LA-UR- 00-3581

c.1

*Approved for public release; distribution is unlimited.*

*Title:* ELECTRON UPGRADE FOR MCNP4B

*Author(s):* Kenneth J. Adams

*Submitted to:* For Lab Distribution on World Wide Web

LOS ALAMOS NATIONAL LABORATORY
3 9338 00791 1141

**Los Alamos**
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (10/96)

**Los Alamos**
NATIONAL LABORATORY
**memorandum**

**Applied Physics Division**
*Diagnostics Applications Group, X-5*
PHONE: (505)667-8747— FAX:(505)665-3046

*To/MS:* Distribution
*From/MS:* Kenneth J. Adams/F663
*Phone/Email:* 5-7000/adamsk
*Symbol:* X-5-RN(U)-00-14
*Date:* 5/25/00

**SUBJECT:** ELECTRON UPGRADE FOR MCNP4B

**Abstract:**

The focus of this effort was the integration into MCNP*(ref. 1) of the ITS3.0(ref. 2) radiative and collisional stopping power (refs. 2 and 3) and bremsstrahlung production model (refs. 2, 4, 17, and 18). The integration was achieved with the development of a patch to MCNP4B and a new electron physics library database, el3w (like el1). SQA of the patch and database are still ongoing. However, improvements in the stopping power model and bremsstrahlung model are presented. Moreover, in the process of successfully achieving this goal several other related (bremsstrahlung angular distribution models) or otherwise unforeseen problems (electron step-size artifacts or thick target bremsstrahlung production problems) were identified and unfortunately not solved yet since they involve much more work. Several bugs have been found and eradicated.

**Overview:**

The memo is organized as follow:

0) Background

1) el3w database. the new database library

2) code modifications for new algorithms to use the new database and other enhancements

2a) radiative stopping power

2b) bremsstrahlung production

2c) density effect

2d) relaxation

2e) variance reduction

---

*MCNP is a trademark of the Regents of the University of California, Los Alamos National Laboratory

2f) bug fixes

3) verification

3a) ITS stopping powers

3b) ITS cross section comparison

3c)Implications

4) preliminary validation based upon Gierga and Adams

4a) comparison with Faddegon for bremsstrahlung production

4b) depth dose in water

5) future work

## Background:

In order to perform radiography analysis, accurate transport models are required to define the bremsstrahlung source and contamination in the object. Several authors (refs. 7 - 10, this is list is not meant to be exhaustive nor definitive) have examined the accuracy of various bremsstrahlung models in transport codes (MCNP[1], ITS[2] and EGS[5]). From these studies, possible deficiencies in MCNP were identified and sought to be corrected. As an example of the results found, figure 1

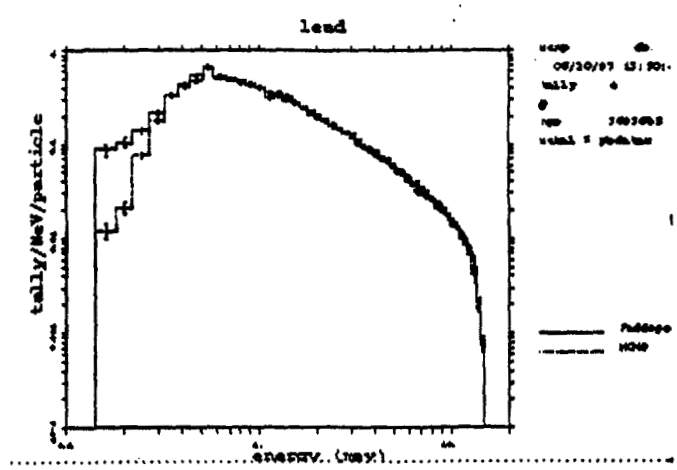



Figure 1: Bremsstrahlung Spectra for 15 MeV electrons on Lead.
Calculation (MCNP) and experiment (Faddegon)

displays the bremsstrahlung spectrum for 15 MeV electrons on lead calculated with MCNP4B and measured by Faddegon where spectral variations at low energies are present. Moreover, the electron physics package in MCNP is based mostly on ITS1.0 (ref. 6) physics with an upgrade to ITS3.0 physics for the calculation of energy loss straggling and calculation of the ionization potential. A more complete upgrade would be desired to include more of the ITS3.0 package and/ or the database from ref. 11..

## el3w database

The previous electron evaluation database was stored in library file, el1. Essentially, this file was a partial rewriting of the ITS1.0 xdata file into a format suitable for reading by MCNP (ACE format.) The initial few words of each data block for a particular Z were peculiar to MCNP but otherwise it just reformatted the data to be read by subroutine getxst for use in the subroutine xsgen with appropriate offsets. As far as I was able to determine there was no processing of the data only rewriting.

Following this philosophy, a code was written to extract and reformat the desired portions of xdata from ITS3.0 and generate the new library file el3w. The code for doing this reprocessing is presented in Appendix A and is available on cfs. The code represents an evolution of what the database needed from its progenitor and could still further evolve.

The following tables are the list of data pointers used in el3w and el1 and some explanation of their use. Table 1 lists the nxs array entries. These are mostly lengths of blocks of data. However, nxs(16) now has special meaning and is a flag for using the new its3.0 data and algorithms. Table 2 lists the jxs array. Finally, the crux of the puzzle is the contents of the exs array which is presented in Table 3.

**Table 1: nxs array**

| nxs | exposition el3w | el1 |
|---|---|---|
| 1 | length of xss data set | length of xss data set |
| 2 | z | z |
| 3 | number of radiation stopping power interpolation points | number of radiation stopping power interpolation points |
| 4 | number of mott scattering cross section corrections | number of mott scattering cross section corrections |
| 5 | number of electron energy points for bremsstrahlung interpolation | number of bremsstrahlung interpolation points |
| 6 | number of photon ratio point for bremsstrahlung interpolation | number of bremsstrahlung interpolation points |
| 7 | *unused* | *unused* |

**Table 1: nxs array**

| nxs | exposition el3w | el1 |
|---|---|---|
| 8 | *unused* | *unused* |
| 9 | number of interpolation points in the bremsstrahlung database for bremsstrahlung spectrum calculation | *unused* |
| 10 | number of interpolation points in the bremsstrahlung database for angular/energy calculation | *unused* |
| 11 | number of oscillator points for density effect calculation | *unused* |
| 12 | *unused* | *unused* |
| 13 | *unused* | *unused* |
| 14 | *unused* | *unused* |
| 15 | *unused* | *unused* |
| 16 | 3 | *unused* |

**Table 2: jxs array exposition**

| jxs | el3w | el1 |
|---|---|---|
| 1 | 1 (offset for line data) | *same* |
| 2 | offset for radiative stopping power numbers | *same* |
| 3 | offset for Mott cross section corrections | *same* |
| 4 | offset for Riley cross section evaluation | *same* |
| 5 | offset for ITS3.0 bremsstrahlung production database | offset for ITS1.0 bremsstrahlung production database |
| 6 | *unused* | bremsstrahlung production database |
| 7 | offset for internally calculated Riley cross sections | *same* |
| 8 | offset for internally calculated functions of Z | *same* |
| 9 | offset for photon energy ratios | *unused* |

**Table 2: jxs array exposition**

| jxs | el3w | el1 |
|---|---|---|
| 10 | offset for photon energy ratio for angular distribution | *unused* |
| 11 | offset for oscillator descriptions for density effect calculation | *unused* |

**Table 3: exs array exposition**

| exs | el3w | el1 |
|---|---|---|
| jxs(1) | K edge below which no electron induced relaxation will occur (renamed for persistence edg) | K edge below which no electron induced relaxation will occur (renamed for persistence edg) |
| jxs(1)+1 | Auger electron emission energy (renamed for persistence eek) $=\bar{E}_K-2\bar{E}_L$ | K-x-ray or Auger electron emission energy (renamed for persistence eek) $=\bar{E}_K$ |
| jxs(1)+2 | *unused* | transition point for bremsstrahlung extrapolation model |
| jxs(1)+3 | *unused* | transition point for bremsstrahlung extrapolation model |
| jxs(2) | energy points for radiation stopping power interpolation | energy points for radiation stopping power interpolation |
| jxs(2)+nxs(3) | corresponding normalized stopping powers | corresponding normalized stopping powers |
| jxs(2)+2*nxs(3) | corresponding electron-electron bremsstrahlung correction ($\eta$) | *unused* |
| jxs(3) | energy of Mott scattering correction points | |
| jxs(3)+nxs(4) | $h(\theta)$ from eq. (A35) of ref 1 at $\theta=0$ | |
| jxs(3)+2*nxs(4) | $h(\theta)$ from eq. (A35) of ref 1 at $\theta=\pi/4$ | |
| jxs(3)+3*nxs(4) | $h(\theta)$ from eq. (A35) of ref 1 at $\theta=\pi/2$ | |
| jxs(3)+4*nxs(4) | $h(\theta)$ from eq. (A35) of ref 1 at $\theta=3\pi/4$ | |

**Table 3: exs array exposition**

| exs | el3w | el1 |
|---|---|---|
| jxs(3)+5*nxs(4) | h(θ) from eq. (A35) of ref 1 at θ=π | |
| jxs(4) | energy of riley set | |
| jxs(4)+1 to jxs(4)+14 | riley scattering cross section parameters at that energy | |
| repeated 8 more times | | |
| jxs(5) | electron energy values for interpolation for bremsstrahlung cross section nxs(5) values | energy values for bremsstrahlung interpolation |
| jxs(5)+nxs(5) | photon energy ratio values for interpolation for bremsstrahlung cross section nxs(6) values | cross section values for interpolation |
| jxs(5)+nxs(5)+nxs(6) | values for interpolation for bremsstrahlung cross section nxs(5)*nxs(6) values | *unused* |
| jxs(6) | *unused* | energy values for bremsstrahlung interpolation<br>cross section values for interpolation |
| jxs(7) | used for riley cross section evaluation (2160 float words long)[a] | |
| jxs(8) | various powers and functions of Z and A (10 floats long, though only three are used as below)[a] | |
| jxs(8) | $Z^{1/3}$ | |
| jxs(8)+1 | log(Z) | |
| jxs(8)+2 | zf*(1./(1.+zf)+.20206-zf*(.0369-zf*(.0083-zf*.002))) where zf=(Z*α)**2 | |
| jxs(9) | ratio values of photon energy over electron energy for bremsstrahlung spectrum calculation (renamed for persistence rkt) | |
| jxs(10) | ratio values of photon energy over electron energy for bremsstrahlung angular distribution calculation (renamed for persistence rka) | |

**Table 3: exs array exposition**

| exs | el3w | el1 |
|---|---|---|
| jxs(11) | number of electrons in each level (occupation number) (< 0 signifying a conductor) used as $f_i = n_i/Z$. | |
| jxs(11)+nxs(11) | oscillator strength or binding energy of the level, $v_i$. | |

a. This explains the 2170 (2160+10) in getxst for the electron offset.

Unfortunarely, the electron side of MCNP has its own peculiar way of dealing with its cross section data and storing it. That is for neutrals all the transport data is stored in the xss arrays and very few arrays are "hardwired" to reflect the physics manifest in the evaluation. (An example may be the rtc array which is hardwired to store only ten different cross section data.) Moreover, the xss data persists after it has undergone the ordeal of expunging; the exs data is overwritten since the relevant physics data has been processed and then stored in various arrays. As such, we have found that even though the electron portion of MCNP seems like an ideal candidate to become a "module", it has intersperced its memory management throughout the code.

The problem is simple, the physics data is stored in the database as a reduced set of data over a range where interpolation is valid while it is used in transport on a much finer or at least different scale. The original code had the size of the expanded or internally calculated arrays parameterized and the dynamic allocation of memory was done based on that those parameters. For the new database, the parameterization has been retained but in such a way as to allow for regression compatibility. It is expected that this way of dealing with data structures will be amended in the future, so that the code can readily accept new data evaluations without being constrained to parameterizations based on different evaluations.

The parameters used in the electron databases are listed and explained in the following table.

| parameter | exposition on parameter | el3w | el1 |
|---|---|---|---|
| mtop* | number of ratio values of photon energy to electron energy for brems production | 89 | 49 |
| mwng* | number of ratio values of photon energy to electron energy for brems detailed brems angular distribution | (mtop+1)/2 actually 34 is used; to be replaced by nxs(10) | (mtop+1)/2 actually 25 |
| mstp* | coarsening factor for electron energy grid for detailed brems angular distribution | 4 | 8 |
| mpng | number of photon angular bins | 21 | 21 |
| mbng | number of possible photon to electron ratio values from the its3.0 | 51 | *unused* |

* These parameters have corresponding variables, ntop, nwng, nstp respectively, to toggle between databases.

## Code and Algorithm Modifications

The patch to achieve these changes is presented in Appendix B.

2a) Radiative stopping power.

The ITS3.0 database has a more recent evaluation of the radiative stopping power. To use the evaluation in the code required a few small changes since the effect of electron bremsstrahlung has been included in the database. The stopping power is

$$\frac{dE}{dx} = (Z+\eta)\Phi_{rad}$$

The database originally contained a table of energy and $\Phi_{rad}$ with the assumed value of unity for the electron bremsstrahlung contribution, $\eta$. The new evaluation includes a newer table of energy and $\Phi_{rad}$ as well as $\eta$.

2b) Bremsstrahlung production

The brems production database has been extended to cover a larger range of photon energies. The photon energy from a bremsstrahlung event is specified in terms of a ratio of the incident electron; the range of ratio values is from one to $10^{-6}$ for the el3w database while the el1 range was from 0.9999 to $10^{-3}$. The following table gives the ratio values for both databases.

el1:

rkt=.001,.00125,.0015,.00175,.002,.0025,.003,.0035,.004,
.0045,.005,.0055,.006,.007,.008,.009,.01,.0125,.015,.0175,.02,
.025,.03,.035,.04,.045,.05,.055,.06,.07,.08,.09,.1,.125,.15,.175,
.2,.25,.3,.35,.4,.45,.5,.55,.6,.7,.8,.9,.9999

el3w:

rkt=0.000001,0.0000015,0.000002,0.000003,0.000004,0.000005,
0.000006,0.000008,0.00001,0.000015,0.00002,0.00003,0.00004,
0.00005,0.00006,0.00008,0.0001,0.00015,0.0002,0.0003,0.0004,
0.0005,0.0006,0.0008,0.001,0.00125,0.0015,0.00175,0.002,0.0025,
0.003,0.0035,0.004,0.0045,0.005,0.0055,0.006,0.007,0.008,
0.009,0.01,0.0125,0.015,0.0175,0.02,0.025,0.03,0.035,0.04,
0.045,0.05,0.055,0.06,0.07,0.08,0.09,0.1,0.125,0.15,0.175,0.2,
0.225,0.25,0.275,0.3,0.325,0.35,0.375,0.4,0.425,0.45,0.475,
0.5,0.525,0.55,0.575,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,0.97,
0.99,0.995,0.9995,1.0

The production of bremsstrahlung from an electron of energy, E, is determined by numerical or analytic integration of the cross section over the above specified ranges of photon

energy ratios. Chosing the best cross section for all energies is non-trivial and has evolved over the years*. For the el1 database, most of the production cross sections are derived from Koch and Motz (ref. 15 and refered to as K&M hereafter) with some exceptions near the "tip" (the "tip" is the photon phase space near where the photon is extremely forward peaked and has nearly all the incident electron energy.) Berger and Seltzer (ref. 16) describe their best evaluation of which cross sections should be used over various energy ranges. The el1 database follows that evaluation except for not using the Schiff formula (see ref. 15 for formula).

In the el3w evaluation or database, the production cross section for bremsstrahlung photons and energy spectra are from the evaluation by Seltzer and Berger (refs. 2, 17, 18 and refered to as S&B collectively hereafter). We summarize the salient features of the evaluation below; more details can be found in the evaluators' documentation. The evaluation uses detailed calculations of the electron-nucleus bremsstrahlung cross section for electrons with energies below 2 MeV and above 50 MeV. The evaluation below 2 MeV uses the results of Pratt, Tseng, and collaborators, based on numerical phase-shift calculations (refs. 22, 19, and 20). For 50 MeV and above, the analytical theory of Davies, Bethe, Maximon, and Olsen, (DBMO ref. 23) is used and is supplemented by the Elwert Coulomb (ref. 26) correction factor and the theory of the high-frequency limit or tip region given by Jabbur and Pratt (ref. 27). Screening effects are accounted for by the use of Hartree-Fock atomic form factors (ref. 25). The values between these firmly grounded theoretical limits are found by a cubic-spline interpolation as described in refs. 64 and S2. Seltzer reports good agreement between interpolated values and those calculated by Tseng and Pratt (ref. 21) for 5 and 10 MeV electrons in aluminum and uranium. Electron-electron bremsstrahlung is also included in the cross section evaluation based on the theory of Haug (ref. 24) with screening corrections derived from Hartree-Fock incoherent scattering factors (ref. 25). The energy spectra for the bremsstrahlung photons are provided in the evaluation. Finally, the comparisons with experiments presented in S&B indicate the validity of their approach**.

In summary, then, the calculation of the production cross section for bremsstrahlung is given by numerical integration of various formula which reflect the best evaluation at the time of the bremsstrahlung process. For el1, this integration is done in the code and most of the formula are related back to the review of K&M, while for el3w, the integration and any corrections are done in the evaluation. Consequently, all of subroutine brem had to be rewritten for the evaluation of the bremsstrahlung cross section, pbr, spectra, eba for el3w. The new algorithms for the el3w database are derived from the ITS code which presumably reflect those authors' recommended interpolation schemes and have been incorporated in brem since the pbr and eba arrays are used for the calculation of thick target brems.

---

*We quote from Seltzer and Berger, "Many theories of the bremsstrahlung process have been developed, each with its own approximations, limitations and regios of applicability. To obtain accurate cross sections over a wide range of conditions, it is necessary to combine, and to interpolate between, the results of several theories."

**The validation problems discussed later will eventually be augmented to include this set.

The detailed angular distribution calculation has been modified only in so far as the resolution of the energy grids has been more refined. For el1, the angular distribution was calculated for electrons with energies $E_n = \frac{E_{max}}{2^n}$ with n being a positive integer. (Recall that electron transport parameters are calculated on an energy grid with $E_n = \frac{E_{max}}{2^{n/8}}$.) For el3w, $E_n = \frac{E_{max}}{2^{n/2}}$. Moreover, for el1 the photon ratios used every other ratio value in the production evaluation. For el3w, the ratios are read from the database and are fewer than every other. (Presumably every other could be used for el3w or even every one. However, this is what ITS does.) The angular distribution is determined by numerical integration of several formulas from K&M using these electron energy values and photon energy ratios. Consequently, the modifications to subroutine brang have been minimal. As a passing note, it is not apparent that 2BS is used in the angular distribution calculation in brang though one may have expected it to be used from ref. 16. It might be speculated that this may account for some of the differences seem between EGS and MCNP.

In MCNP, brems production is sampled along an electron substep. The probability of an event(s) occurring in a distance δ is $1 - e^{-\Sigma_{brem}\delta}$, with the number of events being sampled from a Poisson distribution. If the probability of an event occurring is small, then only the first non-zero term in the Taylor expansion of the probability is kept and used to sample if one and only one event has occurred. This is the procedure used in MCNP for el1. For el3w, the sampling has been modified to sample from the Poisson distribution so that zero, one or more events could occur, though two or more are extremely infrequent. The electron's energy is decremented after each event and used to scale the energy of the photon from the next event. This procedure is consistent with the method employed by ITS3.0.

2c) Density Effect Correction

The stopping power has a correction term to account for the polarization of the media through which the charged particle passes. Sternheimer, Berger and Seltzer (ref. 14 and refered to as SBS) have given a method to calculate the density correction in ref. sbs in terms of the plasma frequency of the media, the oscillator strengths and occupation numbers given by Carlsson (ref. 28), and the ionization potential of the media. The method finds an adjustment factor, ρ, which

gives the ionization potential for the media at the specified density from their eq. (10) where I is

$$\ln I = \sum_{i=1}^{n-1} f_i \ln\left[(h\nu_i\rho) + \frac{2}{3} f_i (h\nu_p)^2\right]^{1/2} + f_n \ln(h\nu_p f_n^{1/2})$$

the ionzation potential of the medium (consistent with the prescriptions of ref. 10), h is Planck's constant, $f_i$ is the fraction of electrons at that level, $\nu_i$ is the oscillator frequency of that level, $\nu_p$ is the plasma frequency of the medium ($\nu_p = \left(\frac{ne^2}{\pi m_e}\right)^{1/2}$ and n is the number desnity of electrons in the medium). The adjustment factor is found in subroutine den1 using a newton search method; for high density materials the newton search method may fail and the code will issue a warning and nullify the density effect correction; this is consistent with the fundamental theory of the density effect. Once the parameter has been found, it is used in subroutine den2 to calculate the density effect correction as specified by Eq. (5) in sbs.

$$\delta(\beta) = \sum_{i=1}^{n} f_i \ln\left[\frac{l_i^2 + l^2}{l_i^2}\right] - l^2(1-\beta^2)$$

where $l_i^2 = \left(\frac{2}{3} f_i + \left\{\frac{\nu_i \rho}{\nu_p}\right\}^2\right)$ and for conductors the nth level has, $l_n^2 = f_n$ and the parameter, l, is the solution of eq. (6) of sbs.

$$\frac{1}{\beta^2} - 1 = \sum_{i=1}^{n} \frac{f_i}{l^2 + \{(\nu_i\rho)/\nu_p\}^2}$$

Previously, MCNP used the published empirical fit formula of Sternheimer and Peierls (ref. 15). The method described in sbs is an improvement upon the sp fit formula as discussed in sbs.

For the sbs treatment, the conduction state of the material is needed which is specified with cond=0 for a non-conductor, and cond=1 for a conductor; any non-conducting isotopes in a conductor are handled as non-conductors. The user can specify this on the m card with the keyword cond. The code will provide defaults which the user can override for the specification of a non-conductor with cond=-1 on the m card. The scheme is:

initially - cond=0 (non-conductor)

in ronge -

check if all isotopes are conductors and set cond=1 if cond=0

check that there is at least one conducting isotope is present if cond=1, if not set cond=0

if cond=-1 override any prior decision on the conduction state and make non-conductor cond=0.

The density effect calculation has a problem for materials of a single element or Z but speicified with multiple isotopes or ZAIDs. This is traced to the implementation of Eq. (10) of sbs and the ionzation sum rule of ICRU 37 (ref. 10). ITS has changed that implementation slightly. Making it consistent with the above rules and such that the same answer is obtained regardless of isotopic specification worsens the verification discussed later. We note this as a problem.

2d) Relaxation

Relaxation has been a problem in MCNP. There are two algorithmic branches for an atom to relax after undergoing an ionizing event depending upon the source of the ionizing event. If the source of the ionization was a photon the atom's relaxation will be described using the model of Carter and Cashwell (ref. 29) where the major K and L line photons may be emitted or an Auger electron with energy exs(jxs(1)) which has been stored in eek for that material. If the source of ionization was an electron then if the atom fluoresces the photon will have only an average K energy given by eek; Auger transitions are treated the same. Under many circumstances. these approximations would not be too conflicting. However, there is an inconsistency, that fluorescent radiation will be emitted with different lines. To remedy the inconsistency, relaxation from an ionizing event has been combined into a single subroutine, flaug, called by either colidp or kxray as needed. Most of flaug is similar to the relaxation code in colidp except for mostly bookkeeping changes needed to accommodate calls from kxray. The only other physics change needed in flaug for a call from kxray is that the atom is required to relax either through an Auger transition or fluorescence; for a photoionization the atom is not required to Auger or fluor (presumably it relaxes through lower energy transistions neglected in our models.) The arguments of flaug are the energy of the ionizing quanta, the photon energy for a photoionization or max K shell energy, edg*1.0001 (1.0001 prevents any round-off problems), the isotope undergoing the interaction (the highest Z isotope for kxray), the material identifier, and a flag for electron or photon ionization event.

Since, eek is not used any longer as the energy given to either a kxray fluor photon or Auger electron, its value has been lowered to the more probable value (used in the POEM code of

ref. 30 and derived by inspection from ref. 31) of $E_A = E_{\overline{K}} - 2E_{\overline{L}}$ as compared to the previous value of $E_A = E_{\overline{K}}$ and is used only to give the value of Auger transition electron.

This model is still quite crude and approximate though no longer inconsistent. Further improvements will be discussed later.

2e) Variance reduction

The variance reduction discussed in this section pertains to brems production only. There are two vr techniques for brems production:

1) bbrem

Bias the brems photons produced toward higher energies.

2) bnum

Produce bnum more photons.

For el3w, the format of the bbrem card has not been changed to allow for backwards compatibility. The entries on the bbrem card are mtop_el1=49 biasing numbers for the production of photons followed by cell numbers where the biasing will be applied. For el3w, the number of relative photon energy bins is larger so to allow compatibility with the previous bbrem card, the biasing probabilities over the low energy photon bins from the card are linearly interpolated over the low energy bins in el3w.

In MCNP, when bnum was invoked on the phys:e card, bnum identical photons were generated from a brems event with wgt=wgt/bnum. For photon thin problems, such a biasing scheme is not effective and for thick problems, it may be of only marginal help. The more common method (refs. 32 and 2) is to sample the production distribution bnum times using the same reduced weight for each photon. The sampling is done in a do loop bnum times over the full Poisson sampling of events. Note for bnum<1, the sampling of energy loss is done once but photons may or may not be produced based on bnum+rang() being zero or one. The sampling of several different photons with reduced weights has been shown to be more effective in photon thin problems (Gierga and Adams). There is a question as to how to deal with the energy loss from the more numerous and different pseudo events. For bnum>0, the first sampling of the photon(s) energy is used to calculate the energy loss. For bnum<0, the average of all photons produced is used to calculate the energy loss. The first method does not conserve energy, but more properly gives the straggling contribution of radiative energy loss. The second method more nearly

conserves energy and gives the CSDA energy loss more closely at the expense of the straggling. The latter method was the one used by Gierga and Adams in their report.

Finally, for any bnum scheme to be effective the number of secondary electrons generated must be controlled; secondary electrons take up most of the time in transport calculations. Controlling the number of secondary electrons generated is done with the enum entry on the phys:e card. The default has been one regardless of the value on the bnum card. This has been changed to scale the production down by the inverse of the number of photons produced; enum=1/bnum. The user can override this default setting.

Another vr scheme suggested by R. A. Forster is to have a brems occur on each substep. This scheme is done when the tenth entry, numb, on the phys:e card is non-zero. On each substep only one photon is produced with weight equal to the probability of an interaction on that substep. If this was a real interaction, then the energy loss is calculated and applied to the electron. The subsequent photons generated in the Poisson sampling are given the nominal, unreduced weight. In this scheme, enum is arbitrarily set to 1%.

2f) Bug fixes

The primary bug has to do with how the electron (charged particle) algorithms handle materials described with isotopic decomposition. The ionization potential has a factor of 1.13 applied to it when the material is a mixture of different elements. However, a pure material, i.e. composed of one type of atom, may be described with its isotopic decomposition and erroneously have the factor of 1.13 applied to the potential. To correct this each isotope is checked to see if its Z is different, if not the factor of 1.13 is not applied. Clearly, more work needs to be done here, since trace elements may be included which should not effect the ionization potential.

This isotopic decompostion bug also appears in the new density effect calculation. The fit parameter calculated from Eq. (10) of sbs is different for single zaid and composite zaid materials. This is because when Eq. (10) is calculated in den1 it uses a different elemental weighting scheme than used for the calculation of the ionzation potential. Since we obtain excellent agreement with ITS on materials, we note this discrepancy; it is usually an error of 0.1%, though can be larger.

Another related bug is in the determination of the kxray production probability. The atom fraction of the first highest Z isotope is used. If several isotopes of the same Z were present, then the probability is too low. This has been corrected to use the sum of the atom fractions of all the highest Z isotopes

Turning off kxray production was not done when xnum=0 on the phys:e card. The test has been changed from ".gt." to ".ge." in subroutine xsgen. Moreover, there appears to be a special treatment for xnum<0 where the probability, d, of kxray production for an electron is calculated for the electron till it has gone below cutoff. This would be the probability of kxray production over the full electron range. For xnum<0 the probability is scaled by lxnuml/d. This treatment of biasing raises a point of discussion related back to the bnum treatment.

## 3) Verification

There exist no analytic problems against which to verify the correctness of the physics or variance reduction models implemented in the code. However, since the database for stopping powers and bremsstrahlung cross sections has been adapted from an existing code system, ITS3.0, verifying the correctness of the implementation will be obtained by comparing to that code system.

3a) Stopping powers

Stopping powers are calculated and printed in the MCNP print table 85 and in the output from ITS code xgenp. Relative differences in ionizing and radiative stopping powers normalized to the ITS values for Z=1 to 92 over the energy range of 1 keV to 1 GeV are presented

in Fig. 2 (ionizing), Fig. 3 (density effect), Fig. 4(radiative). Generally, the agreement is to four

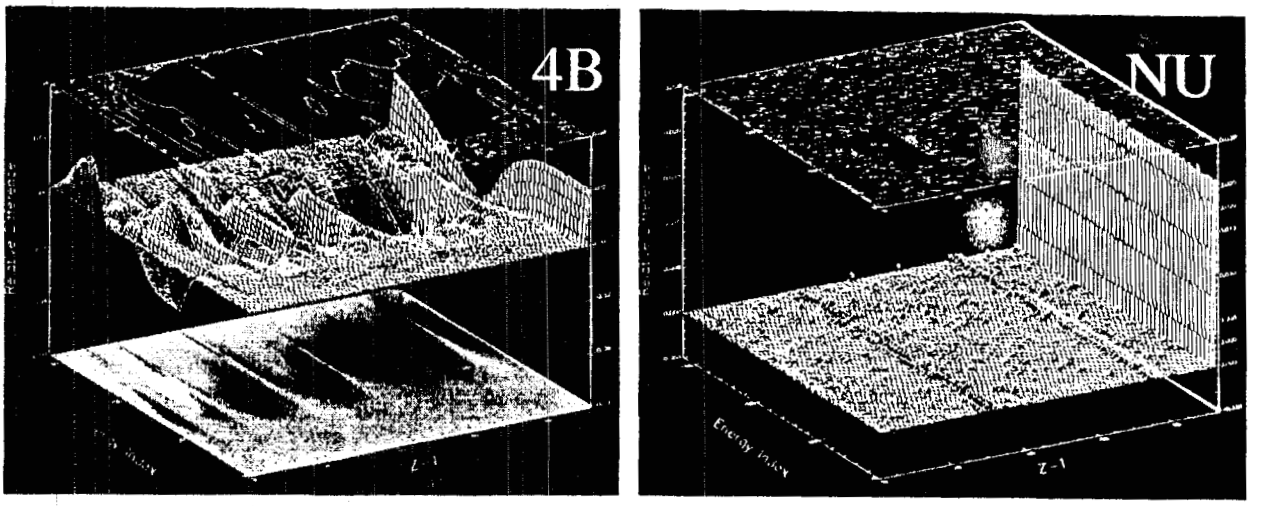



Figure 2: Relative differences between ionizing stopping powers from ITS3.0 and MCNP4B (4B) and ITS3.0 and MCNP4BNU (NU) all normalized to ITS3.0 for Z=1 to 92 from 1 keV to 1 GeV.

