```
Clear["Global`*"]
Needs["DifferentialEquations`InterpolatingFunctionAnatomy`"]
```

(*NDSolve$\left[\left\{D\left[0.5\ v[z,t]^2+\frac{P[z,\ t]}{\Gamma-1},\ t\right]+D\left[\left(0.5\ \rho\ [z,t]v[z,\ t]^2+\frac{P[z,t]}{\Gamma-1}+P[z,\ t]\right)v[z,\ t],\ z\right]=\right.\right.$
$-G\ \frac{z}{r^3}-D[F[z,\ t],\ z],\ F[z,\ t]==\frac{T[z,t]^3}{\rho[z,\ t]}D[T[z,\ t],z],$
$D[\rho[z,t],\ t]==-D[\rho[z,\ t]v[z,\ t],\ z],\ D[v[z,t],\ t]z+v[z,t]\ D[v[z,t],\ z]==\frac{-1}{\rho\ [z,t]},$
$\left.\left.\frac{1}{2}Q\ \mathrm{DiracDelta}[t]+F[H,\ t]-F[0,\ t]-T[H,t]^4,\ \rho[z,0]=\mathrm{Exp}\left[\frac{-z^2}{H^2}\right]\blacksquare\right\}\right]$*)

G = 6.67×10^{-8}; (*Newton's constant in cgs*)
c = 3×10^{10} ; (*Speed of light in cgs*)
α = 0.1; (*Shakura and Sunyaev alpha*)
(*Γ=5/3; (*Adiabatic index*)*)

Msun = 2×10^{33}; (*Mass of the sun in grams*)
M = 10^6 Msun; (*Mass of the black hole*)
Sep = $126\left(G\,\frac{M}{c^2}\right)$; (*Binary separation in cm*)
(*Restrict ourselves to the inner edge of the disk*)
R = 2 Sep;
(*Central temperature and density in cgs*)
Tc0 = 1.3×10^6;
Σ0 = 6.2×10^5;
H = 0.17 R;
(*H=7 10^{11};*)
ρc0 = $\frac{\Sigma 0}{H}$;
(*ρc=1.7 10^{-7};*)

(*Kramer's Opacity assuming free-free absorption*)
(*κ=5 10^{24} ρc $Tc^{-7/2}$;*)
(*Thomson Opacity assuming that the H mass fraction is 1,
with a porosity factor of 0.2 thrown in*)
θ = 0.2;

κ = θ 0.4;

(*Proton mass in grams. Along with mu factor that that represents fully ionized
cosmic gas from King. Also consistent with the μ from Tanaka & Menou 2010*)
μ = 615×10^{-3};
mp = 10^{-24};
(*Boltzmann constant*)
k = 138×10^{-18};

```
(*Stefan-Boltzmann constant in cgs*)
σ = 567 × 10^-7;
a = 4 σ/c;
(*Kinematic viscosity*)
ν = α 2/3 (k T[z])/(μ mp Sqrt[G M/R^3]);
ν0 = ν /. T[z] → Tc0;
(*Pressures*)
pgas = (ρ[z] k T[z])/(μ mp);
prad = (4 σ)/(3 c) (T[z])^4;
pres = pgas + prad;
(*Helper functions for instability criterion*)
b = (3 ρ[z] (k / (μ mp)))/(a (T[z])^3 + 3 ρ[z] (k / (μ mp)));
α1 = (1/T[z] + 4 prad/pgas 1/T[z]);
β = 1/pgas;
β1 = prad/pgas;
γ = 1 + (α1^2 T[z])/(ρ[z] β (3 / 2) (k / (μ mp)));
Γ = b + ((γ - 1) (4 - 3 b )^2)/(b + 12 (γ - 1) (1 - b));


m0 = 10^3;
Ξ = 1;
size = Medium;


(*m0=10^3;
SteadyStateEqns0=
 NDSolve`ProcessEquations[{D[(4σ)/(3 c) (T[z])^4 + (ρ[z] k T[z])/(μ mp), z] == -(G M ρ [z]z)/(R^2+z^2)^(3/2),
    D[(-16 σ T[z]^3)/(3 κ ρ [z]) D[T[z], z], z] == 9/4 ρ[z] ν G M/R^3,  T[0] == Tc0, ρ[0] == ρc0, T'[0] == 0} ,
   { ρ, T, T'} , z, Method→{"EventLocator", "Event"→(T'[z])}, PrecisionGoal→13] //First;
NDSolve`Iterate[SteadyStateEqns0,m0 H ];
NDSolve`ProcessSolutions[SteadyStateEqns0];*)
```

