# POLXINO: Neutralino, chargino and gluino two-loop pole mass calculator

H. Eberl

Institute for High Energy Physics
Austrian Academy of Sciences
Vienna

Tools 2008, MPI Munich, July 2008

# Outline

1. Motivation
   - Challenging experimental accuracy
2. Underlying works
3. Outline of the calc.
   - Tree-level mass matrices
   - Pole mass to two loop order $\mathcal{O}(\alpha\alpha_S, Y^4, \alpha_S Y^2)$
   - The perturbative expansion
   - Automatisms vs. generic calculations
   - Diagrammatics
   - Checks
4. Numerical results
   - Two scenarios
5. Description of Polxino
6. Conclusion & Outlook

# challenging experimental accuracy

- Accuracy at future collider experiments at the ILC and LHC is quite challenging. The physics interplay has been worked out by the LHC/LC study group [G. Weiglein *et al.*, 2004]

accuracy from experiment at LHC + ILC (SPS1a')

| Particle | Mass | "LHC" | "ILC" | "LHC+ILC" |
|---|---|---|---|---|
| $\tilde{\chi}_1^0$ | 97.7 | 4.8 | 0.05 | 0.05 |
| $\tilde{\chi}_2^0$ | 183.9 | 4.7 | 1.2 | 0.08 |
| $\tilde{\chi}_1^\pm$ | 183.7 | | 0.55 | 0.55 |
| $\tilde{q}_R$ | 547.2 | 7 – 12 | – | 5 – 11 |
| $\tilde{q}_L$ | 564.7 | 8.7 | – | 4.9 |
| $\tilde{g}$ | 607.1 | 8.0 | – | 6.5 |

[SPA, J. A. Aguilar-Saavedra *et al.*, 2005 ], all numbers in GeV

- $LHC \sim m^2_{\chi_1^0} - m^2_{\chi_2^0}$, $ILC \sim m_{\chi_1^0}$
- relate observables to $\overline{\mathrm{DR}}$ -input at the two-loop level

# Underlying works

- "Two-loop SUSY QCD corrections to the neutralino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  PLB 649 (2007) 67, arXiv:0612276 [hep-ph].
- "Two-loop SUSY QCD corrections to the chargino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  EPJC 53 (2008) 621, arXiv:0706.0781 [hep-ph].
- "Leading two-loop Yukawa corrections to the pole masses of SUSY fermions in the MSSM"
  by R. Schöfbeck and H. Eberl,
  NPB 798 (2008) 146, arXiv:0711.2731 [hep-ph].
- "Precise predictions for the masses of supersymmetric fermions including two-loop corrections"
  by R. Schöfbeck, Vienna 2008, PhD thesis
  `http://wwwhephy.oeaw.ac.at/pw3/theory/susy/diss_Schoefbeck_R.pdf`
  will be put to the arXiv soon

# Underlying works

- "Two-loop SUSY QCD corrections to the neutralino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  PLB 649 (2007) 67, arXiv:0612276 [hep-ph].
- "Two-loop SUSY QCD corrections to the chargino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  EPJC 53 (2008) 621, arXiv:0706.0781 [hep-ph].
- "Leading two-loop Yukawa corrections to the pole masses of SUSY fermions in the MSSM"
  by R. Schöfbeck and H. Eberl,
  NPB 798 (2008) 146, arXiv:0711.2731 [hep-ph].
- "Precise predictions for the masses of supersymmetric fermions including two-loop corrections"
  by R. Schöfbeck, Vienna 2008, PhD thesis
  `http://wwwhephy.oeaw.ac.at/pw3/theory/susy/diss_Schoefbeck_R.pdf`
  will be put to the arXiv soon

ÖAW Austrian Academy of Sciences

# Underlying works

- "Two-loop SUSY QCD corrections to the neutralino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  PLB 649 (2007) 67, arXiv:0612276 [hep-ph].
- "Two-loop SUSY QCD corrections to the chargino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  EPJC 53 (2008) 621, arXiv:0706.0781 [hep-ph].
- "Leading two-loop Yukawa corrections to the pole masses of SUSY fermions in the MSSM"
  by R. Schöfbeck and H. Eberl,
  NPB 798 (2008) 146, arXiv:0711.2731 [hep-ph].
- "Precise predictions for the masses of supersymmetric fermions including two-loop corrections"
  by R. Schöfbeck, Vienna 2008, PhD thesis
  `http://wwwhephy.oeaw.ac.at/pw3/theory/susy/diss_Schoefbeck_R.pdf`
  will be put to the arXiv soon

ÖAW Austrian Academy of Sciences

# Underlying works

- "Two-loop SUSY QCD corrections to the neutralino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  PLB 649 (2007) 67, arXiv:0612276 [hep-ph].
- "Two-loop SUSY QCD corrections to the chargino masses in the MSSM"
  by R. Schöfbeck and H. Eberl,
  EPJC 53 (2008) 621, arXiv:0706.0781 [hep-ph].
- "Leading two-loop Yukawa corrections to the pole masses of SUSY fermions in the MSSM"
  by R. Schöfbeck and H. Eberl,
  NPB 798 (2008) 146, arXiv:0711.2731 [hep-ph].
- "Precise predictions for the masses of supersymmetric fermions including two-loop corrections"
  by R. Schöfbeck, Vienna 2008, PhD thesis
  `http://wwwhephy.oeaw.ac.at/pw3/theory/susy/diss_Schoefbeck_R.pdf`
  will be put to the arXiv soon