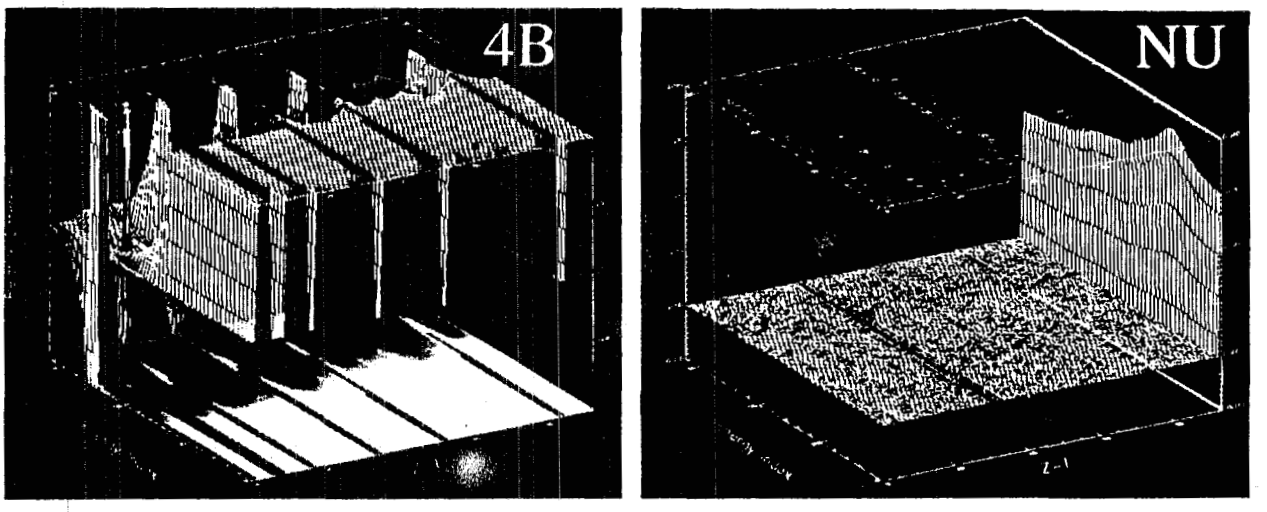



Figure 3: Relative differences between density effect corrections from ITS3.0 and MCNP4B (4B) and ITS3.0 and MCNP4BNU (NU) all normalized to ITS3.0 for Z=1 to 92 from 1 keV to 1 GeV.

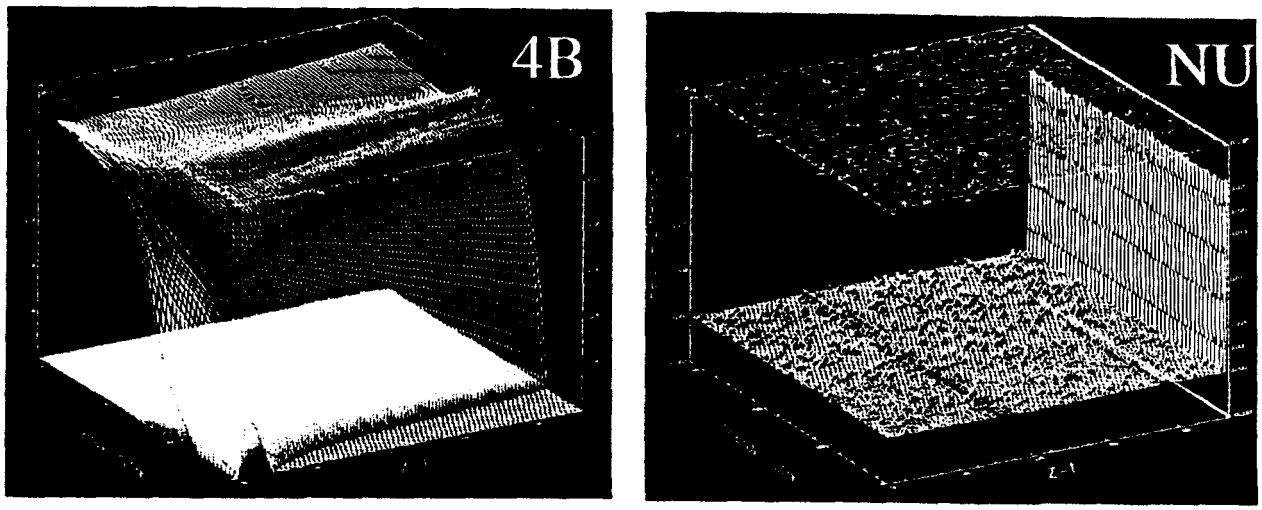



Figure 4: Relative differences between radiative stopping powers from ITS3.0 and MCNP4B (4B) and ITS3.0 and MCNP4BNU (NU) all normalized to ITS3.0 for Z=1 to 92 from 1 keV to 1 GeV.

significant figures and this level of agreement is satisfactory since the fundamental constants and atomic weights in ITS are slightly different from those in MCNP. The only problem material is Pu where the ITS atomic weight is significantly different than MCNP (and most standard tables.) Changing the atomic weight and modifying the constants in MCNP cause the stopping powers to be in agreement to four significant figures.

The stopping powers for materials was also verified. Using the materials of ICRU 37 and a few extra to span composites of a wide range of Z's, the same comparisons were done.

Figure 5 shows the comparison of the collisional or ionizing stopping powers. Figure 6 shows the

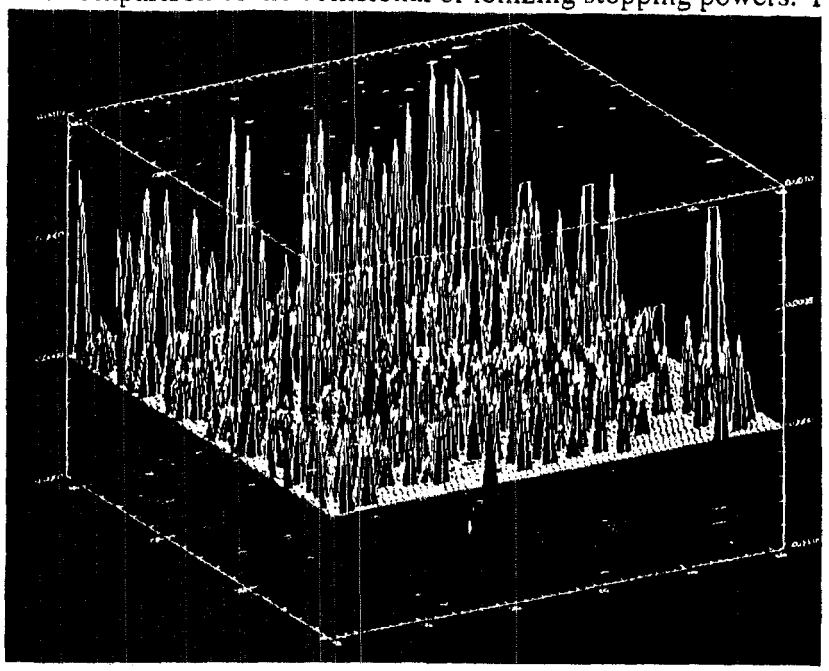


Figure 5: Relative differences between ionizing stopping powers from ITS3.0 and MCNP4BNU all normalized to ITS3.0 for a wide range of materials from 1 keV to 1 GeV.

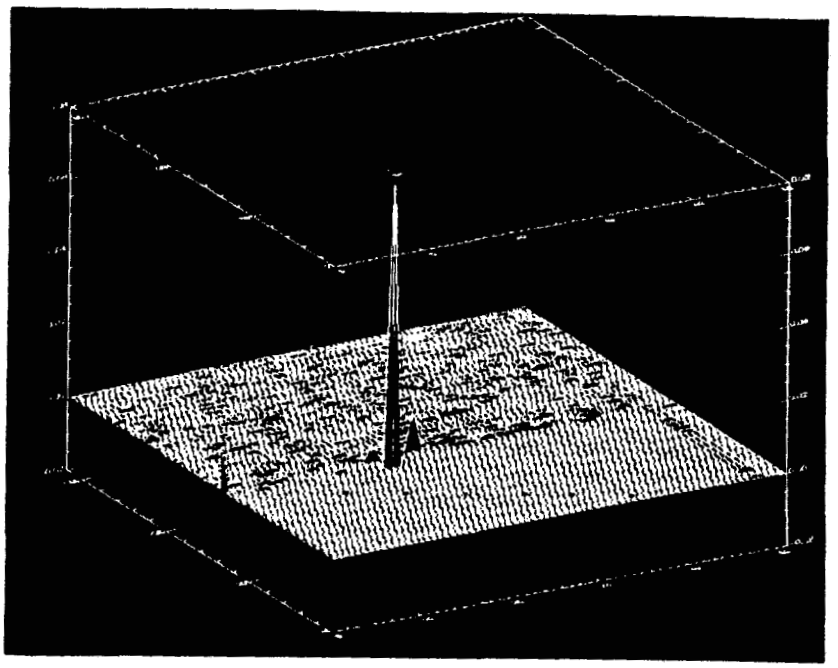


Figure 6: Relative differences between density effect from ITS3.0 and MCNP4BNU all normalized to ITS3.0 for a wide range of materials from 1 keV to 1 GeV.

density effect correction comparison and Figure 7 shows the radiative stopping power

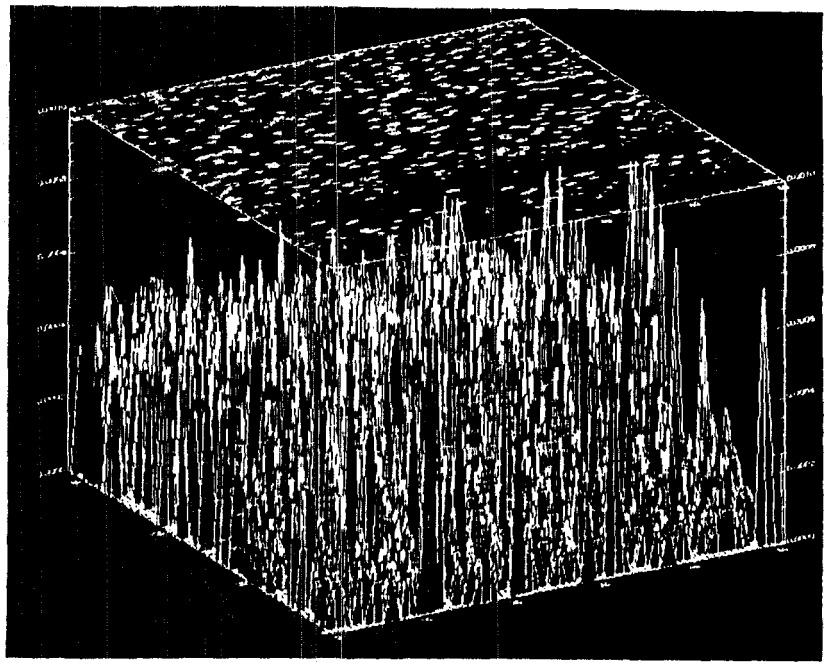


Figure 7: Relative differences between radiative stopping powers from ITS3.0 and MCNP4BNU all normalized to ITS3.0 for a wide range of materials from 1 keV to 1 GeV.

comparison. Once again the results are all within four significant figures except for a spike in the density effect calculation. This spike was for a very small correction which was near zero for both codes and so does not represent any significant problem.

In the patch, the bremsstrahlung production cross section array, pbr, the energy spectrum array, eba, and detailed angular distribution array, ech, are printed to separate files for debugging and verification purposes. H. G. Hughes maintains a version of the ITS3.0 code system where these arrays are printed to output files from the ITS3.0 cross section generating code xgenp. For all the elements for Z= 1 to 92 and over the energy range of 1 keV to 1 GeV, the average relative error in the "pbr" array is $10^{-6}$, in the spectra, $10^{-6}$, and in the angular distribution array, $10^{-6}$. The maximum for pbr is $10^{-5}$. For the "eba" array, the maximum relative difference is $10^{-4}$ and is in the first bin which is typically a small number. For ech the maximum relative difference is $10^{-2}$ again for the same reason as for eba. Since the ITS3.0 code has changed the angular distribution calculation only comestically, we have not changed our routine yet. Though it is recommended that these changes be made when the angular distributions are revised since the coding is more readable and easily related back to fundamental equations. Such revisions would make the inclusion of 2BS in the sampling scheme more straightforward.

The patch also uses an adapted cubic spline fitting routine in correspondence with ITS3.0. The sampling algorithms are in agreement except for the sampling of the angular

distribution. The patch includes what is needed to obtain exact ITS agreement though either method is valid.

The el3w bremsstrahlung cross sections stored in the pbr array are displayed in figure 8. The same array for el1 is also shown in figure 8. The curves are significantly different. Part of

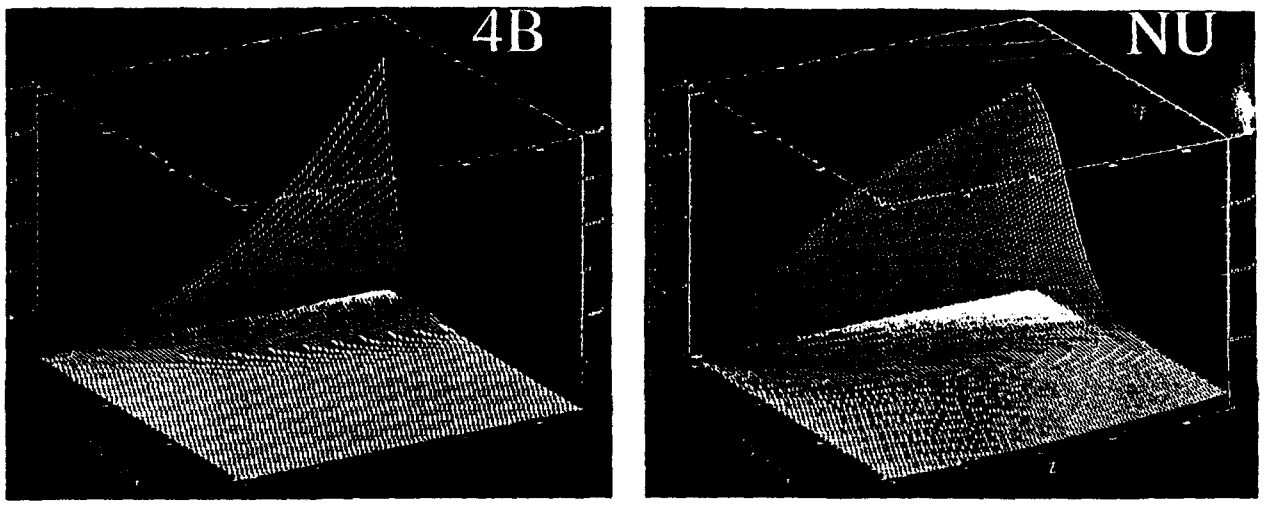



Figure 8: Bremsstrahlung production cross sections (pbr array) from MCNP4B (4B) and MCNP4BNU (NU) for Z=1 to 92 from 1 keV to 1 GeV.

the difference, as noted above, is that the el3w database and the el1 database have different lower limits of the photon energy for which the cross section was calculated. Symbolically, pbr for an electron of kinetic energy, T, is given by the following equation where the minimun photon energy

$$pbr(T) = \int_{k_{min}}^{T} \frac{d\sigma}{dk} dk$$

is kmin. Moreover, knowing that the eba array is the cumulative probability of producing a photon with energy between k/T and kmin/T, a scaling relation between the pbr arrays of the two databases would be

$$pbr_{el1}(T) = pbr_{el3w}(T)(1 - eba(0.001))$$

where eba(0.001) is the value of the eba at that ratio. Thus a fairer comparison is between a scaled pbr for el3w and el1. Figure 9 shows the relative difference between the scaled el3w and

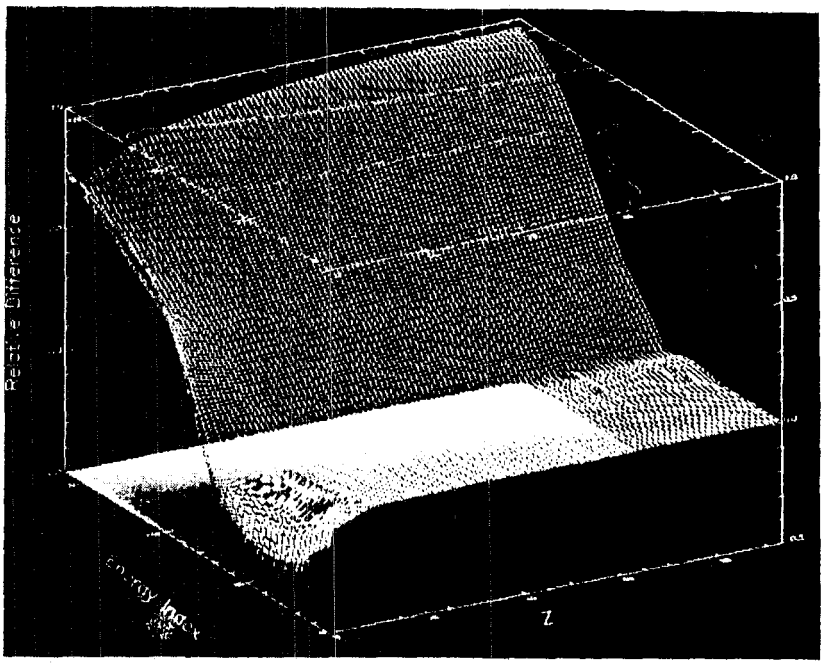



Figure 9: Scaled Bremsstrahlung production cross section relative differences between MCNP4B and MCNP4BNU for Z=1 to 92 from 1 keV to 1 GeV normalized to MCNP4B.

el1 normalized to el1. For low energies the differences are quite large and show that el1 had a very large low energy electron bremsstrahlung cross section.

At this level the algorithms for stopping power calculation and the bremsstrahlung production cross sections have been verified for all the elements and many representative materials over the energy range of 1 keV to 1 GeV. This level of verification meets or exceeds the verification standard of el1 and MCNP4A.

Implications:

Several problems have been suggested that show that the el1 model needed to be updated. Some of the problems still persist, however, improvements have been made. We have considered three energy ranges of problems for geometries similar to nominal converter foils.

The first is a low energy problem suggested by Newton Scientific. 20 keV electrons are normally incident on a 0.001cm W foil. The bremsstrahlung spectrum is tallied. Figure 10

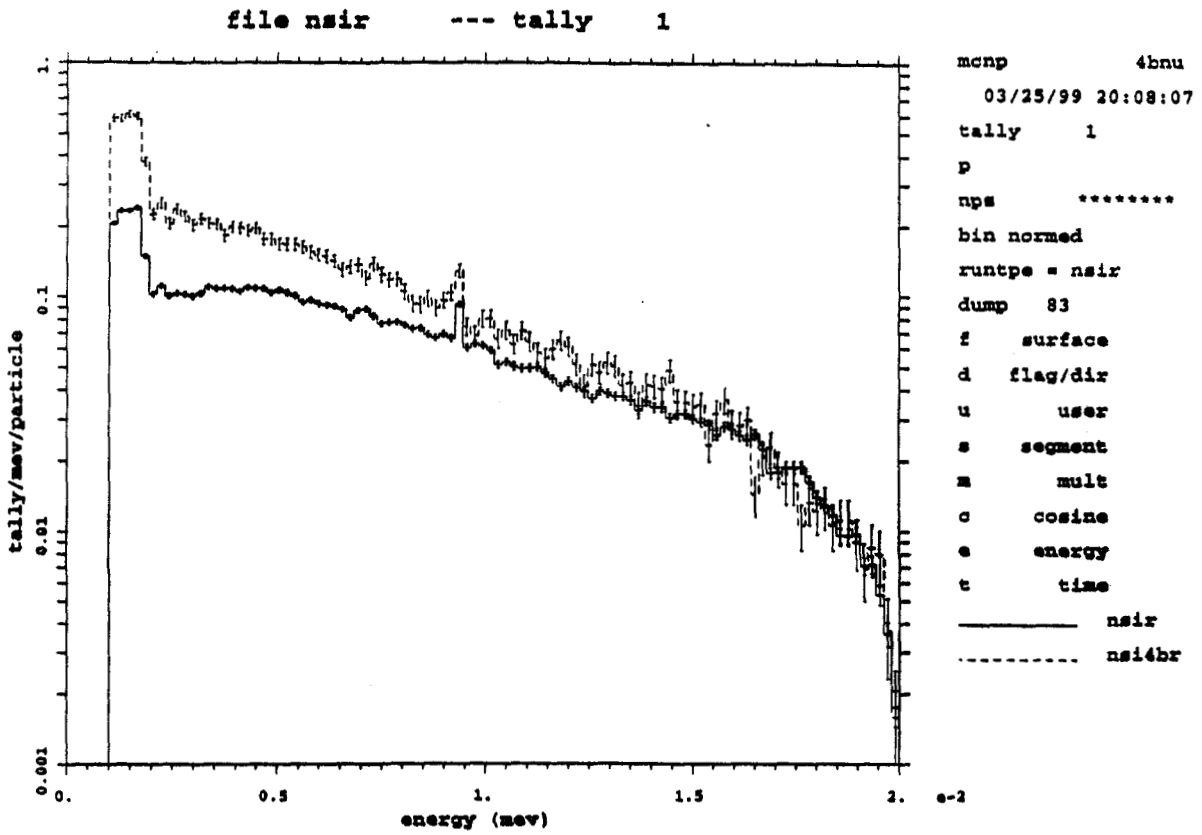



Figure 10: Bremsstrahlung spectra for 20 keV electron incident on 0.001 cm of W calculated with MCNP4B (nsi4br) and MCNP4BNU (nsir).

shows the spectrum calculated with 4bnu (the patched 4b which passed all the MCNP test problems) for the el1(4b in the plots) and el3w (4bnu in the plots) databases. The differences are quite large. In fact the 4b appears to have the M lines which of course it cannot. The 4bnu result is much less intense and more consistent with the NSI ITS calculation.

Higher in energy J. V. Siebers proposed a simple machine head model of a W/Cu slab. 6 MeV electons are normally incident. He has noted that the bremsstrahlung yield downstream depends on the maximum energy of the spectrum. We have not addressed that problem yet but note that it also apparent in the ITS codes. Table JV shows the results of a calculation for that problem with two choices of EMAX (the first entry on the phys:e card). The total bremsstrahlung generated from the summary tables shows a 0.12% difference which is beyond the statisitical bounds one would expect. The tally which scores the forward yield about a 1m from the converter

packs show a much larger variation. Even 4bnu has not answered this problem, however we do see that the yield is more forward directed by about 1%. Another implication apparent in this problem is inconsistency of the relaxation spectra as seen in Figure 11 where 4bnu has removed

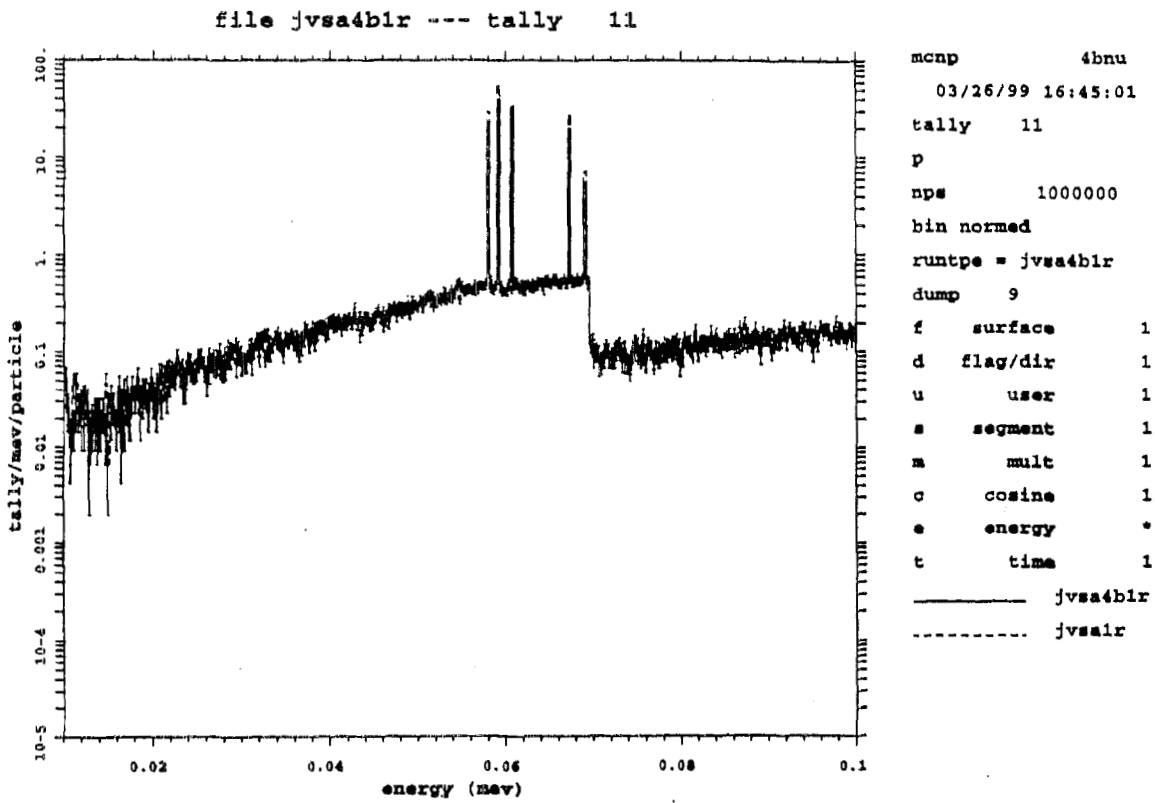



Figure 11: Bremsstrahlung and line spectra calculated with MCNP4B (jvsa4b1r) and MCNP4BNU (jvsa1r). The anomalous line has been removed in the el3w/4BNU calculation.

the anomalous line at 60.69 keV.

| code | EMAX=100 | | EMAX=6 | |
|---|---|---|---|---|
| mcnp4b | | | | |
| bremsstrahlung | 41622998 | 4.1623E+00 | 41551441 | 4.1551E+00 |
| tally in forward cone | 5.29151E-02 | 0.0014 | 5.15433E-02 | 0.0014 |
| mcnp4bnu | | | | |
| bremsstrahlung | 41362833 | 4.1363E+00 | 41283862 | 4.1284E+00 |
| tally in forward cone | 5.35427E-02 | 0.0014 | 5.21917E-02 | 0.0014 |

Finally, at high energies, A. E. Schah von Wittenau suggested a typical converter foil 0.1 cm of W with 85.0 MeV electrons normally incident. The spectra calculated with the two different databases are presented in Fig. 12. The el3w (not labeled asvw4br) results are all

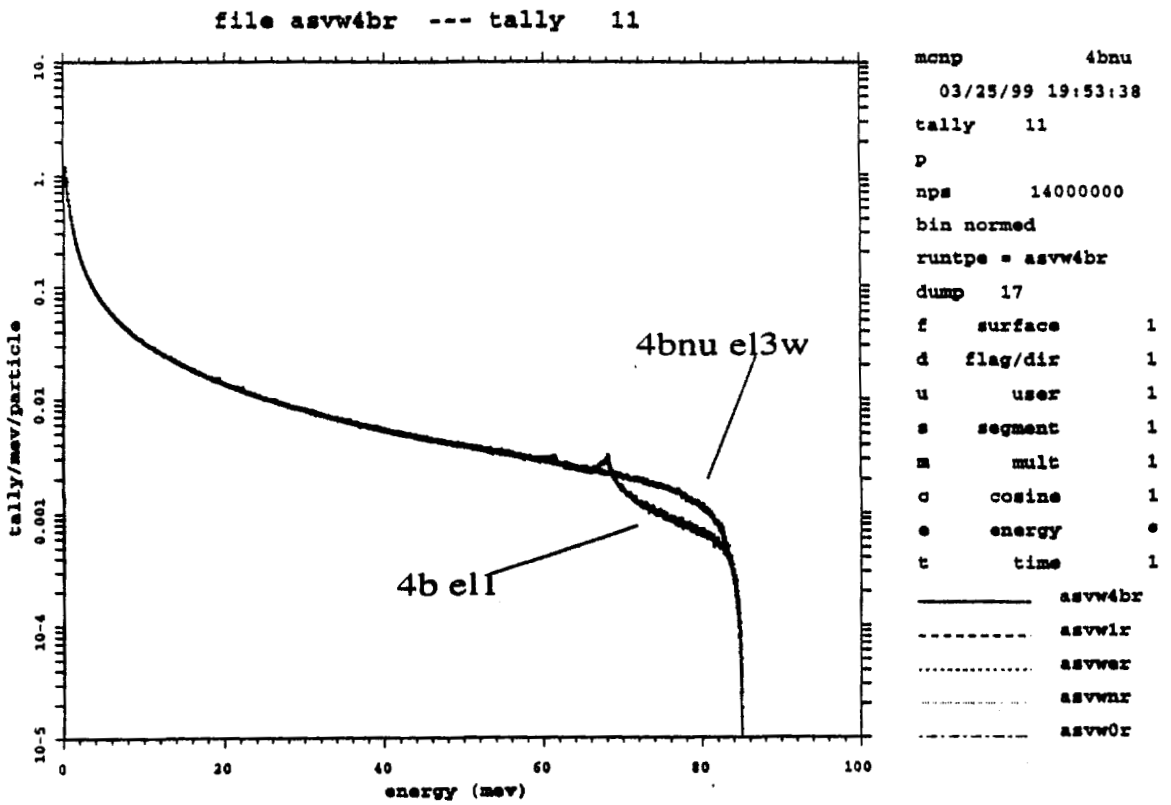



Figure 12: Bremsstrahlung spectra calculated with MCNP4B (asvw4br) and MCNP4BNU (others) for 85 MeV electrons incident on 0.1 cm W. Various variance reduction methods were tested giving nominally the same results.

smoother than the el1. The notches in the 4b calculation are probably due to non-smooth model transistions; 4bnu by definition is immune to such artifacts. All these problems except for asvw0 were run with some variance reduction. asvw1 has the new bnum biasing, asvwe is the same except W is made of it isotopic constituents, asvwn uses the numb biasing, asvw4b uses the old

bnum scheme. Fig 13 shows the spectra calculated with bbrem being used for asvw1 (labeled

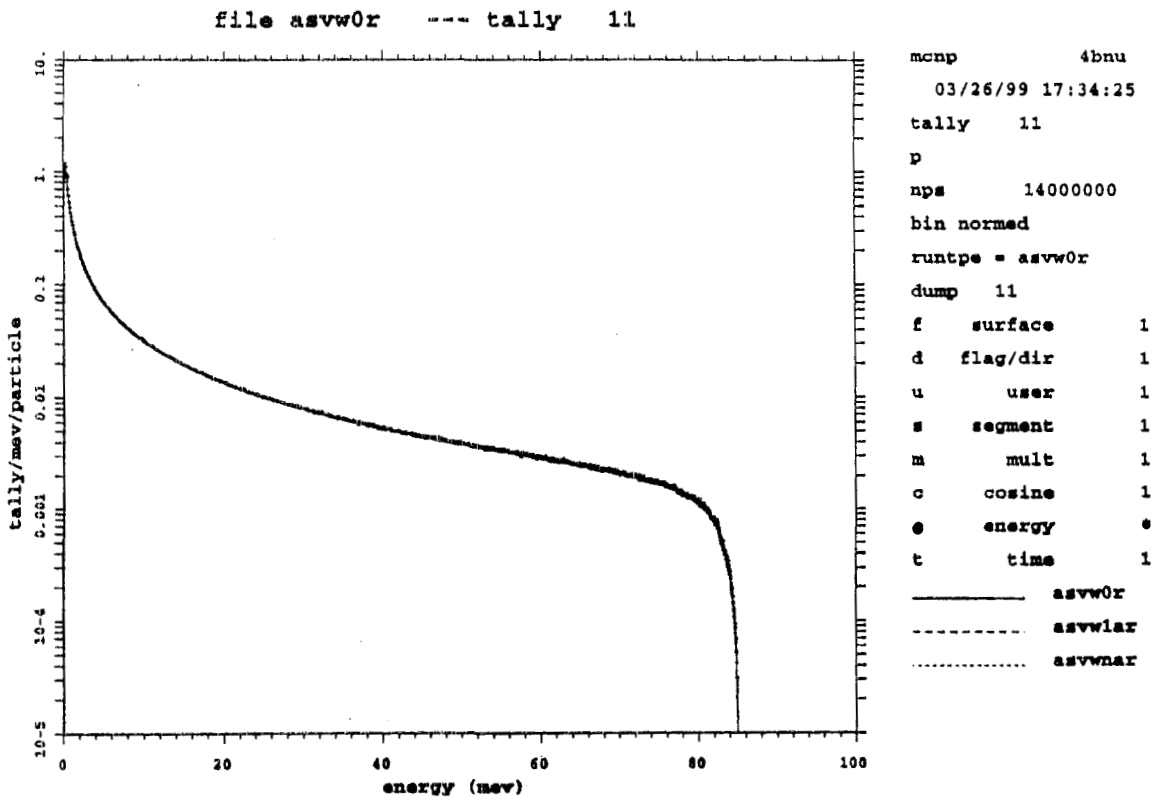
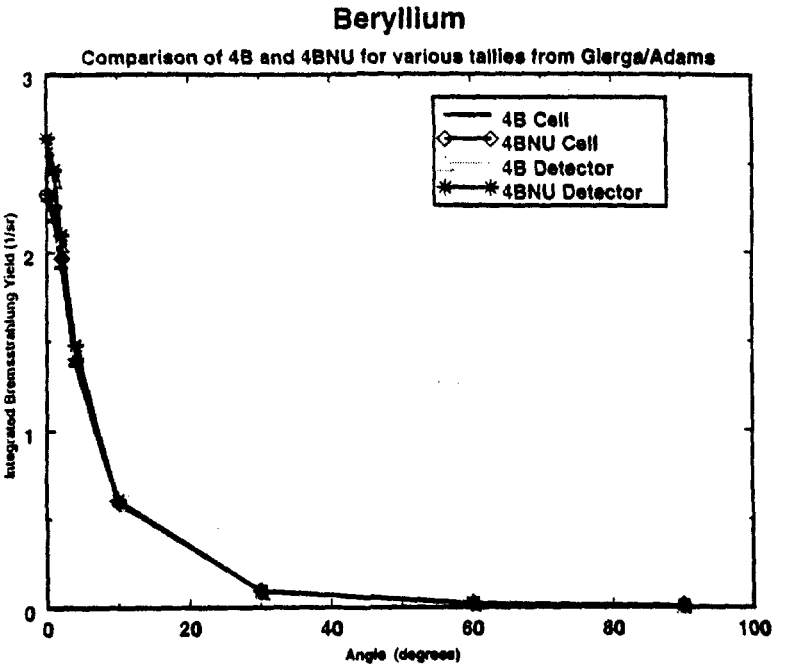


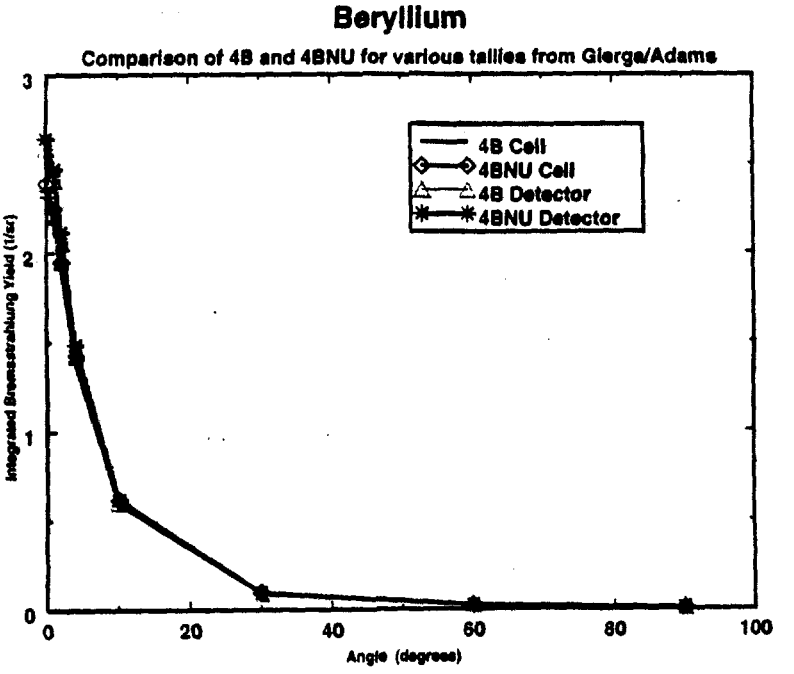

Figure 13: Bremsstrahlung spectra calculated with MCNP4BNU for 85 MeV electrons incident on 0.1 cm W with no variance reduction (asvw0r) and bbrem used in conjunction with new bnum biasins (asvw1ar) and numb biasing (asvwnar).

