Chapter 2

UNIRAD Improvements and subroutine library

In this chapter we describe a series of improvements to the `UNIRAD` package performed by the TREND team at BIRA/IASB. Two types of improvements have been made:

1. the addition of new flux maps from five satellite data sets: AZUR/EI-88, SAMPEX/PET, UARS/PEM, CRRES/MEA, and ISEE/WAPS, and the addition of the ESA-SEE1, CRRESPRO, and CRRESELE models;

2. the development of a library of subroutines.

2.1 Implementation of new trapped particle models in `UNIRAD`

The first series of improvements of `UNIRAD` has been the implementation of new radiation belt models that have been developed during this study (detailed descriptions of the development of the new models can be found in Technical Notes 1, 2, and 5 of this study). In addition, the ESA-SEE1 model developed by A. Vampola (1996) on the basis of a neural network study of the CRRES/MEA data, has been implemented. Finally, the CRRESPRO (Meffert & Gussenhoven 1994) and CRRESELE (Brautigam & Bell 1995) models, developed by Phillips Laboratory Geophysics Directorate (PLGD), have also been adapted to and integrated in `UNIRAD`.

Table 2.1 contains a list of the new radiation belt models, the data sets they were constructed from, and the Technical Note number in which they are described in detail.

2.1.1 The `UNIRAD` program suite

The `UNIRAD` suite of programs provides information about the radiation environment in an arbitrary Earth orbit, predicting satellite exposures to particle fluxes, the resulting radiation dose,

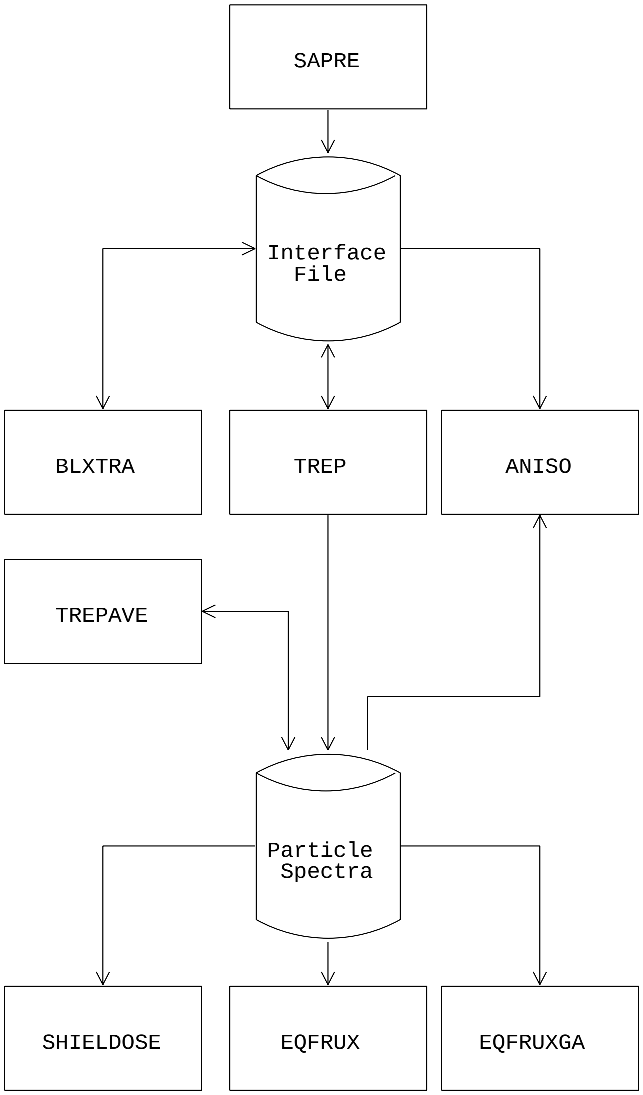

Figure 2.1. Flow diagram of `UNIRAD`

and the resulting damage-equivalent fluences for solar cell degradation calculations (Heynderickx et al. 1996d).

From the orbit parameters, the system will generate a detailed trajectory, magnetic coordinates, integral and differential proton and electron fluences, doses for three shield geometries in four detector materials, and solar cell degradation information, in both printed and graphical form.

The `UNIRAD` package consists of the following programs:

- `SAPRE`: orbit generator which produces a data file used by the next two programs in the package;

- BLXTRA: calculates the geomagnetic coordinates (B, L) from the geographic coordinates generated by SAPRE;

- TREP: determines the radiation flux for the geographic coordinates generated by SAPRE from the NASA trapped radiation models AP-8 and AE-8 and determines the solar proton flux over the mission. It produces a data file with the energy spectra of trapped protons and electrons and of solar protons;

- TREPPOS: calculates the trapped radiation flux for pairs of (B, L) or $(B/B_0, L)$ coordinates interactively input by the user. It produces a data file with the energy spectra of trapped protons and electrons;

- TREPAVE: averages the spectra generated by TREP or TREPPOS for different orbits;

- ANISO: transforms the trapped proton omnidirectional integral flux produced by TREP into unidirectional integral and differential fluxes, taking into account pitch angle and azimuthal dependence. The user can define a set of look directions with respect to a satellite reference frame. The resulting fluxes are averaged over the orbit.

- ANISOPOS: provides the angular distribution (i.e. pitch angle and azimuthal dependence) of the unidirectional integral or differential flux at a given geographic location.

- SHIELDOSE: reads the energy spectra resulting from TREP, TREPAVE, or TREPPOS and converts them to radiation dose-depth curves for different detector materials and simple shielding geometries;

- EQFRUX: determines 1 MeV electron damage equivalent fluences from the TREP spectra to evaluate degradation of Si solar cells;

- EQFRUXGA: idem as EQFRUX, but for GaAs solar cells;

- UNIRAD.PRO: a set of IDL routines to produce graphical output.

The flow diagram of UNIRAD is shown in Fig. 2.1.

2.1.2 The TREP program

TREP Calculates orbital radiation environment fluxes for a spacecraft trajectory generated by SAPRE. It estimates fluxes of trapped particles in function of (B, L) coordinates, and computes solar proton event probabilities to estimate solar proton fluences. The output consists of:

1. the time dependent fluxes of trapped protons, solar protons, and trapped electrons;

2. trajectory-averaged spectra for trapped protons and electrons;

3. solar flare fluence spectra, taking into account geomagnetic shielding.

