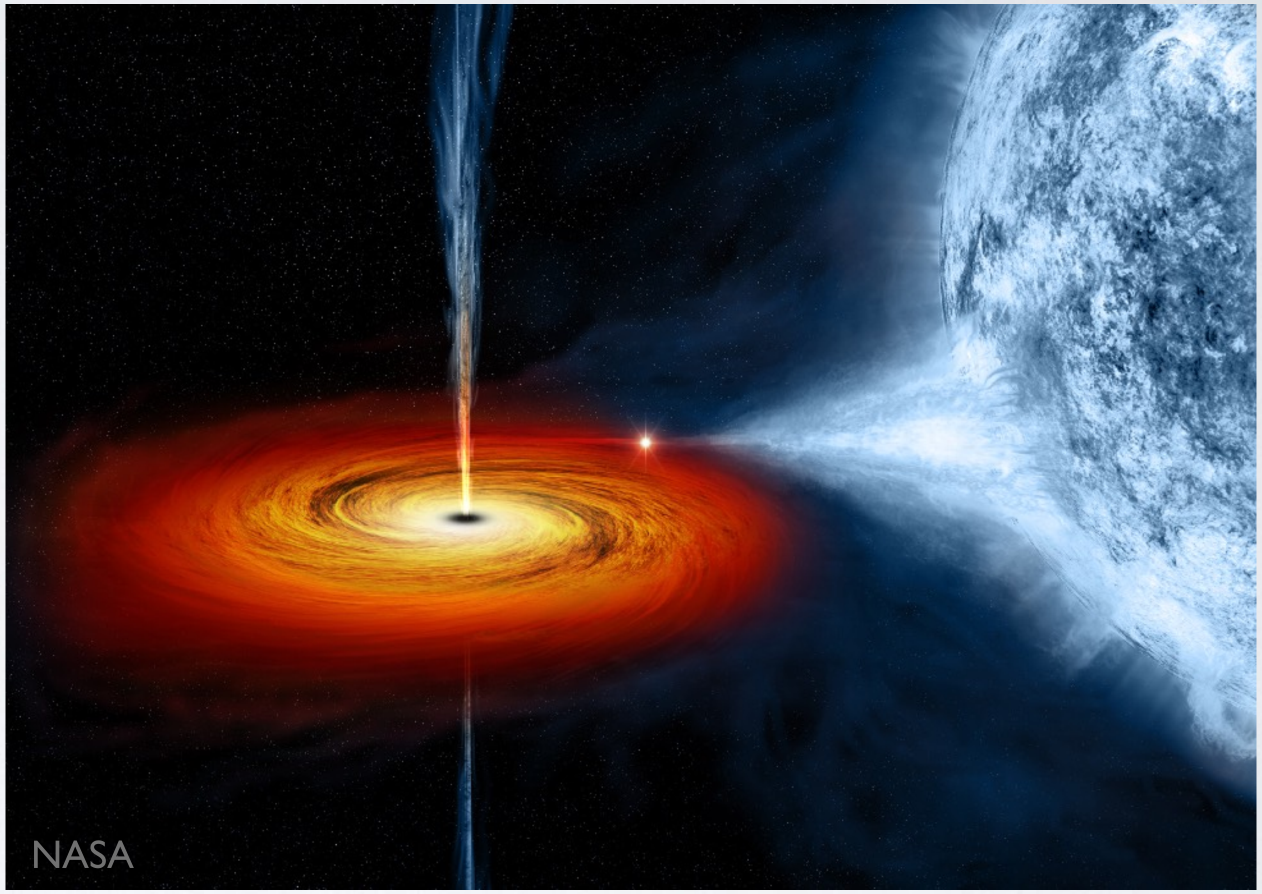
Week 12 - #1

Central Force Motion (I)





Today: Ch 7.10, 8.1-8.4

Next Class: Ch 8.5-8.7, 8.10

Ji-hoon Kim (Seoul National University)

