Introdução à Cosmologia Avançada

Luciano Barosi
Colaboração BINGO
UAF/CCT/UFCG

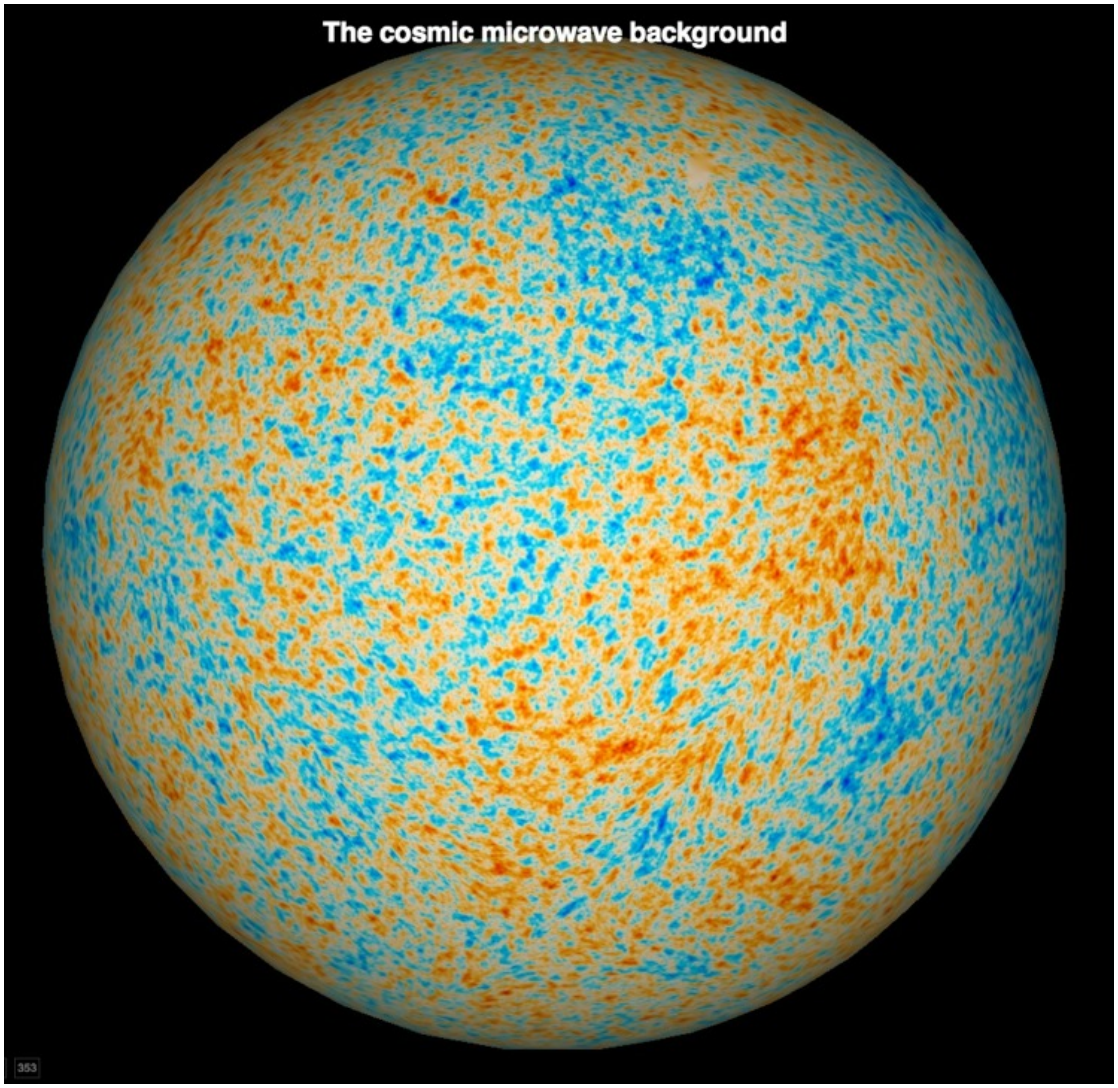
Motivação: Som Cósmico

In[ ]:=

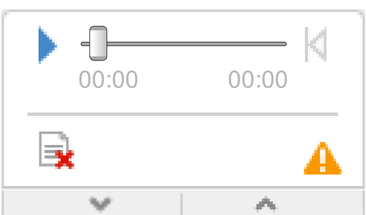
```
SetDirectory[NotebookDirectory[]]
Import["CMB_sphere.png"]
Import["BBSnd30.wav"](*Som pré-gravado*)
```

Out[ ]= /home/lbarosi/Dropbox/ENSINO/2019-1/Cosmologia

Out[•]=



Out[•]=



Créditos John G. Cramer

A radiação cósmica de fundo tem perturbações relacionadas as flutuações do fluido foton-barion na época do desacoplamento. Perturbações em um fluido são usualmente chamadas de perturbações acústicas. Podemos analisar as perturbações no espaço de Fourier, separando em modos. É a amplitude de cada modo que captura informações importantes sobre a física dessas oscilações. A título de ilustração, podemos considerar qual o timbre do som que seria gerado por esse tipo de oscilações. Isso é o que você esta ouvindo.

In[•]:=
```
SetDirectory[NotebookDirectory[]];
fPk=5.0;(*Frequency scale Factor - Change it if you like*)
dur=30;(*Duração do som*)
d2=dur/2;
duSt=ToString[dur];
ffac=fPk*d2^(2/3);
cbrPk=379;(*peak time of radiation at CMB*)
cbrSg=118;(*sigma*)
Temp=ReadList["PlanckData.txt",Number,RecordLists→True];
lenT=Length[Temp];
ListPlot[Table[{Temp[[j,1]],Temp[[j,2]]},{j,lenT}],Frame→True,FrameLabel→{"L","Amplitude",
BBSnd[t_]:=Exp[-((t-d2)/(cbrSg*d2/cbrPk))^2 /2]*Sum[(((Temp[[j,1]]-Temp[[j-1,1]])/Temp[[j,1
Plot[BBSnd[t],{t,0,dur},Frame→True,FrameLabel→{"Time","Amplitude","Sound Function",""},Plo

(*Demora para rodar: Não faça isso em Público!*)
(*BBs=Play[BBSnd[t],{t,0,dur},PlayRange→All];
Export["BBSnd"<>duSt<>".wav",BBs,"WAV"]*)
```

Introdução

Eletromagnetismo Covariante

Apenas um warm - up no uso do Xact, considerando uma variedade 4 - dimensional e as equações covariantes do eletrmagnetismo. Usando um projetor fazemos a decomposição 3+1 e analisamos as equações de Maxwell.

Definições Iniciais

In[1]:=
```
(*Carregando Pacotes*)
Quiet@Block[{Print},
  << xAct`xTensor`;
  << xAct`xCoba`;
 ]

(*Definindo opções úteis*)
$Pre = ScreenDollarIndices;
SetOptions[ContractMetric, AllowUpperDerivatives → True];
$DefInfoQ = False;
```

In[•]:=
```
(*Definindo Variedade*)
DefManifold[M, 4, Complement[IndexRange[a, z], {g, h, n, x, y, z, t}]]
(*Definindo Métrica e derivada covariante CD*)
DefMetric[-1, g[-a, -b], CD, {";", "∇"}, PrintAs → "g"]
```

In[∘]:=

```
(*Função pega uma identidade e transforma numa regra*)
ApplyRuleN[expr_] := MakeRule[Evaluate[List @@ expr], MetricOn → All];
```

Eletromagnetismo

In[∘]:=

```
(*Define tensor timelike como projetor*)
DefTensor[n[a], M]
AutomaticRules[n, MakeRule[{n[a] n[-a], -1}]]
AutomaticRules[n, MakeRule[{g[-a, -b] n[a] n[b], -1}]]
(*Métrica induzida no espaço ortogonal*)
DefMetric[1, h[-a, -b], cd, SymbolOfCovD -> {"|", "D"}, InducedFrom → {g, n}]
PrintAs[epsilonh] ^= "ϵ";(*epsilon induzido*)
(*Vale para Minkowsky, assin não precisa trabalhar com xcoba em componentes*)
IndexSet[CD[a_][n[b_]], 0](*Projetor Constante*);
IndexSet[CD[a_][epsilonh[b_, c_, d_]], 0](*Epsilon COnstante*);
```

Rules {1} have been declared as UpValues for n.

Rules {1, 2} have been declared as UpValues for n.

DefMetric: There are already metrics {g} in vbundle TM.

In[∘]:=

```
(*Relação entre o epsilon em três e quatro dimensões*)
ThreeEpsToFourEpsRule =
  {epsilonh[a_, b_, c_] :→ Module[{d}, n[d] epsilong[-d, a, b, c]]};
nDotThreeEpsRule = MakeRule[{n[d] epsilong[-d, a, b, c], epsilonh[a, b, c]},
   MetricOn → All, ContractMetrics → True];
FourEpsToThreeEpsRule = {epsilong[a_, b_, c_, d_] :→
    -4 Antisymmetrize[n[a] epsilonh[b, c, d], {a, b, c, d}]};
```

In[∘]:=

```
(*Campo Elétrico, Magnético e Potencial Vetor no espaço Ortogonal*)
DefTensor[EE[-a], M, OrthogonalTo → {n[a]},
 ProjectedWith → {h[a, -c]}, PrintAs → "E"]
DefTensor[BB[-a], M, OrthogonalTo → {n[a]},
 ProjectedWith → {h[a, -c]}, PrintAs → "B"]
DefTensor[AA[-a], M, OrthogonalTo → {n[a]},
 ProjectedWith → {h[a, -c]}, PrintAs → "Ã"]
DefTensor[A[-a], M](*Potencial 1-forma*)
DefTensor[F[-a, -b], M, Antisymmetric[{-a, -b}]](*2-forma*)
DefTensor[ϕ[], M](*Potencial Elétrico*)
```

In[•]:=
```
(*Decomposição do potencial 1-forma*)
Adecomp = (InducedDecomposition[A[-a], {h, n}] /. Projectorh → ProjectWith[h]) //
   Simplification;
(*EM termos do potencial escalar e do potencial vetor*)
defAdecomp = A[-a] == AA[b] h[-a, -b] + n[-a] ϕ[];
(*Campo elétrico*)
defEE = CD[b][ϕ[]] == -h[b, -a] EE[a];
(*Campo Magnético*)
defBB = CD[d][AA[f]] == 1/2 epsilonh[-a, d, f] BB[a];
defF = F[-a, -b] == CD[-a][A[-b]] - CD[-b][A[-a]];
```

In[•]:=
```
TIMELIKE[expr_] :=
  ((n[-IndicesOf[Free][expr][[1]]] expr) /. ApplyRuleN[defF] /. ApplyRuleN[
            defAdecomp] /. ApplyRuleN[defEE] /. ApplyRuleN[defBB] /.
         FourEpsToThreeEpsRule // ToCanonical) /. nDotThreeEpsRule //
     Simplification // ContractMetric // ToCanonical
SPACELIKE[expr_] := ((((expr // ProjectWith[h][#] &))) /. ApplyRuleN[defF] /.
            ApplyRuleN[defAdecomp] /. ApplyRuleN[defEE] /. ApplyRuleN[defBB] /.
         FourEpsToThreeEpsRule // ToCanonical) /. nDotThreeEpsRule //
     Simplification // ContractMetric // ToCanonical
```

In[•]:=
```
F[-a, -b] // TIMELIKE
F[-a, -b] // SPACELIKE
```

Out[•]= $-E_b$

Out[•]= $B^c \ \epsilon_{abc}$

In[•]:=
```
(*Equações de Maxwell Covariantes*)
eqM1 = CD[-a][F[a, b]] == 0
eqM2 = CD[-a][1/2 epsilong[a, b, c, d] F[-c, -d]] == 0
```

Out[•]= $\nabla_a F^{ab} == 0$

Out[•]= $\frac{1}{2} \epsilon g^{abcd} \ \nabla_a F_{cd} == 0$

In[ ]:=
```
eqM1 // TIMELIKE
eqM1 // SPACELIKE
eqM2 // TIMELIKE
eqM2 // SPACELIKE
```

Out[ ]= $\nabla_a E^a + n^a \; n^b \; \nabla_b E_a == 0$

Out[ ]= $\epsilon^b{}_{ac} \; \nabla^c B^a + h^b{}_c \; n^a \; \nabla^c E_a == 0$

Out[ ]= $-h_{ab} \; \nabla^b B^a == 0$

Out[ ]= $-h^b{}_c \; n^a \; \nabla_a B^c - \epsilon^b{}_{ac} \; \nabla^c E^a == 0$

Covariance Geral

(Rodar Eletromagnetismo Antes) Apenas mantendo as definições dos objetos geométricos em dia.

Derivada Covariante e Curvatura

In[ ]:=
```
UndefMetric[{g, h}]
DefTensor[X[a], M]
DefTensor[X1[a], M, PrintAs → X̃]
DefTensor[V[a], M]
DefTensor[V1[a], M]
```

```
** UndefCovD: Undefined covariant derivative CD
** UndefTensor: Undefined tetrametric Tetrag†
** UndefTensor: Undefined tetrametric Tetrag
** UndefTensor: Undefined antisymmetric tensor epsilong
** UndefTensor: Undefined symmetric metric tensor g
** UndefInertHead: Undefined projector inert-head Projectorh
** UndefTensor: Undefined acceleration vector Accelerationn
** UndefTensor: Undefined extrinsic curvature tensor ExtrinsicKh
** UndefCovD: Undefined covariant derivative cd
** UndefTensor: Undefined tetrametric Tetrah†
** UndefTensor: Undefined tetrametric Tetrah
** UndefTensor: Undefined antisymmetric tensor epsilonh
** UndefTensor: Undefined symmetric metric tensor h
```

In[•]:=

```
(*Derivadas de Vetores não são Vetores*)
V1[a] == PD[-b][X[a]] V[b]
PD[-c][PD[-b][X[a]] V[b]]
```

Out[•]= $V1^{a} == V^{b}\,\partial_{b}X^{a}$

Out[•]= $\partial_{b}X^{a}\,\partial_{c}V^{b} + V^{b}\,\partial_{c}\partial_{b}X^{a}$

In[•]:=

```
DefMetric[-1, g[-a, -b], CD, {";", "∇"}, PrintAs → "g"]
```

In[•]:=

```
(*Símbolos de Christoffel definem derivada covariante compatível com a métrica*)
Hold[CD[-a][g[-b, -c]]] // CovDToChristoffel // ReleaseHold
Hold[CD[-c][g[-a, -b]]] // CovDToChristoffel // ReleaseHold
Hold[CD[-b][g[-c, -a]]] // CovDToChristoffel // ReleaseHold
% + %% - %%% == 0 // Simplification
eqs = g[f, a] % // ToCanonical
ChristoffelG =
  Equal @@ (Solve[eqs, Union@Cases[eqs, _ChristoffelCD, Infinity]] // Flatten //
      First) // ToCanonical // Simplification
```

Out[•]= $-\Gamma[\nabla]^{d}{}_{ac}\,g_{bd} - \Gamma[\nabla]^{d}{}_{ab}\,g_{dc} + \partial_{a}g_{bc}$

Out[•]= $-\Gamma[\nabla]^{d}{}_{cb}\,g_{ad} - \Gamma[\nabla]^{d}{}_{ca}\,g_{db} + \partial_{c}g_{ab}$

Out[•]= $-\Gamma[\nabla]^{d}{}_{ba}\,g_{cd} - \Gamma[\nabla]^{d}{}_{bc}\,g_{da} + \partial_{b}g_{ca}$

Out[•]= $-2\,\Gamma[\nabla]^{d}{}_{bc}\,g_{ad} - \partial_{a}g_{bc} + \partial_{b}g_{ac} + \partial_{c}g_{ab} == 0$

Out[•]= $-2\,\Gamma[\nabla]^{f}{}_{bc} - g^{fa}\,\partial_{a}g_{bc} + g^{fa}\,\partial_{b}g_{ca} + g^{fa}\,\partial_{c}g_{ba} == 0$

Out[•]= $\Gamma[\nabla]^{f}{}_{bc} == \frac{1}{2}\,g^{fa}\,\left(-\partial_{a}g_{bc} + \partial_{b}g_{ca} + \partial_{c}g_{ba}\right)$

In[•]:=

```
(*Derivadas Covariantes não comutam*)
CD[-a][CD[-b][V[c]]] - CD[-b][CD[-a][V[c]]] // CovDToChristoffel //
   ChristoffelToMetric // Implode // Simplification
```

Out[•]=

$$\frac{1}{4}\,g^{ce}\,\Big(g^{fi}\,\big(\partial g_{adf}\,\partial g_{bei} - \partial g_{afd}\,\partial g_{bei} - \partial g_{adf}\,\partial g_{bie} + \partial g_{afd}\,\partial g_{bie} - \partial g_{bef}\,\partial g_{dia} + \partial g_{bfe}\,\partial g_{dia} + \partial g_{afe}\,\left(\partial g_{bdi} - \partial g_{bid} - \partial g_{dib}\right) + \partial g_{aef}\,\left(-\partial g_{bdi} + \partial g_{bid} + \partial g_{dib}\right) - \partial g_{dib}\,\partial g_{efa} + \partial g_{bdf}\,\partial g_{eia} - \partial g_{bfd}\,\partial g_{eia} - \partial g_{adf}\,\partial g_{eib} + \partial g_{afd}\,\partial g_{eib} + \partial g_{dfa}\,\partial g_{eib}\big) + 2\,\left(\partial\partial g_{adbe} - \partial\partial g_{aebd} - \partial\partial g_{bdae} + \partial\partial g_{bead}\right)\Big)\,V^{d}$$

In[∘]:=
```
(*Tensor de Curvatura mede a não-comutatividade das derivadas*)
CD[-a][CD[-b][V[c]]] - CD[-b][CD[-a][V[c]]] // CovDToChristoffel //
   ChristoffelToMetric // Implode // Simplification
RiemannCD[c, -d, -a, -b] V[d] // RiemannToChristoffel
% // ChristoffelToMetric
% - %%% // Implode // Simplification
```

Out[∘]=
$$\frac{1}{4} g^{ce} \left(g^{fi} \left(\partial g_{adf}\ \partial g_{bei} - \partial g_{afd}\ \partial g_{bei} - \partial g_{adf}\ \partial g_{bie} + \partial g_{afd}\ \partial g_{bie} - \partial g_{bef}\ \partial g_{dia} + \partial g_{bfe}\ \partial g_{dia} + \partial g_{afe} \left(\partial g_{bdi} - \partial g_{bid} - \partial g_{dib} \right) + \partial g_{aef} \left(- \partial g_{bdi} + \partial g_{bid} + \partial g_{dib} \right) - \partial g_{dib}\ \partial g_{efa} + \partial g_{bdf}\ \partial g_{eia} - \partial g_{bfd}\ \partial g_{eia} - \partial g_{adf}\ \partial g_{eib} + \partial g_{afd}\ \partial g_{eib} + \partial g_{dfa}\ \partial g_{eib} \right) + 2 \left(\partial\partial g_{adbe} - \partial\partial g_{aebd} - \partial\partial g_{bdae} + \partial\partial g_{bead} \right) \right) V^{d}$$

Out[∘]=
$$g_{bf}\ g^{ce}\ V^{d} \left(- \Gamma[\nabla]^{f}{}_{ei}\ \Gamma[\nabla]^{i}{}_{da} + \Gamma[\nabla]^{f}{}_{di}\ \Gamma[\nabla]^{i}{}_{ea} + \partial_{d}\Gamma[\nabla]^{f}{}_{ea} - \partial_{e}\Gamma[\nabla]^{f}{}_{da} \right)$$

