RGEs and group theory calculations with the Susyno program

Renato Fonseca

renato.fonseca@ific.uv.es

AHEP Group, Instituto de Física Corpuscular
CSIC/Universitat de València, Spain

Outline

I — The Susyno program: what is it for?

It was created to calculate the RGEs of SUSY models

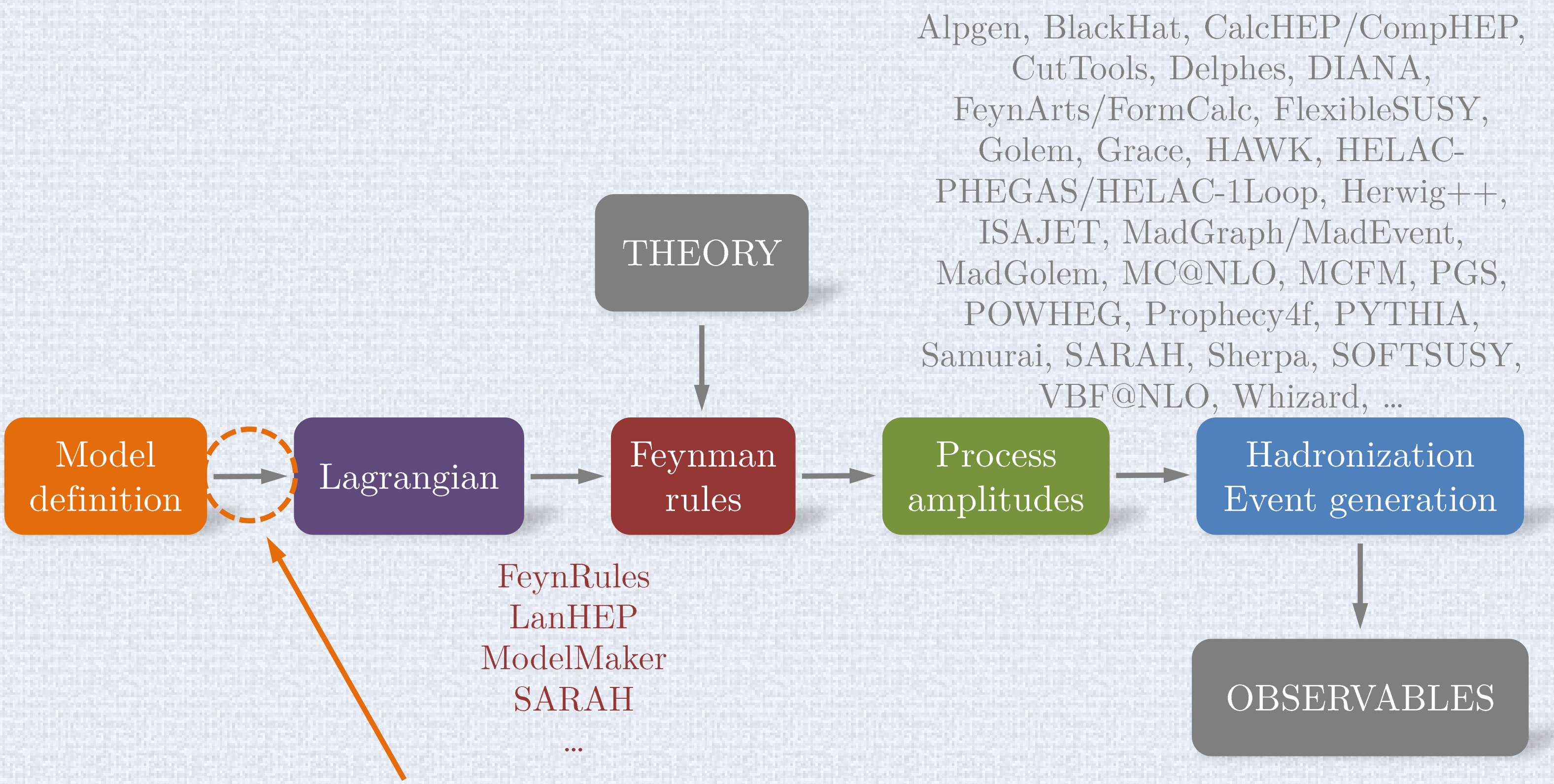
II — Using Susyno to make Group Theory calculations

I will focus mainly on this point in this presentation

III — Summary

The Susyno program: what is it for?

Automation in High Energy Physics



Gauge theories are defined by a **gauge group** and fields transforming under some of its **representations**. The rest are usually just conventions/notation (fixing a specific basis for the representations, naming the parameters, ...)

Handling of Group Theory is needed

Susyno

web.ist.utl.pt/renato.fonseca/susyno.html

Susyno is a Mathematica package which was created to compute the **Renormalization Group Equations (RGEs) of SUSY models**. RF 2012

The two-loop **RGEs of a generic Yang-Mills theory** are known for SUSY (and non-SUSY) models, but in order to apply them to **specific models** it takes some work. One reason for this is that the equations are written **for** and **as a function of** the following **generic tensors**: Martin, Vaughn 1994, 2008 Yamada 1994

$$W = \frac{1}{6} Y^{ijk} \Phi_i \Phi_j \Phi_k + \frac{1}{2} \mu^{ij} \Phi_i \Phi_j + L^i \Phi_i$$

$$-\mathscr{L}_{\text{soft}} = \left(\frac{1}{2} M_a \lambda^a \lambda^a + \frac{1}{6} h^{ijk} \phi_i \phi_j \phi_k + \frac{1}{2} b^{ij} \phi_i \phi_j + s^i \phi_i + \text{h.c.} \right) + \left(m^2\right)^i_j \phi_i \phi_j^*$$

Note that W and $\mathscr{L}_{\text{soft}}$ are to be expanded in all indices (in principle). This includes the gauge indices.

As such, a big part of Susyno's code is dedicated to Group Theory. It can compute various quantities for **any gauge group and field content**. **Such functions are available for other applications**.

To my knowledge, **LieART** is the only other Mathematica package with some (not all) of these functions.

Feger, Kephart 2012

Susyno

Documentation

Published manual

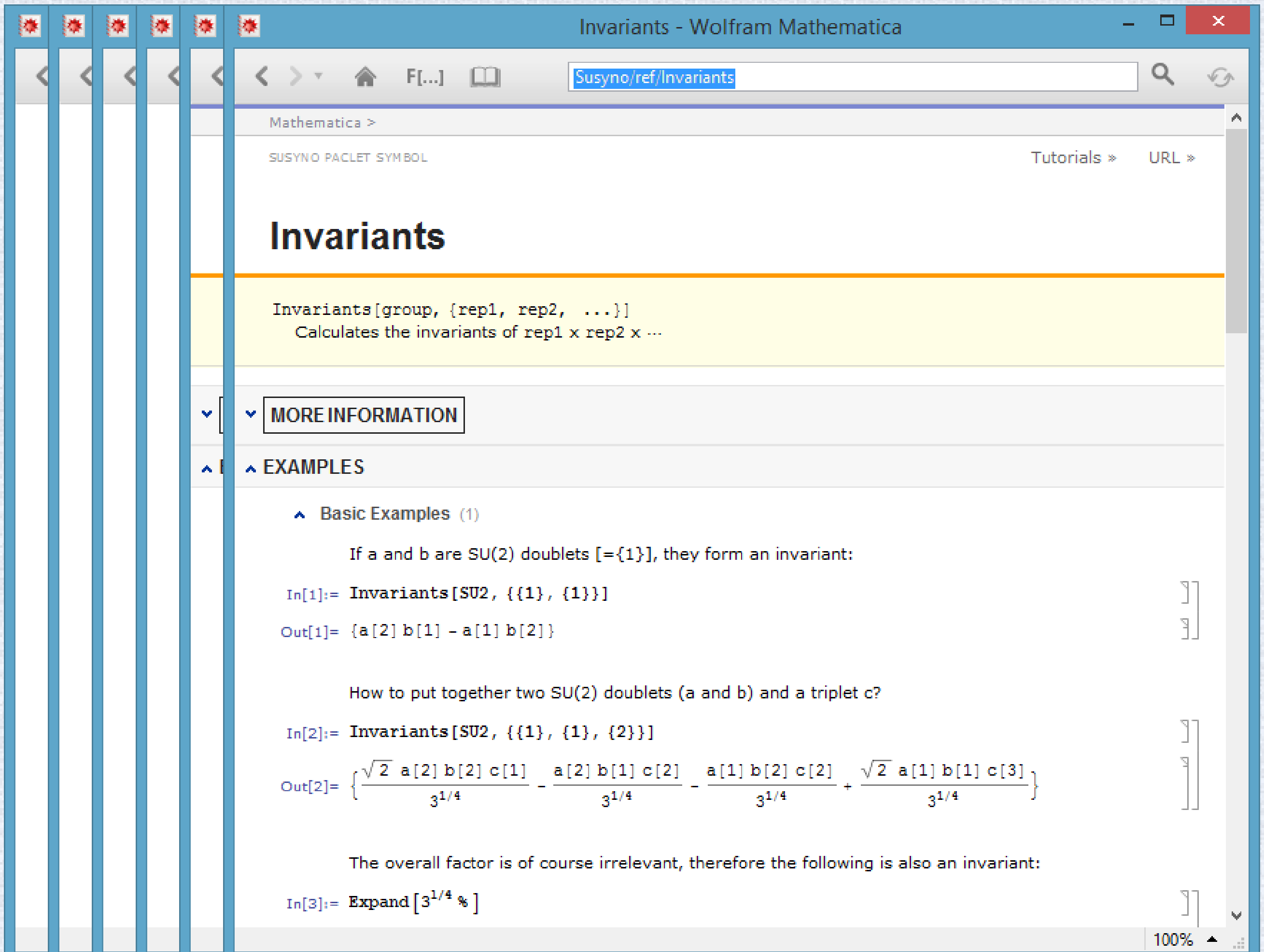
Computer Physics
Communications 183 (2012) 2298
arXiv:1106.5016 [hep-ph]