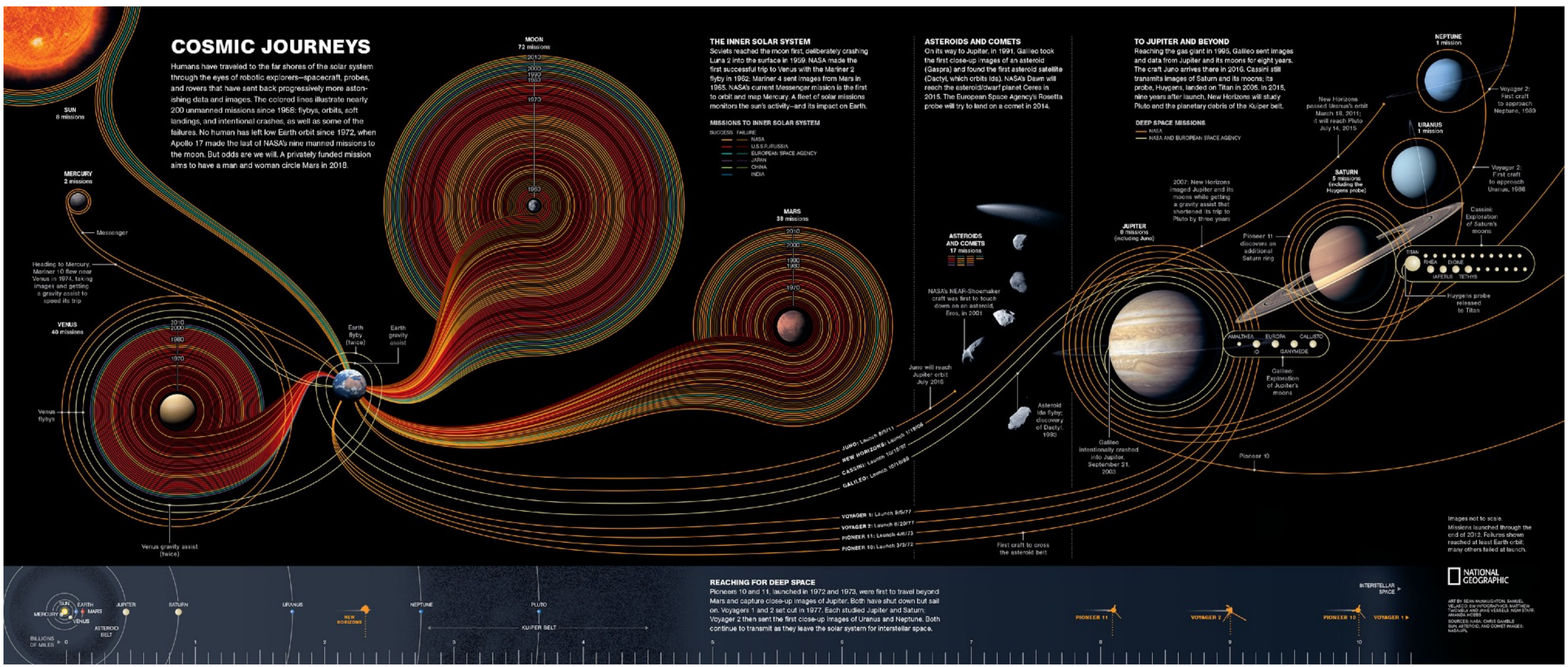
Classical Mechanics I (Spring 2026): Quiz #20

— [open book and open note, but no cellphone or laptop, submit to ETL by 23:00pm, May 31] —

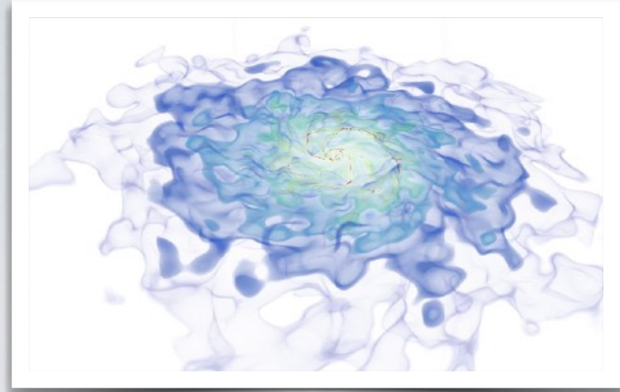
Please write down your name and student ID in the top right corner. (0.0 pt: no paper found with your name / 0.5 pt: paper found with your name and some answers / 1.0 pt: good answers)

1. Thornton & Marion, Problem 7-33.

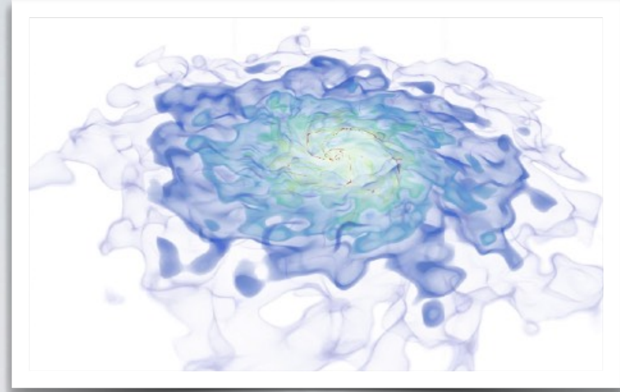
2. Thornton & Marion, Problem 7-28.



