12-2021

Equivalence: A Covariantly Constant Problem in General Relativity

Jaren Hobbs
Utah State University

EQUIVALENCE: A COVARIANTLY CONSTANT PROBLEM IN GENERAL RELATIVITY

by

Jaren Hobbs

A report submitted in partial fulfillment

of the requirements for the degree

of

MASTER OF SCIENCE

in

Physics

Approved:

____
Charles Torre, Ph.D.
Major Professor

____
Mark Riffe, Ph.D.
Committee Member

____
Jim Wheeler, Ph.D.
Committee Member

UTAH STATE UNIVERSITY

Logan, Utah

2021

ABSTRACT

Equivalence: A Covariantly Constant Problem in General Relativity

by

Jaren Hobbs

Utah State University, 2021

Major Professor: Dr. Charles Torre
Department: Physics

The goal of this project is to identify which of the known solutions to the Einstein field equations admit covariantly constant vector fields. These results serve to mitigate the equivalence problem in general relativity. This report includes discussion of background on general relativity, the equivalence problem, methods used, and results obtained; appendices include project code written to automate analysis, and data produced.

(30 pages)

PUBLIC ABSTRACT

Equivalence: A Covariantly Constant Problem in General Relativity

Jaren Hobbs

In studying the space-time structures described by Einstein’s theory of general relativity, it is often useful to identify particular properties referred to as *geometrical invariants*. These are attributes of the space-times which do not change regardless of the underlying coordinate systems used to study them. This project is part of a larger effort to catalogue space-times studied in general relativity. Specifically, computational software was used to identify structures known as covariantly constant vector fields.

Contents

List of Tables

List of Figures

Introduction

This project is part of a unique effort to create a single library of solutions to the Einstein field equations (EFE) embedded within a set of computational tools (Anderson & Torre, 2012, 2017). One of the benefits of this overall effort is the ability to execute semi-autonomous programmatic exploration of properties of EFE solutions en masse; as was done for this project. In particular, EFE solutions were analyzed to identify the vector space of covariantly constant vector fields, the dimensionality of such vector spaces are geometric invariants for those solutions. Identified geometric invariants are included in the library entries for their associated solutions. Comparing geometric invariants between EFE solutions can help to resolve the equivalence problem in general relativity (GR). This report covers GR background, the equivalence problem, methods, and results. Appendices include code written for the project and data produced.

Background on General Relativity

General relativity expresses “space” and “time” as one four-dimensional construct, as opposed to the Newtonian expression of a three-dimensional space evolving along an absolute time. The mathematical objects which are used to express GR space-time are referred to as the manifold and the metric. A manifold can be thought of as a collection of points and a set of rules describing the relationships between the points. In GR points are both spatial and temporal, and are referred to as events. The metric is the infinitesimal line element expressed as a symmetric rank-2 tensor which describes time intervals and distances between events in the manifold. In GR, the presence of energy, momentum, pressure, and stress causes curvature of space-time. A freely falling object in the curved geometry, in the “test particle” approximation, will follow a geodesic described by the metric. For a slow-moving object in weak gravitational field, if the energy density overwhelmingly comes from non-relativistic mass (where the energy equivalence for mass is given by $E = mc^2$), then projecting a space-time geodesic into a classical three-dimensional space produces a path which very closely approximates ballistic trajectories as described by Newton's Law of Universal Gravitation.

The metric may be thought of as describing the curvature of a space-time because the metric tells how to measure positional and directional changes on the manifold. From the metric we can derive the Christoffel connection $\Gamma^{\rho}{}_{\nu\mu}$, which describes the way the geometrical objects change between infinitesimally separated events. The connection is given by:

$$\Gamma^{\rho}{}_{\nu\mu} = \frac{1}{2} g^{\rho\eta}\left(\partial_\mu g_{\eta\nu} + \partial_\nu g_{\eta\mu} + \partial_\eta g_{\nu\mu}\right), \tag{1}$$

where g is the metric and ∂_ν indicates differentiation with respect to the ν^{th} coordinate. Indices in (1) and all subsequent equations are used in accordance with the Einstein summation convention (Misner, Thorne, & Wheeler, 1973). The covariant derivative can be used to describe the change in a vector field across a manifold and is the sum of the ordinary coordinate derivative of the vector field and a linear transformation from the connection. Consequently, the result of a covariant derivative retains its relationship to the vector field when both are subject to the same coordinate transformation. The covariant derivative of some vector field $\boldsymbol{A}$, in the μ^{th} coordinate direction, is given by:

$$\nabla_\mu \boldsymbol{A} = \left(\partial_\mu A^\rho + A^\eta \Gamma^{\rho}{}_{\eta\mu}\right)\boldsymbol{e}_\rho, \tag{2}$$

where A^ρ is the ρ^{th} component of $\boldsymbol{A}$ and $\boldsymbol{e}_\rho$ is the ρ^{th} coordinate basis vector.

