

An X-ray spectral code for optically thin plasmas
Updated version 2.0 12-03-1992

1. Introduction

A new, partly improved and much faster version of Mewe and Gronenschild's plasma code is presented. The structure of the subroutines has been changed, sometimes drastically, and also some in- and output parameters have been changed. For details we refer to the description of the separate subroutines. The primary program is now subroutine FLINE, which calculates the ionisation balance, the continuum and line emission using calls to other subroutines.

The general strategy behind the modification was as follows. First, the older version required input or produced output at several stages in different subroutines, which made it difficult to modify it in order to make it suitable for special purposes of the user. In the new version, no input or output is requested or generated during the run of the program, and all input/output variables are transmitted as subroutine arguments in FLINE. An exception is made for some error messages, which are printed in case of serious problems. These errors should never occur, however, if the program is used properly. Moreover, we removed all usage of common blocks.

Several routines of the old version contained large data blocks; also several constants had to be calculated and if-branchings occurred each time the routines were called, although they could be generated once and forever. We resolved both problems by creating a few data files where those pre-calculated constants are stored; these data files are read once, upon initialisation.

Currently, we especially improved the accuracy of the continuum emissivity to an accuracy of 1 %. A comparison with the old version shows that the old version had sometimes relative errors of some 15 %. Finally, a wrong-placed bracket in one of the formulae for the line-emission in the Leiden version of the old software was put to its right place; this error led to errors of sometimes several tens % in linestrength.

Apart from the "classical" Mewe-Gronenschild routines, there is now also a set of programs which can be used for the calculation of non-equilibrium plasmas. In particular, a program for calculating supernova remnant spectra is included.

There are two types of programs available: a series of first-generation programs in the subdirectory [XSPEC.CREFIL] which generate the basic data files needed for spectral analysis, and which need to run only once; in general it is recommended that normal users of the spectral code do *not* use or modify these programs, but only use the main spectral routines and datafiles in the directory [XSPEC]. Furthermore, there is a directory [XSPEC.SNR] containing programs for non-equilibrium spectra and for supernova remnants.

The programs are open for improvements or extension. Please note the version number of the programs you have. The highest version numbers currently available (19-04-1990) are:

[XSPEC]FLINE	1.3
[XSPEC]GETNAME	1.1
[XSPEC]ERF	1.0
[XSPEC]ABUN	3.0
[XSPEC]IONCON	1.20
[XSPEC]CONEMX	7.41
[XSPEC]LINEMX	7.23
[XSPEC]EXPON1	1.0
[XSPEC]CORFG	1.1
[XSPEC]FBRF	1.0
[XSPEC]HELIUM	1.0
[XSPEC]DENCOR	1.0
[XSPEC]GLINE	1.0

[XSPEC]IONIS	1.0
[XSPEC]FU	1.0
[XSPEC]FACTB	1.0
[XSPEC]RECOMB	1.1
[XSPEC]MEGRON	1.0
[XSPEC]SET1	1.0
[XSPEC]JACOBS	1.0
[XSPEC]INTER	1.0
[XSPEC]SPLCO	1.0
[XSPEC.CREFIL]FILE_CONEMX	1.2
[XSPEC.CREFIL]FILE_FR3NEW	1.0
[XSPEC.CREFIL]FILE_IONIS	1.0
[XSPEC.CREFIL]IONIS	1.0
[XSPEC.CREFIL]POTION	1.0
[XSPEC.CREFIL]NONTE	1.0
[XSPEC.CREFIL]E1	1.0
[XSPEC.CREFIL]FU	1.0
[XSPEC.CREFIL]FACTB	1.0
[XSPEC.CREFIL]EALIBE	1.0
[XSPEC.CREFIL]EAMES	1.0
[XSPEC.CREFIL]SAMP	1.0
[XSPEC.CREFIL]REDA	1.0
[XSPEC.CREFIL]EA3P3D	1.0
[XSPEC.CREFIL]FILE_KARLAT	1.0
[XSPEC.CREFIL]KYLAM	1.0
[XSPEC.CREFIL]RK	1.0
[XSPEC.CREFIL]FILE_LINEFILE	1.0
[XSPEC.CREFIL]TRANS	1.0
[XSPEC.CREFIL]FILE_RECOMB	1.1
[XSPEC.CREFIL]RECOMB	1.0
[XSPEC.SNR]CRERAT	1.01
[XSPEC.SNR]CREFIL	1.01
[XSPEC.SNR]NEISPEC	1.05
[XSPEC.SNR]TIMESTEP	1.02
[XSPEC.SNR]SOLCON	1.03
[XSPEC.SNR]RATESTEP	1.56
[XSPEC.SNR]READ_ARRAY	1.0
[XSPEC.SNR]SNR (driver)	1.0
[XSPEC.SNR]SNRFIT	1.0
[XSPEC.SNR]SNRMOD	2.1
[XSPEC.SNR]SELFSIM	2.0
[XSPEC.SNR]SNRSHELL	2.0
[XSPEC.SNR]ENERG	1.05
[XSPEC.SNR]TIMEHIST	2.11
[XSPEC.SNR]SPINT	1.0
[XSPEC.SNR]BAND	1.03
[XSPEC.SNR]BANDSHAPE	1.01
[XSPEC.SNR]BANDFAST	1.0
[XSPEC.SNR]CHEVAL	1.04
[XSPEC.SNR]OUTER	1.0
[XSPEC.SNR]INNER	1.0
[XSPEC.SNR]SEDOV	1.21
[XSPEC.SNR]LSEDOV	1.0
[XSPEC.SNR]ISOTHERM	1.02
[XSPEC.SNR]ISOINT	1.0
[XSPEC.SNR]PROSPEC	1.0
[XSPEC.SNR]USTEL	1.0

2. Main subroutines in the directory [XSPEC].

Subroutine FLINE(E1, E2, FLUX, NBIN, IC, ITYPE, IEM, ID, CEM, H, T, FACTOR, DILFAC, RADT, XZIN, SION, ALFA, ED, ILX)

This subroutine is the driving routine for the plasma code. It calculates the ionisation balance and the X-ray spectrum for an optically thin, collisionally dominated, non-relativistic plasma.

The program first calculates the ionisation balance. There are 6 possible models for the ionisation balance, and there is also a possibility to choose your own ion concentrations from your calling program. Next the continuum emissivity and subsequently the line emissivity is calculated. If requested the lines are Doppler-broadened, and finally the output is converted into a photon spectrum as received at Earth, or into emissivity (per unit volume).

In your calling program or other subprograms, do not leave logical unit number 80 open while calling FLINE, since FLINE opens and closes this unit several times for reading its internal parameters. Unit 80 is the only unit used internally.

The input/output parameters are as follows.

- * E1(NBIN) – Real*4 array of length NBIN, containing the lower bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * E2(NBIN) – Real*4 array of length NBIN, containing the upper bounds of the energy bins. Units keV. This array is input and unchanged upon output. *Note:* The continuum emissivity (photons/keV/etc.) is determined at the *centre* of the energy bin (i.e. at $0.5(E1(I) + E2(I))$, for $I = 1, \dots, NBIN$, while the line emissivity is the *average* over the entire bin.
- * FLUX(NBIN) – Real*4 array of length NBIN, containing the output emissivity or photon spectrum, depending upon IEM (see there). This array is output.
- * NBIN – Integer*4, number of energy bins requested from the calling program. Input. Unchanged upon output. There is no upper limit to this parameter (apart from the memory size of your computer).
- * IC – Integer*4, type of ion concentrations which should be used. Input, unchanged upon output. Allowed values:
 - IC=1: Arnaud and Rothenflug (1985).
 - IC=2: Gronenschild and Mewe (1985).
 - IC=3: High density plasma, mainly from Jacobs *et al.*
 - IC=4: Input your own ion concentrations from the calling program.
 - IC=5: Raymond and Smith (currently not installed).
 - IC=6: Arnaud and Rothenflug (1985) with Fe XVII from Smith *et al.* (1985).
 - IC=7: Arnaud and Rothenflug (1985), with dielectronic recombination modified by Mewe.
- * ITYPE – Integer*4, type of calculation requested. Input, unchanged upon output. Allowed values:
 - ITYPE=0: Calculate only the transition rates and ion concentrations (no spectrum). This option is used for calculating non-equilibrium plasmas. Do not use with IC=4.

ITYPE=1: Calculate transition rates, ion concentrations (if not yet given) and continuum and line emission.

ITYPE=2: Calculate transition rates, ion concentrations (if not yet given) and only the continuum emission.

ITYPE=3: Calculate only line strength's and line energies; in this case, input E1(1) as the lower energy boundary, E1(2) - as the upper energy boundary; on output E2(1)-E2(NBIN) contain the line energies, FLUX(1)-FLUX(NBIN) the line fluxes or emissivities as specified by IEM, except not per keV (so: phot/cm**2/s or phot/m**3/s). also NBIN is now output!

* IEM - Integer*4, type of output spectrum. Input, unchanged upon output. Allowed values:

IEM=1: the output spectrum FLUX(NBIN) contains a photon spectrum in 10^4 photons s^{-1} $keV^{-1}m^{-2}$. The photon spectrum is the photon spectrum received at Earth (apart from interstellar absorption). A value for CEM should be specified (see later).

IEM=2: the output spectrum FLUX(NBIN) contains the emissivity in photons s^{-1} $keV^{-1}m^{-3}$. A value for CEM is not required (any input value will be denied).

* ID - Integer*4, type of line profile. Input, unchanged upon output. Only required for ITYPE=1. Allowed values:

ID=0: no line broadening (i.e. all spectral lines are assumed to be infinitely narrow).

ID=1: Thermal Doppler broadening of the lines, assuming $T_e = T_i$. Account is taken of the different mass of each element (the internal array AMUS contains the square root of the nuclear mass of each element, expressed in a.m.u., and accounting for the average isotopic abundance ratio's following Lang (1974), table 38. Broadening is taken into account up to at least ~ 3 -20 Doppler widths (depending upon the element).

* CEM - Real*4, reduced emitting volume. Input, unchanged upon output. It defined by $CEM = V/d^2$ where V is the emitting volume and d the distance to the source. Units: $10^{50} cm^3 pc^{-2} = 10^{44} m^3 pc^{-2} = 1.0503 \times 10^{11} m$. If Y is the "classical" emission measure for a homogeneous source, defined by $Y = \int n_e n_H dV$, then $CEM = Y/[n_H^2 (n_e/n_H) d^2]$ where n_e/n_H is the output parameter ED of this subroutine, and n_H the input parameter H. *Note:* the definition of CEM is slightly different from older versions of the code, where it incorporated the square of the density. CEM is only required for IEM=1; for IEM=2 it is denied.

* H - Real*4, Hydrogen density in the source in units of $10^6 m^{-3} = 1 cm^{-3}$. Input, unchanged upon output. Do not try to put H to zero (empty space's spectrum is rather dull). Fitting plasmas with very low density, one could fix H=1.0 and vary CEM.

* T - Real*4, Electron temperature in keV. Input, unchanged upon output.

* FACTOR(15) - Real*4 array, containing the relative abundances of the elements with respect to Solar abundances (for solar abundances, $FACTOR(I) = 1.0$ for $I = 1, \dots, 15$). *Warning:* the Solar abundances have been changed since the previous version of this program (see subroutine ABUN). Currently 15 elements are present.

* DILFAC - Real*4, dilution factor of the radiation. Input, unchanged upon output. This quantity is dimensionless. For a full description, see Mewe and Gronenschild (it is only important for the branching ratio of forbidden He-like lines). If you want to skip this parameter, put DILFAC=0. upon input.

* RADT - Real*4, radiation temperature. Input, unchanged upon output. Used in similar context as DILFAC above. *Warning:* units are Kelvin (instead of keV)! If you are not interested in this

parameter, put $\text{RADT} = 0$.

- * $\text{XZIN}(207)$ – Real*4 array, ionic concentrations with respect to total Hydrogen density. Currently all 207 stages of 15 elements are included. This array is requested for input if $\text{IC}=4$, it is output for $\text{IC}=1-3, 5-7$.
- * $\text{SION}(207)$ – Real*4 array, ionisation rates for all the 207 ions in $10^{-6} \text{ m}^3\text{s}^{-1} = 1 \text{ cm}^3\text{s}^{-1}$. Output. (Used for non-equilibrium calculations). Usually, just declare the array in the calling program and don't bother about its value.
- * $\text{ALFA}(207)$ – Real*4 array, recombination rates for all the 207 ions in $10^{-6} \text{ m}^3\text{s}^{-1} = 1 \text{ cm}^3\text{s}^{-1}$. Output. (Used for non-equilibrium calculations). Usually, just declare the array in the calling program and don't bother about its value. Of course, for $\text{IC}=4$, but also for $\text{IC}=3$ and 5, both SION and ALFA are *not* determined.
- * ED – Real*4, electron density in units of the Hydrogen density. Output.
- * $\text{ILX}(2, \text{NL_MAX})$ – Integer*2 array, identifies the spectral lines; $\text{ILX}(1, \text{I})$ contains the number of the line (as in FILLIN), $\text{ILX}(2, \text{NL_MAX})$ contains the element producing the line (numbered 1–15). NL_MAX is a parameter (currently 3000) which gives the maximum number of lines present in the code.

Subprograms called:

- ++ GETNAME – to get the names of some input data files.
- ++ ABUN – to determine the absolute elemental abundances
- ++ IONCON – to determine the ion concentrations
- ++ CONEMX – to calculate the continuum emission
- ++ LINEMX – to calculate the line emission
- ++ ERF – for Doppler broadening

Subroutine $\text{GETNAME}(\text{FILION}, \text{FILREC}, \text{FILREM}, \text{FILJAC}, \text{FILRAS}, \text{FILLIN}, \text{FILSPE}, \text{FILKAR}, \text{FILCAF})$

In this subroutine the file names of the basic input files are determined. This is the *only* place in all routines where those names are given explicitly. So, upon installing this program, check whether the names and directories of the data files are correct. In all programs, a fixed size of 50 characters for each filename is used.

The parameters (all character*50 and output) are as follows:

- * FILION – filename of ionisation data of Arnaud and Rothenflug
- * FILREC – filename of recombination data of Arnaud and Rothenflug
- * FILREM – filename of recombination data of Arnaud and Rothenflug, with dielectronic recombination modified by Mewe
- * FILJAC – filename of ion concentrations of Jacobs et al.
- * FILRAS – filename of ion concentrations of Raymond and Smith

- * **FILLIN** – filename of spectral line data
- * **FILSPE** – filename of ionic data for continuum emission
- * **FILKAR** – filename of free-free Gaunt-factor table.
- * **FILCAF** – filename of the Ca and Fe satellite data

Function ERF(X)

This subroutine calculates the error function using the approximation of Abramowitz and Stegun 7.1.9 and 7.1.25, which has an absolute accuracy of 2.5×10^{-5} . Although more accurate (but more cpu-consuming) formulae are known, we do not need them here because the present accuracy is far sufficient for our purpose (determining the line-broadening in subroutine FLINE).

The parameters are:

- * X – Real*4, input, unchanged upon output; argument of the error function.
- * ERF – Real*4, error function value; output.

Subroutine ABUN(FACTOR, XE)

This subroutine determines the elemental abundances with respect to Hydrogen from the abundances relative to the Solar values. In the old versions, the Solar abundances were from Allen (1973), who referred to work from the early sixties. In an intermediate code, we adopted the newer values for the abundances from Ross and Aller (1976), where also a discussion on the accuracy of this standard set is found. In the newest code, we use Anders & Grevesse (1989). Table 1 lists the 15 elements used in this program and the abundances (number of nuclei per Hydrogen nucleus) according to Anders & Grevesse, Ross & Aller and Allen. Note that some sets of abundances show large discrepancies, especially for Ar and Ne.

