

# Including $R$ -parity violation in the numerical computation of the spectrum of the minimal supersymmetric standard model: SOFTSUSY3.3.3

B.C. Allanach

*DAMTP, CMS, University of Cambridge, Wilberforce road, Cambridge, CB3 0WA, United Kingdom*

M.A. Bernhard

*Physics Institute, University of Bonn, Nussallee 12, D-53115 Bonn, Germany*

---

## Abstract

Current publicly available computer programs calculate the spectrum and couplings of the minimal supersymmetric standard model under the assumption of  $R$ -parity conservation. Here, we describe an extension to the SOFTSUSY program which includes  $R$ -parity violating effects. The user provides a theoretical boundary condition upon the high-scale supersymmetry breaking  $R$ -parity violating couplings. Successful radiative electroweak symmetry breaking, electroweak and CKM matrix data are used as weak-scale boundary conditions. The renormalisation group equations are solved numerically between the weak scale and a high energy scale using a nested iterative algorithm. This paper serves as a manual to the  $R$ -parity violating mode of the program, detailing the approximations and conventions used.

*Keywords:* sparticle, MSSM

*PACS:* 12.60.Jv

*PACS:* 14.80.Ly

---

## 1. Program Summary

*Program title:* SOFTSUSY

*Program obtainable from:* <http://projects.hepforge.org/softsusy/>

*Distribution format:* tar.gz

*Programming language:* C++, fortran

*Computer:* Personal computer

*Operating system:* Tested on Linux 4.x

*Word size:* 32 bits

*External routines:* None

*Typical running time:* a few seconds per parameter point.

*Nature of problem:* Calculating supersymmetric particle spectrum and mixing parameters in the  $R$ -parity violating minimal supersymmetric standard model. The solution to the renormalisation group equations must be consistent with a high-scale boundary condition on supersymmetry breaking parameters and  $R_p$  parameters, as well as a weak-scale boundary condition on gauge couplings, Yukawa couplings and the Higgs potential parameters.

*Solution method:* Nested iterative algorithm.

*Restrictions:* SOFTSUSY will provide a solution only in the perturbative régime and it assumes that all couplings of the MSSM are real (i.e.  $CP$ -conserving). The iterative SOFTSUSY algorithm will not converge if parameters are too close to a boundary of successful electroweak symmetry breaking, but a warning flag will alert the user to this fact.

## 2. Introduction

Spectrum generators are a widely used tool in particle physics beyond the Standard Model (SM), especially in the case of Supersymmetric (SUSY) models. Spectrum generators can be used in theoretical studies of a SUSY breaking scheme, for example in studies of fine-tuning. Phenomenological investigations of new patterns of SUSY breaking require a calculation of the spectrum. Often, the resulting SUSY spectrum is used to calculate the prospects of high energy experiments such as the Large Hadron Collider (LHC) discovering and measuring SUSY particles, assuming some SUSY breaking scheme [1]. In order to run a realistic collider and detector simulation of a new physics signal, a consistent model is needed as input. Such simulations are required in order to set search and measurement strategies [2, 3] as well as to estimate SUSY backgrounds to some measurement. In the event of discovery of some SUSY signals in LHC data, attention will turn to the question of which patterns of SUSY breaking are consistent with data. In such tests, SUSY spectrum generation would be an essential step. SUSY studies often perform parameter scans, resulting in a large number of generated spectra. There is thus a need for accurate and quick computation of the supersymmetric spectrum as a first step in such studies. There exist several publicly available spectrum generators for the  $R$ -parity ( $R_p$ ) conserving minimal supersymmetric extension of the Standard Model (MSSM): ISASUGRA [4], SOFTSUSY [5], SUSPECT [6] and SPHENO [7]. Spectrum information is typically transferred to decay packages and event generators via a file in the SUSY Les Houches Accord format [8, 9].

The most general renormalisable superpotential of the MSSM contains  $R$ -Parity violating ( $\mathcal{R}_p$ ) couplings, violating baryon and lepton number [11]. A symmetry can be imposed upon the model in order to maintain stability of the proton, for example baryon triality [12] or proton hexality [13]. It has been shown that  $\mathcal{R}_p$  models may have interesting features, such as the generation of non-zero neutrino masses without the addition of right-handed neutrino fields [14], and the gravitino as a viable dark matter candidate [15]. The violation of baryon or lepton number implied by  $\mathcal{R}_p$  leads to additional possibilities for SUSY detection, since such quantum numbers are conserved in the perturbative SM. There are important implications for direct collider searches, since one can lose the classic large missing transverse energy “smoking-gun” signature of SUSY: All of these features make  $\mathcal{R}_p$  worthy of study. There is thus a strong motivation to extend the  $R_p$ -conserving spectrum generating public computer programs to include  $\mathcal{R}_p$  effects. Here, we describe such an extension which has been applied to SOFTSUSY. The latest version of SOFTSUSY including  $\mathcal{R}_p$  effects can be downloaded from address

<http://projects.hepforge.org/softsusy/>

Installation instructions and more detailed technical documentation of the code may also be found there.

The  $R_p$  conserving aspects of SOFTSUSY are already explained in detail in ref. [5], and so they shall not be repeated here or throughout this manual, which will concentrate solely on the  $\mathcal{R}_p$  aspects of the calculation. Adding  $\mathcal{R}_p$  couplings roughly doubles the (already large) number of couplings of the MSSM. The calculation is thus more complicated and so it takes considerably longer than the  $R_p$  case (roughly a factor of three for identical precision). However, this still means that a single point in parameter space can be calculated in a couple of seconds on a modern personal computer. The added complication of  $\mathcal{R}_p$  means that some features of the  $\mathcal{R}_p$  version of SOFTSUSY are less accurate than the  $R_p$ -conserving case, using only one-loop RGEs to evolve the couplings and masses of MSSM fields, as opposed to two-loop RGEs in the  $R_p$  case. Therefore, taking the  $\mathcal{R}_p$  calculation in the limit of small  $\mathcal{R}_p$  couplings, the numerical values of SOFTSUSY outputs will not *exactly* agree with the  $R_p$ -conserving version of SOFTSUSY. We stress though, that if the user does not desire to include  $\mathcal{R}_p$  couplings, the program automatically uses the  $R_p$  calculation with the associated speed and accuracy. Where the accuracy of SOFTSUSY in the  $\mathcal{R}_p$ -mode differs from the  $R_p$ -conserving mode calculation, we shall make a note.

We proceed with a definition of the SOFTSUSY convention for the  $\mathcal{R}_p$  parameters and mixings in section 3. Next, in section 4, we discuss the calculation, making a note of parts which differ in accuracy to the SOFTSUSY  $R_p$  calculation. Installation instructions can be found on the SOFTSUSY web-site, but instructions to run the program can be found in Appendix Appendix A. The output from a SOFTSUSY sample run is displayed and discussed in Appendix Appendix C, whereas a sample main program is shown and explained in Appendix Appendix B. Some more technical information on the structure of the program can be found in Appendix Appendix D. It is expected that the information in Appendix Appendix D will only be of use to users who wish to ‘hack’ SOFTSUSY in some fashion.

### 3. MSSM $\mathcal{R}_p$ Parameters

In this section, we introduce the  $\mathcal{R}_p$  MSSM parameters in the SOFTSUSY conventions. The translations to the actual variable names that are being used in the program code are shown explicitly in appendix Appendix D. The  $\mathcal{R}_p$  SOFTSUSY calculation follows ref. [16] and so the notation and conventions are similar.