Probes sent by humankind into the solar system, National Geographic (2014)



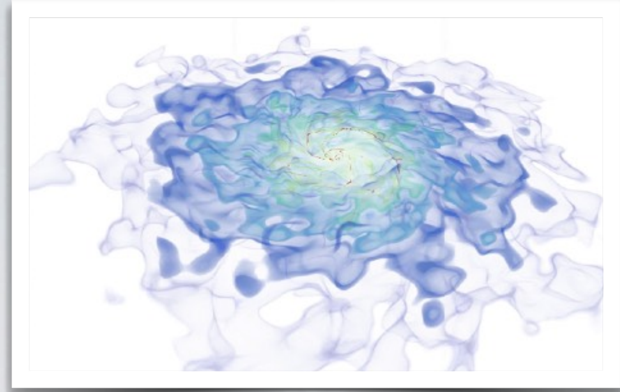
No TA Sessions In Week of 5/25



No TA Session on Jun. 3

↓

Make-up TA Session on Jun. 1 (Mon)

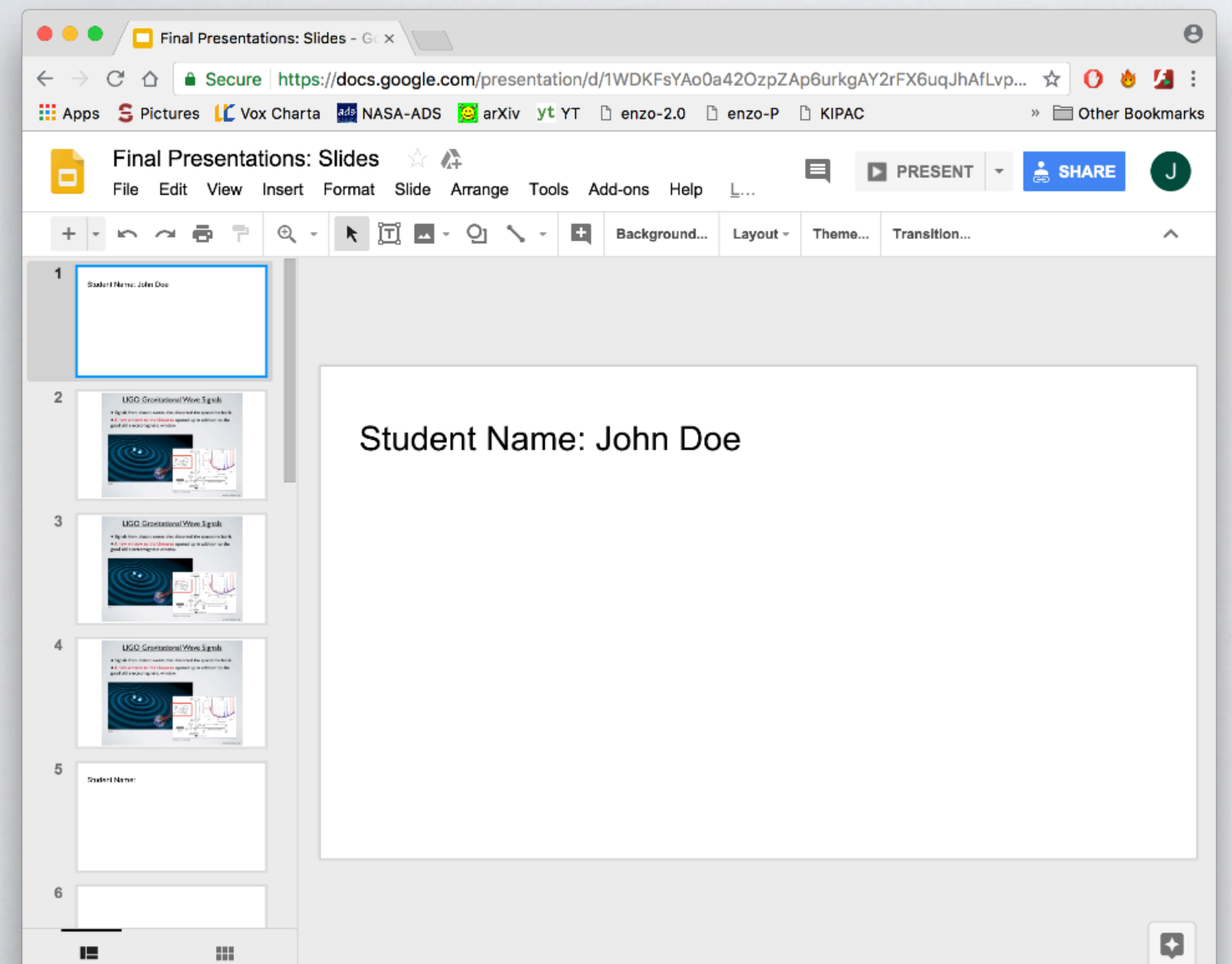


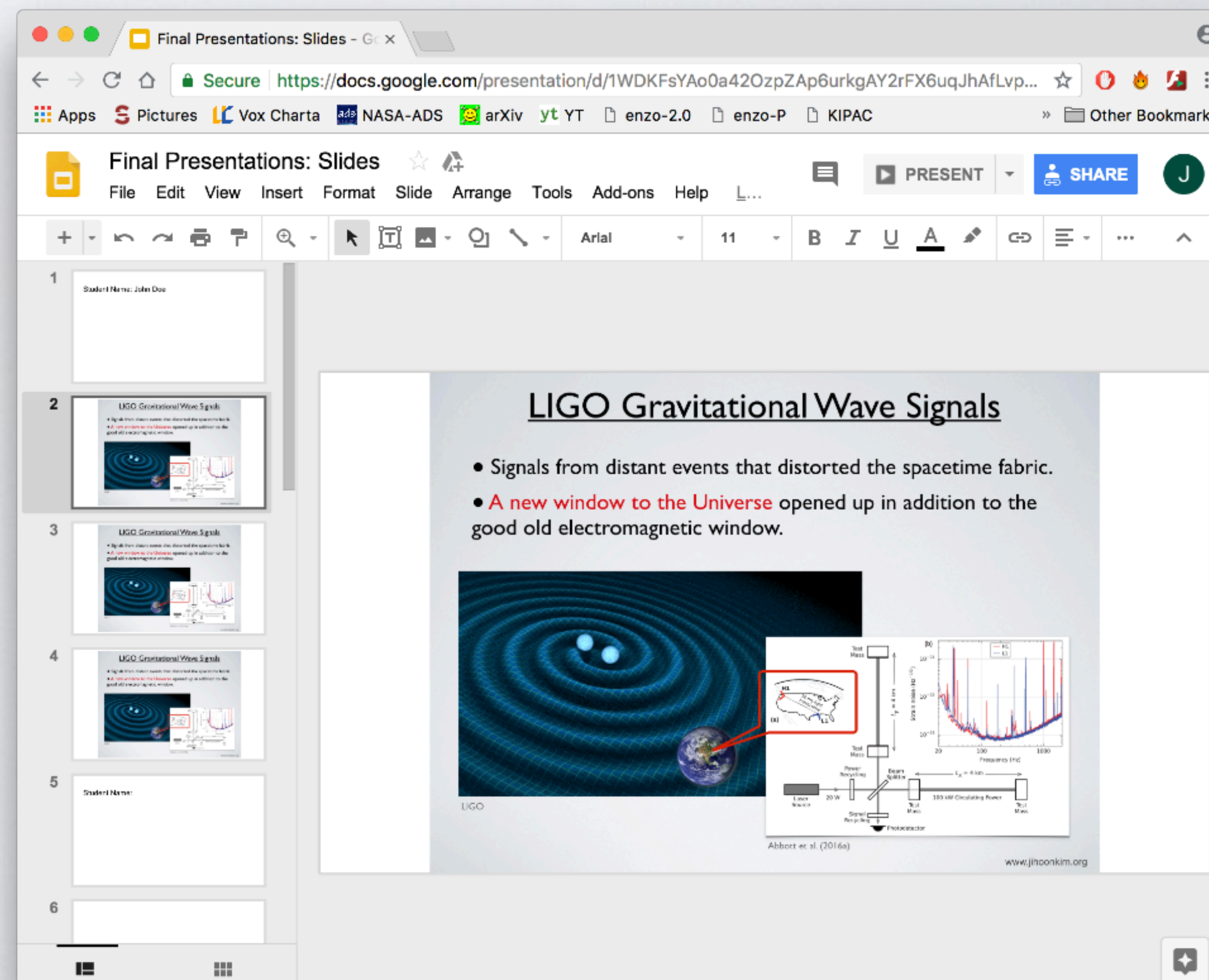
Three Deadlines
Before We Meet In Person On 6/2

