Spectrum prediction in non-minimal supersymmetric models

Alexander Voigt

Technische Universität Dresden
Institut für Kern- und Teilchenphysik

HU Berlin
23 October 2013

Content

1. Motivation
   - Non-minimal supersymmetry
   - Physics situation
2. Calculation of β_v
   - Renormalization of v
   - Background field method
   - Calculation of $\delta\hat{Z}$
   - Result
3. FlexibleSUSY
   - Motivation
   - Design goals
   - Usage
   - Precision test
4. Conclusions

Content

1. Motivation
   - Non-minimal supersymmetry
   - Physics situation
2. Calculation of β_v
   - Renormalization of v
   - Background field method
   - Calculation of $\delta\hat{Z}$
   - Result
3. FlexibleSUSY
   - Motivation
   - Design goals
   - Usage
   - Precision test
4. Conclusions

Motivation – minimal supersymmetry (MSSM)

MSSM gauge group:

$$SU(3)_c \otimes SU(2)_L \otimes U(1)_Y$$

MSSM particle content:

(S)Quarks:	$Q_i,\ \bar{U}_i,\ \bar{D}_i$
(S)Leptons:	$L_i,\ \bar{E}_i$
Higgs(inos):	$H_1,\ H_2$
Gauge(inos):	$V_g^a,\ \vec{V}^W,\ V^Y$

MSSM superpotential:

$$\mathcal{W}_{\text{MSSM}} = \mu H_1 H_2 - y_u (Q H_2)\bar{U} - y_d (H_1 Q)\bar{D} - y_e (H_1 L)\bar{E}$$

Motivation – μ problem of the MSSM

MSSM superpotential:

$$\mathcal{W}_{\text{MSSM}} = \mu H_1 H_2 - y_u (Q H_2) \bar{U} - y_d (H_1 Q) \bar{D} - y_e (H_1 L) \bar{E}$$

Problem:

- $\mathcal{W}_{\text{MSSM}}$ generated at unification scale $M_X \Rightarrow \mu \sim M_X$
- but EWSB conditions imply

$$\frac{1}{2} M_Z^2 = \frac{m_{H_1}^2 - m_{H_2}^2 \tan^2\beta}{\tan^2\beta - 1} - \mu^2 \quad \Rightarrow \quad \mu \sim M_Z$$

Solution: introduce gauge singlet $S \to S + v_s$

$$\mathcal{W}_{\text{NMSSM}} = \mathcal{W}_{\text{MSSM}}(\mu = 0) + \lambda S H_1 H_2 + \kappa S^3$$

Motivation – m_h fine tuning in the MSSM

MSSM: $m_h^2 = (M_Z c_{2\beta})^2 + \Delta m_h^2$

NMSSM: $m_h^2 = (M_Z c_{2\beta})^2 + (\lambda v_s s_{2\beta})^2 + \Delta m_h^2$

E_6SSM: $m_h^2 = (M_Z c_{2\beta})^2 + (\lambda v_s s_{2\beta})^2 + \frac{M_Z^2}{4}\left(1 + \frac{c_{2\beta}}{4}\right)^2 + \Delta m_h^2$

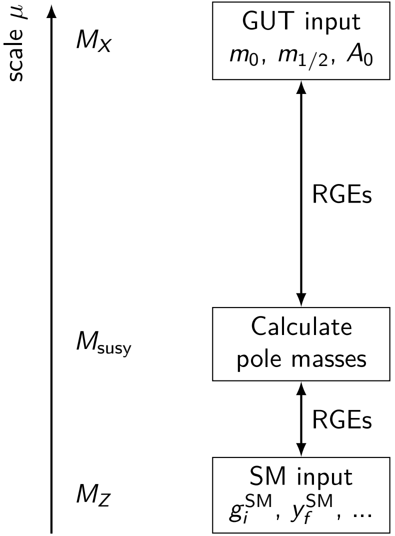
$\Rightarrow$ Non-minmal supersymmetric models can increase the tree-level Higgs mass

