Automating one-loop corrections for general models in RECOLA 2.0

Jean-Nicolas Lang

Universität Würzburg
In collaboration with A. Denner and S. Uccirati
LoopFest XV

August 16, 2016

Efforts in NLO automation

FeynArts/ FormCalc	[Hahn and others]
GoSam	[Cullen, Greiner, Heinrich, Luisoni, Mastrolia, Ossola, Reiter, Tramontano]
NLOX	[Reina, Schutzmeier]
MadGraph5 aMC@NLO	[Alwall, Frederix, Frixione, Hirschi, Maltoni, Mattelaer, Shao, Stelzer, Torrielli, Zaro]
NGluon	[Badger, Biedermann, Uwer, Yundin]
OpenLoops	[Cascioli, Maierhöfer, Pozzorini]
BlackHat	[Bern, Dixon, Cordero, Höche, Ita, Kosower, Maitre, Ozeren]
HELAC-NLO	[Bevilacqua, Czakon, Garzelli, van Hameren, Kardos, Papadopoulos, Pittau, Worek]
RECOLA 1.0	[Actis, Denner, Hofer, JNL, Scharf, Uccirati]
...	

NLOCT	[Degrande]
REPT1L	

Content of this Talk

RECOLA 1.0

BSM models in RECOLA 2.0

Automation of rational terms
and renormalization in REPT1L

Results and conclusion

RECOLA 1.0

RECOLA

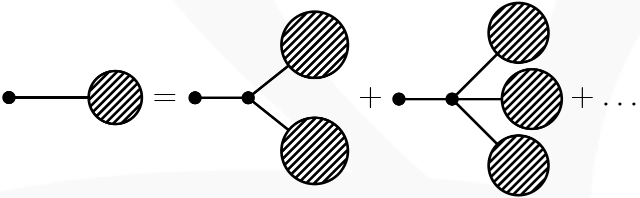
REcursive Computation of One Loop Amplitudes

[Actis, Denner, Hofer, JNL, Scharf, Uccirati]

- Public!
  `https://recola.hepforge.org/`
- Compute any process in the SM at one-loop
  QCD + EW
- Pure Fortran95
- Flexible
  Easily incorporated in monte carlo programs
- Low on memory usage
  Fast and purely numerical

RECOLA algorithm at tree-level

- Off-shell recursion relations
  [Berends Giele '88]

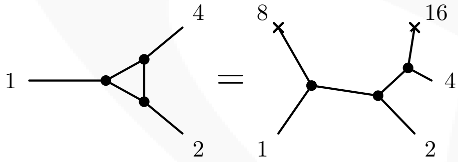


- Off-shell currents represented in binary representation (HELAC)

- Algorithm independent of particle nature

RECOLA algorithm at one-loop order

Algorithm extension to NLO [Van Hameren '09]

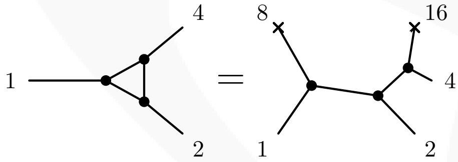


$$i\mathcal{M}_1 = \sum_c c_{\mu_1,\mu_2,\ldots} \; T_c^{\mu_1,\mu_2,\ldots}$$

- Tensor coefficients $c_{\mu_1,\mu_2,\ldots}$ are computed recursively
- Tensor integral T evaluation needs external library (COLLIER [Denner, Dittmaier, Hofer '16])
- Dimensional regularization requires c in D-dim. Include remnants known as rational terms of type R2

RECOLA algorithm at one-loop order

Algorithm extension to NLO [Van Hameren '09]



$$\mathrm{i}\mathcal{M}_1 = \sum_c c_{\mu_1,\mu_2,\dots}\, T_c^{\mu_1,\mu_2,\dots}$$

- Tensor coefficients $c_{\mu_1,\mu_2,\dots}$ are computed recursively
- Tensor integral T evaluation needs external library (COLLIER [Denner, Dittmaier, Hofer '16])
- Dimensional regularization requires c in D-dim. Include remnants known as rational terms of type R2

BSM models in RECOLA 2.0

BSM models in RECOLA 2.0
Framework and Ingredients

RECOLA 2.0

- Generalization of RECOLA done ✓
- Model file support
- Final product is pure Fortran95 ✓

