Metric Symmetry by Design

A Lagrangian perspective with computer algebra adventures

Historical origins

- **The Einstein-Hilbert Lagrangian**

$$S=\int d^4x\sqrt{-g}\left(R-2\Lambda\right)+S_{\text{matter}}$$

- **The corresponding Euler-Lagrange field equation is Einstein's field equation:**

$$R_{\mu\nu}-\frac{1}{2}R\,g_{\mu\nu}+\Lambda\,g_{\mu\nu}=8\pi G\,T_{\mu\nu}.$$

- **But how is it derived?**

The canonical route

- **Lagrangian with second derivatives**

$$\frac{1}{\sqrt{-g}}\left[\frac{\partial \mathcal{L}_{\text{grav}}}{\partial g_{\mu\nu}} - \frac{\partial}{\partial x^{\kappa}}\frac{\partial \mathcal{L}_{\text{grav}}}{\partial g_{\mu\nu,\kappa}} + \frac{\partial^2}{\partial x^{\kappa}\partial x^{\lambda}}\frac{\partial \mathcal{L}_{\text{grav}}}{\partial g_{\mu\nu,\kappa\lambda}}\right] + [\text{matter terms}] = 0.$$

- **Higher derivatives appear only linearly as total derivatives, so we avoid the Ostrogradsky instability**
- **Field equations remain second order**

Suitability for computer algebra

- Straightforward derivation involving field derivatives
- No “magic”, no heuristic assumptions, just deterministic steps
- Can a CAS handle the abstract index formalism efficiently?

MACSYMA, Maxima and itensor

- MACSYMA – one of the oldest CAS, oldest in active development
- Open-sourced (thanks to the late William Shelter) as Maxima around 2000, in active development since
- Contains dedicated tensor algebra and calculus packages

Indexed tensors in CAS

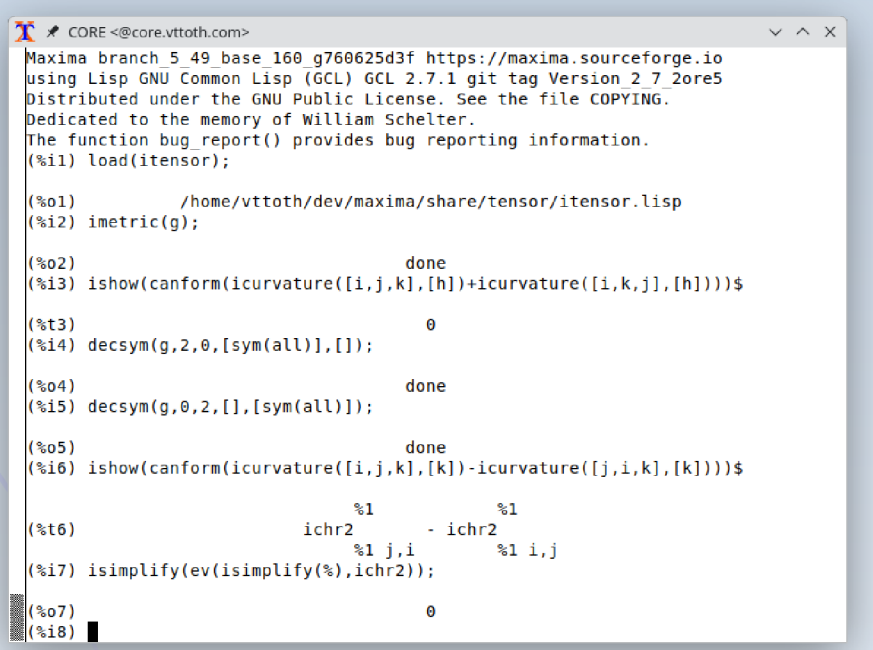
- Maxima's itensor takes the abstract index formalism literally
- Tensors are opaque objects, manipulated without coordinate-dependent representation

A few examples

- `S([],[])` **is a scalar field**
- `V([],[a])` **is** V^a**, a vector field**
- `V([],[a],b)` **is its coordinate derivative with respect to** x^b
- `T([a],[b,c])` **is a rank-3 tensor,** $T_a{}^{bc}$**.**
- **Built-in primitives/representations for the curvature tensor and Christoffel-symbols**
- **Recognizing symmetries and contraction rules**
- **Ability to represent a frame base**

