1-18-2012

New symbolic tools for differential geometry, gravitation, and field theory

Ian Anderson
Utah State University

Charles Torre
Utah State University

New symbolic tools for differential geometry, gravitation, and field theory

I. M. Anderson[1] and C. G. Torre[2,a)]
[1]*Department of Mathematics and Statistics, Utah State University, Logan, Utah 84322, USA*
[2]*Department of Physics, Utah State University, Logan, Utah 84322, USA*

(Received 21 March 2011; accepted 21 December 2011; published online 18 January 2012)

DifferentialGeometry is a *Maple* software package which symbolically performs fundamental operations of calculus on manifolds, differential geometry, tensor calculus, spinor calculus, Lie algebras, Lie groups, transformation groups, jet spaces, and the variational calculus. These capabilities, combined with dramatic recent improvements in symbolic approaches to solving algebraic and differential equations, have allowed for development of powerful new tools for solving research problems in gravitation and field theory. The purpose of this paper is to describe some of these new tools and present some advanced applications involving: Killing vector fields and isometry groups, Killing tensors, algebraic classification of solutions of the Einstein equations, and symmetry reduction of field equations. © *2012 American Institute of Physics*. [doi:10.1063/1.3676296]

I. INTRODUCTION

DifferentialGeometry is a *Maple* software package which symbolically performs fundamental operations of calculus on manifolds, differential geometry, tensor calculus, spinor calculus, Lie algebras, Lie groups, transformation groups, jet spaces, and the variational calculus. These capabilities, combined with dramatic recent improvements in symbolic approaches to solving algebraic and differential equations, have allowed for development of powerful new tools for solving research problems in gravitation and field theory. The purpose of this paper is to briefly describe some of these new tools and present some advanced applications.

In Sec. II, we consider the problem of finding Killing vector fields and determining the structure of the associated isometry group. As an example, we show that the Gödel spacetime is a reductive homogeneous space. Also demonstrated in Sec. II is how to use *DifferentialGeometry* to perform case-splitting analysis, which isolates exceptional values of functions and parameters. This latter point is illustrated with a family of explicit vacuum metrics in the Kundt class which generically admit no isometries.

In Sec. III, we focus on Killing tensors as examples of local invariants of (pseudo-)Riemannian manifolds which can be computed with *DifferentialGeometry*. We show that all rank 2 and rank 3 Killing tensors of the Gödel spacetime are reducible, i.e., they are built from symmetric products of rank-1 Killing tensors.

In Sec. IV, we consider the algebraic classification of curvature tensors. We focus on the ability to perform case splitting when solving equations symbolically to analyze a family of hypersurface homogeneous Einstein spacetimes found in Ref. 11. With one minor correction, we confirm the results there and we show that the family of metrics includes some additional Einstein metrics and Petrov types at exceptional values of parameters appearing in the metric.

Finally, in Sec. V, we consider the problem of reducing field equations by a prescribed group of symmetries. We do this in the context of the Einstein equations for fully nonlinear gravitational

[a)] Author to whom correspondence should be addressed. Electronic mail: torre@cc.usu.edu.

plane waves, which are defined by a 5 parameter isometry group. In particular, we show how *DifferentialGeometry* can be used to find the most general group-invariant metric and to find the residual symmetry group admitted by the reduced field equations. To our knowledge, these symbolic tools, which implement the theory of Ref. 2, are completely new and unique to *DifferentialGeometry*.

The applications we present illustrate a number of important features of *DifferentialGeometry*. First and foremost, it is integrated across mathematical sub-disciplines. The same command is used to create manifolds, manifolds with anholonomic frames, vector bundles, jet spaces, homogeneous spaces, Lie algebras, and Lie algebras with coefficients. All these different computational environments can coexist within the same worksheet. Command syntax is integrated across these environments. We note that many commands support tracking of intermediate computations, and there is an extensive set of utilities to support the rapid development of custom programs. *DifferentialGeometry* also includes an array of Libraries containing tables of abstract Lie algebras, Lie algebras of vector fields, differential equations, and solutions of the Einstein equations (including matter couplings). We also note that the underlying code for all *DifferentialGeometry* commands can be explicitly displayed so that, if desired, one can have complete control over the details of all computations. As part of the *Maple* distributed library, the *DifferentialGeometry* software is constantly tested and maintained.

In *DifferentialGeometry*, all tensor fields, spinor fields, connections, and so forth are built and displayed using index-free notation as is standard in the mathematical literature. Since *DifferentialGeometry* runs as a package in *Maple* (see, e.g., Ref. 7), the syntax of all commands and output which we describe reflects this.[14] We have taken a few liberties with the output of some of the commands to make it more readable in this venue. Much of the capabilities described here appear in *Maple*14 and *Maple*15. Full documentation and tutorials can also be found in these releases. An expanded version of the present paper is available.[4]

II. ISOMETRIES AND KILLING VECTOR FIELDS

Let g be a metric on a manifold M and G a Lie group. A *group action* is a smooth mapping μ: $G \times M \to M$. The *isometry group* of the metric g is a group action which preserves the metric, i.e., for each $h \in G$, $\mu_h^* g = g$. The infinitesimal generators of the isometry group are the Killing vector fields V, which satisfy an over-determined system of linear partial differential equations (PDEs), the Killing equations:

$$L_V g_{ab} = \nabla_a V_b + \nabla_b V_a = 0. \tag{2.1}$$

Using the vector field commutator as the Lie bracket, the vector space of Killing vector fields forms a Lie algebra $\mathfrak{g}$ isomorphic to the Lie algebra of G. In this section, we show how to symbolically determine Killing vector fields and analyze the corresponding isometry group.