Built-in documentation

Up-to-date
More detailed
Easier to use

Susyno

Built-in documentation: easy to use and very handy



Input example: the MSSM

Output: see extra slides

Pick a name for the model (any)

```
author[MSSM] ^= "Me";
date[MSSM] ^= "14:50, 25 July 2014";
```

Provide some optional data

```
group[MSSM] ^= {U1, SU2, SU3};
```

Specify the model's gauge group (any)

```
normalization = Sqrt[3 / 5];
u = {-2 / 3 normalization, {0}, {0, 1}};
d = {1 / 3 normalization, {0}, {0, 1}};
Q = {1 / 6 normalization, {1}, {1, 0}};
e = {normalization, {0}, {0, 0}};
L = {-1 / 2 normalization, {1}, {0, 0}};
Hu = {1 / 2 normalization, {1}, {0, 0}};
Hd = {-1 / 2 normalization, {1}, {0, 0}};
```

Specify the model's representations (any)

```
reps[MSSM] ^= {u, d, Q, e, L, Hu, Hd};
fieldNames[MSSM] ^= {"u", "d", "Q", "e", "L", "Hu", "Hd"};
```

```
nFlavs[MSSM] ^= {3, 3, 3, 3, 3, 1, 1};
discreteSym[MSSM] ^= {-1, -1, -1, -1, -1, 1, 1};
```

Provide the number of flavors and the charges under any abelian symmetry

```
GenerateModel[MSSM, CalculateEverything → True]
```

Tell the program to calculate the model's RGEs (among other things)

Input example: minimal $SO(10)$

Output: see extra slides

```
group[MinimalSUSYSO10GUT] ^= {SO10};

Ψ = {{0, 0, 0, 0, 1}};      (* 16-dim representation *)
Φ = {{0, 0, 0, 1, 1}};      (* 210-dim representation *)
Δ = {{0, 0, 0, 0, 2}};      (* 126-dim representation *)
Δb = {{0, 0, 0, 2, 0}};   (* 126-dim representation (conj.) *)
H = {{1, 0, 0, 0, 0}};    (* 10-dim representation *)

reps[MinimalSUSYSO10GUT] ^= {Ψ, Φ, Δ, Δb, H};
fieldNames[MinimalSUSYSO10GUT] ^= {"Ψ", "Φ", "Δ", "Δ̄", "H"};

nFlavs[MinimalSUSYSO10GUT] ^= {3, 1, 1, 1, 1};
discreteSym[MinimalSUSYSO10GUT] ^= {1, 1, 1, 1, 1};

GenerateModel[MinimalSUSYSO10GUT, CalculateEverything → True]
```

Using Susyno to make Group Theory calculations

Overview

Some useful functions

Casimir | ConjugateIrrep | DynkinIndex | DimR | PermutationSymmetryOfInvariants | ReduceRepProduct | RepName | RepsUpToDimN | TriangularAnomalyValue ...

Basis-independent functions

RepMatrices | Invariants

Basis-dependent functions

HookContentFormula | DecomposeSnProduct | SnClassCharacter | SnClassOrder | SnIrrepDim | SnIrrepGenerators

Permutation group (S_n) functions

DecomposeReps | RegularSubgroupProjectionMatrix | SubgroupEmbeddingCoefficients

NEW

Symmetry breaking functions

Specifying a gauge group/representation

Group

Just type the group's name:
U1, SU2, SU3, ..., SO3, SO5, SO6, SO7, ...,
SP4, SP6, SP8, ..., G2, F4, E6, E7, E8

If the group contains more than one factor, use lists: {U1,SU2,SU3}, {SU5,E8}, ...

Representations

Representations of simple groups have to be given by their **Dynkin indices**, which are a list of non-negative integers
(for U(1)'s: provide the hypercharge).
For example, the complete list of representations of SU(3) is {0,0}, {0,1}, {1,0}, {1,1}, {0,2}, ...
Tables of representations in this notation are available for example in Slansky 1981
But **Susyno itself can be used to identify them**...

RepName, DimR, Casimir, DynkinIndex

RepName	DimR	Casimir	DynkinIndex
Identify by name a representation *	Dimension of representation $d(R)$	$\sum_a T_a^2 = C(R)\mathbb{1}$	$\mathrm{Tr}\left(T_a T_b\right) = T(R)\delta_{ab}$

Example 1

```
rep = {0, 0, 0, 1};

RepName[SU5, rep]
DimR[SU5, rep]
Casimir[SU5, rep]
DynkinIndex[SU5, rep]
```

$\overline{5}$

5

$\frac{12}{5}$

$\frac{1}{2}$

Example 2

```
rep = {4, 0, 0, 1};

RepName[SU5, rep]
DimR[SU5, rep]
Casimir[SU5, rep]
DynkinIndex[SU5, rep]
```

$\overline{315'}$

315

$\frac{88}{5}$

231

* Convention for assigning names to representations: RepName follows the scheme described in Feger, Kephart 2012 (tables in the literature — for example Slansky 1981 — are finite)

RepName, DimR, Casimir, DynkinIndex

and RepsUpToDimN

CODE

```
reps = RepsUpToDimN[SO11, 5000];

Grid[
 Prepend[
  {#, RepName[SO11, #], Casimir[SO11, #],
     DynkinIndex[SO11, #]} & /@ reps,
  Style[#, {Darker[Red], Bold}] & /@
   {"Dynkin indices", "Name", "Casimir",
    "DynkinIndex"}]]
```

OUTPUT

Dynkin indices	Name	Casimir	DynkinIndex
{0, 0, 0, 0, 0}	**1**	0	0
{1, 0, 0, 0, 0}	**11**	5	1
{0, 0, 0, 0, 1}	**32**	$\frac{55}{8}$	4
{0, 1, 0, 0, 0}	**55**	9	9
{2, 0, 0, 0, 0}	**65**	11	13
{0, 0, 1, 0, 0}	**165**	12	36
{3, 0, 0, 0, 0}	**275**	18	90
{1, 0, 0, 0, 1}	**320**	$\frac{99}{8}$	72
{0, 0, 0, 1, 0}	**330**	14	84
{1, 1, 0, 0, 0}	**429**	15	117
{0, 0, 0, 0, 2}	**462**	15	126
{4, 0, 0, 0, 0}	**935**	26	442
{0, 2, 0, 0, 0}	**1144**	20	416
{0, 1, 0, 0, 1}	**1408**	$\frac{135}{8}$	432
{1, 0, 1, 0, 0}	**1430**	18	468
{2, 0, 0, 0, 1}	**1760**	$\frac{151}{8}$	604
{2, 1, 0, 0, 0}	**2025**	22	810
{5, 0, 0, 0, 0}	**2717**	35	1729
{1, 0, 0, 1, 0}	**3003**	20	1092
{0, 0, 1, 0, 1}	**3520**	$\frac{163}{8}$	1304
{0, 0, 0, 0, 3}	**4224**	$\frac{195}{8}$	1872
{1, 0, 0, 0, 2}	**4290**	21	1638

