KfK 4133
Juni 1987

An Updated FORTRAN-77 Version of the 2-D Static Neutron Transport Code Diamant 2 for Regular Triangular Geometry

K. Küfner, J. Burkhard, R. Heger
Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE
Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

KfK 4133

AN UPDATED FORTRAN-77 VERSION OF THE 2-D STATIC NEUTRON TRANSPORT CODE DIAMANT2 FOR REGULAR TRIANGULAR GEOMETRY

Klaus Küfner, Jürgen Burkhard, Renate Heger

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

ABSTRACT

This report describes a major revision of the DIAMANT2 code, which evolved from the DIAMANT-code. DIAMANT2 solves the static multigroup neutron transport equation in planar geometry using the S_N-method. Spatial discretization is accomplished by taking finite differences on a meshgrid composed of equilateral triangles. This new version of DIAMANT2 is written in FORTRAN77 and designed to run on scalar and vector computers. This report contains a detailed documentation of the program and the input description as well as some experiences gained with the code on several vector computers.

EINE ÜBERARBEITETE FORTRAN-77 VERSION DES 2-D NEUTRONEN TRANSPORT PROGRAMMES DIAMANT2 FÜR REGULÄRE DREIECKSGEOMETRIE

ZUSAMMENFASSUNG

Der Bericht beschreibt eine gründlich überarbeitete Version des DIAMANT2 Programms, das aus dem DIAMANT Programm entstand. DIAMANT2 löst die stationäre Multigruppen-Neutronen-Transport-Gleichung in ebener Geometrie mit Hilfe der S_N-Methode. Die räumliche Diskretisierung erfolgt über einem Gitter aus gleichseitigen Dreiecken. Diese neue Version von DIAMANT2 ist in FORTRAN-77 geschrieben und wurde speziell angelegt, um sowohl auf Skalar- als auch auf Vektorrechnern zu laufen. Der Bericht enthält eine ausführliche Programmdokumentation und die Eingabebeschreibung sowie Erfahrungen mit dem Einsatz des Codes auf mehreren Vektorrechnern.

TABLE OF CONTENTS

Chapter 1. Introduction 1

Chapter 2. Difference Equations and Solution Algorithm 3
2.1 Analytic Form of the Static Neutron Transport Equation 3
2.1.1 Divergence Operator 3
2.1.2 Spherical Harmonics Expansion of the Scattering Source Terms 5
2.2 Multigroup Equations 6
2.3 Discrete Ordinates Approach 7
2.3.1 S_N-Constants and Angular Orientation 8
2.3.2 Construction of S_N-Constants 8
2.3.3 Boundary Conditions 10
2.3.3.1 Vacuum Boundary Condition 10
2.3.3.2 Reflective Boundary Conditions 10
2.3.3.3 Boundary Fluxes 10
2.4 Difference Equations 11
2.5 Inner Iterations 13
2.5.1 Diamond Difference Approximation 13
2.5.2 Triangles of Orientation 1 13
2.5.3 Triangles of Orientation 2 15
2.5.4 Negative Flux Fix-Up 16
2.5.5 Progression Through the Space-Angle Mesh 18
2.5.6 Inner Convergence Tests and Convergence Acceleration . . . 19
2.6 Outer Iteration Procedure 21
2.7 Other Code Options 23
2.7.1 Adjoint Calculations 23
2.7.2 Inhomogeneous Calculations 24
2.7.3 Evaluations and Balances 25

Chapter 3. Implementation of DIAMANT2 29
3.1 Program Flow Charts 29
3.1.1 Simplified Call Structure Diagram for DIAMANT2 30
3.1.2 Flow Diagram for Subroutine DRIVER 33
3.1.3 Flow Diagram for Subroutine DIAMAN 34
3.1.4 Flow Diagram for Subroutine OUTER 36
3.1.5 Flow Diagram for Subroutine TRINER 38
3.2 Function of the Different Subroutines 41

3.3 Data Management, COMMON Usage and Data Files 45
3.3.1 Distribution of Working Arrays 45
3.3.2 Use of COMMON Blocks 52
3.3.3 Data Files Used 63
3.3.4 Storage Optimization Levels 65
3.4 Interfaces to Other Programs 65
3.4.1 Structure of Restart Dump Files 65
3.4.2 Structure of Interface File for Perturbation Module 66
3.5 KAPROS Implementation 69
3.6 Use as Stand-alone program 72
3.6.1 General Remarks 72
3.6.2 Subroutine EXTEND 72
3.6.3 Subroutine FILLC 73
3.6.4 Subroutine FRESP1 74
3.6.5 Function TASKTI 74
3.6.6 Function TIMLIM 74
3.6.7 Subroutine KSCC 75
3.6.8 Subroutine KSINIT 75
3.6.9 Subroutine READK0 75

Chapter 4. Test Examples and User Information 77
4.1 Test Calculations Carried Out 77
4.2 Input Test Module DIAPRD 78
4.3 Experience From the Test Calculations 78
4.3.1 How to Choose the Quadrature Set 78
4.3.2 How to Choose Iteration Control Parameters 79
4.3.3 How to Estimate Computing Time 80
4.3.4 How to Estimate Storage Needs 81
4.3.5 Discretization Accuracy 82
4.3.6 The Role of Negative Flux Fix-Up 82
4.4 Experiences on Vector Computers 83
4.5 How to Choose PARAMETER Values 86
4.6 Tuning Binary Input/Ouput 89

Chapter 5. KAPROS Short Description of Module DIAMANT2 91

Chapter 6. Input Description 95
6.1 General Remarks 95
6.2 Input Card Sequence 96
6.3 Remarks and Explanations for Some Input Options 106

6.3.1 Mixture Distribution 106
6.3.2 Nuclear Cross Sections 106
6.3.2.1 Nuclear Cross Section Input by SIGMN Block 107
6.3.2.2 Card Input of Nuclear Cross Sections 110
6.3.3 How to Specify Anisotropic Mixtures 110
6.3.4 External Sources 112
6.3.5 Boundary Fluxes 113
6.3.6 Miscellaneous 114

Chapter 7. Sample Output Interpretation 115

Appendix A. DIAMANT2 Built-in Quadrature Constants 119
A.1 Defining Equations of the Quadrature Sets 119
A.2 P_N Quadrature Constants 121
A.3 DP_N Quadrature Constants 122
A.4 EP_N Quadrature Constants 123

Appendix B. Example of a DIAMANT2 Calculation 125
B.1 Job Control and Input Data Cards 125
B.1.1 Stand-alone Version 125
B.1.2 KAPROS Version 127
B.2 Output List of the Sample Problem 129

Appendix C. DIAMANT2 Stop Codes 147

Acknowledgments 149

References 151

LIST OF ILLUSTRATIONS

Figure 1. Coordinate System Used in DIAMANT2 4
Figure 2. Construction of Quadrature Constants 9
Figure 3. Coefficients of the Differential Equation 13
Figure 4. Possible Orientations of a Triangular Cell 14
Figure 5. Sequence of Directions in an S4 Calculation 16
Figure 6. Space-Angle-Mesh-Sweep Ordering 17
Figure 7. Computational Mesh Sequence for Fixed Direction 18
Figure 8. Possible Orientations of the System Boundary 26
Figure 9. Cross Reference Table for Subroutine Calls 31
Figure 10. Array Allocation in A(NACMAX) resp. DBN=DIAM2ƀREAL*4 . . 46
Figure 11. Usage of Arrays BIL1,BIL2,BIL3 48
Figure 12. Array Allocation in IW(NWMAX) resp. DBN=DIAM2ƀINTEGER*4 . 49
Figure 13. Use of COMMON Blocks by Subroutines 50
Figure 14. Usage of COMMON /COMARR/ 53
Figure 15. Usage of COMMON /COMBND/ and COMMON/COMCHA/ 54
Figure 16. Usage of COMMON /COMLOG/ 54
Figure 17. Usage of COMMON /COMINT/ 55
Figure 18. Usage of COMMON /COMPOI/ 60
Figure 19. Usage of COMMON /COMREA/ 61
Figure 20. Usage of COMMON /COMSTO/ and BLANK COMMON 62
Figure 21. Usage of COMMON /FILLCO/ and COMMON /FILLCX/ 62
Figure 22. Data Files und Their Use 64
Figure 23. Structure of the Label Record in Restart Dump Files . . . 67
Figure 24. Recommended Content of DBN = INITƀREADƀDIAMAN, IND = 1 . 71
Figure 25. Dependence of DIAMANT2 on KAPROS 73
Figure 26. Examples for Storage, CPU Times and Execution Speed . . . 81
Figure 27. CPU Times for Some Vector Computers (Isotropic Scattering) 84
Figure 28. CPU Times for Some Vector Computers (Anisotropic Scattering) 85
Figure 29. Definition of PARAMETER Values Used in DIAMANT2 87
Figure 30. PARAMETER Values Used on Different Computers 88
Figure 31. Basic Input Zones ITYP 101
Figure 32. Convexity of the Model Boundary 107
Figure 33. Specifying the Model in the Reference Parallelogram . . . 108
Figure 34. Storage Sequence in Cross Section Array C 109
Figure 35. Example of an External Source Input 111
Figure 36. Example of a Boundary Flux Input 113

CHAPTER 1. INTRODUCTION

DIAMANT2 (release 2.0) is an improved version of the DIAMANT /1/, /2/ program for two-dimensional neutron transport calculations in planar geometry using a regular triangular mesh. The program solves both regular and adjoint, homogeneous or inhomogeneous, time-independent problems subject to vacuum, reflective or input specified boundary flux conditions.

DIAMANT2 is a fully FORTRAN77 conforming code designed to run on many different computers as efficiently as possible. It successfully solved a set of standard test problems on IBM, FACOM, CYBER-205, CRAY and SIEMENS VP series computers. Many functional improvements have been incorporated:

1. Anisotropic scattering option now operational;
2. New option to create interface files for use in the perturbation module TPTRIA/3/;
3. Use of an effective absorption implied by the input cross sections;
4. New built-in quadrature sets and higher S_N orders;
5. Improved inner-outer iteration strategy;
6. Several convergence criteria as user option;
7. Clean-up of the coding (FORTRAN77) and restructuring;
8. Elimination of installation dependent routines;
9. Centralization of binary input/output operations;
10. Re-organization of computer memory usage;
11. Complete input control print is provided prior to execution;
12. Special care has been taken to allow efficient execution on vector computers.

Details of these features as well as a complete description of all options of DIAMANT2 are given in the following chapters. In the next chapter, difference equations and solution algorithms are discussed. Chapter 3 is a guide for implementation and chapter 4 contains information relevant to the user of DIAMANT2. Chapter 5 is the KAPROS/4/ short description of the module. The input description is contained in Chapter 6. The last chapter gives a complete example of a DIAMANT2 run along with an output description.

Part of this document (especially Chapter 2) has been influenced by the TWOTRAN/5/ manual.

CHAPTER 2. DIFFERENCE EQUATIONS AND SOLUTION ALGORITHM

2.1 ANALYTIC FORM OF THE STATIC NEUTRON TRANSPORT EQUATION

The time independent neutron transport equation is written as

$$\begin{aligned}&\vec{\Omega}.\nabla\psi(\vec{r},E,\vec{\Omega}) + \sigma_t\psi(\vec{r},E,\vec{\Omega}) = \\ &\int dE' \int d\vec{\Omega}'\psi(\vec{r},E',\vec{\Omega}')\sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega}) \\ &+ \chi(E)\int dE' \int d\vec{\Omega}'\psi(\vec{r},E',\vec{\Omega}')\nu\sigma_f(\vec{r},E')/(4\pi) + Q(\vec{r},E,\vec{\Omega})\end{aligned} \tag{1}$$

in which ψ is the particle flux (number density of particles times their speed) defined such that $\psi dV dE d\vec{\Omega}$ is the flux of particles in the volume element dV about $\vec{r}$, in the element of solid angle $d\vec{\Omega}$ about $\vec{\Omega}$, in the energy range dE about E. Similarly, $Q dV dE d\vec{\Omega}$ is the number of particles in the same element of phase space emitted by sources independent of ψ. The macroscopic total interaction cross section is denoted by σ_t, the macroscopic scattering transfer probability (from energy E' to E and from angle $\vec{\Omega}'$ to angle $\vec{\Omega}$) by σ_s, and the macroscopic fission cross section by σ_f. (all these quantities depend on $\vec{r}$, but we have omitted this argument for simplicity). The number of particles emitted isotropically ($1/4\pi$) per fission is ν, and the fraction of those liberated in the range dE about E is $\chi(E)$. This fraction may actually depend on both, the material undergoing fission and the energy of the fission inducing neutron, but such possibilities are not admitted in DIAMANT2. The influence of that effect may be estimated by using the perturbation theory module TPTRIA /3/.

2.1.1 Divergence Operator

The form of $\vec{\Omega}.\nabla\psi$ for the geometry treated by DIAMANT2 is given in terms of the variables defined in the coordinate system sketched in Figure 1 on page 4 as:

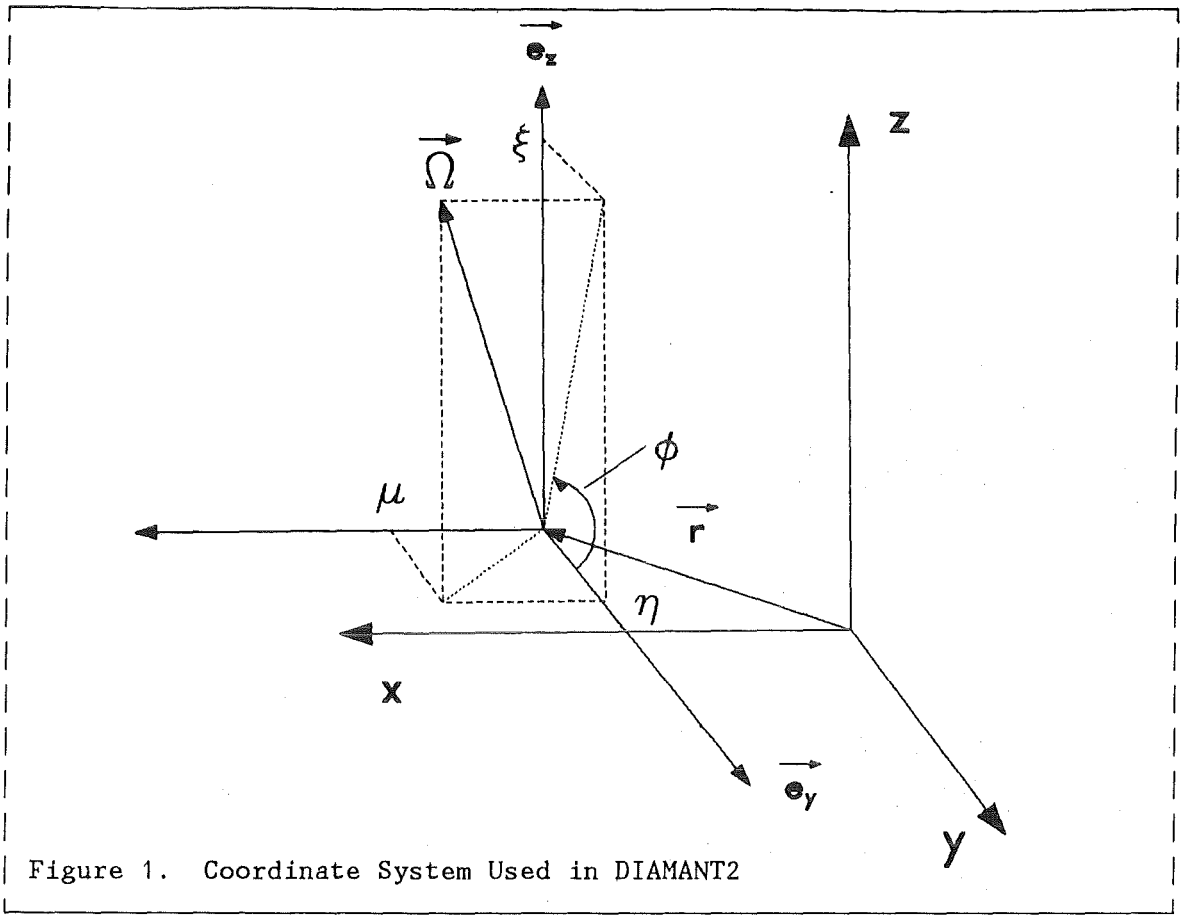


Figure 1. Coordinate System Used in DIAMANT2

Dependence of ψ : $\psi(x,y,\mu,\eta,E)$

Definition of Variables : $\mu = \vec{e}_x.\vec{\Omega}$, $\eta = \vec{e}_y.\vec{\Omega}$

Streaming operator : $\vec{\Omega}.\nabla\psi = (\mu\partial/\partial x + \eta\partial/\partial y)\ \psi$

ϕ is the angle of rotation about the μ axis such that $d\vec{\Omega}$ can be replaced by $d\mu d\phi$, and

$$(2)\quad \begin{aligned} \xi &= \sqrt{(1 - \mu^2)}\ \sin\phi \\ \eta &= \sqrt{(1 - \mu^2)}\ \cos\phi \\ \mu &= \sqrt{(1 - \eta^2 - \xi^2)} \end{aligned}$$

ψ is assumed to be symmetric in ϕ, and only a hemisphere of angular directions needs to be considered. The north pole of the hemisphere points in z-direction.

2.1.2 Spherical Harmonics Expansion of the Scattering Source Terms

The scattering kernel σ_s is assumed to be representable by a finite Legendre polynomial expansion (implicitly assuming that the scattering $\vec{\Omega}'\to\vec{\Omega}$ is dependent on the cosine $\vec{\Omega}'.\vec{\Omega}$ only) :

$$\sigma_s(E'\to E,\mu_0) = 1/(4\pi) \sum_{k=0}^{ISCT} P_k(\mu_0)\sigma_{sk}(E'\to E) \quad , \quad (\mu_0 = \vec{\Omega}.\vec{\Omega}')$$

If this expansion is inserted in Eq. (1), and if the addition theorem is used to expand $P_k(\mu_0)$, we can write

$$(3)\quad \int dE' \int d\vec{\Omega}' \psi(r,E',\vec{\Omega}') \; \sigma_s(E'\to E,\mu_0) =$$

$$\int dE' \sum_{k=0}^{ISC} T_k^* \sigma_{sk}(E'\to E) \sum_{\ell=0}^{k} T_k^\ell(\mu,\phi) \int_{-1}^{1} d\zeta \int_0^{\pi} d\gamma \; T_k^\ell(\zeta,\gamma)\psi(\zeta,\gamma,E'),$$

where $\sigma^*_{sk} = (2k+1)/(4\pi)\sigma_{sk}$.

In deriving this expression, the ϕ symmetry of ψ is used, reducing the domain of $\vec{\Omega}$ to a hemisphere and eliminating expansion terms odd in ϕ. The functions T_k^ℓ are defined by

$$(4)\quad T_k^\ell(\mu,\phi)=[(2-\delta_{\ell k})(k-\ell)!/(\ell+k)!] \; P_k^\ell(\mu)\cos k\phi \quad ,$$

where $\delta_{\ell k}$ is the Kronecker delta (equal to 1 if $\ell = k$, and vanishing otherwise) and the P_k^ℓ are the associated Legendre polynomials.

Hence, if the angular flux is expanded into a series of these functions,

$$(6)\quad \psi = \sum_{k=0}^{ISCT} (2k+1) \sum_{\ell=0}^{k} T_k^\ell \; \psi_k^\ell,$$

the expansion coefficients are given by

$$(7)\quad \psi_k^\ell = \int_{-1}^{1} d\zeta \int_0^{\pi} d\gamma \; T_k^\ell \; \psi(\zeta,\gamma) \; / \; (2\pi)$$

Using this formula, we can rewrite Eq. (1) as

$$(8)\quad \vec{\Omega}.\vec{\nabla}\psi + \sigma_t\psi = \int dE' \sum_{k=0}^{ISCT} (2k+1)\,\sigma_{sk}(E'\to E) \sum_{\ell=0}^{k} T_k^{\ell}(\mu,\phi)\psi_k^{\ell}$$

$$+ \chi(E) \int dE'\, \nu\sigma_f(E')\psi_0^0 + Q$$

As implied by this equation, we have assumed that the source Q is isotropic, e.g., not dependent on the angular variable.

2.2 MULTIGROUP EQUATIONS

The energy domain of interest is assumed to be partitioned into IGM intervals of width ΔE_g, g = 1, 2, ..., IGM. By convention, increasing g represents decreasing energy. If we multiply Eq. (8) by ΔE_g and integrate we can write

$$(9)\quad \vec{\Omega}.\vec{\nabla}\psi_g + \sigma_{t,g}\psi_g = \sum_{h=1}^{IGM} \sum_{k=0}^{ISCT} (2k+1)\sigma_{s,k,h\to g} \sum_{\ell=0}^{k} T_k^{\ell}\psi_{k,h}^{\ell}$$

$$+ \chi_g \sum_{h=1}^{IGM} \nu\sigma_{fh}\psi_{0,h}^0 + Q_g \quad , \; g=1,\ldots,IGM$$

Here, the flux, ψ_g, as well as the fission spectrum, χ_g, for group g

$$(10)\quad \psi_g = \int_{\Delta E_g} dE\ \psi \ , \quad \chi_g = \int_{\Delta E_g} dE\ \chi(E)$$

are no longer distributions in energy, but refer to the total number of particles in the energy interval. For this reason, if group structures are changed, the effect on results must be evaluated by comparing $\psi_g/\Delta E_g$. Because of Eq. (10), energy integrals in DIAMANT2 are evaluated by simple sums.

Cross sections subscripted with g are weighted averages in energy group g, e.g.,

$$(11a)\quad \sigma_{tg} = \left(\int_{\Delta E_g} dE\ \sigma_t \psi \right) / \left(\int_{\Delta E_g} dE\ \psi \right)$$

The transfer cross sections $\sigma_{s,k,h\to g}$ are flux moments weighted averages in group h and integrals over the inscatter group g :

$$(11b)\quad \sigma_{s,k,h\to g} = \left(\int_{\Delta E_g} dE \int_{\Delta E_h} dE'\ \sigma_{s,k}(E'\to E)\ \sum_{\ell} \psi_k^{\ell}(E') T_k^{\ell} \right) / \int_{\Delta E_h} dE'\ \sum_{\ell} \psi_k^{\ell}(E') T_k^{\ell}$$

Of course, ψ is not known in advance and must be approximated by some means. This has to be done in a separate code. If in Eq. (11) the angular dependence of ψ is nonseparable, then σ_{tg} will in principle be angle-dependent. No provision for such dependence is made in DIAMANT2.

Within the code only downscattering problems can be solved which means that the first summation in Eq. (9) does not extend to h=IGM but to h=g only. Equation (9) is solved using a conventional inner-outer iteration procedure as described in the following text.

2.3 DISCRETE ORDINATES APPROACH

To approximate the angular dependence, we select JMM directions on the unit sphere and require Eq.(9) to hold only for these directions. Omitting the group index, we end up with a coupled system of equations

$$(12)\quad \vec{\Omega}_m \cdot \nabla \psi_m + \sigma_t \psi_m = S_m,\ \psi_m = \psi(r,\vec{\Omega}_m)\ ,\ S_m = S(r,\vec{\Omega}_m)$$

The scalar, i.e. angle-integrated, flux is approximated by a quadrature formula

$$(13)\quad \phi(\vec{r}) = \sum_{m=1}^{JMM} w_m\ \psi_m(\vec{r})$$

where the w_m are the integration weights associated with direction $\vec{\Omega}_m$, and JMM is the total number of directions used.

2.3.1 S_N-Constants and Angular Orientation

For a quadrature order ISN, DIAMANT2 requires JMM = ISN*(ISN+2)*3/4+ISN pairs (μ,η) of direction cosines. These are, for ISN up to 12, provided by the subroutine SNCONS (the upper limit 12 is fixed by a PARAMETER statement). The code actually uses only JMM-ISN directions, the remaining directions, having an integration weight zero, are not used by DIAMANT2 and are included only for historical reasons.

2.3.2 Construction of S_N-Constants

S_N-constants are built-in in the code. Presently, a user may choose by input among three different sets : P_N, DP_N and EP_N. Some hints, as to which one of them should be used in a particular application, are given in "How to Choose the Quadrature Set" on page 78. The constants used are described in "Appendix A. DIAMANT2 Built-in Quadrature Constants" on page 119. The construction itself consists of the following steps:

1. We subdivide the upper unit hemisphere into six sixty degree sectors (called "dodecants").
2. For each dodecant we select n=ISN/2 level lines (where ISN is the quadrature order selected and is supposed to be an even number).
3. Each level line is assigned a value of ξ; the ξ-value is taken from a 1d quadrature set (presently possible: P_N, DP_N and EP_N set) as well as the associated integration weight.
4. To calculate μ and η according to Eq. (2), we still have to define the angle Φ. In DIAMANT2 we request:

   (14) $\Phi_j = (2j-1)\ \pi\ /\ 6k,\ j=1,\ldots,k \qquad k=1,\ldots,ISN/2$

   (see Figure 2 on page 9).
5. The integration weight associated with a certain level is distributed uniformly among all angular points on that level.

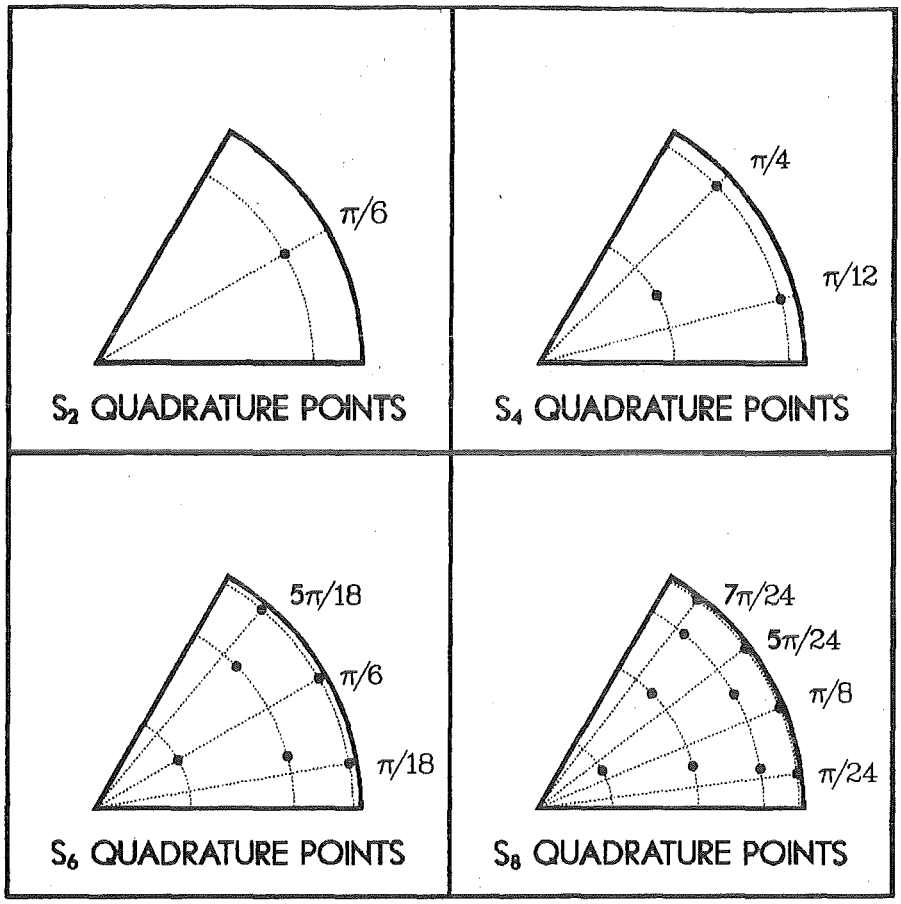


Figure 2. Construction of Quadrature Constants

Several desirable relationships between the projections μ_m and η_m of $\vec{\Omega}_m$ on the coordinate planes, and the integration weights w_m are fulfilled for the built-in quadrature sets (all sums extend from m=1,...,JMM):

(15)

(i) Normalization: $\Sigma\, w_m = 1$

(ii) Equilibrium: $\Sigma\, w_m\mu_m = \Sigma\, w_m\eta_m = 0$

(iii) Diffusion consistency: $\Sigma\, w_m\mu^2{}_m = \Sigma\, w_m\eta^2{}_m = 1/3$

2.3.3 Boundary Conditions

Information about incoming boundary fluxes ($\vec{\Omega}.\vec{n} < 0$) can be specified by the DIAMANT2 user . Details on the input specification are given in "Boundary Fluxes" on page 112. One may select any meaningful combination of the three following options for the geometrical solution domain but on each boundary mesh line only one type of boundary condition can be applied.

2.3.3.1 Vacuum Boundary Condition

The value of the angular flux ψ on such a boundary is set to zero for all incoming directions.

2.3.3.2 Reflective Boundary Conditions

The value of the flux on such a boundary for incoming directions is set equal to the value of the outgoing flux in the reflective direction.

2.3.3.3 Boundary Fluxes

On any side of the solution domain, the user may specify isotropic boundary fluxes. These inflows represent a source of particles and are treated as such by the code. Use of this option implies an inhomogeneous calculation.

2.4 DIFFERENCE EQUATIONS

To discuss the spatial finite difference representation, we write Eq. (9) for a single energy group and a single direction as

$$(16)\quad \vec{\Omega}.\vec{\nabla}\psi + \sigma_t\psi = S \ ,$$

omitting the group and the direction subscript. On the right-hand side of eq. (9) the integral

$$(17)\quad \psi^{\ell}_{k,h} = \int_{-1}^{1} d\mu \int_{0}^{\pi} d\phi \ T^{\ell}_{k}(\mu,\phi)\psi_h/2\pi \ .$$

is approximated for each group h by the sum

$$(18)\quad TF^{\ell}_{k,h,i,j} = \sum_{m=1}^{JMM} w_m T^{\ell}_{k}(\mu_m,\phi_m)FLXANG_{h,i,j,m}$$

where JMM is the total number of directions, ϕ_m is defined by the program as

$$(19a)\quad \phi_m = \tan^{-1}\left(\sqrt{(1 - \mu_m^2 - \eta_m^2)} \ / \ \eta_m\right) \qquad \text{if} \quad \eta_m > 0 \ ,$$

$$(19b)\quad \phi_m = \tan^{-1}\left(\sqrt{(1 - \mu_m^2 - \eta_m^2)} \ / \ \eta_m\right) + \pi \qquad \text{if} \quad \eta_m < 0 \ .$$

and $TF^{\ell}_{k,h,i,j}$ is the value of $\psi^{\ell}_{k,h}$ in spatial cell (i,j).

In the code $TF^{0}_{0,h}$ is identified with the scalar flux $FLXSKA_h$.
The right-hand side of Eq.(16), $S = S_{g,i,j,m}$, is given by the sum of the following three contributions:

$$(20)\quad (SS)_{g,i,j,m} = (\text{Scatter Source})_{g,i,j,m} =$$
$$\sum_{h=1}^{g} \sum_{k=0}^{ISCT} (2k+1)\sigma_{s,k,h\to g} \sum_{\ell=0}^{k} T^{\ell}_{k,m} TF^{\ell}_{k,h,i,j} \ ,$$

$$(21)\quad (FG)_{g,i,j} = (\text{Fission Source})_{g,i,j} = \chi_g \sum_{h=1}^{IGM} \nu\sigma_{f,h} \ TF^{0}_{0,h,i,j}$$

(22) $(\text{EXTSOU})_{g,i,j} = (\text{Inhomogeneous Source})_{g,i,j} = Q_{g,i,j}$.

Note that no angular dependence is allowed in Eqs. (21) and (22). Eq. (20) indicates that only downscattering problems are considered since the summation index h extends only to g (index of the actual energy group).

Apparently the double summations over k and ℓ in the Legendre expansion can be somewhat simplified. In DIAMANT2 the arrays T^{ℓ}_{k} and ψ^{ℓ}_{k} are stored with their non-zero contribution (both are lower triangular matrices with respect to those indices). Moreover, the factor (2k+1) is expected to be contained in the cross-section tables $\sigma_{s,k}$.