A. Computation of Killing vector fields

We consider the Gödel metric, which is a homogeneous perfect fluid solution of the Einstein equations. Expressed in coordinates (t, x, y, z), it takes the form

$$g = -(dt + e^x dz) \otimes (dt + e^x dz) + dx \otimes dx + dy \otimes dy + \frac{1}{2} e^{2x} dz \otimes dz. \tag{2.2}$$

The metric can be defined within *DifferentialGeometry* via

```
> DGsetup([t, x, y, z], M):
> omega := evalDG(dt + exp(x)*dz):
> g := evalDG(- omega &t omega + dx &t dx + dy &t dy + (1/2)
   *exp(2*x)*dz &t dz):
```

The first command defines a chart M. The second command defines a convenient differential 1-form. The third command defines the metric. The asterisk denotes scalar multiplication and the symbol `&t` denotes tensor multiplication.

The command `KillingVectors(g)` computes the Killing Eqs. (2.1) of the metric g and passes the resulting partial differential equations to the *Maple* differential equation solver routines. Applied to the Gödel metric, the output is a list of 5 Killing vector fields which form a basis for the vector space of solutions to Eqs. (2.1):

```
> KV := KillingVectors(g);
```

$$KV := [D_z, D_t, D_x - zD_z, -4e^{-x}D_t + 2zD_x + (-z^2 + 2e^{-2x})D_z, D_y].$$

Here, $D_z \equiv \frac{\partial}{\partial z}$ and so forth. One can verify that these vector fields are indeed Killing vector fields by taking the Lie derivative of the metric with the `LieDerivative` command.

```
> LieDerivative(KV, g);
```

$$[0\,dt\,dt, 0\,dt\,dt, 0\,dt\,dt, 0\,dt\,dt, 0\,dt\,dt],$$

where this list of five vanishing tensors corresponds to the Lie derivative of the metric with respect to each of the vector fields in the list KV.

B. Properties of isometry algebras: Gödel is a reductive homogeneous space

Given the Killing vector fields admitted by a metric, the isometry group they generate can be characterized using a number of techniques available in the *DifferentialGeometry* package. Such a characterization is useful, of course, for better understanding the properties of the geometry defined by the metric. But characterization of the isometry group is also important for analyzing the equivalence problem for metrics, since the isometry group is a diffeomorphism invariant of the metric. Characterization of the isometry group naturally splits into two parts: characterization of the abstract group G and characterization of the group action μ. We illustrate this with the Gödel spacetime whose Killing vector fields we have just determined.

We can characterize the abstract group G by determining its Lie algebra and then computing its Levi decomposition. Recall that every Lie algebra $\mathfrak{g}$ can be decomposed into a semi-direct sum of a unique solvable ideal and a semi-simple Lie algebra, respectively. Given the Killing vectors KV of the Gödel metric, we define the Lie algebra as a data structure LD, which stores the vector space and brackets which define the Lie algebra $\mathfrak{g}$.

```
> LD := LieAlgebraData(KV, A);
> DGsetup(LD):
```

$$LD := [[e1, e3] = -e1, \quad [e1, e4] = 2\,e3, \quad [e3, e4] = -e4].$$

The Lie algebra is now defined internally with a basis $(e_1, e_2, \ldots, e_5)$ corresponding to the 5 vector fields in KV. We next compute the Levi decomposition of $\mathfrak{g}$

```
> LeviDecomposition(A);
```

$$[[e2, e5], [e1, e3, e4]].$$

The output displays a basis for the solvable ideal, $[e2, e5]$, and a basis for a complementary semi-simple sub-algebra, $[e1, e3, e4]$, in terms of the original basis $[e1, \ldots, e5]$ of $\mathfrak{g}$. It is easy to see by inspection of the Lie brackets in $\mathfrak{g}$ that the Levi decomposition is in this case actually a direct sum decomposition of the solvable algebra and $sl(2, R)$. Alternatively, the *DifferentialGeometry* command `Decompose` could have been used to find this direct sum decomposition.

Let us now examine some features of the group action μ. Two salient properties of any group action are the dimension of the orbits and the isotropy subgroups. The dimension of the orbits are easily seen by inspection to be four in the Gödel example—the Killing vectors span the tangent space at each point, so the spacetime is a homogeneous space. In more complicated examples the dimension and signature of the orbits can be computed using the *DifferentialGeometry* command `SubspaceType`. Let us focus on the isotropy subgroups.

