An xTensor/xCoba based derivation.

See section “ADM formalism: spherical symmetry (dim=3)” for description.

(initialization - exec) Setup for xCoba/xTras:

```
(*Setup*)
Block[{Print},
  << xAct`xCoba`
 ];
$Info = False;
(*Hide away definition infos*)
$DefInfoQ = False;
$CVVerbose = False;
$PrePrint = ScreenDollarIndices;

Quiet@Block[{Print},
  << xAct`xTras`
 ]
```

(initialization - exec) xTensor Helper functions:

```
(*As we have explicit components apply simplification rule*)
$CCSimplify = Simplify@ToValues@# &;
(*To throw zero components*)
ThrowZeros[in_] := Cases[in, x_ /; ¬ (x[[2]] === 0) ∧ x[[1]] === ToCanonical@x[[1]]];
(*Do not check canonicalization for extraction*)
ThrowZeros[in_] := Cases[in, x_ /; ¬ (x[[2]] === 0)];
(*
Find expression with indices raised via
 metric. Suppose we have G_ab and a chart Ψ; then
 mutInd[EinsteinCD[a,b],EinsteinCD[-a,-b],Ψ]
 calculates the new form as required
*)
mutInd[newExpr_, oldExpr_, chart_] :=
  ChangeComponents[newExpr // ToBasis[chart], oldExpr // ToBasis[chart]];
(*
Extract non-zero tensor values
*)
extrNZ[tens_, chart_] := ThrowZeros[
   Thread[
    Flatten[(tens // ToBasis[chart] // ComponentArray)] →
     Flatten[(tens // ToBasis[chart] // ComponentArray // ToValues)]
   ]
  ];
(*
Extract non-zero tensor values coupled with taylor expansion in a parameter
*)
extrNZTaySer[tens_, chart_, serpar_] := ThrowZeros[
   Thread[
    Flatten[(tens // ToBasis[chart] // ComponentArray)] → (Series[
        Flatten[(tens // ToBasis[chart] // ComponentArray // ToValues)],
        serpar] // Normal)
   ]
  ];

(*helper functions for extraction of components*)
getCpts[obj_, chart_] := FixedPoint[ToValues,
  obj // ToBasis[chart] // ComponentArray // ToBasis[chart] // TraceBasisDummy
 ]
```

(xCoba) ADMformalism: spherical symmetry (dim=3)

What we do:

Our goal here is to derive explicit (coordinate) expressions for the standard ADM formalism.

We particularize to spherical symmetry and utilize adapted coordinates. Under this assumption a symmetric 2-

tensor $\mathcal{S}_{ij} = \mathcal{S}_{(ij)} \in \mathcal{T}_2(M)$ may be written:

$$\mathcal{S}_{ij}\, dx^i\, dx^j = s_1(r)\, dr^2 + s_2(r)\, d\theta^2 + s_2(r) \sin^2(\vartheta)\, d\varphi^2$$

Relevant equations to work with: (recall $\partial_\perp[\cdot] := (\partial_t - \mathcal{L}_\beta)[\cdot]$):

$$\partial_\perp[\gamma_{ij}] = -2\,\alpha\, K_{ij}$$
$$\partial_\perp[K_{ij}] = -D_i[D_j[\alpha]] + \alpha\left(R_{ij} + K\, K_{ij} - 2\, K_{ik}\, K_j^{\ k}\right)$$

$$\mathcal{H} = R + K^2 - K_{ij}\, K^{ij} = 0$$
$$\mathcal{M}_i = D^j[K_{ij}] - D_i[K] = 0$$

We also note that due to symmetry $\beta^i \doteq (\beta^r(r),\ 0,\ 0)$

During derivation we assume that the manifold M that we work on (and imagine fields as living in the appropriate tangent bundle of) is Σ_t at some fixed t with salient derivative operators defined accordingly.

