On 3+1 splitting in stationary circular space-times

Introducing the decomposition

Preliminary arrangements and prerequisites

Launching xAct

The goals of this chapter are:
• Launch xAct with xTensor package
• Launch xAct package ShowTime with Threshold of 1 second for displaying evaluation time
• Fix indices at print (and only at print, for internal calculation they will still be handled by xAct with different symbols)
• Prepare functions for automatic rule application in two variants: First with(out) metric contraction (all possible index permutations)
• Prepare a constant list $Equations, in which **all** resulting equations are to be stored
• Prepare functions to append our results in $Equations and other constant lists which will be developed in later chapters
• Prepare two functions to display from lists. One with Hold[...] operation which will only display the name of the equation, and one with full display of the equation.

In[ ]:=
```
<< xAct`xTensor`
```

```
------------------------------------------------------------

Package xAct`xPerm`  version 1.2.3, {2015, 8, 23}

CopyRight (C) 2003-2020, Jose M. Martin-Garcia, under the General Public License.

Connecting to external MinGW executable...

Connection established.

------------------------------------------------------------

Package xAct`xTensor`  version 1.2.0, {2021, 10, 17}

CopyRight (C) 2002-2021, Jose M. Martin-Garcia, under the General Public License.

------------------------------------------------------------

These packages come with ABSOLUTELY NO WARRANTY; for details type
  Disclaimer[]. This is free software, and you are welcome to redistribute
  it under certain conditions. See the General Public License for details.

------------------------------------------------------------
```

In[ ]:=
```
$DefInfoQ = False;
```

```
In[ ]:= << xAct`ShowTime1`

In[ ]:= $ShowTimeThreshold

Out[ ]= 1

In[ ]:= $PrePrint = ScreenDollarIndices;

In[ ]:= ApplyRule[expr_] :=
         MakeRule[Evaluate[List @@ expr], MetricOn → All, ContractMetrics → True]
        ApplyRuleN[expr_] := MakeRule[Evaluate[List @@ expr]]
```

Now we define a constant set (which is ensured by $), which will later be filled by equations .
A word of caution: If AddEquation is used on an expression several times (say, by accident), the $Equations list will keep all duplicates!
For functions DisplayHOLD & DisplayEQ we use the command Union to NOT display (possibly) accumulated duplicates

```
In[ ]:= $Equations = {};

In[ ]:= SetAttributes[AddEquation, HoldAll]
        AddEquation[expr_Symbol] := AppendTo[$Equations, Hold[expr]]

        SetAttributes[AddToList, HoldAll]
        AddToList[$list_, expr_Symbol] := AppendTo[$list, Hold[expr]]

        DisplayHOLD[expr_List] := expr // Flatten // MatrixForm
        DisplayEQ[expr_List] :=
         expr // ReleaseHold // ScreenDollarIndices // Flatten // MatrixForm
```

Defining the spacetime, normal and induced 3+1 decomposition

Conventions:

Specifying variables which store used conventions:
• The sign of the metric (stored in **STdetmet**)
• The sign of the normal field (stored in **ϵ**) determining the spacetime character of the normal, with explicit display
- the intent is to display it explicitly whenever we can. As the “time/space-likeness” is crucial geometrical information.
• The sign of the 4-acceleration related to normal vector field (stored in **asign**)
- which is set to **ϵ**
• The sign of the normal-orthogonal splitting induced metric tensor (stored in **IMmetdet**)
• The sign of the extrinsic curvature tensor trace (connected to normal orthogonal split) (stored in **ksign**)

First, the normal is unitary, it’s norm denoted by sign:

```
In[•]:= DefConstantSymbol[ϵ]
        ϵ /: Power[ϵ, n_?EvenQ] := 1
        ϵ /: Power[ϵ, n_?OddQ] := ϵ
```

Then the rest of the notation concerning the convention:

```
In[•]:= STmetdet = -1;
        asign = ϵ;
        IMmetdet = 1;
        ksign = 1;
```

Manifold & metric definitions:

Definition of our space-time manifold begins here.
Overview of the code follows as:
• Manifold definition
• Equip it with pseudometric tensor and it's Levi-Civita type covariant derivative, setting out some printing options for postfix and prefix notation
• Setting the printing out option for covariant derivative as postfix
• Printing out setup for curvature tensors
Note that **we use -1 as the determinant sign of the space-time metric** (defined as the first argument of DefMetric, explicitly specified in "Conventions" chapter)

```
In[•]:= DefManifold[M, 4, {α, β, γ, δ, ι, κ, λ, μ, ν, ρ, σ, υ, χ}]
```

```
In[•]:= DefMetric[STmetdet, g[-μ, -ν], CD, {";", "∇"}]
```

```
In[•]:= $CovDFormat = "Postfix";
        SetOptions[ContractMetric, AllowUpperDerivatives → True]
        {PrintAs[RiemannCD] ^= "R", PrintAs[RicciCD] ^= "R", PrintAs[RicciScalarCD] ^= "R"};
Out[•]= {AllowUpperDerivatives → True, OverDerivatives → False}
```

Print test:

```
In[•]:= {RiemannCD[μ, -ν, -κ, -λ], RicciCD[-ν, -λ], RicciScalarCD[]}
Out[•]= {R^μ_νκλ , R_νλ , R}
```

The normal and it's norm

The normal

At this point we need to specify several properties concerning the normal.

A commentary about structure:
Constant symbol ϵ representing the norm of the normal was already defined.
In this part of code normal field is defined, then it's square in various equivalent (mathematically) definitions, simplifying the definitions step by step until $n^\alpha n_\alpha = \epsilon$ is obtained. Noteworthy is the fact that it is **necessary** to introduce this equation as a relation between **scalars**, not general tensors.
This pivotal fact is then taken as a general rule into **$Rules** list

Further on:

• Normal spacetime character is defined as **ϵ**

• Storing the scalar definition of **ϵ** as general rule into **$Rules** list

• Rules are tested

• 4-velocity (the normal) ⊥ it's 4-acceleration relation is developed

```
In[•]:= $EquationsNormal = {};

In[•]:= DefTensor[n[-α], M]

In[•]:= DefInertHead[ABS]

In[•]:= ϵ == Scalar@ABS[n[-μ] × n[μ]]
        Defineϵ = %;
        AddEquation@Defineϵ;
        AddToList[$EquationsNormal, Defineϵ];
```

Out[•]= $\epsilon == \left(\text{ABS}\left[n_{\mu}\ n^{\mu}\right]\right)$

```
In[•]:= NoScalar /@ Defineϵ
        DefineϵNOSC = %;
        AddEquation@DefineϵNOSC;
        AddToList[$EquationsNormal, Defineϵ];
```

Out[•]= $\epsilon == \text{ABS}\left[n_{\alpha}\ n^{\alpha}\right]$

Following directly from the definitions is:

```
In[•]:= n[-α] × n[α] == ϵ
        PutScalar /@ %
        NormToϵNoABS = %;
        AddEquation@NormToϵNoABS;
        AddToList[$EquationsNormal, NormToϵNoABS];
```

Out[•]= $n_{\alpha}\ n^{\alpha} == \epsilon$

Out[•]= $\left(n_{\alpha}\ n^{\alpha}\right) == \epsilon$

```
In[•]:= AutomaticRules[n, ApplyRule@NormToϵNoABS]
```

```
Rules {1, 2} have been declared as generic Rules.
```

```
In[•]:= $Rules
```

Out[•]= $\left\{\text{HoldPattern}\left[\left(n_{\underline{\alpha}}\ n^{\underline{\alpha}}\right)\right] :\mapsto \text{Module}[\{\}, \epsilon], \text{HoldPattern}\left[\left(n_{\underline{\alpha}}\ n^{\underline{\alpha}}\right)\right] :\mapsto \text{Module}[\{\}, \epsilon]\right\}$

Test whether the rules hold accordingly:

```
In[•]:= n[-α] × n[α] × n[-μ] × n[μ] × n[σ] × n[-σ]
        % // PutScalar
        % /. $Rules
```

Out[•]= $n_{\alpha}\ n^{\alpha}\ n_{\mu}\ n^{\mu}\ n_{\sigma}\ n^{\sigma}$

Out[•]= $\left(n_{\alpha}\ n^{\alpha}\right)\left(n_{\mu}\ n^{\mu}\right)\left(n_{\sigma}\ n^{\sigma}\right)$

Out[•]= ϵ

We have to introduce rule for $\left(n_{\mu}\, n^{\mu}\right)_{;\nu} = 0 \Longrightarrow n_{\mu;\nu}\, n^{\mu} = 0$

```
In[•]:= CD[-ν]@(n[α] × n[-α]) == 0
        Times[%, 1 / 2] // ToCanonical
        NGRADNZERO = %;
        AddEquation@NGRADNZERO;
        AddToList[$EquationsNormal, NGRADNZERO];
```

Out[•]= $n^{\alpha}\ n_{\alpha;\nu} + n_{\alpha}\ n^{\alpha}{}_{;\nu} == 0$

Out[•]= $n^{\alpha}\ n_{\alpha;\nu} == 0$

```
In[•]:= DisplayEQ@$EquationsNormal
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \epsilon == \left(\text{ABS}\left[n_{\mu}\ n^{\mu}\right]\right) \\ \epsilon == \left(\text{ABS}\left[n_{\mu}\ n^{\mu}\right]\right) \\ \left(n_{\alpha}\ n^{\alpha}\right) == \epsilon \\ n^{\alpha}\ n_{\alpha;\nu} == 0 \end{pmatrix}$$

Spatial “3D” metric

The structure followed is:

• Definition of the induced metric

- note that previously stored convention is deployed here

• Fixing the printing out options

• Testing the display

Induced metric

Now we introduce the foliation through defining the induced metric (and it’s Levi-Civita type covariant derivative).

Ignoring the (harmless) warning.

```
In[•]:= DefMetric[IMmetdet, h[-μ, -ν], cd, {"|", "D"}, InducedFrom → {g, n}]
        $ExtrinsicKSign = ksign;
        $AccelerationSign = asign;
```

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

```
1.25882
```

Now for the output print:

```
In[•]:= {PrintAs[ExtrinsicKh] ^= "K", PrintAs[Riemanncd] ^= "(3)R", PrintAs[Riccicd] ^= "(3)R",
          PrintAs[RicciScalarcd] ^= "(3)R", PrintAs[Accelerationn] ^= "a"};
        DisplayTest =
         {Accelerationn[-α], ExtrinsicKh[-α, -β], Riemanncd[-α, -β, -γ, -δ], Riccicd[-α, -β]}
```

Out[•]= $\{a_\alpha,\ K_{\alpha\beta},\ {}^{(3)}R_{\alpha\beta\gamma\delta},\ {}^{(3)}R_{\alpha\beta}\}$

Orthogonality test. Above printed tensors should all be orthogonal to normal, thus projected through the induced metric tensor

```
In[•]:= Times[#, n[α]] & /@ DisplayTest
        Times[#, h[α, ι]] & /@ DisplayTest
```

Out[•]= {0, 0, 0, 0}

Out[•]= $\{a^\iota,\ K_\beta{}^\iota,\ {}^{(3)}R^\iota{}_{\beta\gamma\delta},\ {}^{(3)}R^\iota{}_\beta\}$

```
In[•]:= Clear@DisplayTest
```

Would the orthogonality not hold, another terms would appear:

```
In[•]:= RicciCD[-α, -β] × h[α, ι]
        (ProjectorToMetric[%] // ContractMetric) /. $Rules
```

Out[•]= $h^{\alpha\iota}\ R_{\alpha\beta}$

Out[•]= $-\epsilon\ n^\alpha\ n^\iota\ R_{\alpha\beta} + R^\iota{}_\beta$

Basic induced metric properties and consequences

```
In[•]:= $Equations3Dmetric = {};
```

Basic results of induced decomposition:

Since induced metric is already defined, it is redundant to store following rules between the two existing metrics, they are automatically generated together with induced decomposition. Nevertheless it is desirable to deploy them and store them at will, in order to follow the already lined structure of the code.

Following is:

• Defining $h_{\mu\nu} \leftrightarrow g_{\mu\nu}$ relations

```
In[•]:= h[-μ, -ν] == (h[-μ, -ν] // ProjectorToMetric) /. $Rules
        hTOg = %;
        AddEquation@hTOg;
        AddToList[$Equations3Dmetric, hTOg];
```

Out[•]= $h_{\mu\nu} == g_{\mu\nu} - \epsilon\ n_\mu\ n_\nu$

```
In[•]:= Solve[hTOg, g[-μ, -ν]] // Flatten
        %[[1]] /. Rule → Equal
        gTOh = %;
        AddEquation@gTOh;
        AddToList[$Equations3Dmetric, gTOh];
```

Out[•]= $\{ g_{\mu\nu} \to h_{\mu\nu} + \epsilon\, n_\mu\, n_\nu \}$

Out[•]= $g_{\mu\nu} == h_{\mu\nu} + \epsilon\, n_\mu\, n_\nu$

```
In[•]:= DisplayEQ@$Equations3Dmetric
```

Out[•]//MatrixForm=

$$\begin{pmatrix} h_{\mu\nu} == g_{\mu\nu} - \epsilon\, n_\mu\, n_\nu \\ g_{\mu\nu} == h_{\mu\nu} + \epsilon\, n_\mu\, n_\nu \end{pmatrix}$$

Covariant derivative of the normal

Following is:

• Defining relation between 4D gradient of the normal & 3D extrinsic curvature & normal acceleration field

• Fixing the normal acceleration definition

Now we wish to arrive at equation for the space-time covariant derivative of the normal in terms of Extrinsic Curvature tensor. It is achieved by:

```
In[•]:= CD[-β]@n[-α] == (CD[-β]@n[-α] // GradNormalToExtrinsicK)
        CDN = %;
        AddEquation@CDN;
        AddToList[$EquationsNormal, CDN];
```

Out[•]= $n_{\alpha;\beta} == K_{\beta\alpha} + \epsilon\, a_\alpha\, n_\beta$

```
In[•]:= Times[#, n[β]] & /@ CDN
        ToCanonical /@ ContractMetric /@ %;
        PutScalar /@ %
        % /. $Rules
        NCDNTOAcc = %;
        AddEquation@NCDNTOAcc;
        AddToList[$EquationsNormal, NCDNTOAcc];
```

Out[•]= $n^\beta\, n_{\alpha;\beta} == \left(K_{\beta\alpha} + \epsilon\, a_\alpha\, n_\beta \right) n^\beta$

Out[•]= $n^\beta\, n_{\alpha;\beta} == \epsilon\, a_\alpha \left(n_\beta\, n^\beta \right)$

Out[•]= $n^\beta\, n_{\alpha;\beta} == a_\alpha$

```
In[•]:= Solve[NCDNTOAcc, Accelerationn[-α]];
        % // Flatten
        %[[1]] /. Rule → Equal
        AccTOCDN = %;
        AddEquation@AccTOCDN;
        AddToList[$EquationsNormal, AccTOCDN];
```

Out[•]= $\left\{ a_{\alpha} \to n^{\beta}\ n_{\alpha;\beta} \right\}$

Out[•]= $a_{\alpha} == n^{\beta}\ n_{\alpha;\beta}$

```
In[•]:= DisplayEQ@$EquationsNormal
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \in == \left(\text{ABS}\left[n_{\mu}\ n^{\mu}\right]\right) \\ \in == \left(\text{ABS}\left[n_{\mu}\ n^{\mu}\right]\right) \\ \left(n_{\alpha}\ n^{\alpha}\right) == \in \\ n^{\alpha}\ n_{\alpha;\nu} == 0 \\ n_{\alpha;\beta} == K_{\beta\alpha} + \in a_{\alpha}\ n_{\beta} \\ n^{\beta}\ n_{\alpha;\beta} == a_{\alpha} \\ a_{\alpha} == n^{\beta}\ n_{\alpha;\beta} \end{pmatrix}$$

Extrinsic curvature & 4D-Covariant derivative of the normal

```
In[•]:= $Equations3DExtCurv = {};
```

Following is:

• Defining the extrinsic curvature tensor

• Solving the previously developed relation **CDN** for 3D extrinsic curvature tensor

```
In[•]:= CDN
        ProjectWith[h] /@ CDN
        % // ScreenDollarIndices;
        Solve[%, ExtrinsicKh[-α, -β]] // Flatten
        %[[1]] /. Rule → Equal
        ExtrinsicToGradNormal = %;
```

Out[•]= $n_{\alpha;\beta} == K_{\beta\alpha} + \in a_{\alpha}\ n_{\beta}$

Out[•]= $h_{\alpha}^{\ \gamma}\ h_{\beta}^{\ \delta}\ n_{\gamma;\delta} == K_{\alpha\beta}$

Out[•]= $\left\{ K_{\alpha\beta} \to h_{\alpha}^{\ \gamma}\ h_{\beta}^{\ \delta}\ n_{\gamma;\delta} \right\}$

Out[•]= $K_{\alpha\beta} == h_{\alpha}^{\ \gamma}\ h_{\beta}^{\ \delta}\ n_{\gamma;\delta}$

```
In[•]:= AddEquation@ExtrinsicToGradNormal;
        AddToList[$Equations3DExtCurv, ExtrinsicToGradNormal];
```

```
In[•]:= CDN
        Solve[CDN, ExtrinsicKh[-β, -α]] // Flatten
        %[[1]] /. Rule → Equal
        ExtrinsicKToCDNAcc = %;
        AddEquation@ExtrinsicKToCDNAcc;
        AddToList[$Equations3DExtCurv, ExtrinsicKToCDNAcc];
```

Out[•]= $n_{\alpha;\beta} == K_{\beta\alpha} + \epsilon\, a_\alpha\, n_\beta$

Out[•]= $\left\{K_{\beta\alpha} \to -\epsilon\, a_\alpha\, n_\beta + n_{\alpha;\beta}\right\}$

Out[•]= $K_{\beta\alpha} == -\epsilon\, a_\alpha\, n_\beta + n_{\alpha;\beta}$

Lie derivative of induced metric to extrinsic curvature tensor

Following is:

• Establishing the relation between normal-4D-Lie derivative & the 3D Extrinsic curvature

Now we attempt to show that the Lie derivative along the normal field of induced metric is in fact twice the extrinsic curvature tensor, however this rule is now automated, one only has to account for symmetry of the induced metric, see:

```
In[•]:= LieD[n[α]]@h[-μ, -ν]
        % // ToCanonical
```

Out[•]= $K_{\mu\nu} + K_{\nu\mu}$

Out[•]= $2\, K_{\mu\nu}$

We nevertheless wish to store it. There are two (rather clumsy) options. First of them is to use Defer function, which protects the expression from being evaluated, yet prints it out normally. Second one is, again, to use Hold function, which displays Hold[...] expression. We choose to use the latter, as Defer function works poorly with function Solve when using xAct.

```
In[•]:= (Hold[LieD[n[α]]@h[-μ, -ν]]) == (LieD[n[α]]@h[-μ, -ν] // ToCanonical)
        Solve[%, ExtrinsicKh[-μ, -ν]] // Flatten
        %[[1]] /. Rule → Equal
        ExtrinsicKToLieD = %;
        AddToList[$Equations3DExtCurv, ExtrinsicKToLieD];
```

Out[•]= $\text{Hold}\left[\mathcal{L}_n\, h_{\mu\nu}\right] == 2\, K_{\mu\nu}$

Out[•]= $\left\{K_{\mu\nu} \to \frac{1}{2}\,\text{Hold}\left[\mathcal{L}_n\, h_{\mu\nu}\right]\right\}$

Out[•]= $K_{\mu\nu} == \frac{1}{2}\,\text{Hold}\left[\mathcal{L}_n\, h_{\mu\nu}\right]$

In[•]:=
```
DisplayEQ@$Equations3DExtCurv
```

Out[•]//MatrixForm=

$$\begin{pmatrix} K_{\alpha\beta} == h_{\alpha}^{\ \gamma}\ h_{\beta}^{\ \delta}\ n_{\gamma;\delta} \\ K_{\beta\alpha} == -\epsilon\ a_{\alpha}\ n_{\beta} + n_{\alpha;\beta} \\ K_{\mu\nu} == \frac{1}{2}\ \text{Hold}\left[\mathcal{L}_{n}\, h_{\mu\nu}\right] \end{pmatrix}$$

Induced covariant derivative and 3D Riemann tensor

Induced covariant derivative & it's properties

In[•]:=
```
$Equations3Dderivative = {};
```

General definition & some of it's expressions involving $K_{\alpha\beta}$ & a_{α}

The 3D derivative **cd** was already defined together with induced metric $h_{\mu\nu}$.

Following is:

- Vector V^{α} orthogonal to n_{α} is defined
- It's 3D-covariant derivative is introduced from definition and worked upon
- 3D-covariant divergence is developed

Also worth of mentioning is that these rules albeit expressed for some general vector V^{α} can be

In[•]:=
```
DefTensor[V[α], M, OrthogonalTo → {n[-α]}, ProjectedWith → {h[-α, -β]}];
```

In[•]:=
```
cd[-β]@V[-α] == CD[-δ]@V[-γ] × h[δ, -β] × h[γ, -α]
cd3DCovector = %;
```

Out[•]=
$$V_{\alpha\,|\,\beta} == h^{\gamma}{}_{\alpha}\ h^{\delta}{}_{\beta}\ V_{\gamma;\delta}$$

In[•]:=
```
AddEquation@cd3DCovector;
AddToList[$Equations3Dderivative, cd3DCovector];
```

```
In[•]:= cd[-β]@V[-α] == CD[-δ]@V[-γ] × h[δ, -β] (h[γ, -α] /. ApplyRule@hTOg)
        % // Expand // ScreenDollarIndices // ContractMetric;
        % /. MakeRule[{n[γ] × CD[-δ]@V[-γ], -CD[-δ]@n[γ] × V[-γ]},
          MetricOn → All, ContractMetrics → True]
        (*Same as for normal, since vector V was defined as orthogonal to the normal field,
        it holds that covariant derivative of their contraction is zero,
        thus we deploy reverse Leibnitz rule
         - as this adjustment is not interesting on itself,
        we do not store it for further usage*)
        cd3DCovectorExpanded = %;
        % /. ApplyRule@CDN;
        % // Expand
        cd3CDCovectorExtrK = %;
```

$$\text{Out[•]}= V_{\alpha|\beta} == h^{\delta}{}_{\beta}\ \left(\delta_{\alpha}{}^{\gamma} - \epsilon\ n_{\alpha}\ n^{\gamma}\right)\ V_{\gamma;\delta}$$

$$\text{Out[•]}= V_{\alpha|\beta} == \epsilon\ h^{\delta}{}_{\beta}\ n_{\alpha}\ V_{\gamma}\ n^{\gamma}{}_{;\delta} + h^{\delta}{}_{\beta}\ V_{\alpha;\delta}$$

$$\text{Out[•]}= V_{\alpha|\beta} == \epsilon\ K^{\gamma}{}_{\beta}\ n_{\alpha}\ V_{\gamma} + h^{\delta}{}_{\beta}\ V_{\alpha;\delta}$$

```
In[•]:= AddEquation@cd3DCovectorExpanded;
        AddEquation@cd3CDCovectorExtrK;
        AddToList[$Equations3Dderivative, cd3DCovectorExpanded];
        AddToList[$Equations3Dderivative, cd3CDCovectorExtrK];
```

Important realization: These rules are in fact all we will need for expressing the 3D-derivative of any co/vector field orthogonal to the normal!

Every such field can now substituted for V_α in above gradients via WL Rule For example the normal acceleration field a_α:

```
In[•]:= cd3CDCovectorExtrK
        % /. V → Accelerationn
```

$$\text{Out[•]}= V_{\alpha|\beta} == \epsilon\ K^{\gamma}{}_{\beta}\ n_{\alpha}\ V_{\gamma} + h^{\delta}{}_{\beta}\ V_{\alpha;\delta}$$

$$\text{Out[•]}= a_{\alpha|\beta} == \epsilon\ a_{\gamma}\ K^{\gamma}{}_{\beta}\ n_{\alpha} + h^{\delta}{}_{\beta}\ a_{\alpha;\delta}$$

The same 3D-gradient relations hold for 3D contravariant gradient V^α, $V^{\alpha|\beta}$. The same identities are automatically recognized from xAct pattern matching from the previous $V_{\alpha|\beta}$ identities. Therefore we do not display them separately.

However induced divergence deserves it's own display:

```
In[•]:= cd[-λ]@V[λ] == h[λ, ν] × cd[-λ]@V[-ν]
        cd3DDivTOContr = %;
```

$$\text{Out[•]}= V^{\lambda}{}_{|\lambda} == h^{\lambda\nu}\ V_{\nu|\lambda}$$

```
In[•]:= AddEquation@cd3DDivTOContr;
        AddToList[$Equations3Dderivative, cd3DDivTOContr];
```

```
In[•]:= cd[-λ]@V[λ] == h[λ, ν] × ProjectWith[h][CD[-λ]@V[-ν]]
        cd3DDivTOCDContr = %;
```

Out[•]= $V^{\lambda}{}_{|\lambda} == h^{\beta\alpha}\ V_{\beta;\alpha}$

```
In[•]:= AddEquation@cd3DDivTOCDContr;
        AddToList[$Equations3Dderivative, cd3DDivTOCDContr];
```

```
In[•]:= cd3DDivTOCDContr /. ApplyRule@hTOg;
        % // Expand // ContractMetric
        cd3DDivTOCDExpanded = %;
        % /. MakeRule[{n[β] × CD[-α]@V[-β], -CD[-α]@n[β] × V[-β]},
            MetricOn → All, ContractMetrics → True] // ScreenDollarIndices
        % /. ApplyRule@NCDNTOAcc
        cd3DDivTOAccCD = %;
```

Out[•]= $V^{\lambda}{}_{|\lambda} == V^{\alpha}{}_{;\alpha} - \in n^{\alpha}\ n^{\beta}\ V_{\beta;\alpha}$

Out[•]= $V^{\lambda}{}_{|\lambda} == \in n^{\alpha}\ V_{\beta}\ n^{\beta}{}_{;\alpha} + V^{\alpha}{}_{;\alpha}$

Out[•]= $V^{\lambda}{}_{|\lambda} == \in a^{\beta}\ V_{\beta} + V^{\alpha}{}_{;\alpha}$

```
In[•]:= AddEquation@cd3DDivTOCDExpanded;
        AddEquation@cd3DDivTOAccCD;
        AddToList[$Equations3Dderivative, cd3DDivTOCDExpanded];
        AddToList[$Equations3Dderivative, cd3DDivTOAccCD];
```

Induced 3D-Riemann tensor

The 3D-Riemann tensor arises from relation (note that the **cd** - 3D covariant derivative, is tied to the connection of the Levi-Civita type and thus has vanishing torsion).

Following is:

• The 3D-Riemann tensor commutator definition

```
In[•]:= Riemanncd[μ, -ν, -κ, -λ] × V[-μ] == cd[-λ]@cd[-κ]@V[-ν] - cd[-κ]@cd[-λ]@V[-ν]
        Riemann3Dcommutdef = %;
```

Out[•]= ${}^{(3)}R^{\mu}{}_{\nu\kappa\lambda}\ V_{\mu} == -V_{\nu|\lambda|\kappa} + V_{\nu|\kappa|\lambda}$

```
In[•]:= AddEquation@Riemann3Dcommutdef;
        AddToList[$Equations3Dderivative, Riemann3Dcommutdef];
```

In[•]:=
```
DisplayEQ@$Equations3Dderivative
```

Out[•]//MatrixForm=

$$\begin{pmatrix} V_{\alpha|\beta} == h^{\gamma}{}_{\alpha}\ h^{\delta}{}_{\beta}\ V_{\gamma;\delta} \\ V_{\alpha|\beta} == \in h^{\delta}{}_{\beta}\ n_{\alpha}\ V_{\gamma}\ n^{\gamma}{}_{;\delta} + h^{\delta}{}_{\beta}\ V_{\alpha;\delta} \\ V_{\alpha|\beta} == \in K^{\gamma}{}_{\beta}\ n_{\alpha}\ V_{\gamma} + h^{\delta}{}_{\beta}\ V_{\alpha;\delta} \\ V^{\lambda}{}_{|\lambda} == h^{\lambda\nu}\ V_{\nu|\lambda} \\ V^{\lambda}{}_{|\lambda} == h^{\beta\alpha}\ V_{\beta;\alpha} \\ V^{\lambda}{}_{|\lambda} == V^{\alpha}{}_{;\alpha} - \in n^{\alpha}\ n^{\beta}\ V_{\beta;\alpha} \\ V^{\lambda}{}_{|\lambda} == \in a^{\beta}\ V_{\beta} + V^{\alpha}{}_{;\alpha} \\ {}^{(3)}R^{\mu}{}_{\nu\kappa\lambda}\ V_{\mu} == -V_{\nu|\lambda|\kappa} + V_{\nu|\kappa|\lambda} \end{pmatrix}$$

Projectors

Following is :

• Explanation of an object called Projector (in xAct)

• Fixing printing out option of some projected entities

In further chapters projectors involving induced metric will appear. They are just regular contraction with metric in all indices through $h_{\alpha\beta}$, however they are stored as different object. For example the Riemann tensor projected through induced metric reads:

In[•]:=
```
Projectorh[RiemannCD[-α, -β, -γ, -δ]] == ProjectWith[h][RiemannCD[-α, -β, -γ, -δ]]
```

Out[•]= $\overset{h}{P}\left[R_{\alpha\beta\gamma\delta}\right] == h_{\alpha}{}^{\iota}\ h_{\beta}{}^{\kappa}\ h_{\gamma}{}^{\lambda}\ h_{\delta}{}^{\mu}\ R_{\iota\kappa\lambda\mu}$

Orthogonal projection by those projectors do not change already spatial quantities. We have to explicitly ask xAct to fix the printout for such objects. We enforce this behaviour by function **HoldPattern**. The function **OrthogonalToVectorQ[..]@expr** then checks whether expression is orthogonal to the argument of this function, and returns boolean value. This is implemented by **Condition** , or **/;** in short.

We enforce the behavior for both lie derivative and space-time derivative.

In[•]:=
```
HoldPattern@Projectorh@LieD[n[σ_]]@expr_ :=
 LieD[n[σ]]@expr /; OrthogonalToVectorQ[n]@expr
```

In[•]:=
```
HoldPattern@Projectorh@CD[α_]@expr_ := cd[α]@expr /; OrthogonalToVectorQ[n]@expr
```

And check whether we have prompted the behaviour correctly:

In[•]:=
```
DisplayTest1 = Projectorh@LieD[n[α]]@Accelerationn[-β]
DisplayTest2 = Projectorh@CD[-α]@Accelerationn[-β]
{n[β] × Accelerationn[-β], n[β] DisplayTest1, (n[β]) DisplayTest2 // ToCanonical}
```

Out[•]= $\mathcal{L}_{n}\, a_{\beta}$

Out[•]= $a_{\beta|\alpha}$

Out[•]= {0, 0, 0}

Orthogonality holds.

```
In[•]:= Clear@DisplayTest1
        Clear@DisplayTest2
```

Riemann, Ricci and Ricci scalar tensors 3+1 decomposition

Choice of approach, last preparations before the decomposition

Quick remark about further approach & the GaussCodazzi xAct function.

At this point two options are at hand.

First, working out the decomposition of the Riemann tensor manually as:

```
In[•]:= RiemannCD[-α, -β, -γ, -δ] ==
         RiemannCD[μ, ν, κ, λ] (g[-μ, -α] × g[-ν, -β] × g[-κ, -γ] × g[-λ, -δ] /. ApplyRule@gTOh)
        % // Expand;
        % // ToCanonical
```

$$\text{Out[•]= } R_{\alpha\beta\gamma\delta} == \left(h_{\kappa\gamma} + \epsilon\, n_\gamma\, n_\kappa\right)\left(h_{\lambda\delta} + \epsilon\, n_\delta\, n_\lambda\right)\left(h_{\mu\alpha} + \epsilon\, n_\alpha\, n_\mu\right)\left(h_{\nu\beta} + \epsilon\, n_\beta\, n_\nu\right) R^{\mu\nu\kappa\lambda}$$

$$\begin{aligned}\text{Out[•]= } R_{\alpha\beta\gamma\delta} == {} & h_\alpha{}^\kappa\, h_\beta{}^\lambda\, h_\gamma{}^\mu\, h_\delta{}^\nu\, R_{\kappa\lambda\mu\nu} + \epsilon\, h_\beta{}^\lambda\, h_\gamma{}^\mu\, h_\delta{}^\nu\, n_\alpha\, n^\kappa\, R_{\kappa\lambda\mu\nu} - \\ & \epsilon\, h_\alpha{}^\lambda\, h_\gamma{}^\mu\, h_\delta{}^\nu\, n_\beta\, n^\kappa\, R_{\kappa\lambda\mu\nu} + h_\beta{}^\mu\, h_\delta{}^\nu\, n_\alpha\, n_\gamma\, n^\kappa\, n^\lambda\, R_{\kappa\mu\lambda\nu} - \\ & h_\alpha{}^\mu\, h_\delta{}^\nu\, n_\beta\, n_\gamma\, n^\kappa\, n^\lambda\, R_{\kappa\mu\lambda\nu} - h_\beta{}^\mu\, h_\gamma{}^\nu\, n_\alpha\, n_\delta\, n^\kappa\, n^\lambda\, R_{\kappa\mu\lambda\nu} + \\ & h_\alpha{}^\mu\, h_\gamma{}^\nu\, n_\beta\, n_\delta\, n^\kappa\, n^\lambda\, R_{\kappa\mu\lambda\nu} + \epsilon\, h_\alpha{}^\lambda\, h_\beta{}^\mu\, h_\delta{}^\nu\, n_\gamma\, n^\kappa\, R_{\kappa\nu\lambda\mu} - \epsilon\, h_\alpha{}^\lambda\, h_\beta{}^\mu\, h_\gamma{}^\nu\, n_\delta\, n^\kappa\, R_{\kappa\nu\lambda\mu}\end{aligned}$$

And from here expressing every Riemann projection term from already known identities for 3D-covariant gradient such as

```
In[•]:= (RiemannCD[-μ, -ν, -κ, -λ] × V[μ] == CD[-λ]@CD[-κ]@V[-ν] - CD[-κ]@CD[-λ]@V[-ν] //
           ProjectWith[h]) /. V → n
        CDN
        (CD[-γ] /@ %) // Expand
```

$$\text{Out[•]= } h_\kappa{}^\alpha\, h_\lambda{}^\beta\, h_\nu{}^\gamma\, n^\mu\, R_{\mu\gamma\alpha\beta} == -h_\kappa{}^\alpha\, h_\lambda{}^\beta\, h_\nu{}^\gamma\, n_{\gamma;\beta;\alpha} + h_\kappa{}^\alpha\, h_\lambda{}^\beta\, h_\nu{}^\gamma\, n_{\gamma;\alpha;\beta}$$

$$\text{Out[•]= } n_{\alpha;\beta} == K_{\beta\alpha} + \epsilon\, a_\alpha\, n_\beta$$

$$\text{Out[•]= } n_{\alpha;\beta;\gamma} == \epsilon\, n_\beta\, a_{\alpha;\gamma} + K_{\beta\alpha;\gamma} + \epsilon\, a_\alpha\, n_{\beta;\gamma}$$

This approach is obvious and involves a lot of work.

Second option, using xAct's GaussCodazzi function, which already contains all the projections.

Full decomposition of the Riemann tensor is, after defining the normal and induced metric is as follows:

```
In[•]:= RiemannCD[-α, -β, -γ, -δ] == GaussCodazzi[RiemannCD[-α, -β, -γ, -δ]] /. $Rules
```

$$\begin{aligned}\text{Out}[\bullet]= {} & R_{\alpha\beta\gamma\delta} == \in K_{\alpha\delta} K_{\beta\gamma} - \in K_{\alpha\gamma} K_{\beta\delta} - \in a_\beta a_\delta n_\alpha n_\gamma - K_\beta{}^\perp K_{\delta\perp} n_\alpha n_\gamma + \in a_\alpha a_\delta n_\beta n_\gamma + \\ & K_\alpha{}^\perp K_{\delta\perp} n_\beta n_\gamma + \in a_\beta a_\gamma n_\alpha n_\delta + K_\beta{}^\perp K_{\gamma\perp} n_\alpha n_\delta - \in a_\alpha a_\gamma n_\beta n_\delta - K_\alpha{}^\perp K_{\gamma\perp} n_\beta n_\delta - \\ & n_\beta n_\delta \underset{h}{P}\left[n^\perp K_{\alpha\gamma;\perp}\right] + n_\beta n_\gamma \underset{h}{P}\left[n^\perp K_{\alpha\delta;\perp}\right] + n_\alpha n_\delta \underset{h}{P}\left[n^\perp K_{\beta\gamma;\perp}\right] - n_\alpha n_\gamma \underset{h}{P}\left[n^\perp K_{\beta\delta;\perp}\right] + \\ & {}^{(3)}R_{\alpha\beta\gamma\delta} + \in n_\delta K_{\beta\gamma|\alpha} - \in n_\gamma K_{\beta\delta|\alpha} - \in n_\delta K_{\alpha\gamma|\beta} + \in n_\gamma K_{\alpha\delta|\beta} + \in n_\beta K_{\alpha\delta|\gamma} - \in n_\alpha K_{\beta\delta|\gamma} + \\ & \in \left(\in n_\beta n_\delta a_{\alpha|\gamma} - \in n_\alpha n_\delta a_{\beta|\gamma} - \in n_\beta n_\gamma a_{\alpha|\delta} + \in n_\alpha n_\gamma a_{\beta|\delta}\right) - \in n_\beta K_{\alpha\gamma|\delta} + \in n_\alpha K_{\beta\gamma|\delta}\end{aligned}$$

As we can see term by term, we have every we need already decomposed into 3D entities, **with the exception of the space-time covariant gradient of the extrinsic curvature tensor**. We wish to express the equations which take account every single possible AND unique projection of the space-time Riemann tensor

• Project through every free index → Gauss Equation

• First contract in either one of the first pair of indices with the normal, then project through the remaining three indices → Codazzi Equation

• First contract in either one of the first pair AND in either one of the second pair of indices, then project through remaining two indices → Ricci Equation

We will express the Ricci equation firstly with space-time covariant gradient of the extrinsic curvature explicitly, and then convert the gradient into another terms, one of which will contain the Lie derivative of extrinsic curvature.

