EUR 4779 e

COMMISSION OF THE EUROPEAN COMMUNITIES

CARONTE
THE EURATOM MODULAR CALCULATIONAL SYSTEM

by

G. BUCCARI. G. FATTORI, C. MONGINI-TAMAGNINI (EURATOM)
F. ASTIGLIANO (PRAXIS CALCOLO)

1972

Joint Nuclear Research Centre
Ispra Establishment - Italy
Scientific Data Processing Centre - CETIS
and
Contract EURATOM / PRAXIS CALCOLO S.p.A., Milano - Italy
No. 045-70-06 CETI

LEGAL NOTICE

This report is on sale at the addresses listed on cover page 4

at the price of B.Fr. 150.—

When ordering, please quote the EUR number and the title, which are indicated on the cover of each report.

Printed by L. Vanmelle, Gand
Luxembourg, February 1972

EUR 4779 e

CORRIGENDUM

See cover, page 1 and bibliographical fiche :

Read F. ASTIGIANO instead of Fr. Astigliano

EUR 4779 e

CARONTE THE EURATOM MODULAR CALCULATIONAL SYSTEM
by G. BUCCARI, G. FATTORI, C. MONGINI-TAMAGNINI (EURATOM) and F. ASTIGLIANO (PRAXIS CALCOLO)

Commission of the European Communities
Joint Nuclear Research Centre — Ispra Establishment (Italy)
Scientific Data Processing Centre — CETIS
and
Contract Euratom PRAXIS CALCOLO S.p.A. — Milano (Italy)
No. 045-70-06 CETI
Luxembourg, February 1972 — 112 Pages — 1 Figure — B.Fr. 150.—

The CARONTE system, developed at Euratom CCR Ispra-Cetis, controls the automatic execution of series of programs. The sequence is chosen by the user starting from a library of programs conveniently adapted.

EUR 4779 e

CARONTE THE EURATOM MODULAR CALCULATIONAL SYSTEM
by G. BUCCARI, G. FATTORI, C. MONGINI-TAMAGNINI (EURATOM) and F. ASTIGLIANO (PRAXIS CALCOLO)

Commission of the European Communities
Joint Nuclear Research Centre — Ispra Establishment (Italy)
Scientific Data Processing Centre — CETIS
and
Contract Euratom PRAXIS CALCOLO S.p.A. — Milano (Italy)
No. 045-70-06 CETI
Luxembourg, February 1972 — 112 Pages — 1 Figure — B.Fr. 150.—

The CARONTE system, developed at Euratom CCR Ispra-Cetis, controls the automatic execution of series of programs. The sequence is chosen by the user starting from a library of programs conveniently adapted.

EUR 4779 e

CARONTE THE EURATOM MODULAR CALCULATIONAL SYSTEM
by G. BUCCARI, G. FATTORI, C. MONGINI-TAMAGNINI (EURATOM) and F. ASTIGLIANO (PRAXIS CALCOLO)

Commission of the European Communities
Joint Nuclear Research Centre — Ispra Establishment (Italy)
Scientific Data Processing Centre — CETIS
and
Contract Euratom PRAXIS CALCOLO S.p.A. — Milano (Italy)
No. 045-70-06 CETI
Luxembourg, February 1972 — 112 Pages — 1 Figure — B.Fr. 150.—

The CARONTE system, developed at Euratom CCR Ispra-Cetis, controls the automatic execution of series of programs. The sequence is chosen by the user starting from a library of programs conveniently adapted.

The CARONTE system strongly differs from the existing ones in its main principle, that for the class of programs to be inserted in the CARONTE « PROGRAM LIBRARY », the data to be transferred between the programs need not be classified in a fixed format. Consequently, a program needs no previous significant modifications since its input data and output results do not have to be standardized in any way. They have only to be grouped in numbered models, to be transferred, under CARONTE control, to and/or from a DATAPOOL stored in auxiliary memory. A private datapool on tape can be obtained. Due to this, the models stored in the DATAPOOL by the executed programs may have to be mixed and elaborated to form an input model to a next program. Such an operation is performed, according to the wish of the user, by suitable TRANSFER PROGRAMS running under CARONTE control, and set up for each field of application. The possibility of relying on TRANSFER PROGRAMS for the elaboration of models, has been shown to be extremely advantageous since it permits the handling of programs of any class, without previously studying a classification of data transfer in the class, and with a minimum of modification.

The CARONTE system strongly differs from the existing ones in its main principle, that for the class of programs to be inserted in the CARONTE « PROGRAM LIBRARY », the data to be transferred between the programs need not be classified in a fixed format. Consequently, a program needs no previous significant modifications since its input data and output results do not have to be standardized in any way. They have only to be grouped in numbered models, to be transferred, under CARONTE control, to and/or from a DATAPOOL stored in auxiliary memory. A private datapool on tape can be obtained. Due to this, the models stored in the DATAPOOL by the executed programs may have to be mixed and elaborated to form an input model to a next program. Such an operation is performed, according to the wish of the user, by suitable TRANSFER PROGRAMS running under CARONTE control, and set up for each field of application. The possibility of relying on TRANSFER PROGRAMS for the elaboration of models, has been shown to be extremely advantageous since it permits the handling of programs of any class, without previously studying a classification of data transfer in the class, and with a minimum of modification.

