GALPROP C++ v.41: *Explanatory Supplement*

Andrew W. Strong[*]

Max-Planck-Institut für extraterrestrishe Physik,
Postfach 1603, D-85740 Garching, Germany

&

Igor V. Moskalenko[†]

NASA/Goddard Space Flight Center, Code 660,
Greenbelt, MD 20771, U.S.A.

September 20, 2001

Abstract

These notes provide a description of the current version of the cosmic ray propagation code.

[*]e-mail: `aws@mpe.mpg.de`
[†]e-mail: `imos@milkyway.gsfc.nasa.gov`

Contents

1 Introduction

The origin of cosmic rays have been intriguing scientists since 1912 when V. Hess carried out his famous balloon flight to measure the ionisation rate in the upper atmosphere. The cosmic rays are energetic particles, which come to us from outer space, and are measured either through satellites, balloons, or Earth based experiments. The spectrum of cosmic rays can be approximately described by a single power law with index -3 from ~ 10 GeV to the highest energies ever observed $\sim 10^{20}$ eV. The only feature observed below 10^{18} eV is a knee around 10^{15} eV. Because of this featureless spectrum, it is believed that cosmic-ray production and propagation is governed by the same mechanism over decades of energy, the same mechanism at least works below the knee and the same or another one works above the knee. Meanwhile the origin of the cosmic rays spectrum is not still understood.

Galactic cosmic rays are important part of the interstellar medium. The energy density of relativistic particles is about 1 eV cm^{-3} and is comparable to the energy density of interstellar radiation field, magnetic field, and turbulent motions of the interstellar gas. This makes cosmic rays one of the essential factors determining the dynamics and processes in the interstellar medium.

The sources of cosmic rays are believed to be supernovae and supernova remnants, pulsars, compact objects in close binary systems, and stellar winds. Observations of X-ray and γ-ray emission from these objects reveal the presence of energetic particles thus testifying to efficient acceleration processes near these objects. Particles accelerated near the sources propagate tens of millions years in the interstellar medium where they lose or gain energy, their initial spectra and composition change, they produce secondary particles and γ-rays. The distruction of primary nuclei via spallation gives rise to secondary nuclei and isotopes which are rare in nature, antiprotons, and charged pions that decay producing secondary positrons and electrons.

The variety of isotopes in cosmic rays allows one to study different aspects of their acceleration and propagation in the interstellar medium as well as the source composition. Stable secondary nuclei tell us about the diffusion coefficient and Galactic winds (convection) and/or re-acceleration in the interstellar medium (2nd order Fermi acceleration mechanism). Long-lived radioactive secondaries allow one to constrain global Galactic properties such as Galactic halo size. Abundances of K-capture isotopes, which being stopped in the interstellar gas would decay via electron K-capture, allow one to probe the gas density and acceleration time scale. All these together allow us in principle to build a model of particle acceleration and propagation in the Galaxy.

Such a model is however incomplete. The whole of our knowledge is based on measurements done only at one point on the outskirts of the Galaxy, the solar system, and the assumption that particle spectra and composition are (almost) the same at every point of the Galaxy. The latter may not necessarily be correct. γ-rays are able to deliver the information directly from distant regions thus complementing that obtained from cosmic-ray measurements. Some part of the diffuse γ-rays is produced in energetic nucleons interactions with gas via neutral pion production, another is produced by electrons via inverse Compton scattering and bremsstrahlung. These processes are dominant in different parts of the spectra of γ-rays, therefore, if deciphered the γ-ray spectrum can provide information about the large-scale spectra of nucleonic and leptonic components of cosmic rays.

To extract information which is contained in cosmic ray abundances and γ-ray fluxes one needs to develop a model of particle production and propagation in the Galaxy. Though the basic features of particle diffusion in the Galaxy seem to be well-established, the continuous flow of new more and more accurate data from space, balloon and ground based experiments motivates further development of models. Analytical and semi-analytical models are able to interpret one or only a few features and often fail when they try to deal with the whole variety of data. Therefore more realistic and consistent models are required which would be able to incorporate many processes and astrophysical data of many different kinds simultaneously: nuclear reaction networks and nuclear cross sections, production of antiprotons, positrons, γ-rays and synchrotron emission, realistic gas distribution, radiation field distribution and spectrum, energy losses, convection, diffusive re-acceleration etc.

In several years new missions planned for cosmic ray experiments will tremendously increase the quality and accuracy of cosmic-ray data making new progress impossible without highly developed models. Data will continue to flow from the high resolution detectors on Ulysses, Advanced Composition Explorer and Voyager space missions. During the next few years there will be several flights of ballon-borne high resolution spectrometers that will extend our knowledge of antiproton, positron and electron spectra in cosmic rays. Several more high resolution space experiments are planned to be launched in the next 2–3 years, e.g., PAMELA to measure antiprotons, positrons, electrons, and isotopes H through C over the energy range of 0.1 to 200 GeV, Alpha Magnetic Spectrometer will measure particle and nuclear spectra to TeV energies. The previous γ-ray mission EGRET, one of the four detectors on board of the Compton Gamma-Ray Observatory, gave a detailed map of the Galactic diffuse emission in the range 30 MeV – 10 GeV which traces the cosmic ray distribution in the Galaxy and possibly the acceleration sites of cosmic rays. The future GLAST mission capability covers the range 10 MeV – 300 GeV with a sensivity two orders of magnitude better.

Clearly, a detailed model of cosmic ray propagation in the Galaxy should supplement the high quality data obtained by the spacecraft and balloon-borne missions, providing support for the necessary interpretation and analysis.

2 General Principles

2.1 Transport equation

GALPROP solves the transport equation with a given source distribution and boundary conditions for all cosmic-ray species. This includes Galactic wind (convection), diffusive reacceleration in the interstellar medium, energy losses, nuclear fragmentation, and decay. The numerical solution of the transport equation is based on a Crank-Nicholson (Press et al., 1992) implicit second-order scheme. The spatial boundary conditions assume free particle escape. Since we have a 3-dimensional (R, z, p) or 4-dimensional (x, y, z, p) problem (spatial variables plus momentum) we use "operator splitting" to handle the implicit solution.

The propagation equation is written in the form:

$$\frac{\partial \psi}{\partial t} = q(\vec{r}, p) + \vec{\nabla} \cdot (D_{xx}\vec{\nabla}\psi - \vec{V}\psi) + \frac{\partial}{\partial p} p^2 D_{pp} \frac{\partial}{\partial p} \frac{1}{p^2} \psi - \frac{\partial}{\partial p} \left[\dot{p}\psi - \frac{p}{3}(\vec{\nabla} \cdot \vec{V})\psi\right] - \frac{1}{\tau_f}\psi - \frac{1}{\tau_r}\psi \ , \tag{1}$$

where $\psi = \psi(\vec{r}, p, t)$ is the density per unit of total particle momentum, $\psi(p)dp = 4\pi p^2 f(\vec{p})$ in terms of phase-space density $f(\vec{p})$, $q(\vec{r}, p)$ is the source term, D_{xx} is the spatial diffusion coefficient, $\vec{V}$ is the convection velocity, reacceleration is described as diffusion in momentum space and is determined by the coefficient D_{pp}, $\dot{p} \equiv dp/dt$ is the momentum loss rate, τ_f is the time scale for fragmentation, and τ_r is the time scale for the radioactive decay. The details of the numerical scheme is described in § 3.

For a given halo size the diffusion coefficient as a function of momentum and the reacceleration or convection parameters is determined by boron-to-carbon ratio data. The spatial diffusion coefficient is taken as $D_{xx} = \beta D_0(\rho/\rho_0)^\delta$ if necessary with a break ($\delta = \delta_{1,2}$ below/above rigidity ρ_0), where the factor β $(= v/c)$ is a consequence of a random-walk process. For the case of reacceleration the momentum-space diffusion coefficient D_{pp} is related to the spatial coefficient D_{xx} (Berezinskii et al., 1990; Seo & Ptuskin, 1994), where $\delta = 1/3$ for a Kolmogorov spectrum of interstellar turbulences. The convection velocity (in z-direction only) $V(z)$ is assumed to increase linearly with distance from the plane ($dV/dz > 0$ for all z); this implies a constant adiabatic energy loss. The linear form for $V(z)$ is consistent with cosmic-ray driven MHD wind models (Zirakashvili et al., 1996).

The distribution of cosmic-ray sources (Strong & Moskalenko, 1998) is chosen to reproduce the cosmic-ray distribution determined by analysis of EGRET γ-ray data (Strong & Mattox, 1996). The injection spectrum of nucleons is assumed to be a power law in momentum, $dq(p)/dp \propto p^{-\gamma}$. Energy losses (Strong & Moskalenko, 1998) for nucleons by ionization and Coulomb interactions are included, and for electrons by ionization, Coulomb interactions, bremsstrahlung, inverse Compton, and synchrotron. The total magnetic field distribution is adjusted to match the 408 MHz synchrotron longitude and latitude distributions. This is in agreement with interstellar field estimates (Broadbent et al., 1990) and other magnetic field models (e.g., Heiles, 1996; Vallée, 1996).

2.1.1 Interstellar hydrogen distribution

The interstellar hydrogen distribution uses H I and CO surveys and information on the ionized component (Moskalenko et al., 2001b); the helium fraction of the gas is taken as 0.11 by number. The H_2 gas number density is defined in the form of table (Bronfman et al., 1988), which is interpolated linearly, and the conversion factor is taken as $X \equiv n_{H_2}/\epsilon_{CO} = 1.9 \times 10^{20}$ mols. cm^{-2}/(K km s^{-1}) (Strong & Mattox, 1996). The H I gas number density in the Galactic plane is defined by a table (Gordon & Burton, 1976) which is renormalized to agree with the total integral perpendicular to the plane by Dickey & Lockman (1990). The z-dependence is calculated using the approximation by Dickey & Lockman (1990) for $R < 8$ kpc, using the approximation by

Cox et al. (1986) for $R > 10$ kpc, and interpolated in between. The ionized component H II (atom cm^{-3}) is calculated using a cylindrically symmetrical model (Cordes et al., 1991).

2.1.2 Interstellar radiation field (ISRF)

For calculation of the spectrum of γ-rays arising from inverse Compton scattering and electron energy losses, the full ISRF as function of (R, z, ν) is required, which is not available in the literature. Our ISRF calculation uses emissivities based on stellar populations and dust emission. The infrared emissivities per atom of H I and H_2 are based on COBE/DIRBE data from Sodrowski et al. (1997), combined with the distribution of H I and H_2. The spectral shape is based on the silicate, graphite and PAH synthetic spectrum using COBE data from Dwek et al. (1997). For the distribution of the old stellar disk component we use the model of Freudenreich (1998) based on the COBE/DIRBE few micron survey. The stellar luminosity function is taken from Wainscoat et al. (1992). For each stellar class the local density and absolute magnitude in standard optical and near-infrared bands is given, and these are used to compute the local stellar emissivity by interpolation in wavelength. The z-scaleheight for each class and the spatial functions (disk, halo, rings, arms) given by Wainscoat et al. (1992) then give the volume emissivity as a function of position and wavelength. All their main-sequence and AGB types were explicitly included.