Table 1. Elemental abundances

Element	Anders and Grevesse (1989)	Ross and Aller (1976)	Allen (1973)
H	$\equiv 1$	$\equiv 1$	$\equiv 1$
He	.0977	.0631	.0850
C	3.63×10^{-4}	4.17×10^{-4}	3.31×10^{-4}
N	1.12×10^{-4}	8.71×10^{-5}	9.12×10^{-5}
O	8.51×10^{-4}	6.92×10^{-4}	6.61×10^{-4}
Ne	1.23×10^{-4}	3.72×10^{-5}	8.32×10^{-5}
Na	2.14×10^{-6}	1.91×10^{-6}	1.78×10^{-6}
Mg	3.80×10^{-5}	3.98×10^{-5}	2.63×10^{-5}
Al	2.95×10^{-6}	3.31×10^{-6}	2.45×10^{-6}
Si	3.55×10^{-5}	4.47×10^{-5}	3.31×10^{-5}
S	1.62×10^{-5}	1.59×10^{-5}	1.58×10^{-5}
Ar	3.63×10^{-6}	1.00×10^{-6}	6.31×10^{-6}
Ca	2.29×10^{-6}	2.24×10^{-6}	2.00×10^{-6}
Fe	4.68×10^{-5}	3.16×10^{-5}	3.98×10^{-5}
Ni	1.78×10^{-6}	1.91×10^{-6}	2.00×10^{-6}

The parameters are as follows.

- * FACTOR(15) – Real*4 array, containing the elemental abundances with respect to Solar values for all 15 elements. Input, unchanged upon output. For solar abundances, take FACTOR(I) = 1. for I = 1, ..., 15.
- * XE(15) – Real*4 array, containing the elemental abundances with respect to Hydrogen.

Warning: always take care to take FACTOR(1) = 1!

Subroutine IONCON(T, IC, XE, XZIN, ALFA, SION, ED, FILION, FILREC, FILREM, FILJAC, FILRAS)

This subroutine determines the ionic concentrations and calculates the electron density (with respect to Hydrogen). At the start of the subroutine, there is a branching corresponding to different possibilities for the ionisation balance. There are currently 7 models for the ionisation balance available. *Note:* The Mewe/Gronenschild and Arnaud/Rothenflug models assume that Hydrogen is completely ionised. The parameters have the following meaning:

- * T – Real*4, electron Temperature in keV. Input, unchanged upon output.
- * IC – Integer*4, ion concentrations model which should be used. Input, unchanged upon output. Allowed values:
 - IC=1: Arnaud and Rothenflug (1985).
 - IC=2: Gronenschild and Mewe (1985).
 - IC=3: High density plasma, mainly from Jacobs *et al.*.
 - IC=4: The ion concentrations are not calculated but are given as input from the main program.
 - IC=5: Raymond and Smith.
 - IC=6: Arnaud and Rothenflug (1985) with Fe XVII from Smith *et al.* (1985).
 - IC=7: Arnaud and Rothenflug (1985), with dielectronic recombination modified by Mewe.
- Warning:* Once you have chosen either one of the options 1, 6, or 7 in the calling program, it is impossible to change to any other option of the subset {1, 6, 7} during the same run of the program! (But it is possible to change to modes 2–5).
- * XE(15) – Real*4 array, containing the elemental abundances with respect to Hydrogen. Input, unchanged upon output.
- * XZIN(207) – Real*4 array, ionic concentrations with respect to total Hydrogen density. Currently all 207 stages of 15 elements are included. This array is requested for input if IC=4, it is output for IC=1–3.
- * ALFA(207) – Real*4 array, recombination rates for all the 207 ions in $10^{-6} \text{ m}^3\text{s}^{-1} = 1 \text{ cm}^3\text{s}^{-1}$. Output. For IC=3 or 4, both SION and ALFA are *not* determined.
- * SION(207) – Real*4 array, ionisation rates for all the 207 ions in $10^{-6} \text{ m}^3\text{s}^{-1} = 1 \text{ cm}^3\text{s}^{-1}$. Output.
- * ED – Real*4, electron density with respect to Hydrogen density. Output.
- * FILION – Character*50, filename of ionisation data of Arnaud and Rothenflug. Input, unchanged upon output. item* FILREC – Character*50, filename of recombination data of Arnaud and Rothenflug. Input, unchanged upon output. item* FILREM – Character*50, filename of recombination data of Arnaud and Rothenflug, modified by Mewe. Input, unchanged upon output.
- * FILJAC – Character*50, filename of ion concentrations of Jacobs. Input, unchanged upon output.
- * FILRAS – Character*50, filename of ion concentrations of Raymond and Smith. Input, unchanged upon output.

Subprograms called:

- ++ IONIS – calculates the ionisation rates SION according to Arnaud and Rothenflug.
- ++ RECOMB – calculates the recombination rates ALFA according to Arnaud and Rothenflug.
- ++ MEGRON – calculates ionisation rates SION and recombination rates ALFA according to Mewe and Gronenschild.
- ++ JACOBS – calculates the ionic concentrations according to Jacobs *et al.* and Raymond and Smith.

Table 2. Comparison of free-free Gaunt factor of Gronenschild and Mewe (1978) ($G_{ff,GM}$) with Carson (1987) ($G_{ff,C}$). The tabulated quantity is $G_{ff,GM}/G_{ff,C} - 1$.

$E/kT \setminus Z^2\text{Ryd}/kT$	-4.00	-3.00	-2.00	-1.00	.00	1.00	2.00
-4.00	-.002	.003	.027	.052	.048	.026	-.002
-3.50	-.003	.003	.028	.054	.053	.031	-.004
-3.00	-.003	.003	.029	.056	.057	.035	-.010
-2.50	-.003	.003	.028	.056	.060	.037	-.023
-2.00	-.004	.001	.026	.052	.058	.036	-.038
-1.50	-.005	-.001	.020	.039	.047	.034	-.040
-1.00	-.006	-.003	.009	.012	.023	.035	-.020
-.50	-.007	-.003	-.000	-.029	-.016	.040	.007
.00	-.002	.010	.006	-.063	-.069	.029	.023
.50	.020	.053	.044	-.066	-.121	-.005	.018
1.00	.080	.137	.092	-.035	-.135	-.053	-.005
1.50	.219	.243	.049	-.037	-.079	-.081	-.035
2.00	.489	.269	-.278	-.252	.047	-.029	-.063

Subroutine CONEMX(E1, E2, FLX, NEMX, T, XZIN, FILSPE, FILKAR)

This subroutine calculates the continuum emission for an optically thin plasma, including free-free, free-bound and two-photon emission. This subroutine was farmost the most CPU time consuming part of the code in the older versions. Since then, the required CPU time has been reduced by a factor of ~ 4 , and also several improvements have been made in order to increase its accuracy. These improvements are discussed below. The overall relative accuracy should now be about 1%.

Gronenschild and Mewe (1978) have given a formula for the *free-free Gaunt factor*, based upon the Born approximation and a correction factor. This formula contained a modified Bessel function, square roots and logarithms and was the largest contributor to the CPU time consumed, since the formula was evaluated for each ion contributing to the emission and for all energy bins. The formula was an approximation to the calculations of Karzas and Latter (1961) and had an accuracy of better than 15 %. Karzas and Latter presented their results only graphically; since then, Carson (1987) presented extensive tables of the free-free Gaunt factor, reproducing the results of Karzas and Latter.

Table 2 shows the differences of Gronenschild and Mewe's Gaunt factor with the exact results of Carson. It is evident that especially in the temperature range of 10^4 – 10^7 K (for Hydrogen) and for $E > kT$ the errors can be quite substantially. Therefore we have chosen to calculate the free-free Gaunt factor by interpolation of Carson's table. With bilinear interpolation of the logs of the Gaunt factor on Carson's logarithmic energy and temperature grid the required accuracy of 1 % was reached. Carson lists the Gaunt factor in the energy range $10^{-4}kT < E < 100kT$. For larger energies no Gaunt factor is required since the exponential prefactor cuts off the spectrum at high energies. For lower energies, we used the asymptotic dependence $G = a(\gamma^2) + (\sqrt{3}/\pi) \ln(E/kT)$, where $\gamma^2 = Z^2\text{Ryd}/kT$. As stated before, the present calculations are nonrelativistic; Carson's tables are valid for $T < 136$ keV ($10^{-4} < \gamma^2 < 100$). For larger temperatures (better: smaller γ^2), we simply took the minimum value for $\gamma^2 = 10^{-4}$. Similarly, for very low temperatures ($T < 0.136$ eV, or about 1000 K) we took the maximum value for γ^2 . But in this regime the calculations are irrelevant anyhow since then the atoms are essentially neutral, so free-free radiation is impossible.

The calculations of Carson are non-relativistic. Relativistic corrections were calculated by Kylafis and Lamb (1982), eqn. (B4). These authors calculate the relativistic correction with respect to the Born approximation. The relative accuracy of the Born approximation (in the non-relativistic limit) with respect to the results of Carson is shown in table 3. As is obvious, the Born approximation is good for high energies, and for that reason Kylafis and Lamb made the same approximations as for the Born approximation (apart of course from relativistic effects). Table 4 lists Kylafis and Lamb's correction factor (minus 1) and it is evident from this table that in some parts of our parameter space (especially

Table 3. Comparison of free-free Gaunt factor $G_{ff,Born}$ in the non-relativistic Born approximation with Carson (1987) ($G_{ff,C}$). The tabulated quantity is $G_{ff,Born}/G_{ff,C} - 1$.

$E/kT \setminus Z^2\text{Ryd}/kT$	-4.00	-3.00	-2.00	-1.00	.00	1.00	2.00
-4.00	-.001	-.001	.003	.026	.102	.245	.441
-3.50	-.001	-.001	.003	.028	.114	.277	.506
-3.00	-.001	-.001	.003	.030	.125	.312	.576
-2.50	-.002	-.002	.001	.029	.133	.343	.638
-2.00	-.002	-.005	-.005	.019	.130	.356	.660
-1.50	-.004	-.009	-.019	-.007	.097	.319	.594
-1.00	-.006	-.017	-.044	-.062	.012	.196	.401
-.50	-.010	-.030	-.083	-.158	-.143	-.024	.097
.00	-.015	-.047	-.133	-.282	-.351	-.299	-.238
.50	-.021	-.065	-.184	-.400	-.549	-.551	-.523
1.00	-.026	-.080	-.226	-.487	-.689	-.732	-.721
1.50	-.030	-.090	-.253	-.542	-.769	-.841	-.843
2.00	-.032	-.097	-.270	-.573	-.812	-.898	-.912

Table 4. Comparison of relativistic free-free Gaunt factor ($G_{ff,Rel.}$) with the nonrelativistic Born approximation ($G_{ff,Born}$). The tabulated quantity is $G_{ff,Rel.}/G_{ff,Born} - 1$. Numbers in italics are for photon energies larger than 400 keV, where the approximation to the relativistic Gaunt factor breaks down.

$E/kT \setminus Z^2\text{Ryd}/kT$	-4.00	-3.00	-2.00	-1.00	.00	1.00	2.00
-4.00	.108	.006	.000	.000	.000	.000	.000
-3.50	.111	.006	.000	.000	.000	.000	.000
-3.00	.116	.006	.001	.000	.000	.000	.000
-2.50	.123	.006	.001	.000	.000	.000	.000
-2.00	.132	.007	.001	.000	.000	.000	.000
-1.50	.146	.008	.001	.000	.000	.000	.000
-1.00	.171	.009	.001	.000	.000	.000	.000
-.50	.222	.012	.001	.000	.000	.000	.000
.00	.366	.018	.002	.000	.000	.000	.000
.50	.987	.039	.003	.000	.000	.000	.000
1.00	<i>5.536</i>	.125	.008	.001	.000	.000	.000
1.50	<i>63.045</i>	.655	.026	.002	.000	.000	.000
2.00	<i>1211.239</i>	<i>5.694</i>	.107	.007	.001	.000	.000

for high temperatures) the correction factor can not be neglected. Discarding the difference between the non-relativistic Born approximation and Carson's results at the temperatures and energies where the relativistic correction is important, we simply approximated the true Gaunt factor by Carson's value times Kylafis and Lamb's correction factor. *Note:* The correction of Kylafis and Lamb is calculated for Hydrogen. Since at high temperatures all ions are nearly completely stripped of electrons, $Z^2 \geq 4$ for all ions other than Hydrogen, and thus the corresponding γ^2 is also 4 times larger, resulting in a much smaller (negligable?) relativistic correction.

Mewe, Lemen and van den Oord (1986) approximated the *free-bound Gaunt factor* by assuming that the individual Gaunt factor $g_{fb}(n)$ for each (excited) state with principal quantum number n did not depend upon energy, and that the edge energies $E_n = E_{n_0+1}$ for $n > n_0 + 1$, where n_0 is the principal quantum number of the ground state. Here we take into account the full energy dependence of $g_{fb}(n)$, and also the correct edge energies for all $n \leq 15$. As in Mewe, Lemen and van den Oord, we approximate for the excited states $\zeta_n = 2n^2$, and $z_n = z$, the charge of the recombining ion Z^{+z} .

The individual gaunt factors $g_{fb}(n, E)$ were tabulated by Karzas and Latter (1961). We found the following approximation to their results:

$$g_{fb}(n, E) = (((a_{n,4}x + a_{n,3})x + a_{n,2})x + 1)xa_{n,1}$$

where

$$x \equiv 1/(1 + \sqrt{E/E_1})$$

and

$$E_1 \equiv n^2 E_n = E_H z_n^2.$$

The above formula has an absolute accuracy of < 0.0059 for all n (for $n = 1$ even better than 0.0004). Since in most of the energy region of interest the Gaunt factor is of order unity, this absolute error is sufficiently small for our purposes. (The relative error is everywhere better than 0.023 , with the largest value occurring for very large energies).

The difference between the "old" free-bound Gaunt factor and the new one can be up to 5–10 %.

Finally, we improved the *two-photon Gaunt factor*. Gronenschild and Mewe (1978) approximated this with the square root of a cosine. The accuracy of that approximation is some 14 %. Here we interpolate directly in the table of Spitzer and Greenstein (1961) for Hydrogenic ions, and in a graph in Dalgarno and Drake (1969) for the Helium-like ions. An accuracy of better than 0.007 times the maximum of the distribution function $\phi(y)$ is reached if we interpolate linearly on a grid of 51 points spaced evenly from $y = 0$ to $y = 1$. An impression of the differences with the old version is given in table 5.

In the old version of the program an error was made with the two-photon emission of Calcium. Instead of taking the ionic concentrations for the H- and He-like ions, the concentrations for the Li- and Be- like ions were taken. This error has now been removed.

Finally we modified the normalisation of the two-photon emissivity. The contribution to the total two-photon Gaunt factor of ion (Z, z) can be written as

$$G_{2p,Z,z}(E) = 164995 \frac{E}{E_{Z,z}^2} \phi\left(\frac{E}{E_{Z,z}}\right) f_{ij} \bar{g}\left(\frac{E}{E_{Z,z}}\right) e^{(E-E_{Z,z})/kT}$$

where the constant is $3\pi^2 a_0 m_e^2 c^3 / (2000eh)$ and all input quantities are in SI, and E is expressed in keV, f_{ij} is the oscillator strength and $\bar{g}(y)$ the average excitation gaunt factor. We take f_{ij} from Mewe (1977), which differs in the last digit from Gronenschild and Mewe (1978), i.e. $f=0.415$ for H-like ions and $f = 0.830(1 - 1.34/Z)$ for He-like ions. For the average Gaunt factor we follow Mewe (1972) which seems to be different from Mewe and Schrijver (1978): for the Helium sequence, $\bar{g} = 0.05$ (and not 0.045 as taken by Gronenschild and Mewe (1978) and in the old code); for the Hydrogen sequence, $\bar{g} = 0.055$ for Carbon and higher elements, and g is the expression for the Gaunt factor of Mewe (1972) with $A = 0.08$, $B = -0.16$, $C = 0.11$, $D = 0$ for He II. *Warning:* the two-photon emission of H I and He I still has to be implemented!

