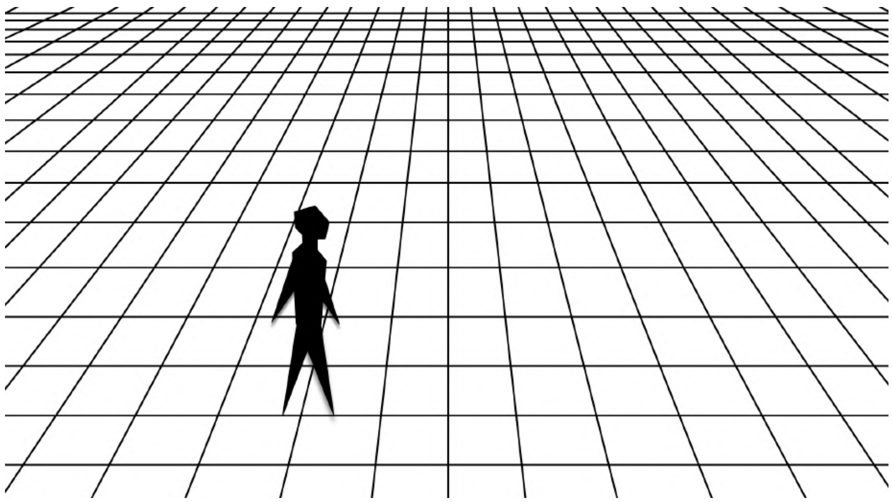
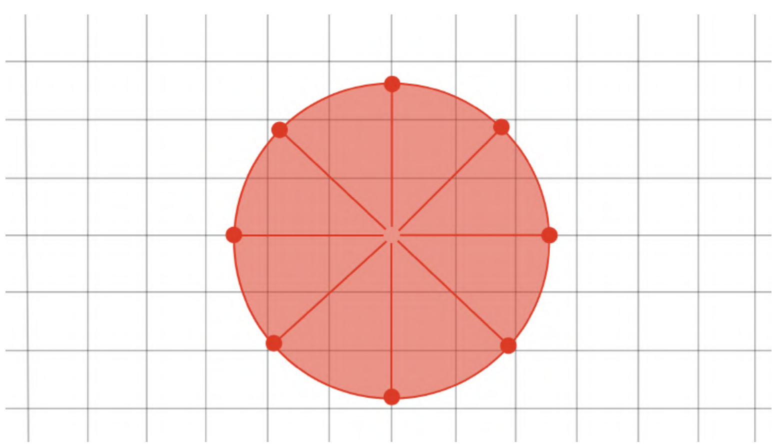
Scalar Curvature

