

# Supercomputers against strong coupling in gravity with curvature and torsion

W.E.V. Barker<sup>1,2,\*</sup>

<sup>1</sup>*Astrophysics Group, Cavendish Laboratory, JJ Thomson Avenue, Cambridge CB3 0HE, UK*

<sup>2</sup>*Kavli Institute for Cosmology, Madingley Road, Cambridge CB3 0HA, UK*

Many theories of gravity are spoiled by strongly coupled modes: the high computational cost of Hamiltonian analysis can obstruct the identification of these modes. We present a computer algebra implementation of the Hamiltonian constraint algorithm for curvature and torsion theories, tailored for surveying the Lagrangian space with high-performance computing resources (clusters and supercomputers). We argue that Poisson brackets can be evaluated in parallel, meaning that Hamiltonian analysis can henceforth be done on silicon, and at scale. We survey the space of ‘outlawed’ Poincaré gauge theories in which a massive parity-even/odd vector or parity-odd tensor accompanies the graviton. Our initial survey spans all configurations of teleparallel-style multiplier fields which might be used to kill-off the strongly coupled modes. All brackets between the known primary and secondary constraints of all theories are made available. This paper also presents the Hamiltonian Gauge Gravity Surveyor (*HiGGS*) package for the *xAct* tensor manipulation suite in *Mathematica*. We demonstrate how *HiGGS* can be used — on a desktop computer — to run the Dirac–Bergmann algorithm on (e.g.) Einstein–Cartan theory, and the ‘outlawed’ extensions above.

PACS numbers: 04.50.Kd, 04.60.-m, 04.20.Fy, 02.70.c, 07.05.Bx

## I. INTRODUCTION

The modern frontier in the search for alternatives to general relativity (GR) is characterised by a very large number of competing models [2]. These models are problematic by dint of their heterogeneity: differences in the mathematical formulation hinder comparison between different classes (e.g. mimetic [3] and MOND [4, 5] theories). This seems intractable, but we expect the number of classes to grow only in proportion to the community. A more serious problem presents when a class contains a large number of parameters, e.g. the couplings in a Lagrangian. Critical theoretical properties, such as the number and health of propagating degrees of freedom (d.o.f) may be sensitive to these parameters in ways which are hard not only to characterise in general, but even to calculate in detail for a given parameter choice.

This scenario commonly arises in the Hamiltonian analysis, used routinely to determine the d.o.fs of a proposed theory [6–10]. If the coupling parameters eliminate some velocity  $\dot{\psi}$  from the motivated Lagrangian  $\mathcal{L}$ , then the total Hamiltonian  $H_T$  must be modified to express the constraint  $\pi_{\dot{\psi}} \sim \partial_{\dot{\psi}} \mathcal{L} \approx 0$ . That constraint is only preserved if  $\dot{\pi}_{\dot{\psi}} \sim \{\pi_{\dot{\psi}}, H_T\} \approx 0$ , which either vanishes identically or constitutes a *new* constraint, ad infinitum. Each constraint subtracts d.o.fs from the countable fields  $\psi$  naïvely present in  $\mathcal{L}$ . The full chain of constraints is systematically elucidated by means of the *Dirac–Bergmann* algorithm [6, 9, 11], the fundamental computational unit of which is the Poisson bracket. In theories constructed from the higher-spin representations of the Lorentz group, including most tensor theories of gravity, a single covariant bracket can be surprisingly cumbersome for humans [12, 13]; less so, as we will find, for computers (see Fig. 1). Through the algorithm, the d.o.f and symmetry structure depend not only on eliminated velocities, but also on all brackets between all con-

straints, which themselves may be contingent on the couplings in ways that are, *ab initio*, unknowable.

The *strong coupling problem* is commonly diagnosed in the Hamiltonian picture. We usually prefer that two gravitational d.o.f propagate [14, 15]. Additional d.o.f may be tolerated unless ghostly or tachyonic<sup>1</sup>, but they must kept under close theoretical and phenomenological control (e.g. by large [18] – or small [19, 20] – masses, screening [21] or other measures). Frequently we start with many fields  $\psi$  (ten in the case of GR), so that reduction to two d.o.f is an achievement in the linearised Hamiltonian picture: strongly coupled modes may *increase* this number the nonlinear analysis [1]. We can imagine strong coupling as a finely-tuned suppression of the kinetic coefficients<sup>2</sup> in a mode’s linear wave equation [22]. The nonlinear operators are generally still present, however, and describe a non-perturbative dynamics which may be unacceptable, having for example an elliptic or parabolic character. The linearisation in this case refers to some motivated exact vacuum solution. This solution need not be Minkowskian: for GR in higher (odd) dimensions, a fine-tuned admixture of the Gauss–Bonnet invariant strongly couples the whole graviton on a maximally symmetric but curved background [23]. In many versions of Hořava gravity [24], the ‘detailed balance’ which defines the Lorentz-asymmetric theory has a similar effect [25, 26].

There are some prominent scenarios where strong coupling seems desirable. The linearised, massive graviton of Fierz and Pauli [27] appears troubled even in the massless limit by its fifth d.o.f — the helicity-0 mode or van Dam–Veltman–Zakharov (vDVZ) scalar [28, 29] — which couples to the matter trace. However, nonlinear completions [30, 31] of Fierz–Pauli theory revealed that rather than persisting as a light d.o.f, the vDVZ scalar instead becomes strongly coupled, i.e. Vainshtein-screened [32, 33]. Unfortunately, these

\* wb263@cam.ac.uk

<sup>1</sup> Perhaps with some interesting exceptions [16, 17].

<sup>2</sup> Where permitted, a canonical normalisation recasts this as a divergent mass.

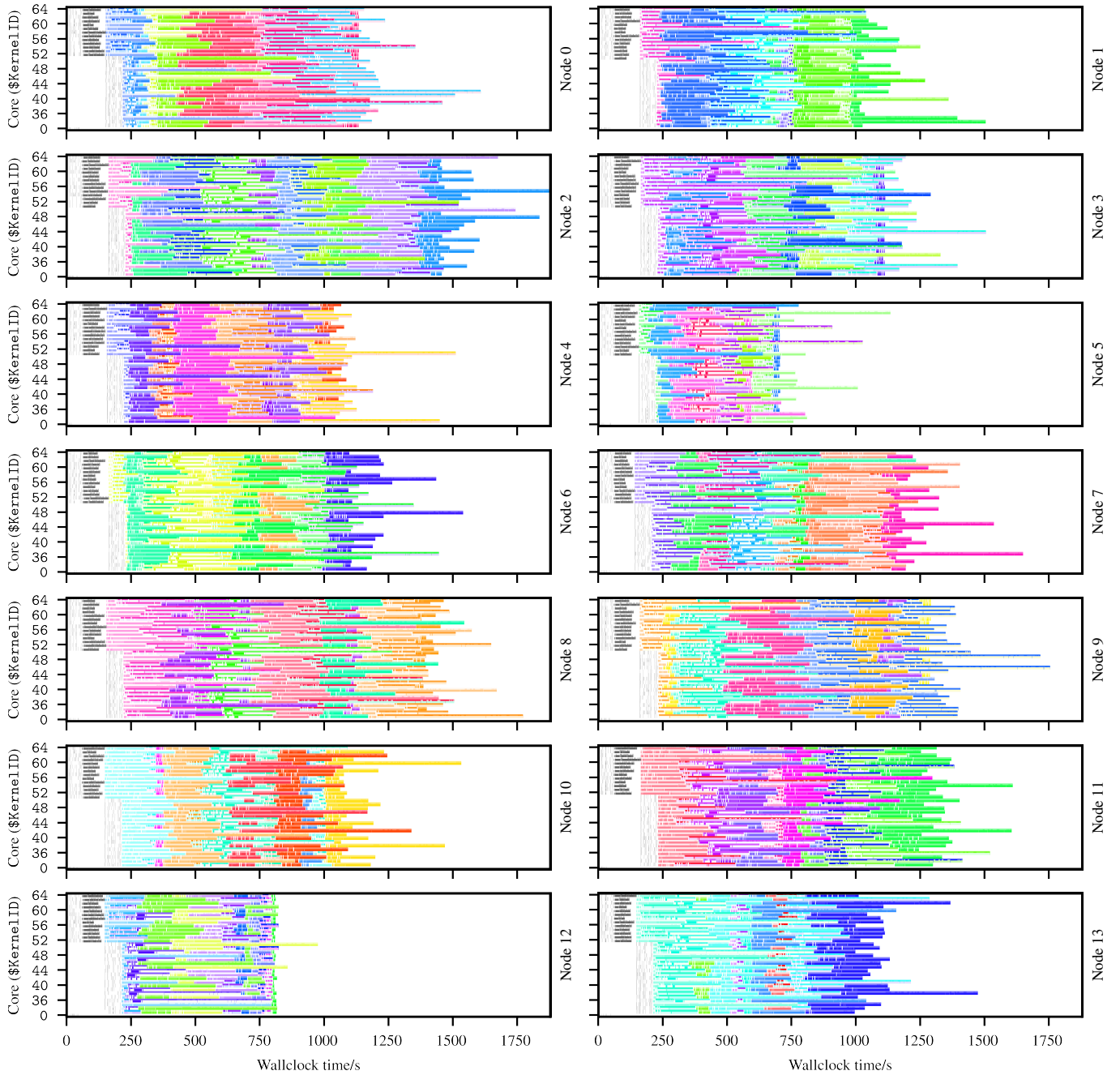


FIG. 1. On a supercomputer, 448 processors obtain simplified, covariant expressions for all nonlinear Poisson brackets, among all discovered primary and secondary constraints, for 192 modified gravity theories. These theories are based on Einstein–Cartan gravity with an extra, massive spin-parity  $1^+$ ,  $1^-$  or  $2^+$  torsion particle, in which various parts ( $2^3 \times 2^3$  choices) of the curvature and torsion are disabled by multipliers. Each colour is a different theory, black/gray is initialisation. We want to know if multipliers can be used to suppress the strongly-coupled  $1^-$ ,  $1^+$  or  $2^-$  modes which respectively ruin these theories [1]. Ready access to the Poisson bracket structure is vital for this analysis.

same theories were also plagued by a sixth d.o.f — the nonlinear Boulware–Deser ghost [30] — whose origin was connected back to the same strong coupling effect [34]. These matters remain contested [35], and massive gravity continues to evolve [36, 37].

Strong coupling in massive gravity has another (contested [35]) association with superluminality [38, 39]. Both

properties are sometimes conflated with acausality, and ill-posedness of the Cauchy–Kovalevskaya problem. While systems might exist in which these are all mutually implicated pathologies, it is by no means general. In each case one should study the characteristic surfaces and sift for coordinate artifacts [40, 41]. Strongly coupled or *singular* surfaces in the Hamiltonian phase space, where the rank of the ma-

trix of constraint brackets changes, have also been accused of causality violation in the *Poincaré gauge theory* (PGT) of gravity [1, 42, 43]. PGT is the general theory which may be constructed from spacetime curvature and torsion. Whether or not the same caveats resolve the causal question here [12], the proclivity for strong coupling in this class remains, and has yet to be seriously addressed. To this end, we target the Poincaré gauge theory class in this paper.

The Poincaré gauge theory encompasses a large sector of the general category of *non-Riemannian* theories, whose current popularity warrants a quick primer. Geometrically interpreted, we retain for non-Riemannian gravity the system of clocks and rulers in the gravitational metric potential  $g_{\mu\nu}$ . We recall however that in GR the connection  $\Gamma^\mu_{\nu\sigma}$  is fixed to the Levi-Civita form  $C^\mu_{\nu\sigma} \equiv C^\mu_{(\nu\sigma)} \equiv \frac{1}{2}g^{\mu\rho}(\partial_\nu g_{\rho\sigma} + \partial_\sigma g_{\rho\nu} - \partial_\rho g_{\nu\sigma})$  – i.e. the non-tensorial Christoffel symbol – by assumption. By relaxing this convention, the geometry of the manifold can be extended to include not only the curvature  $R^\mu_{\nu\sigma\rho} \equiv 2\partial_{[\sigma}\Gamma^\mu_{\rho]\nu} + \dots$ , but also the *torsion*  $T^\mu_{\nu\sigma} \equiv 2\Gamma^\mu_{[\nu\sigma]}$  and *non-metricity*  $Q_{\mu\nu\sigma} \equiv \nabla_\mu g_{\nu\sigma}$ . In this broader context, what is usually meant by ‘GR’ is reached by the Einstein–Hilbert-like action  $S_G \equiv \int d^4x \sqrt{-g} L_G$ , with

$$L_G = -\frac{1}{2\kappa} R + \frac{1}{\kappa} \lambda_\mu^{\rho\sigma} T^\mu_{\rho\sigma} + \frac{1}{\kappa} \hat{\lambda}_\mu^{\rho\sigma} Q^\mu_{\rho\sigma}. \quad (1)$$

The multipliers<sup>3</sup>  $\lambda_\mu^{\rho\sigma}$  and  $\hat{\lambda}_\mu^{\rho\sigma}$  constrain the geometry to be *torsion-free* and *metric*, while the dynamics are those of *curvature*. At the time of writing, the attention of the community is drawn to the remarkable non-uniqueness of (1) as a realisation of GR in this broader, non-Riemannian context. By cycling multipliers onto different pairs in the curvature–torsion–non-metricity triad, and identifying suitable dynamical terms, one can construct the flat, metric *teleparallel* equivalent of GR (TEGR) from pure torsion [45], and the flat, torsion-free *symmetric* alternative (STEGR) from pure non-metricity [46]. With (1), these points in the space of non-Riemannian theories define the *geometrical trinity of gravity* [44].

There is now activity to determine a preferred vertex of the trinity, and the viability of the surrounding non-Riemannian landscape. Various considerations must be balanced in a very large parameter space [47–50]. As we mentioned previously, the PGT may be thought of as the sector of the landscape in which  $Q^\mu_{\nu\sigma}$  is suppressed, e.g. by a multiplier as in (1). PGT is a convenient sector to study — without prejudice to the ultimate rôle of non-metricity in constructing a viable gravity theory — since it has a relatively self-contained history stemming from the Einstein–Cartan model [51], and the gauge-theoretic interpretation pioneered by many authors [7, 52–54], beginning with Kibble [55], Utiyama [56] and also Sciama [57].

The linear Hamiltonian structure of PGT is particularly well developed [7, 58–60]. In the nonlinear structure, strong cou-

pling phenomena are known to be *abundant*, and in this regard PGT is feared to be representative of the broader non-Riemannian landscape. Critically however, the challenge of the Hamiltonian analysis has prevented this structure from being mapped in any comprehensive detail. A few islands in the landscape were probed at the turn of the millenium [1]: minimal extensions to the Einstein–Cartan theory in which a single extra massive spin-parity ( $J^P$ )  $1^+$ ,  $1^-$  or  $2^-$  torsion particle is present. In each case respectively,  $1^-$ ,  $1^+$  and  $2^+$  modes were suggested to be strongly coupled. Since then, the PGT sector of the landscape has been treated with a sense of ‘*hic sunt dracones*’ – i.e. avoided as potentially dangerous – with apparently the only safe configurations being exclusive activation of the  $0^+$  or  $0^-$  scalar torsion particles [43]. In light of our comments above, one can even afford to remain agnostic on the pathology of this strong coupling<sup>4</sup>. We suggest, however, given the recent, promising developments in the non-Riemannian approach [44, 47–50], that the extent of the phenomenon should still be understood, and general tools be developed to that end.

In this paper we present a computer algebra implementation of the nonlinear Hamiltonian analysis for a *generalisation* of the full PGT, in which one may covariantly disable arbitrary irreps of the torsion and curvature by means of Lagrangian multiplier fields. This generalisation was put forward as an anti-strong-coupling measure in [13], and its Hamiltonian structure is elucidated in the companion paper [62]. In the conventions of (1), we may write generalised PGT as

$$\begin{aligned} L_G = & -\frac{\hat{\alpha}_0}{2\kappa} R + \sum_{I=1}^6 \left( \hat{\alpha}_I R^{\mu\nu}_{\sigma\rho} + \bar{\alpha}_I \lambda^{\mu\nu}_{\sigma\rho} \right) I \hat{P}_{\mu\nu}^{\sigma\rho}{}^{\xi\zeta} R^{\kappa\pi}_{\xi\zeta} \\ & + \frac{1}{\kappa} \sum_{M=1}^3 \left( \hat{\beta}_M T^\mu_{\nu\sigma} + \bar{\beta}_M \lambda^\mu_{\nu\sigma} \right) M \hat{P}_\mu^{\nu\sigma}{}^{\xi\zeta} T^\pi_{\xi\zeta} \\ & + \frac{1}{\kappa} \hat{\lambda}_\mu^{\rho\sigma} Q^\mu_{\rho\sigma}, \end{aligned} \quad (2)$$

where the usual ‘quadratic’ PGT is spanned by the ten parameters  $\hat{\alpha}_0$ ,  $\{\hat{\alpha}_I\}$ ,  $\{\hat{\beta}_M\}$ . The projections  $I \hat{P}_{\mu\nu}^{\sigma\rho}{}^{\xi\zeta}$  and  $M \hat{P}_\mu^{\nu\sigma}{}^{\xi\zeta}$  extract the  $SO^+(1, 3)$  field strength irreps. The core of the implementation is (version 1.0.0 of [63]) the *Hamiltonian Gauge Gravity Surveyor (HiGGS)*, a *Mathematica* package grounded in the popular open-source *xAct* tensor manipulation suite.

<sup>4</sup> It is important to understand that the main historical objection to the strongly coupled modes of the PGT has been their ghostly character, as inferred by an inspection of the signs of squared momenta in the Hamiltonian. These signs are fixed by the unitarity requirements of the desired, linearly active modes in each case, and in each case they are *negative* [1]. This ‘*catch-22*’ is particular to the PGT, but, as cogently explained in [61], there are principled reasons to be suspicious of strongly coupled surfaces which appeal to neither the non-perturbative dynamics, nor the causality arguments mentioned above. Generically, a background which is strongly coupled had better not be one which is also seen in nature, since it cannot have been reached by any smooth trajectory through the phase space.

<sup>3</sup> Note some differences in our convention, regarding the dimensionality and density status of these [44].

The *HiGGS* package is suitable for targeted use on a desktop computer. Since it is parallelised over Poisson brackets, *HiGGS* also scales to clusters and supercomputers. We have developed in this direction with the aim of surveying the constraint structure of the non-Riemannian landscape at scale. Modules from *HiGGS*, in particular those concerned with bracket evaluation, can be used as a back-end in searching the parameter space for desirable canonical features — as such features become better understood with time.

For the moment, we perform the brute-force ‘calibration’ survey illustrated in Fig. 1. In this run, which takes a little over 1 h, the  $1^+$  and  $2^-$  Einstein–Cartan extensions are modified with all possible configurations of curvature- and torsion-disabling multipliers: 128 generalised Poincaré gauge theories in total. During the run *HiGGS* obtains, for every theory, all the primary and secondary constraints which can be inferred from a knowledge of the literature, and then computes simple covariant expressions for the nonlinear Poisson brackets between all constraint pairs. All brackets identified in this survey can be found in the supplemental materials [64].

We also provide examples of how to use *HiGGS* to calculate constraint velocities when implementing the Dirac–Bergmann algorithm. We focus on the minimal Einstein–Cartan extensions with strong coupling, and the viable  $0^+$  and  $0^-$  extensions. We also study the unmodified Einstein–Cartan theory, and use *HiGGS* to show that it propagates only the two graviton polarisations. Finally, we will use the results of the initial survey to show how multipliers can be used to suppress strongly coupled fields.

The remainder of this paper is structured as follows. In Section I A we briefly set out our conventions for PGT, using the non-geometric gauge-theoretic formulation. In Section II we describe the *HiGGS* implementation, including general tools for the canonical manipulation curvature and torsion, up to Poisson brackets and higher-level functionality for the theory-specific calculation of constraints and velocities. We also describe our solutions for scaling the Hamiltonian analysis to high-performance computing (HPC) resources. In Section III we present examples of the algorithm, and the results of our initial survey in Fig. 1. Conclusions follow in Section IV.

### A. Conventions

Before proceeding, we introduce the gauge-theoretic formulation of Poincaré gauge theory, as used in [7, 12, 13, 62, 65–67]. The gauge picture does not offer any special advantage over the geometric setup in (1), but it is more consistent with previous Hamiltonian analyses in [1, 7, 43, 58–60, 68].

The geometric covariant derivative appearing in the definition  $Q_{\mu\nu\sigma} \equiv \nabla_\mu g_{\nu\sigma}$ , as it acts on a vector  $V^\mu$ , is written

$$\nabla_\nu V^\mu \equiv \partial_\nu V^\mu + \Gamma^\mu_{\lambda\nu} V^\lambda. \quad (3)$$

In the Poincaré gauge theory we enforce  $Q_{\mu\nu\sigma} = 0$  by assumption, though a multiplier could also be used. The general non-

Riemannian connection

$$\Gamma^\mu_{\nu\sigma} = C^\mu_{\nu\sigma} + K^\mu_{\nu\sigma} + L^\mu_{\nu\sigma}, \quad (4)$$

thus loses its *disformation* part  $L^\mu_{\nu\sigma} \equiv \frac{1}{2}Q^\mu_{\nu\sigma} - Q_{(\nu|\sigma)}^\mu = 0$ , but still conveys torsion through the *contorsion* d.o.fs  $K^\mu_{\nu\sigma} \equiv \frac{1}{2}T^\mu_{\nu\sigma} + T_{(\nu|\sigma)}^\mu$ . Aside from the contorsion, the metric d.o.fs which source the Levi–Civita part are defined, as usual, by tangents to the coordinate curves  $g_{\mu\nu} \equiv e_\mu \cdot e_\nu$ .