#### 3.1. Supersymmetric parameters

The chiral superfield particle content of the MSSM has the following  $SU(3)_c \times SU(2)_L \times U(1)_Y$  quantum numbers:

$$\begin{aligned} L &: (1, 2, -\frac{1}{2}), & \bar{E} &: (1, 1, 1), & Q &: (3, 2, \frac{1}{6}), & \bar{U} &: (\bar{3}, 1, -\frac{2}{3}), \\ \bar{D} &: (\bar{3}, 1, \frac{1}{3}), & H_1 &: (1, 2, -\frac{1}{2}), & H_2 &: (1, 2, \frac{1}{2}). \end{aligned} \quad (1)$$

$L$ ,  $Q$ ,  $H_1$ , and  $H_2$  are the left handed doublet lepton and quark superfields and the two Higgs doublets.  $\bar{E}$ ,  $\bar{U}$ , and  $\bar{D}$  are the lepton, up-type quark and down-type quark right-handed superfield singlets, respectively. Note that the lepton doublet superfields  $L_i^a$  and the Higgs doublet superfield coupling to the down-type quarks,  $H_1$ , have the same SM gauge quantum numbers. The  $\mathcal{R}_p$  part of the renormalisable MSSM superpotential is written, in the interaction eigenbasis,

$$W_{\mathcal{R}_p} = \epsilon_{ab} \left[ \frac{1}{2} \lambda_{ijk} L_i^a L_j^b \bar{E}_k + \lambda'_{ijk} L_i^a Q_j^{xb} \bar{D}_{kx} - \kappa^i L_i^a H_2^b \right] + \frac{1}{2} \epsilon_{xyz} \lambda''_{ijk} \bar{U}_i^x \bar{D}_j^y \bar{D}_k^z. \quad (2)$$

Here, we denote an  $SU(3)$  colour index of the fundamental representation by  $\{x, y, z\} \in \{1, 2, 3\}$ . The  $SU(2)_L$  fundamental representation indices are denoted by  $\{a, b, c\} \in \{1, 2\}$  and the generation indices by  $\{i, j, k\} \in \{1, 2, 3\}$ .  $\epsilon_{xyz} = \epsilon^{xyz}$  and  $\epsilon_{ab} = \epsilon^{ab}$  are totally antisymmetric tensors, with  $\epsilon_{123} = 1$  and  $\epsilon_{12} = 1$ , respectively. Currently, only real couplings in the superpotential and Lagrangian are included.

#### 3.2. $\mathcal{R}_p$ SUSY breaking parameters

We now detail the notation of the soft  $\mathcal{R}_p$  SUSY breaking parameters. The trilinear  $\mathcal{R}_p$  scalar interaction potential is

$$V_{3,\mathcal{R}_p} = \epsilon_{ab} \left[ \frac{1}{2} h_{ijk} \tilde{L}_i^a \tilde{L}_j^b \tilde{e}_k + h'_{ijk} \tilde{L}_i^a \tilde{Q}_j^{bx} \tilde{d}_{kx} + \text{H.c.} \right] + \left[ \frac{1}{2} \epsilon_{xyz} h''_{ijk} \tilde{u}_i^x \tilde{d}_j^y \tilde{d}_k^z + \text{H.c.} \right] \quad (3)$$

where fields with a tilde are the scalar components of the superfield with the identical capital letter. The electric charges of  $\tilde{u}$ ,  $\tilde{d}$ , and  $\tilde{e}$  are  $-\frac{2}{3}$ ,  $\frac{1}{3}$ , and 1, respectively. ‘‘H.c.’’ denotes the Hermitian conjugate of the preceding terms.

The bilinear  $\mathcal{R}_p$  scalar interaction potential is given by

$$V_{2,\mathcal{R}_p} = -\epsilon_{ab} D_i \tilde{L}_i^a H_2^b + m_{\tilde{L}_i H_1}^2 \tilde{L}_{ia}^\dagger H_1^a + \text{H.c.} \quad (4)$$

#### 3.3. Tree-level masses

The mixing of MSSM particles can change in the case that lepton number is violated by the  $\mathcal{R}_p$  interactions. Two cases of lepton number violating mixings are implemented in SOFTSUSY: neutrino-neutralino mixing and chargino-lepton mixing. We neglect sneutrino-anti-sneutrino mixing, because this has been shown to have negligible phenomenological consequences once experimental bounds have been applied [17].

In the presence of lepton number violating  $\mathcal{R}_p$  interactions, the neutrinos mix with the neutralinos. At tree level, this results in one massive neutrino, two massless neutrinos and four massive neutralinos.  $\mathcal{R}_p$ -loop corrections to the neutral fermion mass matrix (currently neglected by SOFTSUSY) can result in all neutrinos acquiring masses and the emergence of a PMNS mixing matrix in lepton charged current interactions. The  $(7 \times 7)$  neutrino-neutralino mass matrix for the three generations of neutrinos is given in [16] and reads

$$\mathcal{L} = -\frac{1}{2} (v_i, -i\tilde{\mathcal{B}}, -i\tilde{\mathcal{W}}^{(3)}, \tilde{h}_1^0, \tilde{h}_2^0) \mathcal{M}_N \begin{pmatrix} \nu_j \\ -i\tilde{\mathcal{B}} \\ -i\tilde{\mathcal{W}}^{(3)} \\ \tilde{h}_1^0 \\ \tilde{h}_2^0 \end{pmatrix}, \quad (5)$$

where

$$\mathcal{M}_N = \begin{pmatrix} 0_{ij} & -\frac{g'}{2}v_i & \frac{g_2}{2}v_i & 0 & -\kappa_i \\ -\frac{g'}{2}v_j & M_1 & 0 & -\frac{g'}{2}v_d & \frac{g'}{2}v_u \\ \frac{g_2}{2}v_j & 0 & M_2 & \frac{g_2}{2}v_d & -\frac{g_2}{2}v_u \\ 0 & -\frac{g'}{2}v_d & \frac{g_2}{2}v_d & 0 & -\mu \\ -\kappa_i & \frac{g'}{2}v_u & -\frac{g_2}{2}v_u & -\mu & 0 \end{pmatrix}, \quad (6)$$

where  $\kappa_i$  are the bilinear mixing parameters in Eq. (2),  $v_i$  are the sneutrino vacuum expectation values (VEVs) and the  $M_1, M_2$  are the gaugino masses of hypercharge and weak isospin, respectively. The matrix (6) has five non-zero eigenvalues, i.e. four neutralinos and one neutrino. We denote the mass eigenstates which are obtained upon diagonalisation of  $\mathcal{M}$  (in ascending order of mass):  $\nu_{i=1,2,3}, \tilde{\chi}_{1,2,3,4}^0$ , with masses along the diagonal of the matrix

$$\mathcal{M}_N^{diag} = O^T \mathcal{M}_N O, \quad (7)$$

where  $O$  is a member of  $O(7)$ . A simple multiplication of rows of  $O$  by factors of  $i$  can absorb any minus signs in  $\mathcal{M}_N^{diag}$ .

In addition, charged leptons mix with the charginos. The Lagrangian contains the  $(5 \times 5)$  lepton-chargino mass matrix

$$\mathcal{L} = -(-i\tilde{W}^-, \tilde{h}_2^-, e_{L_j}^-) \mathcal{M}_C \begin{pmatrix} -i\tilde{W}^+ \\ \tilde{h}_2^+ \\ e_{R_k}^+ \end{pmatrix} + \text{H.c.} \quad (8)$$

The mass eigenstates  $\ell = (e, \mu, \tau), \tilde{\chi}_{1,2}^\pm$  are given upon the diagonalisation of the matrix

$$\mathcal{M}_C = \begin{pmatrix} M_2 & \frac{g_2}{\sqrt{2}}v_u & 0_j \\ \frac{g_2}{\sqrt{2}}v_d & \mu & -(Y_E)_{ij}v_i \frac{1}{\sqrt{2}} \\ \frac{g_2}{\sqrt{2}}v_i & \kappa_i & \frac{1}{\sqrt{2}} \left( (Y_E)_{ij}v_d + \lambda_{kij}v_k \right) \end{pmatrix}, \quad (9)$$

Here,  $Y_E$  is the lepton Yukawa matrix from the  $R_p$  superpotential in ref. [5]. We define the diagonalised mass matrix

$$\mathcal{M}_C^{diag} = U \mathcal{M}_C V^T, \quad (10)$$

$U$  and  $V$  being orthogonal 5 by 5 matrices.

