

# Sectional Curvature in Riemannian Manifolds

Elliott Fairchild  
Francis Owen  
Brendan Burns Healy

The metric structure on a Riemannian or pseudo-Riemannian manifold is entirely determined by its metric tensor, which has a matrix representation in any given chart. Encoded in this metric is the sectional curvature, which is often of interest to mathematical physicists, differential geometers and geometric group theorists alike. In this article, we provide a function to compute the sectional curvature for a Riemannian manifold given its metric tensor. We also define a function to obtain the Ricci tensor, a closely related object.

## ■ Introduction

A *Riemannian manifold* is a differentiable manifold together with a Riemannian metric tensor that takes any point in the manifold to a positive-definite inner product function on its *tangent space*, which is a vector space representing geodesic directions from that point [1]. We can treat this tensor as a symmetric matrix  $G$  with entries denoted by  $g_{ij}$  representing the relationship between tangent vectors at a point in the manifold, once a system of local coordinates has been chosen [2, 3]. In the case of a parameterized surface, we can use the parameters to compute the full metric tensor.

```
In[1]:= ParameterizedSurfaceMetric[parametrization_, {u_, v_}] :=  
Module [  
  {  
    U = D[parametrization, u],  
    V = D[parametrization, v]  
  },  
  { {U.U, U.V}, {V.U, V.V} }  
]
```

A classical parametrization of a surface is the standard parameterization of the sphere. We compute the metric tensor of the standard sphere below.

```
In[2]:= SphereParameterization[u_, v_] :=
        {Cos[u] Sin[v], Sin[u] Sin[v], Cos[v]}
```

```
In[3]:= MatrixForm@
        Simplify@ParameterizedSurfaceMetric[
            SphereParameterization[u, v], {u, v}]
```

Out[3]/MatrixForm=

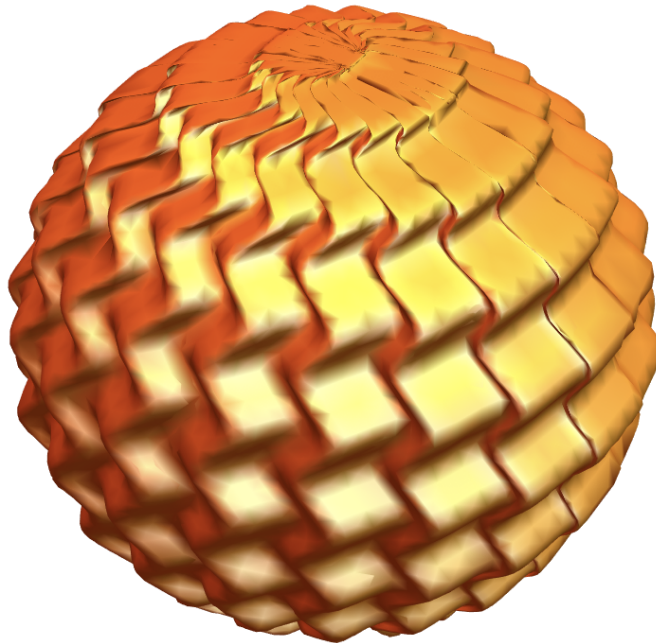
$$\begin{pmatrix} \sin[v]^2 & 0 \\ 0 & 1 \end{pmatrix}$$

This also works for more complicated surfaces. The following is an example taken from [4].

```
In[4]:= BumpySphere[u_, v_] :=
        {Sin[u] Sin[v] + 1 / 20 Cos[20 v], Cos[u] Sin[v] + 1 / 20 Cos[20 u],
        Cos[v]}
```

```
In[5]:= ParametricPlot3D[BumpySphere[u, v], {u, -π, π},
        {v, -π, π}, MaxRecursion → 4,
        PlotStyle → {Orange, Specularity[White, 10]}, Axes → None,
        Mesh → None, Boxed → False]
```