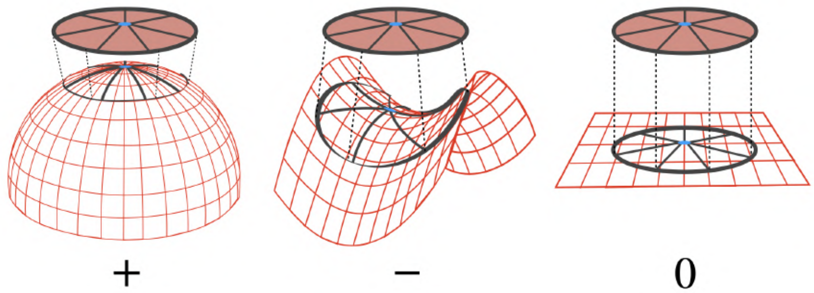
by DiBeos



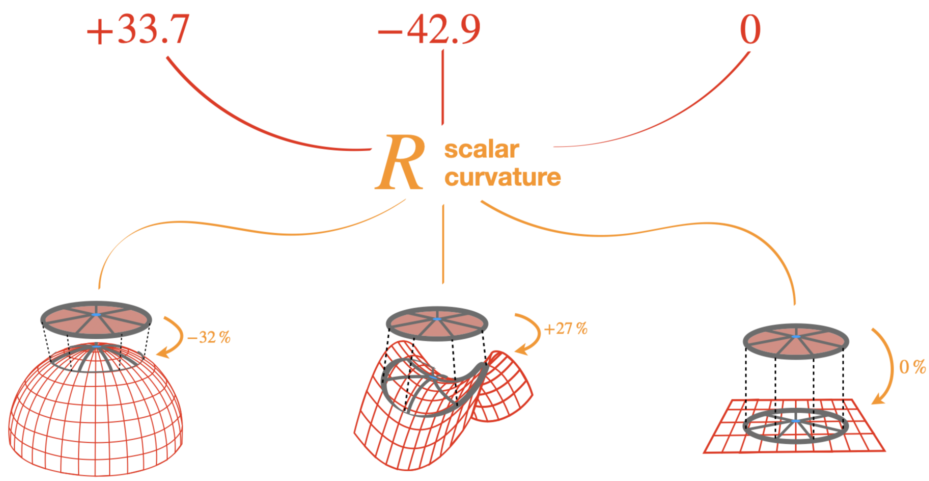
Imagine you are walking on a surface. If the surface is flat, you can draw a tiny circle by rushing on straight lines in many directions, then laying out dots that are all the same distance from your initial (central) position. Now, you measure its area. The result is exactly what you would expect from Euclidean geometry.



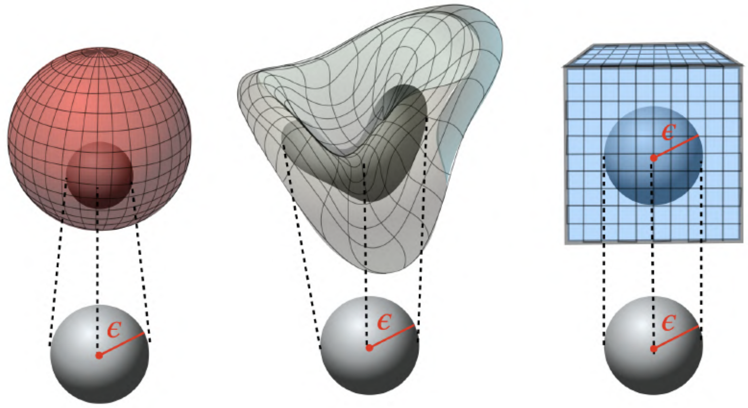
In **Differential Geometry** there are, in general, three kinds of curvature: *positive*, *negative* and *zero*. In a positively curved space, if you draw that circle around you, the area will be *less* than expected. In a negatively curved space, the area will be *larger* than expected. In a flat space (no curvature), the area stays the *same*.



Now, the question is: how can we capture this “circle distortion” as a number that tells us how curved the space is, right where you are standing? That’s where *scalar curvature* comes in! It’s denoted with the capital letter R, and it is a way of capturing how much a space around a point is curving. You can also think of it as the average amount by which circles drawn around that point shrink or expand compared to its flat version.



Of course, this is the $2D$ case, in which we talk about the variation of area of curved circles with respect to flat circles. In 3 (or higher) dimensions, we talk about *volumes* (or *hypervolumes*) of balls, with small radius $\epsilon > 0$, around the point that we are interested in. The scalar curvature tracks how much this volume differs from what it would be in a flat space, for each point. The result is a single number at each point. If it is positive, the ball shrinks. If it is negative, the ball expands.

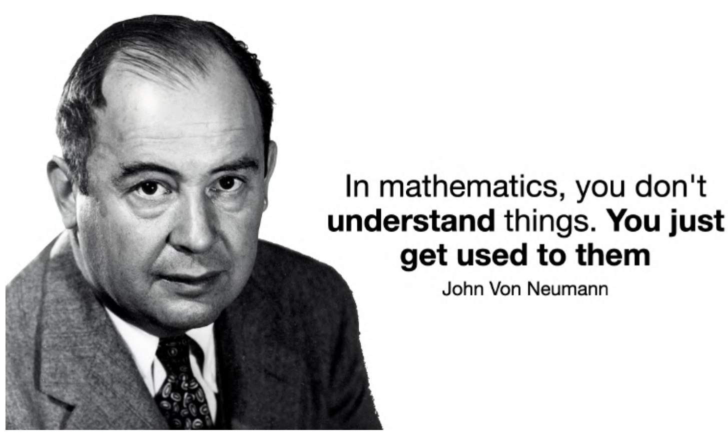


Now it's time to be a little bit more rigorous. We will see an algorithm to calculate the scalar curvature R of a general n-dimensional *Riemannian manifold*.