ReduceRepProduct

Computes the **decomposition** of a product of group representations **into irreducible parts**

Very useful function in model building

Example

$\mathbf{3}\times\mathbf{3}\times\overline{\mathbf{6}} =?$ in $SU(3)$

```
RepName[SU3, {1, 0}]
RepName[SU3, {0, 2}]
3
6̄
```

→

```
result = ReduceRepProduct[SU3, {{1, 0}, {1, 0}, {0, 2}}]
{{{2, 2}, 1}, {{0, 3}, 1}, {{1, 1}, 2}, {{0, 0}, 1}}
{RepName[SU3, #1], #2} & @@@ result
{{27, 1}, {10̄, 1}, {8, 2}, {1, 1}}
```

Output is a list of **representations** with **multiplicity**

So $\mathbf{3}\times\mathbf{3}\times\overline{\mathbf{6}} = \mathbf{27}+\overline{\mathbf{10}}+\mathbf{8}+\mathbf{8}+\mathbf{1}$

ReduceRepProduct

Code and algorithm are very fast

Snow 1990, 1993

$\mathbf{5} \times \mathbf{10} \times \mathbf{15} \times \overline{\mathbf{40}} \times \mathbf{45} = ?$ in $SU(5)$

```
result = ReduceRepProduct[SU5, {{1, 0, 0, 0}, {0, 1, 0, 0}, {2, 0, 0, 0}, {1, 1, 0, 0}, {0, 1, 0, 1}}];
{RepName[SU5, #1], #2} &@@@ result
```

```
{{15750, 1}, {11880, 2}, {16170, 8}, {2400, 8}, {15360, 2}, {7425, 6}, {4620, 10}, {2625', 19},
 {10240, 9}, {8750, 16}, {4410, 21}, {2430, 41}, {2475, 2}, {2625, 21}, {945, 55}, {1470, 9},
 {1200, 33}, {3780, 10}, {2205, 9}, {2520, 34}, {1120, 54}, {420, 10}, {175'', 15}, {450', 14},
 {480, 77}, {280', 12}, {280, 63}, {70, 33}, {45, 50}, {4410', 1}, {1800, 12}, {980, 24},
 {720, 72}, {105, 54}, {50, 36}, {2520''', 1}, {560'', 11}, {1540', 2}, {70', 17}, {5, 13}}
```

Products of groups can also be used

```
d = {1/3 , {0}, {0, 1}};
Q = {1/6 , {1}, {1, 0}};
L = {-1/2 , {1}, {0, 0}};

result = ReduceRepProduct[{U1, SU2, SU3}, {d, Q, L}];
{RepName[{U1, SU2, SU3}, #1], #2} &@@@ result
```

```
{{0⊗3⊗8, 1}, {0⊗3⊗1, 1}, {0⊗1⊗8, 1}, {0⊗1⊗1, 1}}
```

(contains a gauge singlet)

PermutationSymmetryOfInvariants

Does the **same as ReduceRepProduct** but also provides information on **how the irreducible parts transform under permutations** of the representations being multiplied

Leeuwen, Cohen, Lisser 1992

This is relevant only when there are repeated representations

Example

It is well known that **three triplets of SU(3)** form one invariant which is completely anti-symmetric

In other words, the **invariant is in the {1,1,1} irreducible representation of S_3**

```
PermutationSymmetryOfInvariants[SU3, {{1, 0}, {1, 0}, {1, 0}}]

{{{1, 2, 3}}, {{{{1, 1, 1}}, 1}}}
```

Input reps #1, #2, #3 are the same

The given product contains 1 invariant, in an {1,1,1} irrep of S_3

OUTPUT MEANING (see documentation for details and more complex examples)

RepMatrices

RF 2013

This function **builds explicitly the representation matrices of a given gauge group**. To do so, a **particular basis** is chosen by the program

Example 1

Representation matrices of **3** of **SU(3)**

```
MatrixForm /@ RepMatrices[SU3, {1, 0}]
```

$$\left\{\begin{pmatrix}0&\frac{1}{2}&0\\\frac{1}{2}&0&0\\0&0&0\end{pmatrix},\begin{pmatrix}0&0&0\\0&0&\frac{1}{2}\\0&\frac{1}{2}&0\end{pmatrix},\begin{pmatrix}0&0&\frac{i}{2}\\0&0&0\\-\frac{i}{2}&0&0\end{pmatrix},\begin{pmatrix}0&-\frac{i}{2}&0\\\frac{i}{2}&0&0\\0&0&0\end{pmatrix},\begin{pmatrix}0&0&0\\0&0&-\frac{i}{2}\\0&\frac{i}{2}&0\end{pmatrix},\begin{pmatrix}0&0&-\frac{1}{2}\\0&0&0\\-\frac{1}{2}&0&0\end{pmatrix},\begin{pmatrix}\frac{1}{\sqrt{3}}&0&0\\0&-\frac{1}{2\sqrt{3}}&0\\0&0&-\frac{1}{2\sqrt{3}}\end{pmatrix},\begin{pmatrix}0&0&0\\0&\frac{1}{2}&0\\0&0&-\frac{1}{2}\end{pmatrix}\right\}$$

Example 2

Representation matrices of **5** of **SU(2)**

```
MatrixForm /@ RepMatrices[SU2, {4}]
```

$$\left\{\begin{pmatrix}0&1&0&0&0\\1&0&\sqrt{\frac{3}{2}}&0&0\\0&\sqrt{\frac{3}{2}}&0&\sqrt{\frac{3}{2}}&0\\0&0&\sqrt{\frac{3}{2}}&0&1\\0&0&0&1&0\end{pmatrix},\begin{pmatrix}0&-i&0&0&0\\i&0&-i\sqrt{\frac{3}{2}}&0&0\\0&i\sqrt{\frac{3}{2}}&0&-i\sqrt{\frac{3}{2}}&0\\0&0&i\sqrt{\frac{3}{2}}&0&-i\\0&0&0&i&0\end{pmatrix},\begin{pmatrix}2&0&0&0&0\\0&1&0&0&0\\0&0&0&0&0\\0&0&0&-1&0\\0&0&0&0&-2\end{pmatrix}\right\}$$

RepMatrices

650 rep of E(6)

```
Timing[matrices = RepMatrices[E6, {1, 0, 0, 0, 1, 0}];]
{77.046875, Null}
```

Code is fast in most cases

In general, what are the properties of these matrices? They are **hermitian** and conform to the usual **trace condition** used in Particle Physics

```
And @@ Table[matrix == ConjugateTranspose[matrix], {matrix, matrices}]
True
```

✓ $T_a^\dagger = T_a$

```
DynkinIndex[E6, {1, 0, 0, 0, 1, 0}] IdentityMatrix[DimR[E6, Adjoint[E6]]] ==
 Table[Tr[matrix1.matrix2], {matrix1, matrices}, {matrix2, matrices}]
True
```

✓ $\mathrm{Tr}\,(T_a T_b) = T(R)\delta_{ab}$

RepMatrices

Real representations: a word of caution

The basis used by **Susyno always keeps a maximum number of generators in diagonal form** (these generators are always the last ones listed)

```
MatrixForm /@ RepMatrices[SU2, {2}]
```

3 of SU(2) $\longrightarrow$ $\left\{ \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}, \begin{pmatrix} 0 & -\frac{i}{\sqrt{2}} & 0 \\ \frac{i}{\sqrt{2}} & 0 & -\frac{i}{\sqrt{2}} \\ 0 & \frac{i}{\sqrt{2}} & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \right\}$

This is a **perfectly fine basis**, as long as one is aware of it. In fact, this basis is the **best one to read off the quantum numbers** of the representation components

An important consequence of this is that the matrices of real representations are complex. **Adjust results if needed**.

