GDR Terascale - Annecy

Pyr@te

Python RGEs at Two-Loop for Everyone

arXiv:1309.7030

Florian Lyonnet

In collaboration with Ingo Schienbein, Florian Staub, Akın Wingerter

Laboratoire de Physique Subatomique et de Cosmologie
Université Joseph Fourier, Grenoble

November 29, 2013

Motivations

Description

- Generate the **Renormalization Group Equations** for non-supersymmetic theories @ 2-loop

Motivations

Description

- Generate the **Renormalization Group Equations** for non-supersymmetric theories @ 2-loop

- No evidence of SUSY so far :
  - $(g-2)_\mu,\ B_s \to \mu^+\mu^-,\ b \to s\gamma, \ldots$
  - collider experiments
  - direct DM detection experiments

Motivations

Pyr@te

Description

- Generate the **Renormalization Group Equations** for non-supersymmetric theories @ 2-loop

- No evidence of SUSY so far :
  - $(g-2)_\mu, \; B_s \to \mu^+\mu^-, \; b \to s\gamma, \ldots$
  - collider experiments
  - direct DM detection experiments
- Systematic studies of non-SUSY models require the RGEs
- One possible application: constraining non-SUSY BSM models via the stability bound

- RGEs for general gauge theories known for a long time:
  - *M. Machacek and M. T. Vaughn, 1983 Nuc.Phys.B222*
  - *M. Luo et al. Phys.Rev. D67 (2003) 065019*
- Calculation of beta functions "by hand" is time consuming and prone to error ⇒ Difficult to use in practice.
- Full set of 2-loop RGEs known only for few specific cases:
  - SM + Neutrinos

    from *A. Wingerter Phys.Rev. D84 (2011) 095012*
  - SM + chiral fourth generation

    from *C. Cheung et al. JHEP 1207 (2012) 105*
  - SM + real singlet scalar
  - SM + real triplet scalar
  - SM + complex doublet scalar
  - ...

SUSY

- SARAH *Comp. Phys. Com. 182 (2011) pp. 808-833* (spectrum generator generator)
- SUSYNO *Comput.Phys.Commun. 183 (2012) 2298-2306*

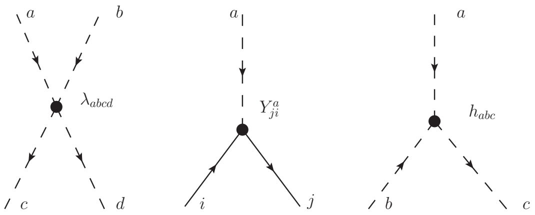
NON-SUSY

- **PyR@TE** cross checked with the beta version of SARAH 4.0.

Outline

Renormalization Group Equations

- Renormalization scale μ
  $\Rightarrow g_{10}, \alpha_{S0}, \lambda_0 \cdots \Rightarrow \tilde{g}_1(\mu), \tilde{\alpha}_S(\mu), \tilde{\lambda}(\mu)$.
- RGEs : ensure the invariance of the observables.
  - e.g. : $\mu \frac{d}{d\mu} \tilde{\alpha}_S(\mu) = \beta_{\alpha_S}$



- β functions depend on the theory i.e. **particles and gauge groups**.
- Can be approximated in perturbation theory.

Definition

- Take a general gauge field theory

$G_1 \times G_2 \times \cdots \times G_n$ direct product of simple groups

$$\begin{aligned}\mathcal{L} \supset \quad & - \quad N_a Y^a_{jk} \psi_j \xi \psi_k \phi_a + h.c. \Rightarrow \beta^a_{jk} \\ & - \quad N_\lambda \lambda_{abcd} \phi_a \phi_b \phi_c \phi_d \Rightarrow \beta_{abcd} \\ & - \quad N_{mf} (mf)_{jk} \psi_j \xi \psi_k + h.c. \Rightarrow (\beta_{mf})_{jk} \\ & - \quad N_{mab} m^2_{ab} \phi_a \phi_b \Rightarrow \beta_{ab} \\ & - \quad N_h \phi_a \phi_b \phi_c \Rightarrow \beta_{abc},\end{aligned}$$

$\Rightarrow$ 6 types of beta functions to calculate:

- $\beta(g) \Rightarrow$ gauge couplings
- $\beta^a_{jk} \Rightarrow$ yukawas
- $\beta_{abcd} \Rightarrow$ quartic couplings
- $\beta_{ab} \Rightarrow$ scalar mass
- $(\beta_{mf})_{jk} \Rightarrow$ fermion mass
- $\beta_{abc} \Rightarrow$ trilinear couplings

Results

- Known @two-loop:
  - Machacek and M. T. Vaughn, 1983 Nuc.Phys.B222
  - Corrected/enhenced M. Luo et al. Phys.Rev. D67 (2003)
  - Multiple $U(1)$ factors, M. Luo et al Phys.Lett. B555 (2003)
    - Also see, R. Fonseca, M. Malinsky, F. Staub, arXiv:1308.1674
- e.g. gauge coupling constant for unique gauge group factor :

$$\beta(g) = -\frac{g^3}{(4\pi)^2}\left\{\frac{11}{3}C_2(G) - \frac{4}{3}\kappa S_2(F) - \frac{1}{6}S_2(S) + 2\frac{\kappa}{(4\pi)^2}Y_4(F)\right\}$$
$$+\frac{g^5}{(4\pi)^4}\left\{\frac{34}{3}[C_2(G)]^2 - \kappa[4C_2(F) + \frac{20}{3}C_2(G)]S_2(F)\right.$$
$$\left.-[2C_2(S) + \frac{1}{3}C_2(G)]S_2(S)\right\},$$

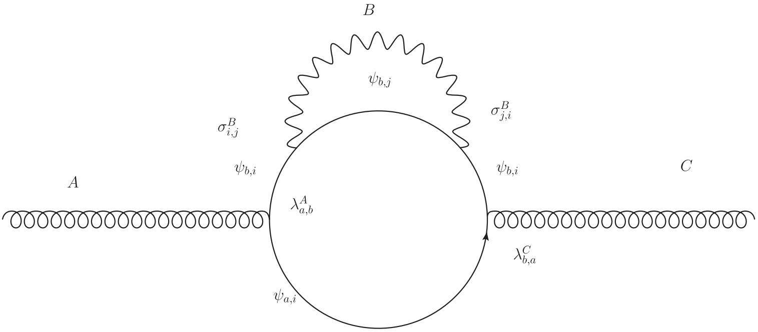
$$Y_4(F) = \frac{1}{d(G)}Tr\left(C_2(F)Y^aY^{\dagger a}\right)$$

Results

- Notation extremely compact, difficult to find the right multiplicity!

E.g.(1) : two-loop gauge couplings beta function

- $(S(R)C(R))_k \equiv$
$$\sum_r \sum_l g_k^2 g_l^2 \mathcal{N}_r \mathcal{S}_k(\Lambda(r)) \mathcal{C}_l(\Lambda(r)) \prod_m \tilde{N}(\Lambda(r))_{mk}$$
- r is running over the scalars ($R = S$) or fermions ($R = F$) of the model.
- $\mathcal{C}_l$ is the quadratic casimir of the irrep $\Lambda(r)$.
- $\mathcal{S}_k$ is the dynkin index of the irrep $\Lambda(r)$.
-
$$\tilde{N}(\Lambda)_{lk} = \begin{cases} N_l(\Lambda) & \text{if } l \neq k, \\ 1 & \text{else if } l = k\,. \end{cases}$$



E.g. (2): $g_2^2 g_3^3$ contribution to g_3 in the SM

$$diag \sim g_2^2 g_3^2 \sum_{a,b,i,j,B} \lambda^A_{a,b} \sigma^B_{i,j} \sigma^B_{j,i} \lambda^C_{b,a}$$

SUSY vs Non-SUSY RGEs

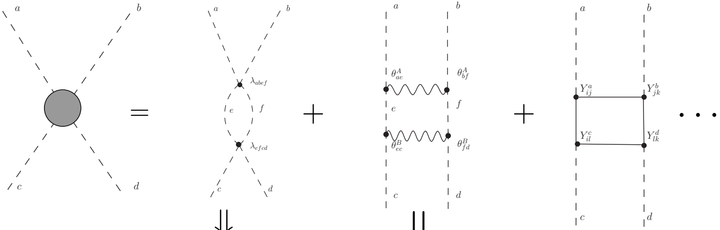
- Non SUSY case ⇒Quartic Terms
- Expressions more involved ⇒ more time consuming
- One needs the explicit matrices of the representation for the scalars and fermions:
  - $D_\mu \phi_a = \partial_\mu \phi_a - ig\theta^A_{ab} V^A_\mu \phi_b$
- θ^A_{ab} assumed purely imaginary and antisymmetric in the calculation. ⇒ Hermitian Basis
  - complex hermitian field with n components ⇒ $2n$ components real vector transforming as