If you don't know what a Riemannian manifold is, check out this video in the channel, where we also attached a very clear written explanation in PDF, just as we do with all of our videos. So, check out the PDF link in the description of this video as well.

Riemannian Manifolds in 12 Minutes
PDF link: Riemannian Manifolds

Don't worry about the technical terms that you don't understand yet. We will get to each one of them. As one of the great mathematicians, John Von Neumann, once said:

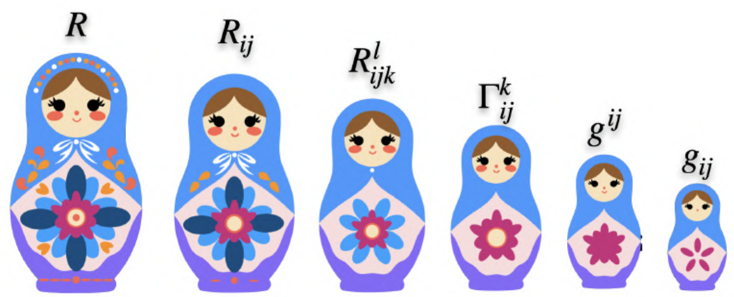


Anyway, let's get to it.

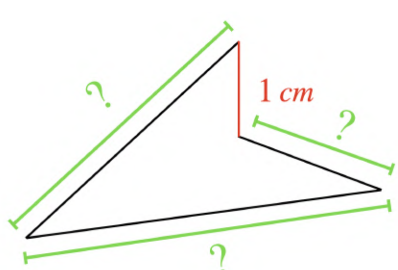
Scalar Curvature Algorithm

1. Choose a **Riemannian metric** for the manifold ($\mathbf{g_{ij}}$)
2. Calculate the **inverse metric** ($\mathbf{g^{ij}}$)
3. Compute the **Christoffel symbols** ($\mathbf{\Gamma^{k}_{ij}}$)
4. Calculate the **Riemannian curvature tensor** ($\mathbf{R^{l}_{ijk}}$)
5. Compute the **Ricci curvature tensor** ($\mathbf{R_{ij}}$)
6. Calculate the **scalar curvature** ($\mathbf{R}$)

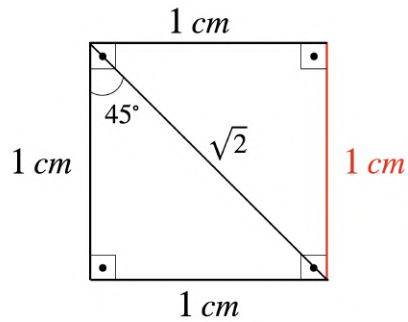
Ok, as you can tell, that's a long process. The main issue here is that there is, what I call, the "*Matryoshka effect*". Each step consists in calculating a quantity that is "nested" inside of the other. Just like *Russian dolls*. This makes the algorithm really cumbersome, and that's one of the reasons *symmetries* play a very important role. They help us to simplify, and shortcut, a lot of calculations.



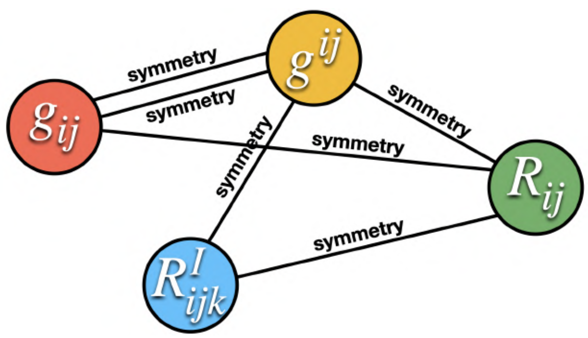
As a very simple illustration of the power of symmetries, imagine I give you a quadrilateral, and tell you that one of its sides measures 1 cm.



