8.962 Pset 10 Solutions

Problem 2

■ b.

We define the constants for the problem:

```
In[1]:=  Γ = 5 / 3; K = 7.32; ρ0c = 7.42*^-4; P0c = K * ρ0c ^ Γ;
```

It will be convenient to express quantities involving ρ entirely in terms of P, so here we solve the equation of state for $\rho 0$ and ρ in terms of P:

```
In[2]:=  ρ0 = Function[P, Evaluate[(P / K) ^ (1 / Γ)]]

Out[2]=  Function[P, 0.302896 P^(3/5)]

In[3]:=  ρ = Function[P, Evaluate[ρ0[P] + P / (Γ - 1)]]

Out[3]=  Function[P, 0.302896 P^(3/5) + 3 P / 2]
```

We are going to have trouble with the 1/r term in the dP/dr equation (not real, physical trouble, but instead numerical trouble---at r = 0, dP/dr = 0/0). So, we will actually start the integration at some small r = ϵ ~ 10^-8, using the first-order taylor series solution to extend our initial condition: m[ϵ] = m[0] + m'[0]*ϵ = m[0], P[ϵ] = P[0] + P'[0]*ϵ = P[0].

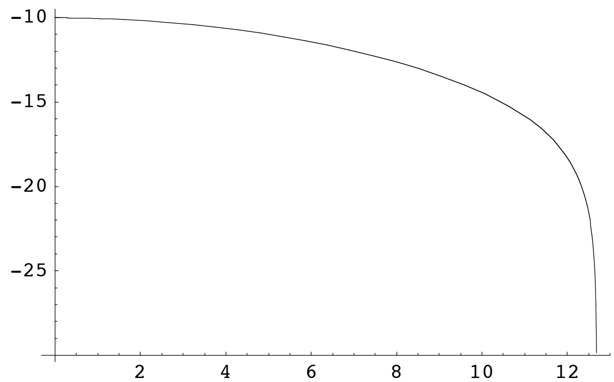
Here are the equations, solved (we choose ϵ=10^-8).

```
In[4]:=  sol = With[{r0 = 1*^-8}, NDSolve[{m'[r] == 4 Pi ρ[P[r]] r^2,
              P'[r] == -(ρ[P[r]] + P[r]) (m[r] + 4 Pi r^3 P[r]) / (r (r - 2 m[r])),
              P[r0] == P0c, m[r0] == 0}, {P, m}, {r, r0, 20}]]

Out[4]=  {{P → InterpolatingFunction[{{1. × 10^-8, 20.}}, <>],
           m → InterpolatingFunction[{{1. × 10^-8, 20.}}, <>]}}
```

Examining the solution, we see that there is a problem when r ~ 12.70:

```
In[6]:= Plot[Log[P[r] /. sol[[1]]], {r, 0, 12.70}]
```

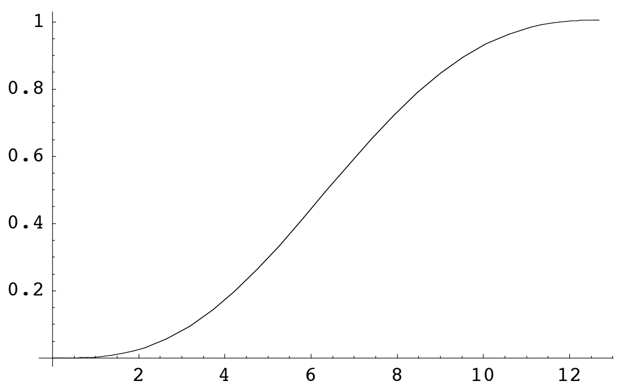
Plot::plnr : Log[P[r] /. sol⟦1⟧] is not a machine-size real number at r = 12.699999470833333` . More…




```
Out[6]= - Graphics -
```

```
In[7]:= Plot[m[r] /. sol[[1]], {r, 0, 12.70}]
```

Plot::plnr : m[r] /. sol⟦1⟧ is not a machine-size real number at r = 12.699999470833333` . More…




```
Out[7]= - Graphics -
```

We can find the exact r where P[r] == 0:

```
In[8]:= FindRoot[P[r] /. sol[[1]], {r, 12.5, 12.6}]

Out[8]= {r → 12.6885}

In[9]:= rStar = r /. %

Out[9]= 12.6885
```

And the mass at that point:

```
In[10]:= mStar = Re[m[rStar] /. sol[[1]]]

Out[10]= 1.00509
```

(Because of numerical inaccuracies in the evolution which occurr after r = rStar, mStar has a 0*i complex part which we have removed using Re[...].)