TREP accesses the NASA trapped particle models AP-8 (Sawyer & Vette 1976) and AE-8 (Vette 1991a) at each orbital point contained in the common interface file produced by SAPRE. The NASA trapped radiation models AP-8 and AE-8 are distributed as tables of omnidirectional fluxes in function of particle energy E and $(B/B_0, L)$, where

$$B_0 = \frac{0.311653}{L^3} \,. \tag{2.1}$$

The NASA models AP-8 and AE-8 are static models built with data obtained in the sixties and seventies. There are two sets of models: AP-8 MAX and AE-8 MAX for solar maximum conditions, and AP-8 MIN and AE-8 MIN for solar minimum conditions. Since these models were built with specific magnetic field models, the same magnetic field models should be used to calculate the $(B/B_0, L)$ values that serve as input to AP-8 and AE-8. The appropriate magnetic field models are: Jensen & Cain (1962) for AE-8 MIN, AP-8 MIN, and AE-8 MAX, and GSFC 12/66 (Cain et al. 1967) updated to epoch 1970 for AP-8 MAX (Heynderickx et al. 1996ab). Two coordinate pairs (B, L) are written to the common interface file PROJECT.INT, corresponding to the field models used with the proton and electron model, respectively.

The position of the SAA in the old magnetic field models to be used with AP-8 and AE-8 is different from its actual position due to the secular variation on the geomagnetic field (Fraser-Smith 1987). This has important effects on the prediction of trapped particle fluxes for low altitude orbits. Therefore, a correction has been built into TREP: when the NAMELIST parameter SAAROT is set to 1 (the default value), a correction term $\Delta\phi(t)$ is added to the geodetic longitude before the calculation of (B, L) (the actual value of the longitude on the interface file is not changed):

$$\Delta\phi(t) = 0.3^\circ \, (t - \texttt{EPOCH}) \,. \tag{2.2}$$

where t is the orbit epoch as written in the header of the common interface, and EPOCH is the epoch of the geomagnetic field model. This correction is also made for the new trapped particle models. The new models were constructed with the IGRF magnetic field model for the epoch of the observations and external magnetic field models for the models that extend beyond the inner radiation belt.

2.1.3 New trapped particle models

Several new proton and electron models have been developed during the TREND-3 study. They are listed in Table 2.1, with their epoch and number of submodels. The models are described in the references given in Table 2.1. The ESA-SEE1 model developed by Vampola (1996) has been implemented as well as the CRRESPRO (Meffert & Gussenhoven 1994) and CRRESELE (Brautigam & Bell 1995) models developed by Phillips Laboratory.

The naming convention for the new models is as follows:

1. the first character of the name indicates whether it is a proton (P) or an electron (E) model;

Table 2.1. List of the new flux models implemented in TREP. N Is the number of submodels.

Name	Particle	Satellite	Instrument	Epoch	N	Reference
PAB97	Protons	AZUR	EI-88	1970	1	Technical Note 5
PSB97	Protons	SAMPEX	PET	1995	1	Technical Note 5
PUB97	Protons	UARS	PEM	1992	1	Technical Note 5
PCP94	Protons	CRRES	PROTEL	1991	2	Meffert & Gussenhoven (1994)
ECS96	Electrons	CRRES	MEA	1991	1	Vampola (1996)
ECM97	Electrons	CRRES	MEA	1991	5	Technical Note 2
EIM97	Electrons	ISEE	WAPS	1979	5	Technical Note 1
ECP95	Electrons	CRRES	HEEF	1991	8	Brautigam & Bell (1995)

2. the second character identifies the spacecraft which collected the data (‘A’ stands for AZUR, ‘S’ for SAMPEX, ‘U’ for UARS, ‘C’ for CRRES, and ‘I’ for ISEE);

3. the third character identifies the institute where the model was developed (‘B’ stands for BIRA/IASB, ‘M’ for MSSL or MPA, ’P’ for PLGD, and ‘S’ for SEE);

4. the last two characters form the year in which the model was developed (not the epoch of the model!).

2.1.4 The ESA-SEE1 model

The ESA-SEE1 model developed by Vampola (1996) is meant to be a replacement of AE-8 MIN, and therefore has the same structure as the NASA model. The model data file ESA-SEE1.DAT provided by Vampola (1996) is not in the same BLOCK DATA format as the version of the NASA models used in TREP. The program VAMPOLA.FOR was developed to rewrite the ESA-SEE1.DAT file in the format used by TREP. The resulting BLOCK DATA file ECS96BD.FOR is linked to TREP, and makes use of the TRARAP subroutine developed by Daly & Evans (1993) to interpolate in the flux map.

2.1.5 The CRRESPRO and CRRESELE models

The CRRESPRO and CRRESELE software packages developed by Phillips Laboratory determine omnidirectional energetic proton and electron fluences for user-defined orbits. The flux models used in these software packages are based on measurements made with the PROTEL (Violet et al. 1993) and HEEF (Hunerwadel et al. 1987) instruments onboard the Combined Release and Radiation Effects Satellite (CRRES), which flew from 25 July 1990 to 12 October

1991 during solar maximum. CRRES was in a geosynchronous transfer orbit with an inclination of $18°$, a perigee of 350 km, and an apogee of 33 000 km. It traversed the radiation belts twice per orbit with a period of $9^{\rm h}52^{\rm m}$.

Both flux models are provided in a series of binary files containing omnidirectional differential flux maps in $(L, B/B_0)$ space. In order to implement these flux maps in `TREP`, we first converted the binary files to ASCII format. Then, new flux map files were created for omnidirectional integral fluxes, which were obtained by integrating the differential fluxes in accordance with the guidelines of Meffert & Gussenhoven (1994) and Brautigam & Bell (1995). Finally, the new flux maps were converted into `BLOCK DATA` format using the procedure outlined in Sect. 2.1.6.

The conversion from differential flux j to integral flux J is defined as:

$$J(> E_i) = \int_{E_i}^{\infty} j(E)\, dE \,, \tag{2.3}$$

where E_i is the lower energy boundary of channel i. The integration is approximated by a summation:

$$J(> E_i) = \sum_{k=i}^{N} j(E_k)\, \Delta E_k \,, \tag{2.4}$$

with ΔE_k the width of channel k and N the number of channels. The summation for channel i begins at the lower boundary of that channel and ends at the upper boundary of the highest energy channel.