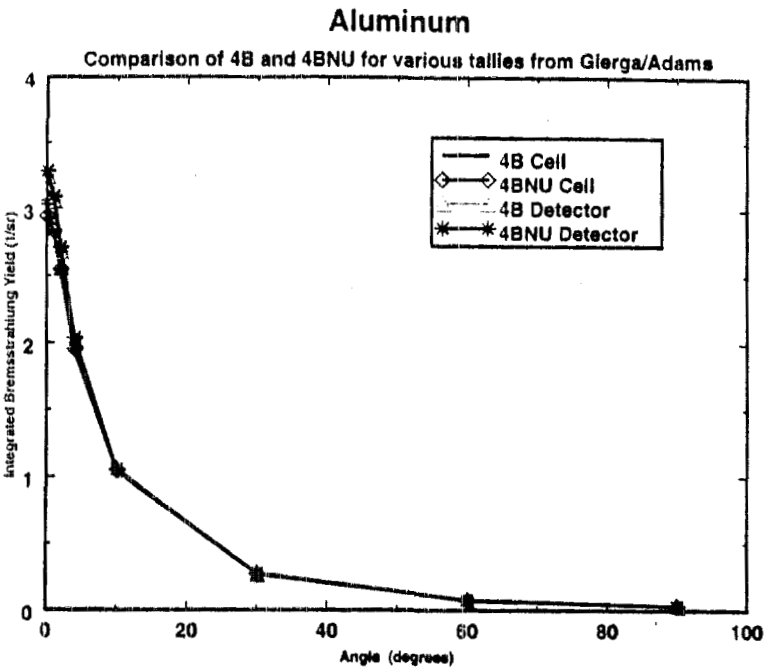
asvw1a) and asvwn (labeled asvwna) and asvw0. The impact of bbrem biasing has introduced no artifacts when used in conjunction with other biasing. It is amusing to note on this particularly thin problem, the largest fom was for no biasing.

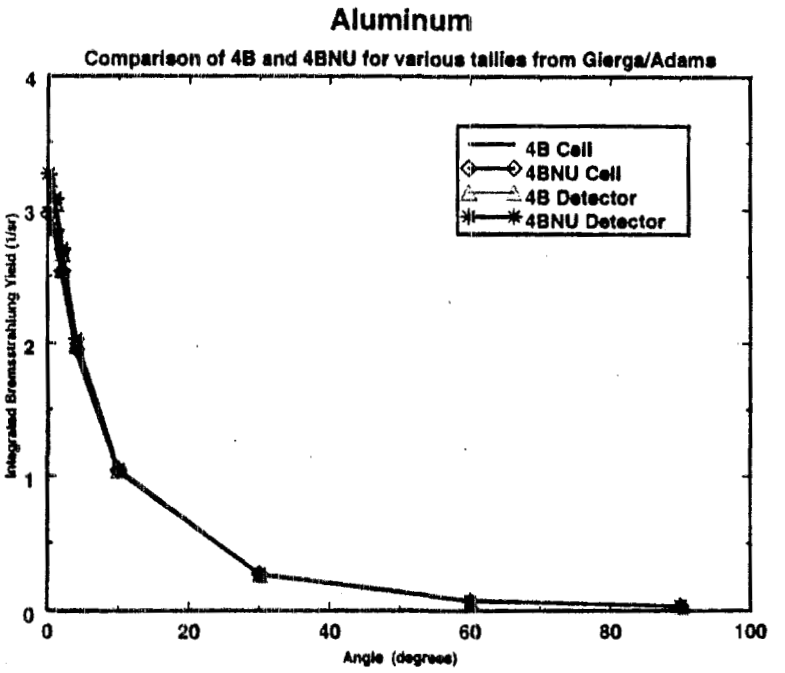
Validation:

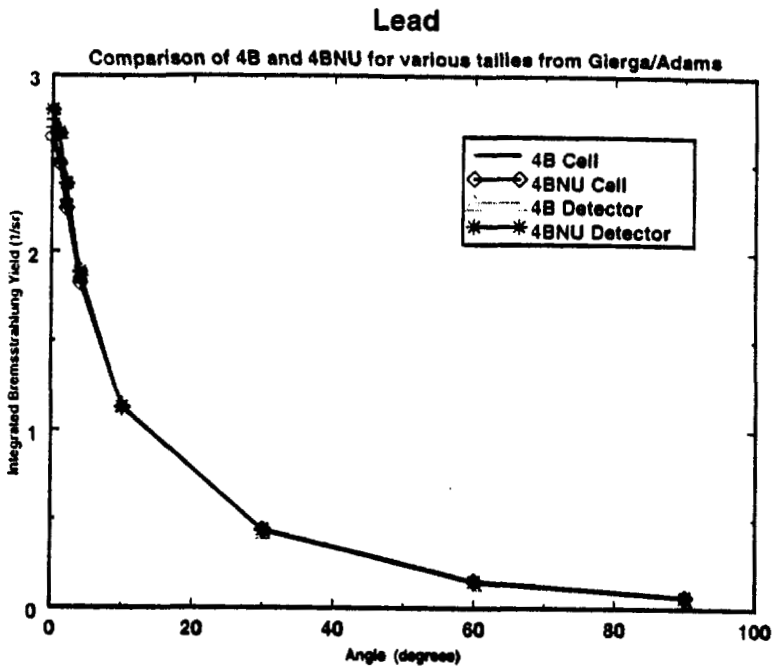
Gierga/Adams provided a very large suite of validation calculation templates. Due to budgetary limitations this set was not expanded. Nor have all the templates been run, only a few of the bremsstrahlung sensitive ones have been exercised. The results follow. Appendix C presents tables of some of the plots for more detailed comparisons.

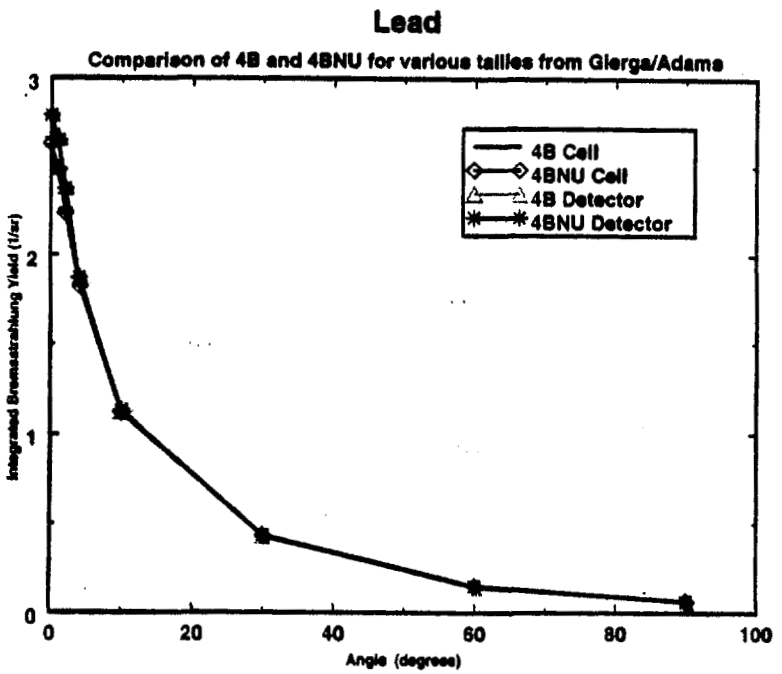

Beryllium
Comparison of 4B and 4BNU for various tallies from Gierga/Adams
4B Cell
4BNU Cell
4B Detector
4BNU Detector
Integrated Bremsstrahlung Yield (1/sr)
Angle (degrees)
0
1
2
3
0
20
40
60
80
100


Beryllium
Comparison of 4B and 4BNU for various tallies from Gierga/Adams
4B Cell
4BNU Cell
4B Detector
4BNU Detector
Integrated Bremsstrahlung Yield (1/sr)
Angle (degrees)
0
1
2
3
0
20
40
60
80
100

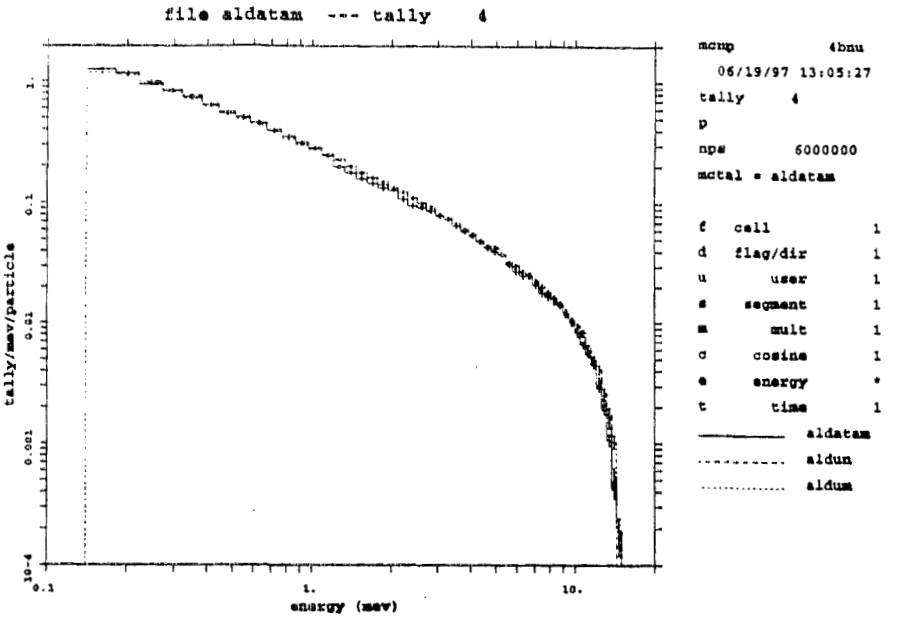

Aluminum
Comparison of 4B and 4BNU for various tallies from Gierga/Adams
4B Cell
4BNU Cell
4B Detector
4BNU Detector
Integrated Bremsstrahlung Yield (1/sr)
Angle (degrees)
0
1
2
3
4
0
20
40
60
80
100

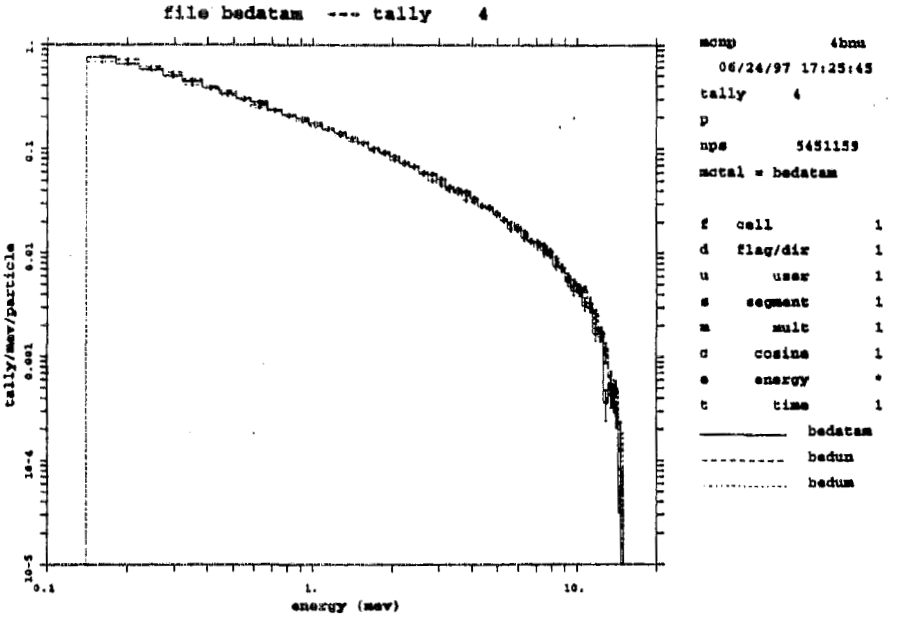

Aluminum
Comparison of 4B and 4BNU for various tallies from Gierga/Adams
4B Cell
4BNU Cell
4B Detector
4BNU Detector
Integrated Bremsstrahlung Yield (1/sr)
Angle (degrees)
0
1
2
3
4
0
20
40
60
80
100

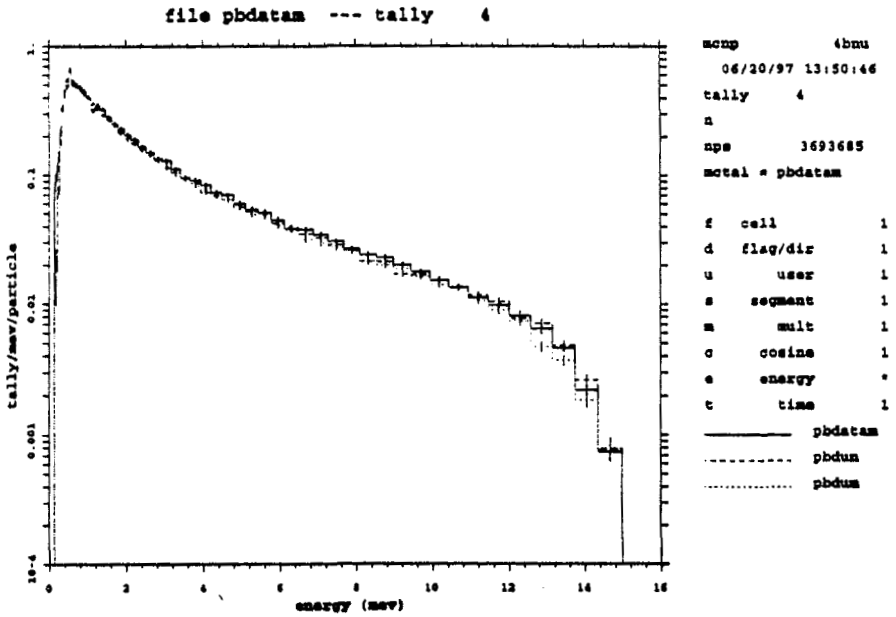

Lead
Comparison of 4B and 4BNU for various tallies from Gierga/Adams
4B Cell
4BNU Cell
4B Detector
4BNU Detector
Integrated Bremsstrahlung Yield (1/sr)
Angle (degrees)
0
1
2
3
0
20
40
60
80
100

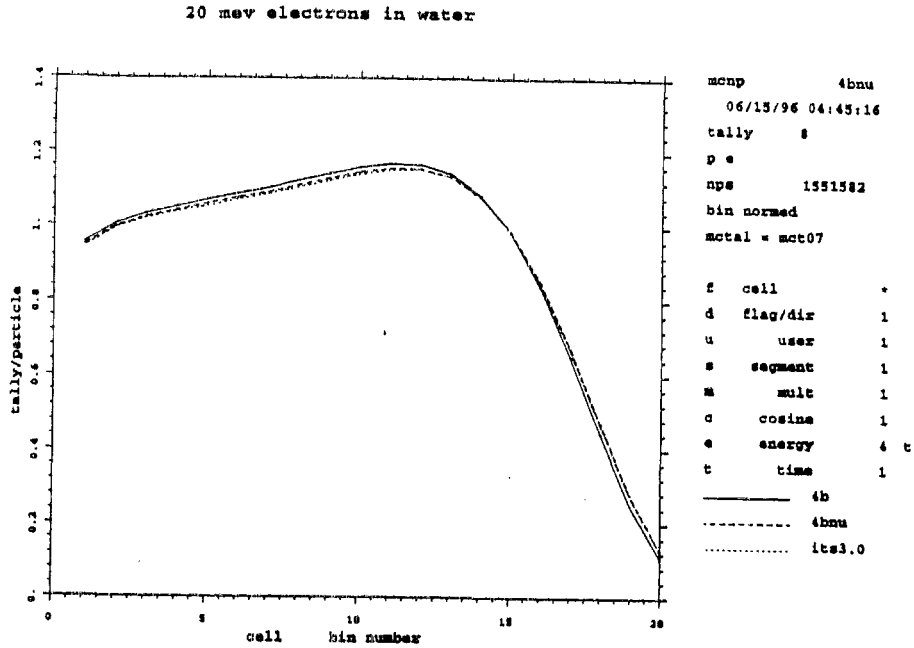

Lead
Comparison of 4B and 4BNU for various tallies from Gierga/Adams
4B Cell
4BNU Cell
4B Detector
4BNU Detector
Integrated Bremsstrahlung Yield (1/sr)
Angle (degrees)
0
1
2
3
0
20
40
60
80
100

Selected spectra comparisons with 4BNU labeled with the suffix "n" and 4B "m"




file pbdatam --- tally 4
tally/mev/particle
energy (mev)
mcnp 4bnu
06/20/97 13:50:46
tally 4
n
nps 3693685
mctal = pbdatam
f cell 1
d flag/dir 1
u user 1
s segment 1
m mult 1
c cosine 1
e energy *
t time 1
pbdatam
pbdum
pbdum


20 mev electrons in water
tally/particle
cell bin number
mcnp 4bnu
06/15/96 04:45:16
tally 8
p e
nps 1551582
bin normed
mctal = mct07
f cell *
d flag/dir 1
u user 1
s segment 1
m mult 1
c cosine 1
e energy 4 t
t time 1
4b
4bnu
its3.0

Testing:

To facillitate merging this patch with the evolving MCNP4C, several test problems have been generated. They fall into two classes. One set are the ones derived from the previous 4B test suite which were selected if they used the el1 database. This set consists of problems 04, 08, 10, 11, 20, 21, 22, 23, and 29. For problems 23 and 29, longer runs were done to assure that the test problems were giving reasonably assured results. A variant of 08 is included to assure that the code crashes and fails to run for mixed databases; this should not be allowed to be overcome with a fatal option since the results will probably be gibberish.

The second set focus on the new features mentioned above and are derived from the relatively straightforward "implication" validation set. inp30 and inp31 are derived from the high energy set. inp30 assures that the new bnum sampling is being done as expected with the bbrem card. inp31 checks the numb option for brems vr. inp32 is derived from the low energy problem with just fewer histories being run. It assures that the low energy portions of the database are being used as expected.

These test problems have exercised by D. Court and myself. Needless to say there are platform dependencies, however the test problems are easily checked with the converged results generated on Openblue.

Future work and conclusions:

The patch and database provide an improvement in the physics modelling of electron transport in MCNP. Comparisons with the progenitor code shows excellent agreement with calculated or derived quantities. Improvements in performance were illustrated with three sample problems. Comparisons with validation runs show improvement, albeit on the 1% level (as one might say the orginal developers did very good work so it is hard to improve it.) Several lessons have been learned from this effort on how new databases should be merged with the code. As such the original electron patch was a bandage and the current patch retains several of the "bandage" features.

There are still many remaining issues to be addressed for the electron/photon transport capabilities of MCNP.

1) The relaxation models are out of date and need to be improved (ref. 30 and 33).

2) Other bremsstrahlung and electron interaction databases (ref. 11) and the bremsstrahlung angular distribution calculation still is open

3) Emax and step size problems

4) Cleaning up the code and allow different databases to be used like the neutron side.

5) No-scatter calculation which is highly platform dependent.

6) Goudsmit-Saunderson treatment and general questions about multiple scattering and transport corrections such as TLC (transverse longitundinal correction).

7) Thick target bremsstrahlung.

Acknowledgements:

J. V. Siebers for helping debug the SGI version of the patch and suggesting the mid range problem. A. Schah von Wittenau. for suggesting the high energy problem and illuminating discussions on the bnum biasing schemes.

