NISTIR 5477

# Updated Calculations for Routine Space-Shielding Radiation Dose Estimates: SHIELDOSE-2

**Stephen M. Seltzer**

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Gaithersburg, MD 20899

Prepared for:

National Aeronautics
and Space Administration
Washington, DC 20024

NIST

NISTIR 5477

# Updated Calculations for Routine Space-Shielding Radiation Dose Estimates: SHIELDOSE-2

**Stephen M. Seltzer**

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Gaithersburg, MD 20899

Prepared for:

National Aeronautics
and Space Administration
Washington, DC 20024

December 1994

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

# Updated Calculations for Routine Space-Shielding Radiation Dose Estimates[1]: SHIELDOSE-2

Stephen M. Seltzer
Ionizing Radiation Division
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

## Abstract

New, more-extensive, depth-dose distributions for electrons, electron-bremsstrahlung, and protons have been calculated, based on improvements in cross-section information since the development of the original SHIELDOSE code. The new database covers incident electron energies from 5 keV to 50 MeV, with the bremsstrahlung tail calculated for depths out to 50 g/cm$^2$, and incident proton energies from 10 keV to 10 GeV. Effects of nuclear interactions on proton depth-dose distributions in aluminum shields have been estimated, and options are provided to include these approximations in test calculations. In addition to the absorbed dose in aluminum, the dose in small volumes of graphite, Si, air, bone, $CaF_2$, GaAs, LiF, $SiO_2$, tissue or water can be evaluated. The functionality of new code is much the same as the old code; however, the resultant dose estimates as a function of depth in aluminum spacecraft can be somewhat different. Through the use of a companion code based on approximate transformations, results can be extended beyond the dose as a function of depth in plane slabs and at centers of solid spheres to include the dose at off-center points in a solid sphere and at the inner surface of spherical shells.

[1] This work was supported by NASA Radiation Health Program, Contract T-9311R.

## Introduction

Quite sophisticated and accurate computer codes are available to calculate the transport of electrons, the electron-produced bremsstrahlung, and protons through spacecraft material to determine the absorbed dose in a volume of interest. Although such calculations are able to take into account complicated multi-material geometry, they usually require a significant understanding of radiation transport physics in conjunction with knowledge of the often complex geometric details involved, and can involve many hours of computation for each problem. In many situations, it is of value to be able to routinely perform rapid dose estimates based on accurate transport data, even if one is restricted to simple geometries.

The SHIELDOSE code package [1,2] was developed in the late 1970's to provide for such rapid calculations of absorbed dose as a function of depth in the aluminum shielding material of spacecraft, given the electron and proton fluence spectra encountered in orbit. The calculation is based on consideration of a single shielding material (the aluminum that represents the bulk of most spacecraft) and assumes that the fluence of incident radiation is isotropic (at least in the time-averaged sense). Shield geometry is basically limited to simple plane slabs, but — under suitable transformations [3] — the calculated dose distributions can be applied to spherical solids and shells. With these assumptions, the problem was reduced to the development of a basic set of depth-dose distributions in aluminum plane slabs for monoenergetic incident radiation. These could then be used repeatedly to predict the dose for any fluence spectra. Furthermore, with information from the basic calculations on the depth-dependent radiation fluence spectra established in the slab, the dose on small non-perturbing volumes of material other than aluminum can — at least approximately — be estimated. Because the transport of electrons and the associated bremsstrahlung is a complicated process, the basic depth-dose data was developed from detailed Monte Carlo calculations. The proton results were based on calculations in the straight-ahead, continuous-slowing-down approximation.

The SHIELDOSE code has found widespread use for dose estimates by the space-radiation-effects community. In the nearly 15 years since the original database was developed, much of the underlying transport cross-section information and algorithms have been improved. With NASA support, new basic calculations have been done using current information to update and expand the database, and to improve the software. This report outlines the new work and compares results with those from the original SHIELDOSE.

## The Database

The new database calculations for electrons and bremsstrahlung are based on the up-to-date cross-section information and Monte Carlo algorithms described in [4], which also form the basis for the latest version of ITS [5]. The Monte Carlo results, which cover incident electron kinetic energies from 5 keV to 50 MeV, are based on much larger numbers of histories than the older calculations, and extend to aluminum depths of 50 g/cm$^2$. The old database covers electron incident energies from 100 keV to 10 MeV for the direct electron ionization dose and from 20 keV to 20 MeV for the bremsstrahlung dose, extending to aluminum depths of 30 g/cm$^2$.

The new proton calculations are based on the use of recent critically-evaluated tables of proton stopping powers and ranges [6], and cover proton energies from 10 keV to 10 GeV. The old database covers proton energies from 2 to 5000 MeV. The original assumption of straight-ahead, continuous-slowing-down-approximation (csda) penetration has been maintained, and depth-dose distributions have been prepared both with nuclear interactions taken into account in the aluminum shield, as well with the neglect of nuclear interactions (as done in the old database).

In addition to the absorbed dose in aluminum, the new work includes estimates of the dose in small volumes of graphite, Si, air, bone, $CaF_2$, GaAs, LiF, $SiO_2$, tissue and water. The old SHIELDOSE includes only Al, Si, $SiO_2$, and water.

*Electron Component*

Monte Carlo calculations for the electron component were carried out using an updated version of the ETRAN code [4], taking into account energy-loss straggling, multiple-elastic-scattering angular deflections, the production and transport of all generations of knock-on electrons, bremsstrahlung photons, and characteristic x rays and Auger electrons subsequent to ionization events. Calculations were done for an incident angular distribution that corresponds to an isotropic fluence of electrons incident on a semi-infinite plane slab aluminum target, with monoenergetic initial kinetic energies of 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0, and 50.0 MeV. Each run was based on the analysis of 100k incident electron histories, with all radiations followed until either they escaped the target or their energy fell below 1 keV. Scores, as a function of depth out to 1.25 times the mean electron range, were kept of the absorbed dose and of the forward-directed and backward-directed fluence spectra of electrons. As the bremsstrahlung component was developed separately to be added to the electron component, secondary electrons from bremsstrahlung photons were not included in these scores.

The absorbed-dose distributions in the semi-infinite aluminum plane slab targets, $D_{\infty}^{\mathrm{Al}}(z,T_o)$, were scaled and smoothed as functions of both depth $z$ and incident energy $T_o$ using least-square cubic splines [7]. This was done to facilitate the interpolation done over incident spectra in the SHIELDOSE code. To convert these depth-dose distributions to those for detector volumes other than aluminum and for finite-thickness aluminum slabs, the cavity theory of Spencer and Attix [8] was used. Here the cavity is assumed to be of small size so as not to perturb the electron fluence, and we further assume that the electron fluence at the transmission face of a finite-thickness plane slab is closely approximated by the forward-directed electron fluence at the corresponding depth in a slab of much greater thickness.

Then for semi-infinite slabs,

$$\frac{D_{\infty}^{\mathrm{det}}(z,T_\mathrm{o})}{D_{\infty}^{\mathrm{Al}}(z,T_\mathrm{o})} = \frac{\displaystyle\int_{\Delta}^{T_\mathrm{o}} F_\mathrm{e}^{\mathrm{tot}}(T,z,T_\mathrm{o}) \left[\frac{L(T,\Delta)}{\rho}\right]^{\mathrm{det}} \mathrm{d}T + F_\mathrm{e}^{\mathrm{tot}}(\Delta,z,T_\mathrm{o}) \left[\frac{S(\Delta)}{\rho}\right]^{\mathrm{det}} \Delta}{\displaystyle\int_{\Delta}^{T_\mathrm{o}} F_\mathrm{e}^{\mathrm{tot}}(T,z,T_\mathrm{o}) \left[\frac{L(T,\Delta)}{\rho}\right]^{\mathrm{Al}} \mathrm{d}T + F_\mathrm{e}^{\mathrm{tot}}(\Delta,z,T_\mathrm{o}) \left[\frac{S(\Delta)}{\rho}\right]^{\mathrm{Al}} \Delta} \ , \qquad (1)$$

and for finite-thickness slabs,

$$\frac{D_{-}^{\mathrm{det}}(z,T_\mathrm{o})}{D_{\infty}^{\mathrm{Al}}(z,T_\mathrm{o})} = \frac{\displaystyle\int_{\Delta}^{T_\mathrm{o}} F_\mathrm{e}^{\mathrm{for}}(T,z,T_\mathrm{o}) \left[\frac{L(T,\Delta)}{\rho}\right]^{\mathrm{det}} \mathrm{d}T + F_\mathrm{e}^{\mathrm{for}}(\Delta,z,T_\mathrm{o}) \left[\frac{S(\Delta)}{\rho}\right]^{\mathrm{det}} \Delta}{\displaystyle\int_{\Delta}^{T_\mathrm{o}} F_\mathrm{e}^{\mathrm{tot}}(T,z,T_\mathrm{o}) \left[\frac{L(T,\Delta)}{\rho}\right]^{\mathrm{Al}} \mathrm{d}T + F_\mathrm{e}^{\mathrm{tot}}(\Delta,z,T_\mathrm{o}) \left[\frac{S(\Delta)}{\rho}\right]^{\mathrm{Al}} \Delta} \ , \qquad (2)$$

where $F_\mathrm{e}(T,z,T_\mathrm{o})$ is the electron fluence spectrum (*for*ward and *tot*al, as indicated) as a function of spectral energy $T$ and depth $z$, $L(T,\Delta)/\rho$ is the electron restricted mass collision stopping power, restricted to energy losses less than $\Delta$, $S(T)/\rho$ is the unrestricted mass collision stopping power, and $\Delta$ is a cut-off energy generally associated with the size of the cavity. The values of $\Delta$ were chosen as max($T_\mathrm{o}$/5000, 1 keV), which places them between 10 and 1 keV; the calculated dose ratios should be rather insensitive to these choices.

Electron stopping powers were calculated according to the methods given in ICRU Report 37 [9]; the composition of bone and tissue was assumed that of cortical bone and of soft tissue given in ICRU Report 44 [10]. Note that the assumptions of our calculations lead to results that pertain only to suitably small volumes of detector material in the aluminum absorber. For example, the results might apply to a bone- or tissue-equivalent detector rather than to an extended biological target. Some scaling and smoothing of the stopping-power ratios were done to facilitate interpolation.

### *Bremsstrahlung Component*

The ETRAN Monte Carlo calculations were carried out in a fashion similar to that for the electron component, treating the same set of monoenergetic incident electron kinetic energies. In these cases, however, each calculation was based on 100k incident electrons and a sample of 10M emitted bremsstrahlung photons, followed down to an energy of 1 keV. The calculation treats the usual photon interactions: pair and triplet production, incoherent (Compton) scattering, coherent scattering, and photoelectric absorption. These Monte Carlo simulations improve on our older calculations in that updated cross-section information is used and coherent scattering and binding corrections in incoherent scattering are included. Secondary charged particles were followed to include their contribution to bremsstrahlung production. The photon fluence spectra were scored out to depths of 50 g/cm$^2$ of aluminum.

With the assumption of charged-particle equilibrium, the absorbed dose from bremsstrahlung photons in the aluminum absorber was calculated according to

$$D_{\infty}^{\mathrm{Al}}(z,T_o) = \int_{o}^{T_o} F_{\gamma}^{\mathrm{tot}}(k,z,T_o)\, k \left[\frac{\mu_{\mathrm{en}}(k)}{\rho}\right]^{\mathrm{Al}} \mathrm{d}k \;, \tag{3}$$

where $F_{\gamma}(k,z,T_o)$ is the photon fluence spectrum as a function of spectral energy $k$ and depth $z$, and $\mu_{\mathrm{en}}(k)/\rho$ is the photon mass energy-absorption coefficient. Dose ratios were obtained for semi-infinite slabs from,

$$\frac{D_{\infty}^{\mathrm{det}}(z,T_o)}{D_{\infty}^{\mathrm{Al}}(z,T_o)} = \frac{\displaystyle\int_{o}^{T_o} F_{\gamma}^{\mathrm{tot}}(k,z,T_o) \left[\frac{\mu_{\mathrm{en}}(k)}{\rho}\right]^{\mathrm{det}} \mathrm{d}k}{\displaystyle\int_{o}^{T_o} F_{\gamma}^{\mathrm{tot}}(k,z,T_o) \left[\frac{\mu_{\mathrm{en}}(k)}{\rho}\right]^{\mathrm{Al}} \mathrm{d}k} \;, \tag{4}$$

and for finite-thickness slabs from,

$$\frac{D_{-}^{\mathrm{det}}(z,T_o)}{D_{\infty}^{\mathrm{Al}}(z,T_o)} = \frac{\displaystyle\int_{o}^{T_o} F_{\gamma}^{\mathrm{for}}(k,z,T_o) \left[\frac{\mu_{\mathrm{en}}(k)}{\rho}\right]^{\mathrm{det}} \mathrm{d}k}{\displaystyle\int_{o}^{T_o} F_{\gamma}^{\mathrm{tot}}(k,z,T_o) \left[\frac{\mu_{\mathrm{en}}(k)}{\rho}\right]^{\mathrm{Al}} \mathrm{d}k} \;. \tag{5}$$

Values for $\mu_{\mathrm{en}}(k)/\rho$ were obtained from calculations recently outlined by Seltzer [11]. Some scaling and smoothing of the $\mu_{\mathrm{en}}$ ratios were done to facilitate interpolation.

For detector materials other than aluminum, the use of dose ratios defined in terms of the photon mass energy-absorption coefficient raises an issue of potential importance. The assumption here is that the detector size is small enough that it does not significantly perturb the photon fluence, but large enough that the energy absorbed in the detector is predominantly from secondary electrons produced by the photons in the detector material and not from those produced in the aluminum absorber or wall material. As an example, for a small graphite-wall air ionization chamber it seems more correct to consider the detector to be the graphite wall, as the ionization in the air is predominantly from the electrons produced in the graphite build-up material. Other situations can be more complicated and may not be adequately treated in our approximation.

*Proton Component*

Depth-dose distributions were calculated for an isotropic fluence of protons incident on a semi-infinite plane slab aluminum targets, with monoenergetic initial kinetic energies of 0.01, 0.015, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.1, .... , 1000, 1500, 2000, 3000, 4000, 5000, 6000, 8000, and 10000 MeV. Current information on proton stopping powers and ranges in the materials of interest has been taken from the recent ICRU Report [6]. A special set of calculations was needed for GaAs which was not included in the work of the ICRU Report Committee. The compositions for tissue and bone were those for ICRU striated muscle and for ICRP cortical bone, both given in reference [9]. In the straight-ahead (neglect of elastic scattering angular deflections) and continuous-slowing-down (neglect of energy-loss straggling) approximations, the depth dose can be calculated from a relatively simple numerical evaluation. A comparison of these results with those from a full Monte Carlo calculation confirm the overall adequacy of using these approximations for such calculations, particularly for isotropic fluences. In the straight-ahead approximation (and very nearly in the actual case), there is no distinction between forward-directed and total fluence. The dose in the various detector materials has been obtained from the integral of the product of the fluence spectrum and the stopping power:

$$\frac{D_{\infty}^{\mathrm{det}}(z,T_o)}{D_{\infty}^{\mathrm{Al}}(z,T_o)} = \frac{\int_{o}^{T_o} F_{\mathrm{p}}^{\mathrm{tot}}(T,z,T_o) \left[\frac{S(T)}{\rho}\right]^{\mathrm{det}} \mathrm{d}T}{\int_{o}^{T_o} F_{\mathrm{p}}^{\mathrm{tot}}(T,z,T_o) \left[\frac{S(T)}{\rho}\right]^{\mathrm{Al}} \mathrm{d}T} , \qquad (6)$$

where here $S(T)/\rho$ is the proton mass stopping power.

The original SHIELDOSE calculations ignored effects of nuclear interactions, i.e., the attenuation of the primary proton beam and the production and transport of nuclear-reaction products. Earlier work by Santoro *et al.* [12] indicated that the neglect of such effects tended to give conservative (somewhat higher) dose estimates, and that the differences can be significant, perhaps as large as 30-40% for hard proton spectra and shield thicknesses of 50 g/cm$^2$. It is difficult to develop guidelines based on such essentially anecdotal findings, so it was considered worthwhile to incorporate in the new code an approximate model of nuclear interactions to provide at least a crude estimate of the effects.

The attenuation of the primary proton beam due to nonelastic nuclear interactions requires knowledge of the total nonelastic-nuclear-interaction cross section. The cross-section data for aluminum shown in Fig. 1 include measured values compiled by Bauhoff [13], the fitted results of intranuclear-cascade calculations [14], the results adopted by Janni [15], the values calculated by Townsend and Wilson [16], and the curve adopted here. Our solid curve has been drawn mainly as a fit to the experimental data, constrained by the high-energy asymptotic value of approximately 456 mb [17] and by a threshold implied by the first excited state in nuclear level diagrams[2]. From the adopted total cross sections, one can calculate the attenuation of primary protons along the direction of travel, as shown in Fig. 2. Integrating over the proton slowing down, the fraction of the number and of the energy of primary protons lost in nuclear interactions can be obtained and is shown in Fig. 3.

The fate of the energy lost in nuclear interactions is, however, a more complicated problem. Results from reference [14] indicate that the relative abundance of nuclear-reaction products in aluminum should be approximately the same as that for oxygen, allowing the use of data recently developed for protons in water [18]. Assuming that the charged-particle nuclear secondaries are short-ranged, their energy can be assumed to be absorbed at the point of production[3]. Figure 4 gives the fraction of primary energy converted to charged-particle secondaries and assumed to be locally absorbed, as obtained from reference [18] and assigned here to proton collisions in aluminum. Then, taking into account attenuation and the local deposition of charged-particle secondaries, an absorbed dose can be calculated as a function of distance along the direction of travel. Such depth-dose curves, with and without nuclear interactions, are shown in Fig. 5, and indicate rather small effects for protons with energies up to a few hundred MeV, but rather significant effects at the higher energies. That proton spectra for typical applications fall off rapidly at such high energies would seem to mitigate the effect of our approximations.

The energy converted to secondary neutron energy (and de-excitation gamma-ray energy) cannot be assumed locally absorbed. The amount of energy involved is shown in Fig. 6 from integrations over the proton slowing down, assuming the partition of primary energy to nuclear-reaction products from reference [18]. These results indicate only a relatively small fraction of primary energy is converted to secondary neutrons for protons up to about one hundred MeV, somewhat larger for the less-abundant high-energy proton. One option is to assume that such energy completely escapes the region of interest. Another option that has been considered is to assume - rather crudely - that the total neutron energy produced by the incident proton is exponentially distributed (from the entrance surface) with a attenuation coefficient of 0.03 $cm^2/g$ in aluminum. This numerical value was taken from tables of the "removal" cross section for fast neutrons in aluminum in references [19-21], and is roughly consistent with the "relaxation" length reported for fast neutrons in aluminum [22].

Figure 7 shows proton absorbed-dose distributions in aluminum from calculations (a) neglecting nuclear interactions, (b) including nuclear attenuation and only our assumption of the local absorption of secondary charged-particle energy, and (c) as in (b) plus adding our approximate distribution of deposited neutron energy. The calculations pertain to protons with simple exponential spectra, extending from 1 MeV to 10 GeV, characterized by e-folding energies $\alpha$ from 10 to 200 MeV. Our estimates of the fraction of the beam energy converted to neutron energy are 0.000033, 0.00061, 0.0090, 0.033, and 0.084, for $\alpha$-values of 10, 20, 50,

[2] Values of the cross section at low energies are rather unimportant in the present application because the effects are small and tend to occur toward the end of the proton range, involving little energy.

[3] Secondary proton spectra have tails extending to energies close to the primary energy, for which our negligible-range approximation is no longer justified. Unfortunately, for high incident primary proton energies, secondary protons dominate the distribution of nuclear-reaction products.

100, and 200 MeV, respectively. The differences are informative, but our approximations cannot be considered particularly reliable. For soft spectra, the small amount of neutron energy distributed exponentially becomes prominent at large depths, but at a level many orders of magnitude lower in dose. For hard spectra, the doses calculated with our approximations for nuclear-interaction effects cam become larger than those without, which does not conform with the results in reference [12] and implies a failure of our simple approximations (see footnote 3 and Fig. 5).

Because of the uncertainties associated with the treatment of nuclear-interaction effects, and because the dose ratios used to convert to dose in detector materials other than aluminum apply strictly only to the primary beam without nuclear attenuation, it is recommended that reliance not be placed on the use of these approximations, but used perhaps to gauge possible effects that might require more accurate follow-up. It should also be kept in mind that only the physical absorbed dose has been addressed here; high-LET effects associated with the heavier secondaries are beyond the scope of this work.

**Comparisons of Results**

It is useful to compare results from the old and new SHIELDOSE codes. Because the volume of monoenergetic data is so large, it is difficult to completely describe the differences between the new and old work. As expected for the direct electron ionization and for the proton depth-dose distributions, the differences are not large (less than a few to perhaps as much 10 percent) for the same incident energy. Differences in the bremsstrahlung depth-dose distributions can be larger, depending on incident energy and depth, due mainly to the use of new bremsstrahlung production spectra.

To facilitate some comparisons, test calculations have been done for simple exponential spectra of electrons and protons incident on semi-infinite aluminum slabs, assuming the detector material to also be Al. Differences between results from SHIELDOSE and from SHIELDOSE-2 are due not only to differences in the basic monoenergetic data but also due to differences in the coverage of the databases and the numerical techniques used in the interpolation and integration over input spectra. To establish some comparability, spectra were integrated from 1 MeV up to 10 GeV for protons and from 10 keV up to 20 MeV for electrons and bremsstrahlung, relying on the automatic (but not reliable) scaling of depth-dose distributions outside the covered energy ranges of the old SHIELDOSE database.

For incident proton spectra with e-folding energies from 5 to 200 MeV, and considering only cases without nuclear attenuation, small differences (less than 3%) are found in the dose at depths less than about 1 g/cm$^2$, as shown in Fig. 8. At larger depths, the new doses are larger by from 3% for the harder spectra to as much as 10-20% for the soft spectra. For incident electron spectra with e-folding energies from 0.1 to 5 MeV, Fig. 9 indicates that the differences are less than about 5% for depths out to 1-3 g/cm$^2$. At larger depths out to 30 g/cm$^2$, the differences in the bremsstrahlung tail of the dose distribution becomes evident, with the new results smaller by from 0-10% for the hardest spectrum to about 40-60% for the softest spectrum.

Some comparisons among SHIELDOSE-2 results highlight the larger list of detector materials included. Figure 10 shows ratios of proton doses for one detector to those for another, for exponential spectra incident on aluminum slabs: Fig. 10a for Si/LiF; 10b for GaAs/Al; 10c for GaAs/Si; 10d for GaAs/$CaF_2$. Similar results, but for the electron and bremsstrahlung dose, are given in Fig. 11: Fig. 11a for Si/$CaF_2$; 11b for GaAs/Al; 11c for GaAs/Si; 11d for GaAs/$CaF_2$. The rather large dose ratios that can be predicted in the bremsstrahlung tail, particularly for detector materials of largely differing effective atomic number, should serve as a reminder about the interpretation of detector size and configuration, as mentioned earlier.

## Conclusions

Improvements in cross-section information and numerical methods have been incorporated into the SHIELDOSE database, and the coverage has been extended. Differences in the resultant dose estimates have been explored, but the significance of possible changes has to be determined for the depths and spectra pertinent to a particular problem.