$\Rightarrow$ avoid m_h fine tuning

Motivation – non-minimal supersymmetric models

	CMSSM	non-minimal SUSY	CE_6SSM
solves μ problem of the MSSM	✗	✓	✓[hep-ph/0510419]
avoid m_h fine tuning	✗	✓	✓[hep-ph/1302.5291]
connection to specific gravity models	✗	✓	✓[hep-ph/0510419]
complete repres. of simple GUT gauge group	✗	✓	✓[hep-ph/0510419]
new particles and phenomena	✓	✓	✓[hep-ph/1302.5291]
explanation of $(g-2)_\mu^{\text{exp.}}$	✗	✓	✗

Physics situation



$$m_{\tilde{q}}(M_X) = m_{\tilde{l}}(M_X) = m_0$$
$$m_1(M_X) = m_2(M_X) = m_{1/2}$$
$$\ldots$$

$$dg_i/d\mu = \beta_{g_i}$$
$$dy_f/d\mu = \beta_{y_f}$$
$$dv/d\mu = \beta_v$$

$$g_i^{\text{susy}}(M_Z) = g_i^{\text{SM}}(M_Z) + \Delta g_i$$
$$y_f^{\text{susy}}(M_Z) = y_f^{\text{SM}}(M_Z) + \Delta y_f$$
$$\ldots$$

Content

1. Motivation
   - Non-minimal supersymmetry
   - Physics situation
2. Calculation of β_v
   - Renormalization of v
   - Background field method
   - Calculation of $\delta\hat{Z}$
   - Result
3. FlexibleSUSY
   - Motivation
   - Design goals
   - Usage
   - Precision test
4. Conclusions

Current status of β_v calculations

Current status of $\beta_v^{\overline{\text{DR}}/\overline{\text{MS}}}$:

Model	$\beta_v^{(1)}$	$\beta_v^{(2)}$
MSSM	✓ [Chankowski Nucl.Phys. B423]	✓ [Yamada 94] $O(g^2 Y^2)$
E_6SSM	✓ [Athron et al. 12]	✗
$\forall$ gauge theory	?	✗
$\forall$ SUSY model	?	✗

We calculate:

$\delta v^{(1,2)}$ and $\beta_v^{(1,2)}$ in a general and supersymmetric gauge theory with R_ξ gauge fixing [Sperling et al. 13]

Renormalization of v

Spontaneously broken gauge theory:

$$\phi \to \phi + v$$

Most generic renormalization transformation:

$$(\phi + v) \to \sqrt{Z}\phi + v + \delta v$$

$$\text{or} \quad (\phi + v) \to \sqrt{Z}(\phi + v + \delta\bar{v})$$

$\delta\bar{v}$ characterizes to what extent v renormalizes differently from ϕ.

With $\sqrt{Z} = 1 + \frac{1}{2}\delta Z \Rightarrow$

$$\delta v = \frac{1}{2}\delta Z v + \delta\bar{v}$$

Influence of global gauge invariance

When does $\delta\bar{v}$ appear?

$$\begin{aligned} \text{global gauge invariance} \quad &\Rightarrow \quad \delta\bar{v} = 0 \\ \text{no global gauge invariance} \quad &\Rightarrow \quad \delta\bar{v} \neq 0 \end{aligned}$$

R_ξ gauge fixing:

$$\mathcal{L}_{\text{fix,gh}} = s\left[\bar{c}^A\left(F^A + \xi B^A/2\right)\right]$$
$$F^A = \partial^\mu V_\mu^A + ig\xi v_a T_{ab}^A \phi_b$$

R_ξ breaks global gauge invariance for $\xi \neq 0 \Rightarrow \delta\bar{v} \neq 0$.