## Appendix A

```
implicit real*8 (a-h,o-z)
character line*80,xline*80,hd*10,rline(15)*3
dimension nxs(16),jxs(32),nxs3(16),jxs3(32)
dimension xss(100000),exs(100000),etb(57),phi(57),eta(57)
dimension emott(100),xmott(100,5),xsc(100000),nxc(16),jxc(32)
dimension xsp(10000),nxp(16),jxp(32)
dimension tb(100),x(100),xi(100),xsec(100,100),xsic(100,100),
     1 r(100),awit(100),ek(100),el(100,15),ea(100),
     1 rtwo(9),rfac(9),riley(9,12),rkac(51)
      data  kmax/89/r/0.000001d0,0.0000015d0,0.000002d0,0.000003d0,
     1 0.000004d0,0.000005d0,0.000006d0,0.000008d0,0.00001d0,
     2 0.000015d0,0.00002d0,0.00003d0,0.00004d0,0.00005d0,0.00006d0,
     3 0.00008d0,0.0001d0,0.00015d0,0.0002d0,0.0003d0,0.0004d0,
     4 0.0005d0,0.0006d0,0.0008d0,0.001d0,0.00125d0,0.0015d0,
     5 0.00175d0,0.002d0,0.0025d0,0.003d0,0.0035d0,0.004d0,0.0045d0,
     5 0.005d0,0.0055d0,0.006d0,0.007d0,0.008d0,0.009d0,0.01d0,
     6 0.0125d0,0.015d0,0.0175d0,0.02d0,0.025d0,0.03d0,0.035d0,
     7 0.04d0,0.045d0,0.05d0,0.055d0,0.06d0,0.07d0,0.08d0,0.09d0,
     8 0.1d0,0.125d0,0.15d0,0.175d0,0.20d0,0.225d0,0.25d0,0.275d0,
     9 0.3d0,0.325d0,0.35d0,0.375d0,0.4d0,0.425d0,0.45d0,0.475d0,0.5d0,
     9 0.525d0,0.55d0,0.575d0,0.6d0,0.65d0,0.7d0,0.75d0,0.8d0,0.85d0,
     9 0.9d0,0.95d0,0.97d0,0.99d0,0.995d0,0.9995d0,1.0d0,11*0.d0/,
     9 aneut/1.008664967d0/
      data rkac/.9999d0,0.8d0,0.6d0,0.5d0,0.4d0,0.3d0,0.2d0,0.15d0,
     1 0.1d0,0.08d0,0.06d0,0.05d0,0.04d0,0.03d0,0.02d0,0.015d0,0.01d0,
     2 0.008d0,0.006d0,0.005d0,0.004d0,0.003d0,0.002d0,0.0015d0,0.001d0,
     3 0.0005d0,0.0002d0,0.0001d0,0.00005d0,0.00002d0,0.00001d0,
     4 0.000005d0,0.000002d0,0.000001d0,17*0.0d0/
iel1=7
imcplib=iel1+1
ixdata=imcplib+1
ixold=ixdata+1
ixnew=ixold+1
ius=ixnew+1
ius1=ius+1
ius2=ius1+1
ius3=ius2+1
iel30=ius3+1
      open(iel1,file='el1')
open(imcplib,file='mcplib02')
open(ixdata,file='xdata.30')
open(ixold,file='xsdir1')
open(ixnew,file='xsdir3w')
open(ius,status='scratch')
open(ius1,status='scratch')
open(ius2,status='scratch')
open(ius3,status='scratch')
      open(iel30,file='el3w')
c
c         read atomic weights and edge data from xdata.
read(ixdata,'(a80)')line
do 1 iz=1,100
read(ixdata,*)kzit,awit(iz),n,nn
```

```
awit(iz)=awit(iz)/aneut
cwrite(6,*)awit(iz)
ek(iz)=0.
ea(iz)=0.
if(n.eq.0)go to 1
      read(ixdata,'(5(a3,f9.1,i4))')(rline(il),el(iz,il),nm,il=1,n)
ek(iz)=el(iz,1)*1d-3
ea(iz)=ek(iz)
if(n.lt.2)go to 1
ae=0.0
nl=0
do 2 il=2,n
if(rline(il)(2:2).ne.'L')go to 2
ae=ae+el(iz,il)*1d-3
nl=nl+1
2continue
ea(iz)=ea(iz)-ae/nl
1continue
c
c         start reading and processing in "jxs" order.
c
c         read and process the radiative stopping powers.
10read(ixdata,'(a80)')line
if(line(1:11).ne.'PHIRADB.DAT')go to 10
read(ixdata,*)lmax,lrax,mrax
read(ixdata,*)(etb(lmax-i+1),i=1,lmax)
do 60 iz=1,100
read(ixdata,*)(phi(lmax-i+1),i=1,lmax),(eta(lmax-i+1),i=1,lmax)
if(iz.le.94)then
read(iel1,'(f9.2,1he,2f12.6,1x,a10)')zaid,awit(iz),tz,hd
hd='6/6/98'
write(ius,'(f9.2,1he,2f12.6,1x,a10)')zaid,awit(iz),tz,hd
call readl(5,iel1,ius)
read(iel1,'(8i9)')(nxs(i),i=1,16),(jxs(i),i=1,32)
read(iel1,'(1p4e20.12)')(xss(i),i=1,nxs(1))
call xs23(nxs,jxs,nxs3,jxs3)
else
call zeroxs(nxs,jxs)
call zeroxs(nxs3,jxs3)
zaid=1000*iz+.01
tz=0.0
hd='6/6/98'
write(ius,'(f9.2,1he,2f12.6,1x,a10)')zaid,awit(iz),tz,hd
line=' '
do 15 i=1,5
15write(ius,'(a80)')line
      endif
jxs(1)=1
nxs3(1)=2+3*lmax
nxs3(2)=iz
nxs3(3)=lmax
jxs3(1)=1
jxs3(2)=3
jxs3(3)=jxs3(2)+3*lmax
c
c         flag the new treatment.
```

```
      nxs3(16)=3
if(iz.le.94)then
call readl(6,imcplib,0)
read(imcplib,'(8i9)')(nxp(i),i=1,16),(jxp(i),i=1,32)
    read(imcplib,'(1p4e20.12)')(xsp(i),i=1,nxp(1))
do 30 i=1,jxs3(2)-1
30exs(i)=xss(i)
c fix the kxray line and auger emission value at least for
c consistency with fluorescence model.
c exs(jxs3(1))=kxray edge threshold.
c exs(jxs3(1)+1)=auger emission energy
      if(nxp(4).ne.0)then
exs(jxs3(1))=xsp(jxp(4)+nxp(4)-1)*1d3+1d-6
exs(jxs3(1)+1)=exs(jxs3(1))
if(nxp(4).gt.2)exs(jxs3(1)+1)=(xsp(jxp(4)+nxp(4)-1)-
     1 2*xsp(jxp(4)+1))*1d3
if(exs(jxs3(1)+1).lt.0.)exs(jxs3(1)+1)=
     1 min(abs(exs(jxs3(1)+1)),
     1 xss(jxs(1)+1))
endif
else
exs(jxs3(1))=ek(iz)
exs(jxs3(1)+1)=ea(iz)
endif
      do 40 i=1,nxs3(3)
exs(jxs3(2)+i-1)=etb(i)
exs(jxs3(2)+i-1+nxs3(3))=phi(i)
40exs(jxs3(2)+i-1+2*nxs3(3))=eta(i)
write(ius,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
60write(ius,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
      rewind ixdata
      rewind ius
c
c       read the old ratios for comparison sake.
      rewind iell
c
c         read mott cross section ratios.
61    read(ixdata,'(a80)')line
if(line(1:9).ne.'MOTTE.DAT')go to 61
read(ixdata,*)imott,ixmott
read(ixdata,*)(emott(i),i=1,imott)
read(ixdata,'(a80)')line
do 63 iz=1,100
read(ixdata,*)((xmott(i,j),i=1,imott),j=1,ixmott)
call readl(6,ius,ius1)
read(ius,'(8i9)')(nxs(i),i=1,16),(jxs(i),i=1,32)
read(ius,'(1p4e20.12)')(xss(i),i=1,nxs(1))
call upto(exs,xss,jxs(3))
      call xs23(nxs,jxs,nxs3,jxs3)
nxs3(4)=imott
nxs3(1)=nxs(1)+(ixmott+1)*nxs3(4)
jxs3(4)=jxs(3)+(ixmott+1)*nxs3(4)
do 62 i=1,imott
62exs(jxs3(3)+i-1)=emott(i)
do 602 j=1,ixmott
jd=j*imott
```

```
do 602 i=1,imott
602exs(jxs3(3)+i-1+jd)=xmott(i,j)
c
c        skip over positron data
read(ixdata,*)((xmott(i,j),i=1,imott),j=1,ixmott)
write(ius1,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
63write(ius1,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
rewind ius
rewind ius1
rewind ixdata
c
c         read riley cross sections.
cRILEY.DAT
64    read(ixdata,'(a80)')line
if(line(1:9).ne.'RILEY.DAT')go to 64
do 67 iz=1,100
do 65 ir=1,9
read(ixdata,*)kz,rtwo(ir),rfac(ir)
65read(ixdata,'(5e15.8)')(riley(ir,ik),ik=1,12)
call readl(6,ius1,ius)
read(ius1,'(8i9)')(nxs(i),i=1,16),(jxs(i),i=1,32)
read(ius1,'(1p4e20.12)')(xss(i),i=1,nxs(1))
call upto(exs,xss,jxs(4))
      call xs23(nxs,jxs,nxs3,jxs3)
jxs3(5)=jxs3(4)+9*14
nxs3(1)=nxs(1)+9*14
io=-1
do 66 ir=1,9
io=io+1
exs(jxs3(4)+io)=rtwo(ir)
io=io+1
exs(jxs3(4)+io)=rfac(ir)
do 66 ik=1,12
io=io+1
66exs(jxs3(4)+io)=riley(ir,ik)
write(ius,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
67write(ius,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
rewind ius
rewind ius1
rewind ixdata
c
c        now do the bremsstrahlung production
70    read(ixdata,'(a80)')line
if(line(1:9).ne.'BREME.DAT')go to 70
      read(ixdata,'(2i6)') nmix,kmix
      read(ixdata,'(6f12.5)') (tb(n),n=1,nmix)
      read(ixdata,'(6f12.5)') (x(k),k=1,kmix)
read(ixdata,'(a80)')line
do 120 iz=1,100
call readl(6,ius,ius2)
read(ius,'(8i9)')(nxs(i),i=1,16),(jxs(i),i=1,32)
read(ius,'(1p4e20.12)')(xss(i),i=1,nxs(1))
call upto(exs,xss,jxs(5))
      call xs23(nxs,jxs,nxs3,jxs3)
nxs3(5)=nmix
nxs3(6)=kmix
```

```
jxs3(6)=0
nxs3(1)=nxs(1)+nmix+kmix+nmix*kmix
read (ixdata,'(6f9.5)') ((xsic(n,k),n=1,nmix),k=1,kmix)
      do 90 n=1,nmix
90exs(jxs3(5)+n-1)=tb(n)
do 100 k=1,kmix
exs(jxs3(5)+k-1+nmix)=x(k)
c
c          1d-3 converts from mb to b.
do 100 n=1,nmix
100exs(jxs3(5)+nmix+kmix+nmix*(k-1)+n-1)=xsic(n,k)*1d-3
c
c          write out the normalized photon grid numbers.
c          these are rkt and rka but to allow for flexibility, they
c          will now be read from el30.
c          the jxs indices are bumped up since jxs(7) and jxs(8) are
c          used already.  Then for consistency nxs is bumped up to have
c          the same indices.
nxs3(9)=kmax
jxs3(9)=nxs3(1)+1
nxs3(1)=nxs3(1)+nxs3(9)
nxs3(10)=34
jxs3(10)=nxs3(1)+1
nxs3(1)=nxs3(1)+nxs3(10)
do 110 i=1,nxs3(9)
110exs(jxs3(9)+i-1)=r(i)
do 115 i=1,nxs3(10)
115exs(jxs3(10)+i-1)=rkac(i)
write(ius2,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
120write(ius2,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
c
c         binding energies and occupation numbers of Carlson for density
c         effect corrections.
      rewind ixdata
      rewind ius2
140   read(ixdata,'(a80)')line
if(line(1:12).ne.'BINDENGY.DAT')go to 140
      do 160 iz=1,100
read(ixdata,*)izx,isets
      read(ixdata,*)(x(i),xi(i),i=1,isets)
      call readl(6,ius2,ius1)
read(ius2,'(8i9)')(nxs(i),i=1,16),(jxs(i),i=1,32)
read(ius2,'(1p4e20.12)')(exs(i),i=1,nxs(1))
      call xs23(nxs,jxs,nxs3,jxs3)
nxs3(11)=isets
nxs3(1)=nxs(1)+2*isets
jxs3(11)=nxs(1)+1
do 150 ij=1,isets
exs(jxs3(11)+ij-1)=x(ij)
150exs(jxs3(11)+ij+isets-1)=xi(ij)
write(ius1,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
160write(ius1,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
      rewind ius1
c
c          all done now write the new data file, el30,
c          directory file, xsdir30.
```

```
nline=1
170read(ixold,'(a80)',end=220,err=220)xline
      iyes=0
line=xline
do 180 j=1,5
jj=j+19
180if(line(jj:jj+2).eq.'ell')iyes=jj
if(iyes.eq.0)then
write(ixnew,'(a80)')xline
go to 170
endif
do 200 i=1,6
      read(ius1,'(a80)')line
if(i.eq.1)then
   do 190 j=1,3
   jj=j+7
190   if(line(jj:jj).eq.'1')line(jj:jj)='3'
   write(6,'(a80)')line
endif
200write(iel30,'(a80)')line
      read(ius1,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
read(ius1,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
write(iel30,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
write(iel30,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
line=xline
line(7:8)='03'
line(iyes+2:iyes+4)=' '
line(iyes+2:iyes+4)='3x '
n0=0
n1=1
t=0.0
write(ixnew,210)line(1:27),n0,n1,nline,nxs3(1),
     1 n0,n0,t
210format(a26,1x,2i2,2i8,2i2,f4.1)
noff=13
if(mod(nxs3(1),4).eq.0)noff=12
      nline=nline+nxs3(1)/4+noff
write(ixnew,'(a80)')xline
go to 170
c
c        add in the final zaids to 100.
220   continue
      do 250 iz=95,100
do 240 i=1,6
      read(ius1,'(a80)')line
if(i.eq.1)then
   do 230 j=1,3
   jj=j+7
230   if(line(jj:jj).eq.'1')line(jj:jj)='3'
   write(6,'(a80)')line
xline=line
if(line(1:1).eq.' ')xline(1:27)=line(2:28)
endif
240write(iel30,'(a80)')line
      read(ius1,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
read(ius1,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
```

```
write(iel30,'(8i9)')(nxs3(i),i=1,16),(jxs3(i),i=1,32)
write(iel30,'(1p4e20.12)')(exs(i),i=1,nxs3(1))
n0=0
n1=1
t=0.0
xline(22:27)=' el3x '
write(ixnew,210)xline(1:27),n0,n1,nline,nxs3(1),
     1 n0,n0,t
noff=13
if(mod(nxs3(1),4).eq.0)noff=12
250   nline=nline+nxs3(1)/4+noff
stop
      end
subroutine readl(n,iin,iout)
character line*80
do 10 i=1,n
      read(iin,'(a80)',end=999)line
cif(i.eq.1)write(6,'(a80)')line
10if(iout.gt.0)write(iout,'(a80)')line
      return
999write(*,*)' problem reading unit ',iin
stop
end
subroutine xs23(nxs,jxs,nxs3,jxs3)
dimension nxs(16),jxs(32),nxs3(16),jxs3(32)
do 10 i=1,16
nxs3(i)=nxs(i)
jxs3(i)=jxs(i)
10    jxs3(i+16)=jxs(i+16)
return
end
subroutine zeroxs(nxs,jxs)
dimension nxs(16),jxs(32)
do 10 i=1,16
nxs(i)=0
jxs(i)=0
10    jxs(i+16)=0
return
end
subroutine upto(exs,xss,jxs)
implicit real*8 (a-h,o-z)
dimension exs(1),xss(1)
do 10 i=1,jxs
10exs(i)=xss(i)
return
end
```

