NPointFunctions: a calculator of amplitudes and observables in FlexibleSUSY

Uladzimir Khasianevich
Wojciech Kotlarski, Dominik Stöckinger, Alexander Voigt

Institut für Kern- und Teilchenphysik, TU Dresden

Computational Tools for High Energy Physics and Cosmology @ IP2I Lyon 2021

INSTITUT FÜR
KERN- UND
TEILCHENPHYSIK

TECHNISCHE
UNIVERSITÄT
DRESDEN

Overview

- Motivation
- What is `FlexibleSUSY`?
- What is `NPointFunctions`?
- Some applications

Overview

- Motivation

now

Workflow of a phenomenologist:

1) define/get $\mathcal{L}$
2) get vertices, masses, RGE
3) calculate observables
4) make parameter scans

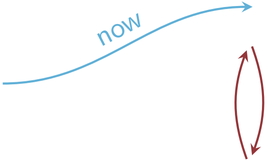
- What is `FlexibleSUSY`?

- What is `NPointFunctions`?

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:

1) define/get $\mathcal{L}_i$

2) get vertices, masses, RGE

3) calculate observables

4) make parameter scans

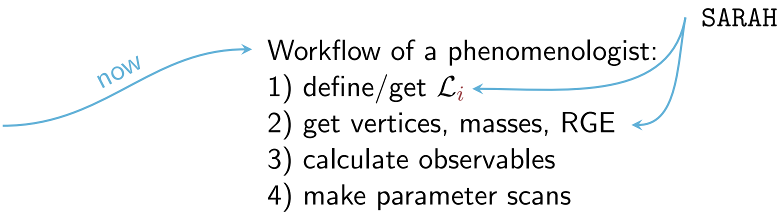
- What is `FlexibleSUSY`?

- What is `NPointFunctions`?

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:

1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables
4) make parameter scans

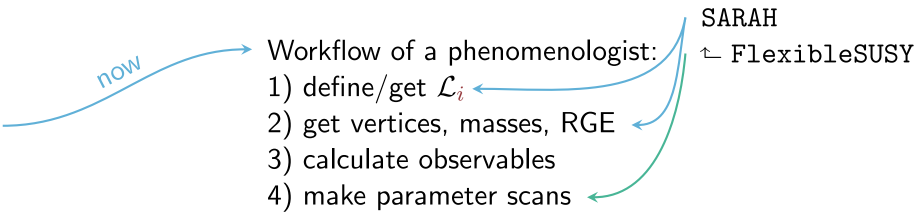
- What is `FlexibleSUSY`?

- What is `NPointFunctions`?

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:
1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables
4) make parameter scans

SARAH
↳ FlexibleSUSY

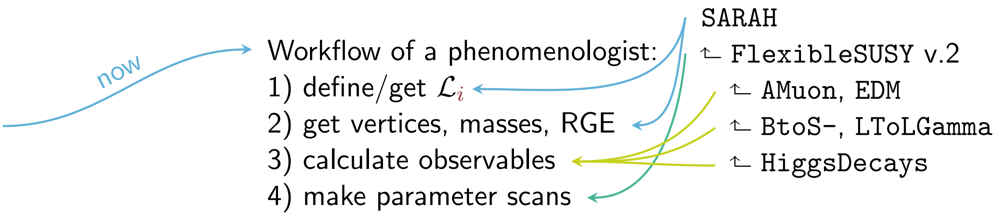
- What is FlexibleSUSY?

- What is NPointFunctions?

- Some applications

Overview

- Motivation



now

Workflow of a phenomenologist:

1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables
4) make parameter scans

`SARAH`
↥ `FlexibleSUSY` v.2
  ↥ `AMuon`, `EDM`
  ↥ `BtoS-`, `LToLGamma`
  ↥ `HiggsDecays`

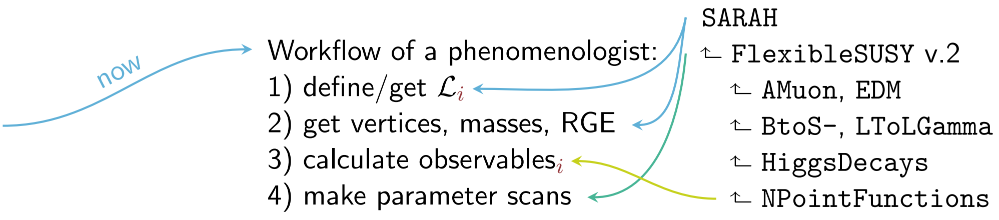
- What is `FlexibleSUSY`?

- What is `NPointFunctions`?

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:
1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables$_i$
4) make parameter scans

SARAH
↿ FlexibleSUSY v.2
  ↿ AMuon, EDM
  ↿ BtoS-, LToLGamma
  ↿ HiggsDecays
  ↿ NPointFunctions

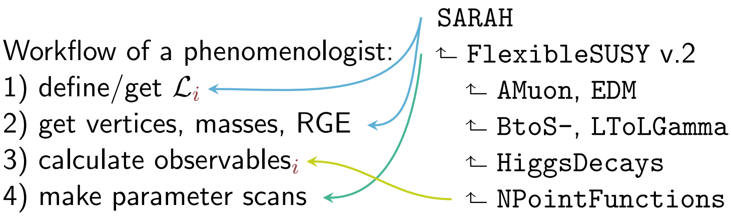
- What is FlexibleSUSY?

