Monte Carlo superposition calculations of resonance integrals in a reactor cell

Kirkegaard, Peter

Publication date:
1970

Document Version
Publisher's PDF, also known as Version of record

A. E. K. Risø

Risø - M - 1257

Risø - M - 1257

Title and author(s)

Monte Carlo Superposition Calculations of Resonance Integrals in a Reactor Cell

by

Peter Kirkegaard

Date July 1970

Department or group

Reactor Physics Department

Group's own registration number(s)

52 pages + 0 tables + 3 illustrations

Abstract

A description is given of a computer program, AEK P-598 MCSUP, which calculates the resonance integral for a single resonance in a square reactor lattice cell, containing a cylindrical fuel zone surrounded by a moderator zone. The fuel zone contains only one nuclide type, while the moderator may be composed by several nuclide types. Scattering interference and Doppler broadening is included.

MCSUP is a Monte Carlo code based on the superposition method devised by Spanier and Gelbard. The code is well-suited for calculations on isolated fuel rods. It is written in FORTRAN IV.

Copies to

Standard distribution

Abstract to

Available on request from the Library of the Danish Atomic Energy Commission (Atomenergikommissionens Bibliotek), Risø, Roskilde, Denmark.
Telephone: (03) 95 51 01, ext. 334, telex: 5072.

FI 23-304

CONTENTS

$$p = \exp\left[\quad \right]$$

Page

1. INTRODUCTION

In a forthcoming report, Mikkelsen[1)] describes a computer program RESAB which carries out a multigroup calculation of the resonance absorption and flux spectrum in a reactor cell, using collision probability technique. His calculation scheme relies on the validity of certain approximations making possible very efficient computing. In developing such an approximate model there arises a natural need of checking the model or parts of it against some exact calculation method; in view of the complexity of the problem Monte Carlo seems to be the only practical way of performing such reference calculations.

This paper describes a Monte Carlo FORTRAN IV computer program, AEK P-598 MCSUP, designed to calculate resonance integrals by the method of superposition. The superposition principle was developed and first put into practical operation by Spanier and Gelbard[2,3)]. MCSUP works fairly efficiently for most problems, and it is possible that it may be valuable also for other purposes than comparative.

2. STATEMENT OF THE PROBLEM

Consider an infinite-cylindrical reactor lattice cell with a cross section as shown in fig. 1. The heterogeneous resonance capture problem is idealized to a two-zone configuration, the cladding zone being neglected. The fuel cylinder radius is a, and the lattice pitch is denoted d. The cross sectional areas of the two zones are $A_1 = \pi a^2$ and $A_2 = d^2 - A_1$.

The fuel zone contains only one isotope (e. g. U-238), while the moderator may be composed of several nuclide types. Scattering in the fuel is assumed to proceed isotropicly in the L-system, scattering in the moderator isotropicly in the L-system or in the CM-system depending on the user's choice. No absorption is allowed in the moderator. The cell is assumed to be a part of an infinite [illegible] lattice, so that the [illegible] [illegible] self starts.

[illegible]

where $\langle \xi \sigma_s \rangle$ is the average slowing-down power of the cell per absorber atom [9].

The energy band taken into account is the interval $E_L < E < E_u$ containing the resonance energy E_o. We assume that the flux at the upper energy cut-off is an unperturbed 1/E-flux. The resonance absorption mechanism to be considered here concerns a single resolved resonance. The resonance cross sections are expressed by the single level Breit-Wigner formulas and include the scattering interference and Doppler broadening.

The scattering cross section of the moderator is assumed to be independent of the energy.

3. THE RESONANCE CROSS SECTIONS

The total microscopic cross section σ_t of the resonance absorber may be written as a sum of a resonance cross section and an energy independent potential scattering cross section:

$$\sigma_t = \sigma_r + \sigma_p \tag{2}$$

The resonance cross section itself is composed by an absorbing and a scattering component

$$\sigma_r = \sigma_a + \sigma_{rs} \tag{3}$$

The expression for σ_a and σ_{rs} read [1,4]

$$\sigma_a(E) = \sigma_o \frac{\Gamma_\gamma}{\Gamma} \sqrt{\pi} \cdot \eta \cdot u(E) \sqrt{\frac{E_o}{E}} \tag{4}$$

$$\sigma_{rs}(E) = \sigma_o \frac{\Gamma_n}{\Gamma} \sqrt{\pi} \cdot \eta \cdot u(E) + 2 \sqrt{\frac{\sigma_o \sigma_p \Gamma_n}{\Gamma}} \cdot \eta \, v(E) \tag{5}$$

with

$$\eta = \sqrt{\frac{A E_o}{16 kT \cdot E}} \tag{6}$$

$u + iv = w$ is the value of the complex error function corresponding to the argument $2\eta \frac{E-E_r}{\Gamma} + i\eta$. The complex error function may be defined by [11]

$$\mathrm{erf}(z) = \mathrm{erf}(x+iy) = \exp(-z^2) \left[1 + \frac{2i}{\sqrt{\pi}} \int_0^z \exp(t^2)\, dt \right] \tag{7}$$

Γ_n, Γ_γ and Γ are the neutron, radiative and total widths of the level, respectively. The parameter σ_o is the value of σ_r for $E = E_o$. It can be shown that [1]

$$\sigma_o = 2.603 \cdot 10^6 \left(\frac{1.00897 + A}{A} \right)^2 \cdot \frac{\Gamma_n}{\Gamma} \frac{1}{E_o} \tag{8}$$

The remaining parameters to be defined is A, the atomic weight of the absorber; k is Boltzmann's constant $k = 8.61 \cdot 10^{-5}$ eV/°K, and T is the absolute temperature.

u and v are connected with the well-known formulas $\phi(\theta,x)$ and $\chi(\theta,x)$ defined in Dresner [4], by the relations

$$\phi(\theta,x) = \sqrt{\pi}\, \eta\, u \tag{9}$$

$$\chi(\theta,x) = 2\sqrt{\pi}\, \eta\, v \tag{10}$$

where

$$x = \frac{E-E_o}{\left(\frac{\Gamma}{2}\right)} \quad \text{and} \quad \theta = 2\eta$$

In this work we are made of a FORTRAN IV version [13] of a code for computing the complex error function, as adapted by [illegible]

4. THE SUPERPOSITION PRINCIPLE

In this section we shall mainly follow Spanier and Gelbard[2] in giving a brief exposition of the superposition principle when applied to resonance absorption calculations.

The superposition principle states that the flux produced by the sum of two sources is the sum of the fluxes produced, separately, by each source. This follows immediately from the linearity of the transport equation. The idea is now that the problem to be solved ("problem α") is superimposed by another problem ("problem β") in such a way that the sum problem has a simple analytical solution. Hence, the Monte Carlo work is confined to problem β, and sometimes this method of attack leads to significant reduction of variance.

This basic approach has to be extended a little in order to yield an efficient model for resonance calculations. Consider again fig. 1 showing the fuel zone R_I and the moderator zone R_{II} with a reflecting boundary. As before we wish to solve problem α. The source $Q_S(E,r)$ in this problem is a slowing-in density in both zones from above E_u. Problem α has the solution (angular flux) $F_\alpha = F_\alpha(E,r,\omega)$, where E is energy, r position, and ω the unit vector of direction. Above E_u, F_α is assumed to be the unperturbed 1/E-flux. We normalize in such a way that $F_\alpha = \frac{1}{4\pi E}$ for $E \geq E_u$.

In the following it is essential to distinguish between resonance and potential scattering in the fuel. The macroscopic fuel cross section is therefore considered as composed by one absorption and two scattering terms:

$$\Sigma_{Tf}(E) = \Sigma_{af}(E) + \Sigma_{Sf}(E) = \Sigma_{af}(E) + \Sigma_{Srf}(E) + \Sigma_{Spf} = \Sigma_{rf}(E) + \Sigma_{Pf} \qquad (11)$$

We now define a problem β with the same geometry but with a source

$$Q_\beta(E) = \frac{\Sigma_{af}(E)}{E}$$

in the interval $E_L \leq E \leq E_u$ before the Monte Carlo [illegible] self starts.

in the fuel. No source is present in the moderator. Let the problem β

have the solution F_β.

The sum problem of α and β has the solution F_T. Now, it is not F_T itself that has a simple expression but that part of it, F_{Tu}, which has not yet undergone resonance scatterings. In fact,

$$F_{Tu}(E,r,\omega) = \frac{1}{4\pi E}, \qquad E_L \leq E \leq E_u \qquad (13)$$

To see this we write down the transport equations satisfied by F_{Tu} in R_I and R_{II}:

$$L F_{Tu}(E,r,\omega) = \int d\omega' \int_{E_L}^{E_u} dE' \Sigma_{Pf}(\omega\cdot\omega', E' \to E, r) F_{Tu}(E', r, \omega')$$

$$+ \frac{1}{4\pi} Q_S(E,r) + \frac{1}{4\pi} Q_f(E), \qquad r \in R_I$$

and

$$L F_{Tu}(E,r,\omega) = \int d\omega' \int_{E_L}^{E_u} dE' \Sigma_S(\omega\cdot\omega', E' \to E, r) F_{Tu}(E', r, \omega')$$

$$+ \frac{1}{4\pi} Q_S(E,r), \qquad r \in R_{II} \qquad (14)$$

$\Sigma_{Pf}(\omega\cdot\omega', E' \to E, r)$ is the differential cross section for potential scattering in the fuel. The operator L is defined by

$$L = \omega\cdot\nabla + \Sigma_{Tf}(E,r) \qquad (27)$$

Otherwise the notation is conventional.

(30) In (14) all scattering cross sections are energy independent. Solution of (13) and (14) [illegible]

$$F_a = F_{Tu} + F_{TS} - F_\beta = \frac{1}{4\pi E} + F_{TS} - F_\beta \tag{16}$$

It is now practical to split up F_β into four components, according to the history of the flux. Let

$F_\beta = F_\beta$ (unscattered before exit, unreturned)

$+ F_\beta$ (unscattered before exit, returned)

$+ F_\beta$ (resonance scattered before exit)

$+ F_\beta$ (potential scattered before exit), or

$$F_\beta = F_{\beta uu} + F_{\beta ur} + F_{\beta sr} + F_{\beta sp} \tag{17}$$

"Exit" means here absorption in or escape from the fuel. By (16) and (17),

$$F_a = \frac{1}{4\pi E} - F_{\beta uu} - F_{\beta ur} - F_{\beta sp} + F_{TS} - F_{\beta sr} \tag{18}$$

We are interested in calculating the resonance capture probability 1-p, or, equivalently, the mean absorption rate per ccm in the fuel

$$A_I = \frac{1}{A_1} \int_{E_L}^{E_u} dE \int_I dr\, \Sigma_{af}\, \phi_a(E,r) \ , \tag{19}$$

where ϕ, of course, stands for a scalar flux and I symbolizes the two-dimensional fuel region R_I. The connection between A_I and p-p is

$$1-p = \frac{A_I}{\langle \xi \Sigma_s \rangle} \tag{20}$$

$\langle \xi \Sigma_s \rangle$ is the slowing-down density past E_u corresponding to $\Sigma_s(E)$, $\Sigma_{Tf}(E)$, and is expressible in terms of $\langle \xi \sigma_s \rangle$ by

$$\langle \xi \Sigma_s \rangle = \frac{N A_1 \langle \xi \sigma_s \rangle}{A_1 + A_2} \tag{21}$$

where N is the number density of the absorber.