Information on running the new SHIELDOSE-2 code can be found in Appendix A. Also included is a companion code DOSCON which can be used to approximately convert the output of SHIELDOSE-2 to obtain the dose for additional geometries: at off-center points in a solid sphere or at the inner surface of spherical shells. Appendix B contains a FORTRAN listing of SHIELDOSE-2, version 2.10, and Appendix C gives the present version of DOSCON. The software for SHIELDOSE-2 should be considered as open, and suggestions are solicited for further development toward enhanced functionality and better flexibility, portability, and ease of use.

## References

[1] S.M. Seltzer, "Electron, Electron-Bremsstrahlung and Proton Depth-Dose Data for Space-Shielding Applications," IEEE Trans. Nucl. Sci. **NS-26**, 4896 (1979).

[2] S.M. Seltzer, "SHIELDOSE: A Computer Code for Space-Shielding Radiation Dose Calculations," National Bureau of Standards Technical Note 1116 (1980).

[3] S.M. Seltzer, "Conversion of Depth-Dose Distributions from Slab to Spherical Geometries for Space-Shielding Applications," IEEE Trans. Nucl. Sci. **NS-33**, 1292 (1986).

[4] S.M. Seltzer, "Electron-Photon Monte Carlo Calculations: The ETRAN Code," Appl. Radiat. Isot. **42**, 917 (1991).

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

[6] M.J. Berger, M. Inokuti, H.H. Andersen, H.Bichsel, D. Powers, S.M. Seltzer, D. Thwaites, and D.E. Watt, "Stopping Powers for Protons and Alpha Particles," Report 49 of the International Commission on Radiation Units and Measurements (1993).

[7] M.J.D. Powell, "Curve Fitting by Cubic Splines," Atomic Energy Research Establishment (Harwell) Report TP 307 (1967).

[8] L.V. Spencer and F.H. Attix, "A Theory of Cavity Ionization," Radiat. Res. **3**, 239 (1955).

[9] M.J. Berger, M. Inokuti, H.H. Andersen, H.Bichsel, J.A. Dennis, D. Powers, S.M. Seltzer, and J.E. Turner, "Stopping Powers for Electrons and Positrons," Report 37 of the International Commission on Radiation Units and Measurements (1984).

[10] D.R. White, J. Booz, R.V. Griffith, J.J. Spokas, and I.J. Wilson, "Tissue Substitutes in Radiation Dosimetry and Measurement," Report 44 of the International Commission on Radiation Units and Measurements (1989).

[11] S.M. Seltzer, "Calculation of Photon Mass Energy-Transfer and Mass Energy-Absorption Coefficients," Rad. Res. **136**, 147 (1993).

[12] R.T. Santoro, R.G. Alsmiller, Jr., and J. Barish, "The Validity of Using Only Primary Protons in Van Allen Belt and Solar-Flare Proton Shielding Studies," Nucl. Sci. and Engr. **49**, 395 (1972).

[13] W. Bauhoff, "Tables of Reaction and Total Cross sections for Proton-Nucleus Scattering Below 1 GeV," At. Data and Nucl. Data Tables **35**, 429 (1986).

[14] R.G. Alsmiller and J. Barish, "NCDATA – Nuclear Collision Data for Nucleon-Nucleus Collisions in the Energy Range 25 to 400 MeV," Oak Ridge National Laboratory Report ORNL-4220 (1968).

[15] J.F. Janni, "Proton Range-Energy Tables, 1 keV - 10 GeV," At. Data and Nucl. Data Tables **27**, 147 (1982).

[16] L.W. Townsend and J.W. Wilson, "Tables of Nuclear Cross Sections for Galactic Cosmic Rays," National Aeronautics and Space Administration Reference Publication 1134 (1985).

[17] J.R. Letaw, R. Silberberg, and C.H. Tsao, "Proton-Nucleus Inelastic Cross Sections: An Empirical Formula for E > 10 MeV," Astrophys. J. Suppl. Ser. **51**, 271 (1983).

[18] S.M. Seltzer, "An Assessment of the Role of Charged Secondaries from Nonelastic Nuclear Interactions by Therapy Proton Beams in Water," National Institute of Standards and Technology Publication NISTIR 5221 (1993).

[19] H.E. Hungerford, "The Nuclear, Physical, and Mechanical Properties of Shielding Materials" in *Reactor Handbook, 2nd Edition, Vol. I. Materials* (Ed. C.R. Tipton, Jr.), Interscience, NY, p. 1027 (1960).

[20] R. Aronson and C.N. Klahr, "Neutron Attenuation," in *Reactor Handbook, 2nd Edition, Vol. III. Materials* (Ed. E.P. Blizard, Jr.), Interscience, NY, p. 63 (1962).

[21] P.N. Stevens, D.K. Trubey, C.W. Garrett, and W.E. Selph, "Radiation Transport," in *Reactor Shielding for Nuclear Engineers* (Ed. N.M. Schaeffer), U.S. Atomic Energy Commission, p. 119 (1973).

[22] D.L. Broder and S.G. Tsypin, "Neutron Attenuation. Attenuation in Non-Hydrogenous Media," in *Engineering Compendium on Radiation Shielding, Vol. I. Shielding Fundamentals and Methods* (Ed. R.G. Jaeger), Springer-Verlag, NY, p. 322 (1968).

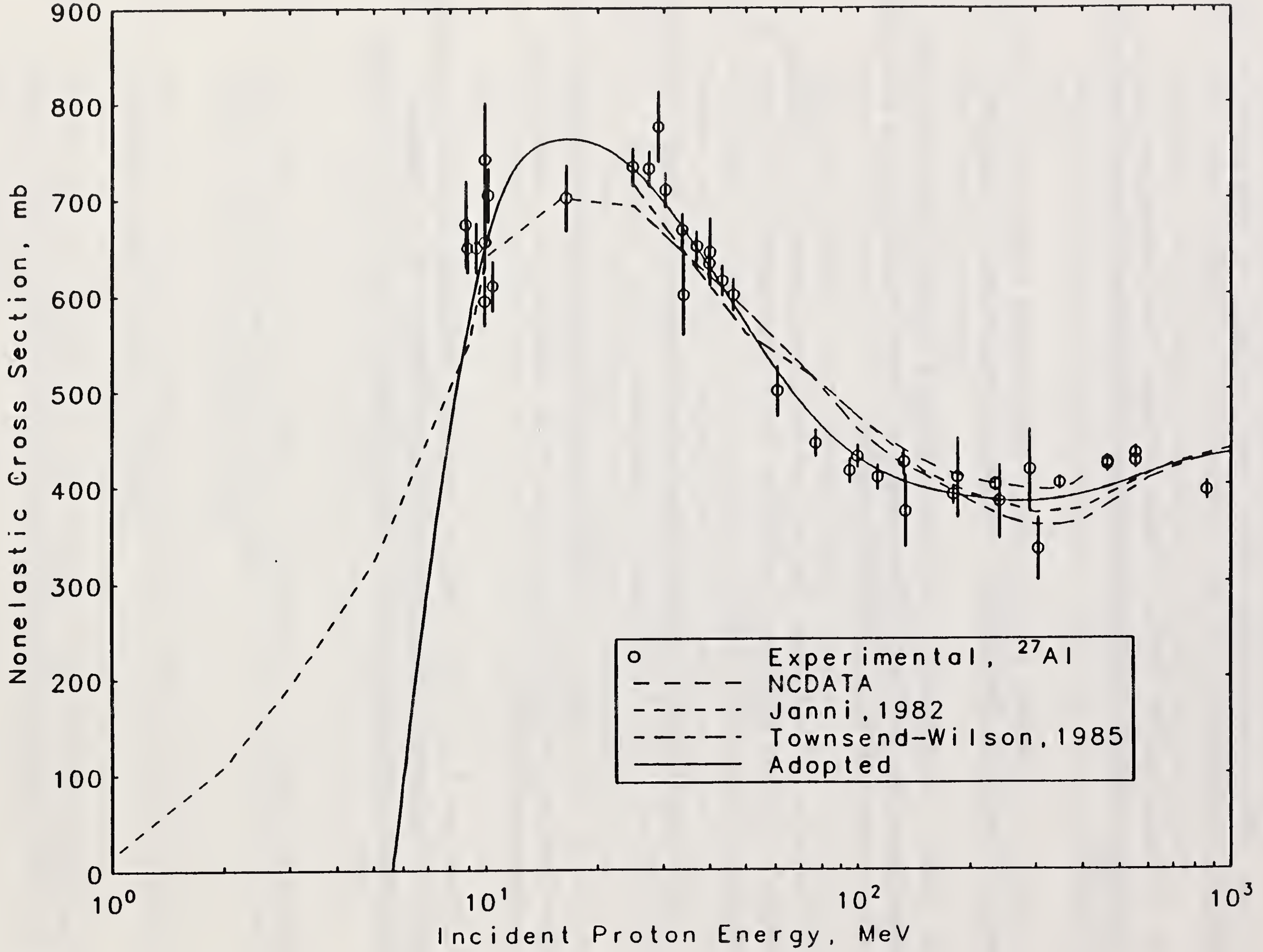

Fig. 1. Total nonelastic nuclear interaction cross section for p+$^{27}$Al. The measured points are from reference [12], the long-dashed curve from results in [13], the short-dash curve from [14], the dot-dash curve from [15], and the solid curve as adopted here.

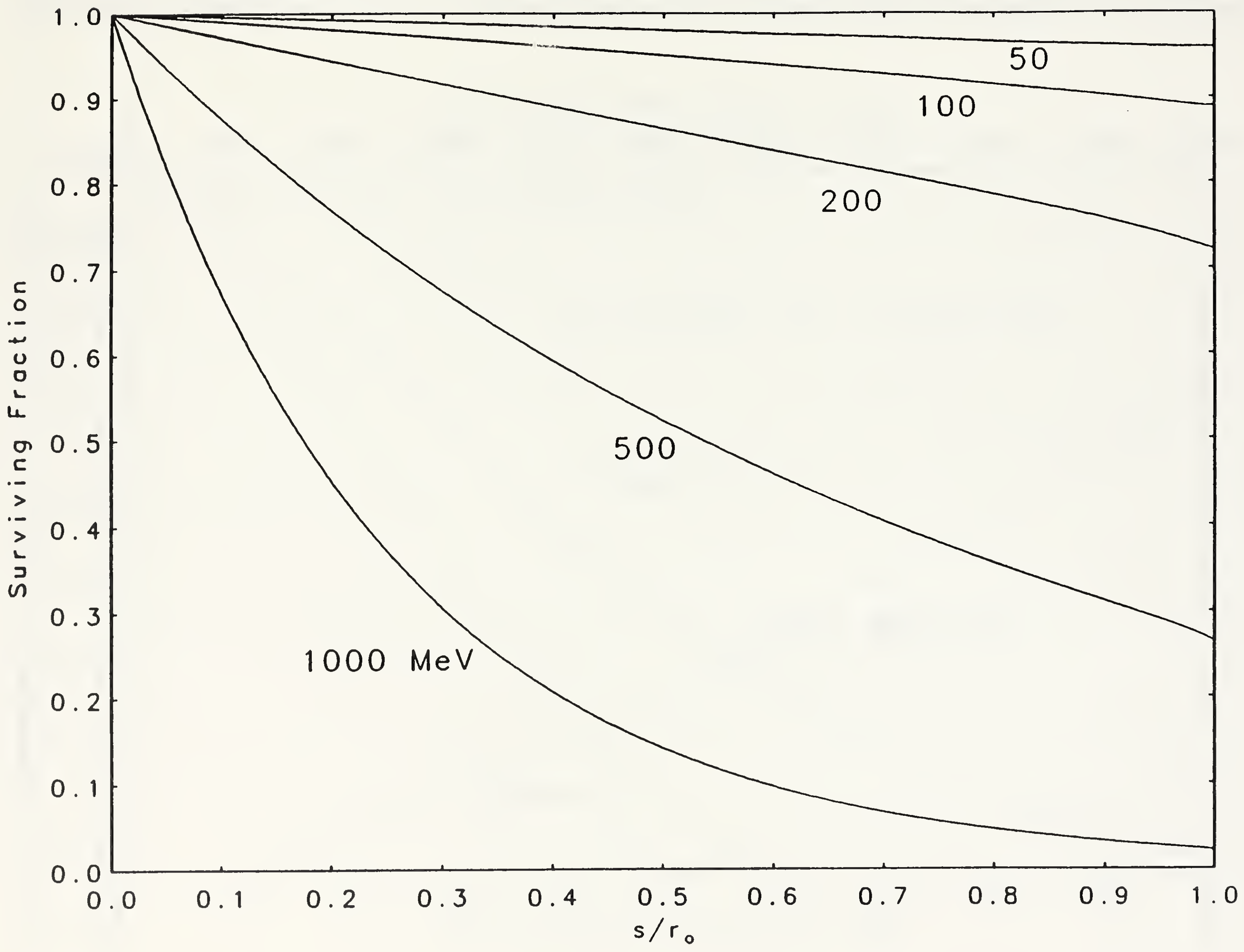

Fig. 2. The attenuation of protons along their direction of travel, due to nonelastic nuclear interactions, evaluated in the continuous-slowing-down approximation (csda). The surviving fraction is plotted as a function of distance s in units of the proton csda or mean range $r_o$.

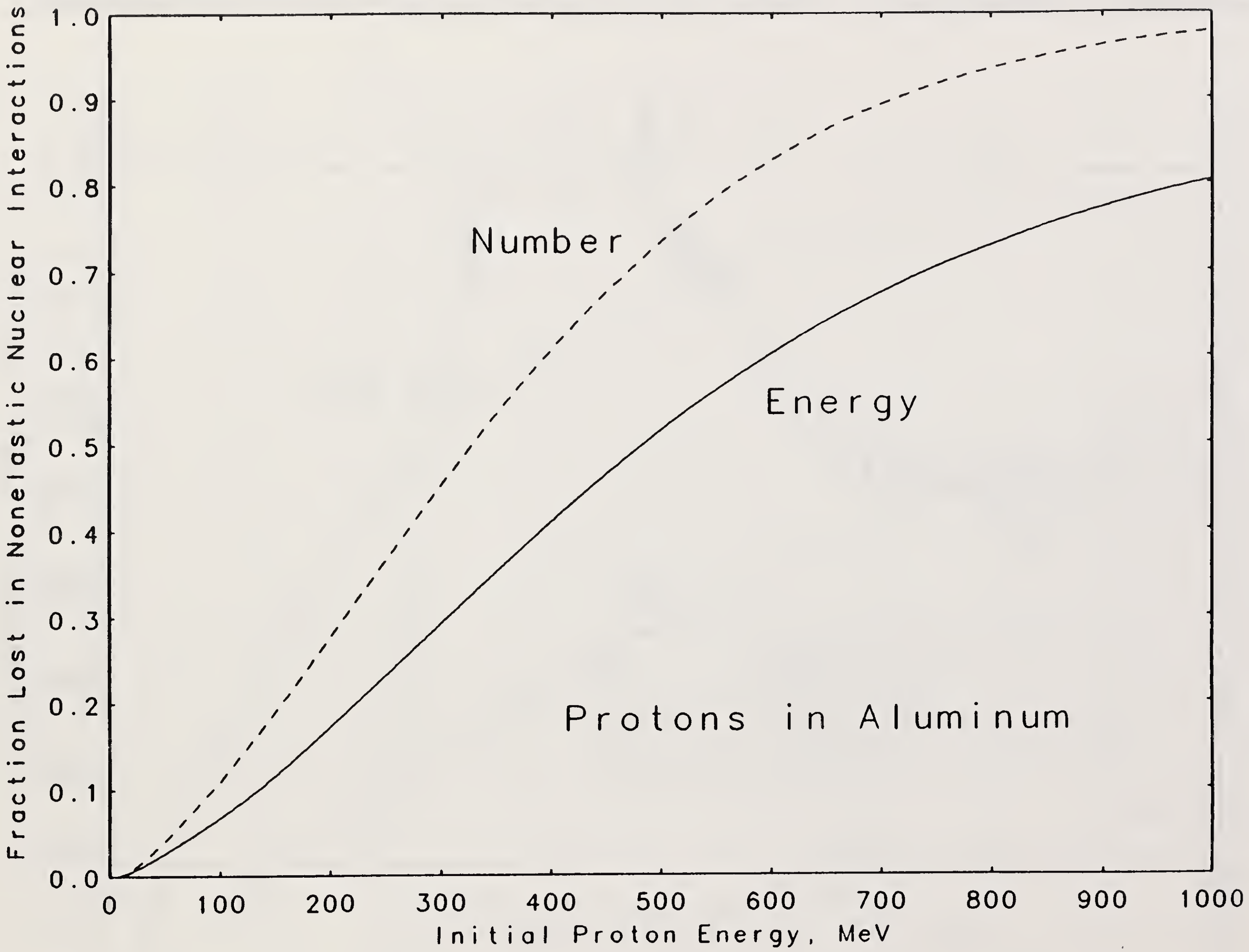

Fig. 3a. The fraction of energy and of number of primary protons lost in nonelastic nuclear interactions, in the course of slowing down in aluminum from their initial energy to rest. The results are from csda calculations. Initial energies up to 1 GeV.

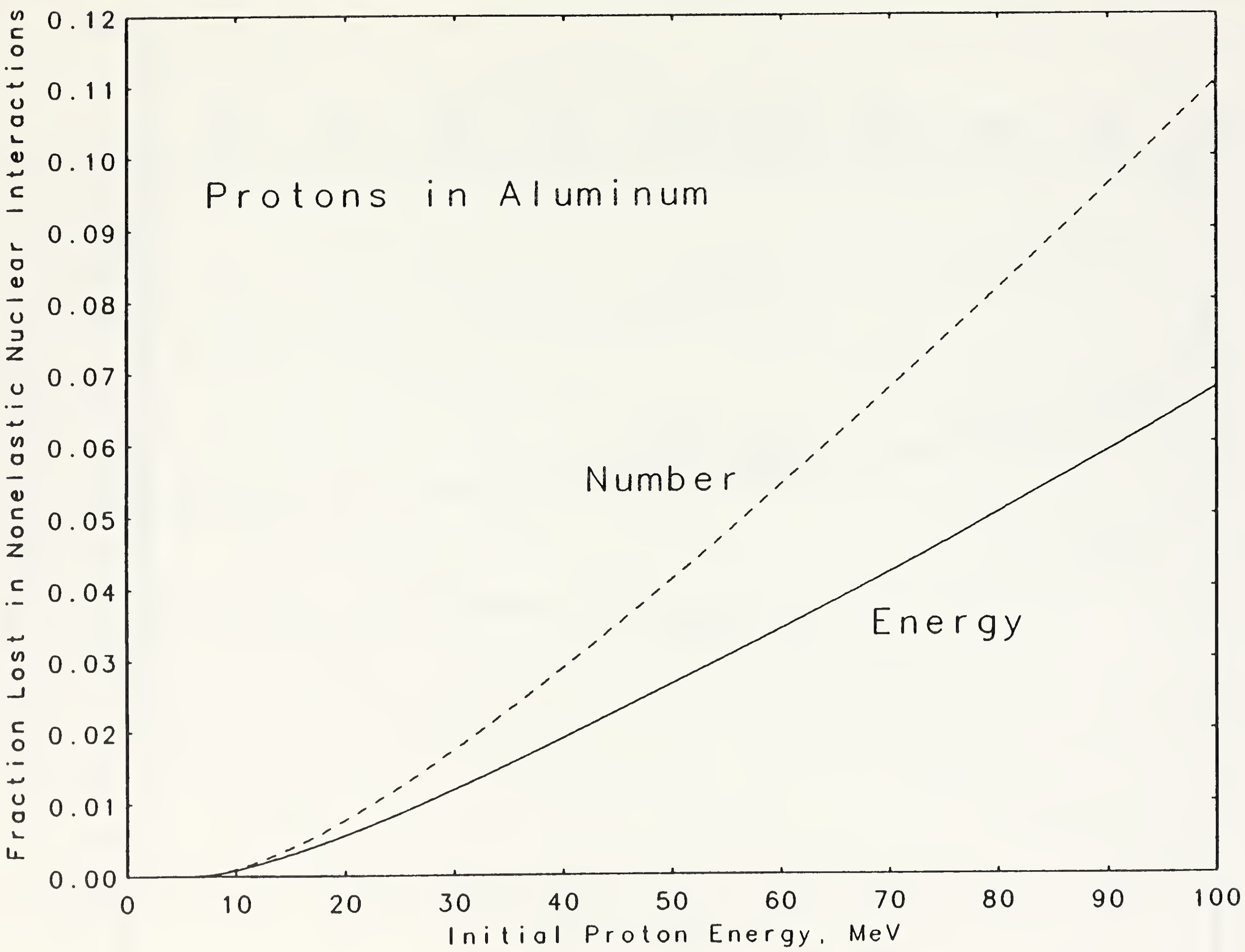

Fig. 3b. The fraction of energy and of number of primary protons lost in nonelastic nuclear interactions, in the course of slowing down in aluminum from their initial energy to rest. The results are from csda calculations. Initial energies up to 100 MeV.

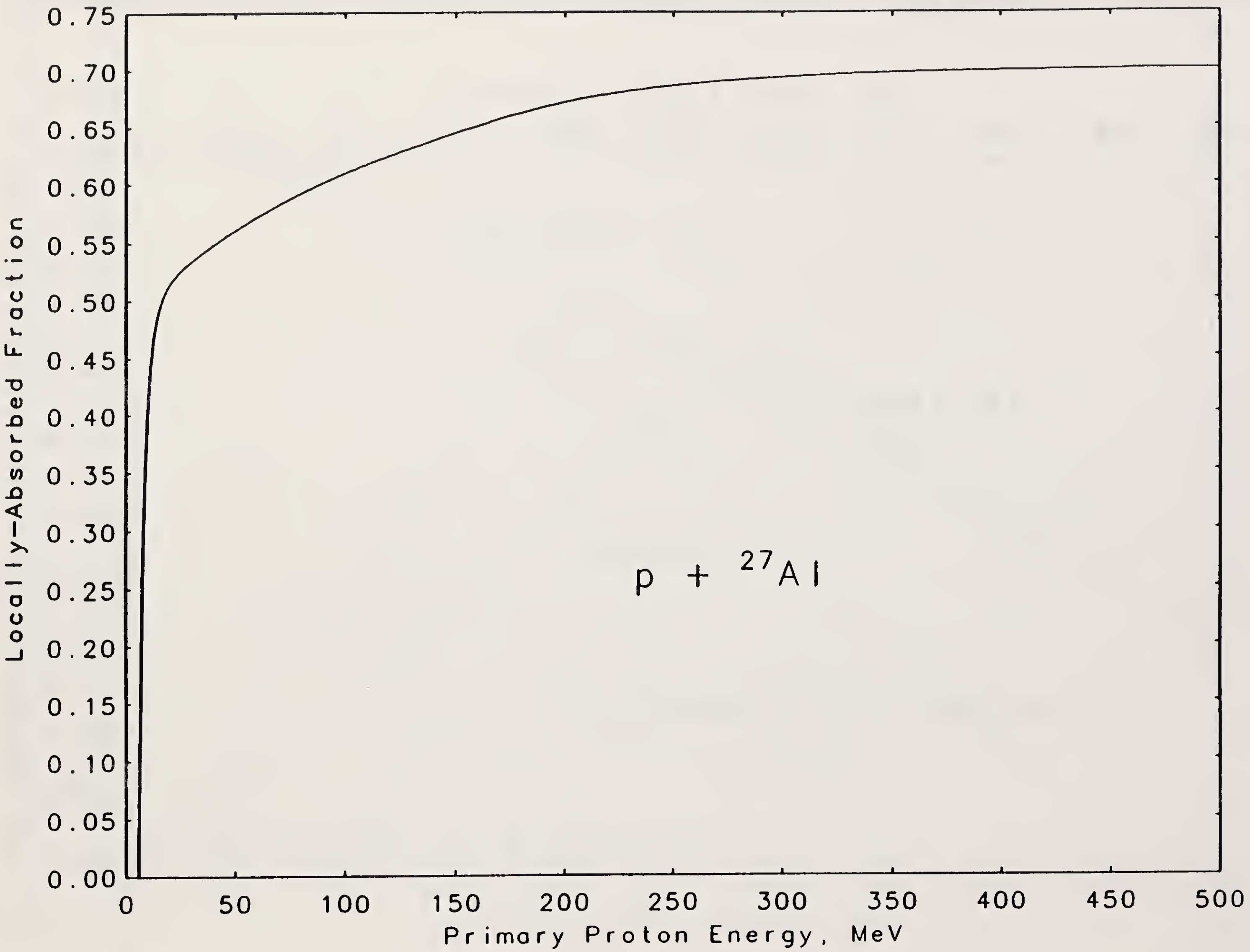

Fig. 4. Assumed fraction of incident primary proton energy converted to kinetic energy of secondary charged particles in nonelastic interactions p+$^{27}$Al. This fraction is assumed to be absorbed locally.