- What is NPointFunctions?

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:

1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables$_i$
4) make parameter scans

SARAH
- ↰ FlexibleSUSY v.2
  - ↰ AMuon, EDM
  - ↰ BtoS-, LToLGamma
  - ↰ HiggsDecays
  - ↰ NPointFunctions

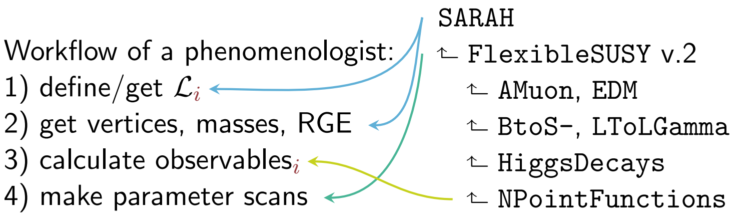
- What is FlexibleSUSY? $\xrightarrow{\text{later}}$ A spectrum generator - generator

- What is NPointFunctions?

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:

1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables$_i$
4) make parameter scans

SARAH

↰ FlexibleSUSY v.2

↰ AMuon, EDM

↰ BtoS-, LToLGamma

↰ HiggsDecays

↰ NPointFunctions

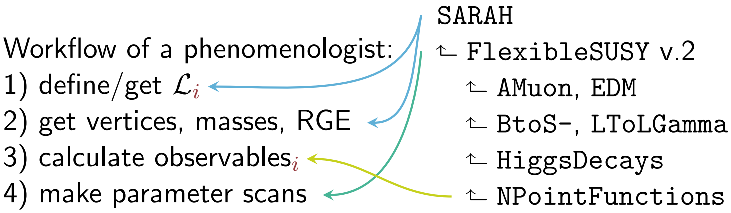
- What is `FlexibleSUSY`? A spectrum generator - generator

- What is `NPointFunctions`? $\xrightarrow{\text{later}}$ A calculator of amplitudes and observables

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:

1) define/get $\mathcal{L}_i$

2) get vertices, masses, RGE

3) calculate observables$_i$

4) make parameter scans

SARAH

↰ FlexibleSUSY v.2

↰ AMuon, EDM

↰ BtoS-, LToLGamma

↰ HiggsDecays

↰ NPointFunctions

- What is FlexibleSUSY?

A spectrum generator - generator

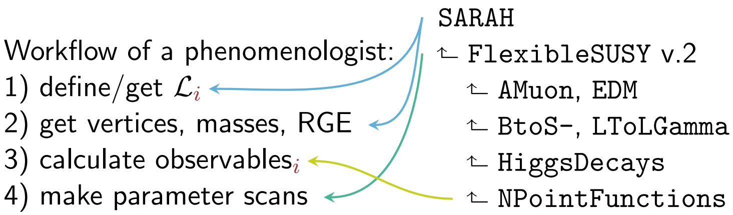
- What is NPointFunctions?

A calculator of amplitudes and observables BRIEFLY

- Some applications

Overview

- Motivation



Workflow of a phenomenologist:
1) define/get $\mathcal{L}_i$
2) get vertices, masses, RGE
3) calculate observables$_i$
4) make parameter scans

`SARAH`
↰ `FlexibleSUSY` v.2
↰ `AMuon`, `EDM`
↰ `BtoS-`, `LToLGamma`
↰ `HiggsDecays`
↰ `NPointFunctions`

- What is `FlexibleSUSY`? A spectrum generator - generator

- What is `NPointFunctions`? A calculator of amplitudes and observables BRIEFLY

- Some applications $\xrightarrow{\text{later}}$ MRSSM, leptoquarks*, Grimus-Neufeld model*

FlexibleSUSY?[1406.2319], [1710.03760] The end-user side.

- Define $\mathcal{L}_i$

#1	#2	#3
`MODEL.m`	`parameters.m`	`particles.m`
model definition	additional information	additional information

`Mathematica` code:

vertices

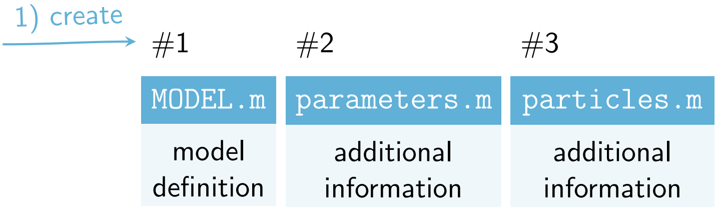
masses

2-loop RGE

EWSB

FlexibleSUSY?[1406.2319], [1710.03760] The end-user side.

- Define $\mathcal{L}_i$



1) create

#1

`MODEL.m`

model definition

#2

`parameters.m`

additional information

#3

`particles.m`

additional information

`Mathematica` code:

vertices

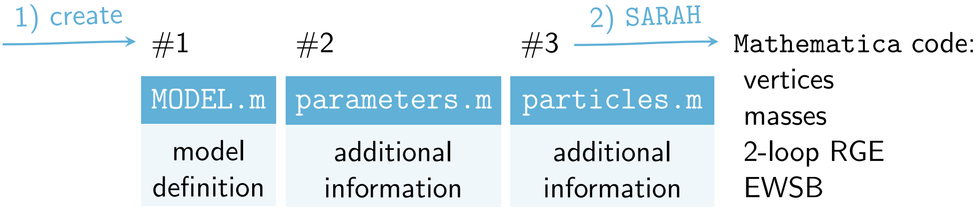
masses

2-loop RGE

EWSB

FlexibleSUSY?[1406.2319], [1710.03760] The end-user side.