Out[∘]=
$$g_{bf}\ g^{ce}\ V^{d} \left(-\frac{1}{4} g^{fj}\ g^{ik} \left(\partial_{e} g_{ij} + \partial_{i} g_{ej} - \partial_{j} g_{ei} \right) \left(\partial_{a} g_{dk} + \partial_{d} g_{ak} - \partial_{k} g_{da} \right) + \frac{1}{4} g^{fl}\ g^{im} \left(\partial_{d} g_{il} + \partial_{i} g_{dl} - \partial_{l} g_{di} \right) \left(\partial_{a} g_{em} + \partial_{e} g_{am} - \partial_{m} g_{ea} \right) + \frac{1}{2} \left(g^{fo} \left(\partial_{d}\partial_{a} g_{eo} + \partial_{d}\partial_{e} g_{ao} - \partial_{d}\partial_{o} g_{ea} \right) - g^{fp}\ g^{oq}\ \partial_{d} g_{pq} \left(\partial_{a} g_{eo} + \partial_{e} g_{ao} - \partial_{o} g_{ea} \right) \right) + \frac{1}{2} \left(- g^{fr} \left(\partial_{e}\partial_{a} g_{dr} + \partial_{e}\partial_{d} g_{ar} - \partial_{e}\partial_{r} g_{da} \right) + g^{fs}\ g^{ru}\ \partial_{e} g_{su} \left(\partial_{a} g_{dr} + \partial_{d} g_{ar} - \partial_{r} g_{da} \right) \right) \right)$$

Out[∘]= 0

Ações e Equações de Movimento

In[∘]:=
```
Quit[]
```

Preparativos

Perturbações

In[•]:=
```
DefMetricPerturbation[g, gpert, ϵ];
PrintAs[gpert] ^= "h";
DefTensor[ϕ[], M, PrintAs → "φ"]
DefTensorPerturbation[Pertϕ[LI[order]], ϕ[], M, PrintAs → "δφ"]
DefScalarFunction[V]
DefScalarFunction[G]
DefTensor[A[-a], M](* Vector field *)
DefConstantSymbol[MPlanck, PrintAs → "mp"]
DefTensor[F[-a, -b], M, Antisymmetric[{-a, -b}]] (* Maxwell strength tensor*)
IndexSetDelayed[F[a_, b_], CD[a][A[b]] - CD[b][A[a]]]
DefTensorPerturbation[PertA[LI[order], -a], A[-a], M, PrintAs → "δA"]
DefScalarFunction[FR]
DefConstantSymbol[μ]
```

In[•]:=
```
eqMotion[Lag_, field_, deriv_] :=
  (Lag // Perturbation // ExpandPerturbation // ContractMetric // ToCanonical //
        VarD[field, deriv]) /. delta[-LI[1], LI[1]] → 1 //
     ToCanonical // Simplify // ContractMetric
```

Campo Escalar

In[•]:=
```
ℒescalar = Sqrt[-Detg[]] (1/2 CD[a]@ϕ[] CD[-a]@ϕ[] - V[ϕ[]])
0 == (ℒescalar // eqMotion[#, Pertϕ[LI[1]], CD] &)
```

Out[•]= $\sqrt{-\tilde{\tilde{g}}}\left(-V[\varphi]+\frac{1}{2}\nabla_a\varphi\,\nabla^a\varphi\right)$

Out[•]= $0 == -\sqrt{-\tilde{\tilde{g}}}\;\nabla_a\nabla^a\varphi-\sqrt{-\tilde{\tilde{g}}}\;V'[\varphi]$

Eletromagnetismo

In[•]:=
```
ℒmaxwell = Sqrt[-Detg[]] F[a, b] F[-a, -b] / 4
0 == (ℒmaxwell // eqMotion[#, PertA[LI[1], a], CD] &)
```

Out[•]= $\frac{1}{4}\sqrt{-\tilde{\tilde{g}}}\;(\nabla_a A_b-\nabla_b A_a)\;(\nabla^a A^b-\nabla^b A^a)$

Out[•]= $0 == \sqrt{-\tilde{\tilde{g}}}\;\nabla_b\nabla_a A^b-\sqrt{-\tilde{\tilde{g}}}\;\nabla_b\nabla^b A_a$

Einstein - Hilbert

In[∘]:=
```
ℒEinstein = Sqrt[-Detg[]] (MPlanck^2 / 2 RicciScalarCD[])
0 == (ℒEinstein // eqMotion[#, gpert[LI[1], a, b], CD] &)
% // RicciToEinstein // Simplification
```

Out[∘]= $\frac{1}{2} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; R[\nabla]$

Out[∘]= $0 == -\frac{1}{2} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; R[\nabla]_{ab} + \frac{1}{4} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; R[\nabla]$

Out[∘]= $0 == -\frac{1}{2} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; G[\nabla]_{ab}$

Outras Teorias de Gravitação

Gravitação F (R)

In[∘]:=
```
FR[expr_] := expr - μ/expr
ℒf = FR[RicciScalarCD[]];
(ℒf // eqMotion[#, gpert[LI[1], a, b], CD] &)
SeriesCoefficient[%, {μ, 0, 1}] // ContractMetric
```

Out[∘]= $-R[\nabla]_{ab} - \frac{\mu\; R[\nabla]_{ab}}{R[\nabla]^2} - \frac{\mu\, \nabla_a \nabla_b R[\nabla]}{R[\nabla]^3} + \frac{6\, \mu\, \nabla_a R[\nabla]\; \nabla_b R[\nabla]}{R[\nabla]^4} - \frac{\mu\, \nabla_b \nabla_a R[\nabla]}{R[\nabla]^3} + \frac{2\, \mu\; g_{ab}\; \nabla_c \nabla^c R[\nabla]}{R[\nabla]^3} - \frac{6\, \mu\; g_{ab}\; \nabla_c R[\nabla]\; \nabla^c R[\nabla]}{R[\nabla]^4}$

Out[∘]= $-\frac{R[\nabla]_{ab}}{R[\nabla]^2} - \frac{\nabla_a \nabla_b R[\nabla]}{R[\nabla]^3} + \frac{6\, \nabla_a R[\nabla]\; \nabla_b R[\nabla]}{R[\nabla]^4} - \frac{\nabla_b \nabla_a R[\nabla]}{R[\nabla]^3} + \frac{2\; g_{ab}\; \nabla_c \nabla^c R[\nabla]}{R[\nabla]^3} - \frac{6\; g_{ab}\; \nabla_c R[\nabla]\; \nabla^c R[\nabla]}{R[\nabla]^4}$

Quintessencia (Acoplamento mínimo e não mínimo)

In[∘]:=
```
LSTmin = ℒEinstein + Sqrt[-Detg[]] * (-1 / 2 CD[a][ϕ[]] CD[-a][ϕ[]] - V[ϕ[]])
(*Perturbação de primeira ordem*)
LSTmin // eqMotion[#, gpert[LI[1], a, b], CD] &
LSTmin // eqMotion[#, Pertϕ[LI[1]], CD] &
```

Out[∘]= $\frac{1}{2} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; R[\nabla] + \sqrt{-\tilde{\tilde{g}}} \left(-V[\varphi] - \frac{1}{2} \nabla_a \varphi\, \nabla^a \varphi\right)$

Out[∘]= $-\frac{1}{2} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; R[\nabla]_{ab} + \frac{1}{4} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; R[\nabla] - \frac{1}{2} \sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; V[\varphi] + \frac{1}{2} \sqrt{-\tilde{\tilde{g}}}\; \nabla_a \varphi\, \nabla_b \varphi - \frac{1}{4} \sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla_c \varphi\, \nabla^c \varphi$

Out[∘]= $\sqrt{-\tilde{\tilde{g}}}\; \nabla_a \nabla^a \varphi - \sqrt{-\tilde{\tilde{g}}}\; V'[\varphi]$

In[•]:=
```
DefScalarFunction[AA]
DefScalarFunction[BB]
LSTnmin =
 AA[ϕ[]]^(-2) ℒEinstein + Sqrt[-Detg[]] * (-BB[ϕ[]] CD[a][ϕ[]] CD[-a][ϕ[]] - V[ϕ[]])
eqR = LSTnmin // eqMotion[#, gpert[LI[1], a, b], CD] &
eqϕ = LSTnmin // eqMotion[#, Pertϕ[LI[1]], CD] &
```

Out[•]=
$$\frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, R[\nabla]}{2\, AA[\varphi]^2} + \sqrt{-\tilde{\tilde{g}}}\, \left(-V[\varphi] - BB[\varphi]\, \nabla_a\varphi\, \nabla^a\varphi\right)$$

Out[•]=
$$-\frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, R[\nabla]_{ab}}{2\, AA[\varphi]^2} + \frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, g_{ab}\, R[\nabla]}{4\, AA[\varphi]^2} - \frac{1}{2}\sqrt{-\tilde{\tilde{g}}}\, g_{ab}\, V[\varphi] +$$
$$BB[\varphi]\sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla_b\varphi - \frac{1}{2} BB[\varphi]\sqrt{-\tilde{\tilde{g}}}\, g_{ab}\, \nabla_c\varphi\, \nabla^c\varphi - \frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\nabla_b\varphi\, AA'[\varphi]}{2\, AA[\varphi]^3} -$$
$$\frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_b\nabla_a\varphi\, AA'[\varphi]}{2\, AA[\varphi]^3} + \frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, g_{ab}\, \nabla_c\nabla^c\varphi\, AA'[\varphi]}{AA[\varphi]^3} + \frac{3\, m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla_b\varphi\, AA'[\varphi]^2}{AA[\varphi]^4} -$$
$$\frac{3\, m_p^2 \sqrt{-\tilde{\tilde{g}}}\, g_{ab}\, \nabla_c\varphi\, \nabla^c\varphi\, AA'[\varphi]^2}{AA[\varphi]^4} - \frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla_b\varphi\, AA''[\varphi]}{AA[\varphi]^3} + \frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, g_{ab}\, \nabla_c\varphi\, \nabla^c\varphi\, AA''[\varphi]}{AA[\varphi]^3}$$

Out[•]=
$$2\, BB[\varphi]\sqrt{-\tilde{\tilde{g}}}\, \nabla_a\nabla^a\varphi - \frac{m_p^2 \sqrt{-\tilde{\tilde{g}}}\, R[\nabla]\, AA'[\varphi]}{AA[\varphi]^3} + \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla^a\varphi\, BB'[\varphi] - \sqrt{-\tilde{\tilde{g}}}\, V'[\varphi]$$

In[•]:=
```
(0 == g[a, b] eqR // ContractMetric // EinsteinToRicci // ToCanonical) //
 Solve[#, RicciScalarCD[]] &
0 == eqϕ /. (% // Flatten) // ToCanonical
```

Out[•]=
$$\Big\{\Big\{R[\nabla] \to \frac{1}{m_p^2\, AA[\varphi]^2}\, 2\, \big(2\, AA[\varphi]^4\, V[\varphi] + AA[\varphi]^4\, BB[\varphi]\, \nabla_a\varphi\, \nabla^a\varphi -$$
$$3\, m_p^2\, AA[\varphi]\, \nabla_a\nabla^a\varphi\, AA'[\varphi] + 9\, m_p^2\, \nabla_a\varphi\, \nabla^a\varphi\, AA'[\varphi]^2 - 3\, m_p^2\, AA[\varphi]\, \nabla_a\varphi\, \nabla^a\varphi\, AA''[\varphi]\big)\Big\}\Big\}$$

Out[•]=
$$0 == 2\, BB[\varphi]\sqrt{-\tilde{\tilde{g}}}\, \nabla_a\nabla^a\varphi - \frac{4\sqrt{-\tilde{\tilde{g}}}\, V[\varphi]\, AA'[\varphi]}{AA[\varphi]} -$$
$$\frac{2\, BB[\varphi]\sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla^a\varphi\, AA'[\varphi]}{AA[\varphi]} + \frac{6\, m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\nabla^a\varphi\, AA'[\varphi]^2}{AA[\varphi]^4} - \frac{18\, m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla^a\varphi\, AA'[\varphi]^3}{AA[\varphi]^5} +$$
$$\sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla^a\varphi\, BB'[\varphi] - \sqrt{-\tilde{\tilde{g}}}\, V'[\varphi] + \frac{6\, m_p^2 \sqrt{-\tilde{\tilde{g}}}\, \nabla_a\varphi\, \nabla^a\varphi\, AA'[\varphi]\, AA''[\varphi]}{AA[\varphi]^4}$$

Galileons

In[•]:=
```
DefConstantSymbol[L3, PrintAs → "Λ3"]
DefConstantSymbol[L4, PrintAs → "Λ4"]
DefTensor[X[], M, PrintAs → "X"]
IndexSet[X[], -1/2 CD[a][ϕ[]] CD[-a][ϕ[]]]
```

Out[•]=
$$-\frac{1}{2}\nabla_a\varphi\,\nabla^a\varphi$$

In[•]:=
```
LGal = Sqrt[-Detg[]] * (X[] + 1/(2 * L3^3) * X[] * CD[-b][CD[b][ϕ[]]] - (2 X[]/L4^6) *
      (X[] * RicciScalarCD[] + 2 * (CD[-a][CD[a][ϕ[]]] CD[-b][CD[b][ϕ[]]] -
           CD[-a][CD[-b][ϕ[]]] CD[a][CD[b][ϕ[]]])) + MPlanck^2/2 * RicciScalarCD[])
eqRgal = LGal // eqMotion[#, gpert[LI[1], a, b], CD] &
eqϕgal = LGal // eqMotion[#, Pertϕ[LI[1]], CD] &
```

Out[•]=
$$\sqrt{-\tilde{\tilde{g}}}\left(\frac{m_p{}^2\,R[\nabla]}{2}-\frac{1}{2}\nabla_c\varphi\,\nabla^c\varphi-\frac{\nabla_b\nabla^b\varphi\,\nabla_d\varphi\,\nabla^d\varphi}{4\,\Lambda_3{}^3}+\frac{\nabla_e\varphi\,\nabla^e\varphi\left(2\left(-\nabla_a\nabla_b\varphi\,\nabla^a\nabla^b\varphi+\nabla_a\nabla^a\varphi\,\nabla_b\nabla^b\varphi\right)-\frac{1}{2}R[\nabla]\,\nabla_f\varphi\,\nabla^f\varphi\right)}{\Lambda_4{}^6}\right)$$

Out[∘]=

$$
\begin{aligned}
&-\frac{1}{2} m_p^2 \sqrt{-\tilde{\tilde{g}}}\; R[\nabla]_{ab} + \frac{1}{4} m_p^2 \sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; R[\nabla] + \frac{1}{2} \sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\varphi + \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\varphi\, \nabla_c\nabla^c\varphi}{4\,\Lambda_3^3} - \\
&\frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_c\varphi\, \nabla_b\varphi\, \nabla^c\varphi}{4\,\Lambda_3^3} - \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\nabla_c\varphi\, \nabla^c\varphi}{4\,\Lambda_3^3} - \frac{1}{4} \sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla_c\varphi\, \nabla^c\varphi + \frac{\sqrt{-\tilde{\tilde{g}}}\; R[\nabla]\; \nabla_a\varphi\, \nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi}{\Lambda_4^6} + \\
&\frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_d\nabla^d\varphi\, \nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi}{\Lambda_4^6} + \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\nabla_d\nabla^d\varphi\, \nabla_c\varphi\, \nabla^c\varphi}{\Lambda_4^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla_a\nabla^d\varphi}{\Lambda_4^6} - \\
&\frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla_b\nabla^d\varphi}{\Lambda_4^6} - \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\varphi\, \nabla_c\nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4^6} + \frac{4\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_c\varphi\, \nabla_b\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4^6} + \\
&\frac{4\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\nabla_a\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4^6} - \\
&\frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\nabla_a\varphi}{\Lambda_4^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\nabla_b\varphi}{\Lambda_4^6} - \frac{4\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_c\varphi\, \nabla_b\nabla_d\varphi\, \nabla^c\varphi\, \nabla^d\varphi}{\Lambda_4^6} - \\
&\frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_b\nabla_d\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi}{\Lambda_4^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\nabla_a\nabla_d\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi}{\Lambda_4^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\; R[\nabla]_{ab}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\varphi\, \nabla^d\varphi}{2\,\Lambda_4^6} - \\
&\frac{\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; R[\nabla]\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\varphi\, \nabla^d\varphi}{4\,\Lambda_4^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla_a\nabla_b\varphi\, \nabla^d\varphi}{\Lambda_4^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla_b\nabla_a\varphi\, \nabla^d\varphi}{\Lambda_4^6} + \\
&\frac{\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi}{4\,\Lambda_3^3} + \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_b\varphi\, \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi}{\Lambda_4^6} + \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\nabla_a\varphi\, \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi}{\Lambda_4^6} - \\
&\frac{2\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla_e\nabla^e\varphi\, \nabla^d\varphi}{\Lambda_4^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\nabla_d\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\nabla_a\varphi}{\Lambda_4^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_d\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\nabla_b\varphi}{\Lambda_4^6} - \\
&\frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\nabla_d\varphi\, \nabla_b\varphi\, \nabla^c\varphi\, \nabla^d\nabla_c\varphi}{\Lambda_4^6} - \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\nabla_d\varphi\, \nabla^c\varphi\, \nabla^d\nabla_c\varphi}{\Lambda_4^6} - \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_b\varphi\, \nabla^c\varphi\, \nabla_d\nabla_a\varphi\, \nabla^d\nabla_c\varphi}{\Lambda_4^6} - \\
&\frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla^c\varphi\, \nabla_d\nabla_b\varphi\, \nabla^d\nabla_c\varphi}{\Lambda_4^6} + \frac{2\sqrt{-\tilde{\tilde{g}}}\; \nabla_a\varphi\, \nabla_b\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\nabla^c\varphi}{\Lambda_4^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi\, \nabla_e\nabla^e\varphi}{\Lambda_4^6} - \\
&\frac{4\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi\, \nabla_e\nabla^e\varphi}{\Lambda_4^6} + \frac{2\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi\, \nabla_e\nabla^e\nabla_d\varphi}{\Lambda_4^6} + \\
&\frac{4\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla^c\varphi\, \nabla^d\varphi\, \nabla_e\nabla_d\varphi\, \nabla^e\nabla_c\varphi}{\Lambda_4^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\; g_{ab}\; \nabla_c\varphi\, \nabla^c\varphi\, \nabla_e\nabla_d\varphi\, \nabla^e\nabla^d\varphi}{\Lambda_4^6}
\end{aligned}
$$

Out[∘]=

$$
\begin{aligned}
&\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi+\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla_b\nabla^b\varphi\ \nabla^a\varphi}{2\,\Lambda_3{}^3}+\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi\ \nabla_b\nabla^b\varphi}{2\,\Lambda_3{}^3}+\\
&\frac{2\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]\ \nabla_a\varphi\ \nabla^a\varphi\ \nabla_b\nabla^b\varphi}{\Lambda_4{}^6}-\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla_c\nabla^c\varphi\ \nabla^a\varphi\ \nabla_b\nabla^b\varphi}{\Lambda_4{}^6}-\\
&\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla^a\varphi\ \nabla_b\nabla^b\nabla_a\varphi}{2\,\Lambda_3{}^3}-\frac{4\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\ \nabla^a\varphi\ \nabla_b\nabla_c\nabla^c\nabla^b\varphi}{\Lambda_4{}^6}+\frac{2\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\ \nabla^a R[\nabla]\ \nabla_b\varphi\ \nabla^b\varphi}{\Lambda_4{}^6}+\\
&\frac{4\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]\ \nabla^a\varphi\ \nabla_b\nabla_a\varphi\ \nabla^b\varphi}{\Lambda_4{}^6}+\frac{16\sqrt{-\tilde{\tilde{g}}}\ \nabla^a\varphi\ \nabla_b\nabla_c\nabla^c\varphi\ \nabla^b\nabla_a\varphi}{\Lambda_4{}^6}-\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_b\nabla_a\varphi\ \nabla^b\nabla^a\varphi}{2\,\Lambda_3{}^3}-\\
&\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla^a\varphi\ \nabla^b\nabla_a\varphi\ \nabla_c\nabla_b\nabla^c\varphi}{\Lambda_4{}^6}-\frac{4\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi\ \nabla_b\nabla^b\varphi\ \nabla_c\nabla^c\varphi}{\Lambda_4{}^6}+\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla^a\varphi\ \nabla_b\nabla^b\varphi\ \nabla_c\nabla^c\nabla_a\varphi}{\Lambda_4{}^6}-\\
&\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla^a\varphi\ \nabla^b\nabla_a\varphi\ \nabla_c\nabla^c\nabla_b\varphi}{\Lambda_4{}^6}+\frac{4\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\ \nabla^a\varphi\ \nabla_c\nabla^c\nabla_b\nabla^b\varphi}{\Lambda_4{}^6}-\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla^b\nabla^a\varphi\ \nabla_c\nabla_b\varphi\ \nabla^c\nabla_a\varphi}{\Lambda_4{}^6}+\\
&\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla_c\nabla_b\varphi\ \nabla^a\varphi\ \nabla^c\nabla^b\varphi}{\Lambda_4{}^6}-\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla^a\varphi\ \nabla_b\nabla_c\nabla_a\varphi\ \nabla^c\nabla^b\varphi}{\Lambda_4{}^6}+\frac{12\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi\ \nabla_c\nabla_b\varphi\ \nabla^c\nabla^b\varphi}{\Lambda_4{}^6}
\end{aligned}
$$