Table 5. Two photon emissivity $\phi(y)$

y	Gronenschild-Mewe	H-like	He-like
0.00	0.00	0.00	0.00
0.05	1.04	0.92	0.47
0.10	1.46	1.48	1.18
0.20	2.01	2.10	2.07
0.30	2.36	2.41	2.54
0.40	2.56	2.56	2.78
0.50	2.62	2.60	2.86

The *structure* of the subroutine CONEMX is as follows. After initialisation of some constants, which is done the first time the routine is called, the subroutine calculates some temperature-dependent coefficients for each of the ions and emission processes involved. Then a large loop over the input energy array follows; for each energy, a loop over all ions in which the free-free, two-photon and free-bound Gaunt factor is calculated is done. The output flux is determined apart from a (temperature dependent)

normalisation factor; this normalisation factor is determined in the calling routine, FLINE, since it is identical to the normalisation factor for the line emission of routine LINEMX. *Warning:* not all the variables called GFF, GFB and G2P are the actual Gaunt factors! Some (exponential) prefactors have been absorbed into them because of computational reasons.

There are some cut-offs in the program to avoid unnecessary calculation of negligible quantities and arithmetic overflow. These are as follows. The relevant scale parameters are C_{\max} and U_{\max} . C_{\max} (currently put to .001) is a cut-off for too small amplitudes of the Gaunt factor. U_{\max} (currently put to 69.0, or $e^{U_{\max}} = 9.25 \times 10^{29}$) is a factor which governs all exponential factors. Continuum emission is only calculated for photon energies

$$E/kT < U_{\max}.$$

The *free-free* emission is only calculated if

$$Z_{eff}^2 X_{(Z,z)} > C_{\max}$$

where Z_{eff} is the effective charge of the ion and $X_{(Z,z)}$ its relative abundance to the total Hydrogen density. Should in any case the photon energy be larger than 511.003 keV, then the relativistic correction of Kylafis and Lamb to the free-free Gaunt factor is calculated assuming $E = 511.003$ keV (this is actually done in the program which generates the free-free Gaunt-factor table). The *two-photon* emission is only calculated if

$$164995 E_{Z,z}^{-1} f_{ij} \bar{g}\left(\frac{E}{E_{Z,z}}\right) X_{(Z,z)} = G_{2p,Z,z}(E) \frac{E_{Z,z}}{E} e^{(E_{Z,z}-E)/kT} \phi\left(\frac{E}{E_{Z,z}}\right) X_{Z,z} > C_{\max}$$

and if

$$E/kT < E_{2p}/kT < U_{\max}.$$

Finally, *free-bound* emission is only calculated if the edge energy E_n satisfies

$$E_n/kT < E/kT < U_{\max},$$

$$(E - E_n)/kT < U_{\max},$$

$$a_{n,1} \zeta_0 z_0^4 X_{(Z,z)} \frac{E_H}{kT} e^{E_n/kT} / n_0^5 < C_{\max}$$

for the ground state and

$$2a_{n,1} z^4 X_{(Z,z)} \frac{E_H}{kT} e^{E_n/kT} / n^3 < C_{\max}$$

for excited states, and if at the energy E the relative contribution of higher states compared to the sum of all non-zero lower states exceeds C_{\max} .

The parameters of CONEMX are:

- * E1(NEMX) – Real*4 array of length NEMX, containing the lower bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * E2(NEMX) – Real*4 array of length NEMX, containing the upper bounds of the energy bins. Units keV. This array is input and unchanged upon output. *Note:* The continuum emissivity is determined at the *centre* of the energy bin (i.e. at $0.5(E1(I) + E2(I))$, for $I = 1, \dots, NEMX$)
- * FLX(NEMX) – Real*4 array of length NEMX, containing the continuum emissivity, to be multiplied by a few constants. This array is output.
- * NEMX – Integer*4, number of energy bins requested from the calling program. Input. Unchanged upon output.
- * T – Real*4, electron Temperature in keV. Input, unchanged upon output.

- * XZIN(207) – Real*4 array, ionic concentrations with respect to total Hydrogen density. Input, unchanged upon output.
- * FILSPE – Character*50, file name of data with ionic constants. Input, unchanged upon output.
- * FILKAR – Character*50, file name of the free-free Gaunt factors. Input, unchanged upon output.

Subroutine LINEMX(FILLIN, FILCAF, E1, E2, XKT, XZIN, ELDEN, DILFAC, RADT, ELX, FLX, ILX, NLX)

This subroutine calculates the line emission. The formulae are essentially unchanged compared to the older version. But the number of Fortran lines has been reduced by 50 %.

The basic parameters are:

- * FILLIN – Character*50, the file name of the file containing the line parameters. Input, unchanged upon output.
- * FILCAF – Character*50, the file name of the file containing the Ca and Fe satellite line parameters. Input, unchanged upon output.
- * E1 – Real*4 containing the lowest energy for which you want to know the line spectrum, in keV. Usually, take it as E1(1) where E1(I) is defined in FLINE. Input, unchanged upon output.
- * E2 – Real*4 containing the largest energy for which you want to know the line spectrum, in keV. Usually, take it as E2(NBIN) where E2(I) is defined in FLINE. Input, unchanged upon output.
- * XKT – Real*4, electron Temperature in keV. Input, unchanged upon output.
- * XZIN(207) – Real*4 array, ionic concentrations with respect to total Hydrogen density. Input, unchanged upon output.
- * ELDEN – Real*4, containing the electron density in $10^6 \text{ m}^{-3} = \text{cm}^{-3}$. Input, unchanged upon output.
- * DILFAC – Real*4, dilution factor of the radiation. Input, unchanged upon output. This quantity is dimensionless. For a full description, see Mewe and Gronenschild (it is only important for the branching ratio of forbidden He-like lines).
- * RADT – Real*4, radiation temperature. Input, unchanged upon output. Used in similar context as DILFAC above. *Warning:* units are Kelvin (instead of keV)!
- * ELX(NL_MAX) – Real*4 array, containing the energies in keV of all the lines which are significant in the spectrum. Output. *Warning:* the program does not check if the actual number of lines is larger than NL_MAX. Currently, NL_MAX = 3000 in FLINE is sufficient.
- * FLX(NL_MAX) – Real*4 array, containing the line photon fluxes, to be multiplied by a few constants. This array is output.
- * ILX(NL_MAX) – Integer*2 array, element number. line nr. I at energy ELX(I) with photon flux FLX(I) originates from element ILX(I). Note that currently 15 elements are included, so $1 \leq \text{ILX} \leq 15$. This parameter is usefull to identify individual lines later and it is requested for calculating Doppler-broadened line profiles. Output.
- * NLX – integer*4, number of lines actually contributing to the line emission between E1 and E2. The arrays ELX, FLX and ILX are filled up to array element NLX. Output.

Subprograms called:

++ GLINE – line Gaunt factor.

++ CORFG – corrects the F*G factor cf. Mewe, Gronenschild and van den Oord (1985).

++ DENCOR – determines a density correction to the line strength.

Function EXPON1(X)

This function calculates the first exponential integral $E_1(x)$, using Abramowitz and Stegun 5.1.53. Accuracy: better than 2×10^{-7} . Parameters:

- * X – Real*4, the argument x . Input, unchanged upon output.
- * EXPON1 – Real*4, $E_1(x)$. Output.

Subroutine CORFG(F, G, FG, EL, Y, T, XZIN, NZZ, RP, LP, ELDEN, DILFAC, RADT)

This subroutine calculates a correction on the F*G value. For a full description of the theory, see Mewe, Gronenschild and van den Oord (1985).

The parameters are as follows

- * F – Real*4, F-value. Input, unchanged upon output.
- * G – Real*4, G-value. Input, unchanged upon output.
- * FG – Real*4, corrected F*G value. Output.
- * EL – Real*4, line energy in keV. Input, unchanged upon output.
- * Y – Real*4, aE/kT , where a is a correction factor defined by MGO. Input, unchanged upon output.
- * T – Real*4, electron Temperature in K. Input, unchanged upon output.
- * XZIN(207) – Real*4 array, ionic concentrations with respect to total Hydrogen density. Input, unchanged upon output.
- * NZZ – Integer*4, index number of the radiating ion (XZIN(NZZ) gives its relative abundance). Input, unchanged upon output.
- * RP(14) – Real*4 array containing some ionic constants. Input, unchanged upon output.
- * LP(9) – Integer*2 array containing some ionic constants. Input, unchanged upon output.
- * ELDEN – Real*4, containing the electron density in $10^6 \text{ m}^{-3} = \text{cm}^{-3}$. Input, unchanged upon output.
- * DILFAC – Real*4, dilution factor of the radiation. Input, unchanged upon output. This quantity is dimensionless. For a full description, see Mewe and Gronenschild (it is only important for the branching ratio of forbidden He-like lines).
- * RADT – Real*4, radiation temperature. Input, unchanged upon output. Used in similar context as DILFAC above. *Warning:* units are Kelvin (instead of keV)!
- * LNUM – Integer*2, containing line number (cf. P. Majer)
- * IDNR(NCAFE) – Integer*2 array, containing the line ident. numbers of the Ca and Fe satellites to be treated separately
- * CDRCAFE(NCAFE) – Real*4 array, containing the correction factors for the Ca and Fe satellites
- * NCAFE – Integer*4, the number of Ca and Fe satellite lines available.

Subprograms called:

- ++ FBRF – Branching ratio for forbidden He-like line
- ++ HELIUM – Forbidden and intercombination He-like lines
- ++ EXPON1 – First exponential integral

Function FBRF(Z, T5, ELDEN, DILFAC, RADT, BRI, IEL)

This function calculates the branching ratio for the forbidden He-like line. For a full description of the theory, see Mewe, Gronenschild and van den Oord (1985).

The parameters are:

- * Z – Real*4, atomic number. Input, unchanged upon output.
- * T5 – Real*4, electron temperature in units of 10^5 K. Input, unchanged upon output.
- * ELDEN – Real*4, containing the electron density in $10^6 \text{ m}^{-3} = \text{cm}^{-3}$. Input, unchanged upon output.
- * DILFAC – Real*4, dilution factor of the radiation. Input, unchanged upon output. This quantity is dimensionless. For a full description, see Mewe and Gronenschild (it is only important for the branching ratio of forbidden He-like lines).
- * RADT – Real*4, radiation temperature. Input, unchanged upon output. Used in similar context as DILFAC above. *Warning:* units are Kelvin (instead of keV)!
- * BRI – Real*4, see Mewe and Gronenschild. Input, unchanged upon output.
- * IEL – Real*4, element number (presently between 1–15).
- * FBRF – Real*4, function value, the branching ratio. Output.

Function HELIUM(BRI, BRF, F5, F6, ITR)

This function calculates the contribution of the forbidden and intercombination He-like lines. For a full description of the theory, see Mewe, Gronenschild and van den Oord (1985).

The parameters are:

- * BRI – Real*4, see Mewe and Gronenschild. Input, unchanged upon output.
- * BRF – Real*4, see Mewe and Gronenschild. Input, unchanged upon output.
- * F5 – Real*4, see Mewe and Gronenschild. Input, unchanged upon output.
- * F6 – Real*4, see Mewe and Gronenschild. Input, unchanged upon output.
- * ITR – Integer*4, transition number. Input, unchanged upon output.

Subroutine DENCOR(DCOR, ETA, TAU, RP, LP3)

This subroutine calculates a density correction factor with which the F*G value has to be multi-

plied.

The parameters are:

- * DCOR – Real*4, the density correction factor. Output.
- * ETA – Real*4, the electron density in units of $10^{18} \text{ m}^{-3} = 10^{12} \text{ cm}^{-3}$. Input, unchanged upon output.
- * TAU – Real*4, the electron temperature in 10^7 K . Input, unchanged upon output.
- * RP(14) – Real*4 array, containing the required line-dependent parameters. Input, unchanged upon output.
- * LP3 – Integer*4, type of ion, i.e. 1: H-like, 2: He-like, 3: Li-like etc.

Function GLINE(Y, A, B, C, D, E)

This function calculates the Gaunt-factor for line emission. The formula used is Mewe and Schrijver (1978), eqn. 36. For

$$g(U) = A + B/U + C/U^2 + 2D/U^3 + E \ln(U)$$

with $U = E_{ex}/kT$ and $y = E/kT$ the Gaunt factor is given by

$$G(y) = A + (By - Cy^2 + Dy^3 + E)e^y E_1(y) + (C + D)y - Dy^2.$$

Since especially for high values of y there is a cancellation of terms proportional to C and D , care should be taken of numerical problems. The following algorithm is sufficient. For $y < 0.3$, we calculate $E_1(y)$ using Abramowitz and Stegun 5.1.53 with only terms up to y^3 , for $0.3 < y < 1$ we derived a new fit for $ye^y E_1(y)$, for $1 < y < 10$ we use Abramowitz and Stegun 5.1.54 and for $y > 10$ we made a fit to $y^3 e^y E_1(y) + y - y^2$.

The parameters are as follows:

- * Y – Real*4, dimensionless photon energy E/kT . Input, unchanged upon output.
- * A, B, C, D, E – Real*4, the Gaunt parameters. Input, unchanged upon output.
- * GLINE – Real*4, function value, the resulting Gaunt factor. Output.

Subroutine IONIS(SION,T,FILION)

This subroutine calculates the ionisation rate according to Arnaud and Rothenflug. The program has been optimised for computational speed. Upon the first call, the subroutine reads the basic datafile FILION. Then in a loop over all possible ions, first the direct ionisation rate is calculated; then in a branching GOTO statement, the auto-ionisation corrections for different cases according to Arnaud and Rothenflug are calculated.

The parameters are:

- * SION(207) – Real*4 array, containing the total ionisation rates for all ions (zero for completely stripped ions, of course). Output. Units are cm^3s^{-1} .
- * T – Real*4, the electron temperature in keV. Input, unchanged upon output.
- * FILION – Character*50, filename of ionisation data of Arnaud and Rothenflug. Input, unchanged upon output. The basic datafile is created by the program [XSPEX.CREFIL]FILE_IONIS.

Subprograms called:

++ FU – $e^x E_1(x)$ in the approximation of Mewe and Schrijver.

++ FACTB – $1 + x - xe^x E_1(x)(2 + x)$ (see Arnaud and Rothenflug).

Function FU(U)

This function calculates $e^u E_1(u)$ using the approximation of Mewe and Schrijver (1978), eqn. (10), which is accurate to 1%. *note: more accurate and faster approximations can be made, see function GLINE.*

The parameters are:

- * U – Real*4, the argument. Input, unchanged upon output.
- * FU – Real*4, the function value. Output.

Subroutine FACTB(Y, FB)

This subroutine calculates the function $\text{FB} = 1 + y - ye^y E_1(y)(2 + y)$ for $y > 10$ according to an algorithm by Arnaud and Rothenflug (a sixteen-point quadrature formula). *note: more accurate and faster approximations can be made, see function GLINE*

The parameters are:

- * Y – Real*4, the argument. Input, unchanged upon output.
- * FB – Real*4, the function value. Output.

Subroutine RECOMB(ALFA, T, FILREC, IC)

This subroutine calculates the recombination rates after Arnaud and Rothenflug. Apart from the bare Arnaud and Rothenflug recombination rate, there is also the possibility for two variants: one with Fe XVII from Smith *et al.* (1985) and the other with the dielectronic recombination modified by Mewe. The variants are: IC=6, Fe XVII: $\alpha_{\text{diel.}} = (2.1754 \times 10^{-11} \text{ cm}^3\text{s}^{-1}) e^{-0.5601/T_{\text{keV}}^{-1.5}}$; and IC=7, Mewe

correction: $\alpha_{\text{diel.}} = \alpha_{\text{diel.,A+R}} R_{\text{Mewe}}$ with Mewe's correction given by $R_{\text{Mewe}} = 1. + 10./(S - 1)^{1.15}$ where S is the stage of the ion, e.g. Ni IV has $S = 4$. In the program, there is a basic loop over all ions; first the radiative recombination is calculated, and then in a branching GOTO the dielectronic recombination. *Note:* Mewe's variant is calculated with the same subroutine, but with another input data file!