Recall that the isotropy group of a point $p \in M$ is the subgroup $G_p \subset G$ such that $\mu_h(p) = p, \forall h \in G_p$. The isotropy subalgebra $\mathfrak{g}_p \subset \mathfrak{g}$ of a point $p \in M$ is the subalgebra of Killing vector fields which vanish at p. This subalgebra can be determined from a given algebra of vector fields with the `IsotropySubalgebra` command. Here we compute the isotropy subalgebra of the origin for the Killing vector fields *KV* of the Gödel metric and express it in terms of the algebra *A* defined above:

```
> Iso := IsotropySubalgebra(KV, [t=0, x=0, y=0, z=0],
    output=[''Vector'', A]);
```

$$Iso := [(2e^{-x} - 2)D_t - zD_x + ((1/2)z^2 - e^{-2x} + 1)D_z], \quad [e1 - 2\,e2 - \frac{1}{2}\,e4].$$

Thus there is a one dimensional isotropy subgroup of the origin whose infinitesimal generator is the Killing vector field,

$$V = 2(e^{-x} - 1)\partial_t - z\partial_x + (\frac{1}{2}z^2 - e^{-2x} + 1)\partial_z = e1 - 2\,e2 - \frac{1}{2}\,e4. \tag{2.3}$$

Since G acts by isometry, the isotropy subgroup of a point $p \in M$ preserves the inner product defined at T_pM by the Lorentz-signature metric, and hence can be identified with a subgroup of the Lorentz group. Infinitesimally, the isotropy subalgebra is a subalgebra of the Lorentz algebra consisting of combinations of boosts, spatial rotations, and null rotations. All such subalgebras have been classified up to conjugation[10] (for four dimensional spacetimes). Given a set of Killing vector fields and a point $p \in M$ the `IsotropyType` command computes this classification of the isotropy subalgebra of p. For the Killing vector fields *KV* of the Gödel metric the isotropy subalgebra of the origin is one dimensional and classified by

```
> IsotropyType(KV, [t=0, x=0, y=0, z=0]);
```

$$\mathrm{F12} \tag{2.4}$$

Since the Gödel spacetime is homogeneous, this is in fact the isotropy type of any point. In the classification scheme of Ref. 10, the subalgebra F12 corresponds to the generator of a spatial rotation. Thus the Gödel spacetime is a locally rotationally symmetric homogeneous space.

Another important property of the group action, which characterizes how the isotropy group sits as a subgroup of G, is whether its Lie algebra $\mathfrak{h} \equiv \mathfrak{g}_p$ admits a complement $\mathfrak{m}$ in $\mathfrak{g}$ such that $[\mathfrak{h}, \mathfrak{m}] \subset \mathfrak{m}$. If so, the vector space decomposition $\mathfrak{g} = \mathfrak{h} + \mathfrak{m}$ is called a *reductive* decomposition of $\mathfrak{g}$ and the group orbit is a *reductive homogeneous space*.[8] We now use *DifferentialGeometry* to show that the isometry algebra of the Gödel spacetime does admit a reductive decomposition. We have already symbolically defined the isometry algebra *A* and its isotropy subalgebra *Iso*; we now compute the general form of its complement:

```
> H := Iso[2];
```

$$H := e_1 - 4e_2 + e_4$$

```
> M := ComplementaryBasis(H, [e1, e2, e3, e4, e5], t);
```

$$M := [(t_1 + 1)e_1 - 2t_1e_2 - \frac{1}{2}t_1e_4, t_2e_1 + (-2t_2 + 1)e_2 - \frac{1}{2}t_2e_4,$$
$$t_3e_1 - 2t_3e_2 + e_3 - \frac{1}{2}t_3e_4, t_4e_1 - 2t_4e_2 - \frac{1}{2}t_4e_4 + e_5], \{t_1, t_2, t_3, t_4\}.$$

The result indicates that there is a 4 parameter family of complementary subspaces (parametrized by $t_1, \ldots, t_4$) and gives a basis for the subspaces. The `Query` command can now determine if any of these complements defines a reductive decomposition and, if so, display a basis for the reductive subspace $\mathfrak{m}$:

```
> Query(H, M, ``ReductivePair'')[-1][1][2];
```

$$[(t_2+1)e_1 + (-2t_2+1)e_2 + (-\frac{1}{2}t_2+\frac{1}{4})e_4, \quad t_2e_1 + (-2t_2+1)e_2 - \frac{1}{2}t_2e_4, \quad e_3,$$

$$t_4e_1 - 2t_4e_2 - \frac{1}{2}t_4e_4 + e_5].$$

Thus there is in fact a two parameter family of reductive decompositions of $\mathfrak{g}$ relative to $\mathfrak{h}$.

To summarize, the Gödel spacetime is a perfect fluid solution of the Einstein equations which is a locally rotationally symmetric, reductive homogeneous space.

C. Case splitting: a vacuum metric with no isometries

The ability to identify exceptional values of parameters or functions when solving a system of equations—case splitting—is an extremely powerful recent development in symbolic solution routines. Here we present an example of case splitting in the analysis of Killing vector fields. The metric we shall consider comes from the class of spacetimes admitting a non-diverging, shear-free, and null geodesic congruence.[9,11,15] The metric g is expressed in coordinates $(u, v, \zeta, \overline{\zeta})$ as

$$g = d\zeta \odot d\overline{\zeta} - du \odot (dv + W d\zeta + \overline{W} d\overline{\zeta} + H du), \tag{2.5}$$

where $\odot$ is the symmetric tensor product and

$$W = -\frac{2v}{\zeta + \overline{\zeta}}, \quad H = (\zeta + \overline{\zeta}) f(u)(e^{\zeta} + e^{\overline{\zeta}}) - \frac{v^2}{(\zeta + \overline{\zeta})^2}. \tag{2.6}$$

A bar on a quantity denotes complex conjugation.