2.1.3 Gamma rays

Gas-related γ-ray intensities are computed from the emissivities as a function of (R, z, E_γ) using the column densities of H I and H_2 for Galactocentric annuli based on 21-cm and CO surveys. Neutral pion production is calculated using a formalism by Dermer (1986a,b) as described in Moskalenko & Strong (1998); bremsstrahlung is calculated using a formalism by Koch & Motz (1959) as described in Strong et al. (2000). The inverse Compton scattering is treated using the formalism for an anisotropic radiation field developed by Moskalenko & Strong (2000a), this uses the interstellar radiation field calculations as described above.

2.1.4 Nuclei H to Ni

In the new version, the code is updated to include the cross-section measurements and energy dependent fitting functions (Strong & Moskalenko, 2001). The nuclear reaction network is built using the Nuclear Data Sheets. Currently, the isotopic cross section database consists of more than 2000 points collected from sources published in 1969–1999. This includes a critical re-evaluation of some data and cross checks. The isotopic cross sections are calculated using the author's fits to major beryllium and boron production cross sections. Other cross sections are calculated using phenomenological approximations by Webber et al. (1990) (code `WNEWTR.FOR` version of 1993) and/or Silberberg and Tsao (code `YIELDX_011000.FOR` version of 2000) renormalized to the data where it exists. The cross sections on the He target are calculated using a parametrization by Ferrando et al. (1988). For pp and pA inelastic cross sections we adapted parametrizations by Tan & Ng (1983a) and Letaw et al. (1983). The reaction network is solved starting at the heaviest nuclei (i.e., ^{64}Ni). The propagation equation is solved, computing all the resulting secondary source functions, and then proceeds to the nuclei with $A - 1$. The procedure is repeated down to $A = 1$. Our preliminary results for all cosmic ray species $Z \leq 28$ are given in Strong & Moskalenko (2001) and re-evaluation of the radioactive isotopes of Be, Al, Cl, Mn is given in Moskalenko et al. (2001a).

2.1.5 Secondary antiprotons

The code calculates production and propagation of secondary antiprotons as described in Moskalenko et al. (1998) and Moskalenko et al. (2001b). Antiproton production in pp-collisions has been calculated using the parametrization of the invariant $\bar{p}$-production cross section given by Tan & Ng (1983b). The antiproton production by nuclei with $Z \geq 2$ is calculated using effective nuclear factors; the latter computed using the Monte Carlo event generator DTUNUC (Roesler et al., 1998; Simon et al., 1998) or scaling factors similar to Gaisser & Schaefer (1992). For the inelastic proton cross sections we adapted parametrizations by Tan & Ng (1983a) and Letaw et al. (1983). The total $\bar{p}p$ inelastic cross section has been calculated using a fit from Tan & Ng (1983a) and parametrization by Groom et al. (2000). The antiproton absorption cross section on nuclear targets is calculated following Moiseev & Ormes (1997). Inelastically scattered antiprotons are treated as a separate "tertiary" component.

2.1.6 Secondary positrons and electrons

Secondary positrons and electrons in cosmic rays are the final product of decay of charged pions and kaons which in turn created in collisions of cosmic-ray particles with gas. Pion production in *pp*-collisions is considered following a method developed by Dermer (1986a,b), which combines isobaric (Stecker, 1970) and scaling (Badhwar et al., 1977; Stephens & Badhwar, 1981) models of the reaction. Secondary positron and electron production is computed as described in Moskalenko & Strong (1998), that includes a critical reevaluation of the charged pion and kaon decay calculations. Primary electrons are computed in the same propagation model.

2.2 Working quantities

The nuclei are aligned on the same kinetic energy per nucleon E_{kin} since this simplifies the secondary-to-primary computation, where primaries produce secondaries of the same E_{kin}. However the basic CR density used has *units* of density per total momentum p since this is natural for propagation. The actual units used internally are $\frac{c}{4\pi}n(p)$, where $n(p) = dn/dp$ in units of cm^{-3} MeV^{-1}.

When the *flux* $I(E_{kin})$ in cm^{-2} sr^{-1} s^{-1}(MeV/nucleon)$^{-1}$ is necessary, it can be simply obtained from

$$I(E_{kin}) = \frac{\beta c}{4\pi}\frac{dn}{dp}\frac{dp}{dE_{kin}} = \frac{c}{4\pi}n(p)A, \tag{2}$$

where A is the nucleus mass number. This follows from $dp = \frac{A}{\beta}dE_{kin}$. The combined requirements of transport and fragmentation are thus elegantly met. The normal units for presentation of CR data are cm^{-2} sr^{-1} s^{-1}(MeV/nucleon)$^{-1}$, and with this scheme the conversion is trivial. The nucleus energy scales are logarithmic in E_{kin}.

The *output* nuclei spectra are fluxes[1] as defined above, multiplied by E_{kin}^2. The spectra are at $z = 0$ as a function of R in kpc. The flux spectra are directly comparable with experimental data at $R = R_0$.

2.3 Abundances and normalization

All calculations are done treating $n(p)$ as the basic quantity; the final step is to normalize to the absolute proton or electron fluxes given in the *galdef* file. All other primary and secondary nuclei follow the same normalization factor as for protons since source abundances relative to protons are specified in the *galdef* file. The global normalization to the absolute proton flux is applied at the end of the entire propagation, in `nuclei_normalize`. Only for output is the *flux* computed for all nuclei, in routine `store_gcr`. The π^0-decay emissivities depend on p, He and the bremsstrahlung and IC emissivities depend on electrons. The nuclei and electron normalizations are therefore done before computing gammas and synchrotron.

2.4 Secondary production

For computation of gamma-ray emissivities and source functions for secondary positrons, electrons and antiprotons, the integral over nucleon energies is required.

The nucleus energy scales are uniform in $\log E_{kin}$. It is easy therefore to replace integration over kinematic variable with summation over $\Delta(\log E_{kin})$. In case of secondary source function for, e.g., antiprotons it can be calculated as following

$$q(p) = \beta c n_H \int dp' \frac{d\sigma(p,p')}{dp} n(p'), \tag{3}$$

where n_H is the gas density (in this case pure Hydrogen), $d\sigma(p,p')/dp$ is the production cross section, $n(p')$ is the CR proton *density*, and p' is the total momentum of a nucleus. Substitution of dp' with $d(\log E_{kin})$ gives:

$$\begin{aligned} q(p) &= c n_H A \int d(\log E_{kin})\, E_{kin}\, n(E_{kin}) \frac{d\sigma(p,E_{kin})}{dp} \\ &= c n_H A\, \Delta(\log E_{kin}) \sum_{E_{kin}} E_{kin}\, n(E_{kin}) \frac{d\sigma(p,E_{kin})}{dp}, \end{aligned} \tag{4}$$

where we used $dp' = \frac{1}{\beta} A E_{kin} d(\log E_{kin})$.

[1] Note this differs from the f90 version which outputs $\frac{c}{4\pi}p^2 n(p)$.

3 The galprop Code

galprop is the main program which calls the routines to read the *galdef* file, read the nuclear data arrays, fill in the gas and radiation field distributions in the Galaxy, create cosmic rays, propagate particles, and finally store the output arrays. Below given are the routines' names and their short description.

The code consists of C++ and FORTRAN 77 files which are compiled separately and then linked to form a single executable file, and data files. Under the Linux an example of the corresponding commands will be:

```
>f77 -c *.f
>g++ -c -O5 -I/software/lheasoft/release/Linux_2.2_i686/include *.cc
>g++ *.o ../libs/*.a -L/usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66
  -L/usr/i386-redhat-linux/lib -lg2c -lm -lgcc -lc
```

The last command is too long to be placed on one line, we therefore broke it into two lines. Those on DEC Alpha will be:

```
>f77 -c *.f
>g++ -c -O5 -I/software/lheasoft/release/OSF1_4.0_alpha/include *.cc
>g++ *.o ../libs/*.a -lUfor -lfor -lFutil -lm -lots -lgcc -lc -lgcc
```

The code uses `cfitsio` package (`libcfitsio.a`).

Header files:

```
Configure.h Galdef.h    constants.h       galprop_classes.h
Distribution.h Particle.h  fort_interface.h  global.h
Galaxy.h Spectrum.h  galprop.h
```

C++ files:

```
B_field_model.cc     isrf_energy_density.cc
Configure.cc     kinematic.cc
D_pp.cc     nH.cc
Distribution.cc     nuc_package.cc
Galaxy.cc     nuclei_normalize.cc
Galdef.cc     nucleon_cs.cc
He_to_H_CS.cc     print_BC.cc
IC_anisotropy_factor.cc     propagate_particles.cc
IC_cross_section.cc     propel.cc
Kcapture_cs.cc     propel_diagnostics.cc
Particle.cc     protri.cc
cr_luminosity.cc     read_COR.cc
create_SNR.cc     read_HIR.cc
create_galaxy.cc     read_gcr.cc
create_gcr.cc     read_isrf.cc
create_transport_arrays.cc     sigma_boron_dec_heinbach_simon.cc
decayed_cross_sections.cc     source_SNR_event.cc
e_KN_loss.cc     source_SNR_event_vec.cc
electrons_normalize.cc     source_distribution.cc
energy_losses.cc     store_IC_skymap.cc
fort_interface1.cc     store_IC_skymap_comp.cc
fort_interface2.cc     store_bremss_emiss.cc
galprop.cc     store_bremss_ionized_skymap.cc
gauss.cc     store_bremss_skymap.cc
gen_IC_emiss.cc     store_gcr.cc
gen_IC_skymap.cc     store_gcr_full.cc
gen_bremss_emiss.cc     store_ionization_rate.cc
```

```
gen_bremss_ionized_skymap.cc     store_pi0_decay_emiss.cc
gen_bremss_skymap.cc     store_pi0_decay_skymap.cc
gen_ionization_rate.cc     store_synch_skymap.cc
gen_isrf_energy_density.cc     test_Distribution.cc
gen_pi0_decay_emiss.cc     test_Particle.cc
gen_pi0_decay_skymap.cc     test_float_accuracy.cc
gen_secondary_antiproton_source.cc  test_isotope_cs.cc
gen_secondary_positron_source.cc    test_nH.cc
gen_secondary_proton_source.cc     test_source_SNR_event.cc
gen_secondary_source.cc     test_suite.cc
gen_synch_emiss.cc     tridag.cc
gen_synch_skymap.cc     tridag_double.cc
gen_tertiary_antiproton_source.cc   tridag_ext.cc
global.cc     tridag_sym.cc
ionization_bethe.cc     tridag_sym_ext.cc
```

FORTRAN files:

```
WNEWTR_FUNC_aws.f     antiproton.f   cfactor.f e_loss_compton.f  pp_meson.f
YIELDX_011000_imos.f  bremss_spec.f  crn6.f inter.f   synchrotron.f
```

Data files:

```
barpol.dat          eval_iso_cs.dat  isotope_cs.dat  nucdata.dat  p_cs_fits.dat
WNEWTR_082693.CDR
```

4 Description of Basic Routines

4.1 galdef.read

Reads the *galdef* file and assigns initial parameter values.