- Define $\mathcal{L}_i$



1) create

#1

MODEL.m

model definition

#2

parameters.m

additional information

#3

particles.m

additional information

2) SARAH

Mathematica code:

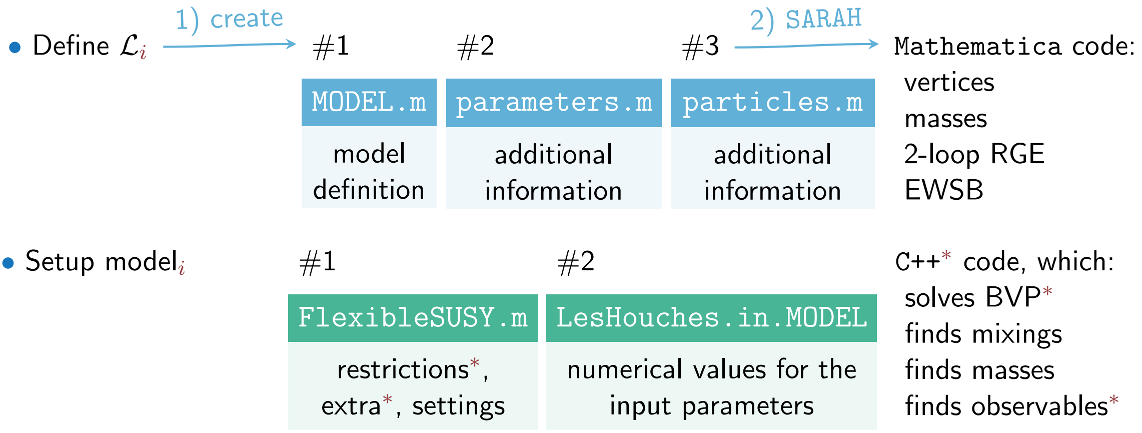
vertices

masses

2-loop RGE

EWSB

`FlexibleSUSY`?[1406.2319], [1710.03760] The end-user side.



- Define $\mathcal{L}_i$ $\xrightarrow{\text{1) create}}$ #1 #2 #3 $\xrightarrow{\text{2) SARAH}}$ `Mathematica` code:

#1	#2	#3
`MODEL.m`	`parameters.m`	`particles.m`
model definition	additional information	additional information

vertices
masses
2-loop RGE
EWSB

- Setup model$_i$

#1	#2
`FlexibleSUSY.m`	`LesHouches.in.MODEL`
restrictions*, extra*, settings	numerical values for the input parameters

`C++`* code, which:
solves BVP*
finds mixings
finds masses
finds observables*

`FlexibleSUSY`?[1406.2319], [1710.03760] The end-user side.

- Define $\mathcal{L}_i$ $\xrightarrow{\text{1) create}}$

  #1 `MODEL.m`: model definition

  #2 `parameters.m`: additional information

  #3 `particles.m`: additional information $\xrightarrow{\text{2) SARAH}}$

  `Mathematica` code:
  vertices
  masses
  2-loop RGE
  EWSB

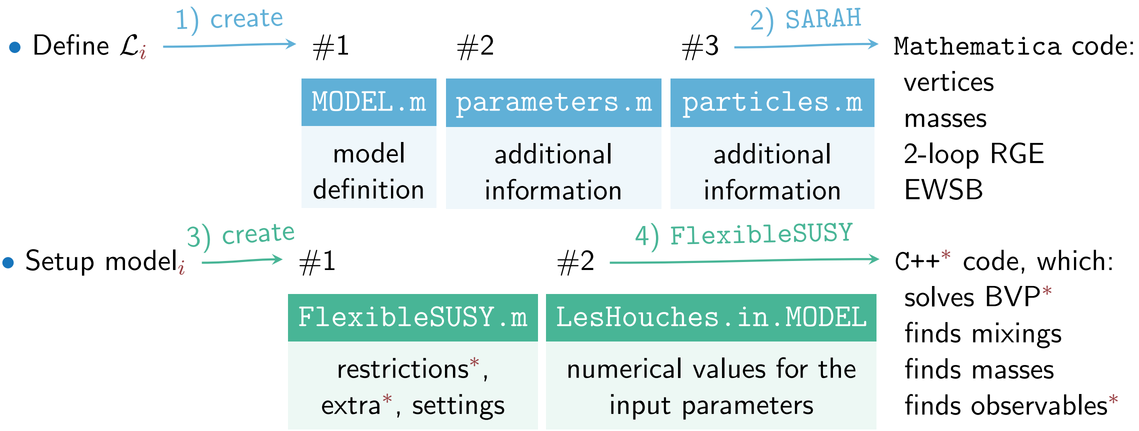
- Setup model$_i$ $\xrightarrow{\text{3) create}}$

  #1 `FlexibleSUSY.m`: restrictions*, extra*, settings

  #2 `LesHouches.in.MODEL`: numerical values for the input parameters

  `C++`* code, which:
  solves BVP*
  finds mixings
  finds masses
  finds observables*

FlexibleSUSY?[1406.2319], [1710.03760] The end-user side.



