

Берчану И. и Игнатович В. К.
Б1-3-9234.

+

С342Г1

Б-528



ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

4799/75

^x
Б1-3-9234

ДЕПОНИРОВАННАЯ ПУБЛИКАЦИЯ

Дубна 1975

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ
Лаборатория нейтронной физики

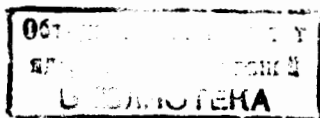
Депонированная публикация

И. Берчану, В. К. Игнатович

Б1-3-9234

СИСТЕМА ПОДПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ
МОЛЕКУЛЯРНОГО ТЕЧЕНИЯ УЛЬТРАХОЛОДНЫХ
НЕЙТРОНОВ ПО НЕЙТРОНОВОДАМ СЛОЖНОЙ
ГЕОМЕТРИЧЕСКОЙ КОНСТРУКЦИИ

Копия
20.06.1975



Дубна, 1975

Эксперименты с ультрахолодными нейтронами (УХН) /1/ включают в себя в качестве составной части транспортировку УХН по трубам от реактора до измерительной аппаратуры. В случае прямолинейных цилиндрических нейтронородов процесс транспортировки рассматривался в работе /2/ как теоретически, так и с помощью численного моделирования методом Монте-Карло. Однако обычно транспортировка производится по нейтронородам сложной геометрической конструкции. Чтобы удовлетворить потребности различных лабораторий и экспериментов, была разработана система подпрограмм, позволяющая рассчитывать транспортировку УХН по нейтронородам, состоящим из цилиндрических, конических участков и угловых поворотов, и имеющим реалистический закон отражения от стенок. Настоящая работа посвящена подробному описанию этих подпрограмм.

Подпрограмма RIGHT(ZI, ZK)

Это основная подпрограмма. Она описывает движение нейтрон по цилиндрической трубе (см.рис.1), начиная от некоторой заданной точки (r_i, z_i) (в цилиндрической системе координат с осью z вдоль оси цилиндра) с заданным направлением вектора скорости v , и кончая последним соударением со стенкой. Продвижение от начальной точки (r_i, z_i) до точки первого соударения (r_f, z_f) со стенкой описывается вектором

$$l = r_f + z_f - r_i - z_i = \Delta r + \Delta z \quad (1)$$

который имеет проекции на ось z (единичный вектор e_z) и на начальный радиус r_i , равные:

$$\begin{aligned} l e_z &= l \cos \theta_z = \Delta z, \\ l r_i &= r_f r_i - r_i^2 = \frac{1}{2} [r_f^2 - r_i^2 - (\Delta r)^2], \end{aligned} \quad (2)$$

квадрат вектора l равен:

$$l^2 = (\Delta r)^2 + (\Delta z)^2 = (\Delta z)^2 / \cos^2 \theta_z, \quad (3)$$

откуда следует (с учётом (2)), что

$$\Delta z = |\Delta r| \cdot \operatorname{ctg} \theta_z. \quad (4)$$

Принимая абсолютное значение начального радиуса равным ρ , а абсолютное значение радиуса трубы (а значит и $|r_f|$) равным I , получаем из (2):

$$|\Delta r| = -\rho \cos \psi_{\rho z} + \sqrt{1 - \rho^2 \sin^2 \psi_{\rho z}} \quad (5)$$

где $\psi_{\rho z}$ - угол между вектором ρ и проекцией v на плоскость, перпендикулярную оси z . Движение от одной точки соударения со стенкой до другой описывается теми же формулами (4) и (5), но в (5) следует ρ положить равным тоже единице. При этом $|\Delta r|$ оказывается равным

$$|\Delta r| = -2 \cos \psi_{\rho z} \quad (6)$$

поскольку $\sqrt{1 - \sin^2 \psi_{\rho z}} = -\cos \psi_{\rho z} > 0$.

На самом деле движение вдоль трубы удобно разложить на поступательное перемещение вдоль оси z и вращение в перпендикулярной плоскости, причем последнее удобно описывать с помощью комплексных чисел, а именно, следует ρ представить как $\rho e^{i\varphi_\rho}$, а из скорости v выделить направление $v_\rho = e^{i\varphi_v}$ и величи-