2.1.5.1 The CRRESPRO model

2.1.5.1.1 The PROTEL instrument The CRRESPRO flux models are based on in situ flux measurements made by the PROton TELescope (PROTEL) onboard CRRES. PROTEL Had two detector heads which together measured protons from 1 to 100 MeV in 24 energy steps, giving a complete spectrum every 1.024 s. The angular resolution of the detector low (high) energy head was $\pm 10° \times \pm 10°$ ($\pm 12° \times \pm 17°$). A full description of PROTEL is available in Violet et al. (1993) and Lynch et al. (1989).

The PROTEL detectors comprised detector stacks and a logic system that required single or double coincidence to verify that the proper energy particle is counted. In addition, both active and passive shielding were used around much of the detector stack. The detectors were extensively calibrated prior to launch. During calibration, it was found that energetic protons (> 60 MeV) incident over a large angular angle with respect to the detector axis could degrade sufficiently in the shielding, pass through the necessary angle in the detector stack, and be counted. This contamination was found to be significant enough for very hard spectra, such as occurs for L values between 1.1 and 1.7, to require correction. A correction scheme was devised based on the assumption of an empty atmospheric loss cone (Gussenhoven et al. 1993).

One channel (15.2 MeV) had significantly lower sensitivity than the other channels and one channel was an overlap channel at 8.5 MeV. To avoid both redundancy and the appearance of

Table 2.2. Integration parameters (MeV) for integral omnidirectional PROTEL flux [from Meffert & Gussenhoven (1994)]

Channel	E_{lo}	E_{mid}	E_{hi}	ΔE
1	1.1	1.5	1.9	0.8
2	1.9	2.1	2.3	0.4
3	2.3	2.5	2.7	0.4
4	2.7	2.9	3.1	0.4
5	3.1	4.3	5.5	2.4
6	5.5	5.7	5.9	0.4
7	5.9	6.8	7.7	1.8
8	7.7	8.5	9.3	1.6
9	9.3	9.7	10.1	0.8
10	10.1	10.7	11.3	1.2
11	11.3	13.2	15.1	3.8
12	15.1	19.4	23.7	8.6
13	23.7	26.3	28.9	5.2
14	28.9	30.9	32.9	4.0
15	32.9	36.3	40.2	7.3
16	40.2	41.1	43.2	3.0
17	43.2	47.0	50.8	7.6
18	50.8	55.0	59.2	8.4
19	59.2	65.7	72.2	13.0
20	72.2	81.3	90.4	18.2

discontinuities in differential and integral spectra at certain L values, the 8.5 MeV and 15.2 MeV channels were not used.

2.1.5.1.2 Proton flux models In March 1991, a magnetic storm caused a reconfiguration of the inner magnetosphere, resulting in, among other features, the formation of a second proton belt over a certain energy range. Because of this change, two CRRES models were created. The "quiet" model uses data from July 1990 to March 1991, and the "active" model uses data from March 1991 to October 1991. Note that the terms "quiet" and "active" have no correspondence to quiet and active as determined by the index K_{p}. The channel characteristics of the PROTEL integral flux maps are given in Table 2.2.

The CRRESPRO models are in the form of omnidirectional differential flux maps organised in $(E, L, B/B_0)$ coordinates. The bin widths of the L range are 0.05 R_{E}, while for B/B_0 the bin limits are chosen such that they cover approximately ±68° magnetic latitude in a dipole field with steps of 2°.

The CRRESPRO models have been implemented in UNIRAD as one model, called PCP94,

Table 2.3. Channel energies (MeV) for CRRES/HEEF [from Brautigam & Bell (1995)]

Channel	E_{lo}	E_{mid}	E_{hi}	ΔE
0	0.50	0.65	0.80	0.30
0.a	0.80	0.825	0.85	0.05
1	0.85	0.95	1.05	0.20
1.a	1.05	1.15	1.25	0.20
2	1.25	1.60	1.70	0.45
3	1.70	2.00	2.10	0.40
4	2.10	2.35	2.50	0.40
5	2.50	2.75	2.90	0.40
6	2.90	3.15	3.30	0.40
7	3.30	3.75	4.10	0.80
8	4.10	4.55	4.95	0.85
9	4.95	5.75	6.60	1.65

containing 2 submodels.

2.1.5.2 The CRRESELE model

2.1.5.2.1 The HEEF instrument The flux models used by CRRESELE are based on in situ flux measurements made by the High Energy Electron Fluxmeter (HEEF) onboard CRRES (Brautigam & Bell 1995). HEEF Was designed to measure the flux of 1–10 MeV electrons in 10 differential number flux channels and to return a complete spectrum every 0.512 s. The telescope consists of a well collimated stack of three particle detection elements, the top two being solid state detectors, and the third a bismuth germanate (BGO) crystal. A passive beryllium shield is placed at the entrance of the telescope to stop lower energy (<0.3 MeV) electrons. A plastic scintillator surrounds the BGO crystal to veto particles that penetrate it from the sides and would otherwise trigger accidental counts.

As the result of extensive post-launch analysis of the actual flight and test data of the HEEF replica, the spectrum energies were revised. The lowest energy differential channel was omitted because of the large uncertainty in its geometric factor, and the highest energy differential channel was omitted because its counting efficiency was so low that it rendered little statistically meaningful data for modelling purposes. The final set of energy channels is listed in Table 2.3.

To facilitate the determination of the integral flux, the fictitious channels 0.a and 1.a have been introduced to fill the gaps in the energy spectrum between channels 0 and 1 and between channels 1 and 2, respectively. The omnidirectional flux at these two fictitious channels is determined by interpolating between the adjacent channels in each case. This procedure is followed to provide a stepwise continuous spectrum, which can then be summed over a discrete set of energies to give the integral omnidirectional flux.