- Define $\mathcal{L}_i$ 1) create → #1 #2 #3 2) SARAH → Mathematica code:

#1	#2	#3
MODEL.m	parameters.m	particles.m
model definition	additional information	additional information

vertices
masses
2-loop RGE
EWSB

- Setup model$_i$ 3) create → #1 #2 4) FlexibleSUSY → C++* code, which:

#1	#2
FlexibleSUSY.m	LesHouches.in.MODEL
restrictions*, extra*, settings	numerical values for the input parameters

solves BVP*
finds mixings
finds masses
finds observables*

FlexibleSUSY?[1406.2319], [1710.03760] The end-user side.

- Define $\mathcal{L}_i$ $\xrightarrow{\text{1) create}}$ #1 #2 #3 $\xrightarrow{\text{2) SARAH}}$ Mathematica code:

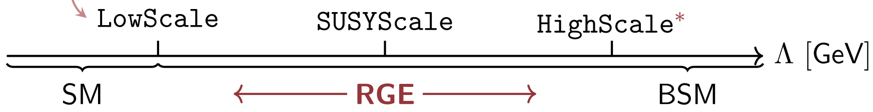
#1	#2	#3
MODEL.m	parameters.m	particles.m
model definition	additional information	additional information

vertices
masses
2-loop RGE
EWSB

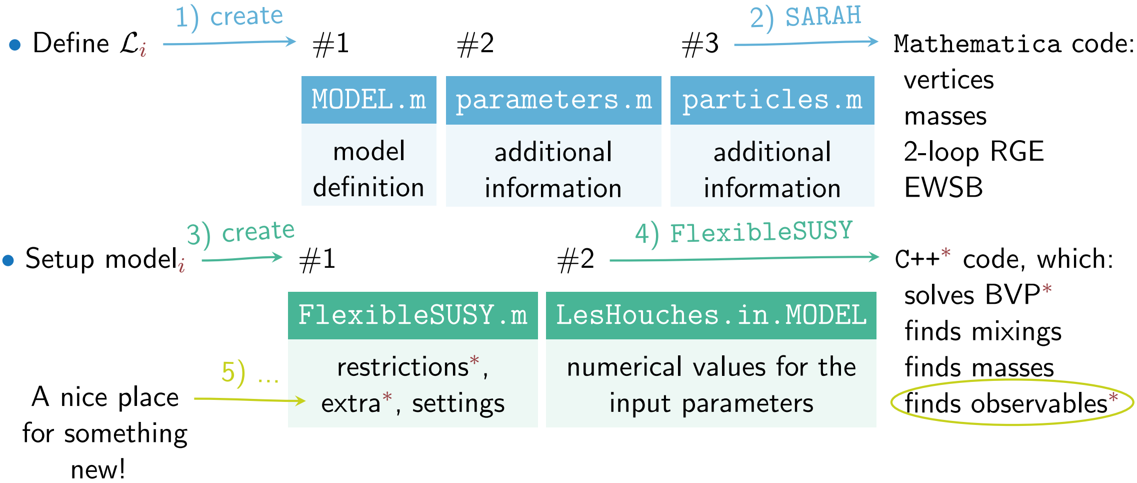
- Setup model$_i$ $\xrightarrow{\text{3) create}}$ #1 #2 $\xrightarrow{\text{4) FlexibleSUSY}}$ C++* code, which:

#1	#2
FlexibleSUSY.m	LesHouches.in.MODEL
restrictions*, extra*, settings	numerical values for the input parameters

solves BVP*
finds mixings
finds masses
finds observables*



FlexibleSUSY?[1406.2319], [1710.03760] The end-user side.



- Define $\mathcal{L}_i$ 1) create → #1 #2 #3 2) SARAH → Mathematica code:

MODEL.m	parameters.m	particles.m
model definition	additional information	additional information

vertices
masses
2-loop RGE
EWSB

- Setup model$_i$ 3) create → #1 #2 4) FlexibleSUSY → C++* code, which:

FlexibleSUSY.m	LesHouches.in.MODEL
restrictions*, extra*, settings	numerical values for the input parameters

solves BVP*
finds mixings
finds masses
finds observables*

A nice place for something new! 5) … →

```
FlexibleSUSYObservable`BrLTo3L[Fe@2 -> {Fe@1, Fe@1, bar@Fe@1}, Scalars, 1]
```

`NPointFunctions?`[soon] The end-user side.

- Setup observable$_i$

#1

settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

1) `Mathematica`* code,
2) `C++` code, which:
adds*, new input blocks
evaluates observable$_i$
evaluates Wilson coefficients

NPointFunctions?[soon] The end-user side.