$$\text{Module}\Big[\{\text{pgas, prad, pres, b}, \beta, \alpha 1, \gamma, \Gamma, \beta 1\},$$

$$\text{pgas} = \frac{\rho\, \text{k}\, \text{T}}{\mu\, \text{mp}};$$

$$\text{prad} = \frac{4\, \sigma}{3\, \text{c}} (\text{T})^4;$$

$$\text{pres} = \text{pgas} + \text{prad};$$

(*Helper functions for instability criterion*)

$$\text{b} = \frac{3\, \rho\, (\text{k} / (\mu\, \text{mp}))}{\text{a}\, (\text{T})^3 + 3\, \rho\, (\text{k} / (\mu\, \text{mp}))};$$

$$\alpha 1 = \left(\frac{1}{\text{T}} + 4\, \frac{\text{prad}}{\text{pgas}}\, \frac{1}{\text{T}}\right);$$

$$\beta = \frac{1}{\text{pgas}};$$

$$\beta 1 = \frac{\text{prad}}{\text{pgas}};$$

$$\gamma = 1 + \frac{\alpha 1^2\ \text{T}}{\rho\, \beta\ (3/2)\ (\text{k} / (\mu\, \text{mp}))};$$

$$\Gamma = \text{b} + \frac{(\gamma - 1)\ (4 - 3\, \text{b})^2}{\text{b} + 12\, (\gamma - 1)\ (1 - \text{b})};$$

$$\gamma\ /.\ \{\text{T} \to 10^4,\ \rho \to 10^{-6}\}\ //\ \text{N};$$

$$(*\text{RegionPlot}\Big[\frac{\text{Abs}\Big[\Big(\frac{\text{pres}}{\rho\ \text{T}}\frac{1}{\text{k}/(\mu\ \text{mp})}\ \frac{\text{b}\ (\Gamma-\text{b})}{(4-3\text{b})\Gamma}\Big) - \Big(\Big(\frac{\text{T}}{\text{pres}}\Big(4\ \text{a}\ \frac{\text{T}^3}{3}+\rho\ \frac{\text{k}}{\mu\ \text{mp}}\Big)+\frac{\rho}{\text{pres}}\ \frac{\text{k T}}{\mu\ \text{mp}}\frac{((3/2)\ \text{pgas} + 12\ \text{prad})}{3\ \text{prad} + \text{pres}}\Big)^{-1}\Big)\Big]}{\Big(\frac{\text{pres}}{\rho\ \text{T}}\frac{1}{\text{k}}\ \frac{\text{b}\ (\Gamma-\text{b})}{(4-3\text{b})\Gamma}\Big)} < 10^{-8},$$

$$\{\rho,\ 10^{-9},\ 10^{-6}\},\ \{\text{T},\ 10^4,\ 10^9\}\Big]*)$$

$$\left(\frac{\text{pres}}{\rho\, \text{T}}\ \frac{1}{\text{k} / (\mu\, \text{mp})}\ \frac{\text{b}\ (\Gamma - \text{b})}{(4 - 3\, \text{b})\ \Gamma}\right)$$