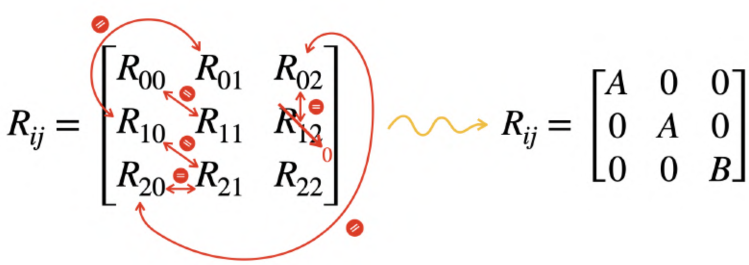
Just by imagining and trying to sketch such a figure you could attempt to approximate the lengths of the other sides, and maybe estimate the angles, but it's still really difficult and imprecise. However, if all of a sudden I introduce just one extra piece of information (i.e., a symmetry), for example that this quadrilateral is actually a *square*, then all of a sudden you can basically calculate everything you want about it!



In a similar way, when working with these tensors (i.e., Riemannian metric, inverse matrix, Riemann tensor, Ricci tensor, and so on) we will find many dependencies between its components. These dependencies are called symmetries, and they simplify our lives a lot!



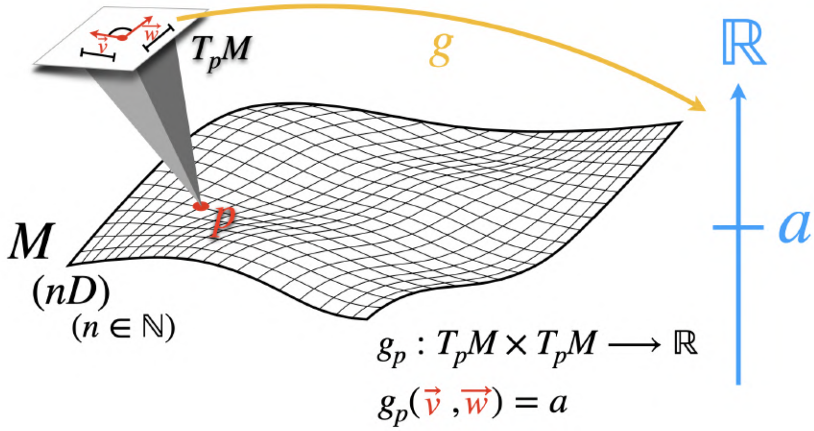
One illustration of it is the fact that the Ricci tensor is symmetric. Therefore, its matrix form can be diagonalized.



Let's revisit the first step of the algorithm:

1. **Riemannian metric** ($\mathbf{g_{ij}}$)

Imagine a smooth manifold M in n dimensions. A Riemannian metric g is a map that assigns, at each point $p \in M$, a way of measuring lengths and angles between vectors.



As you can see, there are many possible "formulas", or maps, that fit this definition of a Riemannian metric. It's usually not unique. So, choosing a metric for a manifold is nothing but that: *a choice*. However, some metrics are "better" or "worse" depending on the manifold you're studying, the symmetries it has, and the properties you want to investigate.

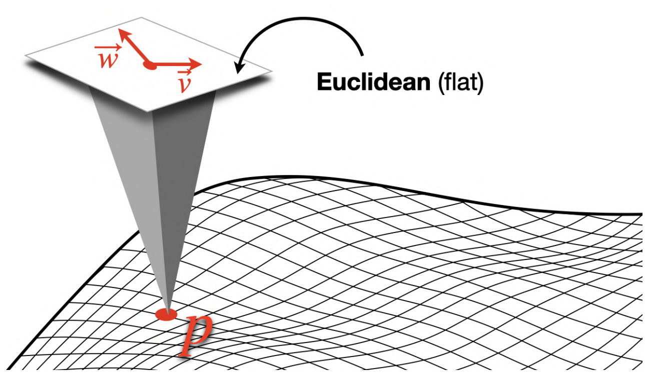
If you feel like you need to understand better what a manifold and a Riemannian metric are check out this video, and PDF, in the channel:

How to Get to Manifolds Naturally
PDF link: Manifolds

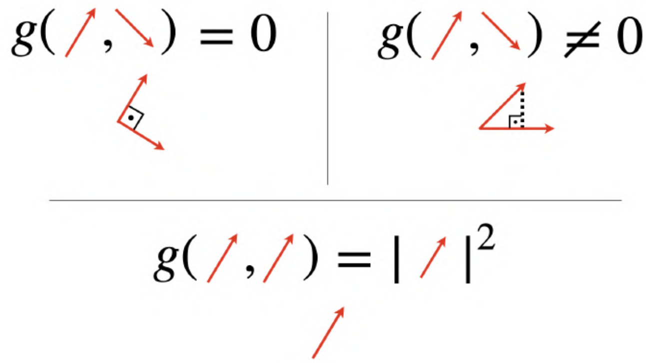
Anyway, you can think of it this way: you "feed" in two tangent vectors at a point p, and the metric "spits out" a real number, which is the result of their dot product, since (by definition) the tangent space at each point is a Euclidean space (i.e., flat).

$$g_p(\vec{v}, \vec{w}) = a \in \mathbb{R}$$

$$\langle \nearrow, \searrow \rangle = a \in \mathbb{R}$$



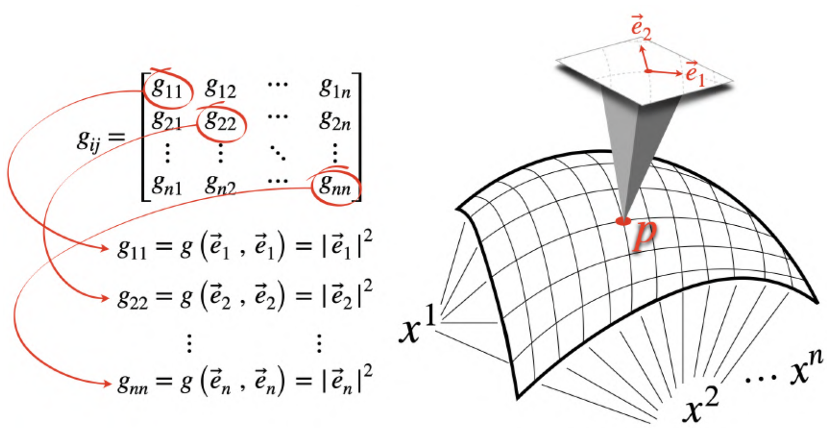
If the output is zero, it means that the vectors are *orthogonal* to one another. If the result is not zero, then it measures "how much they align". If the two vectors are the same, the output is the square length of the vectors.



This metric can be expressed as a matrix:

$$g_{ij} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}$$

The diagonal entries are the squared lengths of the basis vectors $\{\vec{e}_1 \, , \, \vec{e}_2 \, , \, ... \, , \, \vec{e}_n\}$.



If the space is Euclidean, all the diagonal entries are 1, because the basis vectors are *versors*, i.e. unit vectors – their length is 1. But in curved spaces, they can vary with position and be different.

$$g_{ij} = \begin{bmatrix} |\vec{e}_1|^2 & g_{12} & \cdots & g_{1n} \\ g_{21} & |\vec{e}_2|^2 & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & |\vec{e}_n|^2 \end{bmatrix}$$

The off-diagonal entries are inner products between different basis vectors:

$$\begin{gathered} g_{12} = g\left(\vec{e}_1\,,\,\vec{e}_2\right) = \vec{e}_1 \cdot \vec{e}_2 \\ g_{21} = g\left(\vec{e}_2\,,\,\vec{e}_1\right) = \vec{e}_2 \cdot \vec{e}_1 \\ \vdots \\ g_{nm} = g\left(\vec{e}_n\,,\,\vec{e}_m\right) = \vec{e}_n \cdot \vec{e}_m \end{gathered}$$

The "dot" represents the inner product under the Riemannian metric. So, they measure how "non-orthogonal" the coordinate directions are at a point.