In[∘]:=

```
eqϕgal2 = ContractMetric[SortCovDs[eqϕgal, CD], AllowUpperDerivatives → True] //
   ReplaceDummies // ToCanonical
```

Out[∘]=

$$
\begin{aligned}
&\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi+\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi\ \nabla_b\nabla^b\varphi}{2\,\Lambda_3{}^3}+\frac{2\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]\ \nabla_a\varphi\ \nabla^a\varphi\ \nabla_b\nabla^b\varphi}{\Lambda_4{}^6}-\\
&\frac{\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{ab}\ \nabla^a\varphi\ \nabla^b\varphi}{2\,\Lambda_3{}^3}+\frac{2\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\ \nabla^a R[\nabla]\ \nabla_b\varphi\ \nabla^b\varphi}{\Lambda_4{}^6}+\frac{4\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]\ \nabla^a\varphi\ \nabla_b\nabla_a\varphi\ \nabla^b\varphi}{\Lambda_4{}^6}-\\
&\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_b\nabla_a\varphi\ \nabla^b\nabla^a\varphi}{2\,\Lambda_3{}^3}-\frac{4\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\ \nabla^a\varphi\ \nabla^b\varphi\ \nabla_c R[\nabla]_b{}^c}{\Lambda_4{}^6}-\frac{4\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi\ \nabla_b\nabla^b\varphi\ \nabla_c\nabla^c\varphi}{\Lambda_4{}^6}+\\
&\frac{8\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{ab}\ \nabla^a\varphi\ \nabla^b\varphi\ \nabla_c\nabla^c\varphi}{\Lambda_4{}^6}-\frac{16\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{bc}\ \nabla^a\varphi\ \nabla^b\varphi\ \nabla^c\nabla_a\varphi}{\Lambda_4{}^6}-\frac{8\sqrt{-\tilde{\tilde{g}}}\ \nabla^b\nabla^a\varphi\ \nabla_c\nabla_b\varphi\ \nabla^c\nabla_a\varphi}{\Lambda_4{}^6}-\\
&\frac{4\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{bc}\ \nabla_a\varphi\ \nabla^a\varphi\ \nabla^c\nabla^b\varphi}{\Lambda_4{}^6}+\frac{12\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\nabla^a\varphi\ \nabla_c\nabla_b\varphi\ \nabla^c\nabla^b\varphi}{\Lambda_4{}^6}-\frac{8\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{acbd}\ \nabla^a\varphi\ \nabla^b\varphi\ \nabla^d\nabla^c\varphi}{\Lambda_4{}^6}
\end{aligned}
$$

In[∘]:=

```
BianchiID = {CD[-a_]@ϕ[] * CD[-b_]@RicciCD[b_, a_] ⧴
   1/2 CD[-a]@ϕ[] g[b, a] CD[-b]@RicciScalarCD[],
  CD[a_][ϕ[]] * CD[-b_][RicciCD[-a_, b_]] ⧴ 1/2 CD[a]@ϕ[] * CD[-a]@RicciScalarCD[]}
```

Out[∘]=

$$
\left\{\nabla_{\underline{a}}\varphi\ \nabla_{\underline{b}}R[\nabla]^{\underline{b}\,\underline{a}}\ ⧴\ \frac{1}{2}\nabla_a\varphi\ g^{ba}\ \nabla_b R[\nabla],\ \nabla^{\underline{a}}\varphi\ \nabla_{\underline{b}}R[\nabla]_{\underline{a}}{}^{\underline{b}}\ ⧴\ \frac{1}{2}\nabla^a\varphi\ \nabla_a R[\nabla]\right\}
$$

In[∘]:=

```
eqRgal2 = (eqRgal /. BianchiID) // SortCovDs // ToCanonical
```

Out[∘]=

$$-\frac{1}{2} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{ab} + \frac{1}{4} m_p{}^2 \sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ R[\nabla] + \frac{1}{2} \sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla_b\varphi +$$

$$\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla_b\varphi\, \nabla_c\nabla^c\varphi}{4\, \Lambda_3{}^3} - \frac{1}{4} \sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ \nabla_c\varphi\, \nabla^c\varphi + \frac{\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]\ \nabla_a\varphi\, \nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi}{\Lambda_4{}^6} -$$

$$\frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_b\varphi\, \nabla_c\nabla_a\varphi\, \nabla^c\varphi}{4\, \Lambda_3{}^3} - \frac{\sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla_c\nabla_b\varphi\, \nabla^c\varphi}{4\, \Lambda_3{}^3} - \frac{2 \sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla_b\varphi\, \nabla_c\nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4{}^6} +$$

$$\frac{2 \sqrt{-\tilde{\tilde{g}}}\ \nabla_b\nabla_a\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4{}^6} + \frac{4 \sqrt{-\tilde{\tilde{g}}}\ \nabla_b\varphi\, \nabla_c\nabla_a\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4{}^6} + \frac{4 \sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla_c\nabla_b\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi}{\Lambda_4{}^6} -$$

$$\frac{2 \sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{bd}\ \nabla_a\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi}{\Lambda_4{}^6} - \frac{2 \sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{ad}\ \nabla_b\varphi\, \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi}{\Lambda_4{}^6} +$$

$$\frac{\sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{ab}\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\varphi\, \nabla^d\varphi}{2\, \Lambda_4{}^6} - \frac{\sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ R[\nabla]\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\varphi\, \nabla^d\varphi}{4\, \Lambda_4{}^6} - \frac{4 \sqrt{-\tilde{\tilde{g}}}\ \nabla_c\nabla_a\varphi\, \nabla^c\varphi\, \nabla_d\nabla_b\varphi\, \nabla^d\varphi}{\Lambda_4{}^6} +$$

$$\frac{\sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi}{4\, \Lambda_3{}^3} + \frac{4 \sqrt{-\tilde{\tilde{g}}}\ \nabla_b\nabla_a\varphi\, \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi}{\Lambda_4{}^6} - \frac{2 \sqrt{-\tilde{\tilde{g}}}\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla_b\varphi\, \nabla^d\nabla_a\varphi}{\Lambda_4{}^6} -$$

$$\frac{4 \sqrt{-\tilde{\tilde{g}}}\ \nabla_b\varphi\, \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\nabla_a\varphi}{\Lambda_4{}^6} - \frac{4 \sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\nabla_b\varphi}{\Lambda_4{}^6} + \frac{2 \sqrt{-\tilde{\tilde{g}}}\ \nabla_a\varphi\, \nabla_b\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\nabla^c\varphi}{\Lambda_4{}^6} -$$

$$\frac{\sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla_d\nabla^d\varphi\, \nabla_e\nabla^e\varphi}{\Lambda_4{}^6} - \frac{4 \sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ \nabla^c\varphi\, \nabla_d\nabla_c\varphi\, \nabla^d\varphi\, \nabla_e\nabla^e\varphi}{\Lambda_4{}^6} +$$

$$\frac{2 \sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ R[\nabla]_{de}\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi\, \nabla^e\varphi}{\Lambda_4{}^6} - \frac{2 \sqrt{-\tilde{\tilde{g}}}\ R[\nabla]_{adbe}\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla^d\varphi\, \nabla^e\varphi}{\Lambda_4{}^6} +$$

$$\frac{4 \sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ \nabla^c\varphi\, \nabla^d\varphi\, \nabla_e\nabla_d\varphi\, \nabla^e\nabla_c\varphi}{\Lambda_4{}^6} + \frac{\sqrt{-\tilde{\tilde{g}}}\ g_{ab}\ \nabla_c\varphi\, \nabla^c\varphi\, \nabla_e\nabla_d\varphi\, \nabla^e\nabla^d\varphi}{\Lambda_4{}^6}$$

Métrica de Schwarzshild

In[∘]:=

```
UndefTensor[M]
```

... UndefTensor : Unknown tensor M.

In[∘]:=

```
DefScalarFunction[F]
DefConstantSymbol[Ma]
DefConstantSymbol[G]

F[r_] := (1 - 2 Ma G/r)
DefChart[coordS, M, {0, 1, 2, 3}, {t[], r[], θ[], ϕ[]}, ChartColor → Red]
```

... ValidateSymbol: Symbol F is already used as a tensor.

... ValidateSymbol: Symbol G is already used as a scalar function.

In[∘]:=

```
MatrixForm[MatrixSW = DiagonalMatrix[{-F[r[]], 1/F[r[]], r[]^2, r[]^2 Sin[θ[]]^2}]]
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} -1 + \frac{2\,G\,Ma}{r} & 0 & 0 & 0 \\ 0 & \frac{1}{1-\frac{2\,G\,Ma}{r}} & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2\,\mathrm{Sin}[\theta]^2 \end{pmatrix}$$

In[∘]:=

```
MatrixForm@MetricInBasis[g, -coordS, MatrixSW]
MetricCompute[g, coordS, All, Verbose → False]
```

Added independent rule $g_{00} \to -1 + \frac{2\,G\,Ma}{r}$ for tensor g

Added independent rule $g_{01} \to 0$ for tensor g

Added independent rule $g_{02} \to 0$ for tensor g

Added independent rule $g_{03} \to 0$ for tensor g

Added dependent rule $g_{10} \to g_{01}$ for tensor g

Added independent rule $g_{11} \to \frac{1}{1 - \frac{2\,G\,Ma}{r}}$ for tensor g

Added independent rule $g_{12} \to 0$ for tensor g

Added independent rule $g_{13} \to 0$ for tensor g

Added dependent rule $g_{20} \to g_{02}$ for tensor g

Added dependent rule $g_{21} \to g_{12}$ for tensor g

Added independent rule $g_{22} \to r^2$ for tensor g

Added independent rule $g_{23} \to 0$ for tensor g

Added dependent rule $g_{30} \to g_{03}$ for tensor g

Added dependent rule $g_{31} \to g_{13}$ for tensor g

Added dependent rule $g_{32} \to g_{23}$ for tensor g

Added independent rule $g_{33} \to r^2\,\mathrm{Sin}[\theta]^2$ for tensor g

Out[∘]//MatrixForm=

$$\begin{pmatrix} g_{00} \to -1 + \frac{2\,G\,Ma}{r} & g_{01} \to 0 & g_{02} \to 0 & g_{03} \to 0 \\ g_{10} \to 0 & g_{11} \to \frac{1}{1-\frac{2\,G\,Ma}{r}} & g_{12} \to 0 & g_{13} \to 0 \\ g_{20} \to 0 & g_{21} \to 0 & g_{22} \to r^2 & g_{23} \to 0 \\ g_{30} \to 0 & g_{31} \to 0 & g_{32} \to 0 & g_{33} \to r^2\,\mathrm{Sin}[\theta]^2 \end{pmatrix}$$

In[∘]:=

```
(*Função para recuperar valores dos tensores*)
MyArrayComponents[expr_] :=
 expr // ToBasis[coord2] // ComponentArray // ToValues // ToValues // Simplify
```

In[∘]:=

```
ChristoffelCDPDcoordS[{0, coordS}, -{a, coordS}, -{b, coordS}] // ComponentArray //
  ToValues // MatrixForm
ChristoffelCDPDcoordS[{1, coordS}, -{a, coordS}, -{b, coordS}] // ComponentArray //
  ToValues // MatrixForm
ChristoffelCDPDcoordS[{2, coordS}, -{a, coordS}, -{b, coordS}] // ComponentArray //
  ToValues // MatrixForm
ChristoffelCDPDcoordS[{3, coordS}, -{a, coordS}, -{b, coordS}] // ComponentArray //
  ToValues // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & -\frac{G\,Ma}{2\,G\,Ma\,r-r^2} & 0 & 0 \\ -\frac{G\,Ma}{2\,G\,Ma\,r-r^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Out[∘]//MatrixForm=

$$\begin{pmatrix} -\frac{G\,Ma\,(2\,G\,Ma-r)}{r^3} & 0 & 0 & 0 \\ 0 & \frac{G\,Ma}{2\,G\,Ma\,r-r^2} & 0 & 0 \\ 0 & 0 & 2\,G\,Ma-r & 0 \\ 0 & 0 & 0 & (2\,G\,Ma-r)\,\mathrm{Sin}[\theta]^2 \end{pmatrix}$$

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{r} & 0 \\ 0 & \frac{1}{r} & 0 & 0 \\ 0 & 0 & 0 & -\mathrm{Cos}[\theta]\,\mathrm{Sin}[\theta] \end{pmatrix}$$

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{r} \\ 0 & 0 & 0 & \mathrm{Cot}[\theta] \\ 0 & \frac{1}{r} & \mathrm{Cot}[\theta] & 0 \end{pmatrix}$$

In[∘]:=

```
EinsteinCD[-{a, coordS}, -{b, coordS}] // ComponentArray // ToValues // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Precessão de Mercúrio

In[∘]:=

```
Quit[]
```

In[∘]:=

```
<< VariationalMethods`
CorGR = Purple;
CorCl = Brown;
```

In[∘]:=

```
gGR = DiagonalMatrix[{-f[r[τ]], 1/f[r[τ]], r[τ]^2, r[τ]^2 Sin[θ[τ]]^2}];
gCL = DiagonalMatrix[{1, r[τ]^2, r[τ]^2 Sin[θ[τ]]^2}];
XGR = {t[τ], r[τ], θ[τ], ϕ[τ]};
XCL = {r[τ], θ[τ], ϕ[τ]};
V[expr_] := D[#, τ] & /@ expr;
LGR = V[XGR].gGR.V[XGR];
LCL = 1/2 V[XCL].gCL.V[XCL] - Φ[r[τ]];
TraditionalForm@MatrixForm@gGR
LGR // Style[#, FontColor → CorGR] & // TraditionalForm
LCL // Style[#, FontColor → CorCl] & // TraditionalForm
```

Out[∘]//TraditionalForm=

$$\begin{pmatrix} -f(r(\tau)) & 0 & 0 & 0 \\ 0 & \frac{1}{f(r(\tau))} & 0 & 0 \\ 0 & 0 & r(\tau)^2 & 0 \\ 0 & 0 & 0 & r(\tau)^2 \sin^2(\theta(\tau)) \end{pmatrix}$$

Out[∘]//TraditionalForm=

$$\frac{r'(\tau)^2}{f(r(\tau))} - f(r(\tau))\, t'(\tau)^2 + r(\tau)^2\, \theta'(\tau)^2 + r(\tau)^2 \sin^2(\theta(\tau))\, \phi'(\tau)^2$$

Out[∘]//TraditionalForm=

$$\frac{1}{2}\left(r'(\tau)^2 + r(\tau)^2\, \theta'(\tau)^2 + r(\tau)^2 \sin^2(\theta(\tau))\, \phi'(\tau)^2\right) - \Phi(r(\tau))$$

Temos um pacote para fazer a variação da lagrangeana e obter as equações de movimento

In[•]:=
```
EOMGR = EulerEquations[LGR, #, τ] & /@ XGR;
EOMGR // MatrixForm // TraditionalForm //
  Style[#, FontColor → CorGR] & // TraditionalForm
EOMCL = EulerEquations[LCL, #, τ] & /@ XCL;
EOMCL // MatrixForm // TraditionalForm //
  Style[#, FontColor → CorCl] & // TraditionalForm
```

Out[•]//TraditionalForm=

$$\begin{pmatrix} 2\,(f'(r(\tau))\, r'(\tau)\, t'(\tau) + f(r(\tau))\, t''(\tau)) = 0 \\ \frac{f'(r(\tau))\, r'(\tau)^2}{f(r(\tau))^2} - f'(r(\tau))\, t'(\tau)^2 + 2\, r(\tau)\left(\theta'(\tau)^2 + \sin^2(\theta(\tau))\, \phi'(\tau)^2\right) - \frac{2\, r''(\tau)}{f(r(\tau))} = 0 \\ r(\tau)\left(r(\tau)\left(\sin(2\,\theta(\tau))\, \phi'(\tau)^2 - 2\,\theta''(\tau)\right) - 4\, r'(\tau)\, \theta'(\tau)\right) = 0 \\ -2\, r(\tau) \sin(\theta(\tau))\,(2 \sin(\theta(\tau))\, r'(\tau)\, \phi'(\tau) + r(\tau)\,(2 \cos(\theta(\tau))\, \theta'(\tau)\, \phi'(\tau) + \sin(\theta(\tau))\, \phi''(\tau))) = 0 \end{pmatrix}$$

Out[•]//TraditionalForm=

$$\begin{pmatrix} r(\tau)\left(\theta'(\tau)^2 + \sin^2(\theta(\tau))\, \phi'(\tau)^2\right) - \Phi'(r(\tau)) - r''(\tau) = 0 \\ r(\tau)\left(r(\tau)\left(\cos(\theta(\tau)) \sin(\theta(\tau))\, \phi'(\tau)^2 - \theta''(\tau)\right) - 2\, r'(\tau)\, \theta'(\tau)\right) = 0 \\ -r(\tau) \sin(\theta(\tau))\,(2 \sin(\theta(\tau))\, r'(\tau)\, \phi'(\tau) + r(\tau)\,(2 \cos(\theta(\tau))\, \theta'(\tau)\, \phi'(\tau) + \sin(\theta(\tau))\, \phi''(\tau))) = 0 \end{pmatrix}$$

as grandezas conservadas também podem ser obtidas

In[•]:=
```
JcGR = FirstIntegrals[LGR, {t[τ], θ[τ], ϕ[τ]}, τ];
JcCL = FirstIntegrals[LCL, {θ[τ], ϕ[τ]}, τ];
JcGR // MatrixForm // TraditionalForm // Style[#, FontColor → CorGR] & //
 TraditionalForm
JcCL // MatrixForm // TraditionalForm // Style[#, FontColor → CorCl] & //
 TraditionalForm
```

Out[•]//TraditionalForm=

$$\begin{pmatrix} \text{FirstIntegral}(t) \to 2\, f(r(\tau))\, t'(\tau) \\ \text{FirstIntegral}(\phi) \to -2\, r(\tau)^2 \sin^2(\theta(\tau))\, \phi'(\tau) \\ \text{FirstIntegral}(\tau) \to \left(\theta'(\tau)^2 + \sin^2(\theta(\tau))\, \phi'(\tau)^2\right) r(\tau)^2 - \frac{r'(\tau)^2}{f(r(\tau))} - f(r(\tau))\, t'(\tau)^2 \end{pmatrix}$$

Out[•]//TraditionalForm=

$$\begin{pmatrix} \text{FirstIntegral}(\phi) \to -r(\tau)^2 \sin^2(\theta(\tau))\, \phi'(\tau) \\ \text{FirstIntegral}(\tau) \to \Phi(r(\tau)) + \frac{1}{2}\left(r(\tau)^2\left(\theta'(\tau)^2 + \sin^2(\theta(\tau))\, \phi'(\tau)^2\right) - r'(\tau)^2\right) \end{pmatrix}$$

A simetria do problema permite que escolhamos o movimento no plano $\theta = \pi/2$. Simplificamos as equações com o uso das integrais de movimento. (A Hamiltoniana parece vir com um sinal errado!)

In[∘]:=
```
EOMeqGR = EOMGR /. {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;