Defining

$$I_o' = \frac{1}{A_1} \int_{E_L}^{E_u} dE \int_I dr\, \Sigma_{af}(E) \left[\frac{1}{E} - \phi_{\beta uu}(E,r) \right] , \tag{22}$$

$$I_1' = \frac{1}{A_1} \int_{E_L}^{E_u} dE \int_I dr\, \Sigma_{af}(E)\, \phi_{\beta ur}(E,r) \ , \tag{23}$$

$$I_2 = \frac{1}{A_1} \int_{E_L}^{E_u} dE \int_I dr\, \Sigma_{af}(E)\, \phi_{\beta Sp}(E,r) \ , \tag{24}$$

$$J = \frac{1}{A_1} \int_{E_L}^{E_u} dE \int_I dr\, \Sigma_{af}(E) \left[\phi_{TS}(E,r) - \phi_{\beta Sr} \right] . \tag{25}$$

then

$$A_I = I_o' - I_1' - I_2 + J \tag{26}$$

It can be shown that

$$I_o' = \int_{E_L}^{E_u} \frac{dE}{E}\, \Sigma_{af}(E) \left[P_o(E) + \frac{\Sigma_{Pf}}{\Sigma_{Tf}(E)} (1 - P_o(E)) \right] \tag{27}$$

It can be shown that I_1' is the absorption rate produced by an outgoing source on the surface of the fuel

$$Q_{1P}(E,\omega) = \frac{\Sigma_{rf}(E)}{4\pi E} \frac{\mu(1-\exp(-l\cdot\Sigma_{Tf}(E)))}{\Sigma_{Tf}(E)} \tag{28}$$

Here, $\mu = \omega\cdot n > 0$, n being the outward unit normal at the surface point P, and l is the backward intersection path from P in the fuel (due to the scattering interference, $Q_{1P}(E,\omega)$ may become negative at some energies, this tending to increase variance somewhat). Integrating (28) over directions gives $Q_1'(E) = \int Q_{1P}(E,\omega)d\omega$, which is proportional to the density function for selecting starting energies. It follows that [2]

$$Q_1'(E) = \frac{\Sigma_{rf}(E)}{E}\cdot P_o(E) \tag{29}$$

The Monte Carlo calculation of I_1' includes such starters which have been absorbed on their first collision. This contribution will in our model be deleted from I_1' and included in the deterministic term I_o'. In other words, we write

$$I_o' - I_1' = I_o - I_1 \tag{30}$$

where now I_o is to be evaluated deterministicly and I_1 by Monte Carlo. One can show that [2]

$$I_o = \int_{E_L}^{E_u} \frac{dE}{E}\,\Sigma_{af}(E)\left[P^*(E) + \frac{\Sigma_{Pf}}{\Sigma_{Tf}(E)}\,(1-P^*(E))\right] \tag{31}$$

where P^* is the collision probability [2,6] from fuel to moderator. P^* is discussed in Appendix V, and the quadrature evaluation of I_o in Appendix VI. Going from I_1' to I_o, (29) is changed to

$$Q_1(E) = \frac{\Sigma_{rf}(E)}{E}\left[P_o(E) - \frac{\Sigma_{af}(E)}{\Sigma_{Tf}(E)}\,(P_o(E) - P^*(E))\right] \tag{32}$$

Hence, to sample I_1 we have to pick the starting energy from (32) and then sample an ω from (28). How these steps are carried out in detail is discussed in sec. 5. However, if a starter is absorbed on its first collision, it is rejected: a new ω is sampled from (28) without changing E.

The next term in (26), I_2, is generated by neutrons which are potential scattered before exit. The mean rate of such scatterings are

$$Q_2(E) = \frac{\Sigma_{rf}(E)}{E}\,(1-P_o(E))\,\frac{\Sigma_{Pf}}{\Sigma_{Tf}(E)} \tag{33}$$

From $|Q_2(E)|$ we pick a starting energy. Then a starting point P is selected uniformly in the fuel and an ω with isotropic distribution. If the starter leaves the fuel on its first flight it is rejected: without changing E we sample a new P and a new ω. Otherwise we regard its first collision as a potential scattering. Subsequent sampling proceed conventionally.

The last term to be computed by Monte Carlo is J (eq. (25)). We show that this term requires two Monte Carlos. Let us define

$$F_{\Delta S} = F_{TS} - F_{\beta Sr} \tag{34}$$

By writing down the transport equations for F_{TS} and $F_{\beta Sr}$ one sees that $F_{\Delta S}$ satisfies the equations

$$L F_{\Delta S}(E,r,\omega) = \int d\omega' \int_{E_L}^{E_u} dE' \, \Sigma_S(\omega\cdot\omega', E' \to E, r) \, F_{\Delta S}(E', r, \omega'),$$

$$r \in R_{II} \qquad (35)$$

Hence, $F_{\Delta S}$ is generated by resonance scattering in R_I in the flux F_{Tu}-$F_{\beta uu}$. One can show that this latter quantity itself satisfies a transport equation,

$$L\left[F_{Tu}(E,r,\omega) - F_{\beta uu}(E,r,\omega)\right] = \frac{\Sigma_{Pf}}{4\pi E}, \qquad r \in R_I \qquad (36)$$

and

$$F_{Tu}(E,r_P,\omega) - F_{\beta uu}(E,r_P,\omega) = \frac{1}{4\pi E} \qquad (37)$$

where r_P is a point of the fuel-moderator interface, and ω is an inward direction ($\omega\cdot n < 0$).

Eqs. (36) and (37) reflect the fact that F_{Tu}-$F_{\beta uu}$ is produced by the joint action of a uniform isotropic volume source $\Sigma_{Pf}/4\pi E$ and an inward-directed surface source with strength $\omega\, n/4\pi E$.

This leads to the following procedure for the calculation of the corresponding components, I_3 and I_4, of J. In the I_3-case we write

$$Q_3(E) = \frac{\Sigma_{Pf}}{E}\,(1 - P_o(E))\,\frac{\Sigma_{Srf}(E)}{\Sigma_{Tf}(E)}, \qquad (38)$$

select an energy from $|Q_3(E)|$ and proceed just as in the I_2-case (cf. eq. (33) et seq.), but here we force the starter to resonance-scatter in the fuel.

In the I_4-case we write

$$Q_4(E) = [illegible] \frac{P_o(E)}{E} \qquad (39)$$

select an energy from $|Q_4(E)|$ and a starting point from a distribution uniform over the [illegible]. The starting direction has to be inwards, but

otherwise this sampling is quite analogous to the I_1-case. The first collision is forced to be a resonance scattering in R_I. Hereafter, the sampling proceeds conventionally.

To recapitulate, we have expressed the rate of absorption A_f as a sum of five terms

$$A_f = I_o - I_1 - I_2 + I_3 + I_4 \qquad (40)$$

where I_o can be evaluated by quadrature, but I_1, I_2, I_3, I_4 each requires one Monte Carlo. In the following section we discuss some problems in connection with the Monte Carlo sampling.

5. MONTE CARLO SAMPLING IN MCSUP

5.1. Sampling of Initial Energy

From the formulas (32), (33), (38), (39), it is seen that the Monte Carlo for each of the four terms I_1, I_2, I_3, I_4, requires an initial energy sampling from a density function proportional to some known function Q(E). How this is done is shown schematically in fig. 2. Q(E) is approximated by a step function corresponding to a division of the energy interval $E_L \le E \le E_u$ in 1000 equal subintervals (only 8 shown in fig. 2) with step heights equal to Q(E) taken in the subinterval midpoints. This matches the discrete cross section evaluation mentioned in sec. 3. The step function is normalized to a density function and is next converted to a distribution function F(E) of the polygon type. After selection of a random number ζ, E is given by $F^{-1}(\zeta)$.

To increase speed we prefer not to use ζ itself, but its nearest discrete representation ζ_i corresponding to a division of the interval $0 < \zeta < 1$ in 1000 equal subintervals (only 8 shown in fig. 2). The values $E_i = F^{-1}(\zeta_i)$ are calculated and stored before the Monte Carlo is executed. Of course, this procedure involves a sacrifice of some precision.

As pointed out in ref. 2 the technique of antithetic variates [illegible] may be applied here in a very simple way to obtain variance [illegible] history [illegible] of E, the second history [illegible] [illegible] this work. This estimator was first introduced by [illegible]

5.2. Surface Source Sampling

Having selected the initial energy one turns to sample the source position and direction. Starters of types 2 and 3 (cf. sec. 4) have an isotropic and uniform volume distribution which is easily sampled, while the types 1 and 4 have an outgoing, resp. ingoing surface source distribution, which is considered in more detail now. As the surface positions are selected from a uniform distribution, we need only discuss the direction sampling. Because the sampling is quite similar in the two cases, we consider only the outgoing source (type 1).

The relevant sampling formula is (28), which now, when position and energy is known, reduces to

$$Q(\omega) = \mu \left[1 - \exp(-l(\omega)\cdot \Sigma)\right] \quad , \tag{41}$$

apart from a constant factor. As before, $\mu = \omega \cdot n$, and $l(\omega)$ is the backwards intersection path in the fuel.

A straightforward way would be to sample ω from either an isotropic or a cosine distribution and adjust the weight of the starter according to (41). Fluctuating weights have, however, a deleterious influence on variance, unless the fluctuations reflect the importance of the events [2].

A better alternative seems here to be a cosine sampling combined with rejection [2, 19]. As we then sample from a distribution proportional to μ, the quantity in square brackets in (41) governs the rejection. Hence we pick a random number ζ and accepts or rejects the starter according to the events $1 - \exp(-l(\omega)\cdot \Sigma) \gtrless \zeta$. In case of rejection a new ω-sampling is carried out. The efficiency for this rejection is supposed to be high, because the rod for most initial energies is fairly black [2].

5.3. Coveyou Sampling from Truncated Exponential

As mentioned in sec. 4, starters of type 4 are forced to resonance-scatter in the fuel. This leads naturally to sampling from a truncated exponential with the density function

$$f(x) = \frac{\Sigma e^{-\Sigma x}}{1 - e^{-\Sigma l}} \qquad 0 \le x \le l \tag{42}$$

where l is the path from the source point to the rod intersection point along the flight. With a random number ζ, the standard sampling formula corresponding to (42) becomes

$$x = -\frac{1}{\Sigma} \log \left[1 - \zeta(1 - \exp(-\Sigma l))\right] \tag{43}$$

involving both a log- and an exp-calculation. Here, an alternative sampling formula due to Coveyou is used [2, 16, 20]:

$$x = \frac{1}{\Sigma} \ \mathrm{UDR}\left[\frac{-\log \zeta}{\Sigma l}\right] \tag{44}$$

UDR stands for the undivided remainder, e.g. $\mathrm{UDR}\left(\frac{13.83}{6}\right) = 1.83$.

5.4. Conventional Transport Sampling

Apart from the special methods discussed above and the choice of estimator (see 5.5), the sampling of the neutron transport and energy degradation processes elapses quite conventionally and will not be discussed here; details will be found elsewhere [2, 14, 16]. However, it should be quite easy to follow the various steps on the computer code print-out reproduced in Appendix III; it is hoped that the rather numerous comments will facilitate this work.

5.5. Choice of Estimator

Before the program reached its final form, some experimentation was done to find an efficient estimator for the resonance absorption. Three estimators were tried.

1) The Wasow estimator [2] with forbidden absorption and corresponding weight reductions at collisions. History termination only by energy degradation.

2) The analog estimator [2].

3) The modified track length estimator.

From our calculations it appears that the efficiency

track length estimator refers to an analog random walk process. Scoring takes place every time a neutron collides in, or escape from the fuel, after the following rule:

a) When a neutron starts at the fuel point P (a source or scattering point, or a point on the fuel boundary) and thereafter escapes uncollided at E or scatters at S, then the scoring is $\Sigma_a \cdot d$ where d is the track length PE or PS.

b) When a neutron starts at the fuel point P and is absorbed in the fuel at A (whereby the history is terminated), then the scoring is

$$\frac{\Sigma_a}{\Sigma_t} \cdot \frac{1-\exp(-\Sigma_t d)\ \{\Sigma_t d+1\}}{1-\exp(-\Sigma_t d)} \qquad (45)$$

where d = PE, E denoting the hypothetic escape point if no collision had occurred. (45) is constructed by averaging the estimator $\Sigma_a \cdot$ PA over all possible A[2].

5.6. Statistical Processing

In order to carry out a statistical analysis of the Monte Carlo data, histories are collected in groups of 500 each. How many groups that are processed depends on the user's choice of computing time (see 6.1). Group averages are calculated, and from these a final average and a sample variance is found (a confidence analysis based on Student's distribution is not carried out). Finally, the "statistical quality" Q is calculated (see 6.2).