The CARONTE system strongly differs from the existing ones in its main principle, that for the class of programs to be inserted in the CARONTE « PROGRAM LIBRARY », the data to be transferred between the programs need not be classified in a fixed format. Consequently, a program needs no previous significant modifications since its input data and output results do not have to be standardized in any way. They have only to be grouped in numbered models, to be transferred, under CARONTE control, to and/or from a DATAPOOL stored in auxiliary memory. A private datapool on tape can be obtained. Due to this, the models stored in the DATAPOOL by the executed programs may have to be mixed and elaborated to form an input model to a next program. Such an operation is performed, according to the wish of the user, by suitable TRANSFER PROGRAMS running under CARONTE control, and set up for each field of application. The possibility of relying on TRANSFER PROGRAMS for the elaboration of models, has been shown to be extremely advantageous since it permits the handling of programs of any class, without previously studying a classification of data transfer in the class, and with a minimum of modification.

EUR 4779 e

COMMISSION OF THE EUROPEAN COMMUNITIES

CARONTE
THE EURATOM MODULAR CALCULATIONAL SYSTEM

by

G. BUCCARI. G. FATTORI, C. MONGINI-TAMAGNINI (EURATOM)
F. ASTIGLIANO (PRAXIS CALCOLO)

1972

Joint Nuclear Research Centre
Ispra Establishment - Italy
Scientific Data Processing Centre - CETIS
and
Contract EURATOM / PRAXIS CALCOLO S.p.A., Milano - Italy
No. 045-70-06 CETI

ABSTRACT

The CARONTE system, developed at Euratom CCR Ispra-Cetis, controls the automatic execution of series of programs. The sequence is chosen by the user starting from a library of programs conveniently adapted.

The CARONTE system strongly differs from the existing ones in its main principle, that for the class of programs to be inserted in the CARONTE « PROGRAM LIBRARY », the data to be transferred between the programs need not be classified in a fixed format. Consequently, a program needs no previous significant modifications since its input data and output results do not have to be standardized in any way. They have only to be grouped in numbered models, to be transferred, under CARONTE control, to and/or from a DATAPOOL stored in auxiliary memory. A private datapool on tape can be obtained. Due to this, the models stored in the DATAPOOL by the executed programs may have to be mixed and elaborated to form an input model to a next program. Such an operation is performed, according to the wish of the user, by suitable TRANSFER PROGRAMS running under CARONTE control, and set up for each field of application. The possibility of relying on TRANSFER PROGRAMS for the elaboration of models, has been shown to be extremely advantageous since it permits the handling of programs of any class, without previously studying a classification of data transfer in the class, and with a minimum of modification.

KEYWORDS

COMPUTERS
PROGRAMMING
DATA PROCESSING
IBM 360
AUTOMATION
CROSS SECTIONS
REACTORS
BURNUP
TRANSPORT THEORY
COMPUTER CALCULATIONS

INDEX OF CONTENT

GENERAL INTRODUCTION *)

With the increasing diffusion of III generation computers, it was everywhere felt the necessity to dispose of systems apt to control the automatic execution of series of programs. The sequence is chosen by the user starting from a library of programs conveniently adapted. The system CARONTE, developed to the purpose at Euratom CCR Ispra, CETIS, can be utilized for libraries of any type (scientific or administrative).
However the system has been applied at the moment being at Ispra to a library of nuclear reactor design programs. For this particular field a set of macrolanguage instructions has been set up, which greatly facilitate the use of Caronte. A decoding system has been prepared, capable of interpreting the macrolanguage, and of converting it to the normal instruction to be supplied to Caronte.
In the following the various aspects of the system Caronte will be described in detail.

PART I contains the description of the main features of the system, and can be considered the starting point for the understanding of all the others Parts.

PART II is concerned with a very detailed description of the caracteristics of Caronte and with the way one can use Caronte in the case in which a library of programs conveniently adapted to the system has been set up for a particular field, and whenever a macrolanguage for the field does not exist.

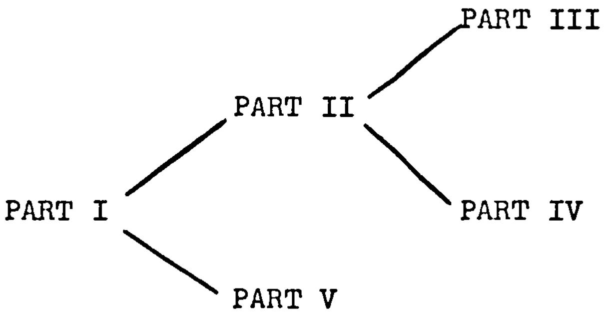
PART III deals with the rules to be followed to add programs to existing libraries, to set up new libraries, and finally to set up macrolanguages instructions.

PART IV describes in detail the adaptation to the system Caronte of a set of programs being the most frequently utilized in nuclear reactor design field. The input cards for the calculation of criticality and fluxes of ISPRA-1 reactor are annexed.

---

*) Manuscript received on 5 February 1972

Finally PART V is the actual user manual for people involved in nuclear reactor design field and whishing to use the special macrolanguage instructions set up for the field. The macrolanguage input cards for the calculation of criticality and fluxes of Ispra-1 reactor are annexed.
The scheme of logical connexion among PART I,II,...,V is the following :



It is meant to periodically distribute to the users updated versions of PART V.