ÖAW
Austrian Academy of Sciences

# Outline of the calculation

Austrian Academy of Sciences

# Tree-level mass matrices

## Chargino mass matrix at tree level

$$
\begin{aligned}
X &= \begin{pmatrix} M_2 & \sqrt{2} M_W \sin\beta \\ \sqrt{2} M_W \cos\beta & \mu \end{pmatrix} \\
M_D^{\tilde{\chi}^+} &= U^* X V^\dagger = \mathrm{diag}(m_{\tilde{\chi}_1^+}, m_{\tilde{\chi}_2^+})
\end{aligned}
$$

## Neutralino mass matrix at tree level

$$
\begin{aligned}
Y &= \begin{pmatrix} M_1 & & & \mathrm{sym.} \\ 0 & M_2 & & \\ -M_Z \cos\beta \sin\theta_W & M_Z \cos\beta \cos\theta_W & 0 & \\ M_Z \sin\beta \sin\theta_W & -M_Z \sin\beta \cos\theta_W & -\mu & 0 \end{pmatrix} \\
M_D^{\tilde{\chi}^0} &= N^* Y N^\dagger = \mathrm{diag}(m_{\tilde{\chi}_1^0}, m_{\tilde{\chi}_2^0}, m_{\tilde{\chi}_3^0}, m_{\tilde{\chi}_4^0})
\end{aligned}
$$

# pole mass to two loop order I

## renormalized 2-point function $\Gamma^{(2)\tilde{\chi}\tilde{\chi}}$

$$G^{(2)-1}_{\tilde{\chi}\tilde{\chi}} = -i\Gamma^{(2)\tilde{\chi}\tilde{\chi}} = -i \begin{pmatrix} -M_D + \hat{\Sigma}^L_m(s) & \sigma \cdot k(1 + \hat{\Sigma}^R_k(s)) \\ \bar{\sigma} \cdot k(1 + \hat{\Sigma}^L_k(s)) & -M_D^\dagger + \hat{\Sigma}^R_m(s) \end{pmatrix}$$

$$s = k^2$$

## Pole mass condition

$$0 = \det\left(s - \left(M_D - \hat{\Sigma}^L_m(s)\right) \cdot \left(1 + \hat{\Sigma}^L_k(s)\right)^{-1} \cdot \left(M_D^\dagger - \hat{\Sigma}^R_m(s)\right) \cdot \left(1 + \hat{\Sigma}^R_k(s)\right)^{-1}\right)$$

## Iterative solution to two-loop order

$$s_{i,pole} = |m_{\tilde{\chi}_i}|^2 - \delta m^{2(1)}_{\tilde{\chi}_i} - \delta m^{2(2)}_{\tilde{\chi}_i} \text{with}$$

$$\delta m^{2(1)}_{\tilde{\chi}_i} = \Pi^{(1)}_{ii} \quad \text{and} \quad \delta m^{2(2)}_{\tilde{\chi}_i} = \Pi^{(2)}_{ii} - \Pi^{(1)}_{ii}\frac{\partial \Pi^{(1)}_{ii}}{\partial s} - \sum_{i \neq j} \frac{\Pi^{(1)}_{ij}\Pi^{(1)}_{ji}}{m_i^2 - m_j^2}$$

# pole mass to two loop order I

## renormalized 2-point function $\Gamma^{(2)\tilde{\chi}\tilde{\chi}}$

$$G^{(2)-1}_{\tilde{\chi}\tilde{\chi}} = -i\Gamma^{(2)\tilde{\chi}\tilde{\chi}} = -i \begin{pmatrix} -M_D + \hat{\Sigma}^L_m(s) & \sigma\cdot k(1+\hat{\Sigma}^R_k(s)) \\ \bar{\sigma}\cdot k(1+\hat{\Sigma}^L_k(s)) & -M_D^\dagger + \hat{\Sigma}^R_m(s) \end{pmatrix}$$

$$s = k^2$$

## Pole mass condition

$$0 = \det\left(s - \left(M_D - \hat{\Sigma}^L_m(s)\right)\cdot\left(1+\hat{\Sigma}^L_k(s)\right)^{-1}\cdot\left(M_D^\dagger - \hat{\Sigma}^R_m(s)\right)\cdot\left(1+\hat{\Sigma}^R_k(s)\right)^{-1}\right)$$

## Iterative solution to two-loop order

$$s_{i,pole} = |m_{\tilde{\chi}_i}|^2 - \delta m^{2(1)}_{\tilde{\chi}_i} - \delta m^{2(2)}_{\tilde{\chi}_i}\text{with}$$

$$\delta m^{2(1)}_{\tilde{\chi}_i} = \Pi^{(1)}_{ii} \quad\text{and}\quad \delta m^{2(2)}_{\tilde{\chi}_i} = \Pi^{(2)}_{ii} - \Pi^{(1)}_{ii}\frac{\partial \Pi^{(1)}_{ii}}{\partial s} - \sum_{i\neq j}\frac{\Pi^{(1)}_{ij}\Pi^{(1)}_{ji}}{m_i^2 - m_j^2}$$

