Assignment 5

Course on Physics of Compact Objects

Instructor: Prof. P. Ajith

Author: Md Arif Shaikh, ICTS-TIFR, Feb 15, 2021

Problem 2

Numerically solve the stellar structure equations

$$\frac{dm}{dr} = 4\pi r^2 \rho, \frac{dP}{dr} = -\frac{Gm\rho}{r^2}$$

assuming the equation of state of non-relativistic degenerate electron gas *including Coulomb corrections*:

$$P(\rho) = K\rho^{5/3}\left[1 - (\rho_0/\rho)^{1/3}\right],$$

where $K \simeq 10^{13}\mu_e^{-5/3}$ (cgs units) and $\rho_0 \simeq 0.4Z^2\mu_e$ g/cm^3. Assume the atomic numeber to be $Z = 6$ (Carbon) an the mean molecular weight per free electron to be $\mu_e = 2$. The obvious boundary conditions are $m(r = 0) = 0$ and $P(r = 0) = P(\rho_c)$, where ρ_c is the central density. Compute the mass-radius of white dwarfs for central densities ranging $\rho_c = 10^6 - 10^9$ g/cm^3 . Compare the mass-radius relation with the same computed earlier neglecting Coulomb corrections.

```
using DifferentialEquations, CairoMakie, StaticArrays
```

```
3.2
```

```
begin
    const G = 6.6743 * 1e-8
    const μe = 2.0
    const K = 1e13 * μe^(-5.0 / 3.0)
    const Z = 2.0
    const ρ0 = 0.4 * Z^2 * μe
end
```

```
stellarStructure (generic function with 1 method)
```

```
function stellarStructure(u, params, r)
    m, ρ = u
    ρ0, G, K = params
    dmdr = 4.0 * pi * r^2 * abs(ρ)
    dρdr = - (3.0 * G * m * abs(ρ)^(1.0 /3.0 )) / (K * r^2 * (5.0 - 4.0 * abs((ρ0 /
ρ))^(1.0 / 3.0)))
    @SVector [dmdr, dρdr]
end
```

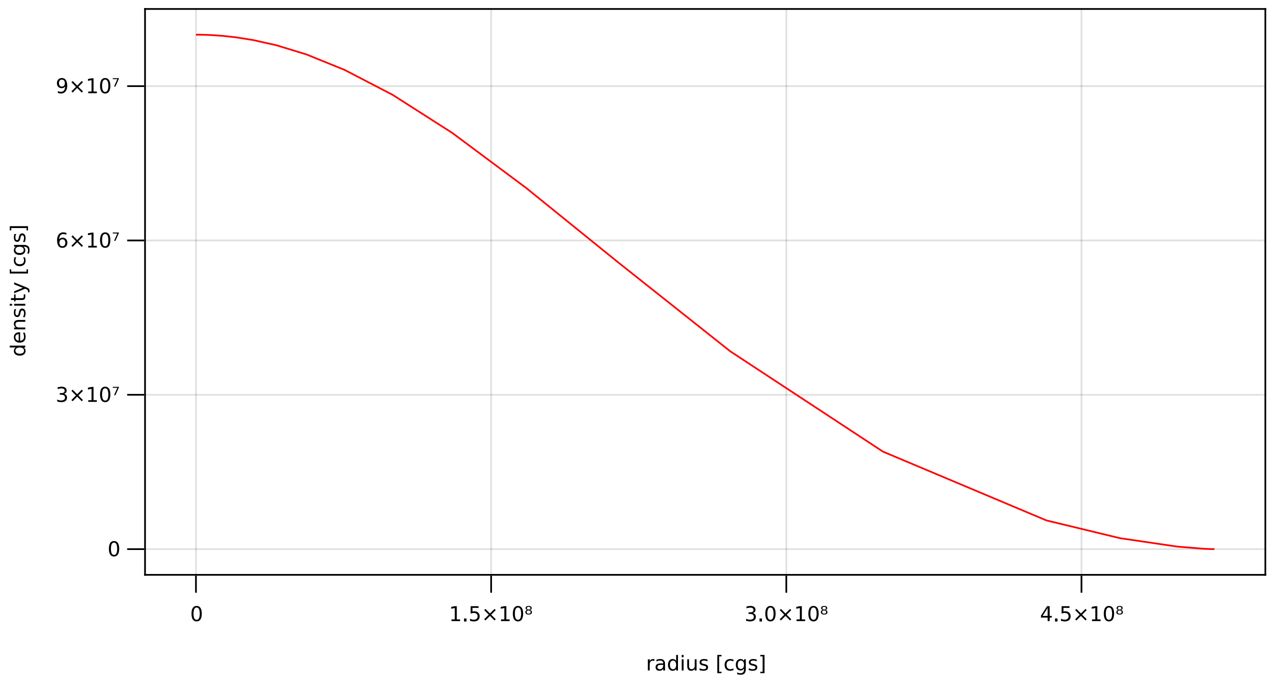
```
▸ ContinuousCallback{typeof(condition),typeof(affect!),typeof(affect!),typeof(DiffEqBase.IN
```

```
begin
    condition(u, r, integrator) =  u[2] - ρ0;
    affect!(integrator) = terminate!(integrator);
    cb = ContinuousCallback(condition, affect!);
end
```