Out[5]=



```
In[6]:= ParameterizedSurfaceMetric[BumpySphere[u, v], {u, v}]
```

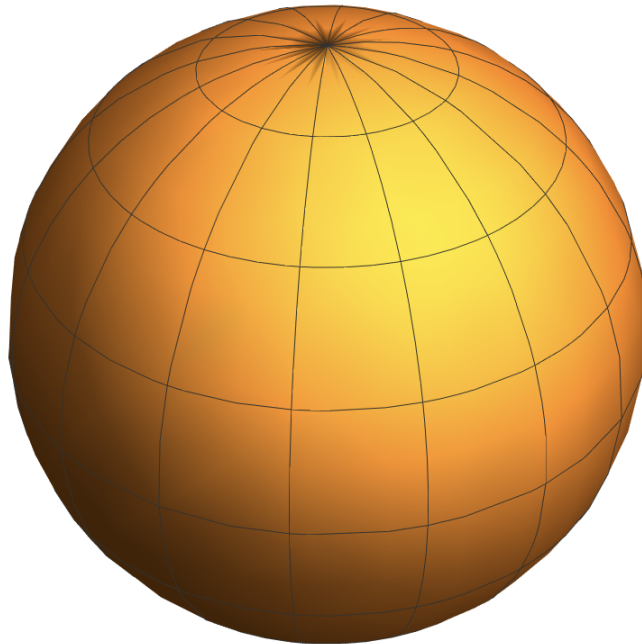
```
Out[6]= { {Cos[u]^2 Sin[v]^2 + (-Sin[20 u] - Sin[u] Sin[v])^2,
           Cos[u] Cos[v] (-Sin[20 u] - Sin[u] Sin[v]) +
           Cos[u] Sin[v] (Cos[v] Sin[u] - Sin[20 v]) },
          {Cos[u] Cos[v] (-Sin[20 u] - Sin[u] Sin[v]) +
           Cos[u] Sin[v] (Cos[v] Sin[u] - Sin[20 v]),
           Cos[u]^2 Cos[v]^2 + Sin[v]^2 + (Cos[v] Sin[u] - Sin[20 v])^2 } }
```

Denoting the coordinates by  $x^i$ , we can then define  $ds^2 := g_{ij} dx^i dx^j$ , where the  $g_{ij}$  are functions of the coordinates  $x^i$ ; this definition uses Einstein notation, which will also apply wherever applicable in the following. From this surprisingly dense description of distance, we can extract many properties of a given Riemannian manifold, including *sectional curvature*, which will be given an explicit formula later. In particular, two-dimensional manifolds, also called *surfaces*, carry a value that measures at any given point how far they are from being flat. This value can be positive, negative or zero. For intuition, we give examples of each of these types of behavior.

The sphere is the prototypical example of a surface of positive curvature.

```
In[7]:= ParametricPlot3D[SphereParameterization[u, v],
                        {u, -π, π}, {v, -π, π}, Axes → None, Boxed → False]
```

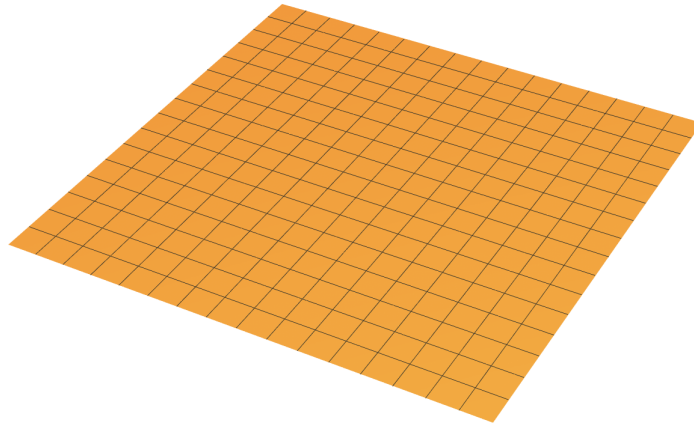
```
Out[7]=
```



Any convex subspace of Euclidean space has zero curvature everywhere.

```
In[8]:= ParametricPlot3D[{u, v, 0}, {u, -10, 10}, {v, -10, 10},
  Boxed → False, Axes → False]
```