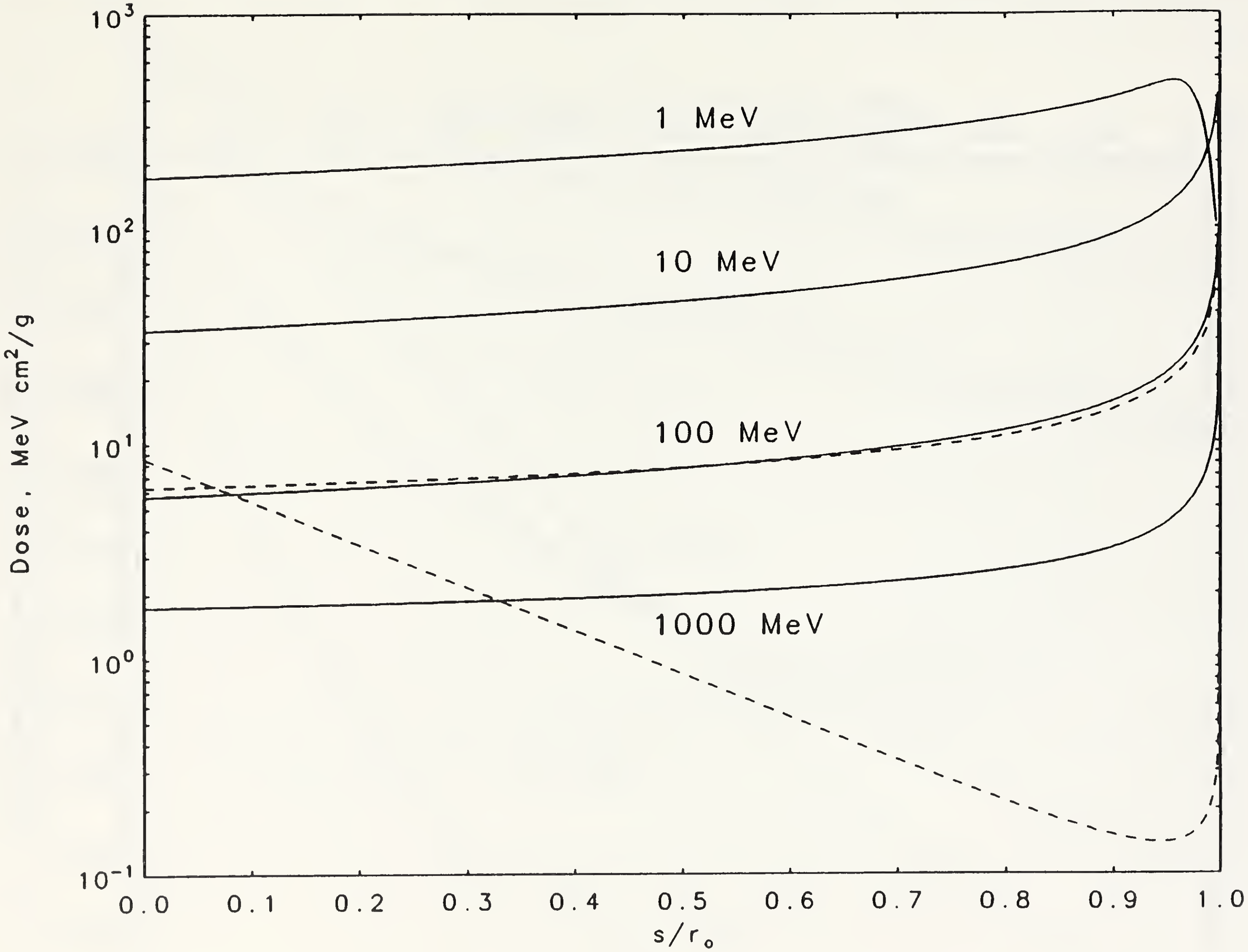

Fig. 5. Proton absorbed dose as a function of distance traveled. The solid curves are from csda calculations neglecting nuclear interactions; the dashed curves are from csda calculations based on the use of the nonelastic nuclear interaction cross sections given in Fig. 1 and the locally-absorbed fractions of nuclear-reaction energy loss given in Fig. 4.

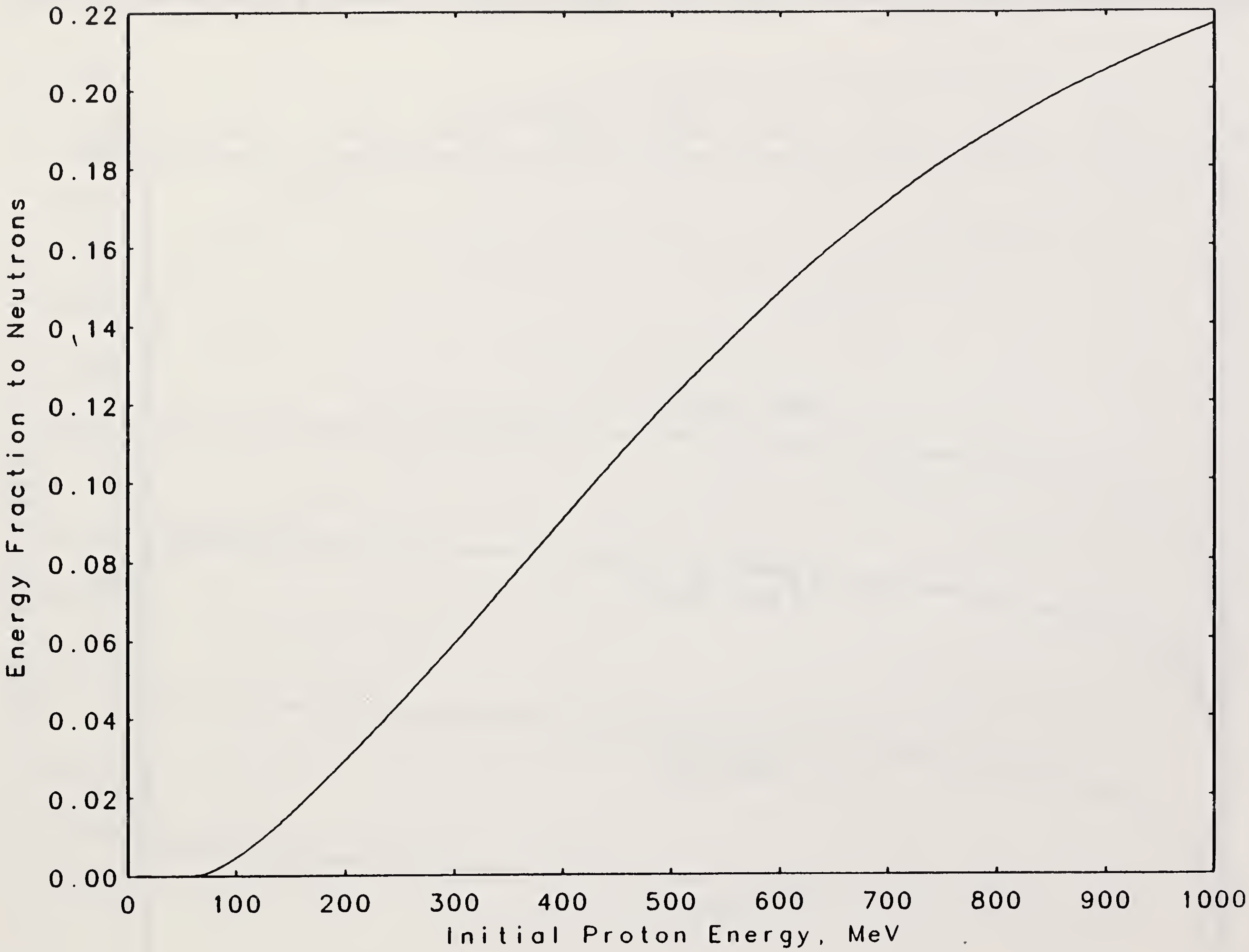

Fig. 6. Assumed fraction of energy of primary protons converted to neutron energy in nonelastic nuclear interactions, in the course of slowing down in aluminum from their initial energy to rest. The results are from csda calculations.

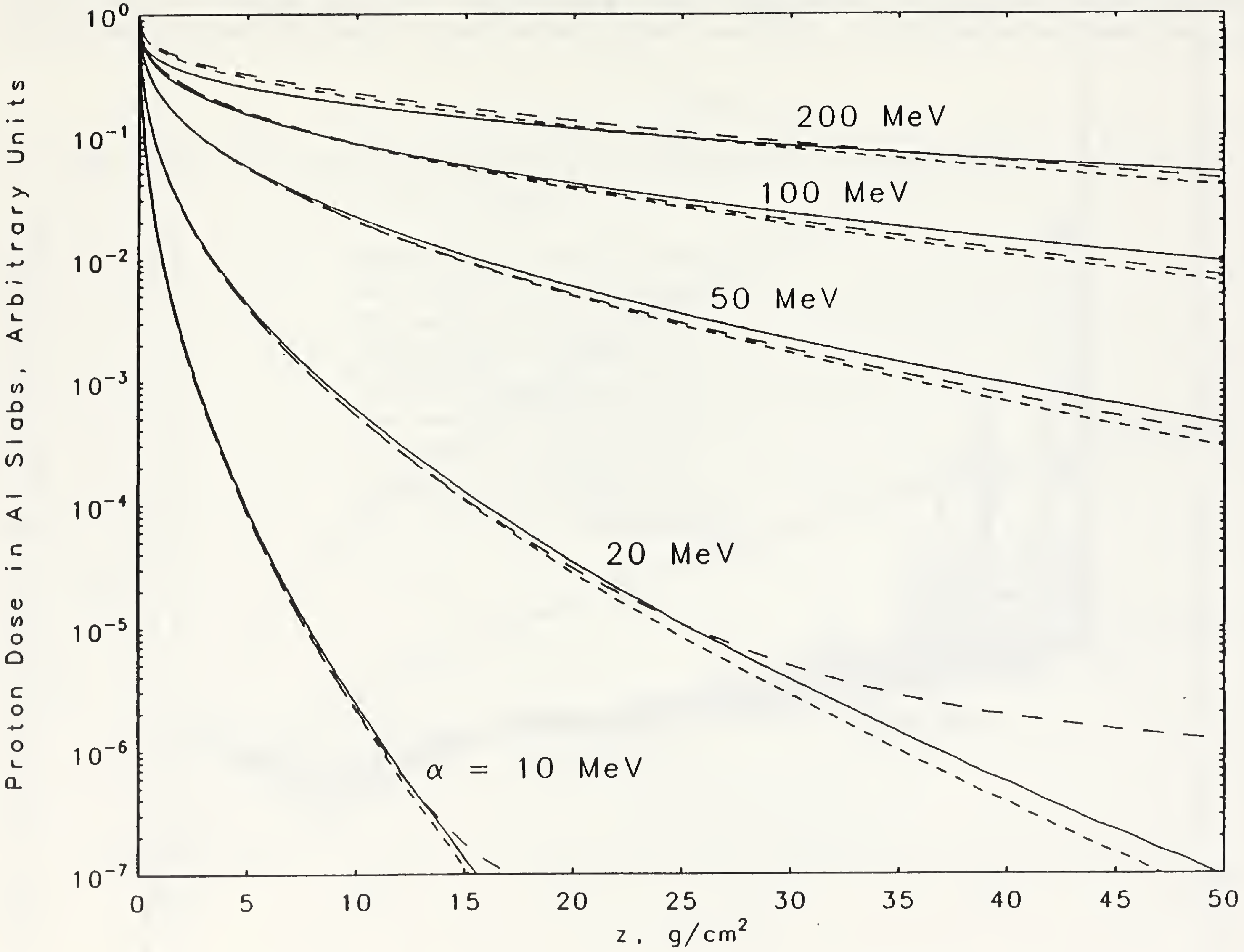

Fig. 7a. Model study of proton depth-dose in aluminum semi-infinite plane slabs. The solid curves are from results obtained with the neglect of nuclear interactions; the short-dash curves include nuclear attenuation and the local absorption of secondary charged-particle energy; and the long-dash curves include, additionally, an assumed simple exponential distribution of deposited secondary neutron energy. Depths to 50 g/cm$^2$, for $\alpha$ = 10, 20, 50, 100, and 200 MeV.

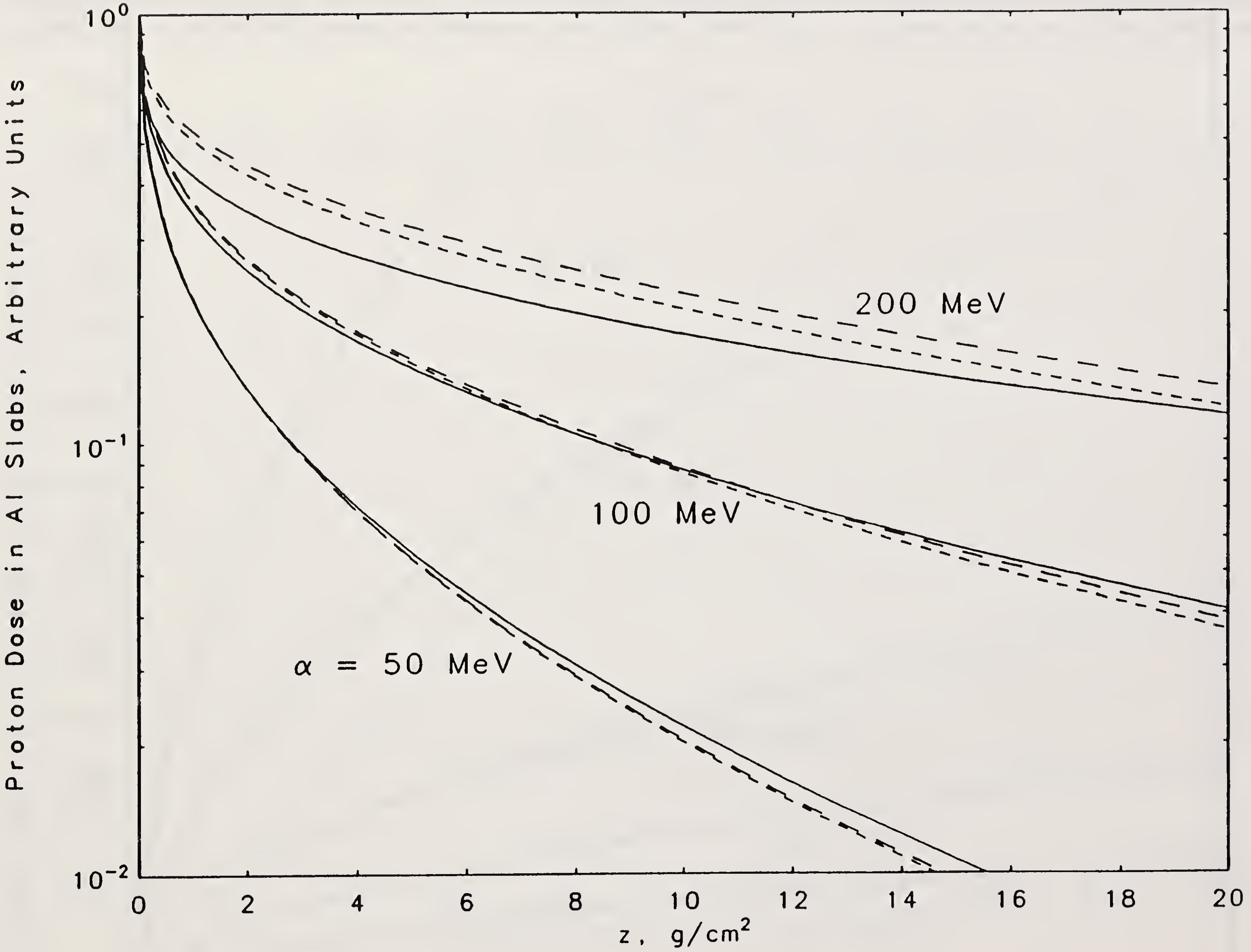

Fig. 7b. Model study of proton depth-dose in aluminum semi-infinite plane slabs. The solid curves are from results obtained with the neglect of nuclear interactions; the short-dash curves include nuclear attenuation and the local absorption of secondary charged-particle energy; and the long-dash curves include, additionally, an assumed simple exponential distribution of deposited secondary neutron energy. An enlargement of the results for depths out to 20 g/cm$^2$.

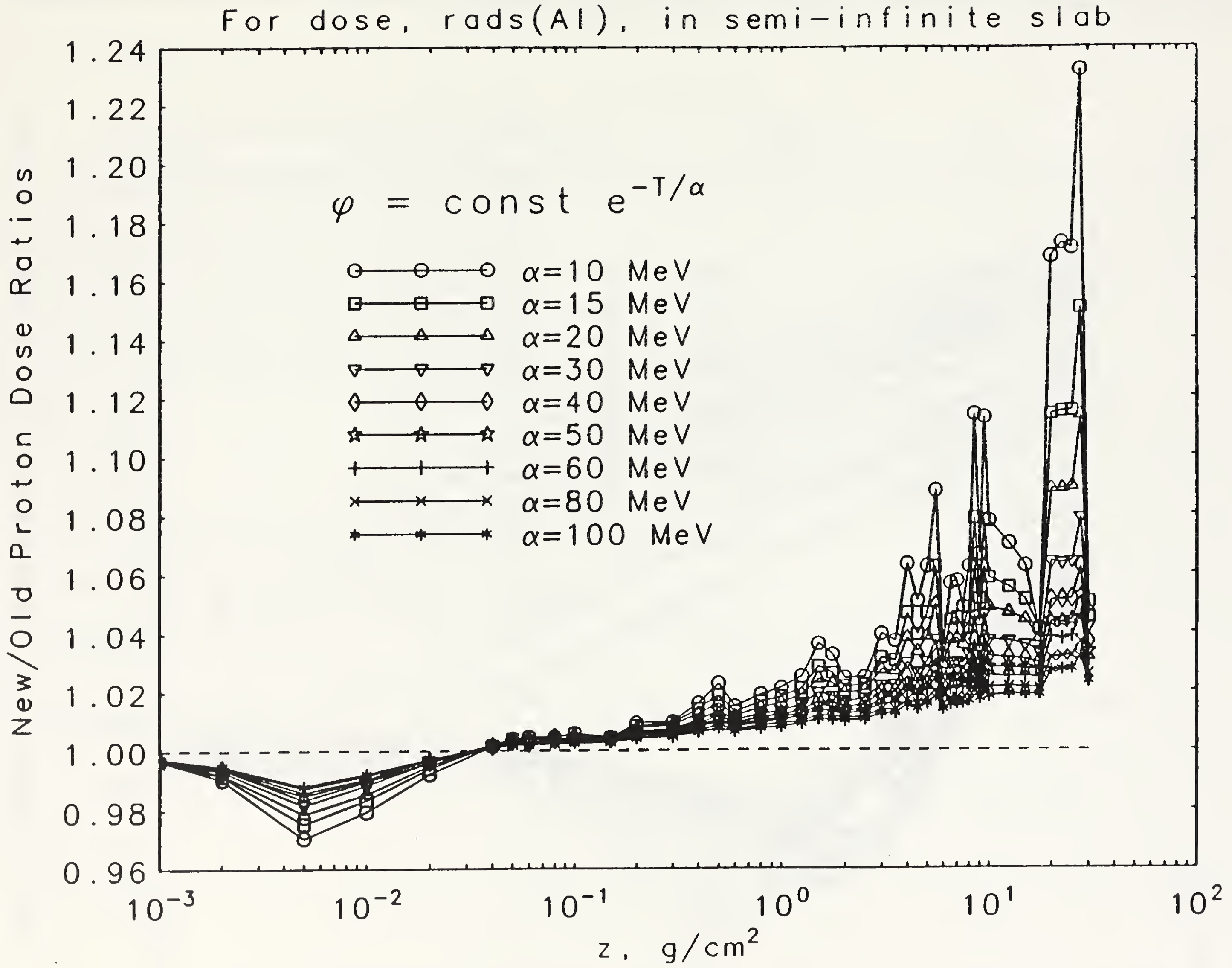

Fig. 8. Comparison of old and new SHIELDOSE results for the dose from protons, as a function of depth z in aluminum semi-infinite slabs. The calculations are for an isotropic fluence of protons with exponential spectra of incident energy T. The irregularities appear to be mainly due to wiggliness in older results.

