Computer tools for EFTs

J. C. Criado (NICPB, Tallinn)

June 16, 2020

Basis

- Generation — BasisGen
- Change — rosetta, wilson
- Reduction

abc_eft*, DEFT

MatchingTools

CoDEx, MatchMaker*

- Matching
- Running — DsixTools, wilson

- Feynman rules — SmeftFR

Observable

- Calculator — EOS, flavio, FlavorKit, FormFlavor, SMEFTsim, SPheno
- Fit — HEPfit, smelli, SMEFiT

1910.11003 (I. Brivio et al.) [SMEFT-Tools 2019]

Connecting tools: `WCxf`

"Data exchange format to interface different tools in particle physics that deal with Wilson coefficients beyond the SM"

1712.05298 (J. Aebischer et al.)

Implemented in:

`DsixTools`, `EOS`, `flavio`, `FlavorKit`, `FormFlavor`, `wilson`, `SmeftFR`, `SMEFTsim`, `smelli`, `SPheno`, `wcxf`, (`MatchingTools?`)

Matching/running codes

	Matching	Running
`DsixTools`	tree-level, d6 SMEFT – d6 WET	SMEFT, WET
`MatchingTools`	tree-level, any fields, any dim.	no
`wilson`	tree-level, d6 SMEFT – d6 WET	SMEFT, WET
`CoDEx`	1-(heavy-)loop, BSM – d6 SMEFT	SMEFT
`MatchMaker*`	1-loop, BSM – d6 SMEFT	no

BSM – d6 SMEFT tree-level dictionary

1711.10391 (J. de Blas, JCC, M. Pérez-Victoria, J. Santiago)

$$(C_{ll})_{ijkl} = \frac{(y_{\mathcal{S}_1})^*_{rjl}(y_{\mathcal{S}_1})_{rik}}{M^2_{\mathcal{S}_{1r}}} + \frac{(y_{\Xi_1})_{rki}(y_{\Xi_1})^*_{rlj}}{M^2_{\Xi_{1r}}} - \frac{(g^l_{\mathcal{B}})_{rkl}(g^l_{\mathcal{B}})_{rij}}{2M^2_{\mathcal{B}_r}} - \frac{(g^l_{\mathcal{W}})_{rkj}(g^l_{\mathcal{W}})_{ril}}{4M^2_{\mathcal{W}_r}} + \frac{(g^l_{\mathcal{W}})_{rkl}(g^l_{\mathcal{W}})_{rij}}{8M^2_{\mathcal{W}_r}},$$