For that we will develop the decomposition of the covariant gradient of extrinsic curvature.

Orthogonal decomposition of $K_{\alpha\beta;\gamma}$

Following is:

• Derivation of the decomposition of term $n^\gamma K_{\alpha\beta;\gamma}$ from the normal-4D-Lie derivative

• Decomposition of the term $K_{\alpha\beta;\gamma}$ into 3D entities

The starting point is the Lie derivative which we write out explicitly as an equation, and expand the Lie derivative into space-time covariant derivative. We wish to first find an equation to decompose $n^\gamma K_{\alpha\beta;\gamma}$ into purely spatial terms, one of which will involve normal-4D-Lie derivative.

Amidst the adjustments a need arises to first decompose the gradient of the normal into extrinsic curvature, then, to get rid of other terms involving normal and extrinsic curvature such as $n^\beta K_{\alpha\beta;\gamma}$, we deploy the fact that $0 = \left(n^\beta K_{\alpha\beta}\right)_{;\gamma} = n^\beta{}_{;\gamma} K_{\alpha\beta} + n^\beta K_{\alpha\beta;\gamma}$ and deploy Leibniz rule as a temporary rule. That gives raise to terms involving the gradient of the normal, thus we again convert those terms into extrinsic curvature.

```
In[•]:= LieD[n[σ]]@ExtrinsicKh[-α, -β] == LieDToCovD[LieD[n[σ]]@ExtrinsicKh[-α, -β], CD]
        ScreenDollarIndices /@ %;
        MapAt[Hold, %, Position[%, n[γ] × CD[-γ][ExtrinsicKh[-α, -β]]]];
        Solve[%, Hold[n[γ] × CD[-γ][ExtrinsicKh[-α, -β]]]] // Flatten // ReleaseHold;
        %[[1]] /. Rule → Equal;
        % // GradNormalToExtrinsicK;
        % // ToCanonical
        NCDExtrinsicK = %;
```

$$\text{Out[•]=}\quad \mathcal{L}_n K_{\alpha\beta} == K_{\gamma\beta}\ n^{\gamma}{}_{;\alpha} + K_{\alpha\gamma}\ n^{\gamma}{}_{;\beta} + n^{\gamma}\ K_{\alpha\beta;\gamma}$$

$$\text{Out[•]=}\quad n^{\gamma}\ K_{\alpha\beta;\gamma} == -2\ K_{\alpha}{}^{\gamma}\ K_{\beta\gamma} - \in a^{\gamma}\ K_{\beta\gamma}\ n_{\alpha} - \in a^{\gamma}\ K_{\alpha\gamma}\ n_{\beta} + \mathcal{L}_n K_{\alpha\beta}$$

```
In[•]:= AddEquation@NCDExtrinsicK;
        AddToList[$Equations3DExtCurv, NCDExtrinsicK];
```

Now we wish to liberate extrinsic curvature gradient along the normal from the normal.
We start with the general relation, when convert the gradient along the normal into terms involving the Lie derivative (using **NCDExtrinsicK**), and we make a temporary rule (as previously) in order to use Leibniz rule.

```
In[•]:= CD[-α]@ExtrinsicKh[-β, -γ] ==
          InducedDecomposition[CD[-α]@ExtrinsicKh[-β, -γ], {h, n}];
        % /. ApplyRule@NCDExtrinsicK;
        Expand /@ %;
        % /. MakeRule[{n[γ] × CD[-α]@ExtrinsicKh[-β, -γ],
            -CD[-α]@n[γ] × ExtrinsicKh[-β, -γ]}, MetricOn → All, ContractMetrics → True];
        % // GradNormalToExtrinsicK;
        % // Expand;
        % /. Projectorh → ProjectWith[h];
        NoScalar /@ %;
        PutScalar /@ %;
        ToCanonical /@ %;
        % /. $Rules
        DecoCDExtrinsicK = %;
```

$$\text{Out[•]=}\quad K_{\beta\gamma;\alpha} == -2 \in K_{\beta}{}^{\delta}\ K_{\gamma\delta}\ n_{\alpha} - \in K_{\alpha}{}^{\delta}\ K_{\gamma\delta}\ n_{\beta} - a^{\delta}\ K_{\gamma\delta}\ n_{\alpha}\ n_{\beta} - \in K_{\alpha}{}^{\delta}\ K_{\beta\delta}\ n_{\gamma} - a^{\delta}\ K_{\beta\delta}\ n_{\alpha}\ n_{\gamma} + K_{\beta\gamma|\alpha} + \in n_{\alpha}\ \left(\mathcal{L}_n K_{\beta\gamma}\right)$$

```
In[•]:= AddEquation@DecoCDExtrinsicK;
        AddToList[$Equations3DExtCurv, DecoCDExtrinsicK];
```

```
In[•]:= DisplayEQ@$Equations3DExtCurv
```

Out[•]//MatrixForm=

$$\begin{pmatrix} K_{\alpha\beta} == h_{\alpha}{}^{\gamma}\ h_{\beta}{}^{\delta}\ n_{\gamma;\delta} \\ K_{\beta\alpha} == -\in a_{\alpha}\ n_{\beta} + n_{\alpha;\beta} \\ K_{\mu\nu} == \frac{1}{2}\,\text{Hold}\left[\mathcal{L}_n h_{\mu\nu}\right] \\ n^{\gamma}\ K_{\alpha\beta;\gamma} == -2\ K_{\alpha}{}^{\gamma}\ K_{\beta\gamma} - \in a^{\gamma}\ K_{\beta\gamma}\ n_{\alpha} - \in a^{\gamma}\ K_{\alpha\gamma}\ n_{\beta} + \mathcal{L}_n K_{\alpha\beta} \\ K_{\beta\gamma;\alpha} == -2 \in K_{\beta}{}^{\delta}\ K_{\gamma\delta}\ n_{\alpha} - \in K_{\alpha}{}^{\delta}\ K_{\gamma\delta}\ n_{\beta} - a^{\delta}\ K_{\gamma\delta}\ n_{\alpha}\ n_{\beta} - \in K_{\alpha}{}^{\delta}\ K_{\beta\delta}\ n_{\gamma} - a^{\delta}\ K_{\beta\delta}\ n_{\alpha}\ n_{\gamma} + K_{\beta} \end{pmatrix}$$

3+1 Riemann tensor decomposition

Every necessity for decomposition is taken care of, thus we embark on expressing the Gauss, Codazzi and Ricci equation for the Riemann tensor.
As stated previously, the approach is to take general (pre-calculated in xAct) decomposition of the Riemann tensor and factor out every term, expand upon possible different (and interesting) formulations, and then factor out the decomposition in somewhat more readable form.
Stressing of the last time that the entire approach could be done without apriori calculated decomposition, however the latter approach is more demanding. The Gauss/Codazzi/Ricci equations are well known results, thus there's little benefit in "verifying them" explicitly here.

Following is:
• Factoring out the Gauss & Codazzi equations (straightforward task) for Riemann tensor
• Factoring out 3 different formulations of the Ricci equation
- First of the formulations is just the basic definition
- Second formulations involves the gradient of the extrinsic curvature along the normal field, $n^\gamma K_{\alpha\beta;\gamma}$
- Third formulation involves somewhat "same" information in the language of the normal 4D-Lie derivative, $\mathcal{L}_{\boldsymbol{n}} K_{\alpha\beta}$

In[•]:=
```
$EquationsRiemann3DDeco = {};
```

In[•]:=
```
RiemannCD[-α, -β, -γ, -δ] == GaussCodazzi[RiemannCD[-α, -β, -γ, -δ], h]
Deco3DRiemannFull = %;
```

Out[•]=
$$R_{\alpha\beta\gamma\delta} == {}^{(3)}R_{\alpha\beta\gamma\delta} - \frac{K_\beta{}^\iota\, K_{\delta\iota}\, n_\alpha\, n_\gamma}{(n_\mu\, n^\mu)^2} + \frac{K_\alpha{}^\iota\, K_{\delta\iota}\, n_\beta\, n_\gamma}{(n_\mu\, n^\mu)^2} + \frac{K_\beta{}^\iota\, K_{\gamma\iota}\, n_\alpha\, n_\delta}{(n_\mu\, n^\mu)^2} - \frac{K_\alpha{}^\iota\, K_{\gamma\iota}\, n_\beta\, n_\delta}{(n_\mu\, n^\mu)^2} -$$
$$\frac{n_\beta\, n_\delta\, P_h\left[n^\iota\, K_{\alpha\gamma;\iota}\right]}{(n_\mu\, n^\mu)^2} + \frac{n_\beta\, n_\gamma\, P_h\left[n^\iota\, K_{\alpha\delta;\iota}\right]}{(n_\mu\, n^\mu)^2} + \frac{n_\alpha\, n_\delta\, P_h\left[n^\iota\, K_{\beta\gamma;\iota}\right]}{(n_\mu\, n^\mu)^2} - \frac{n_\alpha\, n_\gamma\, P_h\left[n^\iota\, K_{\beta\delta;\iota}\right]}{(n_\mu\, n^\mu)^2} +$$
$$\frac{K_{\alpha\delta}\, K_{\beta\gamma}}{(n_\mu\, n^\mu)} - \frac{K_{\alpha\gamma}\, K_{\beta\delta}}{(n_\mu\, n^\mu)} - \frac{a_\beta\, a_\delta\, n_\alpha\, n_\gamma}{(n_\mu\, n^\mu)} + \frac{a_\alpha\, a_\delta\, n_\beta\, n_\gamma}{(n_\mu\, n^\mu)} + \frac{a_\beta\, a_\gamma\, n_\alpha\, n_\delta}{(n_\mu\, n^\mu)} - \frac{a_\alpha\, a_\gamma\, n_\beta\, n_\delta}{(n_\mu\, n^\mu)} +$$
$$\frac{n_\delta\, K_{\beta\gamma\,|\,\alpha}}{(n_\mu\, n^\mu)} - \frac{n_\gamma\, K_{\beta\delta\,|\,\alpha}}{(n_\mu\, n^\mu)} - \frac{n_\delta\, K_{\alpha\gamma\,|\,\beta}}{(n_\mu\, n^\mu)} + \frac{n_\gamma\, K_{\alpha\delta\,|\,\beta}}{(n_\mu\, n^\mu)} + \frac{n_\beta\, K_{\alpha\delta\,|\,\gamma}}{(n_\mu\, n^\mu)} - \frac{n_\alpha\, K_{\beta\delta\,|\,\gamma}}{(n_\mu\, n^\mu)} +$$
$$\in \left(\frac{n_\beta\, n_\delta\, a_{\alpha\,|\,\gamma}}{(n_\mu\, n^\mu)} - \frac{n_\alpha\, n_\delta\, a_{\beta\,|\,\gamma}}{(n_\mu\, n^\mu)} - \frac{n_\beta\, n_\gamma\, a_{\alpha\,|\,\delta}}{(n_\mu\, n^\mu)} + \frac{n_\alpha\, n_\gamma\, a_{\beta\,|\,\delta}}{(n_\mu\, n^\mu)} \right) - \frac{n_\beta\, K_{\alpha\gamma\,|\,\delta}}{(n_\mu\, n^\mu)} + \frac{n_\alpha\, K_{\beta\gamma\,|\,\delta}}{(n_\mu\, n^\mu)}$$

The Gauss equation

Gauss equation arises from projecting space-time Riemann tensor through all it's free indices. We store it twice, once without and once without the explicit norm display

```
In[•]:= ProjectWith[h][Deco3DRiemannFull]
        % /. $Rules
        GaussEquation = %;
```

$$\text{Out[•]}= h_\alpha{}^\iota\ h_\beta{}^\kappa\ h_\gamma{}^\lambda\ h_\delta{}^\mu\ R_{\iota\kappa\lambda\mu} == {}^{(3)}R_{\delta\gamma\beta\alpha} + \frac{K_{\gamma\beta}\ K_{\delta\alpha}}{(n_\mu\ n^\mu)} - \frac{K_{\gamma\alpha}\ K_{\delta\beta}}{(n_\mu\ n^\mu)}$$

$$\text{Out[•]}= h_\alpha{}^\iota\ h_\beta{}^\kappa\ h_\gamma{}^\lambda\ h_\delta{}^\mu\ R_{\iota\kappa\lambda\mu} == \in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}$$

```
In[•]:= AddEquation@GaussEquation;
        AddToList[$EquationsRiemann3DDeco, GaussEquation];
```

The Codazzi equation

Again, from the full decomposition of Riemann we need to get the Codazzi equation, this time through contracting the normal once with the second Riemann index.

```
In[•]:= Deco3DRiemannFull // ScreenDollarIndices;
        PrepCodazzi = Times[%, n[α]] // ToCanonical
```

$$\text{Out[•]}= -n^\alpha\ R_{\beta\alpha\gamma\delta} == -\frac{K_\beta{}^\iota\ K_{\delta\iota}\ n_\alpha\ n^\alpha\ n_\gamma}{(n_\alpha\ n^\alpha)^2} + \frac{K_\beta{}^\iota\ K_{\gamma\iota}\ n_\alpha\ n^\alpha\ n_\delta}{(n_\alpha\ n^\alpha)^2} + \frac{n_\alpha\ n^\alpha\ n_\delta\ P_h[n^\iota\ K_{\beta\gamma;\iota}]}{(n_\alpha\ n^\alpha)^2} -$$
$$\frac{n_\alpha\ n^\alpha\ n_\gamma\ P_h[n^\iota\ K_{\beta\delta;\iota}]}{(n_\alpha\ n^\alpha)^2} - \frac{a_\beta\ a_\delta\ n_\alpha\ n^\alpha\ n_\gamma}{(n_\alpha\ n^\alpha)} + \frac{a_\beta\ a_\gamma\ n_\alpha\ n^\alpha\ n_\delta}{(n_\alpha\ n^\alpha)} -$$
$$\frac{\in n_\alpha\ n^\alpha\ n_\delta\ a_{\beta|\gamma}}{(n_\alpha\ n^\alpha)} - \frac{n_\alpha\ n^\alpha\ K_{\beta\delta|\gamma}}{(n_\alpha\ n^\alpha)} + \frac{\in n_\alpha\ n^\alpha\ n_\gamma\ a_{\beta|\delta}}{(n_\alpha\ n^\alpha)} + \frac{n_\alpha\ n^\alpha\ K_{\beta\gamma|\delta}}{(n_\alpha\ n^\alpha)}$$

```
In[•]:= ProjectWith[h][PrepCodazzi]
        PutScalar /@ %
        ToCanonical /@ %
        CodazziEquation = %;
```

$$\text{Out[•]}= -h_\beta{}^\iota\ h_\gamma{}^\kappa\ h_\delta{}^\lambda\ n^\alpha\ R_{\iota\alpha\kappa\lambda} == -\frac{n_\alpha\ n^\alpha\ K_{\beta\delta|\gamma}}{(n_\alpha\ n^\alpha)} + \frac{n_\alpha\ n^\alpha\ K_{\beta\gamma|\delta}}{(n_\alpha\ n^\alpha)}$$

$$\text{Out[•]}= -h_\beta{}^\iota\ h_\gamma{}^\kappa\ h_\delta{}^\lambda\ n^\alpha\ R_{\iota\alpha\kappa\lambda} == -K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}$$

$$\text{Out[•]}= h_\beta{}^\iota\ h_\gamma{}^\kappa\ h_\delta{}^\lambda\ n^\alpha\ R_{\alpha\iota\kappa\lambda} == -K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}$$

```
In[•]:= Clear[PrepCodazzi];
```

```
In[•]:= AddEquation@CodazziEquation;
        AddToList[$EquationsRiemann3DDeco, CodazziEquation];
```

The Ricci Equation

Same as with Codazzi equation, albeit now we have to contract with normal twice, in first and third index.

We wish to store three versions of the Ricci Equation.

- The “simple” result of projections, orthogonality with the normal field & symmetries of Riemann
- Second involving the spacetime gradient of the extrinsic curvature along the normal field
- Third involving 4D-Lie derivative of the curvature along the normal

First, the “trivial” observation:

```
In[•]:= ProjectWith[h][RiemannCD[-α, -β, -γ, -δ] × n[α] × n[γ]] ==
          (ProjectWith[h][RiemannCD[-α, -β, -γ, -δ] × n[α] × n[γ]] /. ApplyRule@hTOg)
        % // Expand
        ToCanonical /@ %
        RicciEquation1 = %;
```

$$\text{Out[•]= } h_\beta{}^\lambda\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\lambda\gamma\kappa} == n^\alpha\ n^\gamma\ \left(\delta_\beta{}^\lambda - \epsilon\ n_\beta\ n^\lambda\right)\left(\delta_\delta{}^\kappa - \epsilon\ n_\delta\ n^\kappa\right) R_{\alpha\lambda\gamma\kappa}$$

$$\text{Out[•]= } h_\beta{}^\lambda\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\lambda\gamma\kappa} == n^\alpha\ n^\gamma\ R_{\alpha\beta\gamma\delta} - \epsilon\ n^\alpha\ n^\gamma\ n_\delta\ n^\lambda\ R_{\alpha\beta\gamma\lambda} - \epsilon\ n^\alpha\ n_\beta\ n^\gamma\ n^\lambda\ R_{\alpha\lambda\gamma\delta} + n^\alpha\ n_\beta\ n^\gamma\ n_\delta\ n^\lambda\ n^\kappa\ R_{\alpha\lambda\gamma\kappa}$$

$$\text{Out[•]= } h_\beta{}^\lambda\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\lambda\gamma\kappa} == n^\alpha\ n^\gamma\ R_{\beta\alpha\delta\gamma}$$

```
In[•]:= AddEquation@RicciEquation1;
        AddToList[$EquationsRiemann3DDeco, RicciEquation1];
```

For other two expressions, we have to use the full decomposition:

```
In[•]:= Deco3DRiemannFull // ScreenDollarIndices;
        Times[n[α] × n[γ], %];
        % // Expand
        RicciPrep = %;
```

$$\text{Out[•]= } n^\alpha\ n^\gamma\ R_{\alpha\beta\gamma\delta} == -\frac{K_\beta{}^\lambda\ K_{\delta\lambda}\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{\left(n_\mu\ n^\mu\right)^2} -$$

$$\frac{n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ \underset{h}{P}\left[n^\lambda\ K_{\beta\delta;\lambda}\right]}{\left(n_\mu\ n^\mu\right)^2} - \frac{a_\beta\ a_\delta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{\left(n_\mu\ n^\mu\right)} + \frac{\epsilon\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ a_{\beta|\delta}}{\left(n_\mu\ n^\mu\right)}$$

```
In[•]:= ProjectWith[h][RicciPrep]
        % /. Projectorh → ProjectWith[h];
        PutScalar /@ %;
        ToCanonical /@ %;
        % /. $Rules
        RicciEquation2 = %;
```

$$\text{Out[•]= } h_\beta{}^\lambda\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\lambda\gamma\kappa} == -\frac{K_{\lambda\delta}\ K^\lambda{}_\beta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{\left(n_\mu\ n^\mu\right)^2} -$$

$$\frac{h_\beta{}^\kappa\ h_\delta{}^\lambda\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ \underset{h}{P}\left[n^\lambda\ K_{\kappa\lambda;\lambda}\right]}{\left(n_\mu\ n^\mu\right)^2} - \frac{a_\beta\ a_\delta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{\left(n_\mu\ n^\mu\right)} + \frac{\epsilon\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ a_{\beta|\delta}}{\left(n_\mu\ n^\mu\right)}$$

$$\text{Out[•]= } h_\beta{}^\lambda\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\lambda\gamma\kappa} == -\epsilon\ a_\beta\ a_\delta - K_\beta{}^\lambda\ K_{\delta\lambda} + a_{\beta|\delta} - h_\beta{}^\alpha\ h_\delta{}^\gamma\ n^\lambda\ K_{\alpha\gamma;\lambda}$$

```
In[•]:= AddEquation@RicciEquation2;
        AddToList[$EquationsRiemann3DDeco, RicciEquation2];
```

We have to first learn how to decompose the Lie derivative along the normal of the extrinsic curvature tensor. We see, that it is, indeed, purely spatial:

In[•]:=
```
LieD[n[σ]]@ExtrinsicKh[-α, -β]
OrthogonalToVectorQ[n][%]
```

Out[•]= $\mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\alpha\beta}$

Out[•]= True

We first wish to know what is the result of $\mathsf{h}^{\alpha}{}_{\mu}\ \mathsf{h}^{\beta}{}_{\nu}\ \left(\mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\alpha\beta}\right)$

In[•]:=
```
h[α, -μ] × h[β, -ν] × LieD[n[σ]]@ExtrinsicKh[-α, -β]
% /. ApplyRule@hTOg
% // Expand
```

Out[•]= $\mathsf{h}^{\alpha}{}_{\mu}\ \mathsf{h}^{\beta}{}_{\nu}\ \left(\mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\alpha\beta}\right)$

Out[•]= $\left(\delta_{\mu}{}^{\alpha} - \epsilon\ \mathsf{n}^{\alpha}\ \mathsf{n}_{\mu}\right)\ \left(\delta_{\nu}{}^{\beta} - \epsilon\ \mathsf{n}^{\beta}\ \mathsf{n}_{\nu}\right)\ \left(\mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\alpha\beta}\right)$

Out[•]= $\mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\mu\nu}$

So this particular combination of projectors and derivative results in lowering the indices inside the derivative operator. That is however not true generally. Yet, to NOT undergo this process again, we prepare a rule for this (and only this) particular term. This rule will result in a warning, which we will ignore **for this term**, as the step is verified.

In[•]:=
```
h[α, -μ] × h[β, -ν] × LieD[n[σ]]@ExtrinsicKh[-α, -β] == LieD[n[σ]]@ExtrinsicKh[-μ, -ν]
ContrLieDextrinsicK = %;
AddEquation@ContrLieDextrinsicK;
```

Out[•]= $\mathsf{h}^{\alpha}{}_{\mu}\ \mathsf{h}^{\beta}{}_{\nu}\ \left(\mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\alpha\beta}\right) == \mathcal{L}_{\mathsf{n}}\, \mathsf{K}_{\mu\nu}$

```
In[•]:= ProjectWith[h][RicciPrep]
        % /. ApplyRule@NCDExtrinsicK
        % /. Projectorh → ProjectWith[h];
        PutScalar /@ %;
        ToCanonical /@ %;
        % /. $Rules
        % /. ApplyRule@ContrLieDextrinsicK (*we ignore the warning,
        as we confirmed the correctness of this step above*)
        RicciEquation3 = %;
```

$$\text{Out[•]= } h_\beta{}^\iota\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\iota\gamma\kappa} == -\frac{K_{\iota\delta}\ K^\iota{}_\beta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{(n_\mu\ n^\mu)^2} -$$

$$\frac{h_\beta{}^\kappa\ h_\delta{}^\lambda\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ \overset{P}{h}\left[n^\iota\ K_{\kappa\lambda;\iota}\right]}{(n_\mu\ n^\mu)^2} - \frac{a_\beta\ a_\delta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{(n_\mu\ n^\mu)} + \frac{\epsilon\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ a_{\beta|\delta}}{(n_\mu\ n^\mu)}$$

$$\text{Out[•]= } h_\beta{}^\iota\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\iota\gamma\kappa} == -\frac{K_{\iota\delta}\ K^\iota{}_\beta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{(n_\mu\ n^\mu)^2} - \frac{a_\beta\ a_\delta\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma}{(n_\mu\ n^\mu)} +$$

$$\frac{\epsilon\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ a_{\beta|\delta}}{(n_\mu\ n^\mu)} - \frac{h_\beta{}^\iota\ h_\delta{}^\kappa\ n_\alpha\ n^\alpha\ n_\gamma\ n^\gamma\ \left(-2\ \overset{P}{h}\left[K_\iota{}^\lambda\ K_{\kappa\lambda}\right] + \mathcal{L}_n\ K_{\iota\kappa}\right)}{(n_\mu\ n^\mu)^2}$$

$$\text{Out[•]= } h_\beta{}^\iota\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\iota\gamma\kappa} == -\epsilon\ a_\beta\ a_\delta + K_\beta{}^\iota\ K_{\delta\iota} + a_{\beta|\delta} - h_\beta{}^\iota\ h_\delta{}^\alpha\ \left(\mathcal{L}_n\ K_{\iota\alpha}\right)$$

```
** MakeRule: Potential problems moving indices on the LHS.
```

$$\text{Out[•]= } h_\beta{}^\iota\ h_\delta{}^\kappa\ n^\alpha\ n^\gamma\ R_{\alpha\iota\gamma\kappa} == -\epsilon\ a_\beta\ a_\delta + K_\beta{}^\iota\ K_{\delta\iota} + a_{\beta|\delta} - \mathcal{L}_n\ K_{\beta\delta}$$

```
In[•]:= Clear[RicciPrep];

In[•]:= AddEquation@RicciEquation3;
        AddToList[$EquationsRiemann3DDeco, RicciEquation3];
```

3+1 Ricci tensor decomposition

```
In[•]:= $EquationsRicci3DDeco = {};
```

In this chapter the exact same approach as for 3+1 decomposition of the Riemann tensor is followed, this time for the Ricci curvature tensor.

```
In[•]:= RiemannCD[α, -β, -α, -δ] == GaussCodazzi[RiemannCD[α, -β, -α, -δ], h]
        Deco3DRicciFull = %;
```

$$\text{Out[•]}= R_{\beta\delta} == {}^{(3)}R_{\beta\delta} - \frac{K_\beta{}^\gamma\, K_{\delta\gamma}\, n_\alpha\, n^\alpha}{(n_\mu\, n^\mu)^2} - \frac{K_\alpha{}^\gamma\, K^\alpha{}_\gamma\, n_\beta\, n_\delta}{(n_\mu\, n^\mu)^2} - \frac{n_\alpha\, n^\alpha\, \overset{P}{h}\left[n^\gamma\, K_{\beta\delta;\gamma}\right]}{(n_\mu\, n^\mu)^2} +$$
$$\frac{K_{\alpha\delta}\, K_\beta{}^\alpha}{(n_\mu\, n^\mu)} - \frac{K_\alpha{}^\alpha\, K_{\beta\delta}}{(n_\mu\, n^\mu)} - \frac{a_\beta\, a_\delta\, n_\alpha\, n^\alpha}{(n_\mu\, n^\mu)} - \frac{a_\alpha\, a^\alpha\, n_\beta\, n_\delta}{(n_\mu\, n^\mu)} - \frac{n_\beta\, n_\delta\, \left(n^\beta\, K_\alpha{}^\alpha{}_{;\beta}\right)}{(n_\mu\, n^\mu)^2} +$$
$$\frac{n_\delta\, K_\beta{}^\alpha{}_{|\alpha}}{(n_\mu\, n^\mu)} + \frac{n_\beta\, K_{\alpha\delta}{}^{|\alpha}}{(n_\mu\, n^\mu)} - \frac{n_\delta\, K_\alpha{}^\alpha{}_{|\beta}}{(n_\mu\, n^\mu)} + \epsilon \left(\frac{n_\beta\, n_\delta\, a_\alpha{}^{|\alpha}}{(n_\mu\, n^\mu)} + \frac{n_\alpha\, n^\alpha\, a_{\beta|\delta}}{(n_\mu\, n^\mu)}\right) - \frac{n_\beta\, K_\alpha{}^\alpha{}_{|\delta}}{(n_\mu\, n^\mu)}$$

Gauss equation variation for the Ricci Tensor

```
In[•]:= Deco3DRicciFull // ScreenDollarIndices;
        ProjectWith[h][%]
        % /. Projectorh → ProjectWith[h]
        NoScalar /@ %;
        PutScalar /@ %;
        % /. $Rules;
        ToCanonical /@ %;
        NoScalar /@ %;
        % * (-ϵ) // Expand
        RicciGaussEquation1 = %;
```

$$\text{Out[•]}= h_\beta{}^\alpha\, h_\delta{}^\gamma\, R_{\alpha\gamma} == {}^{(3)}R_{\delta\beta} - \frac{K_{\gamma\delta}\, K^\gamma{}_\beta\, n_\alpha\, n^\alpha}{(n_\mu\, n^\mu)^2} -$$
$$\frac{h_\beta{}^\lambda\, h_\delta{}^\kappa\, n_\alpha\, n^\alpha\, \overset{P}{h}\left[n^\gamma\, K_{\lambda\kappa;\gamma}\right]}{(n_\mu\, n^\mu)^2} + \frac{K_{\alpha\delta}\, K^\alpha{}_\beta}{(n_\mu\, n^\mu)} - \frac{K_\alpha{}^\alpha\, K_{\delta\beta}}{(n_\mu\, n^\mu)} - \frac{a_\beta\, a_\delta\, n_\alpha\, n^\alpha}{(n_\mu\, n^\mu)} + \frac{\epsilon\, n_\alpha\, n^\alpha\, a_{\beta|\delta}}{(n_\mu\, n^\mu)}$$

$$\text{Out[•]}= h_\beta{}^\alpha\, h_\delta{}^\gamma\, R_{\alpha\gamma} == {}^{(3)}R_{\delta\beta} - \frac{K_{\gamma\delta}\, K^\gamma{}_\beta\, n_\alpha\, n^\alpha}{(n_\mu\, n^\mu)^2} + \frac{K_{\alpha\delta}\, K^\alpha{}_\beta}{(n_\mu\, n^\mu)} -$$
$$\frac{K_\alpha{}^\alpha\, K_{\delta\beta}}{(n_\mu\, n^\mu)} - \frac{a_\beta\, a_\delta\, n_\alpha\, n^\alpha}{(n_\mu\, n^\mu)} + \frac{\epsilon\, n_\alpha\, n^\alpha\, a_{\beta|\delta}}{(n_\mu\, n^\mu)} - \frac{h_\beta{}^\lambda\, h_\delta{}^\kappa\, n_\alpha\, n^\alpha\, n^\gamma\, K_{\lambda\kappa;\gamma}}{(n_\mu\, n^\mu)^2}$$

$$\text{Out[•]}= -\epsilon\, h_\beta{}^\alpha\, h_\delta{}^\gamma\, R_{\alpha\gamma} == \epsilon\, a_\beta\, a_\delta + K^\alpha{}_\alpha\, K_{\beta\delta} - \epsilon\, {}^{(3)}R_{\beta\delta} - a_{\beta|\delta} + h_\beta{}^\gamma\, h_\delta{}^\lambda\, n^\alpha\, K_{\gamma\lambda;\alpha}$$

```
In[•]:= AddEquation@RicciGaussEquation1;
        AddToList[$EquationsRicci3DDeco, RicciGaussEquation1];
```

Now we wish to use

```
In[•]:= RicciGauss2Prep = RicciEquation2 * (-1) // ScreenDollarIndices
```

$$\text{Out[•]}= -h_\beta{}^\lambda\, h_\delta{}^\kappa\, n^\alpha\, n^\gamma\, R_{\alpha\lambda\gamma\kappa} == \epsilon\, a_\beta\, a_\delta + K_\beta{}^\lambda\, K_{\delta\lambda} - a_{\beta|\delta} + h_\beta{}^\alpha\, h_\delta{}^\gamma\, n^\lambda\, K_{\alpha\gamma;\lambda}$$

Treating the LHS first:

```
In[•]:= RicciGauss2Prep[[1]] /. ApplyRule@hTOg
        % // ToCanonical
        % == RicciGauss2Prep[[2]];
        RicciPrep = % // ScreenDollarIndices
```

Out[•]= $-n^{\alpha}\, n^{\gamma}\, \left(\delta_{\beta}{}^{\iota} - \epsilon\, n_{\beta}\, n^{\iota}\right) \left(\delta_{\delta}{}^{\kappa} - \epsilon\, n_{\delta}\, n^{\kappa}\right) R_{\alpha\iota\gamma\kappa}$

Out[•]= $-n^{\alpha}\, n^{\gamma}\, R_{\beta\alpha\delta\gamma}$

Out[•]= $-n^{\alpha}\, n^{\gamma}\, R_{\beta\alpha\delta\gamma} == \epsilon\, a_{\beta}\, a_{\delta} + K_{\beta}{}^{\iota}\, K_{\delta\iota} - a_{\beta|\delta} + h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, n^{\iota}\, K_{\alpha\gamma;\iota}$

Now solving for the $h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, n^{\prime}\, K_{\alpha\gamma;\prime}$

```
In[•]:= Position[RicciGauss2Prep, h[-β, α] × h[-δ, γ] × n[ι] × CD[-ι]@ExtrinsicKh[-α, -γ]];
        MapAt[Hold, RicciGauss2Prep, %]
        Solve[%, Hold[h[-β, α] × h[-δ, γ] × n[ι] × CD[-ι]@ExtrinsicKh[-α, -γ]]] // Flatten
        %[[1]] /. Rule → Equal
        % // ReleaseHold
        NCDExtrinsicKDeco = %;
```

Out[•]= $-h_{\beta}{}^{\iota}\, h_{\delta}{}^{\kappa}\, n^{\alpha}\, n^{\gamma}\, R_{\alpha\iota\gamma\kappa} == \epsilon\, a_{\beta}\, a_{\delta} + K_{\beta}{}^{\iota}\, K_{\delta\iota} + \text{Hold}\left[h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, n^{\iota}\, K_{\alpha\gamma;\iota}\right] - a_{\beta|\delta}$

Out[•]= $\left\{\text{Hold}\left[h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, n^{\iota}\, K_{\alpha\gamma;\iota}\right] \to -\epsilon\, a_{\beta}\, a_{\delta} - K_{\beta}{}^{\iota}\, K_{\delta\iota} - h_{\beta}{}^{\iota}\, h_{\delta}{}^{\kappa}\, n^{\alpha}\, n^{\gamma}\, R_{\alpha\iota\gamma\kappa} + a_{\beta|\delta}\right\}$

Out[•]= $\text{Hold}\left[h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, n^{\iota}\, K_{\alpha\gamma;\iota}\right] == -\epsilon\, a_{\beta}\, a_{\delta} - K_{\beta}{}^{\iota}\, K_{\delta\iota} - h_{\beta}{}^{\iota}\, h_{\delta}{}^{\kappa}\, n^{\alpha}\, n^{\gamma}\, R_{\alpha\iota\gamma\kappa} + a_{\beta|\delta}$

Out[•]= $h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, n^{\iota}\, K_{\alpha\gamma;\iota} == -\epsilon\, a_{\beta}\, a_{\delta} - K_{\beta}{}^{\iota}\, K_{\delta\iota} - h_{\beta}{}^{\iota}\, h_{\delta}{}^{\kappa}\, n^{\alpha}\, n^{\gamma}\, R_{\alpha\iota\gamma\kappa} + a_{\beta|\delta}$

```
In[•]:= AddEquation@NCDExtrinsicKDeco;
        AddToList[$Equations3DExtCurv, NCDExtrinsicKDeco];
```

```
In[•]:= RicciGaussEquation1 /. ApplyRule@NCDExtrinsicKDeco
        RicciGaussEquation2 = %;
```

Out[•]= $-\epsilon\, h_{\beta}{}^{\alpha}\, h_{\delta}{}^{\gamma}\, R_{\alpha\gamma} == K^{\alpha}{}_{\alpha}\, K_{\beta\delta} - K_{\beta}{}^{\alpha}\, K_{\delta\alpha} - \epsilon\, {}^{(3)}R_{\beta\delta} - h_{\beta}{}^{\iota}\, h_{\delta}{}^{\kappa}\, n^{\alpha}\, n^{\gamma}\, R_{\alpha\iota\gamma\kappa}$

```
In[•]:= Clear[RicciGauss2Prep]
```

```
In[•]:= AddEquation@RicciGaussEquation2;
        AddToList[$EquationsRicci3DDeco, RicciGaussEquation2];
```