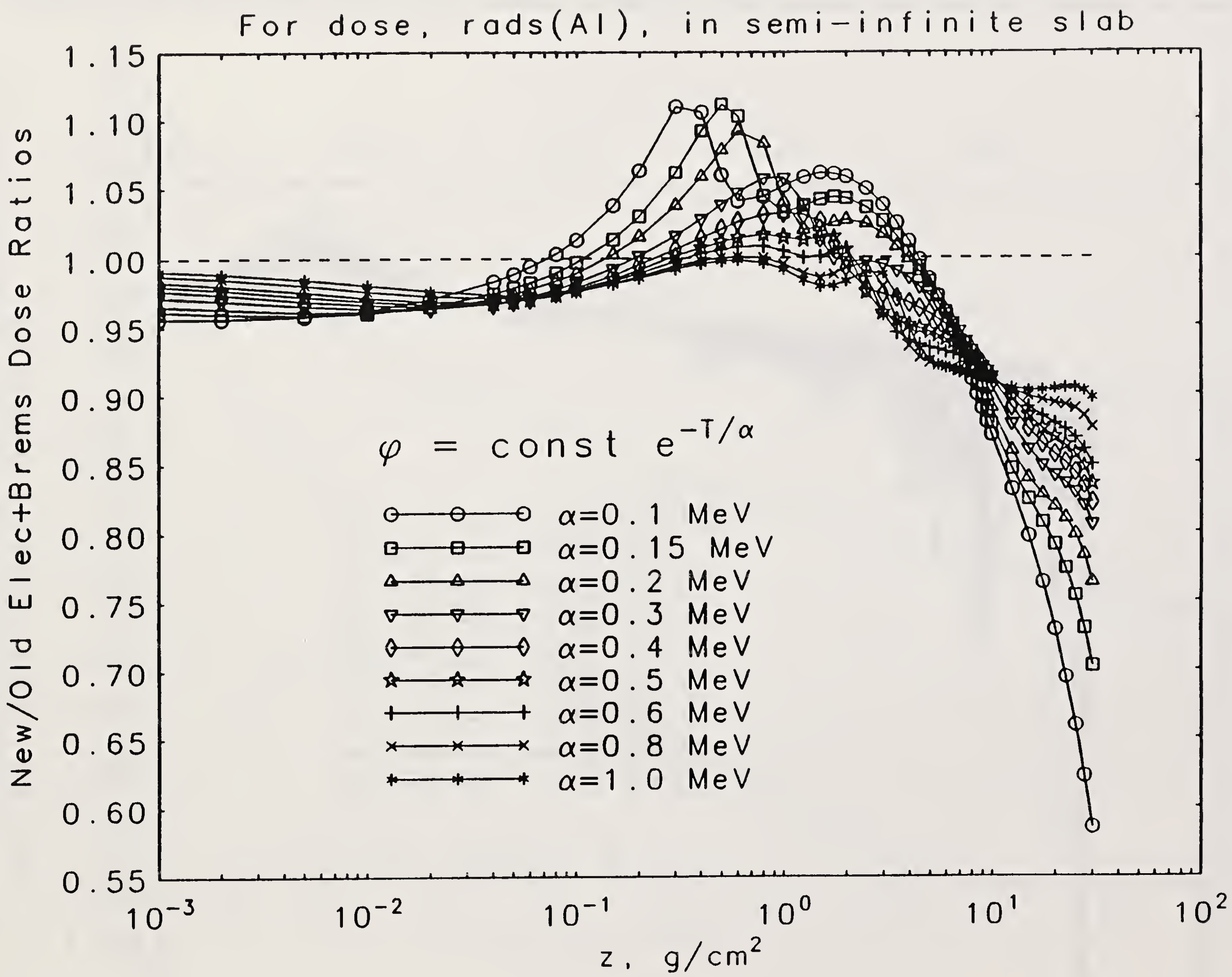

Fig. 9. Comparison of old and new SHIELDOSE results for the dose from electrons and their secondary bremsstrahlung, as a function of depth z in aluminum semi-infinite slabs. The calculations are for an isotropic fluence of electrons with exponential spectra of incident energy T.

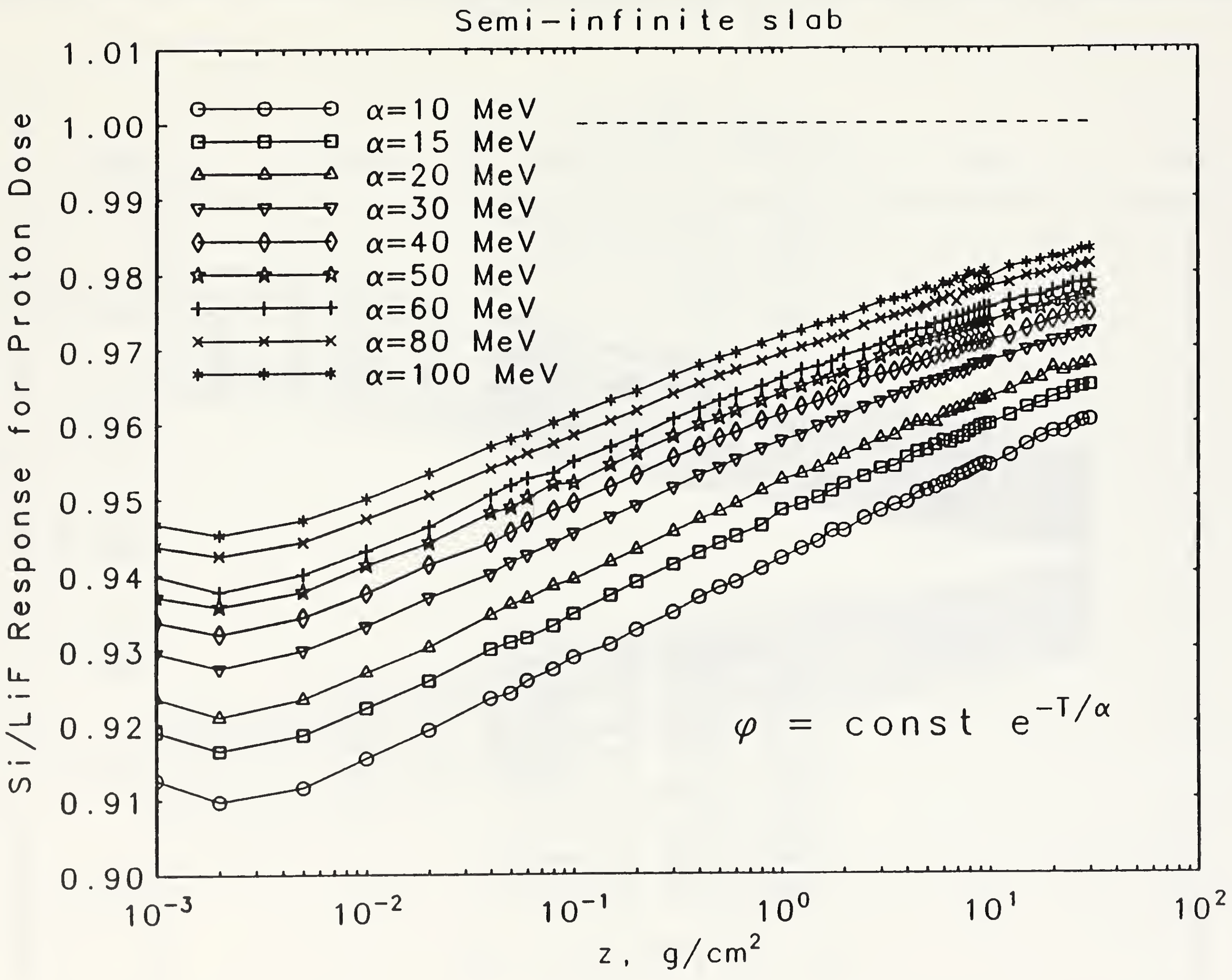

Fig. 10a. Ratio of proton doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. Si/LiF ratios.

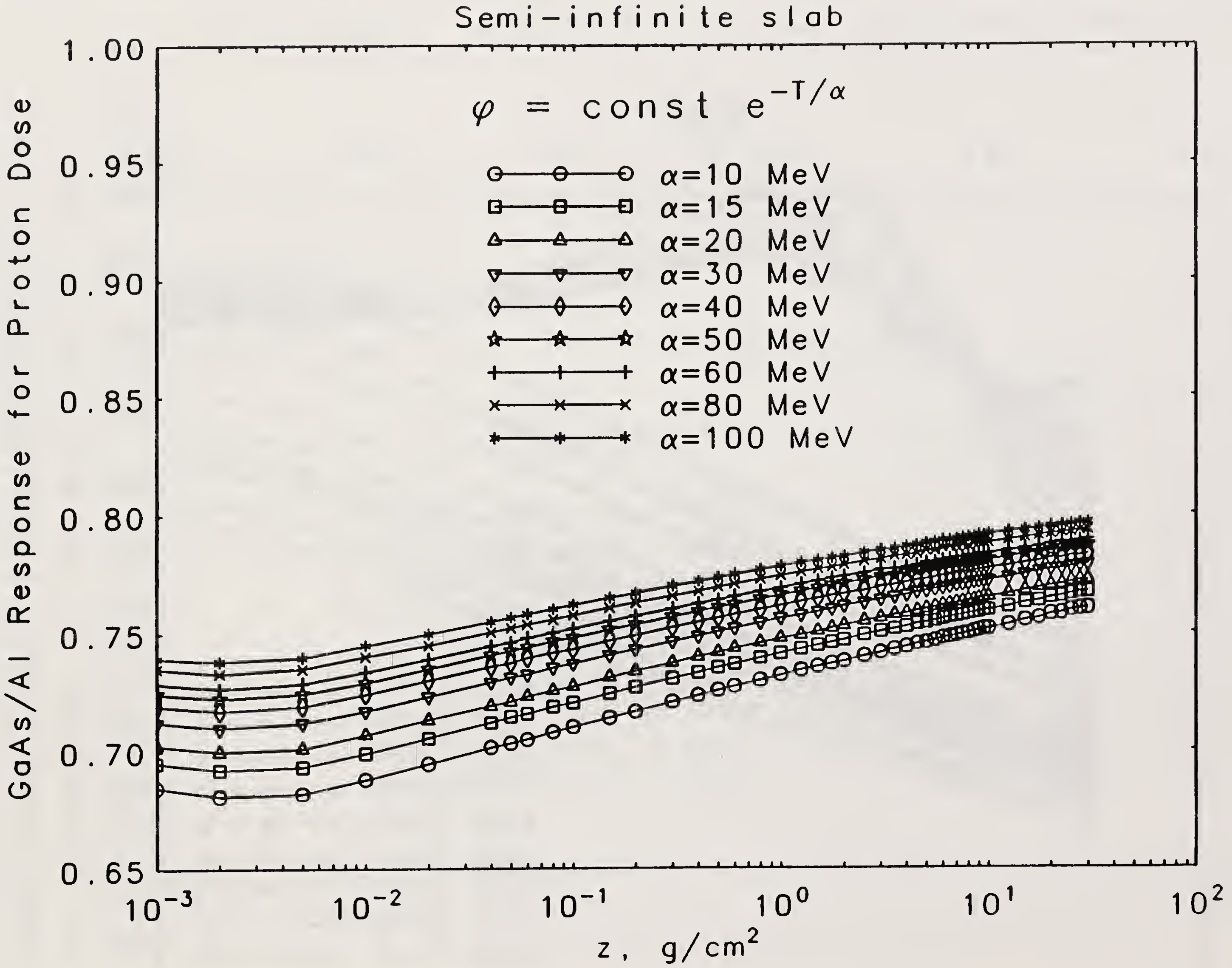

Fig. 10b. Ratio of proton doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. GaAs/Al ratios.

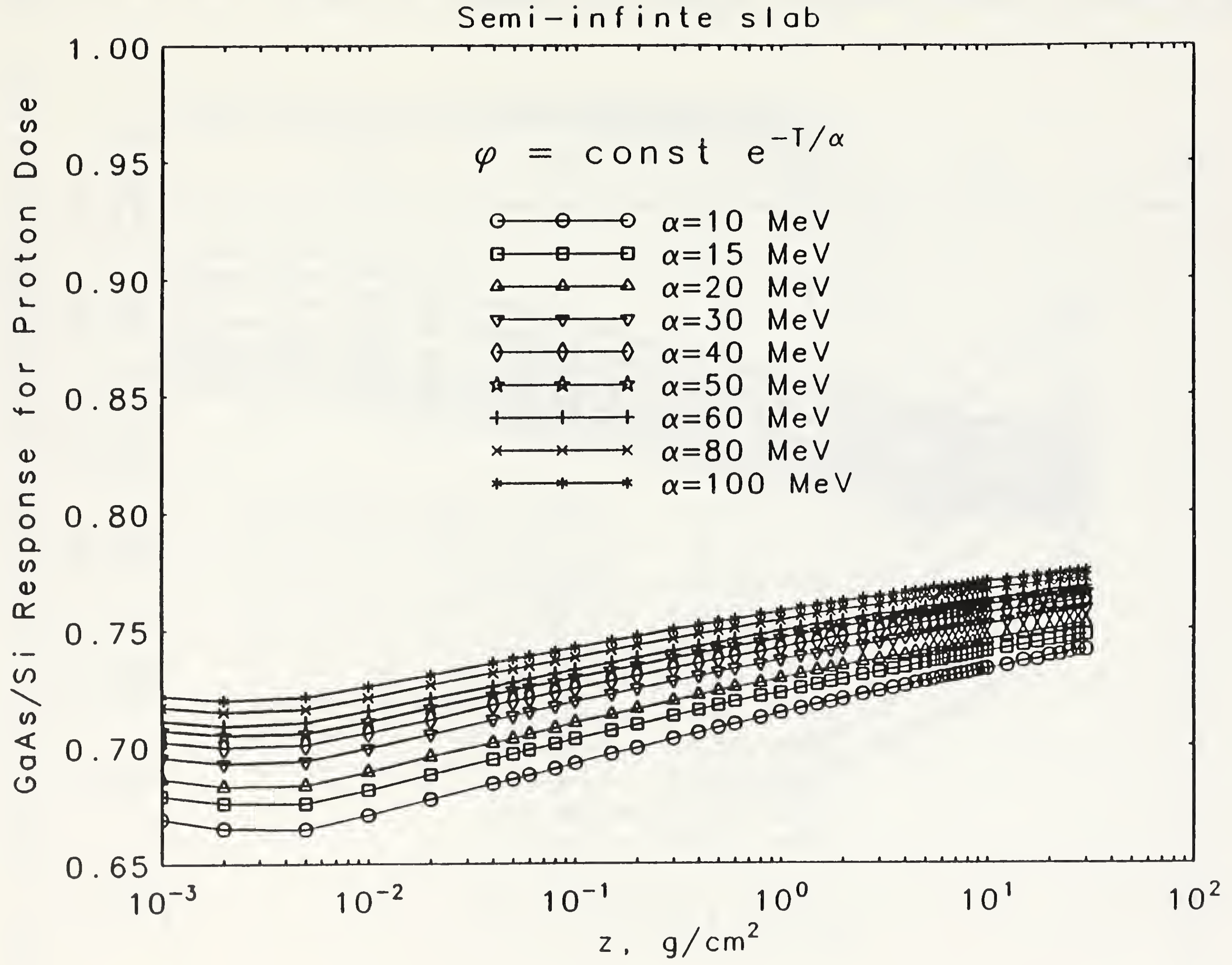

Fig. 10c. Ratio of proton doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. GaAs/Si ratios.

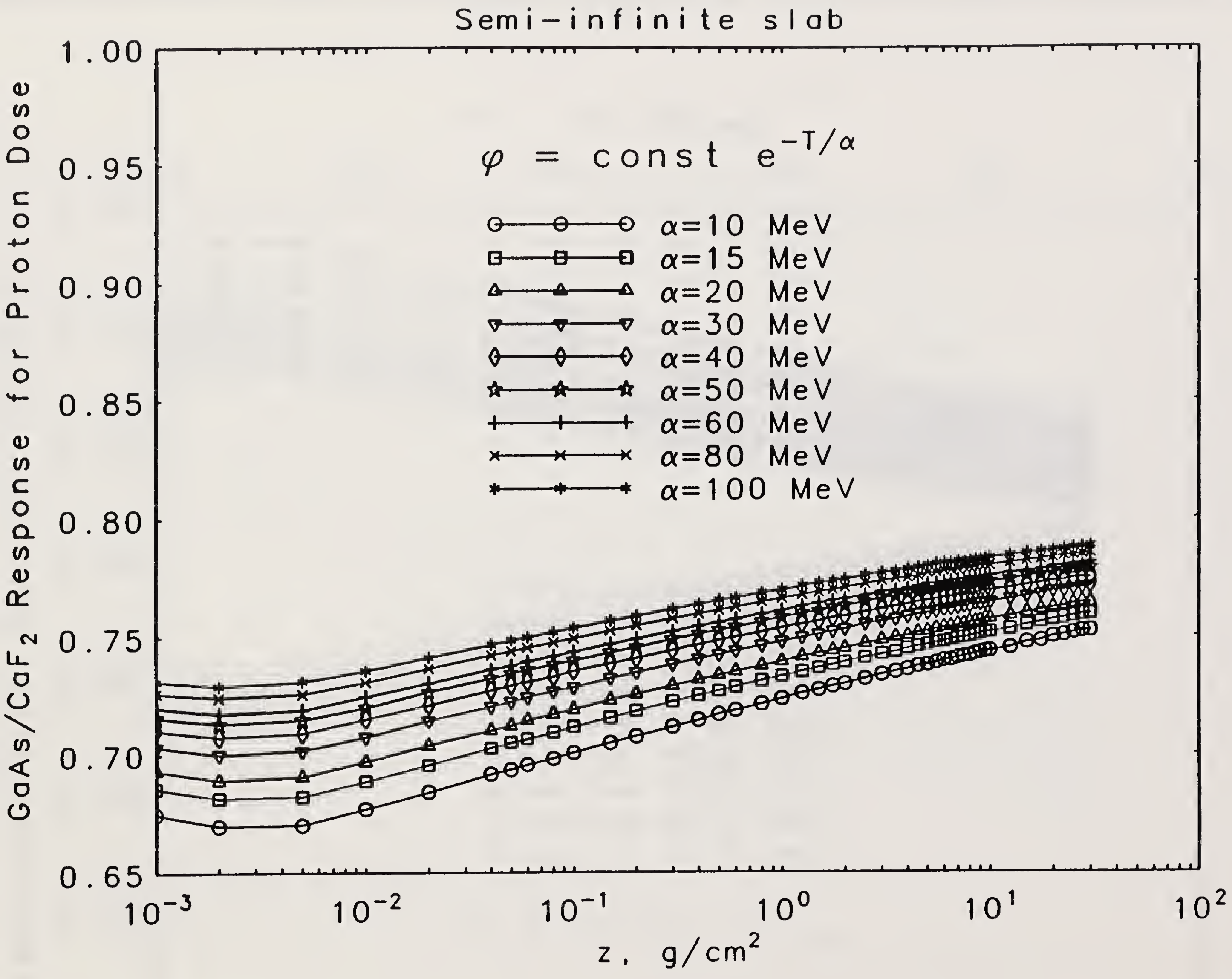

Fig. 10d. Ratio of proton doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. GaAs/CaF$_2$ ratios.

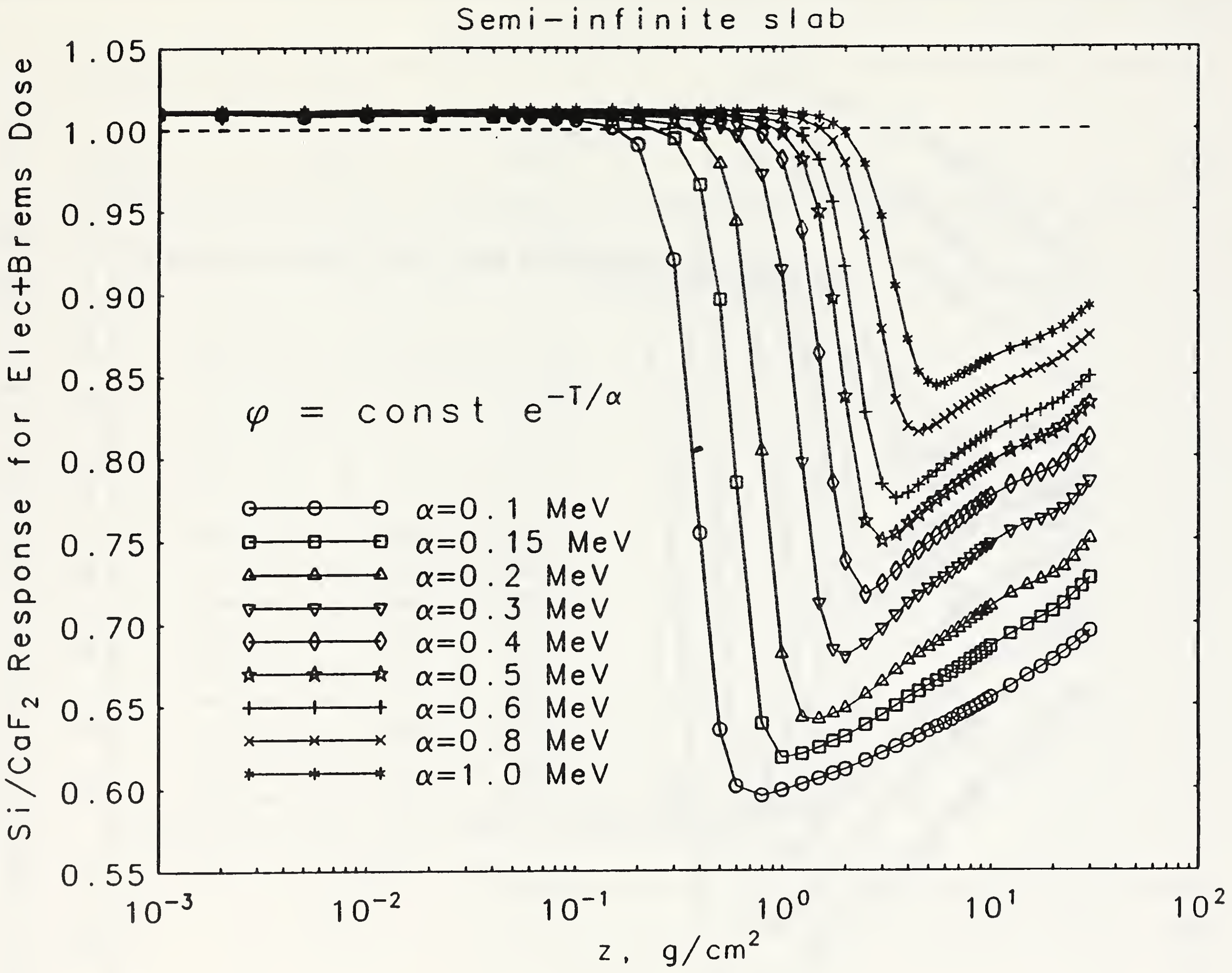

Fig. 11a. Ratio of electron and bremsstrahlung doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. Si/CaF$_2$ ratios.