Background field method – Introducing $\hat{\phi}$

Problem: R_ξ breaks global gauge invariance for $\xi \neq 0 \Rightarrow \delta\bar{v} \neq 0$.

Trick: Keep global gauge invariance in intermediate calculation!

[Kraus,Sibold 95]

Introduce background field $\hat{\phi}$ and shift $\hat{v}$

$$\phi \to \phi_{\text{eff}} := \phi + \hat{\phi} + \hat{v}$$

where $\hat{\phi} + \hat{v}$ has same gauge transformation as ϕ.
Modified R_ξ gauge fixing:

$$F^A = \partial^\mu V_\mu^A + ig\xi(\hat{\phi} + \hat{v})_a T^A_{ab}\phi_b$$

$\Rightarrow$ global gauge invariance! $\Rightarrow \delta\bar{v} = 0$

Background field method – Lagrangian

Modified BRS transformations:

$$s\phi_{\text{eff}} = -igT^A c^A \phi_{\text{eff}}, \quad s\hat{\phi} = \hat{q}, \quad s\hat{q} = 0, \quad s\phi = s\phi_{\text{eff}} - \hat{q}$$

Modified Lagrangian:

$$\mathcal{L} = \mathcal{L}_{\text{inv}}|_{\phi\to\phi_{\text{eff}}} + \mathcal{L}_{\text{fix,gh}} + \mathcal{L}_{\text{ext}}$$

$$\begin{aligned}
\mathcal{L}_{\text{inv}} &= -\frac{1}{4}F^A_{\mu\nu}F^{A\mu\nu} + \frac{1}{2}\left(D_\mu\phi\right)_a\left(D^\mu\phi\right)_a + i\psi^\alpha_p\sigma^\mu_{\alpha\dot{\alpha}}\left(D^\dagger_\mu\bar{\psi}^{\dot{\alpha}}\right)_p \\
&\quad - \frac{1}{2!}m^2_{ab}\phi_a\phi_b - \frac{1}{3!}h_{abc}\phi_a\phi_b\phi_c - \frac{1}{4!}\lambda_{abcd}\phi_a\phi_b\phi_c\phi_d \\
&\quad - \frac{1}{2}\left[\left(m_f\right)_{pq}\psi^\alpha_p\psi_{q\alpha} + \text{h.c.}\right] - \frac{1}{2}\left[Y^a_{pq}\psi^\alpha_p\psi_{q\alpha}\phi_a + \text{h.c.}\right] \\
\mathcal{L}_{\text{fix,gh}} &= s\left[\bar{c}^A\left(F^A + \xi B^A/2\right)\right] \\
\mathcal{L}_{\text{ext}} &= K_{\phi_a}s\phi_a + K_{V^A_\mu}sV^A_\mu + K_{c^A}sc^A + \left[K_{\psi_p}s\psi_p + \text{h.c.}\right]
\end{aligned}$$

Background field method – Renormalization of v

Modified renormalization transformation:

$$\phi \to \sqrt{Z}\phi$$

$$(\hat{\phi} + \hat{v}) \to \sqrt{Z}\sqrt{\hat{Z}}(\hat{\phi} + \hat{v})$$

$$\hat{q} \to \sqrt{Z}\sqrt{\hat{Z}}\hat{q}$$

$$\Rightarrow \phi_{\text{eff}} \to \sqrt{Z}\left(\phi + \sqrt{\hat{Z}}\left(\hat{\phi} + \hat{v}\right)\right)$$

Standard approach:

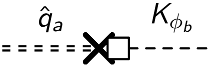
$$(\phi + v) \to \sqrt{Z}\phi + v + \delta v$$