Table 2.4. Summary of CRRES A_{p15} model separation and statistics [from Brautigam & Bell (1995)]

Model Nr.	A_{p15} Range	Average A_{p15}	Days/model	% of total
0	5.0–7.5	6.8	13	3.2
1	7.5–10.0	8.7	83	20.7
2	10.0–15.0	12.5	69	17.1
3	15.0–20.0	17.1	82	20.6
4	20.0–25.0	22.4	72	17.7
5	25.0–55.0	35.7	83	20.7

2.1.5.2.2 Electron flux models The potential problem of energetic proton contamination of HEEF in the inner zone has not been addressed in the construction of the CRRESELE model. The slot region, populated by the temporary third electron belt during the second half of the CRRES mission, would require special treatment in comparison to the outer belt. Therefore, the CRRESELE models are developed for the outer zone electron belts only, and are limited to the interval 2.5$\leq L \leq$6.8.

The CRRESELE models are in the form of omnidirectional differential flux maps organised in $(E, L, B/B_0)$ coordinates. The bin widths of the L range are 0.05 R_{E}, while for B/B_0 the bin limits are chosen such that they cover approximately ±68° magnetic latitude in a dipole field with steps of 2°.

Eight models are available. Six of these models are parametrized by geomagnetic activity, i.e. in terms of the A_{p15} index which is defined as the average of the preceding 15 daily values of A_{p}. Six ranges of A_{p15} were defined, for each of which a full electron model was constructed. The statistics of the six A_{p15} ranges are given in Table 2.4. Two additional models, independent of A_{p15}, are available as well. For model 6, the entire data base was averaged, providing a mission average model. Model 7 was constructed from the maximum flux found at each L bin of the daily averaged data base.

The CRRESELE models have been implemented in UNIRAD as one model, called ECP95, containing 8 submodels.

2.1.6 Structure of the model data files

For all the models listed in Table 2.1, except the ESA-SEE1 model, a new model format and a set of interpolation routines were developed. The flux maps are stored in data files, one for each model, which are transformed into BLOCK DATA files by the program MODTOBD.FOR. The format of the data files is common to all models. The number of submodels in Table 2.1 refers to the number of flux maps in a model corresponding to, for instance, different geomagnetic conditions. For each submodel, the internal and external magnetic field parameters have to be specified.

The flux maps are given on a three-dimensional rectangular grid in E, L, and a third coordinate x which is either B, B/B_0, or the equatorial pitch angle α_0. The third coordinate is identified in the third record of the flux map data file as 1, 2, or 3, corresponding to the above coordinates. The directionality of the flux map is given by a flag: 0 for unidirectional, 1 for omnidirectional. The fluxes are considered to be integral fluxes.

2.1.7 Interpolation in flux maps

The interpolation in the flux maps is performed tri-linearly in (E, L, x) space over the parallepiped containing the point for which the flux $f(E, L, x)$ is required. The vertices of this parallepiped are (E_1, L_1, x_1), (E_1, L_1, x_2), (E_1, L_2, x_1), (E_1, L_2, x_2), (E_2, L_1, x_1), (E_2, L_1, x_2), (E_2, L_2, x_1), and (E_2, L_2, x_2), where $E \,\epsilon\, [E_1, E_2]$, $L \,\epsilon\, [L_1, L_2]$, and $x \,\epsilon\, [x_1, x_2]$. The interpolation function has the form:

$$\begin{aligned} f(E,L,x) \quad = \quad & (1-p)\,(1-q)\,(1-r)\,f(E_1,L_1,x_1) + (1-p)\,(1-q)\,r\,f(E_1,L_1,x_2) \\ & + (1-p)\,q\,(1-r)\,f(E_1,L_2,x_1) + (1-p)\,q\,r\,f(E_1,L_2,x_2) \\ & + p\,(1-q)\,(1-r)\,f(E_2,L_1,x_1) + p\,(1-q)\,r\,f(E_2,L_1,x_2) \\ & + p\,q\,(1-r)\,f(E_2,L_2,x_1) + p\,q\,r\,f(E_2,L_2,x_2)\,, \end{aligned} \tag{2.5}$$

with

$$\left.\begin{aligned} p &= \frac{E-E_1}{E_2-E_1} \\ q &= \frac{L-L_1}{L_2-L_1} \\ r &= \frac{x-x_1}{x_2-x_1} \end{aligned}\right\}\,. \tag{2.6}$$

Interpolation of the flux maps between grid values is handled by the subroutine `MODINT`, which is contained in the program file `MODINT.FOR`. This subroutine takes as input the number of grid bins in each dimension, the grid limits and reference values, a vector containing the flux values over the grid, and a vector of energies (and its length) plus a pair of L and third coordinate for which the flux is required. The subroutine returns a vector of fluxes of the same size as the vector of input energies.

The flux maps can cover regions where no measurements were available when building the map. The corresponding value in these bins has to be set to -1. The interpolation program `MODINT` uses the value -1 as a flag: if a value of -1 is encountered during interpolation, the procedure is stopped and a flux value of -1 is returned. In turn, `TREP` interprets a flux value of -1 as a sign that the geographic point in question is outside the region covered by the model map, and will issue a warning. The same warning is issued when the geographic point or the energy is completely outside of the model map.

MODINT Can be called directly by TREP, or by the routine that converts unidirectional to omnidirectional fluxes.

2.1.8 Conversion to omnidirectional fluxes

The omnidirectional flux J at a given location P and energy E can be expressed in terms of the unidirectional flux j as:

$$J(E) = \int_0^{2\pi} \int_0^{\pi} j[E, L(\alpha), \alpha, \phi] \, \sin\alpha \, d\alpha \, d\phi \tag{2.7}$$

(Roederer, 1970). Using the relation between the local pitch angle α, the local magnetic field strength B, and the equatorial pitch angle α_0:

$$\frac{\sin^2\alpha}{B} = \frac{\sin^2\alpha_0}{B_0} \,, \tag{2.8}$$

and assuming that the dependence of the flux on α is symmetric around $\pi/2$, Eq. (2.7) can be rewritten as:

$$J(E) = 2\,\frac{B}{B_0} \int_0^{2\pi} \int_{\mu_0}^{1} \frac{j_0[E, L(\alpha_0), \alpha_0, \phi] \, \cos\alpha_0}{\sqrt{1 - \dfrac{B}{B_0}\,(1 - \cos^2\alpha_0)}} \, d\cos\alpha_0 \, d\phi \,, \tag{2.9}$$

with

$$\mu_0 = \sqrt{1 - \frac{B_0}{B}} \,. \tag{2.10}$$

The subroutine UNIOMN, contained in MODINT.FOR, uses Eq. (2.9) to derive omnidirectional fluxes from a unidirectional flux map parametrised in terms of E, L, and α_0, and which does not depend on ϕ (the azimuthal dependence of the flux will be implemented later). The integration is carried out using the trapezoid method, so that