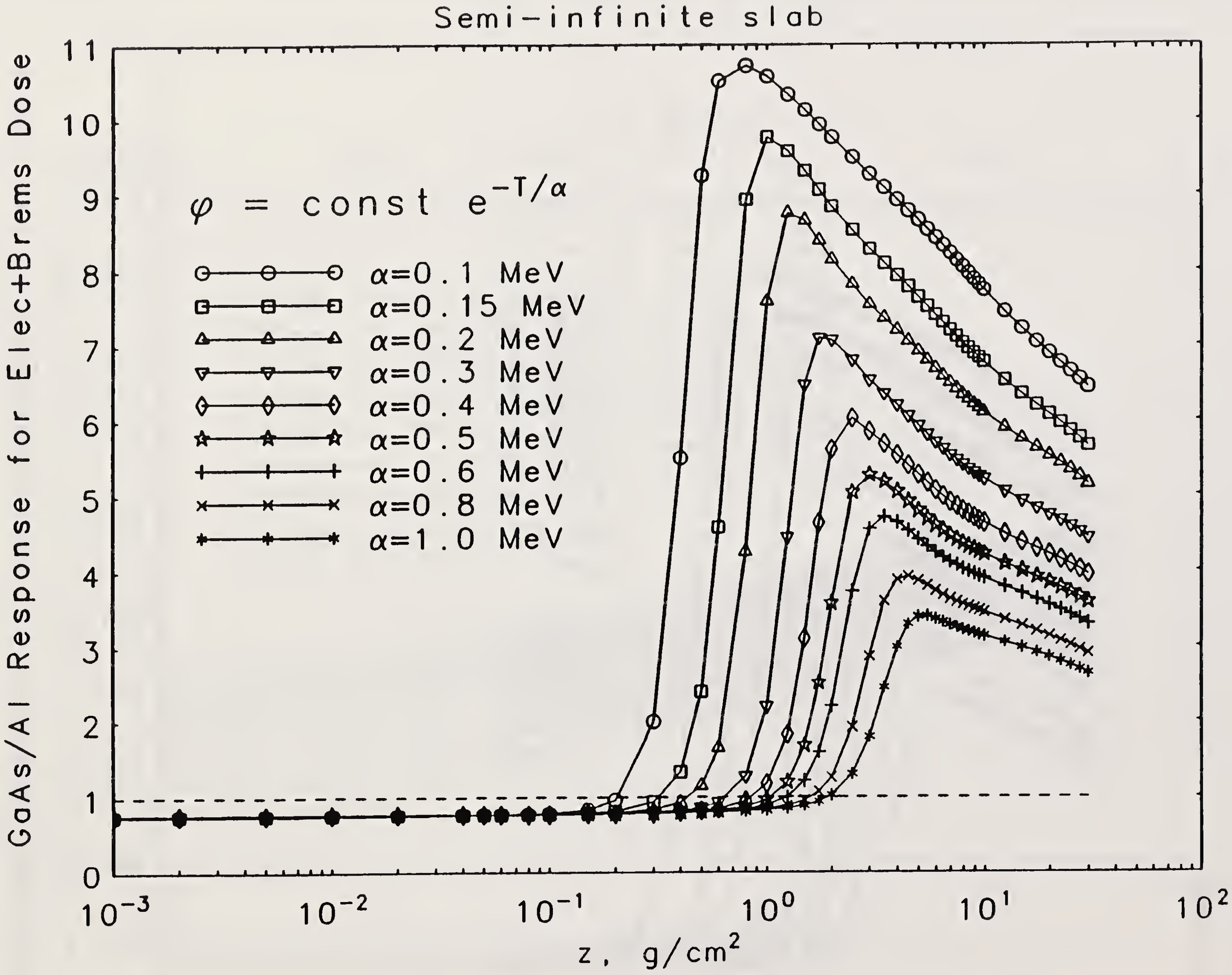

Fig. 11b. Ratio of electron and bremsstrahlung doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. GaAs/Al ratios.

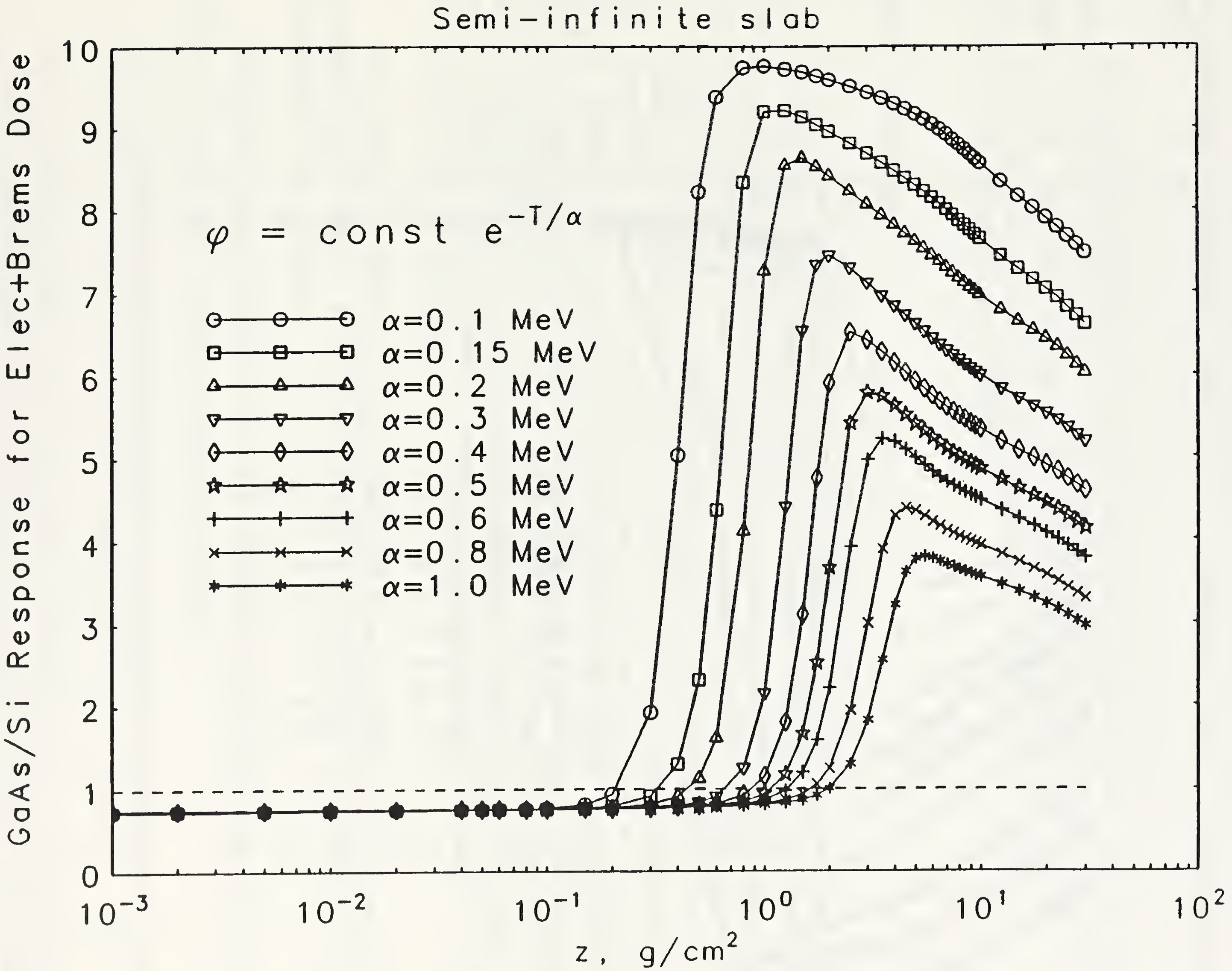

Fig. 11c. Ratio of electron and bremsstrahlung doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. GaAs/Si ratios.

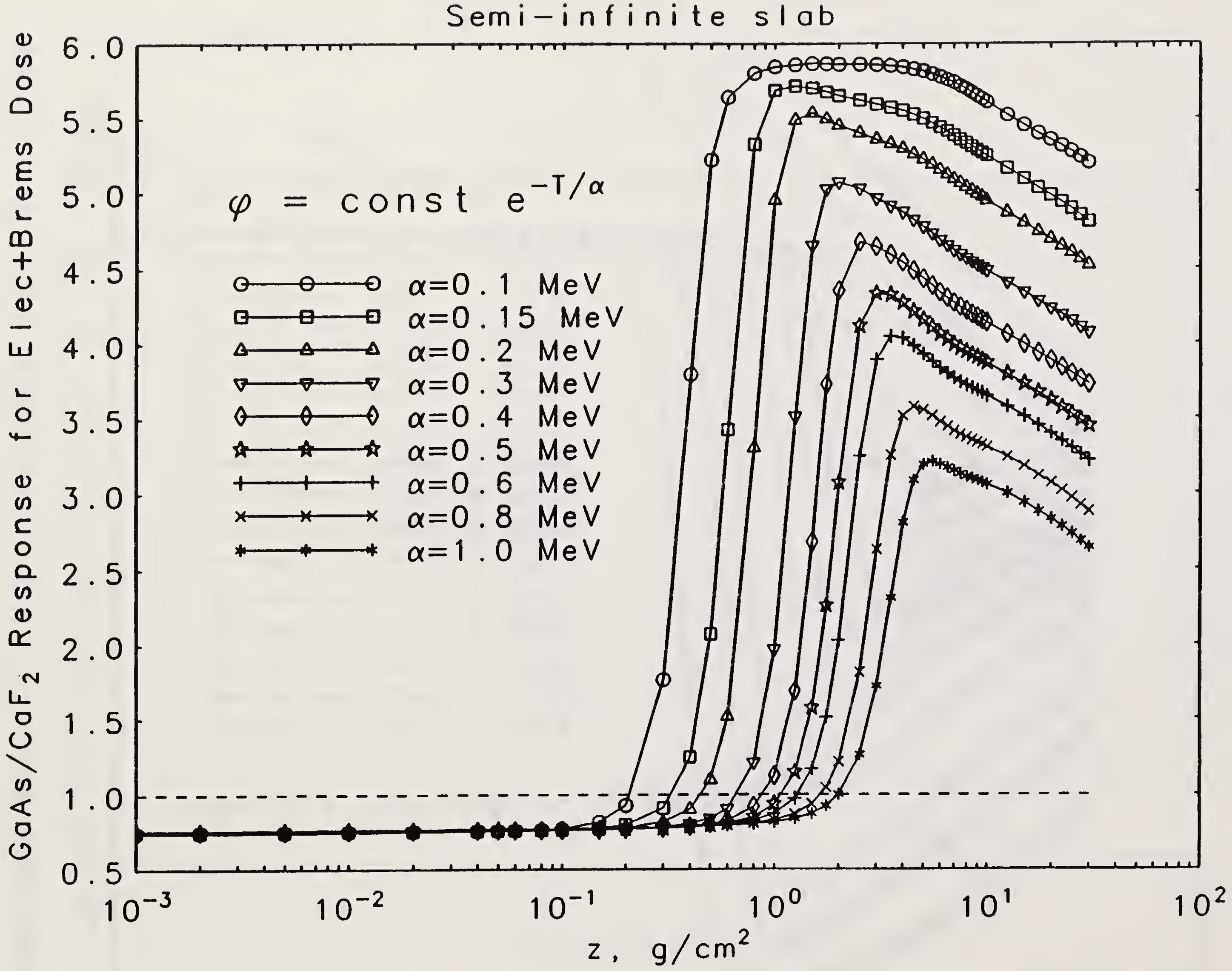

Fig. 11d. Ratio of electron and bremsstrahlung doses calculated for two different detector volumes, as a function of depth z in aluminum semi-infinite slabs. GaAs/CaF$_2$ ratios.

# Appendix A. Comments on Using SHIELDOSE-2

The files supplied include those for SHIELDOSE-2: (a) SD2.FOR, the FORTRAN code; (b) PROTBAS2.DAT, the proton dose database; (c) ELBRBAS2.DAT, the electron and bremsstrahlung dose database; (d) SAMPLE.INP, sample input; and (e) SAMPLE.OUT, sample printout.

Also included is the DOSCON code to convert the SHIELDOSE-2 output to dose as a function of radius in a solid aluminum sphere and to dose at the inner surface of spherical aluminum shells. These files include: (a) DOSCON.FOR, the FORTRAN code; (b) SAMPLE.ARR, the array output from the sample run of SHIELDOSE-2; and (c) SAMPLE.CON, sample output of DOSCON.

*Program SHIELDOSE-2*

The functionality and flow of the present SHIELDOSE-2 code is nearly identical with the original SHIELDOSE code. The FORTRAN code has been compiled and run on a PC. With the dimensions chosen for the distribution file, it is necessary to use a system with a DOS extender to accommodate a 1.7Mb executable code (single precision). Comments in the code will guide you in converting to a double-precision version. A single-precision version that will fit comfortably in the standard lower memory of a PC can be compiled by changing the value of NPTSI from 1001 to 251 in the parameter statements in the main program and in the subroutine SPECTR. This parameter governs the maximum choice for the number of integration points; we have found that a larger number tends to produce somewhat smoother (and more accurate) results, particularly for the proton dose at larger depths. Some further reduction can be done by reducing IMAXI from 71 to 51 in the parameter statements in the main program and in the subroutine SPHERE. This parameter governs the maximum choice for the number of depths at which the doses will be calculated.

The subroutine LOGO may require ANSI.SYS in your CONFIG.SYS to work correctly. The single call to it, near the start of the FORTRAN file, can easily be eliminated to avoid difficulties with your platform.

The input stream is nearly the same, and the user should consult reference [2] for general guidance. There are a few changes to the input parameters:

(1) Two initial records specify the destination of (a) the printout (tables) file, and (b) the file containing calculated dose arrays for possible subsequent analysis.

(2) The remaining input data is list-directed, and need not be in fixed columns but merely separated by appropriate delimiters (e.g., blanks or commas). It is important to know that for list-directed input, FORTRAN treats a blank as a delimiter, so that numbers entered in E format must *not* have a blank between the E and the exponent. E.g., $1.0\times10^{10}$ should be entered as 1.0E+10, and not 1.00E 10 which would be read as the two numbers 1.0 and 10.0.

(3) The first numerical input data record is IDET, INUC, IMAX, IUNT, where IDET is the selection among the detector materials, and INUC is the selection among the options to treat proton nuclear interactions. Selections for both parameters are listed early in the FORTRAN file.

(4) The remaining input parameters are the same as in the old code. In addition to allowing the specification of spectra exponential in energy, the new code also includes spectra exponential in rigidity. The key for these is for EPS(1) $< =$ 0; then S(1) is the e-folding energy, MeV for exponential in energy and MV for exponential in rigidity, S(2) supplies the normalization for the spectrum, and S(3) $>$ 0 selects that the shape is exponential in rigidity rather than in energy. The normalization for these spectra is specified in a somewhat confusing manner, but the value for the integral under the spectrum is given in the printout.

(5) The new code still requires the selection of the energy grid limits for integration over the input spectra for the solar-proton, trapped-proton and electron components, as well as the number of energy points in the grid, that are separate from the actual incident spectral energies that can be read in. This facilitates the preparation of gridded arrays for each component that can be used repeatedly for calculations in different environments within a single run. Rather than using these limits regardless of the spectral energies read in (e.g., forcing dangerous cubic-spline extrapolation in some cases) as was done in the old code, the limits of integration are now set in the following way. For the component in question, let EMIN and EMAX be the selected energy limits, and let $\varepsilon_j$ be the monotonically increasing spectral energies that are read in. Then, the lower limit of integration for that spectrum is max[EMIN,$\varepsilon_1$], and the upper limit is min[EMAX,$\varepsilon_{jmax}$]. This insures that there is no extrapolation of the read-in spectra. For spectra chosen from analytical representations (exponential in energy or rigidity), EMIN and EMAX define the integration limits without modification. For read-in spectra, one should choose EMINs and EMAXs so as not to "waste" large portions of the gridded arrays.

### *Program DOSCON*

The approximate transformations used to convert doses in slabs to those in solid spheres and at the inner surface of spherical shells, for radiation incident isotopically, are outlined in reference [3]. The code requests (1) the name of the SHIELDOSE-2 output array file, which may contain multiple calculations (for different environments), (2) the desired name for the output file, and (3) a value for the outer radius, which holds both for a solid aluminum sphere and for a spherical aluminum shell. In this essentially one-dimensional problem, all dimensions are specified in g/cm$^2$ of Al (actual length can be converted using a density for the aluminum, e.g., 2.7 g/cm$^3$).

The code automatically chooses the range of radii in the solid sphere and thicknesses of the shell that are accessible to the calculation. These ranges are largest when the SHIELDOSE calculation includes depths out to the maximum 50 g/cm$^2$ of aluminum. If no off-center radii in the solid sphere are accessible (i.e., the outer radius is too large), then solid-sphere results are not calculated at all. For large outer radii, the largest accessible shell thickness can be rather small. However, in the limit of infinite outer radius, the dose at the inner surface of the shell approaches twice that behind a plane slab of the same thickness. Therefore, supplemental shell thicknesses are added to cover thicknesses up to the outer radius, but only in the slab approximation. The automatic choices built into the present version of the code make it rather easy to use, but may not be optimum for every problem.

Please let me know of any difficulties, suggested changes, or needed improvements. This is a working version of the package that I did not want to hold back until further refined.

Stephen M. Seltzer
Ionizing Radiation Division
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
(301) 975-5552
Fax: (301) 869-7682
E-mail: seltzer@enh.nist.gov