For $\hat{\phi} = 0 \Rightarrow$

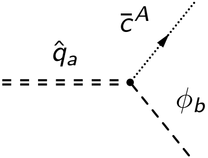
$$\boxed{\delta v = \left(\sqrt{Z}\sqrt{\hat{Z}} - 1\right) v = \frac{1}{2}\left(\delta Z + \delta\hat{Z}\right) v + O(\hbar^2)}$$

$$\begin{aligned}
\mathcal{L}_{\text{ext}} &= K_\phi \; s\phi + \cdots \\
&= -K_\phi \; igT^A c^A \phi - K_\phi \; \hat{q} + \cdots \\
\mathcal{L}_{\text{fix,gh}} &= s\left[\bar{c}^A\left(F^A + \xi B^A/2\right)\right]
\end{aligned}$$

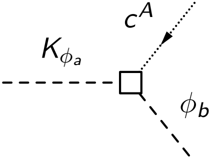
$\Rightarrow$



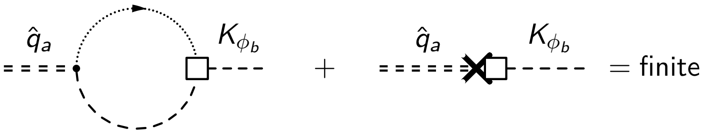
$$= -\frac{i}{2}\delta\hat{Z}^{(1)}_{ba}$$



$$= \xi g T^A_{ab},$$



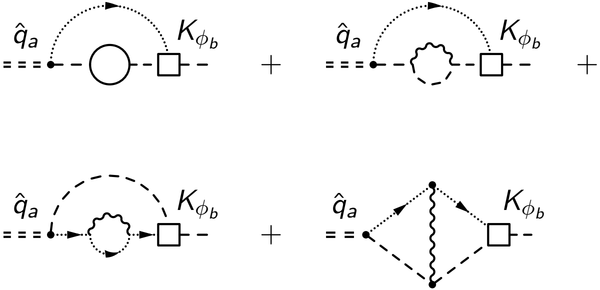
$$= g T^A_{ab}$$



$\Rightarrow$

$$(4\pi)^2 \delta\hat{Z}^{(1)} = 2g^2 \xi C^2(S) \frac{1}{\epsilon}$$

Calculation of $\delta\hat{Z}$ – 2 Loop



$\Rightarrow$

$$(4\pi)^4 \delta\hat{Z}^{(2)} = g^2 \xi C^2(\mathrm{S}) Y^2(\mathrm{S}) \left(\frac{1}{\epsilon^2} - \frac{1}{\epsilon} \right) + O(g^4)$$

Result – β_v

$$\delta v = \left(\sqrt{Z}\sqrt{\hat{Z}} - 1\right) v$$

$$\gamma(S) = \left(\mu\partial_\mu \sqrt{Z}^{-1}\right)\sqrt{Z}\,, \qquad \hat{\gamma}(S) = \left(\mu\partial_\mu \sqrt{\hat{Z}}^{-1}\right)\sqrt{\hat{Z}}$$

$\Rightarrow$

$$\beta_v^{(n)} = \left[\gamma^{(n)}(S) + \hat{\gamma}^{(n)}(S)\right] v$$

$$\hat{\gamma}^{(1)}(\mathrm{S}) = \frac{\xi}{(4\pi)^2} 2g^2 C^2(\mathrm{S})$$

$$\hat{\gamma}^{(2)}(\mathrm{S}) = \frac{\xi}{(4\pi)^4}\Bigg\{ g^4 \left[2\left(1+\xi\right) C^2(\mathrm{S})C^2(\mathrm{S}) + \frac{7-\xi}{2} C_2(\mathrm{G})C^2(\mathrm{S})\right] - 2g^2 C^2(\mathrm{S})Y^2(\mathrm{S})\Bigg\}$$

Calculation of β_v – Summary

- In R_ξ gauge fixings (with $\xi \neq 0$), v renormalizes differently from ϕ

$$\phi \to \sqrt{Z}(\phi + v + \delta\bar{v})$$

- difference $\delta\bar{v}$ can be interpreted as field renormalization $\delta\hat{Z}$ of a background field $\hat{\phi}$

