SMEFT-Tools 2019, Durham, 12 June 2019

Automatic Basis Change for Effective Field Theories

Presented by Peter Stangl

Laboratoire d'Annecy-le-Vieux
de Physique Théorique

LAPTh

Outline

1. Motivation: Why automatic basis change? Why a new code?
2. Strategy
3. Representing the operators
4. Status and Outlook

Based on:

Francesco Sannino, PS, David M. Straub, Anders E. Thomsen [arXiv:1712.07646]
Jason Aebischer, PS [work in progress]

Outline

1. Motivation: Why automatic basis change? Why a new code?
2. Strategy
3. Representing the operators
4. Status and Outlook

Motivation: Why automatic basis change?

Why basis change?

- Different bases can be convenient for different kinds of analysis
- Matching to EFT can yield operators not in a given basis
- Loop computations can yield operators not in a given basis
- Running has only been computed in “Warsaw basis”

Why automatic?

- Possibility to easily construct new bases
- Transform any operator to any basis
- No restriction to specific theory (SMEFT, WET, but also BSM)

New code

- **abc_eft**: **A**utomatic **B**asis **C**hange for **E**ffective **F**ield **T**heories

Aebischer, PS, work in progress

Motivation: Why a new code?

Rosetta	– only SMEFT – translations hard-coded – not full basis – no flavour	+ first code for basis change + fast + interface to event generators
wilson / WCxf talk by J. Kumar	– only SMEFT and WET – translations hard-coded	+ fast + full basis incl. flavour + matching and running
BasisGen talk by J. C. Criado	– only counting	+ any gauge group + any field representations + full flavour + arbitrary operator dimension
DEFT talk by D. Sutherland	– only SMEFT – no flavour	+ arbitrary operator dimension + translate any operator to any basis
abc_eft this talk	– only dimension 6 – only fund. representations and singlets	+ any gauge group + full flavour + translate any operator to any basis

Outline

Matrix of linear dependencies

- Generate all possible **operators** $\boldsymbol{O_j}$
- Apply all possible **transformations** to each operator (Fierz, EOM, IBP, etc.)
  - Each transformation yields relation

    $$\sum_j a_j\, O_j = 0, \qquad a_j = 0 \quad \text{if } O_j \text{ not involved in transformation}$$

    LHS can be interpreted as row a_j of a matrix multiplied by vector O_j
  - All transformations constitute a **matrix** $\boldsymbol{M_{ij}}$ **of linear dependencies**

    $$\boldsymbol{M_{ij}\, O_j = 0}$$

Matrix of linear dependencies

- ***M_{ij}* can be very large**
  - e.g. only four-fermion operators in WET with full flavour structure:

    # of possible (redundant) operators $n_j = \mathcal{O}(10^4)$

    # of (redundant) relations between operators $n_i = \mathcal{O}(10^5)$
- Computing rank, inverse, etc. of very large matrix **not very efficient**

Block matrices

- But not all operators are related to each other
- M_{ij} can be brought into **block-diagonal form** using permutation matrix P

$$\tilde{O} = P\,O, \qquad \tilde{M} = M\,P^T, \qquad \tilde{M}\,\tilde{O} = 0$$

$$\tilde{M} = \begin{pmatrix} M^{(1)} & 0 & \dots & 0 \\ 0 & M^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & M^{(n_k)} \end{pmatrix}, \qquad \tilde{O} = \begin{pmatrix} O^{(1)} \\ O^{(2)} \\ \vdots \\ O^{(n_k)} \end{pmatrix}$$

$$\boldsymbol{M_{ij}^{(k)}\,O_j^{(k)} = 0}$$

- All matrix operations can be performed on **relatively small matrices $\boldsymbol{M_{ij}^{(k)}}$**