# Appendix B. SHIELDOSE-2 Program Listing

```
      PROGRAM SD2
C
C     SHIELDOSE-2, VERSION 2.10, 28 APR 94.
C
C        S.M. SELTZER
C        NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
C        GAITHERSBURG, MD 20899
C        (301) 975-5552
C
C        IDET = 1, AL DETECTOR
C               2, GRAPHITE DETECTOR
C               3, SI DETECTOR
C               4, AIR DETECTOR
C               5, BONE DETECTOR
C               6, CALCIUM FLUORIDE DETECTOR
C               7, GALLIUM ARSENIDE DETECTOR
C               8, LITHIUM FLUORIDE DETECTOR
C               9, SILICON DIOXIDE DETECTOR
C              10, TISSUE DETECTOR
C              11, WATER DETECTOR
C
C        INUC = 1, NO NUCLEAR ATTENUATION FOR PROTONS IN AL
C               2, NUCLEAR ATTENUATION, LOCAL CHARGED-SECONDARY ENERGY
C                     DEPOSITION
C               3, NUCLEAR ATTENUATION, LOCAL CHARGED-SECONDARY ENERGY
C                     DEPOSITION, AND APPROX EXPONENTIAL DISTRIBUTION OF
C                     NEUTRON DOSE
C
C        INCIDENT OMNIDIRECTIONAL FLUX IN /ENERGY/CM2/UNIT TIME
C            (SOLAR-FLARE FLUX IN /ENERGY/CM2).
C
C        EUNIT IS CONVERSION FACTOR FROM /ENERGY TO /MEV,
C             E.G., EUNIT = 1000 IF FLUX IS /KEV.
C
C        DURATN IS MISSION DURATION IN MULTIPLES OF UNIT TIME.
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER  (MMAXPI=133,KMAXPI=30,NMAXPI=49,LMAXPI=51,IMIXI=11,
     1   MMAXEI=81,NMAXEI=14,LMAXSI=33+1,LMAXEI=51,LMAXTI=37,LMAXBI=47,
     2   IMAXI=71,NPTSI=1001,JMAXI=301)
      PARAMETER  (NPTSPI=NPTSI,NPTSEI=NPTSI)
      PARAMETER  (ZCON=0.001*2.540005*2.70,ZMCON=10.0/2.70)
      CHARACTER  FILENM*40,PRTFIL*40,ARRFIL*40,TAG*72,DET(IMIXI)*8,
     1   VERSION*4
      DIMENSION  EP(MMAXPI),RP(MMAXPI),RPB(MMAXPI),RPC(MMAXPI),
     1   RPD(MMAXPI),TEPN(KMAXPI),FEPN(KMAXPI),FEPNB(KMAXPI),
     2   FEPNC(KMAXPI),FEPND(KMAXPI),TP(NMAXPI),ZRP(LMAXPI),
     3   DUM(LMAXPI),DALP(NMAXPI,LMAXPI),DALPB(NMAXPI,LMAXPI),
     4   DALPC(NMAXPI,LMAXPI),DALPD(NMAXPI,LMAXPI),
     5   DRATP(NMAXPI,LMAXPI),DRATPB(NMAXPI,LMAXPI),
     6   DRATPC(NMAXPI,LMAXPI),DRATPD(NMAXPI,LMAXPI)
      DIMENSION  EE(MMAXEI),RE(MMAXEI),REB(MMAXEI),REC(MMAXEI),
     1   RED(MMAXEI),YE(MMAXEI),YEB(MMAXEI),YEC(MMAXEI),YED(MMAXEI),
     2   TE(NMAXEI),AR(NMAXEI),ARB(NMAXEI),ARC(NMAXEI),ARD(NMAXEI),
     3   RS(NMAXEI),RSB(NMAXEI),RSC(NMAXEI),RSD(NMAXEI),BS(LMAXSI),
     4   ZRE(LMAXEI),ZS(LMAXTI),ZB(LMAXBI),DALE(NMAXEI,LMAXSI),
     5   DALEB(NMAXEI,LMAXSI),DALEC(NMAXEI,LMAXSI),DALED(NMAXEI,LMAXSI),
     6   DALB(NMAXEI,LMAXTI),DALBB(NMAXEI,LMAXTI),DALBC(NMAXEI,LMAXTI),
     7   DALBD(NMAXEI,LMAXTI),DRATE(NMAXEI,LMAXEI,2),
     8   DRATEB(NMAXEI,LMAXEI,2),DRATEC(NMAXEI,LMAXEI,2),
     9   DRATED(NMAXEI,LMAXEI,2),DRATB(NMAXEI,LMAXBI,2),
     X   DRATBB(NMAXEI,LMAXBI,2),DRATBC(NMAXEI,LMAXBI,2),
     1   DRATBD(NMAXEI,LMAXBI,2)
      DIMENSION  ZM(IMAXI),Z(IMAXI),ZMM(IMAXI),ZL(IMAXI),TPL(NPTSPI),
     1   TPP(NPTSPI),TEL(NPTSEI),ENEWT(NPTSPI),TEE(NPTSEI),RINE(NPTSEI),
     2   RINS(NPTSEI),ARES(NPTSEI),YLDE(NPTSEI),DIN(LMAXPI),
     3   DINB(LMAXPI),DINC(LMAXPI),DIND(LMAXPI),DRIN(LMAXPI),
     4   GP(NPTSPI,IMAXI),GE(NPTSEI,IMAXI,2),GB(NPTSEI,IMAXI,2),
     5   EPS(JMAXI),S(JMAXI),SOL(NPTSPI),SPG(NPTSPI),SEG(NPTSEI),
     6   G(NPTSI),DOSOL(IMAXI,2),DOSP(IMAXI,2),DOSE(IMAXI,2,2),
     7   DOSB(IMAXI,2,2)
      DATA  DET/'Aluminum','Graphite','Silicon','Air','Bone','CaF2',
     1   'GaAs','LiF','SiO2','Tissue','H2O'/
      DATA  ZMIN/1.0E-06/,RADCON/1.6021892E-08/,NBEGE/1/,ENMU/0.03/
      DATA  VERSION/'2.10'/
      CALL LOGO (VERSION)
      PRINT 10
   10 FORMAT (' Enter input filename: ')
      READ 20,FILENM
   20 FORMAT (A)
      OPEN (UNIT=9,FILE=FILENM)
      READ (9,20) PRTFIL
      OPEN (UNIT=10,FILE=PRTFIL)
      READ (9,20) ARRFIL
      OPEN (UNIT=12,FILE=ARRFIL)
      WRITE (10,30) VERSION
      WRITE (12,30) VERSION
   30 FORMAT (' OUTPUT FROM SHIELDOSE-2, VERSION ',A)
      WRITE (10,32) FILENM
      WRITE (12,32) FILENM
   32 FORMAT (' Input filename: ',A)
      WRITE (10,34) PRTFIL
      WRITE (12,34) PRTFIL
   34 FORMAT (' Print-out filename: ',A)
      WRITE (10,36) ARRFIL
      WRITE (12,36) ARRFIL
   36 FORMAT (' Array output filename: ',A)
      WRITE (10,370)
  370 FORMAT (/'  IDET  INUC  IMAX  IUNT')
      READ (9,*) IDET,INUC,IMAX,IUNT
      WRITE (10,380) IDET,INUC,IMAX,IUNT
  380 FORMAT (12I6)
      WRITE (12,380) IDET,IMAX,INUC
      INATT=2
      IF (INUC.EQ.1) INATT=1
      INEWT=0
      IF (INUC.EQ.3) INEWT=1
      PRINT *,' Reading database and preparing base arrays.............'
      PRINT *,'    Protons.............................................'
      OPEN (UNIT=11,FILE='PROTBAS2.DAT')
      READ (11,20) TAG
      READ (11,*) MMAXP,KMAXP,NMAXP,LMAXP,IMIX
      READ (11,*) (EP(M),M=1,MMAXP)
      READ (11,*) (RP(M),M=1,MMAXP)
      READ (11,*) (TEPN(K),K=1,KMAXP)
      READ (11,*) (FEPN(K),K=1,KMAXP)
      READ (11,*) (TP(N),N=1,NMAXP)
      READ (11,*) (ZRP(L),L=1,LMAXP)
      DO 50 N=1,NMAXP
      DO 38 I=1,2
      READ (11,*) (DUM(L),L=1,LMAXP)
      IF (I.NE.INATT) GO TO 38
      DO 390 L=1,LMAXP
  390 DALP(N,L)=DUM(L)
      DALP(N,LMAXP)=(DALP(N,LMAXP-1)/DALP(N,LMAXP-2))*DALP(N,LMAXP-1)
   38 CONTINUE
      DO 40 I=1,IMIX
      READ (11,*) (DUM(L),L=1,LMAXP)
      IF (I.NE.IDET) GO TO 40
      DO 39 L=1,LMAXP
   39 DRATP(N,L)=DUM(L)
   40 CONTINUE
   50 CONTINUE
```

```
      CLOSE (11)
      DO 60 M=1,MMAXP
      EP(M)=LOG(EP(M))
   60 RP(M)=LOG(RP(M))
      DO 65 K=1,KMAXP
   65 TEPN(K)=LOG(TEPN(K))
      DO 70 N=1,NMAXP
   70 TP(N)=LOG(TP(N))
      CALL SCOF (MMAXP,EP,RP,RPB,RPC,RPD)
      CALL SCOF (KMAXP,TEPN,FEPN,FEPNB,FEPNC,FEPND)
      DO 75 L=1,LMAXP
      DO 72 N=1,NMAXP
   72 DALP(N,L)=LOG(DALP(N,L))
      CALL SCOF (NMAXP,TP,DALP(1,L),DALPB(1,L),DALPC(1,L),DALPD(1,L))
   75 CONTINUE
      PRINT *,'    Electrons and bremsstrahlung.........................'
      OPEN (UNIT=11,FILE='ELBRBAS2.DAT')
      READ (11,20) TAG
      READ (11,*) MMAXE,NMAXE,LMAXS,LMAXE,LMAXT,LMAXB,IMIX
      NLENE=NMAXE-NBEGE+1
      READ (11,*) (EE(M),M=1,MMAXE)
      READ (11,*) (RE(M),M=1,MMAXE)
      READ (11,*) (YE(M),M=1,MMAXE)
      READ (11,*) (TE(N),N=1,NMAXE)
      READ (11,*) (AR(N),N=1,NMAXE)
      READ (11,*) (RS(N),N=1,NMAXE)
      READ (11,*) (BS(L),L=1,LMAXS)
      BS(LMAXS+1)=2.0
      READ (11,*) (ZRE(L),L=1,LMAXE)
      READ (11,*) (ZS(L),L=1,LMAXT)
      READ (11,*) (ZB(L),L=1,LMAXB)
      DO 100 N=1,NMAXE
      READ (11,*) (DALE(N,L),L=1,LMAXS)
      DALE(N,LMAXS+1)=1.0E-07
      READ (11,*) (DALB(N,L),L=1,LMAXT)
      DO 90 I=1,IMIX
      DO 80 M=1,2
      READ (11,*) (DUM(L),L=1,LMAXE)
      IF (I.NE.IDET) GO TO 77
      DO 76 L=1,LMAXE
   76 DRATE(N,L,M)=DUM(L)
   77 READ (11,*) (DUM(L),L=1,LMAXB)
      IF (I.NE.IDET) GO TO 80
      DO 78 L=1,LMAXB
   78 DRATB(N,L,M)=DUM(L)
   80 CONTINUE
   90 CONTINUE
  100 CONTINUE
      LMAXS=LMAXS+1
      CLOSE (11)
      DO 110 M=1,MMAXE
      EE(M)=LOG(EE(M))
      RE(M)=LOG(RE(M))
  110 YE(M)=LOG(YE(M))
      DO 120 N=1,NMAXE
      TE(N)=LOG(TE(N))
      AR(N)=LOG(AR(N))
  120 RS(N)=LOG(RS(N))
      DO 130 L=1,LMAXB
  130 ZB(L)=LOG(ZB(L))
      CALL SCOF (MMAXE,EE,RE,REB,REC,RED)
      CALL SCOF (MMAXE,EE,YE,YEB,YEC,YED)
      CALL SCOF (NMAXE,TE,AR,ARB,ARC,ARD)
      CALL SCOF (NMAXE,TE,RS,RSB,RSC,RSD)
      DO 150 L=1,LMAXS
      DO 140 N=NBEGE,NMAXE
  140 DALE(N,L)=LOG(DALE(N,L))
      CALL LCOF (NLENE,TE(NBEGE),DALE(NBEGE,L),DALEB(NBEGE,L),
     1   DALEC(NBEGE,L),DALED(NBEGE,L))
```

```
C     CALL SCOF (NLENE,TE(NBEGE),DALE(NBEGE,L),DALEB(NBEGE,L),
C    1   DALEC(NBEGE,L),DALED(NBEGE,L))
  150 CONTINUE
      DO 170 L=1,LMAXT
      ZS(L)=LOG(ZS(L))
      DO 160 N=NBEGE,NMAXE
  160 DALB(N,L)=LOG(DALB(N,L))
      CALL LCOF (NLENE,TE(NBEGE),DALB(NBEGE,L),DALBB(NBEGE,L),
     1   DALBC(NBEGE,L),DALBD(NBEGE,L))
C     CALL SCOF (NLENE,TE(NBEGE),DALB(NBEGE,L),DALBB(NBEGE,L),
C    1   DALBC(NBEGE,L),DALBD(NBEGE,L))
  170 CONTINUE
C     PRINT *,' Preparing base arrays for selected detector material...'
      DO 220 L=1,LMAXP
      CALL SCOF (NMAXP,TP,DRATP(1,L),DRATPB(1,L),DRATPC(1,L),
     1   DRATPD(1,L))
  220 CONTINUE
      DO 240 M=1,2
      DO 230 L=1,LMAXE
      CALL LCOF (NLENE,TE(NBEGE),DRATE(NBEGE,L,M),
     1   DRATEB(NBEGE,L,M),DRATEC(NBEGE,L,M),DRATED(NBEGE,L,M))
C     CALL SCOF (NLENE,TE(NBEGE),DRATE(NBEGE,L,M),
C    1   DRATEB(NBEGE,L,M),DRATEC(NBEGE,L,M),DRATED(NBEGE,L,M))
  230 CONTINUE
  240 CONTINUE
      DO 260 M=1,2
      DO 250 L=1,LMAXB
      CALL LCOF (NLENE,TE(NBEGE),DRATB(NBEGE,L,M),
     1   DRATBB(NBEGE,L,M),DRATBC(NBEGE,L,M),DRATBD(NBEGE,L,M))
C     CALL SCOF (NLENE,TE(NBEGE),DRATB(NBEGE,L,M),
C    1   DRATBB(NBEGE,L,M),DRATBC(NBEGE,L,M),DRATBD(NBEGE,L,M))
  250 CONTINUE
  260 CONTINUE
      GO TO (440,470,500), IUNT
  440 WRITE (10,450)
  450 FORMAT (/' SHIELD DEPTH (mils)')
      READ (9,*) (ZM(I),I=1,IMAX)
      WRITE (10,455) (ZM(I),I=1,IMAX)
  455 FORMAT (1P6E12.5)
      DO 460 I=1,IMAX
      IF (ZM(I).LE.ZMIN/ZCON) ZM(I)=ZMIN/ZCON
      Z(I)=ZCON*ZM(I)
  460 ZMM(I)=Z(I)*ZMCON
      GO TO 530
  470 WRITE (10,480)
  480 FORMAT (/' SHIELD DEPTH (g/cm2)')
      READ (9,*) (Z(I),I=1,IMAX)
      WRITE (10,455) (Z(I),I=1,IMAX)
      DO 490 I=1,IMAX
      IF (Z(I).LE.ZMIN) Z(I)=ZMIN
      ZM(I)=Z(I)/ZCON
  490 ZMM(I)=Z(I)*ZMCON
      GO TO 530
  500 WRITE (10,510)
  510 FORMAT (/' SHIELD DEPTH (mm)')
      READ (9,*) (ZMM(I),I=1,IMAX)
      WRITE (10,455) (ZMM(I),I=1,IMAX)
      DO 520 I=1,IMAX
      IF (ZMM(I).LE.ZMIN*ZMCON) ZMM(I)=ZMIN*ZMCON
      Z(I)=ZMM(I)/ZMCON
  520 ZM(I)=Z(I)/ZCON
  530 DO 540 I=1,IMAX
  540 ZL(I)=LOG(Z(I))
      WRITE (12,1435) (Z(I),I=1,IMAX)
      WRITE (10,550)
  550 FORMAT(/'      EMINS      EMAXS      EMINP      EMAXP NPTSP      EMINE
     1     EMAXE NPTSE')
      READ (9,*) EMINS,EMAXS,EMINP,EMAXP,NPTSP,EMINE,EMAXE,NPTSE
      WRITE (10,560) EMINS,EMAXS,EMINP,EMAXP,NPTSP,EMINE,EMAXE,NPTSE
```

```
560 FORMAT (4F10.3,I6,2F10.3,I6)
    EMINU=MIN(EMINP,EMINS)
    EMAXU=MAX(EMAXP,EMAXS)
    DEP=LOG(EMAXU/EMINU)/FLOAT(NPTSP-1)
    EMINUL=LOG(EMINU)
    DELP=DEP/3.0
    CALL EINDEX (EMINU,DEP,NPTSP,EMINS,EMAXS,NFSTSB,NLSTSB,NLENSB)
    CALL EINDEX (EMINU,DEP,NPTSP,EMINP,EMAXP,NFSTPB,NLSTPB,NLENPB)
    DO 570 NP=1,NPTSP
    TPL(NP)=EMINUL+FLOAT(NP-1)*DEP
    TPP(NP)=EXP(TPL(NP))
570 CONTINUE
    WRITE (10,580) TPP(NFSTSB),TPP(NLSTSB),TPP(NFSTPB),TPP(NLSTPB),
   1   NPTSP,EMINE,EMAXE,NPTSE
580 FORMAT (4F10.3,I6,2F10.3,I6,'  ADJUSTED VALUES')
    WRITE (12,560) TPP(NFSTSB),TPP(NLSTSB),TPP(NFSTPB),TPP(NLSTPB),
   1   NPTSP,EMINE,EMAXE,NPTSE
    PRINT *,' Preparing mesh arrays to be integrated over spectra....'
    PRINT *,'    Protons.........................................'
    DO 660 NP=1,NPTSP
    CALL BSPOL (TPL(NP),MMAXP,EP,RP,RPB,RPC,RPD,ANS)
    RINP=EXP(ANS)
    DO 610 L=1,LMAXP
    IF (TPL(NP).LT.TP(NMAXP)) GO TO 590
    ANS=DALP(NMAXP,L)
    ANSR=DRATP(NMAXP,L)
    GO TO 605
590 IF (TPL(NP).GT.TP(1)) GO TO 600
    ANS=DALP(1,L)
    ANSR=DRATP(1,L)
    GO TO 605
600 CALL BSPOL (TPL(NP),NMAXP,TP,DALP(1,L),DALPB(1,L),
   1   DALPC(1,L),DALPD(1,L),ANS)
    ANSR=1.0
    IF (IDET.EQ.1) GO TO 605
    CALL BSPOL (TPL(NP),NMAXP,TP,DRATP(1,L),DRATPB(1,L),
   1   DRATPC(1,L),DRATPD(1,L),ANSR)
605 DIN(L)=ANS+LOG(ANSR)
610 CONTINUE
    ENEWT(NP)=0.0
    BENMU=0.0
    IF (INATT.EQ.1) GO TO 620
    IF (TPL(NP).LE.TEPN(1)) GO TO 615
    CALL BSPOL (TPL(NP),KMAXP,TEPN,FEPN,FEPNB,FEPNC,FEPND,ANS)
    ENEWT(NP)=TPP(NP)*ANS
615 BENMU=ENEWT(NP)*ENMU
620 CALL SCOF (LMAXP,ZRP,DIN,DINB,DINC,DIND)
    DO 650 I=1,IMAX
    ZRIN=Z(I)/RINP
    IF (ZRIN.LT.ZRP(LMAXP)) GO TO 640
    GP(NP,I)=0.0
    GO TO 645
640 CALL BSPOL (ZRIN,LMAXP,ZRP,DIN,DINB,DINC,DIND,ANS)
    ANS=EXP(ANS)
    GP(NP,I)=TPP(NP)*ANS/RINP
645 IF (INEWT.EQ.1.AND.TPL(NP).GT.TEPN(1)) GP(NP,I)=GP(NP,I)+
   1   BENMU*EXP(-ENMU*Z(I))
650 CONTINUE
660 CONTINUE
    PRINT *,'    Electrons and bremsstrahlung........................'
    EMINEL=LOG(EMINE)
    DEE=(LOG(EMAXE)-EMINEL)/FLOAT(NPTSE-1)
    DELE=DEE/3.0
    DO 670 NE=1,NPTSE
    TEL(NE)=EMINEL+FLOAT(NE-1)*DEE
    TEE(NE)=EXP(TEL(NE))
    CALL BSPOL (TEL(NE),MMAXE,EE,RE,REB,REC,RED,ANS)
    RINE(NE)=EXP(ANS)
    CALL BSPOL (TEL(NE),NMAXE,TE,RS,RSB,RSC,RSD,ANS)
```

```
      RINS(NE)=RINE(NE)*EXP(ANS)
      CALL BSPOL (TEL(NE),NMAXE,TE,AR,ARB,ARC,ARD,ANS)
      ARES(NE)=EXP(ANS)
      CALL BSPOL (TEL(NE),MMAXE,EE,YE,YEB,YEC,YED,ANS)
  670 YLDE(NE)=EXP(ANS)
      DO 820 M=1,2
      DO 815 NE=1,NPTSE
      DO 700 L=1,LMAXS
      IF (TEL(NE).LT.TE(NMAXE)) GO TO 680
      DIN(L)=DALE(NMAXE,L)
      GO TO 700
  680 IF (TEL(NE).GT.TE(NBEGE)) GO TO 690
      DIN(L)=DALE(NBEGE,L)
      GO TO 700
  690 CALL BSPOL (TEL(NE),NLENE,TE(NBEGE),DALE(NBEGE,L),DALEB(NBEGE,L),
     1   DALEC(NBEGE,L),DALED(NBEGE,L),DIN(L))
  700 CONTINUE
      DO 715 L=1,LMAXE
      DRIN(L)=1.0
      IF (IDET.EQ.1.AND.M.EQ.1) GO TO 715
      IF (TEL(NE).LT.TE(NMAXE)) GO TO 710
      DRIN(L)=DRATE(NMAXE,L,M)
      GO TO 715
  710 IF (TEL(NE).GT.TE(NBEGE)) GO TO 712
      DRIN(L)=DRATE(NBEGE,L,M)
      GO TO 715
  712 CALL BSPOL (TEL(NE),NLENE,TE(NBEGE),DRATE(NBEGE,L,M),
     1   DRATEB(NBEGE,L,M),DRATEC(NBEGE,L,M),DRATED(NBEGE,L,M),DRIN(L))
      IF (DRIN(L).LT.0.0) DRIN(L)=0.0
  715 CONTINUE
C     CALL LCOF (LMAXS,BS,DIN,DINB,DINC,DIND)
      CALL SCOF (LMAXS,BS,DIN,DINB,DINC,DIND)
      DO 740 I=1,IMAX
      ZRIN=Z(I)/RINS(NE)
      IF (ZRIN.LT.BS(LMAXS)) GO TO 730
  720 GE(NE,I,M)=0.0
      GO TO 740
  730 CALL BSPOL (ZRIN,LMAXS,BS,DIN,DINB,DINC,DIND,ANS)
      ANS=EXP(ANS)
      GE(NE,I,M)=TEE(NE)*ANS*ARES(NE)/RINS(NE)
  740 CONTINUE
C     CALL LCOF (LMAXE,ZRE,DRIN,DINB,DINC,DIND)
      CALL SCOF (LMAXE,ZRE,DRIN,DINB,DINC,DIND)
      DO 745 I=1,IMAX
      ZRIN=Z(I)/RINE(NE)
      IF (ZRIN.LT.ZRE(LMAXE)) GO TO 742
      GE(NE,I,M)=GE(NE,I,M)*DRIN(LMAXE)
      GO TO 745
  742 CALL BSPOL (ZRIN,LMAXE,ZRE,DRIN,DINB,DINC,DIND,ANSR)
      IF (ANSR.LT.0.0) ANSR=0.0
      GE(NE,I,M)=GE(NE,I,M)*ANSR
  745 CONTINUE
      DO 780 L=1,LMAXT
      IF (TEL(NE).LT.TE(NMAXE)) GO TO 760
      DIN(L)=DALB(NMAXE,L)
      GO TO 780
  760 IF (TEL(NE).GT.TE(NBEGE)) GO TO 770
      DIN(L)=DALB(NBEGE,L)
      GO TO 780
  770 CALL BSPOL (TEL(NE),NLENE,TE(NBEGE),DALB(NBEGE,L),DALBB(NBEGE,L),
     1   DALBC(NBEGE,L),DALBD(NBEGE,L),DIN(L))
  780 CONTINUE
      DO 795 L=1,LMAXB
      DRIN(L)=1.0
      IF (IDET.EQ.1.AND.M.EQ.1) GO TO 795
      IF (TEL(NE).LT.TE(NMAXE)) GO TO 790
      DRIN(L)=DRATB(NMAXE,L,M)
      GO TO 795
  790 IF (TEL(NE).GT.TE(NBEGE)) GO TO 792
```

```
      DRIN(L)=DRATB(NBEGE,L,M)
      GO TO 795
  792 CALL BSPOL (TEL(NE),NLENE,TE(NBEGE),DRATB(NBEGE,L,M),
     1   DRATBB(NBEGE,L,M),DRATBC(NBEGE,L,M),DRATBD(NBEGE,L,M),DRIN(L))