$$L_i = \frac{1}{2}\begin{pmatrix} \tilde{L}_i - \tilde{L}^*_i & i(\tilde{L}_i + \tilde{L}^*_i) \\ -i(\tilde{L}_i + \tilde{L}^*_i) & \tilde{L}_i - \tilde{L}^*_i \end{pmatrix}$$

$$L^1_{\phi_h} = \frac{i}{2}\begin{pmatrix} 0 & \tau^1 \\ -\tau^1 & 0 \end{pmatrix}, L^2_{\phi_h} = \frac{1}{2}\begin{pmatrix} \tau^2 & 0 \\ 0 & \tau^2 \end{pmatrix}, L^3_{\phi_h} = \frac{i}{2}\begin{pmatrix} 0 & \tau^3 \\ -\tau^3 & 0 \end{pmatrix}$$

$$\phi_h = (\phi_1, \phi_2, \phi_3, \phi_4)^T, \phi^+ = (\phi_1 + i\phi_2)/\sqrt{2}, \phi^0 = (\phi_3 + i\phi^4)/\sqrt{2}$$

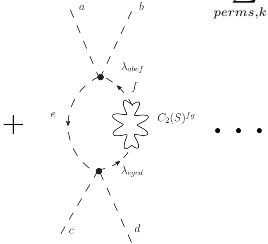
The Quartic Terms



$$\sim \sum_{perm} \lambda_{abef}\lambda_{efcd}$$

$$\sim \sum_{perms,k,l} g^{2k} g^{2l} \{\theta^A,\theta^B\}_{ab}\{\theta^A,\theta^B\}_{cd}$$

$$\sim \sum_{perms}\sum_{i,j,k,l} Y^a_{ij}Y^{b\dagger}_{jk}Y^c_{kl}Y^{d\dagger}_{li}$$



$$\sim \sum_{perm} g^2 C_2^{fg}(S)\lambda_{abef}\lambda_{cdeg}$$

Summury

What are the different ingredients needed ?

- C_2, S_2 for all the representations involved
- θ^A, t^A matrix representation for the scalars and fermions
- Contract the different terms in the Lagrangian into singlets :
  - CGCs, database built from Susyno arxiv: 1106.5016
- Replacement rules to go from single gauge group factor to product :
  - $G \to G_1 \times G_2 \times \cdots \times G_n$
  - e.g. $g^4 C_2(R) C_2(R') \to \sum_{k,l} g_k^2 g_l^2 C_2^k(R) C_2^l(R')$

Outline

Main features

- Public code for any non-SUSY theories, RGEs at 2-loop .
- Version 1.0.2 is out : http://pyrate.hepforge.org
- Gauge Groups : $SU(n), n = 2, ..., 6$ (no kinetic mixing).
- shell and Interactive mode (IPython notebook)

Validation

- Collaborator F. Staub implemented same RGEs in SARAH 4, arXiv: 1309.7223 $\Rightarrow$ independent cross check.
- All the models from C. Cheung et al. JHEP 1207 (2012) 105
- Cross checking the beta functions that are not in the SM :
  - SM + one real scalar field $\Rightarrow$ Trilinear term
  - SM + $t^{'}$ vector like quark $\Rightarrow$ Fermion mass term

Future developments :

- Extend the group part i.e. more groups, more irreps
- Index generation for scalars
- Multiple $U(1) \Rightarrow$ Kinetic mixing
- Running of the vevs, arXiv: 1305.1548
- Include available three loops results

Structure of PyR@TE

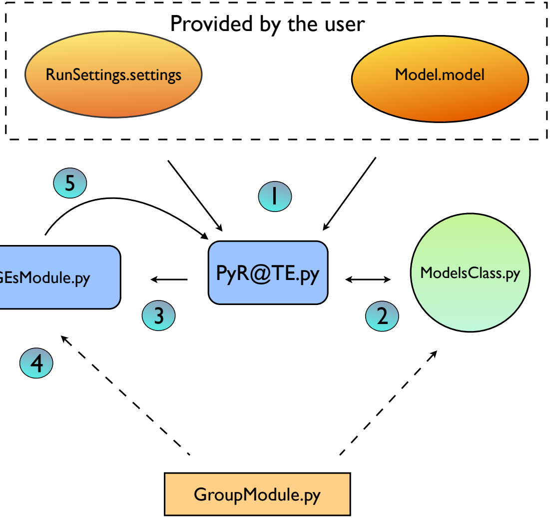
1 : read the settings from the input files