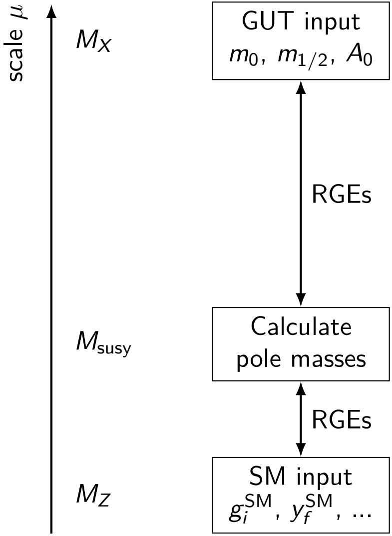
$$\delta\bar{v} = \frac{1}{2}\delta\hat{Z} v$$

- We've calculated $\delta\hat{Z}^{(1,2)}$, $\delta v^{(1,2)}$, $\beta_v^{(1,2)}$, in the $\overline{\text{DR}}$ scheme for arbitrary ξ in a general gauge theory [Sperling et al. 13]

Content

1. Motivation
   - Non-minimal supersymmetry
   - Physics situation
2. Calculation of β_v
   - Renormalization of v
   - Background field method
   - Calculation of $\delta\hat{Z}$
   - Result
3. FlexibleSUSY
   - Motivation
   - Design goals
   - Usage
   - Precision test
4. Conclusions

Physics situation



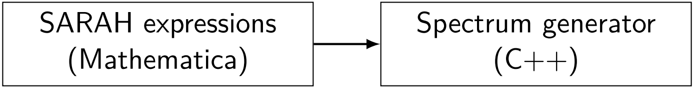
$$m_{\tilde{q}}(M_X) = m_{\tilde{l}}(M_X) = m_0$$
$$m_1(M_X) = m_2(M_X) = m_{1/2}$$
$$\ldots$$

$$dg_i/d\mu = \beta_{g_i}$$
$$dy_f/d\mu = \beta_{y_f}$$
$$dv/d\mu = \beta_v$$

$$g_i^{\text{susy}}(M_Z) = g_i^{\text{SM}}(M_Z) + \Delta g_i$$
$$y_f^{\text{susy}}(M_Z) = y_f^{\text{SM}}(M_Z) + \Delta y_f$$
$$\ldots$$

Motivation – Why creating a new spectrum generator?

FlexibleSUSY provides Mathematica meta code which creates a spectrum generator



Already existing: SARAH $\to$ SPheno (Fortran)

Motivation for FlexibleSUSY:

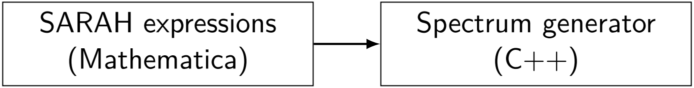
- large variety of supersymmetric models

$\to$ user customization desired

- convergence problems in certain parameter regions

$\to$ provide alternative RG solvers

Design goals



Design goals:

- modular, object oriented C++ code $\Rightarrow$ easy to hack! ✓
- fast (smart linear algebra, multithreading) ✓
- multiple RGE solvers:
  - two-scale running (adaptive Runge-Kutta) ✓
  - lattice method + variants (Jae-hyeon Park) ✓
- SARAH-like user interface ✓
- tower of effective field theories ✗

Usage – 0. Get the source code

Get the source code:

```
$ git clone https://github.com/Expander/FlexibleSUSY
$ cd FlexibleSUSY
```

Usage – 1. Create FlexibleSUSY model file I

```
FSModelName = "NMSSM";

MINPAR = { {1, m0},
           {2, m12},
           {3, TanBeta},
           {5, Azero} };

EXTPAR = { {61, LambdaInput} };

EWSBOutputParameters = { Kappa, vS, ms2 };

SUSYScale = Sqrt[M[Su[1]]*M[Su[6]]];

HighScale = g1 == g2;

HighScaleInput = {
   {mHd2, m0^2}, {mHu2, m0^2}, {mq2, UNITMATRIX[3] m0^2},
   ...
};

LowScale = SM[MZ];

LowScaleInput = { ... };
```

