SuperIso Relic: A program for calculating relic density and flavour physics observables in Supersymmetry

A. Arbey*

Université de Lyon, Lyon, F-69000, France; Université Lyon 1,
Villeurbanne, F-69622, France; Centre de Recherche Astrophysique de Lyon,
Observatoire de Lyon, 9 avenue Charles André, Saint-Genis Laval cedex,
F-69561, France; CNRS, UMR 5574; Ecole Normale Supérieure de Lyon, Lyon, France.

F. Mahmoudi†

Clermont Université, Université Blaise Pascal, CNRS/IN2P3,
LPC, BP 10448, 63000 Clermont-Ferrand, France

Abstract

We describe `SuperIso Relic`, a public program for evaluation of relic density and flavour physics observables in the minimal supersymmetric extension of the Standard Model (MSSM) and in the next-to-minimal supersymmetric extension of the Standard Model (NMSSM). `SuperIso Relic` is an extension of the `SuperIso` program which adds to the flavour observables of `SuperIso` the computation of all possible annihilation and coannihilation processes of the LSP which are required for the relic density calculation. All amplitudes have been generated at the tree level with `LanHEP`/`FeynArts`/`FormCalc`, and widths of the Higgs bosons are computed with `FeynHiggs` or `Hdecay` at the two-loop level. `SuperIso Relic` also provides the possibility to modify the assumptions of the cosmological model, and to study their consequences on the relic density.

PACS numbers: 11.30.Pb, 12.60.Jv, 13.20.-v, 95.35.+d

---

*Electronic address: `alexandre.arbey@ens-lyon.fr`
†Electronic address: `mahmoudi@in2p3.fr`
`URL: http://superiso.in2p3.fr/relic`

Contents

1 Introduction

The dark matter problem remains one of the most puzzling questions in cosmology. Cosmological analyses reveal that dark matter may be composed of non-baryonic particles, but their nature is still to be discovered. Many new physics models provide a natural solution to the dark matter problem. Supersymmetry in particular offers a stable particle, the lightest supersymmetric particle (LSP), if R-parity is conserved, which could be the main component of the dark matter in the Universe. The current density of the LSP can be calculated and is referred as relic density. Compared to the latest precise WMAP measurements of the dark matter density [1], relic density can impose stringent constraints on the supersymmetric parameters.

`SuperIso Relic` is an extension of the `SuperIso` program to the calculation of the relic density. The program calculates the relic density as well as the flavour physics observables using a SUSY Les Houches Accord file (SLHA1 [2] or SLHA2 [3]) as input, either generated automatically via a call to SOFTSUSY [4], ISAJET [5], SPheno [6], SuSpect [7] or NMSSMTools [8], or provided by the user. The calculation can be performed automatically for different supersymmetry breaking scenarios in the minimal supersymmetric extension of the Standard Model (MSSM) or in the next-to-minimal supersymmetric extension of the Standard Model (NMSSM).

One of the most important features of `SuperIso Relic` in comparison to the other public relic density calculation codes, DarkSusy [9], IsaRed [10] and Micromegas [11], is that it provides the possibility to alter the underlying cosmological model, by modifying for example the radiation equation-of-state, the expansion rate or the thermal properties of the Universe in the period before Big-Bang nucleosynthesis (BBN), which is experimentally inaccessible and remains theoretically obscure. In [12–15], we studied the effects of different parametrizations of modification of the expansion rate or of the entropy content of the Universe before BBN on the relic density calculation and showed that they can strongly modify the calculated relic density and therefore change the relic density constraints on supersymmetric parameter space. `SuperIso Relic` makes it possible to evaluate the uncertainties on the relic density due to the cosmological model, and inversely, to make prediction on the early Universe properties using the particle physics constraints.

In the following, first the content of the `SuperIso Relic` package will be presented, as well as the list of the main routines used for the relic density calculation. The procedure to use `SuperIso Relic` will be then explained, and the inputs and outputs of the program will be introduced. Finally, some examples of results obtained with `SuperIso Relic` will be given. In the Appendices, a description of the formulas and models used for computing the relic density will be detailed.

2 Content of the SuperIso Relic package

`SuperIso Relic` is a mixed C / Fortran program devoted to the calculation of the relic density in addition to many flavour observables in Supersymmetry. Sixteen main programs are provided in the package as guidelines, but the users are also invited to write their own main programs. In particular `slha.c` can scan files written following the SUSY Les Houches Accord formats, and calculates the implemented observables. The main programs `cmssm.c`, `amsb.c`, `hcamsb.c`, `gmsb.c`, `mmamsb.c`, and `nuhm.c` have to be linked to at least one of the `SOFTSUSY` [4], the `ISASUGRA/ISAJET` [5], the `SPheno` [6] and/or the `SuSpect` [7] packages, in order to compute supersymmetric mass spectra and couplings within respectively the CMSSM, AMSB, HCAMSB, MMAMSB, GMSB or NUHM scenarios for the MSSM. The programs `cnmssm.c`, `ngmsb.c` and `nnuhm.c` have to be linked to `NMSSMTools` [8] to calculate the spectra within the CNMSSM, NGMSB or NNUHM scenarios for the NMSSM.

The main steps to compute the observables in `SuperIso Relic` are given in the following:

- Generation of a SLHA file with a spectrum generator (or supply of a SLHA file by the user),
- Scan of the SLHA file,
- Calculation of the widths of the Higgs bosons with `FeynHiggs` or `Hdecay`,
- Computation of the squared amplitudes of the annihilation diagrams involved in the relic density calculation,
- Computation of the thermally averaged total annihilation cross section,
- Solving of the Boltzmann equations and computation of the relic density,
- Calculation of the flavour physics observables.

It should be noted that the relic density calculation is performed even if the LSP is a charged particle. A theoretical description of the calculation of the thermally averaged total annihilation cross section can be found in Appendix A and the detail of the calculation of the relic density in the cosmological standard model is given in Appendix B. We refer to [16] for a complete description of the calculation of the flavour observables.

The processes involved in the relic density calculation are all the annihilation and coannihilation processes of the type

$$\tilde{i} + \tilde{j} \to k + l \tag{1}$$

where $\tilde{i}, \tilde{j}$ are supersymmetric particles and k, l are Standard Model particles. The number of involved processes is more than 3000 in the MSSM or 5000 in the NMSSM, and the number of diagrams is even larger. To generate all the squared amplitudes, we have written a `Mathematica` [17] script which uses the `LanHEP` [18] Lagrangians in `FeynArts` format, calls `FeynArts` [19] and `FormCalc` [20] and generates the necessary routines for the numerical computation of the amplitudes. These routines are part of `SuperIso Relic` and can be found in `src/relic` and therefore the user does not need to have `Mathematica` or to install other packages. They rely on `FeynHiggs` [21] or `Hdecay` [22] to calculate the widths of the Higgs bosons at two-loop level.
`Hdecay 3.53` and `FeynHiggs 2.8.0` are included in the `SuperIso Relic v3.1` package, and they can be found in `src/contrib`. Therefore the user does not need to download these programs separately.

The compilation process of all the needed routines is very long (∼ hour), and their calculation can take time. Fortunately, all the squared amplitude routines are not necessary at the same time, as some processes have only negligible effects. Therefore, all the squared amplitudes are not computed for a SUSY parameter space point, and a selection is performed to save time, as described in Appendix A.

