CoEPP
ARC Centre of Excellence for Particle Physics at the Terascale

Next-to-Minimal SOFTSUSY

Ben Allanach, Peter Athron, Lewis Tunstall,
Alexander Voigt, Anthony Williams

coepp.org.au

School of Chemistry & Physics
The University of Adelaide
South Australia 5005
Australia

School of Physics
The University of Sydney
New South Wales 2006
Australia

School of Physics
The University of Melbourne
Victoria 3010
Australia

School of Physics
Monash University
Victoria 3800
Australia

Contents

- Motivation
- NMSSM model
- NMSSM Spectrum calculation
- Code structure
- Download and install
- Running the code
- Conclusions

Spectrum generators

Solve boundary value problem finding:

$\overline{DR}$ Parameters (masses and couplings) at low energies

Spectrum generators

Solve boundary value problem finding:

$\overline{DR}$ Parameters (masses and couplings) at low energies

Pole masses and mixings

Spectrum generators

Solve boundary value problem finding:

$\overline{DR}$ Parameters (masses and couplings) at low energies

Pole masses and mixings

consistent with:

Input boundary conditions on parameters

Low energy data ($m_Z, G_F, m_{top} \ldots$)

Spectrum generators

Solve boundary value problem finding:

$\overline{DR}$ Parameters (masses and couplings) at low energies

Pole masses and mixings

consistent with:

Input boundary conditions on parameters

Low energy data ($m_Z, G_F, m_{top} \ldots$)

➢ Spectrum generators play a crucial role in SUSY phenomenology

- Allow fast exploration of the parameter space with high precision

Spectrum generators

Solve boundary value problem finding:

$\overline{DR}$ Parameters (masses and couplings) at low energies

Pole masses and mixings

consistent with:

Input boundary conditions on parameters

Low energy data ($m_Z, G_F, m_{top}\ldots$)

- ➢ Spectrum generators play a crucial role in SUSY phenomenology

- Allow fast exploration of the parameter space with high precision
- Provides required input for many other HEP tools

  e.g. Dark matter calculators, low energy observables, decay tools

Spectrum generators

Solve boundary value problem finding:

$\overline{DR}$ Parameters (masses and couplings) at low energies

Pole masses and mixings

consistent with:

Input boundary conditions on parameters

Low energy data ($m_Z, G_F, m_{top} \ldots$)

➢ Spectrum generators play a crucial role in SUSY phenomenology

- Allow fast exploration of the parameter space with high precision
- Provides required input for many other HEP tools

  e.g. Dark matter calculators, low energy observables, decay tools

- Typically appears at the top of a long calculation tool chain for doing phenomenology

NMSSM Spectrum generation

SOFTSUSY: popular, fast, precise, reliable tool for MSSM spectra

Next-to-Minimal SOFTUSY update provides all this in NMSSM too

NMSSM Spectrum generation

SOFTSUSY: popular, fast, precise, reliable tool for MSSM spectra

Next-to-Minimal SOFTUSY update provides all this in NMSSM too

- Very reliable, heavily tested, extends very widely used code
- Very readable, easy to understand and modify
- Familiar code structure to SOFTSUSY users

NMSSM Spectrum generation

SOFTSUSY: popular, fast, precise, reliable tool for MSSM spectra

Next-to-Minimal SOFTUSY update provides all this in NMSSM too

- Very reliable, heavily tested, extends very widely used code
- Very readable, easy to understand and modify
- Familiar code structure to SOFTSUSY users

NMSSM previously had one public spectrum generator NMSPEC

(Can also auto-generate codes with FlexibleSUSY, SARAH)

(see talk by A. Voigt next)

Many MSSM spectrum generators

SOFTSUSY, ISASUSY, SPheno, SUSEFLAV, SUSPECT

bug catching, cross-checks, different capabilities, approaches

NMSSM Spectrum generation

SOFTSUSY: popular, fast, precise, reliable tool for MSSM spectra

Next-to-Minimal SOFTUSY update provides all this in NMSSM too

- Very reliable, heavily tested, extends very widely used code
- Very readable, easy to understand and modify
- Familiar code structure to SOFTSUSY users

NMSSM previously had one public spectrum generator NMSPEC

(Can also auto-generate codes with FlexibleSUSY, SARAH)

(see talk by A. Voigt next)

Many MSSM spectrum generators

SOFTSUSY, ISASUSY, SPheno, SUSEFLAV, SUSPECT

bug catching, cross-checks, different capabilities, approaches

With Next-to-Minimal SOFTUSY now get this benefit in NMSSM

NMSSM Spectrum generation

- The μ–problem

$$\text{Why } \mu \approx 0.1 - 1 \text{ TeV?}$$

$$\mathcal{W}_{MSSM} = Y_u \hat{\bar{Q}}_L \hat{H}_u \hat{u}_R - Y_d \hat{\bar{Q}}_L \hat{H}_d \hat{d}_R - Y_e \hat{\bar{E}} \hat{H}_d \hat{d}_R - \mu \hat{H}_u \hat{H}_d$$

NMSSM Spectrum generation

- Solve μ-problem

$$\text{generate } \mu \approx 0.1 - 1 \text{ TeV}$$

$$\mathcal{W}_{PQNMSSM} = Y_u \bar{Q}_L H_u u_R - Y_d \bar{Q}_L H_d d_R - Y_e \bar{E} H_d d_R - \lambda S H_u H_d$$

NMSSM Spectrum generation

- Solve μ-problem generate $\mu \approx 0.1 - 1\ \mathrm{TeV}$

$$\mathcal{W}_{PQNMSSM} = Y_u \bar{Q}_L H_u u_R - Y_d \bar{Q}_L H_d d_R - Y_e \bar{E} H_d d_R - \lambda S H_u H_d$$

- Increase the Higgs mass (low $\tan\beta$, high λ)

$$m_h^2 \approx \underbrace{M_Z^2 \cos^2 2\beta}_{\mathrm{MSSM}} + \frac{\lambda^2}{2} v^2 \sin^2 2\beta$$

NMSSM Spectrum generation

- Solve μ-problem $\quad$ generate $\mu \approx 0.1 - 1$ TeV

$$\mathcal{W}_{PQNMSSM} = Y_u \bar{Q}_L H_u u_R - Y_d \bar{Q}_L H_d d_R - Y_e \bar{E} H_d d_R - \lambda S H_u H_d$$

- Increase the Higgs mass (low $\tan\beta$, high λ)

$$m_h^2 \approx \underbrace{M_Z^2 \cos^2 2\beta}_{\text{MSSM}} + \frac{\lambda^2}{2} v^2 \sin^2 2\beta$$

- Improved naturalness

e.g. [M.Bastero-Gil, C.Hugonie, S.F.King, D.P.Roy , S.Vempati, PLB 489, 359; S. F. King, M. Mhlleitner, R. Nevzorov and K. Walz,Nucl. Phys. B 870 323,;T. Gherghetta, B. von Harling, A. D. Medina,M. A. Schmidt, JHEP 02, 032; D.Kim, PA, C.Balázs, B.Farmer, E.Hutchison arXiv:1312.4150]

NMSSM

$$SU(3)_C \times SU(2)_W \times U(1)_Y$$

$$\mathcal{W}_{PQNMSSM} = Y_u \bar{Q}_L H_u u_R - Y_d \bar{Q}_L H_d d_R - Y_e \bar{E} H_d d_R - \lambda S H_u H_d$$

Chiral superfields

$$\hat{\Psi}_i \to e^{iQ_i^{PQ}\theta} \hat{\Psi}_i$$

Superfield	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	$U(1)_{PQ}$
$\hat{Q}_i$	$\mathbf{3}$	$\mathbf{2}$	$\frac{1}{6}$	-1
$\overline{u}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$-\frac{2}{3}$	0
$\overline{d}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$\frac{1}{3}$	0
$\hat{L}_i$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	-1
$\overline{e}_i$	$\mathbf{1}$	$\mathbf{1}$	1	0
$\hat{H}_u$	$\mathbf{1}$	$\mathbf{2}$	$+\frac{1}{2}$	1
$\hat{H}_d$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	1
$\hat{S}$	$\mathbf{1}$	$\mathbf{1}$	0	-2