REPT1L

- Derive NLO model file for RECOLA ✓

BSM models in RECOLA 2.0
Framework and Ingredients

RECOLA 2.0

- Generalization of RECOLA done ✓
- Model file support
- Final product is pure Fortran95 ✓

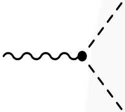
REPT1L

- Derive NLO model file for RECOLA ✓

REPT1L

REnormalization in Python aT 1 Loop

- Starting point: Feynman Rules in UFO Format [Degrande et al. '12]

$\mathcal{L}[\textit{Mathematica}]$ $\underset{\textit{FeynRules}, \textit{SARAH}, \ldots}{\longrightarrow}$  [*UFO*]

[Alloul et al '13] [Staub '12]

REPT1L:

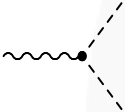
Toolchain in Python, FORM and RECOLA

- Recursive rules for off-shell currents
- Rational terms of type R2
- Renormalization

REPT1L

REnormalization in Python aT 1 Loop

- Starting point: Feynman Rules in UFO Format [Degrande et al. '12]

$\mathcal{L}[\mathit{Mathematica}] \xrightarrow[\text{[Alloul et al '13] [Staub '12]}]{\mathit{FeynRules}, \mathit{SARAH}, \ldots}$  [*UFO*]

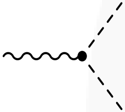
REPT1L:
Toolchain in Python, FORM and RECOLA

- Recursive rules for off-shell currents
- Rational terms of type R2
- Renormalization

REPT1L

REnormalization in Python aT 1 Loop

- Starting point: Feynman Rules in UFO Format [Degrande et al. '12]

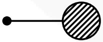
$\mathcal{L}[\textit{Mathematica}] \underset{\substack{\textit{FeynRules},\textit{SARAH},\ldots \\ \text{[Alloul et al '13] [Staub '12]}}}{\longrightarrow}$  $[\textit{UFO}]$

REPT1L:
Toolchain in Python, FORM and RECOLA

- Recursive rules for off-shell currents
- Rational terms of type R2
- Renormalization

Recursive rules for off-shell currents

- Tree currents $w \Rightarrow \mathrm{i}\mathcal{M}_0$
- Loop currents $c \Rightarrow \mathrm{i}\mathcal{M}_1 = \sum_c \sum_r c_r \, T_c^r$

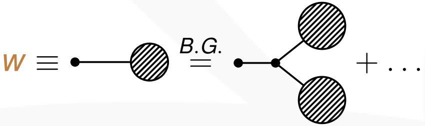
$w \equiv$  $\overset{B.G.}{=} \quad + \dots$

$$\Rightarrow \qquad w_k := \sum_{ij} w_i w_j \times$$

$$\Rightarrow \qquad c_{k,r'} := \sum_{ijr} c_{i,r} w_j \times \Big(\quad \Big)_{rr'}$$

Recursive rules for off-shell currents

- Tree currents $w \Rightarrow \mathrm{i}\mathcal{M}_0$
- Loop currents $c \Rightarrow \mathrm{i}\mathcal{M}_1 = \sum_c \sum_r c_r \; T_c^r$

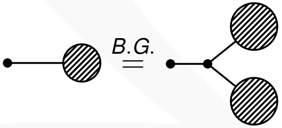


$$\Rightarrow \quad w_k := \sum_{ij} w_i w_j \times$$

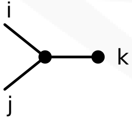
$$\Rightarrow \quad c_{k,r'} := \sum_{ijr} c_{i,r} w_j \times \left(\quad \right)_{rr'}$$

Recursive rules for off-shell currents

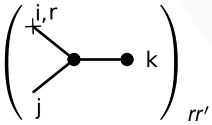
- Tree currents $w \Rightarrow \mathrm{i}\mathcal{M}_0$
- Loop currents $c \Rightarrow \mathrm{i}\mathcal{M}_1 = \sum_c \sum_r c_r\, T_c^r$

$w \equiv$  $+ \ldots$

$$\Rightarrow \quad w_k := \sum_{ij} w_i w_j \times$$



$$\Rightarrow \quad c_{k,r'} := \sum_{ijr} c_{i,r} w_j \times \left(\cdots \right)_{rr'}$$