EnergiaGR =
  E == 1/2 FirstIntegral[t] /. JcGR /. {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
momGR = Jϕ == -1/2 FirstIntegral[ϕ] /. JcGR /.
    {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
HamGR = -1 == LGR /. {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
EOMeqCL = EOMCL /. {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
EnergiaCL =
  Ec == 1/2 FirstIntegral[t] /. JcCL /. {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
momCL = Jϕ == -FirstIntegral[ϕ] /. JcCL /.
    {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
HamCL = 2 Φ[τ] + E == LCL /. {θ[τ] → π/2, θ'[τ] → 0, θ''[τ] → 0} // Simplify;
```

As equações de movimento são :

In[∘]:=
```
EOMeqGR // MatrixForm // TraditionalForm //
  Style[#, FontWeight → "Bold", FontColor → CorGR] &
EOMeqCL // MatrixForm // TraditionalForm //
  Style[#, FontWeight → "Bold", FontColor → CorCl] &
```

Out[∘]=
$$\begin{pmatrix} r'(\tau)\, t'(\tau)\, f'(r(\tau)) + f(r(\tau))\, t''(\tau) = 0 \\ f(r(\tau))\left(t'(\tau)^2 f'(r(\tau)) - 2\, r(\tau)\, \phi'(\tau)^2\right) + 2\, r''(\tau) = \frac{r'(\tau)^2 f'(r(\tau))}{f(r(\tau))} \\ \text{True} \\ r(\tau)\left(2\, r'(\tau)\, \phi'(\tau) + r(\tau)\, \phi''(\tau)\right) = 0 \end{pmatrix}$$

Out[∘]=
$$\begin{pmatrix} r(\tau)\, \phi'(\tau)^2 = r''(\tau) + \Phi'(r(\tau)) \\ \text{True} \\ r(\tau)\left(2\, r'(\tau)\, \phi'(\tau) + r(\tau)\, \phi''(\tau)\right) = 0 \end{pmatrix}$$

In[∘]:=
```
EnergiaGR // TraditionalForm // TraditionalForm // Style[#, FontColor → CorGR] &
EnergiaCL // TraditionalForm // TraditionalForm // Style[#, FontColor → CorCl] &
```

Out[∘]= $E = f(r(\tau))\, t'(\tau)$

Out[∘]= $2\, Ec = \text{FirstIntegral}(t)$

In[•]:=
```
momGR // TraditionalForm // TraditionalForm // TraditionalForm //
  Style[#, FontColor → CorGR] &
momCL // TraditionalForm // TraditionalForm // TraditionalForm //
  Style[#, FontColor → CorCl] &
HamGR // TraditionalForm // TraditionalForm // TraditionalForm //
  Style[#, FontColor → CorGR] &
HamCL // TraditionalForm // TraditionalForm // TraditionalForm //
  Style[#, FontColor → CorCl] &
```

Out[•]= $J\phi = r(\tau)^2\,\phi'(\tau)$

Out[•]= $J\phi = r(\tau)^2\,\phi'(\tau)$

Out[•]= $f(r(\tau))\,t'(\tau)^2 = \frac{r'(\tau)^2}{f(r(\tau))} + r(\tau)^2\,\phi'(\tau)^2 + 1$

Out[•]= $2\left(\mathrm{E} + 2\,\Phi(\tau) + \Phi(r(\tau))\right) = r'(\tau)^2 + r(\tau)^2\,\phi'(\tau)^2$

In[•]:=
```
mom1GR = Solve[momGR, ϕ'[τ]] // Flatten
enGR = Solve[EnergiaGR, t'[τ]] // Flatten
mom1CL = Solve[momCL, ϕ'[τ]] // Flatten
```

Out[•]= $\left\{\phi'[\tau] \to \frac{J\phi}{r[\tau]^2}\right\}$

Out[•]= $\left\{t'[\tau] \to \frac{\mathrm{E}}{f[r[\tau]]}\right\}$

Out[•]= $\left\{\phi'[\tau] \to \frac{J\phi}{r[\tau]^2}\right\}$

In[•]:=
```
eq1GR = HamGR /. mom1GR /. enGR // Simplify;
eq2GR = EOMeqGR[[1]] /. mom1GR /. enGR // Simplify;
% // TraditionalForm // Style[#, FontColor → CorGR] &
%%% // TraditionalForm // Style[#, FontColor → CorGR] &
eq1CL = HamCL /. mom1CL // Simplify;
eq2CL = EOMeqCL[[1]] /. mom1CL // Simplify;
% // TraditionalForm // Style[#, FontColor → CorCl] &
%%% // TraditionalForm // Style[#, FontColor → CorCl] &
```

Out[•]= $\frac{\mathrm{E}\,r'(\tau)\,f'(r(\tau))}{f(r(\tau))} + f(r(\tau))\,t''(\tau) = 0$

Out[•]= $\frac{\mathrm{E}^2}{f(r(\tau))} = \frac{r'(\tau)^2}{f(r(\tau))} + \frac{J\phi^2}{r(\tau)^2} + 1$

Out[•]= $\frac{J\phi^2}{r(\tau)^3} = r''(\tau) + \Phi'(r(\tau))$

Out[•]= $2\left(\mathrm{E} + \Phi(r(\tau)) + 2\,\Phi(\tau)\right) = \frac{J\phi^2}{r(\tau)^2} + r'(\tau)^2$

In[∘]:=
```
eq1GR /. r'[τ] → D[r[ϕ], ϕ] ϕ'[τ] /. mom1GR /. r[τ] → r[ϕ];
eqEnergiaGR = Times[#, f[r[ϕ]]] & /@ % // Expand;
eqEnergiaCL = eq1CL /. r'[τ] → D[r[ϕ], ϕ] ϕ'[τ] /. mom1GR /. r[τ] → r[ϕ];
%% // TraditionalForm // Style[#, FontColor → CorGR] &
%% // TraditionalForm // Style[#, FontColor → CorCl] &
```

Out[∘]= $$\mathrm{E}^2 = \frac{J\phi^2\, f(r(\phi))}{r(\phi)^2} + f(r(\phi)) + \frac{J\phi^2\, r'(\phi)^2}{r(\phi)^4}$$

Out[∘]= $$2\left(\mathrm{E} + \Phi(r(\phi)) + 2\,\Phi(\tau)\right) = \frac{J\phi^2\, r'(\phi)^2}{r(\phi)^4} + \frac{J\phi^2}{r(\phi)^2}$$

In[∘]:=
```
Φ[expr_] := - Gr m / expr
f[expr_] := (1 - 2 m Gr / expr)
```

In[∘]:=
```
eqEnergiaGR // Expand // TraditionalForm // Style[#, FontColor → CorGR] &
eqEnergiaCL // TraditionalForm // Style[#, FontColor → CorCl] &
```

Out[∘]= $$\mathrm{E}^2 = -\frac{2\,\mathrm{Gr}\, J\phi^2\, m}{r(\phi)^3} - \frac{2\,\mathrm{Gr}\, m}{r(\phi)} + \frac{J\phi^2\, r'(\phi)^2}{r(\phi)^4} + \frac{J\phi^2}{r(\phi)^2} + 1$$

Out[∘]= $$2\left(\mathrm{E} - \frac{\mathrm{Gr}\, m}{r(\phi)} - \frac{2\,\mathrm{Gr}\, m}{\tau}\right) = \frac{J\phi^2\, r'(\phi)^2}{r(\phi)^4} + \frac{J\phi^2}{r(\phi)^2}$$

Podemos escrever uma expressão para r (ou u) e ϕ em forma fechada como uma integral elíptica

In[∘]:=
```
eqEnergiaGR /. r'[ϕ] → -u'[ϕ] r[ϕ]^2 /. r[ϕ] → 1/u[ϕ];
binetGR = Times[#, 1/Jϕ^2] & /@ % // Expand;
eqEnergiaCL /. r'[ϕ] → -u'[ϕ] r[ϕ]^2 /. r[ϕ] → 1/u[ϕ];
binetCL = Times[#, 1/Jϕ^2] & /@ % // Expand;
soluGR = Solve[binetGR, u'[ϕ]][[2]];
% // TraditionalForm // Style[#, FontColor → CorGR] &
soluCL = Solve[binetCL, u'[ϕ]][[2]];
% // TraditionalForm // Style[#, FontColor → CorCl] &
```

Out[∘]= $$\left\{u'(\phi) \to \frac{\sqrt{\mathrm{E}^2 + 2\,\mathrm{Gr}\, J\phi^2\, m\, u(\phi)^3 + 2\,\mathrm{Gr}\, m\, u(\phi) - J\phi^2\, u(\phi)^2 - 1}}{J\phi}\right\}$$

Out[∘]= $$\left\{u'(\phi) \to \frac{\sqrt{2\,\mathrm{E}\,\tau - 2\,\mathrm{Gr}\, m\,\tau\, u(\phi) - 4\,\mathrm{Gr}\, m - J\phi^2\,\tau\, u(\phi)^2}}{J\phi\sqrt{\tau}}\right\}$$

O periélio e o afélio estão relacionados com a energia e o momento angular. Nestes pontos u'[ϕ] = 0

In[∘]:=
```
EnergiaMomento = Solve[{rP == 1/u[ϕ] /. Solve[(u'[ϕ] /. soluCL) == 0, u[ϕ]][[1]],
    rA == 1/u[ϕ] /. Solve[(u'[ϕ] /. soluCL) == 0, u[ϕ]][[2]]}, {E, Jϕ}]
```

Out[∘]=
$$\left\{\left\{E \to \frac{Gr\ m\ (2\ rA + 2\ rP + \tau)}{(rA + rP)\ \tau},\ J\phi \to -\frac{\sqrt{2}\ \sqrt{Gr}\ \sqrt{m}\ \sqrt{rA}\ \sqrt{rP}}{\sqrt{-rA - rP}}\right\},\ \left\{E \to \frac{Gr\ m\ (2\ rA + 2\ rP + \tau)}{(rA + rP)\ \tau},\ J\phi \to \frac{\sqrt{2}\ \sqrt{Gr}\ \sqrt{m}\ \sqrt{rA}\ \sqrt{rP}}{\sqrt{-rA - rP}}\right\}\right\}$$

Derivando a equação de primeira ordem acima temos equações de segunda ordem que são as equações de BINET

In[∘]:=
```
BINET2GR = Times[#, Jϕ/u'[ϕ]] & /@ (D[#, ϕ] & /@ binetGR) // FullSimplify;
BINET2CL = Times[#, Jϕ/u'[ϕ]] & /@ (D[#, ϕ] & /@ binetCL) // FullSimplify;
BINET2GR // TraditionalForm // Style[#, FontColor → CorGR] &
BINET2CL // TraditionalForm // Style[#, FontColor → CorCl] &
```

Out[∘]=
$$3\ Gr\ J\phi\ m\ u(\phi)^2 + \frac{Gr\ m}{J\phi} = J\phi\ (u''(\phi) + u(\phi))$$

Out[∘]=
$$\frac{Gr\ m}{J\phi} + J\phi\ (u''(\phi) + u(\phi)) = 0$$

In[∘]:=
```
(*Resolvendo Equação de BINET - Problema de Kepler*)
DSolve[{BINET2CL, u[0] == 1/rPP, u'[0] == 0}, u, ϕ];
uu = u /. First@% ;
uuu = uu[ϕ] /. EnergiaMomento[[2]] /. {rAA → rA, rPP → rP} /. rA → α (1 + ϵ) /.
      rP → α (1 - ϵ) /. α → 𝕃/(1 - ϵ^2) //
    FullSimplify[#, Assumptions → {α > 0, ϵ > 0}] & // Expand;
uCL[x_] := (uuu /. ϕ → x)
uCL[ϕ]
```

Out[∘]=
$$\frac{1}{\mathbb{L}} + \frac{\epsilon\ \mathrm{Cos}[\phi]}{\mathbb{L}}$$

In[•]:=
```
(Coefficient[({1, -1}.List @@ BINET2GR), #] & /@ {u[ϕ], u''[ϕ], u[ϕ]^2}).
  {𝕃/(Gr m) u1[ϕ], 𝕃/(Gr m) u1''[ϕ], uCL[ϕ]^2}
sol = (u1[ϕ] /. DSolve[% == 0, u1, ϕ]) /. {C[1] → 0, C[2] → 0} // Simplify;
uu = (sol // First) /. ϕ → a;
uuS[exp_] := uu /. a → exp;
delta = ((D[uCL[ϕ] + Δ 𝕃/(Gr m) uuS[ϕ], ϕ]) /. ϕ → 2 π + δϕ) // FullSimplify;
Solve[(Series[delta, {δϕ, 0, 1}] // Normal) == 0, δϕ];
(δϕ /. First@%);
Precessao = (% // Series[%, {Δ, 0, 1}] & // Normal)
```

Out[•]= $3\,\mathrm{Gr}\,J\phi\,m\left(\frac{1}{\mathbb{L}}+\frac{\epsilon\,\mathrm{Cos}[\phi]}{\mathbb{L}}\right)^2-\frac{J\phi\,\mathbb{L}\,\mathrm{u1}[\phi]}{\mathrm{Gr}\,m}-\frac{J\phi\,\mathbb{L}\,\mathrm{u1}''[\phi]}{\mathrm{Gr}\,m}$

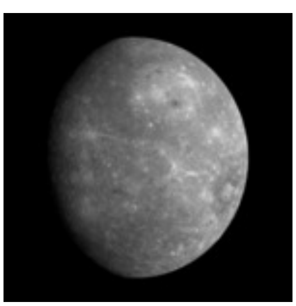
Out[•]= $\frac{6\,\mathrm{Gr}\,m\,\pi\,\Delta}{\mathbb{L}}$

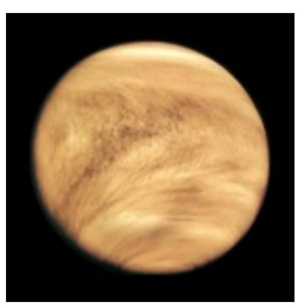
Finalmente, alguns números :

In[•]:=
```
Msun = StarData["Sun", "Mass"];
c = Quantity["SpeedOfLight"] // UnitConvert[#] &;
G = Quantity["GravitationalConstant"] // UnitConvert[#] &;
Ec = PlanetData[PlanetData[], "Eccentricity"];
MajorAcis =
  PlanetData[PlanetData[], "SemimajorAxis"] // UnitConvert[#, "Meters"] &;
Tp = PlanetData[PlanetData[], "OrbitPeriod"] // UnitConvert[#, "Centuries"] &;
Latus = Times[(1 - #^2) & /@ Ec, MajorAcis];
Pres[L_] := (Precessao /. 𝕃 → L) /. m → Msun/c^2 /. Gr → G /. Δ → 1
```

In[•]:=
```
{PlanetData[PlanetData[], "Image"], PlanetData[PlanetData[], "Name"],
  Quantity[(Pres /@ Latus)/Tp, "Radians"] // UnitConvert[#, "ArcSeconds"] &} // TableForm
```

Out[•]//TableForm=


Mercury
42.98" per century


Venus
8.624" per century


Earth
3.839" per century


Mars
1.351" per cent

Cosmologia

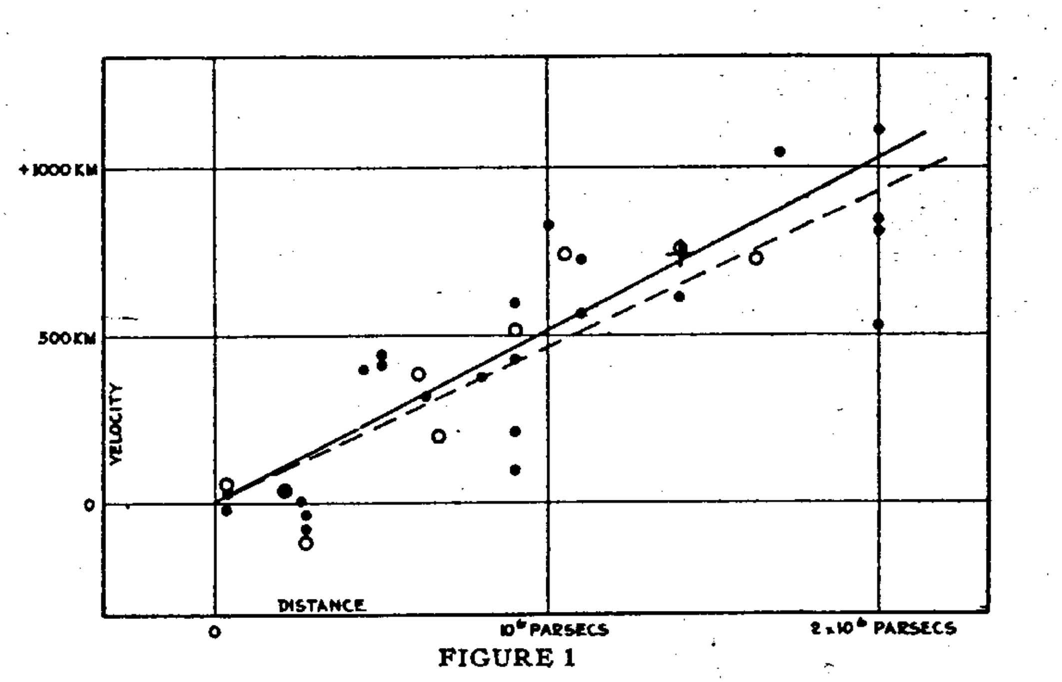
In[•]:=
```
Quit[]
```

Diagrama de Hubble

In[•]:=
```
SetDirectory[NotebookDirectory[]]
Import["HUBBLE_original.gif"]
```

Out[•]= /home/lbarosi/Dropbox/ENSINO/2019-1/Cosmologia

Out[•]=



In[•]:=
```
SN = Import["RiessDATASET.dat"] // Drop[#, {1}] & // Drop[#, {-1}] &;
```

In[•]:=
```
Needs["ErrorBarPlots`"]
```

In[•]:=
```
dadosMassageados = Partition[#, 2] &@Riffle[SN[[All, 3]], SN[[All, 4]]] //
    Riffle[#, ErrorBar /@ SN[[All, 5]]] & // Partition[#, 2] &;
```

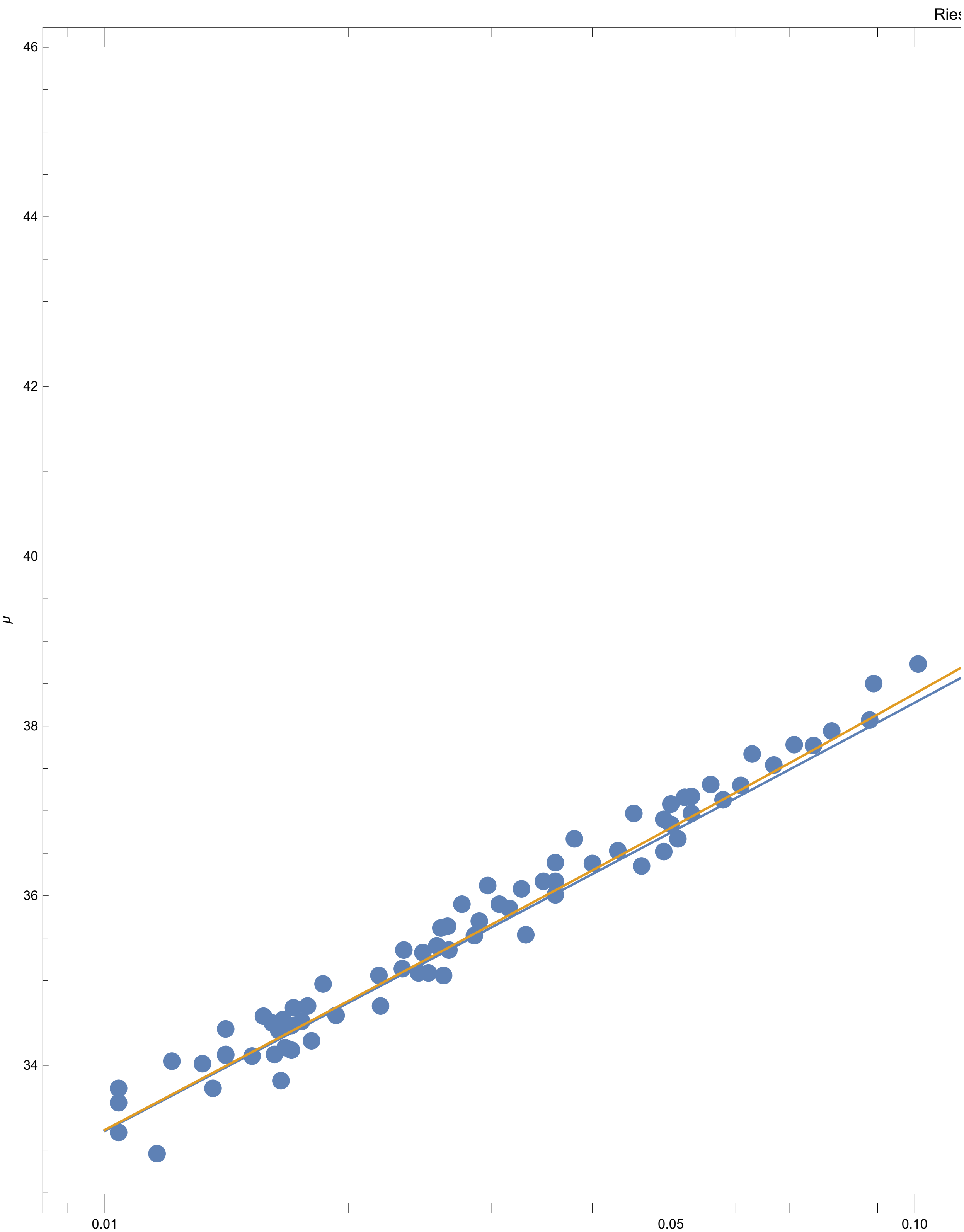
In[∘]:=