# Appendix B

Patches to MCNP for the electron upgrade

Patch to 4xu with variables being defined at the being of patchf:

```
c          <<<<< makemcnp changes for MCNP4B2 >>>>>       loddat 09/22/97
c
c    For each fix, enter five integer parameters on one line in free
c    format, ip(1)-ip(5) described below, followed by the appropriate
c    number of new lines.
c
c    ip(1) = The applicable computer system number or 0 for all systems.
c    ip(2) = The file number:
c        1 = patchf
c        2 = makemcnp
c        3 = patchc
c    ip(3) = The applicable line number to edit-see the MCSETUP source.
c    ip(4) = The alter code:
c        1 = insert the following ip(5) line(s) before line ip(3)
c        2 = replace line ip(3) with the following ip(5) line(s)
c        3 = insert the following line after entry ip(5) of line ip(3)
c        4 = delete lines ip(3) through ip(5)
c    ip(5) = The number of new line(s) if ip(4)=1,2
c            The entry number if ip(4)=3
c            The last line number to delete if ip(4)=4
c
c    The number of new lines that follow ip(1)-ip(5) is given by:
c          = ip(5) if ip(4)=1,2
c          = 1 if ip(4)=3
c          = 0 if ip(4)=4

c******************** FIXES FOR SYSTEM 1 (Cray UNICOS) ****************
c
c     Add the *define t3d for the Cray T3D.
c      1 1 2 1 1
c     *define t3d
c
c     Provide links to the T3D compilers and linker. XTM:GWM-95-128
c     Change the CFT77 compile line.
c     1 2 18 1 4
c     export TARGET; TARGET=CRAY-T3D
c     ln -s /mpp/bin/cc cc
c     ln -s /mpp/bin/cft77 cft77
c     ln -s /mpp/bin/mppldr segldr
c     Provide links to the T3D compilers and linker. XTM:GWM-95-128
c     1 2 26 2 1
c     cft77 compile
c
c     Add CFTLIB library on UNICOS for profile timing.    06/30/95 (GWM)
c     1 2 28 3 10
c      -L/usr/local/lib -lprof

c********************* FIXES FOR SYSTEM 2 (Sun  SunOS ) ****************

c     Add a second X-window include path for some SunOS systems.
c     2 2 18 3 4
```

```
c       -I/usr/openwin/include/X11

c      Add a link to the ANSI C library (libansi.a) on some SunOS systems.
c      2 2 29 3 10
c       -L/home -lansi

c********************** FIXES FOR SYM 3 (IBM  AIX   ) ****************
c
c      PVM libraries must go first for pvm version at some installations.
3 2 30 2 1
xlf -o mcnp *.o -L/usr/lanl/pvm3.3/lib -lfpvm3 -lpvm3 -L/usr/lib -lX11

c********************** FIXES FOR SYSTEM 4 (HP   HPUX  ) ****************
c
c      Add a link to the C math library (libM.a) on some HP systems.
c      4 2 29 3  6
c       -lM

c********************** FIXES FOR SYSTEM 5 (VAX  VMS   ) ****************
c
c      Add an additional *define keyword for PRPR on a VAX VMS system.
c       5 1 2 1  1
c      *define disscgs

c**************** FIXES FOR SYSTEM 6 (DEC UNIX and PC LINUX) ************
c
c      The following 7 lines enable GNU F77 compilation on LINUX.
c      Tim Goorley, jgoorley@mit.edu XTM-RN(U97-028) 7/21/97
c      6 1 2 1 1
c      *define linux
c      6 2 27 3 1
c       -O0
c      6 2 28 4 28
c      6 3 2 1 1
c      *define linux

c********************** FIXES FOR SYSTEM 7 (PC   DOS   ) ****************
c
c      The following lines enable dvf F90 compilation on DOS/WINDOWS.
7 2 4 2 59
if exist flib del flib\*.for
if exist olib del olib\*.obj
if exist compile del compile
if exist newid del newid
if exist patch del patch
copy prpr.id prpr.for
f90 prpr.for
del prpr.for
copy makxs.id codef
type patchf | find "*define" > patch
prpr
rename compile makxsf.for
f90 makxsf.for
del makxsf.for
del codef
del patch
del newid
type patchf | find "*define pcdos" | find "xlib"
if errorlevel 1 goto noxwinl
```

```
if exist mcnpc.c del mcnpc.c
copy mcnpc.id codef
copy patchc patch
prpr
rename compile mcnpc.c
del codef
del patch
del newid
hcx86 -f387 -DMSDOS -Hoff=protection -I\dvx\include -c mcnpc.c
:noxwin1
copy mcnpf.id codef
copy patchf patch
prpr
del codef
del patch
del newid
fsplit compile
if errorlevel 1 goto nosplit
del compile
f90 /compile_only /optimize:4 *.for
goto link
:nosplit
rename compile mcnp.for
lf90 -c -nw -o0 -mod \lf9040\lib mcnp.for
:link
move mcsetup.for mcsetup.id
if exist mcsetup.obj del mcsetup.obj
if not exist flib mkdir flib
move *.for flib
move mcsetup.id mcsetup.for
type patchf | find "*define pcdos" | find "xlib"
if errorlevel 1 goto noxwin2
set lib=\f7713\lib;\hcx3\small;\dvx\lib\hcx87
lf90 @automake.rsp \lf9040\lib\hcx20.obj mcnpc.obj -lib hcx86,hcx87,hcna,x11,sys -
nomap -exe mcnp.exe
goto end
:noxwin2
f90 /exe:mcnp.exe *.obj
:end
if not exist olib mkdir olib
move *.obj olib
7 2 5 4 52

c********************* fixes for system 8 (sun  sol.  ) *****************

c********************* fixes for system 9 (sgi  irix  ) *****************

9 1 2 1 1
*define multt
c
9 2 5 2 1
rm -rf compile newid patch codef flib olib
c
9 2 13 2 1
rm -f newid codef patch *.f *.o
c
9 2 14 2 1
ln -s mcnpc.id codef
9 2 15 2 1
```

```
ln -s patchc patch
9 2 18 2 1
cc -64 -g -c -I/usr/include mcnpc.c
c
9 2 19 2 2
rm -f codef patch
mv newid newidc
c
9 2 20 2 1
ln -s mcnpf.id codef
9 2 21 2 1
ln -s patchf patch
c
9 2 24 2 2
rm -f compile codef patch clog
mv newid newidf
9 2 27 2 1
f77 -64 -mp -O2 -TARG:madd=OFF -OPT:Olimit=3000 -c  *.f
9 2 28 2 1
f77 -64 -mp -O0 -TARG:madd=OFF -OPT:Olimit=3000 -c  trnspt.f
9 2 30 3 1
 -64 -mp
9 2 30 3 12
 \
9 2 31 1 2
 -Wl,-Xlocal,pblcom_  -Wl,-Xlocal,tskcom_  -Wl,-Xlocal,itskpt_


c    Delete lines 2-7 of PATCHF.
 0 1 2 4 10
c    Add the entire following patch to PATCHF.
 0 1 11 2
*/ Integrate your patch with the following. If your patch makes changes
*/ to the ZC, VV, CM, GS, MB, or BD decks, contact MCNP@LANL.GOV for
*/ details on including your patch via the INSTALL.FIX file. When
*/ integrating your patch, be sure the order of the changes (increasing
*/ in line number) is preserved to avoid a PRPR error.  See Appendix
*/ C, page C-8 in the MCNP manual for more details.
*/
*/ brief description of variables
*/ its30: flag only to ensure that its3.0 treatment is done
*/ in xsgen and colidp when nxs may have been zeroed.
*/ ntop and nwng: variables for its1.0 parameterized variables.
*/ mtop and mwng: its3.0 parameterized variables larger than its1.0 ones.
*/ nstp and mstp: number of energy groups spanned in angular brems cal.
*/ nstp is the number used in the calculation and mstp is a parameter to
*/ make enough room is setdas.
*/ rktc: its1.0 photon energy ratios.
*/ crelec: classical electron radius
*/ istern: flag for sbs density effect treatment.
*/ numb: flag for producing brems in each step.
*/ cond, lcon: flag and offset for conductivity assignment in each material.
*/ lmb: memory offset for temporary dynamic memory.  (allocate and deallocate)
*ident its30
*d,zc4xs.3                                                          <21>
      parameter (kod='mcnp',ver='4bnu',loddat='09/21/99')
*d,zc4xu.3 2 mlgc=1000,mpb=5,mpng=21,mseb=301,mspare=3,mtop=49,mwng=25<27-28>
     2 mlgc=1000,mpb=5,mpng=21,mseb=301,mspare=3,
     2 mtop=89,mwng=(mtop+1)/2,mstp=4,mbng=51,
```

```
*d,zc4b.12                                                          <47>
*/ include the fundamental constant crelec=classical electron radius.
*/ crelec=planck*slite/(2d0*pie*fscon*gpt(3))
     9 euler=.577215664901532861d0,crelec=2.8179380d-13,
*d,vv4b.1,vv4a.2                                                  <55-56>
      common /tables/ gpt(mipt),qfiss(23),rktc(mtop),
*/ a fixed array if rka is not read from the database.
*/    1 rkac(mbng),
     1 talb(8,2),vco(mcoh),vic(minc),wco(mcoh),
*d,cm4xu.5          9 )                                           <85-86>
     9 +1*(mtop)+1*(mbng))
*d,cm4xu.9       9 )
     9 +8*(1) )
*d,cm4xu.32
      common /fixcom/ almin,bbrem(mtop),bnum,calph(maxi),coincd,
     1 rkt(mtop),rka(mbng),
*d,cm4xu.44      5 ism(3),issw,istrg,iunr,ivdd(maxf),ivdis(maxv),ivord(maxf),iwwg,
*/ +2
     5 ism(3),issw,istern,its30,istrg,
     5 iunr,ivdd(maxf),ivdis(maxv),ivord(maxf),iwwg,
*d,cm4xu.54      6 nwgma,nww(mipt),nwwm,nwwma,nxnx
     6 nwgma,nww(mipt),nwwm,nwwma,nxnx,
*/ +4
     & ntop,nwng,nstp,numb
*d,cm4xu.69     1 lscr,lemi,lptr,lpts,lrng,lxlk,lyla,lixc,ljmt,lkmt,
*/ +2
     1 lscr,lemi,lcon,lmb,
     1 lptr,lpts,lrng,lxlk,lyla,lixc,ljmt,lkmt,
*d,cm4xu.294       dimension emi(1),ptr(1),pts(1),rng(1),scr(1),ixc(61,1),
      dimension emi(1),cond(1),ptr(1),pts(1),rng(1),scr(1),ixc(61,1),
*d,cm4xu.301,cm4xu.302       equivalence
(das,emi,ptr,pts,rng,scr,ixc,jmt,kdup,kmt,kxd,lbb)
*if -def,pointer,5
      equivalence (das,emi,cond,
     1 ptr,pts,rng,scr,ixc,jmt,kdup,kmt,kxd,lbb)
*d,cm4xu.306,cm4xu.307      pointer (kdy,emi),(kdy,ptr),(kdy,pts),(kdy,rng),(kdy,scr),
*if def,pointer,7
      pointer (kdy,emi),(kdy,cond),
     1 (kdy,ptr),(kdy,pts),(kdy,rng),(kdy,scr),
*i,lkon.12 blue
      if(jlock.eq.-1)call mp_setlock()
*i,lkoff.12 blue
      if(jlock.eq.1)call mp_unsetlock()
*d,bd.14,bd.17                                                    <422-425>
      data rktc/.001,.00125,.0015,.00175,.002,.0025,.003,.0035,.004,
     1 .0045,.005,.0055,.006,.007,.008,.009,.01,.0125,.015,.0175,.02,
     2 .025,.03,.035,.04,.045,.05,.055,.06,.07,.08,.09,.1,.125,.15,.175,
     3 .2,.25,.3,.35,.4,.45,.5,.55,.6,.7,.8,.9,.9999,40*1./
*/      data rkac/.9999,0.8,0.6,0.5,0.4,0.3,0.2,0.15,0.1,0.08,
*/     1   0.06,0.05,0.04,0.03,0.02,0.015,0.01,0.008,0.006,0.005,0.004,
*/     2   0.003,0.002,0.0015,0.001,0.0005,0.0002,0.0001,0.00005,
*/     3   0.00002,0.00001,0.000005,0.000002,0.000001,17*0.0/
*/      data  rktc/0.000001,0.0000015,0.000002,0.000003,0.000004,
*/     1 0.000005,
*/     1 0.000006,0.000008,0.00001,0.000015,0.00002,0.00003,0.00004,
*/     2 0.00005,0.00006,0.00008,0.0001,0.00015,0.0002,0.0003,0.0004,
*/     3 0.0005,0.0006,0.0008,0.001,0.00125,0.0015,0.00175,0.002,0.0025,
*/     4 0.003,0.0035,0.004,0.0045,0.005,0.0055,0.006,0.007,0.008,
*/     5 0.009,0.01,0.0125,0.015,0.0175,0.02,0.025,0.03,0.035,0.04,
```

```
*/      6 0.045,0.05,0.055,0.06,0.07,0.08,0.09,0.1,0.125,0.15,0.175,0.2,
*/      7 0.225,0.25,0.275,0.3,0.325,0.35,0.375,0.4,0.425,0.45,0.475,
*/      8 0.5,0.525,0.55,0.575,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,0.97,
*/      9 0.99,0.995,0.9995,1.0/
*/ Uncomment (delete 1st 3 columns) of next 2 lines to enable LAHEY F90
*/ Uncomment the following for DVF on PCDOS.          02/17/98  (GWM/GWM)
*/ *i,tt4a.3                                                          <989>
*/         use dfport,only:signal
*/ *d,tt4b.5,tt4a.28                                            <1012-1014>
*/ *if def,pcdos,1
*/         k=signal(2,pttyin,-1)
*ident its30
*d,jc4xu.1   for conductor specification                            <1548>
      parameter (nkcd=90,ntalmx=100,mopts=6)
*d,ib.72   enable reading numb option
      data cnm(54),(krq(i,54),i=1,7)/'phys ',0,1, 0,1,-1,  10,0/
*d,ib4a.10   for conductor specification                            <1677>
      data hmopt/'gas','estep','nlib','plib','elib','cond'/
*/ ------------------------------------------------------------ setdas
*d,sd.46                                                            <3348>
*/ el1 way used
*/       ns=8
      ns=mstp
      ledg=lech+mpng*mwng*(nee/ns+1)*nmat1*ke
*d,sd4b.13,sd4b.14 use the space regardless of plotting
      lfcdg=lx85+10*nmat1*nee
*d,sd.169   for conductor specification                             <3488>
      lcon=lemi+nmat1*ke
      lfdd=lcon+nmat1*ke
*i,ce4a.182
      if(bnum.ne.0.0.and.numb.ne.0)call erprnt(1,1,0,0,0,0,0,0,
     1 '"only one kind of brems biasing allowed"')
*d,ce.598  bbrem fix on mtop.
 1060 if(nwc.gt.49)go to 1070
*/ ------------------------------------------------------------ nextit
*d,ny4a.36   for conductor specification                            <5445>
      if(mlc.eq.6.and.nee.gt.0)cond(lcon+nmat)=ritm
      if(mlc.le.2.or.mlc.eq.6)go to 728
*ident bnum
*d,ny.443,ny.444                                                <5529-5530>
      if(nwc.eq.6)then
*/ allow for negative bnum for average energy loss calculation in brems.
*/ automatically reduce the subsequent electron production.  can be over-
*/ ridden by user with enum specification.
         bnum=ritm
         if(bnum.ne.zero)
     1 enum=one/abs(bnum)
      endif
      if(nwc.eq.10)numb=1
      if(ritm.lt.0..or.abs(1.-ritm).lt..00001)return
*ident its30
*d,ny.744,ny.745                                                <5830-5831>
 1480 if(nwc.le.49)bbrem(nwc)=1./ritm
      if(nwc.gt.49)mbi(lmbi+nwc-49)=-iitm
*d,oc4a.26,oc4a.28                                              <6306-6308>
  610 if(nwc.lt.49)call erprnt(2,1,2,nwc,49,0,0,0,
     1 '"too few entries:",i5," were read,",i5," are required."')
      m=49+max(kpt(3),kpt(2)*(1-ides))*nmat1
*/  ------------------------------------------------------------ getxst
```

```
*i,cor4-2.161
      if(nt.ne.0)lmb=mb/ndp2
*i,gt.213                                                          <16597>
*/ define an its30 flag only to ensure that its3.0 treatment is done
*/ in xsgen and colidp when nxs may have been zeroed.
      if(its30.eq.0)its30=nxs(lnxs+16,iex)
      if(its30.ne.nxs(lnxs+16,iex))call erprnt(2,1,0,0,0,0,0,
     1 1,'" electron tables are incompatible."')
*/
*/ ---------------------------------------------------------------- xsgen
*i,xg.11                                                          <17823>
*/      nwng=mwng
      nstp=mstp
*/      do 2 i=1,mbng
*/    2 rka(i)=rkac(i)
      if(its30.eq.3)go to 5
      ntop=49
*/ mwng_el1=(mtop_el1+1)/2
      nwng=25
      nstp=8
      do 1 i=1,mtop
    1 rkt(i)=rktc(i)
      do 3 i=1,nwng
    3 rka(i)=rkt(ntop-2*i+2)
    5 continue
*i,xg.17                                                          <17829>
c       .000579=(electron_radius)**2/fine_structure_constant*1e24
c       and z+1 has been replaced with z+eta, where eta accounts
c       for electron-electron bremsstrahlung.
c       "cross sections for bremsstrahlung production and electron-
c       impact ionization," s. m. selzter, in monte carlo transport
c       of electrons and photons, t. m. jenkins, w. r. nelson, and
c       a. rindi, eds.
      f=0.00579
      if(nxs(lnxs+16,j).eq.3)f=crelec*crelec/fscon*1.d24
*d,xg.18,xg.21   its3.0 radiation yields.                          <17831>
*/ set up this way to allow for tracking the test problems.
      if(nxs(lnxs+16,j).eq.3)go to 11
      do 10 i=1,nxs(lnxs+3,j)
      exs(jxs(ljxs+2,j)+i-1)=log(exs(jxs(ljxs+2,j)+i-1)/gpt(3))
   10 exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)=
     1 .000579*z*(z+1.)*exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)
      go to 18
   11 do 12 i=1,nxs(lnxs+3,j)
      exs(jxs(ljxs+2,j)+i-1)=log(exs(jxs(ljxs+2,j)+i-1)/gpt(3))
      f1=exs(jxs(ljxs+2,j)+2*nxs(lnxs+3,j)+i-1)
   12 exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)=
     1 f*z*(z+f1)*exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)
      if(nxs(lnxs+9,j).gt.mtop)call erprnt(1,1,2,nxs(lnxs+9,j),mtop,0,
     1 0,1,'" nxs(9) greater than mtop."')
*/ load values into rkt, rka, ntop, nwng only once.
*/ if exs is sacrosanct in the future this code can all be expunged.
      if(ntop.ne.0)go to 35
      ntop=nxs(lnxs+9,j)
      nwng=nxs(lnxs+10,j)
      if(nwng.gt.34)call erprnt(1,1,2,nxs(lnxs+10,j),34,0,
     1 0,1,'" nxs(10) greater than 34."')
*/ perform a linear interpolation on bbrem for bins one cannot specify.
*/ interpolate on the low end of the biasing input.  store values in
```

```
*/ eba temporarily and then reload them and zero eba again.
      if(bbrem(1).eq.zero)go to 16
      n7=max(1,ntop-47)
      do 14 i=ntop,n7,-1
   14 eba(leba+i,1)=bbrem(49-ntop+i)
      n8=max(1,ntop-48)
      do 15 i=n8,1,-1
      eba(leba+i,1)=bbrem(1)+(bbrem(2)-bbrem(1))*
     1 (exs(jxs(ljxs+9,j)+i-1)-exs(jxs(ljxs+9,j)))/
     1 (exs(jxs(ljxs+9,j)+n8)-exs(jxs(ljxs+9,j)))
   15 if(eba(leba+i,1).le.zero)eba(leba+i,1)=max(bbrem(1),eba(leba+1,1))
c
c        load photon energy ratios from database into fixed arrays.
c        also load in interpolated bbrem values if needed.
   16 do 17 i=1,nxs(lnxs+9,j)
      rkt(i)=exs(jxs(ljxs+9,j)+i-1)
      if(i.le.nxs(lnxs+10,j))rka(i)=exs(jxs(ljxs+10,j)+i-1)
      if(bbrem(1).eq.zero)go to 17
      bbrem(i)=eba(leba+i,1)
      eba(leba+i,1)=zero
   17 continue
      go to 35
   18 continue
*i,xg.27                                                          <17839>
   35 continue
*i,xg.92                                                          <17892>
*/ istern is a flag for the sbs density effect treatment and if
*/ non-zero is the memory offset.
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3)istern=1
*/ bug repair
*d,xg.122       cn=min(iz,2)*fme(lfme+jm)/(a*3.)
      cn=zero
      do 165 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
  165 if(iz.eq.iza(liza+j)/1000)cn=cn+fme(lfme+j)
      cn=min(iz,2)*cn/(a*3.)
*d,xg.135   correct xnum bug which precludes turning electron impact
*/ ionzation k-xrays off.
      if(xnum.ge.zero)xnm(lxnm+mkc)=xnum
*i,xg.139
*/ to become ink(86)
      if(ink(85).eq.0)go to 180
      af=avgdn/aa
      write(iuo,'(/29helectron secondary production,75x,
     1 14hprint table 86/)')
      write(iuo,'(
     1 9x,6henergy,11x,14hstopping power,11x,5hbrems,3x,9hthick tgt,3x,
     2 7hk x-ray,3x,8hknock-on/
     3 4x,1hn,14x,19hcollision radiation,3x,5htotal,15x,5hbrems,16x,
     4 /10x,3hmev,6x,29hmev barn  mev barn  mev barn ,3x,5hbarn ,
     5 16x,4hbarn,6x,4hbarn)')
      do 175 n=nee-1,1,-1
      nk=n+nee*(mkc-1)
  175 write(iuo,176)n,eee(leee+n),xse85(lx85+1,nk)/af,
     1 xse85(lx85+2,nk)/af,xse85(lx85+3,nk)/af,pbr(lpbr+nk),
     2 pbt(lpbt+nk),pxr(lpxr+nk),pkn(lpkn+nk)
  176 format(i5,1pe12.4,2(1x,5e10.3))
*i,xg.140                                                         <17965>
      if(its30.eq.3)rka(1)=one
*/ debug prints remove for integration.
```

```
      close(17)
      close(18)
      close(19)
*ident bnum
*i,xg.142   set secondary electron to 1% arbitrarily for numb biasing.
      if(numb.eq.1.and.enum.eq.1.)enum=1./100.
*i,xg.145
      if(numb.eq.1)write(iuo,195)
  195 format(' bremsstrahlung generated on each electron substep.',
     1 /,' secondary electron production reduced by 99%.')
*/  ------------------------------------------------------------ ronge
*i,rn4a.4                                                        <18709>
c
c        set up some temporary dynamically allocated arrays for density
c        effect calculation.
      dimension fo(1),eq(1),xq(1)
*if -def,pointer,1
      equivalence (das,fo,eq,xq)
*if def,pointer,1
      pointer (kdy,fo),(kdy,eq),(kdy,xq)
*d,rn.11,rn.12                                                <18728-18729>
c       obtain an average density for a cell with the same material but
c       different densities.
      rh=zero
      nh=0
      ic=0
      do 10 i=1,mxa
      if(mat(lmat+i).ne.mkc)go to 10
      if(ic.eq.0)ic=i
      rh=rh+rho(lrho+i)
      nh=nh+1
   10 continue
      if(nh.eq.0.or.rh.eq.zero)
     & call expirx(0,'ronge','zero density or non-existent material')
      rh=rh/nh
*d,rn4b.16      f=1.13
      iz=iza(liza+jmd(ljmd+mkc))/1000
      do 15 j=jmd(ljmd+mkc)+1,jmd(ljmd+mkc+1)-1
   15 if(iz.ne.iza(liza+j)/1000)go to 16
      go to 21
   16 f=1.13
*d,rn.23,rn.30                                                <18753-18759>
c        az is zoa in the notation of sternhiemer, berger, seltzer.
*/ using this locution gives better agreement with its3.0 instead of crelec.
      cr=planck*slite/(2d0*pie*fscon*gpt(3))
      pl=2d+18*cr*gpt(3)*fscon*sqrt(pie*cr*rh*az)
      su=2.d0*log(sr/pl)
      if(istern.ne.0)go to 40
c
c        sternheimer and peierls treatment.
c        cb=su+1.
      cb=2.*log(sr/(37.1*sqrt(az*rho(lrho+ic))))+1.
      if(emi(lemi+mkc).eq.0.)go to 31
      xb=4+min(1,int(cb/12.25))
      xa=max(.1*one*min(20,max(16,int(2.*cb)-3)),.326*cb-2.5)
      go to 49
   31 xb=2+min(1,int(sr/100.))
      xa=max(.2+zero,.326*cb-.5*xb)
      go to 49
```

```
c
c         sternheimer, berger, and seltzer treatment (sbs):
c         r. m. sternheimer, s. m. seltzer, and m. j. berger,
c         "density effect for the ionization loss of
c         charged particles in various substances", phys rev b, 26, 6067(1982).
   40 lr=0
      l1=0
      l2=0
      do 41 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      ns=nxs(lnxs+11,lme(llme+3,j))
      zs=exs(jxs(ljxs+11,lme(llme+3,j))+ns-1)
      if((cond(lcon+mkc).eq.zero).and.(zs.gt.zero))l1=l1+1
      if((cond(lcon+mkc).gt.zero).and.(zs.lt.zero))l2=l2+1
   41 lr=lr+ns
c
c         check conductor or non-conductor assignment.  non-conductor
c         if assigned by cond<0, or =0 and at least one non-conducting
c         component.  conductor if assigned cond>0 and at least one
c         conducting component, or cond=0 and no non-conducting
c         components.
      if(cond(lcon+mkc).ne.zero)go to 42
      if(l1.ne.0)go to 47
      cond(lcon+mkc)=one
      call erprnt(1,2,1,mkc,0,0,0,0,
     & '"material ",i4," has been set to a conductor"')
      go to 47
   42 if(l2.ne.0)go to 47
      if(cond(lcon+mkc).eq.-one)go to 43
      call erprnt(1,2,1,mkc,0,0,0,0,
     & '"material" ,i4," has been set to a non-conductor"')
      call erprnt(1,2,0,0,0,0,0,0,
     & '"no conduction electrons available"')
   43 cond(lcon+mkc)=zero
c
c         add memory for temporary arrays.
   47 if(istern.eq.1)istern=lmb
      if(lmb-istern.gt.3*lr)go to 48
      l7=(3*lr+istern)*ndp2
      if(lfll.lt.l7)call chgmem(mdas,lfll,l7,'ronge')
      lmb=l7/ndp2
   48 ld=istern
      call den1(az,su,pl,fo(ld+1),eq(ld+1+lr),xq(ld+1+2*lr),nt,wt,tm)
      if(nt.gt.lr)call erprnt(1,1,2,lr,nt,0,0,0,
     1 '"memory problem ",i4," allocated ",i4," used,"')
   49 continue
*i,rn.45
*d,rn.48,rn.49
      in=1
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)bl(k)=bl(k)+
     1 qpol(log(s),exs(jxs(ljxs+2,ix)),exs(jxs(ljxs+2,ix)+
     1 nxs(lnxs+3,ix)),nxs(lnxs+3,ix))*fme(lfme+j)*gpt(3)*(s+1.)
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3)bl(k)=bl(k)+
     1 spol(log(s),exs(jxs(ljxs+2,ix)),exs(jxs(ljxs+2,ix)+
     1 nxs(lnxs+3,ix)),nxs(lnxs+3,ix),in)*fme(lfme+j)*
     1 gpt(3)*(s+1.)
   60 continue
*d,rn.53,rn.57                                                  <18782-18786>
      d=zero
      if(istern.eq.0)go to 65
```

```
      call den2(s,tm,fo(ld+1),eq(ld+1+lr),xq(ld+1+2*lr),nt,wt,d)
      go to 69
   65 x=.4342945*log(sqrt(s*(s+2.)))
      if(x.gt.xa)d=max(zero,4.606*x-cb+(cb-4.606*xa)*
     1 ((xb-min(x,xb))/(xb-xa))**3)
   69 if(k.eq.1)dd=d
*d,rn4b.62,rn4b.66
   80 nk=n+nee*(mkc-1)
      xse85(lx85+1,nk)=el(1)*af
      xse85(lx85+2,nk)=bl(1)*af
      xse85(lx85+3,nk)=(el(1)+bl(1))*af
      if(jovr(1)+jovr(3)+jovr(5).ne.3)go to 85
*i,rn4b.68 load correct bs                                          <18842>
      bs=e*(e+2.*gpt(3))/(e+gpt(3))**2
*d,rn4b.70                                                          <18844>
      xse85(lx85+7,nk)=c1*af*az*dd/bs
*d,rn4b.77                                                          <18856>
     1 1h.,2x,24hmean ionization energy =,1pe12.5,4h ev./)
      if(istern.eq.0.or.nz.ne.1)go to 94
      write(iuo,'('' density effect data'')')
      if(cond(lcon+mkc).eq.zero)write(iuo,'('' non-conductor'')')
      if(cond(lcon+mkc).gt.zero)write(iuo,'('' conductor'')')
      do 91 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      iz=iza(liza+j)/1000
      ns=nxs(lnxs+11,lme(llme+3,j))
      write(iuo,'(''    z =''',i4)') iz
      write(iuo,'(''     occ no, be(ev) pairs'')')
   91 write(iuo,'(6(6x,f3.0,f11.3))')
     & (exs(jxs(ljxs+11,lme(llme+3,j))+nn-1),
     & exs(jxs(ljxs+11,lme(llme+3,j))+ns+nn-1),
     & nn=1,ns)
      write(iuo,'(''        plas(ev)        wt        tmin(mev)'')')
      write(iuo,'(3x,3f12.5)') pl,wt,tm*gpt(3)
      write(iuo,'(/)')
   94 if(nz.eq.1)write(iuo,95)
   95 format(
*addfile ,rn                                                        <18873>
*deck den1
         subroutine den1(az,su,pl,fo,eq,xq,nt,wt,tm)
c
c        calculate oscillator parameters, fo, eq, xq, and the number
c        of oscillators, nt, based on shell data of carlson and the
c        plasma frequency, pl, and log of the ionization potential, su.
c        return the fitting parameter, wt, and minimum
c        normalized kinetic energy, tm, for use in den2.
c        this routine solves eq. (10) of r. m. sternheimer, s. m. seltzer,
c        and m. j. berger, "density effect for the ionization loss of
c        charged particles in various substances", phys rev b, 26, 6067(1982).
*call cm
      dimension fo(1),eq(1),xq(1)
      nt=0
      do 20 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      iz=iza(liza+j)/1000
      ns=nxs(lnxs+11,lme(llme+3,j))
      do 20 n=1,ns
      zs=exs(jxs(ljxs+11,lme(llme+3,j))+n-1)
      nt=nt+1
      fo(nt)=fme(lfme+j)*abs(zs)/az
      if((cond(lcon+mkc).ne.zero).and.(zs.le.zero))go to 10
```

```
      xq(nt)=(exs(jxs(ljxs+11,lme(llme+3,j))+ns+n-1)/pl)**2
      eq(nt)=2.d0*third*fo(nt)
      go to 20
   10 xq(nt)=zero
      eq(nt)=fo(nt)
   20 continue
c
c        find fitting parameter rho termed wt here following
c        sternheimer, berger and seltzer as implemented in its3.0.
      mt=0
      wt=2.d0
   30 xo=wt
      fu=zero
      dr=zero
      do 40 n=1,nt
      pa=wt*wt*xq(n)+eq(n)
      fu=fu+fo(n)*log(pa)
   40 dr=dr+fo(n)*2.d0*wt*xq(n)/pa
      wt=wt-(fu-su)/dr
      mt=mt+1
      if(mt.lt.25)go to 50
      call erprnt(1,2,1,mt,0,0,0,0,
     & 'i10,* iterations den1 fitting parameter has not converged.*')
c
c       when no fitting parameter is found, arbitrarily set it to
c       a small value.
      wt=1.e-4
      go to 70
   50 if(wt.gt.zero)go to 60
      wt=xo/2.d0
      go to 30
   60 if(abs(wt/xo-one).gt.1.d-6) go to 30
   70 w2=wt*wt
      do 80 l=1,nt
      xq(l)=w2*xq(l)
   80 eq(l)=xq(l)+eq(l)
c
c       find minimum normalized kinetic energy.
      tm=zero
      if(cond(lcon+mkc).gt.zero)return
      sm=zero
      do 90 l=1,nt
   90 sm=sm+fo(l)/xq(l)
      bq=one/(one+sm)
      tm=one/sqrt(one-bq)-one
      return
      end
*deck den2
      subroutine den2(s,tm,fo,eq,xq,nt,wt,d)
c
c        calculate the density effect, d, based on fitting parameter, wt
c        and oscillator parameters, fo, eq, xq, at normalized energy, s
c        greater than tm.  d=zero otherwise.
c        this routine solves eqs. (5) and (6) of sternheimer (1982).
*call cm
      dimension fo(1),eq(1),xq(1)
      d=zero
      if(s.le.tm)return
      mt=0
```

```
      re=one/(s*(s+2.d0))
      rc=(s+one)**2
      x=one/re
   10 xo=x
      fn=zero
      dv=zero
      do 20 n=1,nt
      fn=fn+fo(n)/(xq(n)+x)
   20 dv=dv-fo(n)/((xq(n)+x)**2)
      x=x-(fn-re)/dv
      mt=mt+1
      if(mt.lt.25)go to 30
c
c       when no fitting parameter is found, arbitrarily set it to
c       zero.
      call erprnt(1,2,1,mt,0,0,0,0,
     & 'i10,"iterations den2 fitting parameter has not converged."')
      call erprnt(1,2,0,0,0,0,0,0,
     & '"density effect correction set to zero."')
      x=zero
      go to 50
   30 if(x.ge.zero)go to 40
      x=xo/2.d0
      go to 10
   40 if(abs(x/xo-one).gt.1.d-6) go to 10
   50 d=-x/rc
      do 60 l=1,nt
   60 d=d+fo(l)*log(one+x/eq(l))
      return
      end
*ident its30
*/ ------------------------------------------------------------------ brem
*i,bm.6                                                              <19130>
c
c         set up some temporary dynamically allocated arrays for
c         bremsstrahlung production calculation.
      dimension xs(1),x2(1),ti(1),xi(1)
*if -def,pointer,1
      equivalence (das,xs,x2,ti,xi)
*if def,pointer,1
      pointer (kdy,xs),(kdy,x2),(kdy,ti),(kdy,xi)
*i,bm.8                                                              <19132>
      data ld/0/
      save ld
*i,bm.11                                                             <19135>
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)then
*d,bm.23                                                             <19147>
      do 100 k=2,ntop
*d,bm.61                                                             <19185>
      pbr(lpbr+n+nee*(mkc-1))=eba(leba+ntop,n+nee*(mkc-1))/30
*d,bm.64                                                             <19188>
      do 110 k=2,ntop
*d,bm.66                                                             <19190>
    1 eba(leba+ntop,n+nee*(mkc-1))
      else
c
c         bremsstrahlung production cross sections based upon the evaluation
c         described in:
c         "cross sections for bremsstrahlung production and electron-
```

```
c         impact ionization," s. m. selzter, in monte carlo transport
c         of electrons and photons, t. m. jenkins, w. r. nelson, and
c         a. rindi, eds.  algorithm implementation based upon its3.0 and
c         fit values from its3.0 database.
c         set up dynamic memory.  use the values for the first zaid
c         since they are equal.
      ix=lme(llme+3,jmd(ljmd+mkc))
      nm=nxs(lnxs+5,ix)
      km=nxs(lnxs+6,ix)
      lr=(km+nm+(nm+nee)*km)
      if(ld.eq.0)call chgmem(mdas,lfll,(lmb+lr)*ndp2,'brem')
      if(ld.eq.0)lmb=lmb+lr
      ld=lmb-lr
      lx=ld
      lt=lx+km
      ls=lt+nm
      l2=ls+nm*km
      do 320 n=1,lr
  320 xs(ld+n)=zero
      do 330 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      ix=lme(llme+3,j)
      z=nxs(lnxs+2,ix)
      do 330 n=1,nm
      ti(lt+n)=log(exs(jxs(ljxs+5,ix)+n-1))
      do 330 k=1,km
      xs(ls+n+(k-1)*nm)=xs(ls+n+(k-1)*nm)+
     & z*z*exs(jxs(ljxs+5,ix)+km+n-1+k*nm)*fme(lfme+j)
  330 continue
      do 340 k=1,km
      xi(lx+k)=log((one-exs(jxs(ljxs+5,ix)+nm+k-1))+1.d-6)
      do 340 n=1,nm
  340 xs(ls+n+(k-1)*nm)=log(xs(ls+n+(k-1)*nm))
      do 350 k=1,km
      in=1
      do 350 n=1,nee
      tl=log(eee(leee+n))
      bl=log(one-(gpt(3)/(eee(leee+n)+gpt(3)))**2)
*/ quadratic fit if spol is not desired.
*/  350 x2(l2+k+(n-1)*km)=qpol(tl,ti(lt+1),xs(ls+1+(k-1)*nm),nm)
  350 x2(l2+k+(n-1)*km)=spol(tl,ti(lt+1),xs(ls+1+(k-1)*nm),nm,in)
     & -bl
      do 390 n=1,nee
      eba(leba+1,n+nee*(mkc-1))=zero
      in=1
      do 380 k=2,ntop
      de=rkt(k)-rkt(k-1)
      do 360 np=1,11
      ek(np)=rkt(k)-real(np-1)*de*1.d-1
      xp=log((one-ek(np))+1.d-6)
*/ quadratic fit if spol is not desired.
*/  360 g(np)=exp(qpol(xp,xi(lx+1),x2(l2+1+(n-1)*km),km))
  360 g(np)=exp(spol(xp,xi(lx+1),x2(l2+1+(n-1)*km),km,in))
      do 370 i=1,11
  370 g(i)=g(i)/ek(i)
  380 eba(leba+k,n+nee*(mkc-1))=
     & (g(1)+4.d0*(g(2)+g(4)+g(6)+g(8)+g(10))+
     1 2.d0*(g(3)+g(5)+g(7)+g(9))+g(11))*de+
     2 eba(leba+k-1,n+nee*(mkc-1))
c
```

```
c        30=3*(number of integration points -1)
      pbr(lpbr+n+nee*(mkc-1))=eba(leba+ntop,n+nee*(mkc-1))/30.d0
c
c         normalize the cumulative unbiased probability, eba.
      do 390 k=2,ntop
  390 eba(leba+k,n+nee*(mkc-1))=
     & eba(leba+k,n+nee*(mkc-1))/eba(leba+ntop,n+nee*(mkc-1))
      endif
*/ write the cross sections for validation and debug
      do 601 n=1,nee
      write(17,*)eee(leee+n),pbr(lpbr+n+nee*(mkc-1))
      write(18,*)' new group ',eee(leee+n)
      do 601 k=1,ntop
601   write(18,*)rkt(k),eba(leba+k,n+nee*(mkc-1))
*d,bm.75                                                          <19199>
      do 120 k=2,ntop
*d,bm.80                                                          <19204>
    1 q=qpol(e/eee(leee+j),rkt,eba(leba+1,j+nee*(mkc-1)),ntop)
*d,bm.86                                                          <19210>
      do 130 k=2,ntop
*d,bm.88,bm.89                                            <19212-19213>
    1 ebt(lebt+ntop,n+nee*(mkc-1))
      do 140 k=1,ntop
*d,bm.98                                                          <19222>
      do 150 k=2,ntop
*d,bm.103                                                         <19227>
      do 160 k=1,ntop-1
*d,bm.108                                                         <19232>
      do 170 k=2,ntop
*d,bm.113                                                         <19237>
      do 180 k=1,ntop-1
*/
*/ ------------------------------------------------------------ brang
*i,bg.7    correct possible parameter and constant mismatch                <19273>
      save ih,n1,n2,r2
      data ih/0/
*i,bg.8    correct possible parameter and constant mismatch                <19274>
      if(ih.ne.0.or.nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)go to 2
      ih=1
      n1=(mpng+1)/2
      n2=n1-1
      r2=float(n2)
      do 1 i=1,n1
    1 cm(i)=(i-1)/r2
    2 continue
*d,bg.13                                                          <19279>
      do 150 n=1,nee,nstp
*d,bg.15,bg.16                                            <19281-19282>
      do 20 k=1,nwng
   20 ek(k)=rka(k)*s
*d,bg.25,bg.27   correct possible parameter and constant mismatch    <19291-19293>
      xi(l)=(1.+.1*b*(11-l))/(1.+b)
      ch(l)=(11-l+10.*b)/(10.+b*(11-l))
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)go to 30
      xi(l)=(one+b*(n1-l)/r2)/(one+b)
      ch(l)=(n1-l+r2*b)/(r2+b*(n1-l))
   30 continue
      do 150 k=1,nwng
*d,bg.104                                                         <19370>
```

```
  150 ech(lech+1,k,n/nstp+1+(nee/nstp+1)*(mkc-1))=min(max(zero,
*i,bg.105    write the cross sections for validation            <19371>
      do 510 n=1,nee,nstp
      do 510 k=1,nwng
      do 510 l=1,mpng
  510 write(19,*)n,k,l,ech(lech+1,k,n/nstp+1+(nee/nstp+1)*(mkc-1))
*i,mr.6 blue
      integer*4 mz
      mz=ntasks
      call mp_set_numthreads(mz)
*i,mr.92 blue
C$PAR PARALLEL DO
      do 141 i=1,ntasks
  141 call trnspt
*/ get rid of incredibly huge number of restart statements
*i,tn4xu.14        if(t2.ge.cp2(1)-1.)go to 50
      if(nrc.gt.50)then
         nrc=nrc+1
         go to 60
      endif
*d,ut.27 blue
      mz=mp_my_threadnum()
      ktask=mz
*/
*/ ------------------------------------------------------------ colidp
*ident flaug
*b,cp.163                                                      <26798>
      if(its30.ne.3)then
*i,cp4a.16                                                     <26874>
      else
      er=erg
      ie=iex
      ik=ipt
      call flaug(er,ie,mk,ik)
      endif
*addfile ,cp                                                   <26916>
*deck flaug
      subroutine flaug(er,ie,mk,ik)
c
c        a subroutine to generate fluourescent photons or auger
c        electrons to fill vacancies.
c        called by colidp and kxray.
c        er=energy of absorbed photon or edge energy for kxray
c        ie=element that is undergoing the interaction; for kxray it is
c        the highest z element.
c        mk=material and ik=ipt for kxray(3) or colidp(2) for keeping
c        track of parent particle for bookkeeping and vr.
*call cm
      ipt=2
      ntyn=3
      rp=0.
      if(ik.eq.2)go to 5
      vel=slite
      ncp=0
      idx=0
c
c       check for electron emission from impact ionzation.
      if(rang().gt.wwk(lwwk+mk))go to 80
    5 if(dbcn(20).ne.0.)rp=1.
```

```
      if(nxs(lnxs+2,ie).le.11)go to 90
      i1=jxs(ljxs+4,ie)+2
      if(er.le.xss(i1-1))go to 90
      do 10 ka=i1,i1-3+nxs(lnxs+4,ie)
   10 if(er.le.xss(ka))go to 20
   20 ka=ka+nxs(lnxs+4,ie)-1
      k0=ka
   25 ka=k0
      t3=xss(ka)*rang()
      ka=ka+nxs(lnxs+4,ie)
      if(ik.eq.3.and.t3.ge.xss(ka))go to 25
      if(ik.eq.2.and.t3.ge.xss(ka))go to 90
      rp=1.
   30 ka=ka-1
      if(t3.lt.xss(ka))go to 30
c
c        at least single fluorescence or auger electron.
      eg=xss(ka-2*nxs(lnxs+4,ie)+1)
      if(kpt(3).eq.0)go to 40
      if(ik.eq.2.and.rang().gt.wwk(lwwk+mk))go to 80
   40 e=xss(nxs(lnxs+4,ie)+ka+1)
      if(e.lt.elc(2))go to 90
      if(ik.ne.2)go to 41
      paxtc(4,12,2)=paxtc(4,12,2)+1.
      paxtc(5,12,2)=paxtc(5,12,2)+wgt
      paxtc(6,12,2)=paxtc(6,12,2)+wgt*erg
      pwb(kpwb+2,18,icl)=pwb(kpwb+2,18,icl)-wgt
      pan(kpan+2,3,mpan)=pan(kpan+2,3,mpan)+wgt
   41 ed=0.
      if(nxs(lnxs+2,ie).le.30)go to 60
      i=ka+1-2*nxs(lnxs+4,ie)
      if(i.ne.i1.and.i.ne.i1+1)go to 60
      i=i1-2+nxs(lnxs+4,ie)
      ka=nxs(lnxs+4,ie)+i+1
      if(rang().ge.xss(ka)/(1.-xss(i)))go to 60
c
c        double fluorescence.  bank the second photon.
      erg=xss(nxs(lnxs+4,ie)+ka)
      if(erg.lt.elc(2))go to 60
      if(ik.ne.2)go to 42
      paxtc(1,16,2)=paxtc(1,16,2)+1.
      paxtc(2,16,2)=paxtc(2,16,2)+wgt
      paxtc(3,16,2)=paxtc(3,16,2)+wgt*erg
      pwb(kpwb+2,17,icl)=pwb(kpwb+2,17,icl)+wgt
      go to 43
   42 paxtc(1,14,2)=paxtc(1,14,2)+1.
      paxtc(2,14,2)=paxtc(2,14,2)+wgt
      paxtc(3,14,2)=paxtc(3,14,2)+wgt*erg
      pwb(kpwb+2,16,icl)=pwb(kpwb+2,16,icl)+wgt
   43 ntyn=4
      call isos(uuu,lev)
      sw=wgt
      if(ik.ne.2)go to 44
      if(abs(wwp(2,4)).eq.1..and.(idx.eq.0.or.wwp(2,5).lt.0.))
     1 call wtwndo(4,ww)
      if(nter.ne.0)go to 50
      if(espl(2,1).ne.0.)call ergimp
      if(nter.ne.0)go to 50
   44 npa=1
```

```
      call bankit(9)
      ed=erg
      wgt=sw
      go to 60
   50 paxtc(4,nter,2)=paxtc(4,nter,2)+1.
      paxtc(5,nter,2)=paxtc(5,nter,2)+wgt
      paxtc(6,nter,2)=paxtc(6,nter,2)+wgt*erg
      pwb(kpwb+2,jrwb(nter,2),icl)=pwb(kpwb+2,jrwb(nter,2),icl)-wgt
      nter=0
   60 if(ides.eq.0.and.ik.eq.2.and.eg0-eg-ed.gt.elc(3))
     & call emaker(1,eg0-eg-ed)
      erg=e
      ipsc=99+ntyn
      call isos(uuu,lev)
      if(ik.ne.2)go to 61
      paxtc(1,15,2)=paxtc(1,15,2)+1.
      paxtc(2,15,2)=paxtc(2,15,2)+wgt
      paxtc(3,15,2)=paxtc(3,15,2)+wgt*erg
      pwb(kpwb+2,17,icl)=pwb(kpwb+2,17,icl)+wgt
      go to 100
   61 continue
      if(jgp.le.-2)jgp=jgp+2
      paxtc(1,14,2)=paxtc(1,14,2)+1.
      paxtc(2,14,2)=paxtc(2,14,2)+wgt
      paxtc(3,14,2)=paxtc(3,14,2)+wgt*erg
      pwb(kpwb+2,16,icl)=pwb(kpwb+2,16,icl)+wgt
      do 70 i=1,ndx(2)
   70 if((xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+
     1 (zzz-dxx(2,3,i))**2.lt.dxx(2,5,i))idx=i
      if(ndet(2).ne.0)call tallyd
      if(kdb.ne.0)return
      if(ndx(2).gt.1.or.ndx(2).eq.1.and.idx.eq.0)call dxtran
      if(kdb.ne.0)return
      call bankit(18)
      go to 100
c
c        make electrons either auger or photoelectric.
c        set ipt=ik for bookkeeping.
   80 ipt=ik
      if(ides.eq.0.and.eek(leek+mk).gt.elc(3).and.
     1 (dbcn(20).ne.0..or.er.ge.edg(ledg+mk)))
     1 call emaker(4,eek(leek+mk))
   90 if(ides.eq.0.and.eg0-rp*edg(ledg+mk).gt.elc(3).and.
     & ik.eq.2)call emaker(1,eg0-rp*edg(ledg+mk))
      if(ik.ne.2)go to 100
      nter=12
      pan(kpan+2,3,mpan)=pan(kpan+2,3,mpan)+wgt
  100 continue
      return
      end
*/
*/ ------------------------------------------------------------ ttbr
*ident bnum
*d,tb.23   <27185>
      pr=pr*abs(bnum)
*d,tb.46   <27204>
      if(abs(bnum).gt.0..and.bnum.ne.1.)wgt=wgt/abs(bnum)
*ident its30
*d,tb.67,tb.68                                          <27223-27224>
```

```
      pk=qpol(rn,ebt(lebt+1,m+nee*(mkc-1)),rkt,ntop)
      pa=qpol(rn,ebt(lebt+1,m+1+nee*(mkc-1)),rkt,ntop)
*d,tb.73                                                          <27229>
      do 110 j1=2,ntop-1
*d,tb.75                                                          <27231>
  120 do 130 j=2,ntop-1
*/
*/ ------------------------------------------------------------- kxray
*ident flaug
*b,kx.24                                                          <28859>
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)then
*i,kx.51                                                          <28883>
      else
      if(iphot.ne.0.or.fiml(2).eq.0..or.eek(leek+mkc).le.elc(2))
     1 go to 80
c
c        assume the electron has enough energy to excite the atom
c        above the threshold of the highest K shell energy of the
c        highest Z atom of the material.
   71 jm=jmd(ljmd+mkc)
      do 72 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
   72 if(iza(liza+j).gt.iza(liza+jm))jm=j
      ix=lme(llme+2,jm)
      iex=ix
      mk=mkc
*/      er=xss(jxs(ljxs+4,ix)+nxs(lnxs+4,ix)-1)*1.00001
c
c        1.0001 assures round-off does not give spurious results
      er=edg(ledg+mkc)*1.0001
      if(er.gt.eg0)er=eg0
      ik=ipt
      call flaug(er,ix,mk,ik)
      endif
*/
*/ ------------------------------------------------------------- brems
*ident its30
*i,bb.6
c
c        el is dimensioned to keep track of first, possibly real event
c        energy loss, and second and higher order events to keep track of
c        energy loss only for production purposes.
c        ib flags real events in conjunction with numb vr use.
      dimension el(2)
*d,bb.11      if(pr+rang().lt.1.)return
      r1=rang()
      p1=exp(-pr)
      ib=1
      if((nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0
     1 .and.pr+r1.lt.1.).or.
     1 (nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3.and.r1.lt.p1))ib=0
      if(numb+ib.eq.0)return
c
c        set up to produce none, one, or more photons.
      npa=1
      nb=1
c
c        do splitting or roulette if bnum biasing is used.
      if(bnum.ne.one.and.dbcn(23).ne.zero)then
         nb=abs(bnum)+rang()
```

```
          if(nb.eq.0)then
              npa=0
              nb=1
          endif
       endif
       wb=wgt
       if(bnum.ne.zero.and.dbcn(23).ne.zero)
      1 wb=wgt/abs(bnum)
       rb=nb
       w0=wb
       if(numb.eq.1)wb=wb*(one-p1)
       ps=p1
       do 240 n1=1,nb
       ie=min(n1,2)
       el(ie)=zero
       if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0)go to 5
       p1=ps
       ap=p1
       fa=zero
*d,bb.14                                                          <28908>
    5 npb=npb+1
*d,bb.21,bb.23    bnum                                      <28915-28916>
*i,bb.25          bnum                                            <28918>
       wgt=wb
*ident its30
*d,bb4a.4
       es=eg0-qo*dk-el(ie)
       if(es.lt.elc(3))go to 200
*d,bb.40,bb.41                                              <28930-28931>
       pk=qpol(rn,eba(leba+1,m+nee*(mkc-1)),rkt,ntop)
       pa=qpol(rn,eba(leba+1,m+1+nee*(mkc-1)),rkt,ntop)
*i,bb.43          debug print
c       write(20,*)erg
*ident bnum
*d,bb.46    bnum                                                  <28936>
c
c         accumulate radiative energy loss for first photon or all
c         of them for bnum<0.
       el(ie)=el(ie)+erg
       if(n1.gt.1.and.bnum.lt.zero)el(1)=el(1)+erg
*d,cor4-1.163    bnum                                             <28939>
       if(iphot.ne.0.or.fiml(2).eq.0..or.es.lt.elc(2).or.
      & npa.eq.0)go to 200
*ident its30
*d,bb.54,bb.55                                              <28944-28945>
       pk=qpol(rn,ebd(lebd+1,m+nee*(mkc-1)),rkt,ntop)
       pa=qpol(rn,ebd(lebd+1,m+1+nee*(mkc-1)),rkt,ntop)
*d,bb.57                                                          <28947>
       do 60 j1=2,ntop-1
*d,bb.59                                                          <28949>
   70 do 80 j=2,ntop-1
*ident bnum
*d,bb.71,bb.74   bnum                                       <28961-28964>
       if(bnum.eq.1.)go to 110
       if(dbcn(23).ne.zero)go to 110
       npa=bnum+rang()
       if(npa.eq.0)go to 200
       wgt=wgt/bnum
*ident its30
```

```
*d,bb.89,bb.105                                                    <28978-28994>
      k=(m-1)/nstp+1
      ek=eee(leee+(k-1)*nstp+1)-es
      el=es-eee(leee+k*nstp+1)
      do 140 j=2,nwng
  140 if(pj.ge.rka(j))go to 150
  150 rj=rka(j-1)-pj
      sj=pj-rka(j)
      l=rn*(mpng-1)
      pl=rn*(mpng-1)-1
      ql=1.-pl
      i=lech+l+1
      mc=k+(nee/nstp+1)*(mkc-1)
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0)then
      cx=4.*(ql*sj*el*ech(i,j-1,mc)+pl*rj*ek*ech(i+1,j,mc+1)+
     1 pl*sj*el*ech(i+1,j-1,mc)+ql*rj*el*ech(i,j,mc)+
     2 ql*sj*ek*ech(i,j-1,mc+1)+ql*rj*ek*ech(i,j,mc+1)+
     3 pl*sj*ek*ech(i+1,j-1,mc+1)+pl*rj*el*ech(i+1,j,mc))/
     4 ((rka(j-1)-rka(j))*eee(leee+(k-1)*nstp+1))
      else
*/ for exact its3.0 emulation.
*/       rx=one-rn
*/       l=rx*(mpng-1)
*/       pl=rx*(mpng-1)-1
*/       ql=one-pl
*/       i=lech+l+1
      cx=2.d0*(ql*sj*el*ech(i,j-1,mc)+pl*rj*ek*ech(i+1,j,mc+1)+
     1 pl*sj*el*ech(i+1,j-1,mc)+ql*rj*el*ech(i,j,mc)+
     2 ql*sj*ek*ech(i,j-1,mc+1)+ql*rj*ek*ech(i,j,mc+1)+
     3 pl*sj*ek*ech(i+1,j-1,mc+1)+pl*rj*el*ech(i+1,j,mc))/
     4 ((rka(j-1)-rka(j))*(eee(leee+(k-1)*nstp+1)-eee(leee+k*nstp+1)))
      endif
*i,bb.106     debug and verification
c       write(20,*)cx,am
*ident bnum
*d,cor4-1.167          bnum                                        <29052>
      if(dbcn(23).eq.zero.and.bnum.gt.1.)wgt=wgt*bnum
*ident its30
*d,bb.152,bb.154  erg=erg-el
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0.or.ib.eq.0)
     1 go to 240
      if(ib.eq.1.and.numb.eq.1)wb=w0
      fa=fa+one
      pl=pl*pr/fa
      ap=ap+pl
      if(rl.ge.ap)go to 5
  240 continue
      if(ib.eq.0)return
      if(dbcn(23).ne.zero.and.bnum.lt.zero)
     1 el(1)=el(1)/rb
      erg=erg-el(1)
      paxtc(6,12,3)=paxtc(6,12,3)+wgt*el(1)
      if(el(1).ne.zero.and.
     1 nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3)then
         paxtc(4,12,3)=paxtc(4,12,3)+one
         paxtc(5,12,3)=paxtc(5,12,3)+wgt
      endif
      return
*addfile bb
```

```
*deck spol
c     subroutine spol (s,x,y,n,in,t), 15 jan 71.
      function spol (s,x,y,n,in)
*/      function spol (s,x,y,n,m)
c
c        cubic spline interpolation with parabolic runout.
*if def,cheap,1
      implicit double precision (a-h,o-z)
      parameter (ln=1000)
      save
      dimension  x(n),y(n),e(ln),u(ln)
      if(n.gt.ln)call erprnt(1,1,2,n,ln,0,0,1,
     1 '" too large an argument in spol, increase parameter ln"')
      if (in.ne.1) go to 30
      in=2
      n1=n-1
      e(1)=1.d0
      u(1)=0.d0
      b1=x(2)-x(1)
      c1=(y(2)-y(1))/b1
      do 10 j=2,n1
      b2=x(j+1)-x(j)
      c2=(y(j+1)-y(j))/b2
      b=x(j+1)-x(j-1)
      d=(c2-c1)/b
      c=b1/b
      b1=b2
      c1=c2
      p=c*e(j-1)+2.d0
      e(j)=(c-1.d0)/p
   10 u(j)=(d-c*u(j-1))/p
      e(n)=u(n1)/(1.d0-e(n1))
      do 20 kk=1,n1
      k=n-kk
   20 e(k)=e(k)*e(k+1)+u(k)
      id=0
      m2=0
      m1=n
      if (x(1) .le. x(n)) go to 30
      id=1
      m2=n
      m1=0
   30 if(s.lt.x(m1+id).and.s.gt.x(m2+1-id))go to 40
      if(s.ge.x(m1+id))mu=m1+2*id
      if(s.le.x(m2+1-id))mu=m2+2*(1-id)
      go to 60
   40 m1=m2
      mu=m1
   50 m3=(m1+mu)/2
      if(s.lt.x(m3))mu=m3
      if(s.ge.x(m3))m1=m3
      if(iabs(mu-m1).gt.1)go to 50
      mu=mu+id
c
   60 spol=(e(mu-1)*((x(mu)-s)**3)+e(mu)*((s-x(mu-1))**3)+
     1 (y(mu-1)-e(mu-1)*((x(mu)-x(mu-1))**2))*(x(mu)-s)+
     2 (y(mu)-e(mu)*((x(mu)-x(mu-1))**2))*(s-x(mu-1)))/
     3 (x(mu)-x(mu-1))
      return
```

```
      end
*/
*ident ge4b2
*/ *i,ge.4
*/       integer*4 i
*/ Uncomment the following for DVF on PCDOS.          02/17/98  (GWM/GWM)
*/ *i,ge.3                                                          <42172>
*/         use dfport,only:iargc
*/         use dflib,only:getarg
*/ *d,ge4a.8,ge4a.9                                          <42196-42197>
*/ *if def,pcdos,5
*/        hm=' '
*/        j=0
*/        do 10 i=1,iargc()
*/        call getarg(i,hm(j+1:))
*/     10 j=j+leng(hm(j+1:))+1
*/
*/ ------------------------------------------------------------- secnd
*ident se4b2
*/ Uncomment the following for DVF on PCDOS.          02/17/98  (GWM/GWM)
*/ *d,se4a.4,se4a.6                                          <42820-42822>
*/ *d,se4a.8,se4a.15                                         <42837-42844>
*/ *if def,pcdos,1
*/        call cpu_time(t)
```