Codazzi equation variation for Ricci tensor

Again accounting for symmetries, the Codazzi equation for the Ricci tensor arises through contracting the Ricci tensor once with the normal, then projecting it with the induced metric:

```
In[•]:= Deco3DRicciFull // ScreenDollarIndices;
        Times[%, n[β]] // Expand
        PutScalar /@ %;
        % /. $Rules
        NoScalar /@ %
        RicciCodPrep = %;
```

$$\text{Out}[\bullet]= n^{\beta}\ R_{\beta\delta} == -\frac{K_{\alpha}{}^{\gamma}\ K^{\alpha}{}_{\gamma}\ n_{\beta}\ n^{\beta}\ n_{\delta}}{\left(n_{\mu}\ n^{\mu}\right)^{2}} - \frac{a_{\alpha}\ a^{\alpha}\ n_{\beta}\ n^{\beta}\ n_{\delta}}{\left(n_{\mu}\ n^{\mu}\right)} -$$

$$\frac{n_{\beta}\ n^{\beta}\ n_{\delta}\ \left(n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}\right)}{\left(n_{\mu}\ n^{\mu}\right)^{2}} + \frac{\in n_{\beta}\ n^{\beta}\ n_{\delta}\ a_{\alpha}{}^{|\alpha}}{\left(n_{\mu}\ n^{\mu}\right)} + \frac{n_{\beta}\ n^{\beta}\ K_{\alpha\delta}{}^{|\alpha}}{\left(n_{\mu}\ n^{\mu}\right)} - \frac{n_{\beta}\ n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{|\delta}}{\left(n_{\mu}\ n^{\mu}\right)}$$

$$\text{Out}[\bullet]= n^{\beta}\ R_{\beta\delta} == -n_{\delta}\ \left(a_{\alpha}\ a^{\alpha}\right) - \in n_{\delta}\ \left(K_{\alpha}{}^{\gamma}\ K^{\alpha}{}_{\gamma}\right) + \in n_{\delta}\ \left(a_{\alpha}{}^{|\alpha}\right) - \in n_{\delta}\ \left(n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}\right) + K_{\alpha\delta}{}^{|\alpha} - K_{\alpha}{}^{\alpha}{}_{|\delta}$$

$$\text{Out}[\bullet]= n^{\beta}\ R_{\beta\delta} == -a_{\alpha}\ a^{\alpha}\ n_{\delta} - \in K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta}\ n_{\delta} + \in n_{\delta}\ a_{\alpha}{}^{|\alpha} + K_{\alpha\delta}{}^{|\alpha} - K_{\alpha}{}^{\alpha}{}_{|\delta} - \in n^{\beta}\ n_{\delta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}$$

```
In[•]:= ProjectWith[h][RicciCodPrep]
        ToCanonical /@ %
        RicciCodazziEquation = %;
```

$$\text{Out}[\bullet]= h_{\delta}{}^{\alpha}\ n^{\beta}\ R_{\beta\alpha} == K_{\alpha\delta}{}^{|\alpha} - K_{\alpha}{}^{\alpha}{}_{|\delta}$$

$$\text{Out}[\bullet]= h_{\delta}{}^{\alpha}\ n^{\beta}\ R_{\beta\alpha} == K_{\delta}{}^{\alpha}{}_{|\alpha} - K^{\alpha}{}_{\alpha|\delta}$$

```
In[•]:= Clear[RicciCodPrep];
```

```
In[•]:= AddEquation@RicciCodazziEquation;
        AddToList[$EquationsRicci3DDeco, RicciCodazziEquation];
```

Ricci equation variation for Ricci tensor

The only remaining possibility is to contract twice with the normal:

```
In[•]:= Deco3DRicciFull // ScreenDollarIndices;
        Times[%, n[β] × n[δ]];
        % // Expand;
        PutScalar /@ %;
        % /. $Rules;
        NoScalar /@ %
        RicciRicciEquation = %;
```

$$\text{Out}[\bullet]= n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\in a_{\alpha}\ a^{\alpha} - K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + a_{\alpha}{}^{|\alpha} - n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}$$

```
In[•]:= AddEquation@RicciRicciEquation;
        AddToList[$EquationsRicci3DDeco, RicciRicciEquation];
```

We can express this equation in three forms:
First of them is by making a slight adjustment to the last term (converting the gradient of Extrinsic curvature scalar to the normal field):

```
In[•]:= n[β] × CD[-β]@ExtrinsicKh[-α, α] ==
         n[β] × CD[-β]@((h[α, γ] /. ApplyRule@hTOg) ExtrinsicKh[-α, -γ])
        % // Expand // ScreenDollarIndices
        % /. ApplyRule@gTOh
        % // Expand
        NCDScalExtrKTONCDExtrKContr = %;
```

$$\text{Out[•]}= n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} == n^{\beta}\ \left(\left(g^{\gamma\alpha} - \in n^{\alpha}\ n^{\gamma}\right)\ K_{\alpha\gamma;\beta} - \in K_{\alpha\gamma}\ \left(n^{\gamma}\ n^{\alpha}{}_{;\beta} + n^{\alpha}\ n^{\gamma}{}_{;\beta}\right)\right)$$

$$\text{Out[•]}= n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} == g^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta} - \in n^{\alpha}\ n^{\beta}\ n^{\gamma}\ K_{\alpha\gamma;\beta}$$

$$\text{Out[•]}= n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} == -\in n^{\alpha}\ n^{\beta}\ n^{\gamma}\ K_{\alpha\gamma;\beta} + n^{\beta}\ \left(h^{\gamma\alpha} + \in n^{\alpha}\ n^{\gamma}\right)\ K_{\alpha\gamma;\beta}$$

$$\text{Out[•]}= n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} == h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta}$$

```
In[•]:= AddEquation@NCDScalExtrKTONCDExtrKContr;
        AddToList[$Equations3DExtCurv, NCDScalExtrKTONCDExtrKContr];
```

It will be useful to also store the inverse of this relation:

```
In[•]:= NCDScalExtrKTONCDExtrKContr
        %[[2]] == %[[1]]
        NCDExtrKContrTONCDScalExtrK = %;
```

$$\text{Out[•]}= n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} == h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta}$$

$$\text{Out[•]}= h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta} == n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}$$

```
In[•]:= AddEquation@NCDExtrKContrTONCDScalExtrK;
        AddToList[$Equations3DExtCurv, NCDExtrKContrTONCDScalExtrK];
```

Using the obtained information on **RicciRicciEquation** gives us the first desired form:

```
In[•]:= RicciRicciEquation // ScreenDollarIndices
        % /. ApplyRule@NCDScalExtrKTONCDExtrKContr
        RicciRicciEquation1 = %;
```

$$\text{Out[•]}= n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\in a_{\alpha}\ a^{\alpha} - K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + a_{\alpha}{}^{|\alpha} - n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}$$

$$\text{Out[•]}= n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\in a_{\alpha}\ a^{\alpha} - K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + a_{\alpha}{}^{|\alpha} - h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta}$$

```
In[•]:= AddEquation@RicciRicciEquation1;
        AddToList[$EquationsRicci3DDeco, RicciRicciEquation1];
```

Second version of the Ricci equation for Ricci tensor is expressed in terms of the Lie derivative

```
In[•]:= RicciRicciEquation /. ApplyRule@NCDExtrinsicK;
        % /. ApplyRule@gTOh
        % // Expand
        RicciRicciEquation2 = %;
```

$$\text{Out[•]}= n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\in a_{\alpha}\ a^{\alpha} + K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + a_{\alpha}{}^{|\alpha} - \left(h^{\alpha\beta} + \in n^{\alpha}\ n^{\beta}\right)\ \left(\mathcal{L}_{n}\ K_{\alpha\beta}\right)$$

$$\text{Out[•]}= n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\in a_{\alpha}\ a^{\alpha} + K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + a_{\alpha}{}^{|\alpha} - h^{\alpha\beta}\ \left(\mathcal{L}_{n}\ K_{\alpha\beta}\right)$$

```
In[•]:= AddEquation@RicciRicciEquation2;
        AddToList[$EquationsRicci3DDeco, RicciRicciEquation2];
```

Third version uses induced partial divergence equations for $V^\mu = a^\mu$. This substitution will be used often, it is therefore stored separately.

```
In[•]:= cd3DDivTOAccCD
        % /. V → Accelerationn
        cd3DDivAcc = %;
```

Out[•]= $V^{\lambda}{}_{|\lambda} == \in a^{\beta} V_{\beta} + V^{\alpha}{}_{;\alpha}$

Out[•]= $a^{\lambda}{}_{|\lambda} == \in a_{\beta} a^{\beta} + a^{\alpha}{}_{;\alpha}$

```
In[•]:= AddEquation@cd3DDivAcc;
        AddToList[$Equations3Dderivative, cd3DDivAcc];
```

The immediate usage is to treat the RicciRicciEquation, the term $\alpha^{\lambda}{}_{|\lambda}$ gives new information when expanded in this way:

```
In[•]:= RicciRicciEquation
        % /. ApplyRule@cd3DDivAcc
        % // ScreenDollarIndices;
        % /. MakeRule[{CD[-β]@ExtrinsicKh[-α, α],
            CD[-β]@((h[α, γ]) ExtrinsicKh[-α, -γ] /. ApplyRule@hTOg)}]
        RicciRicciEquation3 = %;
```

Out[•]= $n^{\alpha} n^{\beta} R_{\alpha\beta} == -\in a_{\alpha} a^{\alpha} - K_{\alpha}{}^{\beta} K^{\alpha}{}_{\beta} + a_{\alpha}{}^{|\alpha} - n^{\beta} K_{\alpha}{}^{\alpha}{}_{;\beta}$

Out[•]= $n^{\alpha} n^{\beta} R_{\alpha\beta} == - K_{\alpha}{}^{\beta} K^{\alpha}{}_{\beta} + a^{\alpha}{}_{;\alpha} - n^{\beta} K_{\alpha}{}^{\alpha}{}_{;\beta}$

Out[•]= $n^{\alpha} n^{\beta} R_{\alpha\beta} == - K_{\alpha}{}^{\beta} K^{\alpha}{}_{\beta} + a^{\alpha}{}_{;\alpha} - n^{\beta} K_{\alpha}{}^{\alpha}{}_{;\beta}$

```
In[•]:= AddEquation@RicciRicciEquation3;
        AddToList[$EquationsRicci3DDeco, RicciRicciEquation3];
```

3+1 Ricci scalar decomposition

```
In[•]:= $EquationsRicciScalar3DDeco = {};
```

Iterating the same approach we arrive at the full decomposition of the Ricci Scalar

```
In[•]:= RicciCD[] == GaussCodazzi[RicciScalarCD[], h] /. $Rules /. Projectorh → ProjectWith[h]
        PutScalar /@ (-ϵ %) /. $Rules
        NoScalar /@ %;
        % // ContractMetric;
        % // ToCanonical
        RicciScalarFullDeco = %;
```

$$\text{Out[•]=}\quad R == \epsilon\, K_{\alpha}{}^{\beta}\, K_{\beta}{}^{\alpha} - \epsilon\, K_{\alpha}{}^{\alpha}\, K_{\beta}{}^{\beta} - \epsilon\, a_{\alpha}\, a^{\alpha}\, n_{\beta}\, n^{\beta} - K_{\alpha}{}^{\gamma}\, K_{\gamma}{}^{\alpha}\, n_{\beta}\, n^{\beta} - a_{\gamma}\, a^{\gamma}\, n_{\alpha}\, n^{\alpha}\, n_{\beta}\, n^{\beta} - \epsilon\, K_{\gamma}{}^{\delta}\, K^{\gamma}{}_{\delta}\, n_{\alpha}\, n^{\alpha}\, n_{\beta}\, n^{\beta} + {}^{(3)}R - \epsilon\, n_{\alpha}\, n^{\alpha}\, n_{\beta}\, n^{\beta} \left(n^{\beta}\, K_{\alpha}{}^{\alpha}{}_{;\beta}\right) + h^{\alpha\beta}\, n_{\gamma}\, n^{\gamma}\, a_{\alpha|\beta} + \epsilon\, n_{\alpha}\, n^{\alpha}\, n_{\beta}\, n^{\beta}\, a_{\gamma}{}^{|\gamma} - h^{\delta\gamma}\, n_{\alpha}\, n^{\alpha}\, n^{\beta}\, K_{\gamma\delta;\beta}$$

$$\text{Out[•]=}\quad -\epsilon\, R == -\epsilon\, {}^{(3)}R + 2\,\epsilon \left(a_{\alpha}\, a^{\alpha}\right) + \left(K_{\alpha}{}^{\alpha}\right)^{2} + \left(K_{\alpha}{}^{\beta}\, K^{\alpha}{}_{\beta}\right) - \left(a_{\alpha}{}^{|\alpha}\right) - \left(h^{\alpha\beta}\, a_{\alpha|\beta}\right) + \left(h^{\gamma\beta}\, n^{\alpha}\, K_{\beta\gamma;\alpha}\right) + \left(n^{\beta}\, K_{\alpha}{}^{\alpha}{}_{;\beta}\right)$$

$$\text{Out[•]=}\quad -\epsilon\, R == 2\,\epsilon\, a_{\alpha}\, a^{\alpha} + K_{\alpha\beta}\, K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\, K^{\beta}{}_{\beta} - \epsilon\, {}^{(3)}R - 2\, a^{\alpha}{}_{|\alpha} + h^{\beta\gamma}\, n^{\alpha}\, K_{\beta\gamma;\alpha} + n^{\alpha}\, K^{\beta}{}_{\beta;\alpha}$$

```
In[•]:= AddEquation@RicciScalarFullDeco;
        AddToList[$EquationsRicciScalar3DDeco, RicciScalarFullDeco];
```

This result, however straightforwardly it was acquired, should be further worked upon, as we already know how most of the terms relate to each other. We will use the following two equations for the adjustment:

```
In[•]:= cd3DDivAcc
        NCDExtrKContrTONCDScalExtrK
```

$$\text{Out[•]=}\quad a^{\lambda}{}_{|\lambda} == \epsilon\, a_{\beta}\, a^{\beta} + a^{\alpha}{}_{;\alpha}$$

$$\text{Out[•]=}\quad h^{\gamma\alpha}\, n^{\beta}\, K_{\alpha\gamma;\beta} == n^{\beta}\, K_{\alpha}{}^{\alpha}{}_{;\beta}$$

```
In[•]:= RicciScalarFullDeco
        % /. ApplyRule@cd3DDivAcc
        % /. ApplyRule@NCDExtrKContrTONCDScalExtrK
        % // Expand // ToCanonical
        RicciScalar3DDecoEquation1 = %;
```

$$\text{Out[•]=}\quad -\epsilon\, R == 2\,\epsilon\, a_{\alpha}\, a^{\alpha} + K_{\alpha\beta}\, K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\, K^{\beta}{}_{\beta} - \epsilon\, {}^{(3)}R - 2\, a^{\alpha}{}_{|\alpha} + h^{\beta\gamma}\, n^{\alpha}\, K_{\beta\gamma;\alpha} + n^{\alpha}\, K^{\beta}{}_{\beta;\alpha}$$

$$\text{Out[•]=}\quad -\epsilon\, R == 2\,\epsilon\, a_{\alpha}\, a^{\alpha} + K_{\alpha\beta}\, K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\, K^{\beta}{}_{\beta} - \epsilon\, {}^{(3)}R - 2 \left(\epsilon\, a_{\beta}\, a^{\beta} + a^{\alpha}{}_{;\alpha}\right) + h^{\beta\gamma}\, n^{\alpha}\, K_{\beta\gamma;\alpha} + n^{\alpha}\, K^{\beta}{}_{\beta;\alpha}$$

$$\text{Out[•]=}\quad -\epsilon\, R == 2\,\epsilon\, a_{\alpha}\, a^{\alpha} + K_{\alpha\beta}\, K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\, K^{\beta}{}_{\beta} - \epsilon\, {}^{(3)}R - 2 \left(\epsilon\, a_{\beta}\, a^{\beta} + a^{\alpha}{}_{;\alpha}\right) + n^{\alpha}\, K^{\beta}{}_{\beta;\alpha} + n^{\beta}\, K_{\alpha}{}^{\alpha}{}_{;\beta}$$

$$\text{Out[•]=}\quad -\epsilon\, R == K_{\alpha\beta}\, K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\, K^{\beta}{}_{\beta} - \epsilon\, {}^{(3)}R - 2\, a^{\alpha}{}_{;\alpha} + 2\, n^{\alpha}\, K^{\beta}{}_{\beta;\alpha}$$

```
In[•]:= AddEquation@RicciScalar3DDecoEquation1;
        AddToList[$EquationsRicciScalar3DDeco, RicciScalar3DDecoEquation1];
```

We now need to deploy the information about third Ricci tensor decomposition equation

```
In[•]:= RicciRicciEquation3 // ScreenDollarIndices
        Solve[%, CD[-α]@Accelerationn[α]] // Flatten
        %[[1]] /. Rule → Equal
        CDDIVAcc3DRicciRicci3 = %;
```

$$\text{Out[•]}= n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + a^{\alpha}{}_{;\alpha} - n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}$$

$$\text{Out[•]}= \left\{ a^{\alpha}{}_{;\alpha} \to K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} + n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} \right\}$$

$$\text{Out[•]}= a^{\alpha}{}_{;\alpha} == K_{\alpha}{}^{\beta}\ K^{\alpha}{}_{\beta} + n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} + n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}$$

```
In[•]:= AddEquation@CDDIVAcc3DRicciRicci3;
        AddToList[$EquationsRicci3DDeco, CDDIVAcc3DRicciRicci3];
```

Using this result to obtain second expression for Ricci scalar decomposition:

```
In[•]:= RicciScalar3DDecoEquation1
        % /. ApplyRule@CDDIVAcc3DRicciRicci3;
        % // Expand // ToCanonical
        RicciScalar3DDecoEquation2 = %;
```

$$\text{Out[•]}= -\epsilon\, R == K_{\alpha\beta}\ K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta} - \epsilon\ {}^{(3)}R - 2\ a^{\alpha}{}_{;\alpha} + 2\ n^{\alpha}\ K^{\beta}{}_{\beta;\alpha}$$

$$\text{Out[•]}= -\epsilon\, R == -K_{\alpha\beta}\ K^{\alpha\beta} + K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta} - 2\ n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} - \epsilon\ {}^{(3)}R$$

```
In[•]:= AddEquation@RicciScalar3DDecoEquation2;
        AddToList[$EquationsRicciScalar3DDeco, RicciScalar3DDecoEquation2];
```

Vacuum equations, momentum and Hamiltonian constraints, and remarks about independent components

Together with the form of the equations constraints should be mentioned, also worth writing out is the vacuum case.

Independent components of the 3D-Riemann tensor

Following is:

- Expanding the 3D-Riemann tensor in the combination of the 3D-Ricci tensor & induced metric
- Using this knowledge to explicit (and very neat) expression of 3D-Kretschmann scalar

Stressing that 3D-Riemann tensor is fully determined by 3D-Ricci tensor, as it only has 6 independent components :

```
In[•]:= Riemanncd[-μ, -ν, -κ, -λ] ==
         RicciScalarcd[] (h[-μ, -λ] × h[-ν, -κ] - h[-μ, -κ] × h[-ν, -λ]) / 2 +
          h[-μ, -κ] × Riccicd[-ν, -λ] + h[-ν, -λ] × Riccicd[-μ, -κ] -
          h[-μ, -λ] × Riccicd[-ν, -κ] - h[-ν, -κ] × Riccicd[-μ, -λ]
        Riemann3DIndepComp = %;
```

$$\text{Out[•]}= {}^{(3)}R_{\mu\nu\kappa\lambda} == h_{\nu\lambda}\ {}^{(3)}R_{\mu\kappa} - h_{\nu\kappa}\ {}^{(3)}R_{\mu\lambda} - h_{\mu\lambda}\ {}^{(3)}R_{\nu\kappa} + h_{\mu\kappa}\ {}^{(3)}R_{\nu\lambda} + \frac{1}{2}\left(h_{\mu\lambda}\ h_{\nu\kappa} - h_{\mu\kappa}\ h_{\nu\lambda}\right)\ {}^{(3)}R$$

In further chapters the Kretschmann scalar 3+1 decomposition will be worked upon. Now it is worth realizing, that the 3D-Riemann tensor square scalar (the 3D Kretschmann scalar, in fact) is written

out in neat form as:

```
In[•]:= AddEquation@Riemann3DIndepComp;
        AddToList[$EquationsRiemann3DDeco, Riemann3DIndepComp];
```

```
In[•]:= Riemanncd[-μ, -ν, -κ, -λ] × Riemanncd[μ, ν, κ, λ] ==
         (Riemanncd[-μ, -ν, -κ, -λ] × Riemanncd[μ, ν, κ, λ] /. ApplyRule@Riemann3DIndepComp)
        % // Expand;
        % // ToCanonical
        KretschScal3DIndepComp = %;
```

$$\text{Out[•]=}\quad {}^{(3)}R_{\mu\nu\kappa\lambda}\ {}^{(3)}R^{\mu\nu\kappa\lambda} == \left(h_{\nu\lambda}\ {}^{(3)}R_{\mu\kappa} - h_{\nu\kappa}\ {}^{(3)}R_{\mu\lambda} - h_{\mu\lambda}\ {}^{(3)}R_{\nu\kappa} + h_{\mu\kappa}\ {}^{(3)}R_{\nu\lambda} + \frac{1}{2}\ h_{\mu\lambda}\ h_{\nu\kappa}\ {}^{(3)}R - \frac{1}{2}\ h_{\mu\kappa}\ h_{\nu\lambda}\ {}^{(3)}R\right)$$
$$\left(h^{\nu\lambda}\ {}^{(3)}R^{\mu\kappa} - h^{\nu\kappa}\ {}^{(3)}R^{\mu\lambda} - h^{\mu\lambda}\ {}^{(3)}R^{\nu\kappa} + h^{\mu\kappa}\ {}^{(3)}R^{\nu\lambda} + \frac{1}{2}\ h^{\mu\lambda}\ h^{\nu\kappa}\ {}^{(3)}R - \frac{1}{2}\ h^{\mu\kappa}\ h^{\nu\lambda}\ {}^{(3)}R\right)$$

$$\text{Out[•]=}\quad {}^{(3)}R_{\kappa\lambda\mu\nu}\ {}^{(3)}R^{\kappa\lambda\mu\nu} == 4\ {}^{(3)}R_{\kappa\lambda}\ {}^{(3)}R^{\kappa\lambda} - {}^{(3)}R^{2}$$

```
In[•]:= AddEquation@KretschScal3DIndepComp;
        AddToList[$EquationsRiemann3DDeco, KretschScal3DIndepComp];
```

The vacuum case

```
In[•]:= $EquationsNormalChange = {};
```

For the vacuum (obviously) the Ricci tensor & scalar vanish (unlike the Riemann tensor, which still can remain nonzero - remember, the Einstein equations determine only the Ricci tensor, they do not “limit” for the topology of the space-time - therein lies the apriori non-vanishing nature of some of the Riemann tensor components).

Following is:

• Storing the vanishing of Ricci tensor & scalar in the vacuum case

• Preparing a function to convert any equation containing the two mentioned entities into vacuum case

• Expressing the equations for change along the normal (“normal-change equations”)

```
In[•]:= RicciCD[-μ, -ν] == 0
        VacRicci = %;
        RicciScalarCD[] == 0
        VacRicciSc = %;
```

$$\text{Out[•]=}\quad R_{\mu\nu} == 0$$

$$\text{Out[•]=}\quad R == 0$$

```
In[•]:= VacuumEq[expression_] := expression /. ApplyRule@VacRicci /. ApplyRule@VacRicciSc
```

Applying this function to the RicciGauss1/2 equations give us:

```
In[•]:= RicciGaussEquation1
        VacuumEq@%
        % // ScreenDollarIndices
        Solve[%, Riccicd[-β, -δ]] // Flatten;
        %[[1]] /. Rule → Equal;
        % * (-ϵ) // Expand
        NormChangEq1 = %;
```

$$\text{Out[•]= } -\epsilon\, h_{\beta}^{\ \alpha}\ h_{\delta}^{\ \gamma}\ R_{\alpha\gamma} == \epsilon\, a_{\beta}\ a_{\delta} + K^{\alpha}_{\ \alpha}\ K_{\beta\delta} - \epsilon\, {}^{(3)}R_{\beta\delta} - a_{\beta\,|\,\delta} + h_{\beta}^{\ \gamma}\ h_{\delta}^{\ \lambda}\ n^{\alpha}\ K_{\gamma\lambda;\alpha}$$

$$\text{Out[•]= } 0 == \epsilon\, a_{\beta}\ a_{\delta} + K^{\alpha}_{\ \alpha}\ K_{\beta\delta} - \epsilon\, {}^{(3)}R_{\beta\delta} - a_{\beta\,|\,\delta} + h_{\beta}^{\ \gamma}\ h_{\delta}^{\ \lambda}\ n^{\alpha}\ K_{\gamma\lambda;\alpha}$$

$$\text{Out[•]= } 0 == \epsilon\, a_{\beta}\ a_{\delta} + K^{\alpha}_{\ \alpha}\ K_{\beta\delta} - \epsilon\, {}^{(3)}R_{\beta\delta} - a_{\beta\,|\,\delta} + h_{\beta}^{\ \gamma}\ h_{\delta}^{\ \lambda}\ n^{\alpha}\ K_{\gamma\lambda;\alpha}$$

$$\text{Out[•]= } -\epsilon\, {}^{(3)}R_{\beta\delta} == -\epsilon\, a_{\beta}\ a_{\delta} - K^{\alpha}_{\ \alpha}\ K_{\beta\delta} + a_{\beta\,|\,\delta} - h_{\beta}^{\ \gamma}\ h_{\delta}^{\ \lambda}\ n^{\alpha}\ K_{\gamma\lambda;\alpha}$$

```
In[•]:= RicciGaussEquation2
        VacuumEq@%
        % // ScreenDollarIndices;
        Solve[%, Riccicd[-β, -δ]] // Flatten;
        %[[1]] /. Rule → Equal;
        % /. ApplyRule@hTOg;
        % // ToCanonical;
        % * (-ϵ) // Expand
        NormChangEq2 = %;
```

$$\text{Out[•]= } -\epsilon\, h_{\beta}^{\ \alpha}\ h_{\delta}^{\ \gamma}\ R_{\alpha\gamma} == K^{\alpha}_{\ \alpha}\ K_{\beta\delta} - K_{\beta}^{\ \alpha}\ K_{\delta\alpha} - \epsilon\, {}^{(3)}R_{\beta\delta} - h_{\beta}^{\ \lambda}\ h_{\delta}^{\ \kappa}\ n^{\alpha}\ n^{\gamma}\ R_{\alpha\lambda\gamma\kappa}$$

$$\text{Out[•]= } 0 == K^{\alpha}_{\ \alpha}\ K_{\beta\delta} - K_{\beta}^{\ \alpha}\ K_{\delta\alpha} - \epsilon\, {}^{(3)}R_{\beta\delta} - h_{\beta}^{\ \lambda}\ h_{\delta}^{\ \kappa}\ n^{\alpha}\ n^{\gamma}\ R_{\alpha\lambda\gamma\kappa}$$

$$\text{Out[•]= } -\epsilon\, {}^{(3)}R_{\beta\delta} == -K^{\alpha}_{\ \alpha}\ K_{\beta\delta} + K_{\beta}^{\ \alpha}\ K_{\delta\alpha} + n^{\alpha}\ n^{\gamma}\ R_{\beta\alpha\delta\gamma}$$

```
In[•]:= AddEquation@NormChangEq1;
        AddEquation@NormChangEq2;
        AddToList[$EquationsNormalChange, NormChangEq1];
        AddToList[$EquationsNormalChange, NormChangEq2];
```

Momentum and Hamiltonian constraints

```
In[•]:= $EquationsConstraints = {};
```

To finalize the 3+1 decomposition we have to specify the momentum & Hamiltonian constraints, which are:

```
In[•]:= cd[-λ]@ExtrinsicKh[λ, -κ] == cd[-κ]@ExtrinsicKh[-α, α]
        MomentumConstr = %;
```

$$\text{Out[•]= } K^{\lambda}_{\ \kappa\,|\,\lambda} == K_{\alpha}^{\ \alpha}{}_{|\kappa}$$

```
In[•]:= -ϵ RicciScalarcd[] ==
          ExtrinsicKh[λ, -γ] × ExtrinsicKh[γ, -λ] - ExtrinsicKh[α, -α] × ExtrinsicKh[α, -α]
        HamiltContr = %;
```

$$\text{Out[•]= } -\epsilon\, {}^{(3)}R == -K^{\alpha}_{\ \alpha}{}^{2} + K^{\gamma}_{\ \lambda}\ K^{\lambda}_{\ \gamma}$$

```
In[•]:= AddEquation@MomentumConstr;
        AddEquation@HamiltContr;
        AddToList[$EquationsConstraints, MomentumConstr];
        AddToList[$EquationsConstraints, HamiltContr];
```

Last “insurance” has to be has to be made for ${}^{(3)}R^{\nu\lambda}\,h_{\nu\lambda} = {}^{(3)}R$

```
In[•]:= cd[-λ]@Accelerationn[λ] - ϵ Accelerationn[λ] × Accelerationn[-λ] ==
         ExtrinsicKh[λ, -γ] × ExtrinsicKh[γ, -λ] + CD[-γ]@ExtrinsicKh[α, -α] × n[γ]
        InsuranceEq1 = %;
```

$$\text{Out[•]=}\quad -\epsilon\, a_\lambda\; a^\lambda + a^\lambda{}_{|\lambda} == K^\gamma{}_\lambda\; K^\lambda{}_\gamma + n^\gamma\; K^\alpha{}_{\alpha;\gamma}$$

Or, expressing $a^\lambda{}_{|\lambda}$

```
In[•]:= InsuranceEq1
        % /. ApplyRule@cd3DDivAcc
        % // ToCanonical
        InsuranceEq2 = %;
```

$$\text{Out[•]=}\quad -\epsilon\, a_\lambda\; a^\lambda + a^\lambda{}_{|\lambda} == K^\gamma{}_\lambda\; K^\lambda{}_\gamma + n^\gamma\; K^\alpha{}_{\alpha;\gamma}$$

$$\text{Out[•]=}\quad \epsilon\, a_\alpha\; a^\alpha - \epsilon\, a_\lambda\; a^\lambda + a^\alpha{}_{;\alpha} == K^\gamma{}_\lambda\; K^\lambda{}_\gamma + n^\gamma\; K^\alpha{}_{\alpha;\gamma}$$

$$\text{Out[•]=}\quad a^\alpha{}_{;\alpha} == K_{\alpha\gamma}\; K^{\alpha\gamma} + n^\alpha\; K^\gamma{}_{\gamma;\alpha}$$

```
In[•]:= AddEquation@InsuranceEq1;
        AddEquation@InsuranceEq2;
        AddToList[$EquationsConstraints, InsuranceEq1];
        AddToList[$EquationsConstraints, InsuranceEq2];
```

Summary of the 3+1 decomposition results

In order to aesthetically summarize the results, let us now print out the obtained equations:

The Riemann tensor induced 3+1 decomposition summarizes to:

```
In[•]:= DisplayEQ@$EquationsRiemann3DDeco
```

Out[•]//MatrixForm=

$$\begin{pmatrix}
h_\alpha{}^\iota\; h_\beta{}^\kappa\; h_\gamma{}^\lambda\; h_\delta{}^\mu\; R_{\iota\kappa\lambda\mu} == \epsilon\, K_{\gamma\beta}\; K_{\delta\alpha} - \epsilon\, K_{\gamma\alpha}\; K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha} \\
h_\beta{}^\iota\; h_\gamma{}^\kappa\; h_\delta{}^\lambda\; n^\alpha\; R_{\alpha\iota\kappa\lambda} == -K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta} \\
h_\beta{}^\iota\; h_\delta{}^\kappa\; n^\alpha\; n^\gamma\; R_{\alpha\iota\gamma\kappa} == n^\alpha\; n^\gamma\; R_{\beta\alpha\delta\gamma} \\
h_\beta{}^\iota\; h_\delta{}^\kappa\; n^\alpha\; n^\gamma\; R_{\alpha\iota\gamma\kappa} == -\epsilon\, a_\beta\; a_\delta - K_\beta{}^\iota\; K_{\delta\iota} + a_{\beta|\delta} - h_\beta{}^\alpha\; h_\delta{}^\gamma\; n^\iota\; K_{\alpha\gamma;\iota} \\
h_\beta{}^\iota\; h_\delta{}^\kappa\; n^\alpha\; n^\gamma\; R_{\alpha\iota\gamma\kappa} == -\epsilon\, a_\beta\; a_\delta + K_\beta{}^\iota\; K_{\delta\iota} + a_{\beta|\delta} - \mathcal{L}_n K_{\beta\delta} \\
{}^{(3)}R_{\mu\nu\kappa\lambda} == h_{\nu\lambda}\; {}^{(3)}R_{\mu\kappa} - h_{\nu\kappa}\; {}^{(3)}R_{\mu\lambda} - h_{\mu\lambda}\; {}^{(3)}R_{\nu\kappa} + h_{\mu\kappa}\; {}^{(3)}R_{\nu\lambda} + \frac{1}{2}\left(h_{\mu\lambda}\; h_{\nu\kappa} - h_{\mu\kappa}\; h_{\nu\lambda}\right) {}^{(3)}R \\
{}^{(3)}R_{\kappa\lambda\mu\nu}\; {}^{(3)}R^{\kappa\lambda\mu\nu} == 4\; {}^{(3)}R_{\kappa\lambda}\; {}^{(3)}R^{\kappa\lambda} - {}^{(3)}R^2
\end{pmatrix}$$

The Ricci tensor induced 3+1 decomposition summarizes to:

In[ ]:= **DisplayEQ@$EquationsRicci3DDeco**

Out[ ]//MatrixForm=

$$\begin{pmatrix} -\epsilon\, h_\beta^{\ \alpha}\ h_\delta^{\ \gamma}\ R_{\alpha\gamma} == \epsilon\, a_\beta\ a_\delta + K^\alpha_{\ \alpha}\ K_{\beta\delta} - \epsilon\ {}^{(3)}R_{\beta\delta} - a_{\beta|\delta} + h_\beta^{\ \gamma}\ h_\delta^{\ \lambda}\ n^\alpha\ K_{\gamma\lambda;\alpha} \\ -\epsilon\, h_\beta^{\ \alpha}\ h_\delta^{\ \gamma}\ R_{\alpha\gamma} == K^\alpha_{\ \alpha}\ K_{\beta\delta} - K_\beta^{\ \alpha}\ K_{\delta\alpha} - \epsilon\ {}^{(3)}R_{\beta\delta} - h_\beta^{\ \lambda}\ h_\delta^{\ \kappa}\ n^\alpha\ n^\gamma\ R_{\alpha\lambda\gamma\kappa} \\ h_\delta^{\ \alpha}\ n^\beta\ R_{\beta\alpha} == K_\delta^{\ \alpha}{}_{|\alpha} - K^\alpha_{\ \alpha|\delta} \\ n^\alpha\ n^\beta\ R_{\alpha\beta} == -\epsilon\, a_\alpha\ a^\alpha - K_\alpha^{\ \beta}\ K^\alpha_{\ \beta} + a_\alpha^{\ |\alpha} - n^\beta\ K_\alpha^{\ \alpha}{}_{;\beta} \\ n^\alpha\ n^\beta\ R_{\alpha\beta} == -\epsilon\, a_\alpha\ a^\alpha - K_\alpha^{\ \beta}\ K^\alpha_{\ \beta} + a_\alpha^{\ |\alpha} - h^{\gamma\alpha}\ n^\beta\ K_{\alpha\gamma;\beta} \\ n^\alpha\ n^\beta\ R_{\alpha\beta} == -\epsilon\, a_\alpha\ a^\alpha + K_\alpha^{\ \beta}\ K^\alpha_{\ \beta} + a_\alpha^{\ |\alpha} - h^{\alpha\beta}\ \left(\mathcal{L}_n K_{\alpha\beta}\right) \\ n^\alpha\ n^\beta\ R_{\alpha\beta} == -K_\alpha^{\ \beta}\ K^\alpha_{\ \beta} + a^\alpha{}_{;\alpha} - n^\beta\ K_\alpha^{\ \alpha}{}_{;\beta} \\ a^\alpha{}_{;\alpha} == K_\alpha^{\ \beta}\ K^\alpha_{\ \beta} + n^\alpha\ n^\beta\ R_{\alpha\beta} + n^\beta\ K_\alpha^{\ \alpha}{}_{;\beta} \end{pmatrix}$$

The Ricci scalar induced 3+1 decomposition summarizes to:

In[ ]:= **DisplayEQ@$EquationsRicciScalar3DDeco**

Out[ ]//MatrixForm=

$$\begin{pmatrix} -\epsilon\, R == 2\,\epsilon\, a_\alpha\ a^\alpha + K_{\alpha\beta}\ K^{\alpha\beta} + K^\alpha_{\ \alpha}\ K^\beta_{\ \beta} - \epsilon\ {}^{(3)}R - 2\ a^\alpha{}_{|\alpha} + h^{\beta\gamma}\ n^\alpha\ K_{\beta\gamma;\alpha} + n^\alpha\ K^\beta_{\ \beta;\alpha} \\ -\epsilon\, R == K_{\alpha\beta}\ K^{\alpha\beta} + K^\alpha_{\ \alpha}\ K^\beta_{\ \beta} - \epsilon\ {}^{(3)}R - 2\ a^\alpha{}_{;\alpha} + 2\ n^\alpha\ K^\beta_{\ \beta;\alpha} \\ -\epsilon\, R == -K_{\alpha\beta}\ K^{\alpha\beta} + K^\alpha_{\ \alpha}\ K^\beta_{\ \beta} - 2\ n^\alpha\ n^\beta\ R_{\alpha\beta} - \epsilon\ {}^{(3)}R \end{pmatrix}$$

"Normal-change" & momentum and Hamiltonian constraints & "insurance" equations state:

In[ ]:= **DisplayEQ@$EquationsNormalChange**

Out[ ]//MatrixForm=

$$\begin{pmatrix} -\epsilon\ {}^{(3)}R_{\beta\delta} == -\epsilon\, a_\beta\ a_\delta - K^\alpha_{\ \alpha}\ K_{\beta\delta} + a_{\beta|\delta} - h_\beta^{\ \gamma}\ h_\delta^{\ \lambda}\ n^\alpha\ K_{\gamma\lambda;\alpha} \\ -\epsilon\ {}^{(3)}R_{\beta\delta} == -K^\alpha_{\ \alpha}\ K_{\beta\delta} + K_\beta^{\ \alpha}\ K_{\delta\alpha} + n^\alpha\ n^\gamma\ R_{\beta\alpha\delta\gamma} \end{pmatrix}$$

In[ ]:= **DisplayEQ@$EquationsConstraints**

Out[ ]//MatrixForm=

$$\begin{pmatrix} K^\lambda_{\ \kappa|\lambda} == K_\alpha^{\ \alpha}{}_{|\kappa} \\ -\epsilon\ {}^{(3)}R == -K^\alpha_{\ \alpha}{}^2 + K^\gamma_{\ \lambda}\ K^\lambda_{\ \gamma} \\ -\epsilon\, a_\lambda\ a^\lambda + a^\lambda{}_{|\lambda} == K^\gamma_{\ \lambda}\ K^\lambda_{\ \gamma} + n^\gamma\ K^\alpha_{\ \alpha;\gamma} \\ a^\alpha{}_{;\alpha} == K_{\alpha\gamma}\ K^{\alpha\gamma} + n^\alpha\ K^\gamma_{\ \gamma;\alpha} \end{pmatrix}$$

3+1 decomposition of the Kretschmann scalar

In[ ]:= **$EquationsKretsch3DDeco = {};**

With the previous chapter's decomposition equations Kretschmann scalar can be now decomposed.