```
dadosPlot = ErrorListPlot[dadosMassageados, Axes → False,
   Frame → True, FrameLabel → {"z", "μ"}, PlotLabel → "Riess2004 Dataset"];
dadosPlotLog = ErrorListPlot[dadosMassageados, Axes → False,
   Frame → True, FrameLabel → {"Log(z)", "μ"},
   PlotLabel → "Riess2004 Dataset", ScalingFunctions → {"Log10"}];
Parameters = {c → 300 000, H0 → 68};
dL[z_?NumericQ, Ωm_] :=
 c/H0 (1 + z) NIntegrate[1/Sqrt[Ωm (1 + zi)^3 + 1 - Ωm], {zi, 0, z}] /. Parameters
μ[z_?NumericQ, Ωm_] := 25 + 5 Log10[dL[z, Ωm]]
fitPlot = Plot[{μ[z, 1], μ[z, 0.3]}, {z, 0.01, 2}, Axes → False,
   Frame → True, FrameLabel → {"z", "μ"}, PlotLabel → "Riess2004 Dataset"];
fitPlotLog = Plot[{μ[z, 1], μ[z, 0.3]}, {z, 0.01, 2},
   Axes → False, Frame → True, FrameLabel → {"z", "μ"},
   PlotLabel → "Riess2004 Dataset", ScalingFunctions → {"Log10"}];
```

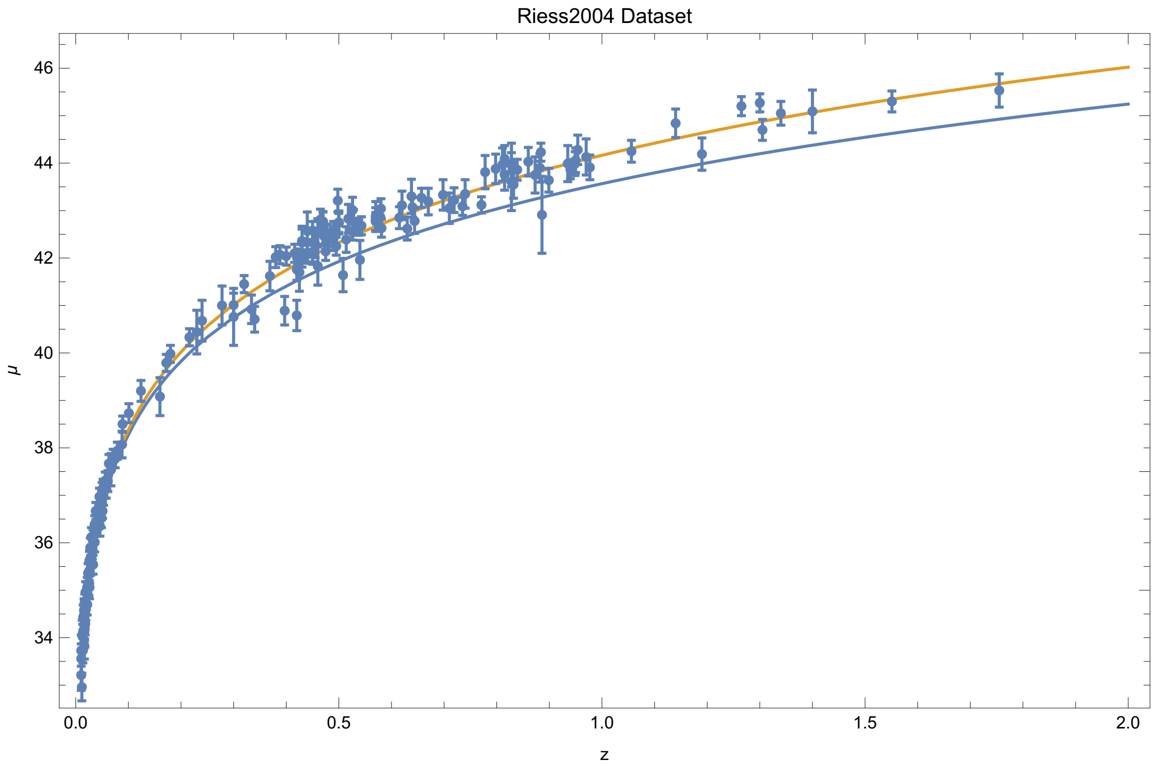
In[∘]:=

```
linearP = Show[fitPlot, dadosPlot];
logPlot = Show[{dadosPlotLog, fitPlotLog}]
GraphicsRow[{linearP, logPlot}]
```

Out[•]=



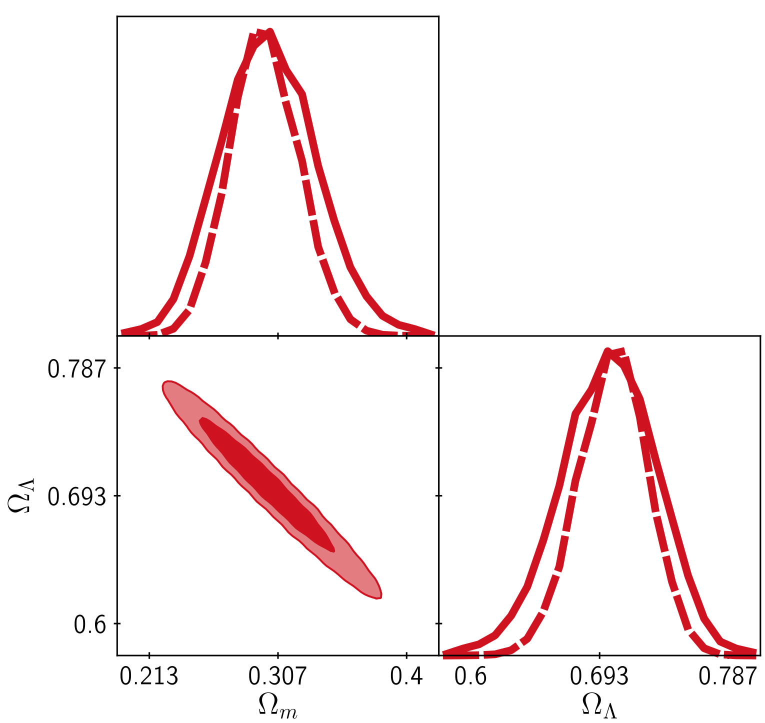
Out[ ]=



In[ ]:=

```
Import["jla_triangle.pdf"] // Show
```

Out[ ]=



In[•]:=

Universo Homogêneo

Métrica de Friedmann - Robertson - Walker

Definições

In[•]:=
```
(*Carregando Pacotes*)
Quiet@Block[{Print},
  << xAct`xTensor`;
  << xAct`xCoba`;
  << xAct`xTras`
 ]
(*Definindo opções úteis*)
$Pre = ScreenDollarIndices;
SetOptions[ContractMetric, AllowUpperDerivatives → True];
$DefInfoQ = False;
(*Função para recuperar valores dos tensores*)
MyArrayComponents[expr_] :=
 expr // ToBasis[coord2] // ComponentArray // ToValues // ToValues // Simplify
```

In[•]:=
```
(*Definindo Variedade*)
DefManifold[M, 4, {α, β, γ, ζ, ι, κ, λ, μ, ν, ξ, ο, σ, υ, χ, ψ}]
(*Definindo Métrica e derivada covariante CD*)
DefMetric[-1, g[-μ, -ν], CD, {";", "∇"}, PrintAs → "g"]
(*Definindo carta tipo cartesiana*)
DefChart[coord1, M, {0, 1, 2, 3}, {η[], x[], y[], z[]}, ChartColor → Blue]
(*Definindo carta tipo esférica*)
DefChart[coord2, M, {0, 1, 2, 3}, {t[], r[], θ[], ϕ[]}, ChartColor → Red]
```

In[•]:=
```
DefConstantSymbol[K](*Curvatura do Universo FRW*)
DefConstantSymbol[Mass, PrintAs → "M"]
DefConstantSymbol[Raio, PrintAs → "R"]
DefConstantSymbol[Λ] (*Constante Cosmológica*)
DefScalarFunction[a](*Fator de escala FRW*)
DefScalarFunction[H](*Parâmetro de Hubble*)
DefScalarFunction[ρ](*Densidade de energia*)
DefScalarFunction[P] (*Pressão*)
DefTensor[T[-α, -β], M, Symmetric[{-α, -β}]] (*Tensor momento-energia*)
```

In[]:=
```
MatrixForm[MatrixFRW =
  DiagonalMatrix[{-1, a[t[]]^2 1/(1 - K r[]^2), a[t[]]^2 r[]^2, a[t[]]^2 r[]^2 Sin[θ[]]^2}]]
MatrixForm@MetricInBasis[g, -coord2, MatrixFRW];
(*Calcula Conexão e Curvatura nas coordenadas da carta*)
MetricCompute[g, coord2, All, Verbose → False]
```

Out[]//MatrixForm=

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & \frac{a[t]^2}{1-K r^2} & 0 & 0 \\ 0 & 0 & a[t]^2 r^2 & 0 \\ 0 & 0 & 0 & a[t]^2 r^2 \mathrm{Sin}[\theta]^2 \end{pmatrix}$$

Added independent rule $g_{00} \to -1$ for tensor g

Added independent rule $g_{01} \to 0$ for tensor g

Added independent rule $g_{02} \to 0$ for tensor g

Added independent rule $g_{03} \to 0$ for tensor g

Added dependent rule $g_{10} \to g_{01}$ for tensor g

Added independent rule $g_{11} \to \frac{a[t]^2}{1 - K r^2}$ for tensor g

Added independent rule $g_{12} \to 0$ for tensor g

Added independent rule $g_{13} \to 0$ for tensor g

Added dependent rule $g_{20} \to g_{02}$ for tensor g

Added dependent rule $g_{21} \to g_{12}$ for tensor g

Added independent rule $g_{22} \to a[t]^2 r^2$ for tensor g

Added independent rule $g_{23} \to 0$ for tensor g

Added dependent rule $g_{30} \to g_{03}$ for tensor g

Added dependent rule $g_{31} \to g_{13}$ for tensor g

Added dependent rule $g_{32} \to g_{23}$ for tensor g

Added independent rule $g_{33} \to a[t]^2 r^2 \mathrm{Sin}[\theta]^2$ for tensor g

In[]:=
```
Ds2FRW = (ComponentArray[g[-{μ, coord2}, -{ν, coord2}]] /. TensorValues[g])
   BasisArray[coord2, coord2][-μ, -ν] // Total[Flatten[#]] &
Ds2FRWK0 = (ComponentArray[g[-{μ, coord2}, -{ν, coord2}]] /. TensorValues[g])
   BasisArray[coord2, coord2][-μ, -ν] // Total[Flatten[#]] /. K → 0 &
```

Out[]= $-e_\mu{}^0\ e_\nu{}^0 + a[t]^2\ e_\mu{}^2\ e_\nu{}^2\ r^2 + \frac{a[t]^2\ e_\mu{}^1\ e_\nu{}^1}{1 - K r^2} + a[t]^2\ e_\mu{}^3\ e_\nu{}^3\ r^2 \mathrm{Sin}[\theta]^2$

Out[]= $-e_\mu{}^0\ e_\nu{}^0 + a[t]^2\ e_\mu{}^1\ e_\nu{}^1 + a[t]^2\ e_\mu{}^2\ e_\nu{}^2\ r^2 + a[t]^2\ e_\mu{}^3\ e_\nu{}^3\ r^2 \mathrm{Sin}[\theta]^2$

In[•]:=

```
MatrixForm[MatrixCONF = DiagonalMatrix[{-a[η[]]^2, a[η[]]^2, a[η[]]^2, a[η[]]^2}]]
MatrixForm@MetricInBasis[g, -coord1, MatrixCONF];
Ds2FRWK0 = (ComponentArray[g[-{μ, coord1}, -{ν, coord1}]] /. TensorValues[g])
   BasisArray[coord1, coord1][-μ, -ν] // Total[Flatten[#]] &
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -a[\eta]^2 & 0 & 0 & 0 \\ 0 & a[\eta]^2 & 0 & 0 \\ 0 & 0 & a[\eta]^2 & 0 \\ 0 & 0 & 0 & a[\eta]^2 \end{pmatrix}$$

Added independent rule $g_{00} \to -a[\eta]^2$ for tensor g

Added independent rule $g_{01} \to 0$ for tensor g

Added independent rule $g_{02} \to 0$ for tensor g

Added independent rule $g_{03} \to 0$ for tensor g

Added dependent rule $g_{10} \to g_{01}$ for tensor g

Added independent rule $g_{11} \to a[\eta]^2$ for tensor g

Added independent rule $g_{12} \to 0$ for tensor g

Added independent rule $g_{13} \to 0$ for tensor g

Added dependent rule $g_{20} \to g_{02}$ for tensor g

Added dependent rule $g_{21} \to g_{12}$ for tensor g

Added independent rule $g_{22} \to a[\eta]^2$ for tensor g

Added independent rule $g_{23} \to 0$ for tensor g

Added dependent rule $g_{30} \to g_{03}$ for tensor g

Added dependent rule $g_{31} \to g_{13}$ for tensor g

Added dependent rule $g_{32} \to g_{23}$ for tensor g

Added independent rule $g_{33} \to a[\eta]^2$ for tensor g

Out[•]= $-a[\eta]^2\, e_\mu{}^0\ e_\nu{}^0 + a[\eta]^2\, e_\mu{}^1\ e_\nu{}^1 + a[\eta]^2\, e_\mu{}^2\ e_\nu{}^2 + a[\eta]^2\, e_\mu{}^3\ e_\nu{}^3$

Equação da Geodésica

In[∘]:=
```
Γ[μ_, ν_, α_] :=
 (ChristoffelCDPDcoord2[{μ, coord2}, {ν, -coord2}, {α, -coord2}] // ComponentArray //
    ToValues // ToValues // Simplify)
Γ[0, -α, -β] /. K → 0 // MatrixForm
Γ[α, 0, -β] /. K → 0 // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a[t]\, a'[t] & 0 & 0 \\ 0 & 0 & a[t]\, r^2\, a'[t] & 0 \\ 0 & 0 & 0 & a[t]\, r^2\, \mathrm{Sin}[\theta]^2\, a'[t] \end{pmatrix}$$

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{a'[t]}{a[t]} & 0 & 0 \\ 0 & 0 & \frac{a'[t]}{a[t]} & 0 \\ 0 & 0 & 0 & \frac{a'[t]}{a[t]} \end{pmatrix}$$

In[∘]:=
```
(*Componentes do Energia e Momento físicos*)
DefScalarFunction[E0]
DefScalarFunction[p]
DefTensor[Pm[μ], M]
```

In[∘]:=
```
(*Componentes dependem apenas de τ devido a homogeneidade*)
PmComponents = {E0[t[]], p[t[]]/a[t[]], 0, 0}
ComponentValue[Pm[{μ, coord2}] // ComponentArray, PmComponents]
(*Tem que informar para usar a métrica para baixar e levantar índices*)
ChangeComponents[Pm[-{μ, coord2}], Pm[{μ, coord2}]];
```

Out[∘]= $\left\{E0[t], \frac{p[t]}{a[t]}, 0, 0\right\}$

Added independent rule $Pm^{0} \to E0[t]$ for tensor Pm

Added independent rule $Pm^{1} \to \frac{p[t]}{a[t]}$ for tensor Pm

Added independent rule $Pm^{2} \to 0$ for tensor Pm

Added independent rule $Pm^{3} \to 0$ for tensor Pm

Out[∘]= $\left\{Pm^{0} \to E0[t],\ Pm^{1} \to \frac{p[t]}{a[t]},\ Pm^{2} \to 0,\ Pm^{3} \to 0\right\}$

Added independent rule $Pm_0 \to g_{00}\ Pm^0 + g_{01}\ Pm^1 + g_{02}\ Pm^2 + g_{03}\ Pm^3$ for tensor Pm

Added independent rule $Pm_1 \to g_{01}\ Pm^0 + g_{11}\ Pm^1 + g_{12}\ Pm^2 + g_{13}\ Pm^3$ for tensor Pm

Added independent rule $Pm_2 \to g_{02}\ Pm^0 + g_{12}\ Pm^1 + g_{22}\ Pm^2 + g_{23}\ Pm^3$ for tensor Pm

Added independent rule $Pm_3 \to g_{03}\ Pm^0 + g_{13}\ Pm^1 + g_{23}\ Pm^2 + g_{33}\ Pm^3$ for tensor Pm

Computed $Pm_\mu \to g_{\alpha\mu}\ Pm^\alpha$ in 0.171008 Seconds

In[∘]:=
```
(*Equação da Geodésica*)
GeodesicEq[ν_] := Pm[μ] CD[-μ]@Pm[ν]
GeodesicEq[ν]
```

Out[∘]= $Pm^\mu\ \nabla_\mu Pm^\nu$

In[∘]:=
```
(*Geodésica em componentes, K→ 0*)
geoC =
 0 == (((GeodesicEq[ν] // ToBasis[coord2] // ToBasis[coord2] // ComponentArray //
        TraceBasisDummy // ToValues) /. K → 0 // Simplify) // First)
(*Da relação de dispersão*)
(*Momento ao quadrado*)
p2 = Pm[μ] Pm[-μ];
(*EM componentes*)
p2C = m² == - (p2 // ToBasis[coord2] // TraceBasisDummy // ToValues // ToValues) /. K → 0
solE = D[p2C, t[]] // Solve[#, E0[t[]]] & // Flatten
(*Componente 0 da geodésica*)
geo0 = (geoC /. solE) /. {p[t[]] → p[a[t[]]], p'[t[]] → p'[a[t[]]] a'[t[]]} //
  FullSimplify[#, Assumptions → {a[t[]] > 0, a'[t[]] > 0, p[a[t[]]] ≠ 0}] &
p == p[a[t[]]] /. DSolve[%, p[a[t[]]], a[t[]]]
(*Momento escala com inverso do fator de escala*)
```

Out[∘]= $0 == \frac{p[t]^2\, a'[t]}{a[t]} + E0[t]\ E0'[t]$

Out[∘]= $m^2 == E0[t]^2 - p[t]^2$

Out[∘]= $\left\{E0[t] \to \frac{p[t]\ p'[t]}{E0'[t]}\right\}$

Out[∘]= $p[a[t]] + a[t]\ p'[a[t]] == 0$

... Attributes: Symbol DSolveDispatchODE not found.

Out[∘]= $\left\{p == \frac{C[1]}{a[t]}\right\}$

Equações de Movimento

Tensor de Einstein

In[∘]:=