We now move away from this geometric picture, to a ‘particle physics’ setup where the underlying manifold is always flat Minkowski space  $\mathcal{M}$ . The metric associated with (curvilinear) coordinate tangents is then  $\gamma_{\mu\nu} \equiv e_\mu \cdot e_\nu$ , and this metric is *flat*. The coordinate basis is accompanied by a Lorentz basis, whose dot products give the Minkowskian metric components  $\eta_{ij} \equiv \hat{e}_i \cdot \hat{e}_j$ . The Lorentz basis can rotate locally under the proper, orthochronous Lorentz rotations, and is non-holonomic. Following on from (3), a vector  $\mathcal{V}^i$  with Lorentz indices has a covariant derivative

$$D_j \mathcal{V}^i \equiv h_j^\mu (\partial_\mu \mathcal{V}^i + A^i_{k\mu} \mathcal{V}^k), \quad (5)$$

where we define the (inverse of the) translational gauge field  $h_i^\mu$  and the rotational gauge field  $A^i_{\mu} \equiv A^{[ij]}_{\mu}$  — we will return to this derivative in Section II C. These fields guarantee invariance under the general coordinate transformations (GCTs or passive diffeomorphisms) on  $\mathcal{M}$ , and Lorentz rotations, and together they *gauge* the Poincaré group  $\mathbb{R}^{1,3} \rtimes \text{SO}^+(1, 3)$ .

How to connect back to the geometric picture? The Minkowskian metric components can be recovered via the identities  $b^i_\mu h_i^\nu \equiv \delta_\mu^\nu$  and  $b^i_\mu h_j^\mu \equiv \delta_j^i$ . On the other hand, the usual system of clocks and rulers can be recovered using  $g_{\mu\nu} \equiv \eta_{ij} b^i_\mu b^j_\nu$  and  $g^{\mu\nu} \equiv \eta^{ij} h_i^\mu h_j^\nu$ . The covariant measures on  $\mathcal{M}$  and  $\tilde{\mathcal{M}}$  are respectively  $\sqrt{-g}$ , i.e. the conventional  $g \equiv \det g_{\mu\nu}$ , and  $b \equiv h^{-1} \equiv \det b^i_\mu$ . Finally, the geometric field strength tensors — referred to as the curvature and the torsion — are provided by the formulae

$$\mathcal{R}^{ij}_{kl} \equiv 2h_k^\mu h_l^\nu (\partial_{[\mu} A^i_{\nu]} + A^i_{m[\mu} A^{mj}_{\nu]}), \quad (6a)$$

$$\mathcal{T}^i_{kl} \equiv 2h_k^\mu h_l^\nu (\partial_{[\mu} b^i_{\nu]} + A^i_{m[\mu} b^m_{\nu]}). \quad (6b)$$

The conversion of these back to the (numerical values of) the geometric components, is done by contraction with the translational gauge fields. The theory (2) meanwhile becomes

$$\begin{aligned} L_G = & -\frac{1}{2} \hat{\alpha}_0 m_p^2 \mathcal{R} \\ & + \sum_{I=1}^6 \left( \hat{\alpha}_I \mathcal{R}^{ij}_{kl} + \bar{\alpha}_I \lambda^{ij}_{kl} \right) \hat{P}^{kl}_{ij}{}^{pq}{}^{nm} \mathcal{R}^{nm}_{pq} \\ & + m_p^2 \sum_{M=1}^3 \left( \hat{\beta}_M \mathcal{T}^i_{jk} + \bar{\beta}_M \lambda^i_{jk} \right) \hat{P}^{jk}_{i}{}^{nm} \mathcal{T}^l_{nm}, \end{aligned} \quad (7)$$

where our metricity assumption dispenses with the need for the final term in (2), and we use the Planck mass rather than the



Einstein constant  $m_p \equiv 1/\sqrt{\kappa}$ . Once again, we reiterate that our use of the gauge theory setup over geometric alternatives (including those which use the tetrad and spin-connection over the metric and contorsion) is merely a matter of convenience in the present work.

In this article we will follow [69] by using the following syntax highlighting for code listings: keywords for *Mathematica* are typeset in **green**, for *xAct* in **blue** and for *HiGGS* in **red**. We use the ‘West coast’ signature (+, −, −, −).

## II. IMPLEMENTATION

We first describe the implementation of the Hamiltonian analysis in the *HiGGS* package. Many aspects of our particular approach will be inefficient: the package is monolithic and expensive in terms of memory and maintenance. We note however that, at the time of writing, HPC is a *cheap* resource. By carefully tuning only a few aspects of the implementation we are therefore able to produce a product which surveys the theory space in a matter of hours. We emphasise that these ‘high-level’ features of *HiGGS* are currently limited to the generalised Poincaré gauge theory in (7). The question of scaling to arbitrary theories will be addressed in Section IV, where we will find that such a scaling is actually likely to *reduce* the complexity of the package, so long as the **PoissonBracket** [] function is upgraded so as to be ‘aware of’ Leibniz’s rule. Moreover, the ‘low-level’ functionality, which is not specific to a Lagrangian, will be of standalone utility. In what follows, we assume a basic familiarity with the *xAct* suite and Wolfram Language.

### A. Geometric setup

We begin by describing the way in which the Riemann–Cartan geometry is implemented in *HiGGS*. Whilst *xAct* is perfectly capable of accommodating not only a Riemann–Cartan curvature, but also a torsion tensor, we prefer to adhere to the ‘particle physics’ picture of gravitational gauge theories [7, 13, 54, 70], and set up all the physics on a *flat* spacetime. The following equivalent *xAct* commands are issued when the *HiGGS* environment is initially built (see Section III A) with the command **BuildHiGGS** []

```
In[] := DefManifold[M4, 4, IndexRange[{a, z}]];
DefMetric[-1, G[-m, -n], CD, {"", "", "\[PartialD]"},
  PrintAs -> "\[Gamma]", FlatMetric -> True,
  SymCovDQ -> True];
```

We thus set up a  $D = 4$  manifold **M4** to represent  $\check{\mathcal{M}}$ , with a *flat* negative-signature metric **G**[-m, -n] to represent  $\gamma_{\mu\nu}$ , and *flat* covariant derivative  $\check{\nabla}_\mu$  represented by **CD**[-m] []. This offers several advantages, and foremost among these is an easy comparison with the very substantial body of literature on the Hamiltonian structure as mentioned in Section I A. We will also not be limited by the fact that the non-Riemannian features

of *xAct* are (currently) less well developed and less flexible than those of a simple Minkowskian setup.

An essential feature of this setup, which goes beyond the particle picture of the literature, is the conflation of holonomic and non-holonomic tangent spaces: a single collection of indices **a, b, c**, etc. is ascribed to **TangentM4**, and these represent both the Lorentz indices  $i, j, k$  and coordinate indices  $\mu, \nu, \sigma$ . This practice might be anathema to the field of differential geometry, but it is acceptable for pragmatic, computational purposes. We assume implicitly that the coordinates are Cartesian, so that the components of the flat metric  $\gamma_{\mu\nu}$  are Minkowskian (equal to those of  $\eta_{ij}$ ), and moreover that a rotational gauge is chosen in which  $e_0 = \hat{e}_0$ ,  $e_1 = \hat{e}_1$ ,  $e_2 = \hat{e}_2$  and  $e_3 = \hat{e}_3$ . As a consequence of this gauge-fixed setup, the covariance of quantities is not guaranteed internally, as it would be if all the features of *xAct* (such as user-defined connections) were fully exploited in *HiGGS*. We can instead check the covariance by visually inspecting the final results for explicit gauge fields: in practice this turns out to be very easy, and the elimination of bare gauge fields in *HiGGS* is reliable. It is important to note that whilst the gauge is fixed to conveniently overload the indices of **TangentM4**, the tensor structure of PGT is wholly preserved. This is in contrast, for example, with the seminal paper [58] in which the *time gauge* imposes  $h_a^0 = 0$ , thereby massively simplifying the various algebraic expressions. On the contrary, the algebraic expressions produced from *HiGGS* are valid for *any gauge*, once the various shared indices are interpreted as being either Greek or Roman. Care is taken in the definitions to avoid any ambiguity over the division of indices.

Following on from our discussion of indices, we note that the 1, 2, 3 spacelike indices  $a, b, c$  and  $\alpha, \beta, \gamma$  are also subsumed into **a, b, c**. These may be extracted by means of the projection operator **G3**[a, b], which represents  $\gamma^{\alpha\beta}$ , and lies at the heart of the Arnowitt–Deser–Misner (ADM) split. The ADM or 3 + 1 split uses a spacelike foliation which is characterised by timelike unit vector  $n_k$ , defined as

$$n_k \equiv h_k^0 / \sqrt{g^{00}}, \quad (8)$$

where we recall that the (gravitational) metric components are recovered by  $g^{\mu\nu} \equiv h_i^\mu h_j^\nu \eta^{ij}$ . Any vector with local Lorentz indices may then be decomposed into perpendicular and parallel components  $\mathcal{V}^i = \mathcal{V}^\perp n^i + \mathcal{V}^\parallel$ . An overbar is used to denote the parallel indices. There are then some identities  $b^k_\alpha h_i^\alpha \equiv \delta_i^k$  and  $b^k_\alpha h_i^\beta \equiv \delta_i^\beta$ , which follow from (8). We denote the gauge fields  $\check{h}_i^\mu$ ,  $b_i^\mu$  and  $A^{ij}_\mu$  by **H**[-a, b], **B**[a, -b] and **A**[a, b, -c]. The timelike vector  $n^i$  is represented by **V**[a], and we see that a variety of identities are then implied within the built *HiGGS* environment

```
In[] := quantity = {H[-a, b], B[a, -b], A[a, b,
  -c], V[a], V[a]*V[-a], A[a, -a, -c], A[a,
  b, -c] + A[b, a, -c], H[-a, i]*B[a, -j]
  ], H[-a, i]*B[c, -i], G3[-a, -b]*G3[b, -
  d], G3[-a, a], B[a, -b]*G3[b, -c]*V[-a],
  - CD[-a][G3[-c, b]]}
Out[] =
```

```

{h_a^b, b_a^b, A^ab_c, n^a, 1, A^a_ac,
  A^ab_c + A^ba_c, delta_j^i, delta_a^c, v^l_ad, 3, 0, 0}
In[] := ToCanonical[quantity]
Out[] =
{h_a^b, b_a^b, A^ab_c, n^a, 1, 0, 0, delta_j^i, delta_a^c, v^l_ad, 3, 0, 0}

```

Note that the covariant derivative `CD[-i][ ]` represents, in our Cartesian gauge choice, the basic coordinate derivative  $\check{\nabla}_\mu \rightarrow \partial_\mu$ . We now define a *lapse* function, and a *shift* vector according to

$$N \equiv n_k b^k_0, \quad N^\alpha \equiv h_{\bar{k}}^\alpha b^{\bar{k}}_0. \quad (9)$$

These functions carry information about the part  $b^{\bar{k}}_0$  of the translational gauge field (which, as we will shortly see, is *non-physical*), whereas  $n^i$  is independent of this quantity. In *HiGGS* the lapse is `Lapse[ ]` and we use also the spatial measure `J[ ]` for the quantity  $J \equiv b/N$ , where  $b \equiv \det b^i_\mu$ . Some further identities are then

$$\begin{aligned} \frac{\partial n_i}{\partial b^k_\mu} &\equiv -n_k h_i^\mu, & \frac{\partial h_i^\nu}{\partial b^k_\mu} &\equiv -h_k^\nu h_i^\mu, & \frac{\partial b}{\partial b^k_\nu} &\equiv b h_k^\nu, \\ \frac{\partial J}{\partial b^k_\nu} &\equiv J h_{\bar{k}}^\nu, & \frac{\partial N}{\partial b^k_\nu} &\equiv N n_k h_{\bar{k}}^\nu, \end{aligned} \quad (10)$$

These identities are also incorporated into the *HiGGS* environment, which prefers to extract from all derivatives of quantities dependent on the translational gauge field the form `CD[-a][B[-b,-c]]`. It is practical to thus replace all instances of `CD[-a][H[-b,-c]]`, since the momentum `BPi[-a,-b]` is defined according to `B[-a,-b]`, and we find accordingly

```

In[] := quantity = ScreenDollarIndices[{J[],
  ↳ Lapse[], CD[-a][V[-j]], CD[-a][H[-j, n
  ↳ ]], CD[-a][J[]], CD[-a][Lapse[]]}]
Out[] =
{J, N, -h_j^b n_a1 a_b a1_b + h_b^b1 n_a1 n^b n_j a_b a1_b1,
  -h_a1^n h_j^b a_b a1_b, h_a1^b J a_b a1_b - h_b^b1 J n_a1 n^b a_b a1_b1,
  h_b^b1 N n_a1 n^b a_b a1_b1}

```

Now that the geometric setup is in place, we introduce the canonical setup [6, 7, 71]. The *canonical* momenta are

$$\pi_i^\mu \equiv \frac{\partial b L_G}{\partial(\partial_0 b^i_\mu)}, \quad \pi_{ij}^\mu \equiv \frac{\partial b L_G}{\partial(\partial_0 A^{ij}_\mu)}. \quad (11)$$

As with our earlier work [12, 13, 62], we will *neglect* the matter Lagrangian  $L_M$ . The field strengths in (6a) and (6b) are independent of the velocities for  $b^k_0$  and  $A^{ij}_0$ , so (11) imply 10 primary constraints of the form

$$\varphi_k^0 \equiv \pi_k^0 \approx 0, \quad \varphi_{ij}^0 \equiv \pi_{ij}^0 \approx 0. \quad (12)$$

From (12) we arrive at the result mentioned above, that the conjugate field  $b^k_0$  is non-physical; the same applies to  $A^{ij}_0$ . We denote by ( $\approx$ ) the *weak* equality, not an approximation. We

term the constraints (12) the ‘sure’ primary, first class (sPFC) constraints: they are a consequence of Poincaré symmetry, and they apply for all choices of the  $\{\hat{\alpha}_I\}$ ,  $\{\hat{\beta}_M\}$ ,  $\{\hat{\alpha}_I\}$ ,  $\{\hat{\beta}_M\}$ . According to (11), *HiGGS* must support a field momentum for both `B[a,b]` and `A[a,b,c]`. We define  $\pi_i^\mu$ ,  $\pi_{ij}^\mu$  and the ‘parallel’  $\hat{\pi}_i^{\bar{j}}$ ,  $\hat{\pi}_{ij}^{\bar{k}}$  as follows

```

In[] := quantity = {BPi[-a, b], APi[-a, -b, c],
  ↳ BPiP[-a, -b], APiP[-a, -b, -c], BPiP[-a,
  ↳ -b]*V[b], APiP[-a, -b, -c]*V[c]}
Out[] =
{pi_a^b, pi_a^b^c, pi_a^b, pi_a^b^c, 0, 0}
In[] := ScreenDollarIndices[ToCanonical[quantity
  ↳ /. PiPToPi]]
Out[] =
{pi_a^b, pi_a^b^c, b_b^a1 pi_a^b1, pi_a^b^b1 b_c^a1, v^l_a1b1, 0, 0}

```

The purpose of the parallel momenta is seen above: they are the physical parts not touched by the sPFCs, and defined according to  $\hat{\pi}_i^{\bar{k}} \equiv \pi_i^\alpha b^{\bar{k}}_\alpha$  and  $\hat{\pi}_{ij}^{\bar{k}} \equiv \pi_{ij}^\alpha b^{\bar{k}}_\alpha$ .

We will see in Section IID that the specific internal rule `PiPToPi` should not often need to be used in practice, and has a more general alternative in the `ToBasicForm[ ]` command which is provided officially by the package. In general, parallel and perpendicular quantities can be accessed with some projections

```

In[] := quantity = {PPerp[-a, b], PPara[-a, b]}
Out[] =
{phi_a^b, phi_a^b}
In[] := ScreenDollarIndices[ToCanonical[quantity
  ↳ /. PADMActivate]]
Out[] =
{n_a n^b, delta_a^b - n_a n^b}

```

The flat manifold of course has vanishing Riemannian curvature  $R^\mu_{\nu\sigma\lambda} = 0$ , and so we must be careful not to use the *xAct* quantities `RiemannCD[-a,-b,-c,-d]`, `RicciCD[-c,-d]` or `RicciScalarCD[ ]`. Instead, the field strengths  $\mathcal{R}_{ijkl}$  and  $\mathcal{T}_{ijk}$  are given by their own tensors, and these expand to give the definitions in Eqs. (6a) and (6b)

```

In[] := quantity = {R[-a, -b, -c, -d], R[a, -a,
  ↳ -c, -d], R[-a, -b, -c, c], T[a, -b, -c],
  ↳ T[a, -b, b]}
Out[] =
{R_abcd, R^a_acd, R^a_b^c, T^a_bc, T^a_b^b}
In[] := ScreenDollarIndices[ToCanonical[quantity
  ↳ /. ExpandStrengths]]
Out[] =
{A^c1_b1 A_b c1 a1 h_c^a1 h_d^b1 - A^c1_a1 A_b c1 b1 h_c^a1 h_d^b1 +
  h_c^a1 h_d^b1 a_a1 A_a b b1 - h_c^a1 h_d^b1 a_b1 A_a b a1,
  0, 0, A^a_ca1 h_b^a1 - A^a_ba1 h_c^a1 +
  h_b^a1 h_c^b1 a_a1 b^a_b1 - h_b^a1 h_c^b1 a_b1 b^a_a1, 0}

```

The final set of fields which we introduce are the multipliers  $\lambda_{ijkl}$  and  $\lambda_{ijk}$  – these are precisely the same shape as  $\mathcal{R}_{ijkl}$  and  $\mathcal{T}_{ijk}$ , and we write them

```

In[]:= quantity = {RLambda[-a, -b, -c, -d],
  ↳ RLambda[a, -a, -c, -d], RLambda[-a, -b,
  ↳ -c, c], TLambda[a, -b, -c], TLambda[a, -
  ↳ b, b]}
Out[]:=
  {λabcd, λacda, λabcc, λbca, λba}
In[]:= ScreenDollarIndices[ToCanonical[quantity
  ↳ ]]
Out[]:=
  {λabcd, 0, 0, λbca, 0}

```

We emphasise again that rules such as `ExpandStrengths`, `PADMActivate` and `PiPToPi` should not often be needed, and are subsumed under the `ToBasicForm[]` command.

## B. Irreducible decompositions

We recall that the Hamiltonian analysis invites decomposition into irreducible representations of the Lorentz group  $SO^+(1,3)$ , and (through the ADM split) the special orthogonal group  $SO(3)$ . We do not often encounter the former in the course of Hamiltonian calculations, but it is implemented across the higher-rank tensors, for example the ‘human-readable’ decomposition (into familiar irreps such as the *Weyl*, the *Ricci* and the *Ricci scalar*, and the *tensor*, *vector* and *pseudoscalar* torsion: see e.g. [1, 43, 72]) may be accessed as follows

```

In[]:= ScreenDollarIndices[ToCanonical[R[-a, -b
  ↳ , -c, -d] /. StrengthS013Activate]]
Out[]:=
  4/3 Rabcd + 2/3 Racdb + 2Rabdc + 3Rabdc +
  1/2 ybd 4Rac - 1/2 ybc 4Rad - 1/2 yad 4Rbc +
  1/2 yac 4Rbd + 1/2 ybd 5Rac - 1/2 ybc 5Rad - 1/2 yad 5Rbc +
  1/2 yac 5Rbd + 1/12 yad ybc 6R - 1/12 yac ybd 6R
In[]:= ScreenDollarIndices[ToCanonical[TLambda
  ↳ [-a, -b, -c] /.
  ↳ StrengthLambdaS013Activate]]
Out[]:=
  2/3 λabc - 2/3 λacb - 1/3 yac 2λb + 1/3 yab 2λc + εabca1 3λa1

```

More commonly, we might need access to the complete, orthonormal  $SO^+(1,3)$  projections which appear in (7). These may be associated with the couplings  $\{\hat{\alpha}_I\}$ ,  $\{\hat{\beta}_M\}$ ,  $\{\tilde{\alpha}_I\}$ ,  $\{\tilde{\beta}_M\}$  as follows, for the purposes of constructing a Lagrangian

```

In[]:= quantity = (Alp1*PR1[-i, -k, -l, -m, a,
  ↳ b, c, d] + Alp2*PR2[-i, -k, -l, -m, a, b,
  ↳ , c, d] + Alp3*PR3[-i, -k, -l, -m, a, b,
  ↳ c, d] + Alp4*PR4[-i, -k, -l, -m, a, b,
  ↳ c, d] + Alp5*PR5[-i, -k, -l, -m, a, b, c,
  ↳ , d] + Alp6*PR6[-i, -k, -l, -m, a, b, c,
  ↳ d])*R[-a, -b, -c, -d]

```

```

Out[]:=
  (α11 Piklmabcd + α22 Piklmabcd + α33 Piklmabcd +
  α44 Piklmabcd + α55 Piklmabcd + α66 Piklmabcd) Rabcd
In[]:= quantity = ScreenDollarIndices[
  ↳ ToCanonical[quantity /. PActivate /. {
  ↳ Alp1 → 1, Alp2 → 1, Alp3 → 1, Alp4 →
  ↳ 1, Alp5 → 1, Alp6 → 1}]]
Out[]:=
  Riklm
In[]:= ScreenDollarIndices[ContractMetric[
  ↳ ToCanonical[T[i, k, l]*(Bet1*PT1[-i, -k,
  ↳ -l, a, b, c])*T[-a, -b, -c] + cAlp6*R[i
  ↳ , k, l, m]*PR6[-i, -k, -l, -m, a, b, c,
  ↳ d]*R[-a, -b, -c, -d] + T[i, k, l]*(cBet2
  ↳ *PT2[-i, -k, -l, a, b, c] + cBet3*PT3[-i
  ↳ , -k, -l, a, b, c])*TLambda[-a, -b, -c]
  ↳ /. PActivate]]]
Out[]:=
  1/6 α6 Racac Rilil + 2/3 β1 Tabc Tabc +
  2/3 β1 Tabc Tbac - 2/3 β1 Taai Tcci +
  1/3 β3 Tabc λabc - 2/3 β3 Tabc λbac + 2/3 β2 Taai λcci

```

From this we see how to access the  $I\hat{\mathcal{P}}_{ij}^{klpq}$  and  $M\hat{\mathcal{P}}_j^{klpq}$  projections.