The parameters of this subroutine are:

- * ALFA(207) – Real*4 array, containing the recombination coefficients. Of course the recombination is zero for neutral atoms. Units: cm^3s^{-1} . Output.
- * T – Real*4, the electron temperature in keV. Input, unchanged upon output.
- * FILREC – Character*50, containing the filename of the recombination parameters. For IC=1 or IC=6, in the calling program the variable "FILREC" should be substituted, for IC=7 this should be "FILREM".
- * IC – Integer*4, the type of ion concentrations requested. Input, unchanged upon output. Allowed values (the program does not check this for you): IC = 1,6,7.

Subroutine MEGRON(SION, ALFA, T)

This subroutine calculates the ionisation and recombination coefficients after Mewe and Gronenschild.

The parameters of this subroutine are:

- * T – Real*4, the electron temperature in keV. Input, unchanged upon output.
- * SION(207) – Real*4 array, containing the total ionisation rates for all ions (zero for completely stripped ions, of course). Output. Units are cm^3s^{-1} .
- * ALFA(207) – Real*4 array, containing the recombination coefficients. Of course the recombination is zero for neutral atoms. Units: cm^3s^{-1} . Output.

Subprograms called:

- ++ SET1 – to preset some temperature-independent constants (only called once).
- ++ EXPON1 – the first exponential integral $E_1(x)$.

Subroutine SET1(IONMAX, RC, NELMX, NIONMX)

This subroutine presets the temperature-independent constants for Mewe-Gronenschild ionisation balance. The parameters are:

- * IONMAX(14) – Real*4 array, containing the nuclear charge of all elements in the model, except Hydrogen. It therefore starts with IONMAX(1) = 2 for Helium, IONMAX(2) = 6 for Carbon etc. Input, unchanged upon output.
- * RC(12,28,14) – Real*4 array, containing the temperature-independent coefficients. Output. The dimensions are as follows: 12 – from 12 parameters (at most) needed for each ion; 28 – the maximum nuclear charge (currently for Ni); 14 – the number of elements included (15 except Hydrogen).
- * NELMX – Integer*4, the last dimension of RC. Currently, take NELMX=14 in the calling program. Input, unchanged upon output.
- * NIONMX – Integer*4, the maximum number of ions per element. Input, unchanged upon output. Currently, in the calling routine NIONMX should be 29 (the maximum number of ions from Ni).

Subroutine JACOBS(XZIN, T, FILCON)

This subroutine calculates the equilibrium concentrations of Jacobs *et al.* and Raymond and Smith, by reading it from a table and interpolation. The basic parameters are:

- * XZIN(207) – Real*4 array, the ionic concentrations with respect to Hydrogen. Output.
- * T – The electron temperature in keV. Input, unchanged upon output.
- * FILCON – Character*50, the filename of the file containing the grid of ion concentrations. Input, unchanged upon output. In the calling program, one can substitute FILCON either by FILJAC, the filename of ion concentrations of Jacobs *et al.*, or by FILRAS, the ion concentrations of Raymond and Smith.

Subprograms called:

++ INTER – The subroutine which does the actual interpolation.

note: This subroutine could be polished further by e.g. taking polynomial interpolation or spline interpolation from routines by Press *et al.* (1986).

Subroutine INTER(TKEV, XZ, XZIN, ITZ)

This subroutine interpolates the relative ionic abundances from a table.

The parameters are:

- * TKEV – Real*4, the electron temperature in keV. Input, unchanged upon output.
- * XZ(4347) – Real*4 array, containing the grid of ion concentrations (207 * 21 entries); units: $-^{10} \log$ of the concentration with respect to Hydrogen. Input, unchanged upon output.
- * XZIN(207) – Real*4 array, containing the ionic concentrations relative to Hydrogen. Output.
- * ITZ(2,207) – Integer*4 array, index array, input, unchanged upon output. ITZ(1,I) contains the starting position and ITZ(2,I) the end position in the array XZ of the ionic concentration grid data for ion number I.

Subprograms called:

++ SPLCO – performs cubic spline interpolation.

Subroutine SPLCO(X, Y, M, C, R, L)

This subroutine performs cubic spline interpolation. The variables are:

- * X(*) – Real*4 array, x-values. Input, unchanged upon output.
- * Y(*) – Real*4 array, y-values. Input, unchanged upon output.
- * M – Integer*4, number of points to be used for the spline determination. Input, unchanged upon output.
- * C(4,*) – Real*4 array, containing spline coefficients. Output.

* $R(8,*)$ – Real*4 array, scratch working space.

* L – Integer*4, offset in the arrays X and Y, i.e. $Y(L+1), \dots, Y(L+M)$ are used. Input, unchanged upon output.

3. Main datafiles in the directory [XSPEC]

All datafiles in this directory are direct access files. They are created by programs in the directory [XSPEC.CREFIL]. Below we give a short list of the datafiles. For details about the contents of the files, see the explanation at the program description in [XSPEC.CREFIL].

Table 6. Datafiles in [XSPEC]

Filename	Created by	Needed for
[XSPEC]IONIS.DAT	[XSPEC.CREFIL]FILE_IONIS	[XSPEC]IONIS
[XSPEC]RECOMB.DAT	[XSPEC.CREFIL]FILE_RECOMB	[XSPEC]RECOMB
[XSPEC]RECOMBM.DAT	[XSPEC.CREFIL]FILE_RECOMB	[XSPEC]RECOMB
[XSPEC]LINEFILE.DAT	[XSPEC.CREFIL]FILE_LINEFILE	[XSPEC]LINEMX
[XSPEC]CAFELINES.DAT	[XSPEC.CREFIL]FILE_LINEFILE	[XSPEC]LINEMX
[XSPEC]IONJAC.DAT	-	[XSPEC]JACOBS
[XSPEC]IONRAS.DAT	-	[XSPEC]JACOBS
[XSPEC]CONEMX.DAT	[XSPEC.CREFIL]FILE_CONEMX	[XSPEC]CONEMX
[XSPEC]KARLAT.DAT	[XSPEC.CREFIL]FILE_KARLAT	[XSPEC]CONEMX

4. Main programs in the directory [XSPEC.CREFIL]

The programs in this subroutine were derived from the original programs in the older versions of the spectral code, or newly written. In the modified programs, the FORTRAN was updated and some unnecessary long sections were shortened. Some of the programs are still not very nice, but they merely serve as working-horses for the generation of basic data files. Note that some of the subroutines pending on main programs in this subdirectory carry the same name as subroutines in the main directory [XSPEC]. Do *not* mix-up these programs! Sometimes they are nearly identical, but there are also slight differences.

In describing these programs, we will focus upon the data-file generated; much of the theory and justification for formulae is found in the relevant original papers. Please change file names in the programs where needed. These file names are generally given explicitly in OPEN statements.

Program FILE_CONEMX

This program presets some of the constants for the free-free and free-bound emission. In old versions of the spectral code, a file named "SPECTAB.DAT" was read. This file was closely related to table I of Mewe, Lemen and van den Oord (1986). In the present program, this file is input. A new output file, called CONEMX.DAT is generated, with 207 records, 1 record for each ion, and a recordlength of 6. The quantities stored in each record are (for the notation, see Mewe, Lemen and van den Oord):

- 1 - $E_{n_0} n_0^2 / E_H$
- 2 - n_0
- 3 - $E_H \zeta_0 z_0^4 / n_0^5$
- 4 - E_{n_0}
- 5 - $2z^4 E_H$
- 6 - $E_H z^2$

Program FILE_FR3NEW

This program reverses the order of indices with which the arrays AR, BR, CR, DR and POTEN were stored in the old file FR3.DAT. These arrays contain the direct ionisation variables. In the old version of IONIS, the ionisation balance routine of Arnaud and Rothenflug, the arrays were declared as AR(28,28,3) etc. These arrays fill a large part of the computer memory, and the majority of entries is simply 0 (there are only 391 non-zero elements). Also the order in which the array elements are used while running IONIS justifies more a declaration AR(3,28,28) etc. In the new version of IONIS, the necessary data are read rapidly and once into a one-dimensional array named "A", and during the run of IONIS, an incremental index IA runs straightforward through this array.

The present program is an intermediate program, it reads the old file FR3.DAT, and a new file [XSPEC.CREFIL]FR3NEW.DAT is created. This new file FR3NEW.DAT is the input for the program [XSPEC.CREFIL]FILE_IONIS, which creates the final input file [XSPEC]IONIS.DAT and which is needed for the new version of subroutine IONIS.

The file FR3NEW.DAT contains 391 records of length 8. It is essentially table 1 of Arnaud and Rothenflug (1985). Each record contains the following (A, B, C and D are defined by Arnaud and Rothenflug, section 2.1):

- 1 - shell nr. according to the nomenclature of Arnaud and Rothenflug (values 1, 2, or 3).
- 2 - ionisation stage e.g. Fe XVII gets the value 17.
- 3 - Z , the nuclear charge.
- 4 - the ionisation potential in eV.
- 5 - A
- 6 - B
- 7 - C
- 8 - D

Program FILE_IONIS

This program converts the direct- and auto-ionisation parameters of Arnaud and Rothenflug into a table suited for rapid computation. It has the following subroutines appended:

++ IONIS, POTION, NONTE, E1, FU, FACTB, EALIBE, EAMES, SAMP, REDA, EA3P3D.

The output of this program consists of a one-dimensional real*4 array A, which is written to the file [XSPEC]IONIS.DAT and which is used as input for the new subroutine IONIS. The array A is written with fixed record length of size 256. This record length has nothing to do with the internal division of the array A. The array A contains subsequently for each ion from 1-207 the following data:

- 1 The number of shells NJ. Allowed values: 0, 1, 2, 3. If NJ=0, the current ion is completely stripped, hence cannot be ionised further, and in that case parameters 3-6 are not present.
- 2 For the first shell, the ionisation potential χ , but now in keV!
- 3-6 For the first shell, Arnaud and Rothenflug's A, B, C, D (in their units) multiplied by $6.692 \times 10^{-11.5}$, in order to obtain the recombination rate with the constant pre-constant absorbed into it (see their formula below at page 426 of their paper), and for the temperature measured in keV.

Steps 2-6 may be repeated if a second and/or third shell is present. Then all following parameters shift 5 (or 10) places.

- 7 This is the index IND to distinguish between the different cases of auto-ionisation. The index may have the following values:

IND=1: no auto-ionisation for this ion. In that case, this is the last parameter for that ion.

IND=2: Cases of Ca I, Ca II, Fe III and Fe IV (in old versions, EA3P3D was called). In step 8-10 the 3 relevant temperature independent parameters follow.

IND=3: The case of Li- like ions (in old versions, EALIBE was called). In step 8-12 follow 5 parameters (similar to step 2-6) in which all temperature independent quantities have been absorbed.

IND=4: Cases where the auto-ionisation is determined by measurements (in old versions EAMES was called). In step 8-9 follow the 2 relevant temperature independent parameters.

IND=5: Auto-ionisation of Sampson (in old versions, SAMP was called). The next 18 * 6 parameters are required for this case.

IND=6: The process of Lagattuta and Hahn (in old versions REDA was called). The next 3 * 12 parameters are required for this case; these are followed by 18 * 6 parameters for Sampson's mode (IND = 5): In IONIS, the step IND = 6 is always followed by a step IND = 5 for the same ion!

Note: In the current implementation, Hydrogen is not treated!

Program FILE_KARLAT

This program generates the table containing the free-free Gaunt factor, including the relativistic correction of Kylafis and Lamb (1982). The basic, non-relativistic Gaunt-factor table of Carson (1988) is presented in a data-statement at the top of the program, and the relativistic correction is calculated using the function KYLLAM. KYLLAM calls the function RK which calculates the ratio of two modified Bessel functions: care is taken of the asymptotic behaviour at small and large values of the argument.

The program creates the file KARLAT.DAT, which contains 1 record with 25 * 25 elements. The energy and temperature grid is identical to the grid used by Carson.

Program FILE.LINEFILE

This program modifies and extends the contents of the old datafile LINEPARN.DAT and CAFE-DATA.DAT and puts the results into the new files LINEFILE.DAT and CAFELINES.DAT. The new file contains a record for each spectral line, and the length of each record is 20. Note that in this file, in order to save memory space, frequently use is made of Integer*2 variables (contrary to the general policy of using normal Integer*4's). The subsequent variables in each record of LINEFILE.DAT are (9 Integer*2's followed by 1 Character*6 and 14 Real*4's):

- I1 - IELEM, the element number (currently 1-15).
- I2 - IS, the ionisation stage (e.g., Ni VII has IELEM = 15 and IS = 7)
- I3 - ION, the transition type (e.g., ION=3 for Li-like transitions)
- I4 - NZZ, the number of the ion (from 1-207)
- I5 - IEX, excitation parameter. In LINEMX, this parameter should be 1 for He6 transitions and 3 for He5 transitions. It may have any other value for other transitions, since it is nowhere else used.
- I6 - IDRS, dielectronic recombination of satellites. If IDRS is not equal to 0, LINEMX takes this process into account.
- I7 - IDR, dielectronic and radiative recombination. IDR may have the values 0-4, with 0: no effect, 1: He S - F S transitions, 2: He4 transition, 3: He5 transition and 4: He6 transition.
- I8 - II, innershell ionisation. The parameter may be 0 (no effect) or 1,2, (or higher?).
- I9 - ITR, the transition number. Only relevant for Satellite lines. For all other lines, the value 100 is assigned. The values 1-19 correspond to He S2-S4, Li S2-S4, Be S2-S4, B S, C S, N S, O S, F S, Ca He SS, Ca Li S2-S4, Mg He S3. Storing this parameter saves a call to the old subroutine TRANS.
- C1 - The name of the transition (copied from bytes 9-14 of the original file LINEMX.DAT).
- R1 - the line energy E in keV
- R2 - the oscillator strength f
- R3 - the a -factor (excitation energy divided by photon energy)
- R4-R8 - the factors A, B, C, D, E entering the line Gaunt factor. Note that there is a slight difference with the old version: in the old version $2 * D$ was stored, which required 2 extra divisions in the calculation of the Gaunt factor; here we store D itself!
- R9-R10 - two factors needed for the radiative recombination case; we have redefined them and absorbed numerical factors into them, so that the radiative recombination rate becomes $R_9 \times T^{R_{10}}$.
- R11-R14 - 4 parameters for the density correction. Unchanged with respect to the old definitions.

Subroutine called:

++ [XSPEC.CREFIL]TRANS (included in the fortran file).

Program FILE_RECOMB

This program generates the basic data file which is required for subroutine RECOMB. There are two options (at the start of the program one is asked for the option): for $IC = 1$, the recombination coefficients are calculated according to the expressions by Arnaud and Rothenflug, for $IC = 7$ these are modified as described by Mewe. An output file is created, called RECOMB.DAT or RECOMBM.DAT, depending upon the option. This file contains 207 records, one for each ion. Each record consists of 13 Real*4 variables. The first of these variables is the recombination mode IND, the following 12 (or less) parameters are the parameters required for the specific mode. The formulae used to calculate the recombination rate (including the dielectronic recombination) can be found in Arnaud and Rothenflug, or in the Fortran source of subroutine RECOMB. The index has the following meaning:

IND= -1: There is no recombination at all (recombination rate 0). All remaining 12 parameters are 0.

IND= 0: radiative recombination of H-like ions. No dielectronic recombination.

IND= 1-7: radiative recombination plus dielectronic recombination.

IND=1: He-like ions

IND=2: normal dielectronic recombination plus Shull's correction for high temperature for Ne VII, Ne VIII, S VI, S XIII, S XIV and Ca XVIII

IND=3: Na and Al

IND=4: normal dielectronic recombination and Shull's correction for low temperature Si II

IND=5: normal dielectronic recombination and Shull's correction for low temperature Si III

IND=6: normal dielectronic recombination and for low temperatures Nussbaumer and Storey, for C II-IV, N II-V and O II-VI.