NMSSM

Break PQ symmetry explictly with cubic term: $\frac{\kappa}{3}S^3$

$\Rightarrow$ usual $\mathcal{Z}_3$ symmetric NMSSM

Superfield	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	$U(1)_{PQ}$
$\hat{Q}_i$	$\mathbf{3}$	$\mathbf{2}$	$\frac{1}{6}$	-1
$\overline{u}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$-\frac{2}{3}$	0
$\overline{d}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$\frac{1}{3}$	0
$\hat{L}_i$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	-1
$\overline{e}_i$	$\mathbf{1}$	$\mathbf{1}$	1	0
$\hat{H}_u$	$\mathbf{1}$	$\mathbf{2}$	$+\frac{1}{2}$	1
$\hat{H}_d$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	1
$\hat{S}$	$\mathbf{1}$	$\mathbf{1}$	0	-2

NMSSM

Break PQ symmetry explictly with cubic term: $\frac{\kappa}{3} S^3$

$\Rightarrow$ usual $\mathcal{Z}_3$ symmetric NMSSM

Superfield	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	$U(1)_{PQ}$
$\hat{Q}_i$	$\mathbf{3}$	$\mathbf{2}$	$\frac{1}{6}$	-1
$\overline{u}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$-\frac{2}{3}$	0
$\overline{d}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$\frac{1}{3}$	0
$\hat{L}_i$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	-1
$\overline{e}_i$	$\mathbf{1}$	$\mathbf{1}$	1	0
$\hat{H}_u$	$\mathbf{1}$	$\mathbf{2}$	$+\frac{1}{2}$	1
$\hat{H}_d$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	1
$\hat{S}$	$\mathbf{1}$	$\mathbf{1}$	0	-2

$\mathcal{Z}_3$ symmetric

$$\mathcal{W}_{\mathcal{Z}_3} = \left[Y_E L H_1 \bar{E} + Y_D Q H_1 \bar{D} + Y_U Q H_2 \bar{U} + \lambda S(H_2 H_1)\right] + \frac{\kappa}{3} S^3$$

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

NMSSM

Break PQ symmetry explictly with cubic term: $\frac{\kappa}{3} S^3$

$\Rightarrow$ usual $\mathcal{Z}_3$ symmetric NMSSM

Use all renormalisable terms

$\Rightarrow$ usual $\mathcal{Z}_3$ violating NMSSM

Superfield	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	$U(1)_{PQ}$
$\hat{Q}_i$	$\mathbf{3}$	$\mathbf{2}$	$\frac{1}{6}$	-1
$\overline{u}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$-\frac{2}{3}$	0
$\overline{d}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$\frac{1}{3}$	0
$\hat{L}_i$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	-1
$\overline{e}_i$	$\mathbf{1}$	$\mathbf{1}$	1	0
$\hat{H}_u$	$\mathbf{1}$	$\mathbf{2}$	$+\frac{1}{2}$	1
$\hat{H}_d$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	1
$\hat{S}$	$\mathbf{1}$	$\mathbf{1}$	0	-2

$\mathcal{Z}_3$ symmetric

$$\begin{aligned} \mathcal{W}_{\mathcal{Z}_3} &= \left[Y_E L H_1 \bar{E} + Y_D Q H_1 \bar{D} + Y_U Q H_2 \bar{U} + \lambda S (H_2 H_1)\right] + \frac{\kappa}{3} S^3 \\ &= \mathcal{W}^{\mu=0}_{\mathrm{MSSM}} + \lambda S (H_2 H_1) + \frac{\kappa}{3} S^3 \end{aligned}$$

NMSSM

Break PQ symmetry explictly with cubic term: $\frac{\kappa}{3}S^3$

$\Rightarrow$ usual $\mathcal{Z}_3$ symmetric NMSSM

Use all renormalisable terms

$\Rightarrow$ usual $\mathcal{Z}_3$ violating NMSSM

Superfield	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	$U(1)_{PQ}$
$\hat{Q}_i$	$\mathbf{3}$	$\mathbf{2}$	$\frac{1}{6}$	-1
$\overline{u}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$-\frac{2}{3}$	0
$\overline{d}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$\frac{1}{3}$	0
$\hat{L}_i$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	-1
$\overline{e}_i$	$\mathbf{1}$	$\mathbf{1}$	1	0
$\hat{H}_u$	$\mathbf{1}$	$\mathbf{2}$	$+\frac{1}{2}$	1
$\hat{H}_d$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	1
$\hat{S}$	$\mathbf{1}$	$\mathbf{1}$	0	-2

$\mathcal{Z}_3$ symmetric

$$\mathcal{W}_{\mathcal{Z}_3} = [Y_E L H_1 \bar{E} + Y_D Q H_1 \bar{D} + Y_U Q H_2 \bar{U} + \lambda S(H_2 H_1)] + \frac{\kappa}{3}S^3$$

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

$\mathcal{Z}_3$ violating

$$\mathcal{W}_{\not{\mathcal{Z}}_3} = \mathcal{W}^{\mu=0}_{MSSM} + \lambda S H_2 H_1 + \mu H_2 H_1 + \xi_F S + \frac{\mu'}{2}S^2 + \frac{\kappa}{3}S^3$$

NMSSM

Break PQ symmetry explictly with cubic term: $\frac{\kappa}{3} S^3$

$\Rightarrow$ usual $\mathcal{Z}_3$ symmetric NMSSM

Use all renormalisable terms

$\Rightarrow$ usual $\mathcal{Z}_3$ violating NMSSM

Superfield	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	$U(1)_{PQ}$
$\hat{Q}_i$	$\mathbf{3}$	$\mathbf{2}$	$\frac{1}{6}$	-1
$\overline{u}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$-\frac{2}{3}$	0
$\overline{d}_i$	$\overline{\mathbf{3}}$	$\mathbf{1}$	$\frac{1}{3}$	0
$\hat{L}_i$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	-1
$\overline{e}_i$	$\mathbf{1}$	$\mathbf{1}$	1	0
$\hat{H}_u$	$\mathbf{1}$	$\mathbf{2}$	$+\frac{1}{2}$	1
$\hat{H}_d$	$\mathbf{1}$	$\mathbf{2}$	$-\frac{1}{2}$	1
$\hat{S}$	$\mathbf{1}$	$\mathbf{1}$	0	-2

Next-to-Minimal SOFTSUSY can generate spectra in both cases

$\mathcal{Z}_3$ symmetric

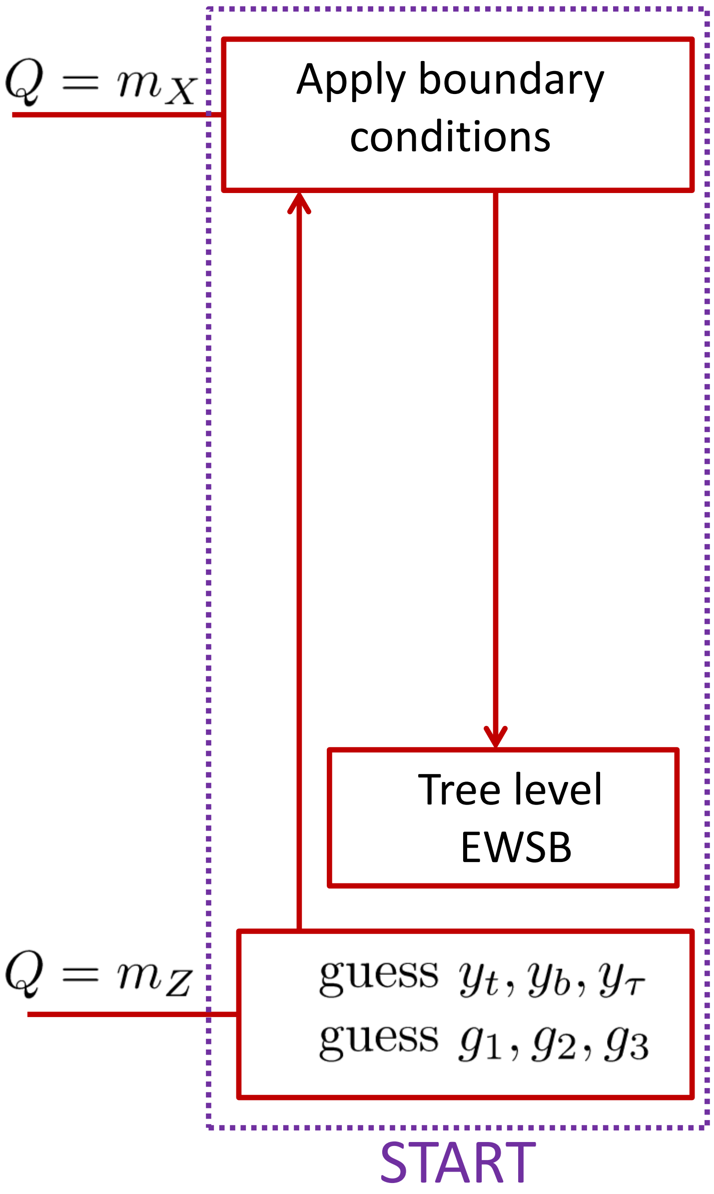
$$\mathcal{W}_{\mathcal{Z}_3} = \left[Y_E L H_1 \bar{E} + Y_D Q H_1 \bar{D} + Y_U Q H_2 \bar{U} + \lambda S (H_2 H_1)\right] + \frac{\kappa}{3} S^3$$