The  $SO(3)$  projections are similarly defined, but of course refer to the vector  $n_i$ , not just the Lorentzian metric  $\eta_{ij}$ . The parallel momenta can be decomposed in this way

```

In[]:= ScreenDollarIndices[ToCanonical[{APiP[-a
  ↳ , -b, -c], BPiP[-a, -b]} /. PiPToPi03]]
Out[]:=
  {-1/2 ybc 1/2 πaa + 1/2 yac 1/2 πbb + 4/3 2/3 πabca +
  1/3 ybc 0/2 πaa na + 1/2 πabca na + 2/3 πabca na -
  1/6 εabca1 0/2 πaa1 - 1/3 yac 0/2 πaa nb - 1/2 πabca nb -
  2/3 πabca nb - 1/2 1/2 πaa nc + 1/2 1/2 πaa nb nc -
  1/3 yab 0/2 πbb + 1/2 πaa nb + 2/3 πaba nb + 1/2 πaa nb nc -
  1/3 0/2 πbb na nb}

```

The parallel parts of the field strengths can also be decomposed, since they are canonical, but we neglect the perpendicular parts which depend on the unphysical fields  $b_0^i$  and  $A_0^{ij}$ , and also on non-canonical velocities. The field strength decomposition is

$$\mathcal{R}_{ijkl} = \mathcal{R}_{ijkl} + 2n_{[k} \mathcal{R}_{ij\perp]l}, \quad (13a)$$

$$\mathcal{T}_{ikl} = \mathcal{T}_{ikl} + 2n_{[k} \mathcal{T}_{i\perp]l}. \quad (13b)$$

Note that both parallel and perpendicular parts of the multipliers are decomposed, since all multiplier fields are assumed to be canonical. The resultant decomposition is

```

In[]:= quantity = ScreenDollarIndices[
  ↳ ToCanonical[{R[-a, -b, -c, -d], RLambda
  ↳ [-a, -b, -c, -d], T[-a, -b, -c], TLambda
  ↳ [-a, -b, -c]} /. StrengthDecompose /.
  ↳ StrengthLambdaDecompose]]
Out[]:=
  {R^1abcd - R^1abd n_c + R^1abc n_d,
   λ^1abcd - λ^1abd n_c + λ^1abc n_d,
   T^1abc - T^1ac n_b + T^1ab n_c, λ^1abc - λ^1ac n_b + λ^1ab n_c}
In[]:= Simplify[ScreenDollarIndices[
  ↳ ContractMetric[ToCanonical[quantity[[1]]
  ↳ /. StrengthPToStrengthP03]]]]
Out[]:=
  1/6 (-6 y_bd T^1ac + 6 y_bc T^1ad + 6 y_bd 2^1R^1ac - 6 y_bc 2^1R^1ad +
    3 y_bd T^1c n_a - 3 y_bc T^1d n_a + 8 2^1R^1cdb n_a -
    ε y_bcd a 1 Q^1R^1 n_a n_a^1 - 8 2^1R^1cda n_b + ε y_acda 1 Q^1R^1 n_a^1 n_b -
    6 R^1abd n_c - y_bd Q^1R^1 n_a n_c + 6 T^1bd n_a n_c -
    6 2^1R^1bd n_a n_c - 6 T^1ad n_b n_c + 6 2^1R^1ad n_b n_c + y_ad
    (- y_bc Q^1R^1 + 6 T^1bc - 6 2^1R^1bc - 3 T^1c n_b + Q^1R^1 n_b n_c) +
    6 R^1abc n_d + y_bc Q^1R^1 n_a n_d - 6 T^1bc n_a n_d +
    6 2^1R^1bc n_a n_d + 6 T^1ac n_b n_d - 6 2^1R^1ac n_b n_d + y_ac
    (y_bd Q^1R^1 - 6 T^1bd + 6 2^1R^1bd + 3 T^1d n_b - Q^1R^1 n_b n_d))
In[]:= Simplify[ScreenDollarIndices[
  ↳ ContractMetric[ToCanonical[quantity[[4]]
  ↳ /. StrengthLambdaPToStrengthLambdaP03
  ↳ /.
  ↳ StrengthLambdaPerpToStrengthLambdaPerp03
  ↳ ]]]]
Out[]:=
  1/6 (8 2^1λ^1bca + 6 T^1λ^1bc n_a - ε y_abc a 1 Q^1λ^1 n_a^1 - 6 T^1λ^1ac n_b -
    6 2^1λ^1ac n_b + 3 T^1λ^1c n_a n_b - 6 T^1λ^1c n_a n_b +
    y_ac (3 T^1λ^1b - 2 Q^1λ^1 n_b) + 6 T^1λ^1ab n_c + 6 2^1λ^1ab n_c -
    3 T^1λ^1b n_a n_c + 6 T^1λ^1b n_a n_c + y_ab (-3 T^1λ^1c + 2 Q^1λ^1 n_c))

```

It is possible, if needed, to recover the explicit ‘human-readable’ projection operators used in the above calculations, such as appear in  $A\hat{\pi}_i \equiv A\check{\mathcal{P}}_{i\bar{k}}^{ij}\hat{\pi}_{ij}^{\bar{k}}$  and  $E\hat{\pi}_i \equiv E\check{\mathcal{P}}_{i\bar{k}}^i\hat{\pi}_{ij}^{\bar{k}}$ , where the  $J^P$  sectors are represented by  $A$  and  $E$  indices according to our conventions in [62]. To reflect the various conventions used in the literature, these projections are sometimes defined differently for spin representations contained within parts of the field strengths and momenta whose tensor structures are identical – for example  $\hat{\pi}_{ijk}$  and  $\mathcal{T}_{ijk}$ . We therefore use different symbol names for momenta and field strengths (or multipliers) in general, and find e.g. for the selection

```

In[]:= quantity = ScreenDollarIndices[
  ↳ ContractMetric[ToCanonical[{PB0p[e, f]*
  ↳ PBPara[-e, -f, a, c]*BPiP[-a, -c], PB1m
  ↳ [-n, f]*PBPerp[-f, a, c]*BPiP[-a, -c],
  ↳ PA2p[-n, -m, e, f]*PAPerp[-e, -f, a, b,
  ↳ c]*APiP[-a, -b, -c], PROp[e, f, g, h]*RP
  ↳ [-e, -f, -g, -h], PROm[e, f, g]*RPerp[-e

```

```

  ↳ , -f, -g], PR1p[-n, -m, e, f, g, h]*
  ↳ RLambdaP[-e, -f, -g, -h], PT1p[-a, -b, c
  ↳ , d]*TPerp[-c, -d], PT2m[-a, -b, -c, d,
  ↳ e, f]*TP[-d, -e, -f]} /. PADMPiActivate
  ↳ ]]]
Out[]:=
  {π_b^1 ae Q^1P_b ae - π_b^1 ae Q^1P_b f e n_a n_f, π_b^1 af T^1P_b nf n_a,
   - π_a^1 aeb 2^1P_a nm ab n_e, Q^1P_a efgh R^1efgh Q^1P_a efgh R^1efg,
   T^1P_a nm efgh λ^1efgh T^1P_a b cd T^1cd 2^1P_a bcd def T^1def}
In[]:= quantity = quantity /. P03PiActivate /.
  ↳ P03TActivate /. P03RActivate /.
  ↳ PiPToPiP03 /. StrengthPToStrengthP03 /.
  ↳ StrengthPerpToStrengthPerp03 /.
  ↳ StrengthLambdaPToStrengthLambdaP03 /.
  ↳ StrengthLambdaPerpToStrengthLambdaPerp03
  ↳ ; quantity = Simplify[
  ↳ ScreenDollarIndices[ContractMetric[
  ↳ ToCanonical[quantity]]]; quantity =
  ↳ Simplify[ScreenDollarIndices[
  ↳ ContractMetric[ToCanonical[quantity]]]]];
Out[]:=
  {Q^1π_b, T^1π_b n, 2^1π_a nm, Q^1R^1, Q^1R^1, T^1λ^1_m n,
   T^1ab, 1/3 (2 2^1T^1abc + 2 T^1acb - 2 T^1bca)}

```

Note above that it is sometimes necessary to repeatedly apply routines in order to recover the desired form. In principle, the *HiGGS* commands are constructed so as to be idempotent, but broadly in *Mathematica* and *xAct* repeated commands are sometimes helpful when multiple functions are nested.

## C. Derivatives

While the gauge-fixed Minkowskian setup initially makes for easy development, it sometimes means that we have to ‘reinvent the wheel’ in order to access machinery for which there is already a very sophisticated implementation in *xAct*. A clear example of this is given by the way in which *HiGGS* handles gauge covariant derivatives in the Poincaré gauge theory. Generalising from (5), we recall from [7, 13, 65, 66] that for some matter field  $\varphi$  we have

$$D_i \varphi \equiv h_i^\mu \left( \partial_\mu + \frac{1}{2} A^{kl} \Sigma_{kl} \cdot \right) \varphi, \quad D_\mu \varphi \equiv b_\mu^i D_i \varphi, \quad (14)$$

where  $\Sigma_{ij}$  are the Lorentz group generators specific to the representation of  $\varphi$ . This construction could be implemented using the *xAct* command `DefCovD[]`, in such a way that the  $\Sigma_{ij}$  generators are automatically calculated for tensorial  $\varphi$  representations of the kind that arise in the Hamiltonian analysis of the matter-free theory. In particular, the derivative  $D_\mu \varphi$  is geometrically interpreted as  $\nabla_\mu \varphi$  as it appears in (3). Rather than following this route, *HiGGS* defines a pair of first derivatives (broadly corresponding to the two definitions in (14)) for every canonical quantity which might be of interest. This process is not very efficient or flexible, but it is sufficient when explicit covariant derivatives are scarce. Indeed, the most common oc-



currence of the gauge covariant derivative is through its commutator

$$2D_{[i}D_{j]}\varphi = \left(\frac{1}{2}\mathcal{R}^{kl}_{ij}\Sigma_{kl} \cdot -\mathcal{T}^k_{ij}D_k\right)\varphi, \quad (15)$$

in the form of the field strengths: as we saw in Section II A, these have a separate implementation. Explicit gradients will tend to arise mostly at the *end* of the calculations for which *HiGGS* is designed. Gradients of the field strengths cannot usually be fed back into the algorithm anyway, since they would require an implementation of the second order Euler–Lagrange equations in order to be processed.

Based on the understanding that we are always dealing with arbitrarily-indexed (possibly mixed) tensorial representations, *HiGGS*’ two preferred forms of the derivative are  $D_\mu\varphi_{\dot{\mu}\dot{\mu}}$  and  $\delta^{\dot{\mu}}_{\dot{\nu}}\delta^{\dot{\nu}}_{\dot{s}}h^{\dot{\mu}}_{\dot{w}}D_{\dot{k}}\varphi_{\dot{\mu}\dot{\mu}}$ . The former is straightforward<sup>5</sup>, and the latter is the parallel projection of the gradient on all indices. Picking a couple of SO(3) irreps at random, we find for example

```
In [] := quantity = {DPiPA2m[-z, -a, -b, -c], CD
  ↳ [-z][PiPA2m[-a, -b, -c]], DRP2p[-z, -a,
  ↳ -b], CD[-z][RP2p[-a, -b]]}
Out [] =
```

```
In [] := quantity = ScreenDollarIndices[ContractMetric[ToCanonical[(G3[-y, z]*#1 & ) /@ quantity
  ↳ [{1, 3}]]]]
Out [] =

In [] := quantity = ScreenDollarIndices[ContractMetric[ToCanonical[quantity /. DpRPActivate /.
  ↳ DpPiPActivate]]]
Out [] =
{b^a1b1 (D^{\hat{\pi}}_{\hat{\pi}})_{a1ab} v^lyb1 - D^{\hat{\pi}}_{\hat{\pi}} b^1_{ba1c} v^lyb1 n_a n^{a1} + D^{\hat{\pi}}_{\hat{\pi}} b^1_{aa1c} v^lyb1 n^{a1} n_b +
  D^{\hat{\pi}}_{\hat{\pi}} b^1_{aba1} v^lyb1 n^{a1} n_c + D^{\hat{\pi}}_{\hat{\pi}} c^1_{ba1b1} v^lyc1 n_a n^{a1} n^{b1} n_c - D^{\hat{\pi}}_{\hat{\pi}} c^1_{aa1b1} v^lyc1 n^{a1} n_b n^{b1} n_c,
  b^a1b1 (D^{\hat{\mathcal{R}}}^l)_{a1ab} v^lyb1 + D^{\hat{\mathcal{R}}}^l b^1_{ba1} v^lyb1 n_a n^{a1} + D^{\hat{\mathcal{R}}}^l b^1_{aa1} v^lyb1 n^{a1} n_b - D^{\hat{\mathcal{R}}}^l c^1_{a1b1} v^lyc1 n_a n^{a1} n_b n^{b1}}
In [] := quantity = ScreenDollarIndices[ContractMetric[ToCanonical[quantity /. DpRPDeactivate /.
  ↳ DpPiPDeactivate]]]
Out [] =
{D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} z abc, D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} abc, D^{\hat{\mathcal{R}}}^l \mathcal{R}^l zab, D^{\hat{\mathcal{R}}}^l \mathcal{R}^l ab}
```

We see in the penultimate expression above the two parallel derivatives. If an expression containing gradients is covariant, velocity-independent and parallel, then the various other terms should cancel among themselves. As we have mentioned, covariance also requires that gauge fields do not explicitly appear, but are implicit in covariant quantities. Overall, the work of packaging an expression into a canonical, covariant form, parallel if possible and free from unphysical fields, is done by the `ToNesterForm[]` command, which we introduce in Section II D.

<sup>5</sup> Note that we use an accent to indicate an arbitrary number of indices, following from [62, 73].

```
{D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} z abc, D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} abc, D^{\hat{\mathcal{R}}}^l \mathcal{R}^l zab, D^{\hat{\mathcal{R}}}^l \mathcal{R}^l ab}
In [] := quantity = ScreenDollarIndices[
  ↳ ContractMetric[ToCanonical[quantity /.
  ↳ DRPDeactivate /. DPiPDeactivate]]]
Out [] =
{A_b^{a1} z \pi_{\hat{\pi}} aa1c + A_c^{a1} z \pi_{\hat{\pi}} aba1 -
  A_a^{a1} z \pi_{\hat{\pi}} ba1c + D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} abc, D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} abc,
  A_b^{a1} z \mathcal{R}^l aa1 + A_a^{a1} z \mathcal{R}^l ba1 + D^{\hat{\mathcal{R}}}^l \mathcal{R}^l ab, D^{\hat{\mathcal{R}}}^l \mathcal{R}^l ab}
In [] := quantity = ScreenDollarIndices[
  ↳ ContractMetric[ToCanonical[quantity /.
  ↳ DRPActivate /. DPiPActivate]]]
Out [] =
{D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} z abc, D^{\hat{\pi}}_{\hat{\pi}} \pi_{\hat{\pi}} z abc - A_b^{a1} z \pi_{\hat{\pi}} aa1c -
  A_c^{a1} z \pi_{\hat{\pi}} aba1 + A_a^{a1} z \pi_{\hat{\pi}} ba1c, D^{\hat{\mathcal{R}}}^l \mathcal{R}^l zab,
  D^{\hat{\mathcal{R}}}^l \mathcal{R}^l zab - A_b^{a1} z \mathcal{R}^l aa1 - A_a^{a1} z \mathcal{R}^l ba1}
```

One cannot generally pass from  $D_\mu\varphi_{\dot{\mu}\dot{\mu}}$  to a parallel derivative, unless only the spacelike indices are involved, i.e.  $D_\alpha\varphi_{\dot{\mu}\dot{\mu}}$ . Since *HiGGS* tries to achieve parallel derivatives wherever possible, this serves as one of the internal checks on the canonical status of a quantity, making sure that any unacceptable time derivatives would appear explicitly in the output. Accordingly, continuing from above

As a final comment on the use of derivatives, we recall that the formulae Eqs. (6a) and (6b) imply a pair of Bianchi identities [7, 72] as follows

$$\epsilon^{\rho\mu\lambda\nu}D_\mu\left(b^i_\lambda b^j_\nu \mathcal{T}^s_{ij}\right) = \epsilon^{\rho\mu\lambda\nu}b^k_\mu b^i_\lambda b^j_\nu \mathcal{R}^s_{kij}, \quad (16a)$$

$$\epsilon^{\rho\mu\lambda\nu}D_\lambda\left(b^k_\mu b^l_\nu \mathcal{R}^{ij}_{kl}\right) = 0. \quad (16b)$$

These can be verified using the tools already introduced, but the process will be easier with the `ToBasicForm[]` and `ToNesterForm[]` commands.

## D. Low-level functions

Throughout Sections II A and II B we have used many rules, such as `DpRPDeactivate`, `DpPiActivate`, etc., which — whilst they are not confined to `xAct` ‘`HiGGS`’ `Private`’ and so are available to the user — should not often be needed within a *HiGGS* science session. Instead these rules are wrapped into two ‘official’ functions: `ToNesterForm[]`, which strives to collect expressions<sup>6</sup>, and `ToBasicForm[]`, which expands them. Naturally `ToNesterForm[]` is the more complicated of the two. It is, in some sense, the extension of the `ToCanonical[]` command from *xAct* into *HiGGS*.

### 1. ToBasicForm

We begin by breaking some expressions which we know to be covariant, using `ToBasicForm[]`

```
In[] := ToBasicForm[DpPiPB1m[-k, -i]]
```

```
Out[] =
```

$$\begin{aligned} & \mathcal{A}_{ia}^{c1} b^{aa1} \pi_b^{bb1} \gamma^{a1b1} \gamma^{cc1} h_k^c n_b + \mathcal{A}_{ac1}^d b^{aa1} \pi_b^{bb1} \gamma^{a1b1} \gamma^{cd} h_k^c n_b n^{c1} n_i - \\ & \mathcal{A}_{ia}^d b^{aa1} \pi_b^{bb1} \gamma^{a1b1} \gamma^{c1d} h^{cc1} n_b n_c n_k - \mathcal{A}_{ad}^{d1} b^{aa1} \pi_b^{bb1} \gamma^{a1b1} \gamma^{c1d1} h^{cc1} n_b n_c n^d n_i n_k + \\ & \pi_b^{aa1} \gamma^{a1b1} \gamma^{bc} h_k^b n_a \partial^c b_i^{b1} + b_i^a \gamma^{ab1} \gamma^{a1c} h_k^{a1} n_b \partial^c \pi_b^{b1} - \\ & \pi_b^{aa1} \gamma^{a1c} \gamma^{bc1} h_k^b n_a n^{b1} n_i \partial^{c1} b_{b1}^c - \pi_b^{aa1} \gamma^{a1c} \gamma^{b1c1} h^{bb1} n_a n_b n_k \partial^{c1} b_i^c - \\ & b_i^a \gamma^{ac} \gamma^{bc1} h^{a1b} n_{a1} n^{b1} n_k \partial^{c1} \pi_b^{b1} + \pi_b^{aa1} \gamma^{a1c1} \gamma^{b1d} h^{bb1} n_a n_b n^c n_i n_k \partial^d b_c^{c1} - \\ & b_i^a \pi_b^{a1b} \gamma^{ab} \gamma^{b1d} h_{a1}^c h_k^{b1} n^{c1} \partial^d b_{c1}^c + b_i^a \pi_b^{a1b} \gamma^{ab} \gamma^{c1d1} h_{a1}^{b1} h^{cc1} n_c n^d n_k \partial^{d1} b_{db1} + \\ & b_i^a \pi_b^{a1b} \gamma^{ab} \gamma^{b1d1} h^{cc1} h_k^{b1} n_{a1} n_c n^d \partial^{d1} b_{dc1} - b_i^a \pi_b^{a1b} \gamma^{ab} \gamma^{de} h^{b1c} h^{c1d} n_{a1} n_{b1} n_{c1} n^{d1} n_k \partial^e b_{d1}^c \end{aligned}$$

The gradient of a more complicated irrep such as `DpPiPA2m[-i,-j,-k]` would span many pages after an application of `ToBasicForm[]`, rendering manual evaluation of brackets impractical, yet we will see in Section III that *there is nothing to stop these terms arising in the analysis*.