Off-shell Currents

REPT1L's current library

- Implemented building blocks:
  - $g^{\mu\nu}$, $\epsilon^{\mu\nu\alpha\beta}$, p^{μ}, 1_{4x4}, γ^{μ}, γ_5, $\sigma^{\mu\nu}$
- Any composite structure possible, e.g.:
  - VVV : $p^{\mu} g^{\nu\sigma} - g^{\mu\sigma} p^{\nu}$
  - $FFFF$: $\sigma^{\mu\nu} \sigma_{\nu\mu}$
- Output as:
  - Optimized Fortran code ⇒ numerical evalution
  - FORM expressions ⇒ analytic evalation

Off-shell Currents

REPT1L's current library

- Implemented building blocks:
  - $g^{\mu\nu}$, $\epsilon^{\mu\nu\alpha\beta}$, p^{μ}, 1_{4x4}, γ^{μ}, γ_5, $\sigma^{\mu\nu}$
- Any composite structure possible, e.g.:
  - VVV : $p^{\mu} g^{\nu\sigma} - g^{\mu\sigma} p^{\nu}$
  - $FFFF$: $\sigma^{\mu\nu}\sigma_{\nu\mu}$
- Output as:
  - Optimized Fortran code ⇒ numerical evalution
  - FORM expressions ⇒ analytic evalation

Off-shell Currents

REPT1L's current library

- Implemented building blocks:
  - $g^{\mu\nu}$, $\epsilon^{\mu\nu\alpha\beta}$, p^{μ}, 1_{4x4}, γ^{μ}, γ_5, $\sigma^{\mu\nu}$
- Any composite structure possible, e.g.:
  - $VVV \quad : p^{\mu} g^{\nu\sigma} - g^{\mu\sigma} p^{\nu}$
  - $FFFF : \sigma^{\mu\nu} \sigma_{\nu\mu}$
- Output as:
  - Optimized Fortran code $\Rightarrow$ numerical evalution
  - FORM expressions $\Rightarrow$ analytic evalation

Automation of rational terms and renormalization in REPT1L

Rational terms

Computation of R2 [Draggiotis, Garzelli, Papadopoulos, Pittau '09]

Step 1 Compute pole part of tensor integrals T

$$P.P. \int \mathrm{d}^n q \frac{q^\mu q^\nu}{D(q+p)D(q)} = \frac{i\pi^2}{6\epsilon} p^2 g^{\mu\nu}$$

Step 2 Compute c_ϵ part ($\epsilon = d-4$) of tensor coefficients c

$$\begin{aligned} g^{\mu\nu} &\equiv \hat{g}^{\mu\nu} + \tilde{g}^{\mu\nu}, \\ \hat{g}^{\mu\nu} &\equiv \mathrm{diag}\,(1,-1,-1,-1) \oplus \mathbf{0}^{d-4}, \\ \tilde{\gamma}^\mu &\equiv \tilde{g}^{\mu\nu}\gamma_\nu, \\ &\dots \end{aligned}$$

Step 3 $R2 = c_\epsilon \times T|_{P.P.}$

Step 4 Repeat for all possible c.

Rational terms

Computation of R2 [Draggiotis, Garzelli, Papadopoulos, Pittau '09]

Step 1 Compute pole part of tensor integrals T

$$P.P. \int \mathrm{d}^n q \frac{q^\mu q^\nu}{D(q+p)D(q)} = \frac{i\pi^2}{6\epsilon} p^2 g^{\mu\nu}$$

Step 2 Compute c_ϵ part ($\epsilon = d - 4$) of tensor coefficients c

$$\begin{aligned} g^{\mu\nu} &\equiv \hat{g}^{\mu\nu} + \tilde{g}^{\mu\nu}, \\ \hat{g}^{\mu\nu} &\equiv \mathrm{diag}\,(1, -1, -1, -1) \oplus \mathbf{0}^{d-4}, \\ \tilde{\gamma}^{\mu} &\equiv \tilde{g}^{\mu\nu} \gamma_\nu, \\ &\ldots \end{aligned}$$

Step 3 $R2 = c_\epsilon \times T|_{P.P.}$

Step 4 Repeat for all possible c.