Algebraic capabilities

- Flexible symmetry property declarations
- Powerful simplification makes it easy to prove many identities

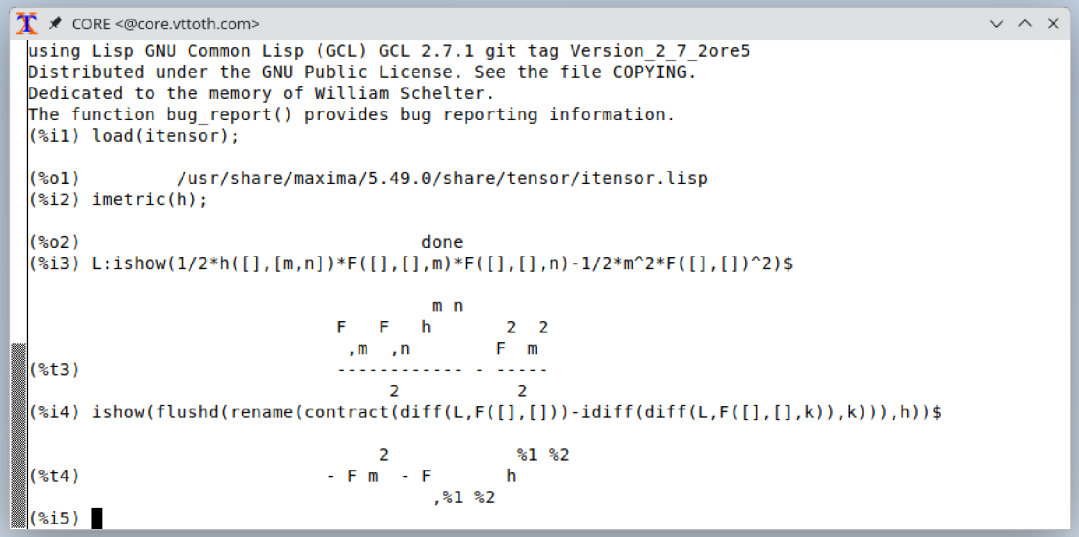


Variational calculus

- The action principle is based on variational calculus
- Formal derivation of field equations or equations of motion requires functional differentiation
- Capability was absent in itensor prior to 2005

A simple example

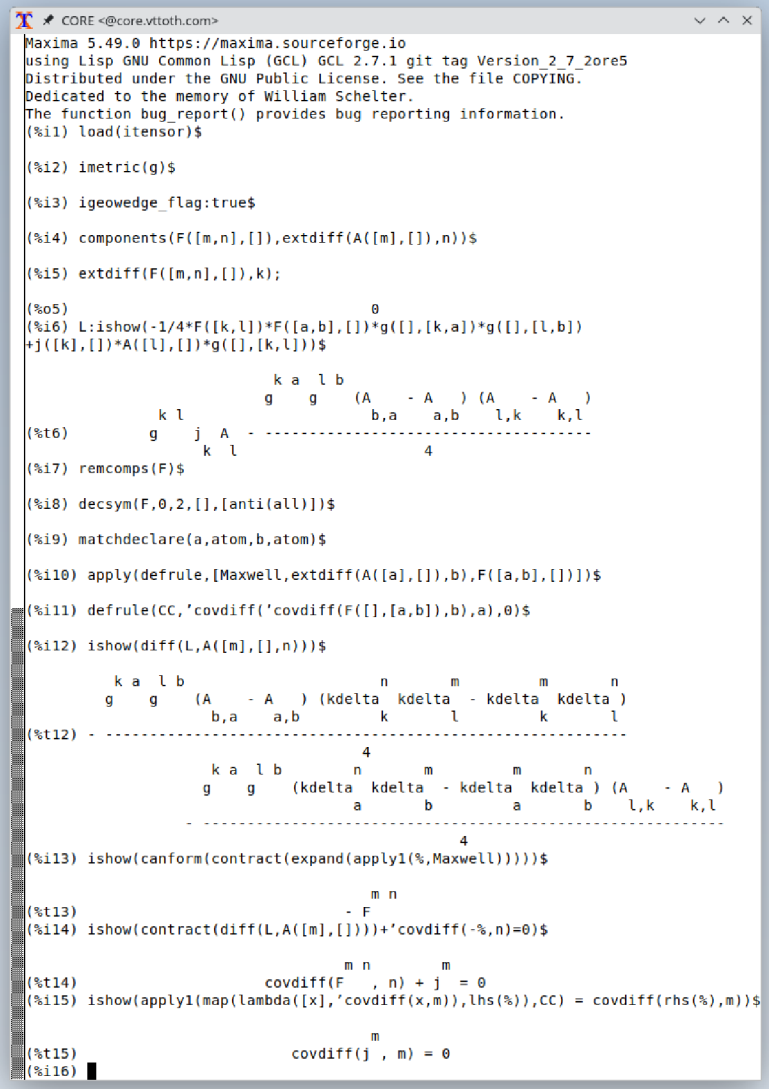
- **Scalar field theory,** $\mathcal{L}=\frac{1}{2}\eta^{\mu\nu}\varphi_\mu\varphi_\nu-\frac{1}{2}m^2\varphi^2$:



- **Which of course we recognize as the field equation** $\left(\nabla_\mu\nabla^\mu+m^2\right)\varphi=0.$

Another example: Maxwell

- Derivation of Maxwell's equations from the EM Lagrangian
- The derivation is valid even in curved spacetime



Gravitational theories

- The metric is present as a field but also as the integration measure of the action integral
- Maxima can properly deal with terms like $\sqrt{-g}$ in the integrand
- Deriving the EFE from the Einstein-Hilbert Lagrangian should be easy

Let us do the derivation

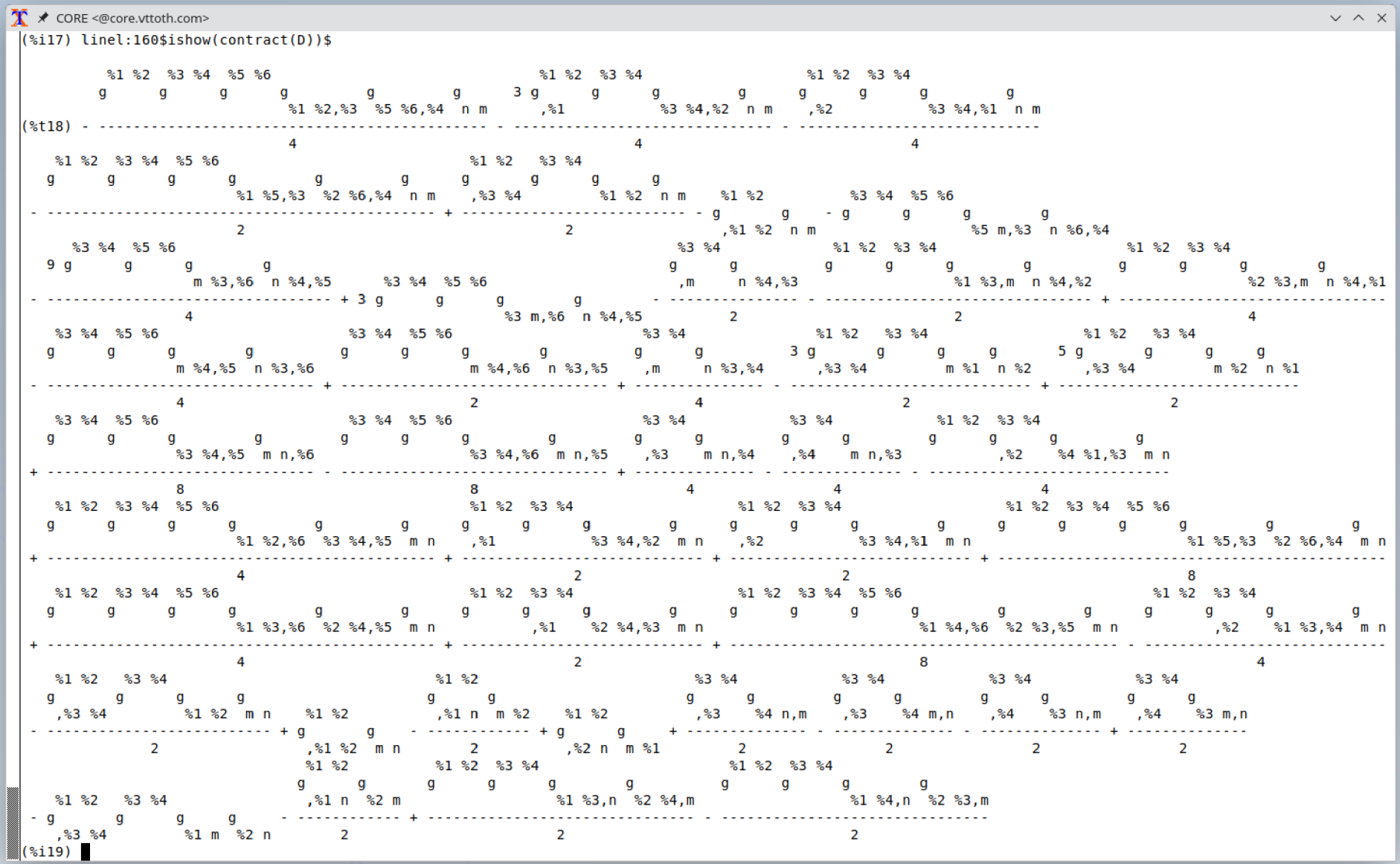
```
load(itensor);
imetric(g);
remcon(g);
defcon(g,g,kdelta);
components(R([a,b,c,d],[]),
          (canform(isimplify(ev(icurvature([a,b,c],[e])*g([d,e],[]),ichr2,eval)))))$
components(R([a,b],[]),(canform(isimplify(ev(icurvature([a,b,c],[c]),ichr2,eval)))))$
components(R([],[]),(canform(isimplify(ev(icurvature([a,b,c],[c])*g([],[a,b]),ichr2,eval)))))$
L0:ishow(1/(16*%pi*G)*((2*L+'R([],[])))*sqrt(-determinant(g)))$
L0:(isimplify(ev(L0,R,eval)))$
EL0:isimplify(diff(L0,g([],[m,n])))$
EL1:isimplify(-idiff(diff(L0,g([],[m,n],k)),k))$
EL2:isimplify(idiff(idiff(diff(L0,g([],[m,n],k,l)),k),l))$
EL:(isimplify(expand((EL0+EL1+EL2)*16*%pi*G/sqrt(-determinant(g)))))$
EFE:ishow('R([m,n],[])-1/2*g([m,n],[])*'R([],[])-L*g([m,n],[]))$
EFE:(isimplify(ev(EFE,R,eval)))$
D:isimplify(contract(isimplify(ev(EL,ichr2,eval))-rename(EFE)))$
```

Nice and clean...

CORE <@core.vttoth.com>

```
Maxima branch_5_49_base_160_g760625d3f https://maxima.sourceforge.io
using Lisp GNU Common Lisp (GCL) GCL 2.7.1 git tag Version_2_7_2ore5