For all events on the manifold, the deviation from flat Euclidean geometry is encoded in the Riemann curvature tensor:

$$R^{\rho}{}_{\mu\sigma\nu} = \partial_\sigma \Gamma^{\rho}{}_{\nu\mu} + \Gamma^{\rho}{}_{\sigma\eta}\Gamma^{\eta}{}_{\nu\mu} - \left(\partial_\nu \Gamma^{\rho}{}_{\sigma\mu} + \Gamma^{\rho}{}_{\nu\eta}\Gamma^{\eta}{}_{\sigma\mu}\right), \tag{3}$$

which arises from the commutator of covariant derivatives of the connection. Curvature is essentially a measure of the degree to which the covariant derivative operation is not commutative. A contraction of the Riemann curvature tensor yields the Ricci curvature tensor:

$$R_{\mu\nu} = R^{\eta}{}_{\mu\eta\nu}. \tag{4}$$

The trace of Ricci curvature tensor yields the scalar curvature:

$$R = g^{\eta\zeta} R_{\zeta\eta}. \tag{5}$$

The scalar curvature and the Ricci curvature tensor may be taken together to derive the Einstein tensor:

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu}. \tag{6}$$

The Einstein field equations are given by:

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = \kappa T_{\mu\nu}, \tag{7}$$

where Λ is the cosmological constant or vacuum energy density, which is associated with the expansion of the universe; $T_{\mu\nu}$ is the energy-momentum tensor, which encodes gravitational sources in GR; and κ is the Einstein gravitational constant, which serves a similar role to Newton's gravitational constant (G):

$$\kappa = \frac{8\pi}{c^4} \mathrm{G}, \tag{8}$$

where c is the speed of light in vacuum.

The form of the energy-momentum tensor depends upon the nature of the gravitational sources in the system. For example, the energy-momentum tensor for an electromagnetic field, without charges, is given by:

$$T_{\mu\nu} = \frac{1}{\mu_0}\left(g_{\mu\zeta} F^{\zeta\eta} F_{\nu\eta} - \frac{1}{4} g_{\mu\nu} F_{\sigma\rho} F^{\sigma\rho} \right), \tag{9}$$

where μ_0 is the vacuum permeability, and F is the electromagnetic tensor which encodes Maxwell's equations in GR space-time. Note the dependence of the energy-momentum tensor upon the metric. Overall, the EFEs are a set of ten, coupled, second-order, non-linear, partial differential equations. In general, an EFE solution is a metric which satisfies (7) for a specified energy-momentum tensor, where the matter fields in the energy-momentum tensor satisfy their own field equations. For (9), the electromagnetic tensor F would satisfy the covariant generalization of the source-free Maxwell equations, where coordinate derivatives become covariant derivatives.

The Equivalence Problem

The complicated geometries involved in GR make EFE solutions particularly vulnerable to what is referred to as the equivalence problem; that is to say, a physically unique solution may take many forms mathematically which look nothing alike but are all related by a coordinate transformation. As an example, consider the following line elements:

$$ds^2 = du^2 + dv^2, \qquad \textbf{(10)}$$

$$dl^2 = dj^2 + j^2 dk^2, \qquad \textbf{(11)}$$

$$dr^2 = da^2 + \sin^2 a \; db^2. \qquad \textbf{(12)}$$

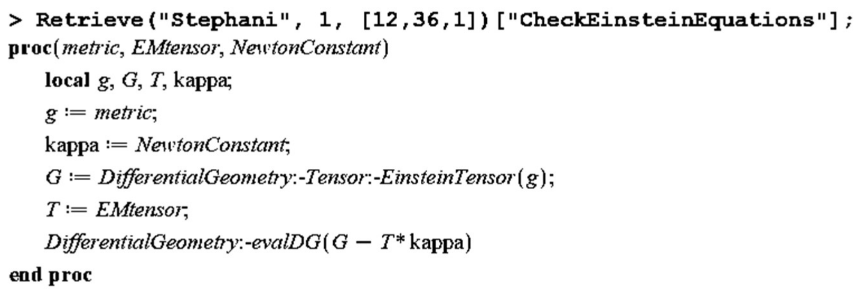
A line element represents infinitesimal displacement; integrating with respect to its square root may be used to determine the length of an arbitrary curve. It is not immediately apparent if (10), (11), and (12) describe the same space: that is to say, if they are related by a coordinate transformation. One can generate an arbitrary set of differentiable functions and attempt to solve for coordinate transformations, and such functions do exist between (10) and (11). However, there are no coordinate transformations from (12) to either (10) or (11). Explicitly proving such non-existence is difficult. Instead, one can resolve this equivalence problem by calculating coordinate invariant properties of the geometries described by these line elements. One of these coordinate invariants is the scalar curvature, which quantifies the deviation of the manifold from a flat Euclidean geometry. For both (10) and (11) the scalar curvature is 0, they are both two-dimensional Euclidean space, for (12) the scalar curvature is 2, it describes a two-sphere; so (12) cannot be equivalent to (10) and (11). For this example, the scalar curvature is sufficient to prove (10) is equivalent to (11). However, in higher dimensions there are additional coordinate invariants which could be calculated to increase confidence in the assertion that they are equivalent; but only one invariant needs to be different to prove they are not equivalent.