#### 4. Calculation Algorithm

In broad terms, the algorithm for the calculation of the  $R_p$  MSSM spectrum follows that of the  $R_p$ -conserving case, although some of the individual steps differ in the  $R_p$  case. It is performed via the iterative algorithm depicted in Fig. 1. An initial estimate of gauge couplings and up quark masses at  $M_Z$  are obtained as in the  $R_p$  version of SOFTSUSY in ref. [5]. However, the charged lepton and down-quark Yukawa couplings receive  $R_p$  corrections, and are detailed in section 4.1. The MSSM parameters are then run to the scale  $M_{SUSY}$ , defined by default to be

$$\sqrt{m_{\tilde{t}_1}(M_{SUSY})m_{\tilde{t}_2}(M_{SUSY})}, \quad (11)$$

where the scale dependence of the electroweak breaking conditions is small [18]. The user may make other choices for the value of  $M_{SUSY}$ , if they desire. Electroweak symmetry breaking (EWSB) conditions are then imposed, as

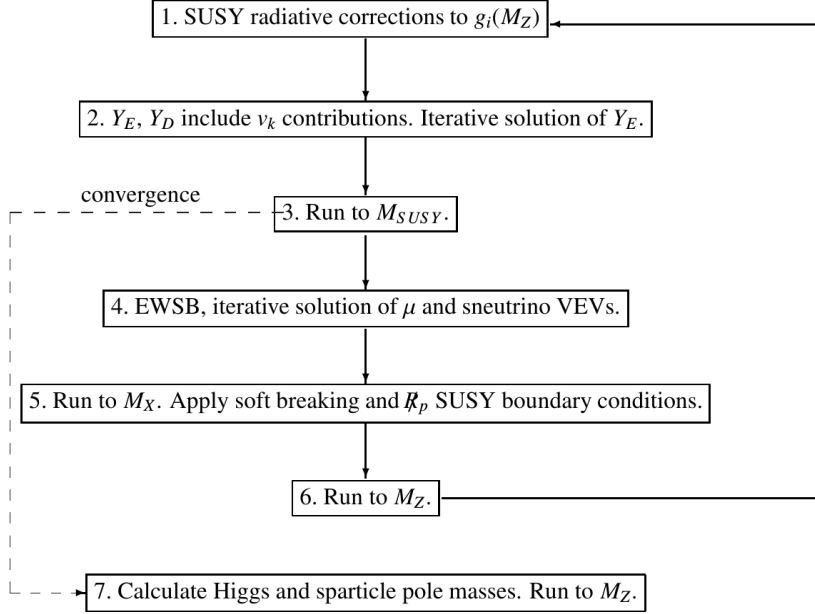


Figure 1: Iterative algorithm used to calculate the  $R_p$  MSSM spectrum. The initial step is the uppermost one.  $M_{SUSY}$  is the scale at which the EWSB conditions are imposed, as discussed in the text.  $M_X$  is the scale at which the high energy SUSY breaking boundary conditions are imposed.

described in section 4.3, taking into account sneutrino VEVs and other lepton-number violating effects. The MSSM parameters are then run up to some high energy scale  $M_X$ , where the soft SUSY breaking terms are fixed by a user-specified boundary condition.  $M_X$  may be the electroweak gauge unification scale, some scale pre-specified by the user or indeed  $M_{SUSY}$ . The running of the MSSM couplings is described in section 4.2. The SUSY  $R_p$  couplings  $\lambda_{ijk}$ ,  $\lambda'_{ijk}$ ,  $\lambda''_{ijk}$ ,  $\kappa_i$  are then fixed at this scale  $M_X$ . The model is then run down to  $M_Z$ , when the iteration is performed again by returning to step 1 in Fig. 1. Iteration proceeds until, at step 3, all parameters evaluated at  $M_{SUSY}$  are identical to within a fractional accuracy of TOLERANCE to the previous iteration's (at step 3). TOLERANCE < 1 is a numerical parameter set by the user, with default value  $10^{-3}$ . Once this has been achieved, the algorithm proceeds to step 7, where the pole masses of sparticles are calculated as in section 4.4.

#### 4.1. Gauge and yukawa couplings

In order to calculate the Yukawa couplings of the down quarks, contributions to the mass matrix from sneutrino VEVs are taken into account:

$$(Y_D)_{ij} = \frac{1}{v_d} \left[ \sqrt{2}(m_D)_{ij} - \lambda'_{kij} \cdot v_k \right], \quad (12)$$

where all parameters are evaluated at  $M_Z$  and are in the  $\overline{DR}$  scheme in the MSSM. The down-quark mass matrix in the weak eigenbasis,  $(m_D)_{ij}$ , is obtained as in the  $R_p$  SOFTSUSY version.  $v_d$  is the VEV of  $H_1$ , as obtained below.

The chargino-lepton mixing in Eq. (9), complicates the matching of  $(Y_E)_{ij}$  to the charged lepton masses. We employ an iterative procedure in order to calculate which  $(Y_E)_{ij}$  predict the empirical input values of charged lepton masses.

1. Initially, we set  $(Y_E)_{ij}$  as in the  $R_p$ -conserving limit, ignoring any charged lepton-chargino mixing or sneutrino VEVs. Thus, the empirical MSSM  $\overline{DR}$  values of charged lepton masses evaluated at  $M_Z$  are written in the matrix  $(m_E^{exp})_{ij}$  as diagonal values. Then,  $(Y_E)_{il} = (m_E^{exp})_{il} \sqrt{2}/v_d$  (no sum on  $l$ ).
2. The resulting matrix  $(Y_E)_{ij}$  is then substituted into Eq. (9) to form the charged lepton-chargino mass matrix, obtaining the  $5 \times 5$   $U$  and  $V$  transformation matrices that diagonalise it via Eq. (10).

3. We denote the 3 by 3 lower right-hand side blocks of  $U$  and  $V$  by  $\tilde{U}$  and  $\tilde{V}$ , respectively. The Yukawa matrix  $Y_E$  is then set to be

$$(Y_E)_{ij} = \frac{1}{v_d} \left[ \sqrt{2} \tilde{U}_{ik}^T (m_E^{exp})_{kl} \tilde{V}_{li} - \lambda_{kij} \cdot v_k \right]. \quad (13)$$

*Physical* lepton mixing is implemented in an extension of this work [19].

4. This result for  $Y_E$  is then inserted back into step 2, leading to a better approximation of  $U$  and  $V$ . Steps 2 to 4 are iterated until successive iterations predict identical diagonal entries of  $Y_E$  within a fractional accuracy of  $10^{-4} \times \text{TOLERANCE}$ .