Rational terms

Computation of R2 [Draggiotis, Garzelli, Papadopoulos, Pittau '09]

Step 1 Compute pole part of tensor integrals T

$$P.P. \int \mathrm{d}^n q \frac{q^\mu q^\nu}{D(q+p)D(q)} = \frac{i\pi^2}{6\epsilon} p^2 g^{\mu\nu}$$

Step 2 Compute c_ϵ part ($\epsilon = d - 4$) of tensor coefficients c

$$\begin{aligned} g^{\mu\nu} &\equiv \hat{g}^{\mu\nu} + \tilde{g}^{\mu\nu}, \\ \hat{g}^{\mu\nu} &\equiv \mathrm{diag}\,(1, -1, -1, -1) \oplus \mathbf{0}^{d-4}, \\ \tilde{\gamma}^{\mu} &\equiv \tilde{g}^{\mu\nu}\gamma_\nu, \\ &\ldots \end{aligned}$$

Step 3 $R2 = c_\epsilon \times T|_{P.P.}$

Step 4 Repeat for all possible c.

Rational terms

Computation of R2 [Draggiotis, Garzelli, Papadopoulos, Pittau '09]

Step 1 Compute pole part of tensor integrals T

$$P.P. \int \mathrm{d}^n q \frac{q^\mu q^\nu}{D(q+p)D(q)} = \frac{i\pi^2}{6\epsilon} p^2 g^{\mu\nu}$$

Step 2 Compute c_ϵ part ($\epsilon = d - 4$) of tensor coefficients c

$$\begin{aligned} g^{\mu\nu} &\equiv \hat{g}^{\mu\nu} + \tilde{g}^{\mu\nu}, \\ \hat{g}^{\mu\nu} &\equiv \mathrm{diag}\,(1, -1, -1, -1) \oplus \mathbf{0}^{d-4}, \\ \tilde{\gamma}^\mu &\equiv \tilde{g}^{\mu\nu} \gamma_\nu, \\ &\ldots \end{aligned}$$

Step 3 $R2 = c_\epsilon \times T|_{P.P.}$.

Step 4 Repeat for all possible c.

Rational terms

REPT1L's features in computing R2

- Automated iteration over all possible contributions
- Selection of specific contributions
- Power counting for renormalizable theories
- Not restricted to renormalizable theories
- Fully parallelized

Renormalization

Step 1: Derive counterterms

Step 2: Setting up and solving renormalization conditions

Renormalization

Step 1: Derive counterterms

REPT1L's autoct tools

- Automated derivation of counterterms.
  User needs to provide expansion rules, e.g.:
  $g \to g + \delta g$

- Wavefunction and mass counterterm can be automatically assigned:
  $$\Phi_{0,i} = \sum_j Z_{ij}\Phi_j, \quad m_0 = m + \delta m_R$$

- Chain rule for parameter dependencies and couplings.

- Support for adding counterterms by hand.

Renormalization

Step 1: Derive counterterms

REPT1L's autoct tools

- Automated derivation of counterterms.
  User needs to provide expansion rules, e.g.:
  $g \to g + \delta g$

- Wavefunction and mass counterterm can be automatically assigned:
  $$\Phi_{0,i} = \sum_j Z_{ij}\Phi_j, \quad m_0 = m + \delta m_R$$

- Chain rule for parameter dependencies and couplings.

- Support for adding counterterms by hand.

Renormalization

Step 2: Setting up and solving renormalization conditions

Predefined renormalization conditions

- On-shell/$\overline{\mathrm{MS}}$/MOM renormalization for 2-point functions
- $\overline{\mathrm{MS}}$ renormalization for *n*-point functions
- α_0, G_F scheme for EW, fixed flavor scheme for QCD

Individual renormalization conditions

- Setup renormalization conditions in Python
- Full access to analytic 1PI expressions
- Compute form factors, e.g. Σ_{T} in $\Sigma^{\mu\nu} = \Sigma_{\mathrm{T}} P_{\mathrm{T}}^{\mu\nu} + \Sigma_{\mathrm{L}} P_{\mathrm{L}}^{\mu\nu}$

Renormalization

Step 2: Setting up and solving renormalization conditions

Predefined renormalization conditions

- On-shell/$\overline{\mathrm{MS}}$/MOM renormalization for 2-point functions
- $\overline{\mathrm{MS}}$ renormalization for *n*-point functions
- α_0, G_F scheme for EW, fixed flavor scheme for QCD

Individual renormalization conditions

- Setup renormalization conditions in Python
- Full access to analytic 1PI expressions
- Compute form factors, e.g. Σ_{T} in $\Sigma^{\mu\nu} = \Sigma_{\mathrm{T}} P_{\mathrm{T}}^{\mu\nu} + \Sigma_{\mathrm{L}} P_{\mathrm{L}}^{\mu\nu}$