ну $V_z = c \operatorname{tg} \theta_z$. При этом

$$\alpha_f = \rho + A \cdot V_p, \quad z_f = z_i + A \cdot V_z \quad (7)$$

$$\psi_{pz} = \psi_p - \psi_v, \quad A = |\Delta z|$$

и при перескоке от стенки к стенке $|\Delta z| = -2 \operatorname{Re}(\rho \cdot V_p^*)$.
Величины z , ρ , V_p , V_z обозначаются в подпрограмме соответственно символами Z , RO , $VR0$, VZ и передаются другим подпрограммам через общий массив $|RV|$. Кроме того, подпрограмма считает число соударений со стенкой, которое обозначено символом NTR , и приписывает индикатору iN значение -1 или $+1$ в зависимости от того вернулась ли разыгрываемая частица обратно через входное отверстие, или благополучно вышла через выходное. Программа предусматривает также возможность поглощения частицы с вероятностью $5 \cdot 10^{-4}$ при каждом соударении со стенкой. В случае поглощения индикатору iN приписывается значение 0. Концы трубы описываются функциями $Z_i(x)$ и $Z_k(x)$ ($x = \operatorname{real} \rho$), которые составляются пользователем. Выход из подпрограммы происходит в том случае, когда координата z очередной точки столкновения со стенкой оказывается больше $Z_k(x)$ или меньше $Z_i(x)$.
Причем при выходе запоминаются координаты и углы вылета, относящиеся к предыдущему соударению. Подпрограмма $RIGHT$ обращается к подпрограмме $ANGLE$, которая вычисляет направление полета после столкновения со стенкой. Перед первым обращением к подпрограмме $RIGHT$ в начале розыгрыша истории очередного нейтрона вызывается подпрограмма $ORIGIN$, которая задает координаты начальной точки z , ρ и направление первого полета V_z, V_p .

ну $V_z = ctg \theta_z$. При этом

$$z_f = \rho + A \cdot V_\rho, \quad z_f = z_i + A \cdot V_z \quad (7)$$

$$\psi_{\rho z} = \psi_\rho - \psi_v, \quad A = |\Delta z|$$

и при перескоке от стенки к стенке $|\Delta z| = -2 \operatorname{Re}(\rho \cdot V_\rho^*)$.
Величины z , ρ , V_ρ , V_z обозначаются в подпрограмме соответственно символами Z , RO , VRO , VZ и передаются другим подпрограммам через общий массив $|RV|$. Кроме того, подпрограмма считает число соударений со стенкой, которое обозначено символом NTR , и приписывает индикатору iN значение $-I$ или $+I$ в зависимости от того вернулась ли разыгрываемая частица обратно через входное отверстие, или благополучно вышла через выходное. Программа предусматривает также возможность поглощения частицы с вероятностью $5 \cdot 10^{-4}$ при каждом соударении со стенкой. В случае поглощения индикатору iN приписывается значение 0. Концы трубы описываются функциями $z_i(x)$ и $z_k(x)$ ($x = z \cos \theta$), которые составляются пользователем. Выход из подпрограммы происходит в том случае, когда координата z очередной точки столкновения со стенкой оказывается больше $z_k(x)$ или меньше $z_i(x)$. При этом при выходе запоминаются координаты и углы вылета, относящиеся к предыдущему соударению. Подпрограмма $RIGHT$ обращается к подпрограмме $ANGLE$, которая вычисляет направление полета после столкновения со стенкой. Перед первым обращением к подпрограмме $RIGHT$ в начале розыгрыша истории очередного нейтрона вызывается подпрограмма $ORIGIN$, которая задает координаты начальной точки z , ρ и направление первого полета V_z, V_ρ .

Подпрограмма CONUS.

Эта подпрограмма описывает движение вдоль конуса и аналогична RIGHT . Перед обращением к подпрограмме должны быть заданы R_i и R_F - начальный и конечный радиусы конуса; $TG = (R_F - R_i) / ZL$ - тангенс угла α при вершине; $CN = R_i / R_F$ - нормировочный множитель, позволяющий при благополучном исходе нормировать выходной радиус конуса на единицу; ZL - полная длина конуса; $Z_i = c \cdot tg \alpha$ - координата входного и $Z_F = Z_i + ZL / R_i$ - координата выходного отверстия. Формулы, характеризующие движение нейтрона имеют тот же вид (I-4) , что и в программе RIGHT . Из них следует, что в случае соударения со стенкой

$$|\Delta z| = \frac{-(\rho \cdot \cos \psi_{sz} - z \cdot tg^2 \alpha \cdot ctg \theta_z) + \sqrt{(\rho \cdot \cos \psi_{sz} - z \cdot tg^2 \alpha \cdot ctg \theta_z)^2 + (z^2 \cdot tg^2 \alpha - \rho^2)(1 - tg^2 \alpha \cdot ctg^2 \theta_z)}}{1 - tg^2 \alpha \cdot ctg^2 \theta_z} \quad (8)$$