# pole mass to two loop order I

## renormalized 2-point function $\Gamma^{(2)\tilde{\chi}\tilde{\chi}}$

$$G^{(2)-1}_{\tilde{\chi}\tilde{\chi}} = -i\Gamma^{(2)\tilde{\chi}\tilde{\chi}} = -i \begin{pmatrix} -M_D + \hat{\Sigma}^L_m(s) & \sigma \cdot k(1+\hat{\Sigma}^R_k(s)) \\ \bar{\sigma}\cdot k(1+\hat{\Sigma}^L_k(s)) & -M^\dagger_D + \hat{\Sigma}^R_m(s) \end{pmatrix}$$

$$s = k^2$$

## Pole mass condition

$$0 = \det\left(s - \left(M_D - \hat{\Sigma}^L_m(s)\right)\cdot\left(1+\hat{\Sigma}^L_k(s)\right)^{-1}\cdot\left(M^\dagger_D - \hat{\Sigma}^R_m(s)\right)\cdot\left(1+\hat{\Sigma}^R_k(s)\right)^{-1}\right)$$

## Iterative solution to two-loop order

$$s_{i,pole} \quad = \quad |m_{\tilde{\chi}_i}|^2 - \delta m^{2(1)}_{\tilde{\chi}_i} - \delta m^{2(2)}_{\tilde{\chi}_i} \text{with}$$

$$\delta m^{2(1)}_{\tilde{\chi}_i} = \Pi^{(1)}_{ii} \quad \text{and} \quad \delta m^{2(2)}_{\tilde{\chi}_i} = \Pi^{(2)}_{ii} - \Pi^{(1)}_{ii}\frac{\partial \Pi^{(1)}_{ii}}{\partial s} - \sum_{i\neq j}\frac{\Pi^{(1)}_{ij}\Pi^{(1)}_{ji}}{m_i^2 - m_j^2}$$

# pole mass to two loop order II

$$
\begin{aligned}
\Pi^{(1)}_{ij} &= m_i m_j^\dagger {\hat\Sigma_k^{(1)L}}_{ij} + |m_j|^2 {\hat\Sigma_k^{(1)R}}_{ij} + m_j^\dagger {\hat\Sigma_m^{(1)L}}_{ij} + m_i {\hat\Sigma_m^{(1)R}}_{ij} \\
\Pi^{(2)}_{ii} &= m_i^\dagger {\hat\Sigma_m^{(2)L}}_{ii} + m_i {\hat\Sigma_m^{(2)R}}_{ii} + |m_i|^2 ({\hat\Sigma_k^{(2)L}}_{ii} + {\hat\Sigma_k^{(2)R}}_{ii}) \\
&\quad - \Bigg( |m_i|^2 \left( {\Sigma_k^{(1)R}}_{ij} {\Sigma_k^{(1)R}}_{ji} + {\Sigma_k^{(1)L}}_{ij} {\Sigma_k^{(1)L}}_{ji} \right) + m_i^\dagger ({\Sigma_k^{(1)R}}_{ij} {\Sigma_m^{(1)L}}_{ji} + {\Sigma_m^{(1)L}}_{ij} {\Sigma_k^{(1)L}}_{ji}) \\
&\quad + m_i {\Sigma_k^{(1)L}}_{ij} {\Sigma_m^{(1)R}}_{ji} + m_j {\Sigma_k^{(1)R}}_{ij} {\Sigma_m^{(1)R}}_{ji} + m_i^\dagger m_j {\Sigma_k^{(1)R}}_{ij} {\Sigma_k^{(1)L}}_{ji} + {\Sigma_m^{(1)L}}_{ij} {\Sigma_m^{(1)R}}_{ji} \Bigg) ,
\end{aligned}
$$

with all selfenergies evaluated at $s = |m_i|^2$.

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

ÖAW Austrian Academy of Sciences

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

ÖAW Austrian Academy of Sciences

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars)
  to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters
    [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

ÖAW
Austrian Academy of Sciences

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars)
  to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars)
  to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters
    [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

Austrian Academy of Sciences

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

Austrian Academy of Sciences

# The perturbative expansion

- regularization: DRED
  gauge fields come with unphysical scalar fields ($\epsilon$-scalars) to restore the counting of DOF for SUSY
- renormalization: $\overline{\mathrm{DR}}\,'$ ($\overline{\mathrm{DR}}$)
  - subtract $1/\epsilon$-piece
  - absorb unphysical $m_\epsilon$ parameter in sfermion mass parameters [Martin '01]
- IR-regulating mass for massless vectors
- EWSB VEVs
  - expand around the loop corrected potential
  - no tadpoles needed
- gauge
  - Landau ($R_{\xi=0}$) gauge for massive vectors
  - Feynman ($R_\xi$) gauge for the gluon

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime

- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

we did both!

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

we did both!

(and happily, the results agree precisely)

Austrian Academy of Sciences

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
    - + use for all kinds of particles, not only fermions
    - + no inherent restriction on scheme, gauge, ...
    - + disentangle graphs and counter-term contributions
    - - complicated code
    - - rather slow in compilation and at runtime
- using generic once-and-for-all results
    - + simple code
    - + very fast in programming and execution
    - - a little less flexible
    - + already available "gluinopole.c" [S.P. Martin, '05]

we did both!

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

we did both!

(and happily, the results agree precisely)

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime
- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

we did both!

(and happily, the results agree precisely)

Austrian Academy of Sciences

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
    - + use for all kinds of particles, not only fermions
    - + no inherent restriction on scheme, gauge, ...
    - + disentangle graphs and counter-term contributions
    - - complicated code
    - - rather slow in compilation and at runtime
- using generic once-and-for-all results
    - + simple code
    - + very fast in programming and execution
    - - a little less flexible
    - + already available "gluinopole.c" [S.P. Martin, '05]

we did both!