For EFE solutions, the coordinate transformation route to resolving the equivalence problem is usually prohibitively difficult. One of the invariants which may be explored to evaluate equivalence involves covariant derivatives. If there exists a vector field such that $\nabla_\mu \boldsymbol{A} = 0$, then $\boldsymbol{A}$ is a Covariantly Constant Vector Field (CCVF). This idea of covariantly constant can be thought of as a generalization of the behavior of unit vectors for Cartesian coordinates: the components of these Cartesian unit vectors are constant with respect to partial derivatives. Because CCVFs can be added and multiplied by constants to from new CCVFs, the set of CCVFs forms a vector space. Therefore, there exist basis-vectors such that any linear combination of these basis-vectors is a CCVF. Since the covariant derivative is coordinate independent, the dimensionality of the vector space of CCVFs is a geometric invariant and may be easily compared between EFE solutions. The goal of the analyses was to identify, for each EFE solution, the vector space of CCVFs.

Methods

The analyses were performed on EFE solutions stored in a digital library transcribed and maintained principally by Dr. Charles Torre; the 802 solutions analyzed are all from Hans Stephani's "Exact Solutions to Einstein's Field Equations, 2nd Edition" (Stephani et al., 2003). The library is a collection of nested tables with particular solutions indexed by author name, a reference number, and an equation number which may be taken from the source publication.

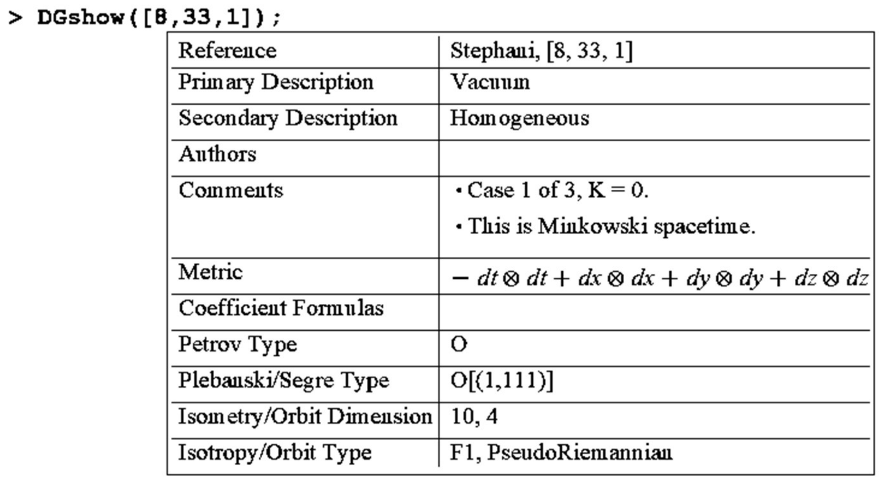
One of the motivations for creating the library was to have a collection of EFE solutions which could be easily verified. Included in every verified solution are the procedures by which this verification is accomplished. Users of the library may view and execute these procedures for themselves. Figure I shows the procedures for a homogeneous pure radiation spacetime.



```
> Retrieve("Stephani", 1, [12,36,1])["CheckEinsteinEquations"];
proc(metric, EMtensor, NewtonConstant)
    local g, G, T, kappa;
    g := metric;
    kappa := NewtonConstant;
    G := DifferentialGeometry:-Tensor:-EinsteinTensor(g);
    T := EMtensor;
    DifferentialGeometry:-evalDG(G − T* kappa)
end proc
```

FIGURE I – EXAMPLE OF EFE VERIFICATION PROCEDURE

With each entry are included known properties of the particular solution, most of which are coordinate invariant and can also be used to address equivalence. Figure II shows some of the data stored for the Minkowski spacetime.



```
> DGshow([8,33,1]);
```

Reference	Stephani, [8, 33, 1]
Primary Description	Vacuum
Secondary Description	Homogeneous
Authors	
Comments	• Case 1 of 3, K = 0. • This is Minkowski spacetime.
Metric	$-dt \otimes dt + dx \otimes dx + dy \otimes dy + dz \otimes dz$
Coefficient Formulas	
Petrov Type	O
Plebanski/Segre Type	O[(1,111)]
Isometry/Orbit Dimension	10, 4
Isotropy/Orbit Type	F1, PseudoRiemannian

FIGURE II – EXAMPLE OF LIBRARY DATA

Depending on the nature of the solution being considered, there are other entries which may be included, and some which may be omitted. Some of these other entries may be data on geometric invariants associated with the solutions. For solutions which are found to admit CCVFs, this data will be added to the library.

The library is stored within the Maple computer algebra system package DifferentialGeometry (DG), created and maintained principally by Dr. Ian Anderson. DG contains tools for the working within manifolds and with tensors. One of these tools is the procedure CovariantlyContantTensors (CCT), which takes a metric or connection and produces the system of linear partial differential equations for the components of the vector field $\boldsymbol{A}$ (or any other type of tensor) defined by the conditions $\nabla_\mu \boldsymbol{A} = 0$. CCT then uses the Maple tool pdsolve to solve for the components of $\boldsymbol{A}$ thereby defining a set of vectors that span the vector space of CCVFs. For this project, code was written to automate the analysis of EFE solutions using Maple and DG.