The matrices of a real representation in a real basis must all be anti-symmetric, therefore they cannot be diagonal

EASY TO CHECK

Invariants

CB coefficients are ubiquitous in model building

They are needed to write down a Lagrangian of a gauge theory

Transformation matrices? As given by RepMatrices

Computes the **Clebsch-Gordon coefficients** of a product of Lie group representations.

In other words, it calculates the **linear combinations of field components which are group invariant**.

field a   field b   ...

Example

2 × **2** in $SU(2)$

```
Invariants[SU2, {{1}, {1}}]
{a[2] b[1] - a[1] b[2]}

Invariants[SU2, {{1}, {1}}, TensorForm → True][[1, 1]] // MatrixForm
```

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Invariants

~16s

27 rep of E(6) **27** rep of E(6) **351** rep of E(6)

```
Invariants[E6, {{1, 0, 0, 0, 0, 0}, {1, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 0, 0}}]
```

$$\left\{-\frac{3^{3/4}\,a[2]\,b[1]\,c[1]}{\sqrt{2}\;13^{1/4}}+\frac{3^{3/4}\,a[1]\,b[2]\,c[1]}{\sqrt{2}\;13^{1/4}}-\frac{3^{3/4}\,a[3]\,b[1]\,c[2]}{\sqrt{2}\;13^{1/4}}+\frac{3^{3/4}\,a[1]\,b[3]\,c[2]}{\sqrt{2}\;13^{1/4}}-\frac{3^{3/4}\,a[4]\,b[1]\,c[3]}{\sqrt{2}\;13^{1/4}}+\frac{3^{3/4}\,a[1]\,b[4]\,c[3]}{\sqrt{2}\;13^{1/4}}-\frac{3^{3/4}\,a[5]\,b[1]\,c[4]}{\sqrt{2}\;13^{1/4}}+\cdots 1851 \cdots+\frac{3^{3/4}\,a[24]\,b[27]\,c[348]}{\sqrt{2}\;13^{1/4}}-\frac{3^{3/4}\,a[26]\,b[25]\,c[349]}{\sqrt{2}\;13^{1/4}}+\frac{3^{3/4}\,a[25]\,b[26]\,c[349]}{\sqrt{2}\;13^{1/4}}-\frac{3^{3/4}\,a[27]\,b[25]\,c[350]}{\sqrt{2}\;13^{1/4}}+\frac{3^{3/4}\,a[25]\,b[27]\,c[350]}{\sqrt{2}\;13^{1/4}}-\frac{3^{3/4}\,a[27]\,b[26]\,c[351]}{\sqrt{2}\;13^{1/4}}+\frac{3^{3/4}\,a[26]\,b[27]\,c[351]}{\sqrt{2}\;13^{1/4}}\right\}$$

large output | **show less** | **show more** | **show all** | **set size limit...**

IMPORTANT QUESTION
How are the Clebsch-Gordon coefficients normalized?

ANSWER: for $\sum_{i1,i2,i3,\cdots,in} c_{i1,i2,\cdots,in}\Phi^{(1)}_{i1}\Phi^{(2)}_{i2}\cdots\Phi^{(n)}_{in}$,

$$\sum_{i1,i2,i3,\cdots,in} \left|c_{i1,i2,\cdots,in}\right|^2=\sqrt{\dim\left(\Phi^{(1)}\right)\dim\left(\Phi^{(2)}\right)\cdots\dim\left(\Phi^{(n)}\right)}$$

ALWAYS

E.g.: if **a[1]b[2]-a[2]b[1]** is an invariant combination of two SU(2) doublets, so is any multiple of this

Two SU(2) doublets form the invariants x(a[1]b[2]-a[2]b[1]) for any x. **Which x does Susyno take?**

$$x^2+(-x)^2=\sqrt{2\times 2}\Rightarrow |x|=1$$

DecomposeRep

Decomposes some (irreducible) representation of a group G into irreducible components of some subgroup of G. In other words, it **calculates branching rules**

Example 1

What are the $SU(3) \times SU(2) \times U(1)$ representations in the **70** of $SU(5)$?

```
group = {SU5}; subgroup = {SU3, SU2, U1}; representation = {{2, 0, 0, 1}} ; (* 70 of {SU(5)} *)
```

$$\text{prjMatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & -4 & -6 & -3 \end{pmatrix};$$ (* Use RegularSubgroupProjectionMatrix to find it *)

```
DecomposeRep[group, representation, subgroup, prjMatrix, UseName → True]
```

$\{15\otimes1\otimes-2,\ \overline{6}\otimes2\otimes-7,\ 8\otimes2\otimes3,\ 3\otimes3\otimes-2,\ 3\otimes1\otimes-2,\ \overline{3}\otimes3\otimes8,\ 1\otimes4\otimes3,\ 1\otimes2\otimes3\}$

DecomposeRep

Example 2

List the $SO(10) \rightarrow SU(5) \times U(1)$ branching rules for all SO(10) representations up to size 700

CODE

```
SO10reps = RepsUpToDimN[SO10, 700];

                       ⎛1  1  0  0   0⎞
                       ⎜0  0  1  0   1⎟
SOtoSU5U1ProjMatrix =  ⎜0  0  0  1   0⎟ ; (* Use RegularSubgroupProjectionMatrix *)
                       ⎜0  1  1  0   0⎟
                       ⎝2  0  2  1  -1⎠

data =
 Table[{Style[RepName[SO10, rep], Darker[Red]],
   DecomposeRep[{SO10}, {rep}, {SU5, U1}, SOtoSU5U1ProjMatrix, UseName → True]}, {rep, SO10reps}];
Grid[Prepend[data, Style[#, Bold] & /@ {"SO(10) rep", "SU(5) x U(1) content"}], Frame → All,
 FrameStyle → LightGray]
```

DecomposeRep

OUTPUT

SO(10) rep	SU(5) x U(1) content
1	$\{1\otimes 0\}$
10	$\{5\otimes 2, \overline{5}\otimes -2\}$
16	$\{10\otimes -1, \overline{5}\otimes 3, 1\otimes -5\}$
$\overline{16}$	$\{5\otimes -3, \overline{10}\otimes 1, 1\otimes 5\}$
45	$\{24\otimes 0, 10\otimes 4, \overline{10}\otimes -4, 1\otimes 0\}$
54	$\{15\otimes 4, 24\otimes 0, \overline{15}\otimes -4\}$
120	$\{\overline{45}\otimes -2, 45\otimes 2, 5\otimes 2, 10\otimes -6, \overline{10}\otimes 6, \overline{5}\otimes -2\}$
$\overline{126}$	$\{15\otimes -6, \overline{45}\otimes -2, 50\otimes 2, 5\otimes 2, \overline{10}\otimes 6, 1\otimes 10\}$
126	$\{\overline{50}\otimes -2, 45\otimes 2, 10\otimes -6, \overline{15}\otimes 6, \overline{5}\otimes -2, 1\otimes -10\}$
144	$\{15\otimes -1, \overline{45}\otimes 3, 24\otimes -5, 5\otimes 7, 40\otimes -1, 10\otimes -1, \overline{5}\otimes 3\}$
$\overline{144}$	$\{\overline{40}\otimes 1, 24\otimes 5, 45\otimes -3, 5\otimes -3, \overline{15}\otimes 1, \overline{10}\otimes 1, \overline{5}\otimes -7\}$
210	$\{\overline{40}\otimes -4, 75\otimes 0, 24\otimes 0, 5\otimes -8, 40\otimes 4, 10\otimes 4, \overline{10}\otimes -4, \overline{5}\otimes 8, 1\otimes 0\}$
$210'$	$\{\overline{35}\otimes 6, 70\otimes 2, \overline{70}\otimes -2, 35\otimes -6\}$
320	$\{70\otimes 2, \overline{40}\otimes 6, \overline{70}\otimes -2, \overline{45}\otimes -2, 45\otimes 2, 5\otimes 2, 40\otimes -6, \overline{5}\otimes -2\}$
560	$\{175\otimes -1, \overline{50}\otimes 3, \overline{70}\otimes 3, \overline{45}\otimes 3, 75\otimes -5, 24\otimes -5, 45\otimes 7, 40\otimes -1, 10\otimes -1, 10\otimes -1, \overline{10}\otimes -9, \overline{5}\otimes 3, 1\otimes -5\}$
$\overline{560}$	$\{70\otimes -3, \overline{175}\otimes 1, \overline{40}\otimes 1, \overline{45}\otimes -7, 75\otimes 5, 24\otimes 5, 50\otimes -3, 45\otimes -3, 5\otimes -3, 10\otimes 9, \overline{10}\otimes 1, \overline{10}\otimes 1, 1\otimes 5\}$
660	$\{\overline{70'}\otimes 8, 160\otimes 4, 200\otimes 0, \overline{160}\otimes -4, 70'\otimes -8\}$
672	$\{\overline{35}\otimes -9, 126\otimes -5, 210\otimes -1, 15\otimes -1, \overline{175''}\otimes 3, \overline{45}\otimes 3, 50\otimes 7, 5\otimes 7, \overline{10}\otimes 11, 1\otimes 15\}$
$\overline{672}$	$\{175''\otimes -3, \overline{210}\otimes 1, \overline{50}\otimes -7, \overline{126}\otimes 5, 45\otimes -3, 35\otimes 9, 10\otimes -11, \overline{15}\otimes 1, \overline{5}\otimes -7, 1\otimes -15\}$