(and happily, the results agree precisely)

Austrian Academy of Sciences

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
  - + use for all kinds of particles, not only fermions
  - + no inherent restriction on scheme, gauge, ...
  - + disentangle graphs and counter-term contributions
  - - complicated code
  - - rather slow in compilation and at runtime

- using generic once-and-for-all results
  - + simple code
  - + very fast in programming and execution
  - - a little less flexible
  - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

Austrian Academy of Sciences

# A sidenote on the approaches

## How to do tensor reduction best

- use CAS (Mathematica) to design one-for-all tools producing auto-generated code
    - + use for all kinds of particles, not only fermions
    - + no inherent restriction on scheme, gauge, ...
    - + disentangle graphs and counter-term contributions
    - - complicated code
    - - rather slow in compilation and at runtime
- using generic once-and-for-all results
    - + simple code
    - + very fast in programming and execution
    - - a little less flexible
    - + already available "gluinopole.c" [S.P. Martin, '05]

**we did both!**

(and happily, the results agree precisely)

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 ($2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d-4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation
up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d - 4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

$\Rightarrow$ completely independent calculation

up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d-4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

$\Rightarrow$ completely independent calculation

up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d - 4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation

up to the evaluation of the basis integrals

Austrian Academy of Sciences

# The hard(er) way

## CAS based approach

- Generation of Diagrams
    - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
    - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
    - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
    - Taylor expansion of Tarcer basis integrals in $d - 4$ to scalar basis integrals
- Numerical evaluation
    - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
    - use some FormCalc subroutines for calculating the SUSY spectra
    - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation
up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams) [J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d-4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

$\Rightarrow$ completely independent calculation

up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d - 4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation
up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams) [J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d - 4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation

up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d-4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

$\Rightarrow$ completely independent calculation

up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

- Generation of Diagrams
  - FeynArts 3.2 ($2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d - 4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation

up to the evaluation of the basis integrals

# The hard(er) way

## CAS based approach

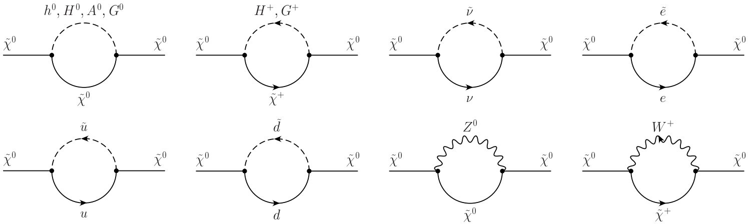
- Generation of Diagrams
  - FeynArts 3.2 $(2 \cdot (36 + 128)$ diagrams)[J.Küblbeck, T. Hahn, et. al, '90]
- Analytic simplification
  - conversion to FeynCalc 4.0.2 and evaluation of color and Dirac traces
  - tensor reduction using Tarcer [R. Mertig, R. Scharf, '98]
  - Taylor expansion of Tarcer basis integrals in $d-4$ to scalar basis integrals
- Numerical evaluation
  - auto-creating Fortran code calling TSIL basis integrals [S.P. Martin and D.G. Robertson, '05]
  - use some FormCalc subroutines for calculating the SUSY spectra
  - use two-loop methods also for one-loop integrals by multiplying with some artificial vacuum bubble

⇒ completely independent calculation

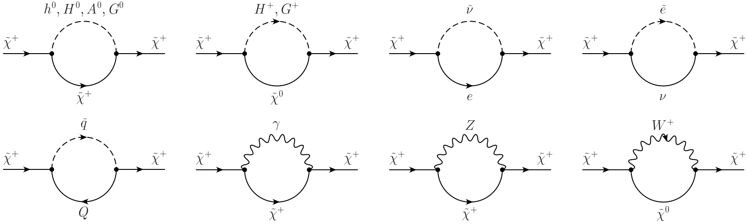
up to the evaluation of the basis integrals