```
(*Informa que métrica deve ser utilizada para subir índices*)
ChangeComponents[EinsteinCD[-{μ, coord2}, {ν, coord2}],
  EinsteinCD[-{μ, coord2}, -{ν, coord2}]];
ChangeComponents[EinsteinCD[{μ, coord2}, {ν, coord2}],
  EinsteinCD[-{μ, coord2}, -{ν, coord2}]];
(*Equação de Einstein*)
EINSTEINT[μ_, ν_] :=
 EinsteinCD[{μ, -coord2}, {ν, -coord2}] + Λ g[{μ, -coord2}, {ν, -coord2}]
```

Added dependent rule $G[\nabla]_0{}^0 \rightarrow G[\nabla]^0{}_0$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^0{}_0 \rightarrow G[\nabla]_{00}\ g^{00} + G[\nabla]_{01}\ g^{01} + G[\nabla]_{02}\ g^{02} + G[\nabla]_{03}\ g^{03}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]_0{}^1 \rightarrow G[\nabla]_{00}\ g^{01} + G[\nabla]_{01}\ g^{11} + G[\nabla]_{02}\ g^{12} + G[\nabla]_{03}\ g^{13}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]_0{}^2 \rightarrow G[\nabla]_{00}\ g^{02} + G[\nabla]_{01}\ g^{12} + G[\nabla]_{02}\ g^{22} + G[\nabla]_{03}\ g^{23}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]_0{}^3 \rightarrow G[\nabla]_{00}\ g^{03} + G[\nabla]_{01}\ g^{13} + G[\nabla]_{02}\ g^{23} + G[\nabla]_{03}\ g^{33}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_1{}^0 \rightarrow G[\nabla]^0{}_1$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^0{}_1 \rightarrow G[\nabla]_{01}\ g^{00} + G[\nabla]_{11}\ g^{01} + G[\nabla]_{12}\ g^{02} + G[\nabla]_{13}\ g^{03}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_1{}^1 \rightarrow G[\nabla]^1{}_1$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^1{}_1 \rightarrow G[\nabla]_{01}\ g^{01} + G[\nabla]_{11}\ g^{11} + G[\nabla]_{12}\ g^{12} + G[\nabla]_{13}\ g^{13}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]_1{}^2 \rightarrow G[\nabla]_{01}\ g^{02} + G[\nabla]_{11}\ g^{12} + G[\nabla]_{12}\ g^{22} + G[\nabla]_{13}\ g^{23}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]_1{}^3 \rightarrow G[\nabla]_{01}\ g^{03} + G[\nabla]_{11}\ g^{13} + G[\nabla]_{12}\ g^{23} + G[\nabla]_{13}\ g^{33}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_2{}^0 \rightarrow G[\nabla]^0{}_2$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^0{}_2 \rightarrow G[\nabla]_{02}\ g^{00} + G[\nabla]_{12}\ g^{01} + G[\nabla]_{22}\ g^{02} + G[\nabla]_{23}\ g^{03}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_2{}^1 \rightarrow G[\nabla]^1{}_2$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^1{}_2 \rightarrow G[\nabla]_{02}\ g^{01} + G[\nabla]_{12}\ g^{11} + G[\nabla]_{22}\ g^{12} + G[\nabla]_{23}\ g^{13}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_2{}^2 \to G[\nabla]^2{}_2$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^2{}_2 \to G[\nabla]_{02}\ g^{02} + G[\nabla]_{12}\ g^{12} + G[\nabla]_{22}\ g^{22} + G[\nabla]_{23}\ g^{23}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]_2{}^3 \to G[\nabla]_{02}\ g^{03} + G[\nabla]_{12}\ g^{13} + G[\nabla]_{22}\ g^{23} + G[\nabla]_{23}\ g^{33}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_3{}^0 \to G[\nabla]^0{}_3$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^0{}_3 \to G[\nabla]_{03}\ g^{00} + G[\nabla]_{13}\ g^{01} + G[\nabla]_{23}\ g^{02} + G[\nabla]_{33}\ g^{03}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_3{}^1 \to G[\nabla]^1{}_3$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^1{}_3 \to G[\nabla]_{03}\ g^{01} + G[\nabla]_{13}\ g^{11} + G[\nabla]_{23}\ g^{12} + G[\nabla]_{33}\ g^{13}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_3{}^2 \to G[\nabla]^2{}_3$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^2{}_3 \to G[\nabla]_{03}\ g^{02} + G[\nabla]_{13}\ g^{12} + G[\nabla]_{23}\ g^{22} + G[\nabla]_{33}\ g^{23}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]_3{}^3 \to G[\nabla]^3{}_3$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^3{}_3 \to G[\nabla]_{03}\ g^{03} + G[\nabla]_{13}\ g^{13} + G[\nabla]_{23}\ g^{23} + G[\nabla]_{33}\ g^{33}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^1{}_0 \to G[\nabla]_0{}^1$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^2{}_0 \to G[\nabla]_0{}^2$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^2{}_1 \to G[\nabla]_1{}^2$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^3{}_0 \to G[\nabla]_0{}^3$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^3{}_1 \to G[\nabla]_1{}^3$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^3{}_2 \to G[\nabla]_2{}^3$ for tensor EinsteinCD

Computed $G[\nabla]_\mu{}^\nu \to G[\nabla]_{\mu\alpha}\ g^{\nu\alpha}$ in 1.656846 Seconds

Found again independent rule
$G[\nabla]^0{}_0 \to G[\nabla]_{00}\ g^{00} + G[\nabla]_{01}\ g^{01} + G[\nabla]_{02}\ g^{02} + G[\nabla]_{03}\ g^{03}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^0{}_1 \to G[\nabla]_{01}\ g^{00} + G[\nabla]_{11}\ g^{01} + G[\nabla]_{12}\ g^{02} + G[\nabla]_{13}\ g^{03}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^0{}_2 \to G[\nabla]_{02}\ g^{00} + G[\nabla]_{12}\ g^{01} + G[\nabla]_{22}\ g^{02} + G[\nabla]_{23}\ g^{03}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^0{}_3 \to G[\nabla]_{03}\ g^{00} + G[\nabla]_{13}\ g^{01} + G[\nabla]_{23}\ g^{02} + G[\nabla]_{33}\ g^{03}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]_0{}^1 \to G[\nabla]_{00}\ g^{01} + G[\nabla]_{01}\ g^{11} + G[\nabla]_{02}\ g^{12} + G[\nabla]_{03}\ g^{13}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^1{}_1 \to G[\nabla]_{01}\ g^{01} + G[\nabla]_{11}\ g^{11} + G[\nabla]_{12}\ g^{12} + G[\nabla]_{13}\ g^{13}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^{1}{}_{2} \to G[\nabla]_{02}\ g^{01} + G[\nabla]_{12}\ g^{11} + G[\nabla]_{22}\ g^{12} + G[\nabla]_{23}\ g^{13}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^{1}{}_{3} \to G[\nabla]_{03}\ g^{01} + G[\nabla]_{13}\ g^{11} + G[\nabla]_{23}\ g^{12} + G[\nabla]_{33}\ g^{13}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]_{0}{}^{2} \to G[\nabla]_{00}\ g^{02} + G[\nabla]_{01}\ g^{12} + G[\nabla]_{02}\ g^{22} + G[\nabla]_{03}\ g^{23}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]_{1}{}^{2} \to G[\nabla]_{01}\ g^{02} + G[\nabla]_{11}\ g^{12} + G[\nabla]_{12}\ g^{22} + G[\nabla]_{13}\ g^{23}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^{2}{}_{2} \to G[\nabla]_{02}\ g^{02} + G[\nabla]_{12}\ g^{12} + G[\nabla]_{22}\ g^{22} + G[\nabla]_{23}\ g^{23}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^{2}{}_{3} \to G[\nabla]_{03}\ g^{02} + G[\nabla]_{13}\ g^{12} + G[\nabla]_{23}\ g^{22} + G[\nabla]_{33}\ g^{23}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]_{0}{}^{3} \to G[\nabla]_{00}\ g^{03} + G[\nabla]_{01}\ g^{13} + G[\nabla]_{02}\ g^{23} + G[\nabla]_{03}\ g^{33}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]_{1}{}^{3} \to G[\nabla]_{01}\ g^{03} + G[\nabla]_{11}\ g^{13} + G[\nabla]_{12}\ g^{23} + G[\nabla]_{13}\ g^{33}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]_{2}{}^{3} \to G[\nabla]_{02}\ g^{03} + G[\nabla]_{12}\ g^{13} + G[\nabla]_{22}\ g^{23} + G[\nabla]_{23}\ g^{33}$ for tensor EinsteinCD

Found again independent rule
$G[\nabla]^{3}{}_{3} \to G[\nabla]_{03}\ g^{03} + G[\nabla]_{13}\ g^{13} + G[\nabla]_{23}\ g^{23} + G[\nabla]_{33}\ g^{33}$ for tensor EinsteinCD

Computed $G[\nabla]_{\alpha}{}^{\nu} \to G[\nabla]_{\alpha\beta}\ g^{\nu\beta}$ in 1.491098 Seconds

Added independent rule
$G[\nabla]^{00} \to G[\nabla]^{0}{}_{0}\ g^{00} + G[\nabla]^{0}{}_{1}\ g^{01} + G[\nabla]^{0}{}_{2}\ g^{02} + G[\nabla]^{0}{}_{3}\ g^{03}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^{01} \to G[\nabla]_{0}{}^{1}\ g^{00} + G[\nabla]^{1}{}_{1}\ g^{01} + G[\nabla]^{1}{}_{2}\ g^{02} + G[\nabla]^{1}{}_{3}\ g^{03}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^{02} \to G[\nabla]_{0}{}^{2}\ g^{00} + G[\nabla]_{1}{}^{2}\ g^{01} + G[\nabla]^{2}{}_{2}\ g^{02} + G[\nabla]^{2}{}_{3}\ g^{03}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^{03} \to G[\nabla]_{0}{}^{3}\ g^{00} + G[\nabla]_{1}{}^{3}\ g^{01} + G[\nabla]_{2}{}^{3}\ g^{02} + G[\nabla]^{3}{}_{3}\ g^{03}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^{10} \to G[\nabla]^{01}$ for tensor EinsteinCD

Replaced independent rule $G[\nabla]^{01} \to G[\nabla]_{0}{}^{1}\ g^{00} + G[\nabla]^{1}{}_{1}\ g^{01} + G[\nabla]^{1}{}_{2}\ g^{02} + G[\nabla]^{1}{}_{3}\ g^{03}$
by $G[\nabla]^{01} \to G[\nabla]^{0}{}_{0}\ g^{01} + G[\nabla]^{0}{}_{1}\ g^{11} + G[\nabla]^{0}{}_{2}\ g^{12} + G[\nabla]^{0}{}_{3}\ g^{13}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^{11} \to G[\nabla]_{0}{}^{1}\ g^{01} + G[\nabla]^{1}{}_{1}\ g^{11} + G[\nabla]^{1}{}_{2}\ g^{12} + G[\nabla]^{1}{}_{3}\ g^{13}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^{12} \to G[\nabla]_{0}{}^{2}\ g^{01} + G[\nabla]_{1}{}^{2}\ g^{11} + G[\nabla]^{2}{}_{2}\ g^{12} + G[\nabla]^{2}{}_{3}\ g^{13}$ for tensor EinsteinCD

Added independent rule
$G[\nabla]^{13} \to G[\nabla]_{0}{}^{3}\ g^{01} + G[\nabla]_{1}{}^{3}\ g^{11} + G[\nabla]_{2}{}^{3}\ g^{12} + G[\nabla]^{3}{}_{3}\ g^{13}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^{20} \rightarrow G[\nabla]^{02}$ for tensor EinsteinCD

Replaced independent rule $G[\nabla]^{02} \rightarrow G[\nabla]_0{}^2\ g^{00} + G[\nabla]_1{}^2\ g^{01} + G[\nabla]^2{}_2\ g^{02} + G[\nabla]^2{}_3\ g^{03}$ by $G[\nabla]^{02} \rightarrow G[\nabla]^0{}_0\ g^{02} + G[\nabla]^0{}_1\ g^{12} + G[\nabla]^0{}_2\ g^{22} + G[\nabla]^0{}_3\ g^{23}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^{21} \rightarrow G[\nabla]^{12}$ for tensor EinsteinCD

Replaced independent rule $G[\nabla]^{12} \rightarrow G[\nabla]_0{}^2\ g^{01} + G[\nabla]_1{}^2\ g^{11} + G[\nabla]^2{}_2\ g^{12} + G[\nabla]^2{}_3\ g^{13}$ by $G[\nabla]^{12} \rightarrow G[\nabla]_0{}^1\ g^{02} + G[\nabla]^1{}_1\ g^{12} + G[\nabla]^1{}_2\ g^{22} + G[\nabla]^1{}_3\ g^{23}$ for tensor EinsteinCD

Added independent rule $G[\nabla]^{22} \rightarrow G[\nabla]_0{}^2\ g^{02} + G[\nabla]_1{}^2\ g^{12} + G[\nabla]^2{}_2\ g^{22} + G[\nabla]^2{}_3\ g^{23}$ for tensor EinsteinCD

Added independent rule $G[\nabla]^{23} \rightarrow G[\nabla]_0{}^3\ g^{02} + G[\nabla]_1{}^3\ g^{12} + G[\nabla]_2{}^3\ g^{22} + G[\nabla]^3{}_3\ g^{23}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^{30} \rightarrow G[\nabla]^{03}$ for tensor EinsteinCD

Replaced independent rule $G[\nabla]^{03} \rightarrow G[\nabla]_0{}^3\ g^{00} + G[\nabla]_1{}^3\ g^{01} + G[\nabla]_2{}^3\ g^{02} + G[\nabla]^3{}_3\ g^{03}$ by $G[\nabla]^{03} \rightarrow G[\nabla]^0{}_0\ g^{03} + G[\nabla]^0{}_1\ g^{13} + G[\nabla]^0{}_2\ g^{23} + G[\nabla]^0{}_3\ g^{33}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^{31} \rightarrow G[\nabla]^{13}$ for tensor EinsteinCD

Replaced independent rule $G[\nabla]^{13} \rightarrow G[\nabla]_0{}^3\ g^{01} + G[\nabla]_1{}^3\ g^{11} + G[\nabla]_2{}^3\ g^{12} + G[\nabla]^3{}_3\ g^{13}$ by $G[\nabla]^{13} \rightarrow G[\nabla]_0{}^1\ g^{03} + G[\nabla]^1{}_1\ g^{13} + G[\nabla]^1{}_2\ g^{23} + G[\nabla]^1{}_3\ g^{33}$ for tensor EinsteinCD

Added dependent rule $G[\nabla]^{32} \rightarrow G[\nabla]^{23}$ for tensor EinsteinCD

Replaced independent rule $G[\nabla]^{23} \rightarrow G[\nabla]_0{}^3\ g^{02} + G[\nabla]_1{}^3\ g^{12} + G[\nabla]_2{}^3\ g^{22} + G[\nabla]^3{}_3\ g^{23}$ by $G[\nabla]^{23} \rightarrow G[\nabla]_0{}^2\ g^{03} + G[\nabla]_1{}^2\ g^{13} + G[\nabla]^2{}_2\ g^{23} + G[\nabla]^2{}_3\ g^{33}$ for tensor EinsteinCD

Added independent rule $G[\nabla]^{33} \rightarrow G[\nabla]_0{}^3\ g^{03} + G[\nabla]_1{}^3\ g^{13} + G[\nabla]_2{}^3\ g^{23} + G[\nabla]^3{}_3\ g^{33}$ for tensor EinsteinCD

Computed $G[\nabla]^{\mu\nu} \rightarrow G[\nabla]_\alpha{}^\nu\ g^{\mu\alpha}$ in 1.777031 Seconds

Tensor Momento - Energia

In[*]:=

```
DefTensor[u[μ], M]
uComponents = {1, 0, 0, 0}
ComponentValue[u[{μ, coord2}] // ComponentArray, uComponents]
(*Tem que informar para usar a métrica para baixar e levantar índices*)
ChangeComponents[u[-{μ, coord2}], u[{μ, coord2}]];
(*Define relações*)
IndexSet[T[α_, β_], ρ[t[]] u[α] u[β] + P[t[]] (g[α, β] + u[α] u[β])];
MatrixForm[MyArrayComponents[T[-α, -β]]]
```

Out[*]= {1, 0, 0, 0}

Added independent rule $u^0 \to 1$ for tensor u

Added independent rule $u^1 \to 0$ for tensor u

Added independent rule $u^2 \to 0$ for tensor u

Added independent rule $u^3 \to 0$ for tensor u

Out[•]= $\{u^0 \to 1,\ u^1 \to 0,\ u^2 \to 0,\ u^3 \to 0\}$

Added independent rule $u_0 \to g_{00}\ u^0 + g_{01}\ u^1 + g_{02}\ u^2 + g_{03}\ u^3$ for tensor u

Added independent rule $u_1 \to g_{01}\ u^0 + g_{11}\ u^1 + g_{12}\ u^2 + g_{13}\ u^3$ for tensor u

Added independent rule $u_2 \to g_{02}\ u^0 + g_{12}\ u^1 + g_{22}\ u^2 + g_{23}\ u^3$ for tensor u

Added independent rule $u_3 \to g_{03}\ u^0 + g_{13}\ u^1 + g_{23}\ u^2 + g_{33}\ u^3$ for tensor u

Computed $u_\mu \to g_{\alpha\mu}\ u^\alpha$ in 0.270915 Seconds

Out[•]//MatrixForm=

$$\begin{pmatrix} \rho[t] & 0 & 0 & 0 \\ 0 & \frac{a[t]^2 P[t]}{1-K r^2} & 0 & 0 \\ 0 & 0 & a[t]^2 P[t] r^2 & 0 \\ 0 & 0 & 0 & a[t]^2 P[t] r^2 \mathrm{Sin}[\theta]^2 \end{pmatrix}$$

In[•]:=

```
(*Equação de Einstein*)
EINSTEIN[μ_, ν_] :=
 EinsteinCD[{μ, -coord2}, {ν, -coord2}] + Λ g[{μ, -coord2}, {ν, -coord2}] ==
  8 π G T[{μ, -coord2}, {ν, -coord2}]
(*Equação de Friedam-Robertson-Walker*)
FRWEq = EINSTEIN[0, 0] // ToValues // ToValues;
(*Equação da aceleração: Raychaudhuri*)
RaycEq = EINSTEIN[2, 2] // ToValues // ToValues // FullSimplify;
FRWEq // TraditionalForm
RaycEq /. K → 0 // TraditionalForm
```

Out[•]//TraditionalForm=

$$\frac{3\left(a'(t())^2+K\right)}{a(t())^2}-\Lambda=8\,\pi\,G\,\rho(t())$$

Out[•]//TraditionalForm=

$$r()\left(2\,a(t())\,a''(t())+a'(t())^2+a(t())^2\,(-(\Lambda-8\,\pi\,G\,P(t())))\right)=0$$

In[•]:=

```
(*Definição do parâmetro de Hubble*)
a'[η_] := H[η] a[η]
a''[η_] := D[H[η] a[η], η]
FRWEq // TraditionalForm
RaycEq // TraditionalForm
```

Out[•]//TraditionalForm=

$$\frac{3\left(a(t())^2\,H(t())^2+K\right)}{a(t())^2}-\Lambda=8\,\pi\,G\,\rho(t())$$

Out[•]//TraditionalForm=

$$r()\left(-a(t())^2\,(\Lambda-8\,\pi\,G\,P(t()))+2\,a(t())\left(a(t())\,H'(t())+a(t())\,H(t())^2\right)+a(t())^2\,H(t())^2+K\right)=0$$

In[•]:=

```
Cons[ν_] := Module[{μ}, CD[-μ]@T[μ, ν]]
(*Função FixedPoint atua com parâmetro até que resultado não mude mais*)
ConsEq =
 TableForm@FixedPoint[ToValues, Cons[-ν] // ToBasis[coord2] // ToBasis[coord2] //
     ComponentArray // ToBasis[coord2] // TraceBasisDummy]
```

Out[•]//TableForm=

$-3\,H[t]\,P[t]-3\,H[t]\,\rho[t]-\rho'[t]$
0
0
0

Matéria no Universo

In[•]:=

```
a'[τ_] =.
FRWEq
RaycEq
ContEq = ConsEq[[1]][[1]] == 0
```

Out[•]= $-\Lambda+\frac{3\,(K+a'[t]^2)}{a[t]^2}==8\,G\,\pi\,\rho[t]$

Out[•]= $r\,\left(K-a[t]^2\,(\Lambda-8\,G\,\pi\,P[t])+a'[t]^2+2\,a[t]\,(H[t]\,a'[t]+a[t]\,H'[t])\right)==0$

Out[•]= $-3\,H[t]\,P[t]-3\,H[t]\,\rho[t]-\rho'[t]==0$

In[•]:=

In[•]:=

```
(*Equação de Estado*)
Psub = P[t[]] → ω ρ[t[]]
Hsub = H[t[]] → a'[t[]]/a[t[]]
Continuidade = ContEq /. {Psub, Hsub}
```

Out[•]= $P[t] \to \omega\,\rho[t]$

Out[•]= $H[t] \to \frac{a'[t]}{a[t]}$

Out[•]= $-\frac{3\,\rho[t]\,a'[t]}{a[t]} - \frac{3\,\omega\,\rho[t]\,a'[t]}{a[t]} - \rho'[t] == 0$

In[•]:=