$$\Big]$$

$$\text{Manipulate}\Big[\text{Module}\Big[\{\nu 0\text{, SteadyState, m, grid, l1, l2, l3, zmax, }\tau\text{tot, dflux,}$$
$$\text{FluxDecrease, FluxDecreasePos, u0, ZValues, uValues, }\tau\text{Values, Tph, TValues,}$$
$$\text{F, plot1, plot2, delad, delonic, delmolli, logp, logT, pres, prad, pgas}\},$$

```
(*SteadyStateEqns=
    NDSolve`Reinitialize[SteadyStateEqns0,{  T[0]==Tc, ρ[0]==ρc, T'[0]==0} ]//First;
   NDSolve`Iterate[SteadyStateEqns,m0 H ];|
   SteadyState=NDSolve`ProcessSolutiZons[SteadyStateEqns];*)

ν0 = ν /. T[z] → Tc;

pgas = (ρ[z] k T[z])/(μ mp);

prad = (4 σ)/(3 c) (T[z])^4;
pres = pgas + prad;
(*SteadyStateEqns=
 NDSolve`Reinitialize[SteadyStateEqns0,{  T[0]==Tc, ρ[0]==ρc, T'[0]==0} ]//First;
NDSolve`Iterate[SteadyStateEqns,m0 H ];
SteadyState=NDSolve`ProcessSolutions[SteadyStateEqns];*)
SteadyState =
 NDSolve[{D[pres, z] == -(G M ρ [z] z)/(R^2 + z^2)^(3/2), D[(-16 σ T[z]^3)/(3 κ ρ [z]) D[T[z], z], z] == 9/4 ρ[z] ν0 G M/R^3,
    T[0] == Tc, ρ[0] == ρc, T'[0] == 0}, {ρ, T, T'}, {z, 0, m0 H},
   Method → {"EventLocator", "Event" → {ρ[z] - 1.05 ρc }, MaxIterations → 1000,
     "EventLocationMethod" → "StepBegin"}, PrecisionGoal → 13] // First;


m = 1/H (InterpolatingFunctionDomain[ρ /. SteadyState])[[1, -1]];
grid = T /. SteadyState // InterpolatingFunctionGrid // Flatten // #[[2 ;;]] &;
l1 = T /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l2 = ρ /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l3 = T' /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
dflux = (-16 σ ( l1)^3)/(3 κ l2) l3 // Differences;
FluxDecrease = If[(dflux // Min) ≥ 0, dflux[[-1]], Select[dflux, # < 0 &] // First];
FluxDecreasePos = Position[dflux, FluxDecrease][[1, 1]];

zmax = grid[[FluxDecreasePos]];
(*zmax=(InterpolatingFunctionDomain[ρ/.SteadyState])[[1,-1]];*)

(*q=0.7;*)

τtot = κ NIntegrate[(ρ[z] /. SteadyState ), {z, 0, zmax}];
u0 = 2 NIntegrate[(ρ[z] /. SteadyState ), {z, 0, zmax}];
```

```
F = (-16 σ (T[z])^3)/(3 κ ρ[z]) T'[z] /. SteadyState /. z → zmax;
(*Ξ=4/5 0.11;*)

ZValues = Range[0, zmax, zmax/100];
(*ρValues=ρ[#]/.SteadyState&/@ZValues;
pValues=(4σ)/(3 c) (T[#])^4+(ρ[#] k T[#])/(μ mp)/.SteadyState&/@ZValues;
logρ=Log[ρ[#]]/.SteadyState&/@ZValues;
logp=Log[(4σ)/(3 c) (T[#])^4+(ρ[#] k T[#])/(μ mp)]/.SteadyState&/@ZValues;*)
delonic = (((pres/(ρ[z] T[z]) 1/(k/(μ mp)) (b (Γ - b))/((4 - 3 b) Γ)) /. SteadyState) /. z → # &) /@ ZValues;

delad = ((T[z]/pres (4 a T[z]^3/3 + ρ[z] k/(μ mp)) + ρ[z]/pres (k T[z])/(μ mp) (1.5 pgas + 12 prad)/(3 prad + pres))^-1 /. SteadyState) /.
    z → # & /@ ZValues;
delmolli = ((1 + β1) (4 β1 + 1))/((12 β1 + 1/(γ-1)) + (4 β1 + 1)^2) /. SteadyState /. z → # & /@ ZValues;

uValues = NIntegrate[ ρ[z] /. SteadyState, {z, 0, #}] & /@ ZValues // Flatten;
τValues = τtot - NIntegrate[κ ρ[z] /. SteadyState, {z, 0, #}] & /@ ZValues // Flatten;

logT = Log[(T[#] /. SteadyState)] & /@ ZValues // Flatten;
logp = Log[(pres /. SteadyState)] /. z → # & /@ ZValues // Flatten;
plot1 = Riffle[uValues, TValues/Tc - 1] // Partition[#, 2] & //