Setup manifold and charts, populate and calculate geometric objects

```
(*Define manifold and associated metric*)
DefManifold[M, 3, IndexRange[a, q]];
DefMetric[1, metric[-a, -b], CD, PrintAs → "γ"];

(*Make charts: one for spherical coords*)
DefChart[Csph, M, {1, 2, 3}, {r[], th[], ph[]}, ChartColor → Green];

DefScalarFunction[gam1, PrintAs → "γ1"];
DefScalarFunction[gam2, PrintAs → "γ2"];

(*Define a matrix that will represent the metric in sph. coords*)
MatrixForm[
  MatMetCsph = DiagonalMatrix[
    {gam1[r[]], gam2[r[]], gam2[r[]] Sin[th[]]^2}]
 ];

(*Insert the definition*)
MatrixForm@MetricInBasis[metric, -Csph, MatMetCsph]
```

$$\begin{pmatrix} \gamma_{11} \to \gamma_1[r] & \gamma_{12} \to 0 & \gamma_{13} \to 0 \\ \gamma_{21} \to 0 & \gamma_{22} \to \gamma_2[r] & \gamma_{23} \to 0 \\ \gamma_{31} \to 0 & \gamma_{32} \to 0 & \gamma_{33} \to \gamma_2[r]\, \mathrm{Sin}[th]^2 \end{pmatrix}$$

```
(*Geometric quantities in Schw/car representation*)
DSimplify[arg_] := arg; (*dummy simplification function [otherwise slow]*)
MetricCompute[metric, Csph, All, CVSimplify → DSimplify];

(* similarily prepare other fields *)
DefTensor[Kextr[-a, -b], M, Symmetric[{1, 2}], PrintAs → "K"];
```

```
(*scalar fields to carry components of extrinsic curvature*)
DefScalarFunction[kap1, PrintAs → "κ1"];
DefScalarFunction[kap2, PrintAs → "κ2"];

(*prepare another symmetric (and diagonal) field*)
tmpKextr = DiagonalMatrix[
   {kap1[r[]], kap2[r[]], kap2[r[]] Sin[th[]]^2}
  ];

(*insert in terms of specified chart*)
ComponentValue[
  ComponentArray[Kextr[-a, -b] // ToBasis[Csph]],
  tmpKextr
 ];

(*lapse*)
DefScalarFunction[alpha, PrintAs → "α"];

(*shift*)
DefTensor[beta[-i], M, PrintAs → "β"];

DefScalarFunction[betar, PrintAs → "β^r"];

(*insert in terms of specified chart*)
ComponentValue[
  ComponentArray[beta[a] // ToBasis[Csph]],
  {betar[r[]], 0, 0}
 ];
```

Inspect some values

```
metric[-a, -b] // ToBasis[Csph] // ComponentArray // ToValues
```

$$\{\{\gamma_1[r],0,0\},\{0,\gamma_2[r],0\},\{0,0,\gamma_2[r]\,\mathrm{Sin}[th]^2\}\}$$

```
Kextr[-a, -b] // ToBasis[Csph] // ComponentArray // ToValues
```

$$\{\{\kappa_1[r],0,0\},\{0,\kappa_2[r],0\},\{0,0,\kappa_2[r]\,\mathrm{Sin}[th]^2\}\}$$

```
(*Look at the Christoffels*)
ChristoffelCDPDCsph[c, -a, -b] // ToBasis[Csph] // ComponentArray // ToValues //
 ToValues
```

$$\Big\{\Big\{\Big\{\frac{\gamma_1'[r]}{2\,\gamma_1[r]},0,0\Big\},\Big\{0,\frac{\gamma_2'[r]}{2\,\gamma_2[r]},0\Big\},\Big\{0,0,\frac{\gamma_2'[r]}{2\,\gamma_2[r]}\Big\}\Big\},$$
$$\Big\{\Big\{0,\frac{\gamma_2'[r]}{2\,\gamma_2[r]},0\Big\},\Big\{-\frac{\gamma_2'[r]}{2\,\gamma_1[r]},0,0\Big\},\{0,0,\mathrm{Cot}[th]\}\Big\},$$
$$\Big\{\Big\{0,0,\frac{\gamma_2'[r]}{2\,\gamma_2[r]}\Big\},\{0,0,\mathrm{Cot}[th]\},\Big\{-\frac{\mathrm{Sin}[th]^2\,\gamma_2'[r]}{2\,\gamma_1[r]},-\mathrm{Cos}[th]\,\mathrm{Sin}[th],0\Big\}\Big\}\Big\}$$

```
(*Ricci tensor components in chart*)
RicCsph = RicciCD[-a, -b] // ToBasis[Csph] // ComponentArray // ToValues;
```

```
RicCsph // Expand
```