The neutron flux in a fixed direction is representative for the average angular flux of the associated angular interval. Integration over each homogeneous basic triangle[1] and use of the divergence theorem gives the discretized form (four-point formula):

(23) $A_1 h\psi_1 + A_2 h\psi_2 + A_3 h\psi_3 + V\sigma_t\psi_0 = VS_0$,

where $A_i = \vec{\Omega}.\vec{n}_i$ (see Figure 3 on page 13), $\vec{n}_i$ being the outer normal to triangle face i, and h and V are length and area of the (equilateral) basic triangle, respectively. The angular flux values in Eq.(23) are defined as

(24) $\psi_i = \int_{si} ds\ \psi / h$, $\psi_0 = \int_V dV\ \psi / V$, $S_0 = \int_V dV\ S / V$

ψ_i and ψ_0 are associated with the edges and the center of mass of each triangle, respectively. The indices s_i (i=1,2,3) identify the three faces of the triangle.

2.5 INNER ITERATIONS

---

[1] At some places these basic equilateral triangles are referred to as "meshes" in the text.

c	A_1	A_2	A_3
$\sqrt{(1-\xi^2)}$	$c*\cos(\pi/6-\Phi)$	$c*\cos(\pi/2-\Phi)$	$c*\cos(\Phi+\pi/6)$

ξ defined by Eq. (2), Φ defined by Eq. (14).

Figure 3. Coefficients of the Differential Equation

2.5.1 Diamond Difference Approximation

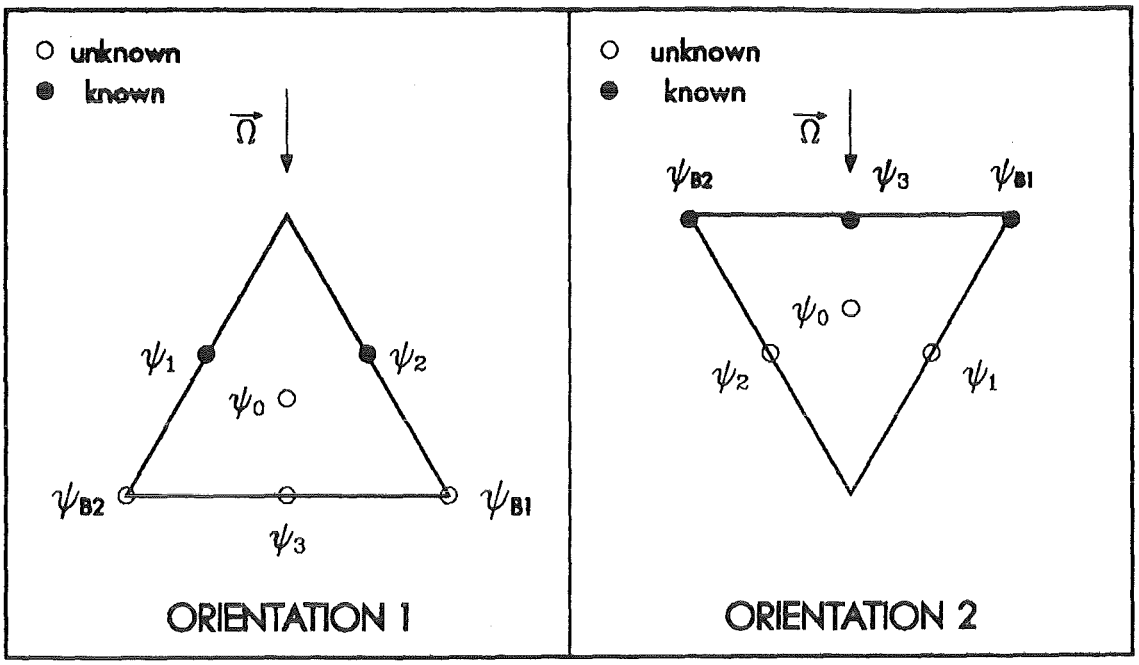
For a fixed angular direction $\vec{\Omega}$, there are two possible orientations of the basic triangles (see Figure 4 on page 14).

2.5.2 Triangles of Orientation 1

For triangles of orientation 1 with respect to the fixed direction, the edge fluxes ψ_1 and ψ_2 are known from boundary conditions or from previous steps (the vertex fluxes ψ_{B1} and ψ_{B2} in Figure 4 on page 14 are needed by triangles of orientation 2 and are defined later). A diamond relation, assuming linearly varying angular fluxes over each basic triangle,

$$\psi_1 + \psi_2 + \psi_3 = 3\psi_0 \tag{25}$$

together with the balance Eq.(23), is therefore sufficient to calculate the unknown box value ψ_0 and the remaining edge value ψ_3; ψ_0 is expressed in terms of the two known values ψ_1 and ψ_2. ψ_3 is calculated using Eq.(25) which implies an extrapolation. For use in the solution step for triangles of orientation 2 we calculate in addition the flux gradient along the "outgoing" side by Eq. (26d).



The arrow at the top of the figure indicates the assumed direction of neutron travelling.

Figure 4. Possible Orientations of a Triangular Cell

(26a) $\psi_0 = (VV*S_0 + (A_3-A_1)*\psi_1 + (A_3-A_2)*\psi_2)/NENN$

(26b) $NENN = 3*A_3 + STOT*VV$ $\qquad VV = V/h$

(26c) $\psi_3 = 3*\psi_0 - \psi_1 - \psi_2$

(26d) $\psi_{B1} - \psi_{B2} = 2(\psi_2 - \psi_1)$

This approximation is locally $O(h^2)$ accurate since only the linearity assumption, Eq.(25) is needed. ("locally $O(h^2)$" means that for small mesh steps h, the magnitude of the (local) truncation error over each basic triangle is proportional to a term of order h^2). Use of a fix-up may be required for certain mesh cells since the extrapolation using Eq.(25) may possibly yield physically unreasonable negative flux values.

2.5.3 Triangles of Orientation 2

For triangles of orientation 2, only ψ_3 of the three edge values is known. To calculate the remaining two, ψ_1, ψ_2, as well as ψ_0, one further relation independent of Eqs.(23) and (25) is needed.

Walters et al. /6/ describe an effective scheme using the flux values at the corners and the centers of the triangle sides (seven-point formulation). Using the basic idea of this scheme for DIAMANT2 requires that the four-point formulas, Eqs. (23) and (25), are extended; instead of using only the centered values on the triangle sides, flux gradients along the triangle sides are calculated as well. For orientation 1 triangles, the auxiliary value $\psi_{B1}-\psi_{B2}$ (see Figure 4 on page 14), is calculated and temporarily stored (thereby using the assumption of linearly varying flux over each triangle cell). Then, for orientation 2 triangles, ψ_3 is known as well as the flux gradient along triangle face s_3 - and therefore it is possible to calculate ψ_1, ψ_2, and ψ_0 using Eqs. (23), (25) and (26) and the linearity of fluxes over the triangle. Use of a fix-up may be required since the extrapolations, Eqs.(25) and (26), may possibly yield physically unreasonable negative flux values.

(27a) $\psi_0 = (VV*S_0 + ((A_1 + A_2)/2 - A_3)\ \psi_3 + (A_1 - A_2)/2(\psi_{B1} - \psi_{B2})) \ /\ NENN$

(27b) $NENN = 3*(A_1+A_2)/2 + STOT*VV \quad , \quad VV = V/h$

(28a) $\psi_1 = (3*\psi_0 - \psi_3 + \psi_{B1} - \psi_{B2})/2$

(28b) $\psi_2 = \psi_1 - \psi_{B1} + \psi_{B2}$

2.5.4 Negative Flux Fix-Up

As mentioned before, Eqs.(26) to (28) may sometimes result in negative angular fluxes. As these are physically unreasonable, they are eliminated by switching to a (positive) step scheme. In any case ψ_0 is always positive

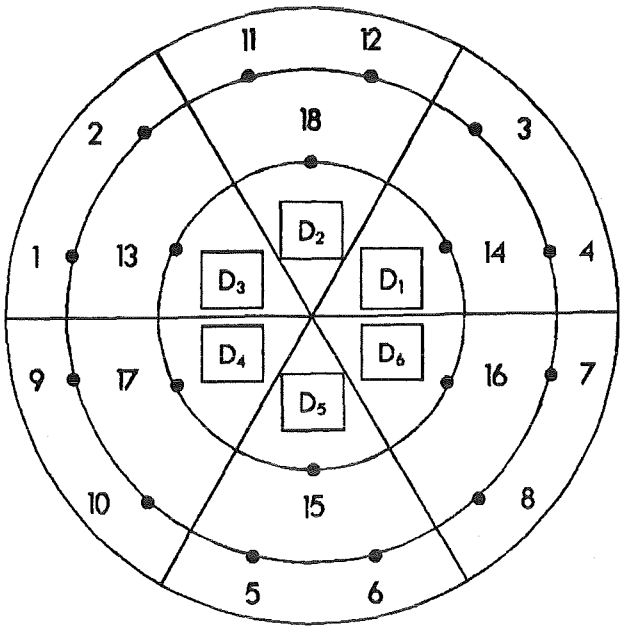


Figure 5. Sequence of Directions in an S4 Calculation

if cross-sections and sources are positive. If ψ_3 is negative in Eq.(26) we use the relation

(29) $\psi_3 = \psi_0$

instead of Eq.(25). Then we have

(30) $\psi_0 = (VV*S_0 - A_1\psi_1 - A_2\psi_2) / (A_3 + STOT*VV)$

For meshes of orientation 2 either ψ_1 or ψ_2 may be negative. In those cases we assume

(31) $\psi_1 = \psi_2 = \psi_0$

which results in

(32) $\psi_0 = (VV*S_0 - A_3\psi_3) / (A_1 + A_2 + STOT*VV)$.

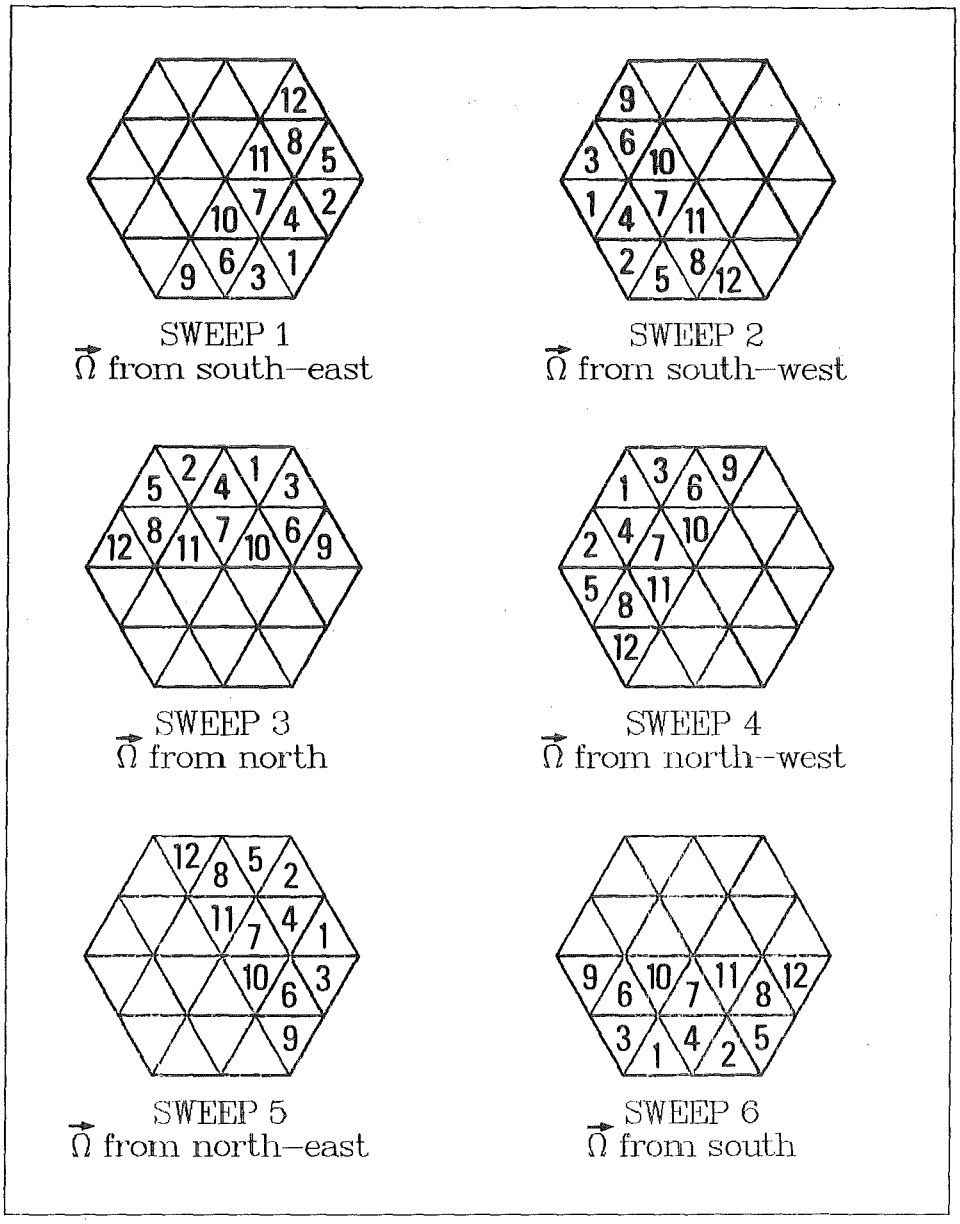


Figure 6. Space-Angle-Mesh-Sweep Ordering

Since it has been observed that using the fix-up slows down the code, we switch off the fix-up (i.e., we allow for a certain time negative fluxes) until outer convergence is met. After that we switch on the fix-up procedure and continue iterating until convergence is achieved again. In our applications only a few additional iterations (1 to 5) have been necessary for convergence to the final solution.

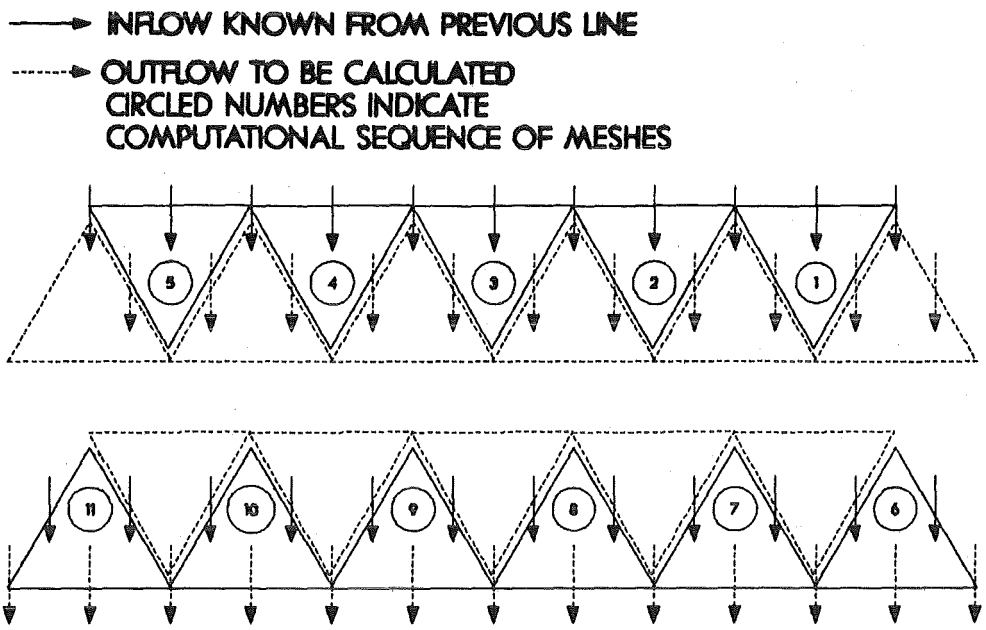


Figure 7. Computational Mesh Sequence for Fixed Direction

2.5.5 Progression Through the Space-Angle Mesh

The unknown ψ are ordered so that the difference scheme is stable and so that the coefficient matrix is lower triangular. Physically this corresponds to proceeding in the direction of particle flow. Organization of the inner iterations is based upon division of the discrete directions into dodecants (see also "Flow Diagram for Subroutine TRINER" on page 38). The following nested loops are executed:

- The outermost loop (variable K) runs over the levels from 1 to ISN/2; (ISN = order of the angular discretization).

- The next loop (variable KT) runs over the dodecants in the sequence 3-1-5-6-4-2 (see Figure 5 and Figure 6).

- The next loop (variable K2) sweeps over all directions on level K in dodecant KT.

- The next to last loop (variable I) runs through the "line" of the discretization grid (the definition of a "line" is dependent upon the dodecant in which the directions are being considered , see Figure 6 on page 17 and Figure 7 on page 18).

- The last loop (variable J) finally processes the individual triangles of each line (see Figure 7 on page 18).

We thus arrive at the sequence of space-angle sweeps shown in Figure 6 on page 17. The control information required for selection of the next direction and the next triangle is kept to a minimum. Moreover, the innermost loop solves Eq. (23) for all meshes on a line simultaneously. This is important with respect to the vectorization potential of the DIAMANT2 code (see "Experiences on Vector Computers" on page 83).

2.5.6 Inner Convergence Tests and Convergence Acceleration

Depending on input specification, there are two criteria to test convergence of inner iterations. Both of them test groupwise the deviation of scalar fluxes during two successive inner iterations.

$$\varepsilon_{inner} = \Sigma \left| \phi_0^{old} - \phi_0^{new} \right| / \Sigma \phi_0^{new} \tag{33}$$

is called "integral" test and is used whenever the input variable EPSA is less then zero. The summation extends over all mesh cells. Note that for regular triangular geometry (constant mesh volume) the summation is equivalent to a spatial integration.

$$\varepsilon_{inner} = \max \left| 1.0 - \phi_0^{old} / \phi_0^{new} \right| \tag{34}$$

is called "point-wise" test and is used whenever EPSA is greater or equal to zero. The maximum is taken over all mesh cells. Note that this criterion puts emphasis on the local behaviour of scalar fluxes. Test (34) is much more stringent than (33).

Inner iterations are stopped if $\varepsilon_{inner} \leq |$ EPSINN $|$. The value of EPSINN depends on the accuracy ERROUT reached in the previous outer iteration. In order to avoid too many inner iterations during the initial phase of outer iterations we set:

EPSINN = max (min(0.1*ERROUT,EPSINN) , |EPSA|)

after each outer iteration. For the first outer iteration, EPSINN is initialized to 1.0E-1 (1.0E-3 if a restart dump is being used).

This special prescription of inner iteration accuracy has been chosen in order to give a more smoothly eigenvalue convergence. It is guaranteed that the flux error is non-increasing. At the cost of more outer iterations, the total number of iterations is usually decreased (for equivalent accuracy) as compared to the older version of DIAMANT2.

Irrespective of the value of ε_{inner}, inner iterations are stopped if the number of iterations is bigger than the value of variable ITMAX. This is initialized with the input value of IIL and increased from one outer iteration to the next by one until the input value of IIM is reached.

Only one inner iteration acceleration method has been implemented in DIAMANT2 up to now, namely global rebalancing for the inner iterations. The group dependent rebalancing factors

$$(35)\quad FUN_g = \Sigma\, S_{g,i} \;/\; \Sigma(S_{g,i} + \sigma_{g\to g,i}*(\phi_{g,i}^{new} - \phi_{g,i}^{old}))$$

are calculated in each inner iteration. $S_{g,i}$ is the sum of the neutron sources in energy group g in mesh cell i. The summation extends over all mesh cells. No rebalancing is attempted in boundary flux cases. If FUN_g is 1.0 within a certain threshold or becomes negative, rebalancing is skipped too. All scalar fluxes and the leaking angular fluxes are then multiplied by the factor FUN_g. Global neutron balance is maintained in this manner (see /8/).

2.6 OUTER ITERATION PROCEDURE

The discretized multigroup neutron transport equations form a linear equation system:

(36a) $L*u = S_f + S_s*u + Q_{ext}$

(36b) $S_f = F*u/k_{eff}$ for homogeneous problems

(36c) $S_f = F*u$ for inhomogeneous problems

L describes the losses, S_f the fission source, S_s the scattering matrix, F the fission matrix and Q_{ext} the (possibly existent) external source. u is the vector of the flux values at the triangle centers, k_{eff} is the desired (dominant) eigenvalue of the problem (criticality factor). S_s is lower triangular since no up-scattering is allowed. Equations (36) are solved by means of the conventional power method ("fission source" or "outer" iteration): (j=0,1,2.....)

(37a) $L*u^{(j+1)} = S_f^{(j)} + S_s*u^{(j+1)}$

(37b) $S_f^{(j+1)} = F*u^{(j+1)}/k^{(j)}$

(37c) $k^{(j+1)} = k^{(j)}*\Sigma(F*u^{(j+1)}) \;/\; \Sigma(F*u^{(j)})$ (homogeneous problems)

(37d) $k^{(j+1)} = 1$ (inhomogeneous problems)

In (37c), the summation extends over all mesh points and all energy groups. $k_{eff}^{(j)}$ converges to the criticality factor, k_{eff}, as the number of iterations increases; $S_f^{(o)}$ is dependent upon the source guess used (source equal to 1.0 for all meshes as a default option).

An additional speed-up of the whole calculation can be achieved by the use of a source and flux guess taken from a previous calculation. Only guesses using an identical space-angle-energy mesh grid will be accepted. This option is very useful in calculations with anisotropic scattering. It is highly desirable first to establish a reasonably converged flux guess using

isotropic scattering and then starting the anisotropic scattering calculation using that guess.

Depending on input specification, there are three criteria to test convergence of outer iterations. All of them test the deviation of the sum of all neutron sources S_0 between two successive outer iterations (sums extending over all meshpoints):

(38) $$\varepsilon_{outer} = \Sigma | S_0^{old} - S_0^{new} | / \Sigma S_0^{new}$$

is called "integral" test and is used whenever the input variable EPS is less then zero.

(39) $$\varepsilon_{outer} = \max | 1.0 - S_0^{old} / S_0^{new} |$$

is called "point-wise" test and is used whenever EPS is greater than zero. This criterion puts emphasis on the local behavior of the source distribution.

The third possibility is a test based on eigenvalue bounds which is used whenever EPS is equal to zero:

(40a) $\varepsilon_{outer} = k_u - k_l$ where

(40b) $k_u = \max | S_0^{old} / S_0^{new} |$

(40c) $k_l = \min | S_0^{old} / S_0^{new} |$

k_u and k_l are upper and lower eigenvalue bounds.

Outer iterations are stopped if $\varepsilon_{outer} \leq |$ EPS $|$. In case EPS = 0.0 the code uses the value of TEPS. Tests (39) and (40) are much more stringent than (38). Test (40) avoids premature iteration termination in slowly convergent cases where the source distribution changes are small from one iteration to the next, but leads to increased CPU time consumption.

Outer iterations are stopped too if either the maximum number of outer iterations is reached (input variable ICM) or if CPU time is near to exhaustion. The code measures the CPU time t_{out} needed for each outer iteration. Using the timing function TIMLIM it gets then the amount of CPU

time still left and breaks the iteration whenever $3*t_{out}$ is smaller than the remaining CPU time. Depending on the accounting system, t_{out} may not have the dimension of a time (e.g. on CYBER-205 it is measured in System Billing Units). Function TIMLIM usually has to be rewritten for a new computer (available for IBM and compatibles, CYBER-205 and CRAY).

2.7 OTHER CODE OPTIONS

2.7.1 Adjoint Calculations

DIAMANT2 solves the adjoint transport equation in the usual way /8/:

1. The scattering matrices are transposed.

2. The role of $\nu\sigma_f$ and χ are exchanged.

3. The energy group order is reversed.

4. The solution for angle $\vec{\Omega}_m$ is identified with angle $-\vec{\Omega}_m$ (of the adjoint problem).

Formally, (1.) transforms the orginal down scattering problem into an up scattering problem, but (3.) converts it to a formal down scattering problem again. It should be noted that the group numbers will be inverted in input and output, i.e., energy group 1 in adjoint computations corresponds to the lowest energy range. Moreover, the order of energy groups and of directions (according to (4.)) are to be inverted for the input of bucklings, boundary fluxes and flux guesses. Only input of nuclear cross sections is the same in real and adjoint problems. The user should be aware that fluxes and source terms are group integrals in direct calculation, whereas adjoint fluxes and source terms must be understood as group averaged values, in accordance with (1.) and (2.) (see also the comments concerning group averaged and group integrated quantities in "Multigroup Equations" on page 6).

2.7.2 Inhomogeneous Calculations

Inhomogeneous calculations can be specified using either boundary fluxes or external sources.

External sources are not allowed to be anisotropic. They may be input in four options :

1. for all groups and all meshes;

2. for all meshes along with a global energy spectrum;

3. for all groups and all mixtures;

4. for all mixtures along with a global energy spectrum.

Boundary fluxes are not allowed to be anisotropic. They may be input in two options :

1. for all groups and all sides containing sources;

2. for all sides containing sources along with a global energy spectrum.

In those cases where a global energy spectrum is used, the group dependent source distribution is determined as the product of the source strength specified for the mesh, mixture or side, respectively, times the input energy spectrum. No normalization to unity of the energy spectrum is done by DIAMANT2.

2.7.3 Evaluations and Balances

It is also possible to carry out elementary evaluations in DIAMANT2. Depending upon the options selected, the reaction densities

$$(41a)\quad \vec{D}(\vec{r}) = \sum_{g=1}^{IGM} \sigma_g(\vec{r}) * \phi_g(\vec{r})$$

and/or the reaction rates:

$$(41b)\quad \vec{R}(\vec{r}) = \vec{D}(\vec{r})*V$$

$$(41c)\quad \text{where } \sigma = \nu\sigma_f,\ \sigma_c,\ \sigma_f,\ \sigma_a = \sigma_c+\sigma_f,\ \text{or } \sigma_t$$

for each mesh are calculated and printed out. ϕ is the scalar flux and V the (constant) triangle volume. Subscripts f, c, a, t refer to fission, capture and absorption, respectively. σ_t means the total cross section.

DIAMANT2 routinely calculates total system balances and selectively balances for the individual mixtures and regions (see input variable KAUSW). Analogous sums are also formed for every spatial point located within one mixture and/or within any contiguous zone having a common mixture number (called "region"). In the present implementation of the code a PARAMETER statement limits the number of regions which can be treated to 30 at maximum. The balances are output by groups and as group sums.

The following list gives the individual terms which are printed out. The (spatial) sums run over all centered fluxes, m runs over all directions and k over the three triangular sides (compare Figure 8 on page 27). $\psi_{m,g,out,k}$ is the angular flux value on the triangular side k. The leakage calculation takes into account that the amounts on triangular sides in the interior of the reactor model cancel one another. For mixture and region balances neither the neutron balance nor the leakages per dodecant are evaluated.

Due to the way of determining the outscattering contribution from the scattering matrix, the term for the total losses exceeds the correct total losses by an amount arising from (n,2n) and (n,3n) contributions. Accordingly, the neutron balance is affected by this deviation too. (Remark: frequently in diffusion codes the outscattering contribution is derived as the difference of removal and absorption rate so that the difficulties mentioned before with (n,2n) and (n,3n) contributions can be avoided.)

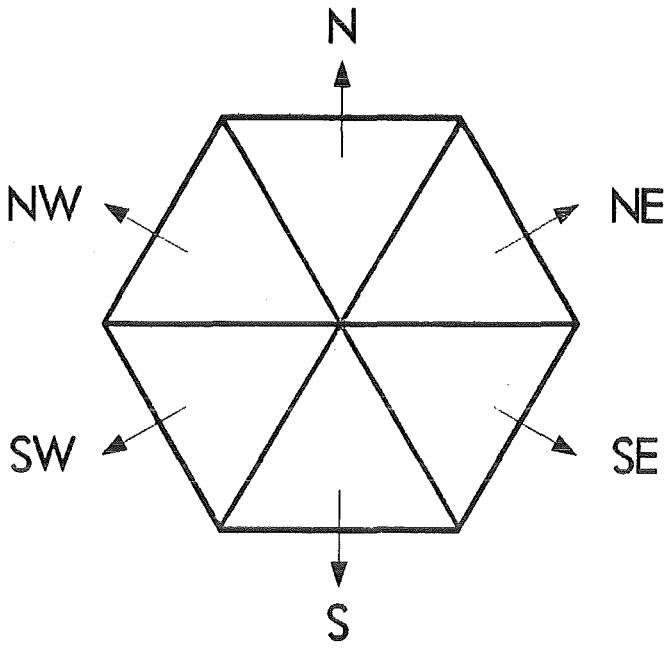


Figure 8. Possible Orientations of the System Boundary

Individual Terms in Balancing Output (Group Dependent)

External source	$\sum_{i=1}^{I} Q_{ext,g}(i)$
Fission source	$\chi_g^* \sum_{h=1}^{G} \sum_{i=1}^{I} \nu\sigma_{f,h}(i)*\phi_h(i)$
Inscatter source in group g	$\sum_{h=1}^{g-1} \sum_{i=1}^{I} \sigma_{h\to g}(i)*\phi_h(i)$
Total production	External source + Fission production + Inscatter source
Total flux	$\sum_{i=1}^{I} \phi_g(i)$
Fission production	Fission source/k_{eff}
Within-Group scattering	$\sum_{i=1}^{I} \sigma_{g\to g}(i)*\phi_g(i)$
Absorption	$\sum_{i=1}^{I} \sigma_{a,g}(i)*\phi_g(i)$

Buckling leakage	$\sum_{i=1}^{I} B_g^2(i)*\phi_g(i)/(3*\sigma_{tr,g}(i))$
Out-scattering from group g	$\sum_{h=g+1}^{G} \sum_{i=1}^{I} \sigma_{g\to h}(i)*\phi_g(i)$
Total leakage	Sum of the 6 individual leakages (see below)
Total losses	Absorption + Leakage + Out-scattering
Capture density	$\sum_{i=1}^{I} \sigma_{c,g}(i)*\phi_g(i)$
Fission density	$\sum_{i=1}^{I} \sigma_{f,g}(i)*\phi_g(i)$
Neutron balance	Total production / Total losses
Leakages from the 6 dodecants	$\sum_{m=1}^{M} \sum_{k=1}^{3} w_m*A_{k,m}*\psi_{m,g,out,k}*h$

Here, I is the total number of spatial meshes, G the number of energy groups and M the number of discrete directions. In the last formula, w_m and $A_{k,m}$ are the constants defined in "Construction of S_N-Constants" on page 8 and "Diamond Difference Approximation" on page 13, respectively. h is the constant length of a triangle side.

CHAPTER 3. IMPLEMENTATION OF DIAMANT2

This chapter describes details of the implementation of the above discussed algorithm. First we give simplified flow diagrams of the main subroutines DRIVER, DIAMAN, OUTER and TRINER. Then a short characterization of each subroutine in the code package is given. COMMON usage, data management and data files are discussed followed by a description of the interface files generated by the code. The last part of this chapter is concerned with the differences in code structure for the KAPROS system version and for the stand-alone code.

3.1 PROGRAM FLOW CHARTS

This chapter describes the global flow of control in DIAMANT2, a subroutine call cross reference list and gives the program flow charts for the main subroutines DRIVER, DIAMAN, OUTER and TRINER.

Arrangement of subroutines in the source file:

1. DIAM2 (main entry point);
2. BLOCK DATA (initialization);
3. DIAMANT2 subroutines common to all versions;
4. Subroutines differing in KAPROS and STAND-ALONE versions;
5. Functions TASKTI, TIMLIM for IBM, SIEMENS VP, CRAY and CYBER-205;
6. Function JTIME in IBM ASSEMBLER.

The code consists presently of 72 subroutines (76 in KAPROS version) with ≈4000 FORTRAN executable statements (≈5000 in KAPROS). The source file contains ≈7500 lines (≈10000 in KAPROS) including comments and continuation cards. The additional statements in the KAPROS version are due to subroutines FILLC and READKO. On IBM systems, compiled with a standard FORTRAN77 compiler, the load module (4 bytes mode) has a length of 240 kB, working arrays not included.

3.1.1 Simplified Call Structure Diagram for DIAMANT2

```
DIAM2  → KSINIT
       → DRIVER → READKO
                → FRESP1
                → DIAMAN → CLEAR
                         → READKO
                         → STODIS → EXTEND
                                  → FRESP1
                         → PLOUT
                         → INFLPR
                         → CROSIN
                         → FILLC
                         → ADJG
                         → SNCONS
                         → ORGA
                         → ANISO
                         → BUCKL
                         → PRINT1 → PLOUT
                                  → LZDZ   → MINMAX
                         → REGION → REGIS  → REGTST
                         → FISSN
                         → OUTER  → SOUSCA
                                  → SOUFIS
                                  → SOUANI  → SOUANJ  → GATHER
                                  → TRINER  → SOUANK
                                            → FLXMOM
                                            → CVGINN
                                  → FISSN
                                  → CVGOUT
                         → BILANZ → PLOUT
                         → PRINT2 → PRINT3
                         → EXTEND
```

The complete call structure is shown in Figure 9.