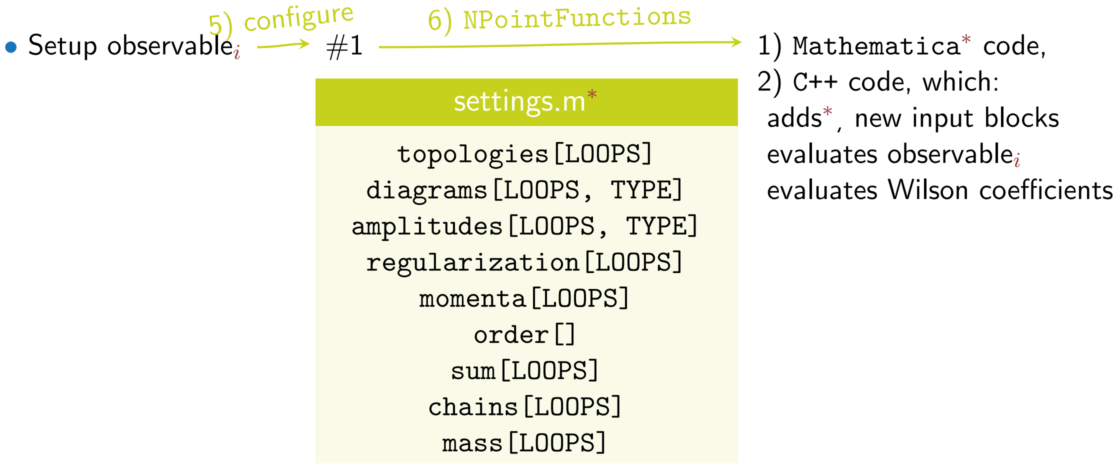
- Setup observable$_i$ $\xrightarrow{\text{5) configure}}$ #1

settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

1) Mathematica* code,
2) C++ code, which:
 adds*, new input blocks
 evaluates observable$_i$
 evaluates Wilson coefficients

NPointFunctions?[soon] The end-user side.



- Setup observable$_i$ $\xrightarrow{\text{5) configure}}$ #1 $\xrightarrow{\text{6) NPointFunctions}}$

settings.m*

topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]

1) Mathematica* code,
2) C++ code, which:
adds*, new input blocks
evaluates observable$_i$
evaluates Wilson coefficients

NPointFunctions?[soon] The end-user side.

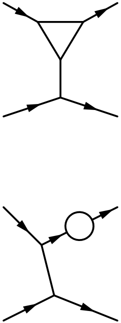
- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code,

2) C++ code, which:

adds*, new input blocks

evaluates observable$_i$

evaluates Wilson coefficients



settings.m*

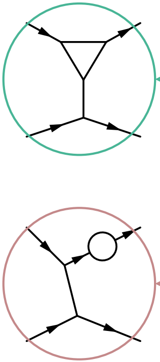
```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ $\xrightarrow{\text{5) configure}}$ #1 $\xrightarrow{\text{6) NPointFunctions}}$

1) Mathematica* code,
2) C++ code, which:
 adds*, new input blocks
 evaluates observable$_i$
 evaluates Wilson coefficients



settings.m*

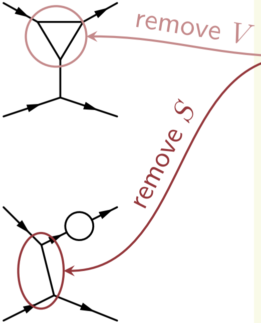
```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
topologies[1] = {
 Scalars -> triangleT,
 Vectors -> outSelfT,..
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code, 2) C++ code, which:
  - adds*, new input blocks
  - evaluates observable$_i$
  - evaluates Wilson coefficients



settings.m*

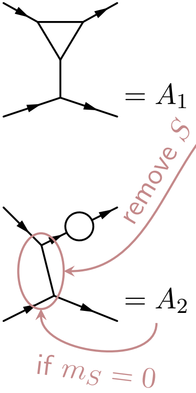
```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
diagrams[1, Plus] = {
 Scalars -> {
  triangleT -> {"No V",
   FreeQ[LoopFields@##,
    FeynArts`V]&},..
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code,
  2) C++ code, which:
  adds*, new input blocks
  evaluates observable$_i$
  evaluates Wilson coefficients



settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
diagrams[1, Minus] = {
 Vectors -> {
  outSelfT -> {"No S",
   FreeQ[#, InternalMass[
    FeynArts`S, 5] -> 0]&},..
```

NPointFunctions?[soon] The end-user side.

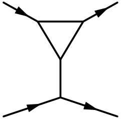
- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code,

2) C++ code, which:

adds*, new input blocks

evaluates observable$_i$

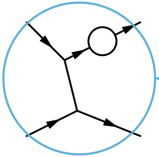
evaluates Wilson coefficients



settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

use $\overline{MS}$

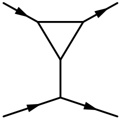


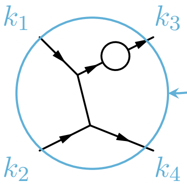
Example: PRELIMINARY

```
regularization[1] = {
 triangleT -> 4,
 outSelfT -> D,..
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code, 2) C++ code, which: adds*, new input blocks; evaluates observable$_i$; evaluates Wilson coefficients





replace k_2

settings.m*

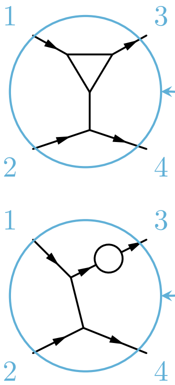
```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
momenta[1] = {
 triangleT -> 4,
 outSelfT -> 2,..
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code, 2) C++ code, which: adds*, new input blocks evaluates observable$_i$ evaluates Wilson coefficients



settings.m*

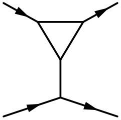
```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
order[] = {1, 2, 4, 3};
```

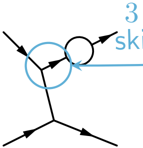
NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code, 2) C++ code, which:
  adds*, new input blocks
  evaluates observable$_i$
  evaluates Wilson coefficients



settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```



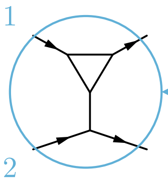
3
skip $F = 3$ in C++

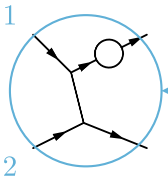
Example: PRELIMINARY

```
sum[1] = {
 outSelfT -> {"Unsame"
  {6, Field[#3, 3]&}},..
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) NPointFunctions → 1) Mathematica* code,
  2) C++ code, which:
  adds*, new input blocks
  evaluates observable$_i$
  evaluates Wilson coefficients





modify spinor

chains

settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
chains[1] = {
 ExceptLoops -> {
  1[k[4|2], ___] -> 0,..
```

NPointFunctions?[soon] The end-user side.