Patch to 4B with a few parallel enhancements:

```
c          <<<<< makemcnp changes for MCNP4B2 >>>>>       loddat 09/22/97
c
c    For each fix, enter five integer parameters on one line in free
c    format, ip(1)-ip(5) described below, followed by the appropriate
c    number of new lines.
c
c    ip(1) = The applicable computer system number or 0 for all systems.
c    ip(2) = The file number:
c        1 = patchf
c        2 = makemcnp
c        3 = patchc
c    ip(3) = The applicable line number to edit-see the MCSETUP source.
c    ip(4) = The alter code:
c        1 = insert the following ip(5) line(s) before line ip(3)
c        2 = replace line ip(3) with the following ip(5) line(s)
c        3 = insert the following line after entry ip(5) of line ip(3)
c        4 = delete lines ip(3) through ip(5)
c    ip(5) = The number of new line(s) if ip(4)=1,2
c            The entry number if ip(4)=3
c            The last line number to delete if ip(4)=4
c
c    The number of new lines that follow ip(1)-ip(5) is given by:
c          = ip(5) if ip(4)=1,2
c          = 1 if ip(4)=3
c          = 0 if ip(4)=4

c******************** FIXES FOR SYSTEM 1 (Cray UNICOS) *****************
c
c     Add the *define t3d for the Cray T3D.
c      1 1 2 1 1
c     *define t3d
c
c     Provide links to the T3D compilers and linker. XTM:GWM-95-128
c     Change the CFT77 compile line.
c     1 2 18 1 4
c     export TARGET; TARGET=CRAY-T3D
c     ln -s /mpp/bin/cc cc
c     ln -s /mpp/bin/cft77 cft77
c     ln -s /mpp/bin/mppldr segldr
c     Provide links to the T3D compilers and linker. XTM:GWM-95-128
c     1 2 26 2 1
c     cft77 compile
c
c     Add CFTLIB library on UNICOS for profile timing.    06/30/95 (GWM)
c     1 2 28 3 10
c      -L/usr/local/lib -lprof

c********************** FIXES FOR SYSTEM 2 (Sun  SunOS ) *****************

c     Add a second X-window include path for some SunOS systems.
c     2 2 18 3 4
c      -I/usr/openwin/include/X11

c     Add a link to the ANSI C library (libansi.a) on some SunOS systems.
c     2 2 29 3 10
c      -L/home -lansi
```

```
c********************** FIXES FOR SYM 3 (IBM  AIX   ) *****************
c
c     PVM libraries must go first for pvm version at some installations.
3 2 30 2 1
xlf -o mcnp *.o -L/usr/lanl/pvm3.3/lib -lfpvm3 -lpvm3 -L/usr/lib -lX11

c********************** FIXES FOR SYSTEM 4 (HP   HPUX  ) *****************
c
c     Add a link to the C math library (libM.a) on some HP systems.
c     4 2 29 3  6
c      -lM

c********************** FIXES FOR SYSTEM 5 (VAX  VMS   ) *****************
c
c     Add an additional *define keyword for PRPR on a VAX VMS system.
c      5 1 2 1  1
c     *define disscgs

c***************** FIXES FOR SYSTEM 6 (DEC UNIX and PC LINUX) ************
c
c     The following 7 lines enable GNU F77 compilation on LINUX.
c     Tim Goorley, jgoorley@mit.edu XTM-RN(U97-028) 7/21/97
c     6 1 2 1 1
c     *define linux
c     6 2 27 3 1
c      -O0
c     6 2 28 4 28
c     6 3 2 1 1
c     *define linux

c********************** FIXES FOR SYSTEM 7 (PC   DOS   ) *****************
c
c     The following lines enable dvf F90 compilation on DOS/WINDOWS.
7 2 4 2 59
if exist flib del flib\*.for
if exist olib del olib\*.obj
if exist compile del compile
if exist newid del newid
if exist patch del patch
copy prpr.id prpr.for
f90 prpr.for
del prpr.for
copy makxs.id codef
type patchf | find "*define" > patch
prpr
rename compile makxsf.for
f90 makxsf.for
del makxsf.for
del codef
del patch
del newid
type patchf | find "*define pcdos" | find "xlib"
if errorlevel 1 goto noxwin1
if exist mcnpc.c del mcnpc.c
copy mcnpc.id codef
copy patchc patch
prpr
rename compile mcnpc.c
```

```
del codef
del patch
del newid
hcx86 -f387 -DMSDOS -Hoff=protection -I\dvx\include -c mcnpc.c
:noxwin1
copy mcnpf.id codef
copy patchf patch
prpr
del codef
del patch
del newid
fsplit compile
if errorlevel 1 goto nosplit
del compile
f90 /compile_only /optimize:4 *.for
goto link
:nosplit
rename compile mcnp.for
lf90 -c -nw -o0 -mod \lf9040\lib mcnp.for
:link
move mcsetup.for mcsetup.id
if exist mcsetup.obj del mcsetup.obj
if not exist flib mkdir flib
move *.for flib
move mcsetup.id mcsetup.for
type patchf | find "*define pcdos" | find "xlib"
if errorlevel 1 goto noxwin2
set lib=\f7713\lib;\hcx3\small;\dvx\lib\hcx87
lf90 @automake.rsp \lf9040\lib\hcx20.obj mcnpc.obj -lib hcx86,hcx87,hcna,xll,sys -
nomap -exe mcnp.exe
goto end
:noxwin2
f90 /exe:mcnp.exe *.obj
:end
if not exist olib mkdir olib
move *.obj olib
7 2 5 4 52

c********************* fixes for system 8 (sun  sol.  ) *****************

c********************* fixes for system 9 (sgi  irix  ) *****************

9 1 2 1 1
*define multt
c
9 2 5 2 1
rm -rf compile newid patch codef flib olib
c
9 2 13 2 1
rm -f newid codef patch *.f *.o
c
9 2 14 2 1
ln -s mcnpc.id codef
9 2 15 2 1
ln -s patchc patch
9 2 18 2 1
cc -64 -g -c -I/usr/include mcnpc.c
c
9 2 19 2 2
```

```
rm -f codef patch
mv newid newidc
c
9 2 20 2 1
ln -s mcnpf.id codef
9 2 21 2 1
ln -s patchf patch
c
9 2 24 2 2
rm -f compile codef patch clog
mv newid newidf
9 2 27 2 1
f77 -64 -mp -O2 -TARG:madd=OFF -OPT:Olimit=3000 -c  *.f
9 2 28 2 1
f77 -64 -mp -O0 -TARG:madd=OFF -OPT:Olimit=3000 -c  trnspt.f
9 2 30 3 1
 -64 -mp
9 2 30 3 12
 \
9 2 31 1 2
 -Wl,-Xlocal,pblcom_  -Wl,-Xlocal,tskcom_  -Wl,-Xlocal,itskpt_


c    Delete lines 2-10 of PATCHF.
 0 1 2 4 10
c    Add this entire patch to PATCHF.
 0 1 11 2
*/ Integrate your patch with the following. If your patch makes changes
*/ to the ZC, VV, CM, GS, MB, or BD decks, contact MCNP@LANL.GOV for
*/ details on including your patch via the INSTALL.FIX file. When
*/ integrating your patch, be sure the order of the changes (increasing
*/ in line number) is preserved to avoid a PRPR error:  See Appendix
*/ C, page C-8 in the MCNP manual for more details.
*/
*ident its30
*d,zc4b.1                                                          <21>
      parameter (kod='mcnp',ver='4bnu')
*d,zc4b.6,zc4a.8                                                 <27-28>
     1 mink=200,mipt=3,mjsf=9,mkft=9,mktc=22,mlgc=1000,mpb=5,mpng=21,
     2 mseb=301,mspare=3,mtop=89,mwng=(mtop+1)/2,mstp=4,mbng=51,
     2 mxdt=20,mxdx=5,mxlv=10,
*d,zc4b.12                                                         <47>
*/ include the fundamental constant crelec=classical electron radius.
*/ crelec=planck*slite/(2d0*pie*fscon*gpt(3))
     9 euler=.57721566490153286 1d0,crelec=2.8179380d-13,
*d,vv4b.1,vv4a.2                                                 <55-56>
      common /tables/ gpt(mipt),qfiss(23),rktc(mtop),
*/ a fixed array if rka is not read from the database.
*/    1 rkac(mbng),
     1 talb(8,2),vco(mcoh),vic(minc),wco(mcoh),
*d,cm4b.2,cm4b.3                                                 <85-86>
      parameter (nfixcm=maxi+3*maxv+2*mtop+mipt*(24+mxdt+7*mxdx)+nsp+
     1 105+mbng,lfixcm=3*mxdt+mink+11*mipt+2*maxv+2*maxf+281+2+6)
*d,cm4b.5 blue
      parameter (nephcm=28+mcpu,lephcm=nptr+novr+ncolor+62)
*d,cm4b.6
      common /fixcom/ bbrem(mtop),bcw(2,3),bnum,calph(maxi),coincd,
     1 rkt(mtop),rka(mbng),
*d,cm.36                                                         <107>
```

```
     4 indt,ink(mink),iphot,iplt,ipty(mipt),isb,issw,istern,its30,
     1 istrg,
*d,cm4a.7                                                          <112>
     7 mtasks,nlat,nsrc,
     & ntop,nwng,nstp,numb,
*d,cm4b.26                                                         <132>
     4 lscr,ldrc,lemi,lcon,lmb,
     4 lfdd,lgnr,lpik,lptb,lptr,lpts,lrng,lrtc,ltgp,
*d,cm4b.34 blue
      common /ephcom/ cp0,cp1,cp2(mcpu),cp3,cpa,ctme,fpi,freq,
*d,cm4a.89                                                         <299>
      dimension scr(1),drc(16,1),emi(1),cond(1),
     & fdd(2,1),genr(1),pik(1),ptr(1),
*d,cm4b.137,cm4b.138                                               <304-305>
      equivalence (das,scr,drc,emi,cond,fdd,genr,pik,ptb,ptr,pts,rng,
     1 rtc,tgp,ifl,ipac2,ixc,jfl,jft,jmt,kmt,ktc,kxd,lbb)
*d,cm.276,cm4b.141                                                 <307-311>
      pointer (kdy,scr),(kdy,drc),(kdy,emi),(kdy,cond),(kdy,fdd),
     1 (kdy,genr),(kdy,pik),(kdy,ptb),(kdy,ptr),(kdy,pts),(kdy,rng),
     2 (kdy,rtc),(kdy,tgp),(kdy,ifl),(kdy,ipac2),(kdy,ixc),(kdy,jfl),
     3 (kdy,jft),(kdy,jmt),(kdy,kmt),(kdy,ktc),(kdy,kxd),(kdy,lbb)
*i,lkon.12 blue
      if(jlock.eq.-1)call mp_setlock()
*i,lkoff.12 blue
      if(jlock.eq.1)call mp_unsetlock()
*d,bd.14,bd.17                                                     <422-425>
      data rktc/.001,.00125,.0015,.00175,.002,.0025,.003,.0035,.004,
     1 .0045,.005,.0055,.006,.007,.008,.009,.01,.0125,.015,.0175,.02,
     2 .025,.03,.035,.04,.045,.05,.055,.06,.07,.08,.09,.1,.125,.15,.175,
     3 .2,.25,.3,.35,.4,.45,.5,.55,.6,.7,.8,.9,.9999,40*1./
*/      data rkac/.9999,0.8,0.6,0.5,0.4,0.3,0.2,0.15,0.1,0.08,
*/     1   0.06,0.05,0.04,0.03,0.02,0.015,0.01,0.008,0.006,0.005,0.004,
*/     2   0.003,0.002,0.0015,0.001,0.0005,0.0002,0.0001,0.00005,
*/     3   0.00002,0.00001,0.000005,0.000002,0.000001,17*0.0/
*/      data  rktc/0.000001,0.0000015,0.000002,0.000003,0.000004,
*/     1 0.000005,
*/     1 0.000006,0.000008,0.00001,0.000015,0.00002,0.00003,0.00004,
*/     2 0.00005,0.00006,0.00008,0.0001,0.00015,0.0002,0.0003,0.0004,
*/     3 0.0005,0.0006,0.0008,0.001,0.00125,0.0015,0.00175,0.002,0.0025,
*/     4 0.003,0.0035,0.004,0.0045,0.005,0.0055,0.006,0.007,0.008,
*/     5 0.009,0.01,0.0125,0.015,0.0175,0.02,0.025,0.03,0.035,0.04,
*/     6 0.045,0.05,0.055,0.06,0.07,0.08,0.09,0.1,0.125,0.15,0.175,0.2,
*/     7 0.225,0.25,0.275,0.3,0.325,0.35,0.375,0.4,0.425,0.45,0.475,
*/     8 0.5,0.525,0.55,0.575,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,0.97,
*/     9 0.99,0.995,0.9995,1.0/
*d,bd4b.3
     3 hsd/'sequential','direct'/,ibin/'fdusmcet'/,loddat/'03/20/99'/,
*/ ------------------------------------------------------------ ttyint
*ident tt4b2
*/ Uncomment (delete 1st 3 columns) of next 2 lines to enable LAHEY F90
*/ compilation on DOS/WINDOWS.                        07/07/97  (GWM/GWM)
*/ *d,tt4a.28                                                      <1014>
*/      call break(lockl)
*/
*/ Uncomment the following for DVF on PCDOS.          02/17/98  (GWM/GWM)
*/ *i,tt4a.3                                                        <989>
*/        use dfport,only:signal
*/ *d,tt4b.5,tt4a.28                                            <1012-1014>
*/ *if def,pcdos,1
```

```
*/        k=signal(2,pttyin,-1)
*ident its30
*d,jc4b.1   for conductor specification                          <1548>
      parameter (nkcd=88,ntalmx=100,mopts=6)
*d,ib.72
      data cnm(54),(krq(i,54),i=1,7)/'phys ',0,1, 0,1,-1,  10,0/
*d,ib4a.10   for conductor specification                         <1677>
      data hmopt/'gas','estep','nlib','plib','elib','cond'/
*/ ------------------------------------------------------------- setdas
*d,sd.46                                                         <3348>
*/ ell way used
*/       ns=8
      ns=mstp
      ledg=lech+mpng*mwng*(nee/ns+1)*nmat1*ke
*d,sd4b.13,sd4b.14
      lfcdg=lx85+10*nmat1*nee
*d,sd.169   for conductor specification                          <3488>
      lcon=lemi+nmat1*ke
      lfdd=lcon+nmat1*ke
*i,ce4a.182
      if(bnum.ne.0.0.and.numb.ne.0)call erprnt(1,1,0,0,0,0,0,0,
     1 '"only one kind of brems biasing allowed"')
*d,ce.598  bbrem fix on mtop.
 1060 if(nwc.gt.49)go to 1070
*/ ------------------------------------------------------------- nextit
*d,ny4a.36   for conductor specification                         <5445>
      if(mlc.eq.6.and.nee.gt.0)cond(lcon+nmat)=ritm
      if(mlc.le.2.or.mlc.eq.6)go to 728
*ident bnum
*d,ny.443,ny.444                                            <5529-5530>
      if(nwc.eq.6)then
*/ allow for negative bnum for average energy loss calculation in brems.
*/ automatically reduce the subsequent electron production.  can be over-
*/ ridden by user with enum specification.
         bnum=ritm
         if(bnum.ne.zero)
     1 enum=one/abs(bnum)
      endif
      if(nwc.eq.10)numb=1
      if(ritm.lt.0..or.abs(1.-ritm).lt..00001)return
*ident its30
*d,ny.744,ny.745                                            <5830-5831>
 1480 if(nwc.le.49)bbrem(nwc)=1./ritm
      if(nwc.gt.49)mbi(lmbi+nwc-49)=-iitm
*d,oc4a.26,oc4a.28                                          <6306-6308>
  610 if(nwc.lt.49)call erprnt(2,1,2,nwc,49,0,0,0,
     1 '"too few entries:",i5," were read,",i5," are required."')
      m=49+max(kpt(3),kpt(2)*(1-ides))*nmat1
*/  ------------------------------------------------------------- getxst
*i,cor4-2.161
      if(nt.ne.0)lmb=mb/ndp2
*i,gt.213                                                        <16597>
*/ define an its30 flag only to ensure that its3.0 treatment is done
*/ in xsgen and colidp when nxs may have been zeroed.
      if(its30.eq.0)its30=nxs(lnxs+16,iex)
      if(its30.ne.nxs(lnxs+16,iex))call erprnt(2,1,0,0,0,0,0,
     1 1,'" electron tables are incompatible."')
*/
*/ ------------------------------------------------------------- xsgen
```

```
*i,xg.11                                                          <17823>
*/       nwng=mwng
      nstp=mstp
*/       do 2 i=1,mbng
*/     2 rka(i)=rkac(i)
      if(its30.eq.3)go to 5
      ntop=49
*/ mwng_ell=(mtop_ell+1)/2
      nwng=25
      nstp=8
      do 1 i=1,mtop
    1 rkt(i)=rktc(i)
      do 3 i=1,nwng
    3 rka(i)=rkt(ntop-2*i+2)
    5 continue
*i,xg.17                                                          <17829>
c        .000579=(electron_radius)**2/fine_structure_constant*1e24
c        and z+1 has been replaced with z+eta, where eta accounts
c        for electron-electron bremsstrahlung.
c        "cross sections for bremsstrahlung production and electron-
c        impact ionization," s. m. selzter, in monte carlo transport
c        of electrons and photons, t. m. jenkins, w. r. nelson, and
c        a. rindi, eds.
      f=0.00579
      if(nxs(lnxs+16,j).eq.3)f=crelec*crelec/fscon*1.d24
*d,xg.18,xg.21   its3.0 radiation yields.                          <17831>
*/ set up this way to allow for tracking the test problems.
      if(nxs(lnxs+16,j).eq.3)go to 11
      do 10 i=1,nxs(lnxs+3,j)
      exs(jxs(ljxs+2,j)+i-1)=log(exs(jxs(ljxs+2,j)+i-1)/gpt(3))
   10 exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)=
     1 .000579*z*(z+1.)*exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)
      go to 18
   11 do 12 i=1,nxs(lnxs+3,j)
      exs(jxs(ljxs+2,j)+i-1)=log(exs(jxs(ljxs+2,j)+i-1)/gpt(3))
      f1=exs(jxs(ljxs+2,j)+2*nxs(lnxs+3,j)+i-1)
   12 exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)=
     1 f*z*(z+f1)*exs(jxs(ljxs+2,j)+nxs(lnxs+3,j)+i-1)
      if(nxs(lnxs+9,j).gt.mtop)call erprnt(1,1,2,nxs(lnxs+9,j),mtop,0,
     1 0,1,'* nxs(9) greater than mtop.*')
*/ load values into rkt, rka, ntop, nwng only once.
*/ if exs is sacrosanct in the future this code can all be expunged.
      if(ntop.ne.0)go to 35
      ntop=nxs(lnxs+9,j)
      nwng=nxs(lnxs+10,j)
      if(nwng.gt.34)call erprnt(1,1,2,nxs(lnxs+10,j),34,0,
     1 0,1,'* nxs(10) greater than 34.*')
*/ perform a linear interpolation on bbrem for bins one cannot specify.
*/ interpolate on the low end of the biasing input.  store values in
*/ eba temporarily and then reload them and zero eba again.
      if(bbrem(1).eq.zero)go to 16
      n7=max(1,ntop-47)
      do 14 i=ntop,n7,-1
   14 eba(leba+i,1)=bbrem(49-ntop+i)
      n8=max(1,ntop-48)
      do 15 i=n8,1,-1
      eba(leba+i,1)=bbrem(1)+(bbrem(2)-bbrem(1))*
     1 (exs(jxs(ljxs+9,j)+i-1)-exs(jxs(ljxs+9,j)))/
     1 (exs(jxs(ljxs+9,j)+n8)-exs(jxs(ljxs+9,j)))
```

```
   15 if(eba(leba+i,1).le.zero)eba(leba+i,1)=max(bbrem(1),eba(leba+1,1))
c
c        load photon energy ratios from database into fixed arrays.
c        also load in interpolated bbrem values if needed.
   16 do 17 i=1,nxs(lnxs+9,j)
      rkt(i)=exs(jxs(ljxs+9,j)+i-1)
      if(i.le.nxs(lnxs+10,j))rka(i)=exs(jxs(ljxs+10,j)+i-1)
      if(bbrem(1).eq.zero)go to 17
      bbrem(i)=eba(leba+i,1)
      eba(leba+i,1)=zero
   17 continue
      go to 35
   18 continue
*i,xg.27                                                          <17839>
   35 continue
*i,xg.92                                                          <17892>
*/ istern is a flag for the sbs density effect treatment and if
*/ non-zero is the memory offset.
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3)istern=1
*/ bug repair
*d,xg.122      cn=min(iz,2)*fme(lfme+jm)/(a*3.)
      cn=zero
      do 165 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
  165 if(iz.eq.iza(liza+j)/1000)cn=cn+fme(lfme+j)
      cn=min(iz,2)*cn/(a*3.)
*d,xg.135   correct xnum bug which precludes turning electron impact
*/ ionzation k-xrays off.
      if(xnum.ge.zero)xnm(lxnm+mkc)=xnum
*i,xg.139
*/ to become ink(86)
      if(ink(85).eq.0)go to 180
      af=avgdn/aa
      write(iuo,'(/29helectron secondary production,75x,
     1 14hprint table 86/)')
      write(iuo,'(
     1 9x,6henergy,11x,14hstopping power,11x,5hbrems,3x,9hthick tgt,3x,
     2 7hk x-ray,3x,8hknock-on/
     3 4x,1hn,14x,19hcollision radiation,3x,5htotal,15x,5hbrems,16x,
     4 /10x,3hmev,6x,29hmev barn  mev barn  mev barn ,3x,5hbarn ,
     5 16x,4hbarn,6x,4hbarn)')
      do 175 n=nee-1,1,-1
      nk=n+nee*(mkc-1)
  175 write(iuo,176)n,eee(leee+n),xse85(lx85+1,nk)/af,
     1 xse85(lx85+2,nk)/af,xse85(lx85+3,nk)/af,pbr(lpbr+nk),
     2 pbt(lpbt+nk),pxr(lpxr+nk),pkn(lpkn+nk)
  176 format(i5,1pe12.4,2(1x,5e10.3))
*i,xg.140                                                         <17965>
      if(its30.eq.3)rka(1)=one
*/ debug prints remove for integration.
       close(17)
       close(18)
       close(19)
*ident bnum
*i,xg.142   set secondary electron to 1% arbitrarily for numb biasing.
      if(numb.eq.1.and.enum.eq.1.)enum=1./100.
*i,xg.145
      if(numb.eq.1)write(iuo,195)
  195 format(' bremsstrahlung generated on each electron substep.',
     1 /,' secondary electron production reduced by 99%.')
```

```
*/  ------------------------------------------------------------ ronge
*i,rn4a.4                                                        <18709>
c
c        set up some temporary dynamically allocated arrays for density
c        effect calculation.
      dimension fo(1),eq(1),xq(1)
*if -def,pointer,1
      equivalence (das,fo,eq,xq)
*if def,pointer,1
      pointer (kdy,fo),(kdy,eq),(kdy,xq)
*d,rn.11,rn.12                                                <18728-18729>
c       obtain an average density for a cell with the same material but
c       different densities.
      rh=zero
      nh=0
      ic=0
      do 10 i=1,mxa
      if(mat(lmat+i).ne.mkc)go to 10
      if(ic.eq.0)ic=i
      rh=rh+rho(lrho+i)
      nh=nh+1
   10 continue
      if(nh.eq.0.or.rh.eq.zero)
     & call expirx(0,'ronge','zero density or non-existent material')
      rh=rh/nh
*d,rn4b.16     f=1.13
      iz=iza(liza+jmd(ljmd+mkc))/1000
      do 15 j=jmd(ljmd+mkc)+1,jmd(ljmd+mkc+1)-1
   15 if(iz.ne.iza(liza+j)/1000)go to 16
      go to 21
   16 f=1.13
*d,rn.23,rn.30                                                <18753-18759>
c        az is zoa in the notation of sternhiemer, berger, seltzer.
*/ using this locution gives better agreement with its3.0 instead of crelec.
      cr=planck*slite/(2d0*pie*fscon*gpt(3))
      pl=2d+18*cr*gpt(3)*fscon*sqrt(pie*cr*rh*az)
      su=2.d0*log(sr/pl)
      if(istern.ne.0)go to 40
c
c       sternheimer and peierls treatment.
c       cb=su+1.
      cb=2.*log(sr/(37.1*sqrt(az*rho(lrho+ic))))+1.
      if(emi(lemi+mkc).eq.0.)go to 31
      xb=4+min(1,int(cb/12.25))
      xa=max(.1*one*min(20,max(16,int(2.*cb)-3)),.326*cb-2.5)
      go to 49
   31 xb=2+min(1,int(sr/100.))
      xa=max(.2+zero,.326*cb-.5*xb)
      go to 49
c
c       sternheimer, berger, and seltzer treatment.
   40 lr=0
      l1=0
      l2=0
      do 41 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      ns=nxs(lnxs+11,lme(llme+3,j))
      zs=exs(jxs(ljxs+11,lme(llme+3,j))+ns-1)
      if((cond(lcon+mkc).eq.zero).and.(zs.gt.zero))l1=l1+1
      if((cond(lcon+mkc).gt.zero).and.(zs.lt.zero))l2=l2+1
```

```
   41 lr=lr+ns
c
c        check conductor or non-conductor assignment.  non-conductor
c        if assigned by cond<0, or =0 and at least one non-conducting
c        component.  conductor if assigned cond>0 and at least one
c        conducting component, or cond=0 and no non-conducting
c        components.
      if(cond(lcon+mkc).ne.zero)go to 42
      if(l1.ne.0)go to 47
      cond(lcon+mkc)=one
      call erprnt(1,2,1,mkc,0,0,0,0,
     & '"material ",i4," has been set to a conductor"')
      go to 47
   42 if(l2.ne.0)go to 47
      if(cond(lcon+mkc).eq.-one)go to 43
      call erprnt(1,2,1,mkc,0,0,0,0,
     & '"material" ,i4," has been set to a non-conductor"')
      call erprnt(1,2,0,0,0,0,0,0,
     & '"no conduction electrons available"')
   43 cond(lcon+mkc)=zero
c
c        add memory for temporary arrays.
   47 if(istern.eq.1)istern=lmb
      if(lmb-istern.gt.3*lr)go to 48
      l7=(3*lr+istern)*ndp2
      if(lfl1.lt.l7)call chgmem(mdas,lfl1,l7,'ronge')
      lmb=l7/ndp2
   48 ld=istern
      call den1(az,su,pl,fo(ld+1),eq(ld+1+lr),xq(ld+1+2*lr),nt,wt,tm)
      if(nt.gt.lr)call erprnt(1,1,2,lr,nt,0,0,0,
     1 '"memory problem ",i4," allocated ",i4," used,"')
   49 continue
*i,rn.45
*d,rn.48,rn.49
      in=1
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)bl(k)=bl(k)+
     1 qpol(log(s),exs(jxs(ljxs+2,ix)),exs(jxs(ljxs+2,ix)+
     1 nxs(lnxs+3,ix)),nxs(lnxs+3,ix))*fme(lfme+j)*gpt(3)*(s+1.)
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3)bl(k)=bl(k)+
     1 spol(log(s),exs(jxs(ljxs+2,ix)),exs(jxs(ljxs+2,ix)+
     1 nxs(lnxs+3,ix)),nxs(lnxs+3,ix),in)*fme(lfme+j)*
     1 gpt(3)*(s+1.)
   60 continue
*d,rn.53,rn.57                                                <18782-18786>
      d=zero
      if(istern.eq.0)go to 65
      call den2(s,tm,fo(ld+1),eq(ld+1+lr),xq(ld+1+2*lr),nt,wt,d)
      go to 69
   65 x=.4342945*log(sqrt(s*(s+2.)))
      if(x.gt.xa)d=max(zero,4.606*x-cb+(cb-4.606*xa)*
     1 ((xb-min(x,xb))/(xb-xa))**3)
   69 if(k.eq.1)dd=d
*d,rn4b.62,rn4b.66
   80 nk=n+nee*(mkc-1)
      xse85(lx85+1,nk)=el(1)*af
      xse85(lx85+2,nk)=bl(1)*af
      xse85(lx85+3,nk)=(el(1)+bl(1))*af
      if(jovr(1)+jovr(3)+jovr(5).ne.3)go to 85
*i,rn4b.68 load correct bs                                    <18842>
```

```
      bs=e*(e+2.*gpt(3))/(e+gpt(3))**2
*d,rn4b.70                                                          <18844>
      xse85(lx85+7,nk)=cl*af*az*dd/bs
*d,rn4b.77                                                          <18856>
     1 1h.,2x,24hmean ionization energy =,1pe12.5,4h ev./)
      if(istern.eq.0.or.nz.ne.1)go to 94
      write(iuo,'('' density effect data'')')
      if(cond(lcon+mkc).eq.zero)write(iuo,'('' non-conductor'')')
      if(cond(lcon+mkc).gt.zero)write(iuo,'('' conductor'')')
      do 91 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      iz=iza(liza+j)/1000
      ns=nxs(lnxs+11,lme(llme+3,j))
      write(iuo,'(''   z ='',i4)') iz
      write(iuo,'(''    occ no, be(ev) pairs'')')
   91 write(iuo,'(6(6x,f3.0,f11.3))')
     & (exs(jxs(ljxs+11,lme(llme+3,j))+nn-1),
     & exs(jxs(ljxs+11,lme(llme+3,j))+ns+nn-1),
     & nn=1,ns)
      write(iuo,'(''        plas(ev)        wt        tmin(mev)'')')
      write(iuo,'(3x,3f12.5)') pl,wt,tm*gpt(3)
      write(iuo,'(/)')
   94 if(nz.eq.1)write(iuo,95)
   95 format(
*addfile ,rn                                                        <18873>
*deck den1
         subroutine den1(az,su,pl,fo,eq,xq,nt,wt,tm)
c
c        calculate oscillator parameters, fo, eq, xq, and the number
c        of oscillators, nt, based on shell data of carlson and the
c        plasma frequency, pl, and log of the ionization potential, su.
c        return the fitting parameter, wt, and minimum
c        normalized kinetic energy, tm, for use in den2.
c        this routine solves eq. (10) of r. m. sternheimer, s. m. seltzer,
c        and m. j. berger, "density effect for the ionization loss of
c        charged particles in various substances", phys rev b, 26, 6067(1982).
*call cm
      dimension fo(1),eq(1),xq(1)
      nt=0
      do 20 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      iz=iza(liza+j)/1000
      ns=nxs(lnxs+11,lme(llme+3,j))
      do 20 n=1,ns
      zs=exs(jxs(ljxs+11,lme(llme+3,j))+n-1)
      nt=nt+1
      fo(nt)=fme(lfme+j)*abs(zs)/az
      if((cond(lcon+mkc).ne.zero).and.(zs.le.zero))go to 10
      xq(nt)=(exs(jxs(ljxs+11,lme(llme+3,j))+ns+n-1)/pl)**2
      eq(nt)=2.d0*third*fo(nt)
      go to 20
   10 xq(nt)=zero
      eq(nt)=fo(nt)
   20 continue
c
c        find fitting parameter rho termed wt here following
c        sternheimer, berger and seltzer as implemented in its3.0.
      mt=0
      wt=2.d0
   30 xo=wt
      fu=zero
```

```
      dr=zero
      do 40 n=1,nt
      pa=wt*wt*xq(n)+eq(n)
      fu=fu+fo(n)*log(pa)
   40 dr=dr+fo(n)*2.d0*wt*xq(n)/pa
      wt=wt-(fu-su)/dr
      mt=mt+1
      if(mt.lt.25)go to 50
      call erprnt(1,2,1,mt,0,0,0,0,
     & 'i10," iterations den1 fitting parameter has not converged."')
c
c       when no fitting parameter is found, arbitrarily set it to
c       a small value.
      wt=1.e-4
      go to 70
   50 if(wt.gt.zero)go to 60
      wt=xo/2.d0
      go to 30
   60 if(abs(wt/xo-one).gt.1.d-6) go to 30
   70 w2=wt*wt
      do 80 l=1,nt
      xq(l)=w2*xq(l)
   80 eq(l)=xq(l)+eq(l)
c
c       find minimum normalized kinetic energy.
      tm=zero
      if(cond(lcon+mkc).gt.zero)return
      sm=zero
      do 90 l=1,nt
   90 sm=sm+fo(l)/xq(l)
      bq=one/(one+sm)
      tm=one/sqrt(one-bq)-one
      return
      end
*deck den2
      subroutine den2(s,tm,fo,eq,xq,nt,wt,d)
c
c        calculate the density effect, d, based on fitting parameter, wt
c        and oscillator parameters, fo, eq, xq, at normalized energy, s
c        greater than tm.  d=zero otherwise.
c        this routine solves eqs. (5) and (6) of sternheimer (1982).
*call cm
      dimension fo(1),eq(1),xq(1)
      d=zero
      if(s.le.tm)return
      mt=0
      re=one/(s*(s+2.d0))
      rc=(s+one)**2
      x=one/re
   10 xo=x
      fn=zero
      dv=zero
      do 20 n=1,nt
      fn=fn+fo(n)/(xq(n)+x)
   20 dv=dv-fo(n)/((xq(n)+x)**2)
      x=x-(fn-re)/dv
      mt=mt+1
      if(mt.lt.25)go to 30
c
```

```
c        when no fitting parameter is found, arbitrarily set it to
c        zero.
      call erprnt(1,2,1,mt,0,0,0,0,
     & 'i10,"iterations den2 fitting parameter has not converged."')
      call erprnt(1,2,0,0,0,0,0,0,
     & '"density effect correction set to zero."')
      x=zero
      go to 50
   30 if(x.ge.zero)go to 40
      x=xo/2.d0
