Hamilcar: MCP-compliant tools for Hamiltonian analysis

This is the main documentation for the Hamilcar package, which extends xAct to perform canonical field theory calculations in a 3+1 dimensional spacetime decomposition.

Hamilcar is a software package for Wolfram (formerly Mathematica) designed to perform canonical field theory calculations in 3+1 dimensions. The canonical formulation assumes the action to have the standard form with the ingredients:
- The dynamical fields are real tensors on the spatial manifold, which may be a collection of distinct fields, each field having some collection of spatial indices, perhaps with some symmetry among the indices.
- The conjugate momenta are the canonical momenta conjugate to the fields.
- The Hamiltonian density is constructed from the fields, momenta, spatial metric, and spatial derivatives.

The package provides essential functions for canonical field theory:
- "DefCanonicalField": defines canonical field pairs and their conjugate momenta
- "PoissonBracket": computes Poisson brackets between operators using variational derivatives
- "TotalFrom"/"TotalTo": expands expressions to canonical variables and converts back
- "FindAlgebra": determines constraint algebra coefficients using ansatz expressions

Introduction

Hamilcar provides tools for defining canonical field pairs, computing Poisson brackets, and analyzing constraint algebras in field theory.

Installation and Setup:
To install Hamilcar, copy the "xAct/Hamilcar" directory to your "~/.Wolfram/Applications/xAct/" or "~/.Mathematica/Applications/xAct/" directory. In "Mathematica", load the package with: "<<xAct`Hamilcar`".

Pre-defined geometry:
When you first run "<<xAct`Hamilcar`" the software defines a three-dimensional spatial hypersurface with the following geometric objects: spatial coordinate indices (corresponding to adapted coordinates in the ADM prescription); "G[-a,-b]" is the induced metric on the spatial hypersurface; "CD[-a]@" is the spatial covariant derivative; "epsilonG[-a,-b,-c]" is the induced totally antisymmetric tensor on the spatial hypersurface.

Key global variables:
- "$DynamicalMetric": Controls whether spatial metric is treated as dynamical field (default: "True")
- "$ManualSmearing": When "True", disables automatic smearing in Poisson brackets (default: "False")

We first load the Hamilcar package, which provides tools for canonical field theory calculations. The package extends xAct to work with time-dependent fields and their conjugate momenta in a 3+1 dimensional spacetime decomposition.

```
Get["xAct`Hamilcar`"]
```

```
-----------------------------------------------------------------
  ---------------

Package xAct`SymManipulator`  version 0.9.5, {2021, 9, 14}

CopyRight (C) 2011-2021, Thomas Bäckdahl, under the General Public License.
-----------------------------------------------------------------
  ---------------

Package xAct`xPert`  version 1.0.6, {2018, 2, 28}

CopyRight (C) 2005-2020, David Brizuela, Jose M. Martin-Garcia
  and Guillermo A. Mena Marugan, under the General Public License.
```

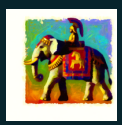
```
** Variable $CovDFormat changed from Prefix to Postfix
** Option AllowUpperDerivatives of ContractMetric changed from False to True
** Option MetricOn of MakeRule changed from None to All
** Option ContractMetrics of MakeRule changed from False to True
```

```
Package xAct`Invar`  version 2.0.5, {2013, 7, 1}
CopyRight (c) 2006-2020, J. M. Martin-Garcia,
  D. Yllanes and R. Portugal, under the General Public License.
** DefConstantSymbol: Defining constant symbol sigma.
** DefConstantSymbol: Defining constant symbol dim.
** Option CurvatureRelations of DefCovD changed from True to False
** Variable $CommuteCovDsOnScalars changed from True to False
```

```
Package xAct`xCoba`  version 0.8.6, {2021, 2, 28}
CopyRight (c) 2005-2021, David Yllanes and
  Jose M. Martin-Garcia, under the General Public License.
```

```
Package xAct`xTras`  version 1.4.2, {2014, 10, 30}
CopyRight (c) 2012-2014, Teake Nutma, under the General Public License.
** Variable $CovDFormat changed from Postfix to Prefix
** Option CurvatureRelations of DefCovD changed from False to True
```

```
Package xAct`Hamilcar` version 0.0.0-developer, {2025, 10, 31}
CopyRight © 2023, Will Barker, under the General Public License.
```



```
These packages come with ABSOLUTELY NO WARRANTY; for details type
  Disclaimer[]. This is free software, and you are welcome to redistribute
  it under certain conditions. See the General Public License for details.
```

Define smearing functions needed for computing Poisson brackets. These functions allow us to evaluate brackets between constraints at different spatial points.