Figure 9. Cross Reference Table for Subroutine Calls

Subroutine		Calls					
ANISO	2	MULT	PUT				
BILANZ	6	GETFLX	MULT	PLOUT	PUTFLX	SUMT	WRITKS
BUCKL	1	PRINT5					
CLEAR	2	PUT	PUTIN				
CROSIN	1	READKR					
DIAMAN	44	ADJG	ANISO	BILANZ	BUCKL	CLEAR	CROSIN
		DMPFND	DMPINI	DMPPOS	DMPWRT	EXTEND	FILLC
		FISSN	GETFLX	INFLPR	INITFA	INITFX	ORGA
		OUTER	PERT	PLOUT	PRINT1	PRINT2	PRINT4
		PRINT5	PRINT6	PUT	PUTIN	READKC	READKI
		READKR	REGION	REWIKS	SNCONS	SOUBND	SOUEXT
		STODIS	SUMUP	TASKTI	TIMLIM	UNCHCK	XSANIS
		XSCHCK	XSHELP				
DIAM2	2	DRIVER	KSINIT				
DMPFND	4	DMPRD	DMPSKP	READKS	REWIKS		
DMPINI	1	WRITKS					
DMPPOS	4	BACKKS	DMPSKP	READKS	REWIKS		
DMPRD	4	DMPSKP	READKS	REWIKS	WRITKS		
DMPSKP	1	SKIPKS					
DMPWRT	3	BACKKS	GETFLX	WRITKS			
DRIVER	6	DIAMAN	FRESP1	PRINT7	READKC	READKO	TASKTI
FISSN	3	GETFLX	MULT	PUT			
FLXMOM	1	TASKTI					
GETFLX	4	EQUAL	READKS	REWIKS	SKIPKS		
INFLPR	1	READKI					
INITFA	3	GETFLX	REWIKS	WRITKS			
INITFX	2	REWIKS	WRITKS				
IOFLMO	5	EQUAL	GETFLX	READKS	REWIKS	WRITKS	
LZDZ	1	MINMAX					
OUTER	13	CVGOUT	EQUAL	FISSN	MULT	PUT	PUTFLX
		REWIKS	SOUANI	SOUFIS	SOUSCA	SUMUP	TASKTI
		TRINER	WRITKS				
PERT	1	WRITKS					

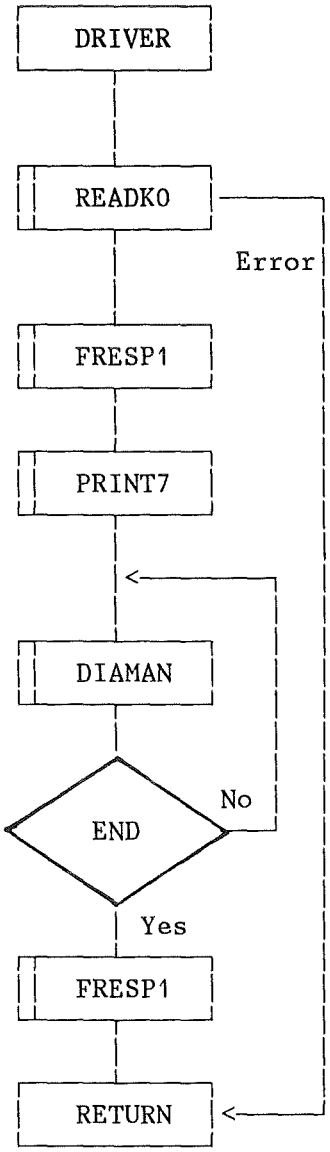
PLOUT	5	PUTIN	READKI				
PRINT1	5	LZDZ	PLOUT				
PRINT2	4	GETFLX	MULT	PRINT3			
PUTFLX	5	EQUAL	REWIKS	SKIPKS	WRITKS		
REGION	3	PUTIN	REGIS				
REGIS	1	REGTST					
SOUANI	8	EQUAL	MULT	PUT	PUTFLX	READKS	REWIKS
		SOUANJ	WRITKS				
SOUANJ	1	GATHER					
SOUANK	1	TASKTI					
SOUBND	2	READKR					
SOUEXT	2	READKR					
SOUSCA	2	GETFLX	PUTFLX				
STODIS	3	EXTEND	FRESP1				
TRINER	9	BACKKS	CVGINN	EQUAL	FLXMOM	MULT	PUT
		SOUANK	TASKTI	WRITKS			
XSANIS	1	PUT					
XSHELP	1	PUT					

Remarks:

1. BACKKS, READKC, READKI, READKR, READKS, REWIKS,SKIPKS and WRITKS are entries in the central I/O module READKO

2. The timing function TASKTI is called at several places (DIAMAN, DRIVER, FLXMOM, OUTER, SOUANK and TRINER). Function TIMLIM is used to prevent a job break-down caused by insufficient CPU time (or equivalent accounting parameter) left. It is called in DIAMAN and machine dependent.

3.1.2 Flow Diagram for Subroutine DRIVER



Initialize READKO-package

Get the amount of working storage at the start

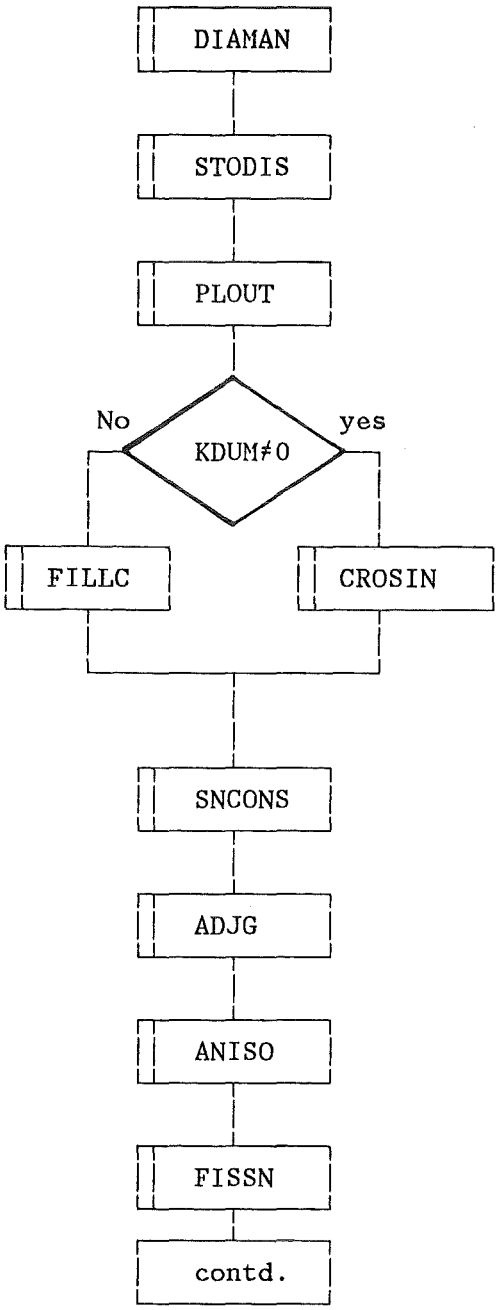
Printout of parameter value table

Each call does one transport calculation

Get the amount of working storage at the end

3.1.3 Flow Diagram for Subroutine DIAMAN

This is the main subroutine of DIAMANT2. It controls input preparation, outer iteration process and balancing and evaluation output.



Calculation of dynamic storage needs

Read and print layout description

Read group cross-sections from cards

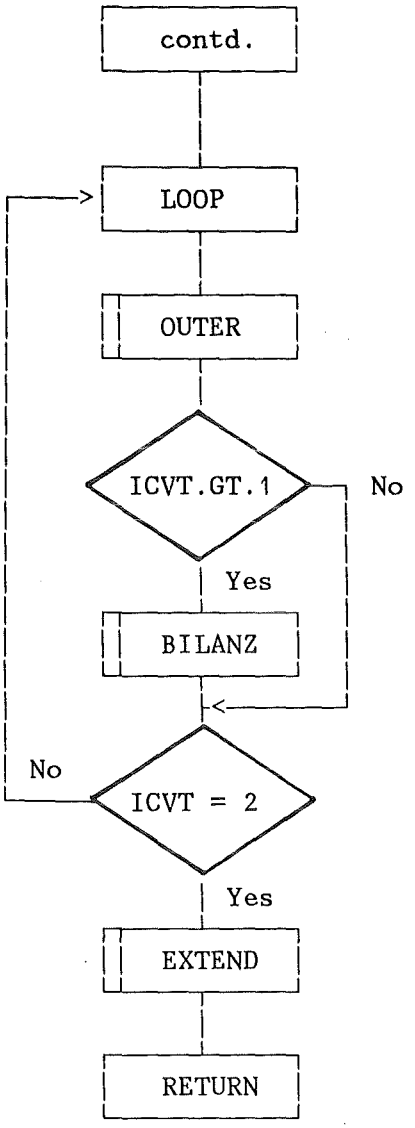
or from SIGMN-file

Define quadrature constants

Transpose scattering matrix (LADJNT=TRUE)

Calculate Legendre polynomials (LANISO=TRUE)

Calculate initial fission source



DO ITOUT = 0,IRM (outer iteration)

A complete outer iteration is performed

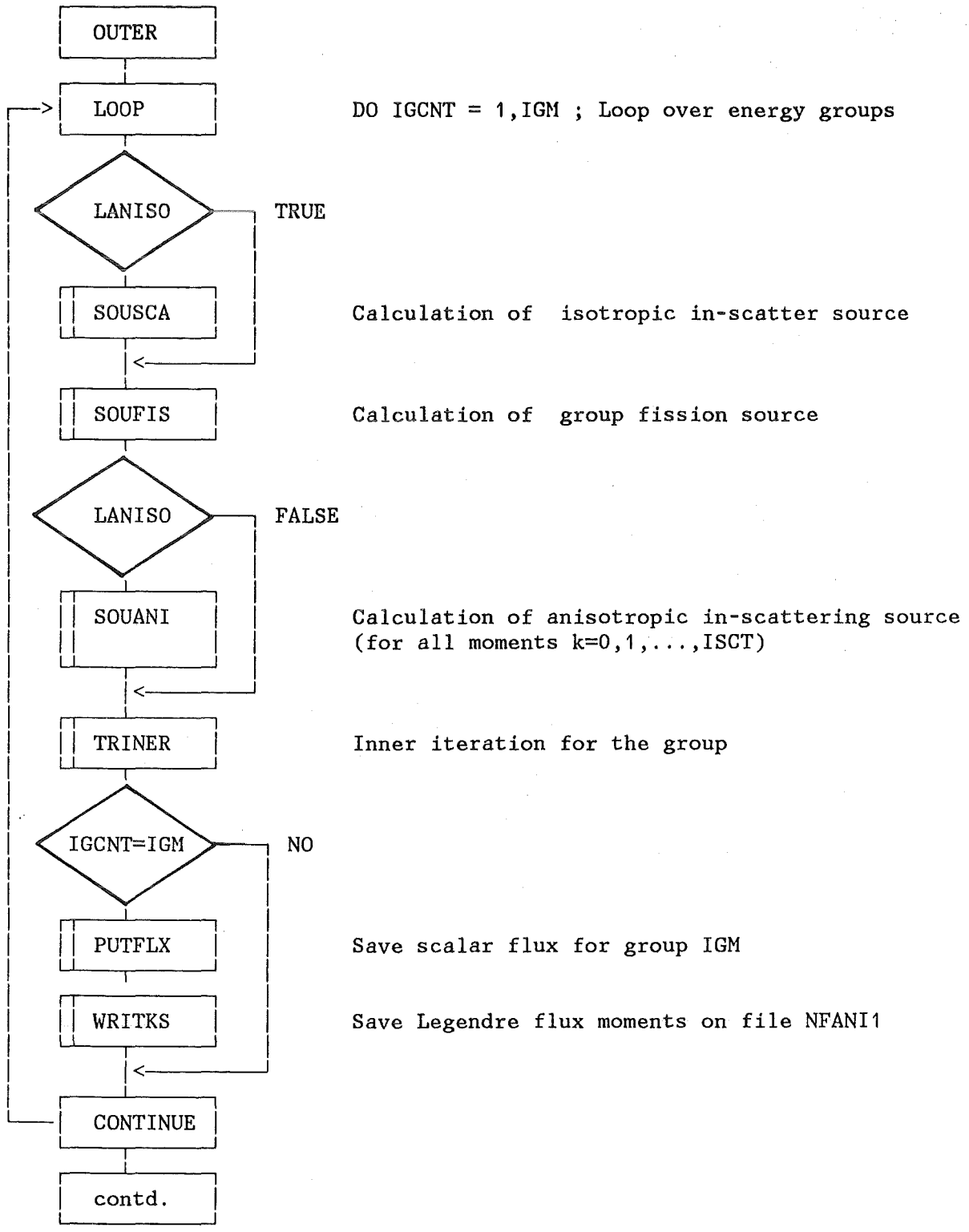
Calculation of balances

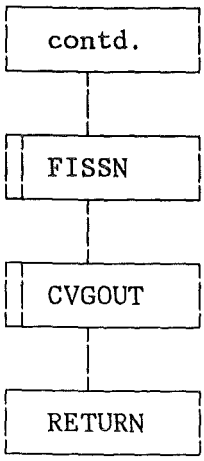
ICVT controls iteration process
and is set in subroutine OUTER

Release all dynamically allocated memory

3.1.4 Flow Diagram for Subroutine OUTER

This subroutine performs one outer iteration.



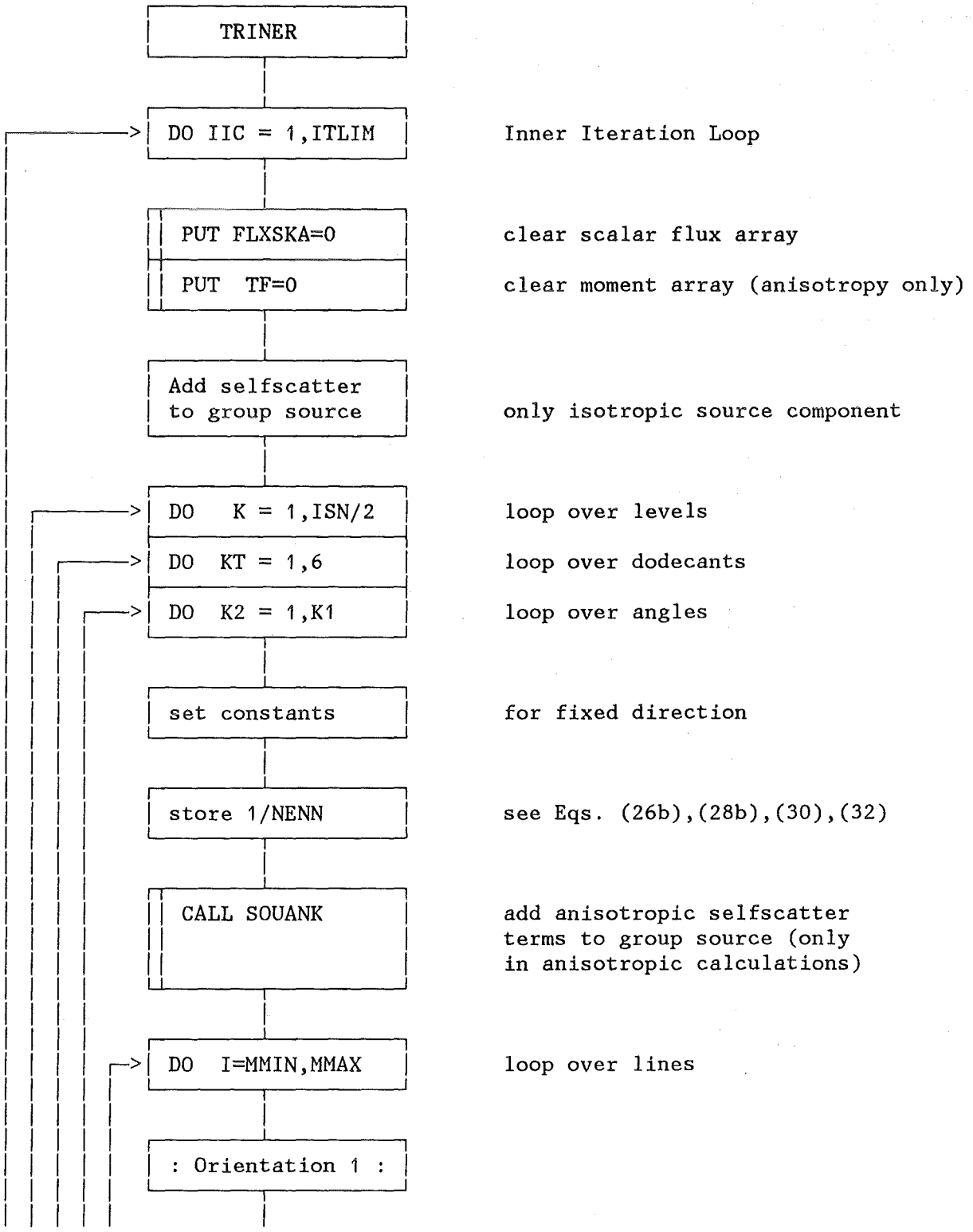


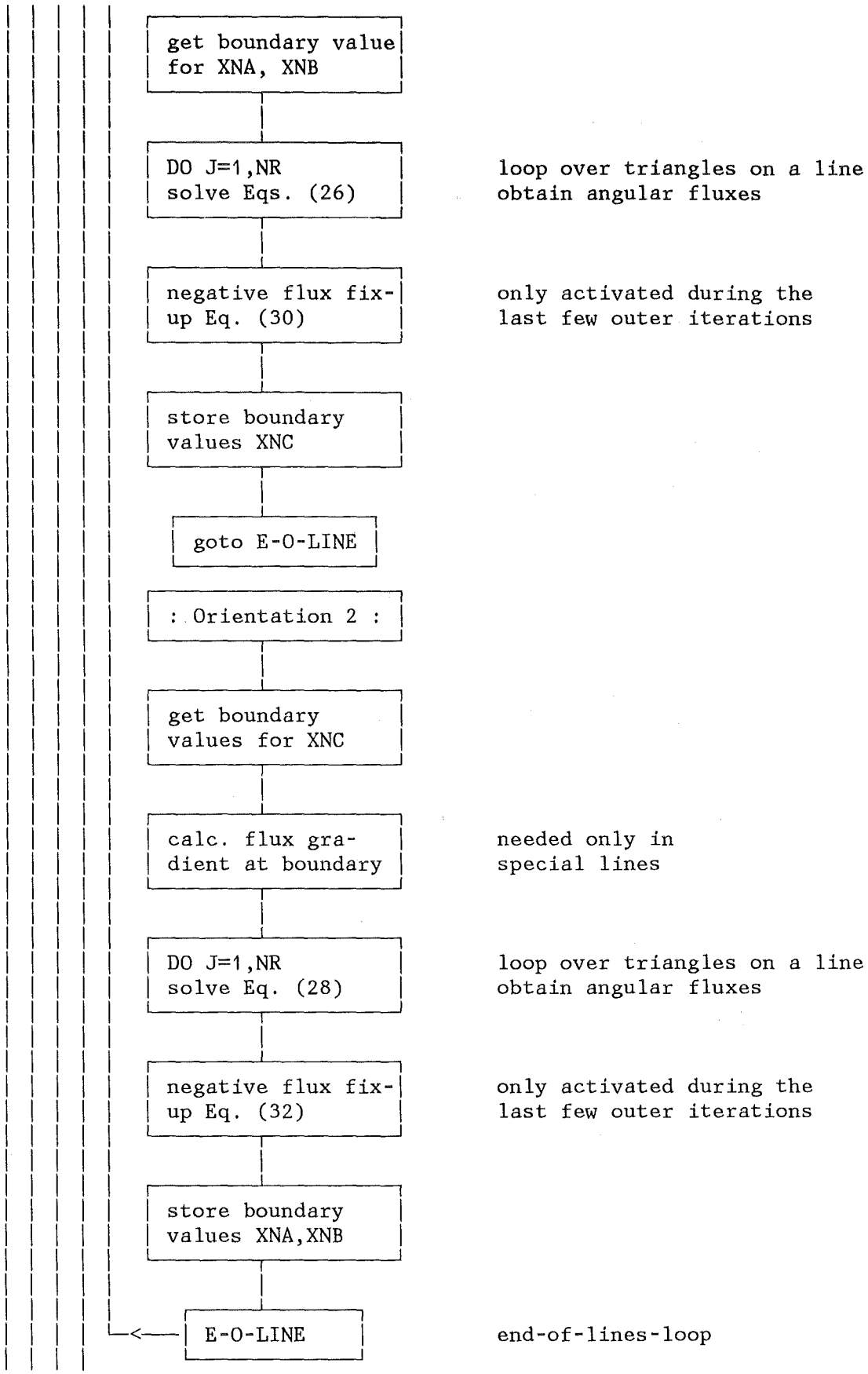
Update fission source for next outer iteration

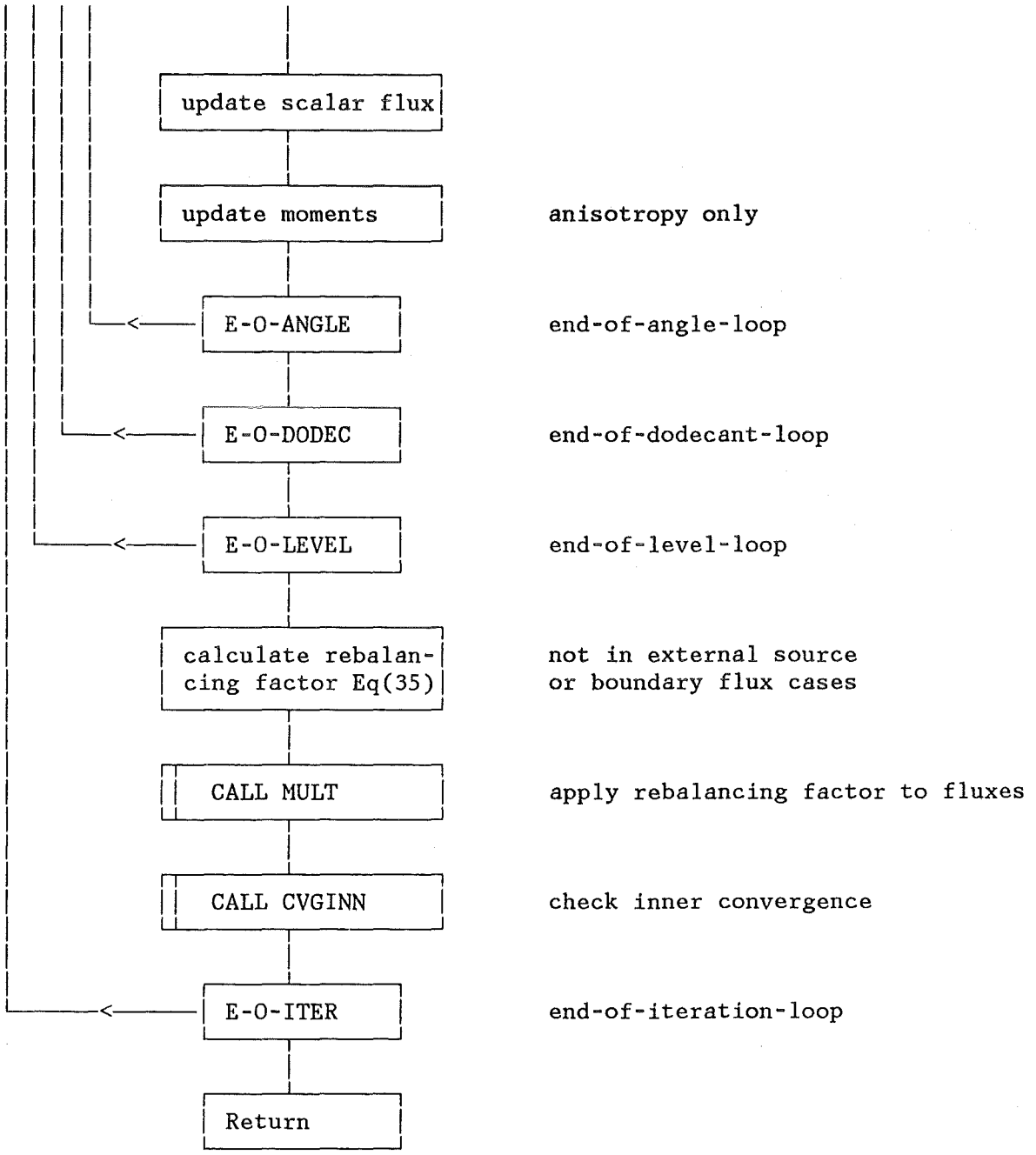
Check convergence of outer iteration

3.1.5 Flow Diagram for Subroutine TRINER

This subroutine performs the inner iteration.




get boundary value for XNA, XNB
DO J=1,NR solve Eqs. (26)
loop over triangles on a line obtain angular fluxes
negative flux fix-up Eq. (30)
only activated during the last few outer iterations
store boundary values XNC
goto E-O-LINE
: Orientation 2 :
get boundary values for XNC
calc. flux gra-dient at boundary
needed only in special lines
DO J=1,NR solve Eq. (28)
loop over triangles on a line obtain angular fluxes
negative flux fix-up Eq. (32)
only activated during the last few outer iterations
store boundary values XNA,XNB
E-O-LINE
end-of-lines-loop


update scalar flux
update moments
anisotropy only
E-O-ANGLE
end-of-angle-loop
E-O-DODEC
end-of-dodecant-loop
E-O-LEVEL
end-of-level-loop
calculate rebalan-
cing factor Eq(35)
not in external source
or boundary flux cases
CALL MULT
apply rebalancing factor to fluxes
CALL CVGINN
check inner convergence
E-O-ITER
end-of-iteration-loop
Return

3.2 FUNCTION OF THE DIFFERENT SUBROUTINES

In the following list, subroutines which contribute much to CPU time consumption are marked with (V), those which may also benefit from vectorization are marked with (V/S). Only these few routines are worth to be considered when optimizing DIAMANT2 for a vector computer.

ADJG	Transposes the scattering cross-section matrix for adjoint calculations
ANISO	Calculates the spherical harmonics functions for use in anisotropic calculations
BILANZ	Calculates neutron balances and performs flux and source normalization
BUCKL	Adds a buckling correction term to total cross-section
CLEAR	Initializes arrays with 0.0 (V)
CROSIN	Reads cross-section input by card
CVGINN	Tests convergence of inner iterations (V)
CVGOUT	Tests convergence of outer iterations (V/S)
DIAM2	Contains code abstract and calls DRIVER
DIAMAN	Controls the total calculational sequence for one case
DMPFND	Tries to find a restart dump on unit NFDMPO
DMPINI	Creates an end record on unit NFDMPN
DMPPOS	Positions unit NFDMPN just before the end record to write out a further restart dump
DMPRD	Reads a restart dump
DMPSKP	Skips one restart dump on unit NFDMPN
DMPWRT	Writes a restart dump
DRIVER	Controls the execution of a sequence of DIAMANT2 calculations
EQUAL	Copies one array into another one (V)
FISSN	Calculates the fission source (V/S)
FLXMOM	Calculates anisotropic flux moments (V)
GATHER	Gathers non-contiguous data into contiguous arrays (V)
GETFLX	Finds for a given group the flux of the previous outer iteration
INFLPR	Reads the group numbers for which fluxes are to be printed
INITFA	Initializes the files NFANI1 and NFANI2 needed in anisotropic calculations
INITFX	Initializes the files NFSCR1 and NFSCR2 needed possibly as scratch files

IOFLMO	Handles I/O of flux moments either on units NFANI1 and NFANI2 or in COMMON block /COMMOM/
LZDZ	Sets up auxiliary arrays for the mesh-angle sweeps during inner iterations
MINMAX	Finds the minimum and maximum limits of the real model in the reference parallelogram
MULT	Multiplication of an one-dimensional array by a scalar (V)
ORGA	Sets up the organization of the mesh-angle sweeps
OUTER	Performs one outer iteration
PERT	Creates interface file for perturbation module (file NFPERT)
PLOUT	Input, interpretation and printout of mixture distribution
POSMRT	Prints relevant information for test purposes and debugging (including COMMON blocks and arrays); see remark at the end of this list.
PRINT1	Tabulates some of the arrays specified by input cards
PRINT2	Tabulates fluxes and/or reaction rates or density tables according to Eqs. (41) after convergence of iterations
PRINT3	Printing of flux values
PRINT4	Prints the external source input values
PRINT5	Prints the input buckling values
PRINT6	Prints input data cards in tabular form along with an explanation
PRINT7	Prints the PARAMETER values relevant for the code implementation
PUT	Initializes real arrays with a constant value (V)
PUTIN	Initializes integer arrays with a constant value (V)
PUTFLX	Stores the flux for a given group in memory or on disk
REGION	Prints the region distribution and sets up array MIXREG
REGIS	Used by subroutine REGION to determine contiguous mixture regions
REGTST	Auxiliary routine used by REGIS
SNCONS	Defines the built-in S_N-quadrature constants
SOUANI	Calculates the anisotropic group scattering source (subroutine SOUANJ) and adds to this the group fission source contribution during outer iterations
SOUANJ	Calculates the anisotropic group scattering source excluding the within-group scattering contribution (V/S)
SOUANK	Adds the anisotropic within-group scattering contribution to group scattering source (during inner iterations) (V)
SOUBND	Reads boundary flux input
SOUEXT	Reads input data for external source cases
SOUFIS	Calculates the fission source contribution to the group source during outer iterations (V/S)

SOUSCA Calculates the scattering source contribution to the group source during outer iterations (called only in isotropic calculations) (V/S)

STODIS Calculates the pointers to distribute working arrays A and IW

SUMT Calculates the sum of two arrays (V)

SUMUP Sums up an array (FUNCTION) (V)

TRINER Performs the inner iterations (V)

UNCHCK Checks the input FORTRAN unit numbers for consistency

XSANIS Stores anisotropic within-group scatter cross section in array XSANI

XSCHCK Compares the calculated absorption cross section with input value

XSHELP Stores total, fission and within-group scatter cross section in separate arrays

Remark:

POSMRT is not called within the present implementation of DIAMANT2. But it may be useful for debugging purposes to insert at suitable places calls to this subroutine. POSMRT prints out the values of all COMMON variables (as namelists) as well as the contents of the different code arrays. POSMRT has no parameter list on the call. If called without modification and especially within inner iterations, this routine may produce a huge amount of output! POSMRT itself calls a couple of auxiliary subroutines which are not included in the above list (but are provided along with the source deck).

Subroutines with different implementation for STAND-ALONE and KAPROS version or on different computers

EXTEND Performs dynamic field extensions and releases (see "Use as Stand-alone program" on page 71).

FILLC Gets cross-section data from a SIGMN-block (see "Use as Stand-alone program" on page 71).

FRESP1 Determines the amount of memory space available for array extensions (see "Use as Stand-alone program" on page 71).

JTIME Supplies the remaining CPU time for the job (system dependent) (see "Use as Stand-alone program" on page 71).

KSCC Sets a stop code in case of any error condition during execution (see "Use as Stand-alone program" on page 71).

KSINIT Sets channel numbers for standard read and print units (see "Use as Stand-alone program" on page 71).

READK0 Central I/O module for binary data transfers (see "Use as Stand-alone program" on page 71).

TASKTI Function; returns the CPU time already used in seconds

TIMLIM Function; returns the CPU time still left in seconds (or some equivalent resource control parameter)

Additional subroutines for KAPROS version

DBLENG Gets the length of KAPROS data blocks

FILSET Auxiliary program to FILLC to initialize arrays

FILTES Auxiliary program to FILLC to test input

FILTRA Performs the transport correction on total cross-section

INITDB Initializes new KAPROS data blocks (belongs to READK0 package)

KS.... All these subroutines belong to the KAPROS system kernel

WQRG Reading of KAPROS SIGMN blocks or specified parts of them

3.3 DATA MANAGEMENT, COMMON USAGE AND DATA FILES

In the following we describe data management and data files as used in the stand-alone version. Differences necessary for the KAPROS version are described in "KAPROS Implementation" on page 69. For an implementation of DIAMANT2 one has to select PARAMETER values activating or deactivating options, defining dimensions and setting certain maximum values. As these values depend on the computer used, we postpone a discussion of them (see "How to Choose PARAMETER Values" on page 86).