Fields	Operators
$\mathcal{S}$	$\mathcal{O}_{\phi 4}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{\phi\Box}$, $\mathcal{O}_{\phi B}$, $\mathcal{O}_{\phi\tilde{B}}$, $\mathcal{O}_{\phi W}$, $\mathcal{O}_{\phi\tilde{W}}$, $\mathcal{O}_{\phi G}$, $\mathcal{O}_{\phi\tilde{G}}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
$\mathcal{S}_1$	$\mathcal{O}_{ll}$
$\mathcal{S}_2$	$\mathcal{O}_{ee}$
φ	$\mathcal{O}_{le}$, $\mathcal{O}^{(1)}_{qu}$, $\mathcal{O}^{(8)}_{qu}$, $\mathcal{O}^{(1)}_{qd}$, $\mathcal{O}^{(8)}_{qd}$, $\mathcal{O}_{ledq}$, $\mathcal{O}^{(1)}_{quqd}$, $\mathcal{O}^{(1)}_{lequ}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
Ξ	$\mathcal{O}_{\phi 4}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{\phi D}$, $\mathcal{O}_{\phi\Box}$, $\mathcal{O}_{\phi WB}$, $\mathcal{O}_{\phi W\tilde{B}}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
Ξ_1	$\mathcal{O}_{\phi 4}$, $\mathcal{O}_5$, $\mathcal{O}_{ll}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{\phi D}$, $\mathcal{O}_{\phi\Box}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
Θ_1	$\mathcal{O}_{\phi}$
Θ_3	$\mathcal{O}_{\phi}$
ω_1	$\mathcal{O}^{(1)}_{qq}$, $\mathcal{O}^{(3)}_{qq}$, $\mathcal{O}^{(1)}_{lq}$, $\mathcal{O}^{(3)}_{lq}$, $\mathcal{O}_{eu}$, $\mathcal{O}^{(1)}_{ud}$, $\mathcal{O}^{(8)}_{ud}$, $\mathcal{O}^{(1)}_{quqd}$, $\mathcal{O}^{(8)}_{quqd}$, $\mathcal{O}^{(1)}_{lequ}$ $\mathcal{O}^{(3)}_{lequ}$, $\mathcal{O}_{duq}$, $\mathcal{O}_{qqu}$, $\mathcal{O}_{qqq}$, $\mathcal{O}_{duu}$
ω_2	$\mathcal{O}_{dd}$
ω_4	$\mathcal{O}_{uu}$, $\mathcal{O}_{ed}$, $\mathcal{O}_{duu}$
Π_1	$\mathcal{O}_{ld}$
Π_7	$\mathcal{O}_{lu}$, $\mathcal{O}_{qe}$, $\mathcal{O}^{(1)}_{lequ}$, $\mathcal{O}^{(3)}_{lequ}$
ζ	$\mathcal{O}^{(1)}_{qq}$, $\mathcal{O}^{(3)}_{qq}$, $\mathcal{O}^{(1)}_{lq}$, $\mathcal{O}^{(3)}_{lq}$, $\mathcal{O}_{qqq}$
Ω_1	$\mathcal{O}^{(1)}_{qq}$, $\mathcal{O}^{(3)}_{qq}$, $\mathcal{O}^{(1)}_{ud}$, $\mathcal{O}^{(8)}_{ud}$, $\mathcal{O}^{(1)}_{quqd}$, $\mathcal{O}^{(8)}_{quqd}$
Ω_2	$\mathcal{O}_{dd}$
Ω_4	$\mathcal{O}_{uu}$
Υ	$\mathcal{O}^{(1)}_{qq}$, $\mathcal{O}^{(3)}_{qq}$
Φ	$\mathcal{O}^{(1)}_{qu}$, $\mathcal{O}^{(8)}_{qu}$, $\mathcal{O}^{(1)}_{qd}$, $\mathcal{O}^{(8)}_{qd}$, $\mathcal{O}^{(8)}_{quqd}$

$$Z^2_\phi C_{\phi D} = -\frac{2(\kappa_\Xi)_r(\kappa_\Xi)_r}{M^4_{\Xi_r}} + \frac{4(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_r}{M^4_{\Xi_{1r}}} - \frac{\mathrm{Re}\left((\hat{g}^\phi_{\mathcal{B}})_r(\hat{g}^\phi_{\mathcal{B}})_r\right)}{M^2_{\mathcal{B}_r}} - \frac{(\hat{g}^\phi_{\mathcal{B}})^*_r(\hat{g}^\phi_{\mathcal{B}})_r}{M^2_{\mathcal{B}_r}}$$
$$+ \frac{(\hat{g}^\phi_{\mathcal{B}_1})^*_r(\hat{g}^\phi_{\mathcal{B}_1})_r}{M^2_{\mathcal{B}_{1r}}} - \frac{\mathrm{Re}\left((\hat{g}^\phi_{\mathcal{W}})_r(\hat{g}^\phi_{\mathcal{W}})_r\right)}{4M^2_{\mathcal{W}_r}} + \frac{(\hat{g}^\phi_{\mathcal{W}})^*_r(\hat{g}^\phi_{\mathcal{W}})_r}{4M^2_{\mathcal{W}_r}} - \frac{(\hat{g}^\phi_{\mathcal{W}_1})^*_r(\hat{g}^\phi_{\mathcal{W}_1})_r}{4M^2_{\mathcal{W}_{1r}}}$$
$$+ \frac{g_1(g^B_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})_s}{M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} - \frac{(h^{(2)}_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})_s}{M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} + \frac{2\,\mathrm{Re}\left((h^{(3)}_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})^*_s\right)}{M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}}$$