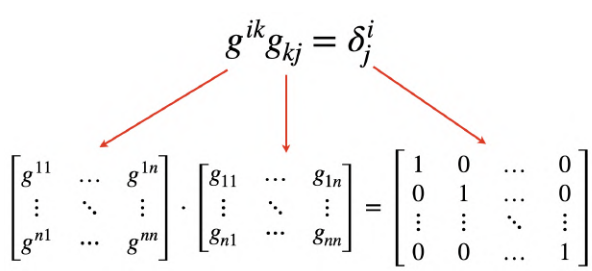
$$g_{ij} = \begin{bmatrix} |\vec{e}_1|^2 & \vec{e}_1 \cdot \vec{e}_2 & \cdots & \vec{e}_1 \cdot \vec{e}_n \\ \vec{e}_2 \cdot \vec{e}_1 & |\vec{e}_2|^2 & \cdots & \vec{e}_2 \cdot \vec{e}_n \\ \vdots & \vdots & \ddots & \vdots \\ \vec{e}_n \cdot \vec{e}_1 & \vec{e}_n \cdot \vec{e}_2 & \cdots & |\vec{e}_n|^2 \end{bmatrix}$$

If one of the entries is zero (i.e., $\exists\, a, b \in \{1\,,\,\ldots\,,n\} :\; g_{ab} = 0$), then the vectors are orthogonal. If it is not zero, then they are tilted with respect to each other.

Ok, once you defined your Riemannian metric, you can move on to step 2:

2. **Inverse matrix ($\mathbf{g^{ij}}$)**

The inverse matrix, by definition, is a matrix that, when multiplied by its original version, results in the identity matrix:

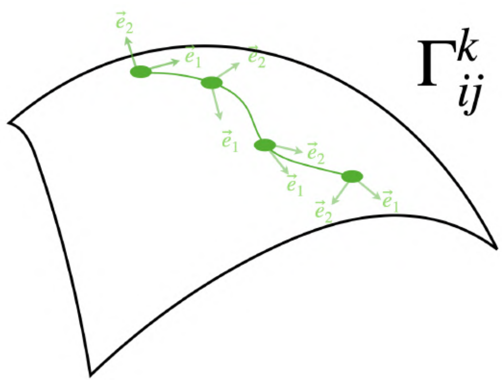


There are many **Linear Algebra** techniques to calculate the inverse matrix, but we won't see them here. If you are interested in learning more check out the video and PDF below:

The Core of Linear Algebra
PDF link: Linear Algebra

3. **Christoffel symbols (Γ^k_{ij})**

These are mathematical objects that track how the coordinate directions tilt across the manifold. They encode how to differentiate vectors smoothly in curved spaces.

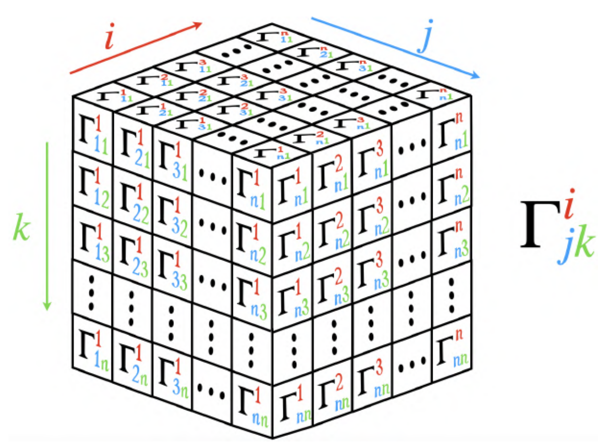


Christoffel symbols are calculated in the following way:

$$\Gamma^k_{ij} = \frac{1}{2} g^{kl} \left(\frac{\partial g_{jl}}{\partial x^i} + \frac{\partial g_{il}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^l} \right)$$

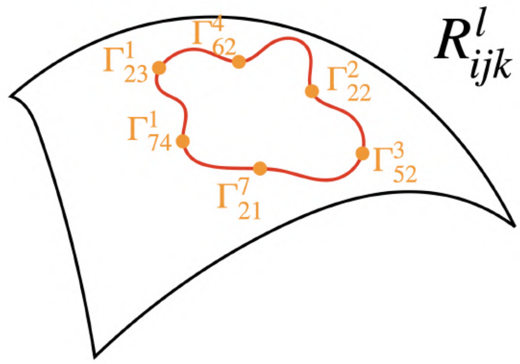
This looks scary, I know. As you can see, the "Matryoshka effect" is escalating quickly!