   ListLinePlot[#, PlotStyle → Directive[Blue], PlotRange → All] &;

Tph = (F/(σ Ξ))^0.25;

plot2 = Plot[(1 - 3 (Tph^4 κ u0)/(4 Tc^4) (u/u0)^2)^0.25 - 1, {u, 0, u0/2},
   PlotStyle → Directive[Red], AxesLabel -> {"u [g cm^-2]", "T/T_c-1"}];
(delad[[2 ;;]] - (logT // Differences)/(logp // Differences)) // Min;
```

```
GraphicsColumn[{Show[{ListPlot[{ delonic, delmolli}], ListPlot[(logT // Differences)/(logp // Differences)]}],
  GraphicsRow[{(*Row[{"τtot=",τtot[[1]],"Σ=",2 τtot/κ[[1]],"Tph (From Flux)=", Tph,
     "Tph=", (4/3)^0.25 Tc/τtot^0.25[[1]],   9/8 ν Σ G M/R^3/.{T[z]:>Tc, Σ:>2 τtot/κ}, F}, "   "],*)
    Plot[(-16 σ T[z]^3)/(3 κ ρ [z]) T'[z] /. SteadyState, {z, 0, zmax}, PlotRange → All, ImageSize → size,
     AxesLabel → {"z [cm]", "Flux [ergs s^-1 cm^-2]"}], Plot[Log[(ρ[z])/ρc] /. SteadyState,
     {z, 0, zmax}, ImageSize → size, AxesLabel → {"z [cm]", "ρ/ρc"}]}],
  GraphicsRow[{Plot[T[z]/Tc /. SteadyState, {z, 0, zmax}, ImageSize → size,
     AxesLabel → {"z [cm]", "T/Tc"}], Show[{plot2, plot1}, PlotRange → All]}]}]],
{ρc, 5. × 10^-8, 5. × 10^-7, Appearance → "Open"}, {{Tc, 1.3 × 10^6},
 10^5,
 10^7, Appearance → "Open"},
AppearanceElements → None,
ControlPlacement → Right,
ContinuousAction → False,
TrackedSymbols :→ {Tc, ρc}]
```

```
Tc = 1.3 × 10^6
ρc = 5. × 10^-8
SteadyState =
  NDSolve[{D[(4 σ)/(3 c) (T[z])^4 + (ρ[z] k T[z])/(μ mp), z] == -(G M ρ [z] z)/((R^2 + z^2)^(3/2)), D[(-16 σ T[z]^3)/(3 κ ρ [z]) D[T[z], z], z] ==
      9/4 ρ[z] ν0 G M/R^3, T[0] == Tc, ρ[0] == ρc, T'[0] == 0}, {ρ, T, T'}, {z, 0, m0 H},
    Method → {"EventLocator", "Event" → {ρ[z] - 1.05 ρc }, MaxIterations → 1000,
      "EventLocationMethod" → "StepBegin"}, PrecisionGoal → 13] // First;


m = 1/H (InterpolatingFunctionDomain[ρ /. SteadyState]) [[1, -1]];
grid = T /. SteadyState // InterpolatingFunctionGrid // Flatten // #[[2 ;;]] &;
l1 = T /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l2 = ρ /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l3 = T ' /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
dflux = (-16 σ ( l1)^3)/(3 κ l2) l3 // Differences;
FluxDecrease = If[(dflux // Min) ≥ 0, dflux[[-1]], Select[dflux, # < 0 &] // First];
FluxDecreasePos = Position[dflux, FluxDecrease][[1, 1]];
zmax = grid[[FluxDecreasePos]];
temps = T /. SteadyState // InterpolatingFunctionValuesOnGrid;
zs = T /. SteadyState // InterpolatingFunctionGrid;
J[x_] := - 1/(4 π κ) 9/4 ν0 G M/R^3 + σ/π x^4
js = J /@ temps;
ListLinePlot[js]

Plot[ (-16 σ ( T[z])^3)/(3 κ ρ[z]) T '[z] /. SteadyState, {z, 0, zmax}, PlotStyle → Directive[Red]]

a = ListLinePlot[{Transpose[{zs[[2 ;;]] // Flatten, (js // Differences)/(zs // Differences) // Flatten}]}];

b = Plot[3/(4 π) κ (16 σ ( T[z])^3)/(3 κ) T '[z] /. SteadyState, {z, 0, zmax}, PlotStyle → Directive[Red]];
Show[a, b]


DensityRatio0 = Table[
    Tc = 10^i;
```

```
ρc = 10^j;
ν0 = ν /. T[z] → Tc;
m0 = 10^3;

(*SteadyStateEqns=
 NDSolve`Reinitialize[SteadyStateEqns0,{  T[0]==Tc, ρ[0]==ρc, T'[0]==0} ]//First;
NDSolve`Iterate[SteadyStateEqns,m0 H ];
SteadyState=NDSolve`ProcessSolutions[SteadyStateEqns];*)
```