$$\begin{aligned} J(E) = 2\pi\,\frac{B}{B_0} \sum_{i=1}^{n} & \left\{ j_{0,i} \left[1 - \frac{B}{B_0}\,(1 - \cos^2\alpha_{0,i})\right]^{-1/2} + j_{0,i+1} \left[1 - \frac{B}{B_0}\,(1 - \cos^2\alpha_{0,i+1})\right]^{-1/2} \right\} \\ & \times (\cos\alpha_{0,i+1} - \cos\alpha_{0,i}) \\ & + 2\pi\,\frac{B}{B_0} \left\{ j_{0,n} \left[1 - \frac{B}{B_0}\,(1 - \cos^2\alpha_{0,n})\right]^{-1/2} + j_{0,n+1} \left[1 - \frac{B}{B_0}\,(1 - \cos^2\alpha_0)\right]^{-1/2} \right\} \\ & \times (\cos\alpha_0 - \cos\alpha_{0,n}) \,, \end{aligned} \tag{2.11}$$

where $j_{0,i}$ represents the flux map entry $j_0[E, L(\alpha_{0i}), \alpha_{0i}]$, and n is the index of the α_0 grid value preceding α_0.

The integration in Eq. (2.9) can be rewritten in terms of B or B_0 to accomodate the flux maps organised in these coordinates. This extension to UNIOMN will be added later.

Table 2.5. New or modified NAMELIST parameters for TREP

Parameter	Data Type	Default	Function
TRPMOD	INTEGER	1	Proton model selection: 1: AP-8 2: PAB97 3: PSB97 4: PUB97 5: PCP94
TREMOD	INTEGER	1	Electron model selection: 1: AE-8 2: ECS96 3: LANL 4: ECM97 5: EIM97 6: ECP95
IMODP	INTEGER	1	Index of proton submodel
IMODE	INTEGER	1	Index of electron submodel
OMNIP	INTEGER	1	Flag for conversion to omnidirectional proton fluxes
OMNIE	INTEGER	1	Flag for conversion to omnidirectional electron fluxes

2.1.9 Modifications to TREP

In order to add the new flux models, the TREP program had to be modified in several ways:

1. addition of models in the same format as the NASA models AP-8 and AE-8;
2. addition of models in a different format, including the development of new interpolation routines;
3. addition of a routine for converting unidirectional fluxes to omnidirectional fluxes;
4. modification and extension of the TREP NAMELIST parameters to accomodate the preceding modifications.

The addition of new models has been described in Sects. 2.1.6 and 2.1.7, and flux conversion in Sect. 2.1.8, respectively. The NAMELIST parameters added or modified are listed in Table 2.5. The new version of TREP has been used to produce the spectra used in Part IV of Technical Note 5 and in Chapter 7 of this Final Report.

2.2 The UNIRAD library

The development of the UNIRAD library was the task of WP 3.2R of the TREND-3 project. The main objectives of the WP are to improve the software used to build radiation belt models and to predict the radiation experienced by spacecraft in orbit around the Earth. To this effect, other coordinates than (B, L) have been investigated to organise trapped particle fluxes, especially at low altitude. The UNIRAD library provides tools to evaluate these new coordinates, which are related to interactions of particles with the environment, such as the atmospheric constituents encountered or the energy loss experienced by the trapped particles. The software library therefore implements tools to compute the geomagnetic field, to trace magnetic field lines and drift shells, to determine the minimum altitude of mirror points and to evaluate physical quantities averaged along a drift trajectory.

2.2.1 Portability

The library has been developed on a DEC/AXP platform under the operating system OpenVMS. The modules of the library have been written conformly to standard FORTRAN 77, extended by the use of STRUCTURE and RECORD statements. These statements greatly increase the user friendliness of the library and are generally supported by all FORTRAN 77 compilers.

Machine specific code has been avoided so that the library can be ported on VAX, AXP, HP, Sun, and PC platforms, under different operating systems. Under the operating systems VAX/VMS and OpenVMS/AXP, the library is callable directly from IDL routines, to allow graphical representation and data analysis.

2.2.2 Sources

The UNIRAD library has been partially derived from the UNIRAD software [programme BLXTRA, see Heynderickx et al. (1996d)] and from the code developed by Hassitt (1965) at the University of California (San Diego) provided to BIRA/IASB by C.E. McIlwain. The UNIRAD library subroutines which evaluate the geomagnetic field intensity and McIlwain's (1961) parameter L are those developed earlier at BIRA/IASB for BLXTRA under the first TREND project.

The Hassitt code calculates averages of the atmospheric density over a drift shell. In this code, the Jensen and Cain (1962) magnetic field model and the atmosphere model of Anderson & Francis (1964) are used, respectively, to trace the drift shell and to evaluate the atmospheric number densities. Hassitt (1965) deduced the averaged atmospheric density over a drift shell by weighting the atmospheric number densities with proton cross sections for charge exchange, ionisation and excitation. The Hassitt code has been extended during the course of the TREND-2 project to incorporate additional magnetic field models and the latest atmosphere and ionosphere models (Heynderickx et al. 1995, 1996c). M. Kruglanski, who was responsible for WP 3.2R, has developed a new version of this drift shell program which differs significantly from that of Hassitt. Both codes have been tested for a number of case studies and give comparable results.

2.2.3 Functionality of the library

A main goal of the UNIRAD library is to evaluate the effects of the interaction of trapped protons with the atmosphere. These effects are usually described through effective scale heights. Different scale heights are obtained from the averages of different physical quantities over one complete longitudinal drift period of the trapped particles. The different physical quantities are, for instance:

- the amount of atmospheric material encountered by the particle;
- the number of charge-exchange collisions;
- the number of ionisation or excitation interactions;
- the energy lost by the trapped particle;
- the pitch angle deviation experienced by the trapped particle.

These quantities are generally expressed per unit of time.

Another key goal of the library is to provide subroutines to compute specific coordinates for the geomagnetically trapped particles, for instance:

- the magnetic field intensity B_m at the mirror points;
- the equatorial pitch angle α_0;
- the McIlwain (1961) parameter L;
- the second and third adiabatic invariants J and Φ ;
- the intersections of the magnetic field line with the Earth's surface.