IND=7: normal dielectronic recombination (all other ions).

Subprogram called:

++ [XSPEC.CREFIL]RECOMB (included in the fortran file).

5. Main data files in the directory [XSPEC.CREFIL]

Below we give a short list of the datafiles in this subdirectory. For details about the contents of the files, see the explanation at the program description in [XSPEC.CREFIL].

Table 7. Datafiles in [XSPEC.CREFIL]

File name [XSPEC.CREFIL]	Created by	Needed for
FR3.DAT	-	[XSPEC.CREFIL]FILE_FR3NEW
FR3NEW.DAT	[XSPEC.CREFIL]FILE_FR3NEW	[XSPEC.CREFIL]FILE_IONIS
LINEPARN.DAT	-	[XSPEC.CREFIL]FILE_LINEFILE
CAFEDATA.DAT	-	[XSPEC.CREFIL]FILE_LINEFILE
SPECTAB.DAT	-	[XSPEC.CREFIL]FILE_CONEMX

6. Hydrodynamical models for supernova remnants and non-equilibrium subroutines in the subdirectory [XSPEC.SNR]

Introduction.

The current subroutines for non-equilibrium spectra were written by J.S. Kaastra and F.A. Jansen. A preliminary description of the basic method in solving the non-equilibrium spectra is given in Jansen's thesis (1988). A fuller description will appear elsewhere. The programs can be found in the directory [XSPEC.SNR]. The main programs CREFIL and CRERAT generate the basic data files needed for the generation of non-equilibrium concentrations. The subroutine NEISPEC calculates the X-ray spectrum for a given temperature- and Hydrogen-density history of an arbitrary, moving mass element in an arbitrary situation (where the plasma is optically thin). So it may be applied to supernova remnants, solar and stellar flares, etc. The subroutines TIMESTEP, SOLCON, RATESTEP and READ_ARRAY are subprograms of NEISPEC. The MAIN program SNR is an example of a driving routine for the important SUBROUTINE SNRFIT, which calculates the spectrum of several theoretical models for (at present) self-similar supernova models. The subroutines PARNOR, PROSPEC, SELFSIM, SNRMOD, SNRSHELL, ENERG, TIMEHIST, SPINT, SEDOV, LSEDOV, ISOINT, ISOTHERM, CHEVAL, OUTER, INNER, and BAND, BANDSHAPE, BANDFAST are subprograms of SNRFIT.

PROGRAMS CRERAT and CREFIL

This program generates the transition rate matrices for all the elements and transforms them to eigenvalue basis as described by Jansen (1988). It is possible to choose between Arnaud-Rothenflug and Mewe-Gronenschild ionisation balance. The currently available datafile (12-03-1992) was done for Arnaud - Rothenflug balance. The program first asks for the type of transition rate (1 is Arnaud & Rothenflug, 2 is Mewe and Gronenschild). Then the start temperature $^{10} \log(T_s)$, end temperature $^{10} \log(T_e)$ and the stepsize (in logarithmic units) for the temperature grid should be specified. A good choice appears to be 3,10,0.05. This was done in the run which created the appending output files. To run the program with those parameters took 2 hour cpu-time on a VAX-785, so be carefull and do not destroy your data files!

In a run over the temperature grid, the program calculates the ionisation and recombination coefficients using subroutine FLINE; for each element, the transition matrix is brought to eigen-value basis, and the eigenvectors and eigenvalues are evaluated in double precision using NAG-subroutines. Small eigenvalues are put to zero, and the eigenvectors are normalised to unity length. Finally, the eigenvector matrix for each element and temperature is inverted. The eigenvalues, eigenvector and inverse eigenvector matrix are stored for each element on separate files, which are opened and closed each time the results for a new temperature become available; this has the desirable property that whenever the program is stopped because of cpu-time limits or other reasons, the calculations up to a certain temperature are saved, and the calculation needs to be extended only to the remaining temperature values.

The output of the program consists of a set of files: the eigenvalues, eigenvector matrix and inverse eigenvector matrix for element no. 4, Nitrogen, are on files called 04.VAL, 04.VEC and 04.INV respectively, and similar names are used for the other 14 elements. *Note:* each time the program runs these files are overwritten! There is also an output file labeled INDEX.DAT, which contains the values of start, end and step in $\log(T)$ which were input. The last output file is XZIN.DAT, which contains the equilibrium concentrations; note however that in this last file, we normalised the ionic concentrations to 1 for the element, i.e. the results are independent of the elemental abundances (although subroutine ABUN is called during the course of the program).

Subprograms called in CRERAT:

++ LIB\$GET_LUN - gets a free logical unit number. This is a VAX-specific system routine; if you do not have this routine available, replace the statement "CALL LIB\$GET_LUN(LUN)" by "LUN=LU" where LU is any concrete logical unit number which is free.

- ++ LIB\$_INIT_TIMER – VAX-specific routine for impatient programmers. Not necessary (may be omitted).
- ++ LIB\$_SHOW_TIMER – VAX-specific routine for impatient programmers. Not necessary (may be omitted).
- ++ ABUN – gets the standard abundances with respect to Hydrogen.
- ++ FLINE – the basic spectral routine, this time used to obtain the transition rates.
- ++ F02AGF – a subroutine from the NAG-library to determine eigenvalues and vectors.
- ++ F04AEF – a subroutine from the NAG-library to invert a matrix.

The second program is CREFIL; this program reads the output from CRERAT and converts the many datafiles into three basic datafiles, called VAL.DAT, VEC.DAT and INV.DAT, containing the eigenvalues, eigenvector matrix and inverse eigenvector matrix as a function of temperature. In this program also the conversion from double to single precision is made.

Subprograms called in CREFIL:

- ++ LIB\$GET_LUN – gets a free logical unit number. This is a VAX-specific system routine; if you do not have this routine available, replace the statement "CALL LIB\$GET_LUN(LUN)" by "LUN=LU" where LU is any concrete logical unit number which is free.

In general, we advise you to run first CRERAT, immediately followed by CREFIL. In order to save disk space, at the end you may delete the files INDEX.DAT and XZIN.DAT, and also all ##.VAL, ##.VEC and ##.INV files where ## is any number between 01–15. Only VAL.DAT, VEC.DAT and INV.DAT are really needed.

Subroutine NEISPEC(T, H, TIME, M, E1, E2, FLX, EDT, ED, NBIN, FACTOR, IC, ID, TISM)

This subroutine is the basic non-equilibrium routine. It calculates the X-ray spectrum for an evolving mass element. On input, arrays giving the temperature and hydrogen density values of the mass element as a function of time must be specified, as well as the starting (equilibrium) temperature. Then the evolution of the ion concentrations is calculated; finally the X-ray spectrum of the end state is calculated and returned to the main program.

The subroutine basically consists of calls to two subroutines which are described in detail later.

The basic parameters are:

- * T(M) – Real*4 array, containing the temperatures in K of the mass element for each time step. Input, unchanged upon output.
- * H(M) – Real*4 array, containing the Hydrogen densities in $10^6 \text{ m}^{-3} = \text{cm}^{-3}$ for each time step. Input, unchanged upon output.
- * TIME(M) – Real*4 array, containing the time in s of the different time steps. Input, unchanged upon output. This array needs not to be evenly spaced. One should only take care that the grid is fine enough to allow a reasonable interpolation (e.g., do not let your temperature steps be much smaller or larger than the typical stepsize in the grid of the basic transition rate files, e.g. 0.05 in logarithm is a reasonable stepsize for the temperature).
- * M – the number of timesteps used. Note that the arrays T, H, and TIME should be ordered in a sense that the first element contains the information at the start and the last element the final state considered.
- * E1(NBIN) – Real*4 array of length NBIN, containing the lower bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * E2(NBIN) – Real*4 array of length NBIN, containing the upper bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * FLX(NBIN) – Real*4 array of length NBIN, containing the output emissivity in photons $\text{s}^{-1} \text{keV}^{-1} \text{m}^{-3}$. Output.
- * EDT – Real*4, output. Contains the integral $n_e dt$ in units of s/cm^3 .
- * ED – Real*4, output, the electron density divided by the Hydrogen density.
- * NBIN – Integer*4, number of energy bins requested from the calling program. Input. Unchanged upon output.
- * FACTOR(15) – Real*4 array, containing the relative abundances of the elements with respect to Solar abundances (for solar abundances, $FACTOR(I) = 1.0$ for $I = 1, \dots, 15$).
- * IC – Integer*4, type of ion concentrations which should be used for the starting concentrations. Input, unchanged upon output. Allowed values are now:
 - IC=1: Arnaud and Rothenflug (1985).
 - IC=2: Gronenschild and Mewe (1985).
 - IC=3: High density plasma, mainly from Jacobs *et al.*

IC=4: not allowed.

IC=5: Raymond and Smith.

IC=6: Arnaud and Rothenflug (1985) with Fe XVII from Smith *et al.* (1985).

IC=7: Arnaud and Rothenflug (1985), with dielectronic recombination modified by Mewe.

* ID – Integer*4, type of line profile. Input, unchanged upon output. Allowed values:

ID=0: no line broadening (i.e. all spectral lines are assumed to be infinitely narrow).

ID=1: Thermal Doppler broadening of the lines, assuming $T_e = T_i$.

* TISM – Real*4, Electron temperature at start in K. Input, unchanged upon output.

Subprograms called:

++ TIMESTEP – converts the input temperature grid into a grid suited for fast integration of the ion concentrations.

++ SOLCON – integrates the differential equation for the ion concentrations.

Subroutine TIMESTEP(T, H, TIME, M, IT, TN, N, TISM, TSTART, TEND, IMAX)

This subroutine transforms the arrays T, H, TIME describing the functional dependence among the temperature, density and time in the evolution of a single mass element into a grid suited for fast integration of the differential equations for the ion concentrations. The basis of the new grid is formed by a "standard" temperature grid on which the transition matrices were calculated (output of the program CRERAT). That grid has a constant logarithmic spacing of e.g. 0.05. The temperatures are rounded to the nearest temperature on the "standard" grid. The data are transformed as follows. On the new grid, each time a new bin is taken in either one of the following situations: the temperature becomes too different from the old situation (e.g., more than 0.05 difference in the logarithm), or the integral $\int n_H dt$ over the old bin becomes significantly larger than the total integral up to the time considered (e.g., 0.05 in logarithm, with a minimum of $10^{12} \text{ m}^{-3}\text{s}^{-1} = 10^6 \text{ cm}^{-3}\text{s}^{-1}$).

The parameters are:

* T(M) – Real*4 array, containing the temperatures in K of the mass element for each time step. Input, unchanged upon output.

* H(M) – Real*4 array, containing the Hydrogen densities in $10^6 \text{ m}^{-3} = \text{cm}^{-3}$ for each time step. Input, unchanged upon output.

* TIME(M) – Real*4 array, containing the time in s of the different time steps. Input, unchanged upon output. This array needs not to be evenly spaced. One should only take care that the grid is fine enough to allow a reasonable interpolation (e.g., do not let your temperature steps be much smaller or larger than the typical stepsize in the grid of the basic transition rate files, e.g. 0.05 in logarithm is a reasonable stepsize for the temperature).

* M – Integer*4, the number of timesteps used. Note that the arrays T, H, and TIME should be ordered in a sense that the first element contains the information at the start and the last element the final state considered.

* IT(N) – Integer*4 array, temperature bin numbers (in units of the basic eigenvalue temperature grid which was used for the creation of the files VAL.DAT etc.) E.g., a temperature equal to

the lowest temperature of the grid has $IT=1$, equal to the second bin in that grid has $IT=2$ etc. Output.

- * $TN(N)$ – Real*4 array, containing the integral $\int n_H dt$ over the present time interval of this array element (Note: the integral over all previous bins is *not* added to $TN(I)$). Output.
- * N – Integer*4, number of final timesteps used. Output. Note that $IT(1)$ contains the temperature bin number at the start of the time integration and $IT(N)$ the bin number at the end of the time integration.
- * $TISM$ – Real*4, Electron temperature at start in K. Input, unchanged upon output.
- * $TSTART$ – Real*4, Electron temperature at start in keV. Output.
- * $TEND$ – Real*4, Electron temperature at end in keV. Output.
- * $IMAX$ – Integer*4, maximum value allowed for $IT(I)$, i.e. maximum size of the eigenvalue temperature grid. Output.

Warning: It is important that the user checks if the (parameter) declarations at the top of the program of the internal variables $TLSTART$, $TLSTEP$ and $IMAX$ are consistent with the eigenvalue temperature grid! The current values for these parameters are sufficient for a temperature grid from $\log(T)$ ranging from 3–10 with steps of 0.05, where T is in K.

Subroutine SOLCON(IT, TN, N, IMAX, E1, E2, FLX, NBIN, FACTOR, IC, ID, TSTART, TEND, H, EDT, ED)

This subroutine solves for the non-equilibrium ion concentrations and also calculates the final X-ray spectrum. The parameters are:

- * $IT(N)$ – Integer*4 array, temperature bin numbers (in units of the basic eigenvalue temperature grid. Input, unchanged upon output.
- * $TN(N)$ – Real*4 array, containing the integral $\int n_H dt$ over the present time interval of this array element. Input, unchanged upon output.
- * N – Integer*4, number of final timesteps used. Input.
- * $IMAX$ – Integer*4, maximum value allowed for $IT(I)$, i.e. maximum size of the eigenvalue temperature grid. Input, unchanged upon output.
- * $E1(NBIN)$ – Real*4 array of length $NBIN$, containing the lower bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * $E2(NBIN)$ – Real*4 array of length $NBIN$, containing the upper bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * $FLX(NBIN)$ – Real*4 array of length $NBIN$, containing the output emissivity in photons $s^{-1} keV^{-1} m^{-3}$. Output.
- * $NBIN$ – Integer*4, number of energy bins requested from the calling program. Input. Unchanged upon output.
- * $FACTOR(15)$ – Real*4 array, containing the relative abundances of the elements with respect to Solar abundances (for solar abundances, $FACTOR(I) = 1.0$ for $I = 1, \dots, 15$).

* IC – Integer*4, type of ion concentrations which should be used for the starting concentrations. Input, unchanged upon output. Allowed values are now:

IC=1: Arnaud and Rothenflug (1985).

IC=2: Gronenschild and Mewe (1985).

IC=3: High density plasma, mainly from Jacobs *et al.*.

IC=4: not allowed.

IC=5: Raymond and Smith.

IC=6: Arnaud and Rothenflug (1985) with Fe XVII from Smith *et al.* (1985).

IC=7: Arnaud and Rothenflug (1985), with dielectronic recombination modified by Mewe.

* ID – Integer*4, type of line profile. Input, unchanged upon output. Allowed values:

ID=0: no line broadening (i.e. all spectral lines are assumed to be infinitely narrow).

ID=1: Thermal Doppler broadening of the lines, assuming $T_e = T_i$.

* TSTART – Real*4, start electron temperature in keV. Input, unchanged upon output.

* TEND – Real*4, final electron temperature in keV. Input, unchanged upon output.

* H – Real*4, Hydrogen density at the end. Input, unchanged upon output.

* EDT – Real*4, output. Contains the integral $n_e dt$ in units of s/cm^3 .

* ED – Real*4, output, the electron density divided by the Hydrogen density.

Subprograms called:

++ FLIN – for the start concentrations and the final spectrum

++ RATESTEP – advances the solution for the ion concentrations.

Subroutine RATESTEP(IPRE, IMAX, TINTNH, YZ, ED, MOD)

This subroutine advances the solution for the ion concentrations in time with one step. The old ion concentrations are used to calculate the electron density. Then for all elements the ion concentrations are brought to the eigenvector basis of the transition matrix. On this new basis, time integration of the differential equations is simply multiplication with an exponential factor (Temperature is roughly constant because in the calling routine the temperature grid is fine enough, and for larger temperature changes, RATESTEP is called with the parameters for the new temperature; moreover, the electron density is also roughly constant, because in the calling routine a new call to RATESTEP is made if the integral $\int n_H dt$ changes too much). After this simple time-integration, the concentrations are brought back to the original basis.