$$\text{SteadyState} = \text{NDSolve}\left[\left\{D\left[\frac{4\,\sigma}{3\,c}\,(T[z])^4 + \frac{\rho[z]\,k\,T[z]}{\mu\,mp},\ z\right] == -\frac{G\,M\,\rho\,[z]\,z}{\left(R^2+z^2\right)^{3/2}},\right.\right.$$

$$\left.D\left[\frac{-16\,\sigma\,T[z]^3}{3\,\kappa\,\rho\,[z]}\,D[T[z],\ z],\ z\right] == \frac{9}{4}\,\rho[z]\,\nu0\,G\,\frac{M}{R^3},\ T[0] == Tc,\ \rho[0] == \rho c,\ T'[0] == 0\right\},$$

$$\{\rho,\ T,\ T'\},\ \{z,\ 0,\ m0\,H\},\ \text{Method} \to \{\text{"EventLocator"},\ \text{"Event"} \to \{\rho[z] - 1.01\,\rho c\},$$

$$\left.\text{"EventLocationMethod"} \to \text{"StepBegin"}\},\ \text{PrecisionGoal} \to 13\right] \ //\ \text{First};$$

```
grid = T /. SteadyState // InterpolatingFunctionGrid // Flatten // #[[2 ;;]] &;
l1 = T /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l2 = ρ /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l3 = T' /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
```

$$\text{dflux} = \frac{-16\,\sigma\,(\text{l1})^3}{3\,\kappa\,\text{l2}}\,\text{l3} \ //\ \text{Differences};$$

```
FluxDecrease = If[(dflux // Min) ≥ 0, dflux[[-1]], Select[dflux, # < 0 &] // First];
FluxDecreasePos = Position[dflux, FluxDecrease][[1, 1]];
zmax = grid[[FluxDecreasePos]];
```

$$\text{test} = \frac{(\rho[\text{zmax}]\ /.\ \text{SteadyState})}{\rho c};$$

```
(*test=(l2//Differences//Max)+1;*)
u0 = 2 NIntegrate[(ρ[z] /. SteadyState ), {z, 0, zmax}];
```

$$F = \frac{-16\,\sigma\,(T[z])^3}{3\,\kappa\,\rho\,[z]}\,T'[z]\ /.\ \text{SteadyState}\ /.\ z \to \text{zmax};$$

```
Ξ = 1;
```

$$\text{ZValues} = \text{Range}\left[0,\ \text{zmax},\ \frac{\text{zmax}}{100}\right];$$

```
uValues = NIntegrate[ ρ[z] /. SteadyState, {z, 0, #}] & /@ ZValues // Flatten;
u0 = 2 NIntegrate[(ρ[z] /. SteadyState ), {z, 0, zmax}];

TValues = (T[#] /. SteadyState) & /@ ZValues // Flatten;
```