——— o ———

The authors wish to aknowledge the eager encouragement in performing the work received by Miss G. POZZI and Mr. A. GAZZANO.

Besides the authors thank Mr. ANDREANI and Mr. PAVESI (PRAXIS Spa) for having programmed parts of the system.

PART I - Basic Concepts

INDEX

PART I - Basic concepts

INTRODUCTION

Given a group of programs, all concerned with a same field of science (Nuclear Reactor Programs, Civil Engineering Programs, Health and Safety Programs, ...) it happens quite frequently that the solution of a certain problem requires the run of a sequence of programs chosen in a given group, with mutual exchange of data among them.

The procedure followed in the past, when the available computers were the so called "first generation computers", was a very cumbersome one since manual interventions were necessary between nearly each program being run, in order to complete the input data of the program to be executed, starting from the results just obtained. In such a procedure the possibility of errors, besides, was quite remarquable.

Since the advent of the "second generation computers", being in respect to the previous ones much faster and bigger, the interest of the run of sequences of programs, has increased rapidly in all the world. An example in this direction was studied early in 1962 at the Martin Marietta Company where the APWRC(Army Pressurized Water Reactor Code) system was set up, consisting of four basic programs which were arbitrarily used individually or in a prescribed automatic combination.

However it was only with the advent of the "third generation computers" that actual detailed systems for the automation of sequences were studied throughout the world. The more advanced systems of the king are specifically constructed for the field of nuclear reactor research and most of them are being developed in the United States.

Generally such systems control the run of sequences of programs specially written to solve separate reactor problems (Cross section averaging, diffusion and transport calculation, burnup determination,..)

The programs inserted in the system are all written in such a way that the input data and the output results (classified in cross sections, fluxes, isotope concentrations, ...) are standardized, and stocked in a fixes format in an auxiliary memory so as to allow any program of the sequence to directly read the results of any pertinent program previously run.

The CARONTE system strongly differs from the existing ones in its main principle, that for the class of programs to be inserted in the CARONTE "PROGRAM LIBRARY", the data to be transferred between the programs need not to be classified in a fixed format. Consequently, a program needs no previous significant modifications since its input data and output results do not have to be standardized in any way. They have only to be grouped in numbered models called PROGRAM MODELS, to be transferred, under CARONTE control, to and/or from a DATAPOOL, stored in auxiliary memory.

Due to this, the PROGRAM MODELS stored in the DATAPOOL by the executed programs may have to be mixed and elaborated to form an input MODEL to a next program. Such an operation is performed, according to the wish of the user, by suitable TRANSFER PROGRAMS running under CARONTE control, and set up for each field of application. The possibility of relying on TRANSFER PROGRAMS for the elaboration of PROGRAM MODELS, has been shown to be extremely advantageous since it permits the handling of programs of any class, without previously studying a classification of data transfers in the class, and with a minimum of modification.

THE PROGRAM MODELS

More in particular the logical organization of CARONTE is the following.
Each program of the CARONTE "PROGRAM LIBRARY" is identified by a name, consisting of up to 8 alphanumeric characters. Each program contains numbered groups of models each of which can be read from or write on a private program DATAPOOL. The transfer from private program DATAPOOL and CARONTE DATAPOOL and viceversa is performed

by CARONTE system.

Whenever a group of data is written from a program on a private DATAPOOL is automatically labelled by the number of the related model. It must be noted besides that whenever a program searches a group of data on the private DATAPOOL, the label consisting in the number of the related model, is checked out.

However, as already stated, since input model and output model of the programs are not standardized in any way, it happens very frequently that a required by a program must be obtained by elaboration of one or more model stored on the DATAPOOL. Such an elaboration is performed by TRANSFER PROGRAMS written for every field of application of CARONTE, running automatically under CARONTE control.

THE TRANSFER PROGRAMS

In order to avoid that the user has to trouble himself with the specifications of the TRANSFER PROGRAMS to be used for the elaboration of the models, the following procedure has been established.

First of all the models contained in the ensemble of the programs are said to have the same "model number" when they contain the same kind of data, written in the same way. In the case of reactor programs field, a "model" can represent, as an example, a vector containing energy group fluxes : $\phi (E_1)$, $\phi (E_2)$..... $\phi (E_N)$ in ascending energies order.

The TRANSFER PROGRAMS operate on one or more "models" to produce another "model". In particular the TRANSFER PROGRAM being able to produce "model" number N, is identified by the number N. As an example, taken from reactor field, a TRANSFER PROGRAM can be the one that starting from microscopic thermal cross section, from microscopic fast cross sections, and from combining specifications, prepares a multigroup cross section library.

In any case the user does not deal with the TRANSFER PROGRAMS, since they are automatically controlled by CARONTE.

It must be noted that in order to give more flexibility to the system it has been taken into account the possibility that more than one transfer program exists, capable to operate on the same data set models, but following different "rules". An example taken as before from reactor field can be the preparation of a macroscopic library containing the diffusion coefficient D, starting from a microscopic library and from the isotope concentrations. As known the D term can be calculated in many different ways.

