Exercise 1: gas plus radiation

A mixture of ionized hydrogen gas and radiation (such as at the center of the Sun) with a density ρ and temperature T has a pressure

$$P = \frac{2\rho kT}{m_p} + \frac{1}{3}aT^4$$

where $k = 1.38 \times 10^{-16}$ erg K^{-1}, $a = 7.56 \times 10^{-15}$ erg $cm^{-3}K^{-4}$, and $m_p = 1.67 \times 10^{-24}$ g.

Find the temperature corresponding to $\rho = 80 \text{ g cm}^{-3}$ and $P = 1.3 \times 10^{18} \text{ dyn cm}^{-2}$ using Brent's method. What are good starting guesses? A lower bound on the temperature is easy, but an upper bound requires more thought. You can find a strict upper limit by noting that since the temperature is always positive, the combined ideal gas and radiation pressure is always larger than either individual term alone.

(Aside: if you're wondering where the (2) came from, the ideal gas pressure has assumed pure H composition so that the mean mass per particle is $m_p/2$.)

In [18]:
```
### CODE HERE
import numpy as np
import scipy.optimize as opt
k = 1.38 * 10**-16
a = 7.56 * 10**-15
m_p = 1.67 * 10**-24
rho = 80
P = 1.3 * 10 **18
def pressure(T):
    gas_pressure = 2 * rho * k * T/m_p
    rad_pres = (1/3) * a * T**4
    return gas_pressure + rad_pres - P
T_high = (3 * P/a) **0.25 #since it was to the power of 4 , and the multiply by 3 s
T_root = opt.brentq(pressure,0,T_high)
print(f"The temperature at the given pressure is {T_root:.2f} K")
```

```
The temperature at the given pressure is 87269196.38 K
```

Exercise 2: Moon missile

A projectile is launched from the surface of the Moon with a speed $v_0 = 10^5 \text{ cm s}^{-1}$ at an angle $\alpha = 45^\circ$ from the surface. This is not enough to achieve orbit, so the projectile falls back to the Moon's surface. Ignoring the Moon's rotation, what is the angle θ_0 subtended by half of the trajectory?

Orbital parameters: (compute from given data)

$$E_0 = \frac{1}{2}v_0^2 - \frac{GM}{R}$$

$$a = -\frac{GM}{2E_0}$$

$$J_0 = Rv_0 \cos\alpha$$

$$e = \left(1 + \frac{2E_0 J_0^2}{G^2 M^2}\right)^{1/2}$$

Ellipse equation: (solve with SciPy)

$$\frac{r^2 \sin^2\theta}{a^2(1-e^2)} + \frac{(r\cos\theta - ae)^2}{a^2} = 1$$

Physical data:

$$M = 7.348 \times 10^{25}\ \mathrm{g} \quad R = 1.737 \times 10^{8}\ \mathrm{cm} \quad G = 6.673 \times 10^{-8}\ \mathrm{cm}^3\,\mathrm{s}^{-2}\,\mathrm{g}^{-1}$$

> $r = R$ at launch and impact, so θ is the only unknown

In [19]:
```python
### CODE HERE
G = 6.673 *10**-8
M = 7.348 * 10**25
R = 1.737 *10**8
v0 = 10**5
alp = np.radians(45) #converting to radians

E0 = 0.5 * v0 **2 -G*M/R
a = -G*M/(2*E0)
J0 = R*v0 * np.cos(alp)
e = (1 + 2 * E0 * J0**2 /(G**2 * M**2))

def ellipse_eq(theta):
    return R **2 * np.sin(theta) **2 /(a**2 * (1 - e**2)) + (R * np.cos(theta) - a

# using bisect method to find sol since it has low cost
sol = opt.bisect(ellipse_eq, 0, np.pi/2)
sol
print(f"The angle of half of the trajectory is {np.degrees(sol):.10f} degrees.")
```

```
The angle of half of the trajectory is 10.1618410681 degrees.
```

Exercise 3: a nonlinear system of equations

Using `scipy.optimize.root`, solve the system of equations

$$9x^2 + 36y^2 + z^2 - 36 = 0$$

$$x^2 - 2y^2 - 20z = 0$$

$$x^2 - y^2 + z^2 = 0$$

for (x, y, z). Good starting guesses are $(\pm 1, \pm 1, 0)$. There are four roots. Try using an error tolerance of 10^{-10}.

Check carefully that you are actually getting four distinct roots – I had to tweak one of the starting guesses in order to pull out the fourth root.

In [20]:
```
### CODE HERE
def eq(var):
    x,y,z = var
    return np.array([ 9*x**2 + 36*y**2 + z**2 - 36, x**2 - 2*y**2 - 20*z, x**2 - y*
init_guess = np.array([
    [1,1,0],
    [-1,1,0],
    [1,-1,0],
    [-1,-1,0]])
sols = []
for guess in init_guess:
    sol = opt.root(eq,guess,method='broyden1')
    if sol.success:
        if not any(np.allclose(sol.x, k, atol=1e-6) for k in sols):
            sols.append(sol.x)
            print(f"found root {sol.x}")
        else:
            print(f" {sol.x} is the multiple root")
            print("Multiple roots found")
    else:
        print(" Found no sol")
```

```
found root [ 0.89368765  0.89458708 -0.04009481]
found root [-0.89368825  0.89458691 -0.0400946 ]
found root [ 0.89368866 -0.89458676 -0.04009449]
 [-0.89368876  0.89458673 -0.04009447] is the multiple root
Multiple roots found
```