Smearing functions are mathematical tools essential for computing Poisson brackets between field operators at different spatial points. In canonical field theory, constraints like the super-Hamiltonian and super-momentum are density-like objects that need to be integrated over spatial regions. Smearing functions provide the test functions for these integrations, allowing us to compute the algebra between smeared constraints. The scalar smearing functions are used with scalar constraints, while vector smearing functions are used with vector constraints like the super-momentum.

Define scalar smearing functions.

```
DefTensor[ScalarSmearingS[], M3, PrintAs → "S"]
```

s (1)

```
DefTensor[ScalarSmearingF[], M3, PrintAs → "f"]
```

f (2)

Define vector smearing functions with both covariant and contravariant versions.

```
DefTensor[VectorSmearingCovariantS[-i], M3, PrintAs → "S"]
```

s_i (3)

```
DefTensor[VectorSmearingContravariantS[i], M3, PrintAs → "S"]
```

s^i (4)

```
DefTensor[VectorSmearingCovariantF[-i], M3, PrintAs → "f"]
```

f_i (5)

```
DefTensor[VectorSmearingContravariantF[i], M3, PrintAs → "f"]
```

f^i (6)

General relativity

Canonical formulation of general relativity using the ADM decomposition.

This section demonstrates the ADM (Arnowitt-Deser-Misner) formulation of general relativity in canonical field theory. In the ADM formalism, spacetime is decomposed as 3+1 dimensional, with spatial slices evolving in time. The key variables are the spatial metric and its conjugate momentum, related to the extrinsic curvature. The constraints are the super-Hamiltonian (Hamiltonian constraint) and super-momentum (momentum constraint) which generate time evolution and spatial diffeomorphisms respectively.

Verify the canonical Poisson bracket for the metric and its conjugate momentum. Define the trace of the conjugate momentum of the metric.

The spatial metric "G[-a,-b]" and its conjugate momentum "ConjugateMomentumG[a,b]" should satisfy the fundamental canonical Poisson bracket relations. These are the basic building blocks for all subsequent bracket calculations in the ADM formalism.

```
DefTensor[TraceConjugateMomentumG[], M3, PrintAs → "π"];
FromTraceConjugateMomentumG =
 MakeRule[{TraceConjugateMomentumG[], Scalar[ConjugateMomentumG[a, -a]]},
  MetricOn → All, ContractMetrics → True]
```

$\pi \to \pi^{a}{}_{a}$ (7)

Register the expansion rule for the trace of conjugate momentum.

```
PrependTotalFrom[FromTraceConjugateMomentumG]
```

Compute all possible combinations of Poisson brackets between the spatial metric and its conjugate momentum.

The fundamental canonical Poisson brackets provide the foundation for all field theory calculations. We systematically compute brackets between the metric "G" and conjugate momentum "ConjugateMomentumG" with all possible index configurations (raised and lowered indices). This

demonstrates the canonical commutation relations that must be satisfied in the ADM formalism and serves as a verification of the canonical structure.

```
$ManualSmearing = False; Perms = {};
Perms //= Join[#1, Permutations[{1, 1, 1, 1}]] &;
Perms //= Join[#1, Permutations[{1, 1, 1, -1}]] &;
Perms //= Join[#1, Permutations[{1, 1, -1, -1}]] &;
Perms //= Join[#1, Permutations[{1, -1, -1, -1}]] &;
Perms //= Join[#1, Permutations[{-1, -1, -1, -1}]] &;
Indices = (MapThread[Times, {{a, b, c, d}, #1}] &) /@ Perms;
AllConfigurations =
 ({G[#1, #2], ConjugateMomentumG[#3, #4]} &) @@@ Indices;
Expr = (Module[{Expr, Answ}, Expr = #1;
     Answ = (PoissonBracket[#1, #2, Parallel → False] &) @@ #1;
     Expr → Answ] &) /@ AllConfigurations;
```

$$\left\{h^{ab},\ \pi^{cd}\right\} \to -\frac{1}{2}\,\alpha^{ab}\,\beta_{ab} - \frac{1}{2}\,\alpha^{ab}\,\beta_{ba} \tag{8}$$

$$\left\{h^{ab},\ \pi^{c}{}_{d}\right\} \to -\frac{1}{2}\,\alpha^{ab}\,\beta_{ab} - \frac{1}{2}\,\alpha^{ab}\,\beta_{ba} \tag{9}$$