Takes around 5 minutes to extend this table up to all reps of size 10000 or smaller

SubgroupEmbeddingCoefficients

Calculates the **relations between the subgroup invariants** (i.e., Clebsch-Gordon coefficients) in a theory which is symmetric under a bigger group

Easier to explain with an example!

Consider the $SO(10)$ invariant combination of the representations $\mathbf{16} \times \mathbf{16} \times \mathbf{10}$. We can see it from a $SU(3) \times SU(2) \times U(1)$ perspective:

$$\mathbf{16} \times \mathbf{16} \times \mathbf{10}$$

$$\begin{matrix} \mathbf{3} \otimes \mathbf{2} \otimes \frac{1}{6} \\ \overline{\mathbf{3}} \otimes \mathbf{1} \otimes -\frac{2}{3} \\ \overline{\mathbf{3}} \otimes \mathbf{1} \otimes \frac{1}{3} \\ \mathbf{1} \otimes \mathbf{2} \otimes -\frac{1}{2} \\ \mathbf{1} \otimes \mathbf{1} \otimes 0 \\ \mathbf{1} \otimes \mathbf{1} \otimes 1 \end{matrix} \times \begin{matrix} \mathbf{3} \otimes \mathbf{2} \otimes \frac{1}{6} \\ \overline{\mathbf{3}} \otimes \mathbf{1} \otimes -\frac{2}{3} \\ \overline{\mathbf{3}} \otimes \mathbf{1} \otimes \frac{1}{3} \\ \mathbf{1} \otimes \mathbf{2} \otimes -\frac{1}{2} \\ \mathbf{1} \otimes \mathbf{1} \otimes 0 \\ \mathbf{1} \otimes \mathbf{1} \otimes 1 \end{matrix} \times \begin{matrix} \mathbf{3} \otimes \mathbf{1} \otimes -\frac{1}{3} \\ \overline{\mathbf{3}} \otimes \mathbf{1} \otimes \frac{1}{3} \\ \mathbf{1} \otimes \mathbf{2} \otimes \frac{1}{2} \\ \mathbf{1} \otimes \mathbf{2} \otimes -\frac{1}{2} \end{matrix}$$

1 SO(10) invariant

=

Linear combination of the 17 subgroup invariants

SubgroupEmbeddingCoefficients

CODE

```
(* INPUT DATA *)
group = {SO10};
rep16 = {{0, 0, 0, 0, 1}};
rep10 = {{1, 0, 0, 0, 0}};
subgroup = {SU3, SU2, U1};
breakInfo = {{1, {2, 1}}, {1, {4}}, {1, 1/6}};

(* CALCULATE THE EMBEDDING COEFFICIENTS *)
result = SubgroupEmbeddingCoefficients[group, {rep16, rep16, rep10}, subgroup, breakInfo];

(* THE REST OF THE CODE BELOW IS JUST TO FORMAT THE OUTPUT IN A NICE WAY *)
coefficients = result[[2, 5, 1]];

productFields = Map[RepName[subgroup, #] &,
   Flatten[ConstantArray[Extract[result[[2, 3]], #[[1]]], #[[2]]] & /@ result[[2, 4, All, {2, 3}]], 1], {2}];

Print["The product of ",
  {Style[RepName[group, #1], {Darker[Blue], Bold}], Style[RepName[group, #2], {Darker[Green], Bold}],
     Style[RepName[group, #3], {Darker[Red], Bold}]} & @@ {rep16, rep16, rep10}, " of SO(10) is the same as"];
table =
  Table[Row[{If[coefficients[[i]] > 0, Style["+", Darker[Gray]], ""], Style[coefficients[[i]], Darker[Gray]],
     {Style[#1, {Darker[Blue], Bold}], Style[#2, {Darker[Green], Bold}], Style[#3, {Darker[Red], Bold}]} & @@
      productFields[[i]]}], {i, Length[coefficients]}];
Print[Row[table, " "]];
Print["in SU(3) x SU(2) x U(1)"];
```

SubgroupEmbeddingCoefficients

OUTPUT

The product of {16, 16, 10} of SO(10) is the same as

$$-\frac{2^{3/4}}{15^{1/4}}\left\{3\otimes2\otimes\frac{1}{6},\ 3\otimes2\otimes\frac{1}{6},\ 3\otimes1\otimes-\frac{1}{3}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{3\otimes2\otimes\frac{1}{6},\ \bar{3}\otimes1\otimes-\frac{2}{3},\ 1\otimes2\otimes\frac{1}{2}\right\}$$

$$+\left(\frac{2}{5}\right)^{1/4}\left\{3\otimes2\otimes\frac{1}{6},\ \bar{3}\otimes1\otimes\frac{1}{3},\ 1\otimes2\otimes-\frac{1}{2}\right\}-\left(\frac{2}{5}\right)^{1/4}\left\{3\otimes2\otimes\frac{1}{6},\ 1\otimes2\otimes-\frac{1}{2},\ \bar{3}\otimes1\otimes\frac{1}{3}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{\bar{3}\otimes1\otimes-\frac{2}{3},\ 3\otimes2\otimes\frac{1}{6},\ 1\otimes2\otimes\frac{1}{2}\right\}$$

$$+\frac{2^{3/4}}{15^{1/4}}\left\{\bar{3}\otimes1\otimes-\frac{2}{3},\ \bar{3}\otimes1\otimes\frac{1}{3},\ \bar{3}\otimes1\otimes\frac{1}{3}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{\bar{3}\otimes1\otimes-\frac{2}{3},\ 1\otimes1\otimes1,\ 3\otimes1\otimes-\frac{1}{3}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{\bar{3}\otimes1\otimes\frac{1}{3},\ 3\otimes2\otimes\frac{1}{6},\ 1\otimes2\otimes-\frac{1}{2}\right\}$$

$$+\frac{2^{3/4}}{15^{1/4}}\left\{\bar{3}\otimes1\otimes\frac{1}{3},\ \bar{3}\otimes1\otimes-\frac{2}{3},\ \bar{3}\otimes1\otimes\frac{1}{3}\right\}-\left(\frac{2}{5}\right)^{1/4}\left\{\bar{3}\otimes1\otimes\frac{1}{3},\ 1\otimes1\otimes0,\ 3\otimes1\otimes-\frac{1}{3}\right\}-\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes2\otimes-\frac{1}{2},\ 3\otimes2\otimes\frac{1}{6},\ \bar{3}\otimes1\otimes\frac{1}{3}\right\}$$

$$+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes2\otimes-\frac{1}{2},\ 1\otimes1\otimes0,\ 1\otimes2\otimes\frac{1}{2}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes2\otimes-\frac{1}{2},\ 1\otimes1\otimes1,\ 1\otimes2\otimes-\frac{1}{2}\right\}-\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes1\otimes0,\ \bar{3}\otimes1\otimes\frac{1}{3},\ 3\otimes1\otimes-\frac{1}{3}\right\}$$

$$+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes1\otimes0,\ 1\otimes2\otimes-\frac{1}{2},\ 1\otimes2\otimes\frac{1}{2}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes1\otimes1,\ \bar{3}\otimes1\otimes-\frac{2}{3},\ 3\otimes1\otimes-\frac{1}{3}\right\}+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes1\otimes1,\ 1\otimes2\otimes-\frac{1}{2},\ 1\otimes2\otimes-\frac{1}{2}\right\}$$

in SU(3) x SU(2) x U(1)