It is readily verified using *DifferentialGeometry* functionality that the vector field ∂_v determines a non-diverging, shearfree, and null geodesic congruence, and furthermore, that this is a vacuum metric of Petrov type N in any open region where $f(u) \neq 0$. The relevant commands are `NPSpinCoefficients` and `PetrovType`. The spacetime is flat wherever $f(u) = 0$; for simplicity, we will suppose $f(u) \neq 0$ in what follows.

The spacetime can be defined within *DifferentialGeometry* via

```
> DGsetup([u, v, zeta, zetab], M):
> xi := evalDG(dv + W*dzeta + W*dzetab + H*du):
> W := - 2*v/(zeta + zetab):
> H := (zeta + zetab)* f(u)* (exp(zeta) + exp(zetab)) - v^2/(zeta
+ zetab)^2:
> g := evalDG(dzeta &s dzetab - du &s xi):
```

Here the symbol `&s` denotes the symmetric tensor product.

We first look for solutions to the Killing equations making no specific choices for the function $f(u)$, in which case the PDE solvers treat $f(u)$ as generic. The result is that there are no non-trivial solutions:

```
> KillingVectors(g);
```

$$[\,]$$

Thus Eq. (2.5) provides an apparently rare instance of an *explicit family of vacuum solutions to the Einstein equations which admit no continuous symmetries*.

For generic choices of $f(u)$ there are no continuous isometries. However, it is possible that specific choices of $f(u)$ allow for the existence of Killing vector fields. This possibility is analyzed by adding keyword arguments "parameters" and "auxiliaryequations" to the `KillingVector` command. These keywords respectively identify the parameters and/or functions which are to be investigated for exceptional values, and impose any desired restrictions on the parameter/functions. The syntax of the command is

```
> KV := KillingVectors(g, parameters=[f(u)], auxiliaryequations
= {f(u) <> 0});
```

The "auxiliaryequations" keyword is used to eliminate the trivial case $f(u) = 0$ from consideration. The output of this command consists of a sequence of lists of Killing vector fields followed by a list of the corresponding conditions on $f(u)$ which allow them:

$$KV := [\,], [(-2C_1 + \frac{1}{2}u^2 + C_2 u + \frac{1}{2}C_2^2)D_u - (\frac{1}{2}\zeta^2 + \zeta\overline{\zeta} + vC_2 + \frac{1}{2}\overline{\zeta}^2 + vu)D_v],$$
$$[f(u) = f(u), f(u) = C_3/(4C_1 - u^2 - 2C_2 u - C_2^2)^2]. \tag{2.7}$$

We thus obtain two classes of results. The first entry is an empty list which corresponds to the condition $f(u) = f(u)$; this recovers the result that for generic choices of f there are no solutions to the Killing equations for this vacuum metric. The second entry is a three parameter family of Killing vector fields corresponding to a three parameter family of choices for $f(u)$,

$$f(u) = \frac{c}{(a - bu - u^2)^2}. \tag{2.8}$$

Thus for each generic choice of a, b, c in Eq. (2.8), there is a single Killing vector field

$$(a - bu - u^2)\partial_u + \left[(\zeta + \overline{\zeta})^2 + v(b + 2u)\right]\partial_v. \tag{2.9}$$

Subsequent case splitting analysis of the Killing vectors for this 3 parameter family of metrics reveals that this is the maximal one-dimensional isometry group for all values of the parameters except $c = 0$, which corresponds to flat spacetime.

III. KILLING TENSORS

Besides Killing vector fields, there are a number of fields whose existence can be used to invariantly characterize spacetimes, e.g., homothetic vector fields, conformal killing vector fields, Killing-Yano tensors, recurrent tensors, etc. Such tensor fields are discussed, e.g., in Ref. 11. All such tensor fields share the feature that they are determined as solutions of systems of over-determined PDEs. As such, they are amenable to symbolic analysis; all the "special tensor fields" of Ref. 11 can be analyzed in *DifferentialGeometry*. Here we focus on the computation of Killing tensors.

Killing tensors are generalizations of Killing vectors. A Killing tensor on a spacetime (M, g) is a symmetric tensor, $K_{a_1 \ldots a_p} = K_{(a_1 \ldots a_p)}$, satisfying

$$\nabla_{(a} K_{b_1 \ldots b_p)} = 0, \tag{3.1}$$

where ∇ is the connection on M compatible with g. Note that a Killing tensor of rank-1 is just the 1-form corresponding via g to a Killing vector field. The metric g is always a Killing tensor of rank 2. The symmetrized product of a Killing tensor of rank p and one of rank q is a Killing tensor of rank $p + q$.

An important question that arises when computing Killing tensors is whether a given Killing tensor is "reducible" in the sense of being constructed from (linear combinations over **R** of) the symmetric products of lower rank Killing tensors. It is possible to construct various symbolic algorithms which answer this question. Within *DifferentialGeometry*, a good strategy is to exploit the command `GetComponents`, which gives the components of any element of a vector space in terms of a given set of elements of that space.