$$C^S_\phi = -\frac{(\lambda_{\mathcal{S}})_{rs}(\kappa_{\mathcal{S}})_r(\kappa_{\mathcal{S}})_s}{M^2_{\mathcal{S}_r}M^2_{\mathcal{S}_s}} + \frac{(\kappa_{\mathcal{S}_3})_{rts}(\kappa_{\mathcal{S}})_r(\kappa_{\mathcal{S}})_t(\kappa_{\mathcal{S}})_s}{M^2_{\mathcal{S}_r}M^2_{\mathcal{S}_s}M^2_{\mathcal{S}_t}} + \frac{(\lambda_\varphi)^*_r(\lambda_\varphi)_r}{M^2_{\varphi_r}} + \frac{4\hat{\lambda}_\phi(\kappa_\Xi)_r(\kappa_\Xi)_r}{M^4_{\Xi_r}}$$
$$- \frac{(\lambda_\Xi)_s(\kappa_\Xi)_s(\kappa_\Xi)_r}{M^2_{\Xi_r}M^2_{\Xi_s}} + \frac{8\hat{\lambda}_\phi(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_r}{M^4_{\Xi_{1r}}} - \frac{2(\lambda_{\Xi_1})_{rs}(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_s}{M^2_{\Xi_{1r}}M^2_{\Xi_{1s}}}$$
$$+ \frac{\sqrt{2}(\lambda'_{\Xi_1})_{rs}(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_s}{M^2_{\Xi_{1r}}M^2_{\Xi_{1s}}} + \frac{(\lambda_{\Theta_1})^*_r(\lambda_{\Theta_1})_r}{6M^2_{\Theta_{1r}}} + \frac{(\lambda_{\Theta_3})^*_r(\lambda_{\Theta_3})_r}{2M^2_{\Theta_{3r}}}$$
$$- \frac{2\,\mathrm{Re}\left((\kappa_{\mathcal{S}\varphi})_{rs}(\lambda_\varphi)^*_s\right)(\kappa_{\mathcal{S}})_r}{M^2_{\mathcal{S}_r}M^2_{\varphi_s}} + \frac{(\kappa_{\mathcal{S}\varphi})^*_{rt}(\kappa_{\mathcal{S}})_r(\kappa_{\mathcal{S}\varphi})_{st}(\kappa_{\mathcal{S}})_s}{M^2_{\mathcal{S}_r}M^2_{\mathcal{S}_s}M^2_{\varphi_t}}$$
$$- \frac{(\lambda_{\mathcal{S}\Xi})_{sr}(\kappa_{\mathcal{S}})_s(\kappa_\Xi)_r}{M^2_{\Xi_r}M^2_{\mathcal{S}_s}} + \frac{(\kappa_{\mathcal{S}\Xi})_{tsr}(\kappa_{\mathcal{S}})_t(\kappa_\Xi)_s(\kappa_\Xi)_r}{M^2_{\Xi_r}M^2_{\Xi_s}M^2_{\mathcal{S}_t}}$$
$$- \frac{2\,\mathrm{Re}\left((\kappa_{\Xi\varphi})_{rs}(\lambda_\varphi)^*_s\right)(\kappa_\Xi)_r}{M^2_{\Xi_r}M^2_{\varphi_s}} + \frac{(\kappa_{\Xi\varphi})^*_{tr}(\kappa_\Xi)_t(\kappa_{\Xi\varphi})_{sr}(\kappa_\Xi)_s}{M^2_{\varphi_r}M^2_{\Xi_s}M^2_{\Xi_t}}$$
$$- \frac{4\,\mathrm{Re}\left((\lambda_{\mathcal{S}\Xi_1})_{rs}(\kappa_{\Xi_1})^*_s\right)(\kappa_{\mathcal{S}})_r}{M^2_{\Xi_{1s}}M^2_{\mathcal{S}_r}} + \frac{2(\kappa_{\mathcal{S}\Xi_1})_{trs}(\kappa_{\mathcal{S}})_t(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_s}{M^2_{\Xi_{1r}}M^2_{\Xi_{1s}}M^2_{\mathcal{S}_t}}$$
$$- \frac{2\sqrt{2}\,\mathrm{Re}\left((\lambda_{\Xi_1\Xi})_{rs}(\kappa_{\Xi_1})^*_r\right)(\kappa_\Xi)_s}{M^2_{\Xi_s}M^2_{\Xi_{1r}}} - \frac{\sqrt{2}(\kappa_{\Xi\Xi_1})_{srt}(\kappa_\Xi)_s(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_t}{M^2_{\Xi_{1r}}M^2_{\Xi_s}M^2_{\Xi_{1t}}} \quad \text{(D.44)}$$
$$- \frac{4\,\mathrm{Re}\left((\kappa_{\Xi_1\varphi})^*_{rs}(\kappa_{\Xi_1})_r(\lambda_\varphi)^*_s\right)}{M^2_{\Xi_{1r}}M^2_{\varphi_s}} + \frac{4(\kappa_{\Xi_1\varphi})^*_{st}(\kappa_{\Xi_1})_s(\kappa_{\Xi_1\varphi})_{rt}(\kappa_{\Xi_1})^*_r}{M^2_{\Xi_{1r}}M^2_{\Xi_{1s}}M^2_{\varphi_t}}$$
$$- \frac{\mathrm{Re}\left((\kappa_{\Xi\Theta_1})_{rs}(\lambda_{\Theta_1})^*_s\right)(\kappa_\Xi)_r}{3M^2\ldots M^2} + \frac{(\kappa_{\Xi\Theta_1})^*_{rs}(\kappa_\Xi)_r(\kappa_{\Xi\Theta_1})_{ts}(\kappa_\Xi)_t}{6M^2\ldots M^2\ldots M^2}$$