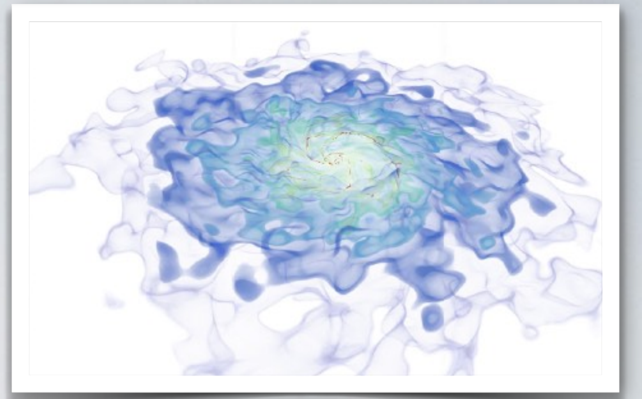
Term Project Presentation (Jun. 9)

- Drag and drop your slides as images by 23:00pm, May 31 (Sun) on a shared Google document below:

https://docs.google.com/presentation/d/17HFeWg4u1mH6Z-aXDSWxQhRqoiKSGJgDXc89C2qHRRg







HW #4 has been posted!

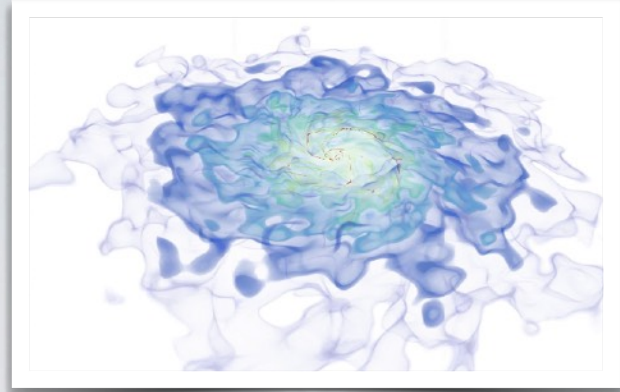
(To be posted on jihoonkim.org, Due: Jun. 1 (Mon), 23:00pm,

Grader TA this time: 서선기, supercap@snu.ac.kr)

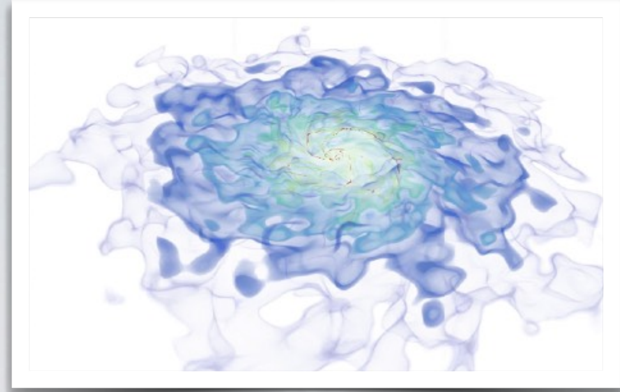
Term Project Paper

- Explore how ideas and methods from classical mechanics play a role in modern research or in everyday phenomena. Must approach the topic creatively and incorporate your own original insights.
- Due: Before the class starts on Jun. 2nd

  (10 page-limit including cover/reference; typewritten report okay)