```
Continuidade /. {ρ[t[]] → ρ[a], ρ'[t[]] → ρ'[a] a'[t[]]} /. {a[t[]] → a, a'[t[]] → 1}
rhoa = (ρ /. First@DSolve[{%, ρ[1] == ρ0}, ρ, a])
ρa[x_, w_] := rhoa[x] /. ω → w
```

Out[•]= $-\frac{3\,\rho[a]}{a} - \frac{3\,\omega\,\rho[a]}{a} - \rho'[a] == 0$

... Attributes: Symbol DSolveDispatchODE not found.

Out[•]= $\text{Function}\left[\{a\},\, a^{-3-3\,\omega}\,\rho0\right]$

In[•]:=

```
FRWEq // Expand
HUBBLEe = FRWEq /. {Λ → 0, K → 0}
```

Out[•]= $-\Lambda + \frac{3\,K}{a[t]^2} + \frac{3\,a'[t]^2}{a[t]^2} == 8\,G\,\pi\,\rho[t]$

Out[•]= $\frac{3\,a'[t]^2}{a[t]^2} == 8\,G\,\pi\,\rho[t]$

In[•]:=

```
(*Definição do parâmetro de Hubble*)
a'[η_] := H[η] a[η]
a''[η_] := D[H[η] a[η], η]
```

In[•]:=
```
(*Densidade Critica*)
Ωc =
  ρc /. First@(HUBBLEe /. {a[t[]] → 1, H[t[]] → H0, ρ[t[]] → ρc} // Solve[#, ρc] &);
wmaterias = {0, 0, 1/3, -1, -1/3, q};
rhot =
  {Ωcold, Ωb, Ωrad, ΩΛ, ΩK, Ωq}.(ρa[a[t[]], #] & /@ wmaterias) Ωc / ρ0 // Simplify;
HUBBLE = (HUBBLEe /. ρ[t[]] → rhot) // Solve[#, H[t[]]] & // Last
```

Out[•]= $\left\{H[t] \to \sqrt{\left(H0^2\,\Omega\Lambda + \frac{H0^2\,\Omega rad}{a[t]^4} + \frac{H0^2\,\Omega b}{a[t]^3} + \frac{H0^2\,\Omega cold}{a[t]^3} + \frac{H0^2\,\Omega K}{a[t]^2} + H0^2\,\Omega q\,a[t]^{-3-3q}\right)}\right\}$

In[•]:=
```
PlanckData = {H0 → 67.44, ΩΛ → 0.6911,
  Ωcold → 0.2589, Ωb → 0.0486, Ωrad → 0.00001, ΩK → 0, Ωq → 0, q → 0}
```

Out[•]=
```
{H0 → 67.44, ΩΛ → 0.6911, Ωcold → 0.2589,
 Ωb → 0.0486, Ωrad → 0.00001, ΩK → 0, Ωq → 0, q → 0}
```

In[•]:=
```
(*Convertendo MegaParsec → km e segundos → anos*)
AgeUniverse =
  (Integrate[1/(H[t[]] a[t[]]) /. HUBBLE /. PlanckData /. a[t[]] → a, {a, 0, 1}] // N)
  10^6 (UnitConvert["Parsecs", "Kilometers"] // QuantityMagnitude)
  UnitConvert["Seconds", "Years"]
```

Out[•]= 1.38949×10^{10} yr

Evolução do Universo

In[•]:=
```
$Assumptions = And[
  ρ0 > 0 , ω > -1, G > 0, t0 > 0, t > 0, q > -1, Ωq > 0, H0 > 0]
```

Out[•]= ρ0 > 0 && ω > -1 && G > 0 && t0 > 0 && t > 0 && q > -1 && Ωq > 0 && H0 > 0

In[∘]:=

```
(*Single Component Universe*)
FRWEq
HUBBLEe
a'[τ_]=.;
SINGLE = {ΩΛ → 0, Ωcold → 0, Ωb → 0, Ωrad → 0, ΩK → 0};
rhot =
  {Ωcold, Ωb, Ωrad, ΩΛ, ΩK, Ωq}.(ρa[a[t[]], #] & /@ wmaterias) Ωc / ρ0 // Simplify;
(HUBBLEe /. ρ[t[]] → rhot /. SINGLE /. Hsub /. t[] → t /. a[t[]] → t)
sol1 = (% // DSolve[{#, a[0] == 0}, a, t] &)
```

Out[∘]= $-\Lambda + \frac{3\,(K + a[t]^2\,H[t]^2)}{a[t]^2} == 8\,G\,\pi\,\rho[t]$

Out[∘]= $3\,H[t]^2 == 8\,G\,\pi\,\rho[t]$

Out[∘]= $\frac{3\,a'[t]^2}{a[t]^2} == 3\,H0^2\,\Omega q\,a[t]^{-3-3\,q}$

... Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

... Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

... Attributes: Symbol DSolveDispatchODE not found.

... Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

... General: Further output of Solve::ifun will be suppressed during this calculation.

Out[∘]= $\left\{\left\{a \to \text{Function}\left[\{t\},\ \left(\frac{3}{2}\right)^{\frac{2}{3\,(1+q)}}\,(1+q)^{\frac{2}{3\,(1+q)}}\,(-H0\,t)^{\frac{2}{3\,(1+q)}}\,\Omega q^{\frac{1}{3\,(1+q)}}\right]\right\},\right.$
$\left.\left\{a \to \text{Function}\left[\{t\},\ \left(\frac{3}{2}\right)^{\frac{2}{3\,(1+q)}}\,(H0\,(1+q)\,t)^{\frac{2}{3\,(1+q)}}\,\Omega q^{\frac{1}{3\,(1+q)}}\right]\right\}\right\}$

In[∘]:=

```
(*Normalização*)
rho0 = Solve[(a[t] /. Last@sol1 /. t → t0) == 1, Ωq] // FullSimplify;
(*Scale factor*)
atSingle = a[t] →  First@(aUniverso1 = a[t] /. Last@sol1 /. rho0 // FullSimplify)
ρtsingle = (ρa[a[t], q] /. atSingle) // Simplify
```

Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

Out[∘]= $a[t] \to \left(\frac{t}{t0}\right)^{\frac{2}{3+3\,q}}$

Out[∘]= $\frac{t0^2\,\rho 0}{t^2}$

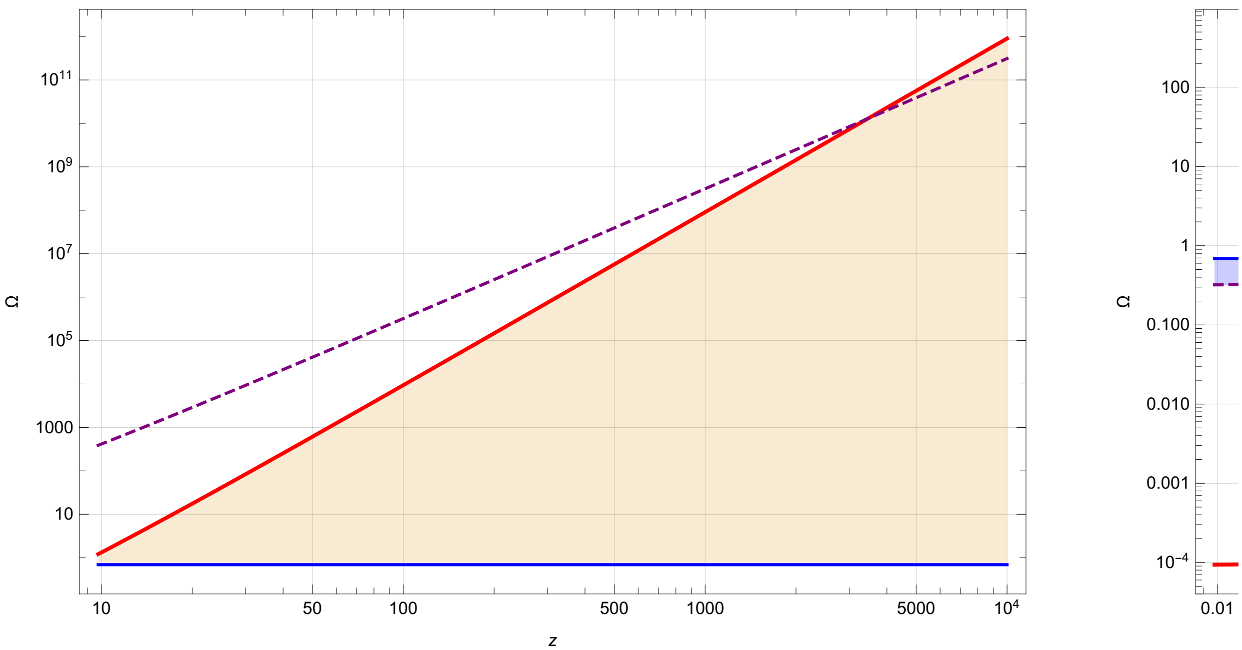
Fases do Universo

In[∘]:=

```
RadMatΛ =
  {ΩΛ → 0.6889, Ωcold → 0.311, Ωb → 0, Ωrad → 0.00009, ΩK → 0, Ωq → 0, q → 0};
rhot = ({Ωcold, Ωb, Ωrad, ΩΛ, ΩK, Ωq}.(ρa[a[t[]], #] & /@ wmaterias) Ωc / ρ0 //
      Simplify) /. RadMatΛ // Expand
DensEnergia = (rhot / Ωc) /. Plus → List /. a[t[]] → 1/(1 + z)
Radiacao = LogLogPlot[DensEnergia, {z, 0, 10 000},
   PlotStyle → {Blue, {Red, Thick}, {Purple, Dashed}}, Filling → {2 → {1}},
   PlotRange → Full, Frame → True, FrameLabel → {z, Ω}, GridLines → Automatic];
EnergiaEscura = LogLogPlot[DensEnergia, {z, 0, 10},
   PlotStyle → {Blue, {Red, Thick}, {Purple, Dashed}}, Filling → {1 → {3}},
   PlotRange → Full, Frame → True, FrameLabel → {z, Ω}, GridLines → Automatic];
GraphicsRow[{Radiacao, EnergiaEscura}]
```

Out[∘]= $\frac{0.0822314\,H0^2}{G} + \frac{0.000010743\,H0^2}{G\,a[t]^4} + \frac{0.0371229\,H0^2}{G\,a[t]^3}$

Out[∘]= $\{0.6889,\ 0.00009\,(1+z)^4,\ 0.311\,(1+z)^3\}$



Ω
z
10
50
100
500
1000
5000
10^4
10^{11}
10^9
10^7
10^5
1000
10
100
1
0.100
0.010
0.001
10^{-4}
0.01

In[•]:=

```
(*Era da Radiação*)
Hubble = H0 → 67.44;
Ωrhot = rhot/Ωc // Simplify
zradiacao = Solve[DensEnergia[[2]] == DensEnergia[[3]], z] // Last
aradiacao = 1/(z + 1) /. zradiacao
tradiacao =
 (Integrate[1/(H0 a[t[]] Sqrt[rhot/Ωc]) /. a[t[]] → a /. Hubble, {a, 0, aradiacao}] // N)
  10^6 (UnitConvert["Parsecs", "Kilometers"] // QuantityMagnitude)
  UnitConvert["Seconds", "Years"]
(*Era da Energia Escura*)
zescura = Solve[DensEnergia[[1]] == DensEnergia[[3]], z] // Last
aescura = 1/(z + 1) /. zescura
tescura =
 (Integrate[1/(H0 a[t[]] Sqrt[(Ωrhot)]) /. a[t[]] → a /. Hubble, {a, 0, aescura}] // N)
  10^6 (UnitConvert["Parsecs", "Kilometers"] // QuantityMagnitude)
  UnitConvert["Seconds", "Years"]
```

Out[•]= $0.6889 + \frac{0.00009}{a[t]^4} + \frac{0.311}{a[t]^3}$

Out[•]= {z → 3454.56}

Out[•]= 0.000289389

Out[•]= 50017. yr

Out[•]= {z → 0.303563}

Out[•]= 0.767128

Out[•]= 1.02656×10^{10} yr

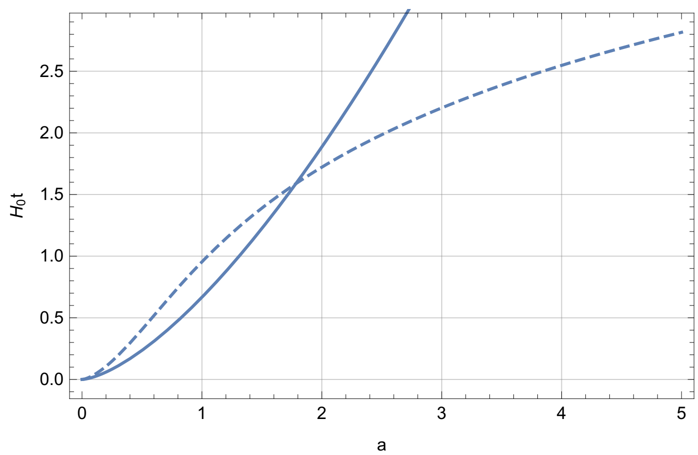
In[•]:=

```
Ta[b_, mat_] :=
 NIntegrate[1/(H[t[]] a[t[]]) /. First@(HUBBLE /. mat /. H0 → 1) /. a[t[]] → a, {a, 0, b}]
```

In[•]:=

```
RadMatΛ =
  {ΩΛ → 0.6889, Ωcold → 0.311, Ωb → 0, Ωrad → 0.00009, ΩK → 0, Ωq → 0, q → 0};
CDM = {ΩΛ → 0, Ωcold → 0.9999, Ωb → 0, Ωrad → 0.00009, ΩK → 0, Ωq → 0, q → 0};
plt1 = Plot[Ta[x, RadMatΛ], {x, 0, 5}, PlotRange → Full, Frame → True,
   FrameLabel → {"a", "H0t"}, GridLines → Automatic, PlotStyle → Dashed];
plt2 = Plot[Ta[x, CDM], {x, 0, 5}];
Show[plt1, plt2]
```

Out[•]=



Distâncias Cosmológicas

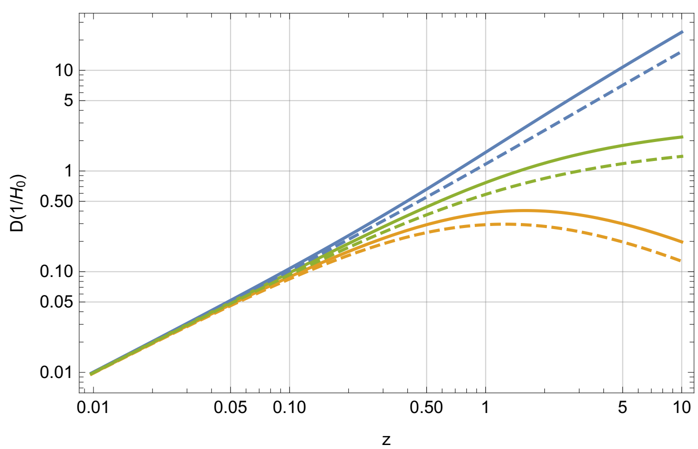
In[•]:=

```
(*Comoving*)
dp[z_, mat_] :=
 NIntegrate[1/H[t[]] /. First@(HUBBLE /. mat /. H0 → 1) /. a[t[]] → 1/(1 + za), {za, 0, z}]
dL[z_, mat_] := (1 + z) dp[z, mat]
dA[z_, mat_] := 1/(1 + z) dp[z, mat]
```

In[∘]:=

```
MatterOnly =
  {ΩΛ → 0, Ωcold → 0.99991, Ωb → 0, Ωrad → 0.00009, ΩK → 0, Ωq → 0, q → 0};
plt3 = Plot[{dL[z, RadMatΛ], dA[z, RadMatΛ], dp[z, RadMatΛ]},
   {z, 0, 10}, ScalingFunctions → {"Log10", "Log10"}, PlotRange → Full,
   Frame → True, FrameLabel → {"z", "D(1/H0)"}, GridLines → Automatic];
plt4 = Plot[{dL[z, MatterOnly], dA[z, MatterOnly], dp[z, MatterOnly]}, {z, 0, 10},
   ScalingFunctions → {"Log10", "Log10"}, PlotRange → Full, Frame → True,
   FrameLabel → {"z", "D(1/H0)"}, GridLines → Automatic, PlotStyle → Dashed];
Show[
 plt3,
 plt4]
```

Out[∘]=



In[∘]:=

```
Age[aa_?NumericQ, mat_] :=
 NIntegrate[1/(H[t[]] a[t[]]) /. HUBBLE /. mat /. H0 → 1 /. a[t[]] → a, {a, 0, aa}]
```

In[∘]:=

```
RadMatΛ
MatterOnly
ClosedUniverse = {ΩΛ → 0, Ωcold → 6, Ωb → 0, Ωrad → 0, ΩK → -5, Ωq → 0, q → 0}
EmptyUniverse = {ΩΛ → 0, Ωcold → 0, Ωb → 0, Ωrad → 0, ΩK → 1, Ωq → 0, q → 0}
ageRadMat = Age[1, RadMatΛ];
ageMatterOnly = Age[1, MatterOnly];
ageClosed = Age[1, ClosedUniverse];
ageEmpty = Age[1, EmptyUniverse];
```

Out[∘]= {ΩΛ → 0.6889, Ωcold → 0.311, Ωb → 0, Ωrad → 0.00009, ΩK → 0, Ωq → 0, q → 0}

Out[∘]= {ΩΛ → 0, Ωcold → 0.99991, Ωb → 0, Ωrad → 0.00009, ΩK → 0, Ωq → 0, q → 0}

Out[∘]= {ΩΛ → 0, Ωcold → 6, Ωb → 0, Ωrad → 0, ΩK → -5, Ωq → 0, q → 0}

Out[∘]= {ΩΛ → 0, Ωcold → 0, Ωb → 0, Ωrad → 0, ΩK → 1, Ωq → 0, q → 0}

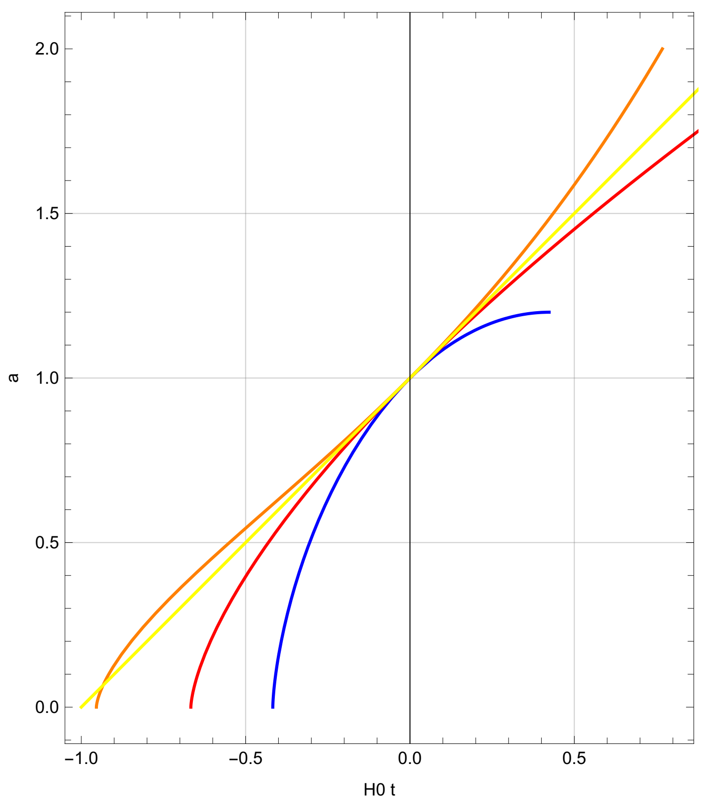
In[∘]:=

```
Clear[H0, t0]
```

In[∘]:=

```
p1 = ParametricPlot[{-ageRadMat + Age[a, RadMatΛ], a},
   {a, 0, 2}, Frame → True, FrameLabel → {"H0 t", "a"},
   GridLines → Automatic, PlotRange → Full, PlotStyle → {Orange}];
p2 = ParametricPlot[{-ageClosed + Age[a, ClosedUniverse], a},
   {a, 0, 2}, PlotStyle → {Blue}];
p3 = ParametricPlot[{-ageMatterOnly + Age[a, MatterOnly], a},
   {a, 0, 2}, PlotStyle → {Red}];
p4 = ParametricPlot[{-ageEmpty + Age[a, EmptyUniverse], a},
   {a, 0, 2}, PlotStyle → {Yellow}];
Show[
 p1,
 p2,
 p3,
 p4]
```

Out[∘]=



História Térmica

Termodinâmica

No equilíbrio termodinâmico podemos escrever uma expressão útil para a densidade de entropia, partindo da definição básica de entropia e utilizando as condições de cauchy-riemann para os potenciais termodinâmicos.

In[∘]:=
```
(*Entropia*)
dS[U, V] == (Dt[S[U, V]])
% /. U → ρ V;
% /. ρ → ρ[T];
% /. {Derivative[1, 0][S][__] → 1/T, Derivative[0, 1][S][__] → P[T]/T};
% // Collect[#, {Dt[V], Dt[T]}] &
(List @@ Last@%) /. {Dt[V] → 1, Dt[T] → 1};
MapThread[D, {%, {T, V}}];
entalpia = Equal @@ % // Simplify // Solve[#, P'[T]] & // First
entropia = S == P'[T] V /. %
```

Out[∘]= $dS[U, V] == Dt[V]\, S^{(0,1)}[U, V] + Dt[U]\, S^{(1,0)}[U, V]$

Out[∘]= $dS[V\rho[T], V] == Dt[V] \left(\frac{P[T]}{T} + \frac{\rho[T]}{T}\right) + \frac{V\, Dt[T]\, \rho'[T]}{T}$

Out[∘]= $\left\{P'[T] \to \frac{P[T] + \rho[T]}{T}\right\}$

Out[∘]= $S == \frac{V\, (P[T] + \rho[T])}{T}$

O Volume escala como $V \propto a^3$, usando a equação de continudade podemos mostrar que a entropia definida acima é uma grandeza conservada.