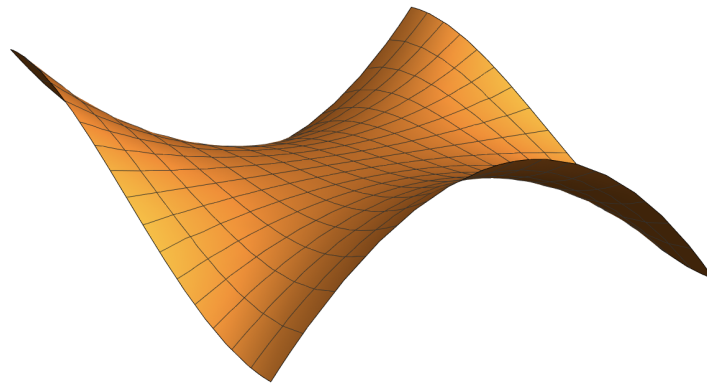
Out[8]=



The monkey saddle is an example of a two-dimensional figure with negative curvature.

```
In[9]:= Plot3D[x (x^2 - 3 y^2), {x, -10, 10}, {y, -10, 10},
  Boxed → False, Axes → False]
```

Out[9]=



Sectional curvature is a locally defined value that gives the curvature of a special type of two-dimensional subspace at a point, where the two dimensions defining the surface are input as tangent vectors. Manifolds may have points that admit sections of both negative and positive curvature simultaneously, as is the case for the Schwarzschild metric discussed in the section “Applications in Physics.” An important property of sectional curvature is that on a Riemannian manifold it varies smoothly with respect to both the point in the manifold being considered and the choice of tangent vectors.

Sectional curvature is given by

$$K(\vec{u}, \vec{v}) = \frac{\langle R(\vec{u}, \vec{v})\vec{u}, \vec{v} \rangle}{\langle \vec{u}, \vec{u} \rangle \langle \vec{v}, \vec{v} \rangle - \langle \vec{u}, \vec{v} \rangle^2},$$

where  $\vec{u}, \vec{v} \in T_p M$ .

In this formula,  $R$  represents the purely covariant Riemannian curvature tensor, a function on tangent vectors that is completely determined by the  $g_{ij}$ . Both  $R$  and the  $g_{ij}$  are treated more thoroughly in the following section, as well as in [1]. Some immediate properties of the curvature formula are that  $K$  is symmetric in its two entries,  $K$  is undefined if the vectors  $\vec{u}$  and  $\vec{v}$  are linearly dependent, and  $K$  does not change when either vector is scaled. Moreover, any two tangent vectors that define the same subspace of the tangent space give the same value. This is important because curvature should only depend on the embedded surface itself and not how it was determined.

While we are primarily concerned with Riemannian manifolds, it is worth noting that all calculations are valid for pseudo-Riemannian manifolds, in which the assumption that the metric tensor is positive-definite is dropped. This generalization is especially important in areas such as general relativity, where the metric tensors that represent spacetime have a different signature than that of traditional Riemannian manifolds. We explore this connection more in the section “Applications in Physics.”

## ■ Coordinate Systems and the Representation of the Metric Tensors

For a differentiable manifold, an *atlas* is a collection of homeomorphisms, called *charts*, from open sets in Euclidean space to the manifold, such that overlapping charts can be made compatible by a differentiable transition map between them. Via these homeomorphisms, we can define coordinates in an open set around any point by adopting the coordinates in the corresponding Euclidean neighborhood. By convention, these coordinates are labelled  $x^i$ , and unless important, we omit the point giving rise to the coordinates. In some cases of interest, it is possible to adopt a coordinate system that is valid over the whole manifold.

From such a coordinate system, whether local or global, we can define a basis for the tangent space using a *coordinate frame* [5]. This will be the basis consisting of the partial derivative operators in each of the coordinate directions, that is,  $\partial_{x^i}$ . Considering the tangent space as a vector space, this set is sometimes referred to in mathematical physics as a *holonomic basis* for the manifold. We use this expression then to define the symmetric matrix  $G = (g_{ij})$  by the following expression for  $p \in M$ :

$$g_{ij}|_p = g_p\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right).$$

From here, we define one more tensor of interest for the purposes of calculating curvature. Using Einstein notation, the Riemannian curvature tensor is

$$R_{abcd} = g_{ae} R_{bcd}^e,$$

$$R_{bcd}^a = \frac{\partial}{\partial x^c} \Gamma_{bd}^a - \frac{\partial}{\partial x^d} \Gamma_{bc}^a + \Gamma_{cs}^a \Gamma_{bd}^s - \Gamma_{ds}^a \Gamma_{bc}^s.$$