4.2 read_nucdata & set_sigma_cc

4.3 create_galaxy

Creates spatial grid and defines distributions of gas (H_2, HI, HII — nH2, nH2_av, nHI, nHI_av, nHII, nHII_av), magnetic field (B_field_model), reads the interstellar radiation field from a file (read_isrf) and defines the ISRF energy density (gen_isrf_energy_density), creates supernova remnants (create_SNR), and defines the skymap parameters for gamma rays and synchrotron emission.

4.3.1 nH2, nH2_av, nHI, nHI_av, nHII, nHII_av

The routines nH2, nHI define the gas number densities in the form of a table, which gives the gas densities in the Galactic plane. The extention of the gas distribution to an arbitrary height above the plane is made using some analytical approximations. The routine nHII uses only analytical approximations.

The routines nH2_av, nHI_av, nHII_av provide a calculation of an average gas density over a step in the z-grid using a smaller step.

nH2: The routine calculates H_2 number density in mol cm^{-3}

$$n_{H_2}(R,Z) = \epsilon_0(R) X\, e^{-\ln 2 (Z-z_0)^2/z_h^2} \quad cm^{-2}\ kpc^{-1}, \tag{5}$$

where $\epsilon_0(R)$ (K km s$^-1$) – CO volume emissivity, and $z_0(R)$, $z_h(R)$ – the height scale and width are taken from Bronfman et al. (1988, Table 3/Cols 4,7,10), and $X = n_{H_2}/n_{CO} = 1.9 \times 10^{20}$ is the conversion factor taken from Strong & Mattox (1996).

nHI: The routine calculates H_I number density in atom cm^{-3}. The relative distribution is from Gordon & Burton (1976, Table 1), but renormalized to agree with Dickey & Lockman (1990), where on page 252 they give their best model for the z-distribution and state the total integral perpendicular to the plane = 6.2×10^{20} cm^{-2}:

$$n_{H_I}(R, Z) = Y(R) f(Z), \tag{6}$$

where $Y(R)$ is the renormalized distribution by Gordon & Burton ($R < 16$ kpc), and $f(Z)$ is the z-dependence which is calculated following Dickey & Lockman for $R < 8$ kpc, following Cox et al. (1986) for $R > 10$ kpc, and interpolated in between. For $R > 16$ kpc the exponential tail is assumed with scale high 3 kpc.

nHII: The ionized component currently calculated using cylindrically symmetrical model by Cordes et al. (1991, Eq.(6) and Table 1). More sophisticated 3D model for the inner Galaxy is developed by Taylor & Cordes (1993) and outer Galaxy by Lazio & Cordes (1998a). Galactic center region has been considered by Lazio & Cordes (1998b). These latter models can be used in future for 3D calculations.

4.3.2 `read_isrf`

Reads 3D interstellar radiation field (ISRF) from a file, curreny `isrf_interp_04_000015`. The units of the ISRF stored is $[\lambda U_\lambda] = \mu$ eV cm^{-3} μ^{-1}. In the routine `read_isrf` the units are changed to $[\nu U_\nu] =$ Hz eV cm^{-3} Hz^{-1}.

4.3.3 `gen_isrf_energy_density`

4.4 `create_gcr`

4.5 `propagate_particles`

The routine organizes a loop over the cosmic ray species: calls a routine to create transport arrays, generates secondary source, calls `propel` to propagate particles and finally normalizes nuclei and electrons.

4.5.1 `create_transport_arrays`

Generates arrays for use by `propel` for the given species: assigns primary source function, diffusion coefficient, fragmentation rate, momentum loss rate, decay rate. Normalizes source spectra according to abundances, interpreting as flux at the same E_{kin} (*kinetic energy per nucleon*) or at the same p_1 (*momentum per nucleon*).

Normalization of the nucleon primary source function: The source function is assumed to be a power-law in rigidity ρ with reference value at ρ_{br}

$$q_A(p) = c_A \left(\frac{\rho}{\rho_{br}}\right)^{-\gamma}, \tag{7}$$

and the source abundance is defined as the ratio

$$X = \frac{Q_A(p_1)}{Q_1(p_1)} = \frac{A q_A(p_1)}{q_1(p_1)} \tag{8}$$

where q_1 refers to protons.

Since $\rho/\rho_{br} = \frac{1}{Z}(p_A/\rho_{br})$ and $p_A = A p_1$, one can get

$$\begin{aligned} X &= \frac{A c_A}{c_1}\left(\frac{p_A}{Z\rho_{br}}\right)^{-\gamma}\left(\frac{p_1}{\rho_{br}}\right)^{\gamma} = \frac{c_A A^{1-\gamma} Z^{\gamma}}{c_1}, \\ \frac{c_A}{c_1} &= X A^{\gamma-1} Z^{-\gamma}. \end{aligned} \tag{9}$$

The factor c_A/c_1 is applied at the end of the source function generation.

The global normalization to the absolute proton flux is applied at the end of the entire propagation, in `nuclei_normalize`. A reference list of abundances relative to protons can be found in Meyer, Drury, & Ellison (1998, Table 1) and is used as the baseline set in the standard *galdef* files.

ρ_{br} can be used as a break rigidity with different exponents above and below, and the formulation ensures continuity at this rigidity. More general method, maybe to be implemented later, valid for any spectrum (above is only for power-law in rididity).

Assigning fragmentation rate

The fragmentation rate is assigned using the total cross section calculation on proton target (`in nucleon_cs_cc`), cross section on He target is calculated using phenomenological scaling (`He_to_H_CS`).

In case of protons "fragmentation" means inelastic scattering in which particle losses a substantial part of its energy. In case of antiprotons "framentation" means inelastic scattering plus annihilation. Finally, the fragmentation rate is calculated in every spatial and energy point as:

$$F = \beta c(n_{H_2} + n_{HI} + n_{HII})(\sigma_p + \sigma_{He}\frac{n_{He}}{n_H}), \quad s^{-1}, \tag{10}$$

where $n_H = n_{H_2} + n_{HI} + n_{HII}$.

Energy losses

The energy losses for nucleons, ionization and Coulomb losses, are calculated in `nucleon_loss`. Those for electrons, ionization, Coulomb losses, bremsstrahlung, syncrotron, and Compton losses (Thompson scattering), are calculated in `electron_loss`. The routine `e_KN_loss` allows to calculate Klein-Nishina energy losses.

Klein-Nishina energy losses are calculated using ISRF calculated separately and read by routine `read_isrf`. The ISRF units are $[\nu U_\nu]$ = Hz eV cm^{-3} Hz^{-1}. The energy losses in every spacial point can be calculated as

$$\frac{dp}{dt} = \int d\nu\, \frac{U_\nu}{\hbar\nu}\frac{dp}{dt}(\nu,\epsilon) = \int d(\log\nu)\, \frac{\nu U_\nu}{\hbar\nu}\frac{dp}{dt}(\nu,\gamma), \quad eV\ s^{-1}, \tag{11}$$

where γ is the electron Lorentz-factor, and $dp(\nu,\gamma)/dt$ is calculated in `e_loss_compton`.

4.5.2 `gen_secondary_source`

Combines calculation of all the secondary source functions.

`gen_secondary_positron_source:` secondary e^+ and e^-

The routine `PP_MESON` written in `FORTRAN-77` is designed to calculate the secondary positron (or sec. electron) production spectrum vs. energy (barn/GeV). Positron/electron energy and total nucleus momentum are used as input parameters as well as beam and target nuclei atomic numbers.

The secondary positron/electron source function as used in *galprop* is defined as following (cm^{-3} s^{-1} sr^{-1} MeV^{-1}):

$$q_e(E_{tot}) = \frac{c}{4\pi}\frac{dn(p)}{dt} = \frac{c}{4\pi}\sum_{i=H,He} n_i \sum_j \int dp'\, \beta n_j(p')\frac{d\sigma_{ij}(E_{tot},p')}{dE_{tot}}, \tag{12}$$

where n_i is the gas density, $d\sigma_i j(E_{tot},p')/dE_{tot}$ is the production cross section, $n_j(p')$ is the CR species *density*, and p' is the total momentum of a nucleus. Substitution of dp' with $d(\log E'_{kin})$ gives:

$$\begin{aligned} q_e(E_{tot}) &= \frac{c}{4\pi}A\sum_{i=H,He} n_i \int d(\log E'_{kin})\, E'_{kin} \sum_j n_j(E'_{kin})\frac{d\sigma_{ij}(E_{tot},E'_{kin})}{dE_{tot}} \\ &= \frac{c}{4\pi}A\,\Delta(\log E'_{kin})\sum_{i=H,He} n_i \sum_{E'_{kin}} E'_{kin}\sum_j n_j(E'_{kin})\frac{d\sigma_{ij}(E_{tot},E'_{kin})}{dE_{tot}}, \end{aligned} \tag{13}$$

where we used $dp' = \frac{1}{\beta}AE'_{kin}d(\log E'_{kin})$.

Since positron/electron is assumed massless $E_{tot} = p$. The units should be transferred to cm^2/MeV, a factor applied is 10^{-27}.

`gen_secondary_antiprotons_source`

The routine `ANTIPROTON` written in `FORTRAN-77` is designed to calculate the antiproton (+antineutron) production spectrum vs. momentum (barn/GeV). Antiproton momentum and nucleus momentum per nucleon are used as input parameters as well as beam and target nuclei atomic numbers.