      IF (DRIN(L).LT.0.0) DRIN(L)=0.0
  795 CONTINUE
      CALL LCOF (LMAXT,ZS,DIN,DINB,DINC,DIND)
C     CALL SCOF (LMAXT,ZS,DIN,DINB,DINC,DIND)
      DO 800 I=1,IMAX
      ZRINL=LOG(Z(I)/RINE(NE))
      CALL BSPOL (ZRINL,LMAXT,ZS,DIN,DINB,DINC,DIND,ANS)
      ANS=EXP(ANS)
      GB(NE,I,M)=TEE(NE)*ANS*YLDE(NE)/RINE(NE)
  800 CONTINUE
      CALL LCOF (LMAXB,ZB,DRIN,DINB,DINC,DIND)
C     CALL SCOF (LMAXB,ZB,DRIN,DINB,DINC,DIND)
      DO 812 I=1,IMAX
      IF (ZL(I).LT.ZB(LMAXB)) GO TO 810
      GB(NE,I,M)=GB(NE,I,M)*DRIN(LMAXB)
      GO TO 812
  810 CALL BSPOL (ZL(I),LMAXB,ZB,DRIN,DINB,DINC,DIND,ANSR)
      IF (ANSR.LT.0.0) ANSR=0.0
      GB(NE,I,M)=GB(NE,I,M)*ANSR
  812 CONTINUE
  815 CONTINUE
  820 CONTINUE
      PRINT *,' Performing calculations for input spectra.............'
  830 WRITE (10,840)
  840 FORMAT (/)
      WRITE (10,850)
  850 FORMAT (' ')
      READ (9,20,END=1440) TAG
      PRINT 860, TAG
  860 FORMAT (4X,A)
      WRITE (10,20) TAG
      WRITE (12,20) TAG
      WRITE (10,870)
  870 FORMAT (/' JSMAX JPMAX JEMAX       EUNIT      DURATN')
      READ (9,*)  JSMAX,JPMAX,JEMAX,EUNIT,DURATN
      WRITE (10,880) JSMAX,JPMAX,JEMAX,EUNIT,DURATN
      WRITE (12,880) JSMAX,JPMAX,JEMAX,EUNIT,DURATN
  880 FORMAT (3I6,1P2E12.5)
      IF (DURATN.LE.0.0) DURATN=1.0
      DELTAS=RADCON*DELP/4.0
      DELTAP=DURATN*RADCON*DELP/4.0
      DELTAE=DURATN*RADCON*DELE/4.0
      IF (EUNIT.LE.0.0) EUNIT=1.0
      ISOL=2
      IF (JSMAX.LT.3) GO TO 900
      ISOL=1
      WRITE (10,885)
  885 FORMAT (//' E(MeV)')
      READ (9,*) (EPS(J),J=1,JSMAX)
      WRITE (10,905) (EPS(J),J=1,JSMAX)
      WRITE (12,905) (EPS(J),J=1,JSMAX)
      WRITE (10,890)
  890 FORMAT (/' SOLAR PROTON SPECTRUM (/energy/cm2)')
      READ (9,*) (S(J),J=1,JSMAX)
      WRITE (10,905) (S(J),J=1,JSMAX)
      WRITE (12,905) (S(J),J=1,JSMAX)
      NLENS=NLENSB
      NFSTS=NFSTSB
      NLSTS=NLSTSB
      CALL SPECTR (JSMAX,EPS,S,EUNIT,EMINU,DEP,NPTSP,NFSTS,NLSTS,NLENS,
     1   TPP,TPL,SOL)
      WRITE (10,891) TPP(NFSTS),TPP(NLSTS),NLENS
  891 FORMAT (/' SPECTRUM INTEGRATED FROM',1PE11.4,' TO',1PE11.4,
     1   ' MeV, USING',I5,' POINTS')
      DO 892 NP=NFSTS,NLSTS
  892 G(NP)=SOL(NP)*ENEWT(NP)
      CALL INTEG (DELP,G(NFSTS),NLENS,ENEUT)
      DO 894 NP=NFSTS,NLSTS
  894 G(NP)=SOL(NP)*TPP(NP)
      CALL INTEG (DELP,G(NFSTS),NLENS,EAV)
      ENEUT=ENEUT/EAV
      WRITE (10,896) ENEUT
  896 FORMAT (/' ASSUMED FRACTION OF BEAM ENERGY INTO NEUTRON ENERGY =',
     1   1PE12.5)
  900 ITRP=2
      IF (JPMAX.LT.3) GO TO 920
      ITRP=1
      WRITE (10,885)
      READ (9,*) (EPS(J),J=1,JPMAX)
      WRITE (10,905) (EPS(J),J=1,JPMAX)
      WRITE (12,905) (EPS(J),J=1,JPMAX)
  905 FORMAT (1P10E12.4)
      WRITE (10,910)
  910 FORMAT (/' TRAPPED PROTON SPECTRUM (/energy/cm2/time)')
      READ (9,*) (S(J),J=1,JPMAX)
      WRITE (10,905) (S(J),J=1,JPMAX)
      WRITE (12,905) (S(J),J=1,JPMAX)
      NLENP=NLENPB
      NFSTP=NFSTPB
      NLSTP=NLSTPB
      CALL SPECTR (JPMAX,EPS,S,EUNIT,EMINU,DEP,NPTSP,NFSTP,NLSTP,NLENP,
     1   TPP,TPL,SPG)
      WRITE (10,891) TPP(NFSTP),TPP(NLSTP),NLENP
      DO 912 NP=NFSTP,NLSTP
  912 G(NP)=SPG(NP)*ENEWT(NP)
      CALL INTEG (DELP,G(NFSTP),NLENP,ENEUT)
      DO 914 NP=NFSTP,NLSTP
  914 G(NP)=SPG(NP)*TPP(NP)
      CALL INTEG (DELP,G(NFSTP),NLENP,EAV)
      ENEUT=ENEUT/EAV
      WRITE (10,896) ENEUT
  920 ILEC=2
      IF (JEMAX.LT.3) GO TO 940
      ILEC=1
      WRITE (10,885)
      READ (9,*) (EPS(J),J=1,JEMAX)
      WRITE (10,905) (EPS(J),J=1,JEMAX)
      WRITE (12,905) (EPS(J),J=1,JEMAX)
      WRITE (10,930)
  930 FORMAT (/' ELECTRON SPECTRUM (/energy/cm2/time)')
      READ (9,*) (S(J),J=1,JEMAX)
      WRITE (10,905) (S(J),J=1,JEMAX)
      WRITE (12,905) (S(J),J=1,JEMAX)
      NLENE=NPTSE
      NFSTE=1
      NLSTE=NPTSE
      CALL SPECTR (JEMAX,EPS,S,EUNIT,EMINE,DEE,NPTSE,NFSTE,NLSTE,NLENE,
     1   TEE,TEL,SEG)
      WRITE (10,891) TEE(NFSTE),TEE(NLSTE),NLENE
  940 GO TO (980,950), ISOL
  950 DO 960 NP=NFSTS,NLSTS
  960 SOL(NP)=0.0
      DO 970 J=1,2
      DO 970 I=1,IMAX
  970 DOSOL(I,J)=0.0
      GO TO 1010
  980 DO 1000 I=1,IMAX
      DO 990 NP=NFSTS,NLSTS
  990 G(NP)=SOL(NP)*GP(NP,I)
      CALL INTEG (DELTAS,G(NFSTS),NLENS,DOSOL(I,1))
 1000 CONTINUE
      CALL SPHERE (ZL,DOSOL(1,1),IMAX,DOSOL(1,2))
 1010 GO TO (1050,1020), ITRP
 1020 DO 1030 NP=NFSTP,NLSTP
```

```
1030 SPG(NP)=0.0
     DO 1040 J=1,2
     DO 1040 I=1,IMAX
1040 DOSP(I,J)=0.0
     GO TO 1080
1050 DO 1070 I=1,IMAX
     DO 1060 NP=NFSTP,NLSTP
1060 G(NP)=SPG(NP)*GP(NP,I)
     CALL INTEG (DELTAP,G(NFSTP),NLENP,DOSP(I,1))
1070 CONTINUE
     CALL SPHERE (ZL,DOSP(1,1),IMAX,DOSP(1,2))
1080 GO TO (1110,1090), ILEC
1090 DO 1100 J=1,2
     DO 1100 M=1,2
     DO 1100 I=1,IMAX
     DOSE(I,M,J)=0.0
1100 DOSB(I,M,J)=0.0
     GO TO 1160
1110 DO 1150 M=1,2
     DO 1130 I=1,IMAX
     DO 1120 NE=NFSTE,NLSTE
     G(NE)=SEG(NE)*GE(NE,I,M)
1120 SPG(NE)=SEG(NE)*GB(NE,I,M)
     CALL INTEG (DELTAE,G(NFSTE),NLENE,DOSE(I,M,1))
     CALL INTEG (DELTAE,SPG(NFSTE),NLENE,DOSB(I,M,1))
1130 CONTINUE
     GO TO (1140,1150), M
1140 CALL SPHERE (ZL,DOSE(1,M,1),IMAX,DOSE(1,M,2))
     CALL SPHERE (ZL,DOSB(1,M,1),IMAX,DOSB(1,M,2))
1150 CONTINUE
1160 J=1
     DO 1340 M=2,1,-1
     GO TO (1190,1170), M
1170 WRITE (10,1180)
1180 FORMAT(//' DOSE AT TRANSMISSION SURFACE OF FINITE ALUMINUM SLAB SH
    1IELDS')
     GO TO 1210
1190 WRITE (10,1200)
1200 FORMAT(//' DOSE IN SEMI-INFINITE ALUMINUM MEDIUM')
1210 WRITE (10,1230) DET(IDET)
1230 FORMAT (/' rads ',A)
     IF (INATT.EQ.1) WRITE (10,1240)
1240 FORMAT (/' Proton results without nuclear attenuation')
     IF (INATT.EQ.2) WRITE (10,1250)
1250 FORMAT (/' Proton results with approximate treatment of nuclear at
    1tenuation')
     IF (INATT.EQ.2.AND.INEWT.EQ.0) WRITE (10,1260)
1260 FORMAT ( '    neglecting transport of energy by neutrons')
     IF (INATT.EQ.2.AND.INEWT.EQ.1) WRITE (10,1270)
1270 FORMAT ( '    and crude exponential transport of energy by neutron
    1s')
     WRITE (10,1310)
1310 FORMAT(//'    Z(mils)      Z(mm)   Z(g/cm2)   ELECTRON    BREMS
    1  EL+BR     TRP PROT   SOL PROT  EL+BR+TRP    TOTAL')
     WRITE (10,850)
     DO 1330 I=1,IMAX
     DOSEB=DOSE(I,M,J)+DOSB(I,M,J)
     DOSEBP=DOSEB+DOSP(I,J)
     DOST=DOSEBP+DOSOL(I,J)
     WRITE (10,1320) ZM(I),ZMM(I),Z(I),DOSE(I,M,J),DOSB(I,M,J),DOSEB,
    1   DOSP(I,J),DOSOL(I,J),DOSEBP,DOST
1320 FORMAT (1P10E11.3)
     IF (FLOAT(I/10).EQ.0.1*FLOAT(I)) WRITE (10,850)
1330 CONTINUE
1340 CONTINUE
     J=2
     M=1
     WRITE (10,1350)
1350 FORMAT (//' 1/2 DOSE AT CENTER OF ALUMINUM SPHERES')
```

```
      WRITE (10,1230) DET(IDET)
      IF (INATT.EQ.1) WRITE (10,1240)
      IF (INATT.EQ.2) WRITE (10,1250)
      IF (INATT.EQ.2.AND.INEWT.EQ.0) WRITE (10,1260)
      IF (INATT.EQ.2.AND.INEWT.EQ.1) WRITE (10,1270)
      WRITE (10,1310)
      WRITE (10,850)
      DO 1410 I=1,IMAX
      DOSEB=DOSE(I,M,J)+DOSB(I,M,J)
      DOSEBP=DOSEB+DOSP(I,J)
      DOST=DOSEBP+DOSOL(I,J)
      WRITE (10,1320) ZM(I),ZMM(I),Z(I),DOSE(I,M,J),DOSB(I,M,J),DOSEB,
     1   DOSP(I,J),DOSOL(I,J),DOSEBP,DOST
      IF (FLOAT(I/10).EQ.0.1*FLOAT(I)) WRITE (10,850)
 1410 CONTINUE
      DO 1437 J=1,2
      WRITE (12,1435) (DOSOL(I,J),I=1,IMAX)
      WRITE (12,1435) (DOSP(I,J),I=1,IMAX)
 1435 FORMAT (1P10E10.3)
 1437 CONTINUE
      DO 1438 M=2,1,-1
      WRITE (12,1435) (DOSE(I,M,1),I=1,IMAX)
      WRITE (12,1435) (DOSB(I,M,1),I=1,IMAX)
 1438 CONTINUE
      WRITE (12,1435) (DOSE(I,1,2),I=1,IMAX)
      WRITE (12,1435) (DOSB(I,1,2),I=1,IMAX)
      GO TO 830
 1440 PRINT 32, FILENM
      PRINT 34, PRTFIL
      PRINT 36, ARRFIL
      print 1500,CHAR(27)
 1500 format (' ',A1,'[0m')
      STOP
      END
C     SUBROUTINE EINDEX (EMINB,DE,NPTS,EMIN,EMAX,NFST,NLST,NLEN), 28 APR 94.
      SUBROUTINE EINDEX (EMINB,DE,NPTS,EMIN,EMAX,NFST,NLST,NLEN)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      NFST=LOG(EMIN/EMINB)/DE+0.5
      NFST=NFST+1
      IF (NFST.LT.1) NFST=1
      NLST=LOG(EMAX/EMINB)/DE+0.5
      NLST=NLST+1
      IF (NLST.GT.NPTS) NLST=NPTS
      NLEN=NLST-NFST+1
      RETURN
      END
C     SUBROUTINE SPECTR (JMAX,EPS,S,EUNIT,EMINB,DEL,NPTS,NFST,NLST,NLEN,
C    1   T,TL,SP), 28 APR 94.
      SUBROUTINE SPECTR (JMAX,EPS,S,EUNIT,EMINB,DEL,NPTS,NFST,NLST,NLEN,
     1   T,TL,SP)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER  (JMAXI=301,NPTSI=1001)
      DIMENSION  EPS(1),S(1),T(1),TL(1),SP(1),BCOF(JMAXI),CCOF(JMAXI),
     1   DCOF(JMAXI),G(NPTSI)
C           ARGLIM = 700.0 (DOUBLE PRECISION), 85.0 (SINGLE PRECISION)
      DATA  EMC2/938.27231/,ARGLIM/85.0/
      DELTA=DEL/3.0
      IF (EPS(1).GT.0.0) GO TO 20
      ALPHA=S(1)
      BETA=S(2)
      IF (BETA.LE.0.0) BETA=1.0
      BETA=BETA/ALPHA
      DO 10 N=NFST,NLST
      SP(N)=0.0
      G(N)=0.0
      IF (S(3).LE.0.0) GO TO 6
      P=SQRT(T(N)*(T(N)+2.0*EMC2))
      ARG=P/ALPHA
      IF (ARG.GT.ARGLIM) GO TO 10
      SP(N)=T(N)*BETA*((T(N)+EMC2)/P)*EXP(-ARG)
      GO TO 8
    6 ARG=T(N)/ALPHA
      IF (ARG.GT.ARGLIM) GO TO 10
      SP(N)=T(N)*BETA*EXP(-ARG)
    8 G(N)=T(N)*SP(N)
   10 CONTINUE
      GO TO 50
   20 CALL EINDEX (EMINB,DEL,NPTS,EPS(1),EPS(JMAX),NFST,NLST,NLEN)
      DO 30 J=1,JMAX
      EPS(J)=LOG(EPS(J))
   30 S(J)=LOG(EUNIT*S(J))
      CALL SCOF (JMAX,EPS,S,BCOF,CCOF,DCOF)
      DO 40 N=NFST,NLST
      CALL BSPOL (TL(N),JMAX,EPS,S,BCOF,CCOF,DCOF,ANS)
      SP(N)=T(N)*EXP(ANS)
   40 G(N)=T(N)*SP(N)
   50 CALL INTEG (DELTA,SP(NFST),NLEN,SIN)
      CALL INTEG (DELTA,G(NFST),NLEN,EBAR)
      EBAR=EBAR/SIN
      WRITE (10,60)
   60 FORMAT (/'    INT SPEC    EAV(MeV)')
      WRITE (10,70) SIN,EBAR
   70 FORMAT (1PE12.4,0PF12.5)
      RETURN
      END
C     SUBROUTINE SPHERE (ZL,DOSE,IMAX,DOSPH), 26 JAN 93.
      SUBROUTINE SPHERE (ZL,DOSE,IMAX,DOSPH)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER  (IMAXI=71)
      DIMENSION  ZL(1),DOSE(1),DOSPH(1),DOSL(IMAXI),BCOF(IMAXI),
     1   CCOF(IMAXI),DCOF(IMAXI)
      DO 10 I=1,IMAX
      IF (DOSE(I).LE.0.0) GO TO 20
   10 DOSL(I)=LOG(DOSE(I))
      I=IMAX+1
   20 IMIX=I-1
      IF (IMIX.LT.3) GO TO 40
      CALL SCOF (IMIX,ZL,DOSL,BCOF,CCOF,DCOF)
      BCOF(IMIX)=BCOF(IMIX-1)+(2.0*CCOF(IMIX-1)+3.0*DCOF(IMIX-1)*
     1   (ZL(IMIX)-ZL(IMIX-1)))*(ZL(IMIX)-ZL(IMIX-1))
      DO 30 I=1,IMIX
   30 DOSPH(I)=DOSE(I)*(1.0-BCOF(I))
   40 IMIX1=IMIX+1
      IF (IMIX1.GT.IMAX) RETURN
      DO 50 I=IMIX1,IMAX
   50 DOSPH(I)=0.0
      RETURN
      END
      SUBROUTINE SCOF(N,X,Y,B,C,D)
C        REINSCH ALGORITHM, VIA MJB, 22 FEB 83
C        Y(S)=((D(J)*(X-X(J))+C(J))*(X-X(J))+B(J))*(X-X(J))+Y(J)
C              FOR X BETWEEN X(J) AND X(J+1)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(1),Y(1),B(1),C(1),D(1)
      N1=N-1
      S=0.0
      DO 10 J=1,N1
      D(J)=X(J+1)-X(J)
      R=(Y(J+1)-Y(J))/D(J)
      C(J)=R-S
   10 S=R
      S=0.0
      R=0.0
      C(1)=0.0
      C(N)=0.0
      DO 20 J=2,N1
      C(J)=C(J)+R*C(J-1)
      B(J)=(X(J-1)-X(J+1))*2.0-R*S
      S=D(J)
   20 R=S/B(J)
      DO 30 JR=N1,2,-1
   30 C(JR)=(D(JR)*C(JR+1)-C(JR))/B(JR)
      DO 40 J=1,N1
      S=D(J)
      R=C(J+1)-C(J)
      D(J)=R/S
      C(J)=3.0*C(J)
   40 B(J)=(Y(J+1)-Y(J))/S-(C(J)+R)*S
      RETURN
      END
      SUBROUTINE BSPOL(S,N,X,Y,B,C,D,T)
C        BINARY SEARCH, X ASCENDING OR DESCENDING
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(1),Y(1),B(1),C(1),D(1)
      IF (X(1).GT.X(N)) GO TO 10
      IDIR=0
      MLB=0
      MUB=N
      GO TO 20
   10 IDIR=1
      MLB=N
      MUB=0
   20 IDIR1=IDIR-1
      IF (S.GE.X(MUB+IDIR)) GO TO 60
      IF (S.LE.X(MLB-IDIR1)) GO TO 70
      ML=MLB
      MU=MUB
      GO TO 40
   30 IF (IABS(MU-ML).LE.1) GO TO 80
   40 MAV=(ML+MU)/2
      IF (S.LT.X(MAV)) GO TO 50
      ML=MAV
      GO TO 30
   50 MU=MAV
      GO TO 30
   60 MU=MUB+IDIR+IDIR1
      GO TO 90
   70 MU=MLB-IDIR-IDIR1
      GO TO 90
   80 MU=MU+IDIR1
   90 Q=S-X(MU)
      T=((D(MU)*Q+C(MU))*Q+B(MU))*Q+Y(MU)
      RETURN
      END
      SUBROUTINE LCOF (NMAX,X,F,B,C,D)
C        26 JAN 93.  SIMPLE LINEAR INTERPOLATION
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  X(1), F(1), B(1), C(1), D(1)
      DO 10 N=1,NMAX-1
      B(N)=(F(N+1)-F(N))/(X(N+1)-X(N))
      C(N)=0.0
      D(N)=0.0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE INTEG (DELTA,G,N,RESULT)
C          INCLUDES N=1
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION G(8)
      NL1=N-1
      NL2=N-2
      IF (REAL (N) -2.0*REAL (N/2)) 100,100,10
   10  IF (N-1) 15,15,20
   15 SIGMA=0.0
      GO TO 70
   20 IF(N-3) 30,30,40
   30 SIGMA=G(1)+4.0*G(2)+G(3)
```

```
      GO TO 70
   40 SUM4=0.0
      DO 50 K=2,NL1,2
   50 SUM4=SUM4+G(K)
      SUM2=0.0
      DO 60 K=3,NL2,2
   60 SUM2=SUM2+G(K)
      SIGMA=G(1)+4.0*SUM4+2.0*SUM2+G(N)
   70 RESULT=DELTA*SIGMA
      RETURN
  100 IF(N-2)110,110,120
  110 SIGMA=1.5*(G(1)+G(2))
      GO TO 70
  120 IF(N-4)130,130,140
  130 SIGMA=1.125*(G(1)+3.0*G(2)+3.0*G(3)+G(4))
      GO TO 70
  140 IF(N-6)150,150,160
  150 SIGMA=G(1)+3.875*G(2)+2.625*G(3)+2.625*G(4)+3.875*G(5)+G(6)
      GO TO 70
  160 IF (N-8)170,170,180
  170 SIGMA=G(1)+3.875*G(2)+2.625*G(3)+2.625*G(4)+3.875*G(5)+2.0*G(6)
     1   +4.0*G(7)+G(8)
      GO TO 70
  180 SIG6=G(1)+3.875*G(2)+2.625*G(3)+2.625*G(4)+3.875*G(5)+G(6)
      SUM4=0.0
      DO 190 K=7,NL1,2
  190 SUM4=SUM4+G(K)
      SUM2=0.0
      DO 200 K=8,NL2,2
  200 SUM2=SUM2+G(K)
      SIGMA=SIG6+G(6)+4.0*SUM4+2.0*SUM2+G(N)
      GO TO 70
      END
      SUBROUTINE LOGO (VERSION)
C        28PR 94.
      CHARACTER  VERSION*4,blue*10,green*7
      CALL CLS
      blue=CHAR(27)//'[40;36;1m'
      print 5,blue
    5 format (1X,A)
      PRINT 10,' '
   10 FORMAT (a80)
      PRINT 20
   20 FORMAT (6X,'┌',66('─'),'┐',6(' '))
      PRINT 25
   25 FORMAT (6X,'│',66X,'│',6(' '))
      PRINT 30
   30 FORMAT (6X,'│',27X,'SHIELDOSE-2',28X,'│',6(' '))
      PRINT 25
      PRINT 40
   40 FORMAT (6X,'│',15X,'A Computer Code for Space-Shielding',16X,'│',
     1   6(' '))
      PRINT 50
   50 FORMAT (6X,'│',19X,'Radiation Dose Calculations',20X,'│',6(' '))
      PRINT 25
      PRINT 60, VERSION
   60 FORMAT (6X,'│',27X,'Version ',A,27X,'│',6(' '))
      PRINT 25
      PRINT 70
   70 FORMAT (6X,'│',28X,'Written by',28X,'│',6(' '))
      PRINT 80
   80 FORMAT (6X,'│',24X,'STEPHEN M. SELTZER',24x,'│',6(' '))
      PRINT 90
   90 FORMAT (6X,'│',10X,'National Institute of Standards and Technology
     1',10x,'│',6(' '))
      PRINT 100
  100 FORMAT (6X,'│',20X,'Gaithersburg, MD 20899, USA',19X,'│',6(' '))
      PRINT 25
      PRINT 110
  110 FORMAT (6X,'┗',66('━'),'┛',6(' '))
      green=CHAR(27)//'[0;32m'
      print 5,green
      RETURN
      END
      SUBROUTINE CLS
C        8 SEP 88.
      PRINT 10,CHAR(27)
   10 FORMAT (' ',A1,'[2J')
      RETURN
      END
```