2.1 Parameter structures

The package `SuperIso Relic` relies on the definition of a main structure in `src/include.h`, which is defined as follows:

```
typedef struct parameters
/* structure containing all the scanned parameters from the SLHA file */
{
int SM;
int model; /* CMSSM=1, GMSB=2, AMSB=3 */
int generator; /* ISAJET=1, SOFTSUSY=3, SPHENO=4, SUSPECT=5, NMSSMTOOLS=6 */
double Q; /* Qmax ; default = M_EWSB = sqrt(m_stop1*mstop2) */
double m0,m12,tan_beta,sign_mu,A0; /* CMSSM parameters */
double Lambda,Mmess,N5,cgrav,m32; /* AMSB, GMSB parameters */
double mass_Z,mass_W,mass_b,mass_top_pole,mass_tau_pole; /* SM parameters */
double inv_alpha_em,alphas_MZ,Gfermi,GAUGE_Q; /* SM parameters */
double charg_Umix[3][3],charg_Vmix[3][3],stop_mix[3][3],sbot_mix[3][3],
stau_mix[3][3],neut_mix[6][6],mass_neut[6],alpha; /* mass mixing matrices */
double Min,M1_Min,M2_Min,M3_Min,At_Min,Ab_Min,Atau_Min,M2H1_Min,M2H2_Min,
mu_Min,M2A_Min,tb_Min,mA_Min; /* optional input parameters at scale Min */
double MeL_Min,MmuL_Min,MtauL_Min,MeR_Min,MmuR_Min,MtauR_Min; /* optional
input parameters at scale Min */
double MqL1_Min,MqL2_Min,MqL3_Min,MuR_Min,McR_Min,MtR_Min,MdR_Min,MsR_Min,
MbR_Min; /* optional input parameters at scale Min */
double N51,N52,N53,M2H1_Q,M2H2_Q; /* optional input parameters (N51...3: GMSB) */
double mass_d,mass_u,mass_s,mass_c,mass_t,mass_e,mass_nue,mass_mu,mass_num,
mass_tau,mass_nut; /* SM masses */
double mass_gluon,mass_photon,mass_Z0; /* SM masses */
double mass_h0,mass_H0,mass_A0,mass_H,mass_dnl,mass_upl,mass_stl,mass_chl,mass_b1,
mass_t1; /* Higgs & superparticle masses */
double mass_el,mass_nuel,mass_mul,mass_numl,mass_tau1,mass_nutl,mass_gluino,
mass_cha1,mass_cha2; /* superparticle masses */
double mass_dnr,mass_upr,mass_str,mass_chr,mass_b2,mass_t2,mass_er,mass_mur,
mass_tau2; /* superparticle masses */
double mass_nuer,mass_numr,mass_nutr,mass_graviton,mass_gravitino; /* superparticle
masses */
double gp,g2,g3,YU_Q,yut[4],YD_Q,yub[4],YE_Q,yutau[4]; /* Yukawa couplings */
double HMIX_Q,mu_Q,tanb_GUT,Higgs_VEV,mA2_Q,MSOFT_Q,M1_Q,M2_Q,M3_Q; /* parameters
at scale Q */
double MeL_Q,MmuL_Q,MtauL_Q,MeR_Q,MmuR_Q,MtauR_Q,MqL1_Q,MqL2_Q,MqL3_Q,MuR_Q,McR_Q,
```

```
MtR_Q,MdR_Q,MsR_Q,MbR_Q; /* masses at scale Q */
double AU_Q,A_u,A_c,A_t,AD_Q,A_d,A_s,A_b,AE_Q,A_e,A_mu,A_tau; /* trilinear couplings */

/* SLHA2 */
int NMSSM,RV,CPV,FV;
double mass_nutau2,mass_e2,mass_nue2,mass_mu2,mass_numu2,mass_d2,mass_u2,mass_s2,mass_c2;
double CKM_lambda,CKM_A,CKM_rhobar,CKM_etabar;
double PMNS_theta12,PMNS_theta23,PMNS_theta13,PMNS_delta13,PMNS_alpha1,PMNS_alpha2;
double lambdaNMSSM_Min,kappaNMSSM_Min,AlambdaNMSSM_Min,AkappaNMSSM_Min,lambdaSNMSSM_Min,
xiFNMSSM_Min,xiSNMSSM_Min,mupNMSSM_Min,mSp2NMSSM_Min,mS2NMSSM_Min,mass_H03,mass_A02,
NMSSMRUN_Q,lambdaNMSSM,kappaNMSSM,AlambdaNMSSM,AkappaNMSSM,lambdaSNMSSM,xiFNMSSM,
xiSNMSSM,mupNMSSM,mSp2NMSSM,mS2NMSSM; /* NMSSM parameters */
double PMNSU_Q,CKM_Q,IMCKM_Q,MSE2_Q,MSU2_Q,MSD2_Q,MSL2_Q,MSQ2_Q,TU_Q,TD_Q,TE_Q;
double CKM[4][4],IMCKM[4][4]; /* CKM matrix */
double H0_mix[4][4],A0_mix[4][4]; /* Higgs mixing matrices */
double sU_mix[7][7],sD_mix[7][7],sE_mix[7][7], sNU_mix[4][4]; /* mixing matrices */
double sCKM_msq2[4][4],sCKM_msl2[4][4],sCKM_msd2[4][4],sCKM_msu2[4][4],
sCKM_mse2[4][4]; /* super CKM matrices */
double PMNS_U[4][4]; /* PMNS mixing matrices */
double TU[4][4],TD[4][4],TE[4][4]; /* trilinear couplings */

/* non-SLHA*/
double mass_b_1S,mass_b_pole,mtmt;
double Lambda5; /* Lambda QCD */

/* Flavor constants */
double f_B,f_Bs,f_Ds,f_D,fK_fpi;
double m_B,m_Bs,m_pi,m_Ds,m_K,m_Kstar,m_D0,m_D;
double life_pi,life_K,life_B,life_Bs,life_D,life_Ds;

/* Decay widths */
int widthcalc; /* 0=none, 1=hdecay, 2=feynhiggs */
double width_h0,width_H0,width_A0,width_H,width_Z,width_W,width_top,
width_H03,width_A02;
double width_gluino,width_t1,width_t2,width_b1,width_b2,width_ul,width_ur,
width_dl,width_dr;
double width_cl,width_cr,width_sl,width_sr,width_el,width_er,width_ml,width_mr,
width_tau1,width_tau2;
double width_nuel,width_numl,width_nutaul,width_c1,width_c2,width_o1,width_o2,
width_o3,width_o4,width_o5;

/* CKM matrix */
double complex Vud,Vus,Vub,Vcd,Vcs,Vcb,Vtd,Vts,Vtb;

/* 2HDM */
int THDM_model;
```

```
double lambda_u[4][4],lambda_d[4][4],lambda_l[4][4];

/* NMSSMTools */
int NMSSMcoll,NMSSMtheory,NMSSMups1S,NMSSMetab1S;
}
parameters;
```

This structure contains all the important parameters and is called by most of the main functions in the program. An additional structure specific to the relic density calculation is also defined:

```
typedef struct relicparam
/* structure containing the cosmological model parameters */
{
int entropy_model;
double dd0,ndd,Tdend;
double sd0,nsd,Tsend;
double Sigmad0,nSigmad,TSigmaend;
double nt0,nnt,Tnend;
double table_eff[276][3];
}
relicparam;
```

This structure is used to define the cosmological model based on which the relic density calculation is performed.