THE TEST ROUTINES

A run of CARONTE consists in the execution of a sequence of programs from the "PROGRAM LIBRARY". Are also permitted loops between two or more programs until certain conditions indicated by the user are satisfied. Such conditions will be examined by TEST routines contained in the pertinent TRANSFER PROGRAMS. An example taken from nuclear reactors can be an iterations between a space independent thermal cross sections averaging program and a space dependent flux calculation program, until self shielding factors are stabilized.

The models stored in the DATAPOOL by CARONTE system can be utilized by any program following in the same run, as well as by programs executed (even months later) in successive runs of CARONTE, by storing models on a Private Datapool. This permits the formation of complex ramified paths, which can be extended indefinitely.

THE SYSTEM ORGANIZATION

The CARONTE system has been written for IBM 360/65, and runs under the standard IBM O.S. control.

The ensamble of the programs inserted in the CARONTE system (that is the CARONTE PROGRAM LIBRARY), the TRANSFER PROGRAMS, and finally the DATAPOOL are all stored on direct access auxiliary storage. The PROGRAM LIBRARY, the ensemble of the TRANSFER PROGRAMS and the DATAPOOL are open ended.

The CARONTE system is organized in three parties, called NUCLEUS PROCESSOR and EXECUTOR. The NUCLEUS always resident in fast memory, first loods the PROCESSOR than loads the EXECUTOR which permits to the NUCLEUS itself to give control to the program required. The EXECUTOR can reside in fast memory with programs or alternatively depending on the memory available.

APPLICATION TO NUCLEAR FIELD

A certain number of nuclear programs have been chosen from those most utilized and have been adapted to CARONTE. For each of them an appropriate group of data has been selected to be transferred to and/or from the DATAPOOL.

Many TRANSFER programs dealing with elaboration or comparison of DATA SETS (models) have been written.

Finally, in order to avoid to the user the cumbersome numerical INPUT CONTROL instructions accepted by CARONTE, an alphabetic language related to the reactor field, has been developed.

A decoding system has been set up, capable of interpreting the alphabetic language and of converting it to the normal numerical control instructions to be supplied to CARONTE. Here again the decoding system is independent from the particular case of reactor programs, since only the "keyword table" is related to the reactor field.

BIBLIOGRAPHY

- C.N. Kelber, G. Jensen, L. Just, J.B. Toppel - "THE ARGONNE REACTOR : COMPUTATION SYSTEM ARC" - Conferencia International Sobra Utilizacion de Reactores de Investigation y Computo y Matematicas de Reactores - Mexico, May 1967

- E.D. Reilly, Jr., W.H. Turner - "THE AUTOMATION OF REACTOR DESIGN CALCULATIONS AT THE KNOLL ATOMIC POWER LABORATORY" - Conference of the Application of Computing Methods to Reactor Problems - Argonne, May 1965

- J.P Bayard, R. Boudet - "THE CODNUC SYSTEM - A MODULAR SYSTEM OF SUBROUTINES FOR REACTOR CALCULATION"- Conference on The Effective Use of Computers in Nuclear Industry - Knoxville, April 1969

- T. Watanabe, K.Arai, T. Noda - "HITACHI NUCLEAR CODES CONTROL SYSTEM, NCCS" - Conference on The Effective Use of Computers in Nuclear Industry - Knoxville, April 1969

- H.C. Honeck, J.E. Suich, J.C. Jensen, C.E. Bailey, J.W. Stewart- "JOSHUA - A REACTOR PHYSICS COMPUTATIONAL SYSTEM" - Conference on The Effective Use of Computers in Nuclear Industry - Knoxville, April 1969

- L.C. Just, P. Walker, B.J. Toppel - "RECENT DEVELOPMENTS AND CAPABILITIES IN THE ARC SYSTEM" - Conference on The Effective Use of Computers in Nuclear Industry - Knoxville, April 1969

PART II User's general manual

INDEX

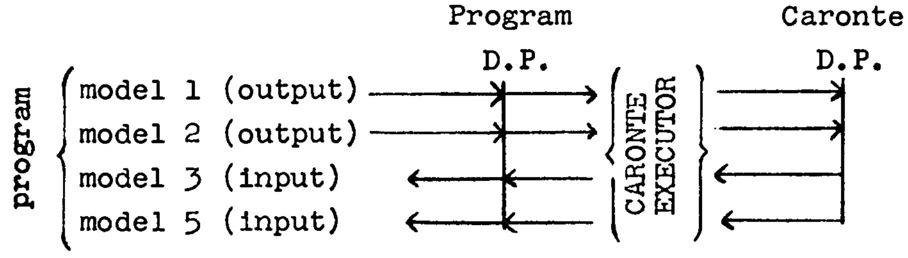
1. LIBRARY OF PROGRAMS, PROGRAM MODELS

The programs to run under Caronte control are stored in one or more program libraries, built up on direct access auxiliary storages with the normal rules of the IBM 360 Operating System.

In such libraries, in the form of partitioned data sets, the programs appear as executable modules each being identified by a name having up to eight alphanumeric characters, of which the first one must be alphabetic, following the normal notations of the IBM 360 Operating System.

For each program the data which may be transmitted from the program datapool to the Caronte DATA POOL (and/or viceversa) are subdivided in models having a certain meaning and a specific reading or writing form.
The models are identified each by a number. The user must know this subdivision and the direction of transmission (from and/or to the D.P.).