Why the strange factors?
The answer is simple: **it is due to the program's default normalization of the invariants** (i.e., Clebsch-Gordon coefficients)

Both the SO(10) Clebsch-Gordon coefficients and the subgroup ones

Here, we can opt to quickly **eliminate the SO(10) Clebsch-Gordon's normalization issue** by just looking at **ratios of these embedding coefficients**

On the other hand, most of the $SU(3)\times SU(2)\times U(1)$ invariants shown here are **normalized as usually expected**

The only ones which are not are the ones involving $\mathbf{3}\times\mathbf{3}\times\mathbf{3}$ of SU(3)

SubgroupEmbeddingCoefficients

y_u

$+\left(\frac{2}{5}\right)^{1/4}\left\{3\otimes2\otimes\frac{1}{6},\ \bar{3}\otimes1\otimes-\frac{2}{3},\ 1\otimes2\otimes\frac{1}{2}\right\}$

y_d

$+\left(\frac{2}{5}\right)^{1/4}\left\{3\otimes2\otimes\frac{1}{6},\ \bar{3}\otimes1\otimes\frac{1}{3},\ 1\otimes2\otimes-\frac{1}{2}\right\}$

The 4 coefficients are the same

y_ν

$+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes2\otimes-\frac{1}{2},\ 1\otimes1\otimes0,\ 1\otimes2\otimes\frac{1}{2}\right\}$

y_ℓ

$+\left(\frac{2}{5}\right)^{1/4}\left\{1\otimes2\otimes-\frac{1}{2},\ 1\otimes1\otimes1,\ 1\otimes2\otimes-\frac{1}{2}\right\}$

Conclusion

$$\underset{\mathbf{16}\times\mathbf{16}\times\mathbf{10}}{SO(10)} \Rightarrow y_u = y_d = y_\ell = y_\nu$$

Now, if we just change in the input {1,0,0,0,0} (the **10**) to {0,0,0,2,0} (the **126**) ...

With minimal code change

$$\underset{\mathbf{16}\times\mathbf{16}\times\mathbf{126}}{SO(10)} \Rightarrow y_u = y_d = -\frac{1}{3}y_\ell = -\frac{1}{3}y_\nu$$

Georgi-Jarlskog relation
1979

Summary

Summary

The Susyno program can calculate

RGEs of a SUSY model

Generate list of model parameters, check anomalies, ...

Group theory quantities

Useful for model building in general

Quick start

1. Download from `http://web.ist.utl.pt/renato.fonseca/susyno.html`
2. Unpack the folder `Susyno` and drop it inside `(Mathematica base directory)/AddOns/Applications`
3. Type in Mathematica's front the following: `<<Susyno``
   The built-in documentation becomes readily available

Thank you

Extra slides

Input example: the MSSM

Pick a name for the model (any)

```
author[MSSM] ^= "Me";
date[MSSM] ^= "14:50, 25 July 2014";
```

Provide some optional data

```
group[MSSM] ^= {U1, SU2, SU3};
```

Specify the model's gauge group (any)

```
normalization = Sqrt[3 / 5];
u = {-2 / 3 normalization, {0}, {0, 1}};
d = {1 / 3 normalization, {0}, {0, 1}};
Q = {1 / 6 normalization, {1}, {1, 0}};
e = {normalization, {0}, {0, 0}};
L = {-1 / 2 normalization, {1}, {0, 0}};
Hu = {1 / 2 normalization, {1}, {0, 0}};
Hd = {-1 / 2 normalization, {1}, {0, 0}};
```

Specify the model's representations (any)

```
reps[MSSM] ^= {u, d, Q, e, L, Hu, Hd};
fieldNames[MSSM] ^= {"u", "d", "Q", "e", "L", "Hu", "Hd"};
```

```
nFlavs[MSSM] ^= {3, 3, 3, 3, 3, 1, 1};
discreteSym[MSSM] ^= {-1, -1, -1, -1, -1, 1, 1};
```

Provide the number of flavors and the charges under any abelian symmetry

```
GenerateModel[MSSM, CalculateEverything → True]
```

Tell the program to calculate the model's RGEs (among other things)

Output example: the MSSM

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

Model name

MSSM

Author

Me

Date

14:50, 25 July 2014

Output example: the MSSM

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

U1 x SU2 x SU3

GOOD NEWS: The model is gauge anomaly free.

>>> Extra information

This data is contained in the group[MSSM] variable.

Output example: the MSSM

Model Information | Gauge group | Representations | Parameters in model | Lagrangian | BetaFunctions

	u	d	Q	e	L	Hu	Hd
U1	$-\frac{2}{\sqrt{15}}$	$\frac{1}{\sqrt{15}}$	$\frac{1}{2\sqrt{15}}$	$\sqrt{\frac{3}{5}}$	$-\frac{\sqrt{\frac{3}{5}}}{2}$	$\frac{\sqrt{\frac{3}{5}}}{2}$	$-\frac{\sqrt{\frac{3}{5}}}{2}$
SU2	{0}	{0}	{1}	{0}	{1}	{1}	{1}
SU3	{0, 1}	{0, 1}	{1, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}
#Flavors	3	3	3	3	3	1	1
R-Charges	-1	-1	-1	-1	-1	1	1

>>> Extra information

This data is contained in the reps[MSSM] variable.

Output example: the MSSM

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

^ Gauge coupling constants

g[1]
g[2]
g[3]

^ Gaugino masses

M[1]
M[2]
M[3]

^ Superpotential trilinear parameters

y[{u, Q, Hu}, {f[1], f[2]}]
y[{d, Q, Hd}, {f[1], f[2]}]
y[{e, L, Hd}, {f[1], f[2]}]

^ Superpotential bilinear parameters

μ[{Hu, Hd}]

^ Superpotential linear parameters

---

^ Soft trilinear parameters

h[{u, Q, Hu}, {f[1], f[2]}]
h[{d, Q, Hd}, {f[1], f[2]}]
h[{e, L, Hd}, {f[1], f[2]}]

^ Soft bilinear parameters

b[{Hu, Hd}]

^ Soft linear parameters

---

^ Soft masses

m2[{u, u}, {f[1], f[2]}]
m2[{d, d}, {f[1], f[2]}]
m2[{Q, Q}, {f[1], f[2]}]
m2[{e, e}, {f[1], f[2]}]
m2[{L, L}, {f[1], f[2]}]
m2[{Hu, Hu}]
m2[{Hd, Hd}]

...