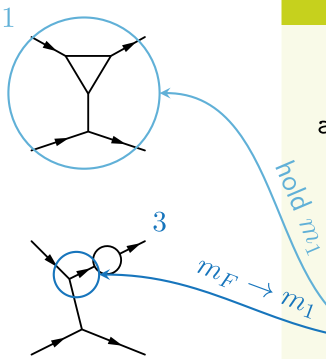
- Setup observable$_i$ $\xrightarrow{\text{5) configure}}$ #1 $\xrightarrow{\text{6) NPointFunctions}}$ 1) Mathematica* code,

2) C++ code, which:

adds*, new input blocks

evaluates observable$_i$

evaluates Wilson coefficients



settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

Example: PRELIMINARY

```
mass[1] = {
 triangleT -> {"Hold it"
  {Hold, ExternalMass[1]}},
 ..
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ $\xrightarrow{\text{5) configure}}$ #1 $\xrightarrow{\text{6) NPointFunctions}}$ 1) Mathematica* code,
  2) C++ code, which:
  adds*, new input blocks
  evaluates observable$_i$
  evaluates Wilson coefficients

settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

- Main* dependencies

```
FeynArts
FormCalc
ColorMath
```

NPointFunctions?[soon] The end-user side.

- Setup observable$_i$ 5) configure → #1 6) `NPointFunctions` → 1) `Mathematica`* code,

2) `C++` code, which:

adds*, new input blocks

evaluates observable$_i$

evaluates Wilson coefficients

settings.m*