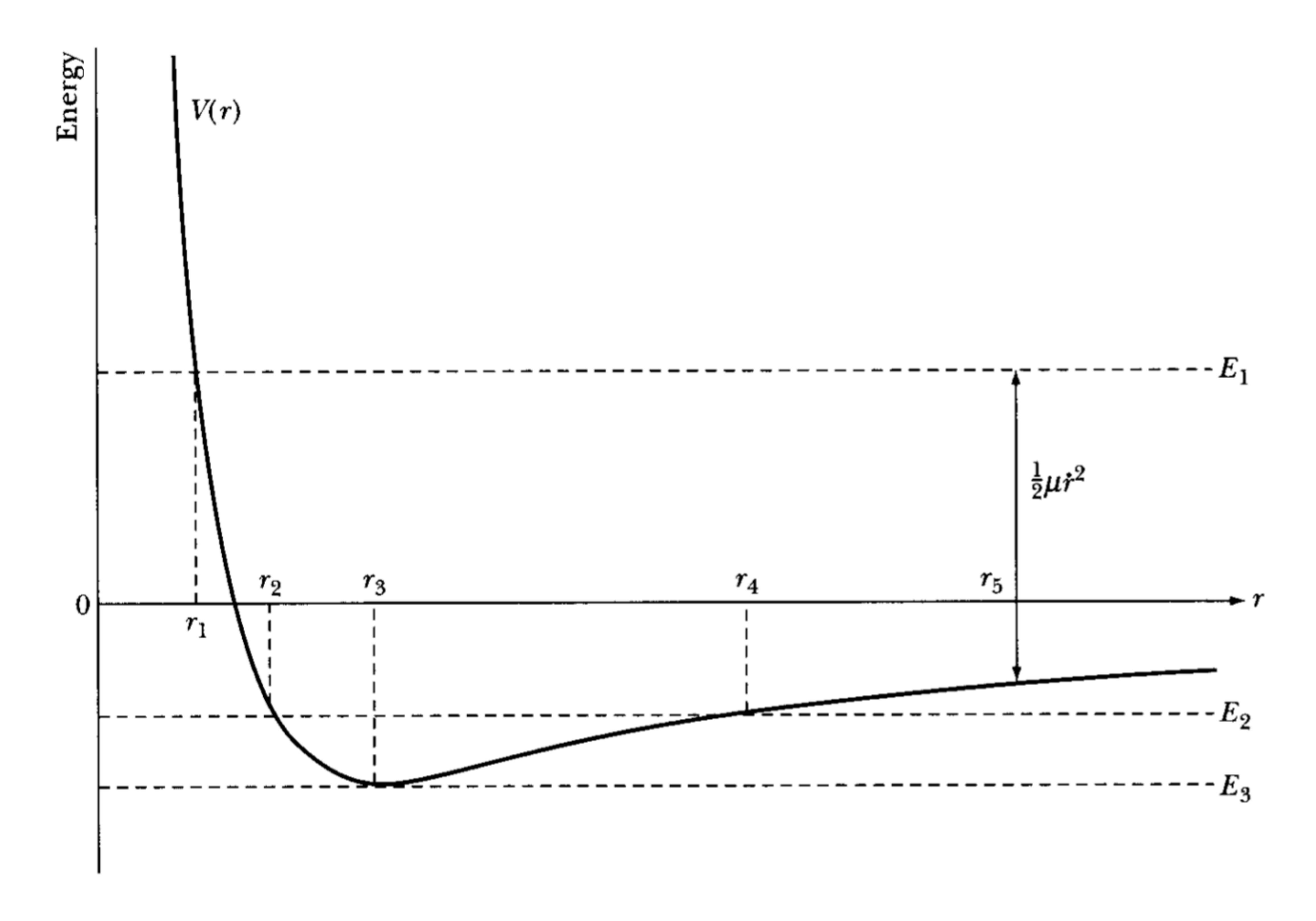


In the first page, please write down the total time you spent to prepare your report!