The antiproton source function as used in *galprop* is defined as following (cm^{-3} s^{-1} sr^{-1} MeV^{-1}):

$$q_{\bar{p}}(p) = \frac{c}{4\pi}\frac{dn(p)}{dt} = \frac{c}{4\pi}\sum_{i=H,He} n_i \sum_j \int dp'\, \beta n_j(p') \frac{d\sigma_{ij}(p,p')}{dp}, \tag{14}$$

where n_i is the gas density, $d\sigma_i j(p,p')/dp$ is the production cross section, $n_j(p')$ is the CR species *density*, and p' is the total momentum of a nucleus. Substitution of dp' with $d(\log E'_{kin})$ gives:

$$\begin{aligned} q_{\bar{p}}(p) &= \frac{c}{4\pi} A \sum_{i=H,He} n_i \int d(\log E'_{kin})\, E'_{kin} \sum_j n_j(E'_{kin}) \frac{d\sigma_{ij}(p,E'_{kin})}{dp} \\ &= \frac{c}{4\pi} A\, \Delta(\log E'_{kin}) \sum_{i=H,He} n_i \sum_{E'_{kin}} E'_{kin} \sum_j n_j(E'_{kin}) \frac{d\sigma_{ij}(p,E'_{kin})}{dp}, \end{aligned} \tag{15}$$

where we used $dp' = \frac{1}{\beta} A E'_{kin} d(\log E'_{kin})$.

The units should be transferred to cm^2/MeV, a factor applied is 10^{-27}.

<u>`gen_secondary_protons_source`</u> and <u>`gen_tertiary_antiprotons_source`</u>

`Gen_secondary_protons_source and gen_tertiary_antiprotons_source` are made very similar using the same formalism. The secondary protons and tertiary antiprotons are essentially those which survived after inelastic scattering with protons and He nuclei of the interstellar gas. It is convenient and correct to consider them as additional populations of particles.

The tertiary $\bar{p}$'s are calculated using

$$q_{\mathrm{tert.}\bar{p}}(p) = c\Delta(\log E'_{kin}) \sum_{i=H,He} n_i \sum_{E'_{kin}} E'_{kin} n_{\mathrm{sec.}\bar{p}}(E'_{kin}) \frac{d\sigma_i(p,E'_{kin})}{dp}, \tag{16}$$

where $d\sigma_i(p,E'_{kin})/dp$ is the "production cross section" of tertiary $\bar{p}$'s. It can be further expanded as

$$\frac{d\sigma_i(p,E'_{kin})}{dp} = \sigma_i^{\mathrm{non}}(E'_{kin})\beta \frac{dN}{dE_{kin}}, \tag{17}$$

where $\sigma_i^{\mathrm{non}}(E'_{kin})$ is the total non-annihilation inelastic cross section, and dN/dE_{kin} is the distribution of $\bar{p}$'s after scattering. Using an approximation given by Tan & Ng (1983):

$$\frac{dN(E'_{kin},E_{kin})}{dE_{kin}} = \frac{1}{E'_{kin}}. \tag{18}$$

The final expression can be obtained by combining eqs. (16)-(18).

secondary source for nuclei: <u>`decayed_cross_sections`</u>

The formalism is similar to the above, except that the cross section of the disintegration process is convenient to write in energy or momentum per nucleon:

$$\frac{d\sigma_{ij}(p,p')}{dp} = \sigma(p')\,\delta\left(p - \frac{A_{sec}}{A_{prim}}p'\right), \tag{19}$$

where A_{prim}, A_{sec} are the atomic numbers of primary and secondary nucleus, correspondingly. Integrating over dp' then yield a factor A_{prim}/A_{sec}:

$$q_{sec}(p) = \beta c \sum_{i=H,He} n_i \sum_j \frac{A_j}{A_{sec}}\, \sigma\left(\frac{A_j p}{A_{sec}}\right) n_j\left(\frac{A_j p}{A_{sec}}\right). \tag{20}$$

The `decayed_cross_sections` routine calculates the cross section σ of a given channel taking into account short-lived intermediate states and nuclear reaction network.

4.5.3 `propel`

The basic routine for calculation of Crank-Nicholson coefficients and numerical solution of the propagation equation.

Numerical solution of propagation equation

The diffusion, reacceleration, convection and loss terms in eq. (1) can all be finite-differenced for each dimension (R, z, p) in the form

$$\frac{\partial \psi_i}{\partial t} = \frac{\psi_i^{t+\Delta t} - \psi_i^t}{\Delta t} = \frac{\alpha_1 \psi_{i-1}^{t+\Delta t} - \alpha_2 \psi_i^{t+\Delta t} + \alpha_3 \psi_{i+1}^{t+\Delta t}}{\Delta t} + q_i \ , \tag{21}$$

where all terms are functions of (R, z, p).

In the Crank-Nicholson implicit method (Press et al., 1992) the updating scheme is

$$\psi_i^{t+\Delta t} = \psi_i^t + \alpha_1 \psi_{i-1}^{t+\Delta t} - \alpha_2 \psi_i^{t+\Delta t} + \alpha_3 \psi_{i+1}^{t+\Delta t} + q_i \Delta t \ . \tag{22}$$

The tridiagonal system of equations,

$$-\alpha_1 \psi_{i-1}^{t+\Delta t} + (1 + \alpha_2) \psi_i^{t+\Delta t} - \alpha_3 \psi_{i+1}^{t+\Delta t} = \psi_i^t + q_i \Delta t, \tag{23}$$

is solved for the $\psi_i^{t+\Delta t}$ by the standard method (Press et al., 1992). Note that for energy losses we use 'upwind' differencing to enhance stability, which is possible since we have only *loss* terms (adiabatic energy *gain* is not included here).

The three spatial boundary conditions

$$\psi(R, z_h, p) = \psi(R, -z_h, p) = \psi(R_h, z, p) = 0 \tag{24}$$

are imposed at each iteration. No boundary conditions are imposed or required at $R = 0$ or in p. Grid intervals are typically $\Delta R = 1$ kpc, $\Delta z = 0.1$ kpc; for p a logarithmic scale with ratio typically 1.2 is used. Although the model is symmetric around $z = 0$ the solution is generated for $-z_h < z < z_h$ since this is required for the tridiagonal system to be valid.

Since we have a 3-dimensional (R, z, p) problem we use 'operator splitting' to handle the implicit solution, as follows. We apply the implicit updating scheme alternately for the operator in each dimension in turn, keeping the other two coordinates fixed. To account for the substeps $\frac{1}{3} q_i$ and $\frac{1}{3\tau}$ are used instead of q_i, $1/\tau$. The coefficients of the Crank-Nicholson scheme we use are given in Table 1.

The method was found to be stable for all α, and this property can be exploited to advantage by starting with $\alpha \gg 1$ (see below). The standard alternating direction implicit (ADI) method, in which the full operator is used to update each dimension implicitly in turn, is more accurate but was found to be unstable for $\alpha > 1$. This is a disadvantage when treating problems with many timescales, but can be used to generate an accurate solution from an approximation generated by the non-ADI method.

A check for convergence is performed by computing the timescale $\frac{\psi}{\partial \psi / \partial t}$ from eq. (1) and requiring that this be large compared to all diffusive and energy loss timescales. The main problem in applying the method in practice is the wide range of time-scales, especially for the electron case, ranging from 10^4 years for energy losses to 10^9 years for diffusion around 1 GeV in a large halo. Use of a time step Δt appropropriate to the smallest time-scales guarantees a reliable solution, but requires a prohibitively large number of steps to reach the long time-scales. The following technique was found to work well: start with a large Δt appropriate for the longest scales, and iterate until a stable solution is obtained. This solution is then accurate only for cells with $\alpha \ll 1$; for other cells the solution is stable but inaccurate. Then reduce Δt by a factor (0.5 was adopted) and continue the solution. This process is repeated until $\alpha \ll 1$ for all cells, when the solution is accurate everywhere. It is found that the inaccurate parts of the solution quickly decay as soon as the condition $\alpha < 1$ is reached for a cell. As soon as all cells satisfy $\alpha < 1$ the solution is continued with the ADI method to obtain maximum accuracy. A typical run starts with $\Delta t = 10^9$ years and ends with $\Delta t = 10^4$ years for nucleons and 10^2 years for electrons performing ~ 60 iterations per Δt. In this way it is possible to obtain reliable solutions in a reasonable computer resources, although the CPU required is still considerable. All results are output as FITS datasets for subsequent analysis.

Table 1: Coefficients for the Crank-Nicholson method.

Process	Coordinate	$\alpha_1/\Delta t$	$\alpha_2/\Delta t$	$\alpha_3/\Delta t$
Diffusion	R	$D_{xx}\frac{2R_i-\Delta R}{2R_i(\Delta R)^2}$	$D_{xx}\frac{2R_i}{R_i(\Delta R)^2}$	$D_{xx}\frac{2R_i+\Delta R}{2R_i(\Delta R)^2}$
	z	$D_{xx}/(\Delta z)^2$	$2D_{xx}/(\Delta z)^2$	$D_{xx}/(\Delta z)^2$
Convection[a]	$z>0$ $(V>0)$	$V(z_{i-1})/\Delta z$	$V(z_i)/\Delta z$	0
	$z<0$ $(V<0)$	0	$-V(z_i)/\Delta z$	$-V(z_{i+1})/\Delta z$
	p $(\frac{dV}{dz}>0)$	0	$\frac{1}{3}\frac{dV}{dz}\frac{p_i}{P_i^{i+1}}$	$\frac{1}{3}\frac{dV}{dz}\frac{p_{i+1}}{P_i^{i+1}}$
Diffusive reacceleration[a]	p	$-\frac{D_{pp,i}-D_{pp,i-1}}{P^{i\,2}_{i-1}} + \frac{2}{P^i_{i-1}}\left(\frac{D_{pp,i}}{P^{i+1}_{i-1}}+\frac{D_{pp,i-1}}{p_{i-1}}\right)$	$-\frac{D_{pp,i}-D_{pp,i-1}}{P^{i\,2}_{i-1}} + \frac{2D_{pp,i}}{P^{i+1}_{i-1}}\left(\frac{1}{P^{i+1}_i}+\frac{1}{P^i_{i-1}}\right) + \frac{2D_{pp,i}}{P^i_{i-1}p_i}$	$\frac{2D_{pp,i+1}}{P^{i+1}_{i-1}P^{i+1}_i}$
Energy loss[a]	p	0	$-\dot{p}_i/P_i^{i+1}$	$-\dot{p}_{i+1}/P_i^{i+1}$
Fragmentation	R,z,p	0	$1/3\tau_f$	0
Radioactive decay	R,z,p	0	$1/3\tau_r$	0

[a] $P^i_j \equiv p_i - p_j$

Diffusion in R

As an example, the coefficients for the radial diffusion term are derived here.

$$\frac{1}{R}\frac{\partial}{\partial R}\left(R\,D_{xx}\frac{\partial\psi}{\partial R}\right) = \frac{2}{R_i}\frac{D_{xx}}{R_{i+1}-R_{i-1}}\left\{R_{i+1}\frac{\psi_{i+1}-\psi_i}{R_{i+1}-R_i} - R_{i-1}\frac{\psi_i-\psi_{i-1}}{R_i-R_{i-1}}\right\}. \tag{25}$$