The various  $\Gamma$  are the *Christoffel symbols*, for which code is presented in the next section. In light of these definitions, we recall sectional curvature once again from the introduction as the following, now considering the special case of the tangent vectors being chosen in coordinate directions:

$$K(\partial_{x^i}, \partial_{x^j}) = \frac{R_{ijij}}{\|\partial_{x^i}\|^2 \|\partial_{x^j}\|^2 - \langle \partial_{x^i}, \partial_{x^j} \rangle^2}.$$

The norm in the denominator is the norm of the tangent vector associated to that partial derivative in the holonomic basis, which is induced by the associated inner product from  $G$ .

## ■ Sectional Curvature

We now create functions to compute these tensors and sectional curvature itself. These values depend on a set of coordinates and a Riemannian metric tensor, so that will be the information that serves as the input for these functions. Coordinates should be a list of coordinate names like  $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ , and `MetricTensor` should be a square symmetric matrix whose size matches the length of the coordinate list. Some not inconsiderable inspiration for the first half of this code was taken from Professor Leonard Parker's Mathematica notebook "Curvature and the Einstein Equation," which is available online as a supplement to [6].

We can now define a function for the Christoffel symbols from the previous section. This calculation consists of taking partial derivatives of the metric tensor components and one tensor operation. In Mathematica, the dot product, typically used for vectors and matrices, is also able to take tensors and contract indices.

```
In[10]:= Christoffel[coordinates_, MetricTensor_] := Module[
  {n = Length[coordinates]},
  Simplify[
    Inverse[MetricTensor].(
      1 / 2 Table[
        D[MetricTensor[[s, j]], coordinates[[k]]] +
        D[MetricTensor[[s, k]], coordinates[[j]]] -
        D[MetricTensor[[j, k]], coordinates[[s]]],
        {s, n}, {j, n}, {k, n}
      )
  ]
]
```

We can now use the formulas stated in the second section to define both the covariant and contravariant forms of the Riemannian curvature tensor.

```
In[11]:= RiemannContravariant[coordinates_, MetricTensor_] := Module[
  {n, c},
  n = Length[coordinates];
  c = Christoffel[coordinates, MetricTensor];
  Simplify@Table[
    D[c[[i, j, l]], coordinates[[k]]] -
    D[c[[i, j, k]], coordinates[[l]]] +
    (c.c)[[i, k, l, j]] - (c.c)[[i, l, k, j]],
    {i, n}, {j, n}, {k, n}, {l, n}
  ]
]
```

We perform one more tensor operation using the dot product to transform our partially contravariant tensor into one that is purely covariant. Both of these will be called at various points later.

```
In[12]:= RiemannCovariant[coordinates_, MetricTensor_] :=
  MetricTensor.RiemannContravariant[coordinates,
  MetricTensor]
```

The full function to return the sectional curvatures consists of computing a scaled version of the covariant Riemannian metric tensor.

```
In[13]:= SectionalCurvature[coordinates_, MetricTensor_] := Module[
  {n, R, listRiemannCovariant, K},
  n = Length[coordinates];
  R = RiemannCovariant[coordinates, MetricTensor];
  listRiemannCovariant = Table[
    R[[i, j, k, l]],
    {i, n}, {j, n}, {k, n}, {l, n}
  ];
  K[u_, u_] = 0;
  K[u_, v_] := Simplify[
    listRiemannCovariant[[u, v, u, v]] /
    (MetricTensor[[u, u]] × MetricTensor[[v, v]] -
    MetricTensor[[u, v]]^2)
  ] /; u ≠ v;
  Table[K[u, v], {u, n}, {v, n}]
]
```

The output consists of a symmetric matrix with zero diagonal entries representing curvatures in the coordinate directions. These diagonal values should not be taken literally, as curvature is undefined given two linearly dependent directions. While this of course does not give all possible sectional curvatures, one may perform a linear transformation on the basis in order

to obtain a new metric tensor with arbitrary (linearly independent) vectors as basis elements. From here, the new tensor may be used for computation.