Automatic BSM – d6 SMEFT dictionary

Soon at: github.com/jccriado/smeft_dict

```
>>> smeft_dict.heavy_fields('1lq')
[['B'], ['U2'], ['X'], ['omega1'], ['zeta']]

>>> smeft_dict.operators('zeta')
['1qq', '3qq', '1lq', '3lq', 'qqq', 'qqqc']

>>> smeft_dict.coeff('1lq', 'zeta')
3*yqlzeta[r, l, j]*yqlzetac[r, k, i]/
(4*Mzeta[r]**2)

>>> smeft_dict.wcxf({'yqlzeta': ...})
wcxf.WC(...)
```

Basis codes

	rosetta	wilson	DEFT	BasisGen	abc_eft*
Purpose	change	change	gen/ch	gen	gen/ch
Group	SM	SM	SM	any	any
Reps	SM	SM	F, Adj, S	any	F
Flavor	no	yes	no	yes	yes
Structs.	–	–	yes	no	yes
Dim.	6	6	any	any	6

A format for operator exchange

Generality:

being able to represent as many operators as possible

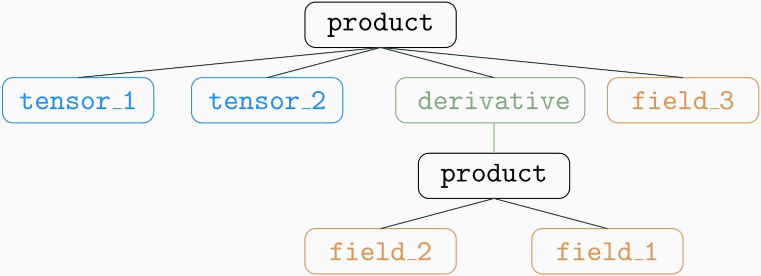
Normalization/canonicalization:

having a unique representation for each operator

A format for operator exchange: proposal

- `tree`: structure of products and derivatives (and sums?)