#### 4.2. Running of MSSM couplings

For the  $R_p$ -conserving parameters, the renormalisation group evolution (RGE) employs two-loop MSSM  $\beta$  functions for the supersymmetric parameters [20], including  $\tan\beta$  and the Higgs VEV parameter  $v$ . Gaugino masses and  $R_p$ -conserving SUSY breaking Higgs parameters are also run to two-loop order in the  $R_p$ -conserving parameters. The other  $R_p$ -conserving SUSY breaking parameters (sfermion mass matrices and some tri-linear couplings) may be set to two-loop or one-loop order by the boolean parameter `INCLUDE_2_LOOP_SCALAR_CORRECTIONS` in the main program. The RGE includes full family dependence and the complete set of 1-loop MSSM  $R_p$   $\beta$  functions in both SUSY-preserving and SUSY-breaking  $R_p$  parameters [16]. The increased number of  $R_p$  couplings and  $\beta$ -functions, as well as other complications, mean that the  $R_p$  mode runs more slowly than the  $R_p$ -conserving mode. In the case of running SOFTSUSY in the  $R_p$  mode the accuracy does not match the extremely high one of the  $R_p$  version, in order to keep the running time down. In either the  $R_p$ -conserving or the  $R_p$  mode, the program can be made to run faster by switching off the two-loop renormalisation of the scalar masses and tri-linear scalar couplings. All  $\beta$  functions are real and include inter-generational quark mixing effects.

#### 4.3. Electroweak symmetry breaking

We now discuss the minimisation the potential of the neutral scalar fields,  $\{h_2^0, h_1^0, \tilde{v}_1, \tilde{v}_2, \tilde{v}_3\}$  at the renormalisation scale  $M_S$ . Following the calculation in Ref. [16], this system of equations is solved using the following definitions [21]

$$\tan\beta \equiv \frac{v_u}{v_d}, \quad v^2 \equiv v_u^2 + v_d^2 + \sum_{i=1}^3 v_i^2 = \frac{4M_W^2}{g_2^2}. \quad (14)$$

The VEVs  $v_d$  and  $v_u$  can be written

$$v_d^2 = \cos^2\beta \left( v^2 - \sum_{i=1}^3 v_i^2 \right), \quad v_u^2 = \sin^2\beta \left( v^2 - \sum_{i=1}^3 v_i^2 \right). \quad (15)$$

We see from Eq. (15) that the presence of sneutrino VEVs does not change the numerical value of  $\tan\beta$ . This convenient formulation was first developed in Ref. [21]. The EWSB condition for the Higgs superpotential mass term  $\mu$  can be written [16] as

$$|\mu|^2 = \frac{1}{\tan^2\beta - 1} \left( \left[ \bar{m}_{H_1}^2 + (m_{\tilde{L},H_1}^2) \frac{v_i}{v_d} + \kappa_i^* \mu \frac{v_i}{v_d} \right] - \left[ \bar{m}_{H_2}^2 + |\kappa_i|^2 - \frac{1}{4}(g^2 + g_2^2)v_i^2 - \tilde{D}_i \frac{v_i}{v_u} \right] \tan^2\beta - \frac{1}{2}M_Z^2 \right), \quad (16)$$

The soft SUSY breaking mass squared term  $m_3^2$  is expressed in terms of  $\mu$ ,  $v_u$ ,  $v_d$ , and  $v_i$ :

$$m_3^2 = \frac{\sin 2\beta}{2} \left\{ \left[ \bar{m}_{H_1}^2 + \bar{m}_{H_2}^2 + 2|\mu|^2 + |\kappa_i|^2 \right] + \left[ (m_{\tilde{L},H_1}^2) + \kappa_i^* \mu \right] \frac{v_i}{v_d} - \tilde{D}_i \frac{v_i}{v_u} \right\}, \quad (17)$$

where we employ the simplifying notation

$$\bar{m}_{H_2}^2 \equiv m_{H_2}^2 + \frac{1}{v_u} \frac{\partial \Delta \mathbf{V}}{\partial v_u}, \quad \bar{m}_{H_1}^2 \equiv m_{H_1}^2 + \frac{1}{v_d} \frac{\partial \Delta \mathbf{V}}{\partial v_d}. \quad (18)$$

The tadpoles  $\frac{\partial \Delta \mathbf{V}}{\partial v_{u,d}}$  currently only contain the  $R_p$ -conserving contributions.  $R_p$  contributions to them are currently neglected, for these are of order  $\frac{\lambda_{ijk}^2}{16\pi^2}, \frac{\lambda'_{ijk}^2}{16\pi^2}$ , so they are small for small  $R_p$  couplings  $\lambda_{ijk}, \lambda'_{ijk}$  as is implied for most  $\{i, j, k\}$  by the experimental bounds [22]. In the  $R_p$ -conserving limit  $\kappa_i, v_i, \widetilde{D}_i, (m_{\widetilde{L}_i H_1}^2) \rightarrow 0$ , Eqs. (16) and (17) tend to the usual  $R_p$ -conserving MSSM Higgs potential minimisation conditions.

The potential minimisation conditions for the sneutrino VEVs may be written as

$$v_j = \sum_i (M_{\widetilde{\nu}}^2)^{-1}_{ji} \left\{ - \left[ (m_{H_1 L_i}^2) + \mu^* \kappa_i \right] v_d + \widetilde{D}_i v_u - \frac{\partial \Delta \mathbf{V}}{\partial v_i} \right\}, \quad (19)$$

where

$$(M_{\widetilde{\nu}}^2)_{ij} \equiv (m_L^2)_{ji} + \kappa_i \kappa_j^* + \frac{1}{2} M_Z^2 \cos 2\beta \delta_{ij} + \frac{(g^2 + g_2^2)}{4} \sin^2 \beta (v^2 - v_u^2 - v_d^2) \delta_{ij}. \quad (20)$$

We now detail the iterative procedure by which SOFTSUSY obtains parameters describing a minimum of the potential with the correct properties, i.e. satisfying Eqs. (16),(17) and (19). All of the running parameters discussed are evaluated at a renormalisation scale  $Q = M_{SUSY}$ .

1. For given value of  $\tan \beta$ , Eq. (14) provides an initial estimate for  $v_u$  and  $v_d$  in the  $R_p$ -conserving limit of  $v_i = 0$ . Eqs. (16) and (17) are also first solved in the  $R_p$  limit. *i.e.*  $v_i = 0, \kappa_i = \widetilde{D}_i = (m_{H_1 \widetilde{L}_i}^2) = 0$ . This provides initial values for  $\mu$  and  $m_3^2$ .
2. The sneutrino VEVs  $v_i$  are now obtained from the left hand side of Eq. (19) by using  $v_u, v_d, \mu$  and  $m_3^2$  as previously derived in the iterative procedure in the right hand side of the equation.
3. The corrected values of  $v_u, v_d$  are then computed including the non-zero sneutrino VEVs  $v_i$  via Eq. (15).
4.  $\mu$  and  $m_3^2$  are then obtained from the left hand sides of Eqs. (16) and (17). The program returns to step 2 and steps 2-4 are iterated until  $\{v_i, \mu, m_3^2\}$  all change by less than a fractional accuracy of  $\text{TOLERANCE} \times 10^{-4}$  on successive iterations.

#### 4.4. MSSM spectrum

Neutralino masses are calculated at tree-level as in Eq. (6). All  $R_p$ -conserving one-loop threshold corrections are then added. The neutrino masses are calculated by diagonalising this mass matrix, and taking the lightest three eigenvalues, whereas neutralino masses are defined to be the largest four eigenvalues. When the chargino masses are calculated by the iterative procedure in section 4.1, one-loop  $R_p$ -conserving corrections to chargino masses are added to the two by two top left-hand corner of Eq. (9). All  $\overline{DR}$  quantities in the mass matrices are taken at the renormalisation scale  $M_{SUSY}$  in the tree-level mass matrices. All other masses are calculated according to the  $R_p$ -conserving SOFTSUSY calculation, i.e. including the one-loop  $R_p$ -conserving threshold contributions.

#### 4.5. Physics applications

The  $R_p$  aspects of prototype versions of SOFTSUSY have already proved useful for various studies, for example: in determining the different possibilities for the lightest supersymmetric particle in the CMSSM framework [23, 24], in defining benchmark points for future comparative collider studies [25], for studying neutrino mass textures [26] and for investigating  $R_p$  mechanisms of neutrinoless double beta decay [27, 28].