Example :



It is important to note that whenever a program has to write one of its output models on the program D.P., it writes before on the D.P. an identifying LABEL, containing the model number. The EXECUTOR checks this label and write the model on the Caronte DATA POOL.

Analogously whenever a program has to read from the program D.P. one of its input models it searches for a LABEL containing its own name and the model number.

Obviously the EXECUTOR copies the model requested from the CARONTE DATA POOL to the program DATA POOL.

In more detail the LABEL consists of 6 words being ;

1) XXXX
2) ⎫ Program name
3) ⎭
4) appearance name of the considered program in the sequence
5) model number
6) indicator = 1 is output model
   indicator = 0 is input model

There is the possibility to elaborate one or more MODELS to obtain a new one. This elaboration is performed by interface programs, called TRANSFER programs, stored in the program library.
Such trasfer programs are defined in an unique way by the models on which they operate and by the order of the models in the transfer expressions.

Let's consider as an example the following transfer expansions.

$$\begin{aligned} &3 + 2 + 4 \rightarrow 13 \\ &2 \rightarrow 14 \\ &9 + 9 \rightarrow 9 \\ &14 + 9 \rightarrow 7 \\ &11 + 12 + 7 \rightarrow 5 \\ &13 + 12 \rightarrow 8 \\ &8 + 15 \rightarrow 6 \\ &15 \rightarrow 7 \end{aligned}$$

However it is possible that may kind of elaborations exist apt to produce one model. In such a case we identify them by RULE NUMBERS:

$$2 + 3 \rightarrow 30 \qquad \begin{cases} \text{RULE 1} \\ \text{RULE 2} \end{cases}$$

The user of Caronte system is supposed to know for the considered field of application all the existing combinations among the models and if it is the case the different rules of elaborations.

MODELS TRANSMISSION

To clarify the way in which the set transmission among the programs of sequence actually is performed, some examples of increasing complexity will now be examined.

a) Let us suppose that the program LMN has to receive model 13 from models 3,1,4 of program ABC.

   At execution time, the program ABC, under Caronte control, stores on the PROGRAMS DATAPOOL models 1,3 and 4 placing into their LABELS its name (ABC), ordering number and the model number. Then the system CARONTE transfers these models in the CARONTE DATAPOOL and, having realized that an interface program type 3+1+4 → 13 is required, prepares to this the models 3,1,4 on the PROGRAMS DATAPOOL and then calls it from the library of programs for execution. The interface program reads and elaborates them and finally stores the results on PROGRAMS DATAPOOL preceded by a LABEL containing the program name LMN, ordering number and models numbers 13. The system CARONTE transfers this model on the CARONTE DATAPOOL and, before execution of program LMN, prepares to this the model 13 in the PROGRAM DATAPOOL.

b) Suppose now that the program PQR has to receive the input data corresponding to model 13 from the program ABC (models 3,1,4), DEF (model 5) and GHI (model 2) The models ABC (1), ABC (3), ABC (4), DEF (5), GHI (2) are transferred by the CARONTE system on the CARONTE DATAPOOL from the PROGRAMS DATAPOOL, after execution of each program.

   Before the PQR run, two interface programs are called by CARONTE being namely:

   3+1+4 → 6
   6+5+2 → 13

The first which operates on the models 1,3,4; the second which operates on the results just obtained named model 6 and on model 5 and 2.

The model 13 is transferred by CARONTE system from PROGRAMS DATAPOOL to CARONTE DATAPOOL and is labelled PQR model 1 which will be read by PQR program, after transfert to the PROGRAM DATAPOOL.

LOOPS

In the case that a loop exists between two (or more) programs the condition of loop exit is that some or all the items of a certain input model to one of the programs in the loop, differs to a given extent from the correspondent items of the input model of the same program in the proceding iteration. To such purpose some of the Transfer programs before storing in the Caronte D.P. the resulting model labelled with name of input program and relative model, compare the considered items with the one, stored with the same label in the previous iteration.
Depending on the result of the comparison, the model is stored or not and the indication is given to the system Caronte which will provide for other iteration or not depending on the case.
The user od the system must know which transfer program actually are constructed for dealing with this case. The loop exit condition must refer to an input model of a program which correspond to a model constructed by such a transfer.

4. THE CONTROL INPUT

4.1 A "run" of Caronte consists in the execution of a sequence of programs (with possibility of loop). A run can produce a private DATA POOL stored on tape or disk. There is the possibility to utilize the previously constructed D.P. for successives runs. This can be the case when a problem is not treated in an unique run due to any king of interruption (time, lines, power break of).

Besides, when the solution of a given problem requires the utilization of results stored in the D.P. by previous problems treated also months before.

When a new problem requiring no previous D.P. is to be treated, the first input control card is a BEGIN card. When a new problem requiring a previous D.P. is dealt with, the first input control card is a CONTINUATION card.

For both the BEGIN and CONTINUATION cases, a restrat after a whatever type of interruption requires as first card a RESTART card, containing the indications pointed out after the last complete executed program of the sequence of the last executed transfer or series transfers whose models output are a complete input for the programs not being executed. In the case of RESTART the input deck is not to be changed at all, except in its first RESTART card which substitutes previous BEGIN or CONTINUATION card. The exact format of above cards is indicated in annexe 1.