Renormalization

Example: Two-Higgs-Doublet Model [1607.07352 Denner, Jenniches, JNL, Sturm]

CP-conserving 2HDM with (softly broken) Z_2 symmetry

$$\mathcal{L}_{\mathrm{Higgs}} = (D^\mu \Phi_1)^\dagger D_\mu \Phi_1 + (D^\mu \Phi_2)^\dagger D_\mu \Phi_2 - V$$

New parameters

$$M_{H_\mathrm{l}}, M_{H_\mathrm{h}}, M_{H_\mathrm{a}}, M_{H^\pm}, \alpha, \beta, M_{\mathrm{sb}}$$

- On-shell renormalization for all particles, fixing mass and (mixing-) wave-function counterterms
- $\overline{\mathrm{MS}}$ renormalization of $\alpha, \beta, M_{\mathrm{sb}}$
- Consistent renormalization of tadpoles $\hat{\mathrm{T}}_{H_\mathrm{l}}, \hat{\mathrm{T}}_{H_\mathrm{h}}$ (see [1607.07352] for details)

Renormalization

Example: Two-Higgs-Doublet Model [1607.07352 Denner, Jenniches, JNL, Sturm]

CP-conserving 2HDM with (softly broken) Z_2 symmetry

$$\mathcal{L}_{\mathrm{Higgs}} = (D^\mu \Phi_1)^\dagger D_\mu \Phi_1 + (D^\mu \Phi_2)^\dagger D_\mu \Phi_2 - V$$

New parameters

$$M_{H_\mathrm{l}}, M_{H_\mathrm{h}}, M_{H_\mathrm{a}}, M_{H^\pm}, \alpha, \beta, M_{\mathrm{sb}}$$

- On-shell renormalization for all particles, fixing mass and (mixing-) wave-function counterterms
- $\overline{\mathrm{MS}}$ renormalization of $\alpha, \beta, M_{\mathrm{sb}}$
- Consistent renormalization of tadpoles $\hat{\mathrm{T}}_{H_\mathrm{l}}, \hat{\mathrm{T}}_{H_\mathrm{h}}$ (see [1607.07352] for details)

Renormalization

Example: Two-Higgs-Doublet Model [1607.07352 Denner, Jenniches, JNL, Sturm]

CP-conserving 2HDM with (softly broken) Z_2 symmetry

$$\mathcal{L}_{\mathrm{Higgs}} = \left(D^\mu \Phi_1\right)^\dagger D_\mu \Phi_1 + \left(D^\mu \Phi_2\right)^\dagger D_\mu \Phi_2 - V$$

New parameters

$$M_{H_\mathrm{l}}, M_{H_\mathrm{h}}, M_{H_\mathrm{a}}, M_{H^\pm}, \alpha, \beta, M_{\mathrm{sb}}$$

- On-shell renormalization for all particles, fixing mass and (mixing-) wave-function counterterms
- $\overline{\mathrm{MS}}$ renormalization of $\alpha, \beta, M_{\mathrm{sb}}$
- Consistent renormalization of tadpoles $\hat{\mathrm{T}}_{H_\mathrm{l}}, \hat{\mathrm{T}}_{H_\mathrm{h}}$ (see [1607.07352] for details)

Renormalization

Example: Two-Higgs-Doublet Model [1607.07352 Denner, Jenniches, JNL, Sturm]

CP-conserving 2HDM with (softly broken) Z_2 symmetry

$$\mathcal{L}_{\mathrm{Higgs}} = (D^\mu \Phi_1)^\dagger D_\mu \Phi_1 + (D^\mu \Phi_2)^\dagger D_\mu \Phi_2 - V$$

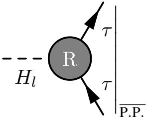
New parameters

$$M_{H_\mathrm{l}}, M_{H_\mathrm{h}}, M_{H_\mathrm{a}}, M_{H^\pm}, \alpha, \beta, M_{\mathrm{sb}}$$

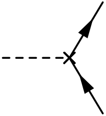
- On-shell renormalization for all particles, fixing mass and (mixing-) wave-function counterterms
- $\overline{\mathrm{MS}}$ renormalization of $\alpha, \beta, M_{\mathrm{sb}}$
- Consistent renormalization of tadpoles $\hat{\mathrm{T}}_{H_\mathrm{l}}, \hat{\mathrm{T}}_{H_\mathrm{h}}$ (see [1607.07352] for details)