Here is an example with diagonal entries that are functions of the last coordinate.

```
In[14]:= SectionalCurvature[{x, y, z},
  {{F[z], 0, 0}, {0, G[z], 0}, {0, 0, H[z]}}
```

$$\text{Out[14]} = \left\{ \left\{ 0, -\frac{F'[z] G'[z]}{4 F[z] \times G[z] \times H[z]}, \frac{F[z] F'[z] H'[z] + H[z] (F'[z]^2 - 2 F[z] F''[z])}{4 F[z]^2 H[z]^2} \right\}, \right.$$

$$\left. \left\{ -\frac{F'[z] G'[z]}{4 F[z] \times G[z] \times H[z]}, 0, \frac{G[z] G'[z] H'[z] + H[z] (G'[z]^2 - 2 G[z] G''[z])}{4 G[z]^2 H[z]^2} \right\}, \right.$$

$$\left. \left\{ \frac{F[z] F'[z] H'[z] + H[z] (F'[z]^2 - 2 F[z] F''[z])}{4 F[z]^2 H[z]^2}, \frac{G[z] G'[z] H'[z] + H[z] (G'[z]^2 - 2 G[z] G''[z])}{4 G[z]^2 H[z]^2}, 0 \right\} \right\}$$

Any good computation in mathematics must stand to scrutiny by known cases, so we evaluate our function with the input of hyperbolic 3-space. The two in the exponent should be imagined as the squaring of the exponential function.

```
In[15]:= SectionalCurvature[{a, b, c},
  {{Exp[-2 c], 0, 0}, {0, Exp[-2 c], 0}, {0, 0, 1}}] //
  MatrixForm
```

```
Out[15]//MatrixForm=
```

$$\begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix}$$

Checking with [7] verifies that this is indeed a global metric tensor for hyperbolic 3-space. As such, we know that it has constant sectional curvature of  $-1$  (recall the diagonal entries do not represent any curvature information).



## ■ Applications in Topology

Continuing with the hyperbolic space metric tensor, it is a well-known result in hyperbolic geometry that one is able to scale these first two dimensions to vary the curvature and produce a *pinched curvature* manifold.

```
In[16]:= pinchedcurvature = {{Exp[-2 a z], 0, 0}, {0, Exp[-2 b z], 0},
      {0, 0, 1}};
```

If we allow for new constant coefficients in the exponents for positive real numbers  $a$  and  $b$ , then we should see explicit bounds on the curvatures.

```
In[17]:= SectionalCurvature[{x, y, z}, pinchedcurvature] //
      MatrixForm
```

Out[17]//MatrixForm=

$$\begin{pmatrix} 0 & -a b & -a^2 \\ -a b & 0 & -b^2 \\ -a^2 & -b^2 & 0 \end{pmatrix}$$

In this vein, the Riemannian structure for *complex hyperbolic space* is similar to the real case, except for a modification to allow for complex variables.

```
In[18]:= complexhyperbolic =
      (2 z) ^ -2 {
      {4 (z + y^2), -4 x y, 0, -2 y},
      {-4 x y, 4 (z + x^2), 0, 2 x},
      {0, 0, 1, 0},
      {-2 y, 2 x, 0, 1}
      };
```

In this setting, a formula for the metric tensor valid over the entire manifold is available from [8], among other places.

```
In[19]:= SectionalCurvature[{x, y, z, w}, complexhyperbolic] //
      MatrixForm
```

Out[19]//MatrixForm=

$$\begin{pmatrix} 0 & -\frac{x^2+y^2+4z}{x^2+y^2+z} & -\frac{4y^2+z}{y^2+z} & -1 \\ -\frac{x^2+y^2+4z}{x^2+y^2+z} & 0 & -\frac{4x^2+z}{x^2+z} & -1 \\ -\frac{4y^2+z}{y^2+z} & -\frac{4x^2+z}{x^2+z} & 0 & -4 \\ -1 & -1 & -4 & 0 \end{pmatrix}$$

One can verify that, although not constant, the entries in the upper-left block are always bounded between  $-1$  and  $-4$  for positive  $z$ . This result agrees with sectional curvature in complex hyperbolic space, and so serves as an example of sectional curvature computation

where the underlying tensor is not diagonal. A careful review of [8] reminds us that this metric is only well-defined up to rescaling, which can change the values of the sectional curvature. What does not change, however, is the ratio of the largest and smallest curvatures, which are always exactly 4. The introduction in [9] takes considerable care to remind us that definitions change between curvatures in  $[-4, -1]$ ,  $[-1, -1/4]$  and even  $[-2, -1/2]$ .