The project code was primarily organized as three subroutines (identified in Maple as *procedures*) with fairly uninspired names: LoopThroughFind (LTF), FindCCVF, and VerifyResults. The code in its entirety is in Appendix A. LTF is the top-level procedure which is called with an author name, reference number, list of equations, and a timeout parameter. The equation list is iterated through, passing the library index and timeout parameter to FindCCVF.

FindCCVF clears variables that are used to save either results or errors. FindCCVF then loads the metric and any coefficient information from the library, establishes the frame, and extracts the coordinate basis from the frame. It then uses DG to calculate the connection, and substitutes any coefficient information into the connection. CCT is then run inside of Maple's timelimit module. For some solutions, CCT could not complete in a reasonable amount of time; the longest any one solution was allowed to run was approximately 72 hours, CCT produced no result in that time. Due to the way Maple manages namespace for errors and the way frames are managed by DG; CCT results and errors both are saved as global variables and are not explicitly passed back to LTF. Upon completion, FindCCVF will exit, and LTF will pick up where it left off.

If the CCT result is null, LTF saves the solution index to a list of null results; another such list exists for any errors that were thrown in the attempt, saving the error message as well. If a result is found, LTF passes the solution index and result to VerifyResults. VerifyResults establishes a Boolean variable to track verification. It then takes each component of the result and calculates the covariant derivative. If any component produces a non-zero covariant derivative, the verification Boolean is set to false. Based on the status of the verification Boolean, solution index and CCT results are saved to lists for either verified or failed.

The following is a manual run through in Maple of the automated analysis. First the metric is retrieved from the library and the manifold is established.

```
> g := Retrieve("Stephani", 1, [12, 36, 1], manifoldname=M,
output=["Metric"])[1];
```

$$g := -2\, e^{2\,\rho x}\, du \otimes du + du \otimes dv + dv \otimes du + dx \otimes dx + dy \otimes dy$$

This metric is part of a solution to the Einstein-Maxwell equations. Next the coefficient information is retrieved from the library, these are generally variable relationships which explicitly define parameters which may be listed in general forms in the metric.

```
M > coef := Retrieve("Stephani", 1, [12, 36,
1])["CoefficientInfo"];
```

$$coef := [\ ]$$

Note that the manifold name has been added to the command prompt, allowing the user to easily identify their current frame if they establish and move between multiple frames. For this solution, there is no coefficient information, so substitution as seen in the project code would be superfluous. It is safe to run a substitution command with an empty set, as the result will not be altered.

```
M > g := subs(op(coef),g);
```

$$g := -2\,\mathrm{e}^{2\_\rho x}\,du \otimes du + du \otimes dv + dv \otimes du + dx \otimes dx + dy \otimes dy$$

Next the coordinate basis will be extracted from the frame which was generated as part of establishing the manifold. Although the manifold is coordinate independent, it is necessary to embed it in a reference frame with coordinates for the system and users to interact with objects on the manifold. For the purposes of these calculations, the manifold and frame are not meaningfully distinct.

```
M > Base := DGinfo("FrameBaseVectors");
```

$$Base := \left[\partial_u, \partial_v, \partial_x, \partial_y\right]$$

Then the connection is calculated.

```
M > CC := Christoffel(g);
```

$$CC := \nabla_{\partial_u} \partial_u = 2\_\rho\,\mathrm{e}^{2\_\rho x}\,\partial_x,\ \nabla_{\partial_u} \partial_x = -2\_\rho\,\mathrm{e}^{2\_\rho x}\,\partial_v,\ \nabla_{\partial_x} \partial_u = -2\_\rho\,\mathrm{e}^{2\_\rho x}\,\partial_v$$

Note the result is a comma-separated list, positions of the partial differential operators and their coordinates variables in reference to “FrameBaseVectors” determines the index positions and values which would be associated with the form of the connection from equation (1). If the “FrameBaseVectors” are taken to relate to their respective coordinate bases from left to right by the numbers 1 through 4, such that u is 1 and y is 4; and the indices on the connection are treated as reference positions like in a matrix: then the values seen in the Maple output for “Christoffel” associate the coordinate of the partial on the right-hand side with the raised connection index, and the two coordinates on the left-hand side with the two lowered indices, any positions not referenced would be zero. For this metric, the connection has only non-zero values of $\Gamma^3{}_{11} = 2\rho\, e^{2\rho x}$, $\Gamma^2{}_{13} = -2\rho\, e^{2\rho x}$, and $\Gamma^2{}_{31} = -2\rho\, e^{2\rho x}$.