# Appendix C. DOSCON Program Listing

```
      PROGRAM DOSCON
C
C     DOSCON, VERSION 1.00, 28 APR 94.
C
C        S.M. SELTZER
C        NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
C        GAITHERSBURG, MD 20899
C        (301) 975-5552
C
C        CONVERSION OF SHIELDOSE RESULTS TO OFF-CENTER DOSE IN SPHERES
C        OR TO DOSE AT SURFACE OF SPHERICAL SHELL
C
      PARAMETER  (IMIXI=11,IMAXI=71,JMAXI=301,NMAXI=51,NRAT=12,NPTS=501)
      PARAMETER  (NRMAX=NMAXI,NTMAX=NMAXI+NRAT-2,NTMAXI=NTMAX+NMAXI)
      CHARACTER  FILEIN*40,FILENM*40,TAG*80,DET(IMIXI)*8
      DIMENSION  Z(IMAXI),ZL(IMAXI),EPS(301),S(301),DOSS(IMAXI),
     1   DOSP(IMAXI),DOSSH(IMAXI),DOSPH(IMAXI),DOSEF(IMAXI),
     2   DOSBF(IMAXI),DOSE(IMAXI),DOSB(IMAXI),DOSEH(IMAXI),DOSBH(IMAXI),
     3   DOSEB(IMAXI),DOSEBH(IMAXI),R(NMAXI),DOSRS(NMAXI),DOSRP(NMAXI),
     4   DOSREB(NMAXI),T(NTMAXI),RT(NTMAX),DOSTS(NTMAX),DOSTP(NTMAX),
     5   DOSTEB(NTMAX),DOSZS(NTMAXI),DOSZP(NTMAXI),DOSZEB(NTMAXI),
     6   BCOFS(IMAXI),CCOFS(IMAXI),DCOFS(IMAXI),BCOFP(IMAXI),
     7   CCOFP(IMAXI),DCOFP(IMAXI),BCOFE(IMAXI),CCOFE(IMAXI),
     8   DCOFE(IMAXI),G(NPTS),TRAT(NRAT)
      DATA  ZMIN/1.0E-06/
      DATA  DET/'Aluminum','Graphite','Silicon','Air','Bone','CaF2',
     1   'GaAs','LiF','SiO2','Tissue','H2O'/
      DATA  TRAT/0.0,0.00001,0.00002,0.00005,0.0001,0.0002,0.0005,0.001,
     1   0.002,0.005,0.01,0.02/
      FNPTS1=NPTS-1
      PRINT 10
   10 FORMAT (' Enter filename for SHIELDOSE-2 data input: ')
      READ 20, FILEIN
   20 FORMAT (A)
      OPEN (UNIT=8,FILE=FILEIN)
      PRINT 22
   22 FORMAT (' Enter filename for output: ')
      READ 20, FILENM
      OPEN (UNIT=9,FILE=FILENM)
      WRITE (9,24) FILEIN
   24 FORMAT (' OUTPUT OF DOSCON FOR SD2 INPUT FILE: ',A)
      PRINT 26
   26 FORMAT (' Enter sphere outer radius in g/cm2 of Al: ')
      READ *,RADIUS
      IF (RADIUS.LE.0.0) STOP
      WRITE (9,28) RADIUS
   28 FORMAT (/' RESULTS FOR SPHERE OUTER RADIUS (g/cm2)= ',1PE12.5)
      RADL=LOG(RADIUS)
      DO 30 K=1,4
      READ (8,20) TAG
   30 CONTINUE
      READ (8,*) IDET,IMAX,INUC
      READ (8,*) (Z(I),I=1,IMAX)
      READ (8,*) SEMIN,SEMAX,TEMIN,TEMAX,NPTSP,EMINE,EMAXE,NPTSE
      ZMAX=Z(IMAX)
      DO 32 I=1,IMAX
   32 ZL(I)=LOG(Z(I))
   33 READ (8,20,END=999) TAG
      WRITE (9,530)
      WRITE (9,530)
      WRITE (9,530)
      WRITE (9,20) TAG
      PRINT 3300, TAG
 3300 FORMAT (' DOSCON: ',A)
```

```
      READ (8,*)  JSMAX,JPMAX,JEMAX,EUNIT,DURATN
      IF (JSMAX.LT.3) GO TO 34
      READ (8,*) (EPS(J),J=1,JSMAX)
      READ (8,*) (S(J),J=1,JSMAX)
   34 IF (JPMAX.LT.3) GO TO 36
      READ (8,*) (EPS(J),J=1,JPMAX)
      READ (8,*) (S(J),J=1,JPMAX)
   36 IF (JEMAX.LT.3) GO TO 38
      READ (8,*) (EPS(J),J=1,JEMAX)
      READ (8,*) (S(J),J=1,JEMAX)
   38 READ (8,40) (DOSS(I),I=1,IMAX)
      READ (8,40) (DOSP(I),I=1,IMAX)
      READ (8,40) (DOSSH(I),I=1,IMAX)
      READ (8,40) (DOSPH(I),I=1,IMAX)
      READ (8,40) (DOSEF(I),I=1,IMAX)
      READ (8,40) (DOSBF(I),I=1,IMAX)
      READ (8,40) (DOSE(I),I=1,IMAX)
      READ (8,40) (DOSB(I),I=1,IMAX)
      READ (8,40) (DOSEH(I),I=1,IMAX)
      READ (8,40) (DOSBH(I),I=1,IMAX)
   40 FORMAT (1P10E10.3)
      ISR=1
      RMAX=MIN(RADIUS,ZMAX-RADIUS)
      IF (RADIUS.GT.ZMAX) THEN
         ISR=0
         RMAX=RADIUS
      ENDIF
      TMAX=RADIUS
      IST=1
      IF (RADIUS.GT.ZMAX) THEN
         IST=0
         TMAX=RADIUS*(1.0-SQRT(1.0-(ZMAX/RADIUS)**2))
      ENDIF
      IF (JSMAX.GE.3) THEN
         DO 60 I=1,IMAX
         IF (DOSS(I).LE.0.0) GO TO 65
         DOSS(I)=LOG(DOSS(I))
   60    DOSSH(I)=LOG(DOSSH(I))
         I=IMAX+1
   65    IMAXS=I-1
C        CALL SCOF (IMAXS,Z,DOSS,BCOFS,CCOFS,DCOFS)
C        CALL BSPOL (0.0,IMAXS,Z,DOSS,BCOFS,CCOFS,DCOFS,ANS)
C        DOSS0=EXP(ANS)
         DOSS0=EXP(DOSS(1))
      ENDIF
      IF (JPMAX.GE.3) THEN
         DO 70 I=1,IMAX
         IF (DOSP(I).LE.0.0) GO TO 75
         DOSP(I)=LOG(DOSP(I))
   70    DOSPH(I)=LOG(DOSPH(I))
         I=IMAX+1
   75    IMAXP=I-1
C        CALL SCOF (IMAXP,Z,DOSP,BCOFP,CCOFP,DCOFP)
C        CALL BSPOL (0.0,IMAXP,Z,DOSP,BCOFP,CCOFP,DCOFP,ANS)
C        DOSP0=EXP(ANS)
         DOSP0=EXP(DOSP(1))
      ENDIF
      IF (JEMAX.GE.3) THEN
         DO 80 I=1,IMAX
         DOSEB(I)=DOSE(I)+DOSB(I)
         DOSEBH(I)=DOSEH(I)+DOSBH(I)
         IF (DOSEB(I).LE.0.0) GO TO 85
         DOSEB(I)=LOG(DOSEB(I))
   80    DOSEBH(I)=LOG(DOSEBH(I))
         I=IMAX+1
   85    IMAXE=I-1
C        CALL SCOF (IMAXE,Z,DOSEB,BCOFE,CCOFE,DCOFE)
C        CALL BSPOL (0.0,IMAXE,Z,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
C        DOSEB0=EXP(ANS)
```

```
         DOSEB0=EXP(DOSEB(1))
      ENDIF
      DR=RMAX/FLOAT(NRMAX-1)
      DO 90 N=1,NRMAX
      DOSRS(N)=0.0
      DOSRP(N)=0.0
      DOSREB(N)=0.0
   90 R(N)=FLOAT(N-1)*DR
      R(NRMAX)=RMAX
      DO 100 N=1,NRAT
      DOSTS(N)=0.0
      DOSTP(N)=0.0
      DOSTEB(N)=0.0
      DOSZS(N)=0.0
      DOSZP(N)=0.0
      DOSZEB(N)=0.0
      T(N)=TRAT(N)*TMAX
  100 RT(N)=RADIUS-T(N)
      DT=TRAT(NRAT)*TMAX
      DO 110 N=NRAT+1,NTMAX
      DOSTS(N)=0.0
      DOSTP(N)=0.0
      DOSTEB(N)=0.0
      DOSZS(N)=0.0
      DOSZP(N)=0.0
      DOSZEB(N)=0.0
      T(N)=T(N-1)+DT
  110 RT(N)=RADIUS-T(N)
      T(NTMAX)=TMAX
      NZMAX=NTMAX
      IF (IST.EQ.1) GO TO 125
      NZMAX=(RADIUS-TMAX)/DT
      IF (NZMAX.GT.NMAXI) NZMAX=NMAXI
      ZTMAX=MIN(RADIUS,ZMAX)
      DZ=(ZTMAX-TMAX)/FLOAT(NZMAX)
      NZMAX=NTMAX+NZMAX
      DO 120 N=NTMAX+1,NZMAX
      DOSZS(N)=0.0
      DOSZP(N)=0.0
      DOSZEB(N)=0.0
  120 T(N)=T(N-1)+DZ
      T(NZMAX)=ZTMAX
C        GET DOSES AT CENTERS OF SPHERES, IF APPLICABLE
  125 IF (ISR.EQ.1.OR.IST.EQ.1) THEN
         IF (JSMAX.LT.3) GO TO 130
            CALL SCOF (IMAXS,ZL,DOSSH,BCOFS,CCOFS,DCOFS)
            CALL BSPOL (RADL,IMAXS,ZL,DOSSH,BCOFS,CCOFS,DCOFS,ANS)
            ANS=2.0*EXP(ANS)
            IF (ISR.EQ.1) DOSRS(1)=ANS
            IF (IST.EQ.1) DOSTS(NTMAX)=ANS
  130    IF (JPMAX.LT.3) GO TO 140
            CALL SCOF (IMAXP,ZL,DOSPH,BCOFP,CCOFP,DCOFP)
            CALL BSPOL (RADL,IMAXP,ZL,DOSPH,BCOFP,CCOFP,DCOFP,ANS)
            ANS=2.0*EXP(ANS)
            IF (ISR.EQ.1) DOSRP(1)=ANS
            IF (IST.EQ.1) DOSTP(NTMAX)=ANS
  140    IF (JEMAX.LT.3) GO TO 150
            CALL SCOF (IMAXE,ZL,DOSEBH,BCOFE,CCOFE,DCOFE)
            CALL BSPOL (RADL,IMAXE,ZL,DOSEBH,BCOFE,CCOFE,DCOFE,ANS)
            ANS=2.0*EXP(ANS)
            IF (ISR.EQ.1) DOSREB(1)=ANS
            IF (IST.EQ.1) DOSTEB(NTMAX)=ANS
  150 CONTINUE
      ENDIF
C        GET DOSE FOR ZERO-THICKNESS SHELL = 2xSLAB FOR MIN Z (e.g.,1.0E-6)
      DOSTS(1)=2.0*DOSS0
      DOSTP(1)=2.0*DOSP0
      DOSTEB(1)=2.0*DOSEB0
C        GET DOSES FOR SLAB APPROX TO SHELL DOSE
```

```
      IF (JSMAX.LT.3) GO TO 180
         CALL SCOF (IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS)
         DO 170 N=1,NZMAX
         IF (T(N).GT.0.0) GO TO 160
         DOSZS(N)=DOSTS(1)
         GO TO 170
  160    TL=LOG(T(N))
         IF (TL.GT.ZL(IMAXS)) GO TO 170
         CALL BSPOL (TL,IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS,ANS)
         DOSZS(N)=2.0*EXP(ANS)
  170    CONTINUE
  180 IF (JPMAX.LT.3) GO TO 210
         CALL SCOF (IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP)
         DO 200 N=1,NZMAX
         IF (T(N).GT.0.0) GO TO 190
         DOSZP(N)=DOSTP(1)
         GO TO 200
  190    TL=LOG(T(N))
         IF (TL.GT.ZL(IMAXP)) GO TO 200
         CALL BSPOL (TL,IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP,ANS)
         DOSZP(N)=2.0*EXP(ANS)
  200    CONTINUE
  210 IF (JEMAX.LT.3) GO TO 240
         CALL SCOF (IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE)
         DO 230 N=1,NZMAX
         IF (T(N).GT.0.0) GO TO 220
         DOSZEB(N)=DOSTEB(1)
         GO TO 230
  220    TL=LOG(T(N))
         IF (TL.GT.ZL(IMAXE)) GO TO 230
         CALL BSPOL (TL,IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
         DOSZEB(N)=2.0*EXP(ANS)
  230    CONTINUE
C            SPHERE INTEGRATIONS
  240 IF (ISR.EQ.0) GO TO 370
      DO 360 N=2,NRMAX
      RMR=RADIUS-R(N)
      RPR=RADIUS+R(N)
      RPRL=LOG(RPR)
      IF (JSMAX.LT.3) GO TO 280
      DRMR=0.0
      DRPR=0.0
      IF (RMR.GT.ZMIN) GO TO 250
      DRMR=DOSS0
      RMR=ZMIN
      RMRL=LOG(RMR)
      GO TO 260
  250 RMRL=LOG(RMR)
      IF (RMRL.GT.ZL(IMAXS)) GO TO 260
      CALL BSPOL (RMRL,IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS,ANS)
      DRMR=EXP(ANS)
  260 IF (RPRL.GT.ZL(IMAXS)) GO TO 265
      CALL BSPOL (RPRL,IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS,ANS)
      DRPR=EXP(ANS)
  265 DS=(RPRL-RMRL)/FNPTS1
      DELTA=DS/3.0
      DO 270 NN=1,NPTS
      G(NN)=0.0
      SL=RMRL+FLOAT(NN-1)*DS
      IF (SL.GT.ZL(IMAXS)) GO TO 270
      CALL BSPOL (SL,IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS,ANS)
      G(NN)=EXP(SL+ANS)
  270 CONTINUE
      CALL INTEG (DELTA,G,NPTS,ANS)
      DOSRS(N)=(RADIUS*(DRMR-DRPR)+ANS)/R(N)
  280 IF (JPMAX.LT.3) GO TO 320
      DRMR=0.0
      DRPR=0.0
      IF (RMR.GT.ZMIN) GO TO 290
      DRMR=DOSPO
      RMR=ZMIN
      RMRL=LOG(RMR)
      GO TO 300
  290 RMRL=LOG(RMR)
      IF (RMRL.GT.ZL(IMAXP)) GO TO 300
      CALL BSPOL (RMRL,IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP,ANS)
      DRMR=EXP(ANS)
  300 IF (RPRL.GT.ZL(IMAXP)) GO TO 305
      CALL BSPOL (RPRL,IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP,ANS)
      DRPR=EXP(ANS)
  305 DS=(RPRL-RMRL)/FNPTS1
      DELTA=DS/3.0
      DO 310 NN=1,NPTS
      G(NN)=0.0
      SL=RMRL+FLOAT(NN-1)*DS
      IF (SL.GT.ZL(IMAXP)) GO TO 310
      CALL BSPOL (SL,IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP,ANS)
      G(NN)=EXP(SL+ANS)
  310 CONTINUE
      CALL INTEG (DELTA,G,NPTS,ANS)
      DOSRP(N)=(RADIUS*(DRMR-DRPR)+ANS)/R(N)
  320 IF (JEMAX.LT.3) GO TO 360
      DRMR=0.0
      DRPR=0.0
      IF (RMR.GT.ZMIN) GO TO 330
      DRMR=DOSEBO
      RMR=ZMIN
      RMRL=LOG(RMR)
      GO TO 340
  330 RMRL=LOG(RMR)
      IF (RMRL.GT.ZL(IMAXE)) GO TO 340
      CALL BSPOL (RMRL,IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
      DRMR=EXP(ANS)
  340 IF (RPRL.GT.ZL(IMAXE)) GO TO 345
      CALL BSPOL (RPRL,IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
      DRPR=EXP(ANS)
  345 DS=(RPRL-RMRL)/FNPTS1
      DELTA=DS/3.0
      DO 350 NN=1,NPTS
      G(NN)=0.0
      SL=RMRL+FLOAT(NN-1)*DS
      IF (SL.GT.ZL(IMAXE)) GO TO 350
      CALL BSPOL (SL,IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
      G(NN)=EXP(SL+ANS)
  350 CONTINUE
      CALL INTEG (DELTA,G,NPTS,ANS)
      DOSREB(N)=(RADIUS*(DRMR-DRPR)+ANS)/R(N)
  360 CONTINUE
C         SHELL INTEGRATIONS
  370 DO 460 N=2,NTMAX
      IF (T(N).EQ.RADIUS) GO TO 460
      H=SQRT(T(N)*(2.0*RADIUS-T(N)))
      TL=LOG(T(N))
      HL=LOG(H)
      IF (JSMAX.LT.3) GO TO 400
      DRMR=DOSZS(N)
      DH=0.0
      IF (HL.GT.ZL(IMAXS)) GO TO 380
      CALL BSPOL (HL,IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS,ANS)
      DH=EXP(ANS)
  380 DS=(HL-TL)/FNPTS1
      DELTA=DS/3.0
      DO 390 NN=1,NPTS
      G(NN)=0.0
      SL=TL+FLOAT(NN-1)*DS
      IF (SL.GT.ZL(IMAXS)) GO TO 390
      CALL BSPOL (SL,IMAXS,ZL,DOSS,BCOFS,CCOFS,DCOFS,ANS)
      G(NN)=EXP(SL+ANS)
```

```
390 CONTINUE
    CALL INTEG (DELTA,G,NPTS,ANS)
    DOSTS(N)=(RADIUS*DRMR-2.0*(H*DH-ANS))/RT(N)
400 IF (JPMAX.LT.3) GO TO 430
    DRMR=DOSZP(N)
    DH=0.0
    IF (HL.GT.ZL(IMAXP)) GO TO 410
    CALL BSPOL (HL,IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP,ANS)
    DH=EXP(ANS)
410 DS=(HL-TL)/FNPTS1
    DELTA=DS/3.0
    DO 420 NN=1,NPTS
    G(NN)=0.0
    SL=TL+FLOAT(NN-1)*DS
    IF (SL.GT.ZL(IMAXP)) GO TO 420
    CALL BSPOL (SL,IMAXP,ZL,DOSP,BCOFP,CCOFP,DCOFP,ANS)
    G(NN)=EXP(SL+ANS)
420 CONTINUE
    CALL INTEG (DELTA,G,NPTS,ANS)
    DOSTP(N)=(RADIUS*DRMR-2.0*(H*DH-ANS))/RT(N)
430 IF (JEMAX.LT.3) GO TO 400
    DRMR=DOSZEB(N)
    DH=0.0
    IF (HL.GT.ZL(IMAXE)) GO TO 440
    CALL BSPOL (HL,IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
    DH=EXP(ANS)
440 DS=(HL-TL)/FNPTS1
    DELTA=DS/3.0
    DO 450 NN=1,NPTS
    G(NN)=0.0
    SL=TL+FLOAT(NN-1)*DS
    IF (SL.GT.ZL(IMAXE)) GO TO 450
    CALL BSPOL (SL,IMAXE,ZL,DOSEB,BCOFE,CCOFE,DCOFE,ANS)
    G(NN)=EXP(SL+ANS)
450 CONTINUE
    CALL INTEG (DELTA,G,NPTS,ANS)
    DOSTEB(N)=(RADIUS*DRMR-2.0*(H*DH-ANS))/RT(N)
460 CONTINUE
    IF (ISR.EQ.0) GO TO 560
    WRITE (9,500)
500 FORMAT (/' DOSE AS A FUNCTION OF RADIUS r IN A SOLID SPHERE')
    WRITE (9,510) DET(IDET)
510 FORMAT (/' rads ',A)
    WRITE (9,520)
520 FORMAT (/'   r(g/cm2)     EL+BR   TRP PROT   SOL PROT EL+BR+TRP
   1TOTAL')
    WRITE (9,530)
530 FORMAT (' ')
    DO 550 N=1,NRMAX
    DOSEBP=DOSREB(N)+DOSRP(N)
    DOST=DOSEBP+DOSRS(N)
    WRITE (9,540) R(N),DOSREB(N),DOSRP(N),DOSRS(N),DOSEBP,DOST
540 FORMAT (2(0PF11.5,1P5E10.2))
550 CONTINUE
560 WRITE (9,600)
600 FORMAT (/' DOSE AT INNER SURFACE, AT RADIUS r, IN A SPHERICAL SHEL
   1L')
    WRITE (9,610)
610 FORMAT ( ' TWICE THE DOSE IS GIVEN ALSO AT EDGE OF PLANE SLAB OF S
   1AME THICKNESS t AS SHELL')
    WRITE (9,510) DET(IDET)
    WRITE (9,615)
615 FORMAT (/'               --------------- SPHERICAL SHELL -----------
   1-----           ---------------- 2 x PLANE SLAB ----------------
   2')
    WRITE (9,620)
620 FORMAT ( '   r(g/cm2)     EL+BR   TRP PROT   SOL PROT EL+BR+TRP
   1TOTAL   t(g/cm2)     EL+BR   TRP PROT   SOL PROT EL+BR+TRP     TOTAL
   2')
```

```
      WRITE (9,530)
      DO 650 N=1,NTMAX
      DOSEBP=DOSTEB(N)+DOSTP(N)
      DOST=DOSEBP+DOSTS(N)
      DUSEBP=DOSZEB(N)+DOSZP(N)
      DUST=DUSEBP+DOSZS(N)
      WRITE (9,540) RT(N),DOSTEB(N),DOSTP(N),DOSTS(N),DOSEBP,DOST,
     1               T(N),DOSZEB(N),DOSZP(N),DOSZS(N),DUSEBP,DUST
  650 CONTINUE
      IF (NZMAX.EQ.NTMAX) GO TO 33
      DO 670 N=NTMAX+1,NZMAX
      DUSEBP=DOSZEB(N)+DOSZP(N)
      DUST=DUSEBP+DOSZS(N)
      WRITE (9,660) T(N),DOSZEB(N),DOSZP(N),DOSZS(N),DUSEBP,DUST
  660 FORMAT (61X,0PF11.5,1P5E10.2)
  670 CONTINUE
      GO TO 33
  999 STOP
      END
      SUBROUTINE SCOF(N,X,Y,B,C,D)
C        REINSCH ALGORITHM, VIA MJB, 22 FEB 83
C        Y(S)=((D(J)*(X-X(J))+C(J))*(X-X(J))+B(J))*(X-X(J))+Y(J)
C             FOR X BETWEEN X(J) AND X(J+1)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(1),Y(1),B(1),C(1),D(1)
      N1=N-1
      S=0.0
      DO 10 J=1,N1
      D(J)=X(J+1)-X(J)
      R=(Y(J+1)-Y(J))/D(J)
      C(J)=R-S
   10 S=R
      S=0.0
      R=0.0
      C(1)=0.0
      C(N)=0.0
      DO 20 J=2,N1
      C(J)=C(J)+R*C(J-1)
      B(J)=(X(J-1)-X(J+1))*2.0-R*S
      S=D(J)
   20 R=S/B(J)
      DO 30 JR=N1,2,-1
   30 C(JR)=(D(JR)*C(JR+1)-C(JR))/B(JR)
      DO 40 J=1,N1
      S=D(J)
      R=C(J+1)-C(J)
      D(J)=R/S
      C(J)=3.0*C(J)
   40 B(J)=(Y(J+1)-Y(J))/S-(C(J)+R)*S
      RETURN
      END
      SUBROUTINE BSPOL(S,N,X,Y,B,C,D,T)
C        BINARY SEARCH, X ASCENDING OR DESCENDING
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(1),Y(1),B(1),C(1),D(1)
      IF (X(1).GT.X(N)) GO TO 10
      IDIR=0
      MLB=0
      MUB=N
      GO TO 20
   10 IDIR=1
      MLB=N
      MUB=0
   20 IDIR1=IDIR-1
      IF (S.GE.X(MUB+IDIR)) GO TO 60
      IF (S.LE.X(MLB-IDIR1)) GO TO 70
      ML=MLB
      MU=MUB
      GO TO 40
   30 IF (IABS(MU-ML).LE.1) GO TO 80
   40 MAV=(ML+MU)/2
      IF (S.LT.X(MAV)) GO TO 50
      ML=MAV
      GO TO 30
   50 MU=MAV
      GO TO 30
   60 MU=MUB+IDIR+IDIR1
      GO TO 90
   70 MU=MLB-IDIR-IDIR1
      GO TO 90
   80 MU=MU+IDIR1
   90 Q=S-X(MU)
      T=((D(MU)*Q+C(MU))*Q+B(MU))*Q+Y(MU)
      RETURN
      END
      SUBROUTINE INTEG (DELTA,G,N,RESULT)
C           INCLUDES N=1
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION G(8)
      NL1=N-1
      NL2=N-2
      IF (REAL (N) -2.0*REAL (N/2)) 100,100,10
   10  IF (N-1) 15,15,20
   15 SIGMA=0.0
      GO TO 70
   20 IF(N-3) 30,30,40
   30 SIGMA=G(1)+4.0*G(2)+G(3)
      GO TO 70
   40 SUM4=0.0
      DO 50 K=2,NL1,2
   50 SUM4=SUM4+G(K)
      SUM2=0.0
      DO 60 K=3,NL2,2
   60 SUM2=SUM2+G(K)
      SIGMA=G(1)+4.0*SUM4+2.0*SUM2+G(N)
   70 RESULT=DELTA*SIGMA
      RETURN
  100 IF(N-2)110,110,120
  110 SIGMA=1.5*(G(1)+G(2))
      GO TO 70
  120 IF(N-4)130,130,140
  130 SIGMA=1.125*(G(1)+3.0*G(2)+3.0*G(3)+G(4))
      GO TO 70
  140 IF(N-6)150,150,160
  150 SIGMA=G(1)+3.875*G(2)+2.625*G(3)+2.625*G(4)+3.875*G(5)+G(6)
      GO TO 70
  160 IF (N-8)170,170,180
  170 SIGMA=G(1)+3.875*G(2)+2.625*G(3)+2.625*G(4)+3.875*G(5)+2.0*G(6)
     1   +4.0*G(7)+G(8)
      GO TO 70
  180 SIG6=G(1)+3.875*G(2)+2.625*G(3)+2.625*G(4)+3.875*G(5)+G(6)
      SUM4=0.0
      DO 190 K=7,NL1,2
  190 SUM4=SUM4+G(K)
      SUM2=0.0
      DO 200 K=8,NL2,2
  200 SUM2=SUM2+G(K)
      SIGMA=SIG6+G(6)+4.0*SUM4+2.0*SUM2+G(N)
      GO TO 70
      END
```