$$\left\{\left\{\frac{\gamma_1'[r]\,\gamma_2'[r]}{2\,\gamma_1[r]\,\gamma_2[r]}+\frac{\gamma_2'[r]^2}{2\,\gamma_2[r]^2}-\frac{\gamma_2''[r]}{\gamma_2[r]},\,0,\,0\right\},\,\left\{0,\,1+\frac{\gamma_1'[r]\,\gamma_2'[r]}{4\,\gamma_1[r]^2}-\frac{\gamma_2''[r]}{2\,\gamma_1[r]},\,0\right\},\right.$$
$$\left.\left\{0,\,0,\,\mathrm{Sin}[th]^2+\frac{\mathrm{Sin}[th]^2\,\gamma_1'[r]\,\gamma_2'[r]}{4\,\gamma_1[r]^2}-\frac{\mathrm{Sin}[th]^2\,\gamma_2''[r]}{2\,\gamma_1[r]}\right\}\right\}$$

Prepare ADMequations: constraints

Build the Hamiltonian

```
tmp = RicciCD[-a, -b] metric[a, b]
```

$R[\nabla]$

```
termH1 = FullSimplify[
  tmp // ToBasis[Csph] // ToValues
 ]
```

$$\frac{1}{2\,\gamma_1[r]^2\,\gamma_2[r]^2}\left(4\,\gamma_1[r]^2\,\gamma_2[r]+2\,\gamma_2[r]\,\gamma_1'[r]\,\gamma_2'[r]+\gamma_1[r]\,\left(\gamma_2'[r]^2-4\,\gamma_2[r]\,\gamma_2''[r]\right)\right)$$

```
tmp = Kextr[-a, -b] metric[a, b]
```

$K_{ab}\ \gamma^{ab}$

```
termTrK = tmp // ToBasis[Csph] // TraceBasisDummy // ToValues
```

$$\frac{\kappa_1[r]}{\gamma_1[r]}+\frac{2\,\kappa_2[r]}{\gamma_2[r]}$$

```
tmp = Kextr[-a, -b] Kextr[-c, -d] metric[a, c] metric[b, d]
```

$K_{ab}\ K_{cd}\ \gamma^{ac}\ \gamma^{bd}$

```
termH3 = tmp // ToBasis[Csph] // TraceBasisDummy // ToValues
```

$$\frac{\kappa_1[r]^2}{\gamma_1[r]^2}+\frac{2\,\kappa_2[r]^2}{\gamma_2[r]^2}$$

```
fullHam = termH1 + termTrK^2 - termH3
```

$$-\frac{\kappa_1[r]^2}{\gamma_1[r]^2}-\frac{2\,\kappa_2[r]^2}{\gamma_2[r]^2}+\left(\frac{\kappa_1[r]}{\gamma_1[r]}+\frac{2\,\kappa_2[r]}{\gamma_2[r]}\right)^2+$$
$$\frac{1}{2\,\gamma_1[r]^2\,\gamma_2[r]^2}\left(4\,\gamma_1[r]^2\,\gamma_2[r]+2\,\gamma_2[r]\,\gamma_1'[r]\,\gamma_2'[r]+\gamma_1[r]\,\left(\gamma_2'[r]^2-4\,\gamma_2[r]\,\gamma_2''[r]\right)\right)$$

```
Ham = FullSimplify[fullHam]
```