	timestamp	value1	value2
1	1.0e-7	0.0	1.0e8
2	0.000426781	0.0325614	1.0e8
3	0.00469359	43.3115	1.0e8
4	0.0473617	44500.9	1.0e8
5	0.474042	4.46211e7	1.0e8
6	4.74085	4.46331e10	1.0e8
7	47.4089	4.46343e13	1.0e8
8	474.09	4.46344e16	1.0e8
9	4740.9	4.46344e19	1.0e8
10	28779.3	9.98454e21	1.0e8
⋮ more			

```
begin
    rspan = [1e-7, 1e10]
    params = [ρ0, G, K]
    ρc = 1e8

    u0 = @SVector [0.0, ρc]
    problem = ODEProblem(stellarStructure, u0, rspan, params)
    solution = solve(problem, callback = cb)
end
```




```
begin
    fig = Figure(resolution = (800, 450), fonts = "Times Roman", fontsize = 12)
    ax = fig[1, 1] = Axis(fig, xlabel = "radius [cgs]", ylabel = "density [cgs]")
    lines!(solution.t, solution[2, :], color = :red)
    fig
end
```

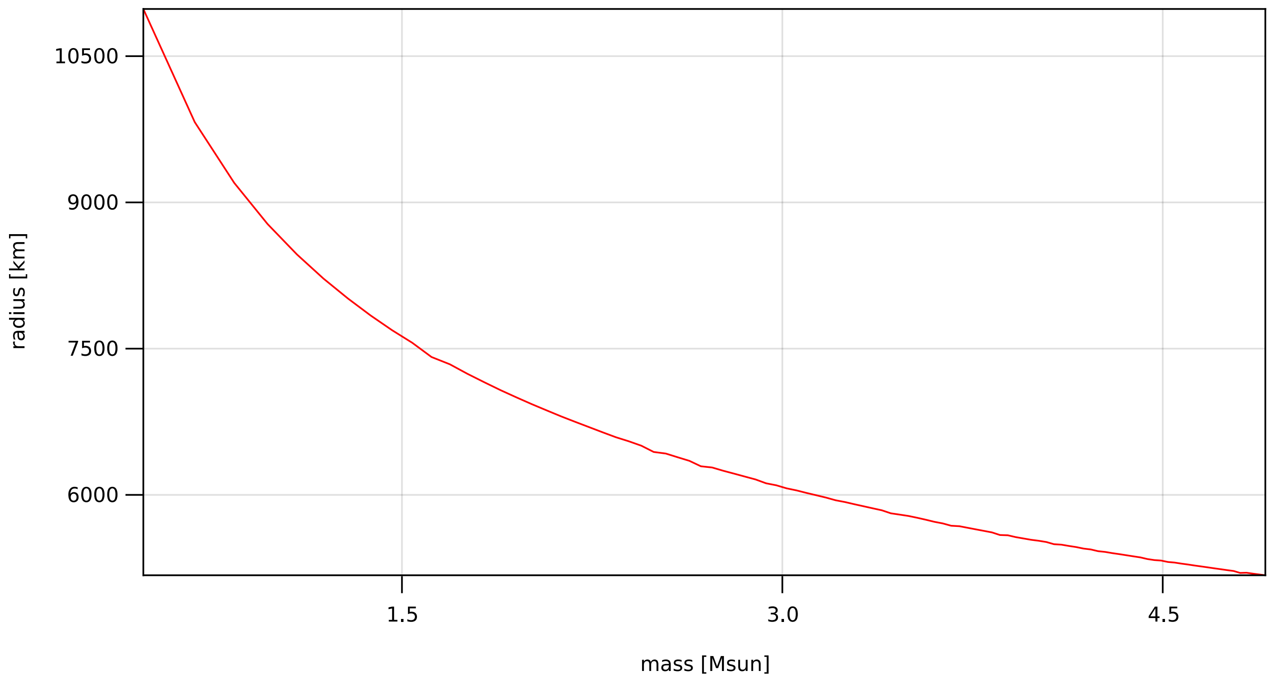
```
begin
    masses = Float64[]
    radii = Float64[]
    ρcs = LinRange(1.e6, 1.e8, 100)
    for ρc in ρcs
        u0 = @SVector [0.0, ρc]
        problem = ODEProblem(stellarStructure, u0, rspan, params)
        sol = solve(problem, callback = cb)
        push!(radii, sol.t[end])
        push!(masses, sol[1, :][end])
    end
end
```

Msun = 1.989e33

```
Msun = 1.989 * 1e33
```

Rsun = 6.9634e10

```
Rsun = 696340 * 1e5
```




```
begin
    figMR = Figure(resolution = (800, 450), fonts = "Times Roman", fontsize = 12)
    axMR = figMR[1, 1] = Axis(figMR,
        ylabel = "radius [km]",
        xlabel = "mass [Msun]")
    lines!(masses / Msun, radii / 1e5, color = :red)
    xlims!(axMR, [masses[1], masses[end]]/Msun)
    ylims!(axMR, [radii[end], radii[1]]/1e5)
    figMR
end
```