### Acknowledgments

We thank S Grab, S Kom and P Slavich for discussions and comments on the manuscript. MB thanks DAMTP, Cambridge, for repeatedly offered, warm hospitality. This work has been partially supported by PPARC and STFC. We thank S Grab, M Hannussek, S Kom, M McCullough and P Slavich for bug-reports.

## Appendix A. Running SOFTSUSY

SOFTSUSY produces an executable called `softpoint.x`. For the calculation of the spectrum of single points in parameter space, we recommend the SUSY Les Houches Accord 2 (SLHA2) [9] input/output option. The user must provide a file (*e.g.* the example file included in the SOFTSUSY distribution `rpvHouchesInput`), that specifies the model dependent input parameters. The program may then be run with

```
./softpoint.x leshouches < rpvHouchesInput
```

For the SLHA2 input option, the output will also be given in SLHA2 format. The example file provided calculates the same point as the CMSSM point we give as an example below. Such output can be used for input into other programs which subscribe to the accord, such as PYTHIA [29] (for simulating sparticle production and decays at colliders), for example. For further details on the necessary format of the input file, see ref. [9]. Note, that SOFTSUSY does not yet support the (optional) setting of the bilinear sneutrino VEVs, these are instead fixed by Eq. 19. It supports the setting of all other SLHA2 input blocks associated with non-complex  $R_p$ . There is an option to have the boundary condition on  $R$ -parity violating parameters to be set at  $M_Z$ , rather than at  $M_{GUT}$ . This is controlled by the boolean global variable `susyRpvBCatMSUSY`, which if set to `true` in the main program, will activate the  $M_{SUSY}$  option ( $M_{GUT}$  being the default). One can instead switch the option on instead within the SLHA2 input file by using a SOFTSUSY specific option in Block SOFTSUSY:

```
Block SOFTSUSY
  8  1.000000e+00 # Switch MSUSY-scale RPV boundary conditions ON
```

Another option has been included in order to interface with programs that expect output only in SLHA 1 format, rather than SLHA 2 format. For this, another Block SOFTSUSY option

```
 10  1.000000e+00 # Try to output object in SLHA 1 format
```

will attempt to produce  $R_p$  output close to the SLHA1 format.

For a quick examination of a single point in CMSSM parameter space, the command

```
./softpoint.x sugra --m0=<value> --m12=<value> --a0=<value> --tanb=<value>
--sgnMu=1 --mgut=unified --lambda 1 2 3 0.1
```

can be utilised. Bracketed entries should be replaced by their numerical values, where all massive parameters ( $m_0$ ,  $M_{1/2}$ ,  $A_0$ ,  $M_{GUT}$ ) should be quoted in GeV. This particular example sets  $\lambda_{123}(M_{GUT}) = 0.1$ , and triggers the  $R_p$  calculation. Instead of (or as well as) `lambda`, one can set `lambdaP`, or `lambdaPP`. `mx` denotes the scale at which the high-energy boundary condition is to be applied. If this is specified as `unified`, as in the  $R_p$  version, the electroweak gauge unification scale  $M_{GUT}$  is used (defined to be the  $\overline{DR}$  scale  $Q$  at which  $g_1(Q) = g_2(Q)$ ). The default output is in SLHA2 format, the conventions of which are explained in Ref. [9].

## Appendix B. Sample Program

In this section we present a sample main program, that illustrates a scan over an  $R_p$  parameter. This main program is included in the `rpvmain.cpp` file with the standard SOFTSUSY distribution and performs a scan in  $\lambda'_{331}(M_{GUT})$ , assuming the CMSSM10.1.1 [10] CMSSM parameters  $m_0 = 125$  GeV,  $M_{1/2} = 500$  GeV,  $A_0 = 0$ ,  $\tan\beta = 10$  and  $\mu > 0$ . The size of the coupling varies from  $\lambda'_{323} \in [0, 0.7]$ . The program prints out the value of the right-handed stop pole mass and any problems associated with the point in question for each value of  $\lambda'_{323}(M_{GUT})$ .

The sample program has the following form:

```
/**
Project:    SOFTSUSY
File:      rpvmain.cpp
Authors:   B.C. Allanach, Markus Bernhardt
Manual:    B.C. Allanach and M.A. Bernhardt, CPC 181 (2010) 232,
           arXiv:0903.1805 and
```



```

        B.C. Allanach, hep-ph/0104145, Comp. Phys. Comm. 143 (2002) 305
Webpage:   http://projects.hepforge.org/softsusy/
Description: main calling program example:
            - scanning CMSSM10.1 with one varying RPV coupling
*/

#include "rpvmain.h"

#ifdef HAVE_CONFIG_H
#include "config.h"
#endif

int main() {
    /// Sets up exception handling
    signal(SIGFPE, FPE_ExceptionHandler);

    bool gaugeUnification = true, ewsBBCscale = false;

    /// Sets format of output: 6 decimal places
    outputCharacteristics(6);

    /// Header
    cout << "# SOFTSUSY" << PACKAGE_VERSION << endl;
    cout << "# If you use SOFTSUSY, please refer to B.C. Allanach, \n";
    cout << "# Comput. Phys. Commun. 143 (2002) 305, hep-ph/0104145;\n";
    cout << "# For RPV aspects, B.C. Allanach and M.A. Bernhardt, \n";
    cout << "# Comp. Phys. Commun. 181 (2010) 232, arXiv:0903.1805.\n";

    /// "try" catches errors in main program and prints them out
    try {
        /// Turn quark mixing on
        int mixing = 1;

        /// Contains default quark and lepton masses and gauge coupling
        /// information
        QcdQcd oneset;      ///< See "lowe.h" for default parameter definitions
        oneset.toMz();      ///< Runs SM fermion masses to MZ
        oneset.runto(oneset.displayPoleMt());    ///< Runs SM fermion masses to mt

        /// Print out the Standard Model data being used, as well as quark mixing
        /// assumption and the numerical accuracy of the solution
        cout << "# Low energy data in SOFTSUSY: mixing=" << mixing << " TOLERANCE="
            << TOLERANCE << endl;

        /// set parameters
        double tanb = 10.;
        int sgnMu = 1;
        double mgutGuess = 2.e16;
        double a0 = 0.0, m12 = 500.0, m0 = 125.0;

        /// number of points for scan
        const int numPoints = 20;

        /// parameter region
        double Start = 0. , End = 0.6;

```

```

DoubleVector pars(3);
/// set basic entries in pars
pars(1) = m0; pars(2) = m12; pars(3) = a0;

cout << "# l''_{323}(M_X) m_stop_R      # Problem flag" << endl;
/// loop over parameter space region
int ii; for (ii=0; ii<=numPoints; ii++){
    double lambda = Start + ((End - Start) / double(numPoints) * double(ii));

    /// define rpvSoftsusy object
    RpvSoftsusy kw;
    kw.setMixing(mixing);

    /// set lambda coupling at mgut
    kw.setLamPrimePrime(3, 2, 3, lambda);

    /// output parameters into double vector pars used by lowOrg
    kw.rpvDisplay(pars);

    /// generate spectrum in RpvSoftsusy object kw
    kw.lowOrg(rpvSugraBcs, mgutGuess, pars, sgnMu,
             tanb, oneset, gaugeUnification, ewsbBCscale);

    /// outputs for this scan
    cout << lambda << " " << kw.displayPhys().mu(2, 3) << " # "
         << kw.displayProblem() << endl;
}
}
catch(const string & a) {
    cerr << a; return -1;
}
catch(const char *a) {
    printf("%s", a); return -1;
}
return 0;
}

```

After including a header file, global variables are defined. These are all described in the  $R_p$  manual [5]. After setting the output accuracy, the program output begins with a title print-out. Then follow some variables specifying the Standard Model input parameters, the mixing switch, which determines how any quark mixing is implemented (as in ref. [5]) and the iteration precision of the output, TOLERANCE. The running masses of the SM fermions and the QED and QCD gauge couplings are determined at  $M_Z$  from data with the method toMz. If the switch gaugeUnification=true, SOFTSUSY will determine mGutGuess as the scale  $M_{GUT}$  of electroweak gauge unification. In order to do this, it requires an initial guess, which must be supplied as the initial value of the variable mGutGuess (in GeV)<sup>1</sup>.