## ■ Applications in Physics

Perhaps the most interesting applications of differentiable manifolds and curvature to physics lie in the area of relativity. This discipline uses the idea of a *Lorentzian manifold*, which is defined as a manifold equipped with a Lorentzian metric that has signature  $(-+++)$  instead of the  $(++++)$  signature for four-dimensional Riemannian manifolds. As noted in the introduction, however, this has no impact on the computations of sectional curvature. Examples of such Lorentzian metrics include the *Minkowski flat spacetime metric*;  $c$  is the familiar constant speed of light.

```
In[20]:= flatspacetime = {{c^2, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0},
                        {0, 0, 0, 1}};
```

Justifying the name of *flat* spacetime, our curvature calculation guarantees all sectional curvatures are identically zero.

```
In[21]:= SectionalCurvature[{x, y, z, t}, flatspacetime] //
        MatrixForm
```

```
Out[21]//MatrixForm=
      ( 0 0 0 0 )
      ( 0 0 0 0 )
      ( 0 0 0 0 )
      ( 0 0 0 0 )
```

More generic Lorentzian manifolds may have nonzero curvature. To this end, we examine the *Schwarzschild metric*, which describes spacetime outside a spherical mass such that the gravitational field outside the mass satisfies Einstein's field equations. This most commonly is viewed in the context of a black hole and how spacetime behaves nearby. More details on the following tensor can be found in [10].

```
In[22]:= SchwarzschildMetric = {
        {-(1 - 2 G m / (r c^2)), 0, 0, 0},
        {0, 1 / (1 - 2 G m / (r c^2)), 0, 0},
        {0, 0, r^2, 0},
        {0, 0, 0, r^2 Sin[θ]^2}
};
```

In the following,  $r$ ,  $\phi$  and  $\theta$  are standard spherical coordinates for three-dimensional space and  $t$  represents time. With this setup, we can calculate the sectional curvature of space-time for areas outside such a spherical mass.

```
In[23]:= SectionalCurvature[{t, r, \theta, \phi}, SchwarzschildMetric] //
MatrixForm
```

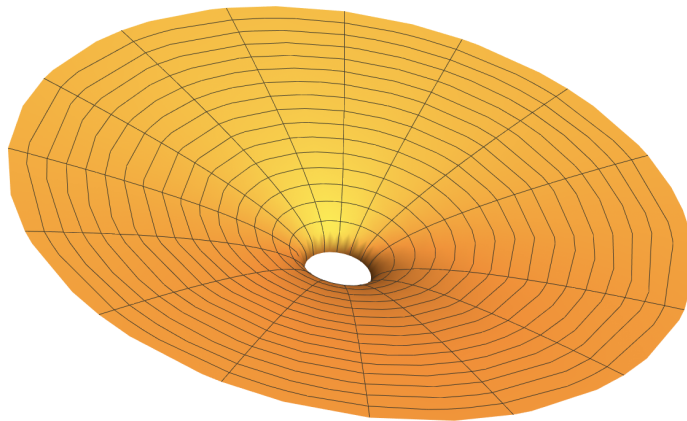
```
Out[23]//MatrixForm=
```

$$\begin{pmatrix} 0 & \frac{2 G m}{c^2 r^3} & -\frac{G m}{c^2 r^3} & -\frac{G m}{c^2 r^3} \\ \frac{2 G m}{c^2 r^3} & 0 & -\frac{G m}{c^2 r^3} & -\frac{G m}{c^2 r^3} \\ -\frac{G m}{c^2 r^3} & -\frac{G m}{c^2 r^3} & 0 & \frac{2 G m}{c^2 r^3} \\ -\frac{G m}{c^2 r^3} & -\frac{G m}{c^2 r^3} & \frac{2 G m}{c^2 r^3} & 0 \end{pmatrix}$$