3.3.1 Distribution of Working Arrays

Following input processing, DIAMANT2 dynamically allocates space in two working arrays.

A storage array A with NACOM words is allocated in BLANK COMMON for real arrays. The variable NACMAX of the PARAMETER statement fixes the maximum number of memory positions possible to store in A. Figure 10 shows the subdivision of this working array (see also remarks to Figure 11).

A storage array IW with NW words is allocated in BLANK COMMON for integer arrays. The variable NWAX of the PARAMETER statement fixes the maximum number of memory positions possible to store in IW. Figure 12 on page 49 illustrates the subdivision of this working array.

Figure 10. Array Allocation in A(NACMAX) resp. DBN=DIAM2ŏREAL*4

Small numbers refer to remarks given at the end of the table

POINTER	LENGTH[1]	CONTENT	NAME	REMARKS
JPOINT		Start of working array relative to A(1)		
JXNA JXNB JXNC	JLIM2 JLIM2 JLIM2	Angular fluxes at triangle sides	XNA XNB XNC	
JHLP1 JHLP2	JLIM2 JLIM2	Angular fluxes at triangle vertex	HLP1 HLP2	
JXNBI JXNBO JXNAI JXNAO JXNCI JXNCO	JJMM IGM*JJMM JJMM IGM*JJMM JJMM IGM*JJMM	In and out leaking fluxes	XNBI XNBO XNAI XNAO XNCI XNCO	
JFLXAN	MCMML	Angular flux	FLXANG	
JFLXNE	MCMML	Scalar flux new	FLXNEW	
JS	MCMML	Source	S	
JFLXOL	MCMML	Scalar flux old	FLXOLD	
JC	IHM*IGM*MTR	XS-tables	C	[5]
JXTOT	MTR*IGM	Total XS	XTOT	[5]
JXSSCA	MTR*IGM	Within-group scattering XS	XSSCAT	[5]
JXSFIS	MTR*IGM	Fission XS	XSFISS	[5]
JGRPSO	MCMML	Source for a group	GRPSOU	
JFG	MCMML	Fission source new	FG	
JFGOLD	MCMML	Fission source old	FGOLD	
JBUCK	MTR*IGM	Buckling values	BUCK	[2]
JEXTSO	IGM*MCMML	External source	EXTSOU	[2]

JXNAIJ JXNBIJ JXNCIJ	JJMM*IGM JJMM*IGM JJMM*IGM	Outleaking fluxes for each triangle side	XNAIJ XNBIJ XNCIJ	
JBIL1 JBIL2 JBIL3	IGP*24 IGP*14*MAT IGP*14*IREGMX	Global ┐ balance Mixture ├ tables Region ┘	BIL1 BIL2 BIL3	see figure 11 and [2]
JFLXSK	IGM*MCMML	Scalar flux for all groups	FLXSKA	[3]
JSS	MCMML*JMM	Anisotropic source	SS	[4]
JT	NRMOM*JMM	Spherical harmonics	T	[4]
JTW	NRMOM*JMM	T times integration weight	TW	[4]
JTF	NRMOM*MCMML	Flux moments new	TF	[4]
JTFOLD	NRMOM*MCMML	Flux moments old	TFOLD	[4]
JXSANI	NRMOM*JMM*IGM	anisotropic within-group scatter XS[5]	XSANI	[4]
JFIN		End of working array		

Remarks:

[1]MCM = No. of intervals along horizontal side
MLM = No. of intervals along oblique side
JLIM2 = MAX(MCM,MLM)
JMM = No. of directions
JJMM = JLIM2*JMM
MCMML = 2*MCM*MLM
IGM = No. of energy groups
IGP = IGM + 1
MAT = No. of cross section tables
MT = No. of mixtures
IHM = IGM + IHT
MTR = MAT + 3
IREGMX = No. of different regions in layout
NRMOM = No. of stored moments

[2]Only allocated if requested by input

[3]Only allocated in storage optimization 1

[4]Only allocated in anisotropic calculations (ISCT > 0)

[5]XS means nuclear cross section

Dimensions :

BIL1 (IGP*24) (global balances)

BIL2 (IGP*14*MAT) (mixture dependent balances)

BIL3 (IGP*14*IREGMX) (zone dependent balances)

I	BIL(I,J,K)	I	BIL(I,J,K)
1	External source	13	Capture density
2	Fission source	14	Fission density
3	In-scattering source	15	Neutron balance
4	Total production	16	Leakage toward NW
5	Total flux	17	Leakage toward NO
6	Fission production	18	Leakage toward S
7	Within-group scattering source	19	Leakage toward SO
8	Absorption	20	Leakage toward SW
9	Buckling leakage	21	Leakage toward N
10	Out-scattering	22	Leakage toward NW+SO
11	Total leakage	23	Leakage toward S+N
12	Total losses	24	Leakage toward NO+SW

Meaning of indices:

J: Mixture or zone index (or = 1 for global balances)

K: Group number for the corresponding balance types; for K = IGM+1(=IGP) the array contains the sum over the group values. Entries K = 15 to 24 are calculated only for global balances (array BIL1). IREGMX is fixed by a PARAMETER value to 30 at maximum, MAT is an input variable (number of cross section tables to be stored). For the interpretation of the leakage directions see Figure 8 on page 26.

Figure 11. Usage of Arrays BIL1, BIL2, BIL3

Small numbers refer to remarks given at the end of the table

POINTER	LENGTH[1]	CONTENT	NAME	REMARKS
KPOINT		Start of working array relative to IW(1)		
KMODBN	18*JLIM	Index array for model boundaries	MODBND	
KMIXDI	MCMML	Mixture distribution	MIXDIS	
KIHLP1	MCMML	Auxiliary array	IHLP1	
KMIXRE	MCMML	Region distribution	MIXREG	[2]
KMODEL	MCMML	array to markup meshes belonging to the real model	MODEL	
KMODAN	MCMML	array to markup meshes using anisotropic scattering	MODANI	[3]
KIHLP2	MCMML	Auxiliary array	IHLP2	[3]
KIFIN		End of working array		

Remarks:

[1]MCM = No. of intervals along horizontal side
MLM = No. of intervals along oblique side
JLIM = MCMM+(MLM-1)*2
MCMML = 2*MCM*MLM

[2]only allocated if requested by input (KAUSW=1 or 11)

[3]only allocated in calculations with anisotropic scattering

Figure 12. Array Allocation in IW(NWMAX) resp. DBN=DIAM2ƀINTEGER*4

Figure 13. Use of COMMON Blocks by Subroutines

SUBROUTINE		COMMON-BLOCKS USED					
ADJG	2	COMARR	COMINT				
ANISO	2	COMARR	COMINT				
BILANZ	7	COMARR	COMBND	COMINT	COMLOG	COMREA	FILLCO
		FILLCX					
BUCKL	5	COMARR	COMINT	COMREA	FILLCO	FILLCX	
CLEAR	1	COMARR					
CROSIN	4	COMARR	COMINT	FILLCO	FILLCX		
CVGINN	3	COMARR	COMINT	COMREA			
CVGOUT	4	COMARR	COMINT	COMLOG	COMREA		
DIAMAN	9	/ /	COMARR	COMCHA	COMINT	COMLOG	COMPOI
		COMREA	FILLCO	FILLCX			
DIAM2	1	COMINT					
DMPFND	1	COMINT					
DMPPOS	1	COMINT					
DMPRD	4	COMARR	COMINT	COMLOG	COMREA		
DMPWRT	3	COMARR	COMINT	COMREA			
DRIVER	2	COMINT	COMREA				
FILLC	2	FILLCO	FILLCX				
FILTRA	2	FILLCO	FILLCX				
FISSN	4	COMARR	COMINT	COMLOG	COMREA		
FLXMOM	2	COMINT	COMREA				
GETFLX	3	COMINT	COMLOG	COMSTO			
INFLPR	2	COMARR	COMINT				
INITFA	1	COMINT					
INITFX	1	COMINT					
IOFLMO	1	COMMOM					
LZDZ	3	COMARR	COMBND	COMINT			
MINMAX	1	COMINT					
ORGA	2	COMARR	COMINT				
OUTER	4	COMARR	COMINT	COMLOG	COMREA		
PERT	7	COMARR	COMCHA	COMINT	COMLOG	COMREA	FILLCO
		FILLCX					
PLOUT	3	COMARR	COMINT	COMLOG			

SUBROUTINE		COMMON-BLOCKS USED					
PRINT1	5	COMARR	COMINT	COMREA	FILLCO	FILLCX	
PRINT2	6	COMARR	COMINT	COMLOG	COMREA	FILLCO	FILLCX
PRINT3	1	COMINT					
PRINT4	1	COMINT					
PRINT5	1	COMINT					
PRINT6	3	COMCHA	COMINT	COMREA			
PUTFLX	3	COMINT	COMLOG	COMSTO			
REGION	1	COMINT					
REGIS	1	COMINT					
REGTST	1	COMINT					
SNCONS	2	COMARR	COMINT				
SOUANI	3	COMINT	COMLOG	COMREA			
SOUANJ	3	COMARR	COMINT	COMREA			
SOUANK	3	COMINT	COMLOG	COMREA			
SOUBND	4	COMARR	COMBND	COMINT	COMLOG		
SOUEXT	1	COMINT					
SOUFIS	3	COMINT	COMLOG	COMREA			
SOUSCA	2	COMINT	COMREA				
STODIS	5	/ /	COMARR	COMINT	COMLOG	COMPOI	
TRINER	5	COMARR	COMBND	COMINT	COMLOG	COMREA	
UNCHCK	1	COMINT					
XSANIS	3	COMARR	COMINT	COMREA			
XSCHCK	2	COMINT	COMREA				
XSHELP	2	COMARR	COMINT				

3.3.2 Use of COMMON Blocks

DIAMANT2 currently uses 11 named COMMON blocks: /COMARR/, /COMBND/, /COMCHA/, /COMINT/, /COMLOG/, /COMMOM/, /COMPOI/, /COMREA/, /COMSTO/, /FILLCO/, /FILLCX/ and a BLANK COMMON / /.

Use of the different COMMON blocks by the various subroutines is shown in Figure 13 on page 50. This table has been produced by the utility OBJXREF /12/. Usage of the variables in different COMMON blocks by subroutines is shown in Figures 14 to 21. These tables have been generated using the static analyzer module of the test system RXVP /13/. The following list gives a short characterization of the function of the different COMMON blocks.

/COMARR/ contains arrays dimensioned by a PARAMETER statement

/COMBND/ contains some arrays needed to organize the mesh-angle sweeps

/COMCHA/ stores the 60 characters specifying the run identification

/COMINT/ stores scalar control integers

/COMLOG/ stores logical switches

/COMMOM/ contains the working array used to replace external storage of anisotropic flux moments if space is sufficient; it is used exclusively by subroutine IOFLMO.

/COMPOI/ pointers identifying subarrays in the working arrays A, IW

/COMREA/ stores scalar real variables

/COMSTO/ stores the number of words allocated in working arrays and an array used to optimize I/O in GETFLUX/PUTFLUX

/FILLCO/ stores pointers to special cross section types and some auxiliary integers

/FILLCX/ stores the sequence of cross section type names

/ / contains the working arrays A ,IW and their maximum lengths

Figure 14. Usage of COMMON /COMARR/

U=used, S=set, X=set and used, C=first used in a call

MODULE / SYMBOL	A	A	B	B	B.C	C	C	C	D.D	D	F	I	L.O	O	P	P	P.P	S	S	S	T	X
	D	N	I	L	U.L	R	V	V	I.M	M	I	N	Z.R	U	E	L	R.R	N	O	T	R	S
	J	I	L	K	C.E	O	G	G	A.P	P	S	F	D.G	T	R	O	I.I	C	U	O	I	H
	G	S	A	D	K.A	S	I	O	M.R	W	S	L	Z.A	E	T	U	N.N	O	B	D	N	E
		O	N	O	L.R	I	N	U	A.D	R	N	P	.	R		T	T.T	N	N	I	E	L
			Z	1	.	N	N	T	N.	T		R	.				1.2	S	D	S	R	P
DEV1					.C				U.				.				.				S	
DMU		U			.C				.				.				U.	X				
DMUA			U		.C				.				.				.	X			U	
DMUB			U		.C				.				.				.	X			U	
DMUC			U		.C				.				.				.	X			U	
ETA		C			.C				.				.				U.	X				
FUNK					.C				.				.				.				S	
FX			C		.C				.C	C	X		.				.U					
ICA				S	.				.				.				.				U	
ICB				S	.				.				.				.				U	
ICC			U	S	.				.				U.				.				U	
IDIM					.C			U	U.				.				.				S	
IFLPR1					.C				.			C	.				.					
IFLPR2					.C				C.			S	.				.U					
ILA				S	.				.				.				.				U	
ILB				S	.				.				.				.				U	
ILC			U	S	.				.				U.				.				U	
INVMAT			C		.C				.				.	C	C	X	.				C	
KOOX					.C				X.				.				.					
KOOY					.C				X.				.				.					
KTUT					.				U.				.				.			X		
KT4			U	S	.				.				U.				.				U	
MATTAB					.C	U			X.				.		C	U	.					
MAXINN					.C		S		C.				.				.					
MOMOAD					U.C				C.				.		C	X	.					U
MR			U		.C				.				.S		C		U.		U		U	
MRA			U		.C				.				.S				U.		U		U	
MRB			U		.C				.				.S				U.		U		U	
MRC			U		.C				.				.S				U.				U	
NEGOV					.C				.				.				.C				X	
REBA					.C				U.				.				.				S	
TINC			U		.C				.				X.				.				U	
VE		X			.C				C.				.		C		.					
WEIGHT			C	U	.C				.				.		C		U.	S			U	
XKE			U		.C				X.			X	.	C			.					
XKI		X			.C	C			X.				.	U			U.					

Figure 15. Usage of COMMON /COMBND/ and COMMON/COMCHA/

U=used, S=set, X=set and used, C=first used in a call

MODULE	BLSTO	IZOUI	LDUBN	AZBNE	ZDR	SYMBOL
ILEA	S U					
ILEB	U S					
ILOG	X U					
IMAX	U S U					
IMIN	X U					

MODULE	DPP	IER	ART	MTN	AT	N6	SYMBOL
C2W	C C U						

Figure 16. Usage of COMMON /COMLOG/

U=used, S=set, X=set and used, C=first used in a call

MODULE	BCDDF.GOPPP.PSSSS.SSTR	IVIMI.EUELR.UOOOO.OTRI	LGAPS.TTROI.TUUUU.UOIN	AOMRS.FETUN.FAAAB.FDNE	NUADN.LR T T.LNNNN.IIER	ZTN .X 2.XIJKD.SSR	SYMBOL
LADJNT	U U X U. U.						
LANISO	X . U U .						
LBNDSO	X . U. U						
LEXTSO	U X U. U.						
LFF	X . X						
LPERT	U X . U						
LSCRAT	U U U .U U.U . X						

Figure 17. Usage of COMMON /COMINT/

U=used, S=set, X=set and used, C=first used in a call

* MODULE *	A	A	B	B	B.C	C	C	D	D.D	D	D	D	D.F	F	G	I	I.I	K	L	M	O.
* *	D	N	I	L	U.R	V	V	I	I.M	M	M	M	R.I	L	E	N	N.N	S	Z	I	R.
* *	J	I	L	K	C.O	G	G	A	A.P	P	P	P	I.S	X	T	F	I.I	C	D	N	G.
* *	G	S	A	D	K.S	I	O	M	M.F	P	R	W	V.S	M	F	L	T.T	C	Z	M	A.
* *		O	N	0	L.I	N	U	A	2.N	O	D	R	E.N	O	L	P	F.F			A	.
* *			Z	1	.N	N	T	N	.D	S		T	R.		M	X	R	A.X			X	.
* SYMBOL * *					.				.				.					.			.
* ICM *					.			X	.				.					.			.
* ICVT *					.		S	X	.				.					.			.
* ID *					.			X	.			C	.					.			.
* IDFL *					.			S	.				.					.			.
* IDMPWR *					.				.	S		X	.					.			.
* ID1 *					.			S	.				.					.			.
* ID2 *					.			S	.				.					.			.
* ID3 *					.			S	.				.					.			.
* ID4 *					.			S	.				.					.			.
* ID5 *					.			S	.				.					.			.
* IEVT *					.			X	.				.					.			.
* IGCNT *			U		.	U		X	.				.					.			.
* IGM *	U		C		U.C		U	X	.		U	C	.U			U	U.U				.
* IGP *			C		.			X	.		C	C	.U					.			.
* IHM *	U				.			X	.				.					.			.
* IHS *	U		U		.			X	.				.					.			.
* IHT *	U		U		U.U			X	.				.					.			.
* IIL *					.			X	.				.					.			.
* IIM *					.		U	X	.				.					.			.
* II1 *					.			S	.				.					.			.
* II2 *					.			X	.				.					.			.
* INBO *					.			X	.				.					.			.
* INDSN *					.			S	.				.					.			.
* INORM *			U		.			X	.				.					.			.
* IQUELL *			U		.			X	.				.					.			.
* IQUER *					.			S	.				.					.			.
* IS *					.			S	.				.					.			.
* ISCT *		U			.U			X	.				.		U			.			.
* ISN *			U		.			X	.			U	C	.				.			U.
* ISTOP *					.S			X	.X	X	X		X.S			S	S	.			.
* ITH *					.			X	.				.					.			.
* ITLIM *					.			X	.				.					.			.
* ITMAX *				S	.			C	.				.					.			.
* ITOUT *					.		U	X	.			X	C	.				.			.
* JJMM *			U		.			S	.			C	C	.				.			.
* JLIM2 *					.			X	.				.					.			.
* JLIM *					.			S	.				.					.			.

U=used, S=set, X=set and used, C=first used in a call

**	*																							
* * MODULE	*	A	A	B	B	B.	C	C	C	D	D.	D	D	D	D	D.	F	F	G	I	I.	I	K	L	M	O.
* *	*	D	N	I	L	U.	R	V	V	I	I.	M	M	M	M	R.	I	L	E	N	N.	N	S	Z	I	R.
* *	*	J	I	L	K	C.	O	G	G	A	A.	P	P	P	P	I.	S	X	T	F	I.	I	C	D	N	G.
* *	*	G	S	A	D	K.	S	I	O	M	M.	F	P	R	W	V.	S	M	F	L	T.	T	C	Z	M	A.
* *	*		O	N	0	L.	I	N	U	A	2.	N	O	D	R	E.	N	O	L	P	F.	F			A	.
* *	*			Z	1	.	N	N	T	N	.	D	S		T	R.		M	X	R	A.	X			X	.
* *	*					.					.					.					.					.
* SYMBOL *	*					.					.					.					.					.
*	**					.					.					.					.					.
* JMM	*		U			.				X	.					.					.					U.
* JM1	*					.				X	.					.					.					.
* JM2	*					.				S	.					.					.					.
* KAUSW	*					.				X	.					.					.					.
* KDUM	*					.				X	.					.					.					.
* KTR	*					.				X	.					.					.					.
* LC	*					.				X	.					.					.					.
* LCDIM	*					.				S	.					.					.					.
* LTAB	*					.C				S	.					.					.					.
* MAT	*	U		U		.		U	U	X	.					.U					.					.
* MAXOUT	*					.			S	X	.					.					.					.
* MBK	*			U		U.				X	.					.					.					.
* MCM	*					.				X	.					.					.					.
* MCMML	*			C		.		U	U	X	.			C	C	.C	U	C			U.	C				.
* MCMM	*			U		.				X	.			U	C	.					.			C		.
* MLM	*			U		.				X	.			U	C	.					.			C		.
* MLM2	*					.				S	.					.					.			X		.
* MT	*					U.U				X	.				C	.					.					.
* MTJ	*			U		.				S	.					.					.					.
* MTR	*					.				X	.					.					.			U		.
* MTV	*			U		.				X	.					.					.				U	.
* MZM	*			U		.				X	.					.					.					.
* NA	*					.					C.					U.					.					.
* NFANI1	*					.				X	.					.				C.						.
* NFANI2	*					.				X	.					.				C.						.
* NFDMPN	*					.				X	.	C		C		.					.					.
* NFDMPO	*					.				X	.X		C			.					.					.
* NFI	*					.C				C	.				X	.			C		.					.
* NFO	*			U		U.U				C	C.C	C	C		C	.U		U	U	U	U.U	U				.
* NFPERT	*			C		.				X	.					.					.					.
* NFSCR1	*			X		.				X	.					.C					.C					.
* NFSCR2	*			X		.				X	.			C	C	.					.C					.
* NOPNTS	*					.				X	.					.					.					.
* NRMOM	*		C			.				U	.					.				U.						.
* NUMDMP	*					.				X	.U	S		C		.					.					.

(COMINT contd.)

U=used, S=set, X=set and used, C=first used in a call

```
----------------------.---------.---------.-----------
**         *          .         .         .          *
* * MODULE * O P P P P.P P P P P.R R R S S.S S S S S *
*  *       * U E L R R.R R R R U.E E E N O.O O O O O *
*   *      * T R O I I.I I I I T.G G G C U.U U U U U *
*    *     * E T U N N.N N N N F.I I T O A.A A B E F *
*     *    * R   T T T.T T T T L.O S S N N.N N N X I *
*      *   *       1 2.3 4 5 6 X.N   T S I.J K D T S *
*       *  *          .         .         .          *
* SYMBOL * *          .         .         .          *
*         **          .         .         .          *
----------------------.---------.---------.-----------
* ICM      *          .      U  .         .          *
* ICVT     * U        .         .         .          *
* ID       *   U      .      U  .         .          *
* IDFL     *          .      U  .         .          *
* IDMPWR   *          .         .         .          *
* ID1      *         U.      U  .         .          *
* ID2      *         U.      U  .         .          *
* ID3      *         U.      U  .         .          *
* ID4      *         U.      U  .         .          *
* ID5      *         U.      U  .         .          *
* IERR     *          .         .X   S    .          *
* IEVT     *   U      .      U  .         .          *
* IGCNT    * X        .         .        C.U U       *
* IGM      * C C   C U.  U U U  .         .    C C   *
* IGP      * U     U U.         .         .          *
* IHM      *   U   U  .         .         .          *
* IHS      *   U      .         .         .U U       *
* IHT      *   U      .         .         .          *
* IIL      *          .      U  .         .          *
* IIM      *          .      U  .         .          *
* II1      *          .         .         .          *
* II2      *          .         .         .          *
* INBO     *   U      .      U  .         .    U     *
* INDSN    *       U  .         .      U  .          *
* INORM    *   U      .      U  .         .          *
* IQUELL   *          .      U  .         .      U U *
* IQUER    *       U  .      U  .         .          *
* IS       *          .         .         .          *
* ISCT     *   U U    .      U  .         .U U       *
* ISN      *   U      .      U  .      U  .    U     *
* ISTOP    * U   S    .        S.        S.    S S   *
* ITH      *   U      .      U  .         .          *
* ITLIM    *          .         .         .          *
* ITMAX    *          .         .         .          *
* ITOUT    *          .         .         .          *
* JJMM     * C        .         .         .          *
* JLIM2    *          .         .         .          *
* JLIM     *       U  .         .         .          *
----------------------.---------.---------.-----------
```

U=used, S=set, X=set and used, C=first used in a call

MODULE / SYMBOL	OUTPUT	PERT	PRINT1	PRINT2	PUNCH1	PUNCH2	PUNCH3	PUNCH4	PUNCH5	PUNCH6	RETOUR	RETOUR1	RETOUR2	SAUT	SAUT1	SAUT2	SAUT3	SAUT4	SAUT5	SAUT6
JMM	U	C		U	U								U	U		U				
JM1				U																
JM2																				
KAUSW								U												
KDUM								U												
KTR					U			U												
LC																				
LCDIM																				
LTAB																				
MAT		C	C	U				U			U				U	U			U	U
MAXOUT																				
MBK		U						U												
MCM			C			C	C	U												
MCMML	C	C	C		C				C	C			C	U	U			U		
MCMM		U	U		U					U		U				U				
MLM		U	U		U	U	U		U		U					U				
MLM2																				
MT		U	C				U	U								C				
MTJ			U																	
MTR			U		U															
MTV			U																	
MZM									X	U	U									
NA																				
NFANI1	X							U					C							
NFANI2	X												C							
NFDMPN								U												
NFDMPO								U												
NFI			C														C	C		
NFO		U	U	U	U	U	U	U	U	U			U				U	U		
NFPERT		C						U												
NFSCR1	X							U					C							
NFSCR2	X				C															
NOPNTS					U															
NRMOM	U	U											U							
NUMDMP								U												

(COMINT contd.)

MODULE / SYMBOL	S O U	S T D	T R I	U N C	X S H	X S E
ICVT			U			
IGCNT	C		C			
IGM		U			U	U
IGP		U				
IHS	U				U	U
IHT					U	U
II1		U				
II2		U				
IQUELL		U				
ISCT		U				
ISN			U			
ISTOP		X		S		
ITLIM			U			
ITOUT			C			
JJMM		U				
JLIM2		U	C			
JLIM		U				
JMM		U				
JM1			U			
JREBA		S	U			
JREBE		S	U			
LC			X			
LCDIM		U				
L4		S				
L5		S				
L6		S				
MAT	U	U	U			
MBK		U				
MCM		U				
MCMML	U	U	C			
MCMM		U	U			
MLM		U				
MT					U	U
MTR		U	U			U
MTV			U			
NFO		C		U	U	
NFPERT			C			
NFSCR1	C	U				
NFSCR2	C	U				
NRMOM		X	U			

U=used, S=set, X=set and used,
C=first used in a call

Figure 18. Usage of COMMON /COMPOI/

U=used, S=set, X=set and used, C=first used in a call

MODULE / SYMBOL	D I A M	S T O D
JBIL1	U	X
JBIL2	U	X
JBIL3	U	X
JBUCK	U	X
JC	U	X
JEXTSO	U	X
JFG	U	X
JFGOLD	U	X
JFIN	U	X
JFLXAN	U	X
JFLXNE	U	X
JFLXOL	U	X
JFLXSK	U	X
JGRPSO	U	X
JHLP1	U	X
JHLP2	U	X
JPOINT	U	X
JS	U	X
JSS	U	X
JT	U	X
JTF	U	X
JTFOLD	U	X
JTW	U	X

MODULE / SYMBOL	D I A M	S T O D
JXAIJ	U	X
JXBIJ	U	X
JXCIJ	U	X
JXNAI	U	X
JXNAO	U	X
JXNA	U	X
JXNBI	U	X
JXNBO	U	X
JXNB	U	X
JXNCI	U	X
JXNCO	U	X
JXNC	U	X
JXSFIS	U	X
JXSSCA	U	X
JXTOT	U	X
KIFIN	U	X
KIHLP1	U	X
KMIXDI	U	X
KMIXRE	U	X
KMODBN	U	X
KMODEL	U	X
KMODAN	U	X
KPOINT	X	X

Figure 19. Usage of COMMON /COMREA/

U=used, S=set, X=set and used, C=first used in a call

SYMBOL \ MODULE	BILANZ	BUCKL	CVGINN	CVGOUT	DIAMAN	DMPRD	DMPWRT	FLXMOMM	FISSN	OUTER	PERT	PRINT1	PRINT2	PRINT6	SOUANI	SOUANJ	SOUANK	SOUFIS	TRINER	XSCHCK
BF		U			S						U			U						
BNDMAX				X	X															
BNDMIN				X	X															
CPTIM0					X															
CPTIM1					U														X	
CPTIM2					U			X												
CPTIM3					U												X			
EPS				U	X						U			U						
EPSA			U	U	X						U			U						
EPSINN					X														U	
EPSOUT				U	X															U
ERROUT				X	X															
EV	C				X	S	U			X	U			U						
EVOLD				U						S										
GINN					X														X	
H	U				X		U				U			U						
HINV					X												U		C	
OVHEAD					X					S									U	
OUTIME					X					S									U	
TEPS					X						U			U						
THRESH					S														U	
V	U				X				U	C	U		C		C		U		U	
VV					S														U	
XLA				U					X											
XLAR				U					S											

Figure 20. Usage of COMMON /COMSTO/ and BLANK COMMON

U=used, S=set, X=set and used, C=first used in a call

MODULE	B	E	F	G	P
	L	X	R	E	U
	K	T	E	T	T
	D	E	S	F	F
	0	N	P	L	L
	1	D	1	X	X
SYMBOL					
IRECNO	S			X	X
LOKLER		X	U		

MODULE	D	E	S
	I	X	T
	A	T	O
	M	E	D
	A	N	I
	N	D	S
SYMBOL			
A	C	C	C
IW	C	C	C
NACOM	X		
NW	X		

Figure 21. Usage of COMMON /FILLCO/ and COMMON /FILLCX/

U=used, S=set, X=set and used, C=first used in a call

MODULE	B	B	C	D	P	P	P	F
	I	U	R	I	E	R	R	I
	L	C	O	A	R	I	I	L
	A	K	S	M	T	N	N	L
	N	L	I	A		T	T	C
	Z		N	N		1	2	
SYMBOL								
IPSCAP	U		S			U	U	S
IPSFIS	U		S		U	U	U	S
IPSN2N			S			U		S
IPSTOT			S			U	U	S
IPSTR		U	S	X		U		S
NTRANS				S				S

MODULE	C	P	F	F
	R	R	I	I
	O	I	L	L
	S	N	L	T
	I	T	C	R
	N	1		A
SYMBOL				
SIGTYP	S	U	S	S

3.3.3 Data Files Used

All data files in DIAMANT2 are accessed by calling entries of subroutine READKO /9/. In this way, all binary I/O operations are centralized and can easily be modified (see "KAPROS Implementation" on page 69).

The data files NFSCR1 and NFSCR2 are used for intermediate binary storage of calculated scalar fluxes. NFSCR1 is an input variable (see input card K4) and NFSCR2 is equal to NFSCR1+1. They are only used if the main memory available is not sufficient to store scalar fluxes for all energy groups. Associated operating system control cards should always be supplied for reasons of safety (example given for IBM environment):

```
//G.FT"NFSCR1"F001 DD DISP=(NEW,DELETE),DCB=(RECFM=VBS,BLKSIZE=.....)
//G.FT"NFSCR2"F001 DD DISP=(NEW,DELETE),DCB=(RECFM=VBS,BLKSIZE=.....)
```

NFANI1 and NFANI2 are binary working data files which are only required for the anisotropy option and contain the calculated flux moments (operating system control cards analogous to those before). NFANI1 is an input variable (see input card K4) and NFANI2 is equal to NFANI1+1. These two files are used only if the space reserved in COMMON block /COMMOM/ (defined by a PARAMETER statement) is not sufficient to hold all flux moments. See also "Tuning Binary Input/Ouput" on page 89.

Data file NFDMPN contains incoming angular fluxes, scalar fluxes, neutron sources, and some auxiliary values in binary form at the end of each outer iteration. NFDMPO contains a flux guess in the same format used in NFDMPN. Both data files are only used if the input values of NFDMPN and/or NFDMPO on input card K2 are greater than 0. If NFDMPN = NFDMPO, the flux guess from NFDMPO is read first and the newly calculated flux is stored (after each outer iteration) behind the last flux guess on NFDMPO. The structure of this data file is described in "Structure of Restart Dump Files" on page 65.

Data file NFPERT serves as a binary interface file for the perturbation module TPTRIA. The structure of this file is described in "Structure of Interface File for Perturbation Module" on page 66.