Setting $R_{i+1}-R_i = R_i - R_{i-1} = \Delta R$, one can obtain the following expressions in terms of our standard form (eq. [1])

$$\begin{aligned}\frac{\alpha_1}{\Delta t} &= D_{xx}\frac{2R_i-\Delta R}{2R_i(\Delta R)^2},\\ \frac{\alpha_2}{\Delta t} &= D_{xx}\frac{2R_i}{R_i(\Delta R)^2},\\ \frac{\alpha_3}{\Delta t} &= D_{xx}\frac{2R_i+\Delta R}{2R_i(\Delta R)^2}.\end{aligned} \tag{26}$$

Diffusive reacceleration

In terms of 3-D momentum phase-space density $f(\vec{p})$ the diffusive reacceleration equation is

$$\frac{\partial f(\vec{p})}{\partial t} = \vec{\nabla}_p\cdot[D_{pp}\vec{\nabla}_p\, f(\vec{p})] = \frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2 D_{pp}\frac{\partial f(p)}{\partial p}\right]. \tag{27}$$

The distribution is assumed isotropic so $f(\vec{p}) = f(p)$ where $p = |\vec{p}|$. First we rewrite the equation in terms of $\psi(p) = 4\pi p^2 f(p)$ instead of $f(p)$ and expand the inner differential:

$$\frac{\partial\psi}{\partial t} = \frac{\partial}{\partial p}\left[p^2 D_{pp}\frac{\partial}{\partial p}\frac{\psi}{p^2}\right] = \frac{\partial}{\partial p}\,D_{pp}\left[\frac{\partial\psi}{\partial p} - \frac{2\psi}{p}\right]. \tag{28}$$

The differencing scheme is then

$$\frac{2}{p_{i+1}-p_{i-1}}\left[D_{pp,i+1}\left(\frac{\psi_{i+1}-\psi_i}{p_{i+1}-p_i}-\frac{2\psi_{i+1}}{p_{i+1}}\right)-D_{pp,i-1}\left(\frac{\psi_i-\psi_{i-1}}{p_i-p_{i-1}}-\frac{2\psi_{i-1}}{p_{i-1}}\right)\right]. \tag{29}$$

In terms of our standard form (eq. [1]) the coefficients for reacceleration are

$$\begin{aligned}
\frac{\alpha_1}{\Delta t} &= \frac{2D_{pp,i-1}}{p_{i+1}-p_{i-1}}\left(\frac{1}{p_i-p_{i-1}}+\frac{2}{p_{i-1}}\right),\\
\frac{\alpha_2}{\Delta t} &= \frac{2}{p_{i+1}-p_{i-1}}\left(\frac{D_{pp,i+1}}{p_{i+1}-p_i}+\frac{D_{pp,i-1}}{p_i-p_{i-1}}\right),\\
\frac{\alpha_3}{\Delta t} &= \frac{2D_{pp,i+1}}{p_{i+1}-p_{i-1}}\left(\frac{1}{p_{i+1}-p_i}-\frac{2}{p_{i+1}}\right).
\end{aligned} \tag{30}$$

One more scheme (#2) comes from further detalization

$$\frac{d\psi}{dt}=\frac{\partial D_{pp}}{\partial p}\frac{\partial\psi}{\partial p}+D_{pp}\frac{\partial^2\psi}{\partial p^2}-2\frac{\partial}{\partial p}\frac{D_{pp}\psi}{p}. \tag{31}$$

Here it is

$$\begin{aligned}
\frac{\alpha_1}{\Delta t} &= -\frac{D_{pp,i}-D_{pp,i-1}}{(p_i-p_{i-1})^2}+\frac{2D_{pp,i}}{(p_{i+1}-p_{i-1})(p_i-p_{i-1})}+\frac{2D_{pp,i-1}}{(p_i-p_{i-1})p_{i-1}};\\
\frac{\alpha_2}{\Delta t} &= -\frac{D_{pp,i}-D_{pp,i-1}}{(p_i-p_{i-1})^2}+\frac{2D_{pp,i}}{p_{i+1}-p_{i-1}}\left(\frac{1}{p_{i+1}-p_i}+\frac{1}{p_i-p_{i-1}}\right)+\frac{2D_{pp,i}}{(p_i-p_{i-1})p_i};\\
\frac{\alpha_3}{\Delta t} &= \frac{2D_{pp,i+1}}{(p_{i+1}-p_{i-1})(p_{i+1}-p_i)}.
\end{aligned} \tag{32}$$

This scheme is used in `propel.cc`.

4.5.4 `nuclei_normalize & electrons_normalize`

Normalizes all nuclei to proton flux from galdef file. Applied at end of all nuclei processing but before computation gamma-rays (which use p, He and electrons). The value of E_{kin} is taken as the reference value for the proton normalization since this is already available in the *galdef* file.

Method: compute the proton flux $\frac{c}{4\pi}n(p)$ at the reference E_{kin} at $R=R_0$ and $z=0$ by interpolation, and hence obtain the normalizing factor to get the correct value as specified in the *galdef* file. Renormalizes the proton and all other nuclei fluxes by this same factor. The units of all spectra are then $\frac{c}{4\pi}n(p)$. Note that the source abundances are already taken into account in *create_transport_arrays*.

4.6 `store_gcr & store_gcr_full`

All distributions are stored in `FITS` files.

4.7 `gen_synch_skymap`

4.8 `gen_bremss_skymap, gen_IC_skymap, & gen_pi0_skymap`

π^o-decay emissivity from CR protons and Helium on interstellar hydrogen and Helium. Units:H atom^{-1} sr^{-1} s^{-1}MeV^{-1}.

The integral is over $\log(E_{kin})$, see section General Principles. This explains the formula used in this routine. The factor includes GeV^{-1} to MeV^{-1} and barn to cm^2.

5 Point sources of CR: spatial and temporal aspects

A new development from version 4 is the inclusion of spatial and temporal variations, which can be followed explicitly in the 3D case. Supernova remnants are produced at some rate (e.g. 1/100 yr) and produced cosmic rays for some time (e.g. 10,000 yr). The first step is to include point sources at arbitrary positions specified by the user, by this is mainly useful for testing and does not address the problem of fluctuations.

The correct approach is to consider the source function as $f(x, y, z, t)$. The problem is that the solution has to be followed on a time scale comparable to the SN rate, i.e. 100 years or less. This would not allow the advantages of the equilibrium solution scheme used in `galprop`. However provided the equilibrium solution is first derived, the short time steps can be started afterwards and will lead to the correct "recent" history of the system.

To model the source function we need an explicit $f(x, y, z, t)$ which has the right properties of producing short-lived events at different positions. This can be done by defining first a regular grid on which SNR can occur, and then generating a random phase for each grid point. The function of time at each point is then a suitable function with the required duty cycle and period.

The basic spatial grid itself can be used, there is no need to define a separate one for the sources. The required parameters are: the probability per unit time and volume of a SNR events, and the time for which it is active. To express both as a time, define `SNR_time_interval` as the time between events in a kpc^3 volume in the Solar vicinity for the first parameter.

The structure Galaxy contains an array giving the phase and rate for each cell. The phases are chosen randomly at the start of the program. The rates are determined from the time interval and the source distribution which is taken as the same as that defined for constant sources. At any time, the on/off state of an SNR in a cell is determined from the time relative to the phase, the time between SNR in the cell (`SNR_cell_time`) and the live time.

The propagation code (`propel.cc`) is divided into two sections to handle the smooth source function using the decreasing timesteps, followed by a section with constant timesteps to handle the SNR events on a fine timescale. These are called `timestep_mode`=1, 2 respectively. The number of timesteps in each mode can be specified independently, but in mode 2 the shortest timestep of mode 1 is used.

If the number of timesteps in mode 1 is set to zero then only mode 2 is used. For a combination of modes it is necessary to normalize the source functions to be consistent: this has to be done by adjusting the input per SNR (`Q_SNR`) since the rate and live time are given. The source function returned by `source_distribution.cc` is defined as production per unit volume (cm^3) per second. (it gets multiplied by spectral and abundance factors later).

The assigned source function in source_SNR_event has to satisfy

$$source\_function * SNR\_livetime/SNR\_cell\_time = source\_distribution(x, y, z)$$

so

$$source\_function = source\_distribution(x, y, z)/SNR\_livetime * SNR\_cell\_time$$

and the SNR input when on should be always same everywhere by construction.
`create_SNR.cc` uses

$$SNR\_cell\_time =$$
$$SNR\_interval * source\_distribution(Sun)/source\_distribution(x, y, z)/cellvolume$$

so the formula used by `source_SNR_event.cc` is

$$source\_function = source\_distribution(x, y, z)/SNR\_livetime * SNR\_interval$$
$$*source\_distribution(Sun)/source\_distribution(x, y, z)/cellvolume$$

which simplifies to

$$source\_function = SNR\_interval/SNR\_livetime * source\_distribution(Sun)/cellvolume$$

and the output rate of an SNR =

$$source\_function * cellvolume = SNR\_interval/SNR\_livetime * source\_distribution(Sun)$$

which is a constant as required. Note that $SNR\_interval$ is for 1 kpc^3 and so has units of years * kpc^3 while $source\_distribution(Sun)$ has units of cm^{-3} s^{-1} so the SNR output rate units are s^{-1} as required. Similarly $source\_function$ has the same units as $source\_distribution$, as required.