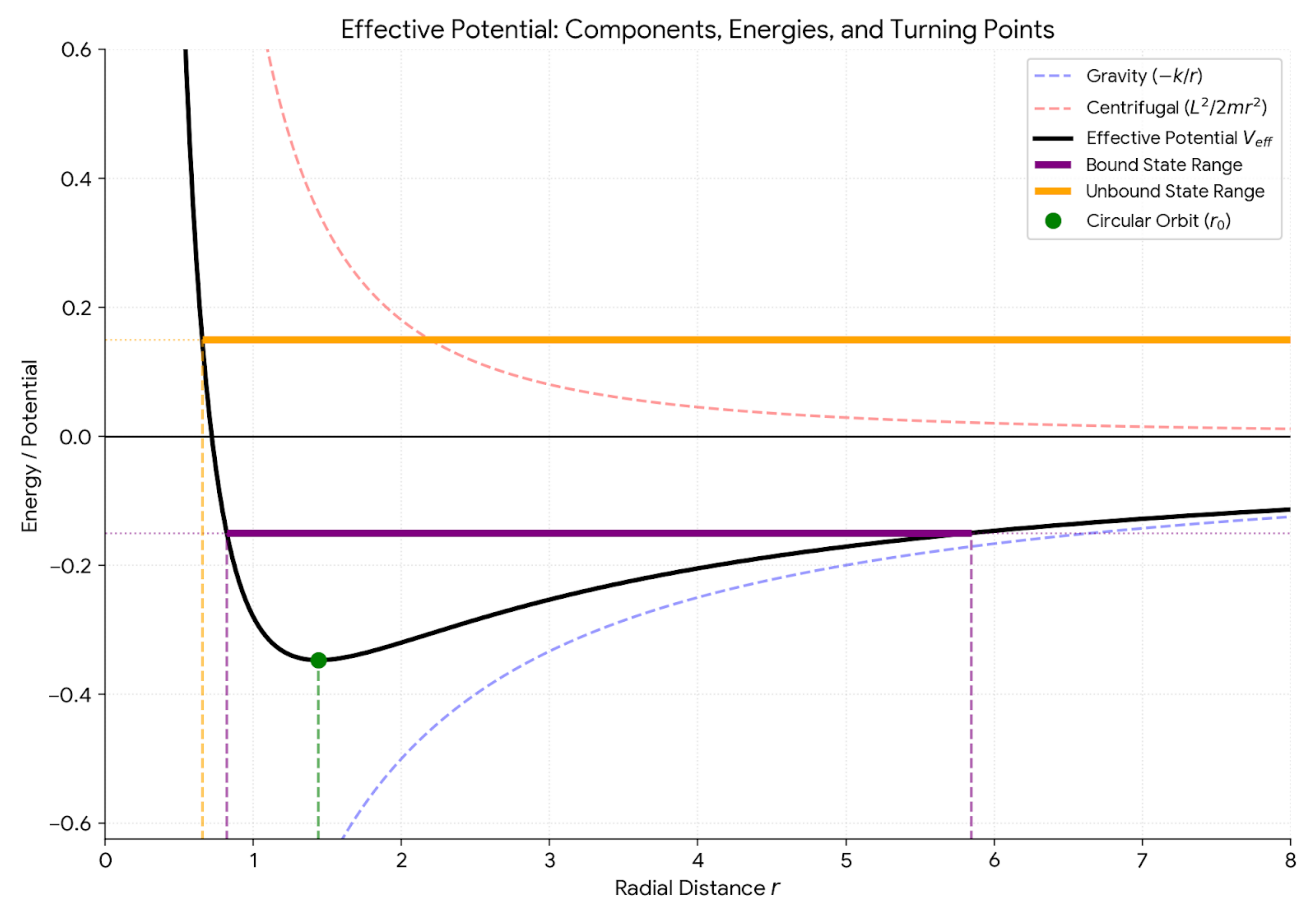


Effective Potential for Central Force

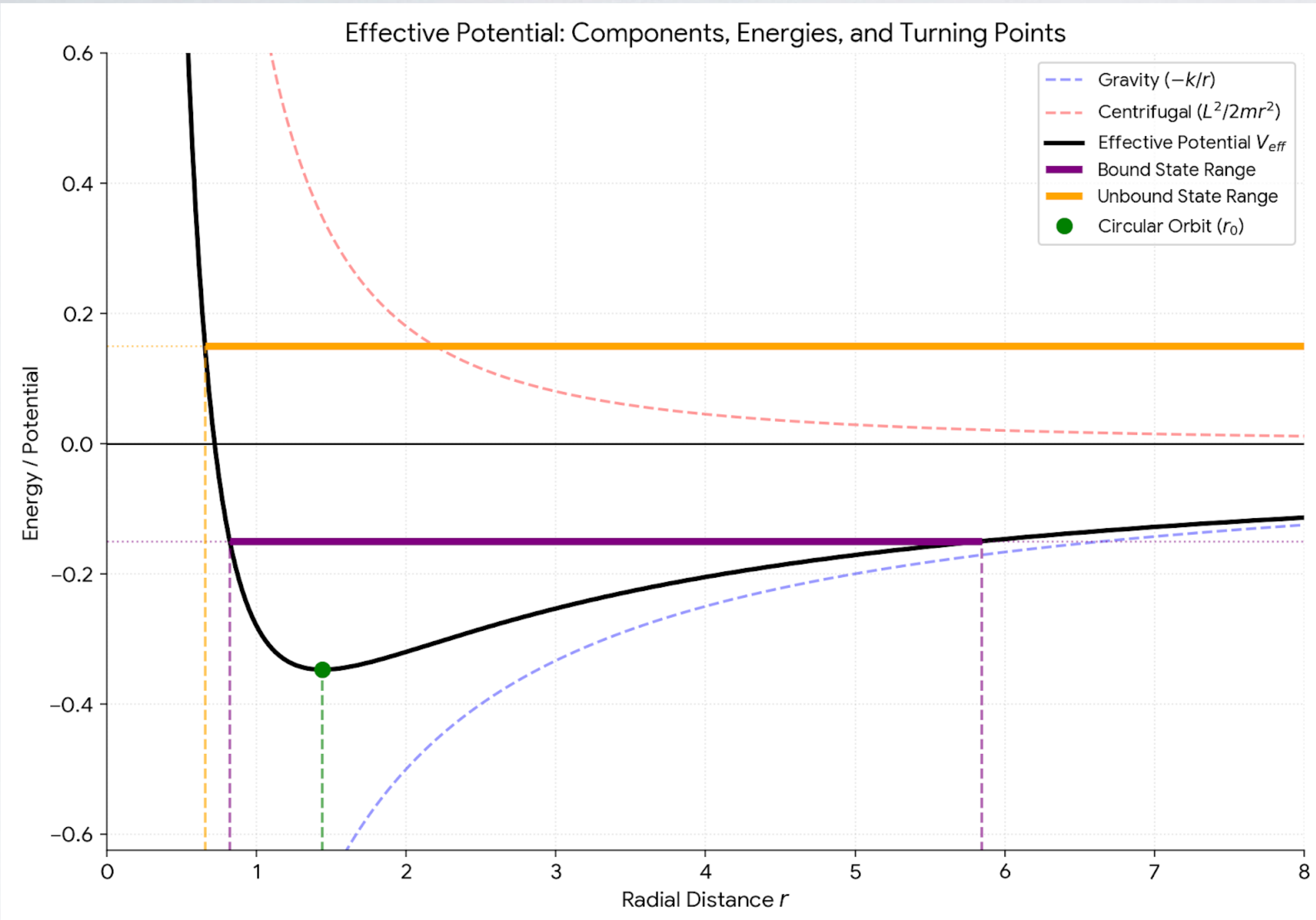
Effective Potential



Effective Potential



Effective Potential



[plots generated with vibe coding in python using Google Gemini]

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

# 1. Physics Constants
k = 1.0    # Force constant
L = 1.2    # Angular momentum
m = 1.0    # Mass

# 2. Define Potential Functions
def v_eff(r, k, L, m):
    return -k/r + L**2/(2*m*r**2)

def v_grav(r, k):
    return -k/r

def v_cent(r, L, m):
    return L**2/(2*m*r**2)

# 3. Setup Data Points
r = np.linspace(0.2, 8.0, 1000)
veff_vals = v_eff(r, k, L, m)
vgrav_vals = v_grav(r, k)
vcent_vals = v_cent(r, L, m)

# 4. Calculate Key Points
# Circular orbit (Minimum of V_eff): r0 = L^2 / (mk)
r0 = L**2 / (m * k)
v_min = v_eff(r0, k, L, m)

# Energy levels for visualization
E_bound = -0.15
E_unbound = 0.15

# Solve for turning points (2mEr^2 + 2mkr - L^2 = 0)
coeffs_bound = [2*m*E_bound, 2*m*k, -L**2]
roots_bound = np.sort(np.roots(coeffs_bound))
rmin_b, rmax_b = roots_bound

coeffs_unbound = [2*m*E_unbound, 2*m*k, -L**2]
roots_unbound = np.roots(coeffs_unbound)
rmin_u = np.max(roots_unbound[roots_unbound > 0])

# 5. Plotting
plt.figure(figsize=(10, 7))

# Components (Dashed lines)
plt.plot(r, vgrav_vals, '--', color='blue', alpha=0.4, label='Gravity ($-k/r$)')