Now is a convenient time to verify the second Bianchi identity in (16b). Setting up the derivative, and using the gauge-fixed  $\epsilon^{\mu\nu\sigma\lambda}$  which follows from  $\gamma_{\mu\nu} - \text{epsilonG}[a,b,c,d]$  — we find

```
In[] := quantity2 = ToBasicForm[epsilonG[r, m, l
  ↳ , n]*(CD[-l][B[k, -m]*B[q, -n]*R[i, j, -
  ↳ k, -q]] + A[i, -x, -l]*B[k, -m]*B[q, -n
  ↳ ]*R[x, j, -k, -q] + A[j, -x, -l]*B[k, -m
  ↳ ]*B[q, -n]*R[i, x, -k, -q]);
```

```
Out[] =
```

$$2 \epsilon^{\gamma r}_{\alpha l m} \partial^m \partial^j \mathcal{A}^{ija}$$

```
In[] := quantity = ToBasicForm[{T[i, -j, -k],
  ↳ PiPB0p[], PiPA1p[-i, -j]}]
Out[] =
```

$$\begin{aligned} & \{ \mathcal{A}_{ka}^i h_j^a - \mathcal{A}_{ja}^i h_k^a + h_j^a h_k^{a1} \partial_a b_{a1}^i - h_j^a h_k^{a1} \partial_{a1} b_a^i, \\ & b^{aa1} \pi_b^b \gamma^{a1b}, \\ & \frac{1}{2} \pi_{ja1}^b b_i^a \gamma^{lab} n^{a1} - \frac{1}{2} \pi_{ia1}^b b_j^a \gamma^{lab} n^{a1} \} \end{aligned}$$

We see that the results are expressed in terms of the bare gauge fields, non-parallel momenta, independent ADM quantities and coordinate derivatives. If a Poisson bracket were to be manually evaluated between the fields in quantity, it would first be necessary to perform the expansion provided by `ToBasicForm[]`, before calculating variational derivatives and multiplying. Indeed, the `PoissonBracket[]` command which we introduce shortly relies on `ToBasicForm[]` as an initial step. In the case above, all the brackets would be relatively straightforward to evaluate by hand: to see why the strong coupling problem demands an implementation such as *HiGGS*, let us break open a simple momentum gradient

Evidently, this quantity vanishes, and so what we are seeing is a side-effect of the *HiGGS* geometric setup: *xAct* does not know that `CD[-i][ ]` is the partial derivative, so we need a final step

```
In[] := quantity2 = (ToCanonical[(1/2)*(#1 +
  ↳ CommuteCovDs[#1, CD, {l, m}]])) & )[
  ↳ quantity2];
```

```
Out[] =
```

0

### 2. ToNesterForm

Let us now undo the breaking, using `ToNesterForm[]`. In general, much of the functionality of `ToNesterForm[]` has to do with fully incorporating the known primary and secondary constraints of the theory during the course of simplification — so as to leave no extra conditions for the user to worry about. Before a theory has been defined (see Section II E) using `DefTheory[]`, we will have to suppress this activity by passing the option `"ToShell" → False`, yielding

<sup>6</sup> The chosen naming refers to the fact that the irrep conventions and notation of collected expressions tends to align most closely with those of a collection of articles — very useful during development — for which Nester is a common author, see e.g. [1, 42, 43, 52, 68, 74–79].

```
In[]:= quantity = ToNesterForm[quantity, "
  ↳ ToShell" → False]
Out[]:=
```

$$\left\{ -\frac{1}{6} \epsilon^{ijk} \mathcal{F}^k + \frac{1}{2} \delta^i_k \mathcal{F}^j - \frac{1}{2} \delta^j_k \mathcal{F}^i + \frac{4}{3} \mathcal{F}^i \mathcal{F}^j \delta^k + \right.$$

$$\left. \mathcal{F}^i \mathcal{F}^j \delta^k - \frac{1}{3} \delta^i_k \mathcal{F}^j n_j - \mathcal{F}^i \mathcal{F}^j \delta^k n_j - \mathcal{F}^i \mathcal{F}^j \delta^k n_j + \right.$$

$$\left. \frac{1}{2} \mathcal{F}^i \mathcal{F}^j \delta^k n_j - \mathcal{F}^i \mathcal{F}^j \delta^k n_j + \frac{1}{3} \delta^i_j \mathcal{F}^k n_k + \mathcal{F}^i \mathcal{F}^j \delta^k n_k + \right.$$

$$\left. \mathcal{F}^i \mathcal{F}^j \delta^k n_k - \frac{1}{2} \mathcal{F}^i \mathcal{F}^j \delta^k n_k + \mathcal{F}^i \mathcal{F}^j \delta^k n_k, \hat{\pi}^i_b, \hat{\pi}^i_{\mathcal{A}ij} \right\}$$

Note that whilst the results are covariant, we do not get back the same form from `ToNesterForm[]` as that which was provided to `ToBasicForm[]`, instead we recover the SO(3) ex-

pansion. The reason for this is that reducible quantities such as `T[-a, -b, -c]` are usually quickly broken up during the course of the Hamiltonian analysis: the output of `ToNesterForm[]` is tuned so as to be useful in that context. Moreover, we note that this operation will not now work in reverse, since input such as `TP2m[-a, -b, -c]//ToBasicForm` will not return a broken expression. There is no special reason behind this: `ToBasicForm[]` is not a sophisticated function, it essentially imposes a list of internal rules on its argument, and the expansion of SO(3) field strength irreps, as with those of momenta, would be perfectly straightforward to implement in the source if and when needed.

When `ToNesterForm[]` is passed a non-covariant quantity, it is unable to return a covariant result. Nonetheless, it tries, returning for the innocuous gradient  $\partial_\mu b^i_\nu$

```
In[]:= ToNesterForm[CD[-i][B[-a, -b]], "ToShell" → False]
Out[]:=
```

$$-\frac{1}{2} \mathcal{A}_{aa1i} b^{a1}_b + \frac{1}{2} \mathcal{A}_{aa1b} b^{a1}_i + \frac{1}{12} b^{a1}_b b^{b1}_i \epsilon^{aa1b1} \mathcal{F}^i - \frac{1}{4} b_{ai} b^{a1}_b \mathcal{F}^i a_1 + \frac{1}{4} b_{ab} b^{a1}_i \mathcal{F}^i a_1 -$$

$$\frac{2}{3} b^{a1}_b b^{b1}_i \mathcal{F}^i a_1 b_1 a - \frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i a_1 b_1 n_a + \frac{1}{6} b_{ai} b^{a1}_b \mathcal{F}^i n_{a1} - \frac{1}{6} b_{ab} b^{a1}_i \mathcal{F}^i n_{a1} + \frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i a_{b1} n_{a1} +$$

$$\frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i a_{b1} n_{a1} - \frac{1}{4} b^{a1}_b b^{b1}_i \mathcal{F}^i b_1 n_a n_{a1} + \frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i b_1 n_a n_{a1} - \frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i a_{a1} n_{b1} -$$

$$\frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i a_{a1} n_{b1} + \frac{1}{4} b^{a1}_b b^{b1}_i \mathcal{F}^i a_1 n_a n_{b1} - \frac{1}{2} b^{a1}_b b^{b1}_i \mathcal{F}^i a_1 n_a n_{b1} + \frac{1}{2} \partial_b b_{ai} + \frac{1}{2} \partial_i b_{ab}$$

As with this case, it is usually easy to identify nonphysical expressions which indicate a human error, through the appearance of bare gauge fields. If needed, this covariance check is easy to automate through an output search with the *Mathematica* `Head[]` function for an unwanted *HiGGS* quantity, such as `A` or `B`. We can also observe in the output above part of the route taken by `ToNesterForm[]`. Gauge field gradients are converted, where possible, to field strengths and covariant derivatives. The residual spin connection terms are then extensively manipulated in an attempt to cancel them. Leftover asymmetric derivatives of the translational gauge field can sometimes be ascribed to covariant quantities through (10), for example the following gradient cannot be expressed through the torsion alone

```
In[]:= ToNesterForm[PPara[-y, v]*H[-v, w]*G3[-w
  ↳ , m]*PPara[-q, b1]*H[-b1, a1]*G3[-a1, 1
  ↳ ]*(G3[-1, n]*V[-k]*CD[-m][B[k, -n]] + G3
  ↳ [-1, n]*B[j, -n]*A[k, -j, -m]*V[-k]) /.
  ↳ PADMActivate, "ToShell" → False]
Out[]:=
```

$$-\frac{1}{2} (\mathcal{D} n)^i q_y - \frac{(\mathcal{D} n)^i y_q}{2} - \frac{\mathcal{F}^i q_y}{2}$$

Note that the above result could equally be written as the single term `-DpV[-q, -y]`, but `ToNesterForm[]` takes the opportunity to separate out the antisymmetric part.

Before moving on, we return to verify the first Bianchi identity in (16b). We can access the canonical (i.e. velocity-

independent) part of this identity by projecting Eq. (16b) with  $b^\perp_\mu$ , or equivalently using in place of  $\epsilon^{\mu\nu\lambda\sigma}$  the foliation equivalent  $\epsilon^{\overline{ijk\perp}}$ , which is given in *HiGGS* by `Eps[i, j, k]`,

```
In[]:= quantity = ToBasicForm[Eps[u, v, w]*H[-u
  ↳ , m]*H[-v, l]*H[-w, n]*(CD[-m][B[i, -l]*
  ↳ B[j, -n]*T[s, -i, -j]]) + A[s, -x, -m]*B[
  ↳ i, -l]*B[j, -n]*T[x, -i, -j]]) /.
  ↳ PADMActivate];
Out[]:=
```

$$2 \mathcal{A}_n^u{}_m \mathcal{A}^{sn}{}_j \epsilon^{il u} h^{ij} h^{lm} - 2 \epsilon^{il n} h^{ij} h^{lm} \partial_m \mathcal{A}^{sn}{}_j$$

```
In[]:= quantity = (ToCanonical[V[-r]*(1/2)*(#1
  ↳ + CommuteCovDs[#1, CD, {j, l}]] & ) [
  ↳ quantity];
Out[]:=
```

$$2 \mathcal{A}_n^u{}_m \mathcal{A}^{sn}{}_j \epsilon^{il u} h^{ij} h^{lm} n_r - 2 \epsilon^{il n} h^{ij} h^{lm} n_r \partial_m \mathcal{A}^{sn}{}_j$$

```
In[]:= quantity = ToNesterForm[quantity, "
  ↳ ToShell" → False];
Out[]:=
```

$$2 \epsilon^{iS}{}_{ai} \mathcal{F}^i \mathcal{R}^{ai} n_r - \mathcal{F}^i \mathcal{R}^i n_r n^S + \frac{4}{3} \epsilon^{ai l} \mathcal{F}^i \mathcal{R}^{al} n_r n^S$$

The output here is then equal to the SO(3) irreps contained within  $\epsilon^{\overline{ijk\perp}} \mathcal{R}^s_{\overline{ki j}}$ , as expected.

### 3. PoissonBracket

The `PoissonBracket[]` command is the third ‘official’ function provided by the *HiGGS* package. The Poisson bracket appearing in this article is defined for general functionals  $\mathcal{A}$  and  $\mathcal{B}$  of the gravitational fields and their conjugate momenta

$$\{\mathcal{A}, \mathcal{B}\} \equiv \int d^3x \left[ \frac{\delta \mathcal{A}}{\delta b^i{}_\mu} \frac{\delta \mathcal{B}}{\delta \pi_i{}^\mu} + \frac{\delta \mathcal{A}}{\delta A^{ij}{}_\mu} \frac{\delta \mathcal{B}}{\delta \pi_{ij}{}^\mu} - \frac{\delta \mathcal{A}}{\delta \pi_i{}^\mu} \frac{\delta \mathcal{B}}{\delta b^i{}_\mu} - \frac{\delta \mathcal{A}}{\delta \pi_{ij}{}^\mu} \frac{\delta \mathcal{B}}{\delta A^{ij}{}_\mu} \right], \quad (17)$$

with a natural extension of the formula when multiplier fields are admitted. The formula (17) may appear no more daunting than a commonplace action variation, but in practice  $\mathcal{A}$  and  $\mathcal{B}$  are frequently *local tensors* rather than nonlocal scalars. Locality signifies that the underlying functionals contain Dirac distributions, themselves subject to the total derivatives of the generalised Euler–Lagrange equations. The full ramifications of covariantly removing these Dirac gradients are detailed in [13, 62], and some special cases are discussed in electrodynamics on the lightcone [80] and noncritical string theory [81]. Currently, *HiGGS* is able to accommodate the *first order* Euler–Lagrange formalism in `PoissonBracket[]`. First order brackets, evaluated by inserting the spatial dependence into (17), produce four terms of the form

$$\begin{aligned} \{\mathcal{A}(\mathbf{x}_1), \mathcal{B}(\mathbf{x}_2)\} \equiv & \int d^3x [J_1(\mathbf{x}) \delta^3(\mathbf{x} - \mathbf{x}_1) \delta^3(\mathbf{x} - \mathbf{x}_2) \\ & + J_2^\alpha(\mathbf{x}) \delta^3(\mathbf{x} - \mathbf{x}_1) \partial_\alpha \delta^3(\mathbf{x} - \mathbf{x}_2) \\ & + J_3^\alpha(\mathbf{x}) \partial_\alpha \delta^3(\mathbf{x} - \mathbf{x}_1) \delta^3(\mathbf{x} - \mathbf{x}_2) \end{aligned}$$

$$+ J_4^{\alpha\beta}(\mathbf{x}) \partial_\alpha \delta^3(\mathbf{x} - \mathbf{x}_1) \partial_\beta \delta^3(\mathbf{x} - \mathbf{x}_2)]. \quad (18)$$

Note that by our conventions in [12, 13, 62], we will in future denote by  $\delta^3$  the equal-time Dirac function  $\delta^3(\mathbf{x}_1 - \mathbf{x}_2)$ . Without any special instructions, `PoissonBracket[]` returns a `List` of the four Dirac coefficients in (18). This behaviour is tied into calls to `PoissonBracket[]` from within `Velocity[]`, which we introduce in Section II E. Note that `PoissonBracket[]` contains calls to `ToNesterForm[]`, and so works to exhaust transformations which can be applied to the output by virtue of the known primary and secondary constraints. In this sense, it depends on the theory introduced by `DefTheory[]` and so for the time being we must again pass the option `"ToShell" → False`. We begin with a very simple bracket, whose output can be understood in terms of our previous `ToNesterForm[]` result for `DpV[-a, -b]`

```
In[] := PoissonBracket[PiPB2p[-a, -b], TP1p[-c,
  ↳ -d], "ToShell" → False];
Out[] =
  {2 n_{ab} · T_{cd}} =
  {-1/4 (D n)^{bd} η^{ac} - 1/4 (D n)^{db} η^{ac} + 1/4 (D n)^{bc} η^{ad} +
    1/4 (D n)^{cb} η^{ad} - 1/4 (D n)^{ad} η^{bc} -
    1/4 (D n)^{da} η^{bc} + 1/4 (D n)^{ac} η^{bd} + 1/4 (D n)^{ca} η^{bd} -
    1/4 η^{bd} T_{ac} + 1/4 η^{bc} T_{ad} - 1/4 η^{ad} T_{bc} +
    1/4 η^{ac} T_{bd} - 1/3 η^{ab} T_{cd}, 0, 0, 0}
```

The bracket is not ‘surficial’, in the sense that the latter three entries vanish and the nonvanishing part of the bracket is a compact, covariant expression. If the latter three coefficients are nonvanishing however, explicit covariance in this expression will be lost, as we can see by modifying the previous example

```
In[] := PoissonBracket[PiPB2p[-a, -b], TP1m[-c], "ToShell" → False];
Out[] =
```

$$\begin{aligned} \{2 n_{ab} · T_{cd}\} = & \left\{ -\frac{1}{2} \mathcal{A}_{bca1} h_a^{a1} - \frac{1}{3} \mathcal{A}_{ca1b1} \eta^{ab} h^{a1b1} - \frac{1}{2} \mathcal{A}_{aca1} h_b^{a1} - \frac{(D\mathcal{T})^l{}_c \eta^{ab}}{3\mathcal{T}} + \frac{(D\mathcal{T})^l{}_b \eta^{ac}}{2\mathcal{T}} + \frac{(D\mathcal{T})^l{}_a \eta^{bc}}{2\mathcal{T}} + \frac{1}{2} \mathcal{A}_{bcb1} h^{a1b1} n_a n_{a1} + \right. \\ & \frac{1}{2} \mathcal{A}_{acb1} h^{a1b1} n_{a1} n_b - \frac{1}{2} \mathcal{A}_{cb1a1} h_b^{a1} n_a n^{b1} - \frac{1}{2} \mathcal{A}_{cb1a1} h_a^{a1} n_b n^{b1} + \frac{1}{2} (D n)^l{}_{ab} n_c + \frac{1}{2} (D n)^l{}_{ba} n_c - \\ & \frac{1}{3} (D n)^l{}_{a1} \eta^{ab} n_c + \frac{1}{3} \mathcal{A}_{cc1b1} \eta^{ab} h^{a1b1} n_{a1} n^{c1} + \mathcal{A}_{cc1b1} h^{a1b1} n_a n_{a1} n_b n^{c1}, 0, \\ & \left. -\frac{1}{2} \eta^{bc} h_a^v - \frac{1}{2} \eta^{ac} h_b^v + \frac{1}{3} \eta^{ab} h_c^v + \frac{1}{2} \eta^{bc} h^{a1v} n_a n_{a1} + \frac{1}{2} \eta^{ac} h^{a1v} n_{a1} n_b - \frac{1}{3} \eta^{ab} h^{a1v} n_{a1} n_c, 0 \right\} \end{aligned}$$

In such ‘surficial’ cases the default output of `PoissonBracket[]` will not be helpful for visual inspection. To resolve this we can recall, again from [13],

that (18) can be alternatively expressed as

$$\begin{aligned} \int d^3x_2 \{ \mathcal{A}_a(\mathbf{x}_1), \mathcal{B}_b(\mathbf{x}_2) \} C^b(\mathbf{x}_2) \equiv & \mathcal{J}_{1b}(\mathbf{x}_1) C^b(\mathbf{x}_1) \\ & + \mathcal{J}_{2b}^\alpha(\mathbf{x}_1) D_\alpha C^b(\mathbf{x}_1) + \mathcal{J}_{3b}^{\alpha\beta}(\mathbf{x}_1) D_\alpha D_\beta C^b(\mathbf{x}_1). \end{aligned} \quad (19)$$



The three-component `List` output corresponding to (19) can be produced by passing the option "Surficial"  $\rightarrow$  `True`. Trying again with this option, we obtain

```
In[]:= PoissonBracket[PiPB2p[-i, -j], TP1m[-1],
  ↳ "ToShell"  $\rightarrow$  False, "Surficial"  $\rightarrow$  True
  ↳ ];
Out[]:=
```

$$\{2^{\wedge} \pi_{bij}, 1^{\wedge} \tau_{ij}\} = \left\{ -\frac{(\mathcal{D}\tau)_{ij}^{\parallel} \eta^{\parallel ij}}{3\mathcal{J}} + \frac{(\mathcal{D}\tau)_{ij}^{\parallel} \eta^{\parallel il}}{2\mathcal{J}} + \frac{(\mathcal{D}\tau)_{ij}^{\parallel} \eta^{\parallel jl}}{2\mathcal{J}} + \right.$$

$$\left. \frac{1}{2}(\mathcal{D}n)_{ij}^{\parallel} n_i + \frac{1}{2}(\mathcal{D}n)_{ji}^{\parallel} n_j - \frac{1}{3}(\mathcal{D}n)_{\parallel a}^{\parallel} \eta^{\parallel ij} n_i, \right.$$

$$\left. \frac{1}{3}(\mathcal{D}\star)_{ij}^{\parallel} \eta^{\parallel ij} - \frac{1}{2}(\mathcal{D}\star)_{ij}^{\parallel} \eta^{\parallel il} - \frac{1}{2}(\mathcal{D}\star)_{ij}^{\parallel} \eta^{\parallel jl}, 0 \right\}$$

We see that the result is indeed covariant, and fairly simple. We will come back to this ‘surficial’ case in Section III B, where we attempt to recover the historical results in [1, 43].

### E. High-level functions

As we mentioned in Section II D, much of the work done by `ToNesterForm[]` has to do with the imposition of the theory shell so as to simplify the argument. The particular shell used is not specifically that of primary vs secondary constraints, but it is restricted to the constraints of which we have prior knowledge from the literature, and does not include new constraints discovered in the course of a *HiGGS* session. In particular, we rely on the so-called *if-constraint* structure which was discovered by Blagojević and Nikolić in [58]. Depending on the Lagrangian parameters in (7), the number and type of primary constraints may be radically different: these contingent primaries are called primary if-constraints (PiC). There are similarly secondary (SiC) and tertiary (TiC) quantities, etc. Returning to [62], the PiCs of the theory (7) take the form

$${}^A\varphi_{\hat{v}} \equiv \frac{1}{J} {}^A\hat{\pi}_{\hat{v}} + 2\hat{\alpha}_0 m_p {}^2A\check{\mathcal{P}}_{\hat{v}\perp\bar{k}}^{\perp\bar{k}} - 8\hat{\alpha}_A^{\perp\perp} {}^A\check{\mathcal{P}}_{\hat{v}jk}^{\perp\bar{m}} \bar{\lambda}^{jk}_{\perp\bar{m}} - 4{}^A\check{\mathcal{P}}_{\hat{v}jk}^{\perp\bar{m}} \left( \bar{\alpha}_A^{\perp\parallel} \lambda^{jk}_{\perp\bar{m}} + 2\hat{\alpha}_A^{\perp\parallel} \mathcal{R}^{jk}_{\perp\bar{m}} \right), \quad (20a)$$

$${}^E\varphi_{\hat{v}} \equiv \frac{1}{J} {}^E\hat{\pi}_{\hat{v}} - 4m_p {}^2\hat{\beta}_E^{\perp\perp} {}^E\check{\mathcal{P}}_{\hat{v}j}^{\perp\bar{m}} \bar{\lambda}^j_{\perp\bar{m}} - 2m_p {}^2E\check{\mathcal{P}}_{\hat{v}j}^{\perp\bar{m}} \left( \bar{\beta}_E^{\perp\parallel} \lambda^j_{\perp\bar{m}} + 2\hat{\beta}_E^{\perp\parallel} \mathcal{T}^j_{\perp\bar{m}} \right), \quad (20b)$$

where we defined e.g.  $\hat{\alpha}_A^{\parallel\parallel} \equiv \sum_I M_{AI}^{\parallel\parallel} \hat{\alpha}_I$ , with the matrix  ${}^A\check{\mathcal{P}}_{\hat{p}}^{\perp\bar{m}} {}^I\hat{\mathcal{P}}_{\bar{n}\bar{q}}^{\perp\bar{r}} {}^{\bar{r}\bar{k}}_{\bar{i}j} \equiv M_{AI}^{\parallel\parallel} {}^A\check{\mathcal{P}}_{\hat{p}ij}^{\perp\bar{r}\bar{k}}$ . In the handling of PiCs, it is extremely useful to refer to the functions

$$\mu(x) \equiv \begin{cases} x^{-1}, & \text{for } x \neq 0 \\ 0, & \text{for } x = 0, \end{cases} \quad \nu(x) \equiv 1 - |\text{sgn}(\mu(x))|, \quad (21)$$

and we note that the PiC functions defined in Eqs. (20a) and (20b) are only constrained when  $\nu(\hat{\alpha}_A^{\perp\perp}) = 1$  or  $\nu(\hat{\beta}_E^{\perp\perp}) = 1$ . The structure of Eqs. (20a) and (20b) is quite useful in that it allows momenta to substituted for parallel field strengths. These substitutions are among those performed every time `ToNesterForm[]` is called, unless the option "ToShell"  $\rightarrow$  `False` is passed as above in Section II D.