6 galdef file parameter explanations

```
Title                 = Zh= 4kpc cs=W i=rigidity^-2.43 D=6.10 0.33 dvdz=0 Va=30// pbar test
```

Descriptive title used to identify the run. With commands like "grep Title galdef*" you can get a summary of all runs.

```
n_spatial_dimensions = 2
```

Specifies whether 2 or 3 spatial dimensions. 2D is cylindrically symmetric (R, z), 3D is (x, y, z) and may be fully asymmetric or with symmetry in x, y, z as specified by the parameter use_symmetry.

```
r_min                =00.0    min r
```

Minimum galactocentric radius, $R_{\rm min}$, for 2D case, in kpc. Normally 0. Ignored for 3D.

```
r_max                =30.00   max r
```

Maximum galactocentric radius, $R_{\rm max}$, for 2D case, in kpc.

```
dr                   = 1.0    delta r
```

Cell size in galactocentric radius, ΔR, for 2D case, in kpc.

```
z_min                =-04.0   min z
```

Minimum height for 2D and 3D case, in kpc. In 3D case with use_symmetry = 1 it must be 0, since in this case only $z > 0$ is explicitly computed.

```
z_max                =+04.0   max z
```

Maximum height for 2D and 3D case, in kpc.

```
dz                   =  0.1   delta z
```

Cell size in z for 2D and 3D case, in kpc.

```
x_min                = 00.0   min x
```

Minimum x for 3D case, in kpc. In 3D case with use_symmetry = 1 it must be 0, since in this case only $x > 0$ is explicitly computed. Ignored for 2D.

```
x_max                =+20.0   max x
```

Maximum x for 3D case, in kpc. Ignored for 2D.

```
dx                   =  0.2   delta x
```

Cell size in x for 3D case, in kpc. Ignored for 2D.

```
y_min                = 00.0   min y
```

See `x_min`, but now for y-axis.

```
y_max                =+20.0   max y
```

See `x_max`, but now for y-axis.

```
dy                   =  0.2   delta y
```

See `dx`, but now for y-axis.

```
p_min                =1000    min momentum (MV)
```

Minimum particle momentum in megavolts (MV), case `p_Ekin_grid = p`. NB do not use except for testing.

```
p_max                =4000    max momentum
```

Maximum particle momentum in megavolts (MV), case `p_Ekin_grid = p`. NB do not use except for testing.

```
p_factor             =1.20        momentum factor
```

The ratio between successive momentum grid points, case `p_Ekin_grid = p`. The momentum grid is on a logarithmic scale, so that $p[i] = p_{\min} \times p^i_{factor}; p[0] = p_{\min}$. The number of grid points is chosen to give a maximum value close to `p_max`. NB do not use except for testing.

```
Ekin_min             =1.0e1  min kinetic energy per nucleon (MeV)
```

Minimum particle kinetic energy per nucleon in MeV, case `p_Ekin_grid = Ekin`.

```
Ekin_max             =1.0e7  max kinetic energy per nucleon
```

Maximum particle kinetic energy per nucleon in MeV, case `p_Ekin_grid = Ekin`.

```
Ekin_factor          =1.20       kinetic energy per nucleon factor
```

The ratio between successive kinetic energy per nucleon grid points, case `p_Ekin_grid = Ekin`. The momentum grid is on a logarithmic scale, so that $E_{kin}[i] = E_{kin\ \min} \times E^i_{kin\, factor}; E_{kin}[0] = E_{kin\ \min}$. The number of grid points is chosen to give a maximum value close to `Ekin_max`.

```
p_Ekin_grid          = Ekin          p||Ekin alignment
```

The grid points are arranged so all particles are aligned the same `Ekin` scale; this is very convenient when computing secondary/primary ratios which are always presented in this form. For details see the explanation elsewhere in this document. (The actually propagation calculation is done in terms of momentum since this is physically more natural, but the user does not have to worry about this.) NB only the case `p_Ekin_grid` is useful for nuclei, p-alignment may however be interesting in some cases, hence this option is kept.

```
E_gamma_min          = 0.1      min gamma-ray energy (MeV)
```

Minimum gamma-ray energy (MeV) for diffuse gamma-ray maps.

```
E_gamma_max          = 1.e6     max gamma-ray energy (MeV)
```

Maximum gamma-ray energy (MeV) for diffuse gamma-ray maps.

```
E_gamma_factor       = 10.      gamma-ray energy factor
```

The ratio between successive gamma-ray energy grid points. The energy grid is on a logarithmic scale, so that $E_\gamma[i] = E_{\gamma\ \min} \times E^i_{\gamma\, factor}; E_\gamma[0] = E_{\gamma\ \min}$. The number of grid points is chosen to give a maximum value close to `E_gamma_max`.

```
nu_synch_min         = 1.0e6   min synchrotron frequency (Hz)
```

Minimum frequency (Hz) for synchrotron maps.

```
nu_synch_max         = 1.0e10  max synchrotron frequency (Hz)
```

Maximum frequency (Hz) for synchrotron maps.

```
nu_synch_factor      = 2.0         synchrotron frequency factor
```

The ratio between successive synchrotron frequency grid points. The frequency grid is on a logarithmic scale, so that $\nu_{synch}[i] = \nu_{synch\ \min} \times \nu^i_{synch\,factor}; \nu_{synch}[0] = \nu_{synch\ \min}$. The number of grid points is chosen to give a maximum value close to `nu_synch_max`.

```
long_min             =  0.5    gamma-ray intensity skymap longitude minimum (deg)
```

Minimum longitude for gamma-ray intensity skymaps (degrees).

```
long_max             =359.5    gamma-ray intensity skymap longitude maximum (deg)
```

Maximum longitude for gamma-ray intensity skymaps (degrees).

```
lat_min              =-89.5    gamma-ray intensity skymap latitude  minimum (deg)
```

Minimum latitude for gamma-ray intensity skymaps (degrees).

```
lat_max              =+89.5    gamma-ray intensity skymap latitude  maximum (deg)
```

Maximum latitude for gamma-ray intensity skymaps (degrees).

```
d_long               = 10.     gamma-ray intensity skymap longitude binsize (deg)
```

Binsize in longitude for gamma-ray intensity skymaps (degrees).

```
d_lat                = 10.     gamma-ray intensity skymap latitude  binsize (deg)
```

Binsize in latitude for gamma-ray intensity skymaps (degrees).

```
D0_xx                =6.10e28  diffusion coefficient at reference rigidity
```

The spatial diffusion coefficient divided by $\beta(= v/c)$ at rigidity `D_rigid_br`. The value at other rigidities is determined via the formula $D = \beta D_{0xx}(\rho/D_{rigid\,br})^{D_{g1}}$ for rigidity < `D_rigid_br`, $D = \beta D_{0xx}(\rho/D_{rigid\,br})^{D_{g2}}$ for rigidity > `D_rigid_br`.

```
D_rigid_br           =4.0e3    reference rigidity for diffusion coefficient in MV
```

Rigidity for `D0_xx` formula, and also break point in case `D_g_1 != D_g_2`.

```
D_g_1                = 0.33    diffusion coefficient index below reference rigidity
```

see formula for `D0_xx`. Kolmogorov turbulence corresponds to a value 1/3.

```
D_g_2                = 0.33    diffusion coefficient index above reference rigidity
```

see formula for `D0_xx`.

```
convection           =0        1=include convection
```

Flag to indicate whether convection is to be included in propagation.

```
v0_conv              =0.       Vo convection in km s-1
```

Convection velocity at $z = 0$ (km s^{-1}). In practice only a value 0 seems possible, from symmetry.

```
dvdz_conv           =7.        dV/dz=grad V  in km s-1 kpc-1
```

Gradient of convection velocity, assumed linear, in km s^{-1} kpc^{-1}.

```
diff_reacc          =1         1=include diffusive reacceleration
```

Flag to indicate whether diffuse reacceleration is to be included in propagation.

```
v_Alfven            =30.       Alfven speed in km s-1
```

Alfvén speed for computation of diffusive reacceleration momentum diffusion coefficient (see explanation elsewhere in document). This parameter is in fact Alfvén speed/$\sqrt{w}$ where w is the ratio of MHD wave energy density to magnetic field energy density, see Strong & Moskalenko (1998).

```
nuc_rigid_br        =1.0e2     reference rigidity for primary nucleus injection index in MV
```

In the case that the primary nucleus injection spectra have a break, this defines the rigidity of the break in MV. It it the same for all primary nuclei.

```
nuc_g_1             =2.43      nucleus injection index below reference rigidity
```

Injection index below `nuc_rigid_br`.

```
nuc_g_2             =2.43      nucleus injection index index above reference rigidity
```

Injection index above `nuc_rigid_br`.

```
inj_spectrum_type   =rigidity  rigidity||beta_rig||Etot nucleon injection spectrum type
```

The primary nucleus injection spectrum may be defined to be a power law in rigidity, $\beta\times$(rigidity) or total energy. Normally rigidity is used since it corresponds to a power-law in momentum favoured by SNR shock acceleration models.

```
electron_rigid_br   =1.0e3     reference rigidity for electron injection index in MV
```

In the case that the primary electron injection spectrum has a break, this defines the rigidity of the break in MV.

```
electron_g_1        =2.50      electron injection index below reference rigidity
```

Injection index below `electron_rigid_br`.

```
electron_g_2        =2.50      electron injection index index above reference rigidity
```

Injection index above `electron_rigid_br`.

```
He_H_ratio          =0.11      He/H of ISM, by number
```

Interstellar gas ratio of Helium to Hydrogen. Used for fragmentation, secondary production, energy loss, gamma-ray production. Values 0.08–0.11 are reasonable, see discussion in Strong & Moskalenko (1998).

```
fragmentation       =1         1=include fragmentation
```

Flag to indicate whether nuclei fragmentation to be included. Useful for testing effect of fragmentation.

```
momentum_losses     =1         1=include momentum losses
```

Flag to indicate whether momentum losses to be included. Useful for testing effect of losses.

```
radioactive_decay    =1         1=include radioactive decay
```

Flag to indicate whether radioactive decay to be included. Useful for testing effect of decay.

```
K_capture            =0         1=include K-capture
```

Flag to indicate whether electron K-capture decay to be included. (NB Disabled in the current version.)

```
start_timestep       =1.0e7   (years)
```

The solution of the propagation equation proceeds starting with a large timestep, repeated typically 20 times (defined by `timestep_repeat`) and reduces the timestep successively. The theory of this is given in Strong & Moskalenko (1998). This parameter is the starting timestep, typically 10^7 years corresponding to the longest timescales in cosmic-ray propagation (20×10^7 years, sufficient for 4 kpc halo).

```
end_timestep         =1.0e1   (years)
```

Final timestep, corresponding to the shortest timescales, dominated by energy-losses; for the highest energy electrons (& positrons) this should be < 100 years for safety, for nuclei-only runs it can be 10^4 years.

```
timestep_factor      =0.50
```

Factor by which timestep is reduced after `timestep_repeat` iterations.

```
timestep_repeat      =20      number of repeats per timestep in  timetep_mode=1
```

Number of iterations of the timestep for each `timestep_factor`. Typically 20. Criterion is that a stable solution is obtained before proceeding to the next smaller timestep, which can be checked by turning on `control_diagnostics` (not normally necessary).

```
timestep_repeat2     =0       number of timesteps in timestep\_mode=2
```

At the end of the series of reduced timesteps the propagation can be continued with the smallest timestep. This is mainly useful in the 3D mode with stochastic SNR sources, where the large timestep technique is not appropriate since we have a time-dependent problem. This parameter specifies how many timesteps are to be performed in this mode.

```
timestep_print       =200     number of timesteps between printings
```

The full cosmic-ray density array can be printed at intervals using this parameter.

```
timestep_diagnostics =10000  number of timesteps between diagnostics
```