```
In[•]:= (*Conservação de Entropia*)
entropia /. {V → V[t], P[T] → P[t], ρ[T] → ρ[t], T → T[t], S → S[t]}
continuidade = ρ'[t] + 3 H (ρ[t] + P[t]) == 0
subscont = First@First@Solve[%, ρ'[t]];
D[%%%, t]
(% /. subscont /. V → Function[x, a[x]^3]) /. H → a'[t]/a[t] // Simplify
% /. P'[t] → P'[T] T'[t];
% /. First@entalpia /. P[T] → P[t] /. ρ[T] → ρ[t] /. T[t] → T
```

Out[•]= $S[t] == \frac{V[t]\,(P[t] + \rho[t])}{T[t]}$

Out[•]= $3\,H\,(P[t] + \rho[t]) + \rho'[t] == 0$

Out[•]= $S'[t] == -\frac{V[t]\,(P[t] + \rho[t])\,T'[t]}{T[t]^2} + \frac{(P[t] + \rho[t])\,V'[t]}{T[t]} + \frac{V[t]\,(P'[t] + \rho'[t])}{T[t]}$

Out[•]= $S'[t] == \frac{a[t]^3\,(T[t]\,P'[t] - (P[t] + \rho[t])\,T'[t])}{T[t]^2}$

Out[•]= $S'[t] == 0$

Distribuições de Fermi - Dirac e Bose - Einstein

Radiação

Nucleossíntese

Recombinação

```
nRAD[T1_, g1_] := BOSONS[[1]] /. T → T1 /. g → g1
```

In[•]:=

```
(nH/(ne np) == nBoltz[T, μH, 4, mH]/(nBoltz[T, μe, 2, me] nBoltz[T, μp, 2, mp]) /. μH → μe + μp)
(% /. mH → mp + me - BH/c^2) // Simplify
% /. mH → mp /. np → ne /. (-BH/c^2 + me + mp) → mp
SAHAnb = % /. ne → X nb /. nH → nb - X nb // Simplify
SAHA = % /. nb → η nRAD[T, 2]
```

Out[•]= $$\frac{\mathrm{nH}}{\mathrm{ne\ np}} == \frac{2\sqrt{2}\, e^{\frac{c^2\,\mathrm{me}}{k\,T}-\frac{c^2\,\mathrm{mH}}{k\,T}+\frac{c^2\,\mathrm{mp}}{k\,T}+\frac{\mu e}{T}+\frac{\mu p}{T}-\frac{\mu e+\mu p}{T}}\, h^3\, \pi^{3/2}\,(k\,\mathrm{mH}\,T)^{3/2}}{(k\,\mathrm{me}\,T)^{3/2}\,(k\,\mathrm{mp}\,T)^{3/2}}$$

Out[•]= $$\frac{\mathrm{nH}}{\mathrm{ne\ np}} == \frac{2\sqrt{2}\, e^{\frac{\mathrm{BH}}{k\,T}}\, h^3\, \pi^{3/2}\left(k\left(-\frac{\mathrm{BH}}{c^2}+\mathrm{me}+\mathrm{mp}\right)T\right)^{3/2}}{k^3\,\mathrm{me}^{3/2}\,\mathrm{mp}^{3/2}\,T^3}$$

Out[•]= $$\frac{\mathrm{nH}}{\mathrm{ne}^2} == \frac{2\sqrt{2}\, e^{\frac{\mathrm{BH}}{k\,T}}\, h^3\, \pi^{3/2}\,(k\,\mathrm{mp}\,T)^{3/2}}{k^3\,\mathrm{me}^{3/2}\,\mathrm{mp}^{3/2}\,T^3}$$

Out[•]= $$\frac{1-X}{\mathrm{nb}\,X^2} == \frac{2\sqrt{2}\, e^{\frac{\mathrm{BH}}{k\,T}}\, h^3\, \pi^{3/2}}{(k\,\mathrm{me}\,T)^{3/2}}$$

Out[•]= $$\frac{c^3\,h^3\,\pi^2\,(1-X)}{2\,k^3\,T^3\,X^2\,\eta\,\mathrm{Zeta}[3]} == \frac{2\sqrt{2}\, e^{\frac{\mathrm{BH}}{k\,T}}\, h^3\, \pi^{3/2}}{(k\,\mathrm{me}\,T)^{3/2}}$$

In[•]:=

```
Last@Solve[SAHA, X]
Xe[T_] = X /. Last@Solve[SAHA, X];
```

Out[•]= $$\left\{X \to \left(e^{-\frac{\mathrm{BH}}{k\,T}}\left(-c^3\,\mathrm{me}\sqrt{\pi}\,\sqrt{k\,\mathrm{me}\,T}+\sqrt{\left(c^6\,k\,\mathrm{me}^3\,\pi\,T+16\,c^3\,e^{\frac{\mathrm{BH}}{k\,T}}\,k^2\,\mathrm{me}\sqrt{2\,\pi}\,T^2\sqrt{k\,\mathrm{me}\,T}\,\eta\,\mathrm{Zeta}[3]\right)}\right)\right)\Big/\left(8\sqrt{2}\,k^2\,T^2\,\eta\,\mathrm{Zeta}[3]\right)\right\}$$

In[∘]:=

```
Xe[T] /. subsValores
Xz[z] = Replace[Xe[T] /. T → Tcmb (1 + z) /. subsValores,
  q_Quantity :→ QuantityMagnitude[UnitConvert[q]], Infinity]
```

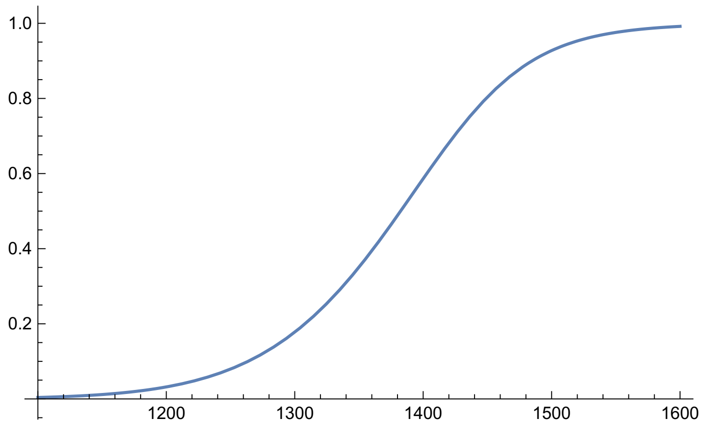
Out[∘]=
$$\frac{1}{T^2} e^{\frac{-13.6\,\text{eV}/k}{T}} \left(1.20542\times10^{8}/k^2\right)\left(\sqrt{T\left(1\,m_e\,k\right)}\left(-\sqrt{\pi}\;m_e\,c^3\right)+\sqrt{\left(e^{\frac{13.6\,\text{eV}/k}{T}}T^2\left(2.9408\times10^{-8}\,m_e\,k^2\,c^3\right)\sqrt{T\left(1\,m_e\,k\right)}+T\left(\pi\,m_e^3\,k\,c^6\right)\right)}\right)$$

Out[∘]=
$$\frac{1}{(1+z)^2}8.56002\times10^{52}\,e^{-\frac{58\,065.3}{1+z}}\left(-2.54353\times10^{-31}\sqrt{1+z}+\sqrt{\left(6.46955\times10^{-62}\,(1+z)+5.94282\times10^{-84}\,e^{\frac{58\,065.3}{1+z}}\,(1+z)^{5/2}\right)}\right)$$

In[∘]:=

```
Plot[Evaluate[Xz[z]], {z, 1100, 1600}]
```

Out[∘]=



In[∘]:=

```
T /. First@Solve[SAHA /. X → 0.5 /. subsValores, T, Reals];
k % /. subsValores // UnitConvert[#, "Electronvolts"] &;
%%/Tcmb - 1 /. subsValores;
(% + 1)^(-3/2) tUniverso /. subsValores;
recombinacao = {Trec → %%%%, Tev → %%%, zrec → %%, trec → %}
```

... Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

Out[∘]= {Trec → 3760.09 K , Tev → 0.324019 eV , zrec → 1382.4, trec → 266 255. yr }

Desacoplamento

In[∘]:=

```
{Γ, H}
% /. {Γ → ne σT c, H → H0 Sqrt[Ωm] (T / Tcmb)^(3/2)}
% /. ne → X nb
% /. nb → (Ωb ρcr)/(mp c^2) a^-3
% /. a → Tcmb/T
expr = % /. criticalDensity
```

Out[∘]= $\{\Gamma, H\}$

Out[∘]= $\left\{c\,\text{ne}\,\sigma T,\ H0\left(\frac{T}{\text{Tcmb}}\right)^{3/2}\sqrt{\Omega m}\right\}$

Out[∘]= $\left\{c\,\text{nb}\,X\,\sigma T,\ H0\left(\frac{T}{\text{Tcmb}}\right)^{3/2}\sqrt{\Omega m}\right\}$

Out[∘]= $\left\{\frac{X\,\rho\text{cr}\,\sigma T\,\Omega b}{a^3\,c\,\text{mp}},\ H0\left(\frac{T}{\text{Tcmb}}\right)^{3/2}\sqrt{\Omega m}\right\}$

Out[∘]= $\left\{\frac{T^3\,X\,\rho\text{cr}\,\sigma T\,\Omega b}{c\,\text{mp}\,\text{Tcmb}^3},\ H0\left(\frac{T}{\text{Tcmb}}\right)^{3/2}\sqrt{\Omega m}\right\}$

Out[∘]= $\left\{\frac{3\,c\,H0^2\,T^3\,X\,\sigma T\,\Omega b}{8\,G\,\text{mp}\,\pi\,\text{Tcmb}^3},\ H0\left(\frac{T}{\text{Tcmb}}\right)^{3/2}\sqrt{\Omega m}\right\}$

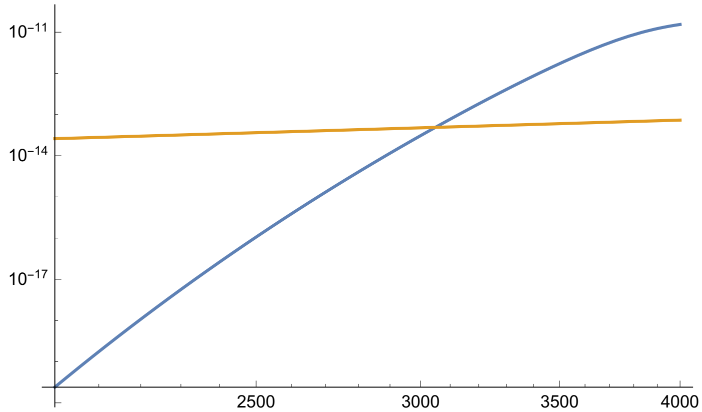
In[∘]:=

```
Γ[Tt_] := Replace[First@expr /. X → Xe[T] /. subsValores,
    q_Quantity :→ QuantityMagnitude[UnitConvert[q]], Infinity] /. T → Tt
H[Tt_] := Replace[Last@expr /. subsValores,
    q_Quantity :→ QuantityMagnitude[UnitConvert[q]], Infinity] /. T → Tt
```

In[•]:=

```
LogLogPlot[{Evaluate[Γ[T]], Evaluate[H[T]]}, {T, 2000, 4000}]
```

Out[•]=



In[•]:=

```
T /. FindRoot[Γ[T]/H[T] == 1, {T, 3000}] // Quantity[#, "Kelvins"] &;
k % /. subsValores // UnitConvert[#, "Electronvolts"] &;
%%/Tcmb - 1 /. subsValores;
(% + 1)^(-3/2) tUniverso /. subsValores;
desacoplamento = {Tdec → %%%%, Tdev → %%%, zdec → %%, tdec → %}
```

Out[•]= {Tdec → 3049.55 K , Tdev → 0.26279 eV , zdec → 1120.98, tdec → 364 537. yr }

Electron Freeze out

In[•]:=

```
FRWeq = H → H0 Sqrt[Ωm] (T / Tcmb)^(3/2);
crossSection = σ → Σ (BH/(k T))^(1/2) Log[BH/(k T)];
crossSectionV = Σ → 9.78 α^2 h^2/(me^2 c);
```

In[•]:=

```
{1/a^3 Dt[ne a^3, t], ne0 np0 σ (nH nnγ/(nH0 nnγ0) - ne np/(ne0 np0))}
% /. nnγ → nnγ0;
% /. np → ne /. ne → Xee nb /. nH → nb (1 - Xee);
% /. Dt[a, t] → 0 /. Dt[nb, t] → 0 // Simplify;
% /.
  {np0 → nBoltz[T, 0, 2, mp], nH0 → nBoltz[T, 0, 4, mH], ne0 → nBoltz[T, 0, 2, me]};
% /. c^2 mH → c^2 mp + c^2 me - BH // Simplify;
% /. mH → mp // Simplify
% /. crossSection;
%/nb /. nb → η nγ /. Xee → Xee[x] /. BH/(k T) → x
% /. Dt[x, t] → H x
% /. FRWeq;
% /. z → T/Tcmb - 1;
% /. T → BH/(k x);
BoltzElectron = Equal @@ % // Simplify
```

Out[•]= $\left\{\frac{3\,a^2\,\text{ne}\,\text{Dt}[a,t]+a^3\,\text{Dt}[\text{ne},t]}{a^3},\ \text{ne0}\left(\frac{\text{nH}\,\text{nn}\gamma}{\text{nH0}\,\text{nn}\gamma\text{0}}-\frac{\text{ne}\,\text{np}}{\text{ne0}\,\text{np0}}\right)\text{np0}\,\sigma\right\}$

Out[•]= $\left\{\text{nb}\,\text{Dt}[\text{Xee},t],\ \frac{1}{4}\,\text{nb}\left(-\frac{\sqrt{2}\,e^{-\frac{\text{BH}}{k\,T}}\left(k\,\text{me}\,T\right)^{3/2}\left(-1+\text{Xee}\right)}{h^3\,\pi^{3/2}}-4\,\text{nb}\,\text{Xee}^2\right)\sigma\right\}$

Out[•]= $\left\{\text{Dt}[x,t]\,\text{Xee}'[x],\ \frac{1}{4}\sqrt{x}\,\Sigma\,\text{Log}[x]\left(-\frac{\sqrt{2}\,e^{-x}\left(k\,\text{me}\,T\right)^{3/2}\left(-1+\text{Xee}[x]\right)}{h^3\,\pi^{3/2}}-\frac{8\,k^3\,T^3\,\eta\,\text{Xee}[x]^2\,\text{Zeta}[3]}{c^3\,h^3\,\pi^2}\right)\right\}$

Out[•]= $\left\{H\,x\,\text{Xee}'[x],\ \frac{1}{4}\sqrt{x}\,\Sigma\,\text{Log}[x]\left(-\frac{\sqrt{2}\,e^{-x}\left(k\,\text{me}\,T\right)^{3/2}\left(-1+\text{Xee}[x]\right)}{h^3\,\pi^{3/2}}-\frac{8\,k^3\,T^3\,\eta\,\text{Xee}[x]^2\,\text{Zeta}[3]}{c^3\,h^3\,\pi^2}\right)\right\}$

Out[•]= $\text{H0}\left(\frac{\text{BH}}{k\,\text{Tcmb}\,x}\right)^{3/2}x\sqrt{\Omega m}\,\text{Xee}'[x]==-\frac{1}{4\,h^3\,\pi^2}\sqrt{x}\,\Sigma\,\text{Log}[x]\left(e^{-x}\sqrt{2\,\pi}\left(\frac{\text{BH}\,\text{me}}{x}\right)^{3/2}\left(-1+\text{Xee}[x]\right)+\frac{8\,\text{BH}^3\,\eta\,\text{Xee}[x]^2\,\text{Zeta}[3]}{c^3\,x^3}\right)$

In[•]:=

```
BoltzElectron /. crossSectionV /. subsValores
eqElectronFreeze = Replace[BoltzElectron /. crossSectionV /. subsValores,
   q_Quantity :→ QuantityMagnitude[UnitConvert[q]], Infinity];
```

Out[•]=

$$\sqrt{\frac{1}{x}}\left(7.79033\times 10^{6}\, H_0\right) \mathrm{Xee}'[x] == \sqrt{x}\, \mathrm{Log}[x] \left(-1.55654\, \alpha^2/(m_e^2 h\, c)\right) \left(e^{-x}\sqrt{2\pi}\left(\frac{13.6\, m_e\, \mathrm{eV}}{x}\right)^{3/2}(-1+\mathrm{Xee}[x]) + \frac{\left(0.0000147558\, \mathrm{eV}^3/c^3\right)\mathrm{Xee}[x]^2}{x^3}\right)$$

In[•]:=

```
cInicial = Xee[BH/(k T) /. T → Trec /. recombinacao /. subsValores] == 0.5;
sol = NDSolve[{eqElectronFreeze, cInicial}, Xee,
  {x, 1, 100}, Method → "StiffnessSwitching", MaxSteps → 1 000 000]
```

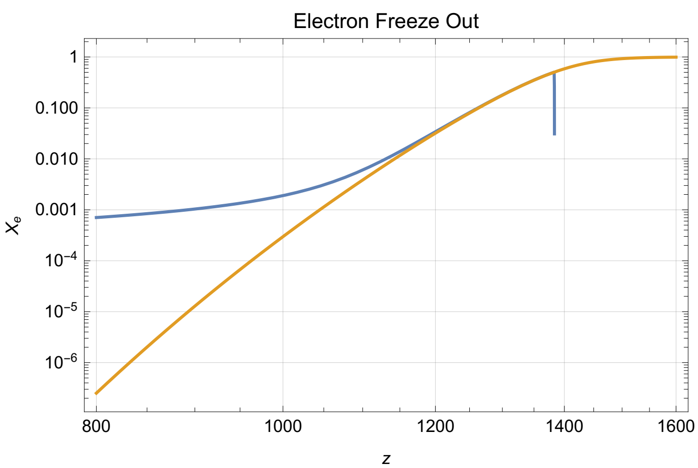
Out[•]=

```
{{Xee → InterpolatingFunction[ Domain: {{1., 100.}} Output: scalar ]}}
```

In[•]:=

```
XZ = BH/(k T) /. T → Tcmb (1 + z) /. subsValores;
Xeee[z_] := Xee[X] /. First@sol /. X → XZ
LogLogPlot[{Evaluate[Xeee[z]], Evaluate[Xz[z]]},
 {z, 800, 1600}, PlotRange → All, Frame → True, FrameLabel → {z, Xe},
 GridLines → Automatic, PlotLabel → "Electron Freeze Out"]
```

Out[•]=



Resumo da História Térmica do Universo

```
Desacoplamento dos Neutrinos
Igualdade Matéria Radiação
BBN
Recombinação
Desacoplamento
{t, T, kT, z, a}
```