$$\left\{h^{ab},\ \pi_{c}{}^{d}\right\} \to -\frac{1}{2}\,\alpha^{ab}\,\beta_{ab} - \frac{1}{2}\,\alpha^{ab}\,\beta_{ba} \tag{10}$$

$$\left\{\delta_{b}{}^{a},\ \pi^{cd}\right\} \to 0 \tag{11}$$

$$\left\{\delta_{a}{}^{b},\ \pi^{cd}\right\} \to 0 \tag{12}$$

$$\left\{h^{ab},\ \pi_{cd}\right\} \to -\frac{1}{2}\,\alpha^{ab}\,\beta_{ab} - \frac{1}{2}\,\alpha^{ab}\,\beta_{ba} \tag{13}$$

$$\left\{\delta_{b}{}^{a},\ \pi^{c}{}_{d}\right\} \to 0 \tag{14}$$

$$\left\{\delta_{b}{}^{a},\ \pi_{c}{}^{d}\right\} \to 0 \tag{15}$$

$$\left\{\delta_{a}{}^{b},\ \pi^{c}{}_{d}\right\} \to 0 \tag{16}$$

$$\left\{\delta_{a}{}^{b},\ \pi_{c}{}^{d}\right\} \to 0 \tag{17}$$

$$\left\{h_{ab},\ \pi^{cd}\right\} \to \frac{1}{2}\ \alpha^{ab}\ \beta_{ab} + \frac{1}{2}\ \alpha^{ab}\ \beta_{ba} \tag{18}$$

$$\left\{\delta_{b}^{\ a},\ \pi_{cd}\right\} \to 0 \tag{19}$$

$$\left\{\delta_{a}^{\ b},\ \pi_{cd}\right\} \to 0 \tag{20}$$

$$\left\{h_{ab},\ \pi^{c}_{\ d}\right\} \to \frac{1}{2}\ \alpha^{ab}\ \beta_{ab} + \frac{1}{2}\ \alpha^{ab}\ \beta_{ba} \tag{21}$$

$$\left\{h_{ab},\ \pi_{c}^{\ d}\right\} \to \frac{1}{2}\ \alpha^{ab}\ \beta_{ab} + \frac{1}{2}\ \alpha^{ab}\ \beta_{ba} \tag{22}$$

$$\left\{h_{ab},\ \pi_{cd}\right\} \to \frac{1}{2}\ \alpha^{ab}\ \beta_{ab} + \frac{1}{2}\ \alpha^{ab}\ \beta_{ba} \tag{23}$$

Define the lapse function and shift vector of the ADM decomposition.

In the ADM formalism, the lapse function and shift vector describe how the spatial slices are embedded in spacetime. The lapse controls the rate of proper time advance between slices, while the shift describes how the spatial coordinate system moves between slices. These are Lagrange multipliers that enforce the super-Hamiltonian and super-momentum constraints respectively.

```
DefTensor[Lapse[], M3, PrintAs → "𝒩"]
```

$$\mathcal{N} \tag{24}$$

```
DefTensor[Shift[m], M3, PrintAs → "𝒩"]
```

$$\mathcal{N}^{m} \tag{25}$$

Define the gravitational coupling constant.

```
DefConstantSymbol[GravitationalCoupling, PrintAs → "κ"]
```

$$\kappa \tag{26}$$

Define the super-Hamiltonian constraint (Hamiltonian constraint) from the ADM formalism. This uses the trace of the conjugate momentum defined in Eq. (7).

```
DefTensor[SuperHamiltonian[], M3, PrintAs → "ℋ"];
FromSuperHamiltonian = MakeRule[{SuperHamiltonian[],
   (1/((1/GravitationalCoupling^2)*Sqrt[DetG[]]))*(ConjugateMomentumG[i, j]*
       ConjugateMomentumG[-i, -j] - (1/2)*TraceConjugateMomentumG[]^2) -
    (1/GravitationalCoupling^2)*Sqrt[DetG[]]*RicciScalarCD[]},
  MetricOn → All, ContractMetrics → True]
```

$$\mathcal{H} \to \frac{-2\,h\,R[\nabla] + \kappa^4\left(2\,\pi_{ab}\,\pi^{ab} - \pi^2\right)}{2\,\kappa^2\,\sqrt{h}} \tag{27}$$

Register the expansion rule for the super-Hamiltonian constraint.

```
PrependTotalFrom[FromSuperHamiltonian]
```