The next step is the definition of the CMSSM parameters  $A_0/\text{GeV}=a_0$ ,  $M_{1/2}/\text{GeV}=m_{12}$ ,  $\tan\beta = \tan b$  and  $m_0/\text{GeV}=m_0$ . Next, a for loop performs the scan over  $\lambda''_{323}(M_{GUT})$ . In the example given (CMSSM), the first three parameters are  $\text{pars}(1) = m_0$ ;  $\text{pars}(2) = m_{12}$ ;  $\text{pars}(3) = a_0$ . The pars vector is needed to keep track of the boundary conditions set at  $M_{GUT}$ . In the iterative SOFTSUSY algorithm the parameters in the RpvSoftsusy object change due to the RGE running. they are re-set in every iteration at  $M_{GUT}$  from the unchanged DoubleVector pars parameters. Users should note that for lepton number violating couplings, users should use the updated RpvNeutrino object as in Ref. [19], rather than a RpvSoftsusy one.

---

<sup>1</sup>If the user wishes to provide this,  $2 \times 10^{16}$  GeV is a good initial guess for  $M_{GUT}$ .

We do not fill the other 102  $R_p$  entries of `pars` explicitly. This would be tedious and an additional source of potential bugs. Instead, we fill the `RpvSoftsusy` object itself using the `setLamPrimePrime` method in this example. We use the `rpvDisplay` method: this fills the `pars` vector automatically with what was set already inside the `RpvSoftsusy` object, while leaving the first nine entries in the vector unchanged. The `rpvDisplay` method automatically changes the length of `pars` appropriately. After this, the actual SOFTSUSY main driving method `lowOrg` is called, the first argument specifying the type of boundary condition (currently `rpvSugraBcs`), which assumes that `pars` has already been prepared by using the `rpvDisplay` object. This is followed by the output of the GUT-scale coupling  $\lambda$ , the pole right-handed stop mass and any problems in the calculation of the parameter point. Finally, the `catch` commands print any errors produced by the program.

### Appendix C. Sample Output

We now present some non-SLHA2 compliant SOFTSUSY output for the example program presented in Section Appendix B. The output obtained from this command is:

```
# SOFTSUSY4.1.1
# If you use SOFTSUSY, please refer to B.C. Allanach,
# Comput. Phys. Commun. 143 (2002) 305, hep-ph/0104145;
# For RPV aspects, B.C. Allanach and M.A. Bernhardt,
# Comp. Phys. Commun. 181 (2010) 232, arXiv:0903.1805.
# Low energy data in SOFTSUSY: mixing=1 TOLERANCE=1.000000e-04
# l''_{323}(M_X) m_stop_R      # Problem flag
0.000000e+00  8.125070e+02 #
3.000000e-02  8.105425e+02 #
6.000000e-02  8.047624e+02 #
9.000000e-02  7.955202e+02 #
1.200000e-01  7.834193e+02 #
1.500000e-01  7.691317e+02 #
1.800000e-01  7.533274e+02 #
2.100000e-01  7.367929e+02 #
2.400000e-01  7.201850e+02 #
2.700000e-01  7.040139e+02 #
3.000000e-01  6.886382e+02 #
3.300000e-01  6.742808e+02 #
3.600000e-01  6.610547e+02 #
3.900000e-01  6.489915e+02 #
4.200000e-01  6.380667e+02 #
4.500000e-01  6.282193e+02 #
4.800000e-01  6.193737e+02 #
5.100000e-01  6.114441e+02 #
5.400000e-01  6.043384e+02 #
5.700000e-01  5.979777e+02 #
6.000000e-01  5.922807e+02 #
```

After a header line labelling the contents of the columns, we see the GUT scale value of  $\lambda''_{323}$  assumed, then the pole value of the right-handed stop mass and any problems associated with the parameter point being examined. For large values of  $\lambda''_{323}$ , a quasi-fixed point occurs in the renormalisation group equations, and no perturbative solution to the RGEs exists.

### Appendix D. Object Structure

We now go on to sketch the objects and their relationship to each other. This is necessary information for any generalisation beyond the  $R_p$  MSSM. Only methods and data which are deemed of possible importance for prospective users are mentioned here, but there are many others within the program itself.

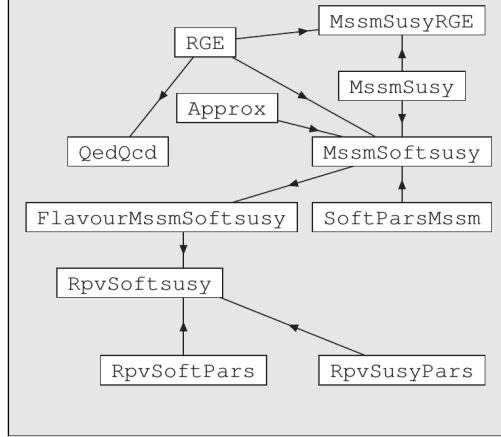


Figure D.2: Heuristic high-level object structure of SOFTSUSY. Inheritance is displayed by the direction of the arrows.

### Appendix D.1. Tensor

The SOFTSUSY program comes with its own linear algebra classes of vectors and matrices (e.g. `DoubleMatrix`) that have been introduced in [5]. New in this version is the class `Tensor`, given in files `tensor.h` and `tensor.cpp`. This class has been added to implement the three-index tensors containing some of the  $\mathcal{R}_p$  couplings into the program. For this reason, the class is specifically designed as a vector of three single objects, each of type `DoubleMatrix`. The dimension of `Tensor` is  $(3, 3, 3)$ . The class also contains linear algebra functions for multiplication, addition or subtraction with matrices and vectors via over loaded operators. For more detail we refer the interested reader to the technical documentation on the SOFTSUSY web-site. The SOFTSUSY internal representation of the trilinear  $\mathcal{R}_p$  couplings is different to the user interface. While the user interface uses the common  $\lambda_{ijk} L_i L_j \bar{E} \in W_{\mathcal{R}_p}$  notation, this coupling is internally represented in the [16] `Tensor` notation in terms of three matrices as in Eq. D.1.

$$(\text{Lambda\_U}_i)_{jk} \equiv \lambda''_{ijk}, \quad (\text{Lambda\_D}_i)_{jk} \equiv \lambda'_{jki}, \quad (\text{Lambda\_E}_i)_{jk} \equiv \lambda_{jki}. \quad (\text{D.1})$$

This does not stop the user from only operating the program using the user interface and the usual  $\lambda_{ijk}$  conventions.  $\lambda'_{jki}$ ,  $\lambda''_{ijk}$  are also stored within analogous `Tensor` representations of three  $3 \times 3$  matrices.