2.2.4 Conventions used in the UNIRAD library

2.2.4.1 Units

The SI units are used in the UNIRAD library except for the quantities listed in Table 2.6. Angles are given in degrees.

Table 2.6. Physical units used in the `UNIRAD` library

Quantity	Unit	Relation to SI
energy	MeV	1.602177×10^{-13} J
date	day	86400 s
dipole geomagnetic moment	Gauss R_{E}^3	2.58621×10^{16} T m^3
distance	km	10^3 m
Earth radius	R_{E}	6,371,200 m
mass	amu	1.66054×10^{-27} kg
mass thickness	g cm^{-2}	10 kg m^{-2}
mass density	g cm^{-3}	10^3 kg m^{-3}
number density	cm^{-3}	10^6 m^{-3}
magnetic flux density	Gauss	10^{-4} T
cross section	mbarn	10^{-31} m^2

2.2.4.2 Coordinate systems

Geographic positions are generally given in the Geocentric Equatorial (GEO) coordinate system in the form of longitude, colatitude and radial distance from the centre of the Earth. The vector attached to a geographic position is given by its spherical components (e.g. V_ρ, V_θ, V_ϕ, where the subscripts ρ, θ and ϕ indicate the radial distance, the colatitude and the longitude in GEO, respectively). The spherical components of a vector are related to its cartesian components by the transformation

$$\begin{bmatrix} V_\rho \\ V_\theta \\ V_\phi \end{bmatrix} = \begin{bmatrix} \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \\ \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} . \tag{2.12}$$

One should note that the GEO spherical components and the Geocentric Inertial (GEI) spherical components of any vector are identical.

The library includes several subroutines to convert GEO coordinates to other coordinate systems, and inversely.

2.2.4.3 Component identifiers

Each component has a unique identifier in the `UNIRAD` library.

The FORTRAN `COMMON BLOCK` components of the library are identified by the two characters `UC` followed by three digits. The FORTRAN subroutine components of the library are identified by two characters, related to the functionality of the component, followed by three digits. The characters `UL`, `UF`, `UD`, `UM`, `UA`, and `UT` correspond to geomagnetic labels, field line tracing, drift shell tracing, magnetic models, atmospheric models, and general tools, respectively. Each component of the library is uniquely defined by its 3-digit code.

Table 2.7. URLs Of the main pages of the UNIRAD library documentation, relative to the root locator http://www.magnet.oma.be/home/unilib/

URL	Content
home.htm	Home page of the library
toc.htm	List of the library components
faq.htm	Frequently asked questions
index.htm	Set of links to other pages ordered by keywords
structure.htm	Description of the FORTRAN structures used by the library

For the variables used in the library, the first character of the identifier indicates the type of the variable. The correspondences between the first character of the identifier and the FORTRAN type of the field are: A–H, P–Y for REAL*8; I, J, K and N for INTEGER*4; L for CHARACTER*(*); M for structure record; and Z for EXTERNAL.

2.2.4.3.1 Error trapping When an error occurs in a subroutine of the library, the error diagnostic is returned by way of a negative integer value of the form −dddii. The first three digits (ddd) are set to the three digits of the name of the subroutine where the error occurs. The last two digits (ii) are used to differentiate between errors inside a same subroutine.

2.2.5 Library documentation

The documentation of the UNIRAD library is provided in the form of HTML pages. The documentation contains a list of Frequently Asked Questions, a list of all the components of the library, and a detailed description of each component. Some examples are provided as well. The Universal Resource Locators (URLs) of the main pages of the documentation are listed in Table 2.7.

2.2.6 Installation and usage

The UNIRAD library contains two items:

1. an object library file (unilib.lib, unilib.olb, or libunilib.a, depending on the operating system);

2. an include file (structure.h).

The object library file is intended to be used by the linker. It includes the object code of all the subroutines contained in the UNIRAD library. The include file contains the definition of all the structures defined in the library. The include file can be called in user source code by the statement `INCLUDE 'STRUCTURE.H'`. Both files can be downloaded from the HTML page `http://www.magnet.oma.be/home/unilib/hreg.htm`.

The functionality of the library has to be accessed with the help of a FORTRAN programme or an IDL routine. Note that except under the operating systems VAX/VMS and OpenVMS/AXP, additional C codes have to be produced to interface the library with the IDL software.

2.2.6.1 Initialization

Before most of the library subroutines can be used, the different `COMMON BLOCK`s of the `UNIRAD` library have to be initialized. The following initalisation subroutines are provided inside the library :

- `UT990`: to initialize the different `COMMON BLOCK`s of the `UNIRAD` library;
- `UM510`: to select a geomagnetic field model;
- `UM520`: to select an external magnetic field model;
- `UA610`: to select an atmospheric, ionospheric and/or plasmaspheric model.

2.2.6.2 Main subroutines

The subroutines of the `UNIRAD` library can be divided into three sets:

1. the main subroutines;
2. the internal subroutines;
3. the miscellaneous subroutines.

The internal subroutines are subroutines called by other subroutines of the library and as such, are not directly accessed by the user. The main subroutines are top-level subroutines, they include:

- `UL220`: to evaluate B_m, L, $K = I\sqrt{B_\mathrm{m}}$ for a set of field line segments passing through a given position where I is the a line integral called the integral invariant function (McIlwain 1961);
- `UD310`: to trace a magnetic drift shell;
- `UD320`: to average physical quantities over a drift shell;
- `UD330`: to evaluate the third invariant;
- `UF420`: to trace a magnetic field line segment passing through a given position;

- UM530: to evaluate the magnetic field vector;
- UA630: to evaluate the atmospheric number and mass densities;
- UA636: to evaluate the atmospheric number or mass densities weighted by cross sections;
- UT980: to print the library error messages.

The miscellaneous subroutines are subroutines called by other subroutines of the library but that users can also use directly (e.g. UM539, UT540, UT541, UT550).

The HTML pages relative to the main subroutines are a good starting point to learn about the UNIRAD library. Another starting point to understand the philosophy of the library are the examples provided in the HTML documentation.