  tensor_1 * tensor_2 * D(field_1 * field_2) * field_3

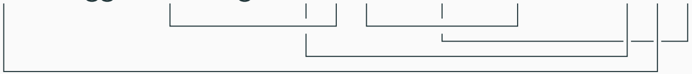


- `contractions`: pattern of index contractions `[(int, int)]`

A format for operator exchange: example

$$D^{\mu}(H^{\dagger}\sigma^{a}D^{\nu}H)W^{a}_{\mu\nu}$$

```
D(0)[Hdagger(1) sigma(2,3,4) D(5)[H(6)]] W(7,8,9)
```



```
{
  "tree": {"op_type": "product", "factors": [...]},
  "contractions": [
    [0, 8], [1, 3], [2, 7], [4, 6], [5, 9]
  ]
}
```

A format for operator exchange: example

$$D^{\mu}(H^{\dagger}\sigma^{a}D^{\nu}H)W^{a}_{\mu\nu}$$

```
{
  "op_type": "product",
  "factors": [
    {"op_type": "field", "name": "Hdagger"},
    {"op_type": "tensor", "name": "sigma"},
    {"op_type": "derivative", "content": {...}}
  ]
}
```

A format for operator exchange: example

$$D^\mu (H^\dagger \sigma^a D^\nu H) W^a_{\mu\nu}$$

$$\downarrow$$


```json
{
  "op_type": "derivative",
  "content": {"op_type": "field", "name": "H"}
}
```

Alternatives

- Apply Leibniz rule + tensor canonicalization to all operators: "unique" representation for each one (not taking into account group theory, IBP and redefinitions). [no $(\phi^\dagger \phi)\Box(\phi^\dagger \phi)$]
- Include more group-theoretical info in the representation: $H^\dagger \sigma^a H =$ `[doublet x doublet -> triplet]`, etc.
- Define "off-shell basis": Fierz identities and IBP.
- ...

Off-shell bases

- Useful for off-shell matching and RGE
- Precomputed translation to reduced "on-shell" bases.
- Easier collaboration:
  `reduced(L1) + reduced(L2)` $\neq$ `reduced(L1 + L2)`
- Sometimes, clearer observable consequences of matching results.

1811.09413 (JCC, M. Pérez-Victoria)

Redundant bases with BasisGen

Using the standard_model.py example script from github.com/jccriado/basisgen:

```
$ python3 standard_model.py --dimension 6
Number of operators:  84
(psi)^4:  38
(F)^3:  4
phi F (psi)^2:  16
(phi)^2 (psi)^2 D: 9
(phi)^3 (psi)^2:  6
(phi)^2 (F)^2:  8
(phi)^4 D^2:  2
(phi)^6:  1
```

Redundant bases with BasisGen

Using the standard_model.py example script from github.com/jccriado/basisgen:

```
$ python3 standard_model.py --dimension 6 --no_eom
Number of operators:  165
(psi)^2 D^3:  5               (phi)^2 (psi)^2 D: 23 [14]
(psi)^4:  38                  (phi)^3 (psi)^2:  6
F (psi)^2 D: 30               (phi)^2 D^4:  1
(F)^2 D^2:  3                 (phi)^2 F D^2:  2
(F)^3:  4                     (phi)^2 (F)^2:  8
phi (psi)^2 D^2:  24          (phi)^4 D^2:  4 [2]
phi F (psi)^2:  16            (phi)^6:  1
```

Conclusions

- Already large set of tools with many applications.
- Even more coming in the near future.
- `smeft_dict`: automatic BSM – d6 SMEFT tree-level dict.
- "Glue":
    - `WCxf`
    - A format for exchanging operators?
- Standardize a "off-shell" bases (`BasisGen` can provide them!)