### 1. DefTheory

In order to discover these PiCs, we must first set up the shell using `DefTheory[]`. Let us consider the simple example of Einstein–Cartan theory, i.e. the substantial restriction of (7) to the simple Einstein–Hilbert term

$$L_G = -\frac{1}{2} \hat{\alpha}_0 m_p^2 \mathcal{R}. \quad (22)$$

We implement the theory (22) by passing to the `DefTheory[]` command the system of equations which deactivates all the  $\{\hat{\alpha}_A\}$ ,  $\{\hat{\beta}_E\}$ ,  $\{\hat{\alpha}_A\}$ ,  $\{\hat{\beta}_E\}$ , while leaving  $\hat{\alpha}_0$  untouched. We want to store our knowledge of the shell once it has been obtained, and so we pass the label "Export"  $\rightarrow$  "EinsteinCartan", which will be used to construct a filename. The input is

```
In[]:= DefTheory[{Alp1 == 0, Alp2 == 0, Alp3 ==
  ↳ 0, Alp4 == 0, Alp5 == 0, Alp6 == 0,
  ↳ Bet1 == 0, Bet2 == 0, Bet3 == 0, cAlp1
  ↳ == 0, cAlp2 == 0, cAlp3 == 0, cAlp4 ==
  ↳ 0, cAlp5 == 0, cAlp6 == 0, cBet1 == 0,
  ↳ cBet2 == 0, cBet3 == 0}, "Export"  $\rightarrow$  "
  ↳ EinsteinCartan", "Order"  $\rightarrow$  Infinity];
** DefTheory: Found the following primary if-
  ↳ constraints:
```

$$0^{\wedge} \phi_b \equiv \mathcal{J}^{-1} 0^{\wedge} \pi_b \approx 0$$

$$1^{\wedge} \phi_{bij} \equiv \mathcal{J}^{-1} 1^{\wedge} \pi_{bij} \approx 0$$

$$1^{\wedge} \phi_{bi} \equiv \mathcal{J}^{-1} 1^{\wedge} \pi_{bi} \approx 0$$

$$2^{\wedge} \phi_{bij} \equiv \mathcal{J}^{-1} 2^{\wedge} \pi_{bij} \approx 0$$

$$0^{\wedge} \phi_{\mathcal{R}} \equiv 3 \hat{\alpha}_0 + \mathcal{J}^{-1} 0^{\wedge} \pi_{\mathcal{R}} \approx 0$$

$$0^{\wedge} \phi_{\mathcal{R}} \equiv \mathcal{J}^{-1} 0^{\wedge} \pi_{\mathcal{R}} \approx 0$$

$$1^{\wedge} \phi_{\mathcal{R}ij} \equiv \mathcal{J}^{-1} 1^{\wedge} \pi_{\mathcal{R}ij} \approx 0$$

$$1^{\wedge} \phi_{\mathcal{R}i} \equiv \mathcal{J}^{-1} 1^{\wedge} \pi_{\mathcal{R}i} \approx 0$$

$$2^{\wedge} \phi_{\mathcal{R}ij} \equiv \mathcal{J}^{-1} 2^{\wedge} \pi_{\mathcal{R}ij} \approx 0$$

$$2^{\wedge} \phi_{\mathcal{R}ijk} \equiv \frac{2}{3} \mathcal{J}^{-1} 2^{\wedge} \pi_{\mathcal{R}ijk} + \frac{1}{3} \mathcal{J}^{-1} 2^{\wedge} \pi_{\mathcal{R}ikj} - \frac{1}{3} \mathcal{J}^{-1} 2^{\wedge} \pi_{\mathcal{R}jki} \approx 0$$

```
** DefTheory: Found the following secondary
  ↳ perpendicular if-constraints:
```

```
** DefTheory: Found the following secondary
  ↳ parallel if-constraints:
```

```
** DefTheory: Found the following secondary
  ↳ singular if-constraints:
```

```
** DefTheory: The super-Hamiltonian is:
```

$$0^{\wedge} \mathcal{H}_b \equiv \frac{\hat{\alpha}_0 \mathcal{J} 0^{\wedge} \mathcal{R}}{2} \approx 0$$

```

** DefTheory: The linear super-momentum is:


$$\mathcal{H}_{\perp} \equiv -\hat{\alpha}_0 \mathcal{T} \mathcal{T}^{\perp} \approx 0$$


** DefTheory: The 1- part of the angular super
    ↪ -momentum is:


$$\mathcal{H}_{\mathcal{A}m} \equiv -\hat{\alpha}_0 \mathcal{T} \mathcal{T}^{\perp} m \approx 0$$


** DefTheory: The 1+ part of the angular super
    ↪ -momentum is:


$$\mathcal{H}_{\mathcal{A}nm} \equiv \hat{\alpha}_0 \mathcal{T} \mathcal{T}^{\perp} m n \approx 0$$


** DefTheory: Exporting the binary at svy/
    ↪ EinsteinCartan.thr.mx

```

In the output above we can see the listing of the PiCs. It is clear from (22) and also from the form of the constraints in Eqs. (20a) and (20b), why we get this specific list. All the coefficients associated with the  $SO^+(1, 3)$  irreps of  $\mathcal{R}_{ijkl}$  and  $\mathcal{T}_{ijk}$  are vanishing, and so all the PiC functions become constraints. Moreover, these constraints are very simple in their form: they are all pure momenta, with the single exception being in the case of the  $0^+$  roton constraint, which contains a constant

$$\varphi_{\perp} = 3\hat{\alpha}_0 m_p^2 + \frac{1}{J} \hat{\pi}_{\perp} \approx 0. \quad (23)$$

Following this listing of the PiCs, there are references to perpendicular, parallel and singular SiCs, none of which are present in the Einstein–Cartan theory. We will return to these in Section III C, but for now we note that they follow from the imposition of multipliers in (7), and their presence is fully understood in [62], just as the presence of the PiCs of the basic Poincaré gauge theory is understood in [58, 60]. This is the intended scope of `DefTheory[]`: to elucidate not only the primary constraints, but all of that part of the constraint structure which is already known from the literature. Every if-constraint identified by `DefTheory[]` is used during the `DefTheory[]` call in the construction of very large internal rule sets `$StrengthPShellToStrengthP03`, `$PiPShellToPiPP03`, `$TheoryCDPiPToCDPiP03` and `$TheoryPiPToPiP03`. These rules are applied – by default – during `ToNesterForm` calls. Most of the if-constraints that can arise, as detailed in [62], are of the form  $\frac{1}{J} A \hat{\pi}_{\hat{u}} + \dots \approx 0$  or  $\frac{1}{J} E \hat{\pi}_{\hat{u}} + \dots \approx 0$ , and so the shell is defined by replacing all possible instances of the momenta in favor of other quantities. There are also the ‘parallel’ SiCs, which deactivate irreps within the canonical parts of the field strength tensors in Eqs. (13a) and (13b) – these are straightforwardly implemented.

Following the listing of the if-constraints, the output above details the structure of the canonical Hamiltonian, which from [62] is written

$$H_C \equiv N H_{\perp} + N^{\alpha} H_{\alpha} - \frac{1}{2} A^{ij} H_{ij} + \partial_{\alpha} \mathcal{D}^{\alpha}. \quad (24)$$

This *Dirac* form [59, 82] is useful because it expresses part of the Hamiltonian as a linear combination of the nonphysical

fields  $N$ ,  $N^{\alpha}$  and  $A^{ij}_0$ . The physical (and canonical) coefficients of these undetermined fields are

$$H_{\perp} \equiv \hat{\pi}_i^{\bar{k}} \mathcal{T}_{\perp \bar{k}}^i + \frac{1}{2} \hat{\pi}_{ij}^{\bar{k}} \mathcal{R}_{\perp \bar{k}}^{ij} - J L_G - n^k D_{\alpha} \pi_k^{\alpha}, \quad (25a)$$

$$H_{\alpha} \equiv \pi_i^{\beta} T_{\alpha \beta}^i + \frac{1}{2} \pi_{ij}^{\beta} R_{\alpha \beta}^{ij} - b_{\alpha}^k D_{\beta} \pi_k^{\beta}, \quad (25b)$$

$$H_{ij} \equiv 2\pi_{[i}^{\alpha} b_{j]\alpha} + D_{\alpha} \pi_{ij}^{\alpha}, \quad (25c)$$

and they form the ‘sure’ secondary FC constraints (sSFCs)

$$H_{\perp} \approx 0, \quad H_{\alpha} \approx 0, \quad H_{ij} \approx 0. \quad (26)$$

We see that these 10 constraints are divided up under  $SO(3)$  to give the final four entries in the output above, with  $H_{ij} = H_{ij}^- + 2n_{[i} H_{\perp j]}$ . The (linearised) values returned for  $H_{\perp}$  and  $H_{\alpha}$  seem sensible in the context of the constraint in (23) – we can see from this that *HiGGS* has begun imposing the PiC shell. What about the angular super-momentum? Let us verify by hand that the answer is correct. The first term in (25c) vanishes on the PiC shell, because *HiGGS* has told us that  $\hat{\pi} \approx \hat{\pi}_{\bar{k}l}^{\wedge} \approx \hat{\pi}_{\perp \bar{k}}^{\sim} \approx \hat{\pi}_{\bar{k}l}^{\sim} \approx 0$ . For the second term, we find from (23) after a few lines that

$$\pi_{kl}^{\beta} \approx -2\hat{\alpha}_0 m_p^2 J n_{[k} h_{l]}^{\beta}, \quad (27)$$

from which we determine

$$\begin{aligned} H_{kl} &\approx -2\hat{\alpha}_0 m_p^2 J \left[ h_{\bar{m}}^{\beta} (\partial_{\alpha} b_{\bar{m}}^{\alpha}) n_{[k} h_{l]}^{\alpha} \right. \\ &\quad \left. + D_{\alpha} (n_{[k} h_{l]}^{\alpha}) \right] \\ &= -2\hat{\alpha}_0 m_p^2 J \left( \tau_{\perp \bar{k}l} + 2n_{[k} \tau_{l]}^{\perp} \right). \end{aligned} \quad (28)$$

These are indeed the  $1^+$  and  $1^-$  parts of the torsion, as *HiGGS* is claiming.

## 2. StudyTheory and Velocity

The greater part of the analysis – i.e. that which is not necessarily encoded in the literature – is requested by means of the `StudyTheory[]` command. We will see in Section II F that this command is just a high-level wrapper for parallelising Poisson matrices and constraint velocities, and also for `DefTheory[]` as discussed in Section II E 1. We will see how to use `StudyTheory[]` for a batch of theories in Section III B. For now we note that having called `DefTheory[]` above, we can pass the option `"Import" → True` to the command

```

In[] := JobsBatch = {"EinsteinCartan", {Alp1 ==
    ↪ 0, Alp2 == 0, Alp3 == 0, Alp4 == 0,
    ↪ Alp5 == 0, Alp6 == 0, Bet1 == 0, Bet2 ==
    ↪ 0, Bet3 == 0, cAlp1 == 0, cAlp2 == 0,
    ↪ cAlp3 == 0, cAlp4 == 0, cAlp5 == 0,
    ↪ cAlp6 == 0, cBet1 == 0, cBet2 == 0,
    ↪ cBet3 == 0}}};
JobsBatch~StudyTheory~("Import" → True);

```

and – since the calculation is expensive – run it through a Wolfram Language package file. The evaluation of the Poisson matrix is fairly straightforward, and made up from calls to `PoissonBracket[]` as discussed in Section IID3. It is better then to turn to `Velocity[]`.

The linearised velocity of (e.g.) some PiC  ${}^B\varphi_{\hat{v}}$  is calculated using the formula

$${}^B\dot{\varphi}_{\hat{v}}(\mathbf{x}_1) = \int d^3x_2 \{ {}^B\varphi_{\hat{v}}(\mathbf{x}_1), \mathcal{H}_T(\mathbf{x}_2) \}, \quad (29)$$

where from (24) we need evaluate only the commutator with  $\mathcal{H}_{\perp}$ , which is re-expressed in [62] on the PiC shell as

$$\begin{aligned} \mathcal{H}_{\perp} \equiv & \frac{J}{64} \sum_A c_A^{\perp} \mu(\hat{\alpha}_A^{\perp\perp})^A \varphi_{\hat{v}}^A \varphi^{\hat{v}} \\ & + \frac{J}{16m_p} \sum_E c_E^{\perp} \mu(\hat{\beta}_E^{\perp\perp})^E \varphi_{\hat{v}}^E \varphi^{\hat{v}} + \frac{1}{2} \hat{\alpha}_0 m_p^2 \underline{\mathcal{R}} \\ & - J \sum_I \left( \hat{\alpha}_I \mathcal{R}^{ij}_{kl} + \bar{\alpha}_I \lambda^{ij}_{kl} \right)^I \hat{\mathcal{P}}_{ij}^{kl} \bar{\mathcal{P}}_{nm}^{\bar{p}\bar{q}} \mathcal{R}^{nm}_{\bar{p}\bar{q}} \quad (30) \\ & - J m_p^2 \sum_M \left( \hat{\beta}_M \mathcal{T}^i_{kl} + \bar{\beta}_M \lambda^i_{kl} \right)^M \hat{\mathcal{P}}_i^{kl} \bar{\mathcal{T}}^n_{\bar{p}\bar{q}} \mathcal{T}^n_{\bar{p}\bar{q}} \\ & - n^k D_{\alpha} \pi_k^{\alpha}. \end{aligned}$$

A particular problem now is that (30) has a *quadratic* structure. If we apply `PoissonBracket[]` to evaluate a velocity for some PiC  $\{ {}^B\varphi_{\hat{v}}, \mathcal{H}_{\perp} \}$ , we will actually be calling `ToBasicForm[]` directly on  $\mathcal{H}_{\perp}$ . This will generally result in a very expensive computation, which we avoid in `Velocity[]` by manually applying the Leibniz rule. Accordingly in the *HiGGS* source there is a collection of prepared formulae for generic expressions such as

$$\begin{aligned} \{ {}^B\varphi_{\hat{v}}, \mathcal{H}_{\perp} \} &= \frac{J}{32} \sum_A c_A^{\perp} \mu(\hat{\alpha}_A^{\perp\perp}) \{ {}^B\varphi_{\hat{v}}, {}^A\varphi_{\hat{u}} \} {}^A\varphi^{\hat{u}} + \dots, \\ &= \frac{1}{32} \sum_A c_A^{\perp} \mu(\hat{\alpha}_A^{\perp\perp}) \left[ J J_{1\hat{v}\hat{u}} {}^A\varphi^{\hat{u}} + \right. \\ &\quad + J_{2\hat{v}\hat{u}}^{\alpha} \partial_{\alpha} (J {}^A\varphi^{\hat{u}}) - \partial_{\alpha} (J J_{3\hat{v}\hat{u}}^{\alpha A} {}^A\varphi^{\hat{u}}) \\ &\quad \left. - \partial_{\alpha} (J_{4\hat{v}\hat{u}}^{\alpha\beta} \partial_{\beta} (J {}^A\varphi^{\hat{u}})) \right] + \dots, \quad (31) \end{aligned}$$

which collectively act as a template for each velocity. For each SO(3) index  $A$  shown in the sum in the second equality in (31), we need the four portions of a single Poisson bracket, as given in (18) and returned as a `List` by a call to `PoissonBracket[]`. Accordingly, the whole of (30) is broken into *blocks*, each of four terms, which follow from a single bracket, and these are evaluated in parallel as discussed in Section IIF.

As we mentioned in Section IID3, the formula (18) is not explicitly covariant, and hence the appearance in (31) of partial derivatives. Ultimately, a call to `ToNesterForm[]` is needed to restore explicit covariance to the quadratic expression (31). This setup is not so good: the use of block formulae restricts us to calculating velocities over the given  $\mathcal{H}_{\perp}$  – it is

also susceptible to human error, and concludes with an expensive simplification process. We propose that any future iterations of *HiGGS* implement velocities based two extensions to `PoissonBracket[]`:

1. The operands should be expressed as sums of products of covariant quantities, and the Leibniz rule be used to distribute the operation.
2. The operation should return the covariant form given in (19), so that covariance is maintained at all steps.

An improvement of `PoissonBracket[]` along these lines should be straightforward to implement, and would also be equally emenable to parallelisation: we defer its development to future work.

### 3. ViewTheory

The final module `ViewTheory[]` is used to produce a human-readable summary of the results from `StudyTheory[]`. Since we already have a summary of the literature constraint structure from the `DefTheory[]` call in Section IIE1, we pass for brevity our earlier theory name with the option "Literature"  $\rightarrow$  `False`, producing

```
In[] := ViewTheory["EinsteinCartanVelocities", "
  ↳ Literature" → False];
** DefTheory: Incorporating the binary at svy
  ↳ /EinsteinCartanVelocities.thr.mx
```

$$\{ {}^0\varphi_{\hat{b}}, {}^0\varphi_{\hat{a}} \} \approx \left\{ -\frac{6\hat{\alpha}_0}{\mathcal{I}}, 0, 0 \right\}$$

$$\{ {}^1\varphi_{\hat{b}ij}, {}^1\varphi_{\hat{a}lm} \} \approx \left\{ \frac{\hat{\alpha}_0 \eta^l{}_{im} \eta^l{}_{jl}}{2\mathcal{I}} - \frac{\hat{\alpha}_0 \eta^l{}_{il} \eta^l{}_{jm}}{2\mathcal{I}}, 0, 0 \right\}$$

$$\{ {}^1\varphi_{\hat{b}i}, {}^1\varphi_{\hat{a}l} \} \approx \left\{ -\frac{2\hat{\alpha}_0 \eta^l{}_{il}}{\mathcal{I}}, 0, 0 \right\}$$

$$\{ {}^2\varphi_{\hat{b}ij}, {}^2\varphi_{\hat{a}lm} \} \approx \left\{ \frac{\hat{\alpha}_0 \eta^l{}_{im} \eta^l{}_{jl}}{2\mathcal{I}} + \frac{\hat{\alpha}_0 \eta^l{}_{il} \eta^l{}_{jm}}{2\mathcal{I}} - \frac{\hat{\alpha}_0 \eta^l{}_{ij} \eta^l{}_{lm}}{3\mathcal{I}}, 0, 0 \right\}$$

$$\frac{d}{dt} {}^0\varphi_{\hat{b}} \approx -\hat{\alpha}_0 \mathcal{N} {}^0\mathcal{R}^l{}_l$$

$$\frac{d}{dt} {}^1\varphi_{\hat{b}ij} \approx -\hat{\alpha}_0 \mathcal{N} {}^1\mathcal{R}^l{}_{ij}$$

$$\frac{d}{dt} {}^2\varphi_{\hat{b}ij} \approx \hat{\alpha}_0 \mathcal{N} {}^2\mathcal{R}^l{}_{ij}$$

$$\frac{d}{dt} {}^0\varphi_{\hat{a}} \approx -\hat{\alpha}_0 \mathcal{N} {}^0\mathcal{T}^l{}_l$$

$$\frac{d}{dt} {}^1\varphi_{\hat{a}i} \approx -\hat{\alpha}_0 \mathcal{N} {}^1\mathcal{T}^l{}_i$$

$$\frac{d}{dt} \phi_{\mathcal{A}ijk} \approx \frac{2}{3} \hat{\alpha}_0 \mathcal{N} \phi_{\mathcal{T}ijk} + \frac{1}{3} \hat{\alpha}_0 \mathcal{N} \phi_{\mathcal{T}ikj} - \frac{1}{3} \hat{\alpha}_0 \mathcal{N} \phi_{\mathcal{T}jki}$$

We see how the results are stored in the theory binary at `svy/EinsteinCartan.thr.mx`. There are *four* nonvanishing Poisson brackets, all between the translational and rotational pairs of the same  $\text{SO}(3)$  irrep,  $\varphi$  vs  $\varphi_\perp$ ,  $\hat{\varphi}_{\bar{ij}}$  vs  $\varphi_\perp$ ,  $\varphi_{\perp\bar{i}}$  vs  $\hat{\varphi}_{\bar{i}}$ , and  $\hat{\varphi}_{\bar{ij}}$  vs  $\hat{\varphi}_{\perp\bar{j}}$ . These are the *conjugate pairs*, whose commutators are proportional to mass parameters in the theory [58] – here mediated by the Einstein–Hilbert term  $\hat{\alpha}_0$ . The involved PiCs are second class (SC).

What about the remaining PiCs  ${}^P\varphi$  and  ${}^T\varphi$ ? These  $0^-$  and  $2^-$  irreps<sup>7</sup> are not represented in the translational sector. Previous analyses tell us that their relevant commutators will be with their own secondaries [58]. Accordingly, we turn to the velocities, which are also provided in the above output. Unlike the if-constraints, the linearised sSFC Hamiltonian constraints are not automatically implemented when "ToShell" → True is passed: we therefore note that since  $\underline{R}^b \approx \mathcal{T}_{\perp\bar{i}}^b \approx 0$ , from the output of `DefTheory[]` above, both rotational and translational (Hamiltonian) multipliers for the  $0^+$  and  $1^-$  sectors will also vanish at linear order. The same is true of the translational  $1^+$  and  $2^+$  Hamiltonian multipliers, but the rotational counterparts will be linearly proportional to  $\underline{R}_{[\bar{ij}]}$  and  $\underline{R}_{\langle\bar{ij}\rangle}$ . Returning to the ‘lonely’  $0^+$  and  $2^-$  sectors, we can go right ahead and calculate the expected brackets mentioned above. We first find

```
In[] := PoissonBracket[PhiA0m[], TP0m[], "
  ↳ ToShell" → True];
Out[] =
```

$$\{\phi_{\mathcal{A}}, \phi_{\mathcal{T}}\} \approx \left\{ -\frac{24}{\mathcal{T}}, 0, 0, 0 \right\}$$

As expected, this commutator survives at linear order: one could determine also the  $0^-$  multiplier, but it is enough to notice that both  ${}^P\varphi$  and its linearised secondary  ${}^P\chi^b \equiv {}^P\mathcal{T}^b \approx 0$  also become SC. Moving on, we find

```
In[] := PoissonBracket[PhiA2m[-i, -j, -k], TP2m
  ↳ [-l, -m, -n], "ToShell" → True];
Out[] =
```

$$\begin{aligned} \{\phi_{\mathcal{A}ijk}, \phi_{\mathcal{T}lmn}\} \approx & \\ & \left\{ -\frac{3\eta_{il}\eta_{jm}\eta_{kl}}{16\mathcal{T}} + \frac{3\eta_{im}\eta_{jn}\eta_{kl}}{16\mathcal{T}} + \frac{3\eta_{in}\eta_{jl}\eta_{km}}{16\mathcal{T}} - \right. \\ & \left. \frac{3\eta_{il}\eta_{jn}\eta_{km}}{16\mathcal{T}} + \frac{3\eta_{im}\eta_{jl}\eta_{kn}}{8\mathcal{T}} - \right. \\ & \left. \frac{3\eta_{il}\eta_{jm}\eta_{kn}}{8\mathcal{T}} - \frac{9\eta_{im}\eta_{jk}\eta_{ln}}{32\mathcal{T}} + \frac{9\eta_{ik}\eta_{jm}\eta_{ln}}{32\mathcal{T}} + \right. \\ & \left. \frac{9\eta_{il}\eta_{jk}\eta_{mn}}{32\mathcal{T}} - \frac{9\eta_{ik}\eta_{jl}\eta_{mn}}{32\mathcal{T}}, 0, 0, 0 \right\} \end{aligned}$$

The same, then, is true of  ${}^T\varphi_{ijk}$  and its linearised secondary

$${}^T\chi^b_{ijk} \equiv {}^T\mathcal{T}^b_{ijk} \approx 0, \text{ and so the algorithm terminates.}$$

What can *HiGGS* tell us about the physics of Einstein–Cartan theory? The PGT contains  $2 \times (24 + 16) = 80$  naïve d.o.fs in its gauge fields. The non-physicality of the lapse and shift (the Poincaré gauge symmetry), remove  $2 \times 10$  d.o.fs through the sure, primary (sPFC) constraints; a further 20 d.o.f are removed via the sSFCs. We further learn from *HiGGS* about the number and class (all SC) of the if-constraints. The final d.o.f count is thus

$$\begin{aligned} & \frac{1}{2}(80 - 2 \times 10 - 2 \times 10 \\ & - (1 + 1 + 3 + 3 + 3 + 3 + 5 + 5) \\ & - (1 + 1 + 5 + 5)) = 2, \end{aligned} \quad (32)$$

i.e. the massless graviton.