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

$\mathcal{Z}_3$ violating

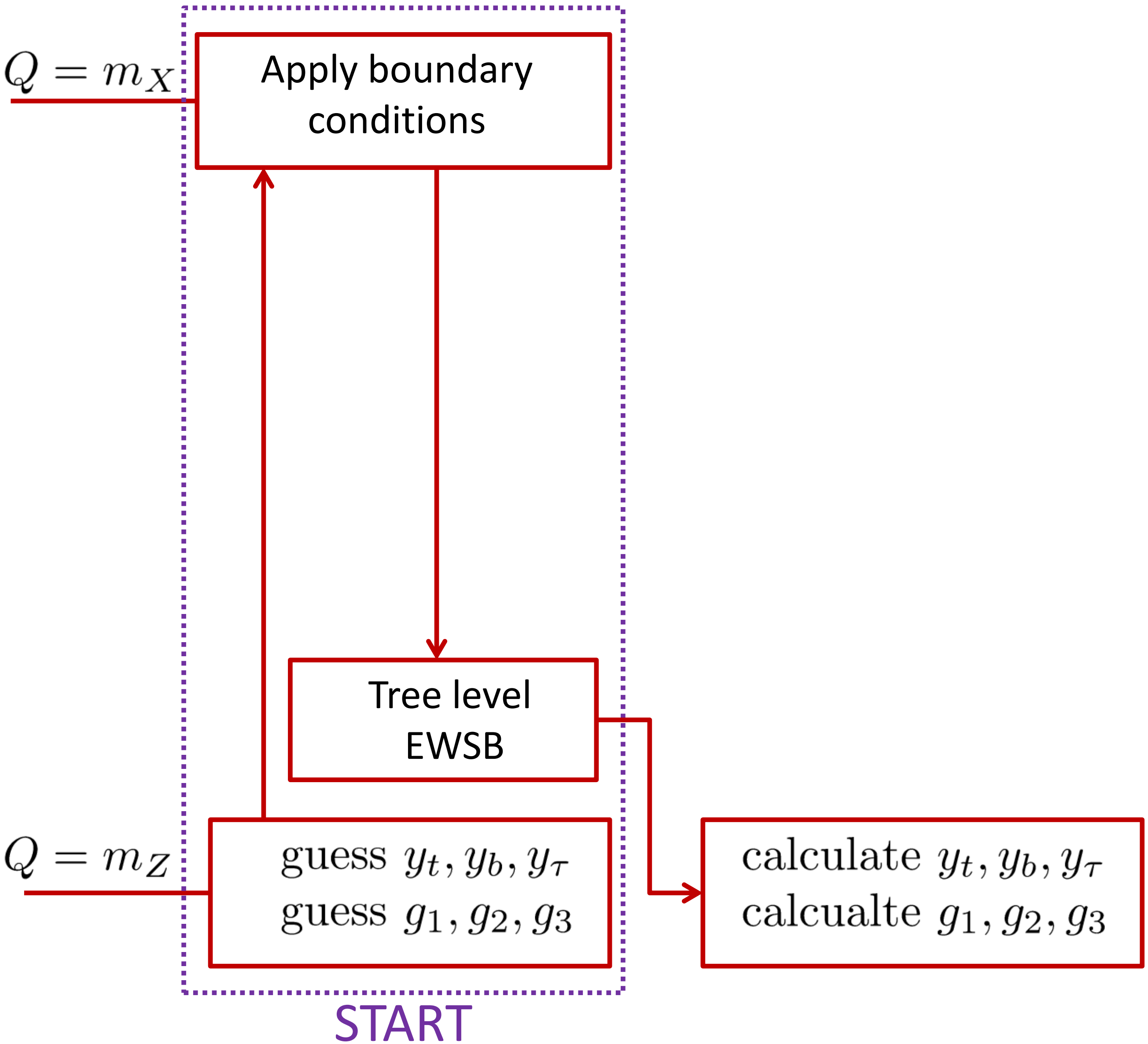
$$\mathcal{W}_{\not{\mathcal{Z}}_3} = \mathcal{W}^{\mu=0}_{MSSM} + \lambda S H_2 H_1 + \mu H_2 H_1 + \xi_F S + \frac{\mu'}{2} S^2 + \frac{\kappa}{3} S^3$$

Algorithm



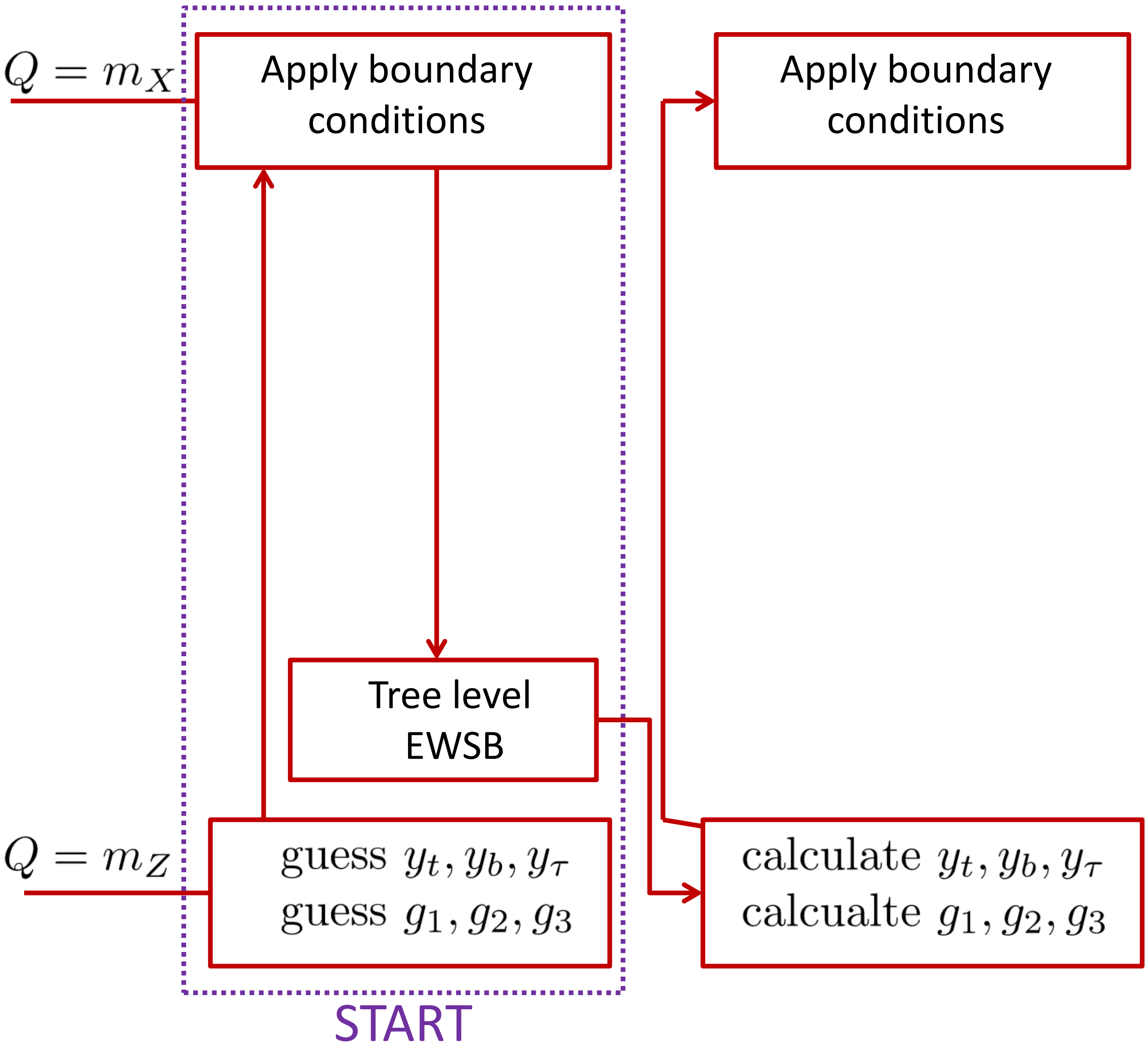
$Q = m_X$
Apply boundary conditions
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
START