Figure 22. Data Files und Their Use

Ref.No.[6]	Name	Type[1]	Used in	Remarks
Input	NFSCR1	I/O	GETFLX/PUTFLX/BILANZ/DIAMAN FISSN/INITFX/OUTER/PRINT6 SOUANI	[2]
Input	NFSCR2	I/O	GETFLX/PUTFLX/BILANZ/DIAMAN DMPRD/DMPWRT/INITFX/OUTER PRINT2	[2]
14	NFI	I	CROSIN/DRIVER/DIAMAN/INFLPR PLOUT/SOUBND/SOUEXT	[3]
Input	NFPERT	O	BILANZ/DIAMAN/PERT/TRINER	[4]
Input	NFDMPN	O	DIAMAN/DMPWRT/DMPOS/PRINT6	[5]
Input	NFDMPO	I	DMPFND/DMPRD/DIAMAN/PRINT6	[5]
Input	NFANI1	I/O	DIAMAN/INITFA/OUTER/PRINT6 SOUANI	[2]
Input	NFANI2	I/O	DIAMAN/INITFA/OUTER/SOUANI	[2]

Remarks:

[1] I = Read only, O = write only, I/O = read and write

[2] Scratch data sets

[3] Input data set (FORTRAN unit number 14)

[4] Interface file for TPTRIA perturbation module
(see "Structure of Interface File for Perturbation Module" on page 66)

[5] Restart files (see "Structure of Restart Dump Files" on page 65)

[6] Input means that the unit reference number is specified as input variable

3.3.4 Storage Optimization Levels

Depending on input variables and on the available computer main memory, DIAMANT2 has two levels of operation:

- in the first one, Optimization 1, the whole problem can be solved core contained (which is preferable from the standpoint of computing time)
- in the second one, Optimization 2, the fluxes of one group are in-core, the other ones being stored on disks NFSCR1 and NFSCR2. For reasons of safety, these two files should be defined in any run.

The length of the working array needed depends on chosen input options (see "How to Estimate Storage Needs" on page 81). Its estimation is a very tedious process. It is recommended to let do this the input check module (see "Input Test Module DIAPRD" on page 78). If possible one should avoid to run the code in optimization level 2 because of performance degradation. In order to avoid such a degradation, one should try the following: for the KAPROS version one simply provides more main memory space by increasing the REGION parameter on the JOB card; for the stand-alone version one has to increase the values NAMAX and NWMAX in the PARAMETER statement (which normally means an increase in the requested job REGION too) and to re-compile the code.

3.4 INTERFACES TO OTHER PROGRAMS

3.4.1 Structure of Restart Dump Files

These dump-files are used to provide a restart capability to DIAMANT2 as well as to provide flux and source guesses. They are read/written in subroutines DMPRD/DMPWRT. The logical structure is subdivided into variable length records. For a definition of the dimensioning variables see Figure 10 on page 46.

Record 1 Label record (see Figure 23 on page 67), 8 integer numbers and 2 reals

Next 3*IGM Records

JMM*JLIM2 real numbers written out separately for each possible orientation of the system boundary and for each energy group.

next IGM records

MCMML real numbers representing the (centered) scalar flux values for each energy group

next record MCMML real numbers representing the fission source density

last record IGM+1 real numbers for the groupwise volume integrated fission source (entry IGM+1 contains the group sum)

All records are read/written in binary form, i.e., without format control. Dumps from several different calculations could be contained on a single dump-file. The program identifies them by their relative position on that file. The end of the dump-file is marked by a record containing 10 negative integer numbers.

If a certain restart dump is requested to be used, the code checks whether the structure of that particular dump fits to the current input data. If not so, DIAMANT2 reads the whole restart file and selects the first dump having suitable dimensions. If this procedure fails too, DIAMANT2 uses a code generated (flat) source guess. Warning messages are printed in any case.

3.4.2 Structure of Interface File for Perturbation Module

This interface file is presently used by the perturbation module TPTRIA /3/. It contains - besides some control information - the angular fluxes calculated during the last outer iteration for each group. Again this is a binary file. For a definition of the dimensioning variables see Figure 10 on page 46.

Record 1 input card K1, 60 characters label

Record 2 from input card K2 variables
ID, ITH, ISN, IGM, IEVT, MLM, MCMM(=2*MCM), MT, MAT, INBO

ID	Identification of the run (60 characters)
ICC	number of the outer iteration
ISN	angular quadrature order
MCMM	2*MCM;number of meshes along the horizontal side of the reference parallelogram
MLM	number of intervals on the oblique side of the reference parallelogram
EV	eigenvalue
MT	number of mixtures
IGM	number of energy groups
NUMFL	number of the flux guess map
H	length of a triangle side (in cm)

Figure 23. Structure of the Label Record in Restart Dump Files

Record 3 contains from input card K4 variables
ISCT, MBK, INORM, and
IHS = position of within-group scattering term in array C
IHT = position of total cross section term in array C
IHM = number of rows in a cross-section table (= IHT+IGM)
IPSFIS = position of SFISS term in array C
JMM = total number of directions used
NRMOM = number of Legendre moments stored in calculations having anisotropic scattering

Record 4 from input card K5 variables
EPS, EV, BF, H, EPSA, TEPS, V = volume of triangular mesh cell (it is supposed that each mesh cell has a height of 1 cm)

Record 5 MT variables from input card K7

Record 6 MT integers identifying the adress of zeroth moments in array C

Record 7 MAT integers denoting which mixture number corresponds to each adress of zeroth moments

Record 8 JMM integration weights

Record 9 cross section table C, IHM*IGM*MAT real values

Record 10 IGM real variables representing the inverse velocities associated with each energy group

Record 11 Input buckling values
if MBK = 1 then BF from input K5 is written

if MBK = 2 then input K14A is written (IGM values)
if MBK = 3 then input K14B is written (IGM*MT values)

Record 12 MCMML integer values containing the mixture numbers associated with each spatial cell, starting from upper left triangle to lower right triangle and row-wise from left to right.

Record 13 JMM integer values identifying the sequence in which discrete directions are worked on

Record 14 (written only in anisotropic cases)
NRMOM*JMM real values containing the spherical harmonics values for each direction.

Records 15 - 17 are repeated for each energy group and for each "real" direction (out of the JMM discrete directions defined, only JMM-ISN are used, the remaining ISN being dummy).

Record 15 one integer number identifying the actual group

Record 16 one integer number identifying the order number of the actual direction

Record 17 MCMML real numbers containing the (centered) angular fluxes for that group and that direction in same order as in Record 12

Records 18 - 22 may appear several times (see remark 4 below).

Record 18 Number of groups (same as variable IGM)

Record 19 Number of records 21,22 (variable IND)

Record 20 Criticality factor

Records 21,22 are repeated IND-times (see remarks 1, 2, 3 below)

Record 21 number of table given as record 22

Record 22 total fluxes (IGM real values)

Remarks

1. Variable IND is equal to 1 for the first set of records 18 to 22. The groupwise volume integrated fluxes then correspond to those of global balances. These total fluxes are always written.

2. Variable IND is equal to MT if KAUSW is equal to 10 or 11 (KAUSW is an input variable on input card K4 and triggers the calculation of balances). In this case the groupwise total fluxes are written separately for each mixture number (i.e., MT times).

3. Variable IND is equal to the number of different regions in the layout if KAUSW = 1 or 11. In this case the total fluxes are written separately for each region.

4. Depending on the value of KAUSW, the set of records 18 to 22 is written once (KAUSW = 0) or twice (KAUSW = 1 or 10) or three times (KAUSW = 11).

5. If the interface file has been generated by an adjoint calculation one should be aware of the different interpretation of group order and angular orientation (see "Adjoint Calculations" on page 23). With exception of the cross section matrix C, all values are given in reverse group ordering; angular fluxes of direction $\vec{\Omega}$ have to be interpreted as belonging to $-\vec{\Omega}$.

3.5 KAPROS IMPLEMENTATION

Within KAPROS it is possible to use datablocks instead of disk data sets. Using the READKO /9/ package one has the possibility to determine at program running time whether a data file should be realized as a KAPROS data block or as a (module-owned, external) tape or disk data file.

Figure 24 gives a description of the (proposed) content of DBN = INITÞREADÞDIAMAN which must be provided in the KAPROS lifeline when calling DIAMANT2. It is read by READKO and controls the realization of the individual data files. Basically, all options listed in /9/ can be selected, although it is advised not to deviate from the recommended values.

If NFDMPN = NFDMPO, unit NFDMPN has the same data block name as NFDMPO; only the space requirement for the actual case under consideration is indicated under file length for NFDMPN in Figure 24.

Within the KAPROS system, the subroutine package FILLC allows DIAMANT2 to read nuclear cross-sections from a binary file having the so-called SIGMN /11/ structure. This option can ease input preparation quite a lot since cross section data has not to be entered by hand.

Within the KAPROS system, NACMAX (length of the real working array) as well as NWMAX (length of the integer working array) are equal to one. After having determined the amount of storage needed (depending on input values), DIAMANT2 calls a special KAPROS routine, KSPUTP, to dynamically extend arrays. The real storage is provided in a KAPROS data block DBN=DIAM2␣REAL*4 with a corresponding pointer JPOINT (relative to A(1)), the integer storage is located in DBN=DIAM2␣INTEGER*4 with a corresponding pointer KPOINT (relative to IW(1)). The distribution of the working arrays is the same as in the stand-alone version.

DIAMANT2 is stored in the KAPROS module library with name DIANEU and is invoked by either the command

```
*GO SM=DIANEU
```

or the call

```
CALL KSEXEC('DIANEU␣␣', ....).
```

See /4/ for further details. Note that at least the input data blocks DBN=DIAMANT2␣EINGABE and DBN=INIT␣READ␣DIAMAN, both with index 1 have to be known at the module level of DIAMANT2.

Ref.No.	Var.Name	DBN	Index	Realization[1]	[2]	[3]
Input	NFSCR1	'SCRATCHbUNITb1bb'	1	6	0	1
Input	NFSCR2	'SCRATCHbUNITb2bb'	1	6	0	1
14	NFI	'DIAMANT2bEINGABE'	1	1	0	0
Input	NFPERT	'TPTRIAbINTERFACE'	1	2/6	1/0	1
Input	NFDMPN	'DIAMANT2bGUESSbN'	1	1/2/6	0/1/0	1
Input	NFDMPO	'DIAMANT2bGUESSbO'	1	1/6	0	0
Input	NFANI1	'SCRATCHbUNITb3bb'	1	6	0	2
Input	NFANI2	'SCRATCHbUNITb4bb'	1	6	0	2

Remarks:

[1] **Realization** 1 = Existing KAPROS data block;
2 = new KAPROS data block;
6 = tape or disk data file (compare also /9/)

[2] **Length** 0 = Not important for existing data files or not generally to be indicated;
1 = variable length depending on input, see "Interfaces to Other Programs" on page 65; a value large enough to contain all data should be entered, though READKO will extend the datablock if necessary.

[3] **Format** 0 = Not important for existing data files or data blocks;
1 = variable record length;
2 = fixed record length.

Figure 24. Recommended Content of DBN = INITbREADbDIAMAN, IND = 1

3.6 USE AS STAND-ALONE PROGRAM

3.6.1 General Remarks

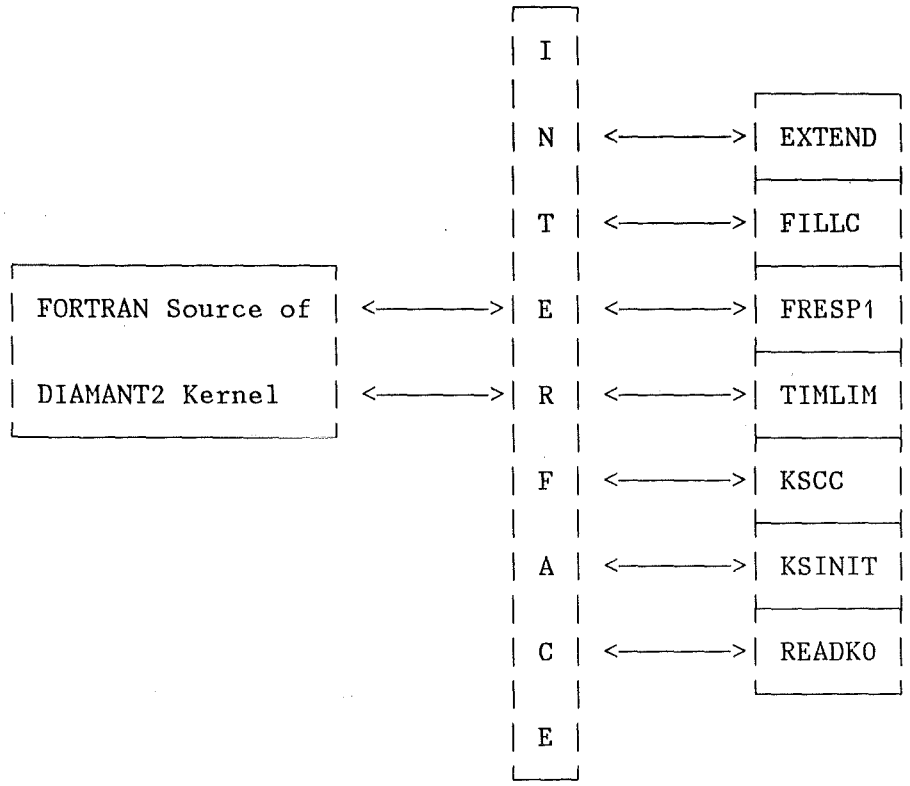
The DIAMANT2 code is designed such that it is very easy to produce a version which runs without support by KAPROS ("stand-alone" version) as well as an equivalent version running under control of KAPROS. The following diagram illustrates schematically those portions of DIAMANT2 which are directly or indirectly dependent upon KAPROS and which differ in coding between the KAPROS and the stand-alone version.

Most of the KAPROS dependent subroutines serve as aids in input preparation (FILLC, READK0) or add dynamic features (EXTEND) to DIAMANT2. As will be illustrated in the following, the KAPROS dependent portions are very easily replaced by stand-alone simulations.

3.6.2 Subroutine EXTEND

Used to distribute working arrays among code arrays.

KAPROS-version	Dynamically allocates pointer data blocks to hold working arrays
Stand-alone version	Allocates in blank COMMON/ / the amount of storage place needed according to input specifications



FUNCTION TASKTI is also dependent on the computer system used but is usually available under some name for most computers.

Figure 25. Dependence of DIAMANT2 on KAPROS

3.6.3 Subroutine FILLC

Special KAPROS interface routine package.

KAPROS-version Transfer of nuclear cross sections from KAPROS SIGMN data blocks and stores them in array C within DIAMANT2 (see Figure 34 on page 109).

Stand-alone version Dummy for stand-alone version; nuclear cross sections have to be provided with the input stream.

3.6.4 Subroutine FRESP1

Used for informative output only.

KAPROS-version Amount of free storage places in the allocated computer main memory

Stand-alone version Gives the amount of working storage still not allocated in working arrays (defined by PARAMETER's NACMAX and NWMAX).

3.6.5 Function TASKTI

Returns the CPU-time (in seconds) used already for the task up to the moment of calling TASKTI. Used to calculate timing statistics.

KAPROS-version Uses a compiler service function

Stand-alone version Uses a compiler service function (such a function is usually provided along with a compiler (like SECOND for CRAY and CYBER, CLOCKM for SIEMENS; or one has to program some simulation like JTIME for IBM in IBM ASSEMBLER).

3.6.6 Function TIMLIM

Returns the remaining CPU-time (in seconds) for the job. Used to prevent a task breakdown due to exhaustion of CPU resources and to allow a controlled termination.

KAPROS-version Uses an ASSEMBLER function JTIME out of the KAPROS kernel system

Stand-alone version A machine dependent routine has to be used (TREMAI for CRAY, JTIME for SIEMENS and IBM, JTIME simulation for CYBER). For IBM, SIEMENS, CRAY and CYBER running ver-

sions of TIMLIM are provided along with the source deck of DIAMANT2.

3.6.7 Subroutine KSCC

Used to respond to erroneous situations.

KAPROS-version	Sets and re-sets error codes; part of the KAPROS kernel
Stand-alone version	executes a STOP statement in case of any error

3.6.8 Subroutine KSINIT

Special KAPROS interface routine.

KAPROS-version	Initialization of the KAPROS kernel
Stand-alone version	Initializes unit numbers for standard read/print files

3.6.9 Subroutine READK0

This is an interface package for binary input/output operation. It is fairly simple for the stand-alone-version and very elaborated for the KAPROS version. For details see /9/.

See "Tuning Binary Input/Ouput" on page 89 for details how one may improve performance of I/O operations on computers having a very large main memory.

CHAPTER 4. TEST EXAMPLES AND USER INFORMATION

4.1 TEST CALCULATIONS CARRIED OUT

An elaborated series of test calculations has been carried out with DIAMANT2. These will be documented separately /14/. Here we will give only an overview about the test cases and the main findings from those calculations. The following is a list of the major test cases executed:

- k_∞ for infinite homogeneous regions;
- Independance of results on the position of the model in the reference frame;
- adjoint calculations;
- anisotropy option;
- buckling options;
- external source calculations;
- usage of restart dumps;
- runs in different storage optimization levels.

In addition to these, a number of parameter variations have been done in order to estimate the influence of the input variables EPS, EPSA, IIM, IIL and ISN on numerical results and computing time. Derived recommendations may be found in "Experience From the Test Calculations" on page 78. Input for most of the test examples is distributed along with the source code.

4.2 INPUT TEST MODULE DIAPRD

A separate module DIAPRD is available for input testing again in a stand-alone version and as KAPROS module. DIAPRD reports the input in clear text and produces a sketch of the mixture distribution. The application of this check module appears to be particularly useful for testing newly produced inputs because it uses almost no computer resources.

In addition to input testing, it also calculates how many words of main memory are required for the individual working fields. DIAPRD provides information concerning these values as well as working data files which may be necessary (see also "Storage Optimization Levels" on page 65).

DIAPRD is stored in the KAPROS module library and is invoked by specifying

*KSIOX DBN=.....,PMN=DIAPRD

See /4/ for further details. Note that at least the input data blocks DBN=DIAMANT2␣EINGABE and DBN=INIT␣READ␣DIAMAN, both with index 1 have to be known at the module level of DIAPRD.

4.3 EXPERIENCE FROM THE TEST CALCULATIONS

4.3.1 How to Choose the Quadrature Set

For problems dominated by the diffusion of neutron through optically thick regions, a P_N approximation may be expected to yield superior results to the DP_N approximation. In contrast, the DP_N quadrature may be preferable for problems dominated by the presence of boundaries, particularly vacuum boundaries. A more detailed discussion of these aspects may be found in /8/. The differences between P_N and EP_N sets are usually small. For details how the quadrature set is constructed see "Construction of S_N-Constants" on page 8 and "Appendix A. DIAMANT2 Built-in Quadrature Constants" on page 119.

S_N-orders ranging from 2 to 12 are built-in. One should bear in mind that DIAMANT2 for a given order ISN has 50% more directions compared to conventional 2d geometry S_N-codes (because of the division of the unit sphere into dodecants instead of the usual octants). In our test cases results did not change very much beyond ISN=8.

4.3.2 How to Choose Iteration Control Parameters

There are basically five input values to control the iteration process. The easiest one to choose is ICM, the maximum number of outer iterations. In our tests we used values between 50 and 100. Compared to the old version /1/, the new code needs more outer iterations but the total number of inner iterations is not changed very much or is even lower. The value of ICM is not very important because a threatening time limit break down of the task will be prevented by the timing function TIMLIM.

The next two parameters, IIL and IIM, are the maximum number of inner iterations for the first outer iteration and the absolute maximum number not to be exceeded in any outer iteration, respectively. IIL is increased from one outer iteration to the next by 1 until IIM is reached. For an efficient iteration, IIL should be small (something like 2 or 5) and IIM large (something like 20 or 30). When using a restart dump, IIL should be larger (something like 10). This is especially important if one uses a dump to start a calculation with anisotropic scattering.

The parameter EPSA is the precision wanted for scalar fluxes. How "precision" is defined and estimated is described in "Inner Convergence Tests and Convergence Acceleration" on page 19. Since the actual precision threshold during inner iterations is set to
EPSINN = MAX {MIN { EPSINN, $0.1*\varepsilon_{outer}$} ,ABS(EPSA) },
which depends on the convergence ε_{outer} achieved for sources, the number of inner iterations does not depend very much on the value of EPSA during the early phase of the iteration process. Values of EPSA in the range 1.0E-3 to 1.0E-5 have been used in our tests.

The parameter ε is the precision wanted in outer iterations. How "precision" is defined and estimated is described in "Outer Iteration Procedure" on page 21. Values in the range 1.0E-4 to 1.0E-5 have been used in our tests.

Since there are several convergence tests available, we should give an indication how tight these tests are. The integral test is less stringent than the pointwise test and this is less stringent than the test based on eigenvalue bounds. But note that in some cases, the integral test must be used because of spurious fluxes which oscillate. Pointwise testing then could result in endless iterations without improving accuracy. Such fluxes may occur at the systems boundaries or especially in "dummy" regions introduced to make the calculational model convex (see "Mixture Distribution" on page 106).

4.3.3 How to Estimate Computing Time

CPU time can be estimated as:

(42a) v*NDIM*NWORK, where

(42b) NDIM = (no. of spatial meshes)*(no. of angles)*(no. of energy groups)

(42c) NWORK = total number of inner iterations

v is the time needed to process one mesh of the real layout (per group and angle). It is printed out in the timing statistics at the end of each calculation.

In our isotropic scattering test cases the code executed at a speed $v_0 \approx 3$ µs per mesh (on IBM3090). In anisotropic scattering cases (P3) we found $v_3 \approx 20$ µs. See Figure 26 for some typical examples of computing times. See "Experiences on Vector Computers" on page 83 for the improvement gained in v_0 and v_3 by using vector computers.

ISCT	Grid size / no. of meshes	Storage kBytes	CPU time sec	Speed µsec	taver msec
0	17x17/ 289	568	23	3.2	16.1
0	38x38/1156	736	78	2.8	55.1
0	57x57/2601	984	175	2.8	122.4
3	17x17/ 289	776	66	19.2	99.4
3	38x38/1156	1308	261	18.4	384.3
3	57x57/2601	2164	585	18.8	880.3

ISCT = anisotropy order; taver = average time per inner iteration; Storage = needed for working arrays.

All values given for S7890 computer; all examples use four energy groups and S4 approximation; convergence criteria are 1.0E-4 for sources and fluxes based on pointwise tests.

Figure 26. Examples for Storage, CPU Times and Execution Speed

4.3.4 How to Estimate Storage Needs

To a first order approximation the storage amount needed is:

(43) Storage = (Codelength) + (Grid Size)*(A + B*IGM) + C*IGM in words

where A = 11, B = 2 and C = 6*JJMM and Grid Size is the number of meshes of the reference parallelogram. Using the anisotropy option these values change to A = 11+2*NRMOM+JMM, C = 6*JJMM+JMM*NRMOM (B same as before). IGM is the number of energy groups, JMM the number of discrete directions, JJMM the number of intervals on one side of the reference parallelogram times JMM and NRMOM the number of stored anisotropic moments. On IBM Codelength ≈ 70000 words.

According to the authors' experience it is easiest to let the code estimate its exact storage need for a particular case. One easily could start the DIAMANT2 or DIAPRD (see "Input Test Module DIAPRD" on page 78) with a small amount of CPU time and then analyse the output. On the first few pages there is a complete information on storage needs. For reasons of efficiency, any calculation should be run in storage optimization 1 (i.e. fully core-contained) as long as there is the possibility to get enough main memory from the operating system. See Figure 26 on page 81 for some typical examples of storage requirements and "Storage Optimization Levels" on page 65 for recommendations.

4.3.5 Discretization Accuracy

In criticality calculations we observed a fairly good linear dependence of k_{eff} on the (constant) mesh volume. This means that at least for this parameter, DIAMANT2 is $O(h^2)$ accurate.

On CYBER-205, the code executes in eight bytes mode. Comparison of results obtained from test calculations on CYBER-205 and IBM led us to the statement that usually single precision is sufficient on IBM equipment (i.e. in four bytes mode) to get reliable results of 4-5 digits in k_{eff} and integrated quantities and 3-4 digits in fluxes.

4.3.6 The Role of Negative Flux Fix-Up

In the present version, negative flux fix-up is activated only after convergence of iterations. Usually only very few meshes need a fix-up and correspondingly only a few (1 to 5) additional iterations are needed for final convergence. The number of meshes which suffered a fix-up step is printed in the iteration monitor. If this number is large compared to the total number of meshes, results will probably have a lower accuracy than mentioned above. In these cases one may try to avoid negative fluxes by using a finer spatial mesh grid and a higher S_N order.

4.4 EXPERIENCES ON VECTOR COMPUTERS

DIAMANT2 has been successfully run on CYBER-205, CRAY, IBM-VF and SIEMENS/FACOM VP vector computers. Only the most time consuming routines (see "Function of the Different Subroutines" on page 41) have been compiled with "autovectorization on". Note that DIAMANT2 is completely written in FORTRAN77 (except for the timing functions). The timings giving in the following refer to a common source deck. At a few places we have inserted compiler directives to improve vectorization. In the case of the CYBER-205, the autovectorizer FTN200 does not substantially vectorize the code. The CYBER-205 entries "autovectorized" given in the table have been obtained using the VAST /16/ preprocessor. It is clear, that using special language extensions on a vector computer could enhance the performance at the cost of creating a less portable code.

In DIAMANT2, the vector length for the most time consuming loops is usually the length of one line of meshes during space-angle mesh sweeps in inner iterations. Moreover, the length of a line can be different from line to line. At the top of the iteration monitor the code prints some information on the vector lengths (minimum, maximum and average) encountered. There are many loops having a length equal to the total number of spatial meshes, but these do normally not contribute much to CPU time consumption.

We have run isotropic calculations with an average vector length of 12, 27 and 41. There we observed speed-ups in the range of 2 to 4 (see Figure 27 on page 84). The vectorization degree (defined as the ratio of total time to vector time) has been measured on VP100 to be 0.5 (for vector length 12) and 0.8 (for vector length 41). Higher speed-up and higher vectorization degree can be expected for larger cases or by hand-tuning of a couple of loops in subroutine TRINER.

The ranking of vector computers in the above examples is CRAY-XMP : CYBER-205 : SIEMENS-VP : IBM-VF, reflecting both, hardware and compiler efficiency.

For cases using a Legendre expansion of the scattering kernel, vectorization has a much more pronounced effect. Using that option necessitates to update

Computer type	Vector length[1]	Speed µsec[2]	Speed up[3]	Speed up[4]	Performance[5]
CYBER[6]	12	5.516 / 3.096	1.8	0.9	0.47
	27	4.780 / 1.732	2.8	1.3	0.48
205	41	4.772 / 1.300	3.7	1.7	0.47
CRAY[6]	12	2.968 / 1.216	2.4	2.2	0.88
	27	2.756 / 0.816	3.4	2.8	0.83
XMP	41	2.552 / 0.692	3.7	3.2	0.88
FACOM[6]	12	2.616 / 1.912	1.4	1.4	
	27	2.284 / 1.008	2.3	2.3	1.00
VP100	41	2.240 / 0.740	3.0	3.0	
IBM[6]	12	2.432 / 2.096	1.2	1.2	1.08
	27	2.116 / 1.408	1.5	1.6	1.08
VF	41	2.108 / 1.328	1.6	1.7	1.06
CYBER[7]	12	6.292 / 3.068	2.1	1.0	0.50
XMP[7]	12	3.682 / 1.272	2.9	2.5	0.86
VP100[7]	12	3.172 / 1.963	1.6	1.6	1.00
VF[7]	12	2.964 / 1.659	1.8	1.9	1.07

[1]The average length of lines is the dominating vector length in cases using isotropic scattering.
[2]Given in the form : scalar / autovectorized CPU time.
[3]Speed-up given as ratio of scalar over autovectorized computing time.
[4]Speed-up given relative to the execution time on a SIEMENS7890
[5]Performance given as ratio of scalar execution time on SIEMENS7890 over scalar execution time on vector computer.
[6]Using 4 energy groups and an S_4 approximation.
[7]Using 26 energy groups and an S_6 approximation.

Figure 27. CPU Times for Some Vector Computers (Isotropic Scattering)

Computer type	Vector length[1]	Speed µsec[2]	Speed up[3]	Speed up[4]	Performance[5]
CYBER[6]	578	34.23 / 5.308	6.4	2.4	0.38
	2592	30.98 / 2.952	10.5	3.9	0.37
205	6050	30.10 / 2.340	12.9	5.1	0.40
CRAY[6]	578	18.23 / 2.336	7.8	5.5	0.71
	2592	16.71 / 1.712	9.8	6.7	0.69
XMP	6050	16.58 / 1.540	10.8	7.8	0.72
FACOM[6]	578	12.92 / 2.972	4.4	4.4	
	2592	11.50 / 1.768	6.5	6.5	1.00
VP100	6050	11.98 / 1.456	8.2	8.2	
IBM[6]	578	11.35 / 6.716	1.7	1.9	1.14
	2592	10.30 / 5.388	1.9	2.1	1.12
VF	6050	10.45 / 5.804	1.8	2.1	1.15
CYBER[7]	578	81.12 / 7.956	10.2	3.3	0.32
XMP[7]	578	47.67 / 4.078	11.7	6.4	0.55
VP100[7]	578	26.23 / 4.890	5.4	5.4	1.00
VF[7]	578	22.20 / 12.30	1.8	2.1	1.18

[1]In anisotropic cases the number of meshes to be calculated is the dominant length.

[2]Given in the form : scalar / autovectorized CPU time.

[3]Speed-up given as ratio of scalar over autovectorized computing time.

[4]Speed-up given relative to the execution time on a SIEMENS7890

[5]Performance given as ratio of scalar execution time on SIEMENS7890 over scalar execution time on vector computer.

[6]Using 4 energy groups, S_4 approximation and P3 expansion.

[7]Using 26 energy groups, S_6 approximation and P3 expansion.

Figure 28. CPU Times for Some Vector Computers (Anisotropic Scattering)

the within-group scattering source term for each discrete angle. Therefore, much of the computing time is spent there (subroutine SOUANK) and in the calculation of flux moments (subroutine FLXMOM). Fortunately, one can formulate those calculations as loops over all meshes of the model and, consequently, one has much larger vectorlengths for the time consuming loops. In the tests mentioned above we got vector lengths 578, 2592 and 6050, respectively (using the same spatial mesh as in the isotropic cases). As can be seen from Figure 28 on page 85 speed-up in P3 calculation increases to values between 6 and 13. Only the IBM-VF shows a smaller speed-up in the range of 2.

The vectorization degree (defined as the ratio of total time to vector time) has been measured on VP100 to lie between 0.9 and 0.95. The ranking of vector computers in the P3 examples is again CYBER-205 : CRAY-XMP : SIEMENS-VP : IBM-VF.

In conclusion, we may say that we have set up a FORTRAN77 version of DIAMANT2 which benefits as much as possible from vector computers without sacrificing scalar performance. Any further improvement with respect to advanced computer architectures (using the same algorithm) seems to be possible only by programming differing versions for scalar and vector computers (and probably the resulting vector versions will be highly machine dependent).

4.5 HOW TO CHOOSE PARAMETER VALUES

Besides setting maximum values for a couple of variables and selecting code options, there are some PARAMETERs which are system dependent. Figure 30 on page 88 shows how these values should be chosen on different computers and show as well the differences between KAPROS and stand-alone version. These values have led to a satisfactory performance during our tests mentioned above. Note that the values (especially the recommendations for MINVEC) may change with the system and the compiler release. The values for working arrays (NACMAX, NWMAX and LENCOM) are adjusted to be big enough to carry out the calculations mentioned in Figure 28 on page 85.