2.2 Main routines

We review here the main routines of the code needed for the relic density calculation. For the main procedures related to the flavour observable calculations we refer the reader to [16].

The most relevant C routines are the following:

- `void Init_param(struct parameters* param)`

  This function initializes the `param` structure, setting all the parameters to 0, apart from the SM masses and the value of the strong coupling constant at the Z-boson mass, which receive the values given in the PDG2010 [23].

- `int Les_Houches_Reader(char name[], struct parameters* param)`

  This routine reads the SLHA file named `name`, and put all the read parameters in the structure `param`. It should be noted that a negative value for `param->model` indicates a problem in reading the SLHA file, or a model not yet included in SuperIso (such as R-parity breaking models). In this case, `Les_Houches_Reader` returns 0, otherwise 1.

- `int test_slha(char name[])`

  This routine checks if the SLHA file is valid, and if so returns 1. If not, -1 means that in the SLHA generator the computation did not succeed (*e.g.* because of tachyonic particles), -2 means that the considered model is not currently implemented in `SuperIso`, and -3 indicates that the provided file is either not in the SLHA format, or some important elements are missing.

- `int softsusy_cmssm(double m0, double m12, double tanb, double A0, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int softsusy_nuhm(double m0, double m12, double tanb, double A0, double mu, double mA, double mtop, double mbot, double alphas_mz, char name[])`
- `int softsusy_gmsb(double Lambda, double Mmess, double tanb, int N5, double cGrav, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int softsusy_amsb(double m0, double m32, double tanb, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`

  The above routines call `SOFTSUSY` to compute the mass spectrum corresponding to the input parameters (more details are given in the next sections), and return a SLHA file whose name has to be specified in the string `name`.

- `int isajet_cmssm(double m0, double m12, double tanb, double A0, double sgnmu, double mtop, char name[])`
- `int isajet_gmsb(double Lambda, double Mmess, double tanb, int N5, double cGrav, double sgnmu, double mtop, char name[])`
- `int isajet_nuhm(double m0, double m12, double tanb, double A0, double mu, double mA, double mtop, char name[])`
- `int isajet_amsb(double m0, double m32, double tanb, double sgnmu, double mtop, char name[])`
- `int isajet_mmamsb(double alpha, double m32, double tanb, double sgnmu, double mtop, char name[])`
- `int isajet_hcamsb(double alpha, double m32, double tanb, double sgnmu, double mtop, char name[])`

  The above routines call `ISAJET` to compute the mass spectrum corresponding to the input parameters (more details are given in the next sections), and return a SLHA file whose name has to be specified in the string `name`.

- `int spheno_cmssm(double m0, double m12, double tanb, double A0, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int spheno_gmsb(double Lambda, double Mmess, double tanb, int N5, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int spheno_amsb(double m0, double m32, double tanb, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`

  The above routines call `SPheno` to compute the mass spectrum corresponding to the input parameters (more details are given in the next sections), and return a SLHA file whose name has to be specified in the string `name`.

- `int suspect_cmssm(double m0, double m12, double tanb, double A0, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int suspect_gmsb(double Lambda, double Mmess, double tanb, int N5, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int suspect_amsb(double m0, double m32, double tanb, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`

  The above routines call `SuSpect` to compute the mass spectrum corresponding to the input parameters (more details are given in the next sections), and return a SLHA file whose name has to be specified in the string `name`.

- `int nmssmtools_cnmssm(double m0, double m12, double tanb, double A0, double lambda, double AK, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int nmssmtools_nnuhm(double m0, double m12, double tanb, double A0, double MHDGUT, double MHUGUT, double lambda, double AK, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`
- `int nmssmtools_ngmsb(double Lambda, double Mmess, double tanb, int N5, double lambda, double AL, double Del_h, double sgnmu, double mtop, double mbot, double alphas_mz, char name[])`

  The above routines call `NMSSMTools` to compute the mass spectrum corresponding to the input parameters (more details are given in the next sections), and return a SLHA file whose name has to be specified in the string `name`.

- `void ModelIni(struct parameters* param, double relicmass, double maxenergy)`

  This routine is an interface between the C routines and the Fortran routines and it defines all the Fortran variables using the C variables.

- `double findrelicmass(struct parameters* param, int *scalar)`

  This function determines the LSP mass, and checks if the LSP is scalar (`*scalar=1`) or fermionic (`*scalar=0`).

- `int Weff(double* res, double sqrtS, struct parameters* param, double relicmass)`

  This function calls the Fortran routines and returns the effective annihilation rate W_{eff} at a given center of mass energy `sqrtS`, following the procedure described in Appendix A.

- `int Init_relic(double Wefftab[NMAX][2], int *nlines_Weff, struct parameters* param)`

  This routine computes for different values of $\sqrt{s}$ the effective annihilation rates W_{eff} needed for the calculation of $\langle \sigma v \rangle$ using the `Weff` function, and collects them in table `Wefftab`.

- `double sigmav(double T, double relicmass, double Wefftab[NMAX][2], int nlines, struct parameters* param)`

  This function computes the averaged annihilation cross section $\langle \sigma v \rangle$ using the effective annihilation rates W_{eff} collected in table `Wefftab`.

- `double heff(double T, struct relicparam* paramrelic)`
  `double sgstar(double T, struct relicparam* paramrelic)`
  `double geff(double T, struct relicparam* paramrelic)`

  These three functions compute respectively h_{eff}, $\sqrt{g_*}$ and g_{eff} at the temperature `T`.

- `double Yeq(double T,struct parameters* param, struct relicparam* paramrelic)`
  `double dYeq_dT(double T,struct parameters* param, struct relicparam* paramrelic)`

  The first function computes Y_{eq} at a temperature `T`, and the second one its derivative.

- `double Tfo(double Wefftab[NMAX][2], int nlines_Weff, double relicmass, struct parameters* param, double d, struct relicparam* paramrelic)`

  This function computes the freeze-out temperature following Eq. (23), using the `Wefftab` generated previously.

-
  ```
  double relic_density(double Wefftab[NMAX][2], int nlines_Weff,
  struct parameters* param, struct relicparam* paramrelic)
  double relic_calculator(char name[])
  ```

  This main procedure computes the relic density using the `Wefftab` generated previously. `relic_calculator` is a container function which scans the SLHA file and computes the relic density.

-
  ```
  void Init_cosmomodel(struct relicparam* paramrelic)
  void Init_modeleff(int model_eff, struct relicparam* paramrelic)
  void Init_dark_density(double dd0, double ndd, double T_end, struct
  relicparam* paramrelic)
  void Init_dark_entropy(double sd0, double nsd, double T_end, struct
  relicparam* paramrelic)
  void Init_dark_entropySigmaD(double Sigmad0, double nSigmad, double T_end,
  struct relicparam* paramrelic)
  ```