4.2. To the purpose of defining the sequence of the program to be executed the user must indicating, separated by commas, each program by its alphanumeric name, followed by a number specifying the ordering of appearance of a same program in the sequence (APPEARANCE NUMBER).

It is to be recalled that we intend for "name" of the program the alphanumeric identification plus the appearance number.
In the sequence specification is forseen the possibility of a loop between two or more programs. The loop exit condition must be specified by the user (see below). In this case the user must ne sure that among the programs appearing inside the loop at least one has a model, the transfer of which provides a compare option.

As for the appearance number in the case of loop, it is to be noted that any program in the loop is identified by an unique appearance number.

The whole of the information the user has to supply to the system concerning the sequence of program must be written down following certain prescriptions:

- the programs name, that is the alphanumeric identification (up to 8 characters) used in the Caronte library, is followed by a minus sign and a number, this last specifying the appearance number;
- the programs names must be separated by commas

EXAMPLE 1 (BEGIN problem)

ABC-1, DEF-1, ABC-2, GHI-1, ABC-3, GHI-2

EXAMPLE 2 (CONTINUATION problem) (suppose that the **PRIVATE** DATA POOL contains the results of the previous example)

DEF-2, LMN-1, ABC-4, ABC-5, ABC-6, LMN-2, ABC-7

- In the case of one or more programs repeated periodically, a given number of times, there is the possibility to indicate briefly as pointed out in examples 3 and 4, being respectively identical to example 1 and 2

EXAMPLE 3 - (BEGIN problem)

ABC-1, DEF-1, 2* (ABC-2, GHT-1)

EXAMPLE 4 -(CONTINUATION problem)

DEF-2, LMN-1, 3* (ABC-4), LMN-2, ABC-7

- In the case of loop the names of the program here in contained must be included in parenthesis

EXAMPLE 5(BEGIN problem)

ABC-1, DEF-1, (LMN-1, ABC-2), DEF-2, ABC-3

- The sequence of the programs must be actually indicated in the second card, called SQ card. The exact card format is given in annexe 1.

4.3 The informations concerning the transmission of models among the programs must be given to the system following the here given indications.

First of all a card of type .CTE DATA TRANSMISSION must preceed all other cards.

A model treated by a program is identified by the program name followed by the model number included in parenthesis

ABC-3(1) indicates model 1 treated by ABC program - appearance number

DEF-1(2) indicates model 2 treated by DEF program - appearance number 1.

In order to indicate that a particular model treated by a program is to be found by an elaboration on some models treated by other programs, the key word FROM is to be utilized.

EXAMPLE 6

```
GHI-1(1) FROM ABC-2(3)
```

EXAMPLE 7

```
LMN-2(4) FROM ABC-1(3), ABC-1(5), ABC-1(4)
```

EXAMPLE 8

```
PQR-1(20) FROM ((ABC-2(1), ABC-2(3), ABC-2(4)), DEF-1(5)).
                                                   GHI-1(2)
```

As for the exact formats in which these cards have to be written, see annexe 1.

It must be noted that in the case of more than one RULE for the transfer program, the indication : USING RULE   has to preceed the word FROM.
Analogously in the case of loops the indication : UNTIL DEVIATION EQUAL. has to preceed the word FROM.
In the case of both presence of RULE and Loop the order of the two indications to be inserted before "FROM" is immaterial.

THE COMPLETE DECK FOR A CARONTE RUN

At this point suppose the user has to prepare the complete input deck for a Caronte run.

The input consists in the normal input cards to be prepared for the various programs of the sequence, keeping in mind that the data which will be supplied to program by the programs D.P. have not to be included : in particular the cards containing data coming all from D.P. must be omitted, and the cards containing some data coming from D.P. must have the corresponding fields left blanck.
Each of such program input deck must be preceeded by a card containing the name of the program (alphanumeric characters + appearance number).
This identification card allows the program input decks not to follow the order of the programs as specified in the sequence card; in particular the presence of the identification card permits the

restart option, and the presence of just one program input deck for any program appearing in a loop. An END card must follow the input control cards, and another END card must close the whole input deck.
The arrangement of the complete input deck (program input + control input) is clearly shown in annexe II.
As for the Job control cards to be prepared after the job card, they are reported in annexe III.

THE MACROLANGUAGE

As for the part of the input control instructions related to the transfer of data among the programs, there exist the possibility to up, and use a macro language, much more sintetic and mnemonic. In this macro language one write down:

$$C\text{-}\beta \quad \text{Keyword} \quad C_1\text{-}\ \beta_1,\ C_2\text{-}\beta_2 \ \ldots\ldots\ldots C_n\text{-}\beta_n$$

where $C\text{-}\beta$ is the complet name of the program (alphanumeric characters appearance number) and the key word can be a 72 characters specificative possibly defined in such a way to have a mnemonic significance.

To set up, for each particular field of application, new developments (related to new key words) the user must follow a prescribed set of rules, being explained the point III.