### Appendix D.2. General object structure

From an RGE point of view, data in a particular quantum field theory consist of a set of parameters defined at some renormalisation scale  $Q$ . A set of  $\beta$  functions describes the evolution of the parameters and masses to a different scale  $Q'$ . This concept is embodied in an *abstract* RGE object, which contains the methods required to run objects of derived classes to different renormalisation scales (their beta functions). The other objects displayed in figure D.2 are particular instances of RGE, and therefore inherit from it. `QedQcd`, `MssmSusy`, `SoftParsMssm`, `MssmSoftsusy` and `Approx` objects encode the  $\mathcal{R}_p$  part of the MSSM and its SM input data, along with the level of approximation of the calculation [5]. `RpvSusyPars` contains all of the supersymmetric  $\mathcal{R}_p$  couplings contained within Eq. (2). `RpvSoftPars` contains the  $\mathcal{R}_p$  soft supersymmetry breaking parameters listed in Eqs. (3) and (4). `RpvSoftsusy` is the  $\mathcal{R}_p$  generalisation of the `MssmSoftsusy` class, and contains all  $\mathcal{R}_p$  MSSM couplings along with their beta functions. Methods for the `RpvSoftsusy` class exist to perform the minimisation of the neutral scalar potential as well as the calculation of Yukawa couplings described in section 4. Neutrino, neutralino and chargino masses and mixings are calculated within this class. Code in the `MssmSoftsusy` class organises and performs the main part of the calculation, using polymorphism to detect the correct  $\beta$  functions to use (in this case, the  $\mathcal{R}_p$ -MSSM  $\beta$  functions). All of the `Rpv` objects contain default constructors and destructors, as well as overloaded `>>`, `<<` operators for input and output. There is always an implicit dependence of running RGE quantities on the current renormalisation scale  $Q$ . Thus, if a method is called that returns one of the object's couplings or masses, that object will return it at the current scale

| data variable  |  | methods  |
|--|--|--|
| Tensor lu, ld, le<br>$\Lambda_E, \Lambda_D, \Lambda_U$ | trilinear $R_p$ superpotential couplings | displayLam, displayLamPrime<br>displayLamPrimePrime<br>setLam, setLamPrime<br>setLamPrimePrime |
| DoubleVector kappa<br>$\kappa_{i=1,2,3}/\text{GeV}$    | bilinear $R_p$ parameter                 | displayKappa<br>setKappa   |

Table D.1: RpvSusyPars class data and accessor methods. See Eq. (D.1) for a translation between the structure of Tensor lu, ld, le.

| data variable  |                                     | methods                              |
|--|-------------------------------------|--------------------------------------|
| DoubleVector mH1lsq<br>$m_{\tilde{L}_{i=1,2,3}H_1}^2/\text{GeV}^2$                                 | bi-linear scalar parameters         | displayMh11Squared<br>setMh11Squared |
| Tensor her, hdr, hur<br>$\{h_{ijk}, h'_{ijk}, h''_{ijk}\}/\text{GeV}$<br>$i, j, k \in \{1, 2, 3\}$ | trilinear $R_p$ scalar interactions | displayHr<br>setHr                   |
| DoubleVector dr<br>$D_{i=1,2,3}/\text{GeV}^2$  | bilinear $R_p$ scalar parameters    | displayDr<br>setDr                   |

Table D.2: RpvSoftPars class data and accessor methods. All parameters are running parameters, evaluated at the  $\overline{DR}$  scale  $\mu$ .

$Q$  of the object. In the following, we provide basic information on the classes associated with  $R_p$ , so that users may program using the class structure of SOFTSUSY. More detailed and technical documentation on the program should be obtained from the SOFTSUSY website.

### Appendix D.3. RpvSusyPars class

Each of the higher level objects described in this appendix have explicitly named `display` and `set` methods that are used to access or change the data contained within each object. In table D.1 (as in the following tables in this section), these accessing methods are listed on the same row as the relevant data variable. The data and input/output methods in the RpvSusyPars class are presented in table D.1. When using Tensor objects lu, ld, le, there exists an enumerated type RpvCouplings  $\in \{\text{LU}, \text{LD}, \text{LE}\}$  used as arguments to the `displayLambda`, `setLambda` methods. This argument selects the type of coupling ( $\lambda''_{ijk}$ ,  $\lambda'_{ijk}$  or  $\lambda_{ijk}$ ).

### Appendix D.4. RpvSoftPars class

The data and input/output methods in the RpvSoftPars class are presented in table D.2. The `displayHr`, `setHr` methods take a parameter of the enumerated type RpvCouplings as their first argument to select a particular tri-linear scalar interaction that is ( $h''_{ijk}$ ,  $h'_{ijk}$  or  $h_{ijk}$  depending upon the argument).

### Appendix D.5. RpvSoftsusy class

The data and important methods in the RpvSoftsusy class are presented in table D.3. As well as standard constructors and destructors, there exists a method `beta`, that calculates the numerical values of the  $\beta$ -function. The `rpvDisplay` method is used in the example program `rpvmain.cpp`, and fills a vector with MSSM running parameters in a certain order. `rpvSet` is used to set MSSM running parameters according to a DoubleVector argument, assuming the same order as `rpvDisplay`. If the user wishes to provide their own function encoding high-scale boundary conditions on the soft supersymmetry breaking terms, they must provide a function

```
void (*boundaryCondition)(MssmSoftsusy &, const DoubleVector &)
```

| data variable  |  | methods                                    |
|--|--|--|
| DoubleVector $\text{snuVevs}$<br>$v_{i=1,2,3}/\text{GeV}$        | sneutrino VEVs   | displaySneutrinoVevs<br>setSneutrinoVevs   |
| DoubleVector $\text{nuMasses}$<br>$m_{\nu_{i=1,2,3}}/\text{GeV}$ | neutrino masses  | setNeutrinoMasses<br>displayNeutrinoMasses |
| DoubleMatrix $\text{neutralFermMixing}$<br>$O$ (7 by 7)          | neutral fermion<br>mixing  | displayNeutralMixing<br>setNeutralMixing   |
| DoubleMatrix $\text{Uch}$<br>$U$ (5 by 5)                        | charged fermion<br>mixing  | displayUch<br>setUch                       |
| DoubleMatrix $\text{Vch}$<br>$V$ (5 by 5)                        | charged fermion<br>mixing  | displayVch<br>setUch                       |
| method   | function   |  |
| void $\text{rpvDisplay}$ (DoubleVector)                          | returns all 102 $\mathcal{R}_p$ MSSM running parameters as a DoubleVector, leaving the first 9 entries untouched         |  |
| void $\text{rpvSet}$ (DoubleVector)                              | sets all 102 $\mathcal{R}_p$ MSSM parameters from a user-provided DoubleVector, starting from its 10 <sup>th</sup> entry |  |

Table D.3: RpvSoftsusy class. The parameters  $v_i, m_{\nu_i}, O, U, V$  are tree-level  $\overline{DR}$  parameters.

which is then passed as the first argument to the main driving method `lowOrg` [5]. One can specify CMSSM conditions for the  $R_p$  parts, plus all specified  $\mathcal{R}_p$  interactions at the high scale by using `rpvSugraBcs`. Alternatively, one can specify  $R_p$ -conserving gauge mediated supersymmetry breaking conditions plus  $\mathcal{R}_p$  interactions at the scale  $M_{mess}$  by using `rpvGmsbBcs`. In this case, the first elements of `pars` should contain  $n$ , the number of vector like 5-plets of messenger fields, the messenger mass scale  $M_{mess}$  in GeV,  $\Lambda$  and  $C_{grav}$ , the constant that determines the gravitino mass [31]. `rpvAmsbBcs` implements the minimal anomaly mediated supersymmetry breaking [32] assumption, but *neglects*  $\mathcal{R}_p$  couplings in the high-scale boundary condition. For this case, the first two parameters of `pars` should be `m32` and `m0`, respectively. This approximation ought to be reasonable for small dimensionless  $\mathcal{R}_p$  couplings. Users should note that for lepton number violating couplings, users should use the updated `RpvNeutrino` object as in Ref. [19], rather than a `RpvSoftsusy` one.