```
topologies[LOOPS]
diagrams[LOOPS, TYPE]
amplitudes[LOOPS, TYPE]
regularization[LOOPS]
momenta[LOOPS]
order[]
sum[LOOPS]
chains[LOOPS]
mass[LOOPS]
```

- Main* dependencies

`FeynArts`

`FormCalc`

`ColorMath`

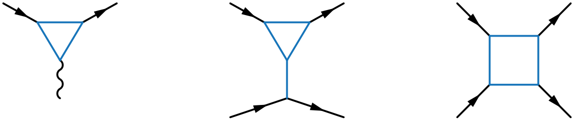
- Implemented*

$l_i \to l_j l_k l_k^C$

$l_i \to l_j$ conversion

connected to $l_i \to l_j \gamma$

LFV processes



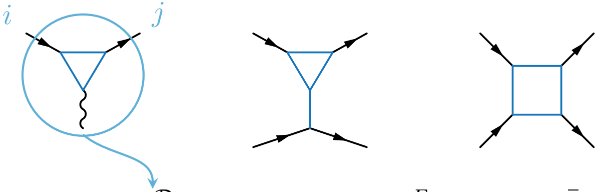
$$\mathcal{L}_{\mathsf{LEFT}} \ni m_\mu C^{\mathcal{D}}_X \left[\bar{e}\sigma^{\mu\nu} P_X \mu\right] F_{\mu\nu} + C^{\Gamma}_{XY,f} \left[\bar{e}\Gamma_X \mu\right] \left[\bar{f}\Gamma_Y f\right]$$

$$\Gamma^{\text{[any]}}_{\mu\to e\gamma} \;\propto\; \sum |C^{\mathcal{D}}|^2$$

$$\Gamma^{\text{[hep-ph/9510309]}}_{\mu\to 3e} \;\propto\; 0.006 \cdot \Gamma_{\mu\to e\gamma} + \sum \left(\mathrm{Re}\, C^{\mathcal{V}}_e C^{\mathcal{D}*} + |C^{\mathcal{S},\mathcal{V}}_e|^2 \right)$$

$$\omega^{\text{[hep-ph/0203110]}}_{\mu-e} \;\propto\; \sum |D C^{\mathcal{D}}_X - \sum (S^{(N)} g^{\mathcal{S}} + V^{(N)} g^{\mathcal{V}})|^2$$

LFV processes



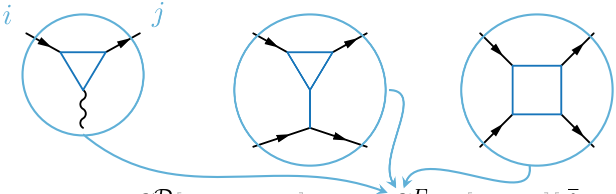
$$\mathcal{L}_{\mathsf{LEFT}} \ni m_\mu C^{\mathcal{D}}_X [\bar e \sigma^{\mu\nu} P_X \mu] F_{\mu\nu} + C^{\Gamma}_{XY,f} [\bar e \Gamma_X \mu][\bar f \Gamma_Y f]$$

$$\Gamma^{\text{[any]}}_{\mu\to e\gamma} \propto \sum |C^{\mathcal{D}}|^2$$

$$\Gamma^{\text{[hep-ph/9510309]}}_{\mu\to 3e} \propto 0.006\cdot\Gamma_{\mu\to e\gamma} + \sum \left(\mathrm{Re}\, C^{\mathcal{V}}_e C^{\mathcal{D}*} + |C^{\mathcal{S},\mathcal{V}}_e|^2\right)$$

$$\omega^{\text{[hep-ph/0203110]}}_{\mu-e} \propto \sum |D C^{\mathcal{D}}_X - \sum(S^{(N)} g^{\mathcal{S}} + V^{(N)} g^{\mathcal{V}})|^2$$

LFV processes



$$\mathcal{L}_{\textsf{LEFT}} \ni m_\mu C^{\mathcal{D}}_X [\bar{e}\sigma^{\mu\nu} P_X \mu] F_{\mu\nu} + C^{\Gamma}_{XY,f} [\bar{e}\Gamma_X \mu][\bar{f}\Gamma_Y f]$$

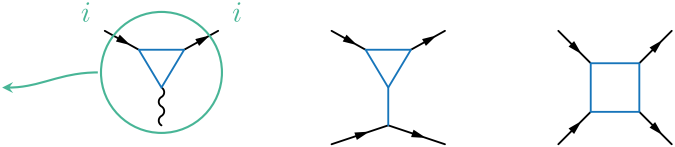
$$\Gamma^{\text{[any]}}_{\mu\to e\gamma} \propto \sum |C^{\mathcal{D}}|^2$$

$$\Gamma^{\text{[hep-ph/9510309]}}_{\mu\to 3e} \propto 0.006 \cdot \Gamma_{\mu\to e\gamma} + \sum \left(\operatorname{Re} C^{\mathcal{V}}_e C^{\mathcal{D}*} + |C^{\mathcal{S},\mathcal{V}}_e|^2 \right)$$

$$\omega^{\text{[hep-ph/0203110]}}_{\mu - e} \propto \sum |D C^{\mathcal{D}}_X - \sum (S^{(N)} g^{\mathcal{S}} + V^{(N)} g^{\mathcal{V}})|^2$$

LFV processes

What is about $(g-2)_i$?



$$\mathcal{L}_{\mathsf{LEFT}} \ni m_\mu C^{\mathcal{D}}_X [\bar e \sigma^{\mu\nu} P_X \mu] F_{\mu\nu} + C^{\Gamma}_{XY,f} [\bar e \Gamma_X \mu][\bar f \Gamma_Y f]$$

$$\Gamma^{\text{[any]}}_{\mu\to e\gamma} \propto \sum |C^{\mathcal{D}}|^2$$

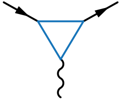
$$\Gamma^{\text{[hep-ph/9510309]}}_{\mu\to 3e} \propto 0.006\cdot\Gamma_{\mu\to e\gamma} + \sum\left(\mathrm{Re}\, C^{\mathcal{V}}_e C^{\mathcal{D}*} + |C^{\mathcal{S},\mathcal{V}}_e|^2\right)$$

$$\omega^{\text{[hep-ph/0203110]}}_{\mu-e} \propto \sum |DC^{\mathcal{D}}_X - \sum(S^{(N)} g^{\mathcal{S}} + V^{(N)} g^{\mathcal{V}})|^2$$

Motivation by ...

- New bounds / results

$(g-2)_\mu$
BNL + FNAL:
$(25.1 \pm 5.9) \cdot 10^{-10}$



$\mu \to e$ conversion COMETs: 3,4
$\mu \to 3e$ Mu3es: 2,4
$\mu \to e\gamma$ MEG-II: 1

They are connected! ... **?**

Q: Do we need / How to use all of that?

- SUSY — Extension of Poincaré algebra | No quadratic divergences | ...
- MRSSM — Different SUSY realization | Absence of MSSM limit | R-symmetry