$$\frac{1}{2\,\gamma_1[r]^2\,\gamma_2[r]^2}\left(4\,\gamma_1[r]^2\,\left(\gamma_2[r]+\kappa_2[r]^2\right)+\right.$$
$$\left.2\,\gamma_2[r]\,\gamma_1'[r]\,\gamma_2'[r]+\gamma_1[r]\,\left(\gamma_2'[r]^2+\gamma_2[r]\,\left(8\,\kappa_1[r]\,\kappa_2[r]-4\,\gamma_2''[r]\right)\right)\right)$$

```
(*expand and suppress arguments with rule*)
Expand[Ham] /. field_[arg_] :→ field
```

$$\frac{2}{\gamma_2}+\frac{4\,\kappa_1\,\kappa_2}{\gamma_1\,\gamma_2}+\frac{2\,\kappa_2{}^2}{\gamma_2{}^2}+\frac{\gamma_1'\,\gamma_2'}{\gamma_1{}^2\,\gamma_2}+\frac{(\gamma_2')^2}{2\,\gamma_1\,\gamma_2{}^2}-\frac{2\,\gamma_2''}{\gamma_1\,\gamma_2}$$

Build momentum constraint

```
tmp = CD[-j][Kextr[-i, -k]] metric[j, k]
```

$$\gamma^{jk}\,\left(\nabla_j K_{ik}\right)$$

```
termM1 = ToValues[
  ComponentArray[
   tmp // ToBasis[Csph] // ToBasis[Csph] // TraceBasisDummy
  ]
 ]
```

$$\left\{-\frac{\kappa_1[r]\,\gamma_1'[r]}{\gamma_1[r]^2}+\frac{\kappa_1[r]\,\gamma_2'[r]}{\gamma_1[r]\,\gamma_2[r]}-\frac{\kappa_2[r]\,\gamma_2'[r]}{\gamma_2[r]^2}+\frac{\kappa_1'[r]}{\gamma_1[r]},\,0,\,0\right\}$$

```
tmp = CD[-i][Kextr[-j, -k]] metric[j, k]
```

$$\gamma^{jk}\,\left(\nabla_i K_{jk}\right)$$

```
termM2 = ToValues[
  ComponentArray[
   tmp // ToBasis[Csph] // ToBasis[Csph] // TraceBasisDummy
  ]
 ]
```

$$\left\{-\frac{\kappa_1[r]\,\gamma_1'[r]}{\gamma_1[r]^2}-\frac{2\,\kappa_2[r]\,\gamma_2'[r]}{\gamma_2[r]^2}+\frac{\kappa_1'[r]}{\gamma_1[r]}+\frac{2\,\kappa_2'[r]}{\gamma_2[r]},\,0,\,0\right\}$$

```
fullMomentum = termM1 - termM2
```

$$\left\{\frac{\kappa_1[r]\,\gamma_2'[r]}{\gamma_1[r]\,\gamma_2[r]}+\frac{\kappa_2[r]\,\gamma_2'[r]}{\gamma_2[r]^2}-\frac{2\,\kappa_2'[r]}{\gamma_2[r]},\,0,\,0\right\}$$

```
Momentum = FullSimplify[fullMomentum]
```

$$\left\{((\gamma_2[r]\,\kappa_1[r]+\gamma_1[r]\,\kappa_2[r])\,\gamma_2'[r]-2\,\gamma_1[r]\,\gamma_2[r]\,\kappa_2'[r])\Big/\left(\gamma_1[r]\,\gamma_2[r]^2\right),\,0,\,0\right\}$$

```
(*expand and suppress arguments with rule*)
Expand[Momentum] /. field_[arg_] :→ field
```

$$\left\{\frac{\kappa_1\,\gamma_2'}{\gamma_1\,\gamma_2}+\frac{\kappa_2\,\gamma_2'}{\gamma_2{}^2}-\frac{2\,\kappa_2'}{\gamma_2},\,0,\,0\right\}$$