Diagnostics to evaluate the quality of the propagation solution can be generated, normally only occasionally as specified by the number of timesteps between diagnostics.

```
control_diagnostics  =0       control detail of diagnostics
```

Controls the amount of detail in diagnostics. Larger $\rightarrow$ more detail.

```
network_iterations   =1       number of iterations of entire network
```

The nuclei fragmentation network starts with the largest A and works downwards in A. More most nuclei this ensures that all secondary, tertiary, etc. products are produced in one complete run. So normally `network_iterations=1`. However there may be cases where a second iteration is required to get all products, and this can be controlled by this parameter.

```
prop_r               = 1  1=propagate in r (2D)
```

Flag to indicate whether to propagate in radial direction (2D only). This and x,y,z,p flags are useful for testing the code, e.g., by allowing propagation in one direction only.

```
prop_x              = 1  1=propagate in x (2D)
```

Flag to indicate whether to propagate in x-direction (3D only).

```
prop_y              = 1  1=propagate in y (3D)
```

Flag to indicate whether to propagate in y-direction (3D only).

```
prop_z              = 1  1=propagate in z (2D, 3D)
```

Flag to indicate whether to propagate in z-direction (2D,3D).

```
prop_p              = 1  1=propagate in momentum
```

Flag to indicate whether to propagate in momentum (2D,3D).

```
use_symmetry        = 1  0=no symmetry, 1=optimized symmetry, 2=xyz symmetry by copying(3D)
```

This is only relevant for 3D. The code will solve the 3D case in general, but this leads to large computer resource requirements, in particular large memory. Often symmetry can be assumed without reducing the usefulness of the results since even with stochastic sources the fact that the sources are symmetically distributed has no effect on the conclusions about, for example, fluctuations. Hence use_symmetry=1 is generally recommended and the code has been specially optimized to take advantage of this. The memory requirements are reduced by a factor 8 relative to the general case.

```
vectorized          = 0  0=unvectorized code, 1=vectorized code
```

Flag to indicate use of vectorized portions of code. Only advantageous on vector processor machines, in 3D. Only applies to timestep_mode=2.

```
source_specification = 0  2D::1:r,z=0 2:z=0  3D::1:x,y,z=0 2:z=0 3:x=0 4:y=0
```

This parameter is for testing only, leave at 0.

```
source_model        = 1  0=zero 1=parameterized 2=Case&B 3=pulsars 4= 5=S&Mattox 6=S&Mattox with cutoff
```

Various CR source distributions. source_model=1 uses the following 3 source parameters, in the formula $R^{\alpha}e^{-\beta R}$, with cutoff at $R = r_{\max}$ (for details see Strong & Moskalenko, 1998).

```
source_parameters_1  = 0.5    model 1:alpha

source_parameters_2  = 1.0    model 1:beta

source_parameters_3  = 20.0   model 1:rmax

n_cr_sources         = 0     number of pointlike cosmic-ray sources   3D only!
```

This allows individual CR sources to be inserted in 3D. Useful for testing the response to a delta function, otherwise has not been used much.

```
cr_source_x_01       = 10.0  x position of cosmic-ray source 1 (kpc)
cr_source_y_01       = 10.0  y position of cosmic-ray source 1
cr_source_z_01       = 0.1   z position of cosmic-ray source 1
cr_source_w_01       = 0.1 sigma width  of cosmic-ray source 1
cr_source_L_01       = 1.0   luminosity of cosmic-ray source 1
cr_source_x_02       = 3.0   x position of cosmic-ray source 2
cr_source_y_02       = 4.0   y position of cosmic-ray source 2
cr_source_z_02       = 0.2   z position of cosmic-ray source 2
cr_source_w_02       = 2.4 sigma width  of cosmic-ray source 2
cr_source_L_02       = 2.0   luminosity of cosmic-ray source 2
```

```
SNR_events           = 0    handle stochastic SNR events
```

Flag to indicate whether to generate CR sources in the form of SNR, random in space and time. Only in 3D and only for `timestep_mode=2`. The spatial distribution of SNR follows `source_specification`, the time dependence is determined by the following 2 parameters.

```
SNR_interval         = 1.0e4 time interval in years between SNR in 1 kpc^-3 volume
```

Determines the rate of SNR, according to method described elsewhere in this document. The total SNR rate corresponding to a given value of `SNR_interval` can be obtained by running galprop and noting the value printed on output. Roughly 1.0 years corresponds to 3 SNR/century.

```
SNR_livetime         = 1.0e4 CR-producing live-time in years of an SNR
```

The time in years for which an SNR remains a "live" source of CR.

```
SNR_electron_sdg     = 0.00      delta electron source index Gaussian sigma
```

Allows for dispersion in the electrons source spectral index. This parameter gives the Gaussian sigma of the index around the values specified in `electron_g_1,2`. A value 0.00 produces no dispersion.

```
SNR_nuc_sdg          = 0.00      delta nucleus  source index Gaussian sigma
```

Allows for dispersion in the nuclei source spectral index. This parameter gives the Gaussian sigma of the index around the value specified in `nuc_g_1,2`. A value 0.00 produces no dispersion.

```
SNR_electron_dgpivot = 5.0e3     delta electron source index pivot rigidity (MeV)
```

In the case of dispersion in the electron source index, this parameter defines the rigidity about which the spectrum pivots.

```
SNR_nuc_dgpivot      = 5.0e3     delta nucleus  source index pivot rigidity (MeV)
```

In the case of dispersion in the nucleon source index, this parameter defines the rigidity about which the spectrum pivots.

```
B_field_model        = 050100020   bbbrrrzzz    bbb=10*B(0)  rrr=10*rscale zzz=10*zscale
```

Specifies the perpendicular component of the magnetic field according to a the law $B = (bbb/10)\times e^{-R/(rrr/10)-z/(zzz/10)}$ microgauss, e.g., this example has $B = 5e^{(-R/10kpc-z/2kpc)}$ microgauss.

```
proton_norm_Ekin     = 1.00e+5 proton kinetic energy for normalization (MeV)
```

kinetic energy for normalization of proton flux in MeV.

```
proton_norm_flux     = 5.50e-9 flux of protons at normalization energy (cm^-2 sr^-1 s^-1 MeV^-1)
```

proton flux at $R = R_\odot = 8.5$ kpc, $z = 0$, at `proton_norm_Ekin` in cm^{-2} sr^{-1} s^{-1} MeV^{-1}.

```
electron_norm_Ekin   = 3.45e4  electron kinetic energy for normalization (MeV)
```

kinetic energy for normalization of electron flux, in MeV.

```
electron_norm_flux   = 4.0e-10 flux of electrons at normalization energy (cm^-2 sr^-1 s^-1 MeV^-1)
```

electron flux at $R = R_\odot = 8.5$ kpc, $z = 0$, at `electron_norm_Ekin`, in cm^{-2} sr^{-1} s^{-1} MeV^{-1}.

```
max_Z                = 28      maximum number of nucleus Z listed
```

Specifies how many parameters `use_Z_..` follow.

```
use_Z_1              = 1
```

Flag to specify that nuclei with $Z = 1$ are to be processed.

```
use_Z_2              = 1
```

Flag to specify that nuclei with $Z = 2$ are to be processed, and so on:

```
use_Z_3              = 1
use_Z_4              = 1
use_Z_5              = 1
use_Z_6              = 1
use_Z_7              = 1
use_Z_8              = 1
use_Z_9              = 1
use_Z_10             = 1
use_Z_11             = 1
use_Z_12             = 1
use_Z_13             = 1
use_Z_14             = 1
use_Z_15             = 1
use_Z_16             = 1
use_Z_17             = 1
use_Z_18             = 1
use_Z_19             = 1
use_Z_20             = 1
use_Z_21             = 1
use_Z_22             = 1
use_Z_23             = 1
use_Z_24             = 1
use_Z_25             = 1
use_Z_26             = 1
use_Z_27             = 1
use_Z_28             = 1
use_Z_29             = 0
use_Z_30             = 0
```

The following parameters give the primary source isotopic abundances. The values are relative, the final CR fluxes are normalized so that the proton flux is as specified by `proton_norm_flux`. Hence the user is free to use abundances relative to any species. The abundances are assumed to be valid at a kinetic energy per nucleon equal to `proton_norm_Ekin`. The example abundances give here have been described in Strong & Moskalenko (2001).

```
iso_abundance_01_001 = 1.430e6        H
```

abundance of $Z = 1,\ A = 1$.

```
iso_abundance_02_004 = 1.350e5        He   was 0.069e6   // Solar system relative isotope abund.:
```

abundance of $Z = 2,\ A = 4$, and so on:

```
iso_abundance_03_006 =    0.          Li
iso_abundance_04_009 =    0.          Be
iso_abundance_05_010 =    0.          B
iso_abundance_06_012 = 2548.  (2573)  C =3000                                          12- 0.955
iso_abundance_06_013 =   25.                                                           13- 0.045
iso_abundance_07_014 =  175.          N  =137.
```

```
iso_abundance_08_016 = 3673.         O
iso_abundance_09_019 =    0.         F
iso_abundance_10_020 =  310.   (403) Ne =???                                         20- 0.88
iso_abundance_10_022 =   93.            22/20 =0.3 in source (DuVernois et al 1996)  22- 0.12
iso_abundance_11_023 =   21.         Na
iso_abundance_12_024 =  626.         Mg =734 *1.1                                    24- 0.78
iso_abundance_12_025 =   80.7                                                        25- 0.10
iso_abundance_12_026 =  100.5                                                        26- 0.12
iso_abundance_13_027 =   45.         Al
iso_abundance_14_028 =  680.   (760) Si =707       Source ab.: Hesse et a. 1996   28/28- 1.00
iso_abundance_14_029 =   60.                                                      29/28- 0.09
iso_abundance_14_030 =   20.                                                      30/28- 0.03
iso_abundance_15_031 =    8.         P  =4.92
iso_abundance_16_032 =   97.0  (105) S  =92.4      Source ab.: Thayer 1997        32/32- 1.00
iso_abundance_16_033 =    2.1                                                     33/32- 0.026
iso_abundance_16_034 =    6.3                                                     34/32- 0.062
iso_abundance_17_035 =    0.9        Cl
iso_abundance_17_037 =    3.2
iso_abundance_18_036 =   20.0        Ar =15.2
iso_abundance_18_038 =    4.0           -introduced by imos
iso_abundance_19_039 =    0.         K  -introduced by imos
iso_abundance_20_040 =   39.0        Ca =42.
iso_abundance_20_041 =    0.8
iso_abundance_21_045 =    0.         Sc
iso_abundance_22_046 =    0.         Ti -introduced by imos
iso_abundance_22_047 =    1.9
iso_abundance_22_048 =    7.3
iso_abundance_22_049 =    0.
iso_abundance_22_050 =    1.4
iso_abundance_23_051 =    0.         V
iso_abundance_24_050 =    1.4        Cr -introduced by imos
iso_abundance_24_051 =    0.
iso_abundance_24_052 =    7.7
iso_abundance_24_053 =    3.4
iso_abundance_24_054 =    1.0
iso_abundance_25_053 =    2.2        Mn -introduced by imos
iso_abundance_25_055 =    6.8
iso_abundance_26_054 =   72.1  (882) Fe =713  Source abund.: Connell & Simpson 1997 54/56= 9.3%
iso_abundance_26_055 =   12.1                                                     55/56= 1.6%
iso_abundance_26_056 =  776.1                                                     56/56= 100%
iso_abundance_26_057 =   28.8                                                     57/56= 3.7%
iso_abundance_26_058 =    1.41                                                    58/56= 0.18%
iso_abundance_27_059 =    0.4        Co =1.28
iso_abundance_28_058 =   30.61 (44.3) Ni =40.2 Source ab.: Connell & Simpson 1997  58/58= 100%
iso_abundance_28_059 =    0.81                                                    59/58= 2.6% =0.69
iso_abundance_28_060 =   13.17                                                    60/58=43.2%
iso_abundance_28_061 =    0.35                                                    61/58 <1.2%
iso_abundance_28_062 =    1.62                                                    62.58= 5.4%
```

```
total_cross_section  =   0    =0 -Letaw83; =1 -WA96 Z>5 & BP01 Z<6; =2 -BP01 (2-best)
```