This result indicates that the sectional curvature is directly proportional to the mass and inversely proportional to the distance from the object. In particular, there is a singularity at  $r = 0$ , indicating that curvature “blows up” near the center of the mass. Indeed, these results are in line with Flamm's paraboloid, the graphical representation of a constant-time equatorial slice of the Schwarzschild metric, whose details can be found in [11].

```
In[24]:= ParametricPlot3D[{r Cos[t], r Sin[t], ((r - 1))^(1/2)},
{r, 1, 10}, {t, 0, 2 Pi}, Boxed -> False, Axes -> False ]
```

```
Out[24]=
```



## ■ Ricci Curvature

In fact, the calculations we have done already allow us to compute one further object of interest for a Riemannian or pseudo-Riemannian manifold: the Ricci curvature. The Ricci curvature is a tensor that contracts the curvature tensor and is computable when one has the contravariant Riemannian curvature tensor. Below we use a built-in function for tensors to contract the first and third indices of the contravariant Riemannian curvature tensor to obtain a matrix containing condensed curvature information (see [12] for more information).

```
In[25]:= RicciCurvature[coordinates_, MetricTensor_] :=
  Simplify@TensorContract[
    RiemannContravariant[coordinates, MetricTensor],
    {{1, 3}}
```

The values 1 and 3 above refer to the dimensions we are contracting. In general, the corresponding indices must vary over sets of the same size; here all dimensions have indices that vary over a set whose size is the number of coordinates. We compute the Ricci curvature for some of the previous examples.

```
In[26]:= RicciCurvature[{x, y, z}, pinchedcurvature] // MatrixForm
```

Out[26]//MatrixForm=

$$\begin{pmatrix} -a(a+b)e^{-2az} & 0 & 0 \\ 0 & -b(a+b)e^{-2bz} & 0 \\ 0 & 0 & -a^2 - b^2 \end{pmatrix}$$

```
In[27]:= RicciCurvature[{x, y, z, w}, complexhyperbolic] //
  MatrixForm
```

Out[27]//MatrixForm=

$$\begin{pmatrix} -\frac{6(y^2+z)}{z^2} & \frac{6xy}{z^2} & 0 & \frac{3y}{z^2} \\ \frac{6xy}{z^2} & -\frac{6(x^2+z)}{z^2} & 0 & -\frac{3x}{z^2} \\ 0 & 0 & -\frac{3}{2z^2} & 0 \\ \frac{3y}{z^2} & -\frac{3x}{z^2} & 0 & -\frac{3}{2z^2} \end{pmatrix}$$

```
In[28]:= RicciCurvature[{t, r,  $\theta$ ,  $\phi$ }, SchwarzschildMetric] //
MatrixForm
```

```
Out[28]//MatrixForm=
```

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

The fact that the Ricci curvature vanishes for the above solution to the Einstein field equation is a consequence of its types of symmetries. In general, the Ricci curvature for other solutions is nonzero. Notice for the `pinchedcurvature` example (and the `SchwarzschildMetric`, trivially), all information from the Ricci tensor is contained in the diagonal elements. This is always the case for a diagonal metric tensor [12]. As such, we may sometimes be interested only in these values, so we take the diagonal in such a case.

```
In[29]:= Diagonal@RicciCurvature[{u, v},
ParameterizedSurfaceMetric[SphereParameterization[u, v],
{u, v}]]
```

```
Out[29]= {Sin[v]^2, 1}
```

```
In[30]:= Diagonal@RicciCurvature[{x, y, z}, pinchedcurvature]
```

```
Out[30]= {-a (a + b) e^{-2 a z}, -b (a + b) e^{-2 b z}, -a^2 - b^2}
```

## ■ Acknowledgments

The supervising author would like to thank Dr. Nicolas Robles for suggesting the submission of this article to *The Mathematica Journal*. We would also like to thank Leonard Parker, who authored the notebook file available at [6], which greatly illuminated some of the calculations. We are also very grateful to the referee and especially the editor, whose contributions have made this article much more accurate, legible and efficient.