Annexe I

```
.CTE  BEGIN
 SQ   ABC-1,DEF-1,ABC-2,GHI-1,ABC-3,GHI-2

.CTE  CONTINUATION
 SQ   DEF-2,LMN-1,ABC-4,ABC-5,ABC-6,LMN-2,ABC-7

.CTE  BEGIN
 SQ   ABC-1,DEF-1,2*(ABC-2,GHI-1)

.CTE  CONTINUATION
 SQ   DEF-2,LMN-1,3*(ABC-4),LMN-2,ABC-7

.CTE  BEGIN
 SQ   ABC-1,DEF-1,(LMN-1,ABC-2),DEF-2,ABC-3
```

Annexe II e III

```
//JOBLIB   DD DSNAME=CTE.CODE.LIBET,DISP=SHR
          EXEC PGM=NUCLEUS
//FT02F001 DD UNIT=SYSSQ,SPACE=((CYL,(5,1))
//FT03F001 DD UNIT=SYSSQ,SPACE=((CYL,(5,1))
//FT04F001 DD UNIT=SYSSQ,SPACE=(CYL,(5,1))
//FT05F001 DD UNIT=SYSSQ,SPACE=((CYL,(5,1))                              c
//               DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//FT06F001 DD SYSOUT=A
//FT07F001 DD SYSOUT=B
//FT08F001 DD UNIT=SYSSP,SPACE=(CYL,(5,1))
//FT99F001 DD UNIT=SYSSQ,SPACE=(CYL,(2,1)),DCB=(,RECFM=VS,BLKSIZE=800)
//CODLIBDD DD DISP=SHR,DSNAME=CTE.CODE.LIBET
//GO.TVCORRSP DD DSN=CTE.CODE.TVCORR,DISP=SHR
//GO.PTVCORRS DD DSN=CTE.CODE.PTVCORR,DISP=SHR
//GO.TABMACRO DD DSN=CTE.CODE.MACRO,DISP=SHR
//GO.PNTMACRO DD DSN=CTE.CODE.PMACRO,DISP=SHR
//GO.INPUTCR  DD DDNAME=SYSIN
//SYSPRINT  DD SYSOUT=A
//GO.PCGETAB DD UNIT=SYSSQ,SPACE=((CYL,(1,1))
//GO.PROGDATA DD UNIT=SYSSQ, SPACE=(CYL,(1,1))
//GO.WRITENT  DD UNIT=SYSSQ, SPACE=(CYL,(1,1))
//GO.WRITEPNT DD UNIT=SYSSQ, SPACE=(CYL,(1,1))
//DATAPOOL    DD UNIT=SYSSQ,SPACE=(CYL,(2,2))
```

```
//GO.PRIVDATA DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
//LIBCODDD   DD  DISP=SHR,DSNAME=CTE.CODE.LABET
//GO.SYSIN   DD *
.CTE   BEGIN
 SQ    PROG1-1,PROG2-1
.CTE   DATA TRANSMISSION
       PROG2-1 KEYWORD FROM PROG1-1
.CTE   END
.CTE   PROG1      1
  Input of PROG1   program

.CTE   PROG2      1
  Input of PROG2   program

.CTE   END
/*
```

If a PRIVATE DATAPOOL on tape is required, the DD card is the following

```
//GO.PRIVDATA   DD UNIT=TP9,LABEL=(1,SL),VOLUME=(PRIVATE,SER=******),    C
//                DSN=PIPPO,DISP=(NEW,PASS),DCB=(,RECFM=VS,BLKSIZE=800)
```

PART III Setting up and maintenance of librairies, tables, macrolanguage

INDEX

PART III : Setting-up and maintenance of libraries, tables and macrolanguage.

INTRODUCTION

This manual is devoted to the users wishing to set-up a new library of programs to be controlled by Caronte (programs concerning a whatever field of application, i.e. nuclear economics, heat transfer, other sciences).

In particular all the rules necessary for the modifications of the considered programs, and for the preparation of the interfaces between programs (that is the TRANSFER routines) will be described in detail. Besides it will be indicated how to prepare the tables concerning models grouping.

The entire organization of libraries and tables on disk will be described. Finally, as concerning the macro language for the specification of transfer of data among programs, the rules to set-up developments of macro instructions (correspondend to keywords) will be indicated.

CARONTE DATA POOL

This data pool is defined on disk storage and is devoted to contain input and output models of all programs and transfers executed in a Caronte run.
All the models are stored sequentially with pointers. These pointers are stored in fast memory by the EXECUTOR which provides to their management.

PRIVATE DATA POOL

If requested, the EXECUTOR can generate a private data pool for using in RESTART or CONTINUATION option selecting only the models which are output of programs and between output of transfers only these which are input of programs. These models are written sequentially on tape or disk depending on users wishing.

Let us see now the way the programs and TRANSFER routines actually operate. All the operations connected with writing the labels, or looking for a LABEL on the programs D.P. are done by a subro tine, called SEARCH, and being the same for all the programs and TRANSFER routines, while the input and output models are actually read and written by the programs and TRANSFER directly.

Let us now describe in same detail the operations performed by the SEARCH routine.

The arguments appearing in the calling sequence are N, IDEVEC, where IDEVEC is a 5-words vector, containing the LABEL (asteriks excluded).

The signification of N is the following : N = 0 causes the search **on the programs D.P. of** the LABEL equal to IDEVEC * ; the record immediately after the LABEL (i - the first record of the data) is then ready to be read.

The control will be next given back to the program or Transfer which will actually read the data.