Choosing a basis

- Operator basis for each block
  - **# of independent relations** in $M_{ij}^{(k)}$: **rank** $\left(\boldsymbol{M_{ij}^{(k)}}\right)$
  - **# of basis operators** in $O_j^{(k)}$: $\boldsymbol{n_b^{(k)} = n_j^{(k)} -}$ **rank** $\left(\boldsymbol{M_{ij}^{(k)}}\right)$
- Choose basis operators among $O_j^{(k)}$
- Permute operators such that
  - $O_b^{(k)}$ contains only basis operators, $b \in [1, n_b^{(k)}]$
  - $O_{\bar{b}}^{(k)}$ contains only non-basis operators, $\bar{b} \in [n_b^{(k)} + 1, n_j^{(k)}]$

$$\begin{pmatrix} M_{ib}^{(k)} & M_{i\bar{b}}^{(k)} \end{pmatrix} \begin{pmatrix} O_b^{(k)} \\ O_{\bar{b}}^{(k)} \end{pmatrix} = 0$$

Removing redundant relations

$$\begin{pmatrix} M_{ib}^{(k)} & M_{i\bar{b}}^{(k)} \end{pmatrix} \begin{pmatrix} O_b^{(k)} \\ O_{\bar{b}}^{(k)} \end{pmatrix} = 0$$

- Matrices $\boldsymbol{M_{ib}^{(k)}}$ and $\boldsymbol{M_{i\bar{b}}^{(k)}}$ still **rank deficient**
- # of non-basis operators equals # of independent relations:

$$n_{\bar{b}}^{(k)} = \text{rank}\left(M_{ij}^{(k)}\right) = \text{rank}\left(M_{i\bar{b}}^{(k)}\right)$$

- Use **QR decomposition** with **column pivoting**:

$$\left(M^{(k)}\right)^T_{\bar{b}i} P_{ir} = \left(Q\,R\right)_{\bar{b}r}$$

(P: permutation, Q: unitary, R: upper triangular)

- $\hat{M}_{r\bar{b}}^{(k)} = P_{ri}^T\, M_{i\bar{b}}^{(k)}$ has full rank for $r \in \left[1, n_{\bar{b}}^{(k)}\right]$

Transforming to the basis

- **Only independent relations**:

$$\begin{pmatrix} \hat{M}^{(k)}_{rb} & \hat{M}^{(k)}_{r\bar{b}} \end{pmatrix} \begin{pmatrix} O^{(k)}_{b} \\ O^{(k)}_{\bar{b}} \end{pmatrix} = 0, \qquad r \in \left[1, n^{(k)}_{\bar{b}}\right]$$

- Multiplying from the left with $\left(\hat{M}^{(k)}\right)^{-1}_{\bar{b}r}$

$$\begin{pmatrix} -T^{(k)}_{\bar{b}b} & \mathbb{1} \end{pmatrix} \begin{pmatrix} O^{(k)}_{b} \\ O^{(k)}_{\bar{b}} \end{pmatrix} = 0$$

$$T^{(k)}_{\bar{b}b} = -\left(\hat{M}^{(k)}\right)^{-1}_{\bar{b}r} \hat{M}^{(k)}_{rb}$$

- **Non-basis operators** expressed **in terms of basis operators**

$$O^{(k)}_{\bar{b}} = T^{(k)}_{\bar{b}b} \, O^{(k)}_{b}$$

Outline

Representing the operators

- Represent operators in a way that is convenient for computer code
- Classify the operators by
  - **Lorentz/spinor structure**
  - **contractions of gauge group indices**
- Example: **four-fermion operator**

$$\frac{1}{2}(\bar{l}_i\,\gamma_\mu\,l_j)(\bar{q}_k\,\gamma^\mu\,q_l)$$

—— SU(2)
······ SU(3)

Representing the operators

- Represent operators in a way that is convenient for computer code
- Classify the operators by
  - **Lorentz/spinor structure**
  - **contractions of gauge group indices**
- Example: **four-fermion operator**

$$\frac{1}{2}(\bar{l}_i\,\gamma_\mu\,l_j)(\bar{q}_k\,\gamma^\mu\,q_l) = (l_j\,q_l)(q_k^\dagger\,l_i^\dagger)$$

—— SU(2)

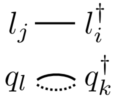
⋯⋯ SU(3)