---------- INTEGER VALUES ----------	
IGMMAX	MAXIMUM NUMBER OF ENERGY GROUPS
IGPMAX	IGMMAX + 1
IREGMX	MAXIMUM NUMBER OF DISJOINT MIXTURE REGIONS
ISCTMX	MAXIMUM ORDER OF ANISOTRPIC SCATTERING (see remark)
ISNMAX	MAXIMUM ORDER OF THE ANGULAR DISCRETIZATION (see remark)
JMMMAX	JMMMAX=ISNMAX+ISNMAX*(ISNMAX+2)*3/4
LENCOM	LENGTH OF WORKING ARRAY IN COMMON BLOCK /COMMOM/
LPMAX	MAXIMUM LENGTH OF A SINGLE DO-LOOP (SIGNIFICANT ONLY FOR CYBER)
MATMAX	MAXIMUM NUMBER OF XS-TABLES
MAXFIL	MAXIMUM VALUE FOR FORTRAN UNIT NUMBERS
MINVEC	MINIMUM VECTOR LENGTH REQUIRED TO EXECUTE LOOPS IN VECTOR MODE
MTJP	MIXTURE NUMBER TO REPRESENT BOUNDARY FLUX CONDITION IN INPUT (= MATMAX + 3)
MTRP	MIXTURE NUMBER TO REPRESENT REFLECTIVE BOUNDARY CONDITION IN INPUT (= MATMAX + 2)
MTVP	MIXTURE NUMBER TO REPRESENT VACUUM BOUNDARY CONDITION IN INPUT (= MATMAX + 1)
NACMAX	NUMBER OF WORDS RESERVED FOR REAL STORAGE
NCMAX	TOTAL NUMBER OF WORDS RESERVED (NCMAX = NACMAX + NWMAX)
NIVMAX	NIVMAX=ISNMAX*(ISNMAX+2)/8
NWMAX	NUMBER OF WORDS RESERVED FOR INTEGER STORAGE
---------- REAL VALUES ----------	
EPSCPU	REAL VALUES BELOW THIS THRESHOLD ARE CONSIDERED TO BE ZERO
EPSMIN	MINIMUM VALUE FOR CONVERGENCE CRITERIA
PI	VALUE OF THE CONSTANT PI
PIHALF	PI/2.
PISIXT	PI/6.
---------- LOGICAL VALUES ----------	
LREBAL	INNER ITERATION REBALANCING ACCELERATION
LTRANS	PERFORM P1 CORRECTION IN XS-TABLE
LXSCHK	REPLACE INPUT ABSORPTION XS BY CALCULATED XS

Remark: Redefining ISCTMX or ISNMAX requires a change in DATA statements in subroutines ANISO and SNCONS (see source deck).

Figure 29. Definition of PARAMETER Values Used in DIAMANT2

PARAMETER	IBM/SIEMENS	CYBER	CRAY	KAPROS
IGMMAX	26	26	26	26
IREGMX	30	30	30	30
ISNMAX[1]	12	12	12	12
ISCTMX[2]	6	6	6	6
LPMAX[3]	65535	65535	65535	65535
MATMAX[4]	89	89	89	89
MAXFIL	40	40	40	40
MINVEC	5	10	6	1
NACMAX	450000	450000	450000	1
NWMAX	50000	50000	50000	1
LENCOM	300000	300000	300000	1
EPSCPU	1.0E-20	1.0E-20	1.0E-20	1.0E-20
EPSMIN	1.0E-6	1.0E-14	1.0E-14	1.0E-6
LREBAL	.TRUE.	.TRUE.	.TRUE.	.TRUE.
LTRANS[5]	.FALSE.	.FALSE.	.FALSE.	.TRUE.
LXSCHK	.TRUE.	.TRUE.	.TRUE.	.TRUE.

Remarks :

[1]If ISNMAX is to be increased, the initialization part in subroutine SNCONS has to be increased too.

[2]If ISCTMX is to be increased, the initialization part in subroutine ANISO has to be increased too.

[3]The value of LPMAX is significant for CYBER-205 only; For all other computers a bigger value may be chosen.

[4]If MATMAX is changed, then in turn MTVP, MTRP and MTJP (the mixture numbers identifying boundary conditions, see Figure 29 on page 87) will change accordingly.

[5]LTRANS is interpreted only in subroutine FILLC; therefore the value given will modify operation only in the KAPROS version of DIAMANT2.

Figure 30. PARAMETER Values Used on Different Computers

4.6 TUNING BINARY INPUT/OUPUT

Scratch units NFANI1 and NFANI2 were needed in the old version in all applications using anisotropic scattering expansion irrespective of the amount of main memory available. Therefore, we have introduced a new COMMON block /COMMOM/ serving as a container array for the data being moved from and to NFANI1 and NFANI2. The size of this block is determined by specifying the PARAMETER value LENCOM. Automatic transition is made from disk I/O to storage in COMMON /COMMOM/ whenever the space reserved there (PARAMETER LENCOM) is sufficient to hold all the data. Using /COMMOM/ instead of disks saves I/O overhead. An informative message is issued whenever LENCOM is big enough to avoid disk I/O.

On a distinct computer there usually exist even more powerful methods to tune I/O (i.e. implicit I/O on a CYBER-205 or using on IBM MVS/XA systems so-called Virtual Input Output scratch datasets which can be located in the extended address area and are very cheap and fast in access), but these methods are not portable.

CHAPTER 5. KAPROS SHORT DESCRIPTION OF MODULE DIAMANT2

1.PROGRAM IDENTIFICATION *DIAMANT2 (VERSION 2.0) /1/*

A MULTIGROUP NEUTRON TRANSPORT CODE IN REGULAR TRIANGULAR GEOMETRY

2.COMPUTER FOR WHICH THE PROGRAM IS DESIGNED AND OTHERS ON WHICH IT IS OPERABLE

IBM/370, IBM30xx, SIEMENS 7.8xx, FUJITSU FACOM 380, CYBER-205, CRAY-1, CRAY-XMP, SIEMENS VP SERIES, FUJITSU VP SERIES, IBM3090-VF SERIES.

3.DESCRIPTION OF FUNCTION

DIAMANT2 SOLVES THE TWO-DIMENSIONAL STATIC MULTIGROUP NEUTRON TRANSPORT EQUATION IN PLANAR REGULAR TRIANGULAR GEOMETRY. BOTH REGULAR AND ADJOINT, INHOMOGENEOUS AND HOMOGENEOUS PROBLEMS SUBJECT TO VACUUM, REFLECTIVE OR INPUT SPECIFIED BOUNDARY FLUX CONDITIONS ARE SOLVED. ANISOTROPY IS ALLOWED FOR THE SCATTERING SOURCE. VOLUME AND SURFACE SOURCES ARE ALLOWED FOR INHOMOGENEOUS PROBLEMS.

4.METHOD OF SOLUTION

THE DISCRETE ORDINATES APPROXIMATION FOR THE ANGULAR VARIABLE IS USED IN FINITE DIFFERENCE FORM WHICH IS SOLVED WITH A DIAMOND DIFFERENCE APPROXIMATION. SPATIAL DISCRETIZATION IS PERFORMED ON A MESH, COMPOSED OF EQUILATERAL TRIANGLES. NEGATIVE FLUXES ARE ELIMINATED BY A STEP SCHEME ALGORITHM. STANDARD INNER (WITHIN-GROUP) ITERATIVE CYCLES ARE ACCELERATED BY GLOBAL REBALANCING.

5.RESTRICTIONS

VARIABLE DIMENSIONING IS USED SO THAT ANY COMBINATION OF PROBLEM PARAMETERS LEADING TO A WORKING ARRAY LESS THAN NCMAX CAN BE ACCOMODATED. ON MOST COMPUTERS, NCMAX CAN BE AS LARGE AS SOME HUNDRED THOUSAND WORDS. THE REMAINING MAXIMUM VALUES ARE SPECIFIED BY A PARAMETER STATEMENT AND CAN EASILY BE CHANGED.

6.PARTICULARITIES OF THE PROGRAM

THE CODE IS WRITTEN IN FORTRAN 77 (ANSI X3.9-1978). QUADRATURE SETS ARE TAILORED TO REGULAR TRIANGULAR GEOMETRY/2/. PROVISION IS MADE FOR CREATION OF INTERFACE OUTPUT FILES FOR ANGULAR INTEGRATED FLUXES AND ANGULAR FLUXES. THESE INTERFACE FILES CAN EASILY BE APPLIED IN PERTURBATION CALCULATIONS, E.G., BY THE TPTRIA CODE/4/. ALL BINARY INPUT/OUTPUT-OPERATIONS ARE LOCALIZED IN THE SUBROUTINE PACKAGE READKO. EDIT OPTIONS AND DUMP AND RESTART CAPABILITIES ARE PROVIDED. DIAMANT2 IS DESIGNED TO BENEFIT FROM VECTOR-COMPUTERS AS MUCH AS POSSIBLE.

7.MACHINE REQUIREMENTS

UP TO FOUR SCRATCH UNITS, UP TO THREE INTERFACE UNITS (USE OPTIONAL) AND THE SYSTEM INPUT/OUTPUT UNITS ARE REQUIRED. A LARGE MEMORY IS DESIRABLE, BUT IT CAN BE REPLACED TO SOME EXTEND BY DISK OR TAPE STORAGE (THE TRANSITION IS MADE AUTOMATICALLY).

8.RELATED PROGRAMS

DIAMANT2/1/ IS AN IMPROVED VERSION BASED ON THE PREVIOUS DIAMANT PROGRAM/3/. MANY COMMENTS WERE ADDED AND SIMPLIFIED PROGRAMMING WAS PERFORMED TO MAKE DIAMANT2 AS EASY TO UNDERSTAND AS POSSIBLE. AN INTERFACE FILE CAN BE CREATED FOR THE TRANSPORT PERTURBATION CODE TPTRIA /4/. A SPECIAL INPUT CHECK MODULE IS PROVIDED ALONG WITH DIAMANT2.

9.MATERIAL AVAILABLE

SOURCE DECK OF DIAMANT2 AND INPUT CHECK MODULE DIAPRD IN UPDATE FORMAT, TEST PROBLEMS INPUT, RESULTS OF EXECUTED TEST PROBLEMS AND THIS REPORT (KFK4233).

10.REFERENCES

/1/ K.KUEFNER, J.BURKHARD, R.HEGER, "AN UPDATED FORTRAN77 VERSION OF THE 2-D NEUTRON TRANSPORT CODE DIAMANT2 FOR TRIANGULAR MESHES", KFK 4233, KARLSRUHE 1987

/2/ K.KUEFNER, B.DELMARMOL, G.MINSART, "CONTINUOUS FOUR POINT TRIANGULAR MESH DIFFERENCE SCHEMES FOR THE MULTIGROUP NEUTRON TRANSPORT EQUATION", PROC. TOP. MEETING ON COMPUTATIONAL METHODS, WILLIAMSBURG 1979

/3/ K.KUEFNER, R.HEGER, "DIAMANT2, EIN MULTIGRUPPEN NEUTRONENTRANSPORT-PROGRAMM FUER DREIECKS- UND HEXAGONALGEOMETRIE", KFK 3033, KARLSRUHE 1980
/4/ K.KOBAYASHI, G.BUCKEL, K.KUEFNER, "TPTRIA, A COMPUTER PROGRAM FOR THE REACTIVITY AND KINETIC PARAMETERS FOR TWO-DIMENSIONAL TRIANGULAR GEOMETRY BY TRANSPORT PERTURBATION THEORY", KFK 4116, KARLSRUHE 1986
/5/ G.BUCKEL, W.HOEBEL, "DAS KARLSRUHER PROGRAMMSYSTEM KAPROS, TEIL 1", KFK 2253, KARSRUHE 1976

11.PROGRAMMING LANGUAGE USED

STANDARD FORTRAN 77 PLUS SYSTEM DEPENDENT TIMING ROUTINES

12.OPERATING SYSTEM/MONITOR UNDER WHICH PROGRAM IS EXECUTED

IBM MVS/XA, CDC VSOS, CRAY COS, SIEMENS BS3000
COMPILER: SIEMENS-FORTRAN77, IBM-VS-FORTRAN, CYBER-FTN200, CRAY-CFT

13.ANY OTHER PROGRAMMING OR OPERATING INFORMATION

THIS CODE WAS DESIGNED TO RUN UNDER CONTROL OF THE MODULAR CODE SYSTEM KAPROS /5/ BUT USE AS STAND ALONE CODE IS EASILY POSSIBLE. IT IS DESIRABLE THAT THE TIMING FUNCTIONS TASKTI AND TIMLIM COULD BE REPLACED BY SOME SYSTEM DEPENDENT TIMING ROUTINE (AVAILABLE FOR: IBM, CYBER, CRAY, FUJITSU/SIEMENS).

14.NAME AND ESTABLISHMENT OF AUTHORS

KLAUS KUEFNER, JUERGEN BURKHARD, RENATE HEGER
NUCLEAR RESEARCH CENTER KARLSRUHE / INR
P. O. BOX 3640
D-7500 KARLSRUHE 1
FEDERAL REPUBLIC OF GERMANY

CHAPTER 6. INPUT DESCRIPTION

6.1 GENERAL REMARKS

DIAMANT2 reads all input cards from logical unit 14. For the KAPROS version this unit is realized as a datablock DBN=DIAMANT2␣EINGABE as shown in "KAPROS Version" on page 127.

Each K_i stands for one logical record of the input stream. Each S_i marks a logical branching point in the otherwise sequential input. All variables are to be input according to the KAPROS input conventions /4/ or, for the stand-alone version, according to the rules of FORTRAN list-directed input /15/. Note also the remarks at the end of this chapter where explanations for special options are given.

All explanations given in the following are valid for the real problem. They apply usually as well to adjoint problems, exceptions from this rule are specially indicated.

All maximum values can be changed by inserting new bounds in the PARAMETER statement and recompiling the code (see Figure 29 on page 87). The maximum value allowed for the mixture number (=89), the number used to identify vacuum (=90), reflective (=91) or boundary flux (=92) boundary conditions are all fixed by the value of the PARAMETER MATMAX and can easily be changed.

6.2 INPUT CARD SEQUENCE

K1: Identification label of the run (60 characters)
(enclosed in apostrophes within KAPROS)

K2: (integer control data)

ID: Identification number of the run (not used in this version)

ITH: Theory option and selector for perturbation file option
0=Real calculation without creation of perturbation theory file
1=Adjoint calculation without creation of perturbation theory file
-1=Adjoint calculation with creation of perturbation theory file
-2=Real calculation with creation of perturbation theory file

ISN: Order of the angular discretization (ISN has to be even and mod(ISN,100) ≤ 12) and selector of quadrature set:
2,...,12 : use P_N set
102,...,112 : use DP_N set of order ISN - 100
202,...,212 : use EP_N set of order ISN - 200

IGM: Number of energy groups (maximum allowed : 26)

IEVT: Problem type (0: External source problem, 1: Eigenvalue problem)

MLM: Number of intervals on the oblique side of the reference parallelogram

MCM: Number of intervals on the horizontal side of the reference parallelogram

MT: Number of mixtures to be used (maximum allowed : 89)

MAT: Number of cross section-tables to be stored (maximum allowed : 89; MAT = MT for isotropic cases, MAT ≥ MT for anisotropic cases). How MAT is to be calculated is shown in "How to Specify Anisotropic Mixtures" on page 110.

ICM: Maximum number of outer iterations (see "How to Choose Iteration Control Parameters" on page 79)

IIM: Maximum number of inner iterations (see "How to Choose Iteration Control Parameters" on page 79)

IIL: Starting number of inner iterations (see "How to Choose Iteration Control Parameters" on page 79)

KTR: Control of neutron flux print (-1/0/1 : All/None/Selected fluxes to be printed)

KDUM: Cross section input by cards (=0: No, SIGMN block used; =1: Yes)

NFPERT: FORTRAN unit for interface file to perturbation module (see "Data Files Used" on page 63).

NFDMPN: Final flux to be written on this unit
=0 : Not used
>0 : FORTRAN unit number of data set
(see "Data Files Used" on page 63).

NFDMPO: Initial flux to be read from this unit
=0 : Not used
>0 : FORTRAN unit number of data set
(see "Data Files Used" on page 63).

INBO: Control of boundary flux option (see "Boundary Fluxes" on page 112).
=0: No boundary flux,
=1: boundary flux by group and interval,
=2: by energy spectrum and source strength

|S1| If no flux guess is used (NFDMPO = 0 on K2) continue with K4

K3: (flux guess map selection)

NUMFL: Position number of the flux guess map on data file NFDMPO which is to be used

IDFL: Identification number of the run that created the flux guess (variable not used in this version)

K4: (integer control data continued)

IQUELL: Distributed source option indicator (see remark on "External Sources" on page 112).
=0: No distributed source
=1: Given by triangle and group
=-1: Given by mixture and group
=2: Given by spectrum and source strength for each triangle
=-2: Given by spectrum and source strength for each mixture

MBK: Buckling correction option
=0: No buckling term
=1: Constant value given on input card K5
=2: Group dependent values
=3: Group and mixture dependent values

IQUER: Print cross section tables; (=0: No, =1: Yes)

ID1: Reaction rates or densities for fission cross section
(0/1/2 : None/Density/Rate)

ID2: Reaction rates or densities for capture cross section
(0/1/2 : None/Density/Rate)

ID3: Reaction rates or densities for absorption cross section
(0/1/2 : None/Density/Rate)

ID4: Reaction rates or densities for fission neutron production cross section
(0/1/2 : None/Density/Rate)

ID5: Reaction rates or densities for total cross section
(0/1/2 : None/Density/Rate)

Note: ID1 - ID5 are edit options for evaluation purposes. If chosen different from zero they create printed output but no data are written on the interface file.

ISCT: Anisotropic scattering order
=0: Isotropic, maximum allowed: 6

KAUSW: Neutron balances selector
=0: Global balances only
=1: Global balances and regionwise balances
=10: Global balances and mixture dependent balances
=11: Global, regionwise and mixture dependent balances

Note: KAUSW is only an edit option for evaluation; but group-wise total fluxes may be transfered to the perturbation interface file NFPERT (see "Data Files Used" on page 63).

NFANI1: FORTRAN unit for anisotropy working set (see "Data Files Used" on page 63)

NFSCR1: FORTRAN unit for auxiliary working set in storage optimization 2 (see "Data Files Used" on page 63 and "Storage Optimization Levels" on page 65)

INORM: Flux normalization selector (see "Miscellaneous" on page 114)
=1: normalization to one fission event
=2: normalization to power = 1 Watt

K5: (real control data)

EPS: Convergence criterion for termination of outer iterations
< 0: integral test using eq. (38)
= 0: eigenvalue bounds test using value of TEPS and eqs. (40)
> 0: pointwise test using eq. (39)

EV: Starting guess for eigenvalue (not relevant for IEVT=0)

BF: Buckling height (Used if MBK = 1 on card K4)

H: Length of triangle side (in cm)

EPSA: Convergence test selector for inner iterations
< 0: integral test using eq. (33)
= 0: integral test using eq. (33) with EPS for EPSA
> 0: pointwise test using eq. (34)

TEPS: Convergence criterion for termination of outer iterations if eigenvalue bounds are requested (EPS = 0.0)

K6: (not read in this version)

K7: (mixture numbers used and anisotropy selector)

MATTAB: MT (see K2) integer numbers to specify mixtures with anisotropic scattering kernel expansion. Those mixtures are indicated by a negative sign in front of the mixture number. (see "How to Specify Anisotropic Mixtures" on page 110 and the example given there).
Additionally, it is possible within KAPROS to apply the MATTAB input for selecting certain mixtures out of a SIGMN-block (see "Nuclear Cross Section Input by SIGMN Block" on page 107 and the example given there). Thereby one can avoid storing cross-section data which are not relevant for the DIAMANT2 calculation but may be needed in subsequent perturbation calculations.

K8: Input of the mixture distribution by means of overlapping basic zones (zones with uniform mixture, see "Mixture Distribution" on page 106); if several basic zones are overlapped, the last value entered applies. This card has to be repeated until the model is completely specified.

ITYP: Type of basic zone, see Figure 31 on page 101.
IZEI: Grid line number for the left upper corner of the basic zone
ISPA: Grid column number for the left upper corner; the left upper corner of the reference parallelogram is ISPA = IZEI = 0
MISCH: Mixture number assigned to the basic zone (see note below and "How to Specify Anisotropic Mixtures" on page 110 and the example given there).
INTZ: number of line intervals of the basic zone
INTS: Number of column intervals of the basic zone

Note: Boundary conditions are specified by the mixture distribution input using mixture numbers 90 (=vacuum), 91 (=reflection) or 92 (=boundary flux). These numbers are fixed in a PARAMETER statement and can easily be changed.

|S2| If mixture input is complete continue with K8A, otherwise K8 again

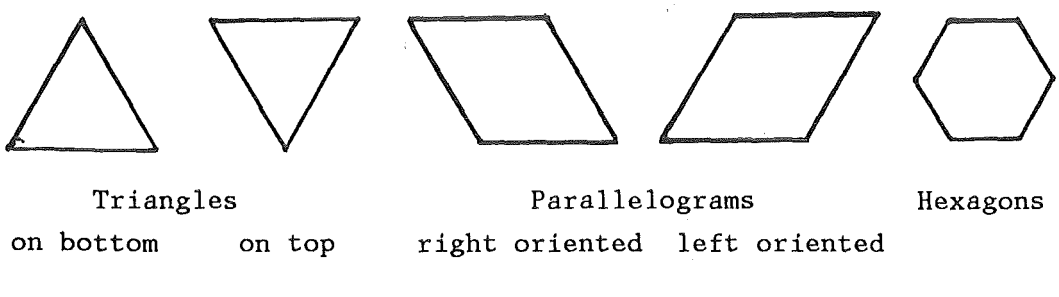


Figure 31. Basic Input Zones ITYP

K8A: (end marker of mixture distribution input)

I1 to I6: enter six times the value 6 (e.g., 6*6)

|S3| If all or no fluxes are to be printed (KTR≠1 on K2) continue with S4

K9: (number of groups used in flux printout)

INGP: Number of energy groups for which neutron fluxes are to be printed. Scalar fluxes for each mesh in the model will be printed.

K10: (groups for which fluxes are printed out)

KNPG(I) (I = 1, INPG) :
Group numbers for printout of fluxes

Note: Remember that in adjoint cases one has to enter these values in reverse group order!

|S4| If cross section input is by SIGMN block (KDUM = 0 on K2) continue with S5

Note: Cross section input sequence is the same for real and adjoint problems; the necessary changes are made automatically by DIAMANT2.

K11: (fission spectrum input)

CHI(I) (I = 1, IGM) :
Input of values for the fission spectrum (assumed to be mixture-independent).

For K = 1, MAT : (MAT see input card K2)

K12: (nuclear cross sections matrix)

C(I,J,K) ((I = 1, IHM), J = 1, IGM) :
Input of the nuclear cross sections (IHM = IGM + IHT; arrangement of the cross sections see "Card Input of Nuclear Cross Sections" on page 110 and Figure 34 on page 109)

|S5| For MBK < 2 (see input card K4) continue with S7

|S6| For MBK = 2 (see input card K4) continue with K14A
For MBK = 3 (see input card K4) continue with K14B

K14A: (buckling values)

B(K) (K = 1, IGM) :
Group-dependent buckling values (mixture-independent)

Note: Remember that in adjoint cases one has to enter these values in reverse group order!

K14B: (buckling values)

B(J,K) ((J = 1, MT), K = 1, IGM) :
Mixture and group-dependent buckling values

Note: Remember that in adjoint cases one has to enter these values in reverse group order!

|S7| If no external source input (IQUELL = 0 on K4) continue with S8

If IQUELL = 1 (input dependent on meshes and energy group)
continue with K15A
If IQUELL = 2 (input dependent on meshes; source equal to product of spectrum times source strength) continue with K15B
If IQUELL =-1 (input dependent on mixtures and energy group)
continue with K15C
If IQUELL =-2 (input dependent on mixtures; source equal to product of spectrum times source strength) continue with K15D

Notes: See "External Sources" on page 112 at the end of this chapter for a discussion of the special type of external source input and definition of input values.
Remember that in adjoint cases one has to enter these values in reverse group order!

K15A: (external source input)

For K = 1, IGM :

Q1(I): ML records with MC(ML1) values for the external source of each triangle
Note: See "External Sources" on page 112 at the end of this chapter for the definition of the variables ML, ML1 and MC(ML1).

Continue with S8

K15B: (external source input)

Q2(K): (K = 1,IGM)
IGM values with the spectrum for the external source

Q1(I): ML records with MC(ML1) values for the external source strength of each triangle

Note: See "External Sources" on page 112 at the end of this chapter for the definition of the variables ML, ML1 and MC(ML1).

Continue with S8

K15C: (external source input)

For K = 1, IGM :

Q3(J,K) (J = 1, MT) :
values of the source strength for each mixture; one record for each energy group.

Continue with S8

K15D: (external source input)

Q2(K): IGM values with the spectrum of the external source

Q3(J): MT values of the source strength for each mixture

|S8| If no boundary flux (INBO=0 on input card K2) continue with K17

If INBO = 1 continue with K16A
If INBO = 2 continue with K16B

Notes: See "Boundary Fluxes" on page 112 at the end of this chapter for a discussion of the special type of boundary flux input and definition of input values.
Remember that in adjoint cases one has to enter these values in reverse group order!

K16A: (boundary flux input)

For K = 1, IGM :
For J = 1, NB :

BND(J,I,K) (I = 1, NI(J)) :
values for the boundary flux
Note: See "Boundary Fluxes" on page 112 at the end of this chapter for the definition of the variables NB and NI.

Continue with K17

K16B: (boundary flux input)

BND(1,1,K): IGM values with the spectrum of the boundary fluxes

K16C:
For J = 1, NB :

BND(J,I,1): (I = 1, NI(J))
values for the source strength
Note: See "Boundary Fluxes" on page 112 at the end of this chapter for the definition of the variables NB and NI.

K17: (end of problem input)

CKK: Four bytes character constant
= ENDE : End of DIAMANT2 input
= CONT : Input for additional case starting from K1
(enclosed in apostrophes within KAPROS)

For the KAPROS version the END-OF-DATABLOCK marker *$*$ completes the whole input sequence.

6.3 REMARKS AND EXPLANATIONS FOR SOME INPUT OPTIONS

6.3.1 Mixture Distribution

The reactor model is embedded in a reference parallelogram having the dimensions MLM and MCM (see K2 and Figure 33 on page 108). Five basic zones are available to produce in a fairly easy way the mixture distribution of the model (see K8). The layout is generated by placing the basic zones adjacent to and overlapping each other. The last input always applies.

The boundary of the (embedded) model **must be convex**. If necessary, the convex outline of the model must be simulated using additional ("dummy") mixtures (see Figure 32 on page 107). In the case of boundaries with vacuum boundary conditions one may take a pure strong absorber (e.g., $\sigma_a = \sigma_{tot} = 1.0$ in all groups and zero entries for all other cross sections). In such cases it is recommended to check the appropriateness of the chosen absorption cross section and the thickness of the artificial absorber region by a careful aposteriori analysis since flux values in the neighborhood of the artificial absorber region as well as integral quantities, e.g., the criticality factor, may depend on the chosen parameters.

The boundary of the model must be surrounded by triangles with mixture numbers MTVP (= vacuum), MTRP (= reflection) or MTJP (= boundary flux) to simulate boundary conditions. MTVP=90, MTRP=91, MTJP=92 are defined in a PARAMETER statement which limits the maximum mixture number to MATMAX=89. If this should not be sufficient in an application, one has to change MATMAX to a higher value in the PARAMETER statement and to recompile the code. The reference parallelogram is initialized with mixture number 0.

6.3.2 Nuclear Cross Sections

The cross section input by cards is described in "Card Input of Nuclear Cross Sections". Use of SIGMN blocks (KDUM = 0 on K2) is strongly recommended for the KAPROS version.

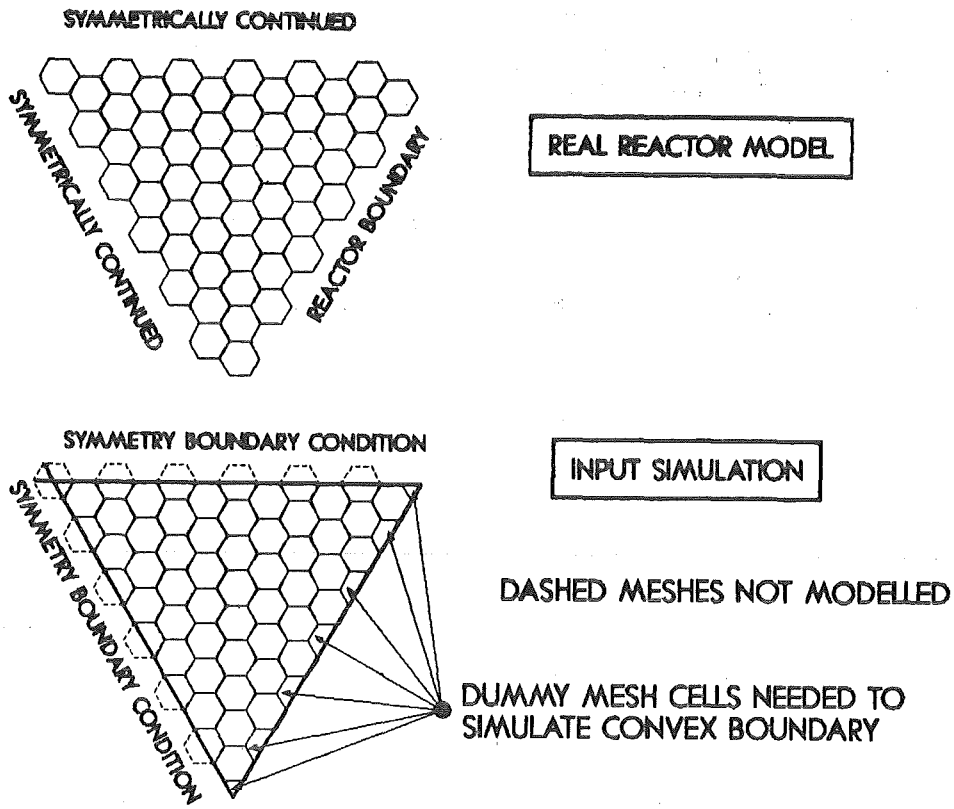


Figure 32. Convexity of the Model Boundary

6.3.2.1 Nuclear Cross Section Input by SIGMN Block

The KAPROS version of DIAMANT2 can take over nuclear cross sections directly from SIGMN /11/ blocks out of the KAPROS lifeline. The expected standard data block name is DBN=SIGMN, IND=1. The block must contain at least the scalar types CHI, NUSF, SCAPT, SFISS, SN2N, STOT or STRTR and STR as well as the vector type SMTOT (see Figure 34 on page 109). For anisotropic calculations, the higher moments of the scattering matrix are expected with the names SMT01 to SMT06, the last two figures indicating the order of the Legendre moment.

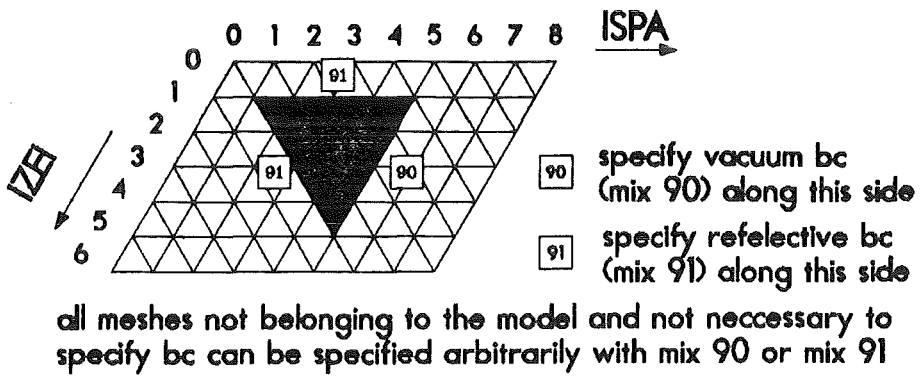


Figure 33. Specifying the Model in the Reference Parallelogram

DIAMANT2 performs a transport correction of the cross sections only for isotropic calculations (see description of subroutine FILLC /10/). STOT is replaced by STRTR and SMTOT(g,g) is replaced by STRTR-SREM; if STRTR is not present in the SIGMN block, type STR is used instead. This modification, which results from the application of the so-called transport correction, can lead to difficulties for the within-group scattering term in the case of material mixtures which contain mainly light scattering substances (e.g. H_2O), where a negative within-group scattering matrix element can occur. In those cases it is recommended to avoid the application of the transport correction and to use instead the anisotropic scattering option, if appropriate nuclear data are available.

The cross sections are expected in the KAPROS lifeline as a data block DBN=SIGMN for KDUM = 0 (on K2). The number of mixtures in the SIGMN block need not agree with the number of mixtures used in DIAMANT2 (variable MT on K2). In particular, mixtures can be selected from the SIGMN block using input record K7.