Following is:

- Printing out option set for 4D-Kretschmann scalar
- General definition recap
- Preparation of $g_{\mu\alpha}\, g_{\nu\beta}\, g_{\kappa\gamma}\, g_{\lambda\delta}$ term decomposition
  - this term is treated separately in order not to "meddle" with automatic contraction option

we have set previously

- the difficulty of treating this term in the general definition is exactly the automatic metric contraction

• Kretschmann scalar “geometrical/general” 3+1 decomposition

• Deploying Gauss, Codazzi & Ricci equations developed in the previous chapter to find out each term of the “general” decomposition

- note that we do this “step by step” in order to save computation time

• Adding the pre-prepared terms, thus arriving at the Kretschmann scalar 3+1 decomposition

Printing out options & preparations:

First, we have to address it’s printing options

```
In[•]:= PrintAs[KretschmannCD] ^= "𝒦";
```

```
In[•]:= KretschmannCD[] == RiemannCD[-μ, -ν, -κ, -λ] × RiemannCD[μ, ν, κ, λ]
```

$$\text{Out[•]= } \mathcal{K} == R_{\mu\nu\kappa\lambda}\ R^{\mu\nu\kappa\lambda}$$

It will be most advantageous to now to lower the fully covariant Riemann indices. However we wish to prevent the expression from automatically contract the spacetime metric with the indices, thus we will decompose following term separately:

```
In[•]:= g[μ, α] × g[ν, β] × g[κ, γ] × g[λ, δ]
        % /. ApplyRule@gTOh
        % // Expand;
        g4FullExpansion = %;
```

$$\text{Out[•]= } g^{\kappa\gamma}\ g^{\lambda\delta}\ g^{\mu\alpha}\ g^{\nu\beta}$$

$$\text{Out[•]= } \left(h^{\kappa\gamma} + \epsilon\ n^{\gamma}\ n^{\kappa}\right)\left(h^{\lambda\delta} + \epsilon\ n^{\delta}\ n^{\lambda}\right)\left(h^{\mu\alpha} + \epsilon\ n^{\alpha}\ n^{\mu}\right)\left(h^{\nu\beta} + \epsilon\ n^{\beta}\ n^{\nu}\right)$$

```
In[•]:= AddEquation@g4FullExpansion;
        AddToList[$Equations3Dmetric, g4FullExpansion];
```

Now for the Kretschmann scalar with two fully covariant Riemann tensors. Bear in mind, that at this point we could simply perform something along the lines of $R_{\mu\nu\kappa\lambda}\,R_{\alpha\beta\gamma\delta}$ ***g4FullExpansion**, then expand and canonicalize the resulting expression. However even at this point, the calculation involving this many terms is quite lengthy.

For the record, this approach took **86s** to calculate.

Whereas treating the multiplication (and canonicalization) term by term is very economic approach to the problem :

```
In[•]:= Times[g4FullExpansion // ScreenDollarIndices, RiemannCD[-α, -β, -γ, -δ]];
        % // ToCanonical;
        Times[%, RiemannCD[-μ, -ν, -κ, -λ]];
        % // ToCanonical;
        KretschmannCD[] == %
        Kretsch3DDeco1 = %;
```

$$\text{Out[•]=}\quad \mathcal{K} == h^{\alpha\beta}\ h^{\gamma\delta}\ h^{\kappa\lambda}\ h^{\mu\nu}\ R_{\alpha\gamma\kappa\mu}\ R_{\beta\delta\lambda\nu} + 4 \in h^{\gamma\delta}\ h^{\kappa\lambda}\ h^{\mu\nu}\ n^{\alpha}\ n^{\beta}\ R_{\alpha\gamma\kappa\mu}\ R_{\beta\delta\lambda\nu} + 4\ h^{\kappa\lambda}\ h^{\mu\nu}\ n^{\alpha}\ n^{\beta}\ n^{\gamma}\ n^{\delta}\ R_{\alpha\kappa\beta\mu}\ R_{\gamma\lambda\delta\nu}$$

```
In[•]:= AddEquation@Kretsch3DDeco1;
        AddToList[$EquationsKretsch3DDeco, Kretsch3DDeco1];
```

As we can see, the terms clearly correspond to the Gauss, Codazzi and Ricci terms of both of the Riemann tensors figuring in the decomposition.
Thus:

```
In[•]:= (ProjectWith[h][RiemannCD[-α, -β, -γ, -δ]])
         (ProjectWith[h][RiemannCD[-μ, -ν, -κ, -λ]])
        % /. ApplyRule@GaussEquation
        % /. ApplyRule@GaussEquation
        Term1 = Times[%, h[λ, δ] × h[κ, γ] × h[ν, β] × h[α, μ]]
```

$$\text{Out[•]=}\quad h_{\alpha}^{\ \iota}\ h_{\beta}^{\ \rho}\ h_{\gamma}^{\ \sigma}\ h_{\delta}^{\ \upsilon}\ h_{\kappa}^{\ \chi}\ h_{\lambda}^{\ \chi 1}\ h_{\mu}^{\ \chi 2}\ h_{\nu}^{\ \chi 3}\ R_{\iota\rho\sigma\upsilon}\ R_{\chi 2\chi 3\chi\chi 1}$$

```
12.0416
```

$$\text{Out[•]=}\quad h_{\kappa}^{\ \iota}\ h_{\lambda}^{\ \rho}\ h_{\mu}^{\ \sigma}\ h_{\nu}^{\ \upsilon}\ \left(\in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}\right)\ R_{\sigma\upsilon\iota\rho}$$

```
12.0254
```

$$\text{Out[•]=}\quad \left(\in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}\right)\ \left(\in K_{\kappa\nu}\ K_{\lambda\mu} - \in K_{\kappa\mu}\ K_{\lambda\nu} + {}^{(3)}R_{\lambda\kappa\nu\mu}\right)$$

$$\text{Out[•]=}\quad h^{\alpha\mu}\ h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(\in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}\right)\ \left(\in K_{\kappa\nu}\ K_{\lambda\mu} - \in K_{\kappa\mu}\ K_{\lambda\nu} + {}^{(3)}R_{\lambda\kappa\nu\mu}\right)$$

```
In[•]:= ProjectWith[h][n[α] × RiemannCD[-α, -β, -γ, -δ]] ×
         ProjectWith[h][n[μ] × RiemannCD[-μ, -ν, -κ, -λ]]
        % /. ApplyRule@CodazziEquation
        % /. ApplyRule@CodazziEquation
        Term2 = Times[%, h[ν, β] × h[κ, γ] × h[λ, δ]]
```

$$\text{Out[•]=}\quad h_{\beta}^{\ \iota}\ h_{\gamma}^{\ \rho}\ h_{\delta}^{\ \sigma}\ h_{\kappa}^{\ \upsilon}\ h_{\lambda}^{\ \chi}\ h_{\nu}^{\ \chi 1}\ n^{\alpha}\ n^{\mu}\ R_{\alpha\iota\rho\sigma}\ R_{\mu\chi 1\upsilon\chi}$$

```
5.28357
```

$$\text{Out[•]=}\quad h_{\kappa}^{\ \alpha}\ h_{\lambda}^{\ \iota}\ h_{\nu}^{\ \rho}\ n^{\mu}\ R_{\mu\rho\alpha\iota}\ \left(-K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}\right)$$

```
5.24414
```

$$\text{Out[•]=}\quad \left(-K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}\right)\ \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)$$

$$\text{Out[•]=}\quad h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(-K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}\right)\ \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)$$

```
In[•]:= ProjectWith[h][n[α] × n[γ] × RiemannCD[-α, -β, -γ, -δ]] ×
         ProjectWith[h][n[μ] × n[κ] × RiemannCD[-μ, -ν, -κ, -λ]]
        % /. ApplyRule@RicciEquation2
        % /. ApplyRule@RicciEquation2
        Times[%, h[β, ν] × h[δ, λ]]
        Term3 = %;
```

Out[•]= $h_{\beta}^{\iota}\ h_{\delta}^{\rho}\ h_{\lambda}^{\sigma}\ h_{\nu}^{\upsilon}\ n^{\alpha}\ n^{\gamma}\ n^{\kappa}\ n^{\mu}\ R_{\alpha\iota\gamma\rho}\ R_{\mu\upsilon\kappa\sigma}$

3.37898

Out[•]= $h_{\lambda}^{\alpha}\ h_{\nu}^{\gamma}\ n^{\kappa}\ n^{\mu}\ R_{\mu\gamma\kappa\alpha}\ \left(-\epsilon\ a_{\beta}\ a_{\delta} - K_{\beta}^{\iota}\ K_{\delta\iota} + a_{\beta|\delta} - h_{\beta}^{\rho}\ h_{\delta}^{\sigma}\ n^{\iota}\ K_{\rho\sigma;\iota}\right)$

3.32938

Out[•]= $\left(-\epsilon\ a_{\beta}\ a_{\delta} - K_{\beta}^{\alpha}\ K_{\delta\alpha} + a_{\beta|\delta} - h_{\beta}^{\gamma}\ h_{\delta}^{\iota}\ n^{\alpha}\ K_{\gamma\iota;\alpha}\right)$
$\left(-\epsilon\ a_{\lambda}\ a_{\nu} - K_{\lambda\kappa}\ K_{\nu}^{\kappa} + a_{\nu|\lambda} - h_{\lambda}^{\rho}\ h_{\nu}^{\mu}\ n^{\kappa}\ K_{\mu\rho;\kappa}\right)$

Out[•]= $h^{\beta\nu}\ h^{\delta\lambda}\ \left(-\epsilon\ a_{\beta}\ a_{\delta} - K_{\beta}^{\alpha}\ K_{\delta\alpha} + a_{\beta|\delta} - h_{\beta}^{\gamma}\ h_{\delta}^{\iota}\ n^{\alpha}\ K_{\gamma\iota;\alpha}\right)$
$\left(-\epsilon\ a_{\lambda}\ a_{\nu} - K_{\lambda\kappa}\ K_{\nu}^{\kappa} + a_{\nu|\lambda} - h_{\lambda}^{\rho}\ h_{\nu}^{\mu}\ n^{\kappa}\ K_{\mu\rho;\kappa}\right)$

Full Kretschmann scalar 3+1 Decomposition

```
In[•]:= KretschmannCD[] == Term1 + 4 ϵ * Term2 + 4 Term3
        Kretsch3DDeco2 = %;
```

Out[•]= $\mathcal{K} ==$
$h^{\alpha\mu}\ h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(\epsilon\ K_{\gamma\beta}\ K_{\delta\alpha} - \epsilon\ K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}\right)\ \left(\epsilon\ K_{\kappa\nu}\ K_{\lambda\mu} - \epsilon\ K_{\kappa\mu}\ K_{\lambda\nu} + {}^{(3)}R_{\lambda\kappa\nu\mu}\right) +$
$4\ \epsilon\ h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(-K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}\right)\ \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right) +$
$4\ h^{\beta\nu}\ h^{\delta\lambda}\ \left(-\epsilon\ a_{\beta}\ a_{\delta} - K_{\beta}^{\alpha}\ K_{\delta\alpha} + a_{\beta|\delta} - h_{\beta}^{\gamma}\ h_{\delta}^{\iota}\ n^{\alpha}\ K_{\gamma\iota;\alpha}\right)$
$\left(-\epsilon\ a_{\lambda}\ a_{\nu} - K_{\lambda\kappa}\ K_{\nu}^{\kappa} + a_{\nu|\lambda} - h_{\lambda}^{\rho}\ h_{\nu}^{\mu}\ n^{\kappa}\ K_{\mu\rho;\kappa}\right)$

```
In[•]:= DisplayEQ@$Equations3DExtCurv
```

Out[•]//MatrixForm=

$$\begin{pmatrix}
K_{\alpha\beta} == h_{\alpha}^{\gamma}\ h_{\beta}^{\delta}\ n_{\gamma;\delta} \\
K_{\beta\alpha} == -\epsilon\ a_{\alpha}\ n_{\beta} + n_{\alpha;\beta} \\
K_{\mu\nu} == \frac{1}{2}\,\text{Hold}\left[\mathcal{L}_{n}\,h_{\mu\nu}\right] \\
n^{\gamma}\ K_{\alpha\beta;\gamma} == -2\ K_{\alpha}^{\gamma}\ K_{\beta\gamma} - \epsilon\ a^{\gamma}\ K_{\beta\gamma}\ n_{\alpha} - \epsilon\ a^{\gamma}\ K_{\alpha\gamma}\ n_{\beta} + \mathcal{L}_{n}\,K_{\alpha\beta} \\
K_{\beta\gamma;\alpha} == -2\,\epsilon\,K_{\beta}^{\delta}\ K_{\gamma\delta}\ n_{\alpha} - \epsilon\,K_{\alpha}^{\delta}\ K_{\gamma\delta}\ n_{\beta} - a^{\delta}\ K_{\gamma\delta}\ n_{\alpha}\ n_{\beta} - \epsilon\,K_{\alpha}^{\delta}\ K_{\beta\delta}\ n_{\gamma} - a^{\delta}\ K_{\beta\delta}\ n_{\alpha}\ n_{\gamma} + K_{\beta\cdot} \\
h_{\beta}^{\alpha}\ h_{\delta}^{\gamma}\ n^{\iota}\ K_{\alpha\gamma;\iota} == -\epsilon\ a_{\beta}\ a_{\delta} - K_{\beta}^{\iota}\ K_{\delta\iota} - h_{\beta}^{\iota}\ h_{\delta}^{\kappa}\ n^{\alpha}\ n^{\gamma}\ R_{\alpha\iota\gamma\kappa} + a_{\beta|\delta} \\
n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta} == h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta} \\
h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta} == n^{\beta}\ K_{\alpha}{}^{\alpha}{}_{;\beta}
\end{pmatrix}$$

```
In[•]:= AddEquation@Kretsch3DDeco2;
        AddToList[$EquationsKretsch3DDeco, Kretsch3DDeco2];
```

Thus the full 3+1 decomposition of the Kretschmann scalar reads (the **main result at this point**)

In[•]:=
```
Kretsch3DDeco1
Kretsch3DDeco2
```

Out[•]=
$$\mathcal{K} == h^{\alpha\beta}\ h^{\gamma\delta}\ h^{\kappa\lambda}\ h^{\mu\nu}\ R_{\alpha\gamma\kappa\mu}\ R_{\beta\delta\lambda\nu} + 4 \in h^{\gamma\delta}\ h^{\kappa\lambda}\ h^{\mu\nu}\ n^{\alpha}\ n^{\beta}\ R_{\alpha\gamma\kappa\mu}\ R_{\beta\delta\lambda\nu} + 4\ h^{\kappa\lambda}\ h^{\mu\nu}\ n^{\alpha}\ n^{\beta}\ n^{\gamma}\ n^{\delta}\ R_{\alpha\kappa\beta\mu}\ R_{\gamma\lambda\delta\nu}$$

Out[•]=
$$\mathcal{K} == h^{\alpha\mu}\ h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(\in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}\right)\left(\in K_{\kappa\nu}\ K_{\lambda\mu} - \in K_{\kappa\mu}\ K_{\lambda\nu} + {}^{(3)}R_{\lambda\kappa\nu\mu}\right) + 4 \in h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(-K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}\right)\left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right) + 4\ h^{\beta\nu}\ h^{\delta\lambda}\ \left(-\in a_{\beta}\ a_{\delta} - K_{\beta}{}^{\alpha}\ K_{\delta\alpha} + a_{\beta|\delta} - h_{\beta}{}^{\gamma}\ h_{\delta}{}^{\iota}\ n^{\alpha}\ K_{\gamma\iota;\alpha}\right)\left(-\in a_{\lambda}\ a_{\nu} - K_{\lambda\kappa}\ K_{\nu}{}^{\kappa} + a_{\nu|\lambda} - h_{\lambda}{}^{\rho}\ h_{\nu}{}^{\mu}\ n^{\kappa}\ K_{\mu\rho;\kappa}\right)$$

The vacuum case, application of constraints.

In this short, however **pivotal** chapter, we will take a closer look at last term of the 3+1 Kretschmann scalar decomposition **in vacuum part of the space-time**.

In[•]:=
```
Term3
```

Out[•]=
$$h^{\beta\nu}\ h^{\delta\lambda}\ \left(-\in a_{\beta}\ a_{\delta} - K_{\beta}{}^{\alpha}\ K_{\delta\alpha} + a_{\beta|\delta} - h_{\beta}{}^{\gamma}\ h_{\delta}{}^{\iota}\ n^{\alpha}\ K_{\gamma\iota;\alpha}\right)\left(-\in a_{\lambda}\ a_{\nu} - K_{\lambda\kappa}\ K_{\nu}{}^{\kappa} + a_{\nu|\lambda} - h_{\lambda}{}^{\rho}\ h_{\nu}{}^{\mu}\ n^{\kappa}\ K_{\mu\rho;\kappa}\right)$$

Considering now the “normal change equation” in vacuum.

In[•]:=
```
NormChangEq1
```

Out[•]=
$$-\in {}^{(3)}R_{\beta\delta} == -\in a_{\beta}\ a_{\delta} - K^{\alpha}{}_{\alpha}\ K_{\beta\delta} + a_{\beta|\delta} - h_{\beta}{}^{\gamma}\ h_{\delta}{}^{\iota}\ n^{\alpha}\ K_{\gamma\iota;\alpha}$$

We can see that most of the terms of the normal change equation are already contained in the **Term3**

In[•]:=
```
<< xAct`xTras`
```

```
------------------------------------------------------------

Package xAct`xPert`  version 1.0.6, {2018, 2, 28}

CopyRight (C) 2005-2020, David Brizuela, Jose M. Martin-Garcia
  and Guillermo A. Mena Marugan, under the General Public License.

** Option MetricOn of MakeRule changed from None to All

** Option ContractMetrics of MakeRule changed from False to True

------------------------------------------------------------

Package xAct`Invar`  version 2.0.5, {2013, 7, 1}

CopyRight (C) 2006-2020, J. M. Martin-Garcia,
  D. Yllanes and R. Portugal, under the General Public License.

** Option CurvatureRelations of DefCovD changed from True to False

** Variable $CommuteCovDsOnScalars changed from True to False

------------------------------------------------------------

Package xAct`xCoba`  version 0.8.6, {2021, 2, 28}

CopyRight (C) 2005-2021, David Yllanes
  and Jose M. Martin-Garcia, under the General Public License.

------------------------------------------------------------

Package xAct`SymManipulator`  version 0.9.5, {2021, 9, 14}

CopyRight (C) 2011-2021, Thomas Bäckdahl, under the General Public License.

------------------------------------------------------------

Package xAct`xTras`  version 1.4.2, {2014, 10, 30}

CopyRight (C) 2012-2014, Teake Nutma, under the General Public License.

** Variable $CovDFormat changed from Postfix to Prefix

** Option CurvatureRelations of DefCovD changed from False to True

------------------------------------------------------------

These packages come with ABSOLUTELY NO WARRANTY; for details type
  Disclaimer[]. This is free software, and you are welcome to redistribute
  it under certain conditions. See the General Public License for details.

------------------------------------------------------------

2.85266
```

In[•]:=
```
NormChangEq1 // ScreenDollarIndices
tmp = SolveTensors[%, -ϵ Accelerationn[-β] × Accelerationn[-δ]];
```

Out[•]= $-\epsilon\ {}^{(3)}R_{\beta\delta} == -\epsilon\ a_\beta\ a_\delta - K^\alpha{}_\alpha\ K_{\beta\delta} + D_\delta\, a_\beta - h_\beta{}^\gamma\ h_\delta{}^\iota\ n^\alpha\ \nabla_\alpha K_{\gamma\iota}$

In[•]:=
```
Term3 /. tmp
%[[1]]
Term3Vac1 = %;
```

Out[•]= $\left\{ h^{\beta\nu}\ h^{\delta\lambda}\ \left(-K_\beta{}^\alpha\ K_{\delta\alpha} + K_{\beta\delta}\ K^\iota{}_\iota - \epsilon\ {}^{(3)}R_{\beta\delta}\right)\ \left(-K_{\lambda\gamma}\ K_\nu{}^\gamma + K^\kappa{}_\kappa\ K_{\nu\lambda} - \epsilon\ {}^{(3)}R_{\nu\lambda}\right)\right\}$

Out[•]= $h^{\beta\nu}\ h^{\delta\lambda}\ \left(-K_\beta{}^\alpha\ K_{\delta\alpha} + K_{\beta\delta}\ K^\iota{}_\iota - \epsilon\ {}^{(3)}R_{\beta\delta}\right)\ \left(-K_{\lambda\gamma}\ K_\nu{}^\gamma + K^\kappa{}_\kappa\ K_{\nu\lambda} - \epsilon\ {}^{(3)}R_{\nu\lambda}\right)$

In[•]:=
```
Clear@tmp
```

Now let's account for only Term1 & Term3 (in vacuum case) of the Kretschmann scalar 3+1

decomposition.

```
In[•]:= KretschTerm2Omit = Term1 + 4 Term3Vac1
        % // Expand;
        % // ToCanonical;
        Collect[%, _Riccicd]
        KretschTerm2OmitExpanded = %;
```

Out[•]= $4\, h^{\beta\nu}\ h^{\delta\lambda}\ \left(-K_{\beta}{}^{\alpha}\ K_{\delta\alpha} + K_{\beta\delta}\ K^{\iota}{}_{\iota} - \epsilon\ {}^{(3)}R_{\beta\delta}\right)\ \left(-K_{\lambda\gamma}\ K_{\nu}{}^{\gamma} + K^{\kappa}{}_{\kappa}\ K_{\nu\lambda} - \epsilon\ {}^{(3)}R_{\nu\lambda}\right) +$
$h^{\alpha\mu}\ h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(\epsilon\ K_{\gamma\beta}\ K_{\delta\alpha} - \epsilon\ K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}\right)\ \left(\epsilon\ K_{\kappa\nu}\ K_{\lambda\mu} - \epsilon\ K_{\kappa\mu}\ K_{\lambda\nu} + {}^{(3)}R_{\lambda\kappa\nu\mu}\right)$

Out[•]= $2\, K_{\alpha}{}^{\gamma}\ K^{\alpha\beta}\ K_{\beta}{}^{\delta}\ K_{\gamma\delta} - 8\, K^{\alpha}{}_{\alpha}\ K_{\beta}{}^{\delta}\ K^{\beta\gamma}\ K_{\gamma\delta} +$
$2\, K_{\alpha\beta}\ K^{\alpha\beta}\ K_{\gamma\delta}\ K^{\gamma\delta} + 4\, K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta}\ K_{\gamma\delta}\ K^{\gamma\delta} + 4\ {}^{(3)}R_{\alpha\beta}\ {}^{(3)}R^{\alpha\beta} +$
$\left(8\epsilon\ K_{\alpha}{}^{\gamma}\ K^{\alpha\beta} - 8\epsilon\ K^{\alpha}{}_{\alpha}\ K^{\beta\gamma}\right)\ {}^{(3)}R_{\beta\gamma} - 4\epsilon\ K^{\alpha\beta}\ K^{\gamma\delta}\ {}^{(3)}R_{\alpha\gamma\beta\delta} + {}^{(3)}R_{\alpha\beta\gamma\delta}\ {}^{(3)}R^{\alpha\beta\gamma\delta}$

```
In[•]:= $CovDFormat = "Postfix";
```

```
In[•]:= Term2
        MapAt[Hold, %, {5}];
        % // ContractMetric;
        Collect[%, _Hold];
        % // ReleaseHold
        Term2Var1 = %;
```

Out[•]= $h^{\kappa\gamma}\ h^{\lambda\delta}\ h^{\nu\beta}\ \left(-K_{\beta\delta\,|\,\gamma} + K_{\beta\gamma\,|\,\delta}\right)\ \left(-K_{\nu\lambda\,|\,\kappa} + K_{\nu\kappa\,|\,\lambda}\right)$

Out[•]= $\left(-K_{\nu\lambda\,|\,\kappa} + K_{\nu\kappa\,|\,\lambda}\right)\ \left(-K^{\lambda\nu\,|\,\kappa} + K^{\kappa\nu\,|\,\lambda}\right)$

```
In[•]:= KretschmannCD[] == KretschTerm2OmitExpanded + 4 ϵ Term2Var1
        KretschVacuum3DDeco1 = %;
```

Out[•]= $\mathcal{K} == 2\, K_{\alpha}{}^{\gamma}\ K^{\alpha\beta}\ K_{\beta}{}^{\delta}\ K_{\gamma\delta} - 8\, K^{\alpha}{}_{\alpha}\ K_{\beta}{}^{\delta}\ K^{\beta\gamma}\ K_{\gamma\delta} + 2\, K_{\alpha\beta}\ K^{\alpha\beta}\ K_{\gamma\delta}\ K^{\gamma\delta} +$
$4\, K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta}\ K_{\gamma\delta}\ K^{\gamma\delta} + 4\ {}^{(3)}R_{\alpha\beta}\ {}^{(3)}R^{\alpha\beta} + \left(8\epsilon\ K_{\alpha}{}^{\gamma}\ K^{\alpha\beta} - 8\epsilon\ K^{\alpha}{}_{\alpha}\ K^{\beta\gamma}\right)\ {}^{(3)}R_{\beta\gamma} -$
$4\epsilon\ K^{\alpha\beta}\ K^{\gamma\delta}\ {}^{(3)}R_{\alpha\gamma\beta\delta} + {}^{(3)}R_{\alpha\beta\gamma\delta}\ {}^{(3)}R^{\alpha\beta\gamma\delta} + 4\epsilon\left(-K_{\nu\lambda\,|\,\kappa} + K_{\nu\kappa\,|\,\lambda}\right)\ \left(-K^{\lambda\nu\,|\,\kappa} + K^{\kappa\nu\,|\,\lambda}\right)$

```
In[•]:= AddEquation@KretschVacuum3DDeco1;
        AddToList[$EquationsKretsch3DDeco, KretschVacuum3DDeco1];
```

As we can see, the vacuum “normal change equation” reduces the decomposition to **ONLY** several contractions of the (3D) extrinsic curvature tensor & 3D Ricci/Riemann tensor contractions.

Two further simplifications arise by considering firstly

```
In[•]:= Riemann3DIndepComp
```

Out[•]= ${}^{(3)}R_{\mu\nu\kappa\lambda} == h_{\nu\lambda}\ {}^{(3)}R_{\mu\kappa} - h_{\nu\kappa}\ {}^{(3)}R_{\mu\lambda} - h_{\mu\lambda}\ {}^{(3)}R_{\nu\kappa} + h_{\mu\kappa}\ {}^{(3)}R_{\nu\lambda} + \frac{1}{2}\left(h_{\mu\lambda}\ h_{\nu\kappa} - h_{\mu\kappa}\ h_{\nu\lambda}\right)\ {}^{(3)}R$

```
In[•]:= Riemanncd[-μ, -ν, -κ, -λ] × Riemanncd[μ, ν, κ, λ] ==
         (Riemanncd[-μ, -ν, -κ, -λ] × Riemanncd[μ, ν, κ, λ] /. ApplyRule@Riemann3DIndepComp)
        % // Expand;
        % // ToCanonical
        Kretsch2DIndepDeco = %;
```

$$\text{Out[•]=}\quad {}^{(3)}R_{\mu\nu\kappa\lambda}\ {}^{(3)}R^{\mu\nu\kappa\lambda} == \left(h_{\nu\lambda}\ {}^{(3)}R_{\mu\kappa} - h_{\nu\kappa}\ {}^{(3)}R_{\mu\lambda} - h_{\mu\lambda}\ {}^{(3)}R_{\nu\kappa} + h_{\mu\kappa}\ {}^{(3)}R_{\nu\lambda} + \frac{1}{2}\ h_{\mu\lambda}\ h_{\nu\kappa}\ {}^{(3)}R - \frac{1}{2}\ h_{\mu\kappa}\ h_{\nu\lambda}\ {}^{(3)}R\right)$$
$$\left(h^{\nu\lambda}\ {}^{(3)}R^{\mu\kappa} - h^{\nu\kappa}\ {}^{(3)}R^{\mu\lambda} - h^{\mu\lambda}\ {}^{(3)}R^{\nu\kappa} + h^{\mu\kappa}\ {}^{(3)}R^{\nu\lambda} + \frac{1}{2}\ h^{\mu\lambda}\ h^{\nu\kappa}\ {}^{(3)}R - \frac{1}{2}\ h^{\mu\kappa}\ h^{\nu\lambda}\ {}^{(3)}R\right)$$

$$\text{Out[•]=}\quad {}^{(3)}R_{\kappa\lambda\mu\nu}\ {}^{(3)}R^{\kappa\lambda\mu\nu} == 4\ {}^{(3)}R_{\kappa\lambda}\ {}^{(3)}R^{\kappa\lambda} - {}^{(3)}R^2$$

```
In[•]:= AddEquation@Kretsch2DIndepDeco;
```

```
In[•]:= Riemanncd[-μ, -ν, -κ, -λ] × ExtrinsicKh[μ, κ] × ExtrinsicKh[ν, λ] ==
         (Riemanncd[-μ, -ν, -κ, -λ] × ExtrinsicKh[μ, κ] × ExtrinsicKh[ν, λ] /.
           ApplyRule@Riemann3DIndepComp)
        % // Expand;
        PutScalar /@ %;
        % // ToCanonical;
        NoScalar /@ %;
        Collect[%, _RicciScalarcd]
        Riemann3DExtrCurv = %;
```

$$\text{Out[•]=}\quad K^{\mu\kappa}\ K^{\nu\lambda}\ {}^{(3)}R_{\mu\nu\kappa\lambda} == K^{\mu\kappa}\ K^{\nu\lambda}$$
$$\left(h_{\nu\lambda}\ {}^{(3)}R_{\mu\kappa} - h_{\nu\kappa}\ {}^{(3)}R_{\mu\lambda} - h_{\mu\lambda}\ {}^{(3)}R_{\nu\kappa} + h_{\mu\kappa}\ {}^{(3)}R_{\nu\lambda} + \frac{1}{2}\ h_{\mu\lambda}\ h_{\nu\kappa}\ {}^{(3)}R - \frac{1}{2}\ h_{\mu\kappa}\ h_{\nu\lambda}\ {}^{(3)}R\right)$$

$$\text{Out[•]=}\quad K^{\alpha\beta}\ K^{\gamma\delta}\ {}^{(3)}R_{\alpha\gamma\beta\delta} == -2\ K_{\alpha}{}^{\gamma}\ K^{\alpha\beta}\ {}^{(3)}R_{\beta\gamma} + 2\ K^{\alpha}{}_{\alpha}\ K^{\beta\gamma}\ {}^{(3)}R_{\beta\gamma} + \left(-\frac{1}{2}\ K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta} + \frac{1}{2}\ K_{\gamma\delta}\ K^{\gamma\delta}\right)\ {}^{(3)}R$$

```
In[•]:= AddEquation@Riemann3DExtrCurv;
```

This can be further simplified by

```
In[•]:= HamiltContr // ScreenDollarIndices
        Sol = SolveTensors[%, ExtrinsicKh[γ, -λ] × ExtrinsicKh[λ, -γ]];
```

$$\text{Out[•]=}\quad -\epsilon\ {}^{(3)}R == -K^{\alpha}{}_{\alpha}{}^{2} + K^{\gamma}{}_{\lambda}\ K^{\lambda}{}_{\gamma}$$

```
In[•]:= Riemann3DExtrCurv /. Sol;
        %[[1]] // ToCanonical
        Collect[%, _Riccicd]
        Riemann3DExtrCurvHamiltConstr = %;
```

$$\text{Out[•]=}\quad K^{\alpha\beta}\ K^{\gamma\delta}\ {}^{(3)}R_{\alpha\gamma\beta\delta} == -2\ K_{\alpha}{}^{\gamma}\ K^{\alpha\beta}\ {}^{(3)}R_{\beta\gamma} + 2\ K^{\alpha}{}_{\alpha}\ K^{\beta\gamma}\ {}^{(3)}R_{\beta\gamma} - \frac{\epsilon\ {}^{(3)}R^2}{2}$$

$$\text{Out[•]=}\quad K^{\alpha\beta}\ K^{\gamma\delta}\ {}^{(3)}R_{\alpha\gamma\beta\delta} == \left(-2\ K_{\alpha}{}^{\gamma}\ K^{\alpha\beta} + 2\ K^{\alpha}{}_{\alpha}\ K^{\beta\gamma}\right)\ {}^{(3)}R_{\beta\gamma} - \frac{\epsilon\ {}^{(3)}R^2}{2}$$

```
In[•]:= AddEquation@Riemann3DExtrCurvHamiltConstr;
```

```
In[•]:= KretschVacuum3DDeco1
        % /. ApplyRule@Kretsch2DIndepDeco
        % /. ApplyRule@Riemann3DExtrCurvHamiltConstr
        Collect[%, ϵ Riccicd[-α, -β] + ExtrinsicKh[-ν, -κ] × ExtrinsicKh[κ, -λ] -
          ExtrinsicKh[-α, α] × ExtrinsicKh[-ν, -λ]]
```

$$\text{Out[•]}= \mathcal{K} == 2\, K_\alpha{}^\gamma\, K^{\alpha\beta}\, K_\beta{}^\delta\, K_{\gamma\delta} - 8\, K^\alpha{}_\alpha\, K_\beta{}^\delta\, K^{\beta\gamma}\, K_{\gamma\delta} + 2\, K_{\alpha\beta}\, K^{\alpha\beta}\, K_{\gamma\delta}\, K^{\gamma\delta} + 4\, K^\alpha{}_\alpha\, K^\beta{}_\beta\, K_{\gamma\delta}\, K^{\gamma\delta} + 4\, {}^{(3)}R_{\alpha\beta}\, {}^{(3)}R^{\alpha\beta} + \left(8 \in K_\alpha{}^\gamma\, K^{\alpha\beta} - 8 \in K^\alpha{}_\alpha\, K^{\beta\gamma}\right) {}^{(3)}R_{\beta\gamma} - 4 \in K^{\alpha\beta}\, K^{\gamma\delta}\, {}^{(3)}R_{\alpha\gamma\beta\delta} + {}^{(3)}R_{\alpha\beta\gamma\delta}\, {}^{(3)}R^{\alpha\beta\gamma\delta} + 4 \in \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)\left(-K^{\lambda\nu|\kappa} + K^{\kappa\nu|\lambda}\right)$$