  These procedures define the cosmological model based on which the relic density is computed. `Init_cosmomodel` has to be called first to initialize the `paramrelic` structure. To alter the QCD equation-of-state as in Appendix C, `Init_modeleff` must be called while specifying the model: `model_eff`= $1 \cdots 5$ corresponds respectively to the models A, B, B2, B3 and C developed in [24], and `model_eff`= 0 to the older model formerly used in `Micromegas` and `DarkSusy`, in which the hadrons are considered as ideal gas. If not specified, the model is set by default to B (`model_eff`= 2). `Init_dark_density` adds a dark energy density as in Eq. (33), with `dd0`=κ_ρ and `ndd`=n_ρ, `Init_dark_entropy` adds a dark entropy density as in Eq. (34), with `sd0`=κ_s and `nsd`=n_s, and `Init_dark_entropySigmaD` adds a dark entropy production as in Eq. (35), with `Sigmad0`=κ_Σ, `nSigmad`=n_Σ and `T_end`=T_r. If these routines are not called, no additional density will be added, and the calculation will be performed in the standard cosmological model.

-
  ```
  double dark_density(double T, struct relicparam* paramrelic)
  double dark_entropy(double T, struct relicparam* paramrelic)
  double dark_entropy_derivative(double T, struct relicparam* paramrelic)
  double dark_entropy_Sigmad(double T, struct relicparam* relicparam)
  ```

  These functions compute energy and entropy densities needed for the alternative cosmological models described in Appendix D.

-
  ```
  int FeynHiggs(char name[], struct parameters* param)
  int Hdecay(char name[], struct parameters* param)
  ```

  These routines call `FeynHiggs` or `Hdecay` to compute the widths and masses of the

Higgs bosons corresponding to the SLHA file `name` at the two-loop level, and puts these variables in the `param` structure.

The complete list of C procedures implemented in `SuperIso Relic` is available in `src/include.h`.

The Fortran routines can be found in `src/relic`. They have been generated automatically by a `Mathematica/FormCalc` script and they perform the computation of all squared amplitudes. Because of the large number of these routines they will not be described further here. For the `FormCalc` specific routines, we refer the reader to the `FormCalc` manual [20].

3 Compilation and installation instructions

The `SuperIso Relic` package can be downloaded from:

`http://superiso.in2p3.fr/relic`

It can be compiled in two different ways:

- the shared library compilation, which compiles the squared amplitude procedures on-the-fly, if they are needed. The initial compilation is fast, but the execution is slightly slower.
- the static library compilation. Here all the squared amplitude routines need to be compiled before running, and therefore the initial compilation process can take about an hour, and the generated executables are large. This compilation enables slightly faster execution than the shared library compilation, and is therefore intended for the scans over a large number of SUSY points.

The shared compilation is recommended.

The following main directory is created after unpacking:

`superiso_vX.X`

This directory contains the `src/` directory, in which all the source files can be found. The main directory contains also a `Makefile`, a `README`, sixteen sample main programs (such as `cmssm.c`, `nnuhm.c`, or `test_modeleff.c`) and one example of input file in the SUSY Les Houches Accord format (`example.lha`).

The paths to the different spectrum generators should be defined in the `Makefile`, if needed by the user. To set the compilation options automatically, simplily type:

```
./configure
```

To use a different C/fortran compiler, type for example:

```
./configure --with-cc=gcc --with-fc=gfortran
```

`SuperIso Relic` is written for a C compiler respecting the C99 standard and a Fortran compiler. In particular, it has been tested successfully with the GNU C and GNU Fortran Compilers and the Intel C and Intel Fortran Compilers on Linux and Mac 32-bits or 64-bits machines. Additional information can be found in the `README` file.
To compile the library, type

`make shared` or `make static`

followed by

```
make
```

This creates `libisospin.a` in `src/` and `librelic.a` in `src/relic`, and compiles `FeynHiggs` and `Hdecay`. After this step, if the paths to the spectrum generators are modified, type: `make resetpaths`.

To compile one of the sixteen programs provided in the main directory, type

`make name` or `make name.c`

where `name` can be `cmssm`, `gmsbm`, ... This generates an executable program with the `.x` extension. Note that `slha.x`, `test_modeleff.x`, `test_standmod.x`, `test_reheating.x`, `test_widthcalc.x` and `sm.x` do not need any spectrum generator.

`slha.x` calculates the implemented observables, using the parameters contained in the SLHA file whose name has to be passed as input parameter.

`amsb.x`, `cmssm.x`, `gmsb.x`, `hcamsb.x`, `mmamsb.x` and `nuhm.x` compute the observables, starting first by calculating the mass spectrum and couplings thanks to `ISAJET`, `SOFTSUSY`, `SPheno` and/or `SuSpect` within respectively the AMSB, CMSSM, GMSB, HCAMSB, MMAMSB or NUHM parameter spaces.

`cnmssm.x`, `ngmsb.x`, and `nnuhm.x` compute the observables, starting first by calculating the mass spectrum and couplings thanks to `NMSSMTools` within respectively the CNMSSM, NGMSB or NNUHM parameter spaces.

`test_modeleff.x`, `test_standmod.x`, `test_reheating.x` and `test_widthcalc.x` calculate the relic density, using the parameters contained in the SLHA file whose name has to be passed as input parameter, in the cosmological models described in the Appendices.

4 Input and output description

The input and output of the SuperIso Relic-specific main programs are described in the following. For the description of the other main programs, please refer to the SuperIso manual [16].

4.1 Alternative QCD equations of state

The program `test_modeleff.x` calculates the relic density while reading the needed parameters in the SLHA file, for the different QCD equations of state (*i.e.* alternative models of $g_{\rm eff}$ and $h_{\rm eff}$) described in Appendix C. For example, the command

```
./test_modeleff.x example.lha
```

returns

```
Dependence of the relic density on the calculation of heff and geff
For model_eff=1 (model A): omega=1.195e+01
For model_eff=2 (model B (default)): omega=1.195e+01
For model_eff=3 (model B2): omega=1.202e+01
For model_eff=4 (model B3): omega=1.187e+01
For model_eff=5 (model C): omega=1.196e+01
For model_eff=0 (old model): omega=1.171e+01
```

4.2 Effective energy and entropy densities

The program `test_standmodel.x` reads the needed parameters in the SLHA file, and calculates the relic density while adding to the standard cosmological model an effective energy density such that

$$\rho_D = \kappa_\rho \rho_{rad}(T_{BBN}) \big(T/T_{BBN}\big)^{n_\rho} \;, \tag{2}$$

and/or an effective entropy density

$$s_D = \kappa_s s_{rad}(T_{BBN}) \big(T/T_{BBN}\big)^{n_s} \;, \tag{3}$$

which modify the Early Universe properties without having observational consequences if chosen adequately [13]. A description of the model and of the related equations can be found in Appendix D. The necessary arguments to this program are[‡]:

- SLHA file name,
- κ_ρ: ratio of dark energy density over radiation energy density at BBN time (preferentially < 1),
- n_ρ: dark energy density decrease exponent (preferentially > 4),
- κ_s: ratio of dark entropy density over radiation entropy density at BBN time (preferentially < 1),
- n_s: dark entropy density decrease exponent (preferentially > 3).

[‡] The preferential values given inside the brackets correspond to cosmological models without observational consequences, *i.e.* as valid as the cosmological standard model.

Two optional parameters can be given:

- T_ρ: temperature in GeV below which the dark energy density is set to 0,
- T_s: temperature in GeV below which the dark entropy density is set to 0.