The statistical procedure has to be modified a little by the introduction of the optimization device discussed in Appendix VIII.

6. CALCULATION WITH THE MCSUP PROGRAM

6.1. Preparation of Input Data

Input data for the FORTRAN IV program P-598 MCSUP is prepared according to the data sheet given in Appendix I.

This input contains data for a single problem concerning a cell with

pure hydrogen as the moderator, and anisotropic moderator scattering is taken into account. Format and type of data items are derived from the sheet. The first item is the problem no; a negative problem no. stops the computations. The two integers in the next line are: random initial number (any nine-digit integer will do) and the number m of different nuclide types in the moderator (here m = 1). The next two lines in the sheet contain real numbers; they are (in the notation previously used): E_0, E_u, E_L, σ_p, Γ_n, Γ_γ, A, T, a, d, number density of absorber $\times 10^{24}$, maximum time t_{max}. All energies are in eV. t_{max} is given in minutes and refers to the time spent in the Monte Carlo phase of the problem; to be safe, the user should not choose a t_{max} greater than $t_{job} - 2$ where t_{job} is the maximum time in minutes stated on the job request card (this rule is relevant for calculations on the IBM 7094 at NEUCC in Denmark; modifications may be necessary for other computing centers). Hereafter come m data lines, one for each nuclide type in the moderator. Each line contains three real numbers and one integer: atomic weight, cross section (barn), number density $\times 10^{24}$, and the scattering mode I which is 0 for isotropic scattering on that nuclide and 1 for anisotropic scattering.

As here only a single job is to be executed, one has written -1 in the last line.

6.2. Interpretation of Output

The result for the sample problem given by the input data in Appendix I appears in Appendix II.

The output interpretation should follow immediately from the text given. It remains only to define the "statistical quality" Q:

$$Q = \frac{Res^2}{\sigma_{Res}^2 \cdot t} \qquad (46)$$

where Res is the output quantity and σ_{Res} its standard deviation, t is the computer time consumed (sec.). Q is substantially independent of t for a fixed computer, and is a measure of the efficiency of the Monte Carlo sampling.

6.3. Solutions of Various Problems by MCSUP

In the following is collected results for MCSUP calculations for a number of problems. Actually, many of these results have been produced by earlier (and less efficient) versions of the code. Due to these variable conditions, computing times and qualities are not stated.

The computations were carried out on the IBM 7094 at NEUCC, Lyngby, Denmark.

Three different resonances with the following data were treated:

Resonance no.	E_o	E_u	E_L	σ_p	Γ_n	Γ_γ
1	6.7	14	3	9.5	0.00152	0.0246
2	36.7	42	32	10.64	0.03114	0.02633
3	190.34	200	180	10.64	0.15038	0.02321

Five different moderators with the following data were treated:

Moderator	σ_1	N_1	σ_2	N_2
H	20.4	0.066633	-	-
D	3.4	0.068633	-	-
C	4.65	0.080307	-	-
H_2O	20.4	0.066633	3.8	0.033317
D_2O	3.4	0.066253	3.8	0.033126

(for compound moderators index 1 refers to the light component).

Six configurations with the following data were treated:

Configuration no.	a	d
1	0.2	$0.3\sqrt{\pi}$
2	1	2
3	1	$1.5\sqrt{\pi}$
4	1	$100\sqrt{\pi}$
5	1.2	$120\sqrt{\pi}$
6	1.7	$170\sqrt{\pi}$

Configurations 4, 5, 6 are, virtually, isolated fuel rods. In all the calculations to be presented, the absorber was assumed to be U-238 (N = 0.046974, T = 300°K).

The MCSUP results are given in the table below. Concerning the scattering mode I (cf. 6.1), this refers for compound moderators to the light components: isotropic scattering is throughout assumed for the heavy component.

Resonance no.	Moderator	I	Config. no.	Resonance integral RI	Standard dev. σ_{RI} *	Escape probability 1-p	Standard dev. σ_{1-p} *
1	H	1	1	4.512	0.014	0.11704	0.00036
1	H	0	2	1.806	0.008	0.20245	0.00094
1	H	0	3	2.7327	0.0022	0.075379	0.000058
2	H	0	3	0.9841	0.0084	0.02683	0.00023
1	H	0	5	2.740	0.009	0.9471_{10-5}	0.0031_{10-5}
3	H	0	3	0.1563	0.0016	0.0043018	0.0000453
1	C	0	5	2.638	0.012	0.21015_{10-3}	0.00097_{10-3}
1	C	0	4	2.9327	0.0113	0.23385_{10-3}	0.00090_{10-3}
1	H	0	4	3.0319	0.0040	0.10478_{10-4}	0.00014_{10-4}
1	D	0	4	3.0609	0.0083	0.87507_{10-4}	0.00238_{10-4}
1	H	1	4	3.0793	0.0044	0.10642_{10-4}	0.00015_{10-4}
2	H	0	4	1.0471	0.0032	0.36188_{10-5}	0.00111_{10-5}
2	H	1	4	1.0557	0.0030	0.36485_{10-5}	0.00105_{10-5}
3	H	0	4	0.15835	0.00168	0.54723_{10-6}	0.00581_{10-6}
1	D_2O	0	4	3.0047	0.0043	0.79084_{10-4}	0.00114_{10-4}
1	H	1	3	2.7639	0.0055	0.076240	0.000151
1	H	0	6	2.2492	0.0050	0.77640_{10-5}	0.00174_{10-7}
1	H	1	6	2.2887	0.0045	0.79003_{10-5}	0.00157_{10-7}
1	H_2O	1	4	3.0575	0.0042	0.10450_{10-4}	0.00015_{10-4}
1	H_2O	0	4	3.0174	0.0045	0.10319_{10-4}	0.00015_{10-4}

* Later experiences indicated that these estimates of σ were low, in many cases by 50% or more. This underestimation was traced back to the history averaging in groups (cf. 5.6). The earlier MCSUP versions had a group size of 100; due to fluctuations induced by the negative starters this is too small for the group averages to settle down to the normal distribution as the statistical analysis requires. The present value of 500 has eliminated or greatly reduced the underestimation of σ.

7. POSSIBLE GENERALISATIONS

The MCSUP program in its present form has a number of limitations, arising from the fact that the code hitherto has served purely comparative purposes. It should, however, be possible to remove some of these limitations fairly easily, if the need occurs.

First, the lattice cell was assumed to be square, but implementation of the hexagonal geometry requires only moderate changes in the main program and in the collision probability subroutine PINCA (cf. Appendix V). The same is true for the inclusion of a cladding zone around the fuel (the RESQ-2 program [3] works in hexagonal geometry with cladding).

Next, the resonance absorption has been confined to single resonances for a single absorbing nuclide type; one could perhaps desire an extension to a mixture of absorbers with quite general resonance cross sections (probably to be read in from tape). Although possible, this extension would call for fairly much editorial programming.

Inclusion of non-absorbing materials in the fuel region would, in contrast, be very easy.

Concerning the sampling method, a useful extension would be to introduce some sort of correlated sampling, which could improve the possibility of studying differential effects (e.g. the difference between isotropic and anisotropic moderator scattering). A simple way to apply correlated sampling is to force the corresponding histories in the two calculations to begin with the same random number [2]. In any case, correlated sampling requires a considerable extension of the statistical analysis.

8. SUMMARY AND CONCLUSIONS

The present report has discussed a Monte Carlo method, based on the superposition principle, devised to calculate the resonance absorption in a reactor lattice cell.

The relevant features of the corresponding FORTRAN IV computer program MCSUP (AEK P-598) have been treated, including instructions for the use of the code.

The results of a serial of problems solved by MCSUP have been presented.

The experience from the work has been that the calculation efficiency proved to be quite high for most problems.

As MCSUP was intended as a reference program, it is not able to solve general resonance problems; as indicated in sec. 7, however, this restriction could probably be removed by extension of the model in various ways.

9. ACKNOWLEDGEMENT

The author thanks Jørn Mikkelsen who gave the incitation to much of the work. Further, Mikkelsen has reviewed part of the manuscript and has detected some errors in an earlier version of the computer code.

10. REFERENCES

1) J. Mikkelsen, Thesis work to be published as a Risö Report in 1970.

2) J. Spanier and E. M. Gelbard, Monte Carlo Principles and Neutron Transport Problems (Addison-Wesley, 1969).

3) B. L. Anderson, E. M. Gelbard, and J. Spanier, RESQ-2 - A Combined Analytic Monte Carlo Calculation of Resonance Absorption Based on Superposition. Bettis Atomic Power Laboratory, WAPD-TM-665 (March, 1967).

4) L. Dresner, Resonance Absorptions in Nuclear Reactors (Pergamon Press, New York, 1960).

5) E. M. Gelbard, L. A. Ondis II, and J. Spanier, A New Class of Monte Carlo Estimators. J. SIAM Appl. Math. 14, 697 (1966).

6) L. Carlvik, A method for calculating collision probabilities in general cylindrical geometry and applications to flux distributions and Dancoff factors. U. N. Int. Conf. on the Peaceful Uses of Atomic Energy, Geneva, 3. 1964, Vol. 2. Geneva 1965, p. 225.

7) G. N. Lance, Numerical Methods for High Speed Computers (Iliffe and Sons Ltd., London, 1960).

8) J. M. Hammersley and D. C. Handscomb, Monte Carlo Methods (Methuen, London, 1964).

9) S. Glasstone and M. C. Edlund, The Elements of Nuclear Reactor Theory (Van Nostrand, 1952).

10) A. M. Weinberg and E. P. Wigner, The Physical Theory of Neutron Chain Reactors (The University of Chicago, Chicago, 1958).

11) V. N. Faddeyeva and N. M. Terent'ev, Tables of values of the function

$$w(Z) = e^{-Z^2}\left(1 + \frac{2i}{\sqrt{\pi}} \int_0^Z e^{t^2}\, dt\right)$$

for complex argument (Pergamon Press, Oxford, 1961).

12) H. Neltrup, Error Integral with Complex Argument erfcpl(x, y, u, v). Risö Computer Library, SA-84/1, 1965.

13) P. Kirkegaard, FORTRAN IV Subroutine for Error Integral with Complex Argument ERFCPL(X, Y, U, V). Risö Computer Library, SF-84, 1970.

14) E. D. Cashwell and C. J. Everett, A Practical Manual of the Monte Carlo Method for Random Walk Problems. Los Alamos Scientific Laboratory, LA-2120 (1957).

15) K. M. Case, F. de Hoffmann, and G. Placzek, Introduction to the Theory of Neutron Diffusion, Vol. I (Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1953).

16) P. Kirkegaard, Nuclear Particle Transport with Emphasis on Monte-Carlo and Shielding Calculations. Danish Atomic Energy Commission, Risö Report No. 136 (1966).

17) D. B. MacMillan, Comparison of Statistical Estimators for Neutron Monte Carlo Calculations. Nucl. Sci. Eng. 26, 366 (1966).

18) M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables (National Bureau of Standards, Applied Mathematics Series 55, 1964).

19) H. Kahn, Applications of Monte Carlo. Rand. Corp., AECU-3259 (1954).

20) M. Leimdörfer, On the Transformation of the Transport Equation for Solving Deep Penetration Problems by the Monte Carlo Method. Trans. Chalmers Univ. Technol., Gothenberg, 286 (1964).

21) I. Carlvik and B. Pershagen, The Dancoff correction in various geometries. AE-16 (Stockholm, 1959).

22) L. W. Nordheim and G. Kuncir, A Program of Research and Calculations of Resonance Absorption. GA-2527 (1961).

23) D. W. Hutchinson, A New Uniform Pseudorandom Number Generator. Com. ACM 9, 432 (1966).

24) I. Carlvik, Dancoff Correction in Square and Hexagonal Lattices. AE-257 (Stockholm, 1966).