2.41683

$$\text{Out[•]}= \mathcal{K} == 2\, K_\alpha{}^\gamma\, K^{\alpha\beta}\, K_\beta{}^\delta\, K_{\gamma\delta} - 8\, K^\alpha{}_\alpha\, K_\beta{}^\delta\, K^{\beta\gamma}\, K_{\gamma\delta} + 2\, K_{\alpha\beta}\, K^{\alpha\beta}\, K_{\gamma\delta}\, K^{\gamma\delta} + 4\, K^\alpha{}_\alpha\, K^\beta{}_\beta\, K_{\gamma\delta}\, K^{\gamma\delta} + 8\, {}^{(3)}R_{\alpha\beta}\, {}^{(3)}R^{\alpha\beta} + \left(8 \in K_\alpha{}^\gamma\, K^{\alpha\beta} - 8 \in K^\alpha{}_\alpha\, K^{\beta\gamma}\right) {}^{(3)}R_{\beta\gamma} - {}^{(3)}R^2 - 4 \in K^{\alpha\beta}\, K^{\gamma\delta}\, {}^{(3)}R_{\alpha\gamma\beta\delta} + 4 \in \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)\left(-K^{\lambda\nu|\kappa} + K^{\kappa\nu|\lambda}\right)$$

1.61742

$$\text{Out[•]}= \mathcal{K} == 2\, K_\alpha{}^\gamma\, K^{\alpha\beta}\, K_\beta{}^\delta\, K_{\gamma\delta} - 8\, K^\alpha{}_\alpha\, K_\beta{}^\delta\, K^{\beta\gamma}\, K_{\gamma\delta} + 2\, K_{\alpha\beta}\, K^{\alpha\beta}\, K_{\gamma\delta}\, K^{\gamma\delta} + 4\, K^\alpha{}_\alpha\, K^\beta{}_\beta\, K_{\gamma\delta}\, K^{\gamma\delta} + 8\, {}^{(3)}R_{\alpha\beta}\, {}^{(3)}R^{\alpha\beta} + \left(8 \in K_\alpha{}^\gamma\, K^{\alpha\beta} - 8 \in K^\alpha{}_\alpha\, K^{\beta\gamma}\right) {}^{(3)}R_{\beta\gamma} - {}^{(3)}R^2 - 4 \in \left(-2\, K_\alpha{}^\gamma\, K^{\alpha\beta}\, {}^{(3)}R_{\beta\gamma} + 2\, K^\alpha{}_\alpha\, K^{\beta\gamma}\, {}^{(3)}R_{\beta\gamma} - \frac{\in {}^{(3)}R^2}{2}\right) + 4 \in \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)\left(-K^{\lambda\nu|\kappa} + K^{\kappa\nu|\lambda}\right)$$

$$\text{Out[•]}= \mathcal{K} == 2\, K_\alpha{}^\gamma\, K^{\alpha\beta}\, K_\beta{}^\delta\, K_{\gamma\delta} - 8\, K^\alpha{}_\alpha\, K_\beta{}^\delta\, K^{\beta\gamma}\, K_{\gamma\delta} + 2\, K_{\alpha\beta}\, K^{\alpha\beta}\, K_{\gamma\delta}\, K^{\gamma\delta} + 4\, K^\alpha{}_\alpha\, K^\beta{}_\beta\, K_{\gamma\delta}\, K^{\gamma\delta} + 8\, {}^{(3)}R_{\alpha\beta}\, {}^{(3)}R^{\alpha\beta} + \left(8 \in K_\alpha{}^\gamma\, K^{\alpha\beta} - 8 \in K^\alpha{}_\alpha\, K^{\beta\gamma}\right) {}^{(3)}R_{\beta\gamma} - {}^{(3)}R^2 - 4 \in \left(-2\, K_\alpha{}^\gamma\, K^{\alpha\beta}\, {}^{(3)}R_{\beta\gamma} + 2\, K^\alpha{}_\alpha\, K^{\beta\gamma}\, {}^{(3)}R_{\beta\gamma} - \frac{\in {}^{(3)}R^2}{2}\right) + 4 \in \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)\left(-K^{\lambda\nu|\kappa} + K^{\kappa\nu|\lambda}\right)$$

```
In[•]:= tmp = %;
```

xAct now lacks ability to convert this equation into somehow more “readable” form. Let’s compare it to following term summation:

```
In[•]:= CompTerm1 = 8 (ϵ Riccicd[-ν, -λ] + ExtrinsicKh[-ν, -κ] × ExtrinsicKh[κ, -λ] -
            ExtrinsicKh[α, -α] × ExtrinsicKh[-ν, -λ]) (ϵ Riccicd[ν, λ] +
            ExtrinsicKh[ν, -γ] × ExtrinsicKh[γ, λ] - ExtrinsicKh[γ, -γ] × ExtrinsicKh[ν, λ])
```

$$\text{Out[•]}= 8\left(K^\kappa{}_\lambda\, K_{\nu\kappa} - K^\alpha{}_\alpha\, K_{\nu\lambda} + \in {}^{(3)}R_{\nu\lambda}\right)\left(K^{\gamma\lambda}\, K^\nu{}_\gamma - K^\gamma{}_\gamma\, K^{\nu\lambda} + \in {}^{(3)}R^{\nu\lambda}\right)$$

```
In[•]:= CompTerm2 = RicciScalarcd[]^2 +
          2 (ExtrinsicKh[-μ, -κ] × ExtrinsicKh[μ, κ]) (ExtrinsicKh[-α, -β] × ExtrinsicKh[α, β]) -
          6 ExtrinsicKh[-μ, -κ] × ExtrinsicKh[μ, λ] × ExtrinsicKh[-ν, -λ] × ExtrinsicKh[ν, κ] +
          8 ExtrinsicKh[-α, α] × ExtrinsicKh[-ν, -κ] × ExtrinsicKh[κ, -λ] × ExtrinsicKh[ν, λ] -
          4 ExtrinsicKh[-α, α] × ExtrinsicKh[-β, β] × ExtrinsicKh[-ν, -λ] × ExtrinsicKh[ν, λ]
```

$$\text{Out[•]}= 2\, K_{\alpha\beta}\, K^{\alpha\beta}\, K_{\mu\kappa}\, K^{\mu\kappa} - 6\, K_{\mu\kappa}\, K^{\mu\lambda}\, K_{\nu\lambda}\, K^{\nu\kappa} + 8\, K_\alpha{}^\alpha\, K^\kappa{}_\lambda\, K_{\nu\kappa}\, K^{\nu\lambda} - 4\, K_\alpha{}^\alpha\, K_\beta{}^\beta\, K_{\nu\lambda}\, K^{\nu\lambda} + {}^{(3)}R^2$$

In[•]:=
```
CompTerm3 = 4 ϵ Term2Var1
```

Out[•]=
$$4\,\epsilon\,\left(-K_{\nu\lambda|\kappa}+K_{\nu\kappa|\lambda}\right)\,\left(-K^{\lambda\nu|\kappa}+K^{\kappa\nu|\lambda}\right)$$

In[•]:=
```
tmp
```

Out[•]=
$$\mathcal{K} == 2\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}\,K_{\beta}{}^{\delta}\,K_{\gamma\delta}-8\,K^{\alpha}{}_{\alpha}\,K_{\beta}{}^{\delta}\,K^{\beta\gamma}\,K_{\gamma\delta}+2\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}+4\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}+8\,{}^{(3)}R_{\alpha\beta}\,{}^{(3)}R^{\alpha\beta}+\left(8\,\epsilon\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}-8\,\epsilon\,K^{\alpha}{}_{\alpha}\,K^{\beta\gamma}\right)\,{}^{(3)}R_{\beta\gamma}-{}^{(3)}R^{2}-4\,\epsilon\left(-2\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}\,{}^{(3)}R_{\beta\gamma}+2\,K^{\alpha}{}_{\alpha}\,K^{\beta\gamma}\,{}^{(3)}R_{\beta\gamma}-\frac{\epsilon\,{}^{(3)}R^{2}}{2}\right)+4\,\epsilon\,\left(-K_{\nu\lambda|\kappa}+K_{\nu\kappa|\lambda}\right)\,\left(-K^{\lambda\nu|\kappa}+K^{\kappa\nu|\lambda}\right)$$

In[•]:=
```
CompTerm1 + CompTerm2 + CompTerm3 - tmp[[2]] // ToCanonical
PutScalar /@ %
```

Out[•]=
$$0$$

Out[•]=
$$0$$

The expression we have proposed matches

In[•]:=
```
tmp
```

Out[•]=
$$\mathcal{K} == 2\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}\,K_{\beta}{}^{\delta}\,K_{\gamma\delta}-8\,K^{\alpha}{}_{\alpha}\,K_{\beta}{}^{\delta}\,K^{\beta\gamma}\,K_{\gamma\delta}+2\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}+4\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}+8\,{}^{(3)}R_{\alpha\beta}\,{}^{(3)}R^{\alpha\beta}+\left(8\,\epsilon\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}-8\,\epsilon\,K^{\alpha}{}_{\alpha}\,K^{\beta\gamma}\right)\,{}^{(3)}R_{\beta\gamma}-{}^{(3)}R^{2}-4\,\epsilon\left(-2\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}\,{}^{(3)}R_{\beta\gamma}+2\,K^{\alpha}{}_{\alpha}\,K^{\beta\gamma}\,{}^{(3)}R_{\beta\gamma}-\frac{\epsilon\,{}^{(3)}R^{2}}{2}\right)+4\,\epsilon\,\left(-K_{\nu\lambda|\kappa}+K_{\nu\kappa|\lambda}\right)\,\left(-K^{\lambda\nu|\kappa}+K^{\kappa\nu|\lambda}\right)$$

In[•]:=
```
Clear@tmp;
```

In[•]:=
```
KretschmannCD[] == CompTerm1 + CompTerm2 + CompTerm3
KretschVacuum3DDeco2 = %;
```

Out[•]=
$$\mathcal{K} == 2\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\mu\kappa}\,K^{\mu\kappa}-6\,K_{\mu\kappa}\,K^{\mu\lambda}\,K_{\nu\lambda}\,K^{\nu\kappa}+8\,K_{\alpha}{}^{\alpha}\,K^{\kappa}{}_{\lambda}\,K_{\nu\kappa}\,K^{\nu\lambda}-4\,K_{\alpha}{}^{\alpha}\,K_{\beta}{}^{\beta}\,K_{\nu\lambda}\,K^{\nu\lambda}+8\left(K^{\kappa}{}_{\lambda}\,K_{\nu\kappa}-K^{\alpha}{}_{\alpha}\,K_{\nu\lambda}+\epsilon\,{}^{(3)}R_{\nu\lambda}\right)\left(K^{\gamma\lambda}\,K^{\nu}{}_{\gamma}-K^{\gamma}{}_{\gamma}\,K^{\nu\lambda}+\epsilon\,{}^{(3)}R^{\nu\lambda}\right)+{}^{(3)}R^{2}+4\,\epsilon\,\left(-K_{\nu\lambda|\kappa}+K_{\nu\kappa|\lambda}\right)\,\left(-K^{\lambda\nu|\kappa}+K^{\kappa\nu|\lambda}\right)$$

In[•]:=
```
AddEquation@KretschVacuum3DDeco2;
AddToList[$EquationsKretsch3DDeco, KretschVacuum3DDeco2];
```

The very last simplification arises from realization, that

In[•]:=
```
CompTerm2
```

Out[•]=
$$2\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\mu\kappa}\,K^{\mu\kappa}-6\,K_{\mu\kappa}\,K^{\mu\lambda}\,K_{\nu\lambda}\,K^{\nu\kappa}+8\,K_{\alpha}{}^{\alpha}\,K^{\kappa}{}_{\lambda}\,K_{\nu\kappa}\,K^{\nu\lambda}-4\,K_{\alpha}{}^{\alpha}\,K_{\beta}{}^{\beta}\,K_{\nu\lambda}\,K^{\nu\lambda}+{}^{(3)}R^{2}$$

Can be written out as $24*\mathrm{Det}\left[K^{\alpha}{}_{\beta}\right]$, except for one additional term!

For computing the determinant we have to summon xCoba and define an arbitrary chart:

In[•]:=
```
<< xAct`xCoba`
```

```
-------------------------------------------------------------

Package xAct`xCoba`  version 0.8.6, {2021, 2, 28}

CopyRight (C) 2005-2021, David Yllanes
  and Jose M. Martin-Garcia, under the General Public License.

-------------------------------------------------------------

These packages come with ABSOLUTELY NO WARRANTY; for details type
  Disclaimer[]. This is free software, and you are welcome to redistribute
  it under certain conditions. See the General Public License for details.

-------------------------------------------------------------
```

In[•]:=
```
DefChart[FM, M, {0, 1, 2, 3}, {t[], x[], y[], z[]}]
```

In[•]:=
```
Determinant[ExtrinsicKh[α, -β], FM]
(24 * %) // Expand // ToCanonical
tmp = %;
```

Out[•]=
$$\frac{1}{12}\,K^{\alpha}{}_{\gamma}\,K^{\beta}{}_{\beta}\,K^{\gamma}{}_{\delta}\,K^{\delta}{}_{\alpha}-\frac{1}{6}\,K^{\alpha}{}_{\beta}\,K^{\beta}{}_{\gamma}\,K^{\gamma}{}_{\delta}\,K^{\delta}{}_{\alpha}-\frac{1}{12}\,K^{\alpha}{}_{\gamma}\,K^{\beta}{}_{\alpha}\,K^{\gamma}{}_{\delta}\,K^{\delta}{}_{\beta}+\frac{1}{6}\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\gamma}\,K^{\gamma}{}_{\delta}\,K^{\delta}{}_{\beta}+\frac{1}{8}\,K^{\alpha}{}_{\beta}\,K^{\beta}{}_{\alpha}\,K^{\gamma}{}_{\delta}\,K^{\delta}{}_{\gamma}-\frac{1}{8}\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K^{\gamma}{}_{\delta}\,K^{\delta}{}_{\gamma}+\frac{1}{12}\,K^{\alpha}{}_{\beta}\,K^{\beta}{}_{\gamma}\,K^{\gamma}{}_{\alpha}\,K^{\delta}{}_{\delta}-\frac{1}{12}\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\gamma}\,K^{\gamma}{}_{\beta}\,K^{\delta}{}_{\delta}-\frac{1}{24}\,K^{\alpha}{}_{\beta}\,K^{\beta}{}_{\alpha}\,K^{\gamma}{}_{\gamma}\,K^{\delta}{}_{\delta}+\frac{1}{24}\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K^{\gamma}{}_{\gamma}\,K^{\delta}{}_{\delta}$$

Out[•]=
$$-6\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}\,K_{\beta}{}^{\delta}\,K_{\gamma\delta}+8\,K^{\alpha}{}_{\alpha}\,K_{\beta}{}^{\delta}\,K^{\beta\gamma}\,K_{\gamma\delta}+3\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}-6\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}+K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K^{\gamma}{}_{\gamma}\,K^{\delta}{}_{\delta}$$

This is the comparison of determinant and CompTerm2

In[•]:=
```
CompTerm2 - tmp
PutScalar /@ %;
% // ToCanonical
tmp = %;
```

Out[•]=
$$6\,K_{\alpha}{}^{\gamma}\,K^{\alpha\beta}\,K_{\beta}{}^{\delta}\,K_{\gamma\delta}-8\,K^{\alpha}{}_{\alpha}\,K_{\beta}{}^{\delta}\,K^{\beta\gamma}\,K_{\gamma\delta}-3\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}+6\,K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K_{\gamma\delta}\,K^{\gamma\delta}-K^{\alpha}{}_{\alpha}\,K^{\beta}{}_{\beta}\,K^{\gamma}{}_{\gamma}\,K^{\delta}{}_{\delta}+2\,K_{\alpha\beta}\,K^{\alpha\beta}\,K_{\mu\kappa}\,K^{\mu\kappa}-6\,K_{\mu\kappa}\,K^{\mu\lambda}\,K_{\nu\lambda}\,K^{\nu\kappa}+8\,K_{\alpha}{}^{\alpha}\,K^{\kappa}{}_{\lambda}\,K_{\nu\kappa}\,K^{\nu\lambda}-4\,K_{\alpha}{}^{\alpha}\,K_{\beta}{}^{\beta}\,K_{\nu\lambda}\,K^{\nu\lambda}+{}^{(3)}R^{2}$$

Out[•]=
$${}^{(3)}R^{2}-\left(K^{\alpha}{}_{\alpha}\right)^{4}+2\left(K^{\alpha}{}_{\alpha}\right)^{2}\left(K_{\alpha\beta}\,K^{\alpha\beta}\right)-\left(K_{\alpha\beta}\,K^{\alpha\beta}\right)^{2}$$

And this is the additional term.

```
In[•]:= AdditionalTerm = Riccicd[]^2 -
          (ExtrinsicKh[-α, α] × ExtrinsicKh[-γ, γ] - ExtrinsicKh[-α, -β] × ExtrinsicKh[α, β]) *
           (ExtrinsicKh[-ρ, ρ] × ExtrinsicKh[-σ, σ] - ExtrinsicKh[-ι, -σ] × ExtrinsicKh[ι, σ])
        PutScalar /@ %
        % // ToCanonical
```

$$\text{Out[•]=} \; -\left(\left(-K_{\alpha\beta}\ K^{\alpha\beta} + K_{\alpha}{}^{\alpha}\ K_{\gamma}{}^{\gamma}\right)\left(-K_{\iota\sigma}\ K^{\iota\sigma} + K_{\rho}{}^{\rho}\ K_{\sigma}{}^{\sigma}\right)\right) + {}^{(3)}R^2$$

$$\text{Out[•]=} \; {}^{(3)}R^2 - \left(K_{\alpha\beta}\ K^{\alpha\beta}\right)\left(K_{\iota\sigma}\ K^{\iota\sigma}\right) + \left(K_{\alpha}{}^{\alpha}\right)\left(K_{\gamma}{}^{\gamma}\right)\left(K_{\iota\sigma}\ K^{\iota\sigma}\right) + \left(K_{\alpha\beta}\ K^{\alpha\beta}\right)\left(K_{\rho}{}^{\rho}\right)\left(K_{\sigma}{}^{\sigma}\right) - \left(K_{\alpha}{}^{\alpha}\right)\left(K_{\gamma}{}^{\gamma}\right)\left(K_{\rho}{}^{\rho}\right)\left(K_{\sigma}{}^{\sigma}\right)$$

$$\text{Out[•]=} \; {}^{(3)}R^2 - \left(K_{\alpha}{}^{\alpha}\right)^4 + 2\left(K_{\alpha}{}^{\alpha}\right)^2\left(K_{\alpha\beta}\ K^{\alpha\beta}\right) - \left(K_{\alpha\beta}\ K^{\alpha\beta}\right)^2$$

This term, however, vanishes! Ricci scalar 3+1 decomposition structure in vacuum case is:

```
In[•]:= 0 == RicciScalar3DDecoEquation2[[2]]
        VacuumEq@%;
        % // ScreenDollarIndices
        Sol = SolveTensors[%, ExtrinsicKh[-α, -β] × ExtrinsicKh[α, β]];
```

$$\text{Out[•]=} \; 0 == -K_{\alpha\beta}\ K^{\alpha\beta} + K_{\alpha}{}^{\alpha}\ K_{\beta}{}^{\beta} - 2\ n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} - \in {}^{(3)}R$$

$$\text{Out[•]=} \; 0 == -K_{\alpha\beta}\ K^{\alpha\beta} + K_{\alpha}{}^{\alpha}\ K_{\beta}{}^{\beta} - \in {}^{(3)}R$$

```
In[•]:= AdditionalTerm /. Sol;
        %[[1]]
        % // Expand
        % // ToCanonical
        (*Some function of xAct treats RicciScalarcd[] as Riccicd[]*)
        % /. RicciScalarcd[] → Riccicd[]
```

$$\text{Out[•]=} \; {}^{(3)}R^2 - \left(K_{\alpha}{}^{\alpha}\ K_{\gamma}{}^{\gamma} - K_{\beta}{}^{\beta}\ K^{\iota}{}_{\iota} + \in {}^{(3)}R\right)\left(-K^{\delta}{}_{\delta}\ K^{\kappa}{}_{\kappa} + K_{\rho}{}^{\rho}\ K_{\sigma}{}^{\sigma} + \in {}^{(3)}R\right)$$

$$\text{Out[•]=} \; K_{\alpha}{}^{\alpha}\ K_{\beta}{}^{\beta}\ K_{\gamma}{}^{\gamma}\ K^{\delta}{}_{\delta} - K_{\alpha}{}^{\alpha}\ K_{\beta}{}^{\beta}\ K^{\gamma}{}_{\gamma}\ K^{\delta}{}_{\delta} + K_{\alpha}{}^{\alpha}\ K_{\beta}{}^{\beta}\ K_{\rho}{}^{\rho}\ K_{\sigma}{}^{\sigma} - K_{\alpha}{}^{\alpha}\ K_{\gamma}{}^{\gamma}\ K_{\rho}{}^{\rho}\ K_{\sigma}{}^{\sigma} + {}^{(3)}R^2 + 2 \in K_{\alpha}{}^{\alpha}\ K_{\beta}{}^{\beta}\ {}^{(3)}R - \in K_{\alpha}{}^{\alpha}\ K_{\gamma}{}^{\gamma}\ {}^{(3)}R - \in K_{\rho}{}^{\rho}\ K_{\sigma}{}^{\sigma}\ {}^{(3)}R - {}^{(3)}R^2$$

$$\text{Out[•]=} \; {}^{(3)}R^2 - {}^{(3)}R^2$$

$$\text{Out[•]=} \; 0$$

Since the additional term is vanishing, we can write out symbolically

```
In[•]:= CompTerm2Var1 = 24 Det[ExtrinsicKh[α, -β]]
```

$$\text{Out[•]=} \; 24\ \text{Det}\left[K_{\alpha}{}^{\alpha}{}_{\beta}\right]$$

```
In[•]:= KretschmannCD[] == CompTerm1 + CompTerm2Var1 + CompTerm3
        KretschVacuum3DDeco3 = %;
```

$$\text{Out[•]=} \; \mathcal{K} == 24\ \text{Det}\left[K_{\alpha}{}^{\alpha}{}_{\beta}\right] + 8\left(K^{\kappa}{}_{\lambda}\ K_{\nu\kappa} - K_{\alpha}{}^{\alpha}\ K_{\nu\lambda} + \in {}^{(3)}R_{\nu\lambda}\right)\left(K^{\gamma\lambda}\ K^{\nu}{}_{\gamma} - K^{\gamma}{}_{\gamma}\ K^{\nu\lambda} + \in {}^{(3)}R^{\nu\lambda}\right) + 4 \in \left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)\left(-K^{\lambda\nu|\kappa} + K^{\kappa\nu|\lambda}\right)$$

```
In[•]:= AddEquation@KretschVacuum3DDeco3;
        AddToList[$EquationsKretsch3DDeco, KretschVacuum3DDeco3];
```

Undefining dummy variables:

In[ ]:=
```
Clear@tmp
Clear@Sol
```

In[ ]:=
```
UndefChart[FM]
```

```
** UndefTensor: Undefined coordinate scalar t
** UndefTensor: Undefined coordinate scalar x
** UndefTensor: Undefined coordinate scalar y
** UndefTensor: Undefined coordinate scalar z
** UndefTensor: Undefined mapping differential tensor diFM
** UndefMapping: Undefined inverse mapping iFM
** UndefTensor: Undefined mapping differential tensor dFM
** UndefCovD: Undefined parallel derivative PDFM
** UndefTensor: Undefined antisymmetric +1 density etaUpFM
** UndefTensor: Undefined antisymmetric -1 density etaDownFM
```

Kretschmann scalar 3+1 decomposition summary in vacuum

In vacuum, considering Hamiltonian constraint & Normal change equations the decomposition is summarized by:

In[ ]:=
```
KretschVacuum3DDeco3
```

Out[ ]=
$$\mathcal{K} == 24\,\text{Det}\left[K^{\alpha}{}_{\beta}\right] + 8\left(K^{\kappa}{}_{\lambda}\;K_{\nu\kappa} - K^{\alpha}{}_{\alpha}\;K_{\nu\lambda} + \epsilon\;{}^{(3)}R_{\nu\lambda}\right)\left(K^{\gamma\lambda}\;K^{\nu}{}_{\gamma} - K^{\gamma}{}_{\gamma}\;K^{\nu\lambda} + \epsilon\;{}^{(3)}R^{\nu\lambda}\right) + 4\epsilon\left(-K_{\nu\lambda|\kappa} + K_{\nu\kappa|\lambda}\right)\left(-K^{\lambda\nu|\kappa} + K^{\kappa\nu|\lambda}\right)$$

Introducing the “Lapse” function

In this chapter we are going to further work upon the geometrical meaning of hypersurface-orthogonality, introduce a scalar function N, which for $\epsilon = -1$, i.e. normal is time-like case **(and only in this case!)**, is called **Lapse function**. It’s meaning is understood immediately as the dilation factor between observers experiencing the 4-velocity n^{μ} & the observers with global coordinate 4-velocity t^{μ}.

The factor N, the Lapse function

First off, since n_{μ} is hypersurface-orthogonal (remember, we defined Σ_t as the hypersurface orthogonal to the normal field at the very beginning), it has to be gradient of the scalar function t.

In[ ]:=
```
$EquationsNFactor = {};
```

We instantly run into a notation problem, as **N** is stored in WL as function for numerical value. For this purpose we use the capital Nu letter.

```
In[ ]:= ? N
```

```
Out[ ]= Symbol

N[expr] gives the numerical value of expr.

N[expr, n] attempts to give a result with n-digit precision.
```

Defining the N factor & the parameter t as scalar functions:

```
In[ ]:= DefTensor[N[], M, PrintAs → "N"];
        DefTensor[t[], M];
```

At this point, we have to specify, that any covariant derivative (PD in xact is just covariant derivative w/o torsion & curvature) of a Scalar tensor should assume the postfix notation (which is most likely a bug to be patched in some future releases), as the default option is prefix and cannot be changed.

```
In[ ]:= (*For this line I sincerely thank to Thomas Bäckdahl of the xAct team*)
        xAct`xTensor`Private`MakeBoxesCovD[der_[i_ /; Head[i] =!= Dir][head_], "Postfix"] :=
          Block[{$WarningFrom = "CovD Formatting"},
           SubsuperscriptBox[head, #1, #2] & @@ xAct`xTensor`Private`AddIndex[{"", ""}, i, der]];
```

```
In[ ]:= CD[-α]@N[]
```

```
Out[ ]= N_;α
```

```
In[ ]:= n[-μ] == ϵ * N[] * PD[-μ]@t[]
        NhypsurfOTGdef = %;
```

```
Out[ ]= n_μ == ϵ N t_,μ
```

```
In[ ]:= NhypsurfOTGdef[[2]] == NhypsurfOTGdef[[1]]
        % * (ϵ) / N[]
        PDtTOn = %;
```

```
Out[ ]= ϵ N t_,μ == n_μ
```

```
Out[ ]= t_,μ == ϵ n_μ / N
```

```
In[ ]:= AddEquation@NhypsurfOTGdef;
        AddToList[$EquationsNFactor, NhypsurfOTGdef];
        AddEquation@PDtTOn;
        AddToList[$EquationsNFactor, PDtTOn];
```

The antisymmetrical term $n_{\mu;\nu} - n_{\nu;\mu}$ can be calculated. Note that the covariant derivative can be exchanged for partial derivative due to symmetry of Christoffel symbols!

```
In[•]:= CD[-ν]@n[-μ] - CD[-μ]@n[-ν] == ChangeCovD[CD[-ν]@n[-μ] - CD[-μ]@n[-ν], CD]
        % // ToCanonical
        antisymn = %;
        %[[1]] == (%[[2]] /. ApplyRule@NhypsurfOTGdef);
        % // ToCanonical;
        % /. ApplyRule@PDtTOn
        antisymnTOPDgradN = %;
```

$$\text{Out[•]=}\quad -n_{\nu;\mu}+n_{\mu;\nu}==\Gamma[\nabla]^{\alpha}{}_{\mu\nu}\ n_{\alpha}-\Gamma[\nabla]^{\alpha}{}_{\nu\mu}\ n_{\alpha}-n_{\nu,\mu}+n_{\mu,\nu}$$

$$\text{Out[•]=}\quad -n_{\nu;\mu}+n_{\mu;\nu}==-n_{\nu,\mu}+n_{\mu,\nu}$$

```
** MakeRule: Potential problems moving indices on the LHS.
```

$$\text{Out[•]=}\quad -n_{\nu;\mu}+n_{\mu;\nu}==-\frac{n_{\nu}\ N_{,\mu}}{N}+\frac{n_{\mu}\ N_{,\nu}}{N}$$

```
In[•]:= AddEquation@antisymn;
        AddEquation@antisymnTOPDgradN;
        AddToList[$EquationsNFactor, antisymn];
        AddToList[$EquationsNFactor, antisymnTOPDgradN];

In[•]:= antisymnTOPDgradN * n[ν] // Expand
        % /. ApplyRule@NGRADNZERO
        % /. ApplyRule@NCDNTOAcc
        % /. ApplyRule@PDtTOn;
        PutScalar /@ %;
        % /. $Rules;
        NoScalar /@ %
        tmp = %;
```

$$\text{Out[•]=}\quad -n^{\nu}\ n_{\nu;\mu}+n^{\nu}\ n_{\mu;\nu}==-\frac{n_{\nu}\ n^{\nu}\ N_{,\mu}}{N}+\frac{n_{\mu}\ n^{\nu}\ N_{,\nu}}{N}$$

$$\text{Out[•]=}\quad n^{\nu}\ n_{\mu;\nu}==-\frac{n_{\nu}\ n^{\nu}\ N_{,\mu}}{N}+\frac{n_{\mu}\ n^{\nu}\ N_{,\nu}}{N}$$

$$\text{Out[•]=}\quad a_{\mu}==-\frac{n_{\nu}\ n^{\nu}\ N_{,\mu}}{N}+\frac{n_{\mu}\ n^{\nu}\ N_{,\nu}}{N}$$

```
** MakeRule: Potential problems moving indices on the LHS.
```

$$\text{Out[•]=}\quad a_{\mu}==\frac{n^{\alpha}\ n_{\mu}\ N_{,\alpha}}{N}-\frac{\in N_{,\mu}}{N}$$

Remembering that:

```
In[•]:= gTOh
        % - g[-μ, -ν] - ϵ n[-μ] × n[-ν]
        % * (-ϵ) // Expand
        nnTOghmetr = %;
```

$$\text{Out[•]= } g_{\mu\nu} == h_{\mu\nu} + \epsilon\ n_\mu\ \ n_\nu$$

$$\text{Out[•]= } -\epsilon\ n_\mu\ \ n_\nu == -g_{\mu\nu} + h_{\mu\nu}$$

$$\text{Out[•]= } n_\mu\ \ n_\nu == \epsilon\ g_{\mu\nu} - \epsilon\ h_{\mu\nu}$$

```
In[•]:= tmp /. ApplyRule@nnTOghmetr
        % // Expand
        AccTOgradLapse = %;
```

$$\text{Out[•]= } a_\mu == \frac{\left(\epsilon\ \delta_\mu{}^\alpha - \epsilon\ h_\mu{}^\alpha\right) N_{,\alpha}}{N} - \frac{\epsilon\ N_{,\mu}}{N}$$

$$\text{Out[•]= } a_\mu == -\frac{\epsilon\ h_\mu{}^\alpha\ N_{,\alpha}}{N}$$

```
In[•]:= Clear@tmp
```

```
In[•]:= AddEquation@AccTOgradLapse;
        AddToList[$EquationsNFactor, AccTOgradLapse];
```

```
In[•]:= ChangeCovD[AccTOgradLapse, PD, CD]
        AccTOCDgradLapse = %;
```

$$\text{Out[•]= } a_\mu == -\frac{\epsilon\ h_\mu{}^\alpha\ N_{;\alpha}}{N}$$

```
In[•]:= AddEquation@AccTOCDgradLapse;
        AddToList[$EquationsNFactor, AccTOCDgradLapse];
```

Note that the numerator is simply 3D covariant derivative of lapse (times the normal norm)

```
In[•]:= Accelerationn[-μ] == -ϵ cd[-μ]@N[] / N[]
        AccTOcdgradLapse = %;
```

$$\text{Out[•]= } a_\mu == -\frac{\epsilon\ N_{|\mu}}{N}$$

```
In[•]:= AddEquation@AccTOcdgradLapse;
        AddToList[$EquationsNFactor, AccTOcdgradLapse];
```

```
In[•]:= AccTOcdgradLapse
        % * N[] * -ϵ
        %[[2]] == %[[1]]
        cdgradLapseTOAcc = %;
```

$$\text{Out[•]= } a_\mu == -\frac{\epsilon\ N_{|\mu}}{N}$$

$$\text{Out[•]= } -\epsilon\ a_\mu\ N == N_{|\mu}$$

$$\text{Out[•]= } N_{|\mu} == -\epsilon\ a_\mu\ N$$

```
In[•]:= AddEquation@cdgradLapseTOAcc;
        AddToList[$EquationsNFactor, cdgradLapseTOAcc];
```

With this in mind, let us discuss how $a_{\alpha|\beta}$ factors out

```
In[•]:= -ϵ cd[-ν]@Accelerationn[-μ] ==
          (-ϵ cd[-ν]@Accelerationn[-μ] /. ApplyRule@AccTOcdgradLapse)
        %[[1]] == (%[[2, 1]] /. ApplyRule@cdgradLapseTOAcc) + %[[2, 2]]
        cdAccToAcccdcdN = %;
```

$$\text{Out[•]=}\quad -\epsilon\ a_{\mu|\nu} == -\frac{N_{|\mu}\,N_{|\nu}}{N^2} + \frac{N_{|\mu|\nu}}{N}$$

$$\text{Out[•]=}\quad -\epsilon\ a_{\mu|\nu} == -a_{\mu}\ a_{\nu} + \frac{N_{|\mu|\nu}}{N}$$

```
In[•]:= AddEquation@cdAccToAcccdcdN;
        AddToList[$EquationsNFactor, cdAccToAcccdcdN];
```

```
In[•]:= -cdAccToAcccdcdN - Accelerationn[-μ] × Accelerationn[-ν];
        % * ϵ // Expand
        ZeroVorticity1 = %;
```

$$\text{Out[•]=}\quad -\epsilon\ a_{\mu}\ \ a_{\nu} + a_{\mu|\nu} == -\frac{\epsilon\, N_{|\mu|\nu}}{N}$$

```
In[•]:= AddEquation@ZeroVorticity1;
        AddToList[$EquationsNFactor, ZeroVorticity1];
```

Due to zero torsion property of the RHS, the LHS is in fact symmetric. Look, the antisymmetric part of RHS vanishes:

```
In[•]:= ZeroVorticity1
        Antisymmetrize[ZeroVorticity1]
        DoesVanish = %[[2]];
```

$$\text{Out[•]=}\quad -\epsilon\ a_{\mu}\ \ a_{\nu} + a_{\mu|\nu} == -\frac{\epsilon\, N_{|\mu|\nu}}{N}$$

$$\text{Out[•]=}\quad \frac{1}{2}\left(-a_{\nu|\mu} + a_{\mu|\nu}\right) == \frac{1}{2}\left(\frac{\epsilon\, N_{|\nu|\mu}}{N} - \frac{\epsilon\, N_{|\mu|\nu}}{N}\right)$$

```
In[•]:= $EquationsNFactor
```

```
Out[•]= {Hold[NhypsurfOTGdef], Hold[PDtTOn], Hold[antisymn], Hold[antisymnTOPDgradN],
         Hold[AccTOgradLapse], Hold[AccTOCDgradLapse], Hold[AccTOcdgradLapse],
         Hold[cdgradLapseTOAcc], Hold[cdAccToAcccdcdN], Hold[ZeroVorticity1]}
```

In[ ]:=
```
DoesVanish
ChangeCovD[%, cd, PD]
% // Expand
(*The error here is irrelevant as we can see the identity holds*)
% // ToCanonical
```

Out[ ]=
$$\frac{1}{2}\left(\frac{\in \mathrm{N}_{|\nu|\mu}}{\mathrm{N}}-\frac{\in \mathrm{N}_{|\mu|\nu}}{\mathrm{N}}\right)$$

Out[ ]=
$$\frac{1}{2}\left(\frac{\in \left(-\Gamma[\mathrm{D}]^{\alpha}{}_{\mu\nu}\,\mathrm{N}_{,\alpha}+\mathrm{N}_{,\nu,\mu}\right)}{\mathrm{N}}-\frac{\in \left(-\Gamma[\mathrm{D}]^{\beta}{}_{\nu\mu}\,\mathrm{N}_{,\beta}+\mathrm{N}_{,\mu,\nu}\right)}{\mathrm{N}}\right)$$

Out[ ]=
$$-\frac{\in \Gamma[\mathrm{D}]^{\alpha}{}_{\mu\nu}\,\mathrm{N}_{,\alpha}}{2\,\mathrm{N}}+\frac{\in \Gamma[\mathrm{D}]^{\alpha}{}_{\nu\mu}\,\mathrm{N}_{,\alpha}}{2\,\mathrm{N}}+\frac{\in \mathrm{N}_{,\nu,\mu}}{2\,\mathrm{N}}-\frac{\in \mathrm{N}_{,\mu,\nu}}{2\,\mathrm{N}}$$

ToCanonical: Detected metric–incompatible derivatives {PD}.