Write operator in terms of 2-component Weyl spinors

Representing the operators

- Represent operators in a way that is convenient for computer code
- Classify the operators by
  - **Lorentz/spinor structure**
  - **contractions of gauge group indices**
- Example: **four-fermion operator**

$$\frac{1}{2}(\bar{l}_i\,\gamma_\mu\,l_j)(\bar{q}_k\,\gamma^\mu\,q_l) = (l_j\,q_l)(q_k^\dagger\,l_i^\dagger) =$$



—— SU(2)

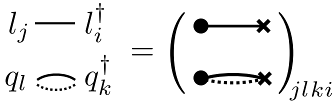
······· SU(3)

Rearrange fields in diagrammatic way

Representing the operators

- Represent operators in a way that is convenient for computer code
- Classify the operators by
  - **Lorentz/spinor structure**
  - **contractions of gauge group indices**
- Example: **four-fermion operator**

$$\frac{1}{2}(\bar{l}_i\,\gamma_\mu\,l_j)(\bar{q}_k\,\gamma^\mu\,q_l) = (l_j\,q_l)(q_k^\dagger\,l_i^\dagger) =$$



—— SU(2)

⋯⋯ SU(3)

● left-handed Wey spinor

✖ right-handed Weyl spinor

Use diagram to represent operator

Representing transformations

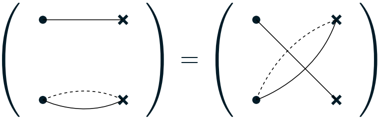
- Example: **Fierz identity** (gauge indices $a, b \in \mathrm{SU}(2)$, $i \in \mathrm{SU}(3)$)
  - in 4-component **Dirac spinor notation**

$$\frac{1}{2}(\bar{l}_a \gamma_\mu l^a)(\bar{q}_{bi} \gamma_\mu q^{bi}) = \frac{1}{2}(\bar{q}_{bi} \gamma_\mu l^a)(\bar{l}_a \gamma_\mu q^{bi})$$

  - in 2-component **Weyl spinor notation**

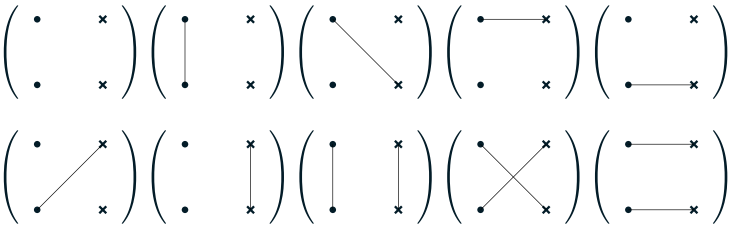
$$(l^a q^{bi})(q^\dagger_{bi} l^\dagger_a) = (l^a q^{bi})(l^\dagger_a q^\dagger_{bi})$$

  - in **diagram notation**



Generating operators

- Construct all contractions for each symmetry group, e.g.



  - + other Lorentz/spinor structures
  - + contractions containing group generators
  - + contractions containing epsilon tensor in $\mathrm{Sp}(N)$ groups, e.g. $\mathrm{SU}(2)$

- Combine contractions for all symmetry groups with each other
  - Contractions fix transformation properties of fields in operator

- Plug in compatible particle content and require cancellation of all $\mathrm{U}(1)$ charges

Outline

Current status

- Code originally developed for **transforming four-fermion operators**

  Sannino, PS, Straub, Thomsen, arXiv:1712.07646

- **Generates** four-fermion operators for EFT with
  - **arbitrary symmetry group**
  - **arbitrary fermion content** (but only (anti)fundamental representations and singlets)
  - **full flavour structure** for **arbitrary # of generations**
- **Relates** operators by
  - **Fierz identities** (including Schouten identity and $\sigma_{\mu\nu}\sigma^{\mu\nu}$ identity)
  - **Completeness relations of group generators** (e.g. “colour Fierz”)
- Possibility to **select basis by general requirements**, e.g.
  (no need to select operators individually)
  - group index contraction inside bilinears (e.g. exclude quark-lepton bilinears)
  - as few tensor operators as possible
  - as few operators containing group generators as possible
  - ...