причём для первого соударения $z \cdot tg \alpha = 1$, а при последующих $z \cdot tg \alpha = \rho$. Выход из подпрограммы осуществляется сравнением радиуса траектории на входном и выходном отверстиях с фактическими радиусами, или сравнением текущей координаты z с Z_i и Z_F . Все вычисления производятся с помощью комплексной арифметики, причём результаты вычисления выдаются в общем массиве $|RV|$, имеющем тот же смысл, что и в подпрограмме RIGHT . Радиусы траектории на входном и выходном отверстиях задаются соответственно формулами:

$$R = |\rho \cdot e^{i\psi_p} + (Z_i - z) \cdot tg \theta_z \cdot e^{i\psi_v}|, \quad R = |\rho \cdot e^{i\psi_p} + (Z_F - z) \cdot tg \theta_z \cdot e^{i\psi_v}| \cdot R_i \quad (9)$$

При благополучном исходе радиус выходного отверстия нормируется

на единицу и соответственно меняется масштаб по оси Z , т.е. Z полагается равным $Z \cdot R_i / R_F$.

Подпрограмма ORIGIN

Эта подпрограмма задает начальные координаты и направление влета нейтрона в нейтроновод. Распределение на входном отверстии предполагается однородным и изотропным, т.е. число нейтронов, влетающих с участка dS в направлении $d\Omega$ равно

$$dn = \cos \theta_z \frac{dS d\Omega}{\pi^2} = dp^2 d\cos^2 \theta_z \frac{d\varphi_p}{2\pi} \frac{d\varphi_v}{2\pi} \quad (10)$$

Поэтому с помощью генератора случайных чисел $RANF(0)$ задаются четыре случайных числа U_1, U_2, U_3, U_4 и с их помощью определяются $\rho = \sqrt{U_1} \exp(2\pi i U_2)$, $v_p = \exp(2\pi i U_3)$, $v_z = \sqrt{U_4 / (1 - U_4)}$ из общего массива $|RV|$.

Подпрограмма ANGLE

В этой подпрограмме определяются углы вылета УХН по отношению к нормали к данному участку поверхности после соударения со стенкой. Затем эти углы приводятся к оси Z , т.е. вычисляются соответствующие элементы общего массива $|RV|$. Угол нормали N по отношению к оси Z задается элементами массива $|AL|$ (см. рис.2а). При отражении от стенки происходит преобразование углов падения Ω_0 в углы отражения Ω . Здесь закон отражения выбран в виде

$$W(\Omega_0, \Omega) = (1 - GA(\Omega)) \delta(\Omega - \Omega_0) + c GA(\Omega_0) A(\Omega) \cos \theta,$$

$$c \int A(\Omega) \cos \theta d\Omega = 1, \quad (\text{II})$$

где $A(\Omega)$ может быть выбрано равным 1 и тогда $c = 1/\pi$, (этот случай соответствует смешанному диффузно - зеркальному отражению) или $A(\Omega) = \cos \theta$, и тогда $c = 3/2\pi$, что соответствует "реалистическому" отражению. Поскольку закон отражения содержит зеркальную и незеркальную часть, то подпрограмма *ANGLE* сначала выбирает между этими возможностями, определяя случайное число U_1 с помощью генератора *RANF(0)* и сравнивая его с величиной $1 - GA(\Omega_0)$. Если

$$U_1 < 1 - GA(\Omega_0) \quad (\text{I2})$$

то отражение происходит зеркально. Чтобы проверить соотношение (I2) в случае $A(\Omega) = \cos \theta$, необходимо найти косинус угла падения:

$$\cos \theta = \frac{v_0 H}{|v_0|} ;$$

$$v_0 = v_{p0} + v_{z0}, \quad H = \rho \cos \mathcal{L} - e_z \sin \mathcal{L}, \quad \rho = e^{i\varphi_\rho}; \quad (\text{I3})$$

$$v_0 H = \cos \mathcal{L} \cdot \text{Re}(\rho \cdot v_{p0}^*) - v_{z0} \sin \mathcal{L}.$$

При зеркальном отражении вектор скорости отраженного нейтрона равен:

$$v = v_0 - 2H(v_0 H);$$

$$v'_p = v_{p0} - 2\rho \cos \mathcal{L} \cdot (v_0 H), \quad (\text{I4})$$

$$v'_z = v_{z0} + 2 \sin \mathcal{L} \cdot (v_0 H) \cdot e_z.$$

Однако v_p согласно определению должна быть комплексным числом с единичным модулем, поэтому, чтобы не менять направления отраженного вектора v и при этом иметь $|v_p| = 1$, необходимо нормировать v на $|v_p'|$. В результате этой процедуры получаем

$$v_p = \frac{v_p'}{|v_p'|} = e^{i\varphi_v}, \quad v_z = \frac{v_z'}{|v_z'|} = \operatorname{ctg} \theta_z. \quad (I5)$$