## F. Parallelisation and HPC

The current version of *HiGGS* is not only designed for the local or desktop operations conducted in Sections II A to II E. The intended use is the evaluation of quantities useful to the Hamiltonian analysis, in an environment where multiple parallel kernels are available. As mentioned in Section I, there are some generic features of the Hamiltonian analysis which lend themselves very well to parallelisation.

Naïvely, we notice that once the primary constraints have been identified, the resulting constraint chains (i.e. the recursive consistency conditions of each primary) can notionally be evaluated in parallel. This seems a natural route down which to parallelise, but in reality chains might seem to continue indefinitely in a kernel with knowledge only of the primary constraint shell. Velocities are the most expensive quantities, and so it seems prudent to pause after each is calculated, to see if the corresponding acceleration is strictly necessary. However, when we parallelise over chains we also find that the velocities, accelerations and jerks rapidly desynchronise: this is to be expected given the varying complexities of the  $\text{SO}(3)$  irreps which underlie each chain. Accordingly, pre-velocity checks would require e.g. a centralised knowledge of the complete shell to be kept on the master kernel, to which sub-kernels could refer as needed. We could even imagine a setup where

<sup>7</sup> Note that the velocity of  ${}^T\varphi_{ijk}$  is not simplified: *HiGGS* inherits this inability from *xAct*, as no efficient algorithm is known for simplifying multi-term symmetries [83] such as that of the  $2^-$  sector.



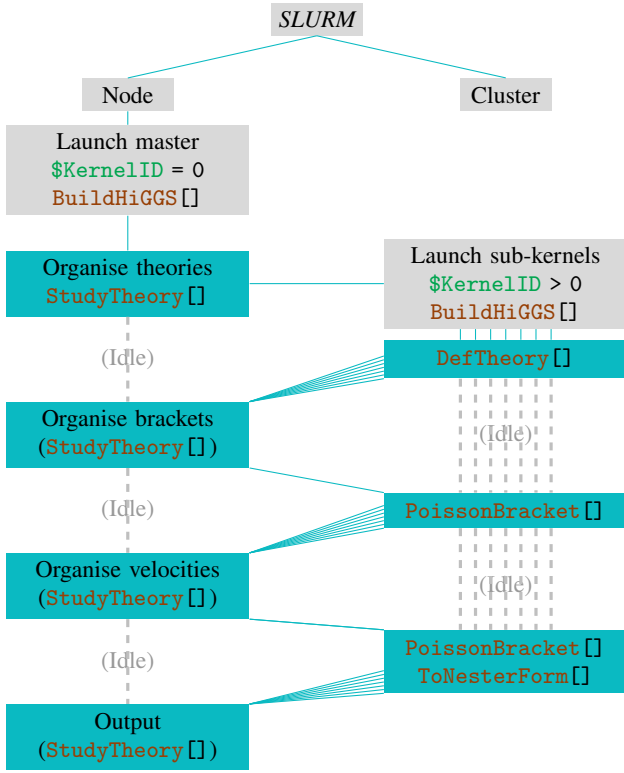


FIG. 2. The parallelisation in *HiGGS* is not sophisticated, but it allows for pragmatic scaling of the analysis to surveys. A large batch of theories can be broken via SBATCH directives over computation nodes, with a reasonable group size of one theory per core per node. Each node is controlled by a master kernel, which coordinates its portion of the survey through a single call to `StudyTheory []`. Three phases of work are then delegated to the sub-kernels. First, the constraint shell is partially reconstructed from the literature knowledge of the theory. The commutators between all constraints of all theories are then evaluated in parallel. Finally, the velocities of the constraints are decomposed into their constituent Poisson brackets: these are evaluated and simplified in parallel before returning to the master kernel for synthesis of the results.

one sub-kernel is interrupted in mid-evaluation on the basis of a report from another.

While it is tempting to develop something sophisticated along these lines, the route turns out not to be practical. In practice, the number of chains is typically very few per theory (and strictly fewer than 11 in the case of the Poincaré gauge theory without multipliers). The desynchronisation effect is then so severe that most kernels would sit idle whilst waiting for the highest-spin chain to complete. More importantly, the question of whether chain (a) need continue based on a new constraint from chain (b), is not a trivial problem in computer algebra. In the case of the PiCs, each constraint has as one term a unique part of the momentum: as mentioned in Section II E 1 this lends itself to a replacement rule which reliably implements the PiC shell, but in more general cases it is less clear how to proceed.

Ultimately, the experience of [1, 43] suggests that evaluation of the accelerations is not usually necessary anyway. Once

the brackets between the PiCs are known, it is often possible to determine from a visual inspection whether any chains are worth continuing. Velocities are expensive because of the structure of the Hamiltonian:  $\mathcal{H}_\perp$  is a sum of terms which are quadratic in nontrivial covariant quantities such as the  ${}^A\varphi_b$  and  ${}^A\tilde{\mathcal{P}}_{\dot{u}nm}{}^{\bar{p}q}\mathcal{R}^{nm}{}_{\bar{p}q}$ . Calculation of an overall bracket with  $\mathcal{H}_\perp$  thus requires very many sub-brackets.

The quadratic structure of  $\mathcal{H}_\perp$  underlies other inefficiencies in the current *HiGGS* implementation, affecting all terms except for the final  $-n^k D_\alpha \pi_k^\alpha$ . As discussed already in Section II E 2, the way in which *HiGGS* calculates velocities is based on (17) rather than (18). As a result, the calculation of a bracket while finding the velocity of  $\mathcal{A}_u(x_2)$ , involving a quadratic term  $\mathcal{B}_b(x_2)\mathcal{C}^b(x_2)$  in  $\mathcal{H}_\perp$  produces – in the first instance – a non-covariant *quadratic* expression which must be covariantised internally by expensive calls to `ToNesterForm []`. This could be readily improvable with development, and in many cases we find that the extra penalty incurred by the quadratic covariantisation remains comparable to the cost of the prerequisite brackets.

Since brackets turn out to be the fundamental unit of the analysis, it is over brackets that *HiGGS* is parallelised. Brackets are extremely cumbersome for humans to evaluate and covariantise: these tasks are relatively easy for computers, now that the infrastructure of Sections II A to II E is in place. On the other hand, shells made from constraints whose format is arbitrary are less easy for computers to use. Once covariantised however, such constraints often comprise straightforward tensor equations which humans can readily manipulate. This suggests a pragmatical division of labour in which *HiGGS* returns an organised structure of covariantised brackets – formerly the primary Poisson matrix (PPM) [1, 12, 13] – and velocities for human inspection. If velocities are expensive because they require multiple brackets, we can break them up accordingly, along with the quadratic covariantisation steps: in this way we can reduce chain desynchronisation.

The final structure of a *HiGGS* survey is as follows;

1. A list of theories is passed to `StudyTheory []` in the master kernel. This function runs `DefTheory []`, for each theory, in its own parallel sub-kernel. The literature knowledge of the constraint shell of each theory is cached as a binary.
2. Within the same `StudyTheory []` call, the master kernel imports the shells and prepares a combined `List` of brackets which need to be evaluated for the Poisson matrices. The `List` elements are delegated to all available parallel sub-kernels – as many of which are launched as necessary. Brackets are transferred to the next available sub-kernel using the Wolfram Symbolic Transfer Protocol (*WSTP*). The *WSTP* overhead is marginal, since each sub-kernel only needs to transfer data after its `PoissonBracket []` call<sup>8</sup>.

<sup>8</sup> The use of `Print []` for debugging and development purposes means that

3. Within the same `StudyTheory[]` call, the master kernel decomposes each linearised velocity into blocks dependent on a single bracket. For the velocity of some  $\text{PiC } {}^B\varphi_{\hat{v}}$  the blocks are;

$$\begin{aligned} & \{ {}^B\varphi_{\hat{v}}, {}^A\varphi_{\hat{u}} \}, \quad \forall A : v(\hat{\alpha}_A^{\perp\perp}) = 1, \\ & \{ {}^B\varphi_{\hat{v}}, {}^E\varphi_{\hat{u}} \}, \quad \forall E : v(\hat{\beta}_E^{\perp\perp}) = 1, \\ & \{ {}^B\varphi_{\hat{v}}, \mathcal{R}_{ijkl} \}, \quad \{ {}^B\varphi_{\hat{v}}, \mathcal{T}_{ikl} \}, \\ & \{ {}^B\varphi_{\hat{v}}, J \}, \quad \{ {}^B\varphi_{\hat{v}}, -n^k D_\alpha \pi_k^\alpha \}. \end{aligned} \quad (33)$$

The blocks are delegated to the sub-kernels again using *WSTP*. After the `PoissonBracket[]` call, each sub-kernel pre-processes its block using `ToNesterForm[]` before passing the result back to the master kernel.

4. Within the same `StudyTheory[]` call the blocks are recombined into velocities, and all results are cached in a final binary.

We illustrate this process in Fig. 2. In the parallel *HiGGS* environment of any node, one has at any one time a collection of theories, each of which has associated with it a complex and growing structure of constraints and commutators. This environment suggests an object-oriented approach. While *Mathematica* allows for object-oriented programming using e.g. `Association[]` (see also [84]), the scope of *HiGGS* is simple enough that we can retain a procedural approach. In fact, the shell structure of any one theory is referenced fairly infrequently, matching the rate of `PoissonBracket[]` calls (see Fig. 1 for an illustration of this). This means that while *WSTP* is necessary for the scheduling of evaluations, the information needed to switch between theories within a sub-kernel can be sourced by using relay system of binary files. Consequently, very minimal use is made by *HiGGS* of `DistributeDefinitions[]` to transfer theory-specific data between kernels. In a more serious implementation of the Dirac–Bergmann algorithm, we would envisage more extensive use of *WSTP* and the object-oriented approach.

How efficient is the above approach? Pending an implementation of the Leibniz rule, the efficacy of Fig. 2 as it applies to the whole constraint algorithm is hard to gauge. The full run is only implemented in Section III B for a handful of previously studied theories (without multipliers). Averaged over those cases, and for the non-fundamental reasons outlined above, a minority of the `Velocity[]` calls contain serial tasks which turn out to dominate the workload. The truly parallel fraction of the workload (including the initial evaluation of all constraint brackets) is then as low as  $p \sim 10^{-2}$ . Modelling the whole implementation in Fig. 2 as an Amdahl task [85], the  $n$ -core speedup

$$S(n) = \frac{1}{1 - p + \frac{p}{n}}, \quad (34)$$

this is not strictly true, and there is also an overhead from the system timing wrappers which we use to produce plots such as Fig. 1. These features are not actually needed when running a survey.

TABLE I. Approximate benchmarks for parallel brackets. The spin-parity  $1^+$  theory in (37) is augmented with all configurations of the multipliers  $\{\tilde{\alpha}_3, \tilde{\alpha}_4, \tilde{\alpha}_6, \tilde{\beta}_1, \tilde{\beta}_2, \tilde{\beta}_3\}$  – see (7) – and all the Poisson brackets are obtained. For survey structure and specifications of the Peta4 cluster, see Section III C.

Cores $n$	Wallclock time/s	Speedup $S(n)$	Efficiency $5 \times S(n)/n$
32	550 $\pm$ 100	3.0 $\pm$ 1.2	0.46 $\pm$ 0.19
28	565 $\pm$ 68	2.88 $\pm$ 0.97	0.51 $\pm$ 0.17
24	590 $\pm$ 100	2.7 $\pm$ 1.1	0.57 $\pm$ 0.22
20	640 $\pm$ 120	2.5 $\pm$ 1.0	0.63 $\pm$ 0.26
16	720 $\pm$ 160	2.3 $\pm$ 1.0	0.71 $\pm$ 0.31
12	835 $\pm$ 81	1.95 $\pm$ 0.61	0.81 $\pm$ 0.26
8	1110 $\pm$ 310	1.46 $\pm$ 0.73	0.91 $\pm$ 0.46

would appear very limited. We get a fairer understanding from the ‘calibration’ survey set out in Section III C. In Table I we run a portion of this survey – spanning only  $2^6 = 64$  modified gravity theories – on a cluster with a variable number of cores per node. We take this to be a task of ‘reasonable size’ when using *HiGGS* to learn about the canonical structure of a theory. Since the survey does not contain `Velocity[]` calls,  $S(n)$  will be more sensitive to the relevant trade-off between *WSTP* overhead and bracket evaluation. There are relatively few brackets, and this portion of the survey is now dominated by the (fast but serial) `BuildHiGGS[]` and `DefTheory[]` calls: the benchmarking is thus quick to run, but it still gives an impression of low efficiency. The normalising time is based on five cores per node: roughly equal to the number of theories. We expect  $S(n)$  to be concave and sublinear, but we do not see it peak in our use-case. No Hamiltonian analysis tools have previously been made that could provide a more meaningful benchmark.

### III. EXAMPLES

Having introduced the implementation in Section II, we now give some basic examples. These will include a reproduction of the analysis in [1, 43] and a simple HPC survey which extends those same ‘minimal’ theories with the use of multipliers.

#### A. Preparing a science session

To prepare a science session, we begin by loading the package into a fresh *Mathematica* kernel

```
In[] := <<xAct'HiGGS';
```

If the *HiGGS* sources have been correctly placed with respect to the *xAct* directory tree, a copyright greeter should be displayed, indicating that the context `xAct'HiGGS'` and its dependancy contexts `xAct'xTensor'`, `xAct'xPerm'`, `xAct'xCore'`, `xAct'xTras'` have been loaded. However, loading the package does not yet introduce the physics: the kernel is still in a fresh *xAct* session, without even a differential manifold. To construct the very many physical definitions

needed for science we must *build* the package using the command

```
In[]:= BuildHiGGS[];
```

This begins an execution in `xAct` 'HiGGS' Private' of `xAct/HiGGS/HiGGS_sources.m`, most of which is taken up with symbol definitions, and in particular calls to `DefTensor[]` and `MakeRule[]`. To shorten the process, the most expensive definitions (including those for  $SO^+(1,3)$  projection operators) have been stored in binary files under `xAct/HiGGS/bin/build/*.mx` — those binaries are imported at this time<sup>9</sup>. In an active front end, the output cells displaying the progress of this build process are periodically deleted, and should finally be replaced by the following message (adjusting for memory)

```
Out[]:= ** BuildHiGGS: The HiGGS environment
        ↳ is now ready to use and is occupying
        ↳ 63184600 bytes in RAM.
```

The context `xAct` 'HiGGS' should now be populated with many physical quantities, and as a result the session alone is very memory-intensive. We can view some of these quantities through the `xAct` variables `$Tensors` and `$ConstantSymbols`. From this point we may perform the various operations in Section II, or proceed directly to science.

### B. Yo-Nester unit tests

As a first application, and to verify that *HiGGS* has been correctly calibrated, we recapitulate the historical analysis in [1, 43].

The theory in which the  $0^+$  mode is active, as considered in [43], is given by imposing the following constraints on (7)

$$\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_3 = \hat{\alpha}_4 = \hat{\alpha}_5 = 2\hat{\beta}_1 + \hat{\beta}_2 = \hat{\beta}_1 + 2\hat{\beta}_3 = 0, \quad (35)$$

$$\hat{\alpha}_0 \neq 0, \quad \hat{\alpha}_6 \neq 0.$$

All the constraint couplings  $\{\tilde{\alpha}_I\}$  and  $\{\tilde{\beta}_M\}$  are of course assumed to vanish, since they are only recently proposed. No other 'simple' linear identities are assumed among the couplings, and this avoids collision with other theories. The other theory in [43], in which the  $0^-$  mode is instead allowed to propagate, is defined by

$$\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_4 = \hat{\alpha}_5 = \hat{\alpha}_6 = 2\hat{\beta}_1 + \hat{\beta}_2 = \hat{\beta}_1 + 2\hat{\beta}_3 = 0, \quad (36)$$

$$\hat{\alpha}_0 \neq 0, \quad \hat{\alpha}_3 \neq 0.$$

In [1] the analysis was extended to 'higher-spin' modes. The theory with only the  $1^+$  mode propagating is

$$\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_3 = \hat{\alpha}_4 = \hat{\alpha}_6 = \hat{\beta}_1 = \hat{\beta}_2 = 0, \quad (37)$$

$$\hat{\alpha}_0 \neq 0, \quad \hat{\alpha}_5 \neq 0, \quad \hat{\beta}_3 \neq 0.$$

The theory with only the  $1^-$  mode propagating is

$$\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_3 = \hat{\alpha}_4 = \hat{\alpha}_6 = \hat{\beta}_1 = \hat{\beta}_3 = 0, \quad (38)$$

$$\hat{\alpha}_0 \neq 0, \quad \hat{\alpha}_5 \neq 0, \quad \hat{\beta}_2 \neq 0,$$

and the theory with only the  $2^-$  mode propagating is

$$\hat{\alpha}_2 = \hat{\alpha}_3 = \hat{\alpha}_4 = \hat{\alpha}_5 = \hat{\alpha}_6 = \hat{\beta}_1 = \hat{\beta}_2 = \hat{\beta}_3 = 0, \quad (39)$$

$$\hat{\alpha}_0 \neq 0, \quad \hat{\alpha}_1 \neq 0.$$

There is also tested in [1] a pair of more complex theories in which the  $0^-$  and  $2^-$  modes are both active at the same time. These are given respectively by

$$\hat{\alpha}_1 = \hat{\alpha}_3 = \hat{\alpha}_4 = \hat{\alpha}_5 = \hat{\alpha}_6 = \hat{\beta}_1 = \hat{\beta}_2 = \hat{\beta}_3 = 0, \quad (40)$$

$$\hat{\alpha}_0 \neq 0, \quad \hat{\alpha}_2 < 0,$$

and

$$\hat{\alpha}_2 = \hat{\alpha}_4 = \hat{\alpha}_5 = \hat{\alpha}_6 = \hat{\beta}_1 = \hat{\beta}_2 = \hat{\beta}_3 = 0, \quad (41)$$

$$\hat{\alpha}_0 \neq 0.$$

The seven configurations Eqs. (35) to (41) can of course be processed in parallel. We set up a list called `JobsBatch`, which stores the coupling conditions and assigns a string label to each theory

```
In[]:= JobsBatch={};
In[]:= JobsBatch~AppendTo~{"spin_0p", {Alp1 ==
    ↳ 0, Alp2 == 0, Alp3 == 0, Alp4 == 0, Alp5
    ↳ == 0, 2Bet1 + Bet2 == 0, Bet1 + 2 Bet3
    ↳ == 0, cAlp1 == 0, cAlp2 == 0, cAlp3 ==
    ↳ 0, cAlp4 == 0, cAlp5 == 0, cAlp6 == 0,
    ↳ cBet1 == 0, cBet2 == 0, cBet3 == 0}};
```

Note that definition in terms of equalities is sufficient: the many strict inequations fixing the absence of other special linear conditions are always assumed by *HiGGS* to be implicit, and signs associated with other inequalities do not affect the constraint structure we wish to probe (though they may well affect the unitarity).

Assuming we have similarly entered the parameters for all theories above, the major undertaking of evaluating all seven PPMs can be initiated with the command

```
In[]:= JobsBatch~StudyTheory~("Import"→True);
```

The option `"Import"→True` assumes that `DefTheory[]` has already been run on all seven cases. In that case binaries such as `svy/spin_0p.thr.mx` etc., will have been created to store a summary of our prior knowledge of each constraint chain.

Even in the case of the small batch in `JobsBatch`, the calculation is barely viable using sub-HPC resources. We perform the run on a dedicated data processing server with 129 GB memory and eight available 2.90 GHz Intel® Xeon® E5-2690 0 CPU cores, corresponding to a maximum of eight parallel *Mathematica* kernels. This computation lasts  $\sim 14$  h, but almost all of this time is spent on the inefficient evaluation of velocities.

<sup>9</sup> Note that the *HiGGS* binaries were compiled on a 64-bit machine: they can, if needed, be recompiled on alternative architecture via minor changes to the source as indicated in `xAct/HiGGS/HiGGS.nb`.