11. APPENDIX I: MCSUP Data Sheet

IBM

IBM Data Centre
Punching Instructions

Name: Peter Hindsgaard		APPLICATION
Date: 24/6-1970	Page of	Sample Input Data for P-598 MCSUP

1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80
1							
555555555	1						
	14.	3.	91.5	0.001152	0.0246	238.	300.
	2.6587	0.046974	8.				
	20.4	0.06633	1				
-1							

D 2023

12. APPENDIX II: RESULT FOR SAMPLE PROBLEM

```
[illegible] CALCULATION AFTER SPANIERS SECOND METHOD

[illegible] NO 1

RANDOM INITIAL INTEGER= 555555555

NUMBER OF MODERATOR NUCLIDE TYPES= 1

RESONANCE ENERGY= 0.67000E 01   UPPER ENERGY LIMIT= 0.14000E 02   LOWER ENERGY LIMIT= 0.30000E 01

[illegible] SCATTERING CROSSSECTION OF ABSORBER= 0.95000E 01

GAMMA-N= 0.15200E-02   GAMMA-GAMMA= 0.24600E-01

ATOMIC WEIGHT OF ABSORBER= 0.23800E 03

TEMPERATURE OF MODERATOR= 0.30000E 03

RADIUS= 0.10000E 01   PITCH= 0.26587E 01

CONCENTRATION OF ABSORBER NUCLEI= 0.46974E-01

[illegible] TIME= 13.00 MINUTES

[illegible] DATA
[illegible]      CROSSSECTION    NUMBER DENSITY    SCATTERING MODE
[illegible]      0.20400E 02     0.66633E-01             1

[illegible] OF THE FIVE TERMS TO RESONANCE INTEGRAL
[illegible] TERM=                         0.39057E 01
[illegible] SURFACE SOURCE TERM=          0.12803E 00   DEVIATION= 0.3168[illegible]E-02
[illegible] SOURCE TERM=                  0.11266E 01   DEVIATION= 0.43621E-02
[illegible] SOURCE TERM=                  0.57935E-01   DEVIATION= 0.31629E-02
[illegible] SURFACE SOURCE TERM=          0.[illegible]1751E-01   DEVIATION= 0.33945E-02

[illegible] INTEGRAL=                     0.[illegible]7600E 01   DEVIATION= 0.71130E-02

[illegible] PROBABILITY=                  0.76134E-01   DEVIATION= 0.19621E-03

[illegible] QUALITY= 190
```