Usage – 1. Create FlexibleSUSY model file II

Minimizer (iterative):

```
LowScaleInput = {
   FSMinimize[{vd,vu},
              (SM[MZ] - Pole[M[VZ]])^2 /
                 STANDARDDEVIATION[MZ]^2
            + (SM[MH] - Pole[M[hh[1]]])^2 /
                 STANDARDDEVIATION[MH]^2 ]
};
```

Root finder (iterative):

```
LowScaleInput = {
   FSFindRoot[{vd,vu}, {SM[MZ] - Pole[M[VZ]],
                        SM[MH] - Pole[M[hh[1]]]} ]
};
```

Usage – 2. Configure and make

Create a NMSSM spectrum generator:

```
$ ./createmodel --models=NMSSM
$ ./configure --with-models=NMSSM
$ make
```

Run it:

```
$ ./models/NMSSM/run_NMSSM.x
```

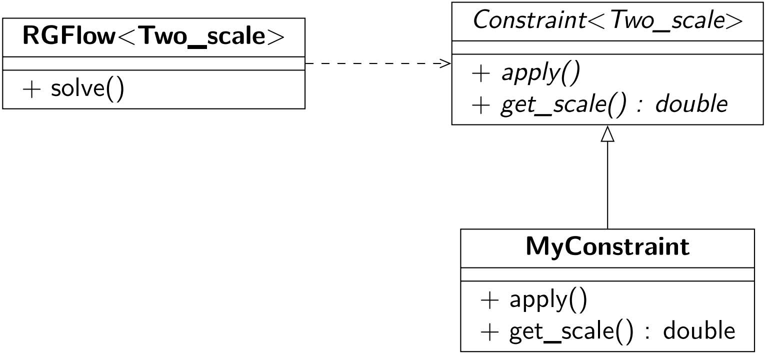
Run with an SLHA input file:

```
$ ./models/NMSSM/run_NMSSM.x \
   --slha-input-file=input.slha \
   --slha-output-file=output.slha
```

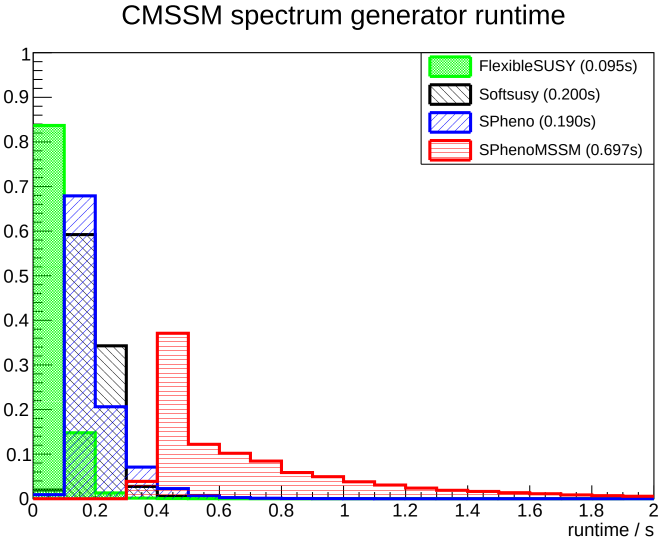
Modularity – generated C++ code

```
typedef Two_scale T; // or Lattice
NMSSM<T> nmssm;
NMSSM_input_parameters input;

std::vector<Constraint<T>*> constraints = {
  new NMSSM_low_scale_constraint<T>(input),
  new NMSSM_susy_scale_constraint<T>(input),
  new NMSSM_high_scale_constraint<T>(input)
};

// solve RG eqs. with the above constraints
RGFlow<T> solver;
solver.add_model(&nmssm, constraints);
solver.solve();

nmssm.calculate_spectrum();
```