plt.plot(r, vcent_vals, '--', color='red', alpha=0.4, label='Centrifugal ($L^2/2mr^2$)')

# Total Effective Potential (Solid black line)
plt.plot(r, veff_vals, 'k', linewidth=2.5, label='Effective Potential $V_{eff}$')

# Bound State Energy and Range
plt.axhline(E_bound, color='purple', linewidth=1, linestyle=':', alpha=0.5)
plt.hlines(E_bound, rmin_b, rmax_b, colors='purple', linewidth=4, label='Bound State Range')
plt.vlines([rmin_b, rmax_b], v_min*2, E_bound, colors='purple', linestyles='--', alpha=0.6)

# Unbound State Energy and Range
plt.axhline(E_unbound, color='orange', linewidth=1, linestyle=':', alpha=0.5)
plt.hlines(E_unbound, rmin_u, 8.0, colors='orange', linewidth=4, label='Unbound State Range')
plt.vlines(rmin_u, v_min*2, E_unbound, colors='orange', linestyles='--', alpha=0.6)

# Circular Orbit Marker
plt.plot(r0, v_min, 'go', markersize=8, label='Circular Orbit ($r_0$)')
plt.vlines(r0, v_min*2, v_min, colors='green', linestyles='--', alpha=0.6)

# 6. Aesthetics and Labels
plt.title('Complete Effective Potential Diagram ($F = -k/r^2$)', fontsize=14)
plt.xlabel('Radial Distance $r$', fontsize=12)
plt.ylabel('Energy / Potential', fontsize=12)
plt.ylim(v_min * 1.8, 0.6)
plt.xlim(0, 8)
plt.grid(True, which='both', linestyle=':', alpha=0.3)
plt.axhline(0, color='black', linewidth=1)
plt.legend(loc='upper right', frameon=True, shadow=True)

plt.tight_layout()
plt.show()
```