13. APPENDIX III: COMPUTER CODE PRINT-OUT

```
      COMMON/COM2/ETA,ER,A1,A2,A3,A4,SPA,CONCA
      COMMON/COM3/U,V,W,SIN2,IX,TCOS(1000),TSIN(1000),AM(5),ASQPLI(5),J
      COMMON/COM4/PI,DIA,SHM,RK,PINVS(46)
      COMMON/COM5/ROI
      DIMENSION PN(1000),SIGT(1000),SIGSR(1000),SAST(1000),TLOG(1000),
     1DEQ(1000,4),EQ(1000,4),RINT(5),PRES(5),DPRS(4),TDEL(4),ISMAX(2,4)
     2,SHMI(5),SHMA(5),CONCM(5),RKSIM(5),ALM(5),ONALM(5),RAND(5)
     3,IAMIS(5)
      LOGICAL LABEL
C SPECIFICATION OF I/O UNITS
      NIN=5
      NOUT=6
  200 FORMAT(8I10)
  201 FORMAT(8E10.5)
  199 FORMAT(3E10.5,I10)
      WRITE(NOUT,101)
  101 FORMAT(1HO,54HSUPERPOSITION CALCULATION AFTER SPANIERS SECOND METH
     1OD)
   66 READ(NIN,200)IPNO
      IF(IPNO.LT.0)STOP
      NMAX=600
      FN=NMAX
      READ(NIN,200)IX,IMOD
      READ(NIN,201)ER,EU,EL,SPA,GN,GG,AA,T,RC1,RK2,CONCA,TIDMAX
      DO 63 I=1,IMOD
   63 READ(NIN,199)AM(I),SHMI(I),CONCM(I),IAMIS(I)
      WRITE(NOUT,103)IPNO
  103 FORMAT(1H1,10HPROBLEM NO,I3)
      WRITE(NOUT,104)IX
  104 FORMAT(1HO,23HRANDOM INITIAL INTEGER=,I10)
      WRITE(NOUT,105)IMOD
  105 FORMAT(1HO,34HNUMBER OF MODERATOR NUCLIDE TYPES=,I2)
      WRITE(NOUT,106)ER,EU,EL
  106 FORMAT(1HO,17HRESONANCE ENERGY=,E12.5,3X,19HUPPER ENERGY LIMIT=,
     1E12.5,3X,19HLOWER ENERGY LIMIT=,E12.5)
      WRITE(NOUT,107)SPA
  107 FORMAT(1HO,46HPOTENTIAL SCATTERING CROSSSECTION OF ABSORBER=,E12.5
     1)
      WRITE(NOUT,108)GN,GG
  108 FORMAT(1HO,8HGAMMA-N=,E12.5,3X,12HGAMMA-GAMMA=,E12.5)
      WRITE(NOUT,109)AA
  109 FORMAT(1HO,26HATOMIC WEIGHT OF ABSORBER=,E12.5)
      WRITE(NOUT,110)T
  110 FORMAT(1HO,22HTEMPERATURE OF MODERATOR=,E12.5)
      WRITE(NOUT,111)RC1,RK2
  111 FORMAT(1HO,7HRADIUS=,E12.5,3X,6HPITCH=,E12.5)
      WRITE(NOUT,112)CONCA
  112 FORMAT(1HO,23HCONCENTRATION OF ABSORBER NUCLEI=,E12.5)
```



```
      WRITE(NOUT,113)TIDMAX
  113 FORMAT(1HO,13HMAXIMUM TIME=,F6.2,2X,7HMINUTES)
      WRITE(NOUT,114)
  114 FORMAT(1HO,14HMODERATOR DATA)
      WRITE(NOUT,115)
  115 FORMAT(1H ,13HATOMIC WEIGHT,4X,12HCROSSSECTION,4X,14HNUMBER DENSIT
     1Y,4X,15HSCATTERING MODE)
      DO 64 I=1,IMOD
   64 WRITE(NOUT,116)AM(I),SHMI(I),CONCM(I),IAMIS(I)
  116 FORMAT(1H ,E12.5,5X,E12.5,4X,E12.5,11X,I2)
      RK=RK2/2.
      PI=3.141592E5
      TOPI=2.*PI
      DO 250 I1=1,1000
      R=FLOAT(I1)/1000.-0.0005
      ARGU=TOPI*R-PI
      TCOS(I1)=COS(ARGU)
      TSIN(I1)=SIN(ARGU)
  250 TLOG(I1)=ALOG(R)
      G=GN+GG
      ETA=G*SQRT(AA/0.0013776/T/ER)
      S0=2.6E3E6*((AA+1.0C8S7)/AA)**2*GN/G/ER
      H1=S0/G*SQRT(PI)*ETA
      A1=H1*GG
      A2=H1*GN
      A3=2.*SQRT(S0*SPA*GN*PI/G)*ETA
      DELE=(EU-EL)/1000.
      A4=2.*ETA/G
      DO 50 I=1,1000
      H6=I
      E=EL+DELE*(H6-0.5)
      H7=A4*(E-ER)
      CALL ERFCPL(H7,ETA,U,V)
      SA=A1*U*SQRT(ER/E)
      SSR=A2*U+A3*V
      SS=SSR+SPA
      ST=SA+SS
      SAST(I)=SA/ST
      SIGSR(I)=SSR*CONCA
   50 SIGT(I)=ST*CONCA
      SPAN=SPA*CONCA
      DIA=2.*ROI
      R1SQ=ROI**2
      AREA1=PI*R1SQ
      AREA2=RK2**2-AREA1
      CPAL1=4.*AA/(AA+1.)**2
      RKSII=6./(3.*AA+2.)
      SOLR=RKSII*SPAN*AREA1
      SHM=0.
      DO 66 I=1,IMOD
      ASQPLI(I)=AM(I)**2+1.
      SHMA(I)=CONCM(I)*SHMI(I)
      SHM=SHM+SHMA(I)
      CPALP(I)=4.*AM(I)/(AM(I)+1.)**2
      ALM(I)=1.-ONALM(I)
      RKSIM(I)=1.
```

```
      IF(AM(I).GT.1.5)RKSIN(I)=1.+ALN(I)/CHALP(I)*ALOG(ALN(I))
   66 SOUR=SOUR+RKSIP(I)*SHMA(I)*AREA2
      CALL PINCA
      IF(IMOD.EQ.1)GOTO67
      RAND(1)=SHMA(1)/SHM
      DO 68 I=2,IMOD
   68 RAND(I)=RAND(I-1)+SHMA(I)/SHM
   67 COEF=DELE/CONCA
      DO 51 I=1,1000
      H1=I
      E=EL+DELE*(H1-0.5)
      STF=SIGT(I)
      OPT=STF*ROI
      PZERO=PO(OPT)
      PSTAR=PS(OPT)
      ASRF=ABS(STF-SPAM)
      DEG(I,1)=ASRF*(PZERO-SAST(I)*(PZERO-PSTAR))/E
      H2=SPAM*(1.-PZERO)/STF/E
      DEG(I,2)=ASRF*H2
      H3=ABS(SIGSR(I))
      DEG(I,3)=H3*H2
   51 DEG(I,4)=H3*PZERO/E
      RINT(6)=SELIN(EL,EU,128,10(C.)/DELE
      PRES(5)=COEF*RINT(6)
      DO 52 L=1,4
      RINT(L)=0.
      DO 52 I=1,1000
   52 RINT(L)=RINT(L)+DEG(I,L)
      DO 53 L=1,4
      J=1
      H4=0.
      H5=0.0005
      SUM=0.
      DO 54 I=1,1000
      H2=DEG(I,L)
      H3=H4
      SUM=SUM+H2
      H4=SUM/RINT(L)
   55 IF(H4.LT.H5)GOTO54
      H1=I-1
      EG(J,L)=EL+DELE*((H5-H3)/(H4-H3)+H1)
      J=J+1
      H5=FLOAT(J)/1000.-0.0005
      GOTO55
   54 CONTINUE
   53 CONTINUE
      TO=0.
      J=1
      DO 302 M=1,2
      TMAX=TIMAX/16.
      DO 350 L=1,4
      IF(M.EQ.2)TMAX=TDEL(L)
      II=0
      CALL TIME(ISTART)
   81 REI=0.
      N=0
```



```
      IEVEN=1
   62 N=N+1
      IF(N.EQ.NMAX+1)GOTO80
C INITIAL ENERGY
      LABEL=.FALSE.
      IEVEN=-IEVEN
      IF(IEVEN)60,60,61
   60 CALL RANDL(IX,R)
      RODD=R
      GOTO62
   61 R=1.-RODD
   62 I1=1000.*R+1.
      E=EG(I1,L)
      I=(E-EL)/DELE+1.
      GOTO(206,202,202,203),L
C STARTER OF TYPE 1
C INITIAL DIRECTION RELATIVE TO SURFACE NORMAL
  206 WG=SIGN(1.,SIGT(I)-SPAM)
      LABEL=.TRUE.
  207 CALL RANDL(IX,R)
      U1=SQRT(R)
      H1=SQRT(1.-R)
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      W=H1*TSIN(I1)
      SIN2=1.-W**2
C REJECTION CALCULATION
      H2=DEXP(DIA*U1/SIN2*SIGT(I))
      CALL RANDL(IX,R)
      IF(H2.LE.R)GOTO207
      V1=H1*TCOS(I1)
C INITIAL SURFACE POSITION AND CORRECTION OF DIRECTION
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      COSV=TCOS(I1)
      SINV=TSIN(I1)
      X=RO1*COSV
      Y=RC1*SINV
      U=COSV*U1-SINV*V1
      V=SINV*U1+COSV*V1
      K=1
      GOTO5
C STARTER OF TYPE 2 OR 3
C INITIAL POSITION IN ZONE 1
  202 CALL RANDL(IX,R)
      ROSQ=R*RISQ
      RC=SQRT(ROSQ)
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      X=RC*TCOS(I1)
      Y=RC*TSIN(I1)
      CALL ISO
C INITIAL PATH LENGTH IN ZONE 1
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      RL=-TLOG(I1)/SIGT(I)
```

```
C REJECTION CALCULATION
      X=X+U*RL
      Y=Y+V*RL
      ROSQ=X**2+Y**2
      IF(ROSQ.GT.R1SQ)GOTO202
      K=1
      IF(L.EQ.3)GOTO204
      WG=SIGN(1.,SIGT(I)-SPAM)
      GOTO15
  204 WG=SIGN(1.,SIGSR(I))
      GOTO15
C STARTER OF TYPE 4
C INITIAL DIRECTION RELATIVE TO SURFACE NORMAL
  203 CALL RANDL(IX,R)
      U1=-SQRT(R)
      V1=SQRT(1.-R)
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      V=V1*TSIN(I1)
      SIN2=1.-V**2
C REJECTION CALCULATION
      H2=-CIA*U1/SIN2*SIGT(I)
      CALL RANDL(IX,R)
      IF(EXP(H2).LE.R)GOTO203
      V1=V1*TCOS(I1)
C INITIAL SURFACE POSITION AND CORRECTION OF DIRECTION
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      COSV=TCOS(I1)
      SINV=TSIN(I1)
      X=RO1*COSV
      Y=RO1*SINV
      U=COSV*U1-SINV*V1
      V=SINV*U1+COSV*V1
C COVEYOU SAMPLING OF INITIAL RESONANCE SCATTERING IN ZONE 1
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      H1=-TLOG(I1)
      I1=H1/H2
      H3=I1
      RL=(H1-H3*H2)/SIGT(I)
C POSITION OF INITIAL SCATTERING
      X=X+U*RL
      Y=Y+V*RL
      ROSQ=X**2+Y**2
      WG=SIGN(1.,SIGSR(I))
      K=1
C NOW BEGINS COMMON HISTORY BLOCK
C SAMPLING OF ENERGY AFTER SCATTERING IN ZONE 1
   15 CALL RANDL(IX,R)
      E=E*(1.-R*ONAL1)
      IF(E.LT.EL)GOTO82
      I=(E-EL)/DELE+1.
C ISOTROPIC DIRECTION ZONE 1
      CALL ISO
C PATH LENGTH IN ZONE 1
```



```
    6 CALL RANDL(IX,R)
      I1=1000.*R+1.
      RL=-TLOG(I1)/SIGT(I)
C DISTANCE TO ZONE INTERFACE
      DELTA=U*X+V*Y
      IF(K.EQ.2)GOTO13
      SL=(SQRT(DELTA**2+SIN2*(R1SQ-ROSQ))-DELTA)/SIN2
      GOTO14
   13 K=1
      SL=-2.*DELTA/SIN2
   14 IF(SL.GE.RL)GOTO4
      X=X+U*SL
      Y=Y+V*SL
C MODIFIED TRACK LENGTH SCORING IN CASE OF PASSAGE
      RES=RES+WG*SAST(I)*SIGT(I)*SL
      GOTO5
C COLLISION IN ZONE 1
    4 CALL RANDL(IX,R)
      IF(R-SAST(I))18,17,17
C ABSORPTION
   18 IF(LABEL)GOTO207
C MODIFIED TRACK LENGTH SCORING IN CASE OF ABSORPTION
      H1=SL*SIGT(I)
      H2=EXP(H1)
      RES=RES+WG*SAST(I)*(1.-(1.-H2)*(H1+1.))/H2
      GOTO82
C SCATTERING IN ZONE 1
   17 IF(LABEL)GOTO19
C MODIFIED TRACK LENGTH SCORING IN CASE OF SCATTERING
      RES=RES+WG*SAST(I)*SIGT(I)*RL
   19 X=X+U*RL
      Y=Y+V*RL
      ROSQ=X**2+Y**2
      GOTO18
C PATH LENGTH IN ZONE 2
    5 CALL RANDL(IX,R)
      I1=1000.*R+1.
      RL=-TLOG(I1)/SHM
      IF(K.EQ.1)GOTO7
      DELTA=U*X+V*Y
      IF(DELTA.GT.0.)GOTO7
    8 D=DELTA**2-SIN2*(ROSQ-R1SQ)
      IF(D.LT.0.)GOTO7
C DIRECTION TOWARDS ZONE 1
      T=(-DELTA-SQRT(D))/SIN2
      IF(T.GT.RL)GOTO9
      X=X+U*T
      Y=Y+V*T
      GOTO6
C DIRECTION TOWARDS OUTER BOUNDARY
    7 K=2
      H1=SIGN(RK,U)
      TX=(H1-X)/U
      H1=SIGN(RK,V)
      TY=(H1-Y)/V
      T=TY
```

```
      IF(TX.LT.TY)T=TX
      IF(T.GT.RL)GOTO9
C REFLECTION
      X=X+U*T
      Y=Y+V*T
      ROSQ=X**2+Y**2
      RL=RL-T
      IF(TX-TY)10,11,11
   10 L=-U
      GOTO12
   11 V=-V
   12 DELTA=U*X+V*Y
      IF(DELTA)8,8,7
C ENERGY SAMPLING FOR ZONE 2 COLLISION
    9 CALL RANDL(IX,R)
      IF(IMOD.EQ.1)GOTO69
      DO 70 J=1,IMOD
   70 IF(R.LT.RAND(J))GOTO71
   71 H1=0.
      IF(J.GT.1)H1=RAND(J-1)
      R=(R-H1)/(RAND(J)-H1)
   69 E=E*(1.-R*ONALN(J))
      IF(E.LT.EL)GOTO82
      I=(E-EL)/DELE+1.
C POSITION FOR ZONE 2 COLLISION
      LABEL=.FALSE.
      X=X+U*RL
      Y=Y+V*RL
      ROSQ=X**2+Y**2
C ISOTROPIC DIRECTION ZONE 2
      IF(IANIS(J).EQ.1)GOTO72
      CALL ISO
      GOTO5
   72 CALL ANIS
      GOTO5
   82 IS=IS+1
      PN(IS)=RES
      CALL TIME(ISLUT)
      TID=FLOAT(ISLUT-ISTART)/3600.
      IF(TID.LT.TMAX.AND.IS.LT.1000)GOTO81
      TO=TO+TID
      H1=0.
      H2=0.
      DO 85 I=1,IS
      H3=PN(I)
      H1=H1+H3
   85 H2=H2+H3**2
      RS=IS
      H1=H1/RS
      H2=H2/RS
      H3=COEF*RIN7(L)/RN
      H4=H1+H3
      H5=SQRT((H2-H1**2)/(RS-1.))*H3
      ISMAX(N,L)=IS
      IF(N.EQ.1)GOTO87
      H1=ISMAX(1,L)
```



```
      H2=ISMAX(2,L)
      H3=H1+H2
      PRES(L)=(H1*PRES(L)+H2*H4)/H3
      DPRS(L)=SQRT((H1*DPRS(L))**2+(H2*H5)**2)/H3
      GOTO350
   87 PRES(L)=H4
      DPRS(L)=H5
  350 CONTINUE
      IF(N.EQ.2)GOTO352
      H1=0.
      DO 353 L=1,4
  353 H1=H1+DPRS(L)
      DO 354 L=1,4
      TDEL(L)=TIDMAX*(DPRS(L)/H1-0.0625)
  354 IF(TDEL(L).LT.0.2)TDEL(L)=0.2
  352 CONTINUE
      P=PRES(5)-PRES(1)-PRES(2)+PRES(3)+PRES(4)
      H1=0.
      DO 351 L=1,4
  351 H1=H1+DPRS(L)**2
      DP=SQRT(H1)
      TO=(P/DP)**2/TD/60.+0.5
      WRITE(NOUT,121)