Если неравенство (I2) не выполняется, то вырабатываются два случайных числа U_2, U_3 , отвечающих распределению по углам вылета:

$$dn = c A(\Omega) \cos \theta d\Omega = \frac{d\varphi}{2\pi} \begin{cases} d\cos^2 \theta, & \text{при } A(\Omega) = 1 \\ d\cos^3 \theta, & \text{при } A(\Omega) = \cos \theta. \end{cases} \quad (I6)$$

Соответственно $\varphi = 2\pi U_2$, $\cos \theta = \sqrt{U_3}$ при $A(\Omega) = 1$ и $\cos \theta = \sqrt[3]{U_3}$ при $A(\Omega) = \cos \theta$. Имея углы φ и θ , можно построить

$$\begin{aligned} v_p' &= (v e_p + i \sin \theta \cos \varphi) \rho, \quad v_z' = v e_z; \\ e_p &= N \cos \mathcal{L} + t \sin \mathcal{L}, \quad e_z = t \cos \mathcal{L} - N \sin \mathcal{L}; \\ vN &= -\cos \theta, \quad vt = \sin \theta \cos \varphi, \end{aligned} \quad (I7)$$

где t — вектор, направленный по образующей в точке соударения. Затем v' нормируется так же, как в (I5). Ко времени обращения к подпрограмме *ANGLE* должен быть задан закон отражения (т.е. $A(\Omega)$), обозначенный символом *LAW* и постоянная G .

Подпрограмма TURN(ALFA)

При необходимости осуществить поворот используется подпрограмма *TURN*, которая выполняет поворот системы координат на угол \mathcal{L} вокруг оси y и находит z, ρ, v_ρ, v_z в новой системе координат:

$$\begin{aligned} z &= -\sin\mathcal{L} \operatorname{Re}(\rho_0) + z_0 \cos\mathcal{L}, \quad \rho = \rho_0 + \operatorname{Re}(\rho_0)(\cos\mathcal{L}-1) + z_0 \sin\mathcal{L}; \\ v_z' &= -\operatorname{Re}(v_{\rho_0}) \sin\mathcal{L} + v_{z_0} \cos\mathcal{L}, \\ v_\rho' &= v_{\rho_0} + \operatorname{Re}(v_{\rho_0})(\cos\mathcal{L}-1); \\ v_\rho &= v_\rho' / |v_\rho'|, \quad v_z = v_z' / |v_z'|. \end{aligned} \tag{19}$$

Угол поворота отсчитывается против часовой стрелки и задается в градусах.

Подпрограмма SUMS(i)

Ведет счет благополучно прошедших нейтронов, причем число прошедших нейтронов, обозначенное $NFZ(i)$ имеет индекс I , указывающий на то, что в случае прямых цилиндрических труб одновременно может рассчитываться несколько трактов (не более 10) и I обозначает номер тракта. Кроме подсчета полного числа прошедших нейтронов, подпрограмма *SUMS* строит распределение по углу и радиусу вылета:

$$f(\cos^2\theta_z) d\cos^2\theta_z; g(\rho^2) d\rho^2. \tag{20}$$

При этом весь интервал ρ^2 и $\cos^2\theta_z$ от 0 до 1 разбивается на 10 ячеек с шагом 0,1. Символы $NRO(i, k)$ и $ANGLE(i, k)$

обозначают число нейтронов I -го тракта, попавших в K -ю ячейку по ρ^2 и $\cos^2\theta_z$ соответственно.

Подпрограмма RESULT

В этой подпрограмме рассчитываются:

коэффициент пропускания тракта PR и его стандартная ошибка

$$PR = \frac{NFZ(i)}{NMAX}, \quad \sigma_{PR} = \sqrt{\frac{PR(1-PR)}{NMAX}}; \quad (21)$$

сопротивление R и его стандартная ошибка

$$R = \frac{1}{PR} - 1, \quad \sigma_R = \frac{\sigma_{PR}}{(PR)^2}. \quad (22)$$

Угловое и радиальное распределения нормируются на единицу путем деления чисел $NRO(i, k)$ и $NANGLE(i, k)$, полученных в подпрограмме $SUMS$ на полное число прошедших нейтронов $NFZ(i)$.

Кроме того, подпрограмма $RESULT$ редактирует и выводит указанные величины на печать.