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) load(itensor);

(%o1)          /home/vttoth/dev/maxima/share/tensor/itensor.lisp
(%i2) imetric(g);

(%o2)                                done
(%i3) remcon(g);

(%o3)                                 [g]
(%i4) defcon(g,g,kdelta);

(%o4)                                done
(%i5) components(R([a,b,c,d],[]),(canform(isimplify(ev(icurvature([a,b,c],[e])*g([d,e],[]),ichr2,eval)))))$

(%i6) components(R([a,b],[]),(canform(isimplify(ev(icurvature([a,b,c],[c]),ichr2,eval)))))$

(%i7) components(R([],[]),(canform(isimplify(ev(icurvature([a,b,c],[c])*g([],[a,b]),ichr2,eval)))))$

(%i8) L0:ishow(1/(16*%pi*G)*((2*L+'R([],[])))*sqrt(-determinant(g)))$

                     (2 L + R) sqrt(- determinant(g))
(%t8)                ---------------------------------
                                 16 %pi G
(%i9) L0:(isimplify(ev(L0,R,eval)))$

(%i10) EL0:isimplify(diff(L0,g([],[m,n])))$

(%i11) EL1:isimplify(-idiff(diff(L0,g([],[m,n],k)),k))$

(%i12) EL2:isimplify(idiff(idiff(diff(L0,g([],[m,n],k,l)),k),l))$

(%i13) EL:(isimplify(expand((EL0+EL1+EL2)*16*%pi*G/sqrt(-determinant(g)))))$

(%i14) EFE:ishow('R([m,n],[])-1/2*g([m,n],[])*'R([],[])-L*g([m,n],[]))$

                                          R g
                                             m n
(%t14)                     R    - L g    - ------
                            m n      m n     2
(%i15) EFE:(isimplify(ev(EFE,R,eval)))$

(%i16) D:isimplify(contract(isimplify(ev(EL,ichr2,eval))-rename(EFE)))$

(%i17)
```