A crucial point in the subroutine is the following. In order to save computing time, the eigenvector matrices etc. are read once and are put into a memory buffer. The size of that buffer is chosen in such a way that it fits within the working space of the computer. This working space is in general smaller than the total memory requirements for the matrices. The program works in such a way that if the temperature exceeds the range corresponding to the buffer in the working space, a new buffer is read;

the subroutine then prompts a message "Reading new buffer". The buffer size should be taken as large as possible, because I/O takes a lot of time; of course, the buffer size is machine dependent. The critical parameter in the subroutine is NSTORE. For a VAX-785 computer, NSTORE = 80 is sufficient.

The parameters of the subroutine are:

- * IPRE – Integer*4, the present temperature bin number (corresponding to the current temperature). Input, unchanged upon output.
- * IMAX – Integer*4, the maximum value for IPRE possible (i.e., the bin number of the highest temperature for which the matrices are available on disk). Input, unchanged upon output. For temperatures exceeding this maximum temperature, the program does as if the temperature equals this maximum temperature, and similarly for temperatures below the smallest available temperature.
- * TINTNH – Real*4, the integral $\int n_H dt$ over the current time step.
- * YZ(207) – Real*4, the ion concentrations with respect to Hydrogen. On input, this contains the starting value, on output, the advanced solution for the concentrations.
- * MOD – Integer*4, input. Designates the mode.

MOD=0: normal mode

MOD=1: special mode (faster) for $T=\text{constant}$ plasmas.

- * ED – Real*4, output, the electron density divided by the Hydrogen density.

Subprograms called:

++ READ_ARRAY – reads a part of the new memory buffer

Logical units used: 81, 82, 83; these units remain open as long as the program is executed.

Subroutine READ_ARRAY(A, N, LUN, IREC)

This subroutine reads a part of the memory buffer containing eigenvalues, eigenvector matrices etc.

The parameters are:

- * A(N) – Real*4 array, Output. Contains the data read.
- * N – Real*4, the number of array elements to be read. Input, unchanged upon output.
- * LUN – Integer*4, logical unit number from which the data have to be read. Input, unchanged upon output.
- * IREC – Integer*4, record number of the data to be read. Input, unchanged upon output.

Program SNR

This is an example of a simple driving routine for subroutine SNRFIT which calculates the supernova remnant spectra. Program SNR may simply be adjusted to the needs of the user. It reads the necessary parameters, generates the energy grid, calls subroutine SNRFIT, and writes output files which can be processed by plotting programs or printed.

Subroutines called: SNRFIT

Subroutine SNRFIT(E, NE, PARAM, PARSNR, FLX)

This subroutine is just an intermediate between the driving program SNR and the working programs. It is written in a form ready to be called by the X-ray fitting package XSPEC.

The parameters of the subroutine are:

- * E(0:NE) – Real*4 array, the boundaries of the energy bins in keV, i.e. E(0) is the lower boundary of the first bin, and E(NE) is the upper boundary of the last bin. Input.
- * NE – Integer*4, the number of energy bins used. Input.
- * PARAM(*) – Real*4 array, containing the input parameters of the model as used by XSPEC.

PARAM(1): the model used; valid values:

0: simple $n_e t$ model.

1: Sedov model.

2: Isothermal model.

3: Chevalier model.

4: Band's isothermal model.

5: Band's isentropic model.

PARAM(2): the number of shells to be used (at most 100).

PARAM(3): projection onto the sky (0=no, 1=yes) (yes may be time-consuming if high energy resolution is used).

PARAM(4): radial parameter profiles (0=no, 1=yes)

PARAM(5): XSPEC parameter (usually 0, but for XSPEC 1); the reason is the different units of the output flux used by me and XSPEC (see parameter FLX below).

PARAM(6): minimum projected radius for integration of the X-ray image (take default 0 for complete remnant).

PARAM(7): maximum projected radius for integration of the X-ray image (take default 1 for complete remnant). Important for spatially-resolved spectroscopy only.

PARAM(8): minimum true radius for integration of the X-ray image (take default 0 for complete remnant).

PARAM(9): maximum true radius for integration of the X-ray image (take default 1 for complete remnant). Important if only a part of the remnant is studied (e.g. only the shocked ejecta).

PARAM(10): minimum energy (keV) for the radial profiles to be calculated (only required if PARAM(4)=1). Input.

PARAM(11): maximum energy (keV) for the radial profiles to be calculated (only required if PARAM(4)=1). Input.

PARAM(12): angular radius of the remnant in arcmin. Input.

PARAM(13): input estimate of the temperature (keV). Note that the actual temperature will be slightly different (this is an output parameter of the model).

PARAM(14): input estimate of the ionisation parameter U (the integral $n_e dt$). Units 10^{10} s/cm³, input.

PARAM(15): the interstellar density gradient s (sometimes called ω). Input.

PARAM(16): the density gradient parameter of the stellar ejecta, called n by Chevalier. Input.

PARAM(17): the α -parameter of Hamilton. This parameter is currently not used. Hamilton's model is currently being programmed and will later be added as a new model.

PARAM(18–31): the relative abundances of all 14 elements from He–Ni used in the code, for the interstellar medium. Note that the Hydrogen abundance, by definition, is always 1.

PARAM(32–45): as 18–31, but for the stellar ejecta.

* PARSNR(28) – Real*4 array, called WORKSP by XSPEC, containing interesting supernova parameters inputted or calculated by the model. For a description of these parameters, see the description of PAR at the subroutine SNRMOD. Output.

* FLX(NE) – Real*4 array, containing the fluxes on output in ph/m²/s/keV, or for XSPEC (if PARAM(5)≠0.5) in other units: ph/cm²/s/bin.

Subroutines called:

++ PARNOR – estimates the input parameters for SNRMOD

++ SNRMOD – the main SNR program

++ PROSPEC – projects the spectrum onto the sky and adds Doppler shifts.

Note: In my explicit XSPEC installation of the SNR programs I have developed somewhat sim-

plified models, allowing less free parameters, since in many practical situation up to now (e.g. detectors with low spectral resolution, or no spatial resolution) the situation is slightly simplified; projection and Dopplershifts can be neglected often. However, the current subroutine SNRFIT is rather complete, especially in order to be able to make detailed theoretical calculations.

Subroutine SNRMOD(E1, E2, FLS, FLX, NBIN, R, R1, R2, TW, HW, EDW, EDTW, TIMEW, ZW, NR, NRMAX, PAR, NPAR, ABUNISM, ABUNEJECT, IC, ID, IM, IW)

This subroutine is the main routine for calculating non-equilibrium, self-similar supernova remnant spectra. The following hydrodynamical models are available: the adiabatic (Sedov) and isothermal model, both with variable interstellar density gradient, and models with a reverse shock, with variable interstellar density gradient as well as variable stellar ejecta gradient; the reverse shock models are Chevalier's adiabatic models, and Band's isothermal and isentropic models. For all models, the elemental abundances can be chosen freely, and for the reverse shock models, the abundances of the stellar ejecta may be chosen differently from the abundances of the interstellar medium. Spectra as a function of radius are generated, so also radial profiles may be obtained. Finally, it is also possible to simulate thermal Doppler line broadening, and Doppler shifts due to the expansion of the supernova remnant.

The program first calculates the basic hydrodynamical model. As a function of radius within the remnant (all expressed in main shock radii) the Lagrangian radius, temperature, Hydrogen density, and expansion velocity are calculated. The Lagrangian radius is the radius a mass element had at the time of shock- or reverse shock passage. Also some mass and energy integrals are calculated. Then the position grid on which the spectra are calculated, is created. In a loop over all these shells, for each position the time history of the mass element is determined using the scaling properties of the supernova remnant model and the Lagrangian radius; these time history arrays (time, temperature and Hydrogen density) are input for the subroutine NEISPEC which calculates the non-equilibrium concentrations and the emissivity. After multiplying with the emitting volume of each shell, the resulting shell spectra are stored and the total spectrum is augmented.

For the set of input parameters, several different combinations are possible. Here we have chosen for the remnant distance, radius, explosion energy and interstellar Hydrogen density. Values of other related parameters, like remnant age, shock temperature, shocked mass etc. can be derived and are given in some of the output parameters.

The parameters of the subroutine are:

- * E1(NBIN) – Real*4 array of length NBIN, containing the lower bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * E2(NBIN) – Real*4 array of length NBIN, containing the upper bounds of the energy bins. Units keV. This array is input and unchanged upon output.
- * FLS(NBIN,NRMAX) – Real*4 array of length (NBIN,NRMAX). Array, containing the shell spectra in the same units as FLX below.
- * FLX(NBIN) – Real*4 array of length NBIN. Output. Contains the total supernova remnant spectrum as detected at earth (apart from interstellar absorption) in units of photons $m^{-2}s^{-1}keV^{-1}$ (*note the SI*). This total spectrum is corrected for thermal line broadening if requested, but not for Doppler shifts due to the expansion of the remnant. That last effect must be treated in a separate subroutine (PROSPEC) which reads the individual shell spectra and expansion velocities.
- * NBIN – Integer*4, number of energy bins requested from the calling program. Input. Unchanged upon output. There is no upper limit to this parameter (apart from the memory size of your computer).
- * R(NR) – Real*4 array of length NR, containing the positions of the shell centres, in units of

the main shock radius (so $R \leq 1$). Output. For models without a reverse shock, $R(\text{IR})$ is taken proportional to $\text{IR}^{0.25}$ and for models with a reverse shock, $R(\text{IR})$ is a linear function between the reverse shock and the main shock.

- * $\text{R1}(\text{NR})$ – Real*4 array, lower position boundaries of the shells, corresponding to $\text{R}(\text{NR})$ above. Output.
- * $\text{R2}(\text{NR})$ – Real*4 array, as R1 above but upper position boundaries. Output.
- * $\text{TW}(\text{NR})$ – Real*4 array, contains the temperatures of the shells in keV. Output.
- * $\text{HW}(\text{NR})$ – Real*4 array, containing the Hydrogen densities of the shells (in cm^{-3}). Output.
- * $\text{EDW}(\text{NR})$ – Real*4 array, containing for each shell the ratio of electron density to Hydrogen density. Output.
- * $\text{EDTW}(\text{NR})$ – Real*4 array, containing for each shell the U value ($n_e t$) in s/cm^{-3} . Output.
- * $\text{TIMEW}(\text{NR})$ – Real*4 array, contains for each shell the plasma age in s, i.e. the time since the main or reverse shock passed the mass element. Output.
- * $\text{ZW}(\text{NR})$ – Real*4 array, containing for each shell the redshift (v/c). Output.
- * NR – Integer*4, the number of position bins. Input, unchanged upon output. For reverse shock models, half of the number of position bins is taken in the shocked ejecta and the other half in the shocked interstellar matter.
- * NRMAX – Integer*4, the maximum number of shells possible. Currently 100. Input. Note that $\text{NR} \leq \text{NRMAX}$!
- * $\text{PAR}(\text{NPAR})$ – Real*4 array of length NPAR , containing partially input, partially output parameters of the supernova model. These parameters are as follows:

$\text{PAR}(1)$ – Input, the SNR distance in kpc. Required for the normalisation of the spectrum.

$\text{PAR}(2)$ – Input, the present main shock radius of the SNR in pc (not to be confused with eventually the reverse shock radius, or the radius of the contact discontinuity).

$\text{PAR}(3)$ – Input, the explosion energy contained within the shocked region, in 10^{43} J. Note that for reverse shock models the actual shock energy is higher than this input value, since there is also an (X-ray invisible) part of the stellar ejecta behind the reverse shock which carry kinetic energy (but no thermal energy due to the low temperature).

$\text{PAR}(4)$ – Input, the interstellar *Hydrogen* density just *before* the main shock, in units of cm^{-3} .

$\text{PAR}(5)$ – Input, ω (called s by some authors) the interstellar density gradient defined by $\rho \sim r^{-\omega}$. Allowed values for ω are $-\infty < \omega < 5$ for $\text{IM}=1$, and $0 \leq \omega < 3$ for $\text{IM}=2-5$.

$\text{PAR}(6)$ – The density gradient n of the stellar ejecta, defined by $\rho \sim r^{-n}$. For $\text{IM}=3-5$, this value should be given as input, and should be in the range $n > 5$. For $\text{IM}=1-2$, $\text{PAR}(6)$ is meaningless; on output we assign it the uniform value 5.

* $\text{PAR}(7)$ – Γ , the adiabatic index. For $\text{IM}=1$ and $\text{IM}=3$ this is a free and required input parameter, which should obey $\Gamma > 1$. Usually, we advise you to take $\Gamma = 5/3$. For all other models, IM is an output parameter, with value 1 for $\text{IM}=4$ and $5/3$ for $\text{IM}=2$ or $\text{IM}=5$.

$\text{PAR}(8)$ – $\eta_r/\eta_{r,\text{min}}$, i.e. the density jump at the reverse shock in units of the minimum density

jump at the reverse shock. This minimum density jump is given by $\eta_{r,min} = 1 + 1/\Gamma$, where Γ is the adiabatic index parameter (PAR(7)). PAR(8) must be an input parameter for IM=4 and IM=5 (Band's models). Allowed values are PAR(8) ≥ 1 . For PAR(8)=1 one obtains the maximum-energy solution. For IM=1-3, PAR(8)=0. on output, and is not requested for input.

PAR(9) – Output, present age of the remnant in s.

PAR(10) – Output, post-shock temperature (K).

PAR(11) – Output, α , the dimensionless energy scaling factor, in the nomenclature of Sedov, i.e. the explosion energy E_0 is related to the shock radius r_s and the present-day pre-shock interstellar mass density ρ_0 by $r_s(t) = (E_0 t^2 / \alpha \rho_0)^{1/(5-\omega)}$.

PAR(12) – Output, swept-up interstellar matter in solar masses (i.e. all interstellar mass within the contact discontinuity and the main shock, for reverse shock models, or within the main shock for models without a reverse shock).

PAR(13) – Output, mass of shocked stellar ejecta in solar masses (zero for models without a reverse shock). It contains all mass between the reverse shock and the contact discontinuity. Note that the total amount of ejected matter may be larger than this number, because the part of the ejecta within the reverse shock is not yet shocked.

PAR(14) – Output, the relative density jump at the forward shock.

PAR(15) – Output, reverse shock radius in units of the main shock radius. Zero if no reverse shock is present.

PAR(16) – Output, radius of the contact discontinuity in units of the main shock radius. Zero if no reverse shock is present.

PAR(17) – Output, the fraction of the total energy within the shocked material divided by the total energy that went into the shocked material. Unity for IM=1-3, but smaller than 1 for Band's models (IM=4 or IM=5).

PAR(18) – Output, emission-measure averaged temperature; unit: keV.

PAR(19) – Output, idem, ejecta only (if present)

PAR(20) – Output, idem, shocked ism only

PAR(21) – Output, emission-measure averaged integral $n_e dt$; unit: 10^{10} s/cm³.

PAR(22) – Output, idem, ejecta only (if present)

PAR(23) – Output, idem, shocked ism only

PAR(24) – Output, reduced emission measure ejecta

PAR(25) – Output, idem, shocked ism

PAR(26) – Output, emission-measure averaged expansion velocity (km/s, ejecta)

PAR(27) – Output, idem, shocked ism

PAR(28) – Output, alpha (Hamilton); currently not used.

* NPAR – Integer*4, number of parameters used for array PAR. Input. Allowed value: 28.

* ABUNISM(15) – Real*4 array of length 15, containing the elemental abundances in the interstellar matter with respect to Hydrogen. Input, unchanged upon output.