## ■ References

- [1] M. do Carmo, *Differential Geometry of Curves & Surfaces*, Mineola, NY: Dover Publications, Inc., 2018.
- [2] J. M. Lee, *Introduction to Smooth Manifolds*, Graduate Texts in Mathematics, 218, New York: Springer, 2003.
- [3] C. Stover and E. W. Weisstein, “Metric Tensor” from MathWorld—A Wolfram Web Resource. [mathworld.wolfram.com/MetricTensor.html](http://mathworld.wolfram.com/MetricTensor.html).
- [4] “ParametricPlot3D,” ParametricPlot3D from Wolfram Language & System Documentation Center—A Wolfram Web Resource. [reference.wolfram.com/language/ref/ParametricPlot3D.html](http://reference.wolfram.com/language/ref/ParametricPlot3D.html).
- [5] F. Catoni, D. Boccaletti, R. Cannata, V. Catoni, E. Nichelatti and P. Zampetti, *The Mathematics of Minkowski Space-Time*, Frontiers in Mathematics, Basel: Birkhäuser Verlag, 2008.
- [6] J. B. Hartle, *Gravity: An Introduction to Einstein’s General Relativity*, San Francisco: Addison-Wesley, 2003. [web.physics.ucsb.edu/~gravitybook/math/curvature.pdf](http://web.physics.ucsb.edu/~gravitybook/math/curvature.pdf).
- [7] J. G. Ratcliffe, *Foundations of Hyperbolic Manifolds*, 2nd ed., Graduate Texts in Mathematics, 149, New York: Springer, 2006.
- [8] J. Parker, “Notes on Complex Hyperbolic Geometry” (Jan 10, 2020). [maths.dur.ac.uk/~dma0jrp/img/NCHG.pdf](http://maths.dur.ac.uk/~dma0jrp/img/NCHG.pdf).
- [9] W. M. Goldman, *Complex Hyperbolic Geometry*, Oxford Mathematical Monographs, Oxford Science Publications, New York: Oxford University Press, 1999.
- [10] R. Adler, M. Bazin and M. Schiffer, *Introduction to General Relativity*, New York: McGraw-Hill, 1965.
- [11] R. T. Eufrazio, N. A. Meholosky and L. Resca, “Curved Space, Curved Time, and Curved Space-Time in Schwarzschild Geodesic Geometry,” *General Relativity and Gravitation*, **50**(159), 2018. doi:10.1007/s10714-018-2481-2.
- [12] L. A. Sidorov, “Ricci Tensor,” *Encyclopedia of Mathematics* (M. Hazewinkel, ed.), Netherlands: Springer, 1990. [www.encyclopediaofmath.org/index.php/Ricci\\_tensor](http://www.encyclopediaofmath.org/index.php/Ricci_tensor).

E. Fairchild, F. Owen and B. Burns Healy, “Sectional Curvature in Riemannian Manifolds,” *The Mathematica Journal*, 2020. <https://doi.org/10.3888/tmj.22-1>.

## About the Authors

Elliott Fairchild is a high-school student at Cedarburg High School. He particularly enjoys problems in analysis, and is always looking for more research opportunities.

Francis Owen is an undergraduate student at the University of Wisconsin-Milwaukee. His major is Applied Mathematics and Computer Science, and he is eager to find new programming opportunities.

Brendan Burns Healy is a Visiting Assistant Professor at the University of Wisconsin-Milwaukee. Though a geometric group theorist and low-dimensional topologist by training, he also enjoys problems of computation and coding.

**Elliott Fairchild**

*Department of Mathematical Sciences  
University of Wisconsin-Milwaukee  
3200 N. Cramer St.  
Milwaukee, WI 53211  
rsquared3.1415926@gmail.com*

**Francis Owen**

*Department of Mathematical Sciences  
University of Wisconsin-Milwaukee  
3200 N. Cramer St.  
Milwaukee, WI 53211*

**Brendan Burns Healy, PhD**

*Department of Mathematical Sciences  
University of Wisconsin-Milwaukee  
3200 N. Cramer St.  
Milwaukee, WI 53211  
www.burnshealy.com*