Подпрограммы READER и ZERON

Эти подпрограммы подготовительные. Подпрограмма $READER$ считывает с карт следующие величины в заданном ниже порядке:

Символ	Обозначение	Формат	Замечания
1	2	3	4
$KMAX$	Количество трактов	$i2$	
$NMAX$	Число разыгрываемых нейтронов	$i5$	

1	2	3	4
G	Постоянная из закона отражения (II)	F 5.3	Значения $0 + I$
$A(\Omega)$	Величина, определяющая тип отражения	A 6	Значение I или $C\Phi STH_{\perp}$
CHECK	Величина, определяющая геометрию	A 6	Значение $CIL_{\perp\perp\perp}$ или $C\Phi NUS_{\perp}$
$ZED(i), i=1, KMAX$	Длины составляющих отрезков	10F 8.3	
Ri	Входной радиус	F 10.2	} на одной карте
RF	Выходной радиус	F 10.2	

В принципе все эти данные могут задаваться в общей диспетчерской программе без использования подпрограммы *READER*, однако нужно помнить, что в *READER* задаются и константы $C = 2\pi$ и $Ci = i$ (мнимая единица), которые также нужно определять при отсутствии *READER*.

Подпрограмма *ZERON* очищает массивы, используемые в *SUMS*, и приводит к нулю счетчик разыгрываемых нейтронов *NEUTR*.

В заключение приводится листинг диспетчерской программы для нейтроновода с поворотом на 45° и всех описанных подпрограмм. Также приводятся примеры по построению функций $Zi(x)$ и $ZK(x)$ для некоторых геометрий: прямая цилиндрическая труба, нейтроновод с одним поворотом на любой угол, коническая труба.

Handwritten signature

Литература

1. Ф.Л.Шапиро. Сообщение ОИЯИ РЗ-7135, Дубна, 1973.
2. И.Берчану, В.К.Игнатович. Препринт ОИЯИ, Р4-7331, Дубна, 1973.
Уаеит 23, 441 (1973)