Define the super-momentum constraint (momentum constraint) from the ADM formalism.

```
DefTensor[SuperMomentum[-i], M3, PrintAs → "ℋ"]; FromSuperMomentum =
 MakeRule[{SuperMomentum[-i], -2*CD[-j][ConjugateMomentumG[j, -i]]},
  MetricOn → All, ContractMetrics → True]
```

$$\mathcal{H}_i \to -2\,\nabla_a \pi_i^a \tag{28}$$

Register the expansion rule for the super-momentum constraint.

```
PrependTotalFrom[FromSuperMomentum]
```

Now we compute the algebra between constraints. This is the Dirac hypersurface deformation algebra of general relativity.

The Dirac hypersurface deformation algebra describes how the constraints of general relativity act on each other under Poisson brackets. This algebra shows that the super-Hamiltonian generates normal deformations of the spatial hypersurface, while the super-momentum generates tangential deformations (spatial diffeomorphisms). The algebra closes in the sense that brackets between constraints produce linear combinations of the same constraints, possibly with structure functions that depend on the smearing functions and their derivatives.

Set manual smearing to true so we can use our defined smearing functions.

```
$ManualSmearing = True
```

Compute the auto-commutator of the super-Hamiltonian using scalar smearing functions.

```
Expr =
 {ScalarSmearingF[] * SuperHamiltonian[], ScalarSmearingS[] * SuperHamiltonian[]}
```

$$\{f\,\mathcal{H}, s\,\mathcal{H}\} \tag{29}$$

```
Expr //= (PoissonBracket[#1, #2, Parallel → True] &) @@ #1 & ;
Expr //= TotalTo
```

$$-2\,\pi_{ab}\,s\,\nabla^b\nabla^a f + 2\,\pi_{ab}\,f\,\nabla^b\nabla^a s \tag{30}$$

Use "FindAlgebra" to express the bracket result in terms of the super-momentum constraint defined in Eq. (28) and spatial derivatives of the smearing functions.

```
Expr //=
  FindAlgebra[#1, {{{SuperMomentum}, {CD, ScalarSmearingF, ScalarSmearingS}}},
    Constraints → {SuperMomentum[i]}, Method → Solve, Verify → True] & ;
```

```
** Verified with respect to the
   effective smearing function ConjugateMomentumG[a, b].
** Verified with respect to the effective smearing function ScalarSmearingF[].
** Verified with respect to the effective smearing function ScalarSmearingS[].
```

$$\mathcal{H}^i\left(-s\,\nabla_i f + f\,\nabla_i s\right) \tag{31}$$

Compute the bracket between the super-Hamiltonian and super-momentum using scalar and contravariant vector smearing.

```
Expr = {ScalarSmearingF[] * SuperHamiltonian[],
   VectorSmearingContravariantS[i] * SuperMomentum[-i]}
```

$$\{f\,\mathcal{H}, \mathcal{H}_i\ s^i\} \tag{32}$$

```
Expr //= (PoissonBracket[#1, #2, Parallel → True] &) @@ #1 & ;
Expr //= TotalTo;
```

$$\frac{2\kappa^2\,\pi^{bc}\,f\,s^a\,\nabla_a\pi_{bc}}{\sqrt{h}}-\frac{\kappa^2\,\pi^b{}_b\,f\,s^a\,\nabla_a\pi^c{}_c}{\sqrt{h}}+\frac{\kappa^2\,\pi_{bc}\,\pi^{bc}\,f\,\nabla_a s^a}{\sqrt{h}}-\frac{\kappa^2\,\pi^b{}_b\,\pi^c{}_c\,f\,\nabla_a s^a}{2\sqrt{h}}-$$
$$\frac{\sqrt{h}\,R[\nabla]\,f\,\nabla_a s^a}{\kappa^2}+\frac{2\sqrt{h}\,\nabla_a s^a\nabla_b\nabla^b f}{\kappa^2}+\frac{2\sqrt{h}\,R[\nabla]_{ab}\,f\,\nabla^b s^a}{\kappa^2}-\frac{\sqrt{h}\,\nabla_a\nabla_b f\,\nabla^b s^a}{\kappa^2}-\frac{\sqrt{h}\,\nabla_b\nabla_a f\,\nabla^b s^a}{\kappa^2}$$