      go to 10
   40 if(abs(x/xo-one).gt.1.d-6) go to 10
   50 d=-x/rc
      do 60 l=1,nt
   60 d=d+fo(l)*log(one+x/eq(l))
      return
      end
*ident its30
*/ ------------------------------------------------------------ sing
*d,sn4b.4,sn4b.6                                          <19016-19018>
c        the constant 3.1329 is 4. * 0.885**2
c        which is (3.d0*pie/4.d0)**(-4.d0*third)
      ea=(1.13+3.76*sqrt(s/(s+1.))*u**2)*z**(2.*third)/(3.1329*t4*
     1 fscon**2)
*/ ------------------------------------------------------------ brem
*i,bm.6                                                        <19130>
c
c        set up some temporary dynamically allocated arrays for
c        bremsstrahlung production calculation.
      dimension xs(1),x2(1),ti(1),xi(1)
*if -def,pointer,1
      equivalence (das,xs,x2,ti,xi)
*if def,pointer,1
      pointer (kdy,xs),(kdy,x2),(kdy,ti),(kdy,xi)
*i,bm.8                                                        <19132>
      data ld/0/
      save ld
*i,bm.11                                                       <19135>
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)then
*d,bm.23                                                       <19147>
      do 100 k=2,ntop
*d,bm.61                                                       <19185>
      pbr(lpbr+n+nee*(mkc-1))=eba(leba+ntop,n+nee*(mkc-1))/30
*d,bm.64                                                       <19188>
      do 110 k=2,ntop
*d,bm.66                                                       <19190>
     1 eba(leba+ntop,n+nee*(mkc-1))
      else
c
c        bremsstrahlung production cross sections based upon the evaluation
c        described in:
c        "cross sections for bremsstrahlung production and electron-
c        impact ionization," s. m. selzter, in monte carlo transport
c        of electrons and photons, t. m. jenkins, w. r. nelson, and
c        a. rindi, eds.  algorithm implementation based upon its3.0 and
c        fit values from its3.0 database.
c        set up dynamic memory.  use the values for the first zaid
c        since they are equal.
      ix=lme(llme+3,jmd(ljmd+mkc))
```

```
      nm=nxs(lnxs+5,ix)
      km=nxs(lnxs+6,ix)
      lr=(km+nm+(nm+nee)*km)
      if(ld.eq.0)call chgmem(mdas,lfll,(lmb+lr)*ndp2,'brem')
      if(ld.eq.0)lmb=lmb+lr
      ld=lmb-lr
      lx=ld
      lt=lx+km
      ls=lt+nm
      l2=ls+nm*km
      do 320 n=1,lr
  320 xs(ld+n)=zero
      do 330 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
      ix=lme(llme+3,j)
      z=nxs(lnxs+2,ix)
      do 330 n=1,nm
      ti(lt+n)=log(exs(jxs(ljxs+5,ix)+n-1))
      do 330 k=1,km
      xs(ls+n+(k-1)*nm)=xs(ls+n+(k-1)*nm)+
     & z*z*exs(jxs(ljxs+5,ix)+km+n-1+k*nm)*fme(lfme+j)
  330 continue
      do 340 k=1,km
      xi(lx+k)=log((one-exs(jxs(ljxs+5,ix)+nm+k-1))+1.d-6)
      do 340 n=1,nm
  340 xs(ls+n+(k-1)*nm)=log(xs(ls+n+(k-1)*nm))
      do 350 k=1,km
      in=1
      do 350 n=1,nee
      tl=log(eee(leee+n))
      bl=log(one-(gpt(3)/(eee(leee+n)+gpt(3)))**2)
*/ quadratic fit if spol is not desired.
*/  350 x2(l2+k+(n-1)*km)=qpol(tl,ti(lt+1),xs(ls+1+(k-1)*nm),nm)
  350 x2(l2+k+(n-1)*km)=spol(tl,ti(lt+1),xs(ls+1+(k-1)*nm),nm,in)
     & -bl
      do 390 n=1,nee
      eba(leba+1,n+nee*(mkc-1))=zero
      in=1
      do 380 k=2,ntop
      de=rkt(k)-rkt(k-1)
      do 360 np=1,11
      ek(np)=rkt(k)-real(np-1)*de*1.d-1
      xp=log((one-ek(np))+1.d-6)
*/ quadratic fit if spol is not desired.
*/  360 g(np)=exp(qpol(xp,xi(lx+1),x2(l2+1+(n-1)*km),km))
  360 g(np)=exp(spol(xp,xi(lx+1),x2(l2+1+(n-1)*km),km,in))
      do 370 i=1,11
  370 g(i)=g(i)/ek(i)
  380 eba(leba+k,n+nee*(mkc-1))=
     & (g(1)+4.d0*(g(2)+g(4)+g(6)+g(8)+g(10))+
     1 2.d0*(g(3)+g(5)+g(7)+g(9))+g(11))*de+
     2 eba(leba+k-1,n+nee*(mkc-1))
c
c        30=3*(number of integration points -1)
      pbr(lpbr+n+nee*(mkc-1))=eba(leba+ntop,n+nee*(mkc-1))/30.d0
c
c         normalize the cumulative unbiased probability, eba.
      do 390 k=2,ntop
  390 eba(leba+k,n+nee*(mkc-1))=
     & eba(leba+k,n+nee*(mkc-1))/eba(leba+ntop,n+nee*(mkc-1))
```

```
      endif
*/ write the cross sections for validation and debug
      do 601 n=1,nee
      write(17,*)eee(leee+n),pbr(lpbr+n+nee*(mkc-1))
      write(18,*)' new group ',eee(leee+n)
      do 601 k=1,ntop
601   write(18,*)rkt(k),eba(leba+k,n+nee*(mkc-1))
*d,bm.75                                                          <19199>
      do 120 k=2,ntop
*d,bm.80                                                          <19204>
     1 q=qpol(e/eee(leee+j),rkt,eba(leba+1,j+nee*(mkc-1)),ntop)
*d,bm.86                                                          <19210>
      do 130 k=2,ntop
*d,bm.88,bm.89                                            <19212-19213>
     1 ebt(lebt+ntop,n+nee*(mkc-1))
      do 140 k=1,ntop
*d,bm.98                                                          <19222>
      do 150 k=2,ntop
*d,bm.103                                                         <19227>
      do 160 k=1,ntop-1
*d,bm.108                                                         <19232>
      do 170 k=2,ntop
*d,bm.113                                                         <19237>
      do 180 k=1,ntop-1
*/
*/ ------------------------------------------------------------- brang
*i,bg.7    correct possible parameter and constant mismatch                <19273>
      save ih,n1,n2,r2
      data ih/0/
*i,bg.8    correct possible parameter and constant mismatch                <19274>
      if(ih.ne.0.or.nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)go to 2
      ih=1
      n1=(mpng+1)/2
      n2=n1-1
      r2=float(n2)
      do 1 i=1,n1
    1 cm(i)=(i-1)/r2
    2 continue
*d,bg.13                                                          <19279>
      do 150 n=1,nee,nstp
*d,bg.15,bg.16                                            <19281-19282>
      do 20 k=1,nwng
   20 ek(k)=rka(k)*s
*d,bg.25,bg.27   correct possible parameter and constant mismatch    <19291-19293>
      xi(l)=(1.+.1*b*(11-l))/(1.+b)
      ch(l)=(11-l+10.*b)/(10.+b*(11-l))
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)go to 30
      xi(l)=(one+b*(n1-l)/r2)/(one+b)
      ch(l)=(n1-l+r2*b)/(r2+b*(n1-l))
   30 continue
      do 150 k=1,nwng
*d,bg.104                                                         <19370>
  150 ech(lech+l,k,n/nstp+1+(nee/nstp+1)*(mkc-1))=min(max(zero,
*i,bg.105    write the cross sections for validation              <19371>
      do 510 n=1,nee,nstp
      do 510 k=1,nwng
      do 510 l=1,mpng
  510 write(19,*)n,k,l,ech(lech+l,k,n/nstp+1+(nee/nstp+1)*(mkc-1))
*i,mr.6 blue
```

```
      integer*4 mz
      mz=ntasks
      call mp_set_numthreads(mz)
*d,mr.17,mr.18 blue
      call secnd(cp2(1))
      cp0=cp2(1)-cpa
*i,mr.92 blue
C$PAR PARALLEL DO
      do 141 i=1,ntasks
  141 call trnspt
*d,tn.52,tn.54 blue
      if(t2.ge.cp2(ktask+1)-1.)go to 50
      write(jtty,40)kod,cp2(ktask+1)/60.,t2/60.
      write(iuo,40)kod,cp2(ktask+1)/60.,t2/60.
*d,tn.59,tn.61 blue
   50 cp1=cp1+t2-cp2(ktask+1)
      cts=cts+t2-cp2(ktask+1)
   60 cp2(ktask+1)=t2
*d,ut.27,ut.28 blue
      mz=mp_my_threadnum()
      ktask=mz
*/
*/ ------------------------------------------------------------ msgcon
*/ Fix a print bug.  Wrong number of arguments.      07/07/97  (GWM/GWM)
*d,me4b.82 blue
      call secnd(cp2(1))
*d,me4b.154 blue
      call secnd(cp2(1))
*ident me4b2
*d,me4b.195                                                     <20311>
      call errprn(0,j,1,one*max(1,ntasks),zero,'ntk',' ',
*/ Force PVM to free some buffers.                  07/07/97  (GWM/GWM)
*/ $20 awarded to Dudley A. Raine, III, ORNL (XTM:JSH-97-176)  (GWM/GWM)
*i,me4b.311                                                     <20498>
c
c         free the receive buffers.
      do 582 nt=1,ntasks
  582 call mfbuf(ib(nt),i)
*d,me4b.338 blue
      call secnd(cp2(1))
*d,me4b.388 blue
      cp1=cp1+t2-cp2(1)-cp3
*/
*/ ------------------------------------------------------------ startp
*ident sp4b2
*/ Wrong index. Affects var. reduction and tracking. 07/07/97  (GWM/GWM)
*d,sp4b.8,sp4b.10                                     <21159-21161>
      if(wc1(i).ge.0.)go to 85
      wcs1tc(i)=-wc1(i)*wgt
      wcs2tc(i)=-wc2(i)*wgt
*/
*/ ------------------------------------------------------------ levcel
*ident l14b2
*/ Silent wrong answers for near-coincident lattices.          08/25/97
*/ $20 to Lee Montierth (INEEL) XTM:JSH-97-208
*d,l14b.25                                                      <22232>
      jsu=-abs(jsu)
      if(l.gt.ll+1.and.lat(llat+1,icl).ne.0)call findel(icl)
      jsu=abs(jsu)
```

```
*/
*/ ------------------------------------------------------------ ypbssp
*ident yp4b2
*/ Wrong index. Affects var. reduction and tracking. 07/07/97  (GWM/GWM)
*d,yp4b.2,yp4b.4                                          <22590-22592>
      if(wc1(i).ge.0.)go to 45
      wcs1tc(i)=-wc1(i)*wgt
      wcs2tc(i)=-wc2(i)*wgt
*/
*/ ------------------------------------------------------------ wwval
*ident we4b2
*/ Fix a print bug.  Wrong number of arguments.      07/23/97  (GWM/GWM)
*d,we4b.145                                                     <23152>
  200 call errprn(1,nwsg(3),-1,1,icl,' ','icl',
*/
*/ ------------------------------------------------------------ track
*ident tr4b2
*/ Silent wrong answers for near-coincident lattices.           08/25/97
*/ $20 to Lee Montierth (INEEL) XTM:JSH-97-208
*d,tr4b.18                                                      <23651>
      if(dl(i)+coincd.lt.dl(levp))go to 445
*/
*/ ------------------------------------------------------------ findel
*ident fn4b2
*/ Silent wrong answers for near-coincident lattices.           08/25/97
*/ $20 to Lee Montierth (INEEL) XTM:JSH-97-208
*d,fn.6                                                         <24481>
      dimension ii(3),jx(14,2)
*i,fn.7  after equivalence                                      <24482>
      data jx/1,1,2,2,3,3,1,1,2,2,2,2,3,3,
     1  1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1/
*d,fn.10  n=                                                    <24485>
*d,fn4b.5,fn4b.16  ii(i)= thru label 10                     <24493-24504>
   10 ii(i)=nint(a)
*d,fn.22,fn.52                                              <24510-24550>
c
c        correct for coincident surfaces and hexagonal prism lattices.
c        check if new location completely inside lattice element.
      jc=0
   30 jc=jc+1
      if(jc.gt.10)go to 70
      do 60 jk=lca(llca+ic),abs(lca(llca+ic+1))-1
      j=abs(lja(llja+jk))
      k=kst(lkst+j)
      l=lsc(llsc+j)
      if(k.eq.1)t4=scf(l+1)*xxx+scf(l+2)*yyy+scf(l+3)*zzz-scf(l+4)
      if(k.ne.1)t4=gpblcm(k-1)-scf(l+1)
      if(jsu.ge.0)go to 40
      if(ksc(lksc-jsu).ne.ksc(lksc+j))go to 40
      if(k.eq.1)t5=scf(l+1)*uuu+scf(l+2)*vvv+scf(l+3)*www
      if(k.ne.1)t5=gpblcm(k+2)
      if(abs(t4).le.coincd*abs(t5))t4=t5
   40 if(lja(llja+jk)*t4.gt.0)go to 60
c
c        location coincident with or beyond surface j, index ix.
      jl=jk-lca(llca+ic)+6*lat(llat+1,ic)-5
      ix=jx(jl,1)
      iy=jx(jl,2)
   50 ii(ix)=ii(ix)+iy
```

```
      xxx=xxx-iy*vcl(lvcl+1,ix,m)
      yyy=yyy-iy*vcl(lvcl+2,ix,m)
      zzz=zzz-iy*vcl(lvcl+3,ix,m)
c
c        if on or beyond hex side 3, increment sides 1 and 2.
      if(ix.ne.2.or.jl.lt.11)go to 30
      iii=iii-iy
      xxx=xxx+iy*vcl(lvcl+1,1,m)
      yyy=yyy+iy*vcl(lvcl+2,1,m)
      zzz=zzz+iy*vcl(lvcl+3,1,m)
      go to 30
   60 continue
      return
   70 kdb=1
*/
*/ ------------------------------------------------------------ findlv
*ident fv4b2
*/ Silent wrong answers for near-coincident lattices.          08/25/97
*/ $20 to Lee Montierth (INEEL) XTM:JSH-97-208
*d,fv.35                                                        <24587>
   60 jsu=-abs(jsu)
      if(lat(llat+1,icl).ne.0)call findel(icl)
      jsu=abs(jsu)
*/
*/ ------------------------------------------------------------ chkcel
*ident cc4b2
*/ Silent wrong answers for near-coincident lattices.          08/25/97
*/ $20 to Lee Montierth (INEEL) XTM:JSH-97-208
*d,cc4b.4                                                       <24700>
      t5=scf(i)*uuu+scf(i+1)*vvv+scf(i+2)*www
      if(abs(t4).le.coincd*abs(t5))t4=t5
*d,cc4b.9                                                       <24705>
      if(abs(t4).le.coincd*abs(gpblcm(k+2)))t4=gpblcm(k+2)
*/
*/ ------------------------------------------------------------ colidp
*ident flaug
*b,cp.163                                                       <26798>
      if(its30.ne.3)then
*i,cp4a.16                                                      <26874>
      else
      er=erg
      ie=iex
      ik=ipt
      call flaug(er,ie,mk,ik)
      endif
*addfile ,cp                                                    <26916>
*deck flaug
      subroutine flaug(er,ie,mk,ik)
c
c        a subroutine to generate fluourescent photons or auger
c        electrons to fill vacancies.
c        called by colidp and kxray.
c        er=energy of absorbed photon or edge energy for kxray
c        ie=element that is undergoing the interaction; for kxray it is
c        the highest z element.
c        mk=material and ik=ipt for kxray(3) or colidp(2) for keeping
c        track of parent particle for bookkeeping and vr.
*call cm
      ipt=2
```

```
      ntyn=3
      rp=0.
      if(ik.eq.2)go to 5
      vel=slite
      ncp=0
      idx=0
c
c       check for electron emission from impact ionzation.
      if(rang().gt.wwk(lwwk+mk))go to 80
    5 if(dbcn(20).ne.0.)rp=1.
      if(nxs(lnxs+2,ie).le.11)go to 90
      i1=jxs(ljxs+4,ie)+2
      if(er.le.xss(i1-1))go to 90
      do 10 ka=i1,i1-3+nxs(lnxs+4,ie)
   10 if(er.le.xss(ka))go to 20
   20 ka=ka+nxs(lnxs+4,ie)-1
      k0=ka
   25 ka=k0
      t3=xss(ka)*rang()
      ka=ka+nxs(lnxs+4,ie)
      if(ik.eq.3.and.t3.ge.xss(ka))go to 25
      if(ik.eq.2.and.t3.ge.xss(ka))go to 90
      rp=1.
   30 ka=ka-1
      if(t3.lt.xss(ka))go to 30
c
c        at least single fluorescence or auger electron.
      eg=xss(ka-2*nxs(lnxs+4,ie)+1)
      if(kpt(3).eq.0)go to 40
      if(ik.eq.2.and.rang().gt.wwk(lwwk+mk))go to 80
   40 e=xss(nxs(lnxs+4,ie)+ka+1)
      if(e.lt.elc(2))go to 90
      if(ik.ne.2)go to 41
      paxtc(4,12,2)=paxtc(4,12,2)+1.
      paxtc(5,12,2)=paxtc(5,12,2)+wgt
      paxtc(6,12,2)=paxtc(6,12,2)+wgt*erg
      pwb(kpwb+2,18,icl)=pwb(kpwb+2,18,icl)-wgt
      pan(kpan+2,3,mpan)=pan(kpan+2,3,mpan)+wgt
   41 ed=0.
      if(nxs(lnxs+2,ie).le.30)go to 60
      i=ka+1-2*nxs(lnxs+4,ie)
      if(i.ne.i1.and.i.ne.i1+1)go to 60
      i=i1-2+nxs(lnxs+4,ie)
      ka=nxs(lnxs+4,ie)+i+1
      if(rang().ge.xss(ka)/(1.-xss(i)))go to 60
c
c        double fluorescence.  bank the second photon.
      erg=xss(nxs(lnxs+4,ie)+ka)
      if(erg.lt.elc(2))go to 60
      if(ik.ne.2)go to 42
      paxtc(1,16,2)=paxtc(1,16,2)+1.
      paxtc(2,16,2)=paxtc(2,16,2)+wgt
      paxtc(3,16,2)=paxtc(3,16,2)+wgt*erg
      pwb(kpwb+2,17,icl)=pwb(kpwb+2,17,icl)+wgt
      go to 43
   42 paxtc(1,14,2)=paxtc(1,14,2)+1.
      paxtc(2,14,2)=paxtc(2,14,2)+wgt
      paxtc(3,14,2)=paxtc(3,14,2)+wgt*erg
      pwb(kpwb+2,16,icl)=pwb(kpwb+2,16,icl)+wgt
```

```
 43 ntyn=4
    call isos(uuu,lev)
    sw=wgt
    if(ik.ne.2)go to 44
    if(abs(wwp(2,4)).eq.1..and.(idx.eq.0.or.wwp(2,5).lt.0.))
   1 call wtwndo(4,ww)
    if(nter.ne.0)go to 50
    if(espl(2,1).ne.0.)call ergimp
    if(nter.ne.0)go to 50
 44 npa=1
    call bankit(9)
    ed=erg
    wgt=sw
    go to 60
 50 paxtc(4,nter,2)=paxtc(4,nter,2)+1.
    paxtc(5,nter,2)=paxtc(5,nter,2)+wgt
    paxtc(6,nter,2)=paxtc(6,nter,2)+wgt*erg
    pwb(kpwb+2,jrwb(nter,2),icl)=pwb(kpwb+2,jrwb(nter,2),icl)-wgt
    nter=0
 60 if(ides.eq.0.and.ik.eq.2.and.eg0-eg-ed.gt.elc(3))
   & call emaker(1,eg0-eg-ed)
    erg=e
    ipsc=99+ntyn
    call isos(uuu,lev)
    if(ik.ne.2)go to 61
    paxtc(1,15,2)=paxtc(1,15,2)+1.
    paxtc(2,15,2)=paxtc(2,15,2)+wgt
    paxtc(3,15,2)=paxtc(3,15,2)+wgt*erg
    pwb(kpwb+2,17,icl)=pwb(kpwb+2,17,icl)+wgt
    go to 100
 61 continue
    if(jgp.le.-2)jgp=jgp+2
    paxtc(1,14,2)=paxtc(1,14,2)+1.
    paxtc(2,14,2)=paxtc(2,14,2)+wgt
    paxtc(3,14,2)=paxtc(3,14,2)+wgt*erg
    pwb(kpwb+2,16,icl)=pwb(kpwb+2,16,icl)+wgt
    do 70 i=1,ndx(2)
 70 if((xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+
   1 (zzz-dxx(2,3,i))**2.lt.dxx(2,5,i))idx=i
    if(ndet(2).ne.0)call tallyd
    if(kdb.ne.0)return
    if(ndx(2).gt.1.or.ndx(2).eq.1.and.idx.eq.0)call dxtran
    if(kdb.ne.0)return
    call bankit(18)
    go to 100
c
c       make electrons either auger or photoelectric.
c       set ipt=ik for bookkeeping.
 80 ipt=ik
    if(ides.eq.0.and.eek(leek+mk).gt.elc(3).and.
   1 (dbcn(20).ne.0..or.er.ge.edg(ledg+mk)))
   1 call emaker(4,eek(leek+mk))
 90 if(ides.eq.0.and.eg0-rp*edg(ledg+mk).gt.elc(3).and.
   & ik.eq.2)call emaker(1,eg0-rp*edg(ledg+mk))
    if(ik.ne.2)go to 100
    nter=12
    pan(kpan+2,3,mpan)=pan(kpan+2,3,mpan)+wgt
100 continue
    return
```

```
      end
*/
*/ ------------------------------------------------------------- ttbr
*ident bnum
*d,tb.23   <27185>
      pr=pr*abs(bnum)
*d,tb.46   <27204>
      if(abs(bnum).gt.0..and.bnum.ne.1.)wgt=wgt/abs(bnum)
*ident its30
*d,tb.67,tb.68                                          <27223-27224>
      pk=qpol(rn,ebt(lebt+1,m+nee*(mkc-1)),rkt,ntop)
      pa=qpol(rn,ebt(lebt+1,m+1+nee*(mkc-1)),rkt,ntop)
*d,tb.73                                                      <27229>
      do 110 j1=2,ntop-1
*d,tb.75                                                      <27231>
  120 do 130 j=2,ntop-1
*/
*/ ------------------------------------------------------------ kxray
*ident flaug
*b,kx.24                                                       <28859>
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).ne.3)then
*i,kx.51                                                       <28883>
      else
      if(iphot.ne.0.or.fiml(2).eq.0..or.eek(leek+mkc).le.elc(2))
     1 go to 80
c
c        assume the electron has enough energy to excite the atom
c        above the threshold of the highest K shell energy of the
c        highest Z atom of the material.
   71 jm=jmd(ljmd+mkc)
      do 72 j=jmd(ljmd+mkc),jmd(ljmd+mkc+1)-1
   72 if(iza(liza+j).gt.iza(liza+jm))jm=j
      ix=lme(llme+2,jm)
      iex=ix
      mk=mkc
*/      er=xss(jxs(ljxs+4,ix)+nxs(lnxs+4,ix)-1)*1.00001
c
c        1.0001 assures round-off does not give spurious results
      er=edg(ledg+mkc)*1.0001
      if(er.gt.eg0)er=eg0
      ik=ipt
      call flaug(er,ix,mk,ik)
      endif
*/
*/ ------------------------------------------------------------ brems
*ident its30
*i,bb.6
c
c        el is dimensioned to keep track of first, possibly real event
c        energy loss, and second and higher order events to keep track of
c        energy loss only for production purposes.
c        ib flags real events in conjunction with numb vr use.
      dimension el(2)
*d,bb.11      if(pr+rang().lt.1.)return
      r1=rang()
      p1=exp(-pr)
      ib=1
      if((nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0
     1 .and.pr+r1.lt.1.).or.
```

```
     1 (nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3.and.r1.lt.p1))ib=0
      if(numb+ib.eq.0)return
c
c        set up to produce none, one, or more photons.
      npa=1
      nb=1
c
c        do splitting or roulette if bnum biasing is used.
      if(bnum.ne.one.and.dbcn(23).ne.zero)then
         nb=abs(bnum)+rang()
         if(nb.eq.0)then
             npa=0
             nb=1
         endif
      endif
      wb=wgt
      if(bnum.ne.zero.and.dbcn(23).ne.zero)
     1 wb=wgt/abs(bnum)
      rb=nb
      w0=wb
      if(numb.eq.1)wb=wb*(one-p1)
      ps=p1
      do 240 n1=1,nb
      ie=min(n1,2)
      el(ie)=zero
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0)go to 5
      p1=ps
      ap=p1
      fa=zero
*d,bb.14                                                          <28908>
    5 npb=npb+1
*d,bb.21,bb.23    bnum                                      <28915-28916>
*i,bb.25          bnum                                            <28918>
      wgt=wb
*ident its30
*d,bb4a.4
      es=eg0-qo*dk-el(ie)
      if(es.lt.elc(3))go to 200
*d,bb.40,bb.41                                              <28930-28931>
      pk=qpol(rn,eba(leba+1,m+nee*(mkc-1)),rkt,ntop)
      pa=qpol(rn,eba(leba+1,m+1+nee*(mkc-1)),rkt,ntop)
*i,bb.43          debug print
c      write(20,*)erg
*ident bnum
*d,bb.46    bnum                                                  <28936>
c
c        accumulate radiative energy loss for first photon or all
c        of them for bnum<0.
      el(ie)=el(ie)+erg
      if(n1.gt.1.and.bnum.lt.zero)el(1)=el(1)+erg
*d,cor4-1.163    bnum                                             <28939>
      if(iphot.ne.0.or.fiml(2).eq.0..or.es.lt.elc(2).or.
     & npa.eq.0)go to 200
*ident its30
*d,bb.54,bb.55                                              <28944-28945>
      pk=qpol(rn,ebd(lebd+1,m+nee*(mkc-1)),rkt,ntop)
      pa=qpol(rn,ebd(lebd+1,m+1+nee*(mkc-1)),rkt,ntop)
*d,bb.57                                                          <28947>
      do 60 j1=2,ntop-1
```

```
*d,bb.59                                                          <28949>
   70 do 80 j=2,ntop-1
*ident bnum
*d,bb.71,bb.74   bnum                                          <28961-28964>
      if(bnum.eq.1.)go to 110
      if(dbcn(23).ne.zero)go to 110
      npa=bnum+rang()
      if(npa.eq.0)go to 200
      wgt=wgt/bnum
*ident its30
*d,bb.89,bb.105                                                <28978-28994>
      k=(m-1)/nstp+1
      ek=eee(leee+(k-1)*nstp+1)-es
      el=es-eee(leee+k*nstp+1)
      do 140 j=2,nwng
  140 if(pj.ge.rka(j))go to 150
  150 rj=rka(j-1)-pj
      sj=pj-rka(j)
      l=rn*(mpng-1)
      pl=rn*(mpng-1)-1
      ql=1.-pl
      i=lech+l+1
      mc=k+(nee/nstp+1)*(mkc-1)
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0)then
      cx=4.*(ql*sj*el*ech(i,j-1,mc)+pl*rj*ek*ech(i+1,j,mc+1)+
     1 pl*sj*el*ech(i+1,j-1,mc)+ql*rj*el*ech(i,j,mc)+
     2 ql*sj*ek*ech(i,j-1,mc+1)+ql*rj*ek*ech(i,j,mc+1)+
     3 pl*sj*ek*ech(i+1,j-1,mc+1)+pl*rj*el*ech(i+1,j,mc))/
     4 ((rka(j-1)-rka(j))*eee(leee+(k-1)*nstp+1))
      else
*/ for exact its3.0 emulation.
*/       rx=one-rn
*/       l=rx*(mpng-1)
*/       pl=rx*(mpng-1)-1
*/       ql=one-pl
*/       i=lech+l+1
      cx=2.d0*(ql*sj*el*ech(i,j-1,mc)+pl*rj*ek*ech(i+1,j,mc+1)+
     1 pl*sj*el*ech(i+1,j-1,mc)+ql*rj*el*ech(i,j,mc)+
     2 ql*sj*ek*ech(i,j-1,mc+1)+ql*rj*ek*ech(i,j,mc+1)+
     3 pl*sj*ek*ech(i+1,j-1,mc+1)+pl*rj*el*ech(i+1,j,mc))/
     4 ((rka(j-1)-rka(j))*(eee(leee+(k-1)*nstp+1)-eee(leee+k*nstp+1)))
      endif
*i,bb.106     debug and verification
c      write(20,*)cx,am
*ident bnum
*d,cor4-1.167          bnum                                      <29052>
      if(dbcn(23).eq.zero.and.bnum.gt.1.)wgt=wgt*bnum
*ident its30
*d,bb.152,bb.154  erg=erg-el
      if(nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.0.or.ib.eq.0)
     1 go to 240
      if(ib.eq.1.and.numb.eq.1)wb=w0
      fa=fa+one
      pl=pl*pr/fa
      ap=ap+pl
      if(rl.ge.ap)go to 5
  240 continue
      if(ib.eq.0)return
      if(dbcn(23).ne.zero.and.bnum.lt.zero)
```

```
     1 el(1)=el(1)/rb
       erg=erg-el(1)
       paxtc(6,12,3)=paxtc(6,12,3)+wgt*el(1)
       if(el(1).ne.zero.and.
     1 nxs(lnxs+16,lme(llme+3,jmd(ljmd+mkc))).eq.3)then
          paxtc(4,12,3)=paxtc(4,12,3)+one
          paxtc(5,12,3)=paxtc(5,12,3)+wgt
       endif
       return
*addfile bb
*deck spol
c      subroutine spol (s,x,y,n,in,t), 15 jan 71.
       function spol (s,x,y,n,in)
*/       function spol (s,x,y,n,m)
c
c          cubic spline interpolation with parabolic runout.
*if def,cheap,1
       implicit double precision (a-h,o-z)
       parameter (ln=1000)
       save
       dimension  x(n),y(n),e(ln),u(ln)
       if(n.gt.ln)call erprnt(1,1,2,n,ln,0,0,1,
     1 '" too large an argument in spol, increase parameter ln"')
       if (in.ne.1) go to 30
       in=2
       n1=n-1
       e(1)=1.d0
       u(1)=0.d0
       b1=x(2)-x(1)
       c1=(y(2)-y(1))/b1
       do 10 j=2,n1
       b2=x(j+1)-x(j)
       c2=(y(j+1)-y(j))/b2
       b=x(j+1)-x(j-1)
       d=(c2-c1)/b
       c=b1/b
       b1=b2
       c1=c2
       p=c*e(j-1)+2.d0
       e(j)=(c-1.d0)/p
    10 u(j)=(d-c*u(j-1))/p
       e(n)=u(n1)/(1.d0-e(n1))
       do 20 kk=1,n1
       k=n-kk
    20 e(k)=e(k)*e(k+1)+u(k)
       id=0
       m2=0
       m1=n
       if (x(1) .le. x(n)) go to 30
       id=1
       m2=n
       m1=0
    30 if(s.lt.x(m1+id).and.s.gt.x(m2+1-id))go to 40
       if(s.ge.x(m1+id))mu=m1+2*id
       if(s.le.x(m2+1-id))mu=m2+2*(1-id)
       go to 60
    40 m1=m2
       mu=m1
    50 m3=(m1+mu)/2
```

```
      if(s.lt.x(m3))mu=m3
      if(s.ge.x(m3))ml=m3
      if(iabs(mu-ml).gt.1)go to 50
      mu=mu+id
c
   60 spol=(e(mu-1)*((x(mu)-s)**3)+e(mu)*((s-x(mu-1))**3)+
     1 (y(mu-1)-e(mu-1)*((x(mu)-x(mu-1))**2))*(x(mu)-s)+
     2 (y(mu)-e(mu)*((x(mu)-x(mu-1))**2))*(s-x(mu-1)))/
     3 (x(mu)-x(mu-1))
      return
      end
*d,ad.43 blue
      tfc(ltfc+3,1,it)=(cts-cpk+t2-cp2(1))*tfc(ltfc+2,1,it)**2
*ident mf4b2
*/ Force PVM to free some buffers.                    07/07/97  (GWM/GWM)
*/ $20 awarded to Dudley A. Raine, III, ORNL (XTM:JSH-97-176)  (GWM/GWM)
*i,mf4b.83                                                        <41720>
      subroutine mfbuf(ib,in)
c        perform function 'free message buffer'.
*call mb
*if def,pvm,1
      call pvmffreebuf(ib,in)
      return
      end
*/
*ident ge4b2
*i,ge.4
      integer*4 i
*/ Uncomment the following for DVF on PCDOS.          02/17/98  (GWM/GWM)
*/ *i,ge.3                                                          <42172>
*/        use dfport,only:iargc
*/        use dflib,only:getarg
*/ *d,ge4a.8,ge4a.9                                          <42196-42197>
*/ *if def,pcdos,5
*/       hm=' '
*/       j=0
*/       do 10 i=1,iargc()
*/       call getarg(i,hm(j+1:))
*/    10 j=j+leng(hm(j+1:))+1
*/
*/ ------------------------------------------------------------ getidt
*ident gi4b2
*/ Provide cleaner separation of LINUX and DEC.       06/04/97  (GWM/GWM)
*d,gi4b.1                                                         <42208>
*if def,linux,3
*d,gi4b.5                                                         <42212>
*i,gi4a.1                                                         <42213>
*if -def,linux,3
*/ ------------------------------------------------------------ secnd
*ident se4b2
*/ Uncomment the following for DVF on PCDOS.          02/17/98  (GWM/GWM)
*/ *d,se4a.4,se4a.6                                          <42820-42822>
*/ *d,se4a.8,se4a.15                                         <42837-42844>
*/ *if def,pcdos,1
*/       call cpu_time(t)
```