Note that $n_\rho = 4$ corresponds to a radiation-like energy density, $n_\rho = 6$ to a quintessence-like energy density and $n_\rho = 8$ to a decaying scalar field energy density. Also, $n_s = 3$ corresponds to a radiation-like entropy density and $n_s = 4$ can appear in reheating models. For example, the command

```
./test_standmod.x example.lha 1e-3 6 1e-3 4
```

returns

```
For the cosmological standard model:
omega=1.195e+01
For the specified model with dark density/entropy:
omega=5.758e+02
```

4.3 Entropy generation and reheating

The program `test_reheating.x` reads the needed parameters in the SLHA file, and calculates the relic density while adding to the standard cosmological model an effective energy density such that

$$\rho_D = \kappa_\rho \rho_{rad}(T_{BBN}) \big(T/T_{BBN}\big)^{n_\rho} \;, \tag{4}$$

and/or an effective entropy production

$$\Sigma_D = \kappa_\Sigma \Sigma_{rad}(T_{BBN}) \big(T/T_{BBN}\big)^{n_\Sigma} \;, \tag{5}$$

which modify the Early Universe properties without having observational consequences if chosen adequately [13]. A description of the model and of the related equations can be found in Appendix D. The necessary arguments to this program are:

- SLHA file name,
- κ_Σ: ratio of dark energy density over radiation energy density at BBN time (preferentially < 1),
- n_Σ: dark energy density decrease exponent (preferentially > 4),
- κ_Σ: ratio of dark entropy production over radiation entropy evolution at BBN time (preferentially < 1),
- n_Σ: dark entropy production exponent (preferentially < 0).
- T_r: reheating temperature in GeV (preferentially $> 10^{-3}$ GeV), above which the dark energy density and entropy production are set to 0.

Note that $n_\rho = 4$ corresponds to a radiation-like energy density, $n_\rho = 6$ to a quintessence-like energy density and $n_\rho = 8$ to a decaying scalar field energy density. Also, $n_\Sigma \sim -1$ corresponds to standard reheating models.
For example, the command

```
./test_reheating.x example.lha 0 0 0.1 -1 1e-3
```

returns

```
For the cosmological standard model:
omega=1.195e+01
For the specified model with reheating:
omega=1.085e+01
```

4.4 Width calculators

The program `test_widthcalc.x` calculates the relic density while reading the needed parameters in the SLHA file, using `Hdecay` at two-loop and at tree level, and `FeynHiggs` at two-loop and at tree level. For example, the command

```
./test_widthcalc.x example.lha
```

returns

```
Dependence of the relic density on the width calculator
With Hdecay: omega=1.195e+01
With Hdecay Tree: omega=1.195e+01
With FeynHiggs: omega=1.195e+01
With FeynHiggs Tree: omega=1.202e+01
```

Using the aforementioned main programs as examples, the user is encouraged to write his/her own programs in order to, for example, perform scans in a given supersymmetric scenario, or test different cosmological models.

5 Results

`SuperIso Relic` computes the relic density, and the results have been compared extensively to those of `DarkSusy` and `Micromegas`. A very good agreement has been found even at the level of the calculation of the effective annihilation rate $W_{\rm eff}$ (see Appendix A), as can be seen in Fig. 1. In general, the results of `DarkSusy`, `Micromegas`, and `SuperIso Relic` differ only by a few percents, but in some rare cases where a Higgs resonance occurs approximately at twice the mass of the LSP, the differences can be large. To avoid this problem, a very precise calculation of the masses and widths of the Higgs bosons is required, and we decided to use the two-loop calculations of `FeynHiggs` and `Hdecay` to obtain a better evaluation of the relic density in this kind of scenarios.

`SuperIso Relic` can also be used in order to constrain SUSY parameter spaces, as it provides many different observables from flavour physics as well as the relic density. It allows in particular to test easily the influence of the cosmological model by modifying for example the QCD equation-of-state (Appendix C) or the expansion rate (Appendix D), as can be seen in Fig. 2.

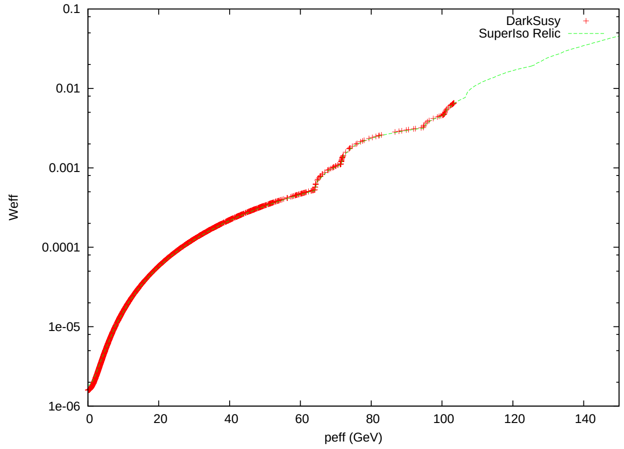

Figure 1: $W_{\rm eff}$ in function of $p_{\rm eff}$, computed with SuperIso Relic (dashed green line), and with DarkSusy (red crosses). This comparison shows an excellent agreement.

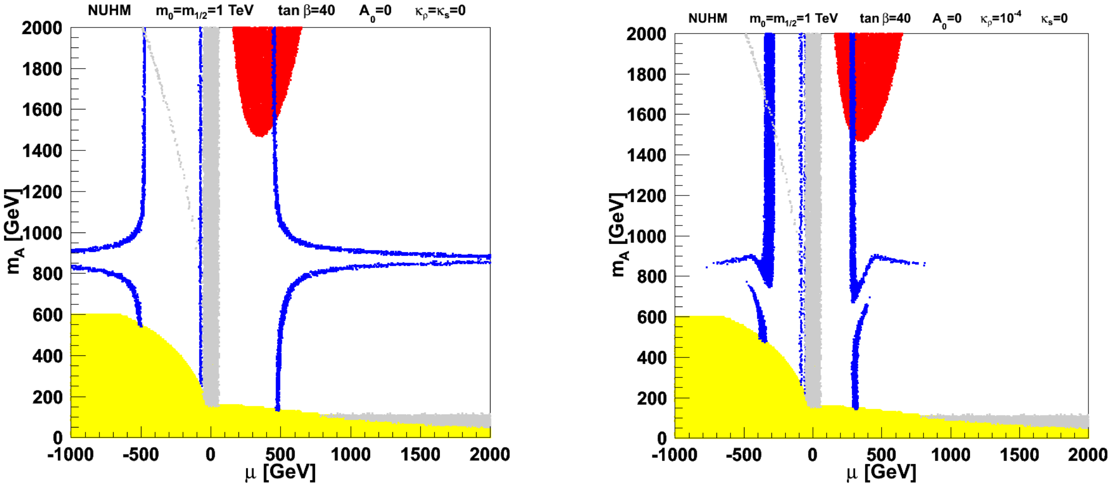

Figure 2: Constraints on the NUHM parameter plane (μ,m_A), in the standard cosmological model (left), and in presence of a tiny energy overdensity with $\kappa_\rho = 10^{-4}$ and $n_\rho = 6$ (right). The red points are excluded by the isospin asymmetry of $B \to K^*\gamma$, the gray area is excluded by direct collider limits, the yellow zone involves tachyonic particles, and the blue strips are favored by the WMAP constraints.