Algorithm

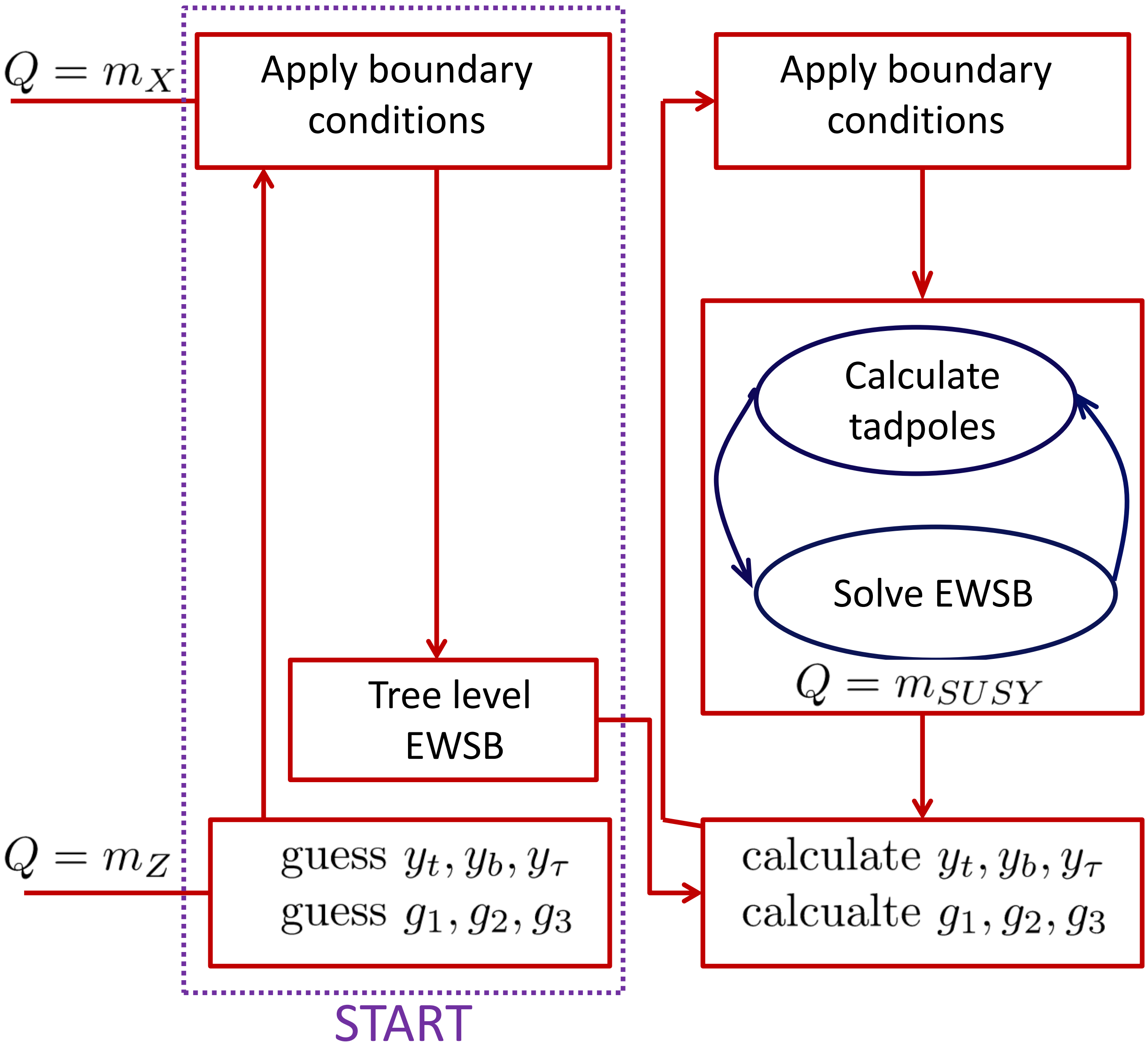


$Q = m_X$
Apply boundary conditions
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
calculate y_t, y_b, y_τ
calcualte g_1, g_2, g_3
START

Algorithm

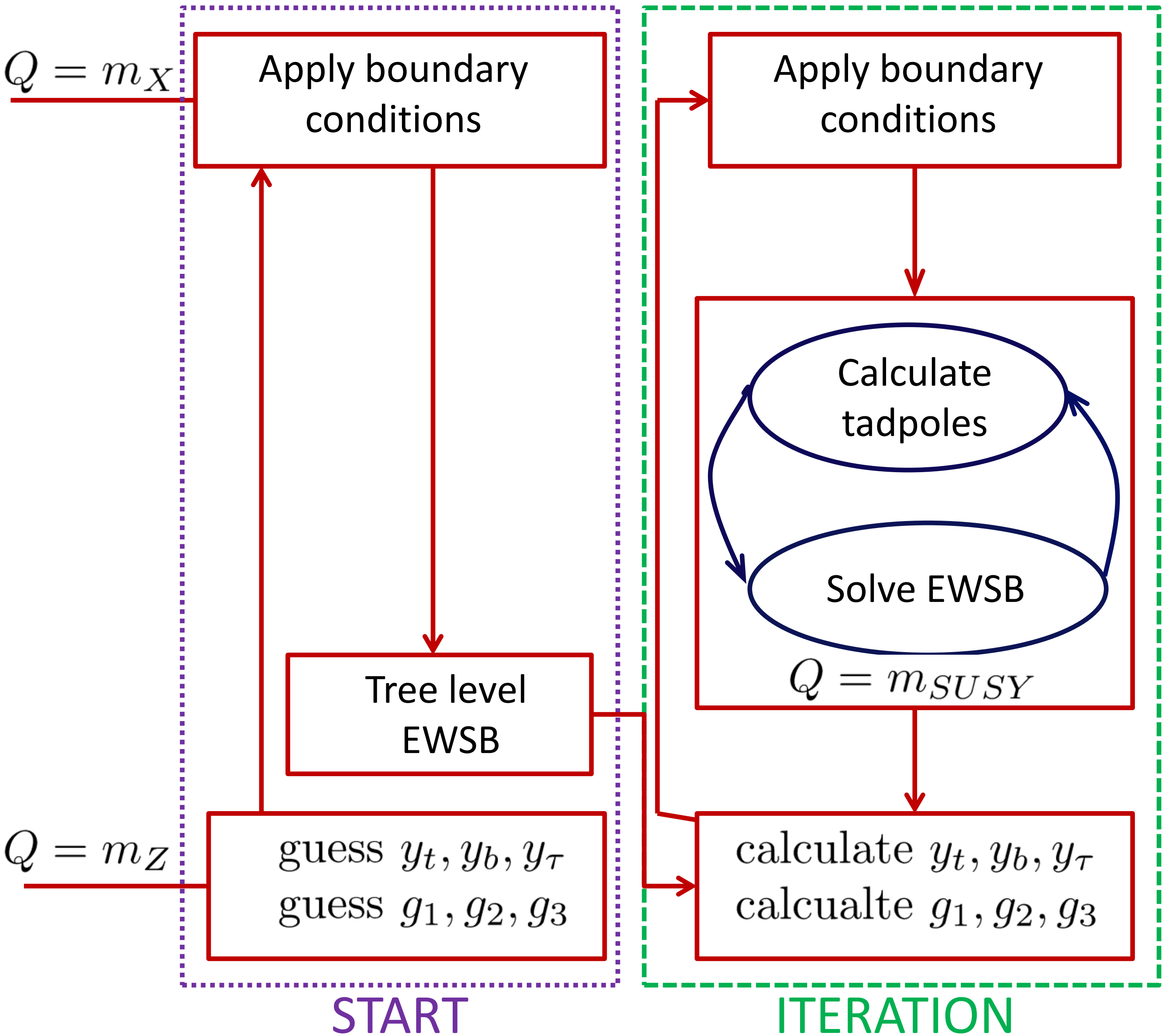

$Q = m_X$
Apply boundary conditions
Apply boundary conditions
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
calculate y_t, y_b, y_τ
calcualte g_1, g_2, g_3
START