Next the CCVF is calculated.

```
M > CCVF := CovariantlyConstantTensors(CC,Base);
```

$$CCVF := \left[\partial_y, \partial_v\right]$$

The CCT results are displayed as a list of vector fields which forms a basis for the vector space of CCVFs. The components of these vector fields in the current frame may be extracted with another DG command. This is the form of the deliverable data for the project.

```
M > GetComponents(CCVF,Base);
```

$$[[0, 0, 0, 1], [0, 1, 0, 0]]$$

Lastly the results will be verified as covariantly constant by calculating the covariant derivative of each basis vector.

```
M > CovariantDerivative(CCVF[1],CC);
```

$$0\,\partial_u \otimes du$$

```
M > CovariantDerivative(CCVF[2],CC);
```

$$0\,\partial_u \otimes du$$

Both covariant derivatives are zero, verifying that each basis vector is covariantly constant, and so that any linear combination of the basis vectors is also covariantly constant.

Results

Of the 802 EFE solutions from the library: 58 were found to admit CCVFs; 604 produced a null result; 109 timed-out; 21 produced various errors generally attributed to transcription errors in the library; 9 were found to be based on metrics with generic functions; and one was a reference to a duplicated problem as listed in original source material. Null results may indicate either a CCVF does not exist for the solution, or Maple was unable to resolve the associated differential equations. Of those solutions which timed-out, four were selected at random and allowed to run without a time-out parameter; processing was interrupted at approximately 24 hours for these without any CCT results having been returned. Time-out is generally attributed to symbolic complexity exceeding the capabilities of pdsolve algorithms. Transcription errors may have occurred while coping the data into the library, and were only identified for solutions which did not have included verification procedures. Generic functions in the metric were in terms of undefined parameters which could possibly have been functions of the coordinate base for the solution. As these parameters were not defined, differentiation of the generic functions with respect to the coordinate base resulted in zeroes which may not be consistent with the conditions of the system. The library already contained comments regarding these solutions indicating the metric is not available.

For the solutions which do permit CCVFs, the equation number and components of a basis of CCVFs were extracted as plain text and are the primary deliverable data produced from the project. This data will be integrated into the EFE solutions library.

Equation	CCVF Basis Components	Dimensionality	Scalar Curvature
[8, 33, 1]	[[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0]]	4	0
[12, 36, 1]	[[0, 0, 0, 1], [0, 1, 0, 0]]	2	0
[24, 47, 1]	[[0, 0, 0, 1], [_b/_a, 1, 0, 0]]	2	0
[12, 8, 4]	[[0, -cos(y), 1/x*sin(y), 0], [0, sin(y), 1/x*cos(y), 0]]	2	$2/B^2$
[12, 8, 5]	[[0, -cos(y), 1/x*sin(y), 0], [0, sin(y), 1/x*cos(y), 0]]	2	$-2/B^2$

TABLE I – SAMPLE OF PROJECT RESULTS

Table I includes a small selection of the data produced, the entire set is in Appendix B. This data helps demonstrate the utility of using the dimensionality of the space of CCVFs to address the equivalence problem. The equation numbers are taken directly from the source material, and are organized by chapter and section numbers. The second column gives the components for a basis of CCVFs; any linear combination of these sets is a CCVF. However, the precise form of any particular CCVF is not relevant to equivalence, nor are the components of the vectors which span the space of CCVFs, since these are coordinate dependent. It is the dimensionality of the space which is significant, and only insofar as it shows which solutions are not equivalent. Consider the solutions referenced in Table I. There is no coordinate information associated with the CCVF coefficients, so the particular values listed may be misleading. In fact, equations [12, 36, 1] and [24, 47, 1] are known to describe the same solution; where [12, 8, 4] and [12, 8, 5] are known to be distinct solutions. In contrast, [8, 33, 1] may be immediately identified as distinct from the others because it is the only one with a four-dimensional vector space of CCVFs. As an example of how these invariants may be taken together to address the equivalence problem, the scalar curvatures for each solution have also been included in the table. Scalar curvature alone would not show [8, 33, 1] to be different than [12, 36, 1] and [24, 47, 1]; though CCVF dimensionality would. And where CCVF dimensionality would not be sufficient to determine [12, 8, 4] and [12, 8, 5] are different, scalar curvature would. Scalar curvature and CCVF dimensionality together do not guarantee [12, 36, 1] and [24, 47, 1] are equivalent, but together they increase confidence in the assertion.

Though verification of CCVFs was performed as part of the automated analysis; additional verification of zero covariant derivative was performed. The deliverable data for the project was manually extracted for the 58 solutions with CCVFs. This data was then imported back into Maple from a spreadsheet and run through the code included at the end of Appendix A. The covariant derivative of each element of the basis was displayed: all are zero. This additional verification also helps ensure that the deliverable data may be merged into the library source without introducing formatting errors.

Concluding Remarks

The code in its entirety is included in Appendix A, so it is available for future researchers and may be modified or used as a model. These same general procedures may be employed to calculate additional invariants for the EFE solutions in the library en masse.

This project has also served as a proof of concept for executing semi-autonomous programmatic analysis on large blocks of solutions in the library. The verification procedure in particular, which checks against a null tensor produced by the DGzero command, could easily be modified for other vanishing results of co- and contra-variant operations on tensors intrinsic to EFE solutions. For example, "scalar curvature invariants," which can be produced by taking various contractions of polynomials in the Riemann curvature tensor. If a particular contraction vanishes in one coordinate system, then it will vanish in all coordinate systems. In this way one can use scalar curvature invariants much as CCVFs were used in this report.