  121 FORMAT(1H0,52HCONTRIBUTION OF THE FIVE TERMS TO RESONANCE INTEGRAL
     1)
      WRITE(NOUT,122)PRES(5)
  122 FORMAT(1H ,37HDETERMINISTIC TERM=                    ,E12.5)
      WRITE(NOUT,123)PRES(1),DPRS(1)
  123 FORMAT(1H ,37HNEGATIVE OUTWARD SURFACE SOURCE TERM=,E12.5,3X,
     1HDEVIATION=,E12.5)
      WRITE(NOUT,124)PRES(2),DPRS(2)
  124 FORMAT(1H ,37HNEGATIVE VOLUME SOURCE TERM=           ,E12.5,3X,
     1HDEVIATION=,E12.5)
      WRITE(NOUT,117)PRES(3),DPRS(3)
  117 FORMAT(1H ,37HPOSITIVE VOLUME SOURCE TERM=           ,E12.5,3X,
     1HDEVIATION=,E12.5)
      WRITE(NOUT,118)PRES(4),DPRS(4)
  118 FORMAT(1H ,37HPOSITIVE INWARD SURFACE SOURCE TERM= ,E12.5,3X,
     1HDEVIATION=,E12.5)
      WRITE(NOUT,119)P,DP
  119 FORMAT(1H0,37HRESONANCE INTEGRAL=                    ,E12.5,3X,
     1HDEVIATION=,E12.5)
      H1=AREA1*CONCA/SOUR
      P=H1*P
      DP=H1*DP
      WRITE(NOUT,120)P,DP
  120 FORMAT(1H0,37HRESONANCE CAPTURE PROBABILITY=         ,E12.5,3X,
     1HDEVIATION=,E12.5)
      WRITE(NOUT,122)TO
  125 FORMAT(1H0,20HSTATISTICAL QUALITY=,I5)
      GOTO88
      END
```

```
      SUBROUTINE ISO
      COMMON/COM3/U,V,W,SIN2,IX,TCOS(1000),TSIN(1000),AM(5),ASGPL1(5),J
      CALL RANDL(IX,R)
      W=2.*R-1.
      SIN2=1.-W**2
      H1=SQRT(SIN2)
      CALL RANDL(IX,R)
      I1=1000.*R+1.
      U=H1*TCOS(I1)
      V=H1*TSIN(I1)
      RETURN
      END



      SUBROUTINE ANIS
      COMMON/COM3/U,V,W,SIN2,IX,TCOS(1000),TSIN(1000),AM(5),ASGPL1(5),J
      CALL RANDL(IX,R)
      IF(AM(J).LT.1.E)GOTO1
      RMY=2.*R-1.
      H1=AM(J)*RMY
      A=(1.+H1)/SQRT(ASGPL1(J)+2.*H1)
      B=SQRT(1.-A**2)
      GOTO2
    1 A=SQRT(R)
      B=SQRT(1.-R)
    2 CALL RANDL(IX,R)
      I1=1000.*R+1.
      C=TCOS(I1)
      D=TSIN(I1)
      ECB=B*C*W
      BC=B*C
      SUIN=SQRT(SIN2)
      UEF=(BCW*U-BD*V)/SUIN+A*U
      V=(BCW*V+BD*U)/SUIN+A*V
      U=UEF
      W=A*W-B*C*SUIN
      SIN2=1.-W**2
      RETURN
      END



      FUNCTION PS(T)
      COMMON/COM4/PI,DIA,SHM,RK,PINVS(46)
      IF(T.GE.6.)GOTO1
      IF(T.GE.1.)GOTO2
      A=20.
      B=1.
      GOTO3
    2 A=8.
      B=16.
    3 H1=A*T+B
      I=H1
      H2=PINVS(I)
      H3=PINVS(I+1)
      H4=H3-H2
      H8=I
      H6=H1-H8
      H7=H6*H4
      PS=1./H7
      RETURN
    1 PS=6./PINVS(46)/T
      RETURN
      END
```

```
      SUBROUTINE PINCA
      COMMON/COM4/PI,DIA,SHM,RK,PINVS(46)
      COMMON/COM5/RO1
      DIMENSION X1(8),Y1(8),WG1(8),U1(64),V1(64),ST1(10),PESC(10),
     1RESULT(10),OPTCH(10),OPT1(10),OPT2(10),CAM(10),RESNOM(10),
     2STMCH(10),PSTAR1(46),YG(8),WG1(8)
      DATA YG,WG1/0.001998,.020642,.062266,.2C6855,.394469,.617311,
     1.825662,.964756,.003255,.017843,.045439,.079200,.106047,.112906,
     2.051119,.044551/
      OPTRO=RO1*SHM
      OPRISQ=OPTRO**2
      OPTRK=RK*SHM
      DO 301 I=1,8
      Y1(I)=OPTRO*(1.-YG(I))
  301 X1(I)=SQRT(OPRISQ-Y1(I)**2)
      CA=PI/128.
      DO 302 J=1,32
      ARGU=CA*(FLOAT(J)-C.5)
      H1=COS(ARGU)
      H2=SIN(ARGU)
      U1(J)=H1
      V1(J)=H2
      J2=65-J
      U1(J2)=H1
  302 V1(J2)=-H2
      DO 303 K=1,10
      H1=1.2*FLOAT(K)
      STMCH(K)=H1
      ST1(K)=H1/DIA
      PESC(K)=PO(H1/2.)
      RESNOM(K)=0.
  303 RESULT(K)=0.
      DO 350 I=1,8
      XC=X1(I)
      YC=Y1(I)
      WG=WG1(I)
      H1=2.*XC/SHM
      DO 304 K=1,10
      H2=H1*ST1(K)
      OPTCH(K)=H2
  304 RESNOM(K)=RESNOM(K)+BIC3(H2)*WG
      DO 350 J=1,64
      U=U1(J)
      V=V1(J)
      X=XC*U-YC*V
      Y=XC*V+YC*U
      DO 305 K=1,10
      CPT1(K)=0.
  305 OPT2(K)=OPTCH(K)
  306 CPTMOD=0.
  307 TX=(OPTRK-X)/U
      H1=SIGN(OPTRK,V)
      TY=(H1-Y)/V
      IF(TY.LT.TX)GOTO311
      CPTMOD=OPTMOD+TX
      X=-OPTRK
      Y=Y+V*TX
      GOTO312
  311 CPTMOD=OPTMOD+TY
      X=X+U*TY
      Y=-H1
  312 D=CPRISQ-(U*Y-V*X)**2
      IF(D)307,313,313
  313 T=-U*X-V*Y-SQRT(D)
```

```
      CPTMOD=OPTMOD+T
      DO 308 K=1,10
      CPTAC=OPT2(K)
      CPTBC=OPT1(K)
      OPTAD=OPTAC+OPTMOD
      OPTBC=OPTBC+OPTMOD
      H1=BIC3(OPTBC)-BIC3(OPTAC)-BIC3(OPTED)+BIC3(OPTAD)
  308 RESULT(K)=RESULT(K)+H1*WG
      X=X+U*T
      Y=Y+V*T
      DELTA=U*X+V*Y
      SL=-2.*DELTA
      X=X+U*SL
      Y=Y+V*SL
      SL=SL/SHM
      IF(OPT1(1)+SL*ST1(1)+OPTMOD.GT.7.)GOTO350
      DO 309 K=1,10
      H1=SL*ST1(K)+OPTMOD
      OPT1(K)=OPT1(K)+H1
  309 OPT2(K)=OPT2(K)+H1
      GOTO306
  350 CONTINUE
      H2=PI/8.
      DO 351 K=1,10
  351 RESULT(K)=RESULT(K)+PESC(K)/64./(H2-RESNCH(K))
      PSTAR1(1)=1.
      PSTAR1(13)=RESULT(1)
      DO 346 K=2,10
      L=2*K+16
  346 PSTAR1(L)=RESULT(K)
      DO 347 K=1,10
      X=STMCH(K)
      P=PESC(K)
      PS1=RESULT(K)
      H1=X*P*PST
  347 GAM(K)=H1/(P-PST+H1)
      GAMMA=GAM(1)
      DO 348 L=2,12
      H1=L
      X=(H1-1.)/10.
      P=PO(X/2.)
  348 PSTAR1(L)=P*GAMMA/(1.-(1.-GAMMA)*(1.-X*P))
      H2=GAM(1)
      H3=GAM(2)-H2
      DO 349 L=14,21
      H1=L
      X=(H1-1.)/10.
      GAMMA=H2+H3/1.2*(X-1.2)
      P=PO(X/2.)
  349 PSTAR1(L)=P*GAMMA/(1.-(1.-GAMMA)*(1.-X*P))
      DO 360 K=3,10
      H1=K
      X1=1.2*H1-1.2
      H3=GAM(K-1)
      H4=GAM(K)-H3
      DO 360 M=1,2
      H2=M
      X=X1+0.4*H2
      GAMMA=H3+H4/3.*H2
      P=PO(X/2.)
      L=2*K+M+13
  360 PSTAR1(L)=P*GAMMA/(1.-(1.-GAMMA)*(1.-X*P))
      DO 361 L=5,46
  361 PSTAR1(L)=1./PSTAR1(L)
```



```
      FUNCTION SELIN(A,B,N,EM)
      COMMON/COM1/XX,YY
      DIMENSION FUNC(129)
      H=(B-A)/FLOAT(N)
      NP1=N+1
      DO 11 I=1,NP1
      XX=A+FLOAT(I-1)*H
      CALL FCT
   11 FUNC(I)=YY
      DEL=0.
      DO 1 I=1,N
    1 DEL=DEL+ABS(FUNC(I+1)-FUNC(I))
      RI=0.
      DO 2 I=1,N
      XL=A+FLOAT(I-1)*H
      NJ=EM*ABS(FUNC(I+1)-FUNC(I))/DEL+1.
      HJ2=H/FLOAT(NJ)
      HJ=HJ2/2.
      RIJ=FUNC(I)+FUNC(I+1)
      XX=XL-HJ
      H1=0.
      DO 3 J=1,NJ
      XX=XX+HJ2
      CALL FCT
    3 H1=H1+YY
      RIJ=RIJ+4.*H1
      IF(NJ.EQ.1)GOTO2
      XX=XL
      NJ1=NJ-1
      H2=0.
      DO 4 J=1,NJ1
      XX=XX+HJ2
      CALL FCT
    4 H2=H2+YY
      RIJ=RIJ+2.*H2
    2 RI=RI+RIJ*HJ/3.
      SELIN=RI
      RETURN
      END



      SUBROUTINE FCT
      COMMON/COM1/XX,YY
      COMMON/COM2/ETA,ER,A1,A2,A3,A4,SPA,CONCA
      COMMON/COM5/RO1
      H7=A4*(XX-ER)
      CALL ERFCPL(H7,ETA,U,V)
      SA=A1*U*SQRT(ER/XX)
      SS=A2*U+A3*V+SPA
      ST=SA+SS
      CPT=ST*CONCA*RO1
      PSTAR=PS(OPT)
      YY=SA*CONCA*(PSTAR+SPA/ST*(1.-PSTAR))/XX
      RETURN
      END
```

```
      SUBROUTINE ERFCPL(X,Y,U,V)
      C=0.
      D=0.
      N=1
      IF(Y.GE.0.)GOTO1
      C=(X+Y)*(X-Y)
      D=-2.*X*Y
      X=-X
      Y=-Y
    1 IF(X.GE.0.)GOTO2
      X=-X
      N=-1
    2 IF(X+Y.GE.3.5.OR.Y+0.2*X.GE.1.7)GOTO3
      A=(X+Y)*(X-Y)
      B=2.*X*Y
      T=0.4*A
      RM=-0.04*(A**2+B**2)
      P1=-14.3+T*3.3
      P3=210.21+T*P1+RM*3.3
      P2=-519.3045+T*P3+RM*P1
      P1=3287.20821+T*P2+RM*P3
      P3=-4455.53759+T*P1+RM*P2
      P2=14448.00588+T*P3+RM*P1
      P1=(-8488.78070+T*P2+RM*P3)/5.
      T1=12096.5125+A*P1+RM*P2
      T2=B*P1
      P3=11.+T
      P2=89.1+T*P3+RM
      P1=514.8+T*P2+RM*P3
      P3=2207.205+T*P1+RM*P2
      P2=7151.3442+T*P3+RM*P1
      P1=17481.0636+T*P2+RM*P3
      P3=31537.26576+T*P1+RM*P2
      P2=35914.35158+T*P3+RM*P1
      P1=(21832.52763+T*P2+RM*P3)/5.
      RN1=12096.5125+A*P1+RM*P2
      RN2=B*P1
      P3=1.1283791671/(RN1**2+RN2**2)
      P2=P3*(RN1*T1+RN2*T2)
      P1=P3*(RN1*T2-RN2*T1)
      T=EXP(-A)
      U=T*COS(B)-X*P1-Y*P2
      V=-T*SIN(B)+X*P2-Y*P1
      GOTO4
    3 P1=Y**2
      U=0.
      V=0.
      T=1.
    5 A=X+T*0.31424037763
      B=0.18147956922/(A**2+P1)
      U=U+B
      V=V+A*B
      A=X+0.9477883912*T
      B=0.06291727763/(A**2+P1)
      U=U+B
      V=V+A*B
      A=X+1.5976826352*T
```



```
      B=0.01642733203/(A**2+P1)
      U=U+B
      V=V+A*B
      A=X+2.2795070805*T
      B=0.001243124432/(A**2+P1)
      U=U+B
      V=V+A*B
      A=X+3.0206370251*T
      B=0.00002725085347/(A**2+P1)
      U=U+B
      V=V+A*B
      A=X+3.8897248579*T
      B=0.00000008462432841/(A**2+P1)
      U=U+B
      V=V+A*B
      IF(T.LT.0.)GOTO6
      T=-1.
      GOTO5
    6 U=U*Y
    4 V=V*FLOAT(N)
      IF(C.EQ.0..AND.D.EQ.0.)RETURN
      C=EXP(-C)
      U=2.*C*COS(D)-U
      V=2.*C*SIN(D)-V
      RETURN
      END
```

```
      FUNCTION BIC3(X)
      IF(X.GE.1.)GOTO1
      BIC3=X*(X*(X*(X*(X*4.16451871E-2-1.73373703E-1)
     1+3.42922057E-1)-5.08602394E-1)+7.48517345E-1)-9.98434740E-1)
     2+7.85279789E-1
      RETURN
    1 IF(X.GE.2.)GOTO2
      BIC3=X*(X*(X*(X* 1.18535064E-2-1.07572888E-1)+4.13391240E-1)
     1-8.23631468E-1)+7.44150729E-1
      RETURN
    2 IF(X.GE.4.)GOTO3
      BIC3=X*(X*(X*(X*(X*(-4.29476343E-4)+8.23936987E-3)-6.59959218E-2)
     1+2.80376616E-1)-6.44209963E-1)+6.53752725E-1
      RETURN
    3 IF(X.GE.8.)GOTO4
      BIC3=X*(X*(X*(X*(X*(-1.50018618E-8)+5.20057758E-4)-7.2824[illegible]-3)
     1+5.17164358E-2)-1.87308876E-1)+2.787517[illegible]E-1
      RETURN
    4 IF(X.GE.10.)GOTO5
      BIC3=[illegible]*(X-10.)
      RETURN
    5 BIC3=0.
      RETURN
      END
```

```
      FUNCTION DEXP(X)
      IF(X.GE.0.96)GOTO1
      DEXP=X*(X*(X*1.0462423E-1-4.6603364E-1)+9.9377340E-1)+1.9328476E-4
      RETURN
    1 IF(X.GE.2.3)GOTO2
      DEXP=X*(X*(X*3.3581449E-2-2.6586457E-1)+7.9497384E-1)+6.9509154E-2
      RETURN
    2 IF(X.GE.4.24)GOTO3
      DEXP=X*(X*(X*6.7157845E-3-8.6419575E-2)+3.8766640E-1)+3.8377218E-1
      RETURN
    3 IF(X.GE.8.)GOTO4
      DEXP=X*(X*(X*4.5466238E-4-9.8122242E-3)+7.1109749E-2)+8.2603369E-1
      RETURN
    4 DEXP=1.
      RETURN
      END
```

```
      FUNCTION PO(T)
      DIMENSION PINVO(46)
      DATA PINVO/1.,1.0652,1.1299,1.1950,1.2610,1.3278,1.3957,1.4646,
     11.5346,1.6056,1.6780,1.7513,1.8256,1.9011,1.9775,2.0548,2.1334,
     22.2128,2.2930,2.3735,2.4561,2.7920,3.1389,3.4952,3.8591,4.2292,
     34.6045,4.9841,5.3668,5.7524,6.1402,6.5295,6.9204,7.3126,7.7059,
     48.0998,8.4947,8.8856,9.2881,9.6815,10.0776,10.4745,10.8719,
     511.2689,11.6673,12.0642/
      IF(T.GE.6.)GOTO1
      IF(T.GE.1.)GOTO2
      A=.5
      B=1.
      GOTO3
    2 A=.5
      B=16.
    3 P1=A*T+B
      I=P1
      P2=PINVO(I)
      P3=PINVO(I+1)
      P4=P3-P2
      P5=I
      P6=P1-P5
      P7=P2+P6*P4
      PO=1./P7
      RETURN
    1 PO=0.5/T*(1.-0.1875/T**2)
      RETURN
      END
```