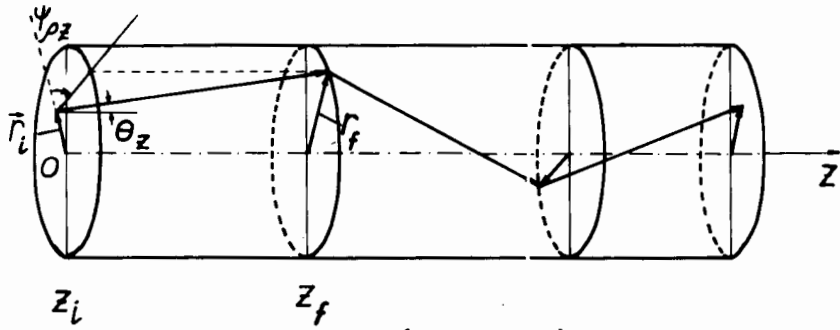


Рис. 1

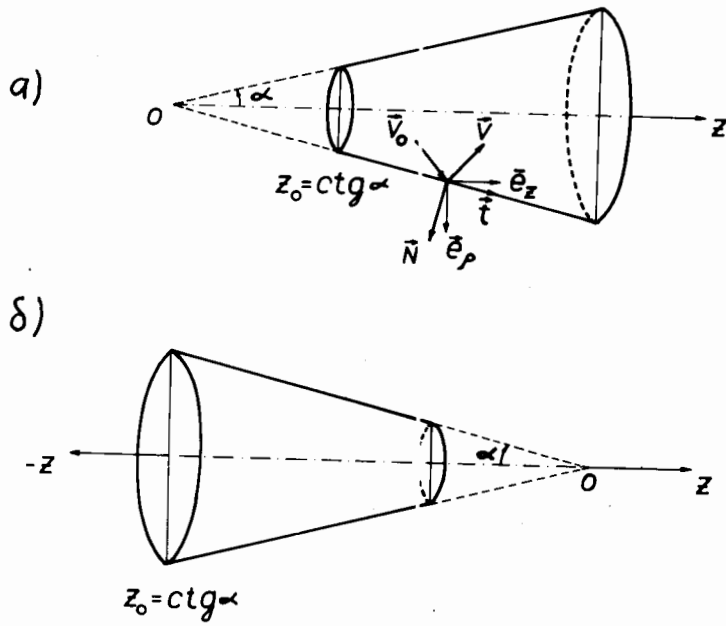


Рис. 2

SUBROUTINE READER

```
C***READING OF DATA *****
COMMON G,CI,C/GEO/GEO(2)/CHECK/CHECK
* /ZED/ZED(10)/LAW/LAW/KMAX/KMAX,NMAX
COMMON/CON/PI,RF,ZI,ZF,ZL,CN,TG
COMPLEX CI
DATA((GEO(I),I=1,2)=6HCIL ,6HCONUS )
READ1,KMAX
READ2,NMAX
READ3,G
READ4,CHECK
READ4,LAW
READ5,(ZED(I),I=1,KMAX)
PRINT 9,LAW
PRINT19,CHECK
PRINT11,G
IF(CHECK.EQ.GEO(2))GO TO 8
RETURN
8 READ 6,RI,RF
PRINT 7,RI,RF
CN=RI/RF
TG=(RF-RI)/ZED(1)
ZF=1./TG+ZED(1)/RI
ZI=1./TG
RETURN
1 FORMAT(I2)
2 FORMAT(I5)
3 FORMAT(F5.3)
4 FORMAT(A6)
5 FORMAT(10F8.3)
6 FORMAT(2F10.2)
7 FORMAT(///* ENTRANCE RADIUS=*,F10.2,3X,* EXIT RADIUS=*,F10.2)
9 FORMAT(//2X,*A(OMEGA)=*,A6 )
10 FORMAT(//2X,* G=*,F5.3)
19 FORMAT(///* GEOMETRY=*,A6)
END
```

```
      SUBROUTINE ZERON
C***PREPARATION OF ALL BLOCKS *****
      COMMON/ZERO/NEUTR,NFZ(10),NANGLE(10,10)
* ,NRO(10,10),NAV(10)
* /KMAX/KMAX,NMAX
      COMMON/ZAKON/ZAKON(2)
      COMMON/GEO/GEO(2)
      COMMON G,CI,C
      COMPLEX CI
      DATA ((GEO(I),I=1,2)=6HCIL ,6HCONUS )
DATA ((ZAKON(I),I=1,2)=6H1 ,6HCOSTH )
      C=6.2831852
      CI=(0.,1.)
      NEUTR=0
      DO 1 I=1,KMAX
      NFZ(I)=0
      NAV(I)=0
      DO 1 J=1,10
      NANGLE(I,J)=0
1      NRO(I,J)=0
      RETURN
      END
```

```
      SUBROUTINE ORIGIN
C***ORIGINAL POINT AND DIRECTION OF FLIGHT*****
      COMMON G,CI,C/RV/Z,RO,VRO,VZ,IN,NTR
      COMPLEX RO,VRO,CI
      NTR=0
      Z=0.
      U1=RANF(0)U2=C*RANF(0)COS2=RANF(0)
      RO=SQRT(U1)*CEXP(CI*U2)
      VZ=SQRT(COS2/(1.-COS2))
      U2=RANF(0)*C
      VRO=CEXP(CI*U2)
      RETURN
      END
```

```

SUBROUTINE RIGHT(ZI,ZK)
COMMON/RV/Z,RO,VRO,VZ,IN,NTR/AL/AL(2)
COMPLEX RO,RO1,VRO
CYLINDER*****
      AL(1)=1.0AL(2)=1.0
C***FIRST FLIGHT *****
      X=1.-CABS(RO)**2
      Y=REAL(RO*CONJG(VRO))
      A=SQRT(Y**2+X)-Y
3      RO1=RO+A*VRO
      FI=REAL(RO1)
      ZO=Z+A*VZ
C***GOING OUT OF THE TUBE*****
      IF(ZO.GT.ZK(FI))GO TO 1
C***RETURN BACK TO CONVERTER *****
      IF(ZO.LT.ZI(FI))GO TO 2
      RO=RO1,Z=ZO,NTR=NTR+1
      U=RANF(1)
C***PROBABILITY OF ABSORPTION*****
      IF(U.LT..0015)GO TO 4
      CALL ANGLE
C***FLIGHT BETWEEN TWO POINTS ON THE WALL*****
      A=-2.*REAL(RO*CONJG(VRO))
      GO TO 3
C***EXIT*****
1      IN=1
      RETURN
C***RETURN*****
2      IN=-1
      RETURN
C***ABSORPTION*****
4      IN=0RETURN
      END
```

```
      SUBROUTINE CONUS
      COMMON/CON/RI,RF,ZI,ZF,ZL,CN,TG
      * /RV/Z,RO,VRO,VZ,IN,NTR
      COMPLEX RO,VRO,R01