Renormalization

Example: $\delta\alpha$ in the 2HDM



$$= 0$$



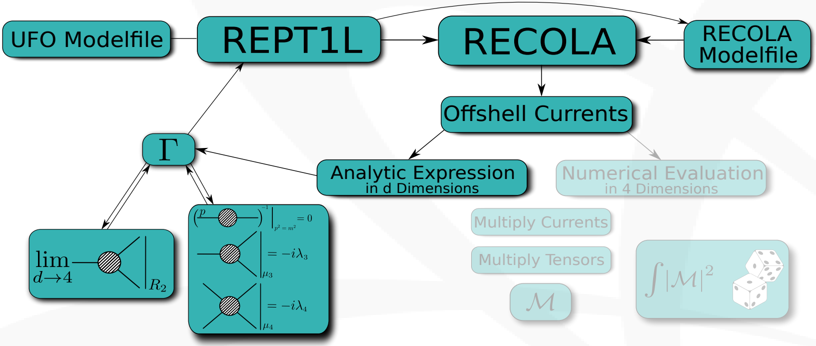
$$= \frac{\mathrm{i}em_\tau}{2m_\mathrm{W}s_\mathrm{w}}\Bigg[(s_{\alpha\beta}+c_{\alpha\beta}t_\beta)\left(\frac{\delta m_\tau}{m_\tau}+\delta Z_e+\frac{c_\mathrm{w}^2-s_\mathrm{w}^2}{(2s_\mathrm{w}^2)}\frac{\delta m_\mathrm{W}^2}{m_\mathrm{W}^2}-\frac{c_\mathrm{w}^2}{2s_\mathrm{w}^2}\frac{\delta m_\mathrm{Z}^2}{m_\mathrm{Z}^2}+t_\beta\delta\beta\right)$$

$$+(c_{\alpha\beta}-s_{\alpha\beta}t_\beta)(\delta\alpha-\delta Z_{H_\mathrm{h}H_\mathrm{l}})\Bigg]$$

```
def renormalize_2HDM_alpha():
  vertex = find_vertex('h1', 'ta+', 'ta-')
  otherct = get_ct(vertex)
  otherct.remove('da')
  da = RenormalizeVertex(vertex,
                         renoscheme='MS',
                         ct='da',
                         reno_sols=otherct)
```

Complete Toolchain

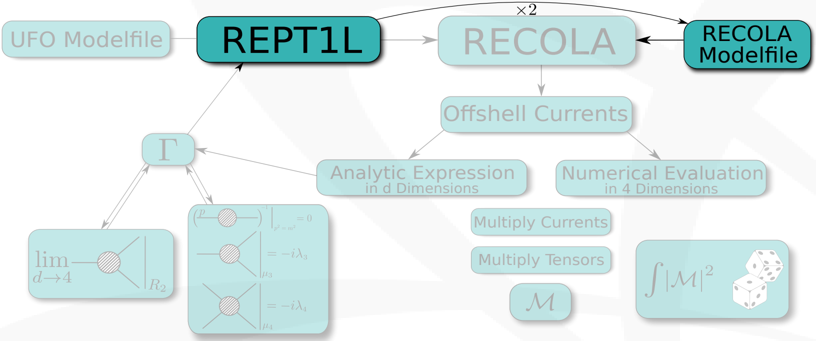
Example: Two-Higgs-Doublet Model




```
# Example THDM
export REPTIL_MODEL_PATH=PATH_TO_UFO_MODEL
./run_model --cct OUTPUT_PATH
./renormalize_qcd
./renormalize_gsw --GFermi
./renormalize_thdm
./run_r2
```

Complete Toolchain

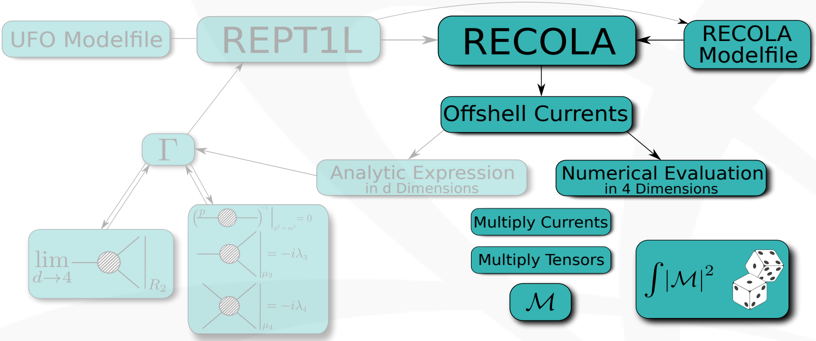
Example: Two-Higgs-Doublet Model




```
# Example THDM
./run_model --cct --cr2 --src OUTPUT_PATH
```