N = -1 causes the search of the final record of the**programs D.P., ie.** STORAGE END. The record will be than cancelled by the writing on it of the LABEL-IDEVEC. The control will be given back to the program or transfer which will write on the programs D.P. the data.

---

* A convenient diagnostic will be printed out in the case of which the LABEL = IDEVEC will not been found.

N = -2 causes the immediate writing on the programs D.P. of the record STORAGE END.

In the case of RESTART or CONTINUATION option the Caronte system reads on the private data pool all the models requested for execution and writes then on the Caronte data pool noting conveniently the pointers.

THE CALLING SEQUENCE OF PROGRAMS AND TRANSFERS

In order to transmit to the programs and transfers a vector of 70 words containing informations among input/output models, the first instruction of each program or transfer must be CALL CIP (NT), where NT is a 70 words vector above mentioned and CIP is subroutine which permits the transmission of these data between NUCLEUS and programs or transfers.

The words of the NT vector used are the following.

NT (1)
NT (2) name of the program or transfer
NT (3) appearance number

Each program can treat 25 models : each model number is stored, if treated, in a word of NT vector starting from NT (21). The position of model number in the vector is determined by an index I.

NT (20+I) (I=1,25)
- 0 means that the model corresponding to I is not to be read as input model
- ≠ 0 means that the model corresponding to I is to be read as input model

NT (46+I) (I=1,25)
- 0 means that the model corresponding to I is not to be written as output model
- ≠ 0 means that the model corresponding to I is to be written as output model.

As for the transfer, the list of the words of the NT vector being used is the following :

NT (11) (used only in the case of loops). The system transmits (in NT (11), floating) to the TRANSFER, if it is the case, the value (TOL) of the maximum admittable difference between some or all the data of the output model of the transfer (being an input model to the next program), and the corresponding data of the same model, in the previous iteration. The Transfer transmits back (in NT (11), integer) to the system an indicator 0 when the above mentionned difference is greater than TOL, and 1 when it is less. Orequal in this last case the loop will be terminated.

NT (20 + I) I = 1,5	label which will be written in the programs D.P. by the TRANSFER.
NT (26 + I) I = 1,5 (32 + I) " " (38 + I) " " (44 + I) " " (50 + I) " " (56 + I) " "	label, some or all which will be looked for in the programs D.P. by the TRANSFER.

THE RULES TO BE FOLLOWED TO INSERT PROGRAMS IN THE CARONTE LIBRARY

It will be here examined in detail what it is do be done in order to modify a program in such a way it can be inserted into the Caronte program library.

1) First of all the program must contain like first instruction Call CIP (NT) as mentioned above and NT must be dimensioned 70 words and must terminate with a STOP 0 CALL EXIT statement.
   The following rules we suggest for the management of the input/output models of the programs.
   A vector IDEVEC of five words must be used as parameter in the calling sequence of the SEARCH subroutine in the following manner

IDEVEC (1) = NT (1) name of the program
" (2) = NT (2)
" (3) = NT (3) appearance number
" (4) = NT (20 + I) model number
" (5) = indicator = 1 for output model, = 0 for input

where I is an integer LESS or equal to 25 which represent the relative ordering number in the program of the model NT (I)

2) For each of the possible INPUT models an IF statement must be introduced into the program to verify whether the model is to be actually read from the programs D.P. or by normal input unit.

As it has been said, the input model NT (20 + I) will be examined in the IF statement.

When NT (20 + I) = 0 a READ statement from the normal input unit must follow.

When NT (20 + I) ≠ 0 the following statements must be inserted

```
IDEVEC (4) = NT (20 + I) = model number
N = 0
CALL SEARCH (IDEVEC,N),
READ (99)..........       (99 is the number assigned to the programs D.P.
```

3) For each of the possible OUTPUT models an IF statement must be introduced into the program to verify whether the model is to be actually stored on the programs D.P. or not.
(The writing on the normal output unit is to be done or not depending on the whish of the user). For the output model I the containt of NT (45 + I) will be examined in the IF statement.

When N (45 + I) ≠ 0 the following statements must be inserted

```
IDEVEC (4) = (number of the output model) = NT(45 + I)

N = -1
CALL SEARCH (IDEVEC, N)

WRITE (99) ............          (at least three words)
```

```
N = - 2
CALL SEARCH (IDEVEC, N)
```

The routine Search is to be added to the program.

The models can be combined and elaborated to form other models (TRANSFER ROUTINES). However it must be recalled that the number of the models to be combined (INPUT MODELS) by a Transfer routine must be up to six*. Just one model (OUTPUT MODEL) must be obtained by a TRANSFER routine. Generally it is understood that to one given output model it is possible to arrive through one TRANSFER only.

Exception to such a rule are

a) the possibility to combine the same input models indifferent way to obtain the same output model, identified by numbered RULES.

EXAMPLE

3 + 4 + 5 → 8 (RULE 1)
3 + 4 + 5 → 8 (RULE 2)

b) the possibility to "add" two input models indentified by the same number to get the same model.

EXAMPLE

3 + 5 → 9
9 + 9 → 9

The possible combinations of models, together with the pertinent TRANSFER routine name, must be organized in model tables, described later on.

---

* Two combinations are said to be different depending on the total number of models involved, the identification numbers of the models, or the order of the input models in the combinations

Parametrize the logical unit from which the program reads data set