Notice how this object has three indices (i, j, k). This means that we can image it as a sort of "3-dimensional matrix", like a cube:



4. **Riemann curvature tensor ($\mathbf{R^l_{ijk}}$)**

Think of the Riemann curvature tensor as describing how the Christoffel symbols vary around loops.



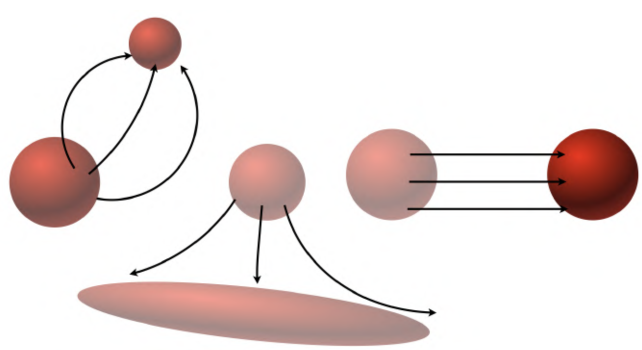
It captures the cumulative tilting effect when transporting vectors around closed paths. It is computed through the following formula:

$$R^l_{ijk} = \frac{\partial \Gamma^l_{jk}}{\partial x^i} - \frac{\partial \Gamma^l_{ik}}{\partial x^j} + \Gamma^m_{jk}\,\Gamma^l_{im} - \Gamma^m_{ik}\,\Gamma^l_{jm}$$

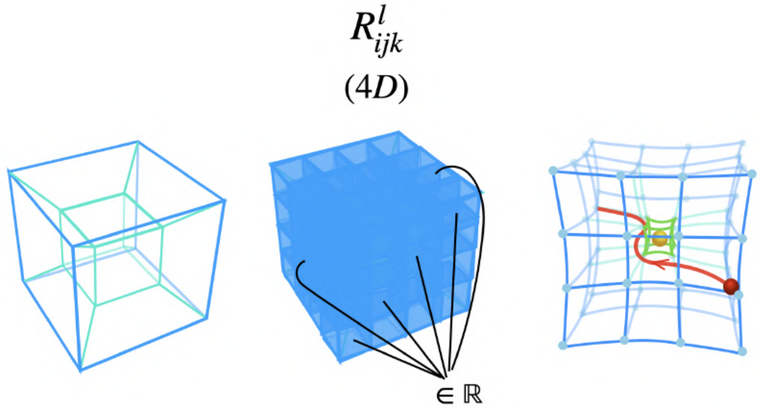
The Christoffel symbols describe how vectors change when moving infinitesimally in a single direction. The Riemann tensor is built from derivatives of the Christoffel symbols and their combinations. It detects how geometry changes across two directions, i.e. over a loop.

5. **Ricci curvature tensor ($\mathbf{R_{ij}}$)**

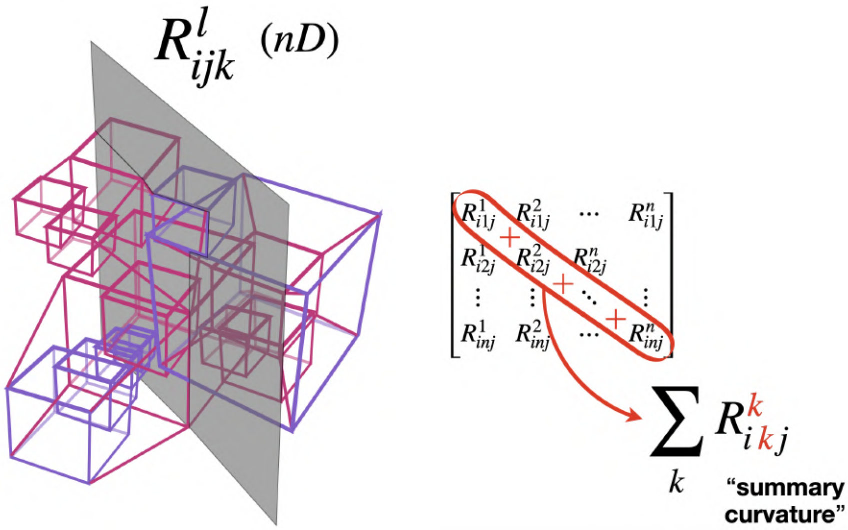
The Ricci tensor measures how volumes deform when you move through space – it tells you how geodesics converge, or diverge, depending on the direction you move.