* ABUNEJECT(15) – Real*4 array of length 15, containing the elemental abundances in the stellar ejecta with respect to Hydrogen. Input, unchanged upon output. This array needs only to be specified for IM=3–5.

* IC – Integer*4, type of ion concentrations which should be used for the cool, preshock medium. Input, unchanged upon output. Allowed values:

IC=1: Arnaud and Rothenflug (1985).

IC=2: Gronenschild and Mewe (1985).

IC=3: High density plasma, mainly from Jacobs *et al.*

IC=5: Raymond and Smith.

IC=6: Arnaud and Rothenflug (1985) with Fe XVII from Smith *et al.* (1985).

IC=7: Arnaud and Rothenflug (1985), with dielectronic recombination modified by Mewe.

* ID – Integer*4, type of line profile. Input, unchanged upon output. Allowed values:

ID=0: no line broadening (i.e. all spectral lines are assumed to be infinitely narrow).

ID=1: Thermal Doppler broadening of the lines, assuming $T_e = T_i$. Account is taken of the different mass of each element.

* IM – Integer*4, supernova remnant model number. Input, unchanged upon output. Allowed values:

IM=1: Sedov's adiabatic model. Free parameters: remnant distance, radius, explosion energy, interstellar density, ω , Γ .

IM=2: The isothermal model. Free parameters: remnant distance, radius, explosion energy, interstellar density, ω .

IM=3: Chevalier's adiabatic reverse shock model. Free parameters: remnant distance, radius, explosion energy, interstellar density, ω , n , Γ .

IM=4: Band's isothermal reverse shock model. Free parameters: remnant distance, radius, explosion energy, interstellar density, ω , n , reverse shock density jump.

IM=5: Band's isentropic reverse shock model. Free parameters: remnant distance, radius, explosion energy, interstellar density, ω , n , reverse shock density jump.

* IW – Integer*4, IW=1: write for each shell calculated a message on the screen (usefull for slow computers or many energy bins to see the progress of the program); IW=0: no such writing (use this for spectral fitting or many calls to this subroutine).

Subprograms called:

++ SELFSIM – calculates a self-similar hydrodynamical model.

++ SNRSHELL – calculates the time history of a single SNR-shell.

++ NEISPEC – calculates the non-equilibrium spectra and ionisation balance.

Note: The preshock temperature and the temperature of the unshocked stellar ejecta are specified in a parameter statement (variables TISM and TEJ) which currently both have the value of 2×10^4 K.

Note: There are implicitly some conversion factors in the program which convert mass density and pressure into temperature. The Hydrodynamical models all use the pressure p and mass density ρ as basic variables (the shock jump conditions are usually defined in terms of those quantities). Now the pressure and temperature are related by $p = n_e k T_e + n_i k T_i$. Further, the ratio's n_e/ρ and n_i/ρ depend upon the elemental abundances and the ionisation balance. In a real supernova remnant, also ionisation energy plays a role, which makes the definition of a single adiabatic index Γ useless. It is unclear how to incorporate these effects into a real self-similar supernova remnant model (in fact, the introduction of extra parameters causes self-similar solutions ceasing to exist). But since the shocked material in general has a high degree of ionisation, we will neglect those effects and simply take the following conversions and conventions:

$$\begin{aligned} T_e &= T_i, \\ n_e &= 1.1376 n_H, \\ n_i &= 1.0645 n_H, \\ \rho &= (2.1305 \times 10^{-27} \text{kg}) n_H, \end{aligned}$$

which are valid for a completely ionised medium of cosmic abundances as given by Ross and Aller (1976) and adopted previously as our standard set of abundances. This still has to be corrected to the abundance set of Anders and Grevesse (1989), our new abundance set.

Subroutine SELFSIM(R, R1, R2, NR, PAR, NPAR, IM, RG, R0G, HG, TG, VG, NG, M, VC)

This subroutine initialises the hydrodynamical parameters for self-similar supernova remnants. The parameters are:

- * R, R1, R2, NR, PAR, NPAR, IM – see the description for the subroutine SNRMOD.
- * RG(NG) – Real*4 array, output, containing the position grid of the hydrodynamical solution, normalised to 1 for the main shock.
- * R0G(NG) – Real*4 array, output, containing the Lagrangian radius (the radius of the remnant at the time the shock or reverse shock passed the mass element), normalised to 1 for the present main shock.
- * HG(NG) – Real*4 array, output, containing the mass density at the position grid, normalised to 1 at the main shock.
- * TG(NG) – Real*4 array, output, containing the temperature at the position grid, again normalised to 1 at the main shock.
- * VG(NG) – Real*4 array, output, containing the expansion velocity of the remnant at the position grid, again normalised to 1 at the main shock.
- * NG – Integer*4, output, the maximum array size of the position grid outputted by the hydrodynamical model.
- * M – Integer*4, output, the actual array size used by the hydrodynamical model.
- * VC – Real*4, output, the scaling factor of the redshift of the shells.

Subprograms called:

- ++ SEDOV – the Sedov model
- ++ ISOTHERM – the isothermal model
- ++ CHEVAL – Chevalier’s model
- ++ BAND – Band’s model
- ++ ENERG – calculate energy and mass integrals

Subroutine ENERG(RC, PV, R, H, T, V, N, A, S, E)

This subroutine calculates the mass and energy integrals for supernova remnant models 1–3. The parameters are:

- * RC – Real*4, position of the contact discontinuity in units of the main shock radius. Input, unchanged upon output. For IM=1 and IM=2, this should be 0.
- * PV – Real*4, a model dependent scaling parameter, proportional to the ratio of thermal to kinetic energy.
- * R(N) – Real*4 array, containing the radii normalised to the shock radius. Input, unchanged upon output.
- * H(N) – Real*4 array, the Hydrogen density normalised to the post-main-shock density. Input, unchanged upon output.
- * T(N) – Real*4 array, the temperature normalised to the post-main-shock temperature. Input, unchanged upon output.
- * V(N) – Real*4 array, the gas velocity normalised to the post-main-shock velocity. Input, unchanged upon output.
- * N – Integer*4, the number of points on which the hydrodynamical model is tabulated. Input, unchanged upon output.
- * A – Real*4, the energy integral (called α by Sedov).
- * S – Real*4, dimensionless swept-up interstellar mass or $(1/4\pi) \int dr (r/r_s)^2 (\rho/\rho_1)$, where ρ_1 is the post-main-shock density and r_s the main shock radius. The integration is from the contact discontinuity to the main shock. Output.
- * E – Real*4, dimensionless shocked ejecta mass, defined similar to S but now integrated from the reverse shock to the contact discontinuity. Output.

Subprograms called:

- ++ SPINT – spline integration.

Subroutine SNRSHELL(R, R1, R2, IR, NR, PAR, NPAR, RG, R0G, HG, TG, VG, M, VC, T, H, TIME, MI, VOL, TW, HW, TIMEW, ZW)

This subroutine calculates the density and temperature evolution of a mass element in a self-similar

supernova remnant. It calls TIMEHIST which returns dimensionless variables and scales all parameters to the correct physical units.

The parameters are:

- * R, R1, R2, NR, PAR, NPAR, RG, R0G, HG, TG, VG, M, VC – see subroutine SELFSIM for a description.
- * IR – Integer*4, the shell number currently treated. Input.
- * T(MI) – Real*4 array, output, the temperature in K (NOT in keV!) as a function of time for the current shell.
- * H(MI) – Real*4 array, output, the Hydrogen density in cm^{-3} as a function of time for the current shell.
- * TIME(MI) – Real*4 array, output, the time grid used for T and H, in units of s.
- * MI – Integer*4, the number of time bins used for this shell. Output.
- * VOL – Real*4, the emissivity normalisation of the shell (essentially the volume of the shell multiplied by a conversion factor from volume emissivity to observed spectrum at earth. Output.
- * TW, HW, TIMEW, ZW – Real*4 variables for the current shell. For a full description, see the corresponding arrays defined in subroutine SNRMOD before.

Subprograms called:

++ TIMEHIST – determines the dimensionless time history of a mass element.

Subroutine TIMEHIST(O, XN, RR, RC, R, RG, R0G, HG, TG, VELG, M, T, H, TIME, VIM, MI)

This subroutine calculates the evolution of a mass element in a self-similar supernova remnant model with a reverse shock. The basis for this program is a grid on which the basic hydrodynamical model is given. On this grid, the time history is reconstructed by travelling from the present mass element position to either the reverse shock (for a position within the shocked ejecta) or to the main shock (for a position in the shocked interstellar matter). Using the Lagrangian radius of each point on the grid, the relevant time in history can be reconstructed and with this time and the known scaling laws for the self-similar models, the actual values of the hydrodynamical variables in the past can be reconstructed. The output data are ordered with increasing time.

The parameters are:

- * O – Real*4, the interstellar density gradient ω . Input, unchanged upon output.
- * XN – Real*4, the stellar ejecta density gradient n . Input, unchanged upon output.
- * RR – Real*4, the reverse shock radius in units of the main shock radius. Input, unchanged upon output.
- * RC – Real*4, the contact discontinuity position in units of the main shock radius. Input, unchanged upon output.
- * R – Real*4, the present position of the mass element in units of the main shock radius. Input, unchanged upon output.

- * RG(M) – Real*4 array, the real position grid (1 at shock). Input, unchanged upon output.
- * R0G(M) – Real*4 array, the Lagrangian radius grid (1 at shock). Input, unchanged upon output.
- * HG(M) – Real*4 array, the mass density grid (1 at shock). Input, unchanged upon output.
- * TG(M) – Real*4 array, the temperature grid (1 at shock). Input, unchanged upon output.
- * VELG(M) – Real*4 array, the gas velocity grid (1 at shock). Input, unchanged upon output.
- * M – Integer*4, the number of bins in the hydrodynamical model grid. Input, unchanged upon output.
- * T(MI) – Real*4 array, the temperature history of the mass element. Units: present post-shock temperature. Output.
- * H(MI) – Real*4 array, the mass density history of the mass element. Units: present post-shock mass density. Output.
- * TIME(MI) – Real*4 array, the time history of the mass element. Units: present remnant age. Output.
- * VIM – Real*4, present gas velocity of the mass element. Units: present post-shock gas velocity. Output.
- * MI – Integer*4, the number of output time history bins. Output.

Subroutine SPINT(X, Y, DT, T, N, YP1, YPN)

This subroutine performs cubic spline integrations. The method of determining the spline was taken from Press *et al.* (1986). The parameters are:

- * X(N) – Real*4 array, the x-values. Input, unchanged upon output.
- * Y(N) – Real*4 array, the y-values (function to be integrated). Input, unchanged upon output.
- * DT(N) – Real*4 array, spline coefficients. On output, DT(I) contains the integral from X(I-1) to X(I) of the function $y(x)$ (and for I=1, DT(1) = 0.)
- * T(N) – Real*4 array, spline coefficients. On output, T(I) contains the integral from X(1) to X(I) of the function $y(x)$.
- * YP1 – Real*4, determines whether to use a natural spline or not at the start point. See Press *et al.* Input, unchanged upon output.
- * YPN – Real*4, determines whether to use a natural spline or not at the end point. See Press *et al.* Input, unchanged upon output.

Subroutine BAND(G, S, XN, EREL, R2, RC, EF, FE, ALPHA, XME, XMI, A, X, H, T, U, N)

This subroutine solves the isothermal or isentropic supernova remnant model of Band (1988). After inputting the value for the relative strength of the reverse shock, the program calculates the solution in the shocked ejecta. This solution yields a parameter δ_0 . The solution in the outer region must have the same value for δ_0 , and the value for the forward shock strength required to do this is determined

iteratively by means of bisection. The resulting solutions in both regions are scaled properly, and the mass integral is determined; this mass integral is finally transformed into the Lagrangian radius. The subroutine uses two variants of the same subroutine for the calculation of the hydrodynamical solution; one is optimised for speed in the loop which determines the parameter δ_0 , and serves solely to determine that parameter; the other one is more extended and computes several other required parameters (but is slightly slower).

The parameters are:

- * G – Real*4, the adiabatic index. Input, unchanged upon output. Allowed values: 1 for the isothermal model, 5/3 for the isentropic model.
- * S – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * XN – Real*4, the stellar ejecta density gradient n defined as before in SNRSPEC. Input, unchanged upon output.
- * EREL – Real*4, $\eta_r/\eta_{r,min}$, i.e. the density jump at the reverse shock in units of the minimum density jump at the reverse shock. This minimum density jump is given by $\eta_{r,min} = 1 + 1/\Gamma$, where Γ is the adiabatic index. Allowed value: ≥ 1 . For EREL extremely large, one might enter numerical problems. For EREL=1 one obtains the maximum-energy solution. Input, unchanged upon output.
- * R2 – Real*4, reverse shock radius in units of the main shock radius. Output.
- * RC – Real*4, radius of the contact discontinuity in units of the main shock. Output.
- * EF – Real*4, the relative density jump at the forward shock. Output.
- * FE – Real*4, the fraction of the total energy within the shocked material divided by the total energy that went into the shocked material. Smaller than 1. Output.
- * ALPHA – Real*4, α , the dimensionless energy scaling factor, in the nomenclature of Sedov, i.e. the explosion energy E_0 is related to the shock radius r_s and the present-day pre-shock interstellar mass density ρ_0 by $r_s(t) = (E_0 t^2 / \alpha \rho_0)^{1/(5-\omega)}$. Output.
- * XME – Real*4, the mass of the shocked stellar ejecta in units of the post-main-shock density times r_s^3 , where r_s is the main shock radius. Output.
- * XMI – Real*4, the mass of the shocked interstellar matter in units of the post-main-shock density times r_s^3 , where r_s is the main shock radius. Output.
- * A(N) – Real*4 array, the Lagrangian radius (1 at shock). Output.
- * X(N) – Real*4 array, the radius (1 at shock). Output.
- * H(N) – Real*4 array, the mass density (1 at shock). Output.
- * T(N) – Real*4 array, the temperature (1 at shock). Output.
- * U(N) – Real*4 array, the gas velocity (1 at shock). Output.
- * N – Integer*4, the number of bins on which the hydrodynamical model is calculated. The bins are ordered from the reverse shock to the main shock. Output.

Subprograms called:

++ BANDSHAPE – to determine the solution of Bands differential equations.

++ BANDFAST – a turbo-version of BANDSHAPE

Subroutine BANDSHAPE(IR, G, XN, S, ETA, D0, ST, SK, XW, UW, HW, AW, N)

This subroutine solves the basic differential equation of Band (1988). The differential equation as given in the paper of Band should be integrated up to a variable boundary. This nasty property was removed by substituting instead of the radial coordinate χ the new variable $p \equiv \chi/U$. The differential equation is then integrable between fixed boundaries. We did the integrations simply in 10000 equidistant steps in p , integrating the equations by the simplest method. The results agreed in nearly all cases within all tabulated digits with the solution in Band's paper.

The parameters are:

- * IR – Integer*4, Input. Unchanged upon output. Allowed values: -1 for the reverse shock, 1 for the forward shock.
- * G – Real*4, the adiabatic index. Input, unchanged upon output. Allowed values: 1 for the isothermal model, 5/3 for the isentropic model.
- * XN – Real*4, the stellar ejecta density gradient n defined as before in SNRSPEC. Input, unchanged upon output.
- * S – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * ETA – Real*4, the density jump at the shock. Input, unchanged upon output.
- * D0 – Real*4, the δ_0 parameter of Band. Output.
- * ST – Real*4, the dimensionless thermal energy integral of Band. Output.
- * SK – Real*4, the dimensionless kinetic energy integral of Band. Output.
- * XW(N) – Real*4 array, Band's χ (dimensionless position). Output.
- * UW(N) – Real*4 array, Band's U (dimensionless velocity). Output.
- * HW(N) – Real*4 array, Band's Ω (dimensionless density). Output.
- * AW(N) – Real*4 array, mass integral $\int \Omega \chi^2 d\chi$ from the shock to the present position χ . Output.
- * N – Integer*4, number of bins used. This number is smaller than the number of integration steps! The bins are ordered from the (reverse) shock to the contact discontinuity.