Modularity – Constraint interface

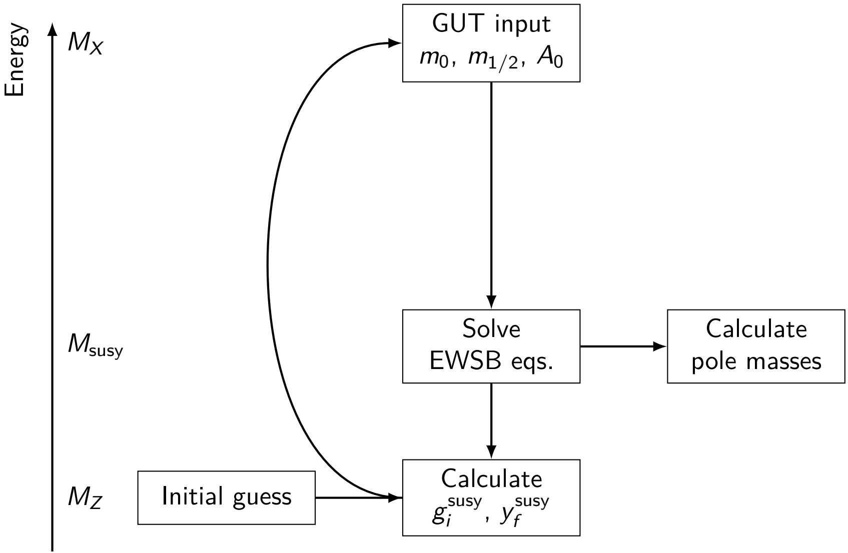


Speed – runtime comparison

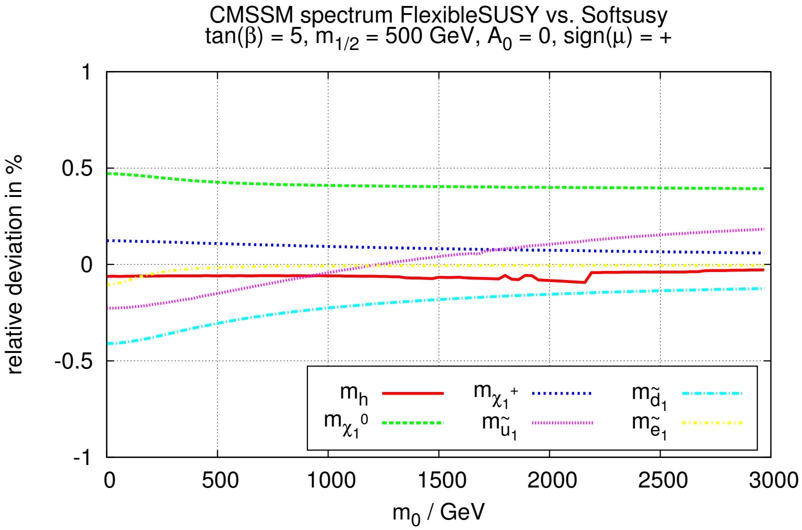


g++ 4.8.0, gfortran 4.8.0

RGE solvers – two-scale algorithm



Spectrum comparison FlexibleSUSY vs. Softsusy



Conclusions

FlexibleSUSY

- is **Modular** (C++ classes, easy to modify and extend)
- is **Fast** (CMSSM runtime $\approx 0.1s$)
- Provides **different RGE solvers**
  - two-scale running (adaptive Runge-Kutta)
  - lattice method + variants

Currently supported models:

- MSSM, NMSSM, SMSSM, UMSSM, E6SSM, MRSSM

Future plans:

- determination of $\sin\theta_W$ from G_μ
- two-loop leading log Higgs mass corrections
- tower of effective field theories