To illustrate all of this, we consider the Killing tensors of rank 2 and 3 for the Gödel spacetime. With the spacetime as defined in Eq. (2.2), we compute the rank-2 Killing tensors via

```
> KT := KillingTensors(g,2):
```

We have suppressed the lengthy output, which is a list of rank-2 tensor fields forming a basis for the vector space K (over **R**) of solutions to Eq. (3.1). The dimension of K is obtained by asking how many elements are in this list:

```
> nops(KT);
```

```
15
```

To compute the "trivial" Killing tensors, we compute a basis *KV* of rank-1 Killing tensors (corresponding to the 5 Killing vector fields of the Gödel metric) and a basis *B* for the algebra *S* of rank-2 symmetric tensors generated by them. Again we suppress the lengthy output.

```
> KV := KillingTensors(g,1):
> B := SymmetricProductsOfKillingTensors(KV, 2):
```

We then check to see which elements of *KT* are in the algebra *S*, by checking if they are in the span of *B* with real, constant coefficients. Rather than displaying the components of *KT* in the basis *B*, we simply ask if the Killing tensors are in the span of *B*:

```
> GetComponents(KT, B, method=``real'', trueorfalse=``on'');
```

$$true. \tag{3.2}$$

The result "true" indicates that *all* the rank-2 Killing tensors are spanned by the rank-1 Killing tensors (cf. Ref. 5). This, of course, could have been deduced from the fact that the dimension of *KT* is 15, which is also the dimension of the symmetric algebra of rank-2 tensors generated by the 5 Killing vector fields. A similar (but more CPU intensive) analysis establishes that there are no non-trivial rank-3 Killing tensors for the Gödel spacetime. First, we compute a basis of solutions to the relevant differential equations; the solution space is 35 dimensional:

```
> KV3 := KillingTensors(g, 3):
> nops(KV3);
```

```
35
```

Since all rank-2 Killing tensors are reducible, we simply need to compute the dimension of the vector space formed by symmetric tensor products of Killing vectors (rank-1 Killing tensors) to see if there are any irreducible rank-3 Killing tensors. This vector space has dimension,

$$\binom{5+3-1}{3} = 35.$$

Thus there are no irreducible rank-3 Killing tensors for the Gödel spacetime.

IV. ALGEBRAIC PROPERTIES OF CURVATURE

The algebraic classification of curvature—"Petrov type" for classification of the Weyl tensor and "Segre type" for classification of the Ricci tensor—is an extremely important tool for invariantly characterizing spacetimes. The analysis needed to find the Petrov or Segre type is relatively straightforward, mathematically speaking, but the calculations can become quite involved, so here is a place where computer algebra systems can be quite useful. The algorithms used by *DifferentialGeometry* for algebraic classification of curvature are a refinement of those devised in Refs. 1 and 13. The input to the algorithm is a null tetrad which the user may supply or which is constructed automatically by *DifferentialGeometry*. The principal difficulties which arise in determining the Petrov/Segre type symbolically include: (i) excessive computation times owing to a poor choice of null tetrad; (ii) inability to determine whether various invariants are equal or vanish ("zero recognition problem" for scalars); and (iii) the possibility of complicated branching of results depending upon exceptional values of parameters or functions.

Finding a computationally effective null tetrad can be a bit of an art if the goal is to determine the Petrov type, but the range of possibilities can be explored by applying various Lorentz transformations via the `NullTetradTransformation` commands. However, if the Petrov type is known `AdaptedNullTetrad` and/or `PrincipalSpinors` can be used to determine a family of adapted null tetrads and/or corresponding principal spinors.

The zero recognition problem is a fundamental issue in symbolic analysis; in *DifferentialGeometry*, this is ameliorated by allowing the user to control and inspect all aspects of the classification process. In particular, the `infolevel` command allows the user to inspect each step of the classification algorithm, the `Preferences` command allows the user to create custom-made simplification rules, and the `output` option allows the user to examine the various algebraic and differential equations which arise in determining the algebraic classification.

In many ways, the most intricate and interesting aspect of algebraic classification of spacetimes is to handle the possible branching of the classification, depending upon particular values of functions or parameters. In *DifferentialGeometry*, this is handled using customizable case-splitting analysis options. This functionality we use below to uncover and classify some new solutions of the Einstein equations which are hidden in a known ansatz.

We consider a family of hypersurface homogeneous spacetimes determined by a parameter *a* and a function *f* (see Ref. 11, Sec. 13.3.3, Eq. (13.67)):

$$ds^2 = -2\left(x^a du - \frac{dy}{(a+1)x}\right)\left[dt + a(tdx + dy)/x + f(t)\left(x^a du - \frac{dy}{(a+1)x}\right)\right] + (dx^2 + dy^2)\frac{(t^2+1)}{2x^2}. \tag{4.1}$$

This metric can be defined as follows:

```
> DGsetup([t, u, x, y], M):
> alpha := evalDG(x^a*du - 1/(a+1)/x*dy);
```

$$\alpha := x^a du - \frac{1}{(a+1)x}dy \tag{4.2}$$

```
> beta := evalDG(dt + a/x*(t*dx + dy) + f(t)*alpha);
```

$$\beta := dt + f(t)x^a du + \frac{at}{x}dx + (\frac{a}{x} - \frac{f(t)}{(a+1)x})dy \tag{4.3}$$

```
> g := evalDG(-alpha &s beta + (t^2+1)/2/x^2*(dx &t dx + dy &t
dy)):
```

