

Status of Developing a Near Real-Time Capability for Estimating Space Radiation Exposure Using EMMREM

Richard B. Hatcher and Lawrence W. Townsend
The University of Tennessee

Nathan A. Schwadron and Kamen Kozarev
Boston University

Copyright © 2009 SAE International

ABSTRACT

The central objective of the Earth-Moon-Mars Radiation Environment Module (EMMREM) project is to develop and validate a numerical module for completely characterizing time-dependent radiation exposure in the Earth-Moon-Mars and Interplanetary space environments. An important step in the process of building this system is the development of the interfaces between EMMREM's internal components, many of which have existed previously as stand-alone simulation codes. This work specifically discusses the development and implementation of the interface, primarily using the Perl scripting language, between two input data set generators, one of which describes the space radiation environment at some desired location, and a space radiation transport and shielding code, BRYNTRN, that provides estimates at fairly short time intervals of dose and dose equivalent behind shielding.

INTRODUCTION

The EMMREM module uses a combination of state-of-the-art particle radiation models to predict astronaut radiation exposures [1]. This presents a unique challenge in that EMMREM requires the smooth interoperation of several distinct simulation codes written by different authors at different times using a variety of computational methods, programming languages, and design philosophies. EMMREM's component codes will run sequentially, wherein output of one component will often be input of the next. This requires an intermediary

translation step between the executions of each component code.

In order to accomplish our goal of constructing a unified framework from multiple simulation codes for accurately analyzing radiation exposure risks to astronauts we decided to implement the inter-component translation scheme by: 1) designing a simple markup language-style data format for passing results between components, 2) modifying the source code of the component simulation codes only where necessary to facilitate interfacing with one another, and 3) writing highly portable interface code that can read properly reformatted output data from one component, translate it into a form that the next component can utilize, and then execute the successor component code.

EMMREM COMPONENTS

In this paper, we examine those parts of the EMMREM module, which are responsible for simulating the evolution through time of particle spectra from solar particle events (SPEs) and the associated radiation doses absorbed by astronauts. The time-dependent solar radiation model is provided through the Energetic Particle Radiation Environment Module (EPREM), a cutting-edge highly parallelized Monte Carlo code developed at the Boston University Department of Astronomy. Once it has been fed input data describing the SPE to be simulated and the location of the simulated observer with respect to the Sun, the EPREM code is executed. The EPREM code then produces a detailed representation of the changing particle energy

spectra over the course of the entire simulated event. This produces a very large amount of data, which are parsed by a script in the EPREM package, which in turn produces a properly formatted output file for processing by the Perl interface script.

BRYNTRN (BaRYon TRAnsport) is a one-dimensional deterministic code developed to simulate the transport of high-energy baryons through arbitrary shielding material. Using a predefined materials interaction database and a straight ahead approximation of the Boltzmann equation, it calculates the amount of dose deposited by particles for a given radiation spectrum as well as the dose equivalent and the lineal energy transfer (LET) as a function of depth in the shielding material [2].

BRYNTRN is not designed to accept a time series of particle spectra over the lifetime of a simulated radiation event like those produced by EPREM. When running simulations of historically significant SPEs, the unmodified BRYNTRN code uses spectra that have already been integrated over time, yielding dose and dose equivalent for the whole of the event.

The interface script enables users to take EPREM generated particle spectra data and feed them into separate instances of BRYNTRN in the form of data blocks representing spectra that are time-integrated only over the length of a single time step. This permits the generation of dosimetric data in near real time with arbitrary step lengths. Early results of this fusion can be seen in data generated by EMMREM in Figure 1. It contains dose and dose rate from an EMMREM run based on the historically powerful August 4, 1972 event over the course of half a day with a relatively large time step of one quarter of a day [3]. The code runs very quickly, however, and time steps on the order of minutes are possible for Solar Energetic Particle spectra.

DATA TRANSFER FORMAT

The first step we took in unifying the EPREM and BRYNTRN codes into a cooperative whole was to define the data transfer format described previously. Table 1 contains an example data block representing output from EPREM that has been formatted as input to the interface script, which passes the data encoded within to BRYNTRN:

Table 1. An example data block in the format of those generated by EPREM.

```
<header>
<format>
time[fractionalJulianDay]
dwelltime[fractionalDay]
data[**Emin(MeV)**Emax(MeV)**J
(cm-2s-1sr-1(MeV_nuc)-1)**]
```

```
<Eformat>
<mass>
1
<Emass>
<charge>
1
<Echarge>
<ne>
5
<Ene>
<time>
2452940.150463
<Etime>
<dwelltime>
0.173612
<Edwelltime>
<Eheader>
<data>
4.0 2.264216e-12
15.0 2.837881e-08
50.0 4.491620e-07
100.0 1.545519e-06
140.0 3.831203e-06
<Edata>
```

The entries in the script input data file are identified by their leading and trailing markup tags, and physical units are defined at the top of the file. Each entry is opened with a tag bearing the name of the data entry and is terminated with a similarly named E-tag. The data tags must appear in this order and format in order to be properly read by the interface script. Note: this is not valid HTML or XML markup, although the style of the format is influenced by them.

Each data block's descriptive header data is opened with a <header> tag and is closed with a matching <Eheader> tag. Within the header, the <format> tags bracket the definition of the data block's physical units. The <mass> tags define the atomic mass number A of the incoming baryons, and likewise the <charge> tags define the baryon electric charge number. The <ne> tags define the number of grid points used to define the differential flux curve used by BRYNTRN to calculate the relevant dosimetric quantities.