Example (assuming MT=3)

K7 : 27 23 35

means that the data for mixture 1 in the DIAMANT2 input are taken from the cross section for mixture 27 in the SIGMN block, mixture 2 (DIAMANT2) from that of mixture 23 in the SIGMN block, etc. The maximum mixture number to

k	C(k,i,j)	Explanation
IPSFIS	σ_{fis}	fission cross section
IPSCAP	σ_{capt}	capture cross section
IPSN2N	$\sigma_{n,2n}$	(n,2n) cross section
IPSTR	σ_{tr}	transport cross section
IHT-2	σ_{abs}	absorption cross section
IHT-1	$\nu\sigma_f$	fission production cross section
IHT	σ_{tot}	total cross section
IHS	$\sigma_{g\to g}$	within-group scatter cross section
IHS+1 : IHM	$\sigma_{g\to h}$	down-scatter cross section

Remarks:

1. Indices i = energy group and j = mixture number have the following ranges: i=1 to IGM and j=1 to MAT.

2. No up-scattering and full down-scattering is assumed (IHS = IHT+1, IHM = IHT+IGM)

Figure 34. Storage Sequence in Cross Section Array C

be used in DIAMANT2 (defined in a PARAMETER statement) is 89 (90, 91 and 92 are reserved to define boundary conditions, see "Mixture Distribution" on page 106).

6.3.2.2 Card Input of Nuclear Cross Sections

DIAMANT2 can also take over cross sections from card input. But it is not possible to mix cross section input from cards with that from SIGMN blocks. A fixed order of input values must be used (see Figure 34).

The cross section block is a three-dimensional array C with the dimensions C(IHM,IGM,MAT) such that IHM = IGM + 7, IGM is the number of energy groups considered and MAT is the number of tables to be stored (see input card K2 and "How to Specify Anisotropic Mixtures"). The cross sections are input in groups for each material from I = 1 to I = MAT. The sequence of cross sections for a fixed group and a fixed mixture is prescribed as: σ_{fiss}, σ_{capt}, $\sigma_{n,2n}$, σ_{tr}, σ_{abs}, $\nu\sigma_f$, σ_{tot}, followed by the scattering terms $\sigma_{g\to1}$, ..., $\sigma_{g\to IGM}$. $\sigma_{g\to g'}$ will be treated as zero if $g > g'$. σ_{tr} is used only for the buckling correction, $\sigma_{n,2n}$ is required only as a dummy (see Figure 34 on page 109). It should be noted that no transport correction is applied with card input of the nuclear cross sections for DIAMANT2.

ISCT (=order of anisotropy) additional tables are stored for each mixture to be calculated anisotropically (indicated by a negative sign for the corresponding MATTAB entry number in the input record K7, which accordingly increases the value for the variables MAT on input card K2 compared to MT; see "How to Specify Anisotropic Mixtures"). All non-scattering terms (i.e., scalar types) are to be input with 0.0 in the anisotropic tables located immediately after the zeroth moment table.

6.3.3 How to Specify Anisotropic Mixtures

The following rule applies to MT and MAT:

MT = MAT for isotropic calculations
MT ≤ MAT for anisotropic calculations.

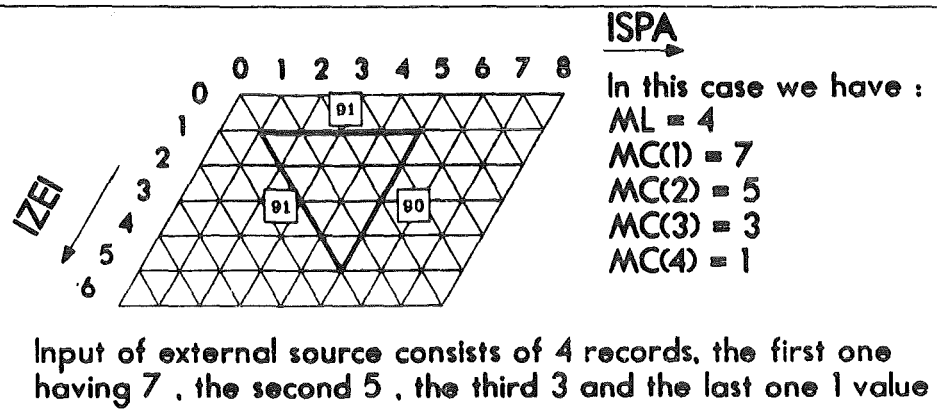


Figure 35. Example of an External Source Input

Each mixture which is to be calculated anisotropically (indicated by a negative sign for the mixture number in the MATTAB array on K7) increases the number of tables to be stored by ISCT. MAT is the total number of cross section tables to be stored.

Consider the following input parameters specifying a mixture input for a calculation without Legendre expansion of the scattering kernel (i.e. ISCT=0) :

Input card K7: MATTAB : 1 2 3 4 5

Input cards K8:

ITYP	IZEI	ISPA	MISCH	INTZ	INTS
4	1	1	3	17	17
4	1	1	2	13	13
4	1	1	1	9	9
		.			
		.			
		.			
6	6	6	6	6	6

This means MT = MAT = 5, ISCT = 0.

In order to choose a Legendre expansion of order ISCT > 0 for mixtures 1 and 3, one has to specify :

Input card K7: MATTAB : -1 2 -3 4 5

Input cards K8: remain unchanged !

This means MT = 5, and because we have two anisotropic mixtures, MAT = MT + 2*ISCT and ISCT > 0. Note that the same order of Legendre expansion (ISCT) is used for all mixtures. In addition to that small input change, now more nuclear cross sections have to be provided and the file numbers for scratch units NFANI1 and NFANI2 on input card K4 must have meaningful values.

6.3.4 External Sources

The following section refers to input cards K15A and K15B used to specify an external source input (see Figure 35).

ML is the total number of grid lines which pass through the real model (the reference parallelogram contains MLM grid lines; the real model is marked as thick lined triangle in Figure 35 on page 111), and MC(ML1) is the number of triangles which are contained in line ML1 of the model (the reference parallelogram contains 2*MCM triangles per line). The model must be swept completely, beginning with the upper left triangle and working to the right. Once a line has been completed, input continues with the next lower line.

In preparing input for external source problems, a user should take care of the influence of reflective boundary conditions if only a sector of the reactor is specified; i.e., he should be aware that at symmetrical positions there will exist equivalent external sources which contribute to the total source strength inserted into the whole reactor.

The user should also be aware of having specified the appropriate physical dimension for the external source. In the real case, evidently the external source should have the dimension of fission sources. This means, e.g., that the dimension of the source strength has to be compatible with the production rate (i.e., neutrons per $cm^3 s$) and the corresponding energy spectrum with the fission spectrum.

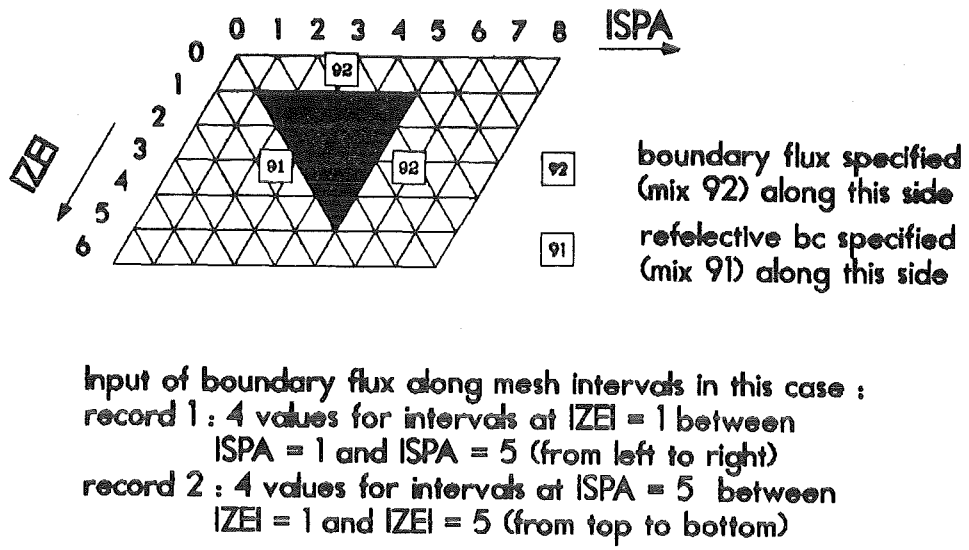


Figure 36. Example of a Boundary Flux Input

6.3.5 Boundary Fluxes

At the outside boundary incoming fluxes can be specified for individual boundary mesh intervals (see Figure 36 on page 113). They are assumed to be isotropic for the halfspace of all incoming directions and are interpreted as particle sources for incoming directions. The boundary of the model is divided into 6 possible directions (north, northeast, southeast, south, southwest, northwest; see Figure 8 on page 26).

Input for each direction begins in the left upper corner of the model and continues in a clockwise direction; only values for those triangles which were attributed the mixture number 92 (92 is related to the maximum number of mixtures allowed fixed by the value of PARAMETER MATMAX, see Figure 29 on page 87 and the last paragraph in "General Remarks" on page 95). in the mixture distribution input have to be input. Note that input has to be given for mesh intervals, i.e., a possible subdivision of the basic triangle side has to be taken into account. Instead of the fluxes, a spectrum (for description of the group dependence) and a source strength can be input in

the sequence described above (INBO = 2). In the latter case, boundary fluxes are calculated as the product of spectrum times source strength.

In preparing input for boundary flux problems, a user should take care of the influence of reflective boundary conditions if only a sector of the reactor is specified; i.e., he should be aware that at symmetrical positions there will exist equivalent boundary fluxes which contribute to the total boundary flux acting on the whole reactor. In the real case, evidently the boundary flux should correspond to angular fluxes. For the adjoint problem, the specification concerns outgoing adjoint fluxes at the boundary according to "Adjoint Calculations" on page 23.

6.3.6 Miscellaneous

- The following constraints apply to all optional data file numbers

  $I < 40$ and $I \neq 5,6,14$;

  Corresponding operating system cards have to define the files used beforehand. Within KAPROS, some of the files may be realized as KAPROS datablocks and may require then a *KSIOX card (see "KAPROS Version" on page 127).
- All data files used by DIAMANT2 must be defined within KAPROS in a separate data block with name INITbREADbDIAMAN (see "KAPROS Implementation" on page 69 and Figure 24 on page 71).
- In normalizing power in a calculation (INORM = 2 on K4) it is assumed that the power portion proportional to the neutron flux is approximately 200 MeV per fission event. The user must take into account the reactor section (because of symmetries) to which the normalization applies.

CHAPTER 7. SAMPLE OUTPUT INTERPRETATION

This is a description of the printed output of a DIAMANT2 calculation. The purpose of this section is to facilitate interpretation of the output for the user (see program output in "Appendix B. Example of a DIAMANT2 Calculation" on page 125).

Part 1 of the Output: Program Constants and Input Protocol

All values of the currently valid PARAMETER statement are printed and briefly explained. Subsequently, the division of working arrays into subarrays is printed.

Then all input values are tabulated, the discrete directions along with the associated integration weights and the angle sweeps sequence are printed. The directions are characterized by the direction cosines of their projections in the x-y plane. The nuclear cross sections may be printed out (if requested by input variable IQUER):

FISSION FRAC	Fission spectrum
GROUP SUM	Group sum for all fission spectra (usually close to 1.0).
CROSS SECTIONS	Nuclear cross section matrix C for each material (including higher Legendre moments if ISCT > 0).

A message may appear beforehand if the input absorption cross section has been replaced by a calculated one. For the KAPROS version some diagnostic output from FILLC will be printed whenever cross sections are read from a SIGMN block.

After that, the reactor model to be calculated is sketched in the printout with heading

MIXTURE DISTRIBUTION	Sketch of the mixture distribution in the reference parallelogram (including boundary conditions)

An error message (ISTOP = 16) will appear if the mixture distribution input will reference a mesh cell outside the reference parallelogram.

Part 2 of the Output: The Iteration Monitor

The following information is output after each outer iteration:

- current outer iteration number;
- total number of outer iterations;
- total number of inner iterations;
- criticality factor (lower estimate);
- criticality factor (k_{eff});
- criticality factor (upper estimate);
- outer iteration error;
- mesh point indices of maximum outer error;
- CPU-time used in seconds;

A special message is output if negative flux fix-up has been activated. The following information is printed for each group (within the outer iteration monitor print):

- energy group considered;
- number of inner iterations in this group;
- inner iteration error;
- mesh point indices of maximum inner error;
- rebalancing factor (if rebalancing is switched on) according to Eq. (35).

Part 3 of the Output: Results During the Final Iteration

- information why iteration stops (convergence or maximum iteration count reached or CPU time exhausted);
- information on the normalization of fluxes;
- global balances followed by mixture and/or region balances (if selected by input);
- various reaction rates and/or reaction densities (as selected by input);
- output of the scalar fluxes (as selected by input);
- volume integrated fission source;
- information if a restart dump has been created (if this option is used, the number NUMFL under which the dump is stored will also be printed).

Part 4 of the Output: Final Iteration Result

The result of the final outer iteration is output in almost the same format as the normal iteration monitor.

- information on the vectorlength for that case;
- information on convergence criteria used;
- outer iteration monitor print;
- negative flux fix-up summary;
- information why iteration stops (convergence or maximum iteration count reached or CPU time exhausted);
- timing statistics (including the execution speed v mentioned in "How to Estimate Computing Time" on page 80).

If the stop code ISTOP is different from zero, a list of stop codes along with an explanation of their meanings is added.

APPENDIX A. DIAMANT2 BUILT-IN QUADRATURE CONSTANTS

A.1 DEFINING EQUATIONS OF THE QUADRATURE SETS

The one dimensional P_N (Gauss quadrature), DP_N (double Gauss quadrature), EP_N (equal weight quadrature) /7/ quadrature sets differ in integration weights w_m and direction cosines ξ_m. These are used to construct twodimensional quadrature sets as described in "Construction of S_N-Constants" on page 8. All three quadrature sets can be selected by input specification (see variable ISN on input card K2). In the following we give a short description of the defining equations of all three quadrature sets.

In the $\mathbf{P_N}$ set, the ordinates ξ_m are taken to be the N (=ISN on input card K2) roots of the N-th Legendre polynomial:

$$(*1*) \quad P_N(\xi_m) = 0, \quad m = 1,\ldots,N$$

The weights are determined such that the quadrature formula correctly integrates P_0 through P_{N-1} :

$$(*2*) \quad \sum_{m=1}^{N} w_m P_\ell(\xi_m) = \delta_{\ell,0} \ , \ \ell = 0,1,\ldots,N-1$$

where $\delta_{i,j}$ is the Kronecker delta, equal to 1 if i equals j and zero otherwise.

In the $\mathbf{DP_N}$ set, Eq.(*1*) is replaced by

$$(*3*) \quad \begin{aligned} P_{(N-2)/2}\ (2\xi_m+1) &= 0, \qquad m=1,2,\ldots,N/2 \\ P_{(N-2)/2}\ (2\xi_m-1) &= 0, \qquad m=N/2+1,\ldots,N \end{aligned}$$

and (*2*) is replaced similarly by

$$(*4*) \quad \sum_{m=1}^{N/2} w_m P_\ell(2\xi_m+1) = \delta_{\ell,0} \ , \ \ell = 0,1,\ldots,(N-2)/2$$

$$\sum_{m=n/2+1}^{N} w_m P_\ell(2\xi_m - 1) = \delta_{\ell,0} \ , \ \ell = 0,1,\ldots,(N-2)/2$$

The EP_N set is a one dimensional analog of the symmetric equal weight quadrature EQN /7/. For a given order ISN, the weights are determined by

(*5*) $$w_m = (4/ISN - m*r_{ISN})/2, \qquad r_{ISN} = 8/(ISN(ISN+2)), \ m=1,\ldots,ISN/2$$

and the ordinates ξ_m are obtained by solving

(*6*) $$\sum_{m=1}^{N} w_m \xi_m^{\ell} = 1/(\ell+1), \qquad \ell = 2,4,\ldots,ISN.$$

For problems dominated by the diffusion of neutrons through optically thick regions, a P_N approximation may be expected to yield superior results to the DP_N approximation. In contrast, the DP_N quadrature may be preferable for problems dominated by the presence of boundaries, particularly vacuum boundary conditions /8/. The differences between P_N and EP_N sets are usually small. Note that the S_2 constants are the same in all three sets.

A.2 P_N QUADRATURE CONSTANTS

Direction Cosines ξ_m of Levels

ISN	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6
2	0.5773502					
4	0.3399810	0.8611363				
6	0.2386192	0.6612094	0.9324695			
8	0.1834346	0.5255324	0.7966665	0.9602899		
10	0.1488743	0.4333954	0.6794096	0.8650634	0.9739065	
12	0.1252334	0.3678315	0.5873180	0.7699027	0.9041173	0.9815606

Weights w_m of Level ξ_m

ISN	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6
2	0.5000000					
4	0.3260726	0.1739274				
6	0.2339570	0.1803808	0.0856622			
8	0.1813419	0.1568533	0.1111905	0.0506143		
10	0.1477621	0.1346334	0.1095432	0.0747257	0.0333357	
12	0.1245735	0.1167463	0.1015837	0.0800392	0.0534697	0.0235877

A.3 DP_N QUADRATURE CONSTANTS

Direction Cosines ξ_m of Levels

ISN	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6
2	0.5773503					
4	0.2113249	0.7886751				
6	0.1127017	0.5000000	0.8872983			
8	0.0694318	0.3300095	0.6699905	0.9305682		
10	0.0469101	0.2307653	0.5000000	0.7692347	0.9530899	
12	0.0337652	0.1693953	0.3806904	0.6193096	0.8306047	0.9662348

Weights w_m of Level ξ_m

ISN	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6
2	0.5000000					
4	0.2500000	0.2500000				
6	0.1388889	0.2222222	0.1388889			
8	0.0869637	0.1630363	0.1630363	0.0869637		
10	0.0592318	0.1196572	0.1422222	0.1196572	0.0592318	
12	0.0428311	0.0901904	0.1169785	0.1169785	0.0901904	0.0428311

A.4 EP_N QUADRATURE CONSTANTS

Direction Cosines ξ_m of Levels

ISN	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6
2	0.5773502					
4	0.3500212	0.8688903				
6	0.2595650	0.6815819	0.9320784			
8	0.2051012	0.5600928	0.8067958	0.9602690		
10	0.1702548	0.4730740	0.7051312	0.8722077	0.9730098	
12	0.1453250	0.4094977	0.6224061	0.7902606	0.9081059	0.9810382

Weights w_m of Level ξ_m

ISN	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6
2	0.5000000					
4	0.3333333	0.1666667				
6	0.2500000	0.1666667	0.0833333			
8	0.2000000	0.1500000	0.1000000	0.0500000		
10	0.1666667	0.1333333	0.1000000	0.0666667	0.0333333	
12	0.1428572	0.1190476	0.0952381	0.0714286	0.0476190	0.0238095

APPENDIX B. EXAMPLE OF A DIAMANT2 CALCULATION

B.1 JOB CONTROL AND INPUT DATA CARDS

B.1.1 Stand-alone Version

```
//.......  JOB (....,...,.....),.......,REGION=2048K,
//  TIME=(,30),MSGCLASS=H,NOTIFY=......,MSGLEVEL=(2,0)
//*MAIN LINES=7
//* ***   DIAMANT2 SAMPLE CASE FOR STANDALONE VERSION
// EXEC F7ASCLG,PARM.C='LANGLVL(77),NOMAP,INCLUDE'
//* ***   COMPILATION OF FORTRAN SOURCE
//C.SYSINC DD DISP=SHR,DSN=INR986.DIANEU.FORT
//C.SYSIN DD DISP=SHR,DSN=INR986.DIANEU.FORT(SAPROG)
//         DD DISP=SHR,DSN=INR986.DIANEU.FORT(FORTY)
//         DD DISP=SHR,DSN=INR986.DIANEU.FORT(SAEXTERN)
//         DD DISP=SHR,DSN=INR986.DIANEU.FORT(F77TIMER)
//* ***   ASSEMBLER STEP FOR TIMING FUNCTION   JTIME
//A.SYSIN DD DISP=SHR,DSN=INR986.DIANEU.FORT(JTIME)
//* ***   EXECUTE DIAMANT2
//G.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,(15,5)),DCB=DCB.VBS
//G.FT10F001 DD UNIT=SYSDA,SPACE=(TRK,(15,5)),DCB=DCB.VBS
//G.FT14F001 DD *
 TESTCASE 1 - K INFINITY CALCULATION IN S4
         1     0     4     4     1     8     8     1     1    20    20
         3    -1     1     0     0     0     0
         0     0     1     0     0     0     0     0     0     0
         0     0     1
     1.0E-04      1.0E+00          0.0       2.3945      1.0E-04          0.0
   1
      4    0    0   90    8   8
      4    1    1   90    7   7
      4    2    2    1    4   4
      1    1    2   91    1   1
      1    1    3   91    1   1
      1    1    4   91    1   1
      1    1    5   91    1   1
      2    2    1   91    1   1
      2    3    1   91    1   1
      2    4    1   91    1   1
      2    5    1   91    1   1
      1    2    6   91    1   1
      1    3    6   91    1   1
      1    4    6   91    1   1
      1    5    6   91    1   1
      2    6    2   91    1   1
      2    6    3   91    1   1
      2    6    4   91    1   1
      2    6    5   91    1   1
      6*6
```

```
  7.67999E-01  2.32000E-01  0.0          0.0
  5.64534E-02  2.99241E-03  3.49752E-01  2.30354E-01  5.94459E-02
  1.52410E-01  2.31315E-01  1.15508E-01  0.0          0.0
  0.0
  5.84630E-02  1.03672E-02  4.27233E-01  3.28800E-01  6.88301E-02
  1.44659E-01  3.45855E-01  2.77019E-01  5.66829E-02  0.0
  0.0
  1.70727E-01  6.29142E-02  8.14570E-01  8.08625E-01  2.33641E-01
  4.13708E-01  8.12944E-01  5.79204E-01  5.81963E-06  2.19795E-05
  0.0
  4.50265E-01  1.92642E-01  1.27163E+00  1.32942E+00  6.42907E-01
  1.09098E+00  1.26985E+00  6.26941E-01  9.90387E-05  2.45335E-08
  2.25968E-07
ENDE
/*
//
```

B.1.2 KAPROS Version

```
//.......  JOB (...,...,.....),.......,REGION=2048K,
//  TIME=(0,30),MSGCLASS=H,NOTIFY=......,MSGLEVEL=(2,0)
//*MAIN LINES=10
//* ***   DIAMANT2 SAMPLE CASE FOR KAPROS VERSION
//TESTDIAM  EXEC  KSG7
//K.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,(15,1)),
//           DISP=(NEW,PASS),DCB=(RECFM=VBS,BLKSIZE=16376)
//K.FT10F001 DD UNIT=SYSDA,SPACE=(TRK,(15,1)),
//           DISP=(NEW,PASS),DCB=(RECFM=VBS,BLKSIZE=16376)
//*  CROSS SECTIONS DERIVED FROM DATA OF GODIVA ARE STORED ON UNIT 17
//K.FT17F001 DD DISP=SHR,DSN=INR986.KSA1.A
//K.SYSIN DD *
*$ ------------------------------------------------------------------
*$ CROSS SECTIONS DERIVED FROM DATA OF GODIVA ARE STORED ON UNIT 17
*KSIOX DBN=SIGMN,IND=1,TYP=ARCI,SPEC=FT17GOD4,DBNA=GODIVA 4
*$ ------------------------------------------------------------------
*KSIOX DBN=DIAMANT2 EINGABE,TYP=CARD,PMN=DIAPRD
*$ K1:************** TEST CASE 1     ****************************************
'TESTCASE 1 - K INFINITY CALCULATION IN S4                          '
*$ K2:
*$       ID    ITH    ISN    IGM   IEVT    MLM    MCM     MT    MAT    ICM    IIM
          1      0      4      4      1      8      8      1      1     20     20
*$      IIL    KTR   KDUM NFPERT NFDMPN NFDMPO INBO
          3     -1      0      0      0      0      0
*$ K4:
*$  IQUELL    MBK IQUER    ID1    ID2    ID3    ID4    ID5   ISCT KAUSW
          0      0      1      0      0      0      0      0      0      0
*$  NFANI1 NFSCR1 INORM
          0      9      1
*$ K5:
*$        EPS            EV            BF             H          EPSA          TEPS
      1.0E-04       1.0E+00           0.0        2.3945       1.0E-04           0.0
*$ K7: MATTAB (USED MIXTURE NUMBERS)
   1
*$  INPUT LAYOUT  CARDS START
*$ K8:
*$ ITYP IZEI ISPA MISH INTZ INTS
      4    0    0   90    8   8
      4    1    1   90    7   7
      4    2    2    1    4   4
      1    1    2   91    1   1
      1    1    3   91    1   1
      1    1    4   91    1   1
      1    1    5   91    1   1
      2    2    1   91    1   1
      2    3    1   91    1   1
      2    4    1   91    1   1
      2    5    1   91    1   1
      1    2    6   91    1   1
      1    3    6   91    1   1
      1    4    6   91    1   1
      1    5    6   91    1   1
```

```
      2    6    2   91    1   1
      2    6    3   91    1   1
      2    6    4   91    1   1
      2    6    5   91    1   1
*$  LAYOUT END
      6*6
*$ CROSS SECTION INPUT IS DONE BY SIGMN BLOCK
'ENDE'
*$*$
*$ ------------------------------------------------------------------------
*KSIOX DBN=INITbREADbDIAMAN,TYP=CARD,PMN=PRDUM
09 'SCRATCHbUNITb1bb' 1 6 0 1
10 'SCRATCHbUNITb2bb' 1 6 0 1
14 'DIAMANT2bEINGABE' 1 1 0 0
*$*$
*$ ------------------------------------------------------------------------
*GO SM=DIANEU
/*
//
```