Algorithm



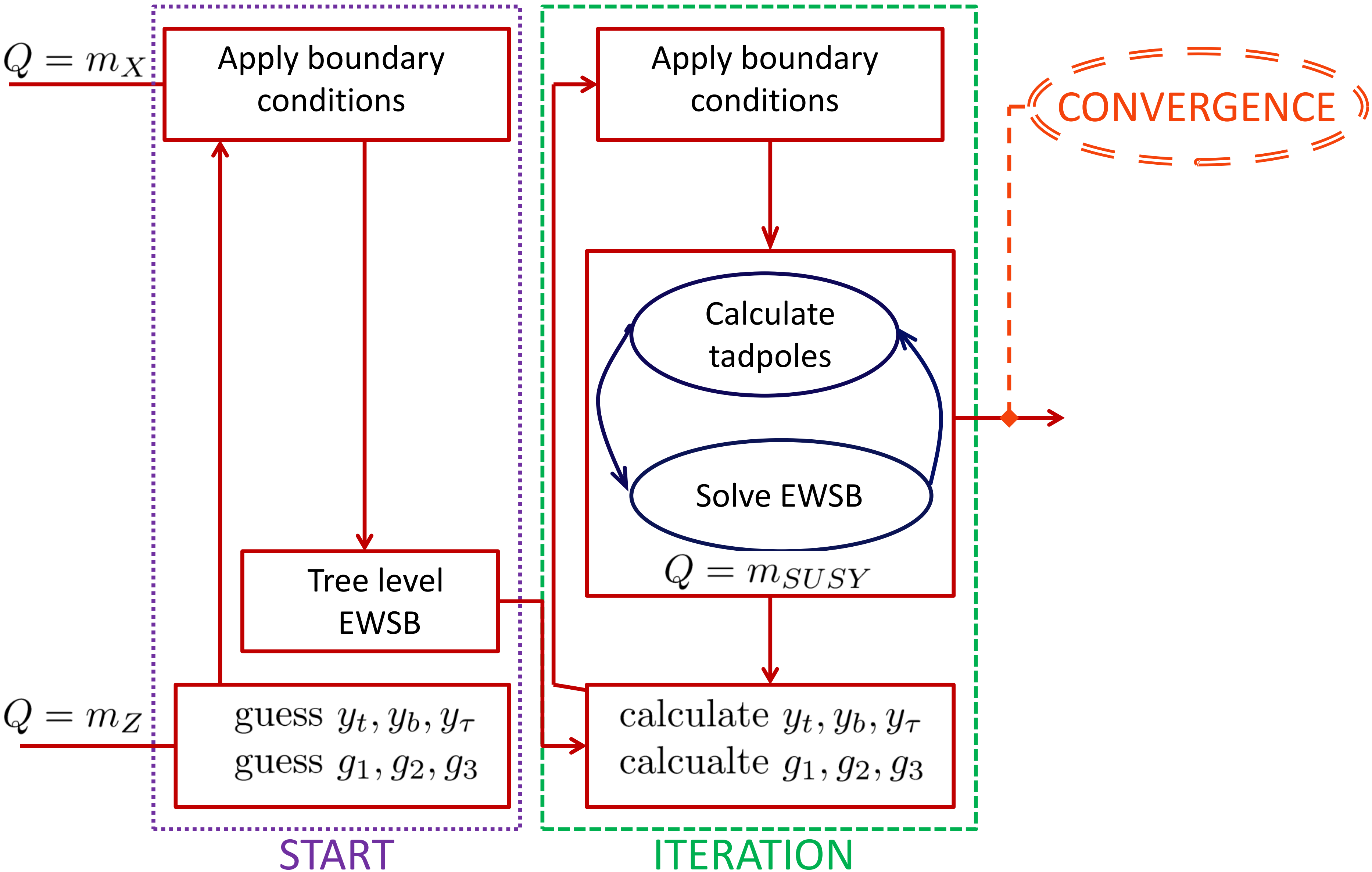
$Q = m_X$
Apply boundary conditions
Apply boundary conditions
Calculate tadpoles
Solve EWSB
$Q = m_{SUSY}$
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
calculate y_t, y_b, y_τ
calcualte g_1, g_2, g_3
START

Algorithm

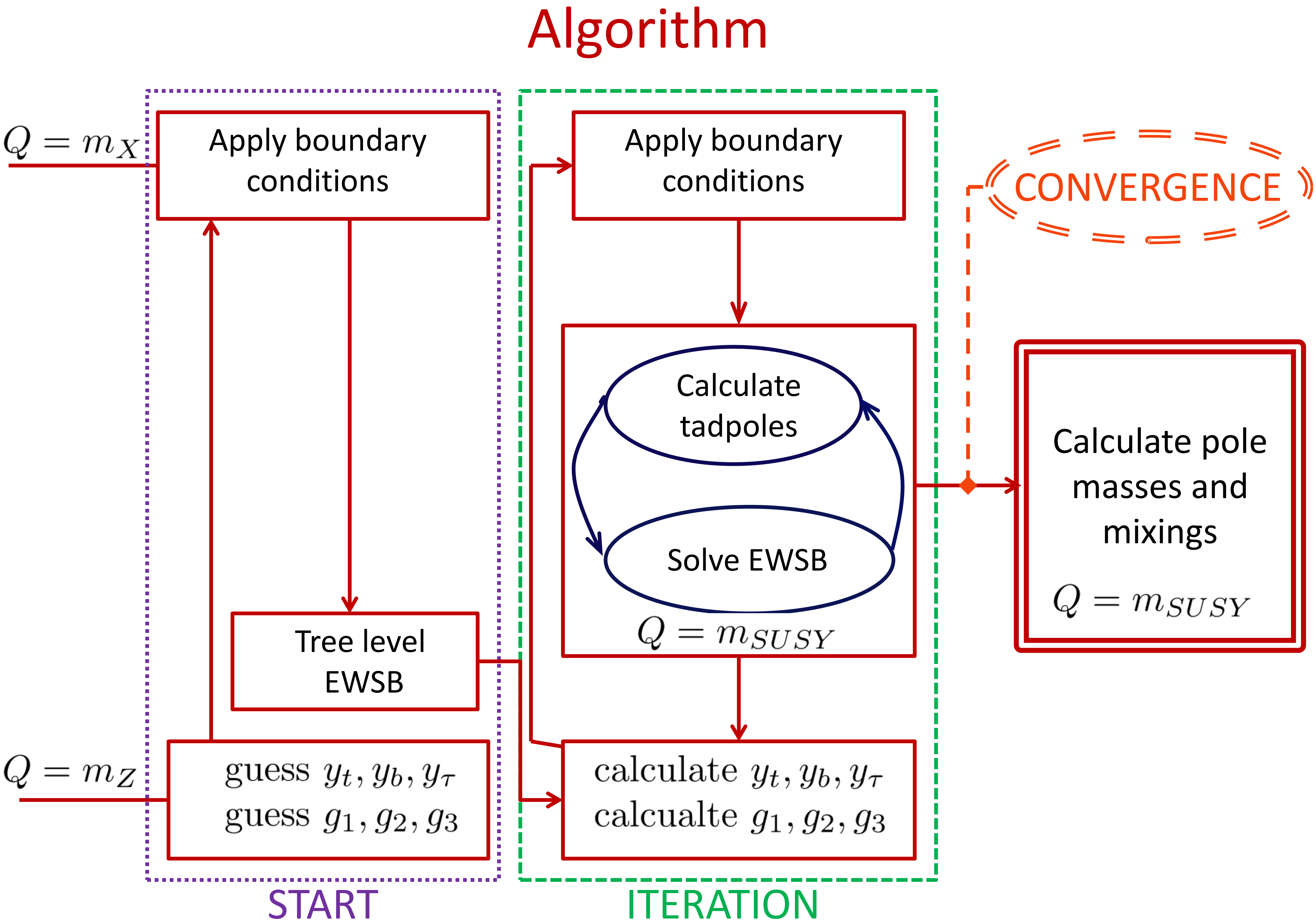


$Q = m_X$
Apply boundary conditions
Apply boundary conditions
Calculate tadpoles
Solve EWSB
$Q = m_{SUSY}$
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
calculate y_t, y_b, y_τ
calcualte g_1, g_2, g_3
START
ITERATION

Algorithm



$Q = m_X$
Apply boundary conditions
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
Apply boundary conditions
Calculate tadpoles
Solve EWSB
$Q = m_{SUSY}$
calculate y_t, y_b, y_τ
calcualte g_1, g_2, g_3
CONVERGENCE
START
ITERATION



Algorithm
$Q = m_X$
Apply boundary conditions
Tree level EWSB
$Q = m_Z$
guess y_t, y_b, y_τ
guess g_1, g_2, g_3
START
Apply boundary conditions
Calculate tadpoles
Solve EWSB
$Q = m_{SUSY}$
calculate y_t, y_b, y_τ
calcualte g_1, g_2, g_3
ITERATION
CONVERGENCE
Calculate pole masses and mixings
$Q = m_{SUSY}$