ToCanonical: Detected metric–incompatible derivatives {PD}.

Out[ ]=
$$0$$

Therefore

In[ ]:=
```
ZeroVorticity1
```

Out[ ]=
$$-\in \mathrm{a}_{\mu}\;\mathrm{a}_{\nu}+\mathrm{a}_{\mu|\nu}==-\frac{\in \mathrm{N}_{|\mu|\nu}}{\mathrm{N}}$$

is fully symmetrical!

The very same holds for the normal, and is immediately seen from the definition of the Extrinsic curvature tensor.

In[ ]:=
```
ExtrinsicKToCDNAcc
%[[2]] == %[[1]]
Antisymmetrize[%] // ToCanonical
```

Out[ ]=
$$\mathrm{K}_{\beta\alpha}==-\in \mathrm{a}_{\alpha}\;\mathrm{n}_{\beta}+\mathrm{n}_{\alpha;\beta}$$

Out[ ]=
$$-\in \mathrm{a}_{\alpha}\;\mathrm{n}_{\beta}+\mathrm{n}_{\alpha;\beta}==\mathrm{K}_{\beta\alpha}$$

Out[ ]=
$$\frac{1}{2}\in \mathrm{a}_{\beta}\;\mathrm{n}_{\alpha}-\frac{1}{2}\in \mathrm{a}_{\alpha}\;\mathrm{n}_{\beta}-\frac{\mathrm{n}_{\beta;\alpha}}{2}+\frac{\mathrm{n}_{\alpha;\beta}}{2}==0$$

Or simpler still from:

```
In[•]:= ExtrinsicKToCDNAcc /. β → μ /. α → ν
        ExtrinsicKToLieD
        %%[[2]] == ExtrinsicKToLieD[[2]]
        % // ReleaseHold
        % // Antisymmetrize // ToCanonical
```

Out[•]= $K_{\mu\nu} == -\epsilon\, a_\nu\, n_\mu + n_{\nu;\mu}$

Out[•]= $K_{\mu\nu} == \frac{1}{2}\,\text{Hold}\left[\mathcal{L}_n\, h_{\mu\nu}\right]$

Out[•]= $-\epsilon\, a_\nu\, n_\mu + n_{\nu;\mu} == \frac{1}{2}\,\text{Hold}\left[\mathcal{L}_n\, h_{\mu\nu}\right]$

Out[•]= $-\epsilon\, a_\nu\, n_\mu + n_{\nu;\mu} == \frac{1}{2}\,\left(K_{\mu\nu} + K_{\nu\mu}\right)$

Out[•]= $-\frac{1}{2}\,\epsilon\, a_\nu\, n_\mu + \frac{1}{2}\,\epsilon\, a_\mu\, n_\nu + \frac{n_{\nu;\mu}}{2} - \frac{n_{\mu;\nu}}{2} == 0$

This is very important realization!

Without delving into this topic deeply, in general it holds that

$u_{\mu;\nu} - \epsilon\, a_\mu\, u_\nu = u_{(\mu;\nu)} - \epsilon\, a_{(\mu}\, u_{\nu)} + u_{[\mu;\nu]} - \epsilon\, a_{[\mu}\, u_{\nu]} := \Theta_{\mu\nu} + \omega_{\mu\nu}$.

Meaning of this is that the hypersurface orthogonality boils down to zero vorticity tensor $\omega_{\mu\nu} = h^\kappa_\mu\, h^\lambda_\nu\, u_{[\kappa;\lambda]} = u_{[\mu;\nu]} - \epsilon\, a_{[\mu}\, u_{\nu]} = 0$, thus the Expansion tensor $\Theta_{\mu\nu} = h^\kappa_\mu\, h^\lambda_\nu\, u_{(\kappa;\lambda)} = u_{(\mu;\nu)} - \epsilon\, a_{(\mu}\, u_{\nu)} = \frac{1}{2}\,\mathcal{L}_{\boldsymbol{u}}\, h_{\mu\nu}$ need not be symmetrized and in general describes the term $u_{\mu;\nu} - \epsilon\, a_\mu\, u_\nu$ fully.

Furthermore, note that for vanishing vorticity tensor it follows that $\Theta_{\mu\nu} \equiv K_{\mu\nu}$! So the Expansion tensor is in fact the Exterior curvature tensor!

- u_μ in this equation stands for n_μ or a_μ

Since hypersurface orthogonality implies $\omega_{\mu\nu} = 0$ & $\omega_{\mu\nu} = 0$ leads to “motion” of the normal being described by the Extrinsic curvature (Expansion), also trivial analogy to hydrodynamics is that u_μ does not “curl” along the normal shift. This (geometrically/intuitively) justifies us to even introduce the 3+1 decomposition in the first place.

Complete insight follows from the Frobenius theorem.

In[ ]:= **DisplayEQ@$EquationsNFactor**

Out[ ]//MatrixForm=

$$\begin{pmatrix}
n_\mu == \in N\, t_{,\mu} \\
t_{,\mu} == \frac{\in n_\mu}{N} \\
-n_{\nu;\mu} + n_{\mu;\nu} == -n_{\nu,\mu} + n_{\mu,\nu} \\
-n_{\nu;\mu} + n_{\mu;\nu} == -\frac{n_\nu N_{,\mu}}{N} + \frac{n_\mu N_{,\nu}}{N} \\
a_\mu == -\frac{\in h_\mu{}^\alpha N_{,\alpha}}{N} \\
a_\mu == -\frac{\in h_\mu{}^\alpha N_{;\alpha}}{N} \\
a_\mu == -\frac{\in N_{|\mu}}{N} \\
N_{|\mu} == -\in a_\mu\, N \\
-\in a_{\mu|\nu} == -a_\mu\, a_\nu + \frac{N_{|\mu|\nu}}{N} \\
-\in a_\mu\, a_\nu + a_{\mu|\nu} == -\frac{\in N_{|\mu|\nu}}{N}
\end{pmatrix}$$

Second foliation, 2+1+1 decomposition preparation

Now somewhat arduous task must be achieved, the second split has to be introduced. However, this time, it cannot be performed by **DefMetric** function. **DefMetric** function has an option to specify what vector induces the metric decomposition (**InducedFrom**), which is the previous approach for introducing 3+1 decomposition.
This option accepts only space time metric and a single (co)vector argument (previously the normal). This restrains the possibilities of how the second split can be introduced to following options:
• We can rework the definition of our space-time to be a product manifold of two 1D manifolds and a 2D manifold and (for convenience) switch to coordinate representation
• Introduce the required tensors (2D metric, 2D exterior curvature, ...) manually, equip it with appropriate symmetries and store every important properties such as raising/lowering indices as separate rule. This can only be done by introducing the 2D metric as 4D metric and then by lengthy set of rules ensuring it behaves exactly like twice induced metric (i.e. orthogonal to two vectors, induced from $g_{\mu\nu}$, describing a surface)

The structure of this notebook is already developed in general space time (without the product manifolds) and at this point it is not yet advantageous to immerse the quantities into particular coordinates (which is a topic of later chapters), thus the latter option is chosen.

Let us now have two congruences.
• First is already defined congruence given by some parameter t, and was deployed in the definition of Σ_t.
• Second congruence is chosen from hypersurface Σ_t, has parameter s, and defines a 2D surface Σ_{ts}. Let r^α be a vector field from tangent bundle of Σ_t. This vector field will act as normal to the surface Σ_{ts}, similarly to how n^α was defined.

Defining the normal of the Σ_{ts} hypersurface

```
In[•]:= $Equations2DNormal = {};
```

First off, similarly to what was already developed for the normal field n^α, a new vector field r^α is introduced and it's norm is stored in the following set of rules.
With $r^\alpha r_\alpha = \zeta$, it immediately follows that for $\epsilon = -1$, $\zeta = 1$.

```
In[•]:= DefConstantSymbol[ζ]
```

```
In[•]:= DefTensor[r[-α], M, OrthogonalTo → {n[α]}]
```

```
In[•]:= ζ /: Power[ζ, r_?EvenQ] := 1
        ζ /: Power[ζ, r_?OddQ] := ζ
```

```
In[•]:= r[α] × n[-α]
Out[•]= 0
```

```
In[•]:= ζ == Scalar@Abs[r[-μ] × r[μ]]
        DefineZ = %;
```

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

```
In[•]:= AddEquation@DefineZ;
        AddToList[$Equations2DNormal, DefineZ];
```

```
In[•]:= NoScalar /@ DefineZ
        DefineZNOSC = %;
```

$$\text{Out[•]= } \zeta == \text{Abs}\left[r_\mu \;\; r^\mu \right]$$

```
In[•]:= AddEquation@DefineZNOSC;
        AddToList[$Equations2DNormal, DefineZNOSC];
```

Again, following from the definition directly:

```
In[•]:= r[-α] × r[α] == ζ
        PutScalar /@ %
        NormToZNoABS = %;
```

$$\text{Out[•]= } r_\alpha \;\; r^\alpha == \zeta$$

$$\text{Out[•]= } \left(r_\alpha \;\; r^\alpha \right) == \zeta$$

```
In[•]:= AddEquation@NormToZNoABS;
        AddToList[$Equations2DNormal, NormToZNoABS];
```

```
In[•]:= AutomaticRules[r, ApplyRule@NormToζNoABS]
```

```
Rules {1, 2} have been declared as generic Rules.
```

```
In[•]:= $Rules
```

```
Out[•]= {HoldPattern[(n_α n^α)] :↦ Module[{}, ∈], HoldPattern[(n_α n^α)] :↦ Module[{}, ∈],
 HoldPattern[(r_α r^α)] :↦ Module[{}, ζ], HoldPattern[(r_α r^α)] :↦ Module[{}, ζ]}
```

A quick test whether the rules hold accordingly

```
In[•]:= r[α] × r[-α] × r[μ] × r[-μ] × r[γ] × r[-γ]
        % // PutScalar
        % /. $Rules
```

```
Out[•]= r_α r^α r_γ r^γ r_μ r^μ
```

```
Out[•]= (r_α r^α) (r_γ r^γ) (r_μ r^μ)
```

```
Out[•]= ζ
```

Induced surface metric definition:

```
In[•]:= $Equations2Dmetric = {};
```

The newly induced metric will be denoted by $f_{\mu\nu}$, and it is clearly orthogonal to both n^μ and r^μ by definition, as
$f_{\mu\nu} = g_{\mu\nu} - \epsilon\, n_\mu\, n_\nu - \zeta\, r_\mu\, r_\nu$

```
In[•]:= IM2Dmetdet = 1;
```

```
In[•]:= DefMetric[IM2Dmetdet, f[-μ, -ν], c2d, {"∥", "d𝕃"}]
```

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

** MakeRule: Potential problems moving indices on the LHS.

** MakeRule: Potential problems moving indices on the LHS.

** MakeRule: Potential problems moving indices on the LHS.
```

```
In[•]:= f[-μ, -ν] == g[-μ, -ν] - ϵ n[-μ] × n[-ν] - ζ r[-μ] × r[-ν]
        fTOg = %;
```

```
Out[•]= f_μν == g_μν - ϵ n_μ n_ν - ζ r_μ r_ν
```

```
In[•]:= AddEquation@fTOg;
        AddToList[$Equations2Dmetric, fTOg];
```

Firstly, $f_{\mu\nu}\, r^\mu = f_{\mu\nu}\, n^\mu = 0$

```
In[•]:= f[-μ, -ν] × r[μ] /. ApplyRule@fTOg
        % // Expand // ContractMetric
        (PutScalar /@ %) /. $Rules
```

$$\text{Out[•]= } r^{\mu}\ \left(g_{\nu\mu} - \epsilon\ n_{\mu}\ n_{\nu} - \zeta\ r_{\mu}\ r_{\nu}\right)$$

$$\text{Out[•]= } r_{\nu} - \zeta\ r_{\mu}\ r^{\mu}\ r_{\nu}$$

$$\text{Out[•]= } 0$$

```
In[•]:= f[-μ, -ν] × n[μ] /. ApplyRule@fTOg
        % // Expand // ContractMetric
        (PutScalar /@ %) /. $Rules
```

$$\text{Out[•]= } n^{\mu}\ \left(g_{\nu\mu} - \epsilon\ n_{\mu}\ n_{\nu} - \zeta\ r_{\mu}\ r_{\nu}\right)$$

$$\text{Out[•]= } n_{\nu} - \epsilon\ n_{\mu}\ n^{\mu}\ n_{\nu}$$

$$\text{Out[•]= } 0$$

We store these results and add them to the set of automatic rules

```
In[•]:= f[-μ, -ν] × r[μ] == 0
        frOTG = %;
        f[-μ, -ν] × n[μ] == 0
        fnOTG = %;
```

$$\text{Out[•]= } f_{\mu\nu}\ r^{\mu} == 0$$

$$\text{Out[•]= } f_{\mu\nu}\ n^{\mu} == 0$$

```
In[•]:= AddEquation@fnOTG;
        AddEquation@frOTG;
        AddToList[$Equations2Dmetric, fnOTG];
        AddToList[$Equations2Dmetric, frOTG];
```

```
In[•]:= AutomaticRules[f, ApplyRule@fnOTG]
```

```
   Rules {1, 2, 3, 4} have been declared as UpValues for f.
```

```
In[•]:= AutomaticRules[f, ApplyRule@frOTG]
```

```
   Rules {1, 2, 3, 4} have been declared as UpValues for f.
```

```
In[•]:= {f[-μ, -ν] × r[μ], f[μ, ν] × n[-μ]}
```

$$\text{Out[•]= } \{0, 0\}$$

```
In[•]:= Solve[fTOg // ScreenDollarIndices, g[-μ, -ν]] // Flatten;
        %[[1]] /. Rule → Equal
        gTOf = %;
```

$$\text{Out[•]= } g_{\mu\nu} == f_{\mu\nu} + \epsilon\ n_{\mu}\ n_{\nu} + \zeta\ r_{\mu}\ r_{\nu}$$

```
In[•]:= AddEquation@gTOf;
        AddToList[$Equations2Dmetric, gTOf];
```

```
In[•]:= f[-μ, -ν] == h[-μ, -ν] - ζ r[-μ] × r[-ν]
        fTOh = %;
```

```
Out[•]= f_μν == h_μν - ζ r_μ r_ν
```

```
In[•]:= AddEquation@fTOh;
        AddToList[$Equations2Dmetric, fTOh];
        AddToList[$Equations3Dmetric, fTOh];
```

```
In[•]:= Solve[fTOh // ScreenDollarIndices, h[-μ, -ν]] // Flatten
        %[[1]] /. Rule → Equal
        hTOf = %;
```

```
Out[•]= {h_μν → f_μν + ζ r_μ r_ν}
```

```
Out[•]= h_μν == f_μν + ζ r_μ r_ν
```

```
In[•]:= AddEquation@hTOf;
        AddToList[$Equations2Dmetric, hTOf];
        AddToList[$Equations3Dmetric, hTOf];
```

Some further properties arise immediately:

```
In[•]:= h[μ, ν] × r[-ν]
        % /. ApplyRule@hTOg
        % // Expand
        % // ContractMetric
```

```
Out[•]= h^μν r_ν
```

```
Out[•]= (g^νμ - ∈ n^μ n^ν) r_ν
```

```
Out[•]= g^νμ r_ν
```

```
Out[•]= r^μ
```

```
In[•]:= h[-μ, -ν] × r[μ] == r[-ν]
        hrTOr = %;
```

```
Out[•]= h_μν r^μ == r_ν
```

```
In[•]:= AddEquation@hrTOr;
        AddToList[$Equations2DNormal, hrTOr];
```

```
In[•]:= f[μ, -α] × h[α, -ν]
        % /. ApplyRule@fTOg
        % /. ApplyRule@hTOg;
        % // Expand;
        PutScalar /@ %;
        % /. $Rules
```

```
Out[•]= f^μ_α h^α_ν
```

```
Out[•]= h^α_ν (δ_α^μ - ∈ n_α n^μ - ζ r_α r^μ)
```

```
Out[•]= δ_ν^μ - ∈ n^μ n_ν - ζ r^μ r_ν
```

Separate rule for **fTOg** variation of one covariant one contravariant index is not yet stored (as Kronecker symbol is protected), however it is clearly seen that the result is

```
In[•]:= h[μ, -α] × f[α, -ν] == f[μ, -ν]
        hfTOf = %;
        f[μ, -ν] == h[μ, -α] × f[α, -ν]
        fTOhf = %;
```

Out[•]= $f^{\alpha}{}_{\nu}\ h^{\mu}{}_{\alpha} == f^{\mu}{}_{\nu}$

Out[•]= $f^{\mu}{}_{\nu} == f^{\alpha}{}_{\nu}\ h^{\mu}{}_{\alpha}$

```
In[•]:= AddEquation@hfTOf;
        AddEquation@fTOhf;
        AddToList[$Equations2Dmetric, hfTOf];
        AddToList[$Equations2Dmetric, fTOhf];
```

Contraction rules for $f^{\mu\nu}$ also have to be developed

```
In[•]:= f[α, -μ] × f[μ, β]
        % /. ApplyRule@fTOg
        % // Expand
        PutScalar /@ %
        % /. $Rules
        % // ContractMetric
        % /. ApplyRule@gTOf
```

Out[•]= $f^{\alpha}{}_{\mu}\ f^{\mu\beta}$

Out[•]= $\left(\delta_{\mu}{}^{\alpha} - \epsilon\ n^{\alpha}\ n_{\mu} - \zeta\ r^{\alpha}\ r_{\mu}\right)\left(g^{\beta\mu} - \epsilon\ n^{\beta}\ n^{\mu} - \zeta\ r^{\beta}\ r^{\mu}\right)$

Out[•]= $g^{\beta\alpha} - \epsilon\ n^{\alpha}\ n^{\beta} - \epsilon\ g^{\beta\mu}\ n^{\alpha}\ n_{\mu} + n^{\alpha}\ n^{\beta}\ n_{\mu}\ n^{\mu} - \zeta\ r^{\alpha}\ r^{\beta} - \zeta\ g^{\beta\mu}\ r^{\alpha}\ r_{\mu} + r^{\alpha}\ r^{\beta}\ r_{\mu}\ r^{\mu}$

Out[•]= $g^{\beta\alpha} - \epsilon\ n^{\alpha}\ n^{\beta} - \epsilon\ g^{\beta\mu}\ n^{\alpha}\ n_{\mu} - \zeta\ r^{\alpha}\ r^{\beta} - \zeta\ g^{\beta\mu}\ r^{\alpha}\ r_{\mu} + n^{\alpha}\ n^{\beta}\left(n_{\mu}\ n^{\mu}\right) + r^{\alpha}\ r^{\beta}\left(r_{\mu}\ r^{\mu}\right)$

Out[•]= $g^{\beta\alpha} - \epsilon\ g^{\beta\mu}\ n^{\alpha}\ n_{\mu} - \zeta\ g^{\beta\mu}\ r^{\alpha}\ r_{\mu}$

Out[•]= $g^{\beta\alpha} - \epsilon\ n^{\alpha}\ n^{\beta} - \zeta\ r^{\alpha}\ r^{\beta}$

Out[•]= $f^{\beta\alpha}$

```
In[•]:= f[α, -μ] × f[μ, β] == f[α, β]
        ffContrTOf = %;
```

Out[•]= $f^{\alpha}{}_{\mu}\ f^{\mu\beta} == f^{\alpha\beta}$

```
In[•]:= AddEquation@ffContrTOf;
        AddToList[$Equations2Dmetric, ffContrTOf];
```

Similarly as in 4D case, orthogonality of 4-Acceleration and 4-Velocity holds (for precisely the same reason - normalization).

```
In[•]:= cd[-ρ]@r[-μ] × r[μ] == 0
        RGRADRZERO = %;
```

Out[•]= $r^{\mu}\ r_{\mu\,|\,\rho} == 0$

```
In[•]:= AddEquation@RGRADRZERO;
        AddToList[$Equations2DNormal, RGRADRZERO];
```

Since $h_{\mu\nu}\, r^{\mu}\, r^{\nu} = \zeta$ it follows that

```
In[•]:= CD[-α]@h[-μ, -ν] (r[μ] × r[ν]) == -h[-μ, -ν] × CD[-α]@(r[μ] × r[ν])
        % /. ApplyRule@hTOg
        % // Expand
```

$$\text{Out[•]=}\quad r^{\mu}\ r^{\nu}\ h_{\mu\nu;\alpha} == -\,h_{\mu\nu}\left(r^{\nu}\ r^{\mu}{}_{;\alpha} + r^{\mu}\ r^{\nu}{}_{;\alpha}\right)$$

$$\text{Out[•]=}\quad -\in r^{\mu}\ r^{\nu}\left(n_{\nu}\ n_{\mu;\alpha} + n_{\mu}\ n_{\nu;\alpha}\right) == -\left(\left(g_{\nu\mu} - \in n_{\mu}\ n_{\nu}\right)\left(r^{\nu}\ r^{\mu}{}_{;\alpha} + r^{\mu}\ r^{\nu}{}_{;\alpha}\right)\right)$$

$$\text{Out[•]=}\quad 0 == -\,g_{\nu\mu}\ r^{\nu}\ r^{\mu}{}_{;\alpha} - g_{\nu\mu}\ r^{\mu}\ r^{\nu}{}_{;\alpha}$$

As we can see, LHS vanishes, thus RHS vanishes aswell, storing this result:

```
In[•]:= CD[-α]@h[-μ, -ν] (r[μ] × r[ν]) == 0
        rrCDh = %;
        h[-μ, -ν] × CD[-α]@(r[μ] × r[ν]) == 0
        hCDrr = %;
```

$$\text{Out[•]=}\quad r^{\mu}\ r^{\nu}\ h_{\mu\nu;\alpha} == 0$$

$$\text{Out[•]=}\quad h_{\mu\nu}\left(r^{\nu}\ r^{\mu}{}_{;\alpha} + r^{\mu}\ r^{\nu}{}_{;\alpha}\right) == 0$$

```
In[•]:= AddEquation@rrCDh;
        AddEquation@hCDrr;
        AddToList[$Equations2DNormal, rrCDh];
        AddToList[$Equations2DNormal, hCDrr];
```

Similarly

```
In[•]:= CD[-α]@f[-μ, -ν] × n[μ] × n[ν]
        % /. ApplyRule@fTOg
        % // Expand
        PutScalar /@ %;
        % /. $Rules;
        ContractMetric /@ %;
        % /. ApplyRule@NGRADNZERO
```

$$\text{Out[•]=}\quad n^{\mu}\ n^{\nu}\ f_{\mu\nu;\alpha}$$

$$\text{Out[•]=}\quad n^{\mu}\ n^{\nu}\left(-\in\left(n_{\nu}\ n_{\mu;\alpha} + n_{\mu}\ n_{\nu;\alpha}\right) - \zeta\left(r_{\nu}\ r_{\mu;\alpha} + r_{\mu}\ r_{\nu;\alpha}\right)\right)$$

$$\text{Out[•]=}\quad -\in n^{\mu}\ n_{\nu}\ n^{\nu}\ n_{\mu;\alpha} - \in n_{\mu}\ n^{\mu}\ n^{\nu}\ n_{\nu;\alpha}$$

$$\text{Out[•]=}\quad 0$$

This last result is LHS of $n^{\mu}\, n^{\nu}\, f_{\mu\nu;\alpha} = -f_{\mu\nu}(n^{\mu}\, n^{\nu})_{;\alpha}$, thus both sides vanish.

```
In[•]:= n[μ] × n[ν] × CD[-α]@f[-μ, -ν] == 0
        nnCDf = %;
        f[-μ, -ν] × CD[-α]@(n[μ] × n[ν]) == 0
        fCDnn = %;
```

Out[•]= $n^{\mu}\ n^{\nu}\ f_{\mu\nu;\alpha} == 0$

Out[•]= $f_{\mu\nu}\ \left(n^{\nu}\ n^{\mu}{}_{;\alpha} + n^{\mu}\ n^{\nu}{}_{;\alpha}\right) == 0$

```
In[•]:= AddEquation@nnCDf;
        AddEquation@fCDnn;
        AddToList[$Equations2Dmetric, nnCDf];
        AddToList[$Equations2Dmetric, fCDnn];
```

Defining the 2D acceleration, second fundamental form, and it's trace.

All the following identities are either the exact copy of already developed identities involving the normal acceleration, 3D second fundamental form and it's trace. However, xAct is not equipped for 2+1+1 split, so some of the functions such as **GradNormalToExtrinsicK** cannot be used this time.

2D acceleration:

```
In[•]:= DefTensor[b[-μ], M, OrthogonalTo → {r[μ], n[μ]}]
```

```
In[•]:= b[-μ] == cd[-ν]@r[-μ] × r[ν]
        Acc2DTOrcd3DDivr = %;
```

Out[•]= $b_{\mu} == r^{\nu}\ r_{\mu|\nu}$

```
In[•]:= AddEquation@Acc2DTOrcd3DDivr;
        AddToList[$Equations2DNormal, Acc2DTOrcd3DDivr];
```

```
In[•]:= r[ν] × cd[-ν]@r[-μ] == b[-μ]
        Acc2Ddef = %;
```

Out[•]= $r^{\nu}\ r_{\mu|\nu} == b_{\mu}$

```
In[•]:= AddEquation@Acc2Ddef;
        AddToList[$Equations2DNormal, Acc2DTOrcd3DDivr];
```

Now we will expand the 3D derivative of a 3D covector to r_{α}.

```
In[•]:= cd3DCovector
        % /. V → r
```

Out[•]= $V_{\alpha|\beta} == h^{\gamma}{}_{\alpha}\ h^{\delta}{}_{\beta}\ V_{\gamma;\delta}$

Out[•]= $r_{\alpha|\beta} == h^{\gamma}{}_{\alpha}\ h^{\delta}{}_{\beta}\ r_{\gamma;\delta}$

```
In[•]:= Acc2DTOrcd3DDivr
        % /. ApplyRule@(cd3DCovector /. V → r)
        % /. ApplyRule@hrTOr
        Acc2DDefCD = %;
```

Out[•]= $b_\mu == r^\nu \; r_{\mu|\nu}$

Out[•]= $b_\mu == h^\alpha{}_\mu \; h^\beta{}_\nu \; r^\nu \; r_{\alpha;\beta}$

Out[•]= $b_\mu == h^\alpha{}_\mu \; r^\beta \; r_{\alpha;\beta}$

```
In[•]:= AddEquation@Acc2DDefCD;
        AddToList[$Equations2DNormal, Acc2DDefCD];
```

```
In[•]:= Acc2DDefCD
        % /. ApplyRule@hTOg;
        % // Expand // ScreenDollarIndices
        % /. MakeRule[{n[α] × CD[-ν]@r[-α], -CD[-ν]@n[-α] × r[α]},
          ContractMetrics → True, MetricOn → All]
        % // GradNormalToExtrinsicK
        % // Expand
        Acc2DTOExtrinsicK = %;
```

Out[•]= $b_\mu == h^\alpha{}_\mu \; r^\beta \; r_{\alpha;\beta}$

Out[•]= $b_\mu == r^\alpha \; r_{\mu;\alpha} - \epsilon \; n^\alpha \; n_\mu \; r^\beta \; r_{\alpha;\beta}$

Out[•]= $b_\mu == r^\alpha \; r_{\mu;\alpha} + \epsilon \; n_\mu \; r^\alpha \; r^\beta \; n_{\alpha;\beta}$

Out[•]= $b_\mu == \epsilon \left(K_{\beta\alpha} + \epsilon \; a_\alpha \; n_\beta \right) n_\mu \; r^\alpha \; r^\beta + r^\alpha \; r_{\mu;\alpha}$

Out[•]= $b_\mu == \epsilon \; K_{\beta\alpha} \; n_\mu \; r^\alpha \; r^\beta + r^\alpha \; r_{\mu;\alpha}$

```
In[•]:= AddEquation@Acc2DTOExtrinsicK;
        AddToList[$Equations2DNormal, Acc2DTOExtrinsicK];
```

```
In[•]:= DisplayEQ@$Equations2DNormal
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \zeta == \text{Abs}\left[r_\mu \; r^\mu \right] \\ \zeta == \text{Abs}\left[r_\mu \; r^\mu \right] \\ \left(r_\alpha \; r^\alpha \right) == \zeta \\ h_{\mu\nu} \; r^\mu == r_\nu \\ r^\mu \; r_{\mu|\rho} == 0 \\ r^\mu \; r^\nu \; h_{\mu\nu;\alpha} == 0 \\ h_{\mu\nu} \left(r^\nu \; r^\mu{}_{;\alpha} + r^\mu \; r^\nu{}_{;\alpha} \right) == 0 \\ b_\mu == r^\nu \; r_{\mu|\nu} \\ b_\mu == r^\nu \; r_{\mu|\nu} \\ b_\mu == h^\alpha{}_\mu \; r^\beta \; r_{\alpha;\beta} \\ b_\mu == \epsilon \; K_{\beta\alpha} \; n_\mu \; r^\alpha \; r^\beta + r^\alpha \; r_{\mu;\alpha} \end{pmatrix}$$

In[•]:=
```
DisplayEQ@$Equations2Dmetric
```

Out[•]//MatrixForm=

$$\begin{pmatrix} f_{\mu\nu} == g_{\mu\nu} - \epsilon\, n_\mu\, n_\nu - \zeta\, r_\mu\, r_\nu \\ \text{True} \\ \text{True} \\ g_{\mu\nu} == f_{\mu\nu} + \epsilon\, n_\mu\, n_\nu + \zeta\, r_\mu\, r_\nu \\ f_{\mu\nu} == h_{\mu\nu} - \zeta\, r_\mu\, r_\nu \\ h_{\mu\nu} == f_{\mu\nu} + \zeta\, r_\mu\, r_\nu \\ f^{\alpha}{}_{\nu}\, h^{\mu}{}_{\alpha} == f^{\mu}{}_{\nu} \\ f^{\mu}{}_{\nu} == f^{\alpha}{}_{\nu}\, h^{\mu}{}_{\alpha} \\ f^{\alpha}{}_{\mu}\, f^{\mu\beta} == f^{\alpha\beta} \\ n^{\mu}\, n^{\nu}\, f_{\mu\nu;\alpha} == 0 \\ f_{\mu\nu}\left(n^{\nu}\, n^{\mu}{}_{;\alpha} + n^{\mu}\, n^{\nu}{}_{;\alpha}\right) == 0 \end{pmatrix}$$

2D Second fundamental form:

In[•]:=
```
$Equations2DExtCurv = {};
```

Similarly how the Extrinsic curvature tensor $K_{\mu\nu}$ was orthogonal to n^μ, the newly arising 2D Extrinsic curvature tensor is orthogonal to both n^μ and r^μ

In[•]:=
```
DefTensor[k[-μ, -ν], M, Symmetric[{-μ, -ν}], OrthogonalTo → {r[μ], n[μ]}]
```

First observation is that (obviously) $k_{\mu\alpha}$ contracts with $f^{\alpha}{}_{\nu}$

In[•]:=
```
k[-μ, -α] × f[α, -ν]
% /. ApplyRule@fTOg
% // Expand
```

Out[•]= $f^{\alpha}{}_{\nu}\, k_{\mu\alpha}$

Out[•]= $k_{\mu\alpha}\left(\delta_{\nu}{}^{\alpha} - \epsilon\, n^{\alpha}\, n_{\nu} - \zeta\, r^{\alpha}\, r_{\nu}\right)$

Out[•]= $k_{\mu\nu}$

In[•]:=
```
k[-μ, -α] × f[α, -ν] == k[-μ, -ν]
fkContr = %;
```

Out[•]= $f^{\alpha}{}_{\nu}\, k_{\mu\alpha} == k_{\mu\nu}$

In[•]:=
```
AddEquation@fkContr;
AddToList[$Equations2DExtCurv, fkContr];
```

In[•]:=
```
k[-μ, -ν] == cd[-β]@r[-α] × f[α, -μ] × f[β, -ν]
ExtrCurv2DDef = %;
```

Out[•]= $k_{\mu\nu} == f^{\alpha}{}_{\mu}\, f^{\beta}{}_{\nu}\, r_{\alpha|\beta}$

In[•]:=
```
AddEquation@ExtrCurv2DDef;
AddToList[$Equations2DExtCurv, ExtrCurv2DDef];
```

An adjustment using once

In[•]:=
```
fTOh
```

Out[•]= $f_{\mu\nu} == h_{\mu\nu} - \xi\ r_{\mu}\ \ r_{\nu}$

In[•]:=
```
k[-μ, -ν] == (h[α, -μ] - ξ r[α] × r[-μ]) f[β, -ν] × cd[-β]@r[-α]
% // Expand
% /. ApplyRule@RGRADRZERO
ExtrCurv2DDeff = %;
```

Out[•]= $k_{\mu\nu} == f^{\beta}{}_{\nu}\ \left(h^{\alpha}{}_{\mu} - \xi\ r^{\alpha}\ \ r_{\mu}\right)\ r_{\alpha|\beta}$

Out[•]= $k_{\mu\nu} == -\xi\ f^{\beta}{}_{\nu}\ \ r^{\alpha}\ \ r_{\mu}\ \ r_{\alpha|\beta} + f^{\beta}{}_{\nu}\ \ r_{\mu|\beta}$

Out[•]= $k_{\mu\nu} == f^{\beta}{}_{\nu}\ \ r_{\mu|\beta}$

In[•]:=
```
AddEquation@ExtrCurv2DDeff;
AddToList[$Equations2DExtCurv, ExtrCurv2DDeff];
```

In[•]:=
```
ExtrCurv2DDeff /. ApplyRule@fTOh
% // Expand
% /. ApplyRule@Acc2Ddef
ExtrCurv2DTOAcc2D = %;
```

Out[•]= $k_{\mu\nu} == \left(h^{\beta}{}_{\nu} - \xi\ r^{\beta}\ \ r_{\nu}\right)\ r_{\mu|\beta}$

Out[•]= $k_{\mu\nu} == -\xi\ r^{\beta}\ \ r_{\nu}\ \ r_{\mu|\beta} + r_{\mu|\nu}$

Out[•]= $k_{\mu\nu} == -\xi\ b_{\mu}\ \ r_{\nu} + r_{\mu|\nu}$

In[•]:=
```
AddEquation@ExtrCurv2DTOAcc2D;
AddToList[$Equations2DExtCurv, ExtrCurv2DTOAcc2D];
```

The last equation is in fact directly $\frac{1}{2}\,{}^{(3)}\mathcal{L}_r\, f_{\mu\nu}$, however this quantity will not be deployed directly. It is still worth a mentions, as this equation is a direct 3D (in fact 2D, as ${}^{(3)}\mathcal{L}_{\mathsf{r}}\ \mathsf{f}_{\mu\nu}$ is an element of (co)tangent space Σ_{nr}) analogy of

In[•]:=
```
ExtrinsicKToLieD
```

Out[•]= $K_{\mu\nu} == \frac{1}{2}\ \text{Hold}\left[\mathcal{L}_{\mathsf{n}}\ h_{\mu\nu}\right]$

In[•]:=
```
ExtrCurv2DDeff
% /. MakeRule[{cd[-β]@r[-μ], h[-β, σ] × h[-μ, ρ] × CD[-σ]@r[-ρ]}]
% /. ApplyRule@hfTOf
ExtrCurv2DDeffhCD = %;
```

Out[•]= $k_{\mu\nu} == f^{\beta}{}_{\nu}\ \ r_{\mu|\beta}$

Out[•]= $k_{\mu\nu} == f^{\beta}{}_{\nu}\ \ h_{\beta}{}^{\gamma}\ \ h_{\mu}{}^{\alpha}\ \ r_{\alpha;\gamma}$

Out[•]= $k_{\mu\nu} == f^{\beta}{}_{\nu}\ \ h_{\mu}{}^{\alpha}\ \ r_{\alpha;\beta}$

In[•]:=
```
AddEquation@ExtrCurv2DDeffhCD;
AddToList[$Equations2DExtCurv, ExtrCurv2DDeffhCD];
```

Now since

In[•]:=
```
hrTOr
```

Out[•]=
$$h_{\mu\nu}\ r^{\mu} == r_{\nu}$$

In[•]:=
```
CD[-β]@(h[μ, ν] × r[-μ]) == CD[-β]@r[ν]
CDhr = %;
```

Out[•]=
$$r_{\mu}\ h^{\mu\nu}{}_{;\beta} + h^{\mu\nu}\ r_{\mu;\beta} == r^{\nu}{}_{;\beta}$$