Prepare ADMequations: dynamics

Assemble 3-metric evolution equation

```
tmpLie = LieDToCovD[LieD[beta[i]][metric[-i, -j]], CD]
```

$$\gamma_{aj}\,\left(\nabla_i \beta^a\right) + \gamma_{ia}\,\left(\nabla_j \beta^a\right)$$

```
tEvoMetRHSLie =
 tmpLie // ToBasis[Csph] // TraceBasisDummy // ComponentArray // ToValues
```

$$\left\{\{2\,\gamma_1[r]\,\beta^{r\prime}[r] + \beta^r[r]\,\gamma_1{}'[r],\ 0,\ 0\},\ \{0,\ \beta^r[r]\,\gamma_2{}'[r],\ 0\},\ \left\{0,\ 0,\ \beta^r[r]\,\mathrm{Sin}[th]^2\,\gamma_2{}'[r]\right\}\right\}$$

```
tEvoMetRHST1 =
 -2 alpha[r[]] Kextr[-i, -j] // ToBasis[Csph] // ComponentArray // ToValues
```

$$\left\{\{-2\,\alpha[r]\,\kappa_1[r],\ 0,\ 0\},\ \{0,\ -2\,\alpha[r]\,\kappa_2[r],\ 0\},\ \left\{0,\ 0,\ -2\,\alpha[r]\,\kappa_2[r]\,\mathrm{Sin}[th]^2\right\}\right\}$$

```
tEvoMetRHS = tEvoMetRHST1 + tEvoMetRHSLie;
```

```
Simplify[tEvoMetRHS]
```

$$\left\{\{-2\,\alpha[r]\,\kappa_1[r] + 2\,\gamma_1[r]\,\beta^{r\prime}[r] + \beta^r[r]\,\gamma_1{}'[r],\ 0,\ 0\},\right.$$
$$\left.\{0,\ -2\,\alpha[r]\,\kappa_2[r] + \beta^r[r]\,\gamma_2{}'[r],\ 0\},\ \left\{0,\ 0,\ \mathrm{Sin}[th]^2\,(-2\,\alpha[r]\,\kappa_2[r] + \beta^r[r]\,\gamma_2{}'[r])\right\}\right\}$$

```
(*examine again the metric*)
tmp = metric[-i, -j] // ToBasis[Csph] // ComponentArray // ToValues
```

$$\left\{\{\gamma_1[r],\ 0,\ 0\},\ \{0,\ \gamma_2[r],\ 0\},\ \left\{0,\ 0,\ \gamma_2[r]\,\mathrm{Sin}[th]^2\right\}\right\}$$

(*γ_1 and γ_2 carry t-dep that we have suppressed, if we reintroduce*)

```
dtgam1 = tEvoMetRHS[[1]][[1]]
```

$$-2\,\alpha[r]\,\kappa_1[r] + 2\,\gamma_1[r]\,\beta^{r\prime}[r] + \beta^r[r]\,\gamma_1{}'[r]$$

```
dtgam2 = tEvoMetRHS[[2]][[2]]
```

$$-2\,\alpha[r]\,\kappa_2[r] + \beta^r[r]\,\gamma_2{}'[r]$$

```
(*and the third component is redundant..*)
```

Assemble extrinsic curvature evolution equation

```
tmpLieKextr = LieDToCovD[LieD[beta[i]][Kextr[-i, -j]], CD]
```

$$\beta^a\,\left(\nabla_a K_{ij}\right) + K_{aj}\,\left(\nabla_i \beta^a\right) + K_{ia}\,\left(\nabla_j \beta^a\right)$$

```
tEvoKextrRHSLie =
 tmpLieKextr // ToBasis[Csph] // ToBasis[Csph] // TraceBasisDummy // ComponentArray //
  ToValues
```

$$\left\{\{2\,\kappa_1[r]\,\beta^{r\prime}[r] + \beta^r[r]\,\kappa_1{}'[r],\ 0,\ 0\},\ \{0,\ \beta^r[r]\,\kappa_2{}'[r],\ 0\},\ \left\{0,\ 0,\ \beta^r[r]\,\mathrm{Sin}[th]^2\,\kappa_2{}'[r]\right\}\right\}$$

```
termKevo1 =
 -CD[-i][CD[-j][alpha[r[]]]] // ToBasis[Csph] // ComponentArray // ToValues
```

$$\left\{\left\{\frac{\alpha'[r]\,\gamma_1{}'[r]}{2\,\gamma_1[r]} - \alpha''[r],\ 0,\ 0\right\},\ \left\{0,\ -\frac{\alpha'[r]\,\gamma_2{}'[r]}{2\,\gamma_1[r]},\ 0\right\},\ \left\{0,\ 0,\ -\frac{\mathrm{Sin}[th]^2\,\alpha'[r]\,\gamma_2{}'[r]}{2\,\gamma_1[r]}\right\}\right\}$$

```
tmp = alpha[r[]] RicciCD[-i, -j] // ToBasis[Csph]
```