Output example: the MSSM

Model Information | Gauge group | Representations | Parameters in model | Lagrangian | BetaFunctions

```
^ Superpotential (trilinear terms)
  y[{e, L, Hd}, {f[1], f[2]}] (e[f[1]] Hd[2] L[f[2]][1] - e[f[1]] Hd[1] L[f[2]][2]) +
   y[{d, Q, Hd}, {f[1], f[2]}] (Hd[2] d[f[1]][1] Q[f[2]][1, 1] + Hd[2] d[f[1]][2] Q[f[2]][1, 2] +
      Hd[2] d[f[1]][3] Q[f[2]][1, 3] - Hd[1] d[f[1]][1] Q[f[2]][2, 1] - Hd[1] d[f[1]][2] Q[f[2]][2, 2] - Hd[1] d[f[1]][3] Q[f[2]][2, 3]) +
   y[{u, Q, Hu}, {f[1], f[2]}] (Hu[2] Q[f[2]][1, 1] u[f[1]][1] - Hu[1] Q[f[2]][2, 1] u[f[1]][1] + Hu[2] Q[f[2]][1, 2] u[f[1]][2] -
      Hu[1] Q[f[2]][2, 2] u[f[1]][2] + Hu[2] Q[f[2]][1, 3] u[f[1]][3] - Hu[1] Q[f[2]][2, 3] u[f[1]][3])
^ Superpotential (bilinear terms)
  μ[{Hu, Hd}] (Hd[2] Hu[1] - Hd[1] Hu[2])
^ Superpotential (linear terms)
  0
^ Soft SUSY breaking Lagrangian (trilinear terms)
  h[{e, L, Hd}, {f[1], f[2]}] (e[f[1]] Hd[2] L[f[2]][1] - e[f[1]] Hd[1] L[f[2]][2]) +
   h[{d, Q, Hd}, {f[1], f[2]}] (Hd[2] d[f[1]][1] Q[f[2]][1, 1] + Hd[2] d[f[1]][2] Q[f[2]][1, 2] +
      Hd[2] d[f[1]][3] Q[f[2]][1, 3] - Hd[1] d[f[1]][1] Q[f[2]][2, 1] - Hd[1] d[f[1]][2] Q[f[2]][2, 2] - Hd[1] d[f[1]][3] Q[f[2]][2, 3]) +
   h[{u, Q, Hu}, {f[1], f[2]}] (Hu[2] Q[f[2]][1, 1] u[f[1]][1] - Hu[1] Q[f[2]][2, 1] u[f[1]][1] + Hu[2] Q[f[2]][1, 2] u[f[1]][2] -
      Hu[1] Q[f[2]][2, 2] u[f[1]][2] + Hu[2] Q[f[2]][1, 3] u[f[1]][3] - Hu[1] Q[f[2]][2, 3] u[f[1]][3])
^ Soft SUSY breaking Lagrangian (bilinear terms)
  b[{Hu, Hd}] (Hd[2] Hu[1] - Hd[1] Hu[2])
^ Soft SUSY breaking Lagrangian (linear terms)
  0
^ Soft SUSY breaking Lagrangian (mass terms)
  m2[{e, e}, {f[1], f[2]}] Conjugate[e][f[2]] e[f[1]] +
   m2[{Hd, Hd}] (Conjugate[Hd][1] Hd[1] + Conjugate[Hd][2] Hd[2]) + m2[{Hu, Hu}] (Conjugate[Hu][1] Hu[1] + Conjugate[Hu][2] Hu[2]) +
   m2[{d, d}, {f[1], f[2]}] (Conjugate[d][f[2]][1] d[f[1]][1] + Conjugate[d][f[2]][2] d[f[1]][2] + Conjugate[d][f[2]][3] d[f[1]][3]) +
   m2[{L, L}, {f[1], f[2]}] (Conjugate[L][f[2]][1] L[f[1]][1] + Conjugate[L][f[2]][2] L[f[1]][2]) +
   m2[{Q, Q}, {f[1], f[2]}] (Conjugate[Q][f[2]][1, 1] Q[f[1]][1, 1] + Conjugate[Q][f[2]][1, 2] Q[f[1]][1, 2] + Conjugate[Q][f[2]][1, 3] Q[f[1]][1, 3] +
      Conjugate[Q][f[2]][2, 1] Q[f[1]][2, 1] + Conjugate[Q][f[2]][2, 2] Q[f[1]][2, 2] + Conjugate[Q][f[2]][2, 3] Q[f[1]][2, 3]) +
   m2[{u, u}, {f[1], f[2]}] (Conjugate[u][f[2]][1] u[f[1]][1] + Conjugate[u][f[2]][2] u[f[1]][2] + Conjugate[u][f[2]][3] u[f[1]][3])

>>> Parameters are shown in dark orange; field heads are shown in blue.
```

...

Output example: the MSSM

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

- g[1]
- g[2]
- g[3]
  - ~~~~~~ $\beta^{(1)}$ ~~~~~~

    $-3\, g[3]^3$
  - ~~~~~~ $\beta^{(2)}$ ~~~~~~

    $\frac{11}{5} g[1]^2 g[3]^3 + 9 g[2]^2 g[3]^3 + 14 g[3]^5 - 4\, \text{Conjugate}[y[\{d, Q, Hd\}, \{f[1], f[3]\}]]\, g[3]^3\, y[\{d, Q, Hd\}, \{f[1], f[3]\}] -$
    $4\, \text{Conjugate}[y[\{u, Q, Hu\}, \{f[1], f[3]\}]]\, g[3]^3\, y[\{u, Q, Hu\}, \{f[1], f[3]\}]$
- M[1]
- M[2]
- M[3]
- y[{u, Q, Hu}, {f[1], f[2]}]
- y[{d, Q, Hd}, {f[1], f[2]}]
- y[{e, L, Hd}, {f[1], f[2]}]
- μ[{Hu, Hd}]
- h[{u, Q, Hu}, {f[1], f[2]}]
- h[{d, Q, Hd}, {f[1], f[2]}]
- h[{e, L, Hd}, {f[1], f[2]}]
- b[{Hu, Hd}]
- m2[{u, u}, {f[1], f[2]}]
- m2[{d, d}, {f[1], f[2]}]
- m2[{Q, Q}, {f[1], f[2]}]
- m2[{e, e}, {f[1], f[2]}]
- m2[{L, L}, {f[1], f[2]}]
- m2[{Hu, Hu}]
- m2[{Hd, Hd}]

>>> Extra information

...

Input example: minimal $SO(10)$

```
group[MinimalSUSYSO10GUT] ^= {SO10};

Ψ = {{0, 0, 0, 0, 1}};    (* 16-dim representation *)
Φ = {{0, 0, 0, 1, 1}};    (* 210-dim representation *)
Δ = {{0, 0, 0, 0, 2}};    (* 126-dim representation *)
Δb = {{0, 0, 0, 2, 0}};   (* 126-dim representation (conj.) *)
H = {{1, 0, 0, 0, 0}};    (* 10-dim representation *)

reps[MinimalSUSYSO10GUT] ^= {Ψ, Φ, Δ, Δb, H};
fieldNames[MinimalSUSYSO10GUT] ^= {"Ψ", "Φ", "Δ", "Δ̄", "H"};

nFlavs[MinimalSUSYSO10GUT] ^= {3, 1, 1, 1, 1};
discreteSym[MinimalSUSYSO10GUT] ^= {1, 1, 1, 1, 1};

GenerateModel[MinimalSUSYSO10GUT, CalculateEverything → True]
```

Output example: minimal SO(10)

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

^ Model name

MinimalSUSYSO10GUT

Output example: minimal SO(10)

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

SO10

GOOD NEWS: The model is gauge anomaly free.

>>> Extra information

This data is contained in the group[MinimalSUSYSO10GUT] variable.

Output example: minimal SO(10)

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

	Ψ	Φ	Δ	$\bar{\Delta}$	H
SO10	{0, 0, 0, 0, 1}	{0, 0, 0, 1, 1}	{0, 0, 0, 0, 2}	{0, 0, 0, 2, 0}	{1, 0, 0, 0, 0}
#Flavours	3	1	1	1	1
R-Charges	1	1	1	1	1

>>> Extra information

This data is contained in the reps[MinimalSUSYSO10GUT] variable.