Using a combination of the commands `DGGramSchmidt` to construct an orthonormal tetrad, `NullTetrad` to construct the corresponding null tetrad, and `NullTetradTransformation` to simplify the result, the we obtain the following null tetrad:

$$\begin{aligned} NT := [&D_t, -f(t)D_t + x^{-a}D_u,\\ &-\frac{a(I+t)}{\sqrt{t^2+1}}D_t + \frac{Ix^{-a}}{\sqrt{t^2+1}(a+1)}D_u + \frac{x}{\sqrt{t^2+1}}D_x + \frac{Ix}{\sqrt{t^2+1}}D_y,\\ &\frac{a(I-t)}{\sqrt{t^2+1}}D_t - \frac{Ix^{-a}}{\sqrt{t^2+1}(a+1)}D_u + \frac{x}{\sqrt{t^2+1}}D_x - \frac{Ix}{\sqrt{t^2+1}}D_y]. \end{aligned}$$

Simply asking for the Petrov type of the spacetime (via the frame *NT*) will classify this metric assuming the parameter *a* and function *f* and the spacetime point are all generic. Generically, the metric is of type II.

```
> PetrovType(NT);
```

“II”.

Generic choices of *a* and *f(t)*, however, do not lead to solutions of the Einstein equations. We now investigate the Petrov types which can occur given that *a* and *f*(*t*) are adjusted so that *g* is an Einstein metric. It is most efficient to perform all tensor computations in the anholonomic frame provided by *NT*. To this end, we initialize the anholonomic frame defined by *NT* and define the metric η in that frame.

```
> FD := FrameData(NT, N): DGsetup(FD):
> eta := evalDG(-Theta1 &s Theta2 + Theta3 &s Theta4);
```

$$\eta := -\Theta 1 \Theta 2 - \Theta 2 \Theta 1 + \Theta 3 \Theta 4 + \Theta 4 \Theta 3.$$

We next compute the vacuum Einstein equations with cosmological constant. We first define these equations as a tensor field *EFE*, then we extract the components of *EFE* to create a system of differential equations denoted by *EQ*. The output is suppressed because of its complexity.

```
> EFE := evalDG(EinsteinTensor(eta) + Lambda * InverseMetric (eta)):
> EQ := DGinfo(EFE, ``CoefficientSet''):
```

The following application of the `PetrovType` command performs the following steps:

1. From the given null tetrad, the equations determining the various PetrovTypes are computed and solved for a, Λ and $f(t)$.
2. For each of these solutions, it is determined when a solution to `EQ` also exists.

In this way, we find all choices of a, Λ and $f(t)$ which yield various possible Petrov types under the condition that the metric is an Einstein metric. The result is presented as a table (PT) whose individual entries are displayed below.[16]

```
> Tetrad := [E1, E2, E3, E4]:
> PT := PetrovType(Tetrad, auxiliaryequations = EQ, parameters =
[f(t), a, Lambda], method = ``real''):
> PT[``I''];
```

$$[\,]$$

```
> PT[``II''];
```

$$\text{“}generic''$$

```
> PT[``III''];
```

$$[a = \frac{1}{2}, \Lambda = -\frac{39}{16}, f(t) = \frac{13}{32}t^2 + \frac{17}{32}]$$

```
> PT[``D''];
```

$$[a = 0, \Lambda = \Lambda, f(t) = (-\frac{1}{6}t^3\Lambda - \Lambda t - t + \frac{1}{2}\frac{\Lambda + 2}{t} + C1)\frac{t}{t^2+1}]$$

```
> PT[``N''];
```

$$[a = 2, \Lambda = -3, f(t) = \frac{1}{2}t^2 - \frac{1}{2}]$$

```
> PT[``O''];
```

$$[[a = 0, \Lambda = -\frac{3}{2}, f(t) = \frac{1}{4} + \frac{1}{4}t^2], [a = -\frac{1}{3}, \Lambda = -\frac{2}{3}, f(t) = \frac{1}{9}t^2 - \frac{1}{9}]].$$

The first entry of the table shows that the spacetime defined by Eq. (4.1) cannot be of Petrov type I. The second table entry shows (again) that the metric is generically of type II, but is not an Einstein metric in this case. The remaining entries show that there is a unique Einstein metric of Petrov type III and of type N. There are two Einstein metrics of type O; these are two anti-De Sitter spaces. Finally, there is a two parameter family of Einstein metrics of Petrov type D. That Eq. (4.1) contains Einstein metrics of types III and N is well-documented,[11] although our result for the type III case corrects an error in Ref. 11 for the value of Λ. The observation that the family of metrics (4.1) contains Einstein spaces of type D is apparently new.

V. SYMMETRY REDUCTION OF FIELD EQUATIONS

Probably the most effective general method for solving nonlinear field equations is to look for solutions admitting a prescribed group G of symmetries. Given a system of G-invariant differential equations for fields on a manifold M, one can convert the problem of finding G-invariant solutions to the problem of solving a generally simpler set of reduced equations for fields on M/G. The *DifferentialGeometry* package provides an environment in which structures from differential geometry, Lie algebras and groups, and fiber bundles are tightly integrated. As such, it is well-suited for geometric analysis of field equations, and in particular, the use of symmetry reduction to analyze group-invariant solutions of field equations.[2] Here we focus on two important tasks which arise when analyzing a symmetry reduction of a field theory and which are here implemented symbolically for the first time: (i) characterizing the most general group invariant field, and (ii) calculation of the induced symmetry group of the reduced field equations. We illustrate this with the symbolic analysis of the problem of finding gravitational plane waves, which can be defined as Ricci-flat spacetimes admitting a particular class of 5 dimensional isometry groups (see Ref. 12, and references therein).