Technical pieces

- Two-loop Renormalisation group equations (full 3 family)
  - Needed for evolution between scales e.g, GUT and EW
  - Obtained from gen. expressions [Martin,Vaughn PRD 50, 2282]
  - checked against NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]

Technical pieces

- Two-loop Renormalisation group equations (full 3 family)
  - Needed for evolution between scales e.g, GUT and EW
  - Obtained from gen. expressions [Martin,Vaughn PRD 50, 2282]
  - checked against NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]
  - Checked numerically via two independent implementations

Technical pieces

- Two-loop Renormalisation group equations (full 3 family)
  - Needed for evolution between scales e.g, GUT and EW
  - Obtained from gen. expressions [Martin,Vaughn PRD 50, 2282]
  - checked against NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]
  - Checked numerically via two independent implementations
  - Checked numerically against FlexibleSUSY

[PA,Park,Stöckinger ,Voigt, arXiv:1406.2319]

(see also talk by A. Voigt)

Technical pieces

- Two-loop Renormalisation group equations (full 3 family)
  - Needed for evolution between scales e.g, GUT and EW
  - Obtained from gen. expressions [Martin,Vaughn PRD 50, 2282]
  - checked against NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]
  - Checked numerically via two independent implementations
  - Checked numerically against FlexibleSUSY

    [PA,Park,Stöckinger ,Voigt, arXiv:1406.2319]

    (see also talk by A. Voigt)

- Full one loop self energies and tadpoles
  - $\mathcal{Z}_3$ expr. taken from literature [G.Degrassii, P.Slavich Nucl.Phys. B 825, 119]

Technical pieces

- Two-loop Renormalisation group equations (full 3 family)
  - Needed for evolution between scales e.g, GUT and EW
  - Obtained from gen. expressions [Martin,Vaughn PRD 50, 2282]
  - checked against NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]
  - Checked numerically via two independent implementations
  - Checked numerically against FlexibleSUSY [PA,Park,Stöckinger ,Voigt, arXiv:1406.2319]

    (see also talk by A. Voigt)

- Full one loop self energies and tadpoles
  - $\mathcal{Z}_3$ expr. taken from literature [G.Degrassii, P.Slavich Nucl.Phys. B 825, 119]
  - $\not{\mathcal{Z}}_3$ couplings added using NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]

Technical pieces

- Two-loop Renormalisation group equations (full 3 family)
  - — Needed for evolution between scales e.g, GUT and EW
  - — Obtained from gen. expressions [Martin,Vaughn PRD 50, 2282]
  - — checked against NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]
  - — Checked numerically via two independent implementations
  - — Checked numerically against FlexibleSUSY [PA,Park,Stöckinger ,Voigt, arXiv:1406.2319]

    (see also talk by A. Voigt)

- Full one loop self energies and tadpoles
  - — $\mathcal{Z}_3$ expr. taken from literature [G.Degrassii, P.Slavich Nucl.Phys. B 825, 119]
  - — $\not{\mathcal{Z}}_3$ couplings added using NMSSM rev. [Ellwanger,Hugonie,Teixeira Phys.Rept. 96, 1]
  - — Checked numerically via two independent implementations (most cases)
  - — Checked numerically against FlexibleSUSY (all cases)

Higgs Mass accuracy

- Obtain $\overline{\text{DR}}$ mass matrix
- Full one loop self energies for all Higgs states
- Two loop corrections using files from Pietro Slavich (for neutral Higgs states)

$\mathcal{O}(\alpha_s\alpha_b)$, $\mathcal{O}(\alpha_s\alpha_t)$ NMSSM zero momentum
[G.Degrassii, P.Slavich Nucl.Phys. B 825, 119]

$\mathcal{O}\left((\alpha_t+\alpha_b)^2\right)$, $\mathcal{O}(\alpha_\tau^2)$ MSSM parts zero momentum

Higgs Mass accuracy

- Obtain $\overline{\text{DR}}$ mass matrix

- Full one loop self energies for all Higgs states

- Two loop corrections using files from Pietro Slavich
  (for neutral Higgs states)

$\mathcal{O}(\alpha_s \alpha_b),\ \mathcal{O}(\alpha_s \alpha_t)$ NMSSM zero momentum
[G.Degrassii, P.Slavich Nucl.Phys. B 825, 119]

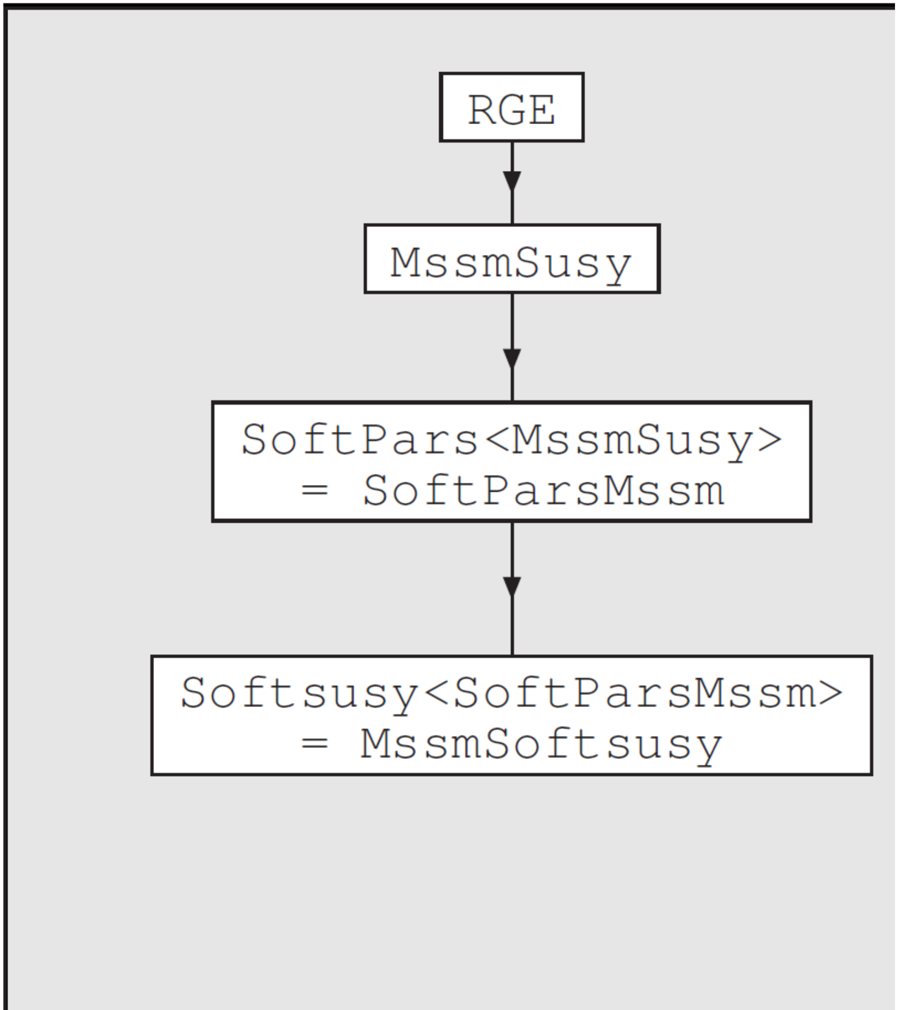
$\mathcal{O}\left((\alpha_t + \alpha_b)^2\right),\ \mathcal{O}(\alpha_\tau^2)$ MSSM parts zero momentum