Once the calculations of the `StudyTheory[]` call are complete, we recall from Section II E 3 that the results are displayed in human-readable form via the `ViewTheory[]` command. We will not provide full analysis here, but focus on the exemplar  $1^+$  case. Skipping the breakdown of PiCs and neglecting the velocities, all the nonlinear brackets are

```
In[]:= ViewTheory["simple__spin_1p", "
  ↳ Literature" → False, "Velocities" →
  ↳ False];
** DefTheory: Incorporating the binary at svy
  ↳ /simple__spin_1p.thr.mx
```

$$\{\phi_b, \phi_{\mathcal{A}}\} \approx \left\{ -\frac{6\hat{a}_0}{\mathcal{T}}, 0, 0 \right\}$$

$$\{\phi_{bi}, \phi_{bj}\} \approx \left\{ \frac{2\hat{v}_{bij}}{\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bi}, \phi_{\mathcal{A}}\} \approx \left\{ -\frac{\hat{v}_{\mathcal{A}i}}{\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bi}, \phi_{\mathcal{A}}\} \approx \left\{ -\frac{2\epsilon^{\mathcal{A}}_{ib1}c1\hat{v}_{\mathcal{A}}^{b1c1}}{\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bi}, \phi_{\mathcal{A}lmn}\} \approx \left\{ -\frac{\eta^{lm}\hat{v}_{\mathcal{A}i}}{6\mathcal{T}^2} + \frac{\eta^{im}\hat{v}_{\mathcal{A}l}}{4\mathcal{T}^2} + \frac{\eta^{il}\hat{v}_{\mathcal{A}m}}{4\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bi}, \phi_{\mathcal{A}lmn}\} \approx \left\{ \frac{3\eta^{mn}\hat{v}_{\mathcal{A}il}}{8\mathcal{T}^2} - \frac{3\eta^{ln}\hat{v}_{\mathcal{A}im}}{8\mathcal{T}^2} + \frac{\eta^{in}\hat{v}_{\mathcal{A}lm}}{2\mathcal{T}^2} + \frac{\eta^{im}\hat{v}_{\mathcal{A}ln}}{4\mathcal{T}^2} - \frac{\eta^{il}\hat{v}_{\mathcal{A}mn}}{4\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bij}, \phi_{bim}\} \approx \left\{ \frac{\eta^{jm}\hat{v}_{bil}}{2\mathcal{T}^2} + \frac{\eta^{jl}\hat{v}_{bim}}{2\mathcal{T}^2} + \frac{\eta^{im}\hat{v}_{bjl}}{2\mathcal{T}^2} + \frac{\eta^{il}\hat{v}_{bjm}}{2\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bij}, \phi_{\mathcal{A}lm}\} \approx \left\{ \frac{\hat{a}_0\eta^{im}\eta^{jl}}{2\mathcal{T}} + \frac{\hat{a}_0\eta^{il}\eta^{jm}}{2\mathcal{T}} - \frac{\hat{a}_0\eta^{ij}\eta^{lm}}{3\mathcal{T}} + \frac{\eta^{jm}\hat{v}_{\mathcal{A}il}}{4\mathcal{T}^2} + \frac{\eta^{jl}\hat{v}_{\mathcal{A}im}}{4\mathcal{T}^2} + \frac{\eta^{im}\hat{v}_{\mathcal{A}jl}}{4\mathcal{T}^2} + \frac{\eta^{il}\hat{v}_{\mathcal{A}jm}}{4\mathcal{T}^2}, 0, 0 \right\}$$

$$\{\phi_{bij}, \phi_{\mathcal{A}lmn}\} \approx \left\{ -\frac{3\eta^{jm}\eta^{ln}\hat{v}_{\mathcal{A}i}}{32\mathcal{T}^2} + \frac{3\eta^{jl}\eta^{mn}\hat{v}_{\mathcal{A}i}}{32\mathcal{T}^2} - \frac{3\eta^{im}\eta^{ln}\hat{v}_{\mathcal{A}j}}{32\mathcal{T}^2} + \frac{3\eta^{il}\eta^{mn}\hat{v}_{\mathcal{A}j}}{32\mathcal{T}^2} + \frac{3\eta^{in}\eta^{jm}\hat{v}_{\mathcal{A}l}}{16\mathcal{T}^2} + \frac{3\eta^{im}\eta^{jn}\hat{v}_{\mathcal{A}l}}{16\mathcal{T}^2} - \frac{3\eta^{ij}\eta^{mn}\hat{v}_{\mathcal{A}l}}{16\mathcal{T}^2} - \frac{3\eta^{in}\eta^{jl}\hat{v}_{\mathcal{A}m}}{16\mathcal{T}^2} + \frac{3\eta^{il}\eta^{jn}\hat{v}_{\mathcal{A}m}}{16\mathcal{T}^2} + \frac{3\eta^{ij}\eta^{ln}\hat{v}_{\mathcal{A}m}}{16\mathcal{T}^2}, 0, 0 \right\}$$

This information (which by itself is obtained within the first minutes of the run) encodes the whole nonlinear primary Poisson matrix (PPM) of the theory: it may be compared with the expressions carefully obtained in [1]. We see that the brackets  $\{\phi_{\perp i}, \phi_{\perp j}\}$ ,  $\{\phi_{\perp i}, \phi_{\perp}\}$ ,  $\{\phi_{\perp i}, {}^P\phi\}$ ,  $\{\phi_{\perp i}, \tilde{\phi}_{\perp lm}\}$ ,  $\{\phi_{\perp i}, {}^T\phi_{lmn}\}$ ,  $\{\tilde{\phi}_{ij}, \tilde{\phi}_{lm}\}$  and  $\{\tilde{\phi}_{ij}, {}^T\phi_{lmn}\}$  are *strictly* nonlinear: they vanish in the linear theory for which the  $1^+$  mode is moving. The effect of this discontinuity in the constraint structure, when moving from the linear to the nonlinear theory, is well described in [1]. In short, a counting analogous to that performed in (32) shows that the three extra d.o.f of the massive  $1^-$  mode are nonlinearly activated: *HiGGS* thus tells us that a vector torsion mode is *strongly coupled*.

The results of the remaining cases in *JobsBatch* are provided in the supplemental materials [64]. We find that, up to terms which vanish due to multi-term symmetries (i.e. which cannot be eliminated in the *xAct* architecture), the nonlinear brackets agree with those found in [1, 43]. There is a possible exception in the case of the general  $\{\hat{\phi}_{ij}, \phi_{\perp}\}$ , which we find to be ‘surfacial’ in ways illustrated already by the bracket in Section II D 3. This subtlety appears only to touch the scalar mode results in [43]: it seems worthwhile to investigate – in future work – if and how this might affect the final d.o.f counting. For brevity we will omit the velocities here<sup>10</sup>, since even the covariantised, linear expressions can be very cumbersome. Examples are provided in Appendix A for the minimal  $1^+$  case above: these illustrate how gradients of the field strengths can arise as the algorithm progresses, necessitating a future extension of *HiGGS* to the second-order Euler–Lagrange formalism.

### C. Simple HPC survey

In this section we further extend our knowledge of the theories examined in Section III B, by running *HiGGS* on ba-

<sup>10</sup> The few velocities which are known from [43] corroborate our results, however we note that `ToNesterForm[]` actually *fails* to fully covariantise the velocity of  $\phi_{\perp i}$  in the simple  $0^-$  case. This ‘bug’ is not known to occur elsewhere, and certainly not during the evaluation of nonlinear brackets: the main feature of *HiGGS*. We suggest that a fully nonlinear ‘Leibniz rule’ implementation of velocities, as recommended above, would greatly reduce the risk of such effects by demanding less of `ToNesterForm[]` in each call.



sic HPC resources. Our starting point will be the ‘minimal’ Poincaré gauge theories set out in Eqs. (35) to (41). Of these, we know that (35) and (36) are the traditional cases in which massive  $0^+$  and  $0^-$  scalars propagate safely alongside the usual  $2^+$  graviton. Of the cases with active higher-spin modes, we know that (40) and (41) propagate both  $0^-$  and  $2^-$  modes, and may be safe.

The minimal ‘problem’ cases are Eqs. (37) to (39). These are ostensibly simple linearised theories which propagate extra  $1^+$ ,  $1^-$  and  $2^-$  modes respectively, but which appear to suffer from mode activation in the fully nonlinear regime. This mode activation is identified by counting the canonical degrees of freedom in the Hamiltonian analysis: its association with a  $J^P$  sector is not shown explicitly, but inferred by examining the unconstrained PiC functions in each case. These are as follows;

- In the minimal  $1^+$  theory (37),  $\hat{\varphi}_{ij}$ ,  $\hat{\varphi}_{\perp ij}$  and  $\vec{\varphi}_i$  are unconstrained, the  $1^-$  mode is thought to be activated.
- In the minimal  $1^-$  theory (38),  $\varphi$ ,  $\varphi_{\perp i}$ ,  $\hat{\varphi}_{\perp ij}$  and  $\vec{\varphi}_i$  are unconstrained, the  $1^+$  mode is thought to be activated.
- In the minimal  $2^-$  theory (39),  $\tilde{\varphi}_{\perp ij}$  and  ${}^T\varphi_{ijk}$  are unconstrained, the  $2^+$  mode is thought to be activated.

We outlined in [62] a potential avenue for preventing mode activation by means of the multiplier fields set out in (7). For any conventional Poincaré gauge theory, there are  $2^6 \times 2^3$  possible multiplier configurations. Any pair of configurations differs, if not by the constrained  $J^P$  sectors, then by the ‘singular’ or ‘parallel’ nature of a constrained sector. These differences may have quite unpredictable consequences in the analysis. In order to efficiently explore the space of multiplier configurations, it is therefore important to have all the commutators between the known constraints to hand.

### 1. Scope of survey

By adding various multipliers, there are  $3 \times 2^6 \times 2^3 = 1536$  separate theories which can be constructed from the minimal PGTs. Some of these can be ruled out immediately on phenomenological grounds. In [62] we focussed on the modification of (37) by  $\tilde{\beta}_2 \neq 0$ . In the Lagrangian picture, this maps to a pair of constraints on the  $1^-$  part of the torsion tensor: suppressing velocities or equating them to gradients. The main phenomenological constraint on the minimal extensions is the requirement that they contain the Einstein–Cartan dynamics, for which the gravitational field still is described by the Riemann–Cartan curvature. For this reason, we suspect that it will generally be safer to impose the  $\{\tilde{\beta}_M\}$  than the  $\{\tilde{\alpha}_I\}$ .

How dangerous is it to suppress parts of the rotational sector? To get an idea, we consider how the two d.o.fs in Einstein–Cartan gravity might be manifest in a gravitational wave-type solution. Rather than studying the Einstein–Cartan waves directly, we instead introduce novel solutions to the special (purely quadratic, i.e.  $\hat{\alpha}_0 = 0$ ) PGT from [12,

13, 66, 67] which describe null pp-waves on the Minkowski background [86]. The wave solutions are formulated in the *Brinkmann* gauge, rather than the more popular transverse-traceless (TT) setup [87]. In a Cartesian coordinate system, the rotation gauge is chosen as in the *HiGGS* environment, so that the local Lorentz and coordinate bases are aligned, and we define ‘perpendicular’ and ‘null’ vectors as

$$\begin{aligned} e_t &\equiv \hat{e}_0, & e_x &\equiv \hat{e}_1, & e_y &\equiv \hat{e}_2, & e_z &\equiv \hat{e}_3, \\ e_{\perp} &\equiv \cos(\theta)e_x + \sin(\theta)e_y, & e_+ &\equiv e_t + e_z. \end{aligned} \quad (42)$$

We will denote the ‘wave coordinate’ as  $\tau \equiv t - z$ ; the wave amplitude is taken in all solutions to be a smooth and compact scalar function  $\mathcal{A} = \mathcal{A}(\tau)$ . The Brinkmann gauge is complementary to the TT gauge in the sense that it confines waves to the time and longitudinal components of the metric perturbation. The wave will therefore alter the definition of the unit timelike vector  $n^i$ , which defines the foliation<sup>11</sup>, and unit spacelike vector  $l_i$  which defines the direction of travel, but not the polarisation vector  $\epsilon_{\bar{i}}$ . Restricting to the case of weak waves, we will then have  $n_i \equiv h_i^t / \sqrt{|g^{tt}|} = (e^t)_i + \mathcal{O}(\mathcal{A})$ ,  $l_i \equiv h_i^z / \sqrt{|g^{zz}|} = (e^z)_i + \mathcal{O}(\mathcal{A})$  and  $\epsilon_{\bar{i}} \equiv \cos(\theta)(e^x)_i + \sin(\theta)(e^y)_i$ . While the Brinkmann gauge is opposed to the TT gauge at the level of the metric, this turns out to be somewhat reversed at the level of the field strengths. Accordingly, it is useful to define the ‘TT symmetric-traceless’ operation on the indices of a general TT tensor  $X_{ij}$  as  $X_{\langle ij \rangle} \equiv X_{(ij)} - \frac{1}{2}X_{\bar{k}}^{\bar{k}}(\eta_{ij} - l_i l_j)$ , where we recall the original ‘symmetric-traceless’ operator  $X_{(\bar{i}\bar{j})} \equiv X_{(ij)} - \frac{1}{3}X_{\bar{k}}^{\bar{k}}\eta_{ij}$  from [12].

The new exact solution which concerns us describes a wave in the Riemann–Cartan curvature, with vanishing torsion and two d.o.fs quantified by a polarisation vector. It has the components

$$\mathcal{R}_{\langle \bar{i}\bar{j} \rangle} = \mathcal{A} \epsilon_{\langle \bar{i}} \epsilon_{\bar{j} \rangle} + \mathcal{O}(\mathcal{A}^2), \quad (43a)$$

$$\mathcal{R}_{\perp \langle \bar{i}\bar{j} \rangle \perp} = \mathcal{A} \epsilon_{\langle \bar{i}} \epsilon_{\bar{j} \rangle} + \mathcal{O}(\mathcal{A}^2), \quad (43b)$$

$${}^T\mathcal{R}_{\perp ijk} = \mathcal{A} \epsilon_{\langle \bar{k}} \epsilon_{\bar{i} \rangle} l_{\bar{j}} + \mathcal{O}(\mathcal{A}^2), \quad (43c)$$

$${}^T\mathcal{R}_{ijk\perp} = -\mathcal{A} \epsilon_{\langle \bar{k}} \epsilon_{\bar{i} \rangle} l_{\bar{j}} + \mathcal{O}(\mathcal{A}^2). \quad (43d)$$

These components turn out to be identical [86] to those of the Riemann tensor in the presence of the vacuum pp-waves known from GR. Recall that Eqs. (37) to (39) were initially set up as modifications of Einstein–Cartan theory. Since the Einstein–Cartan theory differs from GR only by a contact torsion interaction, and since the linear  $1^+$ ,  $1^-$  and  $2^+$  modes are massive, it would seem strange if the the null pp-wave solution Eqs. (43a) to (43d) is not also *mandatory* in the minimal extensions. Referring back to [62], we find that we should then

<sup>11</sup> For this reason the Brinkmann gauge will not be a natural choice for the canonical analysis.

always take  $\bar{\alpha}_1 = 0$ , since we would otherwise encounter the Lagrangian constraints

$$\bar{\alpha}_1 \neq 0 \Rightarrow \mathcal{R}_{\langle ij \rangle} + \mathcal{R}_{\perp \langle ij \rangle \perp} \approx {}^T \mathcal{R}_{\perp jk} - {}^T \mathcal{R}_{ijk\perp} \approx 0, \quad (44)$$

which both force the wave amplitude  $\mathcal{A} \rightarrow 0$ . This already halves the volume of our parameter space, and so we make no attempt exhaustively determine the various other phenomenological constraints.

Other limitations on the allowed multipliers come from the linearised particle spectrum. For the  $1^+$  theory with  $\bar{\alpha}_1 = 0$  only the group  $\{\bar{\alpha}_3, \bar{\alpha}_4, \bar{\alpha}_6, \bar{\beta}_1, \bar{\beta}_2, \bar{\beta}_3\}$  does not immediately constrain  $\hat{\pi}_{\perp ij}$ . Similarly for the  $1^-$  theory the space is  $\{\bar{\alpha}_2, \bar{\alpha}_3, \bar{\alpha}_6, \bar{\beta}_1, \bar{\beta}_2, \bar{\beta}_3\}$ . For the  $2^-$  theory, we consider the group  $\{\bar{\alpha}_3, \bar{\alpha}_5, \bar{\alpha}_6, \bar{\beta}_1, \bar{\beta}_2, \bar{\beta}_3\}$ . Rightly, we ought to allow  $\bar{\alpha}_4 \neq 0$  in this case, especially since the momentum  $\hat{\pi}_{\perp ij}$  of any strongly coupled  $2^+$  mode would then be disabled: just for this initial survey, however, the complexity of the  $2^-$  calculations is such that restricting to  $\bar{\alpha}_4 = 0$  results in a significant economy.

The aim is to obtain covariant expressions for all the possible commutators among all known if-constraints for all  $3 \times 2^3 \times 2^3 = 192$  theories stipulated above. The current *HiGGS* setup is supposed to be able to do this, and produce binaries of the results suitable for use in a database. The requisite calculations would take years on a single desktop computer core, so we distribute them over 14 nodes of the Peta4 supercomputer – the CPU cluster component of the heterogeneous CSD3 facility. Each node has two 2.60 GHz Intel® Xeon® Skylake 6142 CPUs, each having 16 cores with 6 GB of memory apiece, amounting to 448 processors. As described in Section II F, *HiGGS* runs on a master kernel within each node. The master kernel delegates the analysis of a (randomly allocated) batch of theories, and has a total of 32 parallel sub-kernels at its disposal.

An illustration of the survey was shown already in Fig. 1. The node count of 14 is a service-level restriction on Peta4 rather than limitation of the implementation. We can see from a visual inspection that not all theories are equally expensive: some multiplier configurations engender more constraints and more brackets, while others more heavily involve the higher-spin sectors. An examination of the stack trace from each node suggests that the most time-consuming cases are those for which a multiplier is used to constrain a  $J^P$  sector which was thought to be nonlinearly activated. This is an interesting observation in the context of the strong coupling objectives. In particular, delays occur when new ‘parallel’ or ‘singular’ SiCs of the form

$${}^A \chi_{\dot{\nu}}^{\parallel} \equiv {}^A \check{\mathcal{P}}_{\dot{\nu}nm}^{\overline{pq}} \mathcal{R}^{nm}_{\overline{pq}} \approx 0, \quad (45a)$$

$${}^E \chi_{\dot{\nu}}^{\parallel} \equiv m_p {}^2 E \check{\mathcal{P}}_{\dot{\nu}n}^{\overline{pq}} \mathcal{T}^n_{\overline{pq}} \approx 0, \quad (45b)$$

$${}^A \chi_{\dot{\nu}}^{\perp\perp} \equiv {}^A \varphi_{\dot{\nu}} + \frac{8\bar{\alpha}_A^{\perp\perp} \hat{\alpha}_A^{\perp\perp}}{\bar{\alpha}_A^{\perp\perp}} {}^A \check{\mathcal{P}}_{\dot{\nu}jk}^{\overline{lm}} \mathcal{R}^{jk}_{\overline{lm}} \approx 0, \quad (45c)$$

$${}^E \chi_{\dot{\nu}}^{\perp\perp} \equiv {}^E \varphi_{\dot{\nu}} + \frac{4\bar{\beta}_E^{\perp\perp} \hat{\beta}_E^{\perp\perp}}{\bar{\beta}_E^{\perp\perp}} m_p {}^2 E \check{\mathcal{P}}_{\dot{\nu}j}^{\overline{lm}} \mathcal{T}^j_{\overline{lm}} \approx 0, \quad (45d)$$

are introduced. It is known from manual calculations in [13] that these secondaries (specifically their field strength terms) fail to commute with many other constraints, and produce brackets of surprising complexity.

Physics binaries of all nonlinear brackets identified in this survey, along with plaintext stack traces and clocking times, can be found in the supplemental materials [64].

## 2. New physics

An obvious choice of multiplier coupling, again following from (??) and our discussion in ??, is to set  $\bar{\alpha}_4 \neq 0$ . By so doing, we can shut down the  $1^-$  sector by constraining  $\hat{\pi}_{\perp i}$ , with the collateral damage falling in the  $0^+$  and  $2^+$  sectors. Since the PiCs include  $\varphi_{\perp} \approx 0$  and  $\tilde{\varphi}_{\perp ij} \approx 0$ , we might expect the impact to be minimal.

In practice, the constraints listed already in ?? are joined in the  $1^-$  sector by the perpendicular<sup>12</sup> and parallel secondaries

$$\vec{\chi}_{\perp i}^{\perp} \equiv \frac{1}{J} \vec{\hat{\pi}}_i + 2\bar{\alpha}_4 (\lambda_{\perp i} + \lambda_{i\perp}) \approx \vec{\chi}_i^{\parallel} \equiv \mathcal{R}_{\perp i} \approx 0. \quad (46)$$

So far, so good, but in fact the fallout is more substantial across the  $0^+$  and  $2^+$  sectors than we had expected, because no further multipliers can be safely activated. We find accordingly the parallel secondaries

$$\chi_{\perp}^{\parallel} \equiv \mathcal{R} \approx \tilde{\chi}_{\perp ij}^{\parallel} \equiv \mathcal{R}_{\langle ij \rangle} \approx 0. \quad (47)$$

The second constraint in (47) has in particular the potential to become a real phenomenological problem. We reach this conclusion by a consideration of how the two graviton d.o.f might be manifest in a gravitational wave-type solution.

## IV. CONCLUSIONS

In this paper we have presented the package *HiGGS*, written for the tensor manipulation suite *xAct* and the computer algebra software *Mathematica*. The *HiGGS* package performs very heavy calculations – such as Poisson brackets – which frequently arise in the Hamiltonian (canonical) analysis of modified gravity theories with curvature and/or torsion. The current iteration of the package is tailored to the generalised Poincaré gauge theory in Eq. (7): given an action of that class, *HiGGS* can identify primary and secondary constraints, and calculate arbitrary field velocities. For more original torsionful actions,

<sup>12</sup> We will use the same notation for discussing the SO(3) multiplier irreps as we have done for the field strength tensors, merely substituting with the symbol  $(\lambda)$ .

*HiGGS* may still be used to evaluate brackets, and canonicalise expressions by irreducible decomposition.

Parallelisation is a core feature of *HiGGS*. We have argued that those aspects of the Dirac–Bergmann Hamiltonian constraint algorithm which are so daunting to researchers, are actually very well suited to parallel computing. On a laptop or desktop computer, *HiGGS* can take advantage of available cores to shorten the analysis of a given theory. Parallelisation is done *within* the *HiGGS* environment. This capability meshes well with the problem of surveying large numbers of action configurations using HPC, since it avoids the need to educate general-purpose job scheduling tools (such as *SLURM* [88] or *TORQUE* [89]) about the Hamiltonian structure. In our example HPC survey, all job-scheduling decisions were made from within the *HiGGS* environment, to whole-node granularity.