Complete Toolchain

Example: Two-Higgs-Doublet Model



Renormalization

Validation in renormalization

- Separate UV and $\overline{\mathrm{MS}}$ scales
  Numerical check for UV finiteness
- Background Field Method
  R_ξ-gauge
- Consistency checks for onshell renormalization

Further features in renormalization

- Support for switching renormalization schemes
- Light fermions in mass or dimensional regularization
- Soon: Renormalization of effective operators
  (SM D=6 underway)

Renormalization

Validation in renormalization

- Separate UV and $\overline{\mathrm{MS}}$ scales
  Numerical check for UV finiteness
- Background Field Method
  R_ξ-gauge
- Consistency checks for onshell renormalization

Further features in renormalization

- Support for switching renormalization schemes
- Light fermions in mass or dimensional regularization
- Soon: Renormalization of effective operators
  (SM D=6 underway)

Results and conclusion

Results

System successfully applied to:

- Standard Model (diag. CKM) + BFM + R_ξ ($W^\pm, Z$)
- Two-Higgs Doublet Model + BFM + R_ξ ($W^\pm, Z$)
- Toy theories (Φ^8, $\bar{\psi}\gamma_\mu\psi\bar{\psi}\gamma^\mu\psi, \ldots$)

Restrictions

- Spin 0, 1/2 and 1, Majorana fermions underway

Performance:

- Renormalization of the SM/2HDM $\approx$ 30-45min
- Complete set of R2 in SM/2HDM $\geq$ 30min,45min

Results

System successfully applied to:

- Standard Model (diag. CKM) + BFM + R_ξ ($W^\pm, Z$)
- Two-Higgs Doublet Model + BFM + R_ξ ($W^\pm, Z$)
- Toy theories (Φ^8, $\bar{\psi}\gamma_\mu\psi\bar{\psi}\gamma^\mu\psi, \ldots$)

Restrictions

- Spin $0, 1/2$ and 1, Majorana fermions underway

Performance:

- Renormalization of the SM/2HDM $\approx$ 30-45min
- Complete set of R2 in SM/2HDM $\geq$ 30min,45min

Results

System successfully applied to:

- Standard Model (diag. CKM) + BFM + R_ξ ($W^\pm, Z$)
- Two-Higgs Doublet Model + BFM + R_ξ ($W^\pm, Z$)
- Toy theories (Φ^8, $\bar{\psi}\gamma_\mu\psi\bar{\psi}\gamma^\mu\psi, \ldots$)

Restrictions

- Spin $0, 1/2$ and 1, Majorana fermions underway

Performance:

- Renormalization of the SM/2HDM $\approx$ 30-45min
- Complete set of R2 in SM/2HDM $\geq$ 30min,45min

Summary

- RECOLA 2.0 is a high performance one-loop matrix-element generator
- BSM model files
- REPT1L generates one-loop model files from bare UFO model files
- Renormalization automated
  Predefined renormalization conditions
- Results for a gauge-independent renormalization in the 2HDM and beyond
- Soon: NLO corrections to vector-boson fusion Higgs and Higgs-strahlung in the 2HDM

Backup slides

Consistent tadpole renormalization

$\langle \phi \rangle_0 = 0$ at tree-level

- Solution v_0 through potential extremum condition
- v_0 given in terms of bare parameters
  $\Rightarrow$ Gauge independent ✓

$\langle \phi \rangle = 0$ beyond tree-level

- The proper vev v is gauge-dependent
- v potentially enters the definition of physical bare parameters ⚠

Step 1 Define physical bare parameter by bare parameters (v_0 allowed, v not allowed). Include tadpoles in calculation.

Step 2 Get rid of the tadpoles without modifying the theory.