Outlook

- Currently working on generalization to **operators involving bosons**

  Aebischer, PS, work in progress
  - **arbitrary scalar content** (but only (anti)fundamental representations and singlets)
  - Many **new relations** between operators (EOM, IBP, etc.)

- **Interface to other codes**
  - Generate basis files for `WCxf` — Aebischer et al., arXiv:1712.05298
  - Generate basis translators for `wilson` — Aebischer, Kumar, Straub, arXiv:1804.05033
  - Further **suggestions welcome!**

- Convenient **input/output format for operators**
  - A "standardized" file format might be useful to interface to other codes: `OPxf`?
  - Notation for input of user-defined operators
  - **Suggestions welcome!**

Using abc_eft (preliminary)

- Defining an EFT

```
import abc_eft

gauge_indices = ['SU3_C','SU2_L']
fermions = {'Q': [1,1,1/6],
            'L': [0,1,-1/2],
            'u': [-1,0,-2/3],
            'd': [-1,0,1/3],
            'e': [0,0,1]}
scalars = {'H': [0,1,-1/2]}

SMEFT = abc_eft.EFT(gauge_indices, fermions, scalars)
```

- Defining an operator

```
 op_LQ1 = 'L[a] Q[b,c] Q+[b,c] L+[a]'
```

Backup slides

Spinor notation

$$P_L\,\Psi_i = \begin{pmatrix}\chi_i\\ 0\end{pmatrix}, \quad P_R\,\Psi_i = \begin{pmatrix}0\\ \xi_i^\dagger\end{pmatrix}, \quad \bar{\Psi}_i\,P_R = \begin{pmatrix}0 & \chi_i^\dagger\end{pmatrix}, \quad \bar{\Psi}_i\,P_L = \begin{pmatrix}\xi_i & 0\end{pmatrix}.$$

$$\gamma^\mu\,P_L = \begin{pmatrix}0 & 0\\ \bar{\sigma}^\mu & 0\end{pmatrix}, \quad \gamma^\mu\,P_R = \begin{pmatrix}0 & \sigma^\mu\\ 0 & 0\end{pmatrix}$$

$$\sigma^{\mu\nu} = \frac{i}{4}(\sigma^\mu\bar{\sigma}^\nu - \sigma^\nu\bar{\sigma}^\mu), \quad \bar{\sigma}^{\mu\nu} = \frac{i}{4}(\bar{\sigma}^\mu\sigma^\nu - \bar{\sigma}^\nu\sigma^\mu).$$

$$\psi\psi = \psi^\alpha\psi_\alpha = \psi^\alpha\varepsilon_{\alpha\beta}\psi^\beta, \qquad \psi^\dagger\psi^\dagger = \psi^\dagger_{\dot\alpha}\psi^{\dagger\dot\alpha} = \psi^\dagger_{\dot\alpha}\varepsilon^{\dot\alpha\dot\beta}\psi^\dagger_{\dot\beta}$$

Representing four-fermion operators

- Three different Lorentz/spinor structures of four-fermion operators (+ conjugates)

$$\begin{pmatrix} \bullet & \times \\ & \\ \bullet & \times \end{pmatrix} = (\psi_1\,\psi_2)(\psi_3^\dagger\,\psi_4^\dagger) = \frac{1}{2}(\psi_4^\dagger\,\bar\sigma_\mu\,\psi_1)(\psi_3^\dagger\,\bar\sigma^\mu\,\psi_2),$$

$$\begin{pmatrix} \times & \times \\ & \\ \times & \times \end{pmatrix} = (\psi_1^\dagger\,\psi_2^\dagger)(\psi_3^\dagger\,\psi_4^\dagger),$$

$$\begin{pmatrix} \times & \times \\ & \\ \times & \times \end{pmatrix}^{\sigma} = (\psi_1^\dagger\bar\sigma_{\mu\nu}\psi_2^\dagger)(\psi_3^\dagger\bar\sigma^{\mu\nu}\psi_4^\dagger)$$