Function BANDFAST(G, XN, S, ETA)

This function solves the basic differential equation of Band (1988), similar to subroutine BANDSHAPE. This routine is much faster because it does not store intermediate results, neither calculates mass integrals etc. It works only for the main shock, and is optimised to determine the parameter δ_0 . The parameters are:

- * G – Real*4, the adiabatic index. Input, unchanged upon output. Allowed values: 1 for the isothermal model, 5/3 for the isentropic model.

- * XN – Real*4, the stellar ejecta density gradient n defined as before in SNRSPEC. Input, unchanged upon output.
- * S – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * ETA – Real*4, the density jump at the shock. Input, unchanged upon output.
- * BANDEFAST – Real*4, the function value, the δ_0 parameter of Band. Output.

Subroutine CHEVAL(G, S, XN, R2, RC, A, X, Z, P, U, N, NI)

This subroutine solves the differential equations for Chevalier’s reverse shock model. In two calls to different subroutines the solution in the shocked ejecta and interstellar matter is determined. These solutions are scaled properly, and then the Lagrangian radius is determined.

The parameters are:

- * G – Real*4, the adiabatic index. Input, unchanged upon output. G should be larger than 1.
- * S – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * XN – Real*4, the stellar ejecta density gradient n defined as before in SNRSPEC. Input, unchanged upon output.
- * R2 – Real*4, reverse shock radius in units of the main shock radius. Output.
- * RC – Real*4, radius of the contact discontinuity in units of the main shock. Output.
- * A(N) – Real*4 array, the Lagrangian radius (1 at shock). Output.
- * X(N) – Real*4 array, the radius (1 at shock). Output.
- * Z(N) – Real*4 array, the mass density (1 at shock). Output.
- * P(N) – Real*4 array, the temperature (1 at shock). Output. During some phase of the program, it contains the pressure (therefore its name is P).
- * U(N) – Real*4 array, the gas velocity (1 at shock). Output.
- * N – Integer*4, the number of bins on which the hydrodynamical model is calculated. The bins are ordered from the reverse shock to the main shock. Output.
- * NI – Integer*4, the number of bins below the contact discontinuity.

Subprograms called:

- ++ INNER – determines the solution in the shocked ejecta
- ++ OUTER – determines the solution in the shocked interstellar matter.

Subroutine OUTER(DX0, NW, XN, S, G, XD, UD, ZD, PD, AD, N)

This subroutine determines the self-similar solution for the outer blast-wave part of the SNR in

Chevalier's adiabatic model. It is based upon Parker (1963). The algorithm is straightforward integration. The parameters are:

- * DX0 – Real*4, a scale parameter used to set the step size in $\ln(r)$. 0.0001 is a good choice. Input, unchanged upon output.
- * NW – Integer*4, each NW'th integration step is stored. Input, unchanged upon output. NW=25 is a good choice.
- * XN – Real*4, the stellar ejecta density gradient n defined as before in SNRSPEC. Input, unchanged upon output.
- * S – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * G – Real*4, the adiabatic index. Input, unchanged upon output.
- * XD(N) – Real*4 array, the dimensionless positions (1 at the shock). Output.
- * UD(N) – Real*4 array, the dimensionless velocity, i.e. Parker's U . Output.
- * ZD(N) – Real*4 array, dimensionless temperature Z according to Parker. Output.
- * PD(N) – Real*4 array, dimensionless pressure. Output. This pressure is determined apart from a normalisation factor, which has to be derived later.
- * AD(N) – Real*4 array, dimensionless mass integral. Output.
- * N – Integer*4, the number of output bins generated. Output. The bins are ordered from the main shock to the contact discontinuity.

Subroutine INNER(DX0, NW, XN, S, G, XD, UD, ZD, PD, AD, N)

This subroutine determines the self-similar solution for the inner shocked ejecta part of the SNR in Chevalier's adiabatic model. The algorithm is straightforward integration. The parameters are:

- * DX0 – Real*4, a scale parameter used to set the step size in $\ln(r)$. 0.0001 is a good choice. Input, unchanged upon output.
- * NW – Integer*4, each NW'th integration step is stored. Input, unchanged upon output. NW=50 is a good choice.
- * XN – Real*4, the stellar ejecta density gradient n defined as before in SNRSPEC. Input, unchanged upon output.
- * S – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * G – Real*4, the adiabatic index. Input, unchanged upon output.
- * XD(N) – Real*4 array, the dimensionless positions (1 at the reverse shock). Output.
- * UD(N) – Real*4 array, the dimensionless velocity. Output.
- * ZD(N) – Real*4 array, dimensionless temperature Z . Output.

- * PD(N) – Real*4 array, dimensionless pressure. Output. This pressure is determined apart from a normalisation factor, which has to be derived later.
- * AD(N) – Real*4 array, dimensionless mass integral. Output.
- * N – Integer*4, the number of output bins generated. Output. The bins are ordered from the reverse shock to the contact discontinuity.

Subroutine SEDOV(G, O, N, R, R0, RHO, T, VEL)

This subroutine determines the Sedov solution for a blastwave. The formulae used are based upon Sedov (1959), chapter 14. The solution is determined both for ω close to its critical value, as well as for other values. The grid of data points is created equidistant in Sedov's dimensionless velocity parameter V . The program first determines the boundaries of V for the present parameters, then creates the V -grid and determines the solution on that grid. In this subroutine and in subroutine LSEDOV which is called by the subroutine, frequently use is made of double precision calculations. Where this is done, it is essentially for the computational stability, especially near the centre of the remnant the errors become large otherwise.

The parameters are:

- * G – Real*4, the adiabatic index. Input, unchanged upon output. This parameter should be larger than 1.
- * O – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output. Should be smaller than 5.
- * N – Integer*4, the number of bins on which the hydrodynamical model is calculated. The bins are ordered from the centre towards the main shock. Input, unchanged upon output.
- * R(N) – Real*4 array, the radius (1 at shock). Output.
- * R0(N) – Real*4 array, the Lagrangian radius (1 at shock). Output.
- * RHO(N) – Real*4 array, the mass density (1 at shock). Output.
- * T(N) – Real*4 array, the temperature (1 at shock). Output.
- * VEL(N) – Real*4 array, the gas velocity (1 at shock). Output.

Subprograms used:

++ LSEDOV – determines the solution for one value of V .

Subroutine LSEDOV(GG, OO, EE, V, R, R0, RHO, T, VEL)

This subroutine determines the Sedov solution for a given value of Sedov's parameter V . It calculates the radius, Lagrangian radius, density, temperature and velocity of the gas with respect to post-shock quantities. For reasons of computational stability, the routine is written in double precision. Speed is regained by storing several numerical quantities which are required for the same set of input variables, and by using logarithms in the lengthy expressions for the output variables (otherwise many exponentiations would be necessary). The parameters are:

- * GG – Real*4, the adiabatic index. Input, unchanged upon output.

- * OO – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output.
- * EE – Real*4, ϵ . A small parameter (e.g. $\epsilon = 0.02$); for $|\omega - \omega_{cr}| < \epsilon$ the solution is near the critical solution, and the simpler algorithm for the critical solution is used. Input, unchanged upon output.
- * V – Real*8, Sedov’s V . Input, unchanged upon output.
- * R – Real*4, the radius (1 at shock). Output.
- * R0 – Real*4, the Lagrangian radius (1 at shock). Output.
- * RHO – Real*4, the mass density (1 at shock). Output.
- * T – Real*4, the temperature (1 at shock). Output.
- * VEL – Real*4, the gas velocity (1 at shock). Output.

Subroutine ISOTHERM(ETA, O, M, RG, R0G, HG, TG, VG)

This subroutine determines the hydrodynamical solution for the isothermal model of Solinger *et al.* (1985) and Lerche and Vasyliunas (1976). It first solves the basic differential equation, scales it, then by integration the density profile is determined, and finally the Lagrangian radius is determined from the mass-integral. The parameters are:

- * ETA – Real*4, the density jump at the shock. Output.
- * O – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output. Take $0 \leq \omega < 3$.
- * M – Integer*4, the number of bins on which the hydrodynamical model is calculated. The bins are ordered from the centre towards the main shock. Output.
- * RG(M) – Real*4 array, the radius (1 at shock). Output.
- * R0G(M) – Real*4 array, the Lagrangian radius (1 at shock). Output.
- * HG(M) – Real*4 array, the mass density (1 at shock). Output.
- * TG(M) – Real*4 array, the temperature (1 at shock). Output.
- * VG(M) – Real*4 array, the gas velocity (1 at shock). Output.

Subprograms used:

- ++ ISOINT – determines the basic solution of the differential equation.

Subroutine ISOINT(O, DX0, NS, XW, UW, IW, ETA)

This subroutine determines the basic solution to the differential equation for the isothermal supernova remnant model, cf. Lerche and Vasyliunas (1976). The critical density gradient $\omega_{cr} = (13 - \sqrt{160})/9$. For $\omega < \omega_{cr}$, the differential equation is integrated directly from the centre to the shock; in the other case, first the critical points of the equation are calculated; then the integration is done in 3 steps: from the centre to the lower critical point, from the upper to the lower critical point and from the upper critical

point to the shock. The parameters of this subroutine are:

- * O – Real*4, the interstellar density gradient ω defined before in SNRSPEC. Input, unchanged upon output. Take $0 \leq \omega < 3$.
- * DX0 – Real*4, the integration step in the variable ξ . For ω close to 3, this variable should be larger than for other values of ω ! Input, unchanged upon output.
- * NS – Integer*4, input, unchanged upon output. Every NSth step in the integration is stored. The subroutine does not check for you if you have your arrays declared sufficiently large.
- * XW(IW) – Real*4 array, the scaled dimensionless radius. Output.
- * UW(IW) – Real*4 array, the scaled dimensionless velocity. Output.
- * IW – Integer*4, the number of bins on which the hydrodynamical model is actually calculated. The bins are ordered from the centre towards the main shock. Output.
- * ETA – Real*4, the density jump at the shock. Output.

Subroutine PROSPEC(FLS, FLT, FLX, E1, E2, R, R1, R2, Z, NBIN, NR, NX, XP, SR, SMIN, SMAX, EMIN, EMAX)

This subroutine corrects the supernova remnant spectrum for the radial expansion velocity and projects it onto the sky. It also has an option to convolve the spectrum with a Gaussian and to apply intergalactic absorption. In the last case, the common block /SNRS/ should be filled. See below for a description.

The parameters are:

- * FLS(NBIN,NRMAX) – Real*4 array, containing the spectra of each separate shell. Input. Units: photons/s/m²/keV. On output, it contains the shell spectra in photons/s/m²/(shock radius)³.
- * FLT(NBIN) – Real*4 array, scratch. Contains the total SNR spectrum of a radial ring projected onto the sky (on output, the spectrum of the outermost ring).
- * FLX(NBIN) – Real*4 array, contains on output the total SNR spectrum.
- * E1(NBIN) – Real*4 array, input. Contains the lower energy boundaries of the energy bins.
- * E2(NBIN) – Real*4 array, input. Contains the upper energy boundaries of the energy bins.
- * R(NR), R1(NR), R2(NR) – Real*4 arrays, containing the centres, lower and upper boundaries respectively of the positions in the remnant where the spectrum is evaluated (the "shells"). All in units of the main shock radius (thus the values are in general less than 1). Input.
- * Z(NR) – Real*4 array containing the redshifts of the shells (the expansion velocities / c , positive for expansion, negative for contraction (which never happens in our models)). Input.
- * NBIN – Integer*4, the number of energy bins used.
- * NR – Integer*4, the number of shells actually used (smaller than or at most NRMAX). Input.
- * NX – Integer*4, the number of projected rings in which the spectrum must be evaluated. Also at most equal to NRMAX. Input.

- * XP(NX) – Real*4 array, output. The position grid on which the projected spectra are calculated.
- * SR(NX) – Real*4 array, output. Contains the radial X-ray emission profile of the remnant within the energy range specified by EMIN and EMAX, in units of photons/s/m²/keV/(shock radius)².
- * SMIN, SMAX – Real*4, the minimum and maximum projected radius on the sky between which the total spectrum should be evaluated. Units: main shock radius. Input. If the complete remnant is considered, then SMIN=0 and SMAX=1.
- * EMIN, EMAX – Real*4, the minimum and maximum energy (keV) between which the projected profile must be evaluated. Input.

Commonblock /SNRS/CN,HN,ICON

- * ICON – Integer*4, input. If ICON=0, normal mode (no convolution with Gaussian or absorption). If ICON=1, convolution with a Gaussian and intergalactic absorption is applied. In that case only (ICON=1), the energy grid E1, E2 MUST correspond to a FWHM/3 energy grid (i.e. a resolution of 1/3 of the full width at half maximum).
- * HN – Real*4, input, column density in units of 10²⁰ cm⁻².
- * CN – Real*4, currently not used.

Subprogram called:

++ USTEL – the interstellar absorption of Morrison and McCammon (only called ofr ICON=1).

Function USTEL(E)

This function calculates the interstellar absorption cross section of Morrison and McCammon.

Parameters:

- * E – Real*4, the photon energy in keV. Input.

References

- Anders, E., Grevesse, N.: 1989, *Geochimica et Cosmochimica Acta* **53**, 197
- Abramowitz, M., Stegun, I.A.: 1968, *Handbook of mathematical functions*, 5th printing, Dover, New York
- Allen, C.W.: 1973, *Astrophysical quantities*, 3rd Ed., Athlone Press, London
- Arnaud, M., Rothenflug, R.: 1985, *Astron. Astrophys. Suppl.* **60**, 425
- Band, D.L.: 1988, *Astrophys. J.* **332**, 842
- Carson, T.R.: 1988, *Astron. Astrophys.* **189**, 319
- Chevalier, R.A.: 1982, *Astrophys. J.* **258**, 790
- Dalgarno, A., Drake, G.W.F.: 1979, in *Les transitions interdites dans les spectres des astres*, p. 69, ed. Les Congrès et Colloques de l'Université de Liège, Belgium, Vol. 54
- Gronenschild, E.H.B.M., Mewe, R.: 1978, *Astron. Astrophys. Suppl.* **32**, 283
- Jansen, F.A.: 1988, *X-ray photometric morphology of the Cas A and Puppis A supernova remnants*, Thesis, Leiden
- Karzas, W.J., Latter, R.: 1961, *Astrophys. J. Suppl. Series* **6**, 167
- Kylafis, N.D., Lamb, D.Q.: 1982, *Astrophys. J. Suppl. Series* **48**, 239
- Lang, K.R.: 1974, *Astrophysical formulae*, Springer, Berlin
- Lerche, I., Vasyliunas, V.M.: 1976, *Astrophys. J.* **210**, 85
- Mewe, R., Gronenschild, E.H.B.M., van den Oord, G.H.J.: 1985, *Astron. Astrophys. Suppl.* **62**, 197
- Mewe, R., Lemen, J.R., van den Oord, G.H.J.: 1986, *Astron. Astrophys. Suppl.* **65**, 511
- Mewe, R., Schrijver, J.: 1978, *Astron. Astrophys.* **65**, 99
- Parker, E.N.: 1963, *Interplanetary dynamical processes*, Oxford University Press, Oxford, U.K.
- Press, W.H.: Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1986, *Numerical Recipes*; Cambridge University Press, Cambridge, U.K.
- Ross, J.E., Aller, L.H.: 1976, *Science* **191**, 1223
- Sedov, L.: 1959, *Similarity and dimensional methods in mechanics*, Academic Press, New York
- Smith, B.W., Raymond, J.C., Mann, J.B., Cowan, R.D.: 1985, *Astrophys. J.* **298**, 898
- Solinger, A., Rappaport, S., Buff, J.: 1975, *Astrophys. J.* **201**, 381
- Spitzer, L., Greenstein, J.L.: 1951, *Astrophys. J.* **114**, 407