$$\text{TNumerical} = \frac{\text{TValues}}{\text{Tc}} - 1;$$

$$\text{Tph} = \left(\frac{F}{\sigma\,\Xi}\right)^{0.25};$$

```
If[test < 1,
 TAnalytic = ((1 - 3 (Tph^4 κ u0)/(4 (Tc)^4) (#/u0)^2)^0.25 - 1) & /@ uValues, TAnalytic = 1 & /@ uValues];

plot1 = ListLinePlot[Transpose[{uValues, TNumerical}],
  PlotStyle → Directive[Blue], PlotRange → All];
plot2 = ListLinePlot[Transpose[{uValues, TAnalytic}],
  PlotStyle → Directive[Red], AxesLabel -> {"u [g cm^-2]", "T/T_c-1"}];
size = Medium;
logT = Log[(T[#] /. SteadyState)] & /@ ZValues // Flatten;
logp = Log[(pres /. SteadyState)] /. z → # & /@ ZValues // Flatten;

delonic = (((pres/(ρ[z] T[z]) 1/(k / (μ mp)) (b (Γ - b))/((4 - 3 b) Γ)) /. SteadyState) /. z → # &) /@ ZValues;
delmolli = ((1 + β1) (4 β1 + 1))/((12 β1 + 1/(γ-1)) + (4 β1 + 1)^2) /. SteadyState /. z → # & /@ ZValues;

delad =
 (((T[z]/pres (4 a T[z]^3/3 + ρ[z] k/(μ mp)) + ρ[z]/pres (k T[z])/(μ mp) ((3 / 2) pgas + 12 prad)/(3 prad + pres))^-1 /. SteadyState) /.
    z → # & /@ ZValues;
(*profiles=GraphicsColumn[{GraphicsRow[{Plot[(-16 σ T[z]^3)/(3 κ ρ [z]) T'[z]/.SteadyState,
      {z, 0,zmax}, PlotRange→All, ImageSize→size,
      AxesLabel→{ "z [cm]", "Flux [ergs SuperscriptBox[s,-1\]cm^-2]"}], Plot[
      ρ[z]/.SteadyState, {z,0, zmax}, ImageSize→size, AxesLabel→{ "z [cm]", "ρ"}]}],
   GraphicsRow[{Plot[T[z]/.SteadyState, {z, 0, zmax}, ImageSize→size,
      AxesLabel→{ "z [cm]", "T"}], Show[{plot2, plot1}, PlotRange→All]}]}];
If[test<1,Export[NotebookDirectory[]<>"pc"<>ToString[j]<>"Tc"<>
   ToString[i]<>".pdf",profiles]]*)

(*Export[
  NotebookDirectory[]<>"pc"<>ToString[j]<>"Tc"<>ToString[i]<>".pdf",GraphicsColumn[
   {GraphicsRow[{(*Row[{"τ_tot=",τtot[[1]],"Σ=",2 τtot/κ[[1]],"T_ph (From Flux)=", Tph,
        "T_ph=", (4/3)^0.25 Tc/τtot^0.25[[1]],  9/8 ν Σ G M/R^3/.{T[z]:>Tc, Σ:>2 τtot/κ}, F}, "    "],*)
      Plot[(-16 σ T[z]^3)/(3 κ ρ [z]) T'[z]/.SteadyState, {z, 0,zmax}, PlotRange→All, ImageSize→size,
```

```
        AxesLabel→{ "z [cm]", "Flux [ergs SuperscriptBox[s,-1\]cm^-2]"}], Plot[
        ρ[z]/.SteadyState, {z,0, zmax}, ImageSize→size, AxesLabel→{ "z [cm]", "ρ"}]}],
      GraphicsRow[{Plot[T[z]/.SteadyState, {z, 0, zmax}, ImageSize→size,
        AxesLabel→{ "z [cm]", "T"}], Show[{plot2, plot1}, PlotRange→All]}]}]];*)
  {Tc, ρc, If[test < 1, Red, Blue], Max[(TNumerical[[2 ;;]] - TAnalytic[[2 ;;]]) // Abs / TAnalytic[[2 ;;]]],
   (delmolli[[2 ;;]] - (logT // Differences)/(logp // Differences)) // Cases[#, _Real] & // Min,
   (delonic[[2 ;;]] - (logT // Differences)/(logp // Differences)) // Cases[#, _Real] & // Min,
   Max[Abs[(delonic - delmolli)/delonic]]}
  , {i, 4, 9, 0.2}, {j, -9, -6, 0.2} ] // N;
DensityRatio = DensityRatio0 // Flatten[#, 1] &;


WeirdProfiles = Select[DensityRatio, #[[3]] == RGBColor[0., 0., 1.] &];
NormalProfiles = Select[DensityRatio, #[[3]] == RGBColor[1., 0., 0.] &];
TProfileDeviation = Select[DensityRatio, ((#[[4]] // Abs) > 0.01) &];
TProfileDeviation = Complement[TProfileDeviation, WeirdProfiles];
weird = ListLogLogPlot[WeirdProfiles[[All, {1, 2}]], PlotRange → All,
   AxesLabel → {"Tc [K]", "ρc [g cm^-3]" }, PlotStyle → Directive[Blue]];