The manifold is denoted by M with coordinates (u, v, x, y). The desired isometry group action is the 5-dimensional group G of diffeomorphisms of M generated by the vector fields,

$$X_1 = \frac{\partial}{\partial v},\ X_2 = \frac{\partial}{\partial x},\ X_3 = \frac{\partial}{\partial y},\ X_4 = x\frac{\partial}{\partial v} + P(u)\frac{\partial}{\partial x},\ X_5 = y\frac{\partial}{\partial v} + Q(u)\frac{\partial}{\partial y}, \tag{5.1}$$

where $P(u)$ and $Q(u)$ are any functions such that $P'(u) > 0$, $Q'(u) > 0$. Every gravitational plane wave admits G as an isometry group for some choice of the functions P and Q; these functions determine the amplitude and polarization of the wave.

We define the spacetime manifold and the infinitesimal generators of G as follows:

```
> DGsetup([u,v,x,y], M):
> X1 := D_v: X2 := D_x: X3 := D_y: X4 := evalDG(x*D_v + P(u)*D_x):
  X5 := evalDG(y*D_v + Q(u)*D_y):
> KV := [X1, X2, X3, X4, X5]:
```

Our first task is to characterize the G-invariant spacetime metrics, that is, to find the general form of rank-2 symmetric tensor fields which are invariant under the group action generated by the vector fields KV. This can be done using the command `InvariantGeometricObjectFields`, which solves the linear system of PDEs defining any group invariant field. For example, here we find the invariant functions:[17]

```
> InvariantGeometricObjectFields(KV, [1]);
```

$$F1(u).$$

Thus all invariant functions are functions of the invariant u. In particular, the group orbits are the hypersurfaces $u = const.$ and the reduced spacetime manifold is $M/G \approx \mathbf{R}$ with coordinate u.

Similarly, we can compute the most general symmetric tensor of type $\binom{0}{2}$ which is invariant under G. This determines the "reduced fields" from which group invariant solutions of the field equations are obtained. First, we compute a basis $S2$ for symmetric tensors of type $\binom{0}{2}$, then we compute the G invariant fields:

```
> S2:=GenerateSymmetricTensors([du, dv, dx, dy], 2):
> S2invariant := InvariantGeometricObjectFields(KV, S2);
```

$$\begin{aligned} S2invariant := {}& F1(u)\Big(-\frac{d}{du}P(u)\frac{d}{du}Q(u)(du\,dv + dv\,du) + \frac{d}{du}Q(u)dx\,dx + \frac{d}{du}P(u)dy\,dy\Big) \\ &+F2(u)du\,du. \end{aligned} \tag{5.2}$$

Thus the set of G-invariant symmetric tensors of type $\binom{0}{2}$ is spanned by two quadratic forms:

$$\mathcal{Q}_1 = du \otimes du, \quad \mathcal{Q}_2 = -Q'(u)P'(u)du \odot dv + Q'(u)dx \otimes dx + P'(u)dy \otimes dy. \tag{5.3}$$

The most general G-invariant metric takes the form

$$g = a(u)Q_1 + b(u)Q_2, \tag{5.4}$$

where $b(u) > 0$. The symmetry reduced fields on $M/G \approx \mathbf{R}$ are $a(u)$ and $b(u)$.

One obtains symmetry reduced equations for $a(u)$ and $b(u)$ by substituting Eq. (5.4) into the relevant field equations, e.g., the Einstein equations. Generally covariant field equations such as the Einstein equations have the group *Diff*(M) of diffeomorphisms of M (acting by pull-back on the metric) as a symmetry group. The full group of diffeomorphisms does not act on the space of metrics (5.4), so *Diff*(M) cannot be a symmetry group of the symmetry reduced equations. The subgroup of *Diff*(M) which *does* act on the G-invariant metrics and which will also be a symmetry group of the reduced equations is the normalizer of G within *Diff*(M).[2] Thus the normalizer is the *residual symmetry group* after symmetry reduction. The normalizer of G in *Diff*(M) is the smallest subgroup of *Diff*(M) containing G as a normal subgroup. Working infinitesimally, the Lie algebra of the normalizer can be computed as the smallest algebra of vector fields containing the isometry algebra generated by *KV* as an ideal. These vector fields are obtained by solving a large system of linear PDEs which are amenable to symbolic analysis.