C****FIRST FLIGHT *****
      Z=ZI
      N=1
      IF(TG)8,8,7
      8 X=1.-CABS(RO)**2
        Y=REAL(VRO*CONJG(RO))
        B=TG*VZ
        D=Y-B
        B2=1.-B*B
        A=(-D+SQRT(D*D+X*B2))/B2
      3 R01=RO+A*VRO
        ZO=Z+A*VZ
C*****EXIT IN THE CASE OF NARROWING CONE *****
      IF(ZO.GT.ZF)GO TO 1
C****RETURN IN THE CASE OF SPREADING CONE*****
      IF(ZO.LT.ZI)GO TO 2
      U=RANF(0)
C****PROBABILITY OF ABSORPTION*****
      IF(U.LT.0.0005)GO TO 4
      RO=R01$Z=ZO$NTR=NTR+1
      RAB=CABS(RO)
      RO=RO/RAB
      CALL ANGLE
      RO=RO*RAB
      IF(N-1)11,11,12
      11 IF(VZ)5,6,6
      12 IF(VZ)6,7,7
C****FLIGHT BETWEEN TWO POINTS ON THE WALL*****
      6 B=VZ*TG
        A=-2.*(REAL(RO*CONJG(VRO))-RAB*B)/(1.-B*B)
        GO TO 3
C****EXIT*****
      1 IN=1
C****NORMALISATION TO UNIT EXIT RADIUS*****
      Z=Z*CN$RO=RO*CN
      RETURN
C****RETURN*****
      2 IN=-1$RETURN
C****ABSORPTION*****
      4 IN=0$RETURN
C****RETURN IN THE CASE OF NARROWING CONE*****
      5 R2=CABS(RO+(ZI-Z)*VRO/VZ)
        IF(R2-1.)2,2,6
C*****EXIT IN THE CASE OF SPREADING CONE*****
      7 R2=CABS(RO+(ZF-Z)*VRO/VZ)*RI
        IF(R2-RF)1,1,9
      9 IF(N-1)10,10,6
      10 N=2$GO TO 8
      END
*****
```

```
      SUBROUTINE ANGLE
C***ANGLES OF FLIGHT AFTER REFLECTION FROM WALL
      COMMON G,CI,C/ZAKON/ZAKON(2)/LAW/LAW
CCOMMON/RV/Z,RO,VRO,VZ,IN,NTR/AL/AL(2)
      COMPLEX RO,VRO,CI,VROR
      VN1=REAL(VRO*CONJG(RO))*AL(1)+VZ*AL(2)
      U=RANF( )
      IF(LAW.EQ.ZAKON(2))GO TO 1
      COMP=U-1+G
      IF(COMP.LT.0.)GO TO 2
      U=RANF( )
      CS=SQRT(U)
      GO TO 3
      VN=VN1/SQRT(1.+VZ**2)
      COMP=U-1.+G*VN
      IF(COMP.LT.1.)GO TO 2
      U=RANF( )
      CS=U**(1./3.)
3      SN=SQRT(1.-CS**2)
      U=RANF( )
      FI=C*U
      SICOS=SN*CCS(FI)
      Q1=-CS*AL(1)-SICOS*AL(2)
      Q2=SN*SIN(FI)
      VROR=CMPLX(Q1,Q2)*RO
      VZ1=-CS*AL(2)+SICOS*AL(1)
      GO TO 4
2      VROR=VRO-2.*RO*VN1*AL(1)
      VZ1=VZ-2.*VN1*AL(2)
4      VRN=1./CABS(VROR)
      VRO=VROR*VRN
      VZ=VZ1*VRN
      RETURN
      END
```



```
      SUBROUTINE TURN(ALFA)
C***CONTROLOCKWISE ROTATION OF COORDINATE AXIS**
C***ALFA - ANGLE IN DEGREES*****
      COMMON/RV/Z,RO,VRO,VZ,IN,NTR
      COMPLEX RO,VRO,VR01
      BF=ALFA*6.2831852/360.
      CS=COS(BF)
      SN=SIN(BF)
      ROX=REAL(RO)
      RO=RO+ROX*(CS-1.)-Z*SN
      Z=ROX*SN+Z*CS
      VROX=REAL(VRO)
      VR01=VRO+VROX*(CS-1.)-VZ*SN
      VZ1=VROX*SN+VZ*CS
      VRN=1./CABS(VR01)
      VRO=VR01*VRN
      VZ=VZ1*VRN
      RETURN
      END