## References

- [1] B. C. Allanach, SUSY Predictions and SUSY Tools at the LHC, Eur. Phys. J. C59 (2009) 427–443. arXiv:0805.2088, doi:10.1140/epjc/s10052-008-0695-2.
- [2] W. W. Armstrong, et al., ATLAS: Technical proposal for a general-purpose p p experiment at the Large Hadron Collider at CERN, CERN-LHCC-94-43.
- [3] G. L. Bayatian, et al., CMS physics: Technical design report, CERN-LHCC-2006-001.
- [4] F. E. Paige, S. D. Protopopescu, H. Baer, X. Tata, ISAJET 7.69: A Monte Carlo event generator for p p, anti-p p, and e+ e- reactions, arXiv:hep-ph/0312045.
- [5] B. C. Allanach, SOFTSUSY: A C++ program for calculating supersymmetric spectra, Comput. Phys. Commun. 143 (2002) 305–331. arXiv:hep-ph/0104145, doi:10.1016/S0010-4655(01)00460-X.
- [6] A. Djouadi, J.-L. Kneur, G. Moultaka, SuSpect: A Fortran code for the supersymmetric and Higgs particle spectrum in the MSSM, Comput. Phys. Commun. 176 (2007) 426–455. arXiv:hep-ph/0211331, doi:10.1016/j.cpc.2006.11.009.
- [7] W. Porod, SPheno, a program for calculating supersymmetric spectra, SUSY particle decays and SUSY particle production at e+ e- colliders, Comput. Phys. Commun. 153 (2003) 275–315. arXiv:hep-ph/0301101, doi:10.1016/S0010-4655(03)00222-4.
- [8] P. Skands, et al., SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators, JHEP 07 (2004) 036. arXiv:hep-ph/0311123.
- [9] B. Allanach, et al., SUSY Les Houches Accord 2, Comp. Phys. Commun. 180 (2009) 8–25. arXiv:0801.0045, doi:10.1016/j.cpc.2008.08.004.
- [10] S. S. AbdusSalam, B. C. Allanach, H. K. Dreiner, J. Ellis, U. Ellwanger, J. Guion, S. Heinemeyer and M. Kraemer *et al.*, Eur. Phys. J. C 71 (2011) 1835 [arXiv:1109.3859 [hep-ph]].
- [11] H. K. Dreiner, An introduction to explicit R-parity violation, arXiv:hep-ph/9707435; G. Bhattacharyya, A brief review of R-parity-violating couplings, arXiv:hep-ph/9709395; G. Bhattacharyya, R-parity-violating supersymmetric Yukawa couplings: A mini-review, Nucl. Phys. Proc. Suppl. 52A (1997) 83, arXiv:hep-ph/9608415.

- [12] L. E. Ibanez, G. G. Ross, Discrete gauge symmetries and the origin of baryon and lepton number conservation in supersymmetric versions of the standard model, Nucl. Phys. B368 (1992) 3–37. doi:10.1016/0550-3213(92)90195-H.
- [13] H. K. Dreiner, C. Luhn, H. Murayama, M. Thormeier, Proton Hexality from an Anomalous Flavor U(1) and Neutrino Masses - Linking to the String Scale, Nucl. Phys. B795 (2008) 172–200. arXiv:0708.0989, doi:10.1016/j.nuclphysb.2007.11.014.
- [14] L. J. Hall, M. Suzuki, Explicit R-Parity Breaking in Supersymmetric Models, Nucl. Phys. B231 (1984) 419. doi:10.1016/0550-3213(84)90513-3.
- [15] W. Buchmuller, L. Covi, K. Hamaguchi, A. Ibarra, T. Yanagida, Gravitino dark matter in R-parity breaking vacua, JHEP 03 (2007) 037. arXiv:hep-ph/0702184.
- [16] B. C. Allanach, A. Dedes, H. K. Dreiner, The R parity violating minimal supergravity model, Phys. Rev. D69 (2004) 115002. arXiv:hep-ph/0309196, doi:10.1103/PhysRevD.69.115002.
- [17] A. Dedes, H. E. Haber, J. Rosiek, Seesaw mechanism in the sneutrino sector and its consequences, JHEP 11 (2007) 059. arXiv:0707.3718, doi:10.1088/1126-6708/2007/11/059.
- [18] J. A. Casas, V. Di Clemente, M. Quiros, The effective potential in the presence of several mass scales, Nucl. Phys. B553 (1999) 511–530. arXiv:hep-ph/9809275, doi:10.1016/S0550-3213(99)00262-X.
- [19] B.C. Allanach, M. Hanussek and C.H. Kom, Computation of Neutrino Masses in R-parity Violating Supersymmetry: SOFTSUSY3.2 arXiv:1109.3859 [hep-ph]
- [20] V. D. Barger, M. S. Berger, P. Ohmann, The Supersymmetric particle spectrum, Phys. Rev. D49 (1994) 4908–4930. arXiv:hep-ph/9311269, doi:10.1103/PhysRevD.49.4908.
- [21] M. Nowakowski, A. Pilaftsis, W and Z boson interactions in supersymmetric models with explicit R-parity violation, Nucl. Phys. B461 (1996) 19–49. arXiv:hep-ph/9508271, doi:10.1016/0550-3213(95)00594-3.
- [22] R. Barbier, et al., R-parity violating supersymmetry, Phys. Rept. 420 (2005) 1–202. arXiv:hep-ph/0406039, doi:10.1016/j.physrep.2005.08.006.
- [23] H. K. Dreiner, S. Grab, All Possible Lightest Supersymmetric Particles in R-Parity Violating mSUGRA, arXiv:0811.0200.
- [24] M. A. Bernhardt, S. P. Das, H. K. Dreiner, S. Grab, Sneutrino as Lightest Supersymmetric Particle in B3 mSUGRA Models and Signals at the LHC, arXiv:0810.3423.
- [25] B. C. Allanach, M. A. Bernhardt, H. K. Dreiner, C. H. Kom, P. Richardson, Mass Spectrum in R-Parity Violating mSUGRA and Benchmark Points, Phys. Rev. D75 (2007) 035002. arXiv:hep-ph/0609263, doi:10.1103/PhysRevD.75.035002.
- [26] B. C. Allanach, C. H. Kom, Lepton number violating mSUGRA and neutrino masses, JHEP 04 (2008) 081. arXiv:0712.0852, doi:10.1088/1126-6708/2008/04/081.
- [27] B. C. Allanach, C. H. Kom, H. Pas, Large Hadron Collider probe of supersymmetric neutrinoless double beta decay mechanism, arXiv:0902.4697.
- [28] B. C. Allanach, C. H. Kom, H. Pas, LHC and B physics probes of neutrinoless double beta decay in supersymmetry without R-parity, arXiv:0903.0347.
- [29] T. Sjostrand, S. Mrenna, P. Skands, A Brief Introduction to PYTHIA 8.1, Comput. Phys. Commun. 178 (2008) 852–867. arXiv:0710.3820, doi:10.1016/j.cpc.2008.01.036.
- [30] B. C. Allanach, et al., The Snowmass points and slopes: Benchmarks for SUSY searches, Eur. Phys. J. C25 (2002) 113–123. arXiv:hep-ph/0202233, doi:10.1007/s10052-002-0949-3.
- [31] S. Ambrosiano, G. D. Kribs, S. P. Martin, Signals for gauge-mediated supersymmetry breaking models at the CERN LEP2 collider, Phys. Rev. D56 (1997) 1761–1777. arXiv:hep-ph/9703211, doi:10.1103/PhysRevD.56.1761.
- [32] L. Randall, R. Sundrum, Out of this world supersymmetry breaking, Nucl. Phys. B557 (1999) 79–118. arXiv:hep-th/9810155, doi:10.1016/S0550-3213(99)00359-4.