Options for determining total fragmentation cross sections. BP= code from V.S.Barashenkov,A.Polanski and this is the preferred option. (NB Options other then 0 are disabled in the current version.)

```
cross_section_option =   001    100*i+j  i=1: use Heinbach-Simon C,O->B j=kopt j=1=Webber, 2=ST
```

Options for determining cross sections. Details on kopt from nuc_package.cc:

```
//3 kopt  =0 - uses best alghorithm described in comments below (not recommended);
//3       =1 - forces to use Webber'93 code (no renormalization etc.);
//3       =2 - forces to use TS00 code (no renormalization etc.);
//3       =3 - forces to use a const cross section fitted to the data.
//3       =10- forces to use Webber'93 code (renormalized if data exists);
//3       =11- forces to use cross section fit if exists (otherwise equiv. 10);
//3       =12- forces to use a numerical table if exists (otherwise equiv. 11);
//3       =20- forces to use TS00 code (renormalized if data exists).
//3       =21- forces to use cross section fit if exists (otherwise equiv. 20).
//3       =22- forces to use a numerical table if exists (otherwise equiv. 21);
//3 The best values recommended are kopt = 12, 22 (12 is preferrable).
//3 uses file_no[1] and file_no[3] as indicators of the data array and fit params.
```

(NB Options `kopt > 3` are disabled in the current version.)

```
primary_electrons    = 0
```

Flag to indicate whether to propagate primary electrons.

```
secondary_positrons  = 0
```

Flag to indicate whether to propagate secondary positrons.

```
secondary_electrons  = 0
```

Flag to indicate whether to propagate secondary electrons.

```
secondary_antiproton = 2   1=uses nuclear scaling; 2=uses nuclear factors by Simon et al 1998
```

Flag to indicate whether to propagate secondary antiprotons, with 2 options.

```
tertiary_antiproton  = 1
```

Flag to indicate whether to propagate tertiary antiprotons.

```
secondary_protons    = 1
```

Flag to indicate whether to propagate secondary protons.

```
gamma_rays           = 0    1=compute gamma rays
```

Flag to indicate whether to compute diffuse Galactic gamma-ray skymaps and emissivities Requires computation of protons, Helium and electrons.

```
IC_anisotropic       = 0    1=compute anisotropic IC
```

Flag to indicate whether to compute anisotropic inverse Compton scattering according to MS2000. Normally only isotropic IC is computed since the anisotropic is very time consuming. A more efficient algorithm will be provided eventually.

```
synchrotron          = 0    1=compute synchrotron
```

Flag to indicate whether to compute synchrotron skymaps. Requires `primary_electrons =1`.

```
output_gcr_full      = 0  output full galactic cosmic ray array
```

Flag to indicate whether to output the full (R, z, p) or (x, y, z, p) array of nuclei. For 3D this array can be very large but is required for studying spatial effects. It must be set to 1 if subsequent runs (`warm_start`) are to be made using this run as input.

```
warm_start          = 0  read in nuclei file and continue run
```

Flag to indicate the run is to be started with results output from previous run with `output_gcr_full =1`. Useful for long runs which have to be broken up due to CPU time limitations. NB the galdef file should be the same as for the first run, apart from this parameter; the only possible differences should be the values of `end_timestep` and `timestep_repeat2`, and the generation of gamma-rays etc. The grid and selected nuclei, electrons etc. must be the same.

```
verbose             = 0  verbosity: 0=min,10=max
```

Controls level of output: 0,1,2...10.

```
test_suite          = 0  run test suite instead of normal run
```

Instead of normal run, generates test output on many subroutines such as cross-section evaluation.

References

Badhwar, G. D., Stephens, S. A., & Golden, R. L. 1977, Phys. Rev. D 15, 820

Berezinskii, V. S., Bulanov, S. V., Dogiel, V. A., Ginzburg, V. L., & Ptuskin, V. S. 1990, Astrophysics of Cosmic Rays (Amsterdam: North Holland)

Broadbent, A., et al. 1990, Proc. 21st Int. Cosmic Ray Conf. 3, 229

Bronfman, L., Cohen, R. S., Alvarez, H., May, J., & Thaddeus, P. 1988, ApJ 324, 248

Cordes, J. M., Weisberg, J. M., Frail, D. A., Spangler, S. R., & Ryan, M. 1991, Nature 354, 121

Cox, P., Krügel, E., & Mezger, P. G. 1986, A&A 155, 380

Dermer, C. D. 1986a, ApJ 307, 47

Dermer, C. D. 1986b, A&A 157, 223

Dickey, J. M., & Lockman, F. J. 1990, Ann. Rev. Astron. Astrophys. 28, 215

Dwek, E., et al. 1997, ApJ 475, 565

Ferrando, P., Webber, W. R., Goret, P., Kish, J. C., Schrier, D. A., Soutoul, A., & Testerd, O. 1988, Phys. Rev. C 37, 1490

Freudenreich, H. T. 1998, ApJ 492, 495

Gaisser, T. K., & Schaefer, R. K. 1992, ApJ 394, 174

Gordon, M. A., & Burton, W. B. 1976, ApJ 208, 346

Groom, D. E., et al. 2000, Europ. Phys. J. C15, 1

Heiles, C. 1996, in ASP Conf. Ser. 97 Polarimetry of the Interstellar Medium, ed. W. G. Roberge & D. C. B. Whittet (San Francisco: ASP), 457

Koch, H. W., & Motz, J. W. 1959, Rev. Mod. Phys. 31, 920

Lazio, T. J. W., & Cordes, J. M. 1998a, ApJ 497, 238

Lazio, T. J. W., & Cordes, J. M. 1998b, ApJ 505, 715

Letaw, J. R., Silberberg, R., & Tsao, C. H. 1983, ApJS 51, 271

Meyer, J.-P., Drury, L. O'C., & Ellison, D. C. 1998, Space Sci. Rev. 86, 179

Moiseev, A. A., & Ormes, J. F. 1997, Astropart. Phys. 6, 379

Moskalenko, I. V., & Strong, A. W. 1998, ApJ 493, 694

Moskalenko, I. V., & Strong, A. W. 1999, Phys. Rev. D 60, #063003

Moskalenko, I. V., & Strong, A. W. 2000a, ApJ 528, 357

Moskalenko, I. V., & Strong, A. W. 2000b, Astrophys. Space Sci. 272, 247

Moskalenko, I. V., Mashnik, S. G., & Strong, A. W. 2001a, Proc. 27th Int. Cosmic Ray Conf. , 1836 (astro-ph/0106502)

Moskalenko, I. V., Strong, A. W., & Reimer, O. 1998, A&A 338, L75

Moskalenko, I. V., Strong, A. W., Ormes, J. F., & Potgieter, M. S. 2001b, ApJ 564, in press (astro-ph/0106567)

Press, W. H., et al. 1992, Numerical Recipes in FORTRAN (2d ed.; Cambridge: Cambridge Univ. Press)

Roesler, S., Engel, R., & Ranft, J. 1998, Phys. Rev. D 57, 2889

Seo, E. S., & Ptuskin, V. S. 1994, ApJ 431, 705

Simon, M., Molnar, A., & Roesler, S. 1998, ApJ 499, 250

Sodrowski, T. J., et al. 1997, ApJ 480, 173

Stecker, F. W. 1970, Astrophys. Space Sci. 6, 377

Stephens, S. A., & Badhwar, G. D. 1981, Astrophys. Space Sci. 76, 213

Strong, A. W., & Mattox, J. R. 1996, A&A 308, L21

Strong, A. W., & Moskalenko, I. V. 1998, ApJ 509, 212

Strong, A. W., & Moskalenko, I. V. 1999, in Horizons in World Physics 230 Topics in Cosmic Ray Astrophysics, ed. M. A. DuVernois (New York: Nova Scientific), 81

Strong, A. W., & Moskalenko, I. V. 2001, Adv. Space Res. 27, 717 (astro-ph/0101068)

Strong, A. W., Moskalenko, I. V., & Reimer, O. 2000, ApJ 537, 763; Erratum: 2000, ApJ 541, 1109

Strong, A. W., & Moskalenko, I. V. 2001b, Proc. 27th Int. Cosmic Ray Conf. , 1942 (astro-ph/0106504)

Strong, A. W., & Moskalenko, I. V. 2001c, Proc. 27th Int. Cosmic Ray Conf. , 1964 (astro-ph/0106505)

Tan, L. C., & Ng, L. K. 1983a, J. Phys. G: Nucl. Part. Phys. 9, 227

Tan, L. C., & Ng, L. K. 1983b, J. Phys. G: Nucl. Part. Phys. 9, 1289

Taylor, J. H., & Cordes, J. M. 1993, ApJ 411, 674

Vallée, J. P. 1996, Fund. Cosmic Phys. 19, 1

Wainscoat, R. J., et al. 1992, ApJS 83, 111

Webber, W. R., Kish, J. C., & Schrier, D. A. 1990, Phys. Rev. C 41, 566

Zirakashvili, V. N., Breitschwerdt, D., Ptuskin, V. S., & Völk, H. J. 1996, A&A 311, 113