THIS IS A MAJOR REVISION OF THE DIAMANT2 CODE (VERSION 1.4) - NEW RELEASE: 2.0
THIS VERSION IS PROGRAMMED ACCORDING TO THE FORTRAN77 STANDARDS
A GREAT DEAL OF EFFORT HAS GONE INTO CODE RESTRUCTURING, CLARIFICATION AND DOCUMENTATION
THIS VERSION SHOULD RUN EFFICIENTLY ON VECTOR COMPUTERS, TOO

```
**********************************************************************
*                                                                    *
* EINHEIT  9 REALISIERT ALS EXTERNE EINHEIT                          *
* EINHEIT 10 REALISIERT ALS EXTERNE EINHEIT                          *
* EINHEIT 14 REALISIERT ALS NORMALER DB: DIAMANT2 EINGABE     1 *
*                                                                    *
**********************************************************************


FREE MAIN STORAGE PLACE IN THE BEGINNING :    214430 WORDS


     VALUES OF THE PARAMETER STATEMENT USED TO CREATE LOAD MODULE
     ============================================================
 IGMMAX       26     MAXIMUM NUMBER OF ENERGY GROUPS
 IGPMAX       27     IGMMAX + 1
 IREGMX       30     MAXIMUM NUMBER OF DISJOINT MIXTURE REGIONS
 ISNMAX       12     MAXIMUM ORDER OF THE ANGULAR DISCRETIZATION
 ISCTMX        6     MAXIMUM ORDER OF ANISOTRPIC SCATTERING
 JMMMAX      138     JMMMAX=ISNMAX+ISNMAX*(ISNMAX+2)*3/4
 LPMAX     65535     MAXIMUM LENGTH OF A SINGLE DO-LOOP (SIGNIFICANT ONLY FOR CYBER 205)
 MINVEC        6     MINIMUM LENGTH OF A DO-LOOP TO BE EXECUTED IN VECTOR MODE
 MATMAX       89     MAXIMUM NUMBER OF XS-TABLES
 MAXFIL       40     MAXIMUM VALUE FOR FORTRAN UNIT NUMBERS
 MTVP         90     MIXTURE NUMBER TO REPRESENT VACUUM BOUNDARY CONDITION IN INPUT ( = MATMAX + 1)
 MTRP         91     MIXTURE NUMBER TO REPRESENT REFLECTIVE BOUNDARY CONDITION IN INPUT ( = MATMAX + 2)
 MTJP         92     MIXTURE NUMBER TO REPRESENT BOUNDARY FLUX CONDITION IN INPUT ( = MATMAX + 3)
 NACMAX        1     NUMBER OF WORDS RESERVED FOR REAL STORAGE
 NWMAX         1     NUMBER OF WORDS RESERVED FOR INTEGER STORAGE
 NCMAX         2     TOTAL NUMBER OF WORDS OF RESERVED STORAGE ( NCMAX = NACMAX + NWMAX )
 LENCOM        1     LENGTH OF WORKING ARRAY IN COMMON BLOCK /COMMOM/
 NIVMAX       21     NIVMAX=ISNMAX*(ISNMAX+2)/8

 EPSCPU 1.0000E-20     ALL REAL VALUES BELOW THIS THRESHOLD ARE CONSIDERED TO BE ZERO
 EPSMIN 1.0000E-06     MINIMUM VALUE FOR CONVERGENCE CRITERIA (DEPENDS ON MACHINE PRECISION)
 PI     3.1416E+00     VALUE OF THE CONSTANT PI
 PIHALF 1.5708E+00     PI/2.
 PISIXT 5.2360E-01     PI/6.

 LREBAL     TRUE    INNER ITERATION ACCELERATION
 LTRANS     TRUE    PERFORM P1 CORRECTION IN XS-TABLE
 LXSCHK     TRUE    REPLACE INPUT ABSORPTION XS BY CALCULATED XS

 CPU TIME OVERHEAD IN EACH CALL TO FUNCTION TASKTI :    0.0        SECONDS
```

```
TITLE OF THIS RUN:

**************************************************************************
*                                                                        *
*    TESTCASE 1 - K INFINITY CALCULATION IN S4                           *
*                                                                        *
**************************************************************************

   NO  FLUX  GUESS


POINTER DATABLOCK WITH       1054 WORDS OF 4 BYTES EACH CREATED; VALUE OF POINTER IS:      3144

DISTRIBUTION OF WORKING ARRAY IW
================================

ARRAY         MODBND    MIXDIS     IHLP1    MIXREG     MODEL    MODANI     IHLP2
START IN IW     3144      3684      3812      3940      4068      4196      4197
LENGTH           540       128       128       128       128         1         1

START OF WORKING ARRAY AT IW(    3144), END OF WORKING ARRAY AT IW(    4198)

POINTER DATABLOCK WITH       6612 WORDS OF 4 BYTES EACH CREATED; VALUE OF POINTER IS:      4199

DISTRIBUTION OF WORKING ARRAY A
================================

ARRAY           XNA       XNB       XNC      HLP1      HLP2      XNBI      XNBO      XNAI      XNAO      XNCI
START IN A     4199      4207      4215      4223      4231      4239      4415      5119      5295      5999
LENGTH            8         8         8         8         8       176       704       176       704       176

ARRAY          XNCO    FLXANG    FLXNEW         S    FLXOLD         C      XTOT    XSSCAT    XSFISS    GRPSOU
START IN A     6175      6879      7007      7135      7263      7391      7523      7535      7663      7675
LENGTH          704       128       128       128       128       132        12       128        12       128

ARRAY            FG     FGOLD      BUCK    EXTSOU      XAIJ      XBIJ      XCIJ      BIL1      BIL2      BIL3
START IN A     7803      7931      8059      8060      8061      8765      9469     10173     10293     10293
LENGTH          128       128         1         1       704       704       704       120         1         1

ARRAY         FLXSKA        SS         T        TW        TF     TFOLD     XSANI
START IN A     10293     10805     10806     10807     10808     10809     10810
LENGTH           512         1         1         1         1         1         1

START OF WORKING ARRAY AT A(    4199), END OF WORKING ARRAY AT A(   10811)

REAL MAIN STORAGE                  6612
INTEGER MAIN STORAGE               1054
NUMBER OF AUXILIARY DATASETS USED     0
```

```
     DIAMANT2   INPUT   CONTROL
     ==========================

K1:     TESTCASE 1 - K INFINITY CALCULATION IN S4


K2:
ID        1     IDENTIFICATION NUMBER OF THE RUN
ITH       0     THEORY OPTION (0=REGULAR , 1=ADJOINT)
ISN       4     ORDER OF THE ANGLE DISCRETIZATION (MAXIMUM ALLOWED : 12 )
IGM       4     NUMBER OF ENERGY GROUPS (MAXIMUM ALLOWED : 26 )
IEVT      1     PROBLEM TYPE ( 0: EXTERNAL SOURCE PROBLEM ; 1: EIGENVALUE PROBLEM)
MLM       8     NUMBER OF INTERVALS ON THE OBLIQUE SIDE OF THE REFERENCE PARALLELOGRAM
MCM       8     NUMBER OF INTERVALS ON THE HORIZONTAL SIDE OF THE REFERENCE PARALLELOGRAM
MT        1     NUMBER OF MIXTURES TO BE USED
MAT       1     NUMBER OF XS-TABLES ( MAT = MT FOR ISOTROPIC CASE , MAT >= MT FOR ANISOTROPIC CASE)
ICM      50     MAXIMUM NUMBER OF OUTER ITERATIONS
IIM      20     MAXIMUM NUMBER OF INNER ITERATIONS
IIL       3     INITIAL-MAXIMUM NUMBER OF INNER ITERATIONS DURING FIRST OUTER ITERATION
KTR      -1     CONTROL OF PRINTOUT OF NEUTRON FLUXES (-1/0/1 : ALL/NONE/SELECTED FLUXES TO BE PRINTED)
KDUM      0     CROSS-SECTION INPUT BY SIGMN-BLOCK ( 1=NO, 0=YES)
NFPERT    0     FORTRAN UNIT FOR INTERFACE FILE TO PERTURBATION MODULE
NFDMPN    0     FINAL FLUX WRITTEN ON UNIT (0=NOT USED, >0 : FORTRAN UNIT NUMBER OF DATA SET)
NFDMPO    0     INITIAL FLUX READ FROM UNIT (0=NOT USED, >0 : FORTRAN UNIT NUMBER OF DATA SET)
INBO      0     CONTROL OF VARIOUS BOUNDARY FLUX OPTIONS ( 0: NO BOUNDARY FLUX; 1: BOUNDARY FLUX BY INTERVAL AND GROUP;
                                                          2: BY ENERGY SPECTRUM AND SOURCE)

K4:
IQUELL    0     DISTRIBUTED SOURCE OPTION INDICATOR ( 0: NONE; 1: GIVEN BY TRIANGLE AND GROUP; -1: GIVEN BY MIXTURE AND GROUP;
                                                      2: GIVEN BY SPECTRUM AND SOURCE FOR EACH TRIANGLE;
                                                     -2: GIVEN BY SPECTRUM AND SOURCE FOR EACH MIXTURE)
MBK       0     BUCKLING CORRECTION OPTION ( 0: NONE; 1: CONSTANT VALUE; 2: GROUP DEPENDENT VALUES; 3: GROUP AND MIXTURE DEPENDENT)
IQUER     1     PRINT CROSS-SECTION TABLES (0=NO,1=YES)
ID1       0     ACTIVATION RATES OR DENSITIES FOR FISSION CROSS-SECTION ( 0/1/2 : NONE/DENSITY/RATE )
ID2       0     ACTIVATION RATES OR DENSITIES FOR CAPTURE ACTIVATION CROSS SECTION  ( 0/1/2 : NONE/DENSITY/RATE )
ID3       0     ACTIVATION RATES OR DENSITIES FOR ABSORPTION CROSS-SECTION ( 0/1/2 : NONE/DENSITY/RATE )
ID4       0     ACTIVATION RATES OR DENSITIES FOR 'NUSF' CROSS-SECTION ( 0/1/2 : NONE/DENSITY/RATE )
ID5       0     ACTIVATION RATES OR DENSITIES FOR TOTAL CROSS-SECTION ( 0/1/2 : NONE/DENSITY/RATE )
ISCT      0     ANISOTROPY OPTION INDICATOR ( 0: ISOTROPIC; MAXIMUM ALLOWED: 6 )
KAUSW     0     NEUTRON BALANCES SELECTOR ( 0/1/10/11 : GLOBAL/GLOBAL AND ZONES/GLOBAL AND MIXTURES/ALL )
NFANI1    0     FORTRAN UNIT FOR ANISOTROPY WORKING DATASET
NFSCR1    9     FORTRAN UNIT FOR AUXILIARY WORKING DATASET FOR STORAGE OPTIMIZATION 2
INORM     1     FLUX NORMALIZATION INDICATOR ( 1: ONE FISSION; 2: POWER = 1 WATT )

K5:
EPS    1.0000E-04     CONVERGENCE CRITERION FOR OUTER ITERATIONS ( < 0: INTEGRAL; > 0: POINTWISE; = 0: USE TEPS )
EV     1.0000E+00     START VALUE FOR EIGENVALUE
BF     0.0            BUCKLING HEIGHT ( USED IF MBK = 1 )
H      2.3945E+00     LENGTH OF TRIANGLE SIDE (IN CM)
EPSA   1.0000E-04     CONVERGENCE TEST SELECTOR FOR INNER ITERATIONS ( < 0: INTEGRAL; > 0; POINTWISE; = 0: USE EPS )
TEPS   0.0            CONVERGENCE CRITERION FOR OUTER ITERATIONS IF EIGENVALUE BOUNDS ARE REQUESTED (EPS=0.0)

K7:
USED MIXTURE NUMBERS

       1
```

```
TITLE OF THIS RUN:

*************************************************************************
*                                                                       *
*    TESTCASE 1 - K INFINITY CALCULATION IN S4                          *
*                                                                       *
*************************************************************************

NUMBER OF CALCULATIONAL POINTS PER ANGLE AND GROUP:    32

***** SUBROUTINE FILLC (INTERFACING SUBROUTINE TO SIGMN BLOCKS) ENTERED
***** VERSION 3.0 DATED MAY 16TH, 1986

***** CONTROL OUTPUT OF INTEGERS ON ENTRY:

NUMBER OF ENERGY GROUPS                      IGM =   4
TOTAL XS LOCATED IN ROW                      IHT =   7
SELFSCATTERING TERM LOCATED IN ROW           IHS =   8
UPSCATTER WIDTH                             IUPS =   0
DOWNSCATTER WIDTH                          IDOWS =   0
NO. OF ROWS IN XS-TABLE C                    IHM =  11
NUMBER OF MIXTURES USED                   NUMMAT =   1
NUMBER OF TABLES TO STORE                 NUMTAB =   3
ORDER OF ANISOTROPY                         ISCT =   0
IF .NE. 0, STORE MIXTURE DEPENDENT CHI     MDCHI =   0
TRANSPORT CORRECTION DONE IF = 1          NTRANS =   1

***** SIGNIFICANT INTEGERS WHICH WERE POSSIBLY CHANGED BY FILLC:

NUMBER OF ENERGY GROUPS                      IGM =   4
TOTAL XS LOCATED IN ROW                      IHT =   7
SELFSCATTERING TERM LOCATED IN ROW           IHS =   8
UPSCATTER WIDTH                             IUPS =   0
DOWNSCATTER WIDTH                          IDOWS =   3
NO. OF ROWS IN XS-TABLE C                    IHM =  11

*****  7  SCALAR SIGMN TYPES WILL BE STORED IN THE FOLLOWING SEQUENCE INTO ARRAY C:
***** SFISS      SCAPT      SN2N       STR        SABS       NUSF       STR
***** DEFINITION OF TYPE SABS := SCAPT + SFISS

***** SIGMN MIXTURE NUMBERS :
 1
***** ARE STORED IN TABLE    :
 1

******* STORAGE SEQUENCE CHANGED BY FILTRA TO
******* SFISS      SCAPT      SN2N       STR        SABS       NUSF       STRTR

******* TRANSPORT CORRECTION APPLIED TO CROSS SECTION MATRIX C IN SUBROUTINE FILTRA WITHOUT ERROR

***** SUBROUTINE FILLC TERMINATING WITH KONTRO=  0
***** REGULAR END

**** WARNING **** IN XS-TABLE:  1 GROUP:  1 INPUT VALUE OF SABS =   0.60468E-01 SET TO SABS =   0.59931E-01
```

ANGULAR QUADRATURE PERFORMED WITH P N SET

CHECK OF SN CONSTANTS:

SUM (WEIGHT)	:	1.0000E+00
SUM (WEIGHT*MUE)	:	-1.1548E-07
SUM (WEIGHT*ETA)	:	3.1292E-07
SUM (WEIGHT*MUE**2)	:	3.3333E-01
SUM (WEIGHT*ETA**2)	:	3.3333E-01
SUM (WEIGHT*MUE*ETA)	:	0.0

DIRECTIONS MUE	DIRECTIONS ETA	WEIGHTS		
-0.94043E+00	0.0	0.0	1	NOT USED
-0.90839E+00	-0.24340E+00	0.54345E-01	2	
-0.66499E+00	-0.66499E+00	0.54345E-01	3	
-0.24340E+00	-0.90839E+00	0.54345E-01	4	
0.24340E+00	-0.90839E+00	0.54345E-01	5	
0.66499E+00	-0.66499E+00	0.54345E-01	6	
0.90839E+00	-0.24340E+00	0.54345E-01	7	
-0.50837E+00	0.0	0.0	8	NOT USED
-0.44026E+00	-0.25419E+00	0.57976E-01	9	
-0.15962E-06	-0.50837E+00	0.57976E-01	10	
0.44026E+00	-0.25419E+00	0.57976E-01	11	
-0.94043E+00	0.0	0.0	12	NOT USED
-0.90839E+00	0.24340E+00	0.54345E-01	13	
-0.66499E+00	0.66499E+00	0.54345E-01	14	
-0.24340E+00	0.90839E+00	0.54345E-01	15	
0.24340E+00	0.90839E+00	0.54345E-01	16	
0.66499E+00	0.66499E+00	0.54345E-01	17	
0.90839E+00	0.24340E+00	0.54345E-01	18	
-0.50837E+00	0.0	0.0	19	NOT USED
-0.44026E+00	0.25419E+00	0.57976E-01	20	
-0.15962E-06	0.50837E+00	0.57976E-01	21	
0.44026E+00	0.25419E+00	0.57976E-01	22	

DIRECTIONS NUMBER IN ORDER OF EXPLORATION	REFLECTIVE DIRECTIONS LEFT	TOP	RIGHT	
2	5	17	13	1
3	4	18	14	2
6	17	13	5	3
7	18	14	4	4
15	14	4	18	5
16	13	5	17	6
17	16	2	6	7
18	15	3	7	8
13	2	6	16	9
14	3	7	15	10
4	7	15	3	11
5	6	16	2	12
9	10	22	20	13
11	22	20	10	14
21	20	10	22	15
22	21	9	11	16
20	9	11	21	17
10	11	21	9	18

```
FISSION FRACTIONS
0.77655E+00 0.22345E+00 0.0          0.0
GROUP SUM= 1.00000E+00


CROSS SECTIONS


XS-TABLE   1

     GROUP    1             2             3             4

TYPE

SFISS       0.57545E-01   0.58916E-01   0.16362E+00   0.44838E+00
SCAPT       0.29234E-02   0.10859E-01   0.56226E-01   0.23215E+00
SN2N        0.53731E-03   0.0           0.0           0.0
STR         0.23103E+00   0.34149E+00   0.89260E+00   0.10564E+01
SABS        0.59931E-01   0.69775E-01   0.21985E+00   0.68053E+00
NUSF        0.15524E+00   0.14580E+00   0.39661E+00   0.10863E+01
STRTR       0.23188E+00   0.36095E+00   0.91315E+00   0.13231E+01
S  G->G     0.11941E+00   0.29115E+00   0.69319E+00   0.64262E+00
G- 1->G     0.0           0.52518E-01   0.23709E-04   0.10566E-03
G- 2->G     0.0           0.0           0.21618E-04   0.17267E-06
G- 3->G     0.0           0.0           0.0           0.72812E-06
```

MIXTURE DISTRIBUTION ON THE INTERVALS

BOUNDARY CONDITIONS : 90=VACUUM, 91=REFLECTIVE, 92=FIXED BOUNDARY FLUX

```
     1   2   3   4   5   6   7   8
0-------------------------------
    90  90  90  90  90  90  90  90
  90  90  90  90  90  90  90  90
1-------------------------------
    90  90  90  90  90  90  90  90
  90  90  91  91  91  91  90  90
2-------------------------------
    90  91   1   1   1   1  90  90
  90  90   1   1   1   1  91  90
3-------------------------------
    90  91   1   1   1   1  90  90
  90  90   1   1   1   1  91  90
4-------------------------------
    90  91   1   1   1   1  90  90
  90  90   1   1   1   1  91  90
5-------------------------------
    90  91   1   1   1   1  90  90
  90  90   1   1   1   1  91  90
6-------------------------------
    90  90  91  91  91  91  90  90
  90  90  90  90  90  90  90  90
7-------------------------------
    90  90  90  90  90  90  90  90
  90  90  90  90  90  90  90  90
8-------------------------------
```

TITLE OF THIS RUN:

```
**************************************************************************
*                                                                        *
*    TESTCASE 1 - K INFINITY CALCULATION IN S4                           *
*                                                                        *
**************************************************************************
```

INFORMATION ON VECTOR LENGTHS FOR THIS CASE :
MINIMUM VECTORLENGTH 1
MAXIMUM VECTORLENGTH 4
AVERAGE VECTORLENGTH 3
NO. OF SPATIAL MESHES 32

OUTER CONVERGENCE BASED ON POINTWISE TEST
INNER CONVERGENCE BASED ON POINTWISE TEST

***** ITERATION MONITOR *****

OUTER ITERATIONS	INNER ITERATIONS	EV(LOW)	EIGENVALUE	EV(HIGH)	ERR(OUT)	GROUP	INNER ITERATIONS	ERR(INN)	POINTS OF MAX. ERRORS OUT/INN	REBAL. FACTOR	TIME USED SECONDS
0	0	0.0	1.000000	0.0	1.000E+00				(0, 1)		0.0
						1	0	0.0	(0, 1)	0.0	
						2	0	0.0	(0, 1)	0.0	
						3	0	0.0	(0, 1)	0.0	
						4	0	0.0	(0, 1)	0.0	
1	12	0.0	2.262547	2.304833	5.661E-01				(4, 4)		0.04400
						1	3	9.500E-02	(3, 6)	1.011E+00	
						2	3	5.706E-02	(4, 5)	9.799E-01	
						3	3	5.235E-02	(4, 5)	9.792E-01	
						4	3	3.126E-02	(4, 5)	9.964E-01	
2	16	2.270133	2.288019	2.337758	3.217E-02				(3, 6)		0.01600
						1	1	1.550E-02	(3, 6)	1.003E+00	
						2	1	4.740E-02	(3, 6)	1.013E+00	
						3	1	5.085E-02	(3, 6)	1.014E+00	
						4	1	3.344E-02	(3, 6)	1.005E+00	
3	35	2.277327	2.296112	2.368564	3.401E-02				(3, 6)		0.06800
						1	3	1.338E-03	(3, 6)	1.000E+00	
						2	6	2.296E-03	(3, 6)	1.000E+00	
						3	6	5.176E-03	(3, 6)	1.000E+00	
						4	4	3.037E-03	(3, 6)	1.000E+00	
4	44	2.291723	2.295793	2.310397	6.184E-03				(3, 6)		0.03400
						1	3	1.549E-03	(3, 6)	1.000E+00	
						2	1	2.652E-03	(3, 6)	9.998E-01	
						3	3	2.661E-03	(3, 6)	1.000E+00	
						4	2	3.102E-03	(3, 6)	1.000E+00	

5	64	2.293021	2.296497	2.309376	5.882E-03				(3, 6)		0.07200
						1	3	4.894E-04	(3, 6)	1.000E+00	
						2	6	4.245E-04	(3, 6)	1.000E+00	
						3	7	5.367E-04	(3, 6)	1.000E+00	
						4	4	5.605E-04	(3, 6)	1.000E+00	
6	73	2.295539	2.296463	2.299628	1.362E-03				(3, 6)		0.03400
						1	3	3.704E-04	(3, 6)	1.000E+00	
						2	1	5.499E-04	(3, 6)	1.000E+00	
						3	2	5.039E-04	(3, 6)	1.000E+00	
						4	3	3.338E-04	(3, 6)	1.000E+00	
7	93	2.295830	2.296575	2.299557	1.346E-03				(3, 6)		0.07100
						1	3	1.252E-04	(3, 6)	1.000E+00	
						2	6	1.059E-04	(3, 6)	1.000E+00	
						3	7	1.206E-04	(3, 6)	1.000E+00	
						4	4	1.057E-04	(3, 6)	1.000E+00	
8	103	2.296312	2.296577	2.297529	4.163E-04				(3, 6)		0.03500
						1	3	9.733E-05	(3, 6)	1.000E+00	
						2	2	1.326E-04	(3, 6)	1.000E+00	
						3	2	1.325E-04	(3, 6)	1.000E+00	
						4	3	8.345E-05	(3, 6)	1.000E+00	
9	112	2.296438	2.296583	2.297048	2.056E-04				(3, 6)		0.03200
						1	2	8.655E-05	(3, 6)	1.000E+00	
						2	2	9.561E-05	(3, 6)	1.000E+00	
						3	3	8.094E-05	(3, 6)	1.000E+00	
						4	2	7.933E-05	(3, 6)	1.000E+00	
10	116	2.296516	2.296583	2.296751	7.397E-05				(3, 6)		0.01400
						1	1	7.325E-05	(3, 6)	1.000E+00	
						2	1	7.480E-05	(3, 6)	1.000E+00	
						3	1	7.445E-05	(3, 6)	1.000E+00	
						4	1	7.021E-05	(3, 6)	1.000E+00	
11	120	2.296526	2.296588	2.296732	6.568E-05				(3, 6)		0.01800
NEGATIVE FLUX FIXUP ACTIVATED											
			0 TIMES								
			0 TIMES			1	1	6.539E-05	(3, 6)	1.000E+00	
			0 TIMES			2	1	6.515E-05	(3, 6)	1.000E+00	
			0 TIMES			3	1	7.194E-05	(3, 6)	1.000E+00	
			0 TIMES			4	1	6.855E-05	(3, 6)	1.000E+00	

***** INFORMATION *****
FINISHED BY CONVERGENCE OF OUTER ITERATIONS

A FINAL ITERATION IS PERFORMED

FLUX NORMALIZATION PARAMETER INORM= 1
(1: FLUXES NORMALIZED TO UNIT FISSION SOURCE IN THE CALCULATED REGION OF THE REACTOR
2: FLUXES NORMALIZED TO A POWER OF 1 WATT IN THE CALCULATED REGION OF THE REACTOR)

NORMALIZATION FACTOR IS : 1.2587E-02

NEUTRON BALANCE OF THE WHOLE SYSTEM

GR.	EXTERN.SOURCE	FISSION SOURCE	INSCATTER	TOTAL PRODUCT.	TOTAL FLUX	FISSION PROD.	SELFSCATTER
1	0.0	0.7765474E+00	0.0	0.7765474E+00	0.6904290E+01	0.1071836E+01	0.8244231E+00
2	0.0	0.2234526E+00	0.3626016E+00	0.5860543E+00	0.8396058E+01	0.1224117E+01	0.2444489E+01
3	0.0	0.0	0.3483165E-03	0.3483165E-03	0.1583558E-02	0.6280537E-03	0.1097708E-02
4	0.0	0.0	0.6644183E-05	0.6644183E-05	0.9763296E-05	0.1060627E-04	0.6274054E-05
5	0.0	0.1000000E+01	0.3629565E+00	0.1362956E+01	0.1530194E+02	0.2296591E+01	0.3270015E+01

GR.	ABSORPTION	BUCKL.LEAKAGE	OUTSCATTER	TOTAL LEAKAGE	TOTAL LOSSES	CAPTURE DENSITY	FISSION DENSITY
1	0.4137784E+00	0.0	0.3627559E+00	0.0	0.7765343E+00	0.2018370E-01	0.3973048E+00
2	0.5858327E+00	0.0	0.2005084E-03	0.0	0.5860332E+00	0.9117585E-01	0.4946582E+00
3	0.3481456E-03	0.0	0.1673204E-06	0.0	0.3483128E-03	0.8903704E-04	0.2591081E-03
4	0.6644206E-05	0.0	0.0	0.0	0.6644206E-05	0.2266551E-05	0.4377664E-05
5	0.9999658E+00	0.0	0.3629565E+00	0.0	0.1362922E+01	0.1114508E+00	0.8922265E+00

GR.	NEUTRON BAL.	NO-WEST	NO-EAST	SOUTH	SO-EAST	SO-WEST	NORTH
1	0.1000016E+01	0.0	0.0	0.0	0.0	0.0	0.0
2	0.1000035E+01	0.0	0.0	0.0	0.0	0.0	0.0
3	0.1000010E+01	0.0	0.0	0.0	0.0	0.0	0.0
4	0.9999964E+00	0.0	0.0	0.0	0.0	0.0	0.0
5	0.1000025E+01	0.0	0.0	0.0	0.0	0.0	0.0

SCALAR FLUX OF GROUP 1 FOR EACH SPACE POINT

	1	2	3	4	5	6	7	8
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	8.691E-02	8.691E-02	8.691E-02	8.691E-02	0.0	0.0
	0.0	0.0	8.690E-02	8.691E-02	8.691E-02	8.691E-02	0.0	0.0
3	0.0	0.0	8.690E-02	8.691E-02	8.691E-02	8.691E-02	0.0	0.0
	0.0	0.0	8.690E-02	8.690E-02	8.691E-02	8.691E-02	0.0	0.0
4	0.0	0.0	8.690E-02	8.690E-02	8.691E-02	8.691E-02	0.0	0.0
	0.0	0.0	8.689E-02	8.690E-02	8.690E-02	8.691E-02	0.0	0.0
5	0.0	0.0	8.689E-02	8.690E-02	8.690E-02	8.691E-02	0.0	0.0
	0.0	0.0	8.689E-02	8.689E-02	8.690E-02	8.690E-02	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8								

SCALAR FLUX OF GROUP 2 FOR EACH SPACE POINT

	1	2	3	4	5	6	7	8
0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
1	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
2	0.0 0.0	0.0 0.0	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	0.0 0.0	0.0 0.0
3	0.0 0.0	0.0 0.0	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	0.0 0.0	0.0 0.0
4	0.0 0.0	0.0 0.0	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	0.0 0.0	0.0 0.0
5	0.0 0.0	0.0 0.0	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	1.057E-01 1.057E-01	0.0 0.0	0.0 0.0
6	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
7	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
8								

SCALAR FLUX OF GROUP 3 FOR EACH SPACE POINT

	1	2	3	4	5	6	7	8
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.993E-05	1.993E-05	1.994E-05	1.994E-05	0.0	0.0
	0.0	0.0	1.993E-05	1.993E-05	1.994E-05	1.994E-05	0.0	0.0
3	0.0	0.0	1.993E-05	1.993E-05	1.993E-05	1.994E-05	0.0	0.0
	0.0	0.0	1.993E-05	1.993E-05	1.993E-05	1.994E-05	0.0	0.0
4	0.0	0.0	1.993E-05	1.993E-05	1.993E-05	1.993E-05	0.0	0.0
	0.0	0.0	1.993E-05	1.993E-05	1.993E-05	1.993E-05	0.0	0.0
5	0.0	0.0	1.993E-05	1.993E-05	1.993E-05	1.993E-05	0.0	0.0
	0.0	0.0	1.992E-05	1.993E-05	1.993E-05	1.993E-05	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8								

SCALAR FLUX OF GROUP 4 FOR EACH SPACE POINT

	1	2	3	4	5	6	7	8
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
3	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
4	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
5	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
	0.0	0.0	1.229E-07	1.229E-07	1.229E-07	1.229E-07	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8								

VOLUME-INTEGRATED FISSION SOURCE

GROUP	FISSIONS
1	0.77655E+00
2	0.22345E+00
3	0.0
4	0.0
5	0.10000E+01

TITLE OF THIS RUN:

```
**************************************************************************
*                                                                        *
*    TESTCASE 1 - K INFINITY CALCULATION IN S4                           *
*                                                                        *
**************************************************************************
```

********** FINAL ITERATION RESULT **********

INFORMATION ON VECTOR LENGTHS FOR THIS CASE :
MINIMUM VECTORLENGTH 1
MAXIMUM VECTORLENGTH 4
AVERAGE VECTORLENGTH 3
NO. OF SPATIAL MESHES 32

OUTER CONVERGENCE BASED ON POINTWISE TEST

INNER CONVERGENCE BASED ON POINTWISE TEST

OUTER ITERATIONS	INNER ITERATIONS	EV(LOW)	EIGENVALUE	EV(HIGH)	ERR(OUT)	GROUP	INNER ITERATIONS	ERR(INN)	POINTS OF MAX. ERRORS OUT/INN	REBAL. FACTOR	TIME USED SECONDS
12	124	2.296537	2.296588	2.296701	4.989E-05				(3, 6)		0.01700
NEGATIVE FLUX FIXUP ACTIVATED											
0 TIMES						1	1	4.590E-05	(3, 6)	1.000E+00	
0 TIMES						2	1	5.412E-05	(3, 6)	1.000E+00	
0 TIMES						3	1	6.372E-05	(3, 6)	1.000E+00	
0 TIMES						4	1	5.770E-05	(3, 6)	1.000E+00	

NEGATIVE FLUX FIX UP SUMMARY FOR LAST OUTER ITERATION
IN EACH GROUP THERE ARE 576 ANGULAR FLUXES CALCULATED

NEGATIVE ANGULAR FLUXES IN GROUP 1 : 0 TIMES ENCOUNTERED (= 0.0 PERCENT)
NEGATIVE ANGULAR FLUXES IN GROUP 2 : 0 TIMES ENCOUNTERED (= 0.0 PERCENT)
NEGATIVE ANGULAR FLUXES IN GROUP 3 : 0 TIMES ENCOUNTERED (= 0.0 PERCENT)
NEGATIVE ANGULAR FLUXES IN GROUP 4 : 0 TIMES ENCOUNTERED (= 0.0 PERCENT)

***** INFORMATION *****
FINISHED BY CONVERGENCE OF OUTER ITERATIONS

```
**********  TIMING STATISTICS  **********

TIME USED BY THE CASE WITH ID=     1            0.5100 SECONDS
AVERAGE TIME PER OUTER ITERATION                0.0425 SECONDS
INNER ITERATION USED                            0.4190 SECONDS =   82.2 PER CENT
GROUP SOURCE CALCULATIONS USED                  0.0    SECONDS =    0.0 PER CENT
FLUX MOMENT CALCULATION USED                    0.0    SECONDS =    0.0 PER CENT
AVERAGE TIME PER INNER ITERATION                0.0041 SECONDS
AVERAGE TIME PER MESH                           7.1404 MICROSECONDS

TIME USED BY THE WHOLE JOB UP TO NOW            0.5110 SECONDS =      0.0085 MINUTES

KEYWORD  ENDE  ENCOUNTERED IN INPUT STOPS EXECUTION

FREE MAIN STORAGE PLACE AT THE END :    214378 WORDS
```

```
 ****   *   ***   *   *   ***   *   * *****   **
 *   *  *  *   *  ** **  *   *  **  *   *     * *
 *   *  *  *****  * * *  *****  * * *   *      *
 *   *  *  *   *  *   *  *   *  *  **   *      *
 ****   *  *   *  *   *  *   *  *   *   *     ****

*   *   ***   ****   *   *   ***   *        *****  *   *  ****
**  *  *   *  *   *  ** **  *   *  *        *      **  *  *   *
* * *  *   *  ****   * * *  *****  *        ****   * * *  *   *
*  **  *   *  *  *   *   *  *   *  *        *      *  **  *   *
*   *   ***   *   *  *   *  *   *  *****    *****  *   *  ****
```

APPENDIX C. DIAMANT2 STOP CODES

ISTOP	MEANING
0	JOB ENDED WITHOUT ERRORS
1	INVALID OR DOUBLY DEFINED UNIT NUMBERS
2	MISSING UNITS FOR STORAGE OPTIMIZATION 2
3	INSUFFICIENT MAIN STORAGE
4	ERROR IN READKO
5	END OF DATA ON UNIT NFI
6	TRANSMISSION ERROR ON UNIT NFI
7	END OF DATA ON UNIT NFDMPO OR NFDMPN
8	TRANSMISSION ERROR ON UNIT NFDMPO OR NFDMPN
9	END OF DATA ON UNIT NFANI1 OR NFANI2
10	TRANSMISSION ERROR ON UNIT NFANI1 OR NFANI2
11	END OF DATA ON UNIT NFSCR1 OR NFSCR2
12	TRANSMISSION ERROR ON UNIT NFSCR1 OR NFSCR2
13	ZERO FISSION SOURCE STOPS CALCULATION
14	FISSION SPECTRUM VANISHES
15	TYPE IN LAYOUT CARD IS INVALID
16	INPUT VALUES OF THE LAYOUT CARD ARE OUT OF BOUNDS
17	INCORRECT SN-ORDER
18	IEVT = 0 AND IQUELL = 0 NOT PERMITTED
19	NUMBER OF FLUXES TO PRINT OUT NOT CORRECT
-20	PROGRAM FINISHED BY MAXIMUM NUMBER OF OUTER ITERATIONS
-21	PROGRAM FINISHED BY MAXIMUM CPU TIME
22	ERROR IN FILLC
23	INCONSISTENT INPUT VALUES FOR MT, MAT, MATTAB

Whereas positive error codes usually signal severe errors, a negative error code means that the iteration process has not been finished yet. Nevertheless, in the latter case results might be meaningful.

ACKNOWLEDGMENTS

The authors want to thank B.delMarmol and G.Minsart for providing the very first version of the code. Special thanks are due to E.Kiefhaber who accompanied our efforts with helpful discussions and support. Especially, he has influenced the re-design of the iteration strategy and also put much time in carefully reviewing the many drafts of this reports. Many thanks are due to R.Böhme, C.Broeders, I.Broeders, G.Buckel, H.Giese, A.Polch and D.Thiem for useful remarks concerning the draft version of this report. Special thanks are due to J.Marek for his help with some of the figures. D.Sanitz at KfK, GSI Darmstadt, KFA Jülich and the universities of Karlsruhe and Kaiserslautern helped in providing access to the different vector computers.

REFERENCES

/1/ K.Küfner, R.Heger,
"DIAMANT2 - Ein Multigruppen Neutronentransportprogramm für Dreiecks- und Hexagonalgeometrie",
KfK 3033, Karlsruhe 1980 (in German; english translation available from RSIC)

/2/ K.Küfner, B.DelMarmol, G.Minsart,
"Continuous Four Point Triangular Mesh Difference Schemes for the Multigroup Neutron Transport Equation",
Proc. Top. Meeting on Computational Methods, pp.4-35 to 4-47, Williamsburg 1979

/3/ K.Kobayashi, G.Buckel, K.Küfner,
"TPTRIA, A Computer Program for the Reactivity and Kinetic Parameters for Two-Dimensional Triangular Geometry by Transport Perturbation Theory",
KfK 4116, Karlsruhe 1986

/4/ G.Buckel, W.Höbel,
"Das Karlsruher Programmsystem KAPROS, Teil 1",
KfK 2253, Karlsruhe 1976 (in German; english translation: Risley Trans 5016, UKAEA 1976)

/5/ K.D.Lathrop, F.W.Brinkley,
"TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport",
LA-4848-MS, Los Alamos 1973

/6/ W.F.Walters et al.,
Trans. Am. Nucl. Soc., 28, 257 (1978)

/7/ B.G.Carlson,
"Tables of Equal Weight Quadrature EQN Over the Unit Sphere"
LA-4743, Los Alamos 1971

/8/ E.E. Lewis, W.F. Miller,Jr.
"Computational Methods of Neutron Transport"
Wiley & Sons New York, 1984

/9/ K.Küfner,
"READKO - Ein Unterprogrammpaket für zentralisierte Ein- und Ausgabeoperationen in KAPROS (Version(1.8))"
KfK 3333, Karlsruhe 1982 (in German)

/10/ K.Küfner,
unpublished note,1986 (in German)
/11/ H.Bachmann, D.Sanitz,
unpublished note,1970 (in German)
/12/ H.Borgwaldt,
unpublished note,1986 (in German)
/13/ "RXVP80 - The Verification and Validation System for FORTRAN",
Version 4.0, User's manual
General Research Corp., Santa Barbara, California, 1985
/14/ K.Küfner, J.Burkhard, R.Heger,
"A Set of Test Cases for the 2-d Static Neutron Transport Code DIAMANT2
for Regular Triangular Geometry",
report in preparation (1987)
/15/ International Organization for Standardization,
"Programming Language FORTRAN",
ISO 1539-1980 or ANSI X3.9-1978,
/16/ Cyber 205 Service,
"VAST Automatic Vectorizer User Guide",
Control Data 1983