Appendix A Thermally averaged annihilation cross section

The computation of the thermally averaged annihilation cross section $\langle \sigma v \rangle$ is the most time consuming part of the relic density computation, as it requires the computation of the many annihilation and co-annihilation amplitudes. One can define the annihilation rate of supersymmetric particles i and j into SM particles k and l [25, 26]:

$$W_{ij\to kl} = \frac{p_{kl}}{16\pi^2 g_i g_j S_{kl}\sqrt{s}} \sum_{\text{internal d.o.f.}} \int |\mathcal{M}(ij \to kl)|^2 \, d\Omega \,, \tag{6}$$

where $\mathcal{M}$ is the transition amplitude, s is the center-of-mass energy, g_i is the number of degrees of freedom of the particle i, p_{kl} is the final center-of-mass momentum such as

$$p_{kl} = \frac{\left[s - (m_k + m_l)^2\right]^{1/2} \left[s - (m_k - m_l)^2\right]^{1/2}}{2\sqrt{s}} \,, \tag{7}$$

S_{kl} is a symmetry factor equal to 2 for identical final particles and to 1 otherwise, and the integration is over the outgoing directions of one of the final particles. Moreover, an average over initial internal degrees of freedom is performed.

We also define an effective annihilation rate W_{eff} by

$$g_{LSP}^2 p_{\text{eff}} W_{\text{eff}} \equiv \sum_{ij} g_i g_j p_{ij} W_{ij} \tag{8}$$

with

$$p_{\text{eff}}(\sqrt{s}) = \frac{1}{2}\sqrt{(\sqrt{s})^2 - 4m_{LSP}^2} \,. \tag{9}$$

In `SuperIso Relic` we compute

$$\frac{dW_{\text{eff}}}{d\cos\theta} = \sum_{ijkl} \frac{p_{ij} p_{kl}}{8\pi g_{LSP}^2 p_{\text{eff}} S_{kl}\sqrt{s}} \sum_{\text{helicities}} \left| \sum_{\text{diagrams}} \mathcal{M}(ij \to kl) \right|^2 \,, \tag{10}$$

where θ is the angle between particles i and k. We integrate over $\cos\theta$ numerically by means of the Gauss-Legendre quadrature of order 5.

Since $W_{\text{eff}}(\sqrt{s})$ does not depend on the temperature T, it can be tabulated once for each model point. It is however important to make sure that the maximum $\sqrt{s}$ in the table is large enough to include all important resonances, thresholds and coannihilation thresholds.

To improve the calculation speed, we use two different thresholds:

- a cut such that the coannihilation of SUSY particles i and j is only taken into account if

  $$m_i + m_j < \sqrt{s}_{\text{cut coann}} \,, \tag{11}$$

  where we have taken

  $$\sqrt{s}_{\text{cut coann}} = 3\, m_{LSP} \,, \tag{12}$$

- a maximum energy up to which $W_{\rm eff}(\sqrt{s})$ is calculated, such that

  $$\sqrt{s}_{max} = 2m_{LSP} - T_{fo}\log(B_\epsilon)\ , \tag{13}$$

  where $T_{fo} = 25$ GeV is a typical upper limit freeze-out temperature, and B_ϵ is the Boltzmann suppression factor limit that we fixed at 10^{-6} [27].

The thermal average of the effective cross section is then

$$\langle\sigma_{\rm eff}v\rangle = \frac{\displaystyle\int_0^\infty dp_{\rm eff}p_{\rm eff}^2 W_{\rm eff}(\sqrt{s})K_1\left(\frac{\sqrt{s}}{T}\right)}{m_{LSP}^4 T\left[\displaystyle\sum_i \frac{g_i}{g_{LSP}}\frac{m_i^2}{m_1^2}K_2\left(\frac{m_i}{T}\right)\right]^2}\ , \tag{14}$$

where K_1 and K_2 are the modified Bessel functions of the second kind of order 1 and 2 respectively. The average is performed numerically using a Gaussian integration, and the ∞ limit can be safely replaced by $p_{\rm eff}(\sqrt{s}_{max})$ using the properties of K_1.

Appendix B Cosmological Standard Model

The cosmological standard model is based on a Friedmann-Lemaître Universe filled with radiation, baryonic matter and cold dark matter, approximately flat and incorporating a cosmological constant accelerating its expansion. Before recombination, the Universe expansion was dominated by a radiation density, and therefore the expansion rate H of the Universe is determined by the Friedmann equation

$$H^2 = \frac{8\pi G}{3}\rho_{rad}\ , \tag{15}$$

where

$$\rho_{rad}(T) = g_{\rm eff}(T)\frac{\pi^2}{30}T^4 \tag{16}$$

is the radiation density and $g_{\rm eff}$ is the effective number of degrees of freedom of radiation. The computation of the relic density is based on the solution of the Boltzmann evolution equation [25, 26]

$$dn/dt = -3Hn - \langle\sigma_{\rm eff}v\rangle(n^2 - n_{\rm eq}^2)\ , \tag{17}$$

where n is the number density of all supersymmetric particles, $n_{\rm eq}$ their equilibrium density, and $\langle\sigma_{\rm eff}v\rangle$ is the thermal average of the annihilation rate of the supersymmetric particles to the Standard Model particles. By solving this equation, the density number of supersymmetric particles in the present Universe and consequently the relic density can be determined.

The ratio of the number density to the radiation entropy density, $Y(T) = n(T)/s(T)$ can be defined, where

$$s(T) = h_{\rm eff}(T)\frac{2\pi^2}{45}T^3\ . \tag{18}$$

$h_{\rm eff}$ is the effective number of entropic degrees of freedom of radiation. Combining Eqs. (15) and (17) and defining $x = m_{\rm LSP}/T$, the ratio of the LSP mass over temperature, yield

$$\frac{dY}{dx} = -\sqrt{\frac{\pi}{45G}} \frac{g_*^{1/2} m_{\rm LSP}}{x^2} \langle \sigma_{\rm eff} v \rangle (Y^2 - Y_{\rm eq}^2) \;, \tag{19}$$

with

$$g_*^{1/2} = \frac{h_{\rm eff}}{\sqrt{g_{\rm eff}}} \left(1 + \frac{T}{3 h_{\rm eff}} \frac{d h_{\rm eff}}{dT} \right) \;. \tag{20}$$

The freeze-out temperature T_f is the temperature at which the LSP leaves the initial thermal equilibrium when $Y(T_f) = (1+\delta) Y_{\rm eq}(T_f)$, with $\delta \simeq 1.5$. The relic density is obtained by integrating Eq. (19) from $x = 0$ to $m_{\rm LSP}/T_0$, where $T_0 = 2.726$ K is the temperature of the Universe today [25, 26]:

$$\Omega_{\rm LSP} h^2 = \frac{m_{\rm LSP} s(T_0) Y(T_0) h^2}{\rho_c^0} \approx 2.755 \times 10^8 \frac{m_{\rm LSP}}{1 \text{ GeV}} Y(T_0) \;, \tag{21}$$

where ρ_c^0 is the critical density of the Universe, such as

$$H_0^2 = \frac{8\pi G}{3} \rho_c^0 \;, \tag{22}$$

H_0 being the Hubble constant.