normal = ListLogLogPlot[NormalProfiles[[All, {1, 2}]], PlotRange → All,
   AxesLabel → {"Tc [K]", "ρc [g cm^-3]" }, PlotStyle → Directive[Red]];
tdeviation := ListLogLogPlot[TProfileDeviation[[All, {1, 2}]], PlotRange → All,
   AxesLabel → {"Tc [K]", "ρc [g cm^-3]" }, PlotStyle → Directive[Green]];
If[(TProfileDeviation // Length) == 0, Show[weird, normal, PlotRange → All],
 Show[weird, normal, tdeviation, PlotRange → All]]




Convective =
  Select[Cases[WeirdProfiles[[All, {1, 2, -2}]], {_Real , _Real , _Real }], #[[3]] < 0 &] ⋃
   Select[Cases[NormalProfiles[[All, {1, 2, -2}]], {_Real , _Real , _Real }], #[[3]] < 0 &];


Show[ normal, weird, ListLogLogPlot[Convective[[All, 1 ;; 2]], PlotStyle → Directive[Purple],
  PlotRange → All, AxesLabel → {"Tc [K]", "ρc [g cm^-3]" }], tdeviation, PlotRange → All]


(*Manipulate[
 pt={{DensityRatio0[[i,j,1]], DensityRatio0[[i, j,2]]}}//
   ListLogLogPlot[#, PlotStyle->Directive[Purple]]&;
 GraphicsColumn[{
   If[(TProfileDeviation//Length)==0, Show[weird, normal,pt, PlotRange→All],
    Show[weird, normal, tdeviation,pt, PlotRange→All]],
   DensityRatio0[[i, j, 5]] }
 ],
 {i,1, DensityRatio0//Dimensions//First}, {j,1, DensityRatio0//Dimensions//#[[2]]&}]*)
```

```
SteadyState =
  NDSolve[{D[(4 σ)/(3 c) (T[z])^4 + (ρ[z] k T[z])/(μ mp), z] == -(G M ρ [z] z)/(R^2 + z^2)^(3/2), D[(-16 σ T[z]^3)/(3 κ ρ [z]) D[T[z], z], z] ==
      9/4 ρ[z] ν0 G M/R^3, T[0] == 3.981071705534969`*^7, ρ[0] == 6.309573444801943`*^-9,
     T'[0] == 0} , {ρ, T, T'} , {z, 0, m0 H}, Method → {"EventLocator", "Event" → (T'[z])},
    PrecisionGoal → 14, AccuracyGoal → 14, WorkingPrecision → 50] // First;
grid = T /. SteadyState // InterpolatingFunctionGrid // Flatten // #[[2 ;;]] &;
l1 = T /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l2 = ρ /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l3 = T' /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
dflux = (-16 σ ( l1)^3)/(3 κ l2) l3 // Differences;
FluxDecrease = If[(dflux // Min) > 0, dflux[[-1]], Select[dflux, # < 0 &] // First];
FluxDecreasePos = Position[dflux, FluxDecrease][[1, 1]];

zmax = grid[[FluxDecreasePos]];

Plot[(-16 σ T[z]^3)/(3 κ ρ [z]) T'[z] /. SteadyState, {z, 0, zmax}]

SteadyState =
  NDSolve[{D[(4 σ)/(3 c) (T[z])^4 + (ρ[z] k T[z])/(μ mp), z] == -(G M ρ [z] z)/(R^2 + z^2)^(3/2), D[(-16 σ T[z]^3)/(3 κ ρ [z]) D[T[z], z], z] ==
      9/4 ρ[z] ν0 G M/R^3, T[0] == 3.981071705534969`*^7, ρ[0] == 6.309573444801943`*^-9,
     T'[0] == 0} , {ρ, T, T'} , {z, 0, m0 H}, Method → {"EventLocator", "Event" → (T'[z])},
    WorkingPrecision → 50, PrecisionGoal → 20, AccuracyGoal → 13] // First;
grid = T /. SteadyState // InterpolatingFunctionGrid // Flatten // #[[2 ;;]] &;
l1 = T /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l2 = ρ /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
l3 = T' /. SteadyState // InterpolatingFunctionValuesOnGrid // Flatten // #[[2 ;;]] &;
dflux = (-16 σ ( l1)^3)/(3 κ l2) l3 // Differences;
FluxDecrease = If[(dflux // Min) > 0, dflux[[-1]], Select[dflux, # < 0 &] // First];
FluxDecreasePos = Position[dflux, FluxDecrease][[1, 1]];

zmax = grid[[FluxDecreasePos]];

Plot[(-16 σ T[z]^3)/(3 κ ρ [z]) T'[z] /. SteadyState, {z, 0, zmax}]
```