$\alpha[r]\ R[\nabla]_{ij}$

```
termKevo2 = FullSimplify[
  tmp // ToBasis[Csph] // ComponentArray // ToValues
 ]
```

$$\{\{(\alpha[r]\ (\gamma_1[r]\ \gamma_2'[r]^2 + \gamma_2[r]\ (\gamma_1'[r]\ \gamma_2'[r] - 2\,\gamma_1[r]\ \gamma_2''[r])))\,/\,(2\,\gamma_1[r]\ \gamma_2[r]^2),\ 0,\ 0\},$$
$$\{0,\ (\alpha[r]\ (4\,\gamma_1[r]^2 + \gamma_1'[r]\ \gamma_2'[r] - 2\,\gamma_1[r]\ \gamma_2''[r]))\,/\,(4\,\gamma_1[r]^2),\ 0\},$$
$$\{0,\ 0,\ (\alpha[r]\ \mathrm{Sin}[th]^2\ (4\,\gamma_1[r]^2 + \gamma_1'[r]\ \gamma_2'[r] - 2\,\gamma_1[r]\ \gamma_2''[r]))\,/\,(4\,\gamma_1[r]^2)\}\}$$

```
tmp = alpha[r[]] Kextr[-k, -l] metric[k, l] Kextr[-i, -j] // ToBasis[Csph]
```

$\alpha[r]\ K_{ab}\ K_{ij}\ \gamma^{ab}$

```
termKevo3 = FullSimplify[
  tmp // TraceBasisDummy // ComponentArray // ToValues
 ]
```

$$\left\{\left\{\alpha[r]\ \kappa_1[r]\ \left(\frac{\kappa_1[r]}{\gamma_1[r]} + \frac{2\,\kappa_2[r]}{\gamma_2[r]}\right),\ 0,\ 0\right\},\ \left\{0,\ \alpha[r]\ \kappa_2[r]\ \left(\frac{\kappa_1[r]}{\gamma_1[r]} + \frac{2\,\kappa_2[r]}{\gamma_2[r]}\right),\ 0\right\},\right.$$
$$\left.\{0,\ 0,\ (\alpha[r]\ \kappa_2[r]\ (\gamma_2[r]\ \kappa_1[r] + 2\,\gamma_1[r]\ \kappa_2[r])\ \mathrm{Sin}[th]^2)\,/\,(\gamma_1[r]\ \gamma_2[r])\}\right\}$$

```
tmp = -2 alpha[r[]] Kextr[-i, -k] Kextr[-j, -l] metric[k, l] // ToBasis[Csph]
```

$-2\,\alpha[r]\ K_{ia}\ K_{jb}\ \gamma^{ab}$

```
termKevo4 = FullSimplify[
  tmp // TraceBasisDummy // ComponentArray // ToValues
 ]
```

$$\left\{\left\{-\frac{2\,\alpha[r]\ \kappa_1[r]^2}{\gamma_1[r]},\ 0,\ 0\right\},\ \left\{0,\ -\frac{2\,\alpha[r]\ \kappa_2[r]^2}{\gamma_2[r]},\ 0\right\},\ \left\{0,\ 0,\ -\frac{2\,\alpha[r]\ \kappa_2[r]^2\ \mathrm{Sin}[th]^2}{\gamma_2[r]}\right\}\right\}$$

```
tEvoKextrRHS =
 FullSimplify[termKevo1 + termKevo2 + termKevo3 + termKevo4 + tEvoKextrRHSLie]
```