References

I. M. Anderson and C. G. Torre, New symbolic tools for differential geometry, gravitation, and field theory J. Math. Phys. 53, 013511 (2012); DOI: 10.1063/1.3676296.

I. Anderson and C. Torre, Introduction to the USU Library of Solutions to the Einstein Field Equations, https://digitalcommons.usu.edu/dg_tutorial/8/ , (2017).

Misner, C. W.; Thorne, K. S.; Wheeler, J. A. (1973). Gravitation. San Francisco: W. H. Freeman. ISBN 978-0-7167-0344-0.

Stephani, H; Kramer, D; MacCallum, M. A. H.; Hoenselaers, C. A. and Herlt, E. (2003). Exact solutions of Einstein's field equations, 2nd edition. Cambridge University Press, Cambridge. ISBN 0-521-46136-7

APPENDICIES

Appendix A – Project Code

```
LoopThroughFind := proc(Author,Reference,List,timlim)
    description "Loop given list elements through FindCCVF":
    global CCVFnull, CCVFtimed, CCVFerror, CCVFverified, CCVFfailed, CCVFweird, g, coef, M, CCVF, errored:
    local Equation, index, partial, now:

    index := 1:
        #Establish indexing variable
    for Equation in List do
        #For each equation in the List of solutions to the Einstein Field Equations
        index := index + 1:
            #Increment index
        partial := List[index..]:
            #Save portion of list not attempted
        try
            #Try to find CCVF
            FindCCVF([Author, Reference, Equation],timlim):
                #Run FindCCVF on the current solution
            if evalb(CCVF = []) or evalb(CCVF = ())
                #If the output is an empty list
                then CCVFnull := [op(CCVFnull), [Author, Reference, Equation, [], []]]:
                    #Add to the list of equations without a CCVF
            elif evalb(errored[-1] = "time expired")
                #If time expired
                then CCVFtimed := [op(CCVFtimed), [Author, Reference, Equation, [CCVF], [errored]]]:
                        #Add to the list of equations with FindCCVF timed out on
            elif (type(errored[1],procedure) or type(errored[1],`module`))
                #First entry in other error messages is procedure or module name
                then   CCVFerror := [op(CCVFerror), [Author, Reference, Equation, [CCVF], [errored]]]:
                    #Add to the list of solutions that errored out in FindCCVF
            else
                #Otherwise it worked
                VerifyResults([Author, Reference, Equation, [CCVF], []]):
                    #Verify CCVF
            end if:
        catch:
            #Otherwise an error is thrown just trying to pass to FindCCVF
            CCVFerror := [op(CCVFerror), [Author, Reference, Equation, [], [lastexception]]]:
                #Add to the list of solutions that errored out in FindCCVF
        end try:
        now:=Date():
            #Mark date for checking progress from another maple server
        save(now, partial, CCVFnull, CCVFtimed, CCVFerror, CCVFverified, CCVFfailed, "Stephani_1_partial.m"):
            #save partial progress
    end do:
end proc:
```

```
FindCCVF := proc (Solution,timlim)
    description "Given a Solution from The USU library of Solutions to the Einstein Field Equations, as a list of Author, reference, and equation number; outputs vector field basis for which the covariant derivative is zero with respect to the Connection defined by metric.":
    global g, coef, M, CCVF, errored:
    local Base, CC:
    uses DifferentialGeometry, DifferentialGeometry:-Tensor, DifferentialGeometry:-Library, DifferentialGeometry:-Tools:

    unassign('errored'):
        #Clear variable for error tracking
    unassign('CCVF'):
        #Clear variable for CCVF
    try
        #Try with coefficient info
        g, coef := op(Retrieve(op(Solution), manifoldname = M, output = ["Metric", "CoefficientInfo"])):
            #Establish frame and load metric from library.
    catch:
        #Otherwise give it an empty set for coefficients
        g, coef := op(Retrieve(op(Solution), manifoldname = M, output = ["Metric"])), []:
            #Establish frame and load only metric is entry does not include coefficient information.
    end try:
    try:
        #Try to find CCVF
        Base := DGinfo("FrameBaseVectors"):
            #Identify the basis for the frame.
        CC := Christoffel(g):
            #Calculate the connection
        CC := subs(op(coef),CC):
            #Substitute coefficient information into the connection
        CCVF := timelimit(timlim, CovariantlyConstantTensors(CC,Base)):
            #Find CCVFs timeout after # seconds
    catch:
        #Otherwise save error
      errored := lastexception:
            #Save error
    end try:
end proc:
```

```
VerifyResults := proc(Result)
    description "Verify the results for each solution in the provided group.":
    local BaseSet, Component, CoDev, IsVer:
    global CCVFverified, CCVFfailed, CCVFerror, g, coef, M:
    uses DifferentialGeometry, DifferentialGeometry:-Tensor, DifferentialGeometry:-Library, DifferentialGeometry:-Tools:

    try
        IsVer := true:
            #bool for if verified
        for BaseSet in Result[4] do
            #for each set of basis produced by CCT
            for Component in BaseSet do
                #for each individual base vector
                CoDev := subs(op(coef),CovariantDerivative(Component,Christoffel(g))):
                    #calculate the covariant derivative and substitute coefficients
                IsVer := IsVer and DGequal(CoDev, DGzero["DGvector"](M)):
                    #if the covariant derivative is zero, and the covariant derivative for each previous base was also zero
            end do:
        end do:
    catch:
        IsVer := false:
            #if an error occured assume not verified
    end try:
    if IsVer then
        #if verified
        CCVFverified := [op(CCVFverified), Result]:
            #save to list of verified solutions
    else
        #if verification failed or errored
        CCVFfailed := [op(CCVFfailed), Result]:
            #save to list of failed solutions
    end if:
end proc:
```

```
i:=1:
#Initialize index variable
for n in Array(1..58) do
    #Iterate through loop 58 times
    i:=i+1:
        #Increment index
    g,coef:=op(Retrieve("Stephani",1,parse(ver[i][1]),manifoldname=M,output=["Metric","CoefficientInfo"])):
        #Retrieve metric and coefficient info
    base:=DGinfo("FrameBaseVectors"):
        #Retrieve coordinate system from frame
    CCVFcoef:=parse(ver[i][2]):
        #Extract sets of CCVF space base coefficients from results
    for each in CCVFcoef do
        #For each set in results
        CCbase:=DGzip(each,base):
            #Create vector from base
        CD:=CovariantDerivative(CCbase,Christoffel(g)):
            #Calculate covariant derivative
        CDsub:=subs(op(coef),CD):
            #Substitute coef in covariant derviative
        print(DGsimplify(CDsub));
            #Simplify and display covariant derviative
    end do:
end do:
```