But...



What about symmetries?

- In the calculation so far, we assumed nothing about $g_{\mu\nu}$.
- The metric tensor is supposed to be symmetrical:

CORE <@core.vttoth.com>

```
(%i19) linel:80$decsym(g,2,0,[sym(all)],[]);

(%o20)                                done
(%i21) decsym(g,0,2,[],[sym(all)]);

(%o21)                                done
(%i22) ishow(isimplify(contract(isimplify(D))))$

(%t22)                                 0
(%i23)
```

Configuration space

- Of course... the “configuration space” is restricted to symmetric metrics only
- But the variational principle does not “know” this
- We introduce symmetrization “by hand” at some point, whenever convenient
- But this is neither rigorous nor robust

A naive example

- **Object falling in homogeneous gravitational field with lateral freedom of motion:**

$$L=\frac{1}{2}m\left(\dot{x}^2+\dot{y}^2\right)-mgy\,.$$

- **The corresponding Euler-Lagrange equations are straighforward:**

$$m\,\ddot{x}=0\,,$$
$$m\,\ddot{y}=-mg\,.$$

- **But this is free fall.**

A naive example (cont'd)

- **What about sliding down an incline?**
- **We just changed the "configuration space" confining the solution to** $dx = dy \tan\varphi$
- **We could thus modify our equation of motion "by hand":**

$$m\left(1+\tan^2\varphi\right)\ddot{y}=-mg\,.$$

- **But this is not really the "right" way, is it.**

The power of the Lagrangian machinery

- **The textbook solution: add a Lagrange-multiplier!**
- **Doing so "informs" the variational principle about the correct configuration space:**

$$L=\frac{1}{2}m\left(\dot{x}^2+\dot{y}^2\right)-mgy+m\lambda\left(x^2-y^2\tan^2\varphi\right).$$

- **As a result, the right equations of motion are produced automatically:**

$$x=y\tan\varphi\,,$$
$$m\left(1+\tan^2\varphi\right)\ddot{y}=-mg\,.$$

The case of metric symmetry

- The symmetry constraint is $g_{[\mu\nu]} = 0$.
- Conceptually, it is similar to the "inclined plane" constraint in the simple example
- It restricts the "configuration space"
- Why apply this constraint *post hoc*, instead of informing the Lagrangian machinery up front?

Why not just symmetrize?

- We could, of course, use the substitution, $g_{\mu\nu} \Rightarrow g_{(\mu\nu)}$, in the Lagrangian used in a CAS
- This was my first approach, when I treated this as basically just a “software problem”
- Nastier than it sounds: the substitution has to be tracked through the curvature tensor, the integration measure, the connection
- Feels brittle, fragile, *ad hoc* in any case

A Lagrange-multiplier

- **How about following instead the canonical approach?**

$$S=\int d^4x\sqrt{-g}\left(R-2\Lambda+\lambda^{\mu\nu}g_{[\mu\nu]}\right)+S_{\text{matter}}$$

Variation with respect to λ

- **Variation with respect to the nondynamical Lagrange-multiplier is straightforward:**

$$g_{[\mu\nu]}=0.$$

- **Exactly what we wanted in the first place.**

What about the EFE?