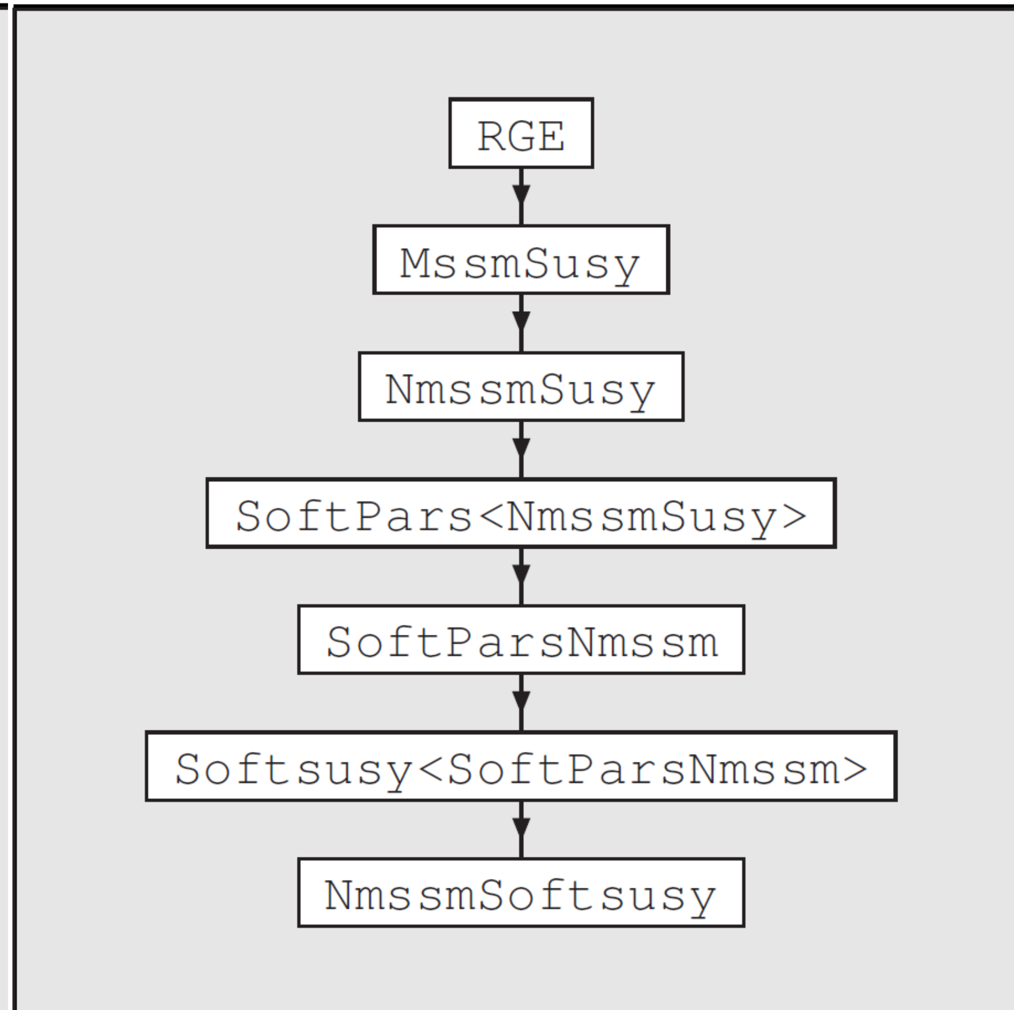
- one loop self energies use at $p^2 = m_{h_i^2}$ via iteration:

$$M_H^2 + \Sigma(p^2 = m_{h_i}^2) \xrightarrow[\text{for eigenvalues}]{\text{diagonalise}} m_{h_i}^2$$

Code structure

- SOFTSUSY code structure has been updated a little
- minimises code duplication, errors
- methods maintain familiarity for expert users





Download and install

Next-to-Minimal SOFTSUSY is an extension of SOFTSUSY

Distributed as part of SOFTSUSY from version 3.4.0 onwards

View/Download latest version from: http://softsusy.hepforge.org/

Download

```
wget http://www.hepforge.org/archive/softsusy/softsusy-3.4.1.tar.gz
tar -xvzf softsusy-3.4.0.tar.gz
cd softsusy
```

Install

```
./configure
make -jN
```

N is the number of jobs to run simultaneously for parallel make

Test Run

```
./softsusy-nmssm.x > inOutFiles/outputTest-nmssm
./softpoint.x leshouches < inOutFiles/nmssmSLHAnoZ3Input > nmSLHAnoZ3Output
./softpoint.x leshouches < inOutFiles/nmssmSLHAZ3Input > nmSLHAZ3Output
```

Running the code

Option 1

Use the SLHA interface by passing an SLHA2 Input file

[CPC 180 (2009) 8]

To run:

call the softpoint executable as shown on the last slide:

```
./softpoint.x leshouches < inOutFiles/nmssmSLHAnoZ3Input > nmSLHAnoZ3Output
./softpoint.x leshouches < inOutFiles/nmssmSLHAZ3Input > nmSLHAZ3Output
```

User written SLHA2 input file

SLHA2 output file

Running the code

WARNING: this interface has changed in version 3.4.0 onwards
Details here are for the new version for 3.4.0 and later

Option 2

Use the command line interface

To run:
call the softpoint executable again
but now without leshouches option
and passing parameters as options like:

```
./softpoint.x  --m0=<m0> --m12=<m12> --a0=<a0> --tanBeta=<tb> --lambda=<lam>
```

Running the code

NMSSM flags	description
`--lambdaAtMsusy`	input λ at scale M_{SUSY}

NMSSM parameters	description
`--m0=<value>`	unified soft scalar mass
`--m12=<value>`	unified soft gaugino mass
`--a0=<value>`	unified trilinear coupling
`--tanBeta=<value>`	$\tan\beta$
`--mHd2=<value>`	soft down-type Higgs mass squared $m_{H_1}^2$
`--mHu2=<value>`	soft up-type Higgs mass squared $m_{H_2}^2$
`--mu=<value>`	μ parameter
`--m3SqrOverCosBetaSinBeta=<value>`	$m_3^2/(\cos\beta\sin\beta)$
`--lambda=<value>`	trilinear superpotential coupling λ
`--kappa=<value>`	trilinear superpotential coupling κ
`--Alambda=<value>`	trilinear soft coupling A_λ
`--Akappa=<value>`	trilinear soft coupling A_κ
`--lambdaS=<value>`	$\lambda\langle S\rangle = \lambda s/\sqrt{2}$
`--xiF=<value>`	linear superpotential coupling ξ_F
`--xiS=<value>`	linear soft coupling ξ_S
`--muPrime=<value>`	bilinear superpotential coupling μ'
`--mPrimeS2=<value>`	bilinear soft coupling $m_S'^2$
`--mS2=<value>`	bilinear soft mass m_S^2

Running the code

Option 3

Write program calling the fixed point iteration e.g distributed sample program: src/main-nmssm.cpp

This sample code was called earlier in the TestRun example:

```
./softsusy-nmssm.x > inOutFiles/outputTest-nmssm
```

Running the code

Option 3

Write program calling the fixed point iteration e.g distributed sample program: src/main-nmssm.cpp

This sample code was called earlier in the TestRun example:

```
./softsusy-nmssm.x > inOutFiles/outputTest-nmssm
```

Call fixed point iteration

```
NmssmSoftsusy n;
n.lowOrg(SemiMsugraBcs, mGutGuess, pars, nmpars, sgnMu, tanb, oneset, uni);
```

Using the code

Get SM inputs at MZ:

```
QedQcd oneset; ///< See "lowe.h" for default definitions parameters
double alphasMZ = 0.1187, mtop = 173.5, mbmb = 4.18;
oneset.setAlpha(ALPHAS, alphasMZ); oneset.setPoleMt(mtop);
oneset.setMass(mBottom, mbmb);
oneset.toMz(); ///< Runs SM fermion masses to MZ
```

Fix parameters :

```
double m12 = 300., a0 = -300., mGutGuess = 2.0e16, tanb = 10.0, m0 = 500.;
int sgnMu = 1; ///< sign of effective mu parameter
double lambda = 0.1, s = 0.0, xiF = 0.0, mupr = 0.0;
```

Fill

```
DoubleVector pars(5), nmpars(5);
pars(1) = m0; pars(2) = m12; pars(3) = a0; pars(4) = a0, pars(5) = a0;
nmpars(1) = lambda; nmpars(2) = kappa; nmpars(3) = s; nmpars(4) = xiF; nmpars(5) = mupr;
bool uni = true; // MGUT defined by g1(MGUT)=g2(MGUT)
softsusy::Z3 = true; // choose Z3 conserving case (selects EWSB conditions)
```

Call fixed point iteration

```
NmssmSoftsusy n;
n.lowOrg(SemiMsugraBcs, mGutGuess, pars, nmpars, sgnMu, tanb, oneset, uni);
```