$$\left\{\left\{\frac{1}{2\,\gamma_1[r]\ \gamma_2[r]^2}\ \left(\gamma_2[r]^2\ (\alpha'[r]\ \gamma_1'[r] + 2\,\gamma_1[r]\ (2\,\kappa_1[r]\ \beta^{r\prime}[r] + \beta^r[r]\ \kappa_1'[r] - \alpha''[r])) + \right.\right.\right.$$
$$\alpha[r]\ (-2\,\gamma_2[r]^2\ \kappa_1[r]^2 + \gamma_1[r]\ \gamma_2'[r]^2 +$$
$$\left.\left.\gamma_2[r]\ (4\,\gamma_1[r]\ \kappa_1[r]\ \kappa_2[r] + \gamma_1'[r]\ \gamma_2'[r] - 2\,\gamma_1[r]\ \gamma_2''[r]))\right),\ 0,\ 0\right\},$$
$$\left\{0,\ \frac{1}{4\,\gamma_1[r]^2}\ (2\,\gamma_1[r]\ (-\alpha'[r]\ \gamma_2'[r] + 2\,\beta^r[r]\ \gamma_1[r]\ \kappa_2'[r]) + \right.$$
$$\left.\alpha[r]\ (\gamma_1'[r]\ \gamma_2'[r] + 2\,\gamma_1[r]\ (2\,\gamma_1[r] + 2\,\kappa_1[r]\ \kappa_2[r] - \gamma_2''[r]))),\ 0\right\},$$
$$\left\{0,\ 0,\ \frac{1}{4\,\gamma_1[r]^2}\ \mathrm{Sin}[th]^2\ (2\,\gamma_1[r]\ (-\alpha'[r]\ \gamma_2'[r] + 2\,\beta^r[r]\ \gamma_1[r]\ \kappa_2'[r]) + \right.$$
$$\left.\left.\alpha[r]\ (\gamma_1'[r]\ \gamma_2'[r] + 2\,\gamma_1[r]\ (2\,\gamma_1[r] + 2\,\kappa_1[r]\ \kappa_2[r] - \gamma_2''[r])))\right\}\right\}$$

```
(*similar to the above we find evo eqns*)
dtkap1 = Expand[tEvoKextrRHS[[1, 1]]]
```

$$-\frac{\alpha[r]\,\kappa_1[r]^2}{\gamma_1[r]}+\frac{2\,\alpha[r]\,\kappa_1[r]\,\kappa_2[r]}{\gamma_2[r]}+2\,\kappa_1[r]\,\beta^{r\prime}[r]+\frac{\alpha'[r]\,\gamma_1'[r]}{2\,\gamma_1[r]}+\frac{\alpha[r]\,\gamma_1'[r]\,\gamma_2'[r]}{2\,\gamma_1[r]\,\gamma_2[r]}+\frac{\alpha[r]\,\gamma_2'[r]^2}{2\,\gamma_2[r]^2}+\beta^r[r]\,\kappa_1'[r]-\alpha''[r]-\frac{\alpha[r]\,\gamma_2''[r]}{\gamma_2[r]}$$

```
dtkap2 = Expand[tEvoKextrRHS[[2, 2]]]
```

$$\alpha[r]+\frac{\alpha[r]\,\kappa_1[r]\,\kappa_2[r]}{\gamma_1[r]}-\frac{\alpha'[r]\,\gamma_2'[r]}{2\,\gamma_1[r]}+\frac{\alpha[r]\,\gamma_1'[r]\,\gamma_2'[r]}{4\,\gamma_1[r]^2}+\beta^r[r]\,\kappa_2'[r]-\frac{\alpha[r]\,\gamma_2''[r]}{2\,\gamma_1[r]}$$