In practice, to improve the speed of the code, the freeze-out temperature T_f is determined by solving the implicit equation:

$$\frac{dY_{\rm eq}}{dx} = -\sqrt{\frac{\pi}{45G}} \frac{g_*^{1/2} m_{\rm LSP}}{x_f^2} \langle \sigma_{\rm eff} v \rangle \delta (2+\delta) Y_{\rm eq}^2 \;. \tag{23}$$

and the evolution equation (17) is only solved from $T = T_f$ to T_0, with the initial condition $Y(T_f) = (1+\delta) Y_{\rm eq}$. This method is known to provide results with less than a few percent error for the calculation of the relic density.

Appendix C QCD equation of state

To evaluate the relic density, it is necessary to know the number of effective degrees of freedom $g_{\rm eff}$ and $h_{\rm eff}$ which give access to the energy and entropy densities of radiation. To compute them, one generally assumes that above the QCD phase transition critical temperature $T_c \sim 200$ MeV, the primordial plasma is weakly interacting because of asymptotic freedom, and can therefore be treated as an ideal gas.

However, non-perturbative studies have shown that the QCD plasma departs strongly from the ideal gas behavior at high temperatures, and more realistic models have been studied in [24]. In these models, below T_c the hadronic degrees of freedom are modeled by an interacting gas of hadrons, while above T_c the quarks and gluons are taken to interact and are replaced by hadronic models. In `SuperIso Relic`, the models depicted in [24] are available, and can be selected in the routine `Init_modeleff(int model_eff, struct relicparam* paramrelic)` by setting the value of `model_eff` as given below (see subsection 2.2). The different models are:

- Model A (`model_eff`=1): ignores hadrons completely.
- Model B (`model_eff`=2): considers $T_c = 154$ MeV, and models hadrons as a gas of free mesons and hadrons, with a sharp switch to the hadronic gas at $T_{hg} = T_c$.
- Model B2 (`model_eff`=3): variation of model B constructed by scaling the pressure and energy density lattice data by 0.9.
- Model B3 (`model_eff`=4): variation of model B constructed by scaling the pressure and energy density lattice data by 1.1.
- Model C (`model_eff`=5): assumes $T_c = 185.5$ MeV, and models hadrons as a gas of free mesons and hadrons, with a sharp switch to the hadronic gas at $T_{hg} = 200$ MeV.
- Old Model (`model_eff`=0): models hadrons as an ideal gas.

An example main program is provided as `test_modeleff.c`. For more information about these models, the reader is referred to [24].

Appendix D Modified Cosmological Model

The density number of supersymmetric particles is determined by the Boltzmann equation:

$$\frac{dn}{dt} = -3Hn - \langle\sigma v\rangle(n^2 - n_{eq}^2) + N_D \ , \tag{24}$$

where n is the number density of supersymmetric particles, $\langle\sigma v\rangle$ is the thermally averaged annihilation cross-section, H is the Hubble parameter, n_{eq} is the relic particle equilibrium number density. The term N_D has been added to provide a parametrization of the non-thermal production of SUSY particles. The expansion rate H is determined by the Friedmann equation:

$$H^2 = \frac{8\pi G}{3}(\rho_{rad} + \rho_D) \ , \tag{25}$$

where ρ_{rad} is the radiation energy density, which is considered as dominant before BBN in the standard cosmological model. Following [12, 13], ρ_D is introduced as an effective dark density which parametrizes the expansion rate modification. The entropy evolution reads:

$$\frac{ds}{dt} = -3Hs + \Sigma_D \ , \tag{26}$$

where s is the total entropy density. Σ_D parametrizes here effective entropy fluctuations due to unknown properties of the Early Universe. The radiation energy and entropy densities can be written as usual:

$$\rho_{rad} = g_{\rm eff}(T)\frac{\pi^2}{30}T^4 \ , \qquad\qquad s_{rad} = h_{\rm eff}(T)\frac{2\pi^2}{45}T^3 \ . \tag{27}$$

Separating the radiation entropy density from the total entropy density, *i.e.* setting $s \equiv s_{rad} + s_D$ where s_D is an effective entropy density, the following relation between s_D and Σ_D can be derived:

$$\Sigma_D = \sqrt{\frac{4\pi^3 G}{5}}\sqrt{1+\tilde{\rho}_D}T^2\left[\sqrt{g_{\rm eff}}s_D - \frac{1}{3}\frac{h_{\rm eff}}{g_*^{1/2}}T\frac{ds_D}{dT}\right] \ . \tag{28}$$

Following the standard relic density calculation method [25,26], we introduce $Y \equiv n/s$, and Eq. (24) becomes

$$\frac{dY}{dx} = -\frac{m_{LSP}}{x^2}\sqrt{\frac{\pi}{45G}}g_*^{1/2}\left(\frac{1+\tilde{s}_D}{\sqrt{1+\tilde{\rho}_D}}\right)\left[\langle\sigma v\rangle(Y^2-Y_{eq}^2)+\frac{Y\Sigma_D - N_D}{\left(h_{\rm eff}(T)\frac{2\pi^2}{45}T^3\right)^2(1+\tilde{s}_D)^2}\right] , \tag{29}$$

where $x = m_{LSP}/T$, m_{LSP} being the mass of the relic particle, and

$$\tilde{s}_D = \frac{s_D}{h_{\rm eff}(T)\frac{2\pi^2}{45}T^3} , \qquad \tilde{\rho}_D \equiv \frac{\rho_D}{g_{\rm eff}\frac{\pi^2}{30}T^4} , \tag{30}$$

and

$$Y_{eq} = \frac{45}{4\pi^4 T^2 h_{\rm eff}}\frac{1}{(1+\tilde{s}_D)}\sum_i g_i m_i^2 K_2\left(\frac{m_i}{T}\right) , \tag{31}$$

where i runs over all supersymmetric particles of mass m_i and with g_i degrees of freedom. Following the methods described in Appendix B, the relic density can then be calculated:

$$\Omega h^2 = 2.755 \times 10^8 Y_0 m_{LSP}/\text{GeV} . \tag{32}$$

where Y_0 is the present value of Y. In the limit where $\rho_D = s_D = \Sigma_D = N_D = 0$, usual relations are retrieved. We should note here that s_D and Σ_D are not independent variables.

In `SuperIso Relic`, we adopt the parametrizations described in [12,13] for ρ_D and s_D:

$$\rho_D = \kappa_\rho \rho_{rad}(T_{BBN})\big(T/T_{BBN}\big)^{n_\rho} \tag{33}$$

and

$$s_D = \kappa_s s_{rad}(T_{BBN})\big(T/T_{BBN}\big)^{n_s} , \tag{34}$$

where T_{BBN} stands for the BBN temperature. κ_ρ and κ_s are respectively the ratio of effective dark energy/entropy density over radiation energy/entropy density, and n_ρ and n_s are parameters describing the behavior of the densities. We refer the reader to [12,13] for detailed descriptions and discussions of these parametrizations.