The HPC survey we have performed in this work targets minimal extensions to the Einstein–Cartan theory, in which a single extra massive spin-parity  $1^+$ ,  $1^-$  or  $2^-$  torsion particle is present, i.e. extending the two usual graviton polarisations by three or five extra degrees of freedom. These minimal extensions were believed [90–92] to be ghost free according to the linearised analysis. Subsequent nonlinear Hamiltonian analysis [1] suggested that the linear regime strongly couples extra modes of spin-parity  $1^-$ ,  $1^+$  or  $2^+$  respectively, so that these modes spoil the theories’ viability. Based on the hypothesis that multiplier fields could selectively suppress these modes – à la teleparallel gravity – our survey lists all the commutators among all the known primary and secondary ‘if-constraints’ (i.e. constraints which may arise due to a choice of Lagrangian couplings), for the various possible multiplier configurations. Our level of analysis at least matches that of mode activation<sup>13</sup> in [1], where the primary Poisson matrices of the minimal Einstein–Cartan extensions are obtained. We perform our own inspection of the results (which are available in the supplemental materials [64]) in [62].

While the *HiGGS* package may be of use to researchers in the ways described above, we must make an effort to observe some of its limitations;

- The `PoissonBracket[]` module automatically expands its operands, rather than first taking advantage of the Leibniz rule wherever those operands are products of covariant factors. This can result in costly attempts by `ToNesterForm[]` to covariantise arbitrarily complex brackets.
- The `Velocity[]` module is directly associated with the canonical Hamiltonian in (30) of the generalised Poincaré gauge theory in (7). This could easily be avoided with Leibniz rule functionality in `PoissonBracket[]`.
- The `PoissonBracket[]` module is incapable of processing the second-order Euler–Lagrange formulation.

The lack (to our knowledge) of a general formula in Poincaré gauge theory for the second-order bracket was also a limiting factor for our previous analysis in [12].

- In general, *HiGGS* relies quite heavily on the gauge-covariant derivatives  $D_\mu$  or  $\mathcal{D}_\mu$ . The *xAct* suite already has a very sophisticated functionality to accommodate the definition of such derivatives, which is not exploited by the *HiGGS* implementation.
- In general, *HiGGS* relies very heavily on the  $\text{SO}^+(1,3)$  and (particularly) the  $\text{SO}(3)$  decompositions of tensorial fields. However, these are manually defined in the implementation, for *each decomposed field*. A clear case is made for a general decomposition tool, more closely integrated with the existing functionality in `xAct` ‘`SymManipulator`’.
- Notwithstanding the Lagrangian structure implied by `Velocity[]`, inclusion of new dynamical variables constitutes a different problem. We are confident that extension to the metric affine gauge theory (MAGT) structure would be a straightforward, if time-consuming, exercise. More work would likely be needed for the inclusion of (e.g. fermionic) matter fields.
- The background assumed by *HiGGS* when the option “ToOrder” $\rightarrow$ 0 or “ToOrder” $\rightarrow$ 1 is passed to modules such as `ToNesterForm[]`, is Minkowski space-time with vanishing background torsion. There are, however, clear motivations for considering curved backgrounds, such as de Sitter or Schwarzschild. Moreover in [12, 13, 66, 67] it is argued that non-minimal gravitational gauge theories, which are detached from the geometrical trinity, only become viable in an attractor background of *constant axial torsion*.

Whether these limitations are best addressed by improving the *HiGGS* implementation, or beginning ab initio with an improved understanding of the challenges posed by canonical computer algebra, remains to be seen. For the moment, we hope at least to have shown that modified gravity is now ready for computer algebra assistance *at scale*. In this sense we build on the recent work of Lin, Hobson and Lasenby [73, 93], who used computer algebra to systematically obtain all ghost and tachyon-free cases of the linearised Poincaré gauge theory, and did so with far more limited resources than are brought to bear in this paper. A future is then suggested in which vague concerns – viz, a potential for strong coupling demonstrated among a handful of cases – are no longer valid grounds upon which to dismiss a rich class of theories.

## ACKNOWLEDGMENTS

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service ([www.csd3.cam.ac.uk](http://www.csd3.cam.ac.uk)), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and

<sup>13</sup> We do not, however, consider the problem of constraint bifurcation using the Poisson matrix pseudodeterminant.

DiRAC funding from the Science and Technology Facilities Council ([www.dirac.ac.uk](http://www.dirac.ac.uk)).

This manuscript was improved by the kind suggestions of Amel Duraković and Tom Złosnik.

I am grateful for the kind hospitality of Leiden University

and the Lorentz Institute, and the support of Girton College, Cambridge.

The current version of *HiGGS* incorporates elements of Cyril Pitrou's code from the repository at [www.github.com/xAct-contrib/examples](https://www.github.com/xAct-contrib/examples).

- 
- [1] H.-J. Yo, J. M. Nester, and W. T. Ni, *Int. J. Mod. Phys. D* **11**, 747 (2002), [arXiv:gr-qc/0112030 \[gr-qc\]](#).
  - [2] A. Golovnev and M.-J. Guzmán, (2022), [arXiv:2203.16610 \[gr-qc\]](#).
  - [3] L. Sebastiani, S. Vagnozzi, and R. Myrzakulov, *Adv. High Energy Phys.* **2017**, 3156915 (2017), [arXiv:1612.08661 \[gr-qc\]](#).
  - [4] B. Famaey and S. McGaugh, *Living Rev. Rel.* **15**, 10 (2012), [arXiv:1112.3960 \[astro-ph.CO\]](#).
  - [5] C. Skordis and T. Złosnik, *Phys. Rev. Lett.* **127**, 161302 (2021), [arXiv:2007.00082 \[astro-ph.CO\]](#).
  - [6] M. Henneaux and C. Teitelboim, *Quantization of gauge systems* (Princeton University Press, Princeton, New Jersey, 1992).
  - [7] M. Blagojević, *Gravitation and Gauge Symmetries*, Series in high energy physics, cosmology, and gravitation (Institute of Physics Publishing, Bristol, UK, 2002).
  - [8] J. L. Anderson and P. G. Bergmann, *Phys. Rev.* **83**, 1018 (1951).
  - [9] P. G. Bergmann and I. Goldberg, *Phys. Rev.* **98**, 531 (1955).
  - [10] L. Castellani, *Ann. Physics (N. Y.)* **143**, 357 (1982).
  - [11] P. A. M. Dirac, *Proc. Math. Phys. Eng. Sci.* **246**, 326 (1958).
  - [12] W. E. V. Barker, A. N. Lasenby, M. P. Hobson, and W. J. Handley, *Phys. Rev. D* **104**, 084036 (2021), [arXiv:2101.02645 \[gr-qc\]](#).
  - [13] W. E. V. Barker, *Gauge theories of gravity*, Ph.D. thesis, Wolfson College, University of Cambridge (2021).
  - [14] S. Weinberg and E. Witten, *Physics Letters B* **96**, 59 (1980).
  - [15] J. W. York, Jr., *Phys. Rev. Lett.* **26**, 1656 (1971).
  - [16] N. Arkani-Hamed, H.-C. Cheng, M. A. Luty, and S. Mukohyama, *J. High Energy Phys.* **05**, 074 (2004), [arXiv:hep-th/0312099](#).
  - [17] J. S. Bagla, H. K. Jassal, and T. Padmanabhan, *Phys. Rev. D* **67**, 063504 (2003), [arXiv:astro-ph/0212198](#).
  - [18] D. Carney *et al.*, (2022), [arXiv:2203.06508 \[hep-ph\]](#).
  - [19] C. de Rham, J. T. Deskins, A. J. Tolley, and S.-Y. Zhou, *Rev. Mod. Phys.* **89**, 025004 (2017), [arXiv:1606.08462 \[astro-ph.CO\]](#).
  - [20] V. B. Klaer and G. D. Moore, *JCAP* **11**, 049 (2017), [arXiv:1708.07521 \[hep-ph\]](#).
  - [21] C. Burrage and J. Sakstein, *Living Rev. Rel.* **21**, 1 (2018), [arXiv:1709.09071 \[astro-ph.CO\]](#).
  - [22] D. Baumann, L. Senatore, and M. Zaldarriaga, *JCAP* **05**, 004 (2011), [arXiv:1101.3320 \[hep-th\]](#).
  - [23] C. Charmousis and A. Padilla, *JHEP* **12**, 038 (2008), [arXiv:0807.2864 \[hep-th\]](#).
  - [24] A. Wang, *Int. J. Mod. Phys. D* **26**, 1730014 (2017), [arXiv:1701.06087 \[gr-qc\]](#).
  - [25] C. Charmousis, G. Niz, A. Padilla, and P. M. Saffin, *JHEP* **08**, 070 (2009), [arXiv:0905.2579 \[hep-th\]](#).
  - [26] A. Papazoglou and T. P. Sotiriou, *Phys. Lett. B* **685**, 197 (2010), [arXiv:0911.1299 \[hep-th\]](#).
  - [27] M. Fierz and W. Pauli, *Proc. Roy. Soc. Lond. A* **173**, 211 (1939).
  - [28] H. van Dam and M. J. G. Veltman, *Nucl. Phys. B* **22**, 397 (1970).
  - [29] V. I. Zakharov, *JETP Lett.* **12**, 312 (1970).
  - [30] D. G. Boulware and S. Deser, *Phys. Rev. D* **6**, 3368 (1972).
  - [31] N. Arkani-Hamed, H. Georgi, and M. D. Schwartz, *Annals Phys.* **305**, 96 (2003), [arXiv:hep-th/0210184](#).
  - [32] A. I. Vainshtein, *Phys. Lett. B* **39**, 393 (1972).
  - [33] C. Deffayet, G. R. Dvali, G. Gabadadze, and A. I. Vainshtein, *Phys. Rev. D* **65**, 044026 (2002), [arXiv:hep-th/0106001](#).
  - [34] C. Deffayet and J.-W. Rombouts, *Phys. Rev. D* **72**, 044003 (2005), [arXiv:gr-qc/0505134](#).
  - [35] C. de Rham, *Living Rev. Rel.* **17**, 7 (2014), [arXiv:1401.4173 \[hep-th\]](#).
  - [36] C. de Rham and G. Gabadadze, *Phys. Rev. D* **82**, 044020 (2010), [arXiv:1007.0443 \[hep-th\]](#).
  - [37] C. de Rham, G. Gabadadze, and A. J. Tolley, *Phys. Rev. Lett.* **106**, 231101 (2011), [arXiv:1011.1232 \[hep-th\]](#).
  - [38] P. Motloch, W. Hu, A. Joyce, and H. Motohashi, *Phys. Rev. D* **92**, 044024 (2015), [arXiv:1505.03518 \[hep-th\]](#).
  - [39] K. Hinterbichler, A. Nicolis, and M. Porrati, *JHEP* **09**, 089 (2009), [arXiv:0905.2359 \[hep-th\]](#).
  - [40] J. Magueijo, *Phys. Rev. D* **79**, 043525 (2009), [arXiv:0807.1689 \[gr-qc\]](#).
  - [41] N. Afshordi, D. J. H. Chung, and G. Geshnizjani, *Phys. Rev. D* **75**, 083513 (2007), [arXiv:hep-th/0609150 \[hep-th\]](#).
  - [42] H. Chen, J. M. Nester, and H.-J. Yo, *Acta Phys. Pol. B* **29**, 961 (1998).
  - [43] H.-J. Yo and J. M. Nester, *Int. J. Mod. Phys. D* **08**, 459 (1999), [arXiv:gr-qc/9902032 \[gr-qc\]](#).
  - [44] J. Beltrán Jiménez, L. Heisenberg, and T. S. Koivisto, *Universe* **5**, 173 (2019), [arXiv:1903.06830 \[hep-th\]](#).
  - [45] R. Aldrovandi and J. G. Pereira, *Teleparallel Gravity: An Introduction* (Springer, 2013).
  - [46] J. M. Nester and H.-J. Yo, *Chin. J. Phys.* **37**, 113 (1999), [arXiv:gr-qc/9809049](#).
  - [47] J. Beltrán Jiménez, L. Heisenberg, and T. S. Koivisto, *Int. J. Mod. Phys. D* **28**, 1944012 (2019), [arXiv:1903.12072 \[gr-qc\]](#).
  - [48] D. K. Blixt, *Hamiltonian analysis of covariant teleparallel theories of gravity*, Ph.D. thesis, Tartu U., Tartu University (2021).
  - [49] J. Beltrán Jiménez and T. S. Koivisto, *Phys. Rev. D* **105**, L021502 (2022), [arXiv:2111.04716 \[gr-qc\]](#).
  - [50] J. Beltrán Jiménez, D. De Andrés, and A. Delhom, *Class. Quant. Grav.* **37**, 225013 (2020), [arXiv:2006.07406 \[gr-qc\]](#).
  - [51] A. Trautman, (2006), [arXiv:gr-qc/0606062](#).
  - [52] F. W. Hehl, P. Von Der Heyde, G. D. Kerlick, and J. M. Nester, *Rev. Mod. Phys.* **48**, 393 (1976).
  - [53] Y. N. Obukhov, *Int. J. Geom. Meth. Mod. Phys.* **3**, 95 (2006), [arXiv:gr-qc/0601090](#).
  - [54] A. Lasenby, C. Doran, and S. Gull, *Philos. Trans. R. Soc A* **356**, 487 (1998), [arXiv:gr-qc/0405033 \[gr-qc\]](#).
  - [55] T. W. B. Kibble, *J. Math. Phys.* **2**, 212 (1961).
  - [56] R. Utiyama, *Phys. Rev.* **101**, 1597 (1956).
  - [57] D. W. Sciama, *Rev. Mod. Phys.* **36**, 463 (1964).
  - [58] M. Blagojević and I. A. Nikolić, *Phys. Rev. D* **28**, 2455 (1983).
  - [59] M. Blagojević and M. Vasilčić, *Phys. Rev. D* **35**, 3748 (1987).
  - [60] M. Blagojević and B. Cvetković, *Phys. Rev. D* **98**, 024014 (2018), [arXiv:1804.05556 \[gr-qc\]](#).
  - [61] J. Beltrán Jiménez and A. Jiménez-Cano, *JCAP* **01**, 069 (2021), [arXiv:2009.08197 \[gr-qc\]](#).



- [62] W. E. V. Barker, *Geometric multipliers and partial teleparallelism in Poincaré gauge theory* (In prep. for Phys. Rev. D, 2022).
- [63] W. E. V. Barker, *Hamiltonian Gauge Gravity Surveyor (HiGGS): Source Files* (2022) See Supplemental Material at [www.github.com/wevbarker/HiGGS](http://www.github.com/wevbarker/HiGGS).
- [64] W. E. V. Barker, *Hamiltonian Gauge Gravity Surveyor (HiGGS): Calibration Survey* (2022) See Supplemental Material at [www.github.com/wevbarker/HiGGS-calibration-survey](http://www.github.com/wevbarker/HiGGS-calibration-survey).
- [65] A. N. Lasenby and M. P. Hobson, *Journal of Mathematical Physics* **57**, 092505 (2016), arXiv:1510.06699 [gr-qc].
- [66] W. E. V. Barker, A. N. Lasenby, M. P. Hobson, and W. J. Handley, *Phys. Rev. D* **102**, 024048 (2020), arXiv:2003.02690 [gr-qc].
- [67] W. E. V. Barker, A. N. Lasenby, M. P. Hobson, and W. J. Handley, *Phys. Rev. D* **102**, 084002 (2020), arXiv:2006.03581 [gr-qc].
- [68] C.-M. Chen, J. M. Nester, and R.-S. Tung, *Int. J. Mod. Phys. D* **24**, 1530026 (2015), arXiv:1507.07300 [gr-qc].
- [69] M. Hohmann, *Eur. Phys. J. C* **81**, 504 (2021), arXiv:2012.14984 [gr-qc].
- [70] A. N. Lasenby and M. P. Hobson, *J. Math. Phys.* **57**, 092505 (2016), arXiv:1510.06699 [gr-qc].
- [71] E. Ş. Kutluk, *Adiabatic Solutions in General Relativity and Boundary Symmetries*, Ph.D. thesis, Bogazici University (2019).
- [72] K. Hayashi and T. Shirafuji, *Prog. Theor. Phys.* **64**, 866 (1980).
- [73] Y.-C. Lin, M. P. Hobson, and A. N. Lasenby, arXiv e-prints, arXiv:2005.02228 (2020), arXiv:2005.02228 [gr-qc].
- [74] K.-F. Shie, J. M. Nester, and H.-J. Yo, *Phys. Rev. D* **78**, 023522 (2008), arXiv:0805.3834 [gr-qc].
- [75] H. Chen, F.-H. Ho, J. M. Nester, C.-H. Wang, and H.-J. Yo, *J. Cosmology Astropart. Phys.* **2009**, 027 (2009), arXiv:0908.3323 [gr-qc].
- [76] P. Baekler, F. W. Hehl, and J. M. Nester, *Phys. Rev. D* **83**, 024001 (2011), arXiv:1009.5112 [gr-qc].
- [77] F.-H. Ho and J. M. Nester, in *Journal of Physics Conference Series*, Journal of Physics Conference Series, Vol. 330 (2011) p. 012005, arXiv:1105.5001 [gr-qc].
- [78] F.-H. Ho and J. M. Nester, *Int. J. Mod. Phys. D* **20**, 2125 (2011).
- [79] F.-H. Ho, H. Chen, J. M. Nester, and H.-J. Yo, *Chin. J. Phys.* **53**, arXiv:1512.01202 (2015), arXiv:1512.01202 [gr-qc].
- [80] K. Sundermeyer, *Constrained dynamics* (Springer, Berlin, Germany, 1982) p. 68.
- [81] M. Blagojevic, M. Vasilic, and T. Vukasinac, *Class. Quant. Grav.* **11**, 2143 (1994), arXiv:hep-th/9311146.
- [82] I. A. Nikolić, *Phys. Rev. D* **30**, 2508 (1984).
- [83] T. Nutma, *Comput. Phys. Commun.* **185**, 1719 (2014), arXiv:1308.3493 [cs.SC].
- [84] R. Maeder, *The Mathematica Programmer* (AP Professional, Elsevier, Cambridge, Massachusetts, 1994).
- [85] D. P. Rodgers, *SIGARCH Comput. Archit. News* **13**, 225–231 (1985).
- [86] A. N. Lasenby, (2021), unpublished.
- [87] A. N. Lasenby, *Adv. Appl. Clifford Algebras* **29**, 79 (2019), arXiv:1912.05960 [gr-qc].
- [88] A. B. Yoo, M. A. Jette, and M. Grondona, in *Job Scheduling Strategies for Parallel Processing*, edited by D. Feitelson, L. Rudolph, and U. Schwiegelshohn (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003) pp. 44–60.

- [89] G. Staples, in *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, SC '06 (Association for Computing Machinery, New York, NY, USA, 2006) p. 8–es.
- [90] E. Sezgin and P. van Nieuwenhuizen, *Phys. Rev. D* **21**, 3269 (1980).
- [91] E. Sezgin, *Phys. Rev. D* **24**, 1677 (1981).
- [92] K. Hayashi and T. Shirafuji, *Prog. Theor. Phys.* **64**, 1435 (1980).
- [93] Y.-C. Lin, M. P. Hobson, and A. N. Lasenby, *Phys. Rev. D* **101**, 064038 (2020), arXiv:1910.14197 [gr-qc].

## Appendix A: Linear velocities of the simple spin-1<sup>+</sup> case

In this appendix we provide the linearised velocities of the PiCs appearing in the simple 1<sup>+</sup> extension to Einstein–Cartan theory:

```
In[] := ViewTheory["simple_spin_1p", "Literature"
  ↳ " → False, "PPM" → False, "Velocities"
  ↳ → True];
** DefTheory: Incorporating the binary at svy
  ↳ /simple_spin_1p.thr.mx
```

$$\begin{aligned} \frac{d}{dt} \phi_b &\approx \hat{\alpha}_0 N \phi^{\mathcal{I}} \\ \frac{d}{dt} \phi_{\mathcal{I}} &\approx -\frac{2 \left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}a} N}{\mathcal{I}} - \frac{\hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}i}}{4 \hat{\alpha}_5 \mathcal{I}} + \hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \mathcal{I}_i \\ \frac{d}{dt} \phi_{\mathcal{I}ij} &\approx \hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \mathcal{I}^{\mathcal{I}}_{ij} \\ \frac{d}{dt} \phi_{\mathcal{I}} &\approx -\frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}a} N}{\mathcal{I}} \\ \frac{d}{dt} \phi_{\mathcal{I}} &\approx -\frac{2 \left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}a} \mathcal{I}^{\mathcal{I}}_{\mathcal{I}b} \mathcal{I}^{\mathcal{I}}_{\mathcal{I}a} N}{\mathcal{I}} - \hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \mathcal{I}^{\mathcal{I}} - 8 \hat{\beta}_3 N \mathcal{I}^{\mathcal{I}} \mathcal{I}^{\mathcal{I}} \\ \frac{d}{dt} \phi_{\mathcal{I}ij} &\approx \frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}ij} N}{4 \mathcal{I}} + \frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}ji} N}{4 \mathcal{I}} - \frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}a} \mathcal{I}^{\mathcal{I}}_{\mathcal{I}ij} N}{6 \mathcal{I}} \\ \frac{d}{dt} \phi_{\mathcal{I}ijk} &\approx \frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}ijk} N}{4 \mathcal{I}} - \frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}jik} N}{4 \mathcal{I}} - \frac{\left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}kij} N}{2 \mathcal{I}} \\ &\quad - \frac{3 \left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}ja} \mathcal{I}^{\mathcal{I}}_{\mathcal{I}ik} N}{8 \mathcal{I}} + \frac{3 \left( \mathcal{D}^{\mathcal{I}} \hat{\pi}_{\mathcal{I}} \right)^{\mathcal{I}}_{\mathcal{I}ia} \mathcal{I}^{\mathcal{I}}_{\mathcal{I}jk} N}{8 \mathcal{I}} + \\ &\quad \frac{2}{3} \hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \mathcal{I}^{\mathcal{I}}_{ijk} + \frac{1}{3} \hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \mathcal{I}^{\mathcal{I}}_{ikj} - \frac{1}{3} \hat{\alpha}_0 N \mathcal{I}^{\mathcal{I}} \mathcal{I}^{\mathcal{I}}_{jki} \end{aligned}$$

These expressions are expressed on the PiC shell, but not on the sSFC shell. We note in particular the general presence of field strength gradients, i.e. second derivative quantities even in the first-order formulation of gravity that is PGT.