## Appendix C

Tables of values calculated using David Gierga's input files. Each suite of runs is presented. The final three tables compare to data showing only a marginal change in the results due to the upgrade.

**Table 1: Beryllium cell flux tallies without stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.31E+00 | 0.0181 | 2.33E+00 | 0.0212 | -0.0072 |
| 1 | 2.18E+00 | 0.0066 | 2.25E+00 | 0.0076 | -0.0334 |
| 2 | 1.91E+00 | 0.005 | 1.97E+00 | 0.0058 | -0.0275 |
| 4 | 1.37E+00 | 0.0041 | 1.40E+00 | 0.0048 | -0.0223 |
| 10 | 5.90E-01 | 0.004 | 5.98E-01 | 0.0047 | -0.0142 |
| 30 | 9.37E-02 | 0.0059 | 9.42E-02 | 0.007 | -0.0060 |
| 60 | 2.29E-02 | 0.0091 | 2.34E-02 | 0.0106 | -0.0222 |
| 90 | 1.08E-02 | 0.0123 | 1.11E-02 | 0.0144 | -0.0281 |

**Table 2: Beryllium detector flux tallies without stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.58E+00 | 0.0025 | 2.64E+00 | 0.003 | -0.0215 |
| 1 | 2.40E+00 | 0.0024 | 2.46E+00 | 0.0028 | -0.0218 |
| 2 | 2.05E+00 | 0.0024 | 2.10E+00 | 0.0027 | -0.0246 |
| 4 | 1.43E+00 | 0.0024 | 1.48E+00 | 0.0027 | -0.0328 |
| 10 | 5.90E-01 | 0.0028 | 6.06E-01 | 0.0033 | -0.0267 |
| 30 | 9.20E-02 | 0.0034 | 9.27E-02 | 0.004 | -0.0077 |
| 60 | 2.26E-02 | 0.003 | 2.28E-02 | 0.0034 | -0.0063 |
| 90 | 1.05E-02 | 0.0027 | 1.06E-02 | 0.0038 | -0.0114 |

**Table 3: Beryllium cell flux tallies with stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.33E+00 | 0.0181 | 2.39E+00 | 0.0209 | -0.0249 |
| 1 | 2.18E+00 | 0.0066 | 2.25E+00 | 0.0076 | -0.0319 |
| 2 | 1.92E+00 | 0.005 | 1.95E+00 | 0.0058 | -0.0158 |
| 4 | 1.39E+00 | 0.0041 | 1.42E+00 | 0.0048 | -0.0240 |
| 10 | 5.98E-01 | 0.004 | 6.15E-01 | 0.0046 | -0.0294 |
| 30 | 9.34E-02 | 0.006 | 9.53E-02 | 0.0069 | -0.0207 |
| 60 | 2.37E-02 | 0.009 | 2.38E-02 | 0.0105 | -0.0044 |
| 90 | 1.10E-02 | 0.0124 | 1.10E-02 | 0.0144 | -0.0055 |

**Table 4: Beryllium detector flux tallies with stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.59E+00 | 0.0026 | 2.64E+00 | 0.003 | -0.0211 |
| 1 | 2.41E+00 | 0.0025 | 2.47E+00 | 0.0028 | -0.0228 |
| 2 | 2.06E+00 | 0.0024 | 2.11E+00 | 0.0027 | -0.0269 |
| 4 | 1.44E+00 | 0.0024 | 1.49E+00 | 0.0027 | -0.0288 |
| 10 | 6.03E-01 | 0.0028 | 6.17E-01 | 0.0032 | -0.0235 |
| 30 | 9.34E-02 | 0.0034 | 9.49E-02 | 0.004 | -0.0158 |
| 60 | 2.34E-02 | 0.0038 | 2.33E-02 | 0.0037 | 0.0014 |
| 90 | 1.08E-02 | 0.0027 | 1.09E-02 | 0.004 | -0.0176 |

**Table 5: Aliminum cell flux tallies without stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0.1 | 3.04E+00 | 0.0151 | 2.94E+00 | 0.0168 | 0.0325 |
| 1 | 2.81E+00 | 0.0056 | 2.84E+00 | 0.0061 | -0.0100 |
| 2 | 2.51E+00 | 0.0042 | 2.55E+00 | 0.0045 | -0.0149 |
| 4 | 1.94E+00 | 0.0034 | 1.95E+00 | 0.0037 | -0.0081 |
| 10 | 1.03E+00 | 0.0029 | 1.04E+00 | 0.0032 | -0.0089 |
| 30 | 2.66E-01 | 0.0034 | 2.68E-01 | 0.0037 | -0.0064 |
| 60 | 7.28E-02 | 0.0049 | 7.14E-02 | 0.0054 | 0.0199 |
| 90 | 3.29E-02 | 0.0068 | 3.28E-02 | 0.0075 | 0.0039 |

**Table 6: Aliminum detector flux tallies without stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0.1 | 3.24E+00 | 0.0022 | 3.28E+00 | 0.0023 | -0.0107 |
| 1 | 3.06E+00 | 0.0021 | 3.09E+00 | 0.0023 | -0.0103 |
| 2 | 2.68E+00 | 0.002 | 2.71E+00 | 0.0022 | -0.0101 |
| 4 | 2.02E+00 | 0.002 | 2.04E+00 | 0.0022 | -0.0071 |
| 10 | 1.04E+00 | 0.0024 | 1.05E+00 | 0.0026 | -0.0053 |
| 30 | 2.67E-01 | 0.0027 | 2.66E-01 | 0.003 | 0.0055 |
| 60 | 7.28E-02 | 0.0028 | 7.25E-02 | 0.0029 | 0.0031 |
| 90 | 3.28E-02 | 0.0027 | 3.26E-02 | 0.003 | 0.0056 |

**Table 7: Aliminum cell flux tallies with stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0.1 | 2.99E+00 | 0.0153 | 2.96E+00 | 0.0188 | 0.0110 |
| 1 | 2.80E+00 | 0.0056 | 2.84E+00 | 0.0068 | -0.0125 |
| 2 | 2.49E+00 | 0.0042 | 2.54E+00 | 0.0051 | -0.0190 |
| 4 | 1.93E+00 | 0.0034 | 1.96E+00 | 0.0041 | -0.0132 |
| 10 | 1.03E+00 | 0.0029 | 1.04E+00 | 0.0035 | -0.0117 |
| 30 | 2.69E-01 | 0.0034 | 2.68E-01 | 0.0041 | 0.0040 |
| 60 | 7.32E-02 | 0.0049 | 7.26E-02 | 0.006 | 0.0081 |
| 90 | 3.33E-02 | 0.0068 | 3.32E-02 | 0.0083 | 0.0004 |

**Table 8: Aliminum detector flux tallies with stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0.1 | 3.22E+00 | 0.0022 | 3.26E+00 | 0.0026 | -0.0117 |
| 1 | 3.04E+00 | 0.0021 | 3.08E+00 | 0.0025 | -0.0106 |
| 2 | 2.67E+00 | 0.002 | 2.70E+00 | 0.0025 | -0.0123 |
| 4 | 2.01E+00 | 0.002 | 2.03E+00 | 0.0025 | -0.0117 |
| 10 | 1.04E+00 | 0.0023 | 1.05E+00 | 0.0028 | -0.0060 |
| 30 | 2.67E-01 | 0.0027 | 2.67E-01 | 0.0033 | 0.0011 |
| 60 | 7.29E-02 | 0.0027 | 7.21E-02 | 0.0032 | 0.0105 |
| 90 | 3.31E-02 | 0.0027 | 3.29E-02 | 0.0033 | 0.0072 |

**Table 9: Lead cell flux tallies without stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.72E+00 | 0.0222 | 2.65E+00 | 0.0199 | 0.0260 |
| 1 | 2.51E+00 | 0.0082 | 2.49E+00 | 0.0072 | 0.0060 |
| 2 | 2.28E+00 | 0.006 | 2.24E+00 | 0.0054 | 0.0168 |
| 4 | 1.84E+00 | 0.0048 | 1.82E+00 | 0.0042 | 0.0090 |
| 10 | 1.13E+00 | 0.0038 | 1.13E+00 | 0.0034 | 0.0063 |
| 30 | 4.34E-01 | 0.0037 | 4.37E-01 | 0.0032 | -0.0079 |
| 60 | 1.44E-01 | 0.0048 | 1.48E-01 | 0.0042 | -0.0231 |
| 90 | 6.03E-02 | 0.007 | 6.23E-02 | 0.0061 | -0.0335 |

**Table 10: Lead detector flux tallies without stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.81E+00 | 0.0032 | 2.80E+00 | 0.0028 | 0.0032 |
| 1 | 2.68E+00 | 0.0031 | 2.67E+00 | 0.0027 | 0.0054 |
| 2 | 2.40E+00 | 0.003 | 2.38E+00 | 0.0027 | 0.0080 |
| 4 | 1.90E+00 | 0.0031 | 1.88E+00 | 0.0027 | 0.0104 |
| 10 | 1.14E+00 | 0.0036 | 1.13E+00 | 0.0032 | 0.0083 |
| 30 | 4.31E-01 | 0.0045 | 4.37E-01 | 0.004 | -0.0142 |
| 60 | 1.43E-01 | 0.0053 | 1.47E-01 | 0.0045 | -0.0233 |
| 90 | 5.86E-02 | 0.0065 | 6.13E-02 | 0.0055 | -0.0464 |

**Table 11: Lead cell flux tallies with stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.64E+00 | 0.0207 | 2.63E+00 | 0.0199 | 0.0045 |
| 1 | 2.46E+00 | 0.0076 | 2.48E+00 | 0.0073 | -0.0086 |
| 2 | 2.25E+00 | 0.0056 | 2.24E+00 | 0.0054 | 0.0086 |
| 4 | 1.83E+00 | 0.0044 | 1.83E+00 | 0.0042 | 0.0049 |
| 10 | 1.12E+00 | 0.0036 | 1.12E+00 | 0.0034 | 0.0041 |
| 30 | 4.30E-01 | 0.0034 | 4.34E-01 | 0.0032 | -0.0098 |
| 60 | 1.43E-01 | 0.0045 | 1.47E-01 | 0.0042 | -0.0324 |
| 90 | 6.03E-02 | 0.0064 | 6.21E-02 | 0.0061 | -0.0296 |

**Table 12: Lead detector flux tallies with stainless steel**

| angle | Flux(4b) | Rel. Err. | Flux(nu) | Rel. Err. | rel. diff. |
|---|---|---|---|---|---|
| 0 | 2.80E+00 | 0.0029 | 2.79E+00 | 0.0028 | 0.0028 |
| 1 | 2.67E+00 | 0.0028 | 2.65E+00 | 0.0027 | 0.0070 |
| 2 | 2.38E+00 | 0.0028 | 2.37E+00 | 0.0027 | 0.0022 |
| 4 | 1.88E+00 | 0.0028 | 1.87E+00 | 0.0027 | 0.0061 |
| 10 | 1.13E+00 | 0.0032 | 1.13E+00 | 0.0032 | -0.0018 |
| 30 | 4.31E-01 | 0.0042 | 4.33E-01 | 0.004 | -0.0057 |
| 60 | 1.42E-01 | 0.0046 | 1.46E-01 | 0.0045 | -0.0278 |
| 90 | 6.00E-02 | 0.0117 | 6.14E-02 | 0.0058 | -0.0226 |

**Table 13: Data comparison with Faddegon for Lead**

| | 4bnu cell | | 4bnu detect | | Faddegon |
|---|---|---|---|---|---|
| 0 | 2.63E+00 | 0.0199 | 2.79E+00 | 0.0028 | 2.92E+00(5.0) |
| 1 | 2.48E+00 | 0.0073 | 2.65E+00 | 0.0027 | 2.80E+00(5.0) |
| 2 | 2.24E+00 | 0.0054 | 2.37E+00 | 0.0027 | 2.48E+00(5.0) |
| 4 | 1.83E+00 | 0.0042 | 1.87E+00 | 0.0027 | 1.99E+00(5.0) |
| 10 | 1.12E+00 | 0.0034 | 1.13E+00 | 0.0032 | 1.2E+00(5.0) |
| 30 | 4.37E-01 | 0.0032 | 4.37E-01 | 0.004 | 4.47E-01(5.0) |
| 60 | 1.48E-01 | 0.0042 | 1.47E-01 | 0.0045 | 1.29E-01(5.0) |
| 90 | 6.23E-02 | 0.0061 | 6.13E-02 | 0.0055 | 5.19E-02(7.0) |

**Table 14: Data comparison with Faddegon for Beryllium**

| | 4bnu cell | | 4bnu detect | | Faddegon |
|---|---|---|---|---|---|
| 0 | 2.39E+00 | 0.0209 | 2.64E+00 | 0.003 | 2.73E+00(5.1) |
| 1 | 2.25E+00 | 0.0076 | 2.47E+00 | 0.0028 | 2.57E+00(5.1) |
| 2 | 1.95E+00 | 0.0058 | 2.11E+00 | 0.0027 | 2.14E+00(5.1) |
| 4 | 1.42E+00 | 0.0048 | 1.49E+00 | 0.0027 | 1.54E+00(5.0) |
| 10 | 6.15E-01 | 0.0046 | 6.17E-01 | 0.0032 | 6.30E-01(5.1) |
| 30 | 9.42E-02 | 0.007 | 9.27E-02 | 0.004 | 9.49E-02(5.1) |
| 60 | 2.34E-02 | 0.0106 | 2.28E-02 | 0.0034 | 2.38E-02(5.9) |
| 90 | 1.11E-02 | 0.0144 | 1.06E-02 | 0.0038 | 1.06E-02(7.0) |

**Table 15: Data comparison with Faddegon for Aluminum**

| | 4bnu cell | | 4bnu detect | | Faddegon |
|---|---|---|---|---|---|
| 0 | 2.96E+00 | 0.0188 | 3.26E+00 | 0.0026 | 3.42E+00(5.0) |
| 1 | 2.84E+00 | 0.0068 | 3.08E+00 | 0.0025 | 3.21E+00(5.0) |
| 2 | 2.54E+00 | 0.0051 | 2.70E+00 | 0.0025 | 2.78E+00(5.0) |
| 4 | 1.96E+00 | 0.0041 | 2.03E+00 | 0.0025 | 2.14E+00(5.0) |
| 10 | 1.04E+00 | 0.0035 | 1.05E+00 | 0.0028 | 1.06E+00(5.0) |
| 30 | 2.68E-01 | 0.0037 | 2.66E-01 | 0.003 | 2.65E-01(5.0) |
| 60 | 7.14E-02 | 0.0054 | 7.25E-02 | 0.0029 | 6.66E-02(6.0) |
| 90 | 3.28E-02 | 0.0075 | 3.26E-02 | 0.003 | 2.87E-02(6.0) |

References:


1. J.F. Briesmeister, Ed., "MCNP™ - A General Monte Carlo N-Particle Transport Code, Version 4B," Los Alamos National Laboratory report, LA-12625-M (1997).

2. J. A. Halbleib, R. P. Kensek, T. A. Mehlhorn, G. D. Valdez, S. M. Seltzer, and M. J. Berger, "ITS Version 3.0: Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes," SAND91-1634 (1992).

3. M. Berger, "Electron Stopping Powers for Transport Calculations," in *Monte Carlo Transport of Electrons and Photons*, T. M. Jenkins, W. R. Nelson, and A. Rindi, eds., Plenum Press, (1987).

4. S. M. Seltzer, "Cross Sections for Bremsstrahlung Production and Electron-Impact Ionization," in *Monte Carlo Transport of Electrons and Photons*, T. M. Jenkins, W. R. Nelson, and A. Rindi, eds., Plenum Press, (1987).

5. W. R. Nelson, H. Hirayama, and D. W. O. Rogers, "The EGS4 Code System," SLAC-265, December, (1985).

6. J. A.Halbleib and T. A. Melhorn, "ITS: The Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes," SAND84-0573 (1984).

7. D. P. Gierga and K. J. Adams, "Electron/Photon Verification Calculations Using MCNP4B," LA in preparation (1999).

8. J. J. DeMarco, T. D. Soldberg, R. E. Wallace, and J. B. Smathers, "A Verification of the Monte Carlo Code MCNP for Thick Target Bremsstrahlung Calculations," *Med. Phys.* **22**, p. 11-16, (1995).

9. B. A. Faddegon and D. W. O Rogers, "Comparisons of Thick-Target Bremsstrahlung Calculations by EGS4/PRESTA and ITS version 2.1," *Nuc. Inst. Meth.* A **327**, p. 556-565 (1993).

10. T. Kauppila, K. Mueller, and C. Lebeda, "Radiographic Analysis of a Copper Step Wedge at 15 MeV Bremsstrahlung Energy," LA-UR-97-4801, (1998)

11. S. T. Perkins, D. E. Cullen,S. M. Seltzer, "Tables and Graphs of Electron-Interaction Cross Sections from 10 eV to 100 GeV from the LLNL Evaluated Atomic Data Library (EEDL), Z=1 - 100," UCRL-50400, Vol 31 (1991)

12. ICRU (International Commission of Radiation Units and Measurements) Report 37 (1984).

13. R. M. Sternheimer and R. F. Peierls, "General Expression for the Density Effect for the Ionization of Charged Particles," *Phys. Rev. B* **3**, p. 3681-3692, (1971).

14. R. M. Sternheimer, S. M. Seltzer and M. J. Berger, "Density effect for the ionization loss of charged particles in various substances," *Phys. Rev. B* **26**, p. 6067-6076, (1982).

15. H. W. Koch and J. W. Motz, "Bremsstrahlung Cross-Section Formulas and Related Data," Rev. Mod. Phys. **31** p. 920, (1959)

16. M. J. Berger and S. M. Seltzer, "Bremsstrahlung and Photoneutrons from Thick Tungsten and Tantalum Targets," *Phys. Rev.* C **2**, p. 621-631, (1970).

17. S. M. Seltzer and M. J. Berger, "Bremsstrahlung Spectra from Electron Interactions with Screened atomic Nuclei and Orbital Electrons", Nucl. Instr. Meth. B12 (1985) 95

18. S. M. Seltzer and M. J. Berger, "Bremsstrahlung Energy Spectra from Electrons with Kinetic Energy 1 keV - 10 GeV Incident on Screened Nuclei and Orbital Electrons of Neutral Atoms with Z=1 to 100", Atom. Data and Nuc. Data Tables 35, (1986) 345

19. H. K. Tseng and R. H. Pratt, "Exact Screened Calculations of Atomic-Field Bremsstrahlung," *Phys. Rev.* **A3** (1971) 100

20. H. K. Tseng and R. H. Pratt, "Electron Bremsstrahlung from Neutral Atoms," *Phys. Rev. Lett.* **33** (1974) 516

21. H. K. Tseng and R. H. Pratt, "Electron Bremsstrahlung Energy Spectra Above 2 MeV," *Phys. Rev.* **A19** (1979) 1525

22. R. H. Pratt, H. K. Tseng, C. M. Lee, L. Kissel, C. MacCallum, and M. Riley, "Bremsstrahlung Energy Spectra from Electrons of Kinetic Energy 1 keV $\leq T \leq$ 2000 keV Incident on Neutral Atoms $2 < Z < 92$," *Atom. Data and Nuc. Data Tables* **20**, (1977) 175; errata in **26** (1981) 477

23. H. Davies, H. A. Bethe, and L. C. Maximom, "Theory of Bremsstrahlung and Pair Production. II. Integral Cross Section for Pair Production," *Phys. Rev.* **93** (1954) 788; and H. Olsen, "Outgoing and Ingoing Waves in Final States and Bremsstrahlung," *Phys. Rev.* **99** (1955) 1335.

24. E. Haug, "Bremsstrahlung and Pair Production in the field of free Electrons," *Z. Naturforsch.* **30a** (1975) 1099

25. J. H. Hubbell, W. J. Veigele, E. A. Briggs, R. T. Brown, D. T. Cromer, and R. J. Howerton, "Atomic Form Factors, Incoherent Scattering Functions, and Photon Scattering Cross Sections," *J. Phys. Chem. Ref. Data* **4** (1975) 471; and J. H. Hubbell and I. Overbo, "Relativistic Atomic Form Factors and Photon Coherent Scattering Cross sections," *J. Phys. Chem. Ref. Data* **8** (1979) 69

26. G. Elwert, "Verscharte Berechnung von Intensitat und Polarisation im Kontinuierlichen Rontgenspektrum," *Ann. Physick* **34** (1939)178

27. R. J. Jabbur and R. H. Pratt, "High-Frequency Region of the Spectrum of Electron and Positron Bremsstrahlung," *Phys. Rev.* **129** (1963) 184; and "High-Frequency Region of the Spectrum of Electron and Positron Bremsstrahlung II," *Phys. Rev.* **133** (1964) 1090

28. T. A. Carlson, *Photoelectron and Auger Spectroscopy*, Plenum Press, New York, N.Y. 1975

29. C. J. Everett and E. D. Cashwell, "MCP Code Fluorescence-Routine Revision", LA-5240-MS (1973)

30. W. L. Chadsey, "POEM: A fast Monte Carlo code for the calculation of X-ray photoemission and transistion zone dose and current," SAI-74-624-WA, AFCRL-TR-75-0324, (1975)

31. S. T. Perkins, D. E. Cullen, M. H. Chen, J. H. Hubbell, J. Rathkopf, J. Scofield, "Tables and Graphs of Atomic Subshell and Relaxation Data Derived from the LLNL Evaluated Atomic Data Library (EADL), Z=1 - 100," UCRL-50400, Vol 30 (1991)

32. A. F. Bielajew, R. Mohan, and C. S. Chou, "Improved bremsstrahlung photon angular sampling in the EGS4 code system," PIRS-0203, (1989)

34. D. E. Cullen, M. H. Chen, J. H. Hubbell, S. T. Perkins, E. F. Plechaty, J. Rathkopf, and J. Scofield, "Tables and Graphs of Photon-Interaction Cross Sections from 10 eV to 100 GeV from the LLNL Evaluated Atomic Data Library (EPDL), Z=1 - 100," UCRL-50400, Vol 6 Parts A & B (1989)

**Distribution list:**

X-5 files
K. J. Adams, X-5 F663
A. R. Heath, X-5 F663
T. Seed, X-5 F663
Eolus Team
Data Team

without appendices:

J. A. Favorite, X-5 F663
C. F. Lebeda, X-5 F663

J. S. Saracino, X-4 F664
A. R. Mathews, X-4 F664

H. G. Hughes, X-6 D409
L. S. Waters, APT-TPO H816