Another parametrization of entropy inspired by reheating scenarios is present in `SuperIso Relic`. In this reheating-like parametrization the entropy production Σ_D evolves like [15]

$$\Sigma_D = \kappa_\Sigma \Sigma_{rad}(T_{BBN})\left(\frac{T}{T_{BBN}}\right)^{n_\Sigma} . \tag{35}$$

κ_Σ is the ratio of effective dark entropy production over radiation entropy production, and n_Σ is a parameter describing the behavior of this entropy production ($n_\Sigma \sim -1$ in most reheating scenarios). The radiation entropy production reads:

$$\Sigma_{rad}(T_{BBN}) = \left(\frac{4\pi^3 G}{5}g_{\rm eff}(T_{BBN})\right)^{1/2} T_{BBN}^2 s_{rad}(T_{BBN}) . \tag{36}$$

The dark entropy density can then be calculated by:

$$s_D(T) = 3\sqrt{\frac{5}{4\pi^3 G}} h_{\text{eff}} T^3 \int_0^T dT' \frac{g_*^{1/2} \Sigma_D(T')}{\sqrt{1 + \frac{\rho_D}{\rho_{rad}} h_{\text{eff}}^2(T') T'^6}} \;. \tag{37}$$

An extra-parameter has to be introduced to remain consistent with cosmological observations: the reheating temperature T_r below which entropy production stops. This parametrization is further described in [15].

Acknowledgments

The authors would like to thank Joackim Edsjö and Geneviève Bélanger for useful discussions, and Sven Heinemeyer and Thomas Hahn for their help with FeynHiggs and FormCalc.

References

[1] J. Dunkley *et al.*, "Five-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Likelihoods and Parameters from the WMAP data", Astrophys. J. Suppl. **180**, 306 (2009) [arXiv:0803.0586].

[2] P. Skands *et al.*, "SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators", JHEP **0407**, 036 (2004) [hep-ph/0311123].

[3] B. Allanach *et al.*, "SUSY Les Houches Accord 2", Comput. Phys. Commun. **180**, 8 (2009) [arXiv:0801.0045].

[4] B.C. Allanach, "SOFTSUSY: A C++ program for calculating supersymmetric spectra", Comput. Phys. Commun. **143**, 305 (2002) [hep-ph/0104145].

[5] H. Baer *et al.*, "ISAJET 7.69: A Monte Carlo event generator for pp, $\bar{p}p$, and e^+e^- reactions", hep-ph/0312045.

[6] W. Porod, "SPheno, a program for calculating supersymmetric spectra, SUSY particle decays and SUSY particle production at e^+e^- colliders", Comput. Phys. Commun. 153, 275 (2003) [hep-ph/0301101].

[7] A. Djouadi, J.L. Kneur and G. Moultaka, "SuSpect: A Fortran code for the supersymmetric and Higgs particle spectrum in the MSSM", Comput. Phys. Commun. 176, 426 (2007) [hep-ph/0211331].

[8] U. Ellwanger and C. Hugonie, "NMSPEC: A Fortran code for the sparticle and Higgs masses in the NMSSM with GUT scale boundary conditions", Comput. Phys. Commun. 177, 399 (2007) [hep-ph/0612134]; "NMHDECAY 2.0: An Updated program for sparticle masses, Higgs masses, couplings and decay widths in the NMSSM", Comput. Phys. Commun. 175, 290 (2006) [hep-ph/0508022].

[9] P. Gondolo *et al.*, "DarkSUSY: Computing supersymmetric dark matter properties numerically", JCAP **0407**, 008 (2004) [astro-ph/0406204].

[10] H. Baer, C. Balazs and A. Belyaev, “Neutralino relic density in minimal supergravity with coannihilations”, JHEP **0203**, 042 (2002) [hep-ph/0202076].

[11] G. Bélanger, F. Boudjema, A. Pukhov and A. Semenov, “micrOMEGAs: Version 1.3”, Comput. Phys. Commun. **174**, 577 (2006) [hep-ph/0405253].

[12] A. Arbey and F. Mahmoudi, “SUSY constraints from relic density: High sensitivity to pre-BBN expansion rate”, Phys. Lett. B**669**, 46 (2008) [arXiv:0803.0741].

[13] A. Arbey and F. Mahmoudi, “SUSY Constraints, Relic Density, and Very Early Universe”, JHEP**1005** (2010), 051 [arXiv:0906.0368].

[14] A. Arbey and F. Mahmoudi, “LHC and ILC Data and the Early Universe Properties”, Nuovo Cim. C**33**, 151 (2010) [arXiv:1002.4096].

[15] A. Arbey, A. Deandrea and A. Tarhini, “Anomaly mediated SUSY breaking scenarios in the light of cosmology and in the dark (matter)”, JHEP **1105**, 078 (2011) [arXiv:1103.3244].

[16] F. Mahmoudi, “SuperIso: A Program for calculating the isospin asymmetry of $B \to K^* \gamma$ in the MSSM”, Comput. Phys. Commun. **178**, 745 (2008) [arXiv:0710.2067];
F. Mahmoudi, “SuperIso v2.3: A Program for calculating flavour physics observables in Supersymmetry”, Comput. Phys. Commun. **180**, 1579 (2009) [arXiv:0808.3144];
F. Mahmoudi, “SuperIso v3.0, flavor physics observables calculations: Extension to NMSSM”, Comput. Phys. Commun. **180**, 1718 (2009), `http://superiso.in2p3.fr` .

[17] S. Wolfram, “The Mathematica Book”, ISBN 1579550223.

[18] A. Semenov, “LanHEP: A Package for the automatic generation of Feynman rules in field theory. Version 3.0”, Comput. Phys. Commun. **180**, 431 (2009) [arXiv:0805.0555].

[19] T. Hahn, “Generating Feynman Diagrams and Amplitudes with FeynArts 3”, Comput. Phys. Commun. **140**, 418 (2001) [hep-ph/0012260];
T. Hahn and C. Schappacher, “The Implementation of the Minimal Supersymmetric Standard Model in FeynArts and FormCalc”, Comput. Phys. Commun. **143**, 54 (2002) [hep-ph/0105349].

[20] T. Hahn and M. Perez-Victoria, “Automatized One-Loop Calculations in 4 and D dimensions”, Comput. Phys. Commun. **118**, 153 (1999) [hep-ph/9807565];
T. Hahn and M. Rauch, “News from FormCalc and LoopTools”, hep-ph/0601248.

[21] S. Heinemeyer, W. Hollik and G. Weiglein, “FeynHiggs: A Program for the calculation of the masses of the neutral CP even Higgs bosons in the MSSM”, Comput. Phys. Commun. **124**, 76 (2000) [hep-ph/9812320].

[22] A. Djouadi, J. Kalinowski and M. Spira, “HDECAY: A program for Higgs boson decays in the standard model and its supersymmetric extension”, Comput. Phys. Commun. **108**, 56 (1998) [arXiv:hep-ph/9704448].

[23] K. Nakamura *et al.* [Particle Data Group], “Review of particle physics”, J. Phys. G**37**, 075021 (2010).

[24] M. Hindmarsh and O. Philipsen, “WIMP dark matter and the QCD equation of state”, Phys. Rev. D**71**, 087302 (2005) [hep-ph/0501232].

[25] P. Gondolo and G. Gelmini, “Cosmic abundances of stable particles: Improved analysis”, Nucl. Phys. B**360**, 145 (1991).

[26] J. Edsjö and P. Gondolo, “Neutralino relic density including coannihilations”, Phys. Rev. D**56**, 1879 (1997) [hep-ph/9704361].

[27] G. Bélanger, F. Boudjema, A. Pukhov and A. Semenov, “MicrOMEGAs: A Program for calculating the relic density in the MSSM”, Comput. Phys. Commun. **149**, 103 (2002) [hep-ph/0112278].