■ d.

We have

```
In[11]:= mP = Re[NIntegrate[4 Pi ρ[P[r]] r^2 (1 - 2 m[r] / r) ^ (-1 / 2) /. sol[[1]], {r, 0, rStar}]]

Out[11]= 1.08501
```

We see that 1.08501 - 1.00509 = 0.079925 km of mass are accounted for by the gravitational binding energy.

```
In[12]:= (mP - mStar) / mStar

Out[12]= 0.0795204
```

Problem 3

■ a.

Note: we reset the Kernel here, so that old definitions do not contanimate the calculations. Below we duplicate the setup of Problem 2, but ρ0c is now an array of values.

```
In[1]:= Γ = 5 / 3; K = 7.32;
        ρ0c = 7.42 {10^ (-5), 10^ (-4), 10^ (-3), 10^ (-2), 10^ (-1)}; P0c = K * ρ0c^Γ;

In[2]:= ρ0 = Function[P, Evaluate[(P / K) ^ (1 / Γ)]]

Out[2]= Function[P, 0.302896 P^(3/5)]

In[3]:= ρ = Function[P, Evaluate[ρ0[P] + P / (Γ - 1)]]

Out[3]= Function[P, 0.302896 P^(3/5) + 3 P/2]
```

Now we solve the TOV equations for the various values of ρ0c (starting from a small value of r0 = 10^-8.

```
In[4]:= r0 = 1*^-8; r1 = 36;
```

r1, above, was chosen with some "feedback" so that every solution reached the maximum number of steps *before* reaching r1. The idea is that the integrator will be taking very small steps to integrate the solutions past the point at which P = 0, so by extending the region so that every region of integration includes this point, we can be sure that we've captured the transition to P = 0 accurately for each central mass.

```
In[5]:= sols = Table[NDSolve[{m'[r] == 4 Pi ρ[P[r]] r^2,
            P'[r] == -(ρ[P[r]] + P[r]) (m[r] + 4 Pi r^3 P[r]) / (r (r - 2 m[r])),
            P[r0] == P0c[[i]], m[r0] == 0}, {P, m}, {r, r0, r1}][[1]], {i, 1, Length[P0c]}]

        NDSolve::mxst : Maximum number of 10000 steps reached at the point r == 35.15360439385316` . More…

        NDSolve::mxst : Maximum number of 10000 steps reached at the point r == 20.564551231693855` . More…

        NDSolve::mxst : Maximum number of 10000 steps reached at the point r == 12.192785315996783` . More…

        General::stop : Further output of NDSolve::mxst will be suppressed during this calculation. More…

Out[5]= {{P → InterpolatingFunction[{{1.×10^-8, 35.1536}}, <>],
          m → InterpolatingFunction[{{1.×10^-8, 35.1536}}, <>]},
         {P → InterpolatingFunction[{{1.×10^-8, 20.5646}}, <>],
          m → InterpolatingFunction[{{1.×10^-8, 20.5646}}, <>]},
         {P → InterpolatingFunction[{{1.×10^-8, 12.1928}}, <>],
          m → InterpolatingFunction[{{1.×10^-8, 12.1928}}, <>]},
         {P → InterpolatingFunction[{{1.×10^-8, 11.1016}}, <>],
          m → InterpolatingFunction[{{1.×10^-8, 11.1016}}, <>]},
         {P → InterpolatingFunction[{{1.×10^-8, 12.575}}, <>],
          m → InterpolatingFunction[{{1.×10^-8, 12.575}}, <>]}}
```

We can already see that R_* has what appears to be a dip, so it looks like something interesting is going on.

Here we compute the rStar for each solution.

```
In[6]:= rStars = Table[FindRoot[(P[r] /. sols[[i]]) == 0, {r, 3, 4}], {i, 1, Length[P0c]}]

        FindRoot::nlnum :
         The function value {InterpolatingFunction[{{1.×10^-8, 35.1536}}, «3», {Automatic}][«19» - «1»]}
           is not a list of numbers with dimensions {1} at {r} = {20.7324 - 1.12517×10^-13 i}. More…

        FindRoot::nlnum :
         The function value {InterpolatingFunction[{{1.×10^-8, 20.5646}}, «3», {Automatic}][«1»]} is
           not a list of numbers with dimensions {1} at {r} = {12.6827 - 2.15303×10^-13 i}. More…

        FindRoot::nlnum : The function value {InterpolatingFunction[{{1.×10^-8, 12.575}}, «3», {Automatic}][«1»]}
           is not a list of numbers with dimensions {1} at {r} = {5.94891 - 2.75297×10^-11 i}. More…

        General::stop : Further output of FindRoot::nlnum will be suppressed during this calculation. More…

Out[6]= {{r → 20.7324}, {r → 12.6827}, {r → 6.66337}, {r → 4.61717}, {r → 5.94891}}
```