Appendix B – Data Produced

Equation	Covariantly Constant Vector Field Basis Components	Dimensionality
[8, 33, 1]	[[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0]]	4
[12, 7, 1]	[[0, 1, 0, 0]]	1
[12, 8, 2]	[[1/z*exp(-t), 0, 0, exp(-t)], [-1/z*exp(t), 0, 0, exp(t)]]	2
[12, 8, 4]	[[0, -cos(y), 1/x*sin(y), 0], [0, sin(y), 1/x*cos(y), 0]]	2
[12, 8, 5]	[[0, -cos(y), 1/x*sin(y), 0], [0, sin(y), 1/x*cos(y), 0]]	2
[12, 8, 7]	[[1/z*exp(-t), 0, 0, exp(-t)], [-1/z*exp(t), 0, 0, exp(t)]]	2
[12, 12, 1]	[[0, 1, 0, 0]]	1
[12, 12, 2]	[[0, 1, 0, 0]]	1
[12, 12, 3]	[[0, 1, 0, 0]]	1
[12, 12, 4]	[[0, 1, 0, 0]]	1
[12, 13, 1]	[[0, 1, 0, 0]]	1
[12, 23, 1]	[[1, 0, 0, 0]]	1
[12, 23, 2]	[[1, 0, 0, 0]]	1
[12, 24.1, 1]	[[1, 0, 0, 0]]	1
[12, 24.2, 1]	[[1, 0, 0, 0]]	1
[12, 24.3, 1]	[[1, 0, 0, 0]]	1
[12, 26, 1]	[[0, 0, 1, 0]]	1
[12, 36, 1]	[[0, 0, 0, 1], [0, 1, 0, 0]]	2
[12, 37, 1]	[[0, 1, 0, 0]]	1
[12, 37, 2]	[[0, 1, 0, 0]]	1
[12, 37, 3]	[[0, 1, 0, 0]]	1

Equation	Covariantly Constant Vector Field Basis Components	Dimensionality
[12, 37, 4]	[[0, 1, 0, 0]]	1
[12, 37, 5]	[[0, 1, 0, 0]]	1
[12, 37, 6]	[[0, 0, 0, 1]]	1
[12, 37, 7]	[[0, 1, 0, 0]]	1
[12, 37, 8]	[[0, 1, 0, 0]]	1
[12, 37, 9]	[[0, 1, 0, 0]]	1
[15, 18, 1]	[[0, 0, 0, 1]]	1
[18, 65, 2]	[[0, 0, 1, 0]]	1
[22, 5, 1]	[[0, 0, 0, 1]]	1
[22, 70, 1]	[[0, 0, 0, 1]]	1
[24, 35, 1]	[[0, 1, 0, 0]]	1
[24, 40, 1]	[[0, 0, 0, 1]]	1
[24, 40, 5]	[[0, 0, 0, 1]]	1
[24, 40, 6]	[[0, 0, 0, 1]]	1
[24, 40, 7]	[[0, 0, 0, 1]]	1
[24, 40, 8]	[[0, 0, 0, 1]]	1
[24, 40, 10]	[[0, 0, 0, 1]]	1
[24, 40, 11]	[[0, 0, 0, 1]]	1
[24, 47, 1]	[[0, 0, 0, 1], [_b/_a, 1, 0, 0]]	2
[24, 51, 1]	[[0, 1, 0, 0]]	1
[35, 19, 1]	[[0, 0, 0, 1]]	1
[35, 33, 1]	[[0, 0, 0, 1]]	1