14. APPENDIX IV: ESCAPE PROBABILITY P_o FOR INFINITE CYLINDERS

The development in sec. 3 indicated the need for efficient subroutines calculating the escape probability P_o as well as the collision probability P^*. Spanier and Gelbard [2] suggest the methods to be applied here for both quantities. P_o is treated in this appendix, while P^* is treated in Appendix V.

The escape probability P_o for a medium is the probability that an isotropic starter picked from a uniform distribution in the medium will escape from it uncollided. P_o for an infinite cylinder is a well-defined function of the optical radius $\zeta = a\Sigma$. In fact it is given in terms of the modified Bessel functions by the following expression [15, 21]

$$P_o(\zeta) = \frac{2\zeta}{3}\left\{2\left[\zeta\,(I_oK_o + I_1K_1) - 1\right] + I_oK_1 - I_1K_o + \frac{I_1K_1}{\zeta}\right\} \quad . \quad (47)$$

the arguments of I_o, K_o, I_1, K_1 being ζ.

Spanier and Gelbard [2] notice that $1/P_o(\zeta)$ is an almost linear function of ζ and give a table* of this function for $0 \leq \zeta \leq 6$; the arguments are $\zeta = 0,\ 0.05,\ 0.1,\ \ldots,\ 0.95,\ 1.0,\ 1.2,\ \ldots,\ 5.8,\ 6.0$. They claim that linear interpolation in this table yields P_o to within 0.1%. Beyond the table range they recommend the expression

$$P_o(\zeta) \approx \frac{1}{2\zeta} \quad (48)$$

In MCSUP the above technique is utilized in constructing the subroutine "PO". However, in order to keep within the 0.1% accuracy for all ζ, (48) was replaced by the following more accurate expression, obtained by retaining one more term in the asymptotic expansion for $P_o(\zeta)$ [15]

$$P_o(\zeta) \approx \frac{1}{2\zeta}\left(1 - \frac{3}{10\zeta^2}\right) \quad (49)$$

---

*Three errors in Spanier and Gelbard's table have been detected. The values of $1/P_o$ at ζ = 1.4, 2.4, and 3.6, should read 3.1389, 4.6041, and 7.7059.

15. APPENDIX V: FUEL-TO-MODERATOR COLLISION PROBABILITY P^*

We now turn to discuss the calculation of the fuel-to-moderator collision probability P^*, which occurs in (31) and (32).

Also here we shall follow Spanier and Gelbard [2], but only to a certain extent; we found it much more satisfactory to compute P^* by purely deterministic methods instead of using Monte Carlo.

$P^*(E)$ is defined as the probability that a source neutron with energy E drawn from a uniform distribution in the fuel will make its first collision in the moderator.

As the moderator cross section is assumed energy independent, we first notice that P^* for a fixed configuration is a function only of the optical radius $\zeta = a\Sigma$ of the fuel.

As for P_o (Appendix IV) we want to construct a table of $P^*(\zeta)$, or rather $1/P^*(\zeta)$, this quantity behaving more linearly. This table is of course not universal but depends on the actual configuration. This table has (like the $1/P_o$-table) 46 entries with $\zeta = 0,\ 0.05,\ 0.1,\ \ldots,\ 0.95,\ 1.0,\ 1.2,\ \ldots,\ 5.8,\ 6.0$, and linear interpolation should be quite adequate to find $P^*(\zeta)$ for arbitrary ζ. Now Spanier and Gelbard [2] point out that it is not necessary to calculate all the corresponding 46 P^*-values. They take advantage of Nordheim's approximate formula [22]

$$P^*(\zeta) \approx \frac{P_o(\zeta)\, G_m}{1-(1-G_m)(1-G_f(\zeta))} \tag{50}$$

where G_m and $G_f(\zeta)$ is the sticking probability [2, 22] for the moderator and fuel, respectively, and define a quantity $\gamma_m(\zeta)$ which exactly satisfies the following equation

$$P^*(\zeta) = \frac{P_o(\zeta)\, \gamma_m(\zeta)}{1-(1-\gamma_m(\zeta))(1-G_f(\zeta))} \tag{51}$$

Due to (50), $\gamma_m(\zeta)$ is a slowly varying function of ζ. Now, $G_f(\zeta)$ is given by [2]

$$G_f(\zeta) = 2\zeta P_o(\zeta) \ , \tag{52}$$

so, assuming $P_o(\zeta)$ to be known, $\gamma_m(\zeta)$ becomes a unique function of $P^*(\zeta)$. The idea is now to compute (in some way) $P^*(\zeta)$ at rather few ζ-values, e.g. $\zeta = 0.6,\ 1.2,\ \ldots,\ 6.0$. Via (51) one obtains 10 corresponding γ_m-values, and linear interpolation in this coarse γ_m-table renders next a full, 46-entry table for $\gamma_m(\zeta)$, which finally, again via (51), is converted to the desired 46-entry table for $P^*(\zeta)$ *.

Now it still remains to calculate $P^*(\zeta)$ at $\zeta = 0.6,\ 1.2,\ \ldots,\ 6.0$, and in doing so, we now digress from the exposition of Spanier and Gelbard [2], who recommend a Monte Carlo procedure for this job. We have found that an approach, inspired by Carlvik [6, 24], yield a both faster and more accurate calculation scheme.

Consider fig. 3 which shows a part of the infinite lattice system. P^* may be expressed by the double integral

$$P^* = \frac{2}{\pi} \int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} d\alpha \int_0^a dy\, l(y)\, \frac{1}{\frac{1}{2}\pi a^2}\, p(y,\alpha) \tag{53}$$

where $p(y,\alpha)$ is the probability that a starter drawn uniformly from the chord of length $l(y)$ after travelling in the direction α makes its first collision in the moderator. Referring to the optical scale on the α-ray depitched in fig. 3 below, one can show that $p(y,\alpha)$ is given by the expression

$$p(y,\alpha) = \frac{\sum_{i=1}^{\infty} \left\{ \left[Ki_3(t_i) - Ki_3(t_i+\lambda) \right] - \left[Ki_3(u_i) - Ki_3(u_i+\lambda) \right] \right\}}{\Sigma \cdot l(y)} \equiv \frac{\varphi(y,\alpha)}{\Sigma \cdot l(y)} \tag{54}$$

where $Ki_n(x)$ is the Bickley function

* Mikkelsen [1] has noticed a pronounced maximum in the relative error of (50) in the range $0 < \zeta < 0.6$ for small cells, perhaps making the above procedure questionable for such cells. We have, however, not revealed any significant errors for the final calculation results.

$$Ki_n(x) = \int_0^{\frac{\pi}{2}} \exp\left(-\frac{x}{\cos\theta}\right) \cos^{n-1}\theta \, d\theta \qquad (55)$$

Then (53) may be written

$$P^* = \frac{4}{\pi^2 \Sigma a^2} \int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} d\alpha \int_0^a dy \; \varphi(y,\alpha) \qquad (56)$$

In the numerical calculation of the double integral (56), a discrete representation is chosen for both variables:

1) α-integration: 64 equidistant α-values in $-\frac{\pi}{4} < \alpha < \frac{\pi}{4}$

2) y-integration: 8 non-equidistant y-values in $0 < y < a$.

The y-integration is of the Gaussian type. We choose a weight function mimicking the behaviour of the integrand near $y = 0$. For small y, $\varphi(y,\alpha)$ tends to be proportional to $1(y)$ and thereby to $y^{1/2}$. Hence, by introducing the new variable $x = (y/a)^2$, one could hope that the Gaussian representation

$$\int_0^1 x\, f(x)\, dx = \sum_{i=1}^{n} w_i f(x_i) \qquad \text{with} \quad \sum_{i=1}^{n} w_i = \frac{1}{2} \qquad (57)$$

would yield a good approximation to the integral in question. Ref. 18 gives tables of the x_i and w_i for different choices of n. With such a set of (x_i, w_i) one would then write

$$\int_0^a dy \; \varphi(y,\alpha) \approx 2a \sum_{i=1}^{n} w_i \, \varphi(a\eta_i, \alpha) \qquad (58)$$

where $\eta_i = x_i^2$. In fact, (58) has proved to be a surprisingly accurate approximation, and it seems rather difficult to justify mathematically the success of this procedure, which has been used widely in collision probability problems *. In our case, n = 8 was found to give an adequate compromise between speed and precision.

Hence, (56) is replaced by

$$P^* \approx \frac{1}{16\,\pi\Sigma a} \sum_{i=1}^{8} w_i \sum_{j=1}^{64} \varphi(a\eta_i, \alpha_j) \qquad (59)$$

where η_i and w_i are given below

i	η_i	w_i
1	0.001992	0.003295
2	0.020842	0.017843
3	0.082268	0.045439
4	0.206855	0.079200
5	0.394469	0.106047
6	0.617311	0.112506
7	0.825693	0.091119
8	0.964756	0.044551

We go a step further in refining the formula for P^* by reducing the discretization error. We observe that an expression quite similar to (59) exists for P_o:

$$P_o \approx \frac{1}{16\pi\Sigma a} \sum_{i=1}^{8} w_i \sum_{j=1}^{64} \varphi_o(a\eta_i, \alpha_j) \qquad (60)$$

where φ_o is the function $\varphi(y,\alpha)$ appropriate for an isolated fuel rod; (54) gives

---

* The method should presumably be credited to H. Tollander and I Carlvik, Sweden.

$$\varphi_0(y,a) = Ki_3(0) - Ki_3(\tau) = \frac{\pi}{4} - Ki_3(\tau) \qquad (61)$$

As we already have an efficient algorithm for P_o to our disposal we recast (59) to the following form

$$P^* \approx \frac{\sum_{i=1}^{8} w_i \sum_{j=1}^{64} \varphi(a\eta_i, a_j)}{\sum_{i=1}^{8} w_i \sum_{j=1}^{64} \varphi_0(a\eta_i, a_j)} \cdot P_o \qquad (62)$$

Formula (62) has the further advantage of yielding the exact limit P_o when the cell expands towards the isolated fuel rod.

The evaluation of the Ki_3-functions is performed by the function subprogram BIC3, developed originally by B. Tollander in connection with Carlvik's work[6]. This procedure zeros $Ki_3(x)$ for $x > 10$, and this leads to a natural truncation of the infinite series in (54).