In[•]:=
```
AddEquation@CDhr;
AddToList[$Equations2DNormal, CDhr];
```

Or

In[•]:=
```
h[μ, ν] × CD[-β]@r[-μ] == CD[-β]@r[ν] - r[-μ] × CD[-β]@h[μ, ν]
hCDr = %;
```

Out[•]=
$$h^{\mu\nu}\ r_{\mu;\beta} == -r_{\mu}\ h^{\mu\nu}{}_{;\beta} + r^{\nu}{}_{;\beta}$$

In[•]:=
```
AddEquation@hCDr;
AddToList[$Equations2DNormal, hCDr];
```

In[•]:=
```
ExtrCurv2DDeffhCD // ScreenDollarIndices
% /. ApplyRule@hCDr // Expand
% /. ApplyRule@hTOg // Expand
% // GradNormalToExtrinsicK // Expand
ExtrCurv2DTOExtrinsicK = %;
```

Out[•]=
$$k_{\mu\nu} == f^{\beta}{}_{\nu}\ h_{\mu}{}^{\alpha}\ r_{\alpha;\beta}$$

Out[•]=
$$k_{\mu\nu} == -f^{\beta}{}_{\nu}\ r_{\alpha}\ h^{\alpha}{}_{\mu;\beta} + f^{\beta}{}_{\nu}\ r_{\mu;\beta}$$

Out[•]=
$$k_{\mu\nu} == \in f^{\beta}{}_{\nu}\ n_{\mu}\ r_{\alpha}\ n^{\alpha}{}_{;\beta} + f^{\beta}{}_{\nu}\ r_{\mu;\beta}$$

Out[•]=
$$k_{\mu\nu} == \in K_{\beta}{}^{\alpha}\ f^{\beta}{}_{\nu}\ n_{\mu}\ r_{\alpha} + f^{\beta}{}_{\nu}\ r_{\mu;\beta}$$

In[•]:=
```
AddEquation@ExtrCurv2DTOExtrinsicK;
AddToList[$Equations2DExtCurv, ExtrCurv2DTOExtrinsicK];
```

In[•]:=
```
k[-μ, μ] == f[μ, ν] × k[-μ, -ν]
ExtrCurv2DTRDef = %;
```

Out[•]=
$$k_{\mu}{}^{\mu} == f^{\mu\nu}\ k_{\mu\nu}$$

In[•]:=
```
AddEquation@ExtrCurv2DTRDef;
AddToList[$Equations2DExtCurv, ExtrCurv2DTRDef];
```

```
In[•]:= ExtrCurv2DTRDef
        %[[1]] == (%[[2]] /. ApplyRule@ExtrCurv2DDef)
        % /. ApplyRule@ffContrTOf;
        % /. ApplyRule@ffContrTOf;
        %[[1]] == (%[[2]] /. ApplyRule@fTOh // Expand)
        % /. ApplyRule@RGRADRZERO
        % // ContractMetric
        ExtrCurv2DTOcdDivr = %;
```

$$\text{Out[•]=}\quad k_{\mu}{}^{\mu} == f^{\mu\nu}\ k_{\mu\nu}$$

$$\text{Out[•]=}\quad k_{\mu}{}^{\mu} == f^{\alpha}{}_{\mu}\ f^{\beta}{}_{\nu}\ f^{\mu\nu}\ r_{\alpha|\beta}$$

$$\text{Out[•]=}\quad k_{\mu}{}^{\mu} == h^{\beta\alpha}\ r_{\alpha|\beta} - \zeta\ r^{\alpha}\ r^{\beta}\ r_{\alpha|\beta}$$

$$\text{Out[•]=}\quad k_{\mu}{}^{\mu} == h^{\beta\alpha}\ r_{\alpha|\beta}$$

$$\text{Out[•]=}\quad k_{\mu}{}^{\mu} == r^{\alpha}{}_{|\alpha}$$

```
In[•]:= AddEquation@ExtrCurv2DTOcdDivr;
        AddToList[$Equations3DExtCurv, ExtrCurv2DTOcdDivr];
```

At this point an equation developed earlier for induced divergence of a 3D entity is needed:

```
In[•]:= cd3DDivTOAccCD
        % /. V → r
```

$$\text{Out[•]=}\quad V^{\lambda}{}_{|\lambda} == \in\ a^{\beta}\ V_{\beta} + V^{\alpha}{}_{;\alpha}$$

$$\text{Out[•]=}\quad r^{\lambda}{}_{|\lambda} == \in\ a^{\beta}\ r_{\beta} + r^{\alpha}{}_{;\alpha}$$

```
In[•]:= ExtrCurv2DTOcdDivr /. ApplyRule@(cd3DDivTOAccCD /. V → r)
        ExtrCurv2DToAccCDDivR = %;
```

$$\text{Out[•]=}\quad k_{\mu}{}^{\mu} == \in\ a^{\alpha}\ r_{\alpha} + r^{\alpha}{}_{;\alpha}$$

```
In[•]:= AddEquation@ExtrCurv2DToAccCDDivR;
        AddToList[$Equations2DExtCurv, ExtrCurv2DToAccCDDivR];
```

Another note about orthogonality. Since $k^{\mu\nu}$ is tangent to Σ_{ts}, it follows that:

```
In[•]:= {b[μ] × r[-μ], k[-μ, -ν] × r[ν], b[μ] × n[-μ], k[-μ, -ν] × n[ν]}
Out[•]= {0, 0, 0, 0}
```

Lastly, the main result - a correspondence with **GradNormalToExtrinsicK** has to be found for 2D derivative.

```
In[•]:= ExtrCurv2DTOAcc2D // ScreenDollarIndices
        Solve[%, cd[-ν]@r[-μ]] // Flatten
        %[[1]] /. Rule → Equal
        Grad2DNormalTO2DExtrCurv = %;
```

Out[•]= $k_{\mu\nu} == -\zeta\ b_\mu\ r_\nu + r_{\mu|\nu}$

Out[•]= $\left\{ r_{\mu|\nu} \to k_{\mu\nu} + \zeta\ b_\mu\ r_\nu \right\}$

Out[•]= $r_{\mu|\nu} == k_{\mu\nu} + \zeta\ b_\mu\ r_\nu$

```
In[•]:= AddEquation@Grad2DNormalTO2DExtrCurv;
        AddToList[$Equations2DExtCurv, Grad2DNormalTO2DExtrCurv];
```

```
In[•]:= DisplayEQ@$Equations2DExtCurv
```

Out[•]//MatrixForm=

$$\begin{pmatrix} f^\alpha{}_\nu\ k_{\mu\alpha} == k_{\mu\nu} \\ k_{\mu\nu} == f^\alpha{}_\mu\ f^\beta{}_\nu\ r_{\alpha|\beta} \\ k_{\mu\nu} == f^\beta{}_\nu\ r_{\mu|\beta} \\ k_{\mu\nu} == -\zeta\ b_\mu\ r_\nu + r_{\mu|\nu} \\ k_{\mu\nu} == f^\beta{}_\nu\ h_\mu{}^\alpha\ r_{\alpha;\beta} \\ k_{\mu\nu} == \epsilon\ K_\beta{}^\alpha\ f^\beta{}_\nu\ n_\mu\ r_\alpha + f^\beta{}_\nu\ r_{\mu;\beta} \\ k_\mu{}^\mu == f^{\mu\nu}\ k_{\mu\nu} \\ k_\mu{}^\mu == \epsilon\ a^\alpha\ r_\alpha + r^\alpha{}_{;\alpha} \\ r_{\mu|\nu} == k_{\mu\nu} + \zeta\ b_\mu\ r_\nu \end{pmatrix}$$

Induced surface covariant derivative & Riemann tensor

```
In[•]:= $Equations2Dderivative = {};
```

```
In[•]:= DefTensor[B[-μ], M, OrthogonalTo → {n[μ], r[μ]}]
```

```
In[•]:= c2d[-β]@B[-α] == f[-α, μ] × f[-β, ν] × cd[-ν]@B[-μ]
        cd2DCovector = %;
```

Out[•]= $B_{\alpha\|\beta} == f_\alpha{}^\mu\ f_\beta{}^\nu\ B_{\mu|\nu}$

```
In[•]:= AddEquation@cd2DCovector;
        AddToList[$Equations2Dderivative, cd2DCovector];
```

```
In[•]:= cd2DCovector
```

Out[•]= $B_{\alpha\|\beta} == f_\alpha{}^\mu\ f_\beta{}^\nu\ B_{\mu|\nu}$

```
In[•]:= c2d[-β]@B[-α] == cd[-ν]@B[-μ] × f[-β, ν] (f[-α, μ] /. ApplyRule@fTOh)
        % // Expand
        % /. MakeRule[{r[μ] × cd[-ν]@B[-μ], -cd[-ν]@r[μ] × B[-μ]}]
        % /. ApplyRule@Grad2DNormalTO2DExtrCurv;
        % // Expand // ContractMetric
        % /. ApplyRule@fkContr
        cd2DCovectorTOExtrCurv2D = %;
```

$$\text{Out[•]= } B_{\alpha\|\beta} == f_\beta{}^\nu \left(h_\alpha{}^\mu - \zeta\, r_\alpha\, r^\mu\right) B_{\mu|\nu}$$

$$\text{Out[•]= } B_{\alpha\|\beta} == f_\beta{}^\nu\, B_{\alpha|\nu} - \zeta\, f_\beta{}^\nu\, r_\alpha\, r^\mu\, B_{\mu|\nu}$$

$$\text{Out[•]= } B_{\alpha\|\beta} == f_\beta{}^\nu\, B_{\alpha|\nu} + \zeta\, B_\gamma\, f_\beta{}^\nu\, r_\alpha\, r^\gamma{}_{|\nu}$$

$$\text{Out[•]= } B_{\alpha\|\beta} == \zeta\, B_\gamma\, f_\beta{}^\nu\, k^\gamma{}_\nu\, r_\alpha + f_\beta{}^\nu\, B_{\alpha|\nu}$$

$$\text{Out[•]= } B_{\alpha\|\beta} == \zeta\, B_\gamma\, k^\gamma{}_\beta\, r_\alpha + f_\beta{}^\nu\, B_{\alpha|\nu}$$

```
In[•]:= AddEquation@cd2DCovectorTOExtrCurv2D;
        AddToList[$Equations2Dderivative, cd2DCovectorTOExtrCurv2D];
```

Further adjustment is to be made by

```
In[•]:= cd3CDCovectorExtrK
```

$$\text{Out[•]= } V_{\alpha|\beta} == \in K^\gamma{}_\beta\, n_\alpha\, V_\gamma + h^\delta{}_\beta\, V_{\alpha;\delta}$$

```
In[•]:= cd2DCovectorTOExtrCurv2D /. ApplyRule@(cd3CDCovectorExtrK /. V → B)
        % // Expand
        % /. ApplyRule@hfTOf
        cd2DCovectorTOCDExtrKExtrCurv2D = %;
```

$$\text{Out[•]= } B_{\alpha\|\beta} == \zeta\, B_\gamma\, k^\gamma{}_\beta\, r_\alpha + f_\beta{}^\nu \left(\in B_\delta\, K^\delta{}_\nu\, n_\alpha + h^\gamma{}_\nu\, B_{\alpha;\gamma}\right)$$

$$\text{Out[•]= } B_{\alpha\|\beta} == \in B_\gamma\, K^\gamma{}_\nu\, f_\beta{}^\nu\, n_\alpha + \zeta\, B_\gamma\, k^\gamma{}_\beta\, r_\alpha + f_\beta{}^\nu\, h^\gamma{}_\nu\, B_{\alpha;\gamma}$$

$$\text{Out[•]= } B_{\alpha\|\beta} == \in B_\gamma\, K^\gamma{}_\nu\, f_\beta{}^\nu\, n_\alpha + \zeta\, B_\gamma\, k^\gamma{}_\beta\, r_\alpha + f^\gamma{}_\beta\, B_{\alpha;\gamma}$$

```
In[•]:= AddEquation@cd2DCovectorTOCDExtrKExtrCurv2D;
        AddToList[$Equations2Dderivative, cd2DCovectorTOCDExtrKExtrCurv2D];
```

Vectorial entities abide the very same equations (and they are immidiately accessible due to xAct pattern matching). For example:

```
In[•]:= c2d[β]@B[α] == (c2d[β]@B[α] /. ApplyRule@cd2DCovectorTOCDExtrKExtrCurv2D)
        ** MakeRule: Potential problems moving indices on the LHS.
```

$$\text{Out[•]= } B^{\alpha\|\beta} == \in B_\delta\, K^\delta{}_\gamma\, f^{\beta\gamma}\, n^\alpha + \zeta\, B_\gamma\, k^{\gamma\beta}\, r^\alpha + f^{\gamma\beta}\, B^\alpha{}_{;\gamma}$$

However for divergence terms it is worth expressing these identities (and a few more) explicitly:

```
In[•]:= c2d[-α]@B[α] == f[α, γ] × c2d[-α]@B[-γ]
        cd2DCovDivdef = %;
```

$$\text{Out[•]= } B^\alpha{}_{\|\alpha} == f^{\alpha\gamma}\, B_{\gamma\|\alpha}$$

```
In[•]:= AddEquation@cd2DCovDivdef;
        AddToList[$Equations2Dderivative, cd2DCovDivdef];

In[•]:= cd2DCovDivdef[[1]] == (cd2DCovDivdef[[2]] /. ApplyRule@cd2DCovector)
        (*Make rule problem involving indices is announced,
        however this is an artefact deriving in
          the fact that xact does not involve 2+1+1 splitting*)
        % /. ApplyRule@ffContrTOf
        % /. ApplyRule@ffContrTOf
        cd2DDIVTOfcd3D = %;

        ** MakeRule: Potential problems moving indices on the LHS.
```

Out[•]= $B^{\alpha}{}_{||\alpha} == f_{\alpha}{}^{\delta}\ f^{\alpha\gamma}\ f_{\gamma}{}^{\beta}\ B_{\beta|\delta}$

Out[•]= $B^{\alpha}{}_{||\alpha} == f_{\alpha}{}^{\gamma}\ f^{\alpha\beta}\ B_{\beta|\gamma}$

Out[•]= $B^{\alpha}{}_{||\alpha} == f^{\beta\alpha}\ B_{\alpha|\beta}$

```
In[•]:= AddEquation@cd2DDIVTOfcd3D;
        AddToList[$Equations2Dderivative, cd2DDIVTOfcd3D];

In[•]:= cd2DDIVTOfcd3D /. ApplyRule@fTOh
        % // Expand
        % /. MakeRule[{r[α] × cd[-β]@B[-α], -cd[-β]@r[α] × B[-α]}]
        % /. ApplyRule@Acc2Ddef
        % // ContractMetric
        cd2DDIVTOcd3DDIVAccCov = %;
```

Out[•]= $B^{\alpha}{}_{||\alpha} == \left(h^{\beta\alpha} - \zeta\ r^{\alpha}\ r^{\beta}\right)\ B_{\alpha|\beta}$

Out[•]= $B^{\alpha}{}_{||\alpha} == h^{\beta\alpha}\ B_{\alpha|\beta} - \zeta\ r^{\alpha}\ r^{\beta}\ B_{\alpha|\beta}$

Out[•]= $B^{\alpha}{}_{||\alpha} == \zeta\ B_{\beta}\ r^{\alpha}\ r^{\beta}{}_{|\alpha} + h^{\beta\alpha}\ B_{\alpha|\beta}$

Out[•]= $B^{\alpha}{}_{||\alpha} == \zeta\ b^{\alpha}\ B_{\alpha} + h^{\beta\alpha}\ B_{\alpha|\beta}$

Out[•]= $B^{\alpha}{}_{||\alpha} == \zeta\ b^{\alpha}\ B_{\alpha} + B^{\alpha}{}_{|\alpha}$

```
In[•]:= AddEquation@cd2DDIVTOcd3DDIVAccCov;
        AddToList[$Equations2Dderivative, cd2DDIVTOcd3DDIVAccCov];

In[•]:= cd2DDIVTOcd3DDIVAccCov /. ApplyRule@(cd3DDivTOAccCD /. V → B)
        cd2DDIVTOCDDIVAcc2DAcc3DCov = %;
```

Out[•]= $B^{\alpha}{}_{||\alpha} == \in a^{\alpha}\ B_{\alpha} + \zeta\ b^{\alpha}\ B_{\alpha} + B^{\alpha}{}_{;\alpha}$

```
In[•]:= AddEquation@cd2DDIVTOCDDIVAcc2DAcc3DCov;
        AddToList[$Equations2Dderivative, cd2DDIVTOCDDIVAcc2DAcc3DCov];
```

2D Riemann tensor

The Riemann tensor is given (abstractly) by already defined 2D (albeit introduced as another (full)spacetime derivative) as

```
In[•]:= Riemannc2d[α, β, γ, δ]
```

Out[•]= $R[\text{dl}]^{\alpha\beta\gamma\delta}$

```
In[•]:= {PrintAs[Riemannc2d] ^= "(2)R",
         PrintAs[Riccic2d] ^= "(2)R", PrintAs[RicciScalarc2d] ^= "(2)R"}
```

Out[•]= $\left\{{}^{(2)}R,\ {}^{(2)}R,\ {}^{(2)}R\right\}$

```
In[•]:= {Riemannc2d[-α, -β, -γ, -δ], Riccic2d[-μ, -ν], RicciScalarc2d[]}
```

Out[•]= $\left\{{}^{(2)}R_{\alpha\beta\gamma\delta},\ {}^{(2)}R_{\mu\nu},\ {}^{(2)}R\right\}$

The 2D-Riemann tensor definitions is given by:

```
In[•]:= Riemannc2d[-μ, -ν, -κ, -λ] × B[μ] == c2d[-λ]@c2d[-κ]@B[-ν] - c2d[-κ]@c2d[-λ]@B[-ν]
        Riemann2DCommutatordef = %;
```

Out[•]= $B^{\mu}\ {}^{(2)}R_{\mu\nu\kappa\lambda} == -B_{\nu||\lambda||\kappa} + B_{\nu||\kappa||\lambda}$

```
In[•]:= AddEquation@Riemann2DCommutatordef;
        AddToList[$Equations2Dderivative, Riemann2DCommutatordef];
```

Albeit, as Σ_{ts} is a surface, it's Riemann tensor is given by a single component (corresponding Ricci Scalar) as:

```
In[•]:= Riemannc2d[-μ, -ν, -κ, -λ] ==
         RicciScalarc2d[] (f[-μ, -κ] × f[-ν, -λ] - f[-μ, -λ] × f[-ν, -κ]) / 2
        Riemann2DRicci2Ddef = %;
```

Out[•]= $${}^{(2)}R_{\mu\nu\kappa\lambda} == \frac{1}{2}\left(-f_{\mu\lambda}\ f_{\nu\kappa} + f_{\mu\kappa}\ f_{\nu\lambda}\right)\ {}^{(2)}R$$

```
In[•]:= AddEquation@Riemann2DRicci2Ddef;
        AddToList[$Equations2Dderivative, Riemann2DRicci2Ddef];
```

```
In[•]:= B[μ] × f[-μ, λ] == B[λ]
        ContractVecf = %;
```

Out[•]= $B^{\mu}\ f_{\mu}{}^{\lambda} == B^{\lambda}$

```
In[•]:= AddEquation@ContractVecf;
        AddToList[$Equations2Dmetric];
```

Hence

```
In[•]:= Riemann2DRicci2Ddef * B[μ]
        % // Expand
        % /. ApplyRule@ContractVecf
        % /. ApplyRule@Riemann2DCommutatordef
        cd2DCommutTORicci = %;
```

$$\text{Out[•]= } B^{\mu}\ {}^{(2)}R_{\mu\nu\kappa\lambda} == \frac{1}{2}\ B^{\mu}\ \left(-f_{\mu\lambda}\ f_{\nu\kappa} + f_{\mu\kappa}\ f_{\nu\lambda}\right)\ {}^{(2)}R$$

$$\text{Out[•]= } B^{\mu}\ {}^{(2)}R_{\mu\nu\kappa\lambda} == -\frac{1}{2}\ B^{\mu}\ f_{\mu\lambda}\ f_{\nu\kappa}\ {}^{(2)}R + \frac{1}{2}\ B^{\mu}\ f_{\mu\kappa}\ f_{\nu\lambda}\ {}^{(2)}R$$

$$\text{Out[•]= } B^{\mu}\ {}^{(2)}R_{\mu\nu\kappa\lambda} == -\frac{1}{2}\ B_{\lambda}\ f_{\nu\kappa}\ {}^{(2)}R + \frac{1}{2}\ B_{\kappa}\ f_{\nu\lambda}\ {}^{(2)}R$$

$$\text{Out[•]= } -B_{\nu\,||\,\lambda\,||\,\kappa} + B_{\nu\,||\,\kappa\,||\,\lambda} == -\frac{1}{2}\ B_{\lambda}\ f_{\nu\kappa}\ {}^{(2)}R + \frac{1}{2}\ B_{\kappa}\ f_{\nu\lambda}\ {}^{(2)}R$$

```
In[•]:= AddEquation@cd2DCommutTORicci;
        AddToList[$Equations2Dderivative, cd2DCommutTORicci];
```

Obtaining the lost relations

For this chapter no list will be created, as it is desired to make all yet unobtained rules automatic.

In this chapter we have to “cover” for what xAct lacks - full 2+1+1 formalism. We have already introduced a metric, derivative and all the curvature tensors and “2D” 4-velocity. However now we have to instruct xAct on several important relations:

Following is:

• How metrics $f_{\mu\nu}$ act on any of the developed tensors

• What are all the orthogonal relations of all said developed tensors with respect to r^{α} & n^{α}

We will take inspiration from the 3+1 case, as the geometrical setup is the exact same.

The 2D metric

We have already obtained some rules previously

• $f_{\mu\nu}$ & the normal vectors r^{μ}, n^{ν} are orthogonal.

• how the metrics acts on itself and the 3D metric

```
In[•]:= ffContrTOf
        hfTOf
        fTOhf
```

$$\text{Out[•]= } f^{\alpha}{}_{\mu}\ f^{\mu\beta} == f^{\alpha\beta}$$

$$\text{Out[•]= } f^{\alpha}{}_{\nu}\ h^{\mu}{}_{\alpha} == f^{\mu}{}_{\nu}$$

$$\text{Out[•]= } f^{\mu}{}_{\nu} == f^{\alpha}{}_{\nu}\ h^{\mu}{}_{\alpha}$$

Metric action on the 2D extrinsic curvature tensor

```
In[•]:= h[μ, ν] × ExtrinsicKh[-μ, -α]
```

Out[•]= $K_{\alpha}^{\ \nu}$

```
In[•]:= f[μ, ν] × k[-μ, -α] == k[ν, -α]
        MA2DExtrCurv = %;
```

Out[•]= $f^{\mu\nu}\ k_{\mu\alpha} == k^{\nu}_{\ \alpha}$

```
In[•]:= AddEquation@MA2DExtrCurv;
```

```
In[•]:= AutomaticRules[k, ApplyRule@MA2DExtrCurv]
```

```
   Rules {1, 2, 3, 4, 5, 6, 7, 8} have been declared as UpValues for k.
```

And a test:

```
In[•]:= f[μ, ν] × k[-α, -μ]
```

Out[•]= $k^{\nu}_{\ \alpha}$

The action on “2D” 4-Acceleration

```
In[•]:= h[μ, ν] × Accelerationn[-ν]
```

Out[•]= a^{μ}

```
In[•]:= f[μ, ν] × b[-ν] == b[μ]
        MA2DAcc = %;
```

Out[•]= $b_{\nu}\ f^{\mu\nu} == b^{\mu}$

```
In[•]:= AddEquation@MA2DAcc;
```

```
In[•]:= AutomaticRules[b, ApplyRule@MA2DAcc]
```

```
   Rules {1, 2, 3, 4} have been declared as UpValues for b.
```

And a test:

```
In[•]:= f[-μ, ν] × b[μ]
```

Out[•]= b^{ν}

How the metric acts on the 2D Riemann tensor

```
In[•]:= Riemanncd[-μ, -ν, -κ, -λ] × h[μ, α]
```

Out[•]= $^{(3)}R^{\alpha}_{\ \nu\kappa\lambda}$

```
In[•]:= Riemannc2d[-μ, -ν, -κ, -λ] × f[μ, α] == Riemannc2d[α, -ν, -κ, -λ]
        MA2DRiemann1 = %;
```

Out[•]= $f^{\mu\alpha}\ {}^{(2)}R_{\mu\nu\kappa\lambda} == {}^{(2)}R^{\alpha}_{\ \nu\kappa\lambda}$

```
In[•]:= AddEquation@MA2DRiemann1;
```

```
In[•]:= AutomaticRules[Riemannc2d, ApplyRule@MA2DRiemann1]
```

```
   Rules {1, 2, 3, 4, 5, 6, 7, 8} have been declared as UpValues for Riemannc2d.
```

```
In[•]:= Riemannc2d[-α, β, -γ, δ] × f[-β, μ]
```

```
Out[•]= - (2)R^μ_αγ^δ
```

However the Ricci tensor is not produced:

```
In[•]:= Riemannc2d[-α, -β, -γ, -δ] × f[α, γ]
```

```
Out[•]= (2)R^γ_βγδ
```

```
In[•]:= Riemannc2d[-α, -β, -γ, -δ] × f[α, γ] == Riccic2d[-β, -δ]
        MA2DRiemann2 = %;
```

```
Out[•]= (2)R^γ_βγδ == (2)R_βδ
```

```
In[•]:= AddEquation@MA2DRiemann2;
        AutomaticRules[Riemannc2d, ApplyRule@MA2DRiemann2]
```

```
   Rules {1, 2, 3, 4} have been declared as DownValues for Riemannc2d.
```

Again a quick test:

```
In[•]:= Riemannc2d[-α, -β, -γ, -δ] × f[α, γ]
```

```
Out[•]= (2)R_βδ
```

```
In[•]:= Riemannc2d[-α, -β, -δ, -γ] × f[α, γ]
```

```
Out[•]= - (2)R_βδ
```

2D Ricci tensor contractions:

```
In[•]:= Riccicd[-μ, -ν] × h[μ, α]
```

```
Out[•]= (3)R^α_ν
```

```
In[•]:= Riccic2d[-μ, -ν] × f[μ, α] == Riccic2d[α, -ν]
        MA2DRicci1 = %;
```

```
Out[•]= f^μα (2)R_μν == (2)R^α_ν
```

```
In[•]:= AddEquation@MA2DRicci1;
        AutomaticRules[Riccic2d, ApplyRule@MA2DRicci1]
```

```
   Rules {1, 2, 3, 4, 5, 6, 7, 8} have been declared as UpValues for Riccic2d.
```

Again a quick test:

```
In[•]:= Riccic2d[-μ, -ν] × f[μ, -α]
```

```
Out[•]= (2)R_αν
```

Yet again the Ricci scalar does not arise:

```
In[•]:= Riccic2d[-μ, -ν] × f[μ, ν]
```

```
Out[•]= (2)R^ν_ν
```

```
In[•]:= Riccic2d[-μ, -ν] × f[μ, ν] == RicciScalarc2d[]
        MA2DRicci2 = %;
```

Out[•]= $^{(2)}R^{\nu}{}_{\nu} == {}^{(2)}R$

```
In[•]:= AddEquation@MA2DRicci2;
        AutomaticRules[Riccic2d, ApplyRule@MA2DRicci2]
```

```
Rules {1, 2} have been declared as DownValues for Riccic2d.
```

Test:

```
In[•]:= Riccic2d[-α, α]
```

Out[•]= $^{(2)}R$

Orthogonality relations w.r. to normal fields:

Again getting inspiration from 3+1 split.
Already obtained relations are:

```
In[•]:= Accelerationn[α] × n[-α]
```

Out[•]= 0

```
In[•]:= b[α] × r[-α]
```

Out[•]= 0

```
In[•]:= n[α] × ExtrinsicKh[-α, -β]
```

Out[•]= 0

```
In[•]:= r[α] × k[-α, -β]
```

Out[•]= 0

```
In[•]:= NGRADNZERO
```

Out[•]= $n^{\alpha}\ n_{\alpha;\nu} == 0$

```
In[•]:= RGRADRZERO
```

Out[•]= $r^{\mu}\ r_{\mu|\rho} == 0$

Considering now that every tensor existing co/tangent bundle of Σ_t is already orthogonal to n^{α}. Decomposition of Σ_t into Σ_{ts} & r^{α} direction should not change the orthogonality obtained.

```
In[•]:= r[-α] × n[α]
```

Out[•]= 0

```
In[•]:= f[α, β] × n[-β]
```

Out[•]= 0

For example the three main decomposition equations:

The gauss equation:

In[•]:= **GaussEquation**

Out[•]= $h_\alpha{}^{\iota}\ h_\beta{}^{\kappa}\ h_\gamma{}^{\lambda}\ h_\delta{}^{\mu}\ R_{\iota\kappa\lambda\mu} == \epsilon\ K_{\gamma\beta}\ K_{\delta\alpha} - \epsilon\ K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}$

In[•]:= **Riemanncd[-μ, -ν, -κ, -λ] × n[μ]**

Out[•]= **0**

In[•]:= **ExtrinsicKh[γ, β] × n[-β]**

Out[•]= **0**

Which was already obtained.

The Codazzi equation

In[•]:= **CodazziEquation**

Out[•]= $h_\beta{}^{\iota}\ h_\gamma{}^{\kappa}\ h_\delta{}^{\lambda}\ n^{\alpha}\ R_{\alpha\iota\kappa\lambda} == -K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}$

In[•]:= **cd[-γ]@ExtrinsicKh[-β, -δ] × n[γ]**

Out[•]= **0**

Was not yet obtained in 2D, let's develop it now.

In[•]:= **c2d[-γ]@k[-β, -δ] × r[γ] == 0**
rc2dgradk = %;

Out[•]= $r^{\gamma}\ k_{\beta\delta\|\gamma} == 0$

In[•]:= **AddEquation@rc2dgradk;**
AutomaticRules[k, ApplyRule@rc2dgradk]

```
** MakeRule: Potential problems moving indices on the LHS.

   Rules {1, 2} have been declared as generic Rules.
```

In[•]:= **$Rules**

Out[•]= $\{\text{HoldPattern}[(n_{\underline{\alpha}}\ n^{\underline{\alpha}})] :\to \text{Module}[\{\}, \epsilon], \text{HoldPattern}[(n_{\underline{\alpha}}\ n^{\underline{\alpha}})] :\to \text{Module}[\{\}, \epsilon],$
$\text{HoldPattern}[(r_{\underline{\alpha}}\ r^{\underline{\alpha}})] :\to \text{Module}[\{\}, \zeta], \text{HoldPattern}[(r_{\underline{\alpha}}\ r^{\underline{\alpha}})] :\to \text{Module}[\{\}, \zeta],$
$\text{HoldPattern}[r^{\underline{\gamma}}\ k^{\underline{\beta\delta}}{}_{\|\underline{\gamma}}] :\to \text{Module}[\{\}, 0], \text{HoldPattern}[r_{\underline{\gamma}}\ k^{\underline{\beta\delta\|\gamma}}] :\to \text{Module}[\{\}, 0]\}$

In[•]:= **{r[-γ] × c2d[γ]@k[-β, -δ],**
r[-γ] × c2d[γ]@k[-β, δ]}
% /. $Rules

Out[•]= $\{r_\gamma\ k_{\beta\delta}{}^{\|\gamma}, r_\gamma\ k_\beta{}^{\delta\|\gamma}\}$

Out[•]= **{0, 0}**

The Ricci equation

In[•]:= **RicciEquation2**

Out[•]= $h_{\beta}^{\ \iota}\ h_{\delta}^{\ \kappa}\ n^{\alpha}\ n^{\gamma}\ R_{\alpha\iota\gamma\kappa} == -\in a_{\beta}\ a_{\delta} - K_{\beta}^{\ \iota}\ K_{\delta\iota} + a_{\beta\,|\,\delta} - h_{\beta}^{\ \alpha}\ h_{\delta}^{\ \gamma}\ n^{\iota}\ K_{\alpha\gamma;\iota}$

In[•]:= **h[-β, α] × h[-δ, γ] × n[ι] × CD[-ι]@ExtrinsicKh[-α, -γ] × n[β]**

Out[•]= **0**

In[•]:= **f[-β, α] × f[-δ, γ] × r[ι] × cd[-ι]@k[-α, γ] × r[β]**

Out[•]= **0**

In[•]:= **n[δ] × cd[-δ]@Accelerationn[-β]**

Out[•]= **0**

In[•]:= **n[β] × cd[-δ]@Accelerationn[-β]**

Out[•]= **0**

In[•]:= **c2d[-δ]@b[-β] × r[δ] == 0**
rc2dgradb = %;

Out[•]= $r^{\delta}\ b_{\beta\,||\,\delta} == 0$

In[•]:= **AutomaticRules[b, ApplyRule@rc2dgradb]**

** MakeRule: Potential problems moving indices on the LHS.

Rules {1, 2} have been declared as generic Rules.

In[•]:= **r[-δ] × c2d[δ]@b[-β]**
% /. $Rules

Out[•]= $r_{\delta}\ b_{\beta}^{\ ||\delta}$

Out[•]= **0**

In[•]:= **c2d[-δ]@b[-β] × r[β] == 0**
c2dbrcontr = %;

Out[•]= $r^{\beta}\ b_{\beta\,||\,\delta} == 0$

In[•]:= **AutomaticRules[b, ApplyRule@c2dbrcontr]**

** MakeRule: Potential problems moving indices on the LHS.

Rules {1, 2} have been declared as generic Rules.

In[•]:= **r[-β] × c2d[δ]@b[β]**
% /. $Rules

Out[•]= $r_{\beta}\ b^{\beta\,||\,\delta}$

Out[•]= **0**

And finally, every (already) 3D tensor should remain orthogonal to n

In[•]:= `n[β] {b[-β], r[-β], f[α, -β], Riccic2d[-ν, -β], Riemannc2d[-α, -β, -γ, -δ], k[α, -β]}`

Out[•]= $\left\{0, 0, 0, n^{\beta}\ {}^{(2)}R_{\nu\beta}, n^{\beta}\ {}^{(2)}R_{\alpha\beta\gamma\delta}, 0\right\}$

In[•]:= `n[β] × Riccic2d[-α, -β] == 0`
`nRiccic2d1 = %;`

Out[•]= $n^{\beta}\ {}^{(2)}R_{\alpha\beta} == 0$

In[•]:= `AutomaticRules[Riccic2d, ApplyRule@nRiccic2d1]`

Rules {1, 2, 3, 4} have been declared as UpValues for Riccic2d.

In[•]:= `n[-β] × Riccic2d[-α, β]`

Out[•]= 0

In[•]:= `n[α] × Riemannc2d[-α, -β, -γ, -δ] == 0`
`nRiemannc2d1 = %;`

Out[•]= $n^{\alpha}\ {}^{(2)}R_{\alpha\beta\gamma\delta} == 0$

In[•]:= `AutomaticRules[Riemannc2d, ApplyRule@nRiemannc2d1]`

Rules {1, 2, 3, 4} have been declared as UpValues for Riemannc2d.

In[•]:= `n[β] × Riemannc2d[-α, -β, -γ, -δ]`

Out[•]= 0

In[•]:= `n[-α] × Riemannc2d[α, -β, -γ, -δ]`

Out[•]= 0

In[•]:= `n[γ] × Riemannc2d[-α, -β, -γ, -δ] == 0`
`nRiemannc2d2 = %;`

Out[•]= $n^{\gamma}\ {}^{(2)}R_{\alpha\beta\gamma\delta} == 0$

In[•]:= `AutomaticRules[Riemannc2d, ApplyRule@nRiemannc2d2]`

Rules {1, 2} have been declared as UpValues for Riemannc2d.

In[•]:= `n[δ] × Riemannc2d[-α, -β, -γ, -δ] == 0`
`nRiemannc2d3 = %;`

Out[•]= $n^{\delta}\ {}^{(2)}R_{\alpha\beta\gamma\delta} == 0$

In[•]:= `AutomaticRules[Riemannc2d, ApplyRule@nRiemannc2d3]`

Rules {1, 2} have been declared as UpValues for Riemannc2d.

In[•]:= `{n[α], n[-β], n[-δ], n[γ]} Riemannc2d[-α, β, -γ, δ]`

Out[•]= {0, 0, 0, 0}

2+1 decomposition of the 3D Riemann tensor:

```
In[•]:= $EquationsRiemann2DDeco = {};
```

Since $f_{\mu\nu}$ has not been introduced as an induced metric (again running to problems with the absence of 2+1+1 split in xAct), GaussCodazzi function cannot be used in the previous manner. That does not matter, however, as the structure between Spacetime and it's 3+1 induced split, and 3+1 split and it's corresponding induced 2+1 split is in terms of geometry (and indeed all relations concerning the decomposition - Gauss, Codazzi, Ricci equations) are exactly the same, only with substituting all the corresponding 3D ↔ 2D entities such as the Riemann tensors, Extrinsic curvature tensors, accelerations, normals and their norms. And of course, the derivatives.

