Overview Of Meta-Tools

Tools to make tools

(plus a few tools to check things along the way)

Ben O'Leary

Julius-Maximilians-Universität Würzburg

Post-Planck Bethe Forum, BCTP Bonn, May 29th, 2013

Tools to make tools

Introduction

I'm going to talk about some *very specific* meta-tools:

- `SARAH`
- `FeynRules` (+ `ASperGe`)
- `LanHEP`
- `Susyno`
- `PyR@te`

These are meta-tools in the sense that they generate code to assist in turning parameter points of models into observables to compare against experiment.

Introduction, continued

I will also mention some other supplementary tools concerned with getting SSB correct:

- `ScannerS`
- `Vscape`
- `CosmoTransitions`
- `Vevacious`– you should definitely use this!

Also I will mention tools that can be adapted to solving SSB:

- `HOM4PS2`
- `paramotopy`
- `StringVacua`

The main point though is that you should use `Vevacious`.

Meta-tools for Lagrangian to Feynman rules

Idea → observables

- human brain(s): idea → QFT as fields + symmetries
- meta-tools / student: fields + symmetries $(\rightarrow \mathcal{W}) \rightarrow \mathcal{L}$
- meta-tools: $\mathcal{L} \rightarrow$ Feynman rules (as understood by other codes)
- MC simulators / other tools: Feynman rules + $\mathcal{L}$ parameter values → observables

Meta-tools for $\mathcal{L} \to$ Feynman rules

`SARAH`, `FeynRules`, and `LanHEP` automate $\mathcal{L} \to$ Feynman rules.

- Feynman rules should come from expansion around minimum of potential.
- SSB not easy to automate (especially with extended gauge groups).
- Usual R_ξ gauges require knowledge of shift to minimum.

Meta-tools for RGEs

Parameters also may need to be run from one scale to another; the derivation of renormalization group equations from $\mathcal{L}$ can be automated.

- `SARAH` creates `SPheno` with 2-loop SUSY RGEs
- `Susyno` creates 2-loop SUSY β functions in `Mathematica`
- `PyR@te` will create 2-loop non-SUSY β functions in `Python` and `C++`

Meta-tool inputs

Meta-tool	written in	required input
SARAH	Mathematica	SUSY: $\mathcal{G}$, $\Phi_i^{\mathcal{G}}$, $\mathcal{W}$, SSB, Φ_i^m
		non-SUSY: $\mathcal{G}$, $\Phi_i^{\mathcal{G}}$, V, SSB, Φ_i^m
FeynRules	Mathematica	SUSY: $\mathcal{G}$, $\Phi_i^{\mathcal{G}}$, $\mathcal{W}$, $\mathcal{L}_{\not{\text{SUSY}}}$, SSB, ξ, Φ_i^m
		non-SUSY: $\mathcal{G}$, $\Phi_i^{\mathcal{G}}$, V, SSB, ξ, Φ_i^m
LanHEP	C	$\mathcal{L}$ (Φ_i^m) (can use $\mathcal{W} \to V$), SSB, ξ
Susyno	Mathematica	$\mathcal{G}$, $\Phi_i^{\mathcal{G}}$, $\mathbf{Z}$
PyR@te	Python	$\mathcal{G}$, $\Phi_i^{\mathcal{G}}$, V

$\mathcal{G}$: gauge group
$\Phi_i^{\mathcal{G}}$: gauge eigenstates
$\mathcal{L}$: Lagrangian
ξ: gauge-fixing
$\mathcal{W}$: superpotential

$\mathbf{Z}$: global symmetries
Φ_i^m: mass eigenstates
V: potential (including Yukawa terms)
SSB: spontaneous symmetry breaking
$\mathcal{L}_{\not{\text{SUSY}}}$: soft SUSY-breaking terms

Meta-tool outputs

Meta-tool	$\mathcal{L}$	RGEs	spectrum	UFO *etc.*
SARAH	LaTeX	SPheno	SPheno (NLO)	✔
FeynRules	LaTeX	✗	ASperGe (LO)	✔
LanHEP	LaTeX	✗	✗	✔
Susyno	ugly	Mathematica	✗	✗
PyR@te	LaTeX	LaTeX, Python, C++	✗	✗

MC simulator support (all support UFO):

Meta-tool	FeynArts /FORMCalc	CalcHEP /CompHEP	WHIZARD /O'Mega	MadGraph /MadEvent	Sherpa
SARAH	✔	✔	✔	through UFO	✗
FeynRules	✔	✔	✔	✔	✔
LanHEP	✔	✔	✗	through UFO	✗

Tools for spontaneous symmetry breaking

Getting desired SSB minima

- Relatively easy to engineer tree-level V with extremum at desired VEVs.
- Saddle points / maxima can be rejected by presence of tachyonic scalar masses-squared.
- `ScannerS`: automatic solution of $\mathcal{L}$ parameters to get desired VEVs, + calculates mass eigenstates for scalars.
  - Advertized as finding all minima of potentials, fine print shows that it doesn't do that yet.
- `Vscape`: automatic generation of 1-loop V from $\mathcal{W}$, finds minimum near given start points.
  - Doesn't do vector superfield contributions yet.

Metastability

- Desired VEVs don't have to be global minimum if false vacuum is long-lived enough.
- $\Gamma/\text{vol.} = A\exp(-B)$, should be $\lesssim (13\text{ Gyr})^{-4}$
- `CosmoTransitions`: calculates B for given $V+$ false vacuum VEVs + true vacuum VEVs, also does finite T, can calculate critical T and phases.
- A usually estimated on dimensional grounds, not so important, as $B \approx 400$ for $\tau \approx 13$ Gyr $(100\text{ GeV}/A^{1/4})$: 1% change in $B \to$ factor 2.8 change in τ

Vevacious

`Vevacious`: code to check V for *global* minimum.

- Guaranteed to find *all* tree-level extrema for given V!
  - Uses homotopy continuation via `HOM4PS2`.
  - Alternative homotopy continuation codes include `Bertini`, `paramotopy`.
  - Generally considered faster than Gröbner bases (implemented in `StringVacua`, for example).
- Tree-level extrema used as starting points for `MINUIT` (through `PyMinuit`) for 1-loop potential.
- If input VEVs correspond to false vacuum, `CosmoTransitions` is called to calculate tunneling time.
- Quick: typically a few seconds for 4 to 6 "VEVing" scalars.
- Takes SLHA file as input.

Available now(-ish) from HepForge!

Summary

Summary

There are several HEP meta-tools to make your lives easier:

- Meta-tools take over from the stage of having written down $\mathcal{L}$.
- Some meta-tools even find $\mathcal{L}$ for you.
- There are meta-tools to run $\mathcal{L}$ parameters from one Q to another.
- SSB still needs to be put in by hand to some extent.
- SSB can be automatically checked, at least.

Thank you for your attention!

Backup slides

Just trust me, OK?