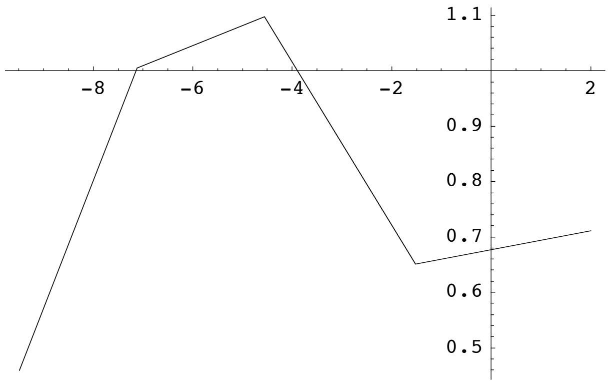
And here we compute mStar for each solution:

```
In[7]:= mStars = Table[Re[m[r /. rStars[[i]]] /. sols[[i]]], {i, 1, Length[P0c]}]

        General::spell1 :
         Possible spelling error: new symbol name "mStars" is similar to existing symbol "rStars". More…

Out[7]= {0.459139, 1.00509, 1.09707, 0.651181, 0.711202}
```

It's a little suspicious that we didn't exactly reproduce the results from the last problem, but we'll soldier on anyway.

Plotting mStar versus Log[ρ[P0c]] = Log[ρc], we see evidence of a turnover in derivative around ρc = 10^-2. (Note that, since ρc > 0, dmStar/dLog[ρc] has the same sign as dmStar/dρc.)

```
In[8]:= ListPlot[Table[{ Log[ρ[P0c[[i]]]], mStars[[i]]}, {i, 1, Length[P0c]}],
         PlotRange → All, PlotJoined → True]
```




```
Out[8]= - Graphics -
```

▪ b.

Note: once again, we reset the Kernel so that our results from previous computations do not contaminate the following.

Below we generate 100 initial conditions, logarithmically spaced between the two points at which we saw the turnaround in the last problem:

```
In[1]:= Γ = 5 / 3; K = 7.32; ρ0c = 7.42 Table[10 ^ (-4 + i / 50), {i, 0, 100}]; P0c = K * ρ0c ^ Γ;

In[2]:= ρ0 = Function[P, Evaluate[(P / K) ^ (1 / Γ)]]

Out[2]= Function[P, 0.302896 P^(3/5)]

In[3]:= ρ = Function[P, Evaluate[ρ0[P] + P / (Γ - 1)]]

Out[3]= Function[P, 0.302896 P^(3/5) + 3 P/2]
```

Now we solve the TOV equations for the models (we have chosen r0 small, and r1 = 8 so we definitely calculate through P = 0 for each model in the class, based on our rStar results from part a).

```
In[4]:= r0 = 1*^-8; r1 = 13;

In[5]:= sols = Table[NDSolve[{m '[r] == 4 Pi ρ[P[r]] r^2,
              P '[r] == - (ρ[P[r]] + P[r]) (m[r] + 4 Pi r^3 P[r]) / (r (r - 2 m[r])),
              m[r0] == 0, P[r0] == P0c[[i]]}, {P, m}, {r, r0, r1}], {i, 1, 101}];

        NDSolve::mxst : Maximum number of 10000 steps reached at the point r == 12.960465220791548` . More…

        NDSolve::mxst : Maximum number of 10000 steps reached at the point r == 12.881716567595713` . More…

        NDSolve::mxst : Maximum number of 10000 steps reached at the point r == 12.711732107456085` . More…

        General::stop : Further output of NDSolve::mxst will be suppressed during this calculation. More…
```

Below we calculate rStar and mStar for each central pressure. (Note that we use Re[...] to eliminate the spurious imaginary parts we have found in previous solutions.)