Using the code

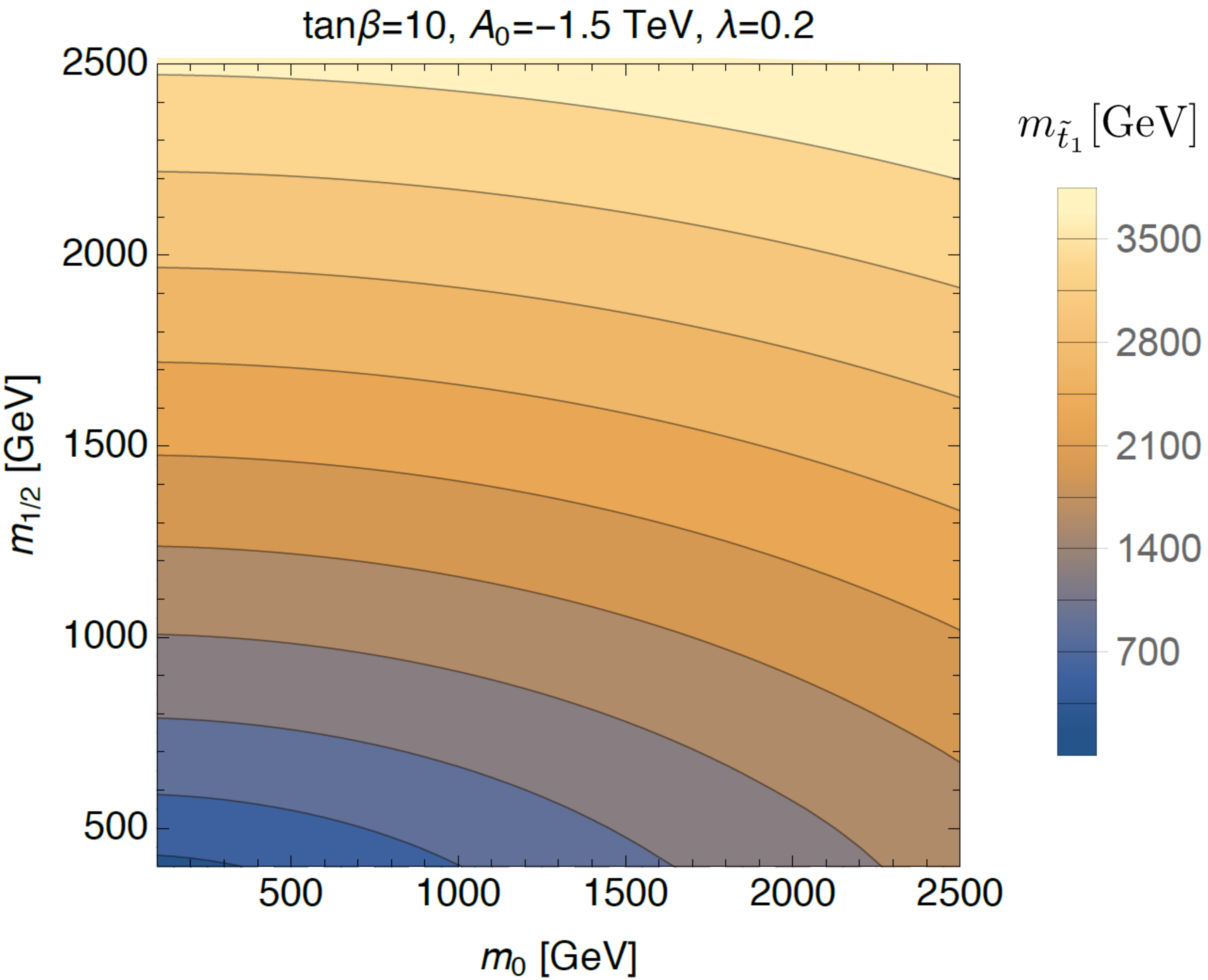
Option 3

Gives great flexibility and control to the user
Wrap grid or random scan of parameters
around last step

→ Produce plots of the physics

For example...

Using the code



Running the code

Interface with NMSSMTools

Setup:

```
$ cd /path/to/NMSSMTools/
$ wget http://www.th.u-psud.fr/NMHDECAY/NMSSMTools_4.1.2.tgz
$ tar xf NMSSMTools_4.1.2.tgz
$ cd /path/to/softsusy/
$ ./setup_nmssmtools.sh \
     --nmssmtools-dir=/path/to/NMSSMTools/NMSSMTools_4.1.2 \
     --compile
```

Run:

```
$ ./softsusy_nmssmtools.x leshouches < slhaInput > slhaOutput
```

SLHA options:

```
Block SOFTSUSY
   15    1       # NMSSMTools compatible output (default: 0)
   16    4       # Select Micromegas option for NMSSMTools
   17    1       # 1:sparticle decays via NMSDECAY (default: 0)
```

Conclusions

- Next-to-Minimal SOFTSUSY is here now!
- Next-to-Minimal SOFTSUSY is fast, reliable, easy to use
- Go to the SOFTSUSY website now!

Download it!

Install it!

Use it!

Hack it!

Tell us what you think of it!

Back up slides

EWSB

$$V_{\rm Higgs} = V_F^H + V_D^H + V_{\rm soft}^H$$
$$= V_{\rm MSSM}^{\mu=0} + V_F^{HN} + V_{\rm soft}^{HN}$$

$$\langle H_1^0 \rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} v_1 \\ 0 \end{pmatrix}, \quad \langle H_2^0 \rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ v_2 \end{pmatrix},$$
$$\langle S \rangle = \frac{1}{\sqrt{2}} s$$

$$V_F^{HN} = |\lambda S + \mu|^2 (|H_2|^2 + |H_1|^2) + |\lambda H_2 H_1 + \kappa S^2 + \mu' S + \xi_S|^2 ,$$
$$V_{\rm soft}^{HN} = m_S^2 |S|^2 + \left(\lambda A_\lambda S H_2 H_1 + \frac{\kappa}{3} A_\kappa S^3 + \frac{m_S'^2}{2} S^2 + \xi_S S + \text{h.c.} \right) .$$

$$\mu_{\rm eff} \equiv \mu + \frac{\lambda s}{\sqrt{2}}, \quad B_{\rm eff} \equiv A_\lambda + \frac{\kappa s}{\sqrt{2}} \quad (m_3^2)_{\rm eff} \equiv \frac{\lambda s}{\sqrt{2}} B_{\rm eff} + \widehat{m}_3^2 \quad \widehat{m}_3^2 \equiv m_3^2 + \lambda\left(\frac{\mu' s}{\sqrt{2}} + \xi_F \right)$$

$$m_{H_1}^2 = -\frac{M_Z^2}{2}\cos(2\beta) - \frac{\lambda^2}{2} v_2^2 + (m_3^2)_{\rm eff} \tan\beta - |\mu_{\rm eff}|^2 ,$$
$$m_{H_2}^2 = \frac{M_Z^2}{2}\cos(2\beta) - \frac{\lambda^2}{2} v_1^2 + \frac{(m_3^2)_{\rm eff}}{\tan\beta} - |\mu_{\rm eff}|^2 ,$$
$$m_S^2 = -\kappa^2 s^2 - \frac{\lambda^2}{2} v^2 + \kappa\lambda v_2 v_1 + \lambda A_\lambda \frac{v_2 v_1}{\sqrt{2} s} - \kappa A_\kappa s - m_S'^2 - \mu'^2 + 2\kappa\xi_F - 3\kappa s \mu'$$

EWSB

Options for parameters fixed by EWSB conditions:

$\mathcal{Z}_3$ (covering $\{s, \kappa, m_S\}$ and $\{m_{H_u}^2, m_{H_d}^2, m_S^2\}$)

$\not{\mathcal{Z}_3}$ (covering $\{m_{H_u}^2, m_{H_d}^2, m_S^2\}$ and $\{\mu, m_3^2, \xi_S\}$)

$$\{s, \kappa, m_S\} \qquad \{m_{H_u}^2, m_{H_d}^2, m_S^2\} \qquad \{\mu, m_3^2, \xi_S\}$$

$$\mu_{\rm eff} \equiv \mu + \frac{\lambda s}{\sqrt{2}} \quad B_{\rm eff} \equiv A_\lambda + \frac{\kappa s}{\sqrt{2}} \quad (m_3^2)_{\rm eff} \equiv \frac{\lambda s}{\sqrt{2}} B_{\rm eff} + \widehat{m}_3^2 \quad \widehat{m}_3^2 \equiv m_3^2 + \lambda\left(\frac{\mu' s}{\sqrt{2}} + \xi_F\right)$$

$$m_{H_1}^2 = -\frac{M_Z^2}{2}\cos(2\beta) - \frac{\lambda^2}{2} v_2^2 + (m_3^2)_{\rm eff} \tan\beta - |\mu_{\rm eff}|^2 ,$$

$$m_{H_2}^2 = \frac{M_Z^2}{2}\cos(2\beta) - \frac{\lambda^2}{2} v_1^2 + \frac{(m_3^2)_{\rm eff}}{\tan\beta} - |\mu_{\rm eff}|^2 ,$$

$$m_S^2 = -\kappa^2 s^2 - \frac{\lambda^2}{2} v^2 + \kappa\lambda v_2 v_1 + \lambda A_\lambda \frac{v_2 v_1}{\sqrt{2} s} - \kappa A_\kappa s - m_S'^2 - \mu'^2 + 2\kappa\xi_F - 3\kappa s \mu'$$