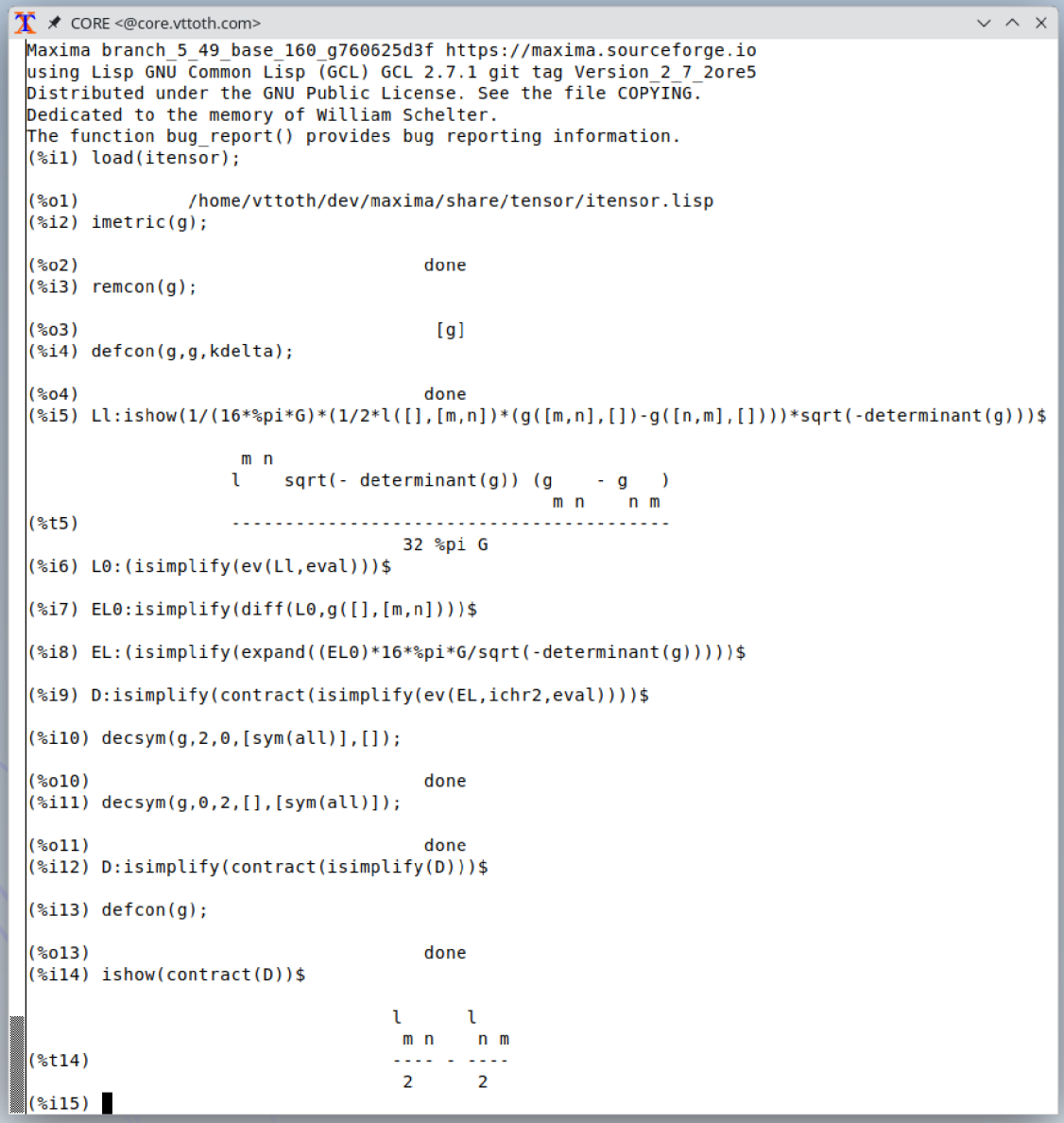
- **This is where things get interesting:**

$$R_{\mu\nu}-\frac{1}{2}R\,g_{\mu\nu}+\Lambda\,g_{\mu\nu}+\lambda_{[\mu\nu]}=8\pi G\,T_{\mu\nu}.$$

- **Given that the first three terms are now manifestly symmetric, we can rewrite this as**

$$R_{\mu\nu}-\frac{1}{2}R\,g_{\mu\nu}+\Lambda\,g_{\mu\nu}=8\pi G\,T_{(\mu\nu)},$$
$$\lambda_{[\mu\nu]}=8\pi G\,T_{[\mu\nu]}.$$

Verification using Maxima



What does it mean?