# Generating Amplitudes using FeynArts

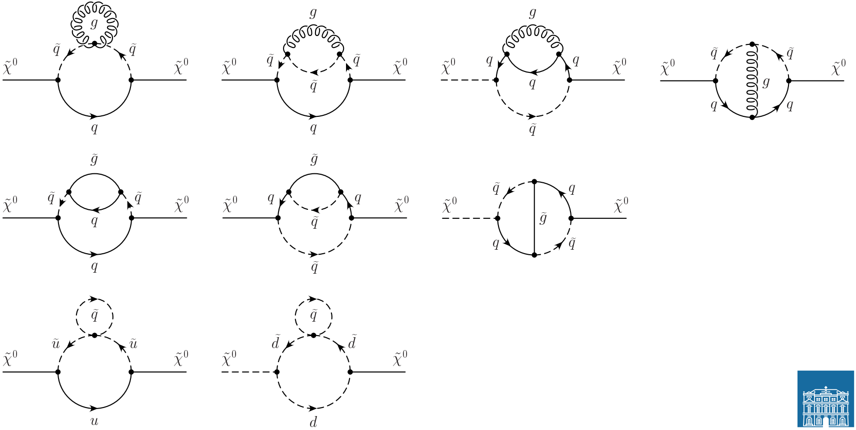
## Neutralinos



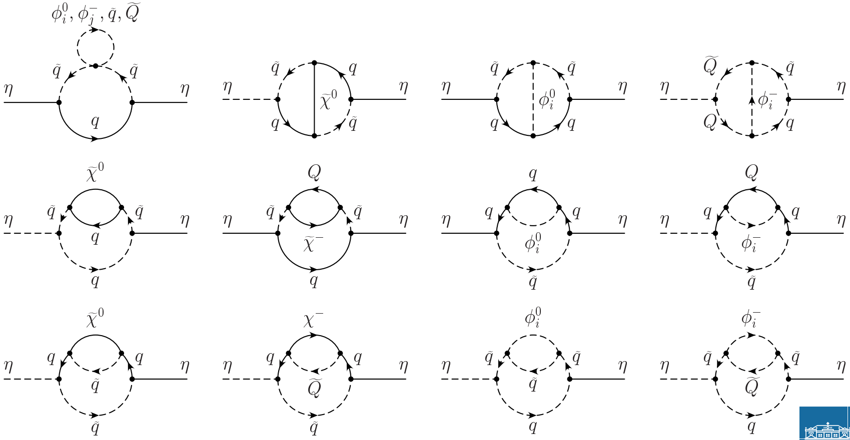
## Charginos



# Two loop SQCD diagrams for neutralinos

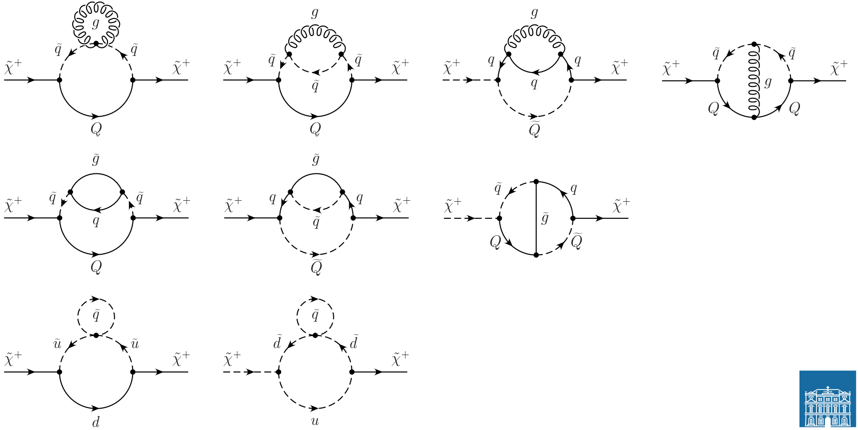


# Two loop Yukawa corr. diagrams for neutralinos

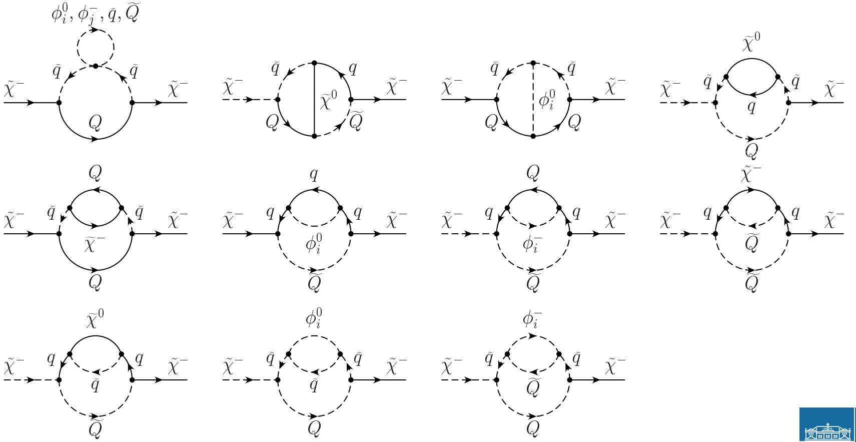


$$\eta = \tilde{\chi}^0$$

# Two loop SQCD diagrams for charginos



# Two loop Yukawa corr. diagrams for charginos



## Checking the Result

### Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

### Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno [W. Porod, '03]

### Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

Austrian Academy of Sciences

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

Austrian Academy of Sciences

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

Austrian Academy of Sciences

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

Austrian Academy of Sciences

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde\chi_i}$ against Spheno [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno

  [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

Austrian Academy of Sciences

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno

  [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

# Checking the Result

## Analytic Checks

- gauge parameter independence of the gluon gauge parameter
- explicit cancellation of $m_\epsilon$ contributions in $\overline{\mathrm{DR}}\,'$
- IR finiteness by extraction of the IR divergent parts in each diagram
- rather non-trivial: RGE check (UV finiteness) on $\mathrm{tr}(Y^\dagger Y) = \mathrm{tr}(M_D^\dagger M_D)$
- one-loop results agree with previous work [W. Öller '03, T. Fritzsche '04]

## Checks on the Numerics

- RGE check of mass shifts $\delta m_{\tilde{\chi}_i}$ against Spheno

  [W. Porod, '03]

## Check on the whole strategy

- SQCD corrections to the gluino pole mass agree with known result [S.P. Martin, '05]

ÖAW Austrian Academy of Sciences

# Numerics

# Two scenarios

- Gaugino scenario:
  the SPS1a′ benchmark point J.A. Aguilar-Saavedra '05 *et.al.*, Porod '03
  The $\overline{\mathrm{DR}}$ SUSY Lagrangian parameters at the scale $Q = 1\ \mathrm{TeV}$

| Parameter | SPS1a′ value | Parameter | SPS1a′ value |
|---|---|---|---|
| $g'$ | 0.3636 | $M_1$ | 103.3 |
| $g$ | 0.6479 | $M_2$ | 193.2 |
| $g_s$ | 1.0844 | $M_3$ | 571.7 |
| $Y_\tau$ | 0.1034 | $A_\tau$ | −445.2 |
| $Y_t$ | 0.8678 | $A_t$ | −565.1 |
| $Y_b$ | 0.1354 | $A_b$ | −943.4 |
| $\mu$ | 401.6 | $\tan\beta$ | 10.0 |