Equation	Covariantly Constant Vector Field Basis Components	Dimensionality
[35, 34, 1]	[[0, 0, 0, 1]]	1
[35, 34, 2]	[[0, 0, 0, 1]]	1
[35, 73, 1]	[[0, 1, 0, 0]]	1
[36, 34, 1]	[[1, 0, 0, 0]]	1
[37, 40, 1]	[[0, 0, 0, 1]]	1
[37, 64, 1]	[[0, 1, 0, 0]]	1
[37, 65, 1]	[[0, 0, 0, 1]]	1
[37, 66, 1]	[[0, 1, 0, 0]]	1
[37, 104, 1]	[[0, 0, 0, 1]]	1
[24, 40, 3]	[[0, 0, 0, 1]]	1
[35, 80, 1]	[[0, 0, 0, 1]]	1
[12, 29, 1]	[[1, 0, 0, 0]]	1
[22, 34, 3]	[[0, 0, 1, 0]]	1

Equation	Covariantly Constant Vector Field Basis Components	Dimensionality
[17, 24, 1]	[[-exp(t)*rho/((rho^2+z^2)^(1/2)+z)^(1/2), -2*(z^2+z*(rho^2+z^2)^(1/2)+1/2*rho^2)*exp(t)/((rho^2+z^2)^(1/2)+z)^(3/2), 0, exp(t)/((rho^2+z^2)^(1/2)+z)^(1/2)], [exp(-t)*rho/((rho^2+z^2)^(1/2)+z)^(1/2), exp(-t)*(2*z*(rho^2+z^2)^(1/2)+rho^2+2*z^2)/((rho^2+z^2)^(1/2)+z)^(3/2), 0, exp(-t)/((rho^2+z^2)^(1/2)+z)^(1/2)], [-cos(phi)*((rho^2+z^2)^(1/2)+z)^(1/2), 1/((rho^2+z^2)^(1/2)+z)^(1/2)*cos(phi)*rho, sin(phi)/rho*((rho^2+z^2)^(1/2)+z)^(1/2), 0], [sin(phi)*((rho^2+z^2)^(1/2)+z)^(1/2), -1/((rho^2+z^2)^(1/2)+z)^(1/2)*sin(phi)*rho, cos(phi)/rho*((rho^2+z^2)^(1/2)+z)^(1/2),0]]	4
[24, 38, 1]	[[-1, -xi1*xi, xi/r, xi1/r], [0, -xi, 0, 1/r], [0, -xi1, 1/r, 0], [0, 1, 0, 0]]	4

VITAE

Jaren Hobbs

OVERVIEW

Six years of experience operating a field deployable biosurveillance laboratory for the US Army. Three years contracted as an engineering analyst for the Air Force Nuclear Weapons Center through BAE Systems Incorporated. Motivated and versatile individual recognized for technical expertise and efficiency.

EDUCATION

Bachelor of Science Physics — Fall 2013

- Utah State University — *Logan, Utah*
- Research Project
    - Forward Model for Temperature Derivation from Atmospheric Lidar

Master of Science Physics — Summer 2021

- Utah State University — *Logan, Utah*
- Research Project
    - Equivalence: A Covariantly Constant Problem in General Relativity

Military Education

- Chemical, Biological, Radiological & Nuclear Specialist
    - Nord Hall Training Center — *Fort Leonard Wood, Missouri*
    - Placed on the Commandant's List
    - Earned an Army Achievement Medal for outstanding performance
- Biological Integrated Detection System Operator
    - United States Army Chemical School — *Fort McClellan, Alabama*

RELEVANT SKILLS & EXPERIENCE

Engineering Analyst

- Evaluate engineering proposals, maintenance/operational procedures, technical data
- Verify compliance with applicable military and industrial system safety standards
- Ensure weapon system security and safety as defined by DoDM 3150.02
- Identify potential fault and threat vectors in new or updated designs and procedures

Training Coordinator

- Wrote and conducted Nuclear Surety training, emphasis on Normal Accident Theory
- Conducted military operational sustainment retraining
- Briefed command officers on system capabilities and emplacement strategies
- Explained scientific principles of biodetection to senior enlisted personnel

Lab Technician

- Operate, monitor and maintain automated biosurveillance equipment
- Perform manual inoculation of immunoassays and process samples
- Safely package and transport potential biological contagions
- Maintain operation logs and evidence chain of custody forms

Troubleshooting

- Access and evaluate system generated error codes
- Visually and physically inspect equipment for faults

Maintenance

- Field replacement of core system components
- Preventative maintenance checks and services

Leadership

- Assistant leader for a remote, autonomous team
- Fulfilled a non-commissioned officer's shift leader position

Computer Science

- Experience programming in IDL, C++, and Python
- Data analysis in IDL, Maple, and Spyder CAS