Keeping in mind the structure of substitution, some substitutions have to come prior to other, due to orthogonality and symmetries of tensorial quantities involved in the decomposition expressions. An example of a poorly done substitution is:

```
In[•]:= CodazziEquation
        % /. f → h /. RiemannCD → Riemanncd /. ExtrinsicKh → k
        (*RHS vanishes due to orthogonality*)
```

Out[•]= $h_{\beta}^{\ \perp}\ h_{\gamma}^{\ \kappa}\ h_{\delta}^{\ \lambda}\ n^{\alpha}\ R_{\alpha\perp\kappa\lambda} == -K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}$

Out[•]= $0 == -k_{\beta\delta|\gamma} + k_{\beta\gamma|\delta}$

And yet another example of poor usage:

```
In[•]:= GaussEquation
        GaussEquation /. RiemannCD → Riemanncd /. Riemanncd → Riemannc2d
        (*Note that RiemannCD is switched for 3D Riemann in RHS,
        then both 3D Riemann tensors are switched for 2D Riemann tensors →
         the correct order is the exact opposite*)
```

Out[•]= $h_{\alpha}^{\ \perp}\ h_{\beta}^{\ \kappa}\ h_{\gamma}^{\ \lambda}\ h_{\delta}^{\ \mu}\ R_{\perp\kappa\lambda\mu} == \in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}$

Out[•]= ${}^{(2)}R_{\delta\gamma\beta\alpha} == \in K_{\gamma\beta}\ K_{\delta\alpha} - \in K_{\gamma\alpha}\ K_{\delta\beta} + {}^{(2)}R_{\delta\gamma\beta\alpha}$

Defining a general function to perform the substitution:

```
In[•]:= CorrespondenceRules[expression_] :=
         (expression) /. n → r /. Accelerationn → b /. h → f /. cd → c2d /. CD → cd /.
                  Riemanncd → Riemannc2d /. Riccicd → Riccic2d /.
                RicciScalarcd → RicciScalarc2d /. ∈ → ξ /. ExtrinsicKh → k /.
            RiemannCD → Riemanncd /. RicciCD → Riccicd /. RicciScalarCD → RicciScalarcd
```

And deploying this function to previously obtained Gauss, Codazzi & Ricci equations for 3+1 splitting of spacetime Riemann, Ricci and Ricci scalar tensors:

Gauss, Codazzi & Ricci equations for the 3D Riemann tensor

Gauss equation

```
In[•]:= GaussEquation
        CorrespondenceRules@GaussEquation
        Gauss2DEquation = %;
```

Out[•]= $h_\alpha{}^\iota \; h_\beta{}^\kappa \; h_\gamma{}^\lambda \; h_\delta{}^\mu \; R_{\iota\kappa\lambda\mu} == \epsilon \; K_{\gamma\beta} \; K_{\delta\alpha} - \epsilon \; K_{\gamma\alpha} \; K_{\delta\beta} + {}^{(3)}R_{\delta\gamma\beta\alpha}$

Out[•]= $f_\alpha{}^\iota \; f_\beta{}^\kappa \; f_\gamma{}^\lambda \; f_\delta{}^\mu \; {}^{(3)}R_{\iota\kappa\lambda\mu} == \zeta \; k_{\gamma\beta} \; k_{\delta\alpha} - \zeta \; k_{\gamma\alpha} \; k_{\delta\beta} + {}^{(2)}R_{\delta\gamma\beta\alpha}$

```
In[•]:= AddEquation@Gauss2DEquation;
        AddToList[$EquationsRiemann2DDeco, Gauss2DEquation];
```

Codazzi equation

```
In[•]:= CodazziEquation
        CorrespondenceRules@CodazziEquation
        Codazzi2DEquation = %;
```

Out[•]= $h_\beta{}^\iota \; h_\gamma{}^\kappa \; h_\delta{}^\lambda \; n^\alpha \; R_{\alpha\iota\kappa\lambda} == -K_{\beta\delta|\gamma} + K_{\beta\gamma|\delta}$

Out[•]= $f_\beta{}^\iota \; f_\gamma{}^\kappa \; f_\delta{}^\lambda \; r^\alpha \; {}^{(3)}R_{\alpha\iota\kappa\lambda} == -k_{\beta\delta||\gamma} + k_{\beta\gamma||\delta}$

```
In[•]:= AddEquation@Codazzi2DEquation;
        AddToList[$EquationsRiemann2DDeco, Codazzi2DEquation];
```

Ricci equation

```
In[•]:= RicciEquation1
        CorrespondenceRules@RicciEquation1
        Ricci2DEquation1 = %;
```

Out[•]= $h_\beta{}^\iota \; h_\delta{}^\kappa \; n^\alpha \; n^\gamma \; R_{\alpha\iota\gamma\kappa} == n^\alpha \; n^\gamma \; R_{\beta\alpha\delta\gamma}$

Out[•]= $f_\beta{}^\iota \; f_\delta{}^\kappa \; r^\alpha \; r^\gamma \; {}^{(3)}R_{\alpha\iota\gamma\kappa} == r^\alpha \; r^\gamma \; {}^{(3)}R_{\beta\alpha\delta\gamma}$

```
In[•]:= RicciEquation2
        CorrespondenceRules@RicciEquation2
        Ricci2DEquation2 = %;
```

Out[•]= $h_\beta{}^\iota \; h_\delta{}^\kappa \; n^\alpha \; n^\gamma \; R_{\alpha\iota\gamma\kappa} == -\epsilon \; a_\beta \; a_\delta - K_\beta{}^\iota \; K_{\delta\iota} + a_{\beta|\delta} - h_\beta{}^\alpha \; h_\delta{}^\gamma \; n^\iota \; K_{\alpha\gamma;\iota}$

Out[•]= $f_\beta{}^\iota \; f_\delta{}^\kappa \; r^\alpha \; r^\gamma \; {}^{(3)}R_{\alpha\iota\gamma\kappa} == -\zeta \; b_\beta \; b_\delta - k_\beta{}^\iota \; k_{\delta\iota} + b_{\beta||\delta} - f_\beta{}^\alpha \; f_\delta{}^\gamma \; r^\iota \; k_{\alpha\gamma|\iota}$

```
In[•]:= RicciEquation3
        CorrespondenceRules@RicciEquation3
        Ricci2DEquation3 = %;
```

Out[•]= $h_\beta{}^\iota \; h_\delta{}^\kappa \; n^\alpha \; n^\gamma \; R_{\alpha\iota\gamma\kappa} == -\epsilon \; a_\beta \; a_\delta + K_\beta{}^\iota \; K_{\delta\iota} + a_{\beta|\delta} - \mathcal{L}_n \, K_{\beta\delta}$

Out[•]= $f_\beta{}^\iota \; f_\delta{}^\kappa \; r^\alpha \; r^\gamma \; {}^{(3)}R_{\alpha\iota\gamma\kappa} == -\zeta \; b_\beta \; b_\delta + k_\beta{}^\iota \; k_{\delta\iota} + b_{\beta||\delta} - \mathcal{L}_r \, k_{\beta\delta}$

```
In[•]:= AddEquation@Ricci2DEquation1;
        AddEquation@Ricci2DEquation2;
        AddEquation@Ricci2DEquation3;
        AddToList[$EquationsRiemann2DDeco, Ricci2DEquation1];
        AddToList[$EquationsRiemann2DDeco, Ricci2DEquation2];
        AddToList[$EquationsRiemann2DDeco, Ricci2DEquation3];

In[•]:= DisplayEQ@$EquationsRiemann2DDeco
```

Out[•]//MatrixForm=

$$\begin{pmatrix} f_\alpha{}^\iota\ f_\beta{}^\kappa\ f_\gamma{}^\lambda\ f_\delta{}^\mu\ {}^{(3)}R_{\iota\kappa\lambda\mu} == \zeta\, k_{\gamma\beta}\ k_{\delta\alpha} - \zeta\, k_{\gamma\alpha}\ k_{\delta\beta} + {}^{(2)}R_{\delta\gamma\beta\alpha} \\ f_\beta{}^\iota\ f_\gamma{}^\kappa\ f_\delta{}^\lambda\ r^\alpha\ {}^{(3)}R_{\alpha\iota\kappa\lambda} == -k_{\beta\delta\,||\,\gamma} + k_{\beta\gamma\,||\,\delta} \\ f_\beta{}^\iota\ f_\delta{}^\kappa\ r^\alpha\ r^\gamma\ {}^{(3)}R_{\alpha\iota\gamma\kappa} == r^\alpha\ r^\gamma\ {}^{(3)}R_{\beta\alpha\delta\gamma} \\ f_\beta{}^\iota\ f_\delta{}^\kappa\ r^\alpha\ r^\gamma\ {}^{(3)}R_{\alpha\iota\gamma\kappa} == -\zeta\, b_\beta\ b_\delta - k_\beta{}^\iota\ k_{\delta\iota} + b_{\beta\,||\,\delta} - f_\beta{}^\alpha\ f_\delta{}^\gamma\ r^\iota\ k_{\alpha\gamma\,|\,\iota} \\ f_\beta{}^\iota\ f_\delta{}^\kappa\ r^\alpha\ r^\gamma\ {}^{(3)}R_{\alpha\iota\gamma\kappa} == -\zeta\, b_\beta\ b_\delta + k_\beta{}^\iota\ k_{\delta\iota} + b_{\beta\,||\,\delta} - \mathcal{L}_r\, k_{\beta\delta} \end{pmatrix}$$

Gauss, Codazzi & Ricci equations for the 3D Ricci tensor

```
In[•]:= $EquationsRicci2DDeco = {};
```

A quick reminder how to proceed without GaussCodazzi function is due.
Simply realizing that ${}^{(3)}R_{\mu\nu} = {}^{(3)}R_{\alpha\beta}\, h^\alpha_\mu\, h^\beta_\nu = {}^{(3)}R_{\alpha\beta}\left(f^\alpha_\mu + \xi\, r^\alpha\, r_\mu\right)\left(f^\beta_\nu + \xi\, r^\beta\, r_\nu\right)$ and expanding each term, accounting for symmetries gives us:

```
In[•]:= Riccicd[-ν, -λ] == Riccicd[-α, -β] (h[α, -ν] × h[β, -λ] /. ApplyRule@hTOf) // Expand
        Ricci2DDecodef = %;
```

Out[•]= ${}^{(3)}R_{\nu\lambda} == f^\alpha{}_\nu\ f^\beta{}_\lambda\ {}^{(3)}R_{\alpha\beta} + \zeta\, f^\alpha{}_\nu\ r^\beta\ r_\lambda\ {}^{(3)}R_{\alpha\beta} + \zeta\, f^\beta{}_\lambda\ r^\alpha\ r_\nu\ {}^{(3)}R_{\alpha\beta} + r^\alpha\ r^\beta\ r_\lambda\ r_\nu\ {}^{(3)}R_{\alpha\beta}$

Then, we would, again, need to decompose every term & deploy already stored equations. Let us not repeat this process in full detail.

Gauss equations:

```
In[•]:= AddEquation@Ricci2DDecodef;
        AddToList[$EquationsRicci2DDeco, Ricci2DDecodef];

In[•]:= CorrespondenceRules@(-RicciGaussEquation1)
        Ricci2DGaussEquation1 = %;
```

Out[•]= $\zeta\, f_\beta{}^\alpha\ f_\delta{}^\gamma\ {}^{(3)}R_{\alpha\gamma} == -\zeta\, b_\beta\ b_\delta - k^\alpha{}_\alpha\ k_{\beta\delta} + \zeta\, {}^{(2)}R_{\beta\delta} + b_{\beta\,||\,\delta} - f_\beta{}^\gamma\ f_\delta{}^\iota\ r^\alpha\ k_{\gamma\iota\,|\,\alpha}$

```
In[•]:= CorrespondenceRules@(-RicciGaussEquation2)
        Ricci2DGaussEquation2 = %;
```

Out[•]= $\zeta\, f_\beta{}^\alpha\ f_\delta{}^\gamma\ {}^{(3)}R_{\alpha\gamma} == -k^\alpha{}_\alpha\ k_{\beta\delta} + k_\beta{}^\alpha\ k_{\delta\alpha} + \zeta\, {}^{(2)}R_{\beta\delta} + f_\beta{}^\iota\ f_\delta{}^\kappa\ r^\alpha\ r^\gamma\ {}^{(3)}R_{\alpha\iota\gamma\kappa}$

```
In[•]:= AddEquation@Ricci2DGaussEquation1;
        AddEquation@Ricci2DGaussEquation2;
        AddToList[$EquationsRicci2DDeco, Ricci2DGaussEquation1];
        AddToList[$EquationsRicci2DDeco, Ricci2DGaussEquation2];
```

Codazzi equation:

```
In[•]:= RicciCodazziEquation
        CorrespondenceRules@RicciCodazziEquation
        Ricci2DCodazziEquation = %;
```

Out[•]= $h_{\delta}^{\alpha}\ n^{\beta}\ R_{\beta\alpha} == K_{\delta\ |\alpha}^{\ \alpha} - K^{\alpha}_{\ \alpha|\delta}$

Out[•]= $f_{\delta}^{\alpha}\ r^{\beta}\ {}^{(3)}R_{\beta\alpha} == k_{\delta\ ||\alpha}^{\ \alpha} - k^{\alpha}_{\ \alpha||\delta}$

```
In[•]:= AddEquation@Ricci2DCodazziEquation;
        AddToList[$EquationsRicci2DDeco, Ricci2DCodazziEquation];
```

Ricci equation:

```
In[•]:= RicciRicciEquation1
        CorrespondenceRules@RicciRicciEquation1
        Ricci2DRicciEquation1 = %;
```

Out[•]= $n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\epsilon\ a_{\alpha}\ a^{\alpha} - K_{\alpha}^{\ \beta}\ K^{\alpha}_{\ \beta} + a_{\alpha}^{\ |\alpha} - h^{\gamma\alpha}\ n^{\beta}\ K_{\alpha\gamma;\beta}$

Out[•]= $r^{\alpha}\ r^{\beta}\ {}^{(3)}R_{\alpha\beta} == -\zeta\ b_{\alpha}\ b^{\alpha} - k_{\alpha}^{\ \beta}\ k^{\alpha}_{\ \beta} + b_{\alpha}^{\ ||\alpha} - f^{\gamma\alpha}\ r^{\beta}\ k_{\alpha\gamma\,|\beta}$

```
In[•]:= RicciRicciEquation2
        CorrespondenceRules@RicciRicciEquation2
        Ricci2DRicciEquation2 = %;
```

Out[•]= $n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -\epsilon\ a_{\alpha}\ a^{\alpha} + K_{\alpha}^{\ \beta}\ K^{\alpha}_{\ \beta} + a_{\alpha}^{\ |\alpha} - h^{\alpha\beta}\ \left(\mathcal{L}_{n}\,K_{\alpha\beta}\right)$

Out[•]= $r^{\alpha}\ r^{\beta}\ {}^{(3)}R_{\alpha\beta} == -\zeta\ b_{\alpha}\ b^{\alpha} + k_{\alpha}^{\ \beta}\ k^{\alpha}_{\ \beta} + b_{\alpha}^{\ ||\alpha} - f^{\alpha\beta}\ \left(\mathcal{L}_{r}\,k_{\alpha\beta}\right)$

```
In[•]:= RicciRicciEquation3
        CorrespondenceRules@RicciRicciEquation3
        Ricci2DRicciEquation3 = %;
```

Out[•]= $n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} == -K_{\alpha}^{\ \beta}\ K^{\alpha}_{\ \beta} + a^{\alpha}_{\ ;\alpha} - n^{\beta}\ K_{\alpha\ ;\beta}^{\ \alpha}$

Out[•]= $r^{\alpha}\ r^{\beta}\ {}^{(3)}R_{\alpha\beta} == -k_{\alpha}^{\ \beta}\ k^{\alpha}_{\ \beta} + b^{\alpha}_{\ |\alpha} - r^{\beta}\ k_{\alpha\ |\beta}^{\ \alpha}$

```
In[•]:= AddEquation@Ricci2DRicciEquation1;
        AddEquation@Ricci2DRicciEquation2;
        AddEquation@Ricci2DRicciEquation3;
        AddToList[$EquationsRicci2DDeco, Ricci2DRicciEquation1];
        AddToList[$EquationsRicci2DDeco, Ricci2DRicciEquation2];
        AddToList[$EquationsRicci2DDeco, Ricci2DRicciEquation3];
```

Ricci scalar & 3D Ricci tensor ↔ 2D Ricci scalar relation

```
In[•]:= $EquationsRicciScalar2DDeco = {};
```

And similarly 2D Riemann tensor is fully determined by 2D curvature scalar

In[•]:= **Riemann2DRicci2Ddef**

Out[•]= $${}^{(2)}R_{\mu\nu\kappa\lambda} == \frac{1}{2}\left(-f_{\mu\lambda}\ f_{\nu\kappa} + f_{\mu\kappa}\ f_{\nu\lambda}\right)\ {}^{(2)}R$$

Thus the 2+1 decomposition reduces to relations between 3D Ricci tensor & 2D curvature scalar (!)
For the 3D Ricci scalar we obtain:

In[•]:=
```
-RicciScalar3DDecoEquation1
CorrespondenceRules@-RicciScalar3DDecoEquation1
RicciScalar2DDeco1 = %;
```

Out[•]= $$\epsilon\, R == -K_{\alpha\beta}\ K^{\alpha\beta} - K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta} + \epsilon\ {}^{(3)}R + 2\ a^{\alpha}{}_{;\alpha} - 2\ n^{\alpha}\ K^{\beta}{}_{\beta;\alpha}$$

Out[•]= $$\zeta\ {}^{(3)}R == -k_{\alpha\beta}\ k^{\alpha\beta} - k^{\alpha}{}_{\alpha}\ k^{\beta}{}_{\beta} + \zeta\ {}^{(2)}R + 2\ b^{\alpha}{}_{|\alpha} - 2\ r^{\alpha}\ k^{\beta}{}_{\beta|\alpha}$$

In[•]:=
```
AddEquation@RicciScalar2DDeco1;
AddToList[$EquationsRicciScalar2DDeco, RicciScalar2DDeco1];
```

In[•]:=
```
-RicciScalar3DDecoEquation2
CorrespondenceRules@%
RicciScalar2DDeco2 = %;
```

Out[•]= $$\epsilon\, R == K_{\alpha\beta}\ K^{\alpha\beta} - K^{\alpha}{}_{\alpha}\ K^{\beta}{}_{\beta} + 2\ n^{\alpha}\ n^{\beta}\ R_{\alpha\beta} + \epsilon\ {}^{(3)}R$$

Out[•]= $$\zeta\ {}^{(3)}R == k_{\alpha\beta}\ k^{\alpha\beta} - k^{\alpha}{}_{\alpha}\ k^{\beta}{}_{\beta} + 2\ r^{\alpha}\ r^{\beta}\ {}^{(3)}R_{\alpha\beta} + \zeta\ {}^{(2)}R$$

In[•]:=
```
AddEquation@RicciScalar2DDeco2;
AddToList[$EquationsRicciScalar2DDeco, RicciScalar2DDeco2];
```

To summarize the obtained 3D → 2D Ricci scalar decompositions read:

In[•]:= **DisplayEQ@$EquationsRicciScalar2DDeco**

Out[•]//MatrixForm=

$$\begin{pmatrix} \zeta\ {}^{(3)}R == -k_{\alpha\beta}\ k^{\alpha\beta} - k^{\alpha}{}_{\alpha}\ k^{\beta}{}_{\beta} + \zeta\ {}^{(2)}R + 2\ b^{\alpha}{}_{|\alpha} - 2\ r^{\alpha}\ k^{\beta}{}_{\beta|\alpha} \\ \zeta\ {}^{(3)}R == k_{\alpha\beta}\ k^{\alpha\beta} - k^{\alpha}{}_{\alpha}\ k^{\beta}{}_{\beta} + 2\ r^{\alpha}\ r^{\beta}\ {}^{(3)}R_{\alpha\beta} + \zeta\ {}^{(2)}R \end{pmatrix}$$

2+1 decompoisition of the 3D Kretschmann scalar:

In[•]:= **$EquationsKretschmann2DDeco = {};**

We wish to obtain the decomposition as:

In[•]:= **KretschScal3DIndepComp**

Out[•]= $${}^{(3)}R_{\kappa\lambda\mu\nu}\ {}^{(3)}R^{\kappa\lambda\mu\nu} == 4\ {}^{(3)}R_{\kappa\lambda}\ {}^{(3)}R^{\kappa\lambda} - {}^{(3)}R^{2}$$

Thus we need to find the decomposition of $${}^{(3)}R_{\kappa\lambda}\ {}^{(3)}R^{\kappa\lambda}$$

The decomposition of ${}^{(3)}R_{\kappa\lambda}\ {}^{(3)}R^{\kappa\lambda}$

Obtaining simplification of ${}^{(2)}\mathrm{R}_{\kappa\lambda}$

First let us look at two important pieces of information. Firstly

In[•]:=
```
f[-μ, μ] == 2
metric2Dtrace = %;
AddEquation@metric2Dtrace;
AddToList[$Equations2Dmetric, metric2Dtrace];
```

Out[•]= $\mathrm{f}_{\mu}^{\ \mu} == 2$

Secondly:

In[•]:=
```
Riemann2DRicci2Ddef
```

Out[•]= $${}^{(2)}\mathrm{R}_{\mu\nu\kappa\lambda} == \frac{1}{2}\left(-\mathrm{f}_{\mu\lambda}\ \mathrm{f}_{\nu\kappa} + \mathrm{f}_{\mu\kappa}\ \mathrm{f}_{\nu\lambda}\right)\ {}^{(2)}\mathrm{R}$$

In[•]:=
```
Riemann2DRicci2Ddef
f[μ, κ] * %
% // Expand;
% /. ApplyRule@ffContrTOf;
% /. ApplyRule@ffContrTOf;
% /. ApplyRule@metric2Dtrace
Ricci2DToRicciTensor2DIndepComp = %;
```

Out[•]= $${}^{(2)}\mathrm{R}_{\mu\nu\kappa\lambda} == \frac{1}{2}\left(-\mathrm{f}_{\mu\lambda}\ \mathrm{f}_{\nu\kappa} + \mathrm{f}_{\mu\kappa}\ \mathrm{f}_{\nu\lambda}\right)\ {}^{(2)}\mathrm{R}$$

Out[•]= $${}^{(2)}\mathrm{R}_{\nu\lambda} == \frac{1}{2}\ \mathrm{f}^{\mu\kappa}\ \left(-\mathrm{f}_{\mu\lambda}\ \mathrm{f}_{\nu\kappa} + \mathrm{f}_{\mu\kappa}\ \mathrm{f}_{\nu\lambda}\right)\ {}^{(2)}\mathrm{R}$$

Out[•]= $${}^{(2)}\mathrm{R}_{\nu\lambda} == \frac{1}{2}\ \mathrm{f}_{\nu\lambda}\ {}^{(2)}\mathrm{R}$$

Obtaining simplification of ${}^{(3)}R_{\kappa\lambda}\ {}^{(3)}R^{\kappa\lambda}$

In[•]:=
```
h[κ, γ] × h[λ, δ] == (h[κ, γ] × h[λ, δ] /. ApplyRule@hTOf)
h2FullExpansion = %;
```

Out[•]= $\mathrm{h}^{\kappa\gamma}\ \mathrm{h}^{\lambda\delta} == \left(\mathrm{f}^{\kappa\gamma} + \zeta\ \mathrm{r}^{\gamma}\ \mathrm{r}^{\kappa}\right)\ \left(\mathrm{f}^{\lambda\delta} + \zeta\ \mathrm{r}^{\delta}\ \mathrm{r}^{\lambda}\right)$

In[•]:=
```
AddEquation@h2FullExpansion;
AddToList[$Equations2Dmetric, h2FullExpansion];
```

```
In[•]:= Times[h2FullExpansion, Riccicd[-κ, -λ]]
        % // ToCanonical;
        Times[%, Riccicd[-γ, -δ]];
        % // ToCanonical
        Ricci3DsqFullDeco = %;
```

Out[•]= $^{(3)}R^{\delta\gamma} == \left(f^{\kappa\gamma} + \zeta\, r^{\gamma}\ r^{\kappa}\right)\left(f^{\lambda\delta} + \zeta\, r^{\delta}\ r^{\lambda}\right)\ ^{(3)}R_{\kappa\lambda}$

Out[•]= $^{(3)}R_{\gamma\delta}\ ^{(3)}R^{\gamma\delta} == f^{\gamma\delta}\ f^{\kappa\lambda}\ ^{(3)}R_{\gamma\kappa}\ ^{(3)}R_{\delta\lambda} + 2\,\zeta\, f^{\kappa\lambda}\ r^{\gamma}\ r^{\delta}\ ^{(3)}R_{\gamma\kappa}\ ^{(3)}R_{\delta\lambda} + r^{\gamma}\ r^{\delta}\ r^{\kappa}\ r^{\lambda}\ ^{(3)}R_{\gamma\delta}\ ^{(3)}R_{\kappa\lambda}$

```
In[•]:= AddEquation@Ricci3DsqFullDeco;
```

This approach does not promising, however further simplifications can be done using previously obtained relation between 2D Ricci tensor & 2D Ricci scalar. We need to prepare every term separately, as with the decomposition of the 4D Kretschmann scalar, and then rearrange the terms correctly.

Starting by terms 2 & 3, as term 1 will be worked upon much more deeply:

```
In[•]:= (ProjectWith[f][Riccicd[-γ, -κ] × r[γ]]) * f[κ, λ]
        % /. ApplyRule@Ricci2DCodazziEquation
        % // Expand;
        % /. MakeRule[{f[κ, λ] × c2d[-α]@k[-κ, α], c2d[-α]@k[λ, α]}];
        % /. MakeRule[{f[κ, λ] × c2d[-κ]@k[α, -α], c2d[λ]@k[α, -α]}];
        Times[(%), (ProjectWith[f][Riccicd[-δ, -λ] × r[δ]])]
        % /. ApplyRule@Ricci2DCodazziEquation
        Ricci3DsqTerm2 = %;
```

Out[•]= $f_{\kappa}^{\ \alpha}\ f^{\kappa\lambda}\ r^{\gamma}\ ^{(3)}R_{\gamma\alpha}$

Out[•]= $f^{\kappa\lambda}\left(k_{\kappa}^{\ \alpha}{}_{\|\alpha} - k^{\alpha}{}_{\alpha\|\kappa}\right)$

```
** MakeRule: Potential problems moving indices on the LHS.

** MakeRule: Potential problems moving indices on the LHS.
```

Out[•]= $f_{\lambda}^{\ \gamma}\ r^{\delta}\ ^{(3)}R_{\delta\gamma}\left(k^{\lambda\alpha}{}_{\|\alpha} - k^{\beta}{}_{\beta}{}^{\|\lambda}\right)$

Out[•]= $\left(k_{\lambda}^{\ \gamma}{}_{\|\gamma} - k^{\gamma}{}_{\gamma\|\lambda}\right)\left(k^{\lambda\alpha}{}_{\|\alpha} - k^{\beta}{}_{\beta}{}^{\|\lambda}\right)$

```
In[•]:= (Riccicd[-γ, -κ] × r[γ] × r[κ]) (Riccicd[-δ, -λ] × r[δ] × r[λ])
        % /. ApplyRule@Ricci2DRicciEquation1
        % /. ApplyRule@Ricci2DRicciEquation1
        Ricci3DsqTerm3 = %;
```

Out[•]= $r^{\gamma}\ r^{\delta}\ r^{\kappa}\ r^{\lambda}\ ^{(3)}R_{\gamma\kappa}\ ^{(3)}R_{\delta\lambda}$

Out[•]= $r^{\delta}\ r^{\lambda}\ ^{(3)}R_{\delta\lambda}\left(-\zeta\ b_{\alpha}\ b^{\alpha} - k_{\alpha}^{\ \beta}\ k^{\alpha}{}_{\beta} + b_{\alpha}{}^{\|\alpha} - f^{\kappa\gamma}\ r^{\iota}\ k_{\gamma\kappa|\iota}\right)$

Out[•]= $\left(-\zeta\ b_{\alpha}\ b^{\alpha} - k_{\alpha}^{\ \beta}\ k^{\alpha}{}_{\beta} + b_{\alpha}{}^{\|\alpha} - f^{\iota\gamma}\ r^{\delta}\ k_{\gamma\iota|\delta}\right)\left(-\zeta\ b_{\kappa}\ b^{\kappa} - k_{\kappa}^{\ \lambda}\ k^{\kappa}{}_{\lambda} + b_{\kappa}{}^{\|\kappa} - f^{\rho\mu}\ r^{\nu}\ k_{\mu\rho|\nu}\right)$

Now for the first term:

```
In[•]:= (ProjectWith[f]@Riccicd[-γ, -κ])
        % /. ApplyRule@(ζ Ricci2DGaussEquation1);
        % /. ApplyRule@Ricci2DToRicciTensor2DIndepComp
        Times[%, f[γ, κ]]
        % // Expand;
        % /. ApplyRule@ffContrTOf
        % /. MakeRule[{f[γ, κ] × c2d[-κ]@b[-γ], c2d[-κ]@b[κ]}];
        % /. ApplyRule@ffContrTOf;
        % /. ApplyRule@metric2Dtrace
        tmp1 = %;
```

$$\text{Out[•]=}\quad f_{\gamma}{}^{\alpha}\ f_{\kappa}{}^{\beta}\ {}^{(3)}R_{\alpha\beta}$$

$$\text{Out[•]=}\quad -b_{\gamma}\ b_{\kappa}-\zeta\ k^{\alpha}{}_{\alpha}\ k_{\gamma\kappa}+\frac{1}{2}\ f_{\gamma\kappa}\ {}^{(2)}R+\zeta\ b_{\gamma\shortparallel\kappa}-\zeta\ f_{\gamma}{}^{\beta}\ f_{\kappa}{}^{\delta}\ r^{\alpha}\ k_{\beta\delta\,|\,\alpha}$$

$$\text{Out[•]=}\quad f^{\gamma\kappa}\ \left(-b_{\gamma}\ b_{\kappa}-\zeta\ k_{\gamma\kappa}\ k^{\iota}{}_{\iota}+\frac{1}{2}\ f_{\gamma\kappa}\ {}^{(2)}R+\zeta\ b_{\gamma\shortparallel\kappa}-\zeta\ f_{\gamma}{}^{\beta}\ f_{\kappa}{}^{\delta}\ r^{\alpha}\ k_{\beta\delta\,|\,\alpha}\right)$$

$$\text{Out[•]=}\quad -b_{\gamma}\ b^{\gamma}-\zeta\ k^{\alpha}{}_{\alpha}\ k^{\kappa}{}_{\kappa}+\frac{1}{2}\ f_{\gamma}{}^{\gamma}\ {}^{(2)}R+\zeta\ f^{\gamma\kappa}\ b_{\gamma\shortparallel\kappa}-\zeta\ f_{\gamma}{}^{\beta}\ f^{\gamma\delta}\ r^{\alpha}\ k_{\beta\delta\,|\,\alpha}$$

```
        ** MakeRule: Potential problems moving indices on the LHS.
```

$$\text{Out[•]=}\quad -b_{\gamma}\ b^{\gamma}-\zeta\ k^{\alpha}{}_{\alpha}\ k^{\kappa}{}_{\kappa}+{}^{(2)}R+\zeta\ b^{\alpha}{}_{\shortparallel\alpha}-\zeta\ f^{\beta\gamma}\ r^{\alpha}\ k_{\beta\gamma\,|\,\alpha}$$

```
In[•]:= (ProjectWith[f]@Riccicd[-δ, -λ])
        % /. ApplyRule@(ζ Ricci2DGaussEquation1);
        % /. ApplyRule@Ricci2DToRicciTensor2DIndepComp;
        Times[%, f[δ, λ]]
        % // Expand;
        % /. ApplyRule@ffContrTOf
        % /. MakeRule[{f[γ, κ] × c2d[-κ]@b[-γ], c2d[-κ]@b[κ]}];
        % /. ApplyRule@ffContrTOf;
        % /. ApplyRule@metric2Dtrace
        tmp2 = %;
```

$$\text{Out[•]=}\quad f_{\delta}{}^{\alpha}\ f_{\lambda}{}^{\beta}\ {}^{(3)}R_{\alpha\beta}$$

$$\text{Out[•]=}\quad f^{\delta\lambda}\ \left(-b_{\delta}\ b_{\lambda}-\zeta\ k_{\delta\lambda}\ k^{\iota}{}_{\iota}+\frac{1}{2}\ f_{\delta\lambda}\ {}^{(2)}R+\zeta\ b_{\delta\shortparallel\lambda}-\zeta\ f_{\delta}{}^{\beta}\ f_{\lambda}{}^{\gamma}\ r^{\alpha}\ k_{\beta\gamma\,|\,\alpha}\right)$$

$$\text{Out[•]=}\quad -b_{\delta}\ b^{\delta}-\zeta\ k^{\alpha}{}_{\alpha}\ k^{\lambda}{}_{\lambda}+\frac{1}{2}\ f_{\delta}{}^{\delta}\ {}^{(2)}R+\zeta\ f^{\delta\lambda}\ b_{\delta\shortparallel\lambda}-\zeta\ f_{\delta}{}^{\beta}\ f^{\delta\gamma}\ r^{\alpha}\ k_{\beta\gamma\,|\,\alpha}$$

```
        ** MakeRule: Potential problems moving indices on the LHS.
```

$$\text{Out[•]=}\quad -b_{\delta}\ b^{\delta}-\zeta\ k^{\alpha}{}_{\alpha}\ k^{\lambda}{}_{\lambda}+{}^{(2)}R+\zeta\ b^{\alpha}{}_{\shortparallel\alpha}-\zeta\ f^{\beta\gamma}\ r^{\alpha}\ k_{\beta\gamma\,|\,\alpha}$$

```
In[•]:= Ricci3DsqTerm1 = tmp2 * tmp1
```

$$\text{Out[•]=}\quad \left(-b_{\gamma}\ b^{\gamma}-\zeta\ k^{\kappa}{}_{\kappa}\ k^{\mu}{}_{\mu}+{}^{(2)}R+\zeta\ b^{\nu}{}_{\shortparallel\nu}-\zeta\ f^{\beta\iota}\ r^{\alpha}\ k_{\beta\iota\,|\,\alpha}\right)$$
$$\left(-b_{\delta}\ b^{\delta}-\zeta\ k^{\lambda}{}_{\lambda}\ k^{\chi}{}_{\chi}+{}^{(2)}R+\zeta\ b^{\chi 1}{}_{\shortparallel\chi 1}-\zeta\ f^{\sigma\upsilon}\ r^{\rho}\ k_{\sigma\upsilon\,|\,\rho}\right)$$

In[•]:=
```
Riccicd[-ν, -λ] × Riccicd[ν, λ] == Ricci3DsqTerm1 + 2 ζ Ricci3DsqTerm2 + Ricci3DsqTerm3
```

Out[•]=
$$
{}^{(3)}R_{\nu\lambda}\ {}^{(3)}R^{\nu\lambda} ==
$$
$$
2\,\zeta\,\left(k_{\lambda}{}^{\gamma}{}_{\|\gamma}-k^{\gamma}{}_{\gamma\|\lambda}\right)\left(k^{\lambda\alpha}{}_{\|\alpha}-k^{\beta}{}_{\beta}{}^{\|\lambda}\right)+\left(-\zeta\,b_{\alpha}\;b^{\alpha}-k_{\alpha}{}^{\beta}\;k^{\alpha}{}_{\beta}+b_{\alpha}{}^{\|\alpha}-f^{\iota\gamma}\;r^{\delta}\;k_{\gamma\iota\,|\,\delta}\right)
$$
$$
\left(-\zeta\,b_{\kappa}\;b^{\kappa}-k_{\kappa}{}^{\lambda}\;k^{\kappa}{}_{\lambda}+b_{\kappa}{}^{\|\kappa}-f^{\rho\mu}\;r^{\nu}\;k_{\mu\rho\,|\,\nu}\right)+
$$
$$
\left(-b_{\gamma}\;b^{\gamma}-\zeta\,k^{\kappa}{}_{\kappa}\;k^{\mu}{}_{\mu}+{}^{(2)}R+\zeta\,b^{\nu}{}_{\|\nu}-\zeta\,f^{\beta\iota}\;r^{\alpha}\;k_{\beta\iota\,|\,\alpha}\right)
$$
$$
\left(-b_{\delta}\;b^{\delta}-\zeta\,k^{\lambda}{}_{\lambda}\;k^{\chi}{}_{\chi}+{}^{(2)}R+\zeta\,b^{\chi 1}{}_{\|\chi 1}-\zeta\,f^{\sigma\upsilon}\;r^{\rho}\;k_{\sigma\upsilon\,|\,\rho}\right)
$$

In[•]:=
```
Ricci3DsqFinalDecomposition = %;
```

In[•]:=
```
AddEquation@Ricci3DsqFinalDecomposition;
```

In[•]:=
```
AddToList[$EquationsKretschmann2DDeco, Ricci3DsqFinalDecomposition];
```

Which is the final result of this chapter.