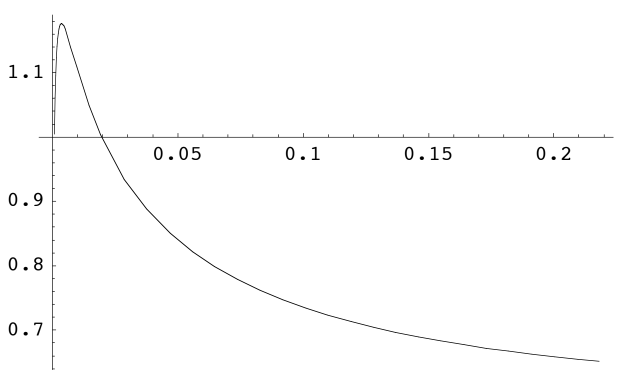
```
In[6]:= rStars = Table[Re[r /. FindRoot[P[r] /. sols[[i]][[1]], {r, 3, 4}]], {i, 1, 101}];

FindRoot::nlnum : The function value
   {InterpolatingFunction[{{1. × 10^-8, 13.}}, ≪1≫, {≪1≫}, {{≪1≫}, {≪1≫}}, {Automatic}][≪1≫]}
   is not a list of numbers with dimensions {1} at {r} = {12.1118 - 1.27456 × 10^-12 i}. More…

FindRoot::nlnum : The function value
   {InterpolatingFunction[{{1. × 10^-8, 13.}}, ≪1≫, {≪1≫}, {{≪1≫}, {≪1≫}}, {Automatic}][≪1≫]}
   is not a list of numbers with dimensions {1} at {r} = {11.9787 - 1.88019 × 10^-10 i}. More…

FindRoot::nlnum : The function value
   {InterpolatingFunction[{{1. × 10^-8, 13.}}, ≪1≫, {≪1≫}, {{≪1≫}, {≪1≫}}, {Automatic}][≪1≫]}
   is not a list of numbers with dimensions {1} at {r} = {11.8383 - 6.37846 × 10^-10 i}. More…

General::stop : Further output of FindRoot::nlnum will be suppressed during this calculation. More…
```

```
In[7]:= mStars = Table[Re[m[rStars[[i]]] /. sols[[i]][[1]]], {i, 1 , 101}];

General::spell1 :
 Possible spelling error: new symbol name "mStars" is similar to existing symbol "rStars". More…
```

```
In[8]:= mStar = Interpolation[Table[{ρ[P0c[[i]]], mStars[[i]]}, {i, 1, 101}]]

General::spell1 :
 Possible spelling error: new symbol name "mStar" is similar to existing symbol "mStars". More…

Out[8]= InterpolatingFunction[{{0.000808774, 0.218061}}, <>]
```

A plot of mStar vs ρc:

```
In[9]:= Plot[mStar[ρc], {ρc, 0.000808775, 0.218061}]

General::spell : Possible spelling error: new symbol name "ρc" is similar to existing symbols {ρ, ρ0c}. More…
```




```
Out[9]= - Graphics -
```

And here is the central density at which the turnover occurrs:

```
In[10]:= FindRoot[mStar'[ρc] == 0, {ρc, 0.000808775, 0.0009}]

Out[10]= {ρc → 0.00361846}
```

This corresponds to a central pressure of

```
In[11]:= Solve[ρ[P] == ρc /. %, P]

Out[11]= {{P → 0.000444571}}
```

And here is the solution for the max physical central pressure:

```
In[12]:= maxSol = NDSolve[{m'[r] == 4 Pi ρ[P[r]] r^2,
            P'[r] == -(ρ[P[r]] + P[r]) (m[r] + 4 Pi r^3 P[r]) / (r (r - 2 m[r])),
            m[r0] == 0, P[r0] == (P /. %[[1]])}, {P, m}, {r, r0, r1}]

Out[12]= {{P → InterpolatingFunction[{{1. × 10^-8, 13.}}, <>],
           m → InterpolatingFunction[{{1. × 10^-8, 13.}}, <>]}}
```

And R_* and M_* for this solution are

```
In[13]:= rStarMax = r /. FindRoot[Re[P[r]] /. maxSol[[1]], {r, 3, 4}]

Out[13]= 8.74332

In[14]:= mStarMax = Re[m[rStarMax] /. maxSol[[1]]]

         General::spell1 :
          Possible spelling error: new symbol name "mStarMax" is similar to existing symbol "rStarMax". More…

Out[14]= 1.17656
```

We see that $M_* = 1.17656$ km. This is converted into solar masses in the associated PDF solutions.