Use "FindAlgebra" to express this bracket in terms of the super-Hamiltonian constraint defined in Eq. (27).

```
Expr //= FindAlgebra[#1,
    {{{SuperHamiltonian}, {CD, ScalarSmearingF, VectorSmearingContravariantS}}},
    Constraints → {SuperHamiltonian[]}, Method → Solve, Verify → True] & ;
```

```
** Verified with respect to the effective smearing function ScalarSmearingF[].
** Verified with respect to the
  effective smearing function VectorSmearingContravariantS[a].
```

$$-\mathcal{H}\,s^a\,\nabla_a f \tag{34}$$

Compute the auto-commutator of the super-momentum using contravariant vector smearing.

```
Expr = {VectorSmearingContravariantF[i] * SuperMomentum[-i],
  VectorSmearingContravariantS[j] * SuperMomentum[-j]}
```

$$\left\{\mathcal{H}_i\,f^i,\,\mathcal{H}_j\,s^j\right\} \tag{35}$$

```
Expr //= (PoissonBracket[#1, #2, Parallel → True] & ) @@ #1 & ;
Expr //= TotalTo;
```

$$2\,\pi_{bc}\,\nabla_a s^c\,\nabla^b f^a+2\,\pi_{ab}\,\nabla^b f^a\,\nabla_c s^c+2\,s^a\,\nabla_a\pi_{bc}\,\nabla^c f^b-2\,\pi_{ac}\,\nabla^b f^a\,\nabla^c s_b-2\,f^a\,\nabla_a\pi_{bc}\,\nabla^c s^b-2\,\pi_{bc}\,\nabla_a f^a\,\nabla^c s^b \tag{36}$$

Use "FindAlgebra" to express this bracket in terms of the super-momentum constraint defined in Eq. (28).

```
Expr //= FindAlgebra[#1, {{{SuperMomentum},
      {CD, VectorSmearingContravariantF, VectorSmearingContravariantS}}},
    Constraints → {SuperMomentum[i]}, Method → Solve, Verify → True] & ;
```

** Verified with respect to the
effective smearing function ConjugateMomentumG[a, b].
** Verified with respect to the
effective smearing function VectorSmearingContravariantF[a].
** Verified with respect to the
effective smearing function VectorSmearingContravariantS[a].

$$\mathcal{H}^i\left(-s^a \nabla_a f_i + f^a \nabla_a s_i\right) \tag{37}$$

Compute the auto-commutator of the super-momentum using covariant vector smearing for comparison with Eq. (37).

```
Expr = {VectorSmearingCovariantF[-i] * SuperMomentum[i],
  VectorSmearingCovariantS[-j] * SuperMomentum[j]}
```

$$\left\{\mathcal{H}^i\ f_i, \mathcal{H}^j\ s_j\right\} \tag{38}$$

```
Expr //= (PoissonBracket[#1, #2, Parallel → True] &) @@ #1 & ;
Expr //= TotalTo;
```

$$\begin{aligned}&2\,\pi_{bc}\,\nabla_a s^c \nabla^b f^a - 2\,s^a\,\nabla_a f^b \nabla_c \pi_b{}^c + 2\,f^a\,\nabla_a s^b \nabla_c \pi_b{}^c - 2\,s^a\,\nabla^b f_a \nabla_c \pi_b{}^c + 2\,f^a\,\nabla^b s_a \nabla_c \pi_b{}^c + \\ &\quad 2\,\pi_{ab}\,\nabla^b f^a \nabla_c s^c + 2\,s^a\,\nabla_a \pi_{bc} \nabla^c f^b - 2\,\pi_{ac}\,\nabla^b f^a \nabla^c s_b - 2\,f^a\,\nabla_a \pi_{bc} \nabla^c s^b - 2\,\pi_{bc}\,\nabla_a f^a \nabla^c s^b\end{aligned} \tag{39}$$

Use "FindAlgebra" to express this bracket in terms of the super-momentum constraint with covariant smearing, comparing to Eq. (37).

```
Expr //= FindAlgebra[#1, {{{SuperMomentum},
      {CD, VectorSmearingCovariantF, VectorSmearingCovariantS}}},
    Constraints → {SuperMomentum[i]}, Method → Solve, Verify → True] & ;
```

** Verified with respect to the
effective smearing function ConjugateMomentumG[a, b].
** Verified with respect to the
effective smearing function VectorSmearingCovariantF[-a].
** Verified with respect to the
effective smearing function VectorSmearingCovariantS[-a].

$$\mathcal{H}^i\left(s^a \nabla_i f_a - f^a \nabla_i s_a\right) \tag{40}$$