The *DifferentialGeometry* command `InfinitesimalPseudoGroupNormalizer` solves these PDEs and thus computes the Lie algebra of the normalizer within the Lie algebra of all vector fields.[18] In the present example, we get

```
> Nor := InfinitesimalPseudoGroupNormalizer(KV);
```

$$Nor := (2\_C1\, v + \_F1(u))D_v + \_C1\, x\, D_x + \_C1\, y\, D_y.$$

Thus the residual symmetry group consists of a scaling of (v, x, y) and a u dependent translation of v. These transformations can be computed explicitly using the *DifferentialGeometry* `Flow` command as follows. First, for simplicity, we break the vector field *Nor* into its translational and scaling parts, respectively:

```
> NorTrans := eval(Nor, _C1=0, _F1(u) = f(u));
```

$$NorTrans := f(u)D_v$$

```
> NorScale := eval(Nor, _C1=1, _F1(u) = 0);
```

$$NorScale := 2v\, D_v + x\, D_x + y\, D_y.$$

Then we compute each of the flows using parameters α and β, respectively. The result is two sets of transformations, specified as relations between the original and transformed coordinate values:

```
> Trans := Flow(NorTrans, alpha);
```

$$Trans := [u = u, v = v + \alpha f(u), x = x, y = y]$$

```
> Scale := Flow(NorScale, beta);
```

$$Trans := [u = u, v = e^{2\beta} v, x = e^{\beta} x, y = e^{\beta} y].$$

Any generally covariant field theory (not just general relativity, or a theory of a metric) will admit these transformations as symmetries upon symmetry reduction by the symmetry group defined by Eq. (5.1).

We can verify that the residual symmetry group does indeed act on the G-invariant metric (5.4) via the `Pullback` command. Defining `Q1` and `Q2` as in Eq. (5.3), we have

```
> Pullback(Trans, Q1);
```

$$du\, du$$

```
> Pullback(Trans, Q2)
```

$$-2\alpha \frac{dP(u)}{du}\frac{dQ(u)}{du}\frac{df(u)}{du} du\, du - \frac{dP(u)}{du}\frac{dQ(u)}{du}(du\, dv + dv\, du) + \frac{dQ(u)}{du} dx\, dx + \frac{dP(u)}{du} dy\, dy$$

```
> Pullback(Scale, Q1);
```

$$du\, du$$

```
> Pullback(Scale, Q2);
```

$$e^{2\beta}(-\frac{dP(u)}{du}\frac{dQ(u)}{du}(du\, dv + dv\, du) + \frac{dQ(u)}{du}dx\, dx + \frac{dP(u)}{du}dy\, dy).$$

Thus the action of the residual group on the functions $(a(u), b(u))$ defining the G-invariant metrics can be expressed as

$$a(u) \longrightarrow a(u) - 2\alpha P'(u)Q'(u)f'(u)b(u), \qquad b(u) \longrightarrow e^{2\beta}b(u). \tag{5.5}$$

From this action of the residual symmetry group, we see that the function $a(u)$ can be "gauged" to zero via `Trans`.

The reduced field equations, obtained by restricting the Einstein equations to metrics of the form Eq. (5.4), are equivalent to a single ordinary differential equation which can easily be computed using *DifferentialGeometry*. It can also be verified using *DifferentialGeometry* that the Einstein–Hilbert Lagrangian density vanishes identically for plane wave metrics (5.4), so that the reduced equations do not follow from the Einstein–Hilbert variational principle, in accord with the general theory of the principle of symmetric criticality.[3,6]

ACKNOWLEDGMENTS

It is a pleasure to thank Edgardo Cheb-Terrab; first, for proposing the development of general relativity tools within the framework of *DifferentialGeometry*, and second, for his time and efforts to integrate this work into the *Maple* distributed library of packages. Supports from the National Science Foundation (DMS-0713830) and Maplesoft are gratefully acknowledged. We thank the referee for skillfully helping us to improve this paper.

[1] Acvevedo, M., Enciso-Aguilar, M., and Lopez-Bonilla, J., Electron. J. Theor. Phys **9**, 79 (2006).

[2] Anderson, I., Fels, M., and Torre, C. G., Commun. Math. Phys. **212**, 653 (2000).

[3] Anderson, I., Fels, M., and Torre, C. G., CRM Proc. Lect. Notes **29**, 95 (2001) ISBN-10: 0-8218-2803-7.

[4] Anderson, I. M. and Torre, C. G., arXiv:1103.1608 [math-ph].

[5] Cook, S., *Killing spinors and affine symmetry tensors in Gödel's universe*, Ph.D. dissertation, Oregon State University, 2009.

[6] Fels, M. and Torre, C. G., Class. Quantum Grav. **19**, 641 (2002).

[7] Heck, A., *Introduction to Maple* (Springer, New York, 2003).

[8] Kobayashi, S. and Nomizu, K., *Foundations of Differential Geometry* (Wiley, New York, 1969), Vol. II.

[9] Kundt, W., Z. Phys. **163**, 77 (1961).

[10] Patera, J., Winternitz, P., and Zassenhaus, H., J. Math. Phys. **16**, 1597 (1975).

[11] Stephani, H., Kramer, D., MacCallum, M., Hoenselaers, C., and Herlt, E., *Exact Solutions of Einstein's Field Equations* (Cambridge University Press, Cambridge, England, 2003).

[12] Torre, C. G., Gen. Relativ. Gravit. **38**, 653 (2006).

[13] Zakhary, E. and Carminati, J., Gen. Relativ. Gravit. **36**, 1015 (2004).

[14] Note, in particular, that a command terminated with a colon will suppress displaying the results of the command, while a command terminated with a semi-colon will display the results of the command.

[15] We thank Dillon Morse for this example.

[16] The option `method = ''real''` limits the solving routines to real solutions.

[17] The second argument of this command is a list which defines a basis (with respect to linear combinations with coefficients given by functions on M) for the invariant tensor fields of the desired type.

[18] The default for this computation is to compute the normalizer within the set Γ of all vector fields. Other choices for Γ can be made.