Consistent tadpole renormalization

$\langle\phi\rangle_0 = 0$ at tree-level

- Solution v_0 through potential extremum condition
- v_0 given in terms of bare parameters
  $\Rightarrow$ Gauge independent ✓

$\langle\phi\rangle = 0$ beyond tree-level

- The proper vev v is gauge-dependent
- v potentially enters the definition of physical bare parameters ⚠

Step 1 Define physical bare parameter by bare parameters (v_0 allowed, v not allowed). Include tadpoles in calculation.

Step 2 Get rid of the tadpoles without modifying the theory.

The FJ Tadpole Scheme

Consistent renormalization of tadpoles [Fleischer Jegerlehner '81] and generalization thereof [1607.07352]

- Renormalize the tadpoles via:

$$\phi(x) \to \phi(x) + \Delta v \quad \text{or} \quad v_0 \to v_0 + \Delta v$$

- Relate Δv to the tadpole counterterm $\delta t(\Delta v)$
- Choose Δv such that $\delta t = -\mathrm{T}$
- $\langle\phi\rangle = 0$ ✓

The FJ Tadpole Scheme

Different tadpole schemes

- Technically, the schemes differ in the way the tadpole counterterms are introduced.
- Problem: Tadpoles are accidentally absorbed in bare physical parameters
  0709.1075 (SM),
  hep-ph/9206257, hep-ph/0207010, 0807.4668, ... (MSSM),
  hep-ph/9701257,hep-ph/0408364,... (2HDM)
- Observation:
  Schemes indistinguishable when all parameters are renormalized at fixed points in momentum space (e.g. on-shell, MOM).
- MS or $\overline{\mathrm{MS}}$ is sensitive to the specific scheme and S-matrix potentially becomes gauge-dependent.

The FJ Tadpole scheme

Why choose the FJ tadpole scheme? [1607.07352]

- Theory is independent of Δv_i:
  $\hat{\mathrm{T}}_i = 0$ is equivalent to $\delta t_i = 0$ in general.
- No tadpoles are absorbed into the definition of physical bare parameters.
- Counterterms associated to physical parameters are gauge independent.
- *S*-Matrix is gauge independent
- In the 'standard schemes' the renormalization of β is gauge-dependent already at one-loop order (applies to the MSSM and THDM).

Current optimizations

Colourflow representation

- $G^a \hat{=} G^i_j \quad \Leftrightarrow \quad 8 \oplus 1 = 3 \otimes \bar{3}$
- UFO vertices automatically transformed to colourflow vertices

Helicity conservation

- Automatically derives helicity conservation rules for any current

Massless Fermion loops

- Avoid computing equal fermion loops (only for SM like theories, CKM diagonal)

Current optimizations

Colourflow representation

- $G^a \hat{=} G^i_j \quad \Leftrightarrow \quad 8 \oplus 1 = 3 \otimes \bar{3}$
- UFO vertices automatically transformed to colourflow vertices

Helicity conservation

- Automatically derives helicity conservation rules for any current

Massless Fermion loops

- Avoid computing equal fermion loops (only for SM like theories, CKM diagonal)

Current optimizations

Colourflow representation

- $G^a \hat{=} G^i_j \quad \Leftrightarrow \quad 8 \oplus 1 = 3 \otimes \bar{3}$
- UFO vertices automatically transformed to colourflow vertices

Helicity conservation

- Automatically derives helicity conservation rules for any current

Massless Fermion loops

- Avoid computing equal fermion loops (only for SM like theories, CKM diagonal)

Results

Testing and validation

- Validated against RECOLA 1.0, OpenLoops, Madgraph for the SM
- Renormalization validated in the 2HDM with L. Jenniches (Würzburg)
- Validation of $H \to 4f$ in the 2HDM with L. Altenkamp (Freiburg)
- REPT1L equipped with unittests and doctests
- Complete testing routine for the SM and 2HDM

Rational terms

Limitations in computing R2

- Pole parts for n-point tensor integrals implemented up to rank $n+2$ for $n=4,5,6$.
- NDR-scheme
- Missing rules for open fermion lines in eff. field theory, e.g.:

$$\lim_{d\to 4}\left(\sigma^{\mu\nu}\right)_{ij}\left(\sigma_{\nu\mu}\right)_{kl} = \left(\hat{\sigma}^{\mu\nu}\right)_{ij}\left(\hat{\sigma}_{\nu\mu}\right)_{kl} + \mathcal{O}\left(d-4\right)$$