- Higgsino scenario:
  $\mu = 130\ \mathrm{GeV}$, $M_2 = 400\ \mathrm{GeV}$, $M_1 = M_2/2$,
  other parameters from SPS1a′

# Gaugino scenario

- Neutralino masses [GeV]

$m_{\tilde{\chi}^0_1} = \ 100.68 \ -3.04 \ \ -0.07 \ +0.00$
$m_{\tilde{\chi}^0_2} = \ 181.32 \ +2.83 \ \ -0.19 \ +0.00$
$m_{\tilde{\chi}^0_3} = \ 407.35 \ -1.65 \ \ +0.16 \ +0.30$
$m_{\tilde{\chi}^0_4} = \ 421.85 \ -3.14 \ \ +0.05 \ +0.83$

- Chargino masses [GeV]

$m_{\tilde{\chi}^+_1} = \ 180.94 \ +2.98 \ \ -0.19 \ +0.00$
$m_{\tilde{\chi}^+_2} = \ 422.27 \ -2.02 \ \ -0.16 \ +0.59$

- Gluino mass [GeV]

$m_{\tilde{g}} = \ 572.33 \ +39.14 \ \ +11.06 \ +0.00$

Austrian Academy of Sciences

# Gaugino scenario

- Neutralino masses [GeV]

  $m_{\tilde{\chi}^0_1} = 100.68 - 3.04 \; - 0.07 + 0.00$

  $m_{\tilde{\chi}^0_2} = 181.32 + 2.83 \; - 0.19 + 0.00$

  $m_{\tilde{\chi}^0_3} = 407.35 - 1.65 \; + 0.16 + 0.30$

  $m_{\tilde{\chi}^0_4} = 421.85 - 3.14 \; + 0.05 + 0.83$

- Chargino masses [GeV]

  $m_{\tilde{\chi}^+_1} = 180.94 + 2.98 \; - 0.19 + 0.00$

  $m_{\tilde{\chi}^+_2} = 422.27 - 2.02 \; - 0.16 + 0.59$

- Gluino mass [GeV]

  $m_{\tilde{g}} = 572.33 + 39.14 \; + 11.06 + 0.00$

Austrian Academy of Sciences

# Gaugino scenario

- Neutralino masses $[\mathrm{GeV}]$

  $m_{\tilde{\chi}_1^0} = 100.68 - 3.04 \quad -0.07 + 0.00$
  $m_{\tilde{\chi}_2^0} = 181.32 + 2.83 \quad -0.19 + 0.00$
  $m_{\tilde{\chi}_3^0} = 407.35 - 1.65 \quad +0.16 + 0.30$
  $m_{\tilde{\chi}_4^0} = 421.85 - 3.14 \quad +0.05 + 0.83$

- Chargino masses $[\mathrm{GeV}]$

  $m_{\tilde{\chi}_1^+} = 180.94 + 2.98 \quad -0.19 + 0.00$
  $m_{\tilde{\chi}_2^+} = 422.27 - 2.02 \quad -0.16 + 0.59$

- Gluino mass $[\mathrm{GeV}]$

  $m_{\tilde{g}} = 572.33 + 39.14 \quad +11.06 + 0.00$

Austrian Academy of Sciences

# Gaugino scenario

- Neutralino masses [GeV]

$$m_{\tilde{\chi}^0_1} = 100.68 \; -3.04 \; -0.07 \; +0.00$$
$$m_{\tilde{\chi}^0_2} = 181.32 \; +2.83 \; -0.19 \; +0.00$$
$$m_{\tilde{\chi}^0_3} = 407.35 \; -1.65 \; +0.16 \; +0.30$$
$$m_{\tilde{\chi}^0_4} = 421.85 \; -3.14 \; +0.05 \; +0.83$$

- Chargino masses [GeV]

$$m_{\tilde{\chi}^+_1} = 180.94 \; +2.98 \; -0.19 \; +0.00$$
$$m_{\tilde{\chi}^+_2} = 422.27 \; -2.02 \; -0.16 \; +0.59$$

- Gluino mass [GeV]

$$m_{\tilde{g}} = 572.33 \; +39.14 \; +11.06 \; +0.00$$

Austrian Academy of Sciences

# Gaugino scenario

- Neutralino masses $[\mathrm{GeV}]$
  $m_{\tilde\chi^0_1} = 100.68 - 3.04 \;\; -0.07 + 0.00$
  $m_{\tilde\chi^0_2} = 181.32 + 2.83 \;\; -0.19 + 0.00$
  $m_{\tilde\chi^0_3} = 407.35 - 1.65 \;\; +0.16 + 0.30$
  $m_{\tilde\chi^0_4} = 421.85 - 3.14 \;\; +0.05 + 0.83$
- Chargino masses $[\mathrm{GeV}]$
  $m_{\tilde\chi^+_1} = 180.94 + 2.98 \;\; -0.19 + 0.00$
  $m_{\tilde\chi^+_2} = 422.27 - 2.02 \;\; -0.16 + 0.59$
- Gluino mass $[\mathrm{GeV}]$
  $m_{\tilde g} = 572.33 + 39.14 \;\; +11.06 + 0.00$