Output example: minimal SO(10)

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

^ Gauge coupling constants

g[1]

^ Gaugino masses

M[1]

^ Superpotential trilinear parameters

y[{Ψ, Ψ, Δ̄}, {f[1], f[2]}] (symmetric under a permutation of the flavor indices {f[1], f[2]})
y[{Ψ, Ψ, H}, {f[1], f[2]}] (symmetric under a permutation of the flavor indices {f[1], f[2]})
y[{Φ, Φ, Φ}]
y[{Φ, Δ, Δ̄}]
y[{Φ, Δ, H}]
y[{Φ, Δ̄, H}]

^ Superpotential bilinear parameters

μ[{Φ, Φ}]
μ[{Δ, Δ̄}]
μ[{H, H}]

^ Superpotential linear parameters

---

^ Soft trilinear parameters

h[{Ψ, Ψ, Δ̄}, {f[1], f[2]}] (symmetric under a permutation of the flavor indices {f[1], f[2]})
h[{Ψ, Ψ, H}, {f[1], f[2]}] (symmetric under a permutation of the flavor indices {f[1], f[2]})
h[{Φ, Φ, Φ}]
h[{Φ, Δ, Δ̄}]
h[{Φ, Δ, H}]
h[{Φ, Δ̄, H}]

^ Soft bilinear parameters

b[{Φ, Φ}]
b[{Δ, Δ̄}]
b[{H, H}]

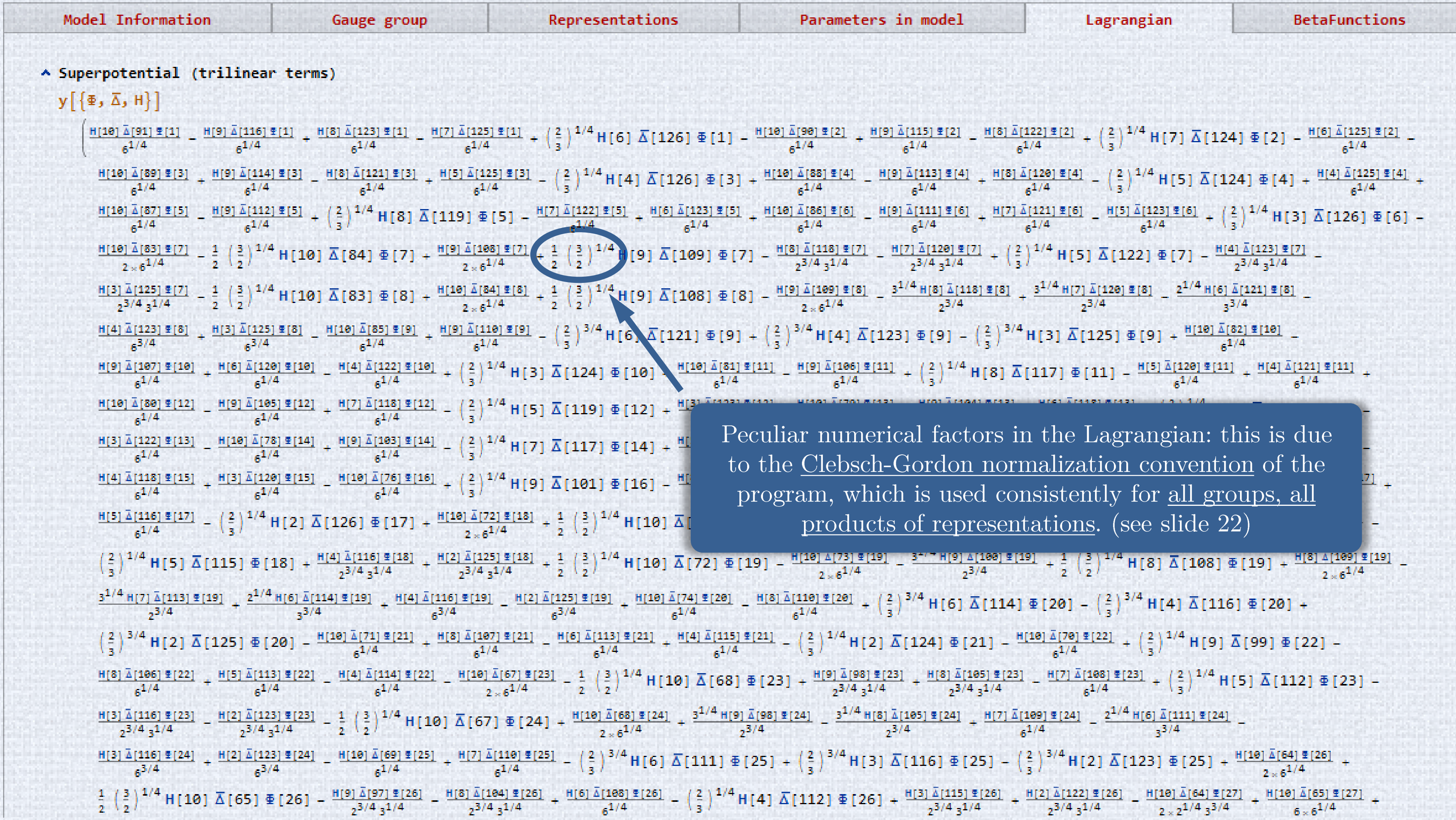
^ Soft linear parameters

---

^ Soft masses

...

Output example: minimal SO(10)



Peculiar numerical factors in the Lagrangian: this is due to the Clebsch-Gordon normalization convention of the program, which is used consistently for all groups, all products of representations. (see slide 22)

…

Output example: minimal SO(10)

Model Information	Gauge group	Representations	Parameters in model	Lagrangian	BetaFunctions

g[1]

M[1]

~~~~~ $\beta^{(1)}$ ~~~~~

$218\, g[1]^2 M[1]$

~~~~~ $\beta^{(2)}$ ~~~~~

$1624\sqrt{\frac{2}{3}}\,\text{Conjugate}[y[\{\Phi,\bar{\Delta},H\}]]\,g[1]^2\,h[\{\Phi,\bar{\Delta},H\}] + 1624\sqrt{\frac{2}{3}}\,\text{Conjugate}[y[\{\Phi,\Delta,H\}]]\,g[1]^2\,h[\{\Phi,\Delta,H\}] +$

$2072\sqrt{\frac{42}{5}}\,\text{Conjugate}[y[\{\Phi,\Delta,\bar{\Delta}\}]]\,g[1]^2\,h[\{\Phi,\Delta,\bar{\Delta}\}] + 112\sqrt{210}\,\text{Conjugate}[y[\{\Phi,\Phi,\Phi\}]]\,g[1]^2\,h[\{\Phi,\Phi,\Phi\}] +$

$56\sqrt{\frac{2}{5}}\,\text{Conjugate}[y[\{\Psi,\Psi,H\},\{f[1],f[3]\}]]\,g[1]^2\,h[\{\Psi,\Psi,H\},\{f[1],f[3]\}] +$

$\frac{152}{3}\sqrt{14}\,\text{Conjugate}[y[\{\Psi,\Psi,\bar{\Delta}\},\{f[1],f[3]\}]]\,g[1]^2\,h[\{\Psi,\Psi,\bar{\Delta}\},\{f[1]$ …

$1624\sqrt{\frac{2}{3}}\,\text{Conjugate}[y[\{\Phi,\bar{\Delta},H\}]]\,g[1]^2\,M[1]\,y[\{\Phi,\bar{\Delta},H\}] - 1624\sqrt{\frac{2}{3}}\,\text{Con}$ …

$2072\sqrt{\frac{42}{5}}\,\text{Conjugate}[y[\{\Phi,\Delta,\bar{\Delta}\}]]\,g[1]^2\,M[1]\,y[\{\Phi,\Delta,\bar{\Delta}\}] - 112\sqrt{210}\,\text{Co}$ …

$56\sqrt{\frac{2}{5}}\,\text{Conjugate}[y[\{\Psi,\Psi,H\},\{f[1],f[3]\}]]\,g[1]^2\,M[1]\,y[\{\Psi,\Psi,H\},\{f[1],f[3]\}] -$

$\frac{152}{3}\sqrt{14}\,\text{Conjugate}[y[\{\Psi,\Psi,\bar{\Delta}\},\{f[1],f[3]\}]]\,g[1]^2\,M[1]\,y[\{\Psi,\Psi,\bar{\Delta}\},\{f[1],f[3]\}]$

$y[\{\Psi,\Psi,\bar{\Delta}\},\{f[1],f[2]\}]$

$y[\{\Psi,\Psi,H\},\{f[1],f[2]\}]$

$y[\{\Phi,\Phi,\Phi\}]$

$y[\{\Phi,\Delta,\bar{\Delta}\}]$

$y[\{\Phi,\Delta,H\}]$

$y[\{\Phi,\bar{\Delta},H\}]$

$\mu[\{\Phi,\Phi\}]$

Peculiar numerical factors in the RGEs are a consequence of the normalization of the Clebsch-Gordon factors used by the program. Conversion to other normalizations is easy (see documentation)

…