- Rich phenomenology[2014-...] — Electroweak precision observables | Higgs boson mass | Dark matter relic density | Coloured sector
- Other models? — Leptoquarks S_1 and R_2 | Grimus-Neufeld model

Well motivated!

Q: What's the model contribution / parameter dependence?

MRSSM?[0712.2039]

$U(1)_R$ symmetry: $\theta \to e^{i\alpha Q_\theta}\theta, \quad Q_\theta := +1$

Same superfield → related Q_*

Assertion	$Q_{\mathsf{V}} = 0$	$Q(v_{d,u}) = 0$	Yukawas form	All previous
Result	**no** Majorana gauginos	**no** μ-term	Q_{SM} are* fixed	**no** L/R mixing **no** A-terms
Consequence	Dirac masses			sfermion masses*
$-\mathcal{L} \ni$	$M_B^D(\tilde{B}\tilde{S} - \sqrt{2}D_B S)$			$(m^2_{\tilde{l}})_{ij}\tilde{l}^*_i\tilde{l}_j$
		higgsino masses	usual Yukawas	new "Yukawas"
$W \ni$		$\mu_u R_u \cdot H_u$	$-Y^e_{ij}\ \bar{E}_i L_j \cdot H_d$	$\lambda_u S R_u \cdot H_u$

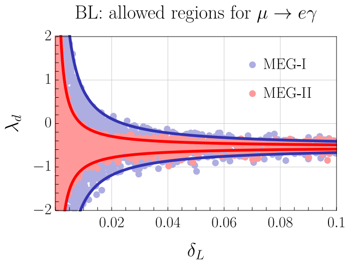
Parameters

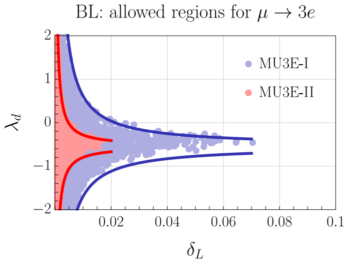
Dirac masses	higgsino masses	sfermion masses*	new “Yukawas”
$M_B^D(\tilde{B}\tilde{S} - \sqrt{2}D_B S)$	$\mu_u R_u \cdot H_u$	$(m^2_{\tilde{l}})_{ij}\tilde{l}^*_i\tilde{l}_j$	$\lambda_u S R_u \cdot H_u$
M_B^D or M_W^D should be light!	μ_d – dipole μ_u – restricted	$\delta_L = \frac{(m^2_{\tilde{l}})_{12}}{(m_{\tilde{l}})_{11}(m_{\tilde{l}})_{22}}$ and / or $\delta_R = \frac{(m^2_{\tilde{e}})_{12}}{(m_{\tilde{e}})_{11}(m_{\tilde{e}})_{22}}$	λ_d, λ_u, Λ_d, Λ_u – dependent

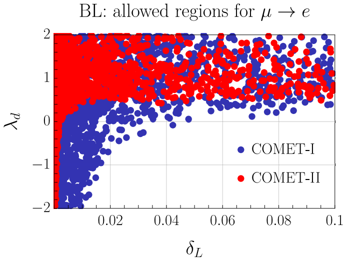
So many! **Q:** What to do?

Simplified scenarios: i.e. BHL

Scattering plots


BL: allowed regions for $\mu \to e\gamma$


BL: allowed regions for $\mu \to 3e$


BL: allowed regions for $\mu \to e$

expectation:

$$\mathrm{Br}_{\mu\to e\gamma} \propto \delta_L^2(\lambda_d + \Delta)^2$$

check!

expectation:

dipole dominance if

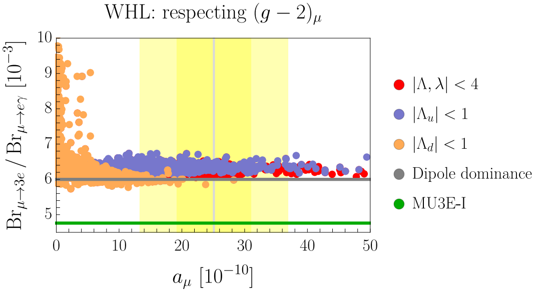
$$\mathrm{Br}_{\mathsf{MEG}} \to \mathrm{Br}_{\mathsf{MU3E}}/0.006$$

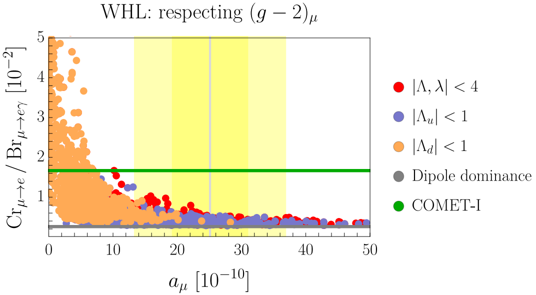
correlation?

cancellations,

non-correlation!

What if ... $(g-2)_\mu$?





$$\frac{\mathrm{Br}_{\mu\to 3e}}{\mathrm{Br}_{\mu\to e\gamma}} \approx 0.006 \qquad \frac{\mathrm{Cr}_{\mu\to e}}{\mathrm{Br}_{\mu\to e\gamma}} \approx 0.0026$$

Chirality flip *aka* $\sigma_{\mu\nu}$, **no** μ-term $\to$ Λ_d, λ_d enhancement.

Conclusions

- **N**PointFunctions @ FlexibleSUSY

  Fast (thanks to `C++` and `FORTRAN`)
  Customizable (on `Mathematica` and `C++` levels)
  Extendable (due to a modular structure)
  Consistent checks / constraints from different scales

- **A**pplications

  MRSSM[**U.Kh**, W.Kotlarski, D.Stöckinger, H.Stöckinger-Kim]
  Leptoquarks S_1 and R_2[**U.Kh**, D.Stöckinger, H.Stöckinger-Kim, J.Wünsche]
  Grimus-Neufeld model[V.Dūdėnas, T.Gajdosik, **U.Kh**, W.Kotlarski, D.Stöckinger]
  ...

- **T**ODOs:

  New observables
  Some guide
  Bug fixes / structure simplification
  More options / loops / ...

Conclusions

- **N**PointFunctions @ FlexibleSUSY

  Fast (thanks to `C++` and `FORTRAN`)
  Customizable (on `Mathematica` and `C++` levels)
  Extendable (due to a modular structure)
  Consistent checks / constraints from different scales

- **A**pplications

  MRSSM[**U.Kh**, W.Kotlarski, D.Stöckinger, H.Stöckinger-Kim]
  Leptoquarks S_1 and R_2[**U.Kh**, D.Stöckinger, H.Stöckinger-Kim, J.Wünsche]
  Grimus-Neufeld model[V.Dūdėnas, T.Gajdosik, **U.Kh**, W.Kotlarski, D.Stöckinger]
  ... WRITE ME :D

- **T**ODOs:

  New observables
  Some guide
  Bug fixes / structure simplification
  More options / loops / ...