ÖAW Austrian Academy of Sciences

# Higgsino scenario

- Neutralino masses [GeV]

  $m_{\tilde{\chi}^0_1} = 106.54 \; -8.69 \; -2.19 \; +2.11$
  $m_{\tilde{\chi}^0_2} = 137.22 \; +1.92 \; +1.52 \; -1.36$
  $m_{\tilde{\chi}^0_3} = 212.81 \; -5.66 \; -1.85 \; +1.88$
  $m_{\tilde{\chi}^0_4} = 417.87 \; -3.97 \; -1.90 \; +1.94$

- Chargino masses [GeV]

  $m_{\tilde{\chi}^+_1} = 121.54 \; +8.63 \; -1.42 \; +1.36$
  $m_{\tilde{\chi}^+_2} = 417.75 \; -9.16 \; -1.72 \; +1.74$

- Gluino mass [GeV]

  $m_{\tilde{g}} = 572.33 \; +39.11 \; +11.08 \; -0.15$

Austrian Academy of Sciences

# Higgsino scenario

- Neutralino masses [GeV]

  $m_{\tilde{\chi}^0_1} = 106.54 \;\; -8.69 \;\; -2.19 \;\; +2.11$

  $m_{\tilde{\chi}^0_2} = 137.22 \;\; +1.92 \;\; +1.52 \;\; -1.36$

  $m_{\tilde{\chi}^0_3} = 212.81 \;\; -5.66 \;\; -1.85 \;\; +1.88$

  $m_{\tilde{\chi}^0_4} = 417.87 \;\; -3.97 \;\; -1.90 \;\; +1.94$

- Chargino masses [GeV]

  $m_{\tilde{\chi}^+_1} = 121.54 \;\; +8.63 \;\; -1.42 \;\; +1.36$

  $m_{\tilde{\chi}^+_2} = 417.75 \;\; -9.16 \;\; -1.72 \;\; +1.74$

- Gluino mass [GeV]

  $m_{\tilde{g}} = 572.33 \;\; +39.11 \;\; +11.08 \;\; -0.15$

# Higgsino scenario

- Neutralino masses [GeV]

  $m_{\tilde{\chi}^0_1} = 106.54 - 8.69 \quad -2.19 \quad +2.11$

  $m_{\tilde{\chi}^0_2} = 137.22 + 1.92 \quad +1.52 \; -1.36$

  $m_{\tilde{\chi}^0_3} = 212.81 - 5.66 \quad -1.85 \; +1.88$

  $m_{\tilde{\chi}^0_4} = 417.87 - 3.97 \quad -1.90 \; +1.94$

- Chargino masses [GeV]

  $m_{\tilde{\chi}^+_1} = 121.54 + 8.63 \quad -1.42 \; +1.36$

  $m_{\tilde{\chi}^+_2} = 417.75 - 9.16 \quad -1.72 \; +1.74$

- Gluino mass [GeV]

  $m_{\tilde{g}} = 572.33 + 39.11 \quad +11.08 \; -0.15$

Austrian Academy of Sciences

# Higgsino scenario

- Neutralino masses [GeV]

$$m_{\tilde{\chi}^0_1} = 106.54 \; -8.69 \; -2.19 \; +2.11$$
$$m_{\tilde{\chi}^0_2} = 137.22 \; +1.92 \; +1.52 \; -1.36$$
$$m_{\tilde{\chi}^0_3} = 212.81 \; -5.66 \; -1.85 \; +1.88$$
$$m_{\tilde{\chi}^0_4} = 417.87 \; -3.97 \; -1.90 \; +1.94$$

- Chargino masses [GeV]

$$m_{\tilde{\chi}^+_1} = 121.54 \; +8.63 \; -1.42 \; +1.36$$
$$m_{\tilde{\chi}^+_2} = 417.75 \; -9.16 \; -1.72 \; +1.74$$

- Gluino mass [GeV]

$$m_{\tilde{g}} = 572.33 \; +39.11 \; +11.08 \; -0.15$$

Austrian Academy of Sciences

# Higgsino scenario

- Neutralino masses [GeV]

$$m_{\tilde{\chi}^0_1} = 106.54 - 8.69 - 2.19 + 2.11$$
$$m_{\tilde{\chi}^0_2} = 137.22 + 1.92 + 1.52 - 1.36$$
$$m_{\tilde{\chi}^0_3} = 212.81 - 5.66 - 1.85 + 1.88$$
$$m_{\tilde{\chi}^0_4} = 417.87 - 3.97 - 1.90 + 1.94$$

- Chargino masses [GeV]

$$m_{\tilde{\chi}^+_1} = 121.54 + 8.63 - 1.42 + 1.36$$
$$m_{\tilde{\chi}^+_2} = 417.75 - 9.16 - 1.72 + 1.74$$

- Gluino mass [GeV]

$$m_{\tilde{g}} = 572.33 + 39.11 + 11.08 - 0.15$$

Austrian Academy of Sciences

# Description of Polxino

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct mssm

Austrian Academy of Sciences

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct `mssm`

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct mssm

Austrian Academy of Sciences

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct mssm

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct mssm

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct mssm

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct mssm

# is a C-program

- Needs the Two-loop Self-energy Integral Library TSIL
  `http://zippy.physics.nui.edu/TSIL`
- Central data C-struct is the `MSSM_DATA` mssm
- provide interface to
  SUSY Les Houches Accord 1.0 [Skands et.al. '03]
  e.g. SPheno produces a .spc file
- loadSPC("SPheno.spc",&mssm); loads the file into the struct
  As an example, the file SPheno.spc contains the data for the SPS1a' point
- If there is need, re-adjust parameters manually
  e.g. for our higgsino scenario, we write
  `mssm.mue = 130.L; mssm.M1 = 200.L; mssm.M2 = 400.L`
- `mssm_digest(&mssm)`
  re-diagonalizes mixing systems
  (some routines taken over from LoopTools) [Hahn, Perez '98]
- `mssm_print(mssm);`
  prints out all the information stored in the struct `mssm`