2 : construct the Model class

3 : each RGE is calculated by the RGEsModule

4 : the calculation of each RGE requires information from GroupModule.py

5 : the result is sent back to PyR@TE.py for final output



Input files

- *.model* required to run and *.settings*.

.model

- we are using text files for the input (YAML)
- keys :
  - Author Date Name

    e.g. Name : SMtp

Input files

- *.model* required to run and *.settings*.

.model

- we are using text files for the input (YAML)
- keys :
  - Author Date Name

    e.g. Name : SMtp
  - *Groups* : GroupName: Group

    e.g. SU2L: SU2

Input files

- *.model* required to run and *.settings*.

.model

- we are using text files for the input (YAML)
- keys :
  - Author Date Name

    e.g. Name : SMtp
  - *Groups* : GroupName: Group

    e.g. SU2L: SU2
  - *Fermions*: Label: Gen: ng, Qnb:{GroupName:value}

    e.g. Q: Gen: ng, Qnb:{ U1: 1/3, SU2L: 2, SU3c: 3}

Input files

- *.model* required to run and *.settings*.

.model

- we are using text files for the input (YAML)
- keys :
  - Author Date Name

    e.g. Name : SMtp
  - *Groups* : GroupName: Group

    e.g. SU2L: SU2
  - *Fermions*: Label: Gen: ng, Qnb:{GroupName:value}

    e.g. Q: Gen: ng, Qnb:{ U1: 1/3, SU2L: 2, SU3c: 3}
  - Potential ⇒ is given in a similar way:

    e.g. Yukawas: Y_u : $Fields : [Qbar, H*, uR], Norm : 1$

Running

We run PyR@TE for the SMtp model with options :

- -a: all contributions
- -res results/DemoSMtp: result directory set to 'results/DemoSMtp'
- -v: verbose mode

```
%run pyR@TE.py -m models/SMtp.model -a -res results/DemoSMtp -v
```

```
                ==================================================================================

                                PyR@TE version 1.0.1 released September 27 2013

                        F. Lyonnet, I. Schienbein, F.Staub, A.Wingerter, arxiv 1309.7030
                ==================================================================================


Starting a new run ...
Run Settings :
                verbose: True
                Gauge-Couplings: False
                LatexOutput: True
                Yukawas: False
                LogFile: logFile.log
                ExportFile: BetaFunction.py
                PickleFile: RGEsOutput.pickle
```

We can extract all the results from RGEs list e.g. the GaugeCouplings are given by :

```
GaugeCouplings = getoneloop(RGEs[0])
GaugeCouplings
```

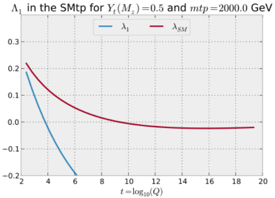
$$\left[-\frac{19}{6} g_{SU2L}^3, \quad \frac{131}{18} g_1^3, \quad -\frac{19}{3} g_{SU3c}^3 \right]$$

Running

Let's see how y_t modifies the running

```
yt = 0.5
mtp = 2e3
Y0SMtp[-6],Y0SMtp[-7] = mtp,yt
rge.Y0 = Y0SMtp
rge.solve_rges(t0,tf,ts)
```

```
plot(rge.Sol['t'],rge.Sol['bl1'],label='$\lambda_1$')
plot(tSM,LambdaSM,label='$\lambda_{SM}$')
title('$\Lambda_1$ in the SMtp for $Y_t(M_z)={}$ and $mtp = {}$ GeV'.format(yt,mtp))
xlabel('$t=\log_{10}(Q)$')
axis([2,20,-0.2,0.4])
legend(loc=9,ncol=3)
savefig('LambdaSMtp.pdf')
```



Conclusion and outlook

- For a more systematic study of non SUSY models RGEs are needed.
- We developed a tool that generates the RGEs @2-loop $\Rightarrow$ PyR@TE
- Have fun !