```

```
      SUBROUTINE SUMS(I)
      COMMON/RV/Z,RO,VRO,VZ,IN,NTR/A/A//G,CI,C
      COMMON/ZERC/NEUTR,NFZ(10),NANGLE(10,10)
* ,NRO(10,10),NAV(10)
      COMPLEX RO,VRO,RC1,CI
      NFZ(I)=NFZ(I)+1
      NAV(I)=NTR+NAV(I)
      EU=VZ**2/CT2=EU/(1.+EU)
      K=1+CT2*10.
      NANGLE(I,K)=NANGLE(I,K)+1
      RC1=RO+(A-Z)*VRO/VZ & ROF=CABS(RC1)
      K=1+ROF*ROF*10
1 NRO(I,K)=NRO(I,K)+1
      RETURN
      END
```

```
      SUBROUTINE RESULT
      COMMON/ZERO/NEUTR,NFZ(10),NANGLE(10,10)
*   ,NRO(10,10),NAV(10)//G,CI,C
      /ZED/ZED(10)/LAW/LAW/KMAX/KMAX,NMAX
      DIMENSION ANGLE(10,10),ANRO(10,10),DIV(10)
      DIMENSION CAV(10)
      COMPLEX CI
      PRINT 6,LAW
6   FORMAT(///* A(OMEGA)=*,A6)
      PRINT 8,G
8   FORMAT(///* G=*,F5.3///)
      PRINT 1
      DO 2 I=1,KMAX
      DIV(I)=1./NFZ(I)
      CAV(I)=NAV(I)*DIV(I)
C***PR-PROBABILITY OF TRANSMISSION *****
      PR=FLOAT(NFZ(I))/NMAX
C***SIG-ERROR OF PR RES-RESISTANCE*****
      SIG=SQRT(PR*(1.-PR)/NMAX)
      RES=1./PR-1.
C***SIGRES-ERROR OF RESISTANCE*****
      SIGRES=SIG/PR**2
      PRINT 3,ZED(I),NFZ(I),PR,SIG,RES,SIGRES
      DO 2 J=1,10
      ANGLE(I,J)=NANGLE(I,J)*DIV(I)
2   ANRO(I,J)=NRO(I,J)*DIV(I)
      PRINT 4
      K=KMAX
      PRINT 5,((ANGLE(I,J),J=1,10),I=1,K)
      PRINT 7
      PRINT 5,((ANRO(I,J),J=1,10),I=1,K)
      PRINT 10
10  FORMAT(///* AVERAGE COLLISION NUMBER*)
      PRINT 11,(CAV(I),I=1,K)
11  FORMAT(10(2X,F9.2))
1   FORMAT(*LENGTH*,2X,*NFZ*,2X,*TRANSMISSION*
*   ,* COEF.*,2X,*SIGMA*,2X,*RESISTANCE*,4X,*SIGMA*)
3   FORMAT(F6.2,1X,I5,6X,F6.4,9X,F6.4,
*1X,F8.4,4X,F8.4)
4   FORMAT(///// * ANGULAR DISTRIBUTION*)
      FORMAT(10(3X,F7.3))
5   FORMAT(///// * RADIAL DISTRIBUTION*)
      RETURN
      END
```

```
PROGRAM KOLEN(INPUT,OUTPUT)
COMMON G,CI,C/AL/AL(2)/ZED/ZED(10)/A/A
COMMON/ZERO/NEUTR,NFZ(10),NANGLE(10,10)
*,NRO(10,10),NAV(10)/KMAX/KMAX,NMAX
COMMON/RV/Z,RO,VRO,VZ,IN,NTR/X/X
COMPLEX RO,VRO,CI
DIMENSION GN(5)
DATA((GN(I),I=1,5)=.125,.25,.5,1.,0.)
EXTERNAL ZA,ZB,ZC,ZD
CALL READER
AL(1)=1. $ AL(2)=0.
BETA=45. $ X=1.
DO 10 I=1,2
DO 11 J=1,5
1 CALL ZERON
2 NEUTR=NEUTR+1
IF(NEUTR.GT.NMAX)GO TO5
CALL ORIGIN
6 A=ZED(1)
CALL RIGHT(ZA,ZB) $ IF(IN)2,2,7
7 Z=Z-ZED(1) $ CALL TURN(BETA) $ A=ZED(2)
CALL RIGHT(ZC,ZD) $ IF(IN)4,2,3
4 Z=Z+ZED(1) $ CALL TURN(-BETA) $ GO TO 6
3 CALL SUMS(1) $ GO TO 2
5 KMAX=1
CALL RESULT
11 G=GN(J)
10 READ13,LAW
13 FORMAT(A6)
STOP $ END
```

```
FUNCTION ZA(X)
COMMON/X/B/A/A
ZA=0. $ RETURN
ENTRY ZB
ZA=A-X*B $ RETURN
ENTRY ZC
ZA=X*B $ RETURN
ENTRY ZD
ZA=A $ RETURN
END
```