## Basic parameters of the struct `MSSM_DATA`

| **MSSM_DATA** member | description |
|---|---|
| **gs, g1, g2** | gauge coupling constants |
| **mue** | $\mu$-parameter |
| **TB** | $\tan\beta$ |
| **VEV** | $\sqrt{v_1^2+v_2^2}$, $v_i$ are the Higgs VEVs |
| **MA02** | squared mass of $A_0$ |
| **MQU[3], MQD[3], MLE[3]** | quark and lepton-masses |
| **M1, M2, M3** | gaugino masses |
| **MSL[3], MSE[3], MSQ[3], MSU[3], MSD[3]** | bilinear soft Susy breaking parameters |
| **Af[4][3]** | trilinear soft Susy breaking parameters |

Austrian Academy of Sciences

# The subroutines
## using generic functions

- generic one- and two-loop functions were extracted from gluinopole.c [S.P.Martin '05]

**getneupole**(⟨int⟩ i, ⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1, ⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);
**getchpole**(⟨int⟩ i, ⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1, ⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);
**getgluinopole**(⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1, ⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);

calculate the one- and two-loop pole masses of chargino (neutralino) *i* or the gluino, stores the one- and two-loop results in the variables **Mpole1** resp. **Mpole2**.

The tree-level and one-loop widths are stored in **Gamma1** and **Gamma2**

# The subroutines

## using generic functions

- generic one- and two-loop functions were extracted from gluinopole.c [S.P.Martin '05]

**getneupole(**⟨int⟩ i, ⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1,
⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);
**getchpole(**⟨int⟩ i, ⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1,
⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);
**getgluinopole(**⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1,
⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);

calculate the one- and two-loop pole masses of chargino (neutralino) $i$ or the gluino, stores the one- and two-loop results in the variables **Mpole1** resp. **Mpole2**.

The tree-level and one-loop widths are stored in **Gamma1** and **Gamma2**

# The subroutines

## using generic functions

- generic one- and two-loop functions
  were extracted from gluinopole.c [S.P.Martin '05]

**getneupole(**⟨int⟩ i, ⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1,
⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);
**getchpole(**⟨int⟩ i, ⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1,
⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);
**getgluinopole(**⟨MSSM_DATA⟩, ⟨POL_REAL*⟩ Mpole1, ⟨POL_REAL*⟩ Gamma1,
⟨POL_REAL*⟩ Mpole2, ⟨POL_REAL*⟩ Gamma2);

calculate the one- and two-loop pole masses of chargino (neutralino) *i* or the gluino, stores the one- and two-loop results in the variables **Mpole1** resp. **Mpole2**.

The tree-level and one-loop widths are stored in **Gamma1** and **Gamma2**

# Download

The pole mass calculator for $\chi$-inos

- can be downloaded from `http://wwwhephy.oeaw.ac.at/tools/polxino`

# Download

The pole mass calculator for $\chi$-inos

- can be downloaded from
  http://wwwhephy.oeaw.ac.at/tools/polxino

# Download

Polxino

- can be downloaded from http://wwwhephy.oeaw.ac.at/tools/polxino

# Download

Polxino

- can be downloaded from
  http://wwwhephy.oeaw.ac.at/tools/polxino

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with Polxino (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with Polxino (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

ÖAW Austrian Academy of Sciences

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with Polxino (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with Polxino (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with `Polxino` (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

ÖAW Austrian Academy of Sciences

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with `Polxino` (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with `Polxino` (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

ÖAW Austrian Academy of Sciences

# Conclusion

Motivation:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with Polxino (using generic formulae)

Results:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

# Conclusion

`Motivation`:

- high precision experiments demand pole mass calculations to two-loop order [LHC/LC Study Group]
- combining existing software packages it was possible to calculate the leading NNLO corrections (SQCD + Yukawas)
- auto-generated code: FeynArts→FeynCalc→Tarcer, Fortran, TSIL, LoopTools
- agrees with `Polxino` (using generic formulae)

`Results`:

- the remaining renormalization scale dependence is weakened
- For gaugino-like $\tilde{\chi}$'s two-loop SQCD is mandatory
- For higgsino-like $\tilde{\chi}$'s two-loop Yukawa corr. have to be included too

ÖAW
Austrian Academy of Sciences

# Outlook

`Full two-loop calculation`:

- best of two worlds approach
  combine the strength of
  - existing Mathematica tools for diagram generation
  - well-tested generic results

  to a package capable of doing the full two-loop calculation ($2 \cdot 2500$ diagrams)

Austrian Academy of Sciences

# Outlook

`Full two-loop calculation`:

- best of two worlds approach
  combine the strength of
    - existing Mathematica tools for diagram generation
    - well-tested generic results

  to a package capable of doing the full two-loop calculation ($2 \cdot 2500$ diagrams)

# Thank you for our attention!

Here the address for the tools again:

- `http://wwwhephy.oeaw.ac.at/tools/polxino`