2.2.7 Models included in the library

2.2.7.1 Magnetic field models

The library includes some internal and external magnetic field models. The set of internal magnetic field models is composed of:

1. the International Geomagnetic Reference Field (IGRF) model;
2. the Jensen & Cain (1962) model;
3. the GSFC12/66 model (Cain et al., 1967);
4. a simple centred tilted dipole magnetic field.

The IGRF model is the empirical representation of the Earth's magnetic field recommended for scientific use by the International Association of Geomagnetism and Aeronomy (IAGA). The Jensen & Cain (1962) model and the GSFC12/66 model (Cain et al. 1967) are included in the library since they have been used to produce the NASA trapped radiation belt models AP-8 and AE-8 (Heynderickx et al. 1996ab). The simple dipole magnetic field model is deduced from a truncation to the second order of the expansion of the IGRF geomagnetic field models.

Eight external magnetic field models are included in the library:

1. Mead & Fairfield (1975) model;
2. Tsyganenko (1987) short model;
3. Tsyganenko (1987) long model;
4. Tsyganenko (1989) model;
5. Olson & Pfitzer (1977) quiet model;

6. Olson & Pfitzer (Olson et al. 1988) dynamic model;
7. model T96 (Tsyganenko & Stern, 1996);
8. Ostapenko-Maltsev (1997) model.

The Mead & Fairfield (1975) and Tsyganenko (1987, 1989) magnetic field models depend on levels of magnetic disturbance parameterized by K_{p}. The Tsyganenko (1989) model is tilt dependent and was primarily developed as a tail model. The Olson & Pfitzer (1977) model is an average model fit to quiet conditions without parameter. The dynamic model (Olson et al. 1988) is a scalable model depending upon the activity level but without a tilt dependence. The scale factors are determined by the activity index D_{st} and the standoff distance of the magnetopause determined by the solar wind density and velocity. The model T96 (Tsyganenko & Stern 1996) depends on the solar wind pressure and the B_Z and B_Y components of the interplanetary magnetic field. The Ostapenko-Maltsev (1997) model was obtained by a least-square fit of fourth order polynomials to 14,000 vector field measurements from the data base of Fairfield et al. (1994). The model depends on the D_{st} and K_{p} indices, as well as on the solar wind dynamic pressure and the z component of the interplanetary magnetic field.

The different magnetic field models are accessible through subroutines `UM510`, `UM520` and `UM530`.

2.2.7.2 Atmospheric models

Some atmospheric models as well as ionospheric and plasmaspheric extensions are accessible from the library. The library includes:

1. the neutral atmosphere empirical model MSISE-90 (Hedin 1991), improved by Chabrillat (1995);
2. the Anderson and Francis (1964) atmosphere and ionosphere model;
3. the international reference ionosphere IRI-90 (Bilitza 1990);
4. a plasmaspheric extension of IRI-90 (Carpenter & Anderson 1992);
5. a simple atmospheric model based on a table of Allen (1985);
6. the atmospheric model used by Pfitzer (1990).

The MSISE-90 atmospheric model is a very comprehensive reference model for the upper atmosphere based on measurements from several rockets, satellites and incoherent scatter radars. It is extended to the middle and lower atmosphere on the base of a tabulation of zonal average temperature and pressure. The IRI-90 ionospheric model is a reference model for the ionospheric densities and temperatures. Although outdated and obsolete, the Anderson and Francis (1964) model is included in the library since it had been originally used by Hassitt (1965). The last two models are simplified models which provide only the total atmospheric mass density.

The atmospheric models, the ionospheric model and its plasmaspheric extension are accessible through the use of the subroutines `UA610` and `UA630`.

2.2.7.3 Collisional cross section models

Two models to evaluate the cross sections for collisions between protons and atmospheric particles are included in the library:

1. the Hassitt (1965) cross sections;
2. a collection of cross sections compiled by Pierrard (1994).

The Hassitt model is included for historical reason and includes cross sections from unreferenced origin. Pierrard's (1994) compilation includes cross sections of charge exchange, ionisation, excitation and nuclear collision from several sources. Both models are accessible through the use of the subroutines `UA636` and `UA637`.

2.2.8 Magnetic drift shell

The tracing of magnetic drift shells is the central part of the `UNIRAD` library: when a drift shell is traced with the help of the library, more than twenty subroutines of the library are used. The tracing of is drift shell is mainly controlled by the subroutines `UD310`, `UF410` and `UF420`. The algorithm is separated in two parts, the magnetic field line tracing and the drift shell tracing.

2.2.8.1 Field line tracing

A segment of a magnetic field line is described as a set of elementary segments. Each segments is characterized by its GEO location, arc length, radius of curvature,. . . (see the definition of the FORTRAN structure `/zseg/`). The field line segments are traced by the subroutine `UF420` and its dependencies. The algorithm has been derived from Pfitzer's (1991) one and is still based on Gill's (1951) 4-order Runge-Kutta integration method. The input arguments of the subroutine `UF420` are the GEO coordinates of a point on the magnetic field line, a set of mirror-point magnetic field intensities and, eventually, an altitude. The subroutine traces the segment of the magnetic field line that includes all the pair of mirror points and eventually the points of the magnetic field line that reaches the altitude provided as input. Note that the resulting field segment does not necessarily include the starting point.

When the subroutine `UF420` is called, it checks first that the starting point is located inside the most external boundary conditions (the boundary conditions are defined by the mirror-point magnetic field intensities and the given altitude). If this is not the case, the field line is traced in the direction of decreasing field intensity until a boundary condition is reached. The field line segment is then firstly traced in the increasing field intensity direction until all the boundary conditions have been reached. Afterwards, this first part of the field line segment is reversed and the rest of the field line segment is traced in the opposite direction. Each time when a boundary condition is overstepped, the field line tracing is suspended and the exact location on the field line corresponding to the boundary condition is calculated by an iterative search. Such an iterative search is also applied when an extremum of the magnetic field intensity is overreached.

Note that the integral invariant function I is not evaluated during the field line tracing, but that the intermediate B value of the Runge-Kutta integration step are stored in order to calculate the value of I afterwards (see subroutine `UL230`).

2.2.8.2 Drift shell tracing

Magnetic drift shells are defined by their mirror-point magnetic field intensity B_m and McIlwain shell parameter L: the drift shell (B_m, L) consists of a family of magnetic field line segments

- which are bounded by the magnetic field intensity B_m, and,
- the integral invariant function I which corresponds to the requested L value.