- If the matter stress-energy tensor is symmetric, nothing.
- However: gravity does not constrain matter to have a symmetric stress-energy tensor.
- In fact, gravity is not at all sensitive to antisymmetric contributions.

But wait...

- We have $T^{\mu\nu} = \left(-2/\sqrt{-g}\right)\delta S_{\text{matter}}/\delta g_{\mu\nu}$
- If $g_{\mu\nu}$ is constrained to be symmetric, doesn't that imply the symmetry of $T^{\mu\nu}$?
- But recall, unless we vary with respect to $g_{(\mu\nu)}$, we vary 16 degrees of freedom independently
- E.g., if the matter action contains the scalar product of two vector fields, the result will not be symmetrical.

Canonical example

- **The conserved angular momentum current is**

$$J^{\lambda\mu\nu} = x^{\mu} T^{\lambda\nu} - x^{\nu} T^{\lambda\mu},$$
$$\nabla_{\lambda} J^{\lambda\mu\nu} = 0.$$

- **If $T^{\mu\nu}$ is conserved, we have**

$$\nabla_{\lambda} J^{\lambda\mu\nu} = 2\, T^{[\mu\nu]} = 0,$$

so the symmetry is assured.

Spin current

- **But the presence of a spin current changes the picture:**

$$J^{\lambda\mu\nu} = x^{\mu} T^{\lambda\nu} - x^{\nu} T^{\lambda\nu} + S^{\lambda\mu\nu}$$

- **Since typically, $\nabla_{\lambda} S^{\lambda\mu\nu} \neq 0$, conservation of angular momentum no longer implies $T^{[\mu\nu]} = 0$.**

Examples

- **Canonical case: spinor field**

$$S^{\lambda\mu\nu} = \frac{1}{2}\overline{\psi}\,\gamma^{\lambda}\Sigma^{\mu\nu}\psi\,.$$

- **Torsion in Einstein-Cartan theory is related, but distinguish between symmetries of the Lagrangian functional vs. the variational degrees of freedom**

Belinfante-Rosenfeld tensor

- The problem is “cured” by introducing a symmetrized stress-energy tensor

$$\Theta^{\mu\nu} = T^{\mu\nu} + \frac{1}{2}\nabla_\lambda\left(S^{\mu\nu\lambda} + S^{\nu\mu\lambda} - S^{\lambda\mu\nu}\right).$$

- This is usually justifed by observing that what remains is precisely the stress-energy tensor of the EFE
- This justification becomes unnecessary in our formalism.

Some concluding thoughts

- A CAS can serve as a powerful consistency checker
- The symmetry of the stress-energy tensor is not enforced by gravity
- It is imposed on the matter side (e.g., isotropic perfect fluid in FLRW)
- Far from this idealized regime (early universe?) there may be tangible implications

Summary

- A “simple” CAS problem: Why is it so hard to reproduce the derivation of the EFE from the EH Lagrangian?
- Post-hoc imposition of metric symmetry works but feels “dirty”
- Introducing the constraint up-front in the Lagrangian is the rigorous approach
- Doing so by way of a Lagrange multiplier leads to interesting results.

References

- Metric symmetry by design in general relativity,
  Viktor T. Toth, arXiv:2412.10607 [gr-qc],
  Class. Quant. Grav. 42 (2025) 027001

- Field theory with the Maxima computer algebra system,
  Viktor T. Toth, arXiv:2308.09837 [cs.SC],
  Int. J. Mod. Phys. C36 (2025) 6, 2450234

- Tensor manipulation in GPL Maxima,
  Viktor T. Toth, arXiv:cs/0503073