The calculation technique suggested in this Appendix has been programmed as to form the subroutine PINCA. PINCA delivers the 46 P^*-values discussed earlier. Computation of $P^*(\zeta)$ by linear interpolation in the $1/P^*$-table so constructed is done by another subprogram, PS. For $\zeta > 6$ PS uses the approximation[2]

$$P^*(\zeta) \approx P^*(6) \cdot \frac{6}{\zeta} \qquad (63)$$

16. APPENDIX VI: A SELECTIVE INTEGRATION ROUTINE

The evaluation of the integral (31) is carried out by a special, selective integration routine SELIN. This was found useful because the integrand in (31) varies rather sharply around the resonance energy; as one wishes to limit the number N of integrand computations (here, N = 1000 was chosen) it is important to make a selective choice of the corresponding N abscissas.

SELIN does this in a crude, yet adequate manner. It works roughly in such a way that it divides the integration interval in a coarse equidistant subdivision. Thereafter, each of these subintervals Δ_i is submitted to a further subdivision in m_i new subintervals δ_{ij}, m_i being roughly proportional to the derivative of the integrand at Δ_i. Finally, Simpson's rule is applied to all subintervals and the results added. SELIN is in no way an "optimal" routine, but has nevertheless proved useful in the calculation of various definite integrals with a pronounced variation of the integrand.

SELIN is here supplied with function values from the function subprogram FCT.

17. APPENDIX VII: A FAST EXPONENTIAL ROUTINE

During the Monte Carlo calculations performed by MCSUP it is frequently required to calculate the exponential e^{-x}, or, more often, the function $1 - e^{-x}$. Of course the standard exponential function available at the computer may be applied, but in view of the rather limited demands for precision in Monte Carlo calculations it is better policy to use a faster routine and sacrify some accuracy. Spanier and Gelbard[2] recommend the use of the so-called g-function defined by

$$g(x) = \frac{x}{1-e^{-x}} \qquad (64)$$

After tabulating g(x) from e. g. 0 to 7 with steps of 0.2 it is possible by linear interpolation to calculate g(x) to an accuracy within 0.1%. For $x > 7$ they set $1 - e^{-x}$ equal to one.

Another alternative is to construct a simple polynomial approximation of $1 - e^{-x}$, and this method performed quite well on the IBM 7094 and was

adopted in MCSUP as the function subprogram OEXP. It was decided to divide the interval $0 < x < 8$ in four intervals in such a manner that third order polynomials chosen in the best possible way would yield absolute errors of almost the same magnitude in the four intervals. "Best possible" is understood in the Chebyshev sense, i.e. one minimizes the max. absolute error in each interval. The relevant approximation formula is given by Lance [7]:

$$\exp(kx) = I_0(k) + 2 \sum_{n=1}^{\infty} I_n(k)\, T_n(x), \quad -1 \leq x \leq 1 \qquad (65)$$

where I_n is the modified Bessel function of first kind and order n, while $T_n(x)$ is the Chebyshev polynomial of degree n. By truncating (65) at T_3 and making a variable transformation, this formula yields the polynomial coefficients. After some numerical experimentation the interval subdivision and coefficients stated in OEXP (Appendix III) was obtained. The OEXP approximations are not quite those of the Chebyshev type, because small constant terms had been added in order to keep the mean arithmetic error in the whole range $0 < x < 8$ as low as possible, whereby $\int_0^{\infty} e^{-x}\,dx$ calculated by OEXP becomes 1 with a high degree of precision (OEXP sets $1-e^{-x}$ equal to one for $x > 8$).

The performance of the g-function method (G) and OEXP has been measured on an IBM 7094 (model I) with the result shown in the table below.

	G	OEXP
netto time (μS) for a single call	240	180
root-mean-square error	1.1_{10-4}	1.27_{10-4}
max. error	3.01_{10-4}	2.37_{10-4}
abscissa for max. error	0.90	0.96
mean arithmetic error	-0.73_{10-4}	10^{-8}
mean absolute error	0.73_{10-4}	1.13_{10-4}
truncation error	9.1_{10-4}	3.35_{10-4}
zero value	0	1.93_{10-4}

The time for a call of the IBM 7094 standard function $1 - e^{-x}$ was found to 262 μS.

18. APPENDIX VIII: OPTIMIZATION OF COMPUTER TIME FOR THE FOUR TYPES OF STARTERS

The resonance absorption rate was in sec. 3 expressed as the sum of 5 terms (eq. (40)), where one is deterministic and the remaining four are Monte Carlo terms. The combined Monte Carlo expression

$$R = -I_1 - I_2 + I_3 + I_4 = R_1 + R_2 + R_3 + R_4 \qquad (66)$$

was composed of four mutually independent terms to be calculated separately according to the principles laid down in sec. 3. Now, some possibilities exist for carrying out a more refined composition of the terms in (66). Spanier and Gelbard [2] propose a *sampling technique*, where the initial energy E is sampled from a combined density function, derived from the four individual density functions, whereafter the category of starter is selected. This procedure implies saving of computer storage.

In our case, small computing time rather than storage saving is at a premium, and then another approach is feasible.

We still calculate the four terms in (66) separately, but we assume that a certain amount of computer time T is at our disposal and try to find a division

$$T = T_1 + T_2 + T_3 + T_4 \quad , \qquad (67)$$

where T_i is the time spent in calculation of R_i, so that a minimum-variance estimate of R is obtained. We have

$$\sigma^2(R) = \sum_1^4 \sigma^2(R_i) \qquad (68)$$

At this point we assume that

$$\sigma^2(R_i) = \frac{k_i}{T_i} \tag{69}$$

where the k_i are known positive constants. Hence we minimize the functional

$$\varphi(T_1, T_2, T_3, T_4) = \sum_1^4 \frac{k_i}{T_i} \tag{70}$$

under the constraint (67). The method of Lagrange's multipliers yields easily the solution

$$T_i = \frac{\sqrt{k_i}}{\sum_1^4 \sqrt{k_i}} T \tag{71}$$

In practice we have to estimate the k_i. This is done in MCSUP by applying a preliminary computation with time length T' distributed equally with $T_i' = \frac{T}{4}$ to each type of starters. Preliminary variance estimates

$$\sigma_i'^2 \cong \sigma^2(R_i') \tag{72}$$

are used to calculate k_i as

$$k_i = T_i' \sigma_i'^2 \tag{73}$$

Now (71) is recast to

$$T_i = \frac{\sigma_i'}{\sum_1^4 \sigma_i'} \cdot T \tag{74}$$

Hereafter the main calculation is executed. Utilizing also the preliminary calculation, we have

$$T_i'' = T_i - T_i' \tag{75}$$

where T_i'' is the time spent in calculation of R_i in the main calculation.

The MCSUP program uses $T' = \frac{T}{4}$ and with this choice we finally obtain

$$T_i'' = \left(\frac{\sigma_i'}{\sum \sigma_i'} - \frac{1}{16} \right) T \tag{76}$$

19. APPENDIX IX: COMPUTER DEPENDENT FEATURES OF MCSUP

MCSUP in its present form is intended for use on the IBM 7094 at the NEUCC center in Lyngby, Denmark. In view of the possibility of its use on other computers, we give in this appendix those features of the code that are specific for our system, and which perhaps have to be implemented in other ways for other machines.

19.1. The Timing Routine

It is essential that a timing routine is available. The call is of the type

CALL TIME (ICLOCK)

where the integer ICLOCK measures the computer time in units of 1/60 sec. The actual position of the zero on the time scale is irrelevant for our purpose.

19.2. The Random Number Generator

The random number calls are of the type

CALL RANDL (IX, R)

In our case, a standard multiplicative generator[23] is used. IX is a current integer and is not used (yet it has to be initialized, cf. the input description in 6.1). R is the resulting random number between 0 and 1.

19.3. Input/Output Units

The symbolic nos. of the input and output units are 5 and 6 for the IBM 7094 at NEUCC. Only the statements

NIN = 5

NOUT = 6

(at the beginning of the code print-out, cf. Appendix III) should be changed to fit other conventions.

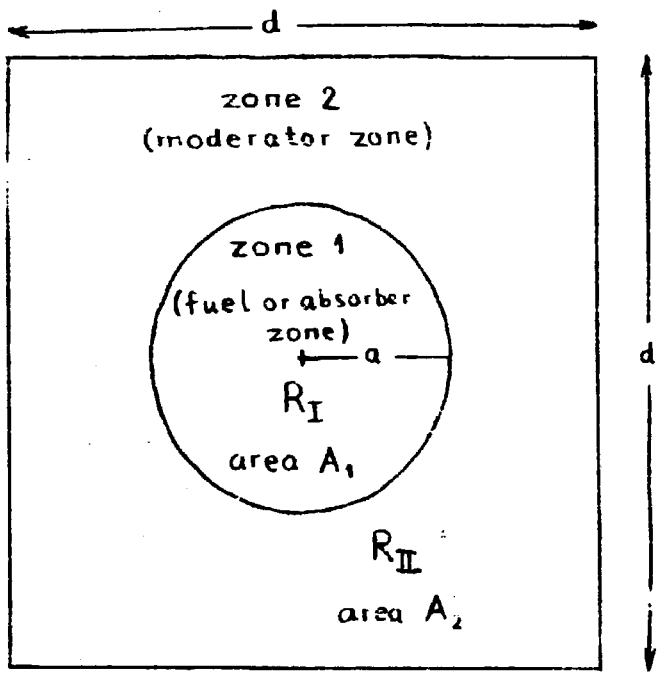
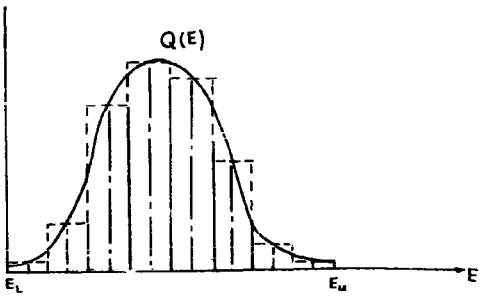


Fig. 1. MCSUP geometry



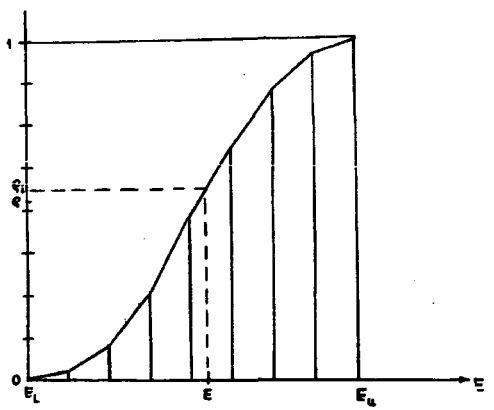
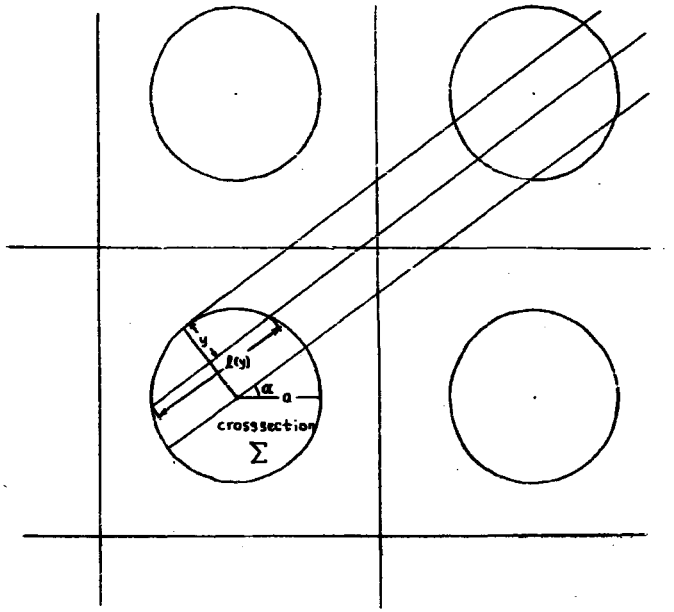
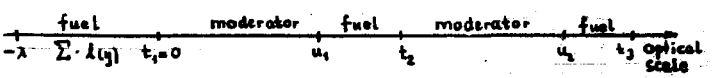


Fig. 2. Sampling of initial energy, schematic.





α - ray

Fig. 3. Calculation of [redacted]