We start from the Riemann tensor R^l_{ijk}, which has four indices. It can be thought of as a 4-dimensional hypercube (if we are describing the 4-dimensional spacetime, for example), with entries that are just real numbers. Of course, the more general case would be the one in which we have a n-dimensional cube.



In order to compute the Ricci tensor, we "slice through" this cuboid and then we add up a row of values in a particular *cross-section*. The result is a sort of "summary" of the curvature in each direction.



This process is called a *contraction*. It is actually just a fancy word for summing along a repeated index which shrinks the dimensionality of the object (like from 4 to 2 dimensions) – it is similar to taking the trace of a matrix, but extended to higher dimensions.

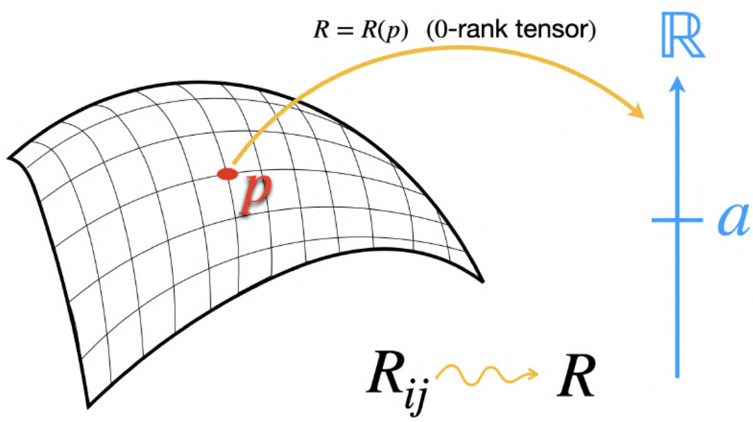
$$R_{ij} = R^{k}_{ikj} \iff R_{ij} = \sum_{k=1}^{n} R^{k}_{ikj} \iff$$

$$\iff R_{ij} = R^{1}_{i1j} + R^{2}_{i2j} + \ldots + R^{n}_{inj}$$

Finally!

6. **Scalar curvature (R)**

As we've seen before, the scalar curvature is a single number at each point that summarizes how the space is curved on average in all directions.



R is a real-valued function (i.e., a 0-*rank tensor*), which is obtained by collapsing the Ricci tensor into a single number. This is done by summing over both indices of the Ricci matrix using the inverse metric (a double contraction)

$$R = g^{ij} R_{ij} \iff R = \sum_{i=1}^{n} \sum_{j=1}^{n} g^{ij} R_{ij} \iff$$

$$\iff R = g^{11} R_{11} + \ldots + g^{nn} R_{nn}$$

In conclusion, the Ricci matrix R_{ij} tells us how much the space curves, or distorts, volumes when moving in a specific direction. The inverse metric g^{ij} tells us how to "average" these distortions, and provides a geometric tool for doing so. The product between the inverse metric and the Ricci tensor ($g^{ij} \cdot R_{ij}$) gives a sort of weighted average of curvature in all directions – and this is the scalar curvature R.

This quantity (R) answers the following question: at a specific point on the manifold, is the curvature, on average, positive, negative or zero around its neighborhood?

`If you found this document useful let us know. If you found typos and things to improve, let us know as well. Your feedback is very important to us. We're working hard to deliver the best material possible. Contact us at: dibeos.contact@gmail.com`