Drift shells can be traced by subroutine `UD310`. The input arguments of the subroutine are B_m, L and the number n of field line segments to be traced. As output the subroutine returns the set of field line segments such that the longitudes of the points with the lowest magnetic field intensity along each field line segment are equidistant. The subroutine `UD310` calls the subroutine `UF410`, n times, to find and trace the different field line segments.

The arguments of the subroutine `UF410` are B_m, L, a longitude and an altitude range. When the subroutine is entered, it seeks first a magnetic field line for which the point P_min with the lowest magnetic field intensity has the requested longitude and lies in the given altitude range. If the magnetic field intensity at P_min is greater than B_m, the lower limit of the altitude range is set to the altitude of the current P_min and the programme seeks for an other magnetic field line.

When a field line which satisfied all conditions on P_min is obtained, the I value of the field line segment (delimited by the magnetic field intensity B_m) is evaluated and compared to the expected value. According to the value obtained for the integral invariant function, the altitude range of the point P_min is adapted:

- when the invariant function is larger than the requested I, the upper limit of the altitude range is set to the altitude of the current P_min;
- when the invariant function is lower than the requested I, the lower limit of the altitude range is set to the altitude of the current P_min.

Afterwards, a new magnetic field line is sought again.

This iterative process is repeated until the integral invariant function is found to be equal to the expected I within a tolerance specified in the common block `UC190`. During the iterations, the programme is aborted when

1. the number of iterations exceeds 20;

Table 2.8. Subroutines implemented in the UNIRAD library

Name	Usage
`UL220`	get information on a magnetic field line segment
`UL230`	evaluate the integral invariant coordinate I
`UL240`	evaluate the Hilton's function
`UL242`	inverse the Hilton function
`UL245`	equatorial pitch angle
`UD310`	trace a magnetic drift shell
`UD315`	search the mirror point with the lowest altitude
`UD316`	search the equatorial point with the lowest magnetic field intensity
`UD319`	transfer a field line segment from common block `UC170` to `UC130`
`UD320`	evaluate a double time integral over a drift shell
`UD321`	evaluate a time integral over a magnetic field line
`UD327`	search an extremum
`UD328`	evaluate the second derivative of B
`UD329`	interpolate between two or three points
`UD330`	evaluate the third invariant
`UD331`	evaluate the magnetic flux through a spherical cap
`UD332`	evaluate the magnetic flux through a spherical pie
`UF410`	search the geographic position of a magnetic field line
`UF411`	search a local magnetic equator
`UF415`	rebuild the labels of the field line
`UF420`	trace a magnetic field line segment passing through a given position
`UF421`	initialize and close a field line segment
`UF422`	follow a field line until a boundary condition is reached
`UF423`	Runge Kutta step
`UF424`	search the lowest B value
`UF425`	evaluate the curvature of the field lines
`UF426`	interpolating an extremum of B
`UF427`	interpolating a value of B
`UF428`	interpolating an altitude
`UF429`	transpose a field line segment
`UM510`	select a geomagnetic field model
`UM511`	set the Jensen & Cain model coefficients
`UM512`	set the GSFC 12/66 model coefficients
`UM513`	set the DGRF/IGRF model coefficients
`UM515`	perform a Schmidt normalisation
`UM517`	transform from Schmidt to Kluge normalisation
`UM520`	select an external magnetic field model
`UM521`	ground disturbances from $\mathrm{K_p}$
`UM522`	position of the Sun

Table 2.8. (continued)

UM523	GEO to GSM transformation
UM524	GEO to SM transformation
UM530	evaluate the magnetic field vector
UM531	geomagnetic field
UM532	external magnetic field
UM533	distance to the magnetopause
UM535	geocentric to geodetic transformation
UM536	geodetic to geocentric transformation
UM537	Kluge evaluation of the geomagnetic field
UM539	evaluate the magnetic field
UT540	compute modified Julian Day from date
UT541	convert spherical coordinates to cartesian coordinates
UT542	convert spherical vector components to cartesian components
UT545	compute date from modified Julian Day
UT546	convert cartesian coordinates to spherical coordinates
UT547	convert cartesian vector components to spherical components
UT550	select a coordinate transformation
UT551	initialize an Euler rotation matrix
UT552	initialize a quaternion rotation matrix
UT555	coordinate conversion
UT556	vector conversion
UA610	select an atmospheric, ionospheric and/or plasmaspheric model
UA612	provide a list of atmospheric constituents
UA630	evaluate the density numbers and mass densities
UA631	Anderson and Francis atmosphere
UA632	MSISE-90 atmosphere
UA633	IRI-90 ionosphere
UA634	compute mass density from number density
UA635	evaluate the Debye length
UA636	evaluate a weighted atmospheric mass
UA637	proton cross sections
UA638	Hassitt proton cross sections
UA639	equatorial electron density
UT980	print the error messages
UT981	search information on a subroutine
UT982	print the link between two subroutines
UT985	transfer a field line of the drift shell to IDL
UT986	pass general variables to IDL
UT990	initialize the UNIRAD library
UT991	print the magnetic field line

Table 2.8. (continued)

UT992	print the magnetic drift shell
UT993	store a magnetic field drift shell
UT998	evaluate the magnetic field vector and the normal to the field line
UT999	radius of curvature in a dipolar magnetic field

2. the upper and lower limits of the altitude range become equal;

3. an unrecoverable error occurs during the field line tracing.

In that case, the subroutine UF410 returns with a specific error diagnostic.

The subroutine UD310 succeeds only when all the magnetic field line segments of the drift shell have been successfully traced.

2.2.9 Content of the library

The complete set of subroutines implemented in the library is listed in Table 2.8. The user documentation for the UNIRAD library has been written in the HyperText Mark-up Language (HTML) and has not been reproduced in this Final Report. The documentation consists of:

- Frequently Asked Questions (FAQs) and their answers, available at:
  http://www.magnet.oma.be/home/unilib/faq.htm

- a set of pages with a detailed description of the subroutines making up the library, one HTML page per subroutine. These HTML pages are made up of the following sections: Name, Synopsis, Arguments, Description, Dependencies, See Also, and Examples. The pages can be accessed directly from a Table of Contents (TOC) page. In addition, links are provided between related pages. The URL of the TOC page is
  http://www.magnet.oma.be/home/unilib/toc.htm.