ADMequation summary

Collectively (suppressing arguments):

```
(*expand and suppress arguments with rule*)
Thread[{dtγ1, dtγ2, dtκ1, dtκ2} ==
   ({dtgam1, dtgam2, dtkap1, dtkap2} /. field_[arg_] :> field)
 ] // MatrixForm
```

$$\begin{pmatrix} \mathrm{dt}\gamma 1 == -2\,\alpha\,\kappa_1+2\,\gamma_1\,\beta^{r\prime}+\beta^r\,\gamma_1' \\ \mathrm{dt}\gamma 2 == -2\,\alpha\,\kappa_2+\beta^r\,\gamma_2' \\ \mathrm{dt}\kappa 1 == -\frac{\alpha\,\kappa_1^2}{\gamma_1}+\frac{2\,\alpha\,\kappa_1\,\kappa_2}{\gamma_2}+2\,\kappa_1\,\beta^{r\prime}+\frac{\alpha'\,\gamma_1'}{2\,\gamma_1}+\frac{\alpha\,\gamma_1'\,\gamma_2'}{2\,\gamma_1\,\gamma_2}+\frac{\alpha\,(\gamma_2')^2}{2\,\gamma_2^2}+\beta^r\,\kappa_1'-\alpha''-\frac{\alpha\,\gamma_2''}{\gamma_2} \\ \mathrm{dt}\kappa 2 == \alpha+\frac{\alpha\,\kappa_1\,\kappa_2}{\gamma_1}-\frac{\alpha'\,\gamma_2'}{2\,\gamma_1}+\frac{\alpha\,\gamma_1'\,\gamma_2'}{4\,\gamma_1^2}+\beta^r\,\kappa_2'-\frac{\alpha\,\gamma_2''}{2\,\gamma_1} \end{pmatrix}$$

```
Thread[{ℋ, ℳ1} == ({Ham, Momentum[[1]]} /. field_[arg_] :> field)] // MatrixForm
```

$$\begin{pmatrix} \mathcal{H} == \frac{4\,\gamma_1^2\,(\gamma_2+\kappa_2^2)+2\,\gamma_2\,\gamma_1'\,\gamma_2'+\gamma_1\,((\gamma_2')^2+\gamma_2\,(8\,\kappa_1\,\kappa_2-4\,\gamma_2''))}{2\,\gamma_1^2\,\gamma_2^2} \\ \mathcal{M}1 == \frac{(\gamma_2\,\kappa_1+\gamma_1\,\kappa_2)\,\gamma_2'-2\,\gamma_1\,\gamma_2\,\kappa_2'}{\gamma_1\,\gamma_2^2} \end{pmatrix}$$

Let us check compatibility with Minkowski

```
Ham /. {gam1 → Function[r, 1], gam2 → Function[r, r^2]}
```

$$\frac{1}{2\,r^4}\left(4\,r^2+(-8+8\,\kappa_1[r]\,\kappa_2[r])\,r^2+4\,\left(\kappa_2[r]^2+r^2\right)\right)$$

```
Momentum /. {
  gam1 → Function[r, 1], gam2 → Function[r, r^2]
 }
```

$$\left\{\frac{1}{r^4}\left(2\,r\,\left(\kappa_2[r]+\kappa_1[r]\,r^2\right)-2\,r^2\,\kappa_2'[r]\right),\ 0,\ 0\right\}$$

```
(*need to satisfy constraints*)
```

```
rule = {
   gam1 → Function[r, 1], gam2 → Function[r, r^2],
   kap1 → Function[r, f[r]], kap2 → Function[r, g[r]]
  };
```

```
tmpH = FullSimplify[Ham /. rule]
```

$$\frac{2\, g[r]\, (g[r] + 2\, f[r]\, r^2)}{r^4}$$

```
tmpK = FullSimplify[Momentum /. rule]
```

$$\left\{\frac{2\, (g[r] + r\, (f[r]\, r - g'[r]))}{r^3},\, 0,\, 0\right\}$$

```
(*so we immediately see that taking f=g=0 would be a compatible choice*)
rule = {
   gam1 → Function[r, 1], gam2 → Function[r, r^2],
   kap1 → Function[r, 0], kap2 → Function[r, 0]
  };
```

```
tmpH = FullSimplify[Ham /. rule]
```

0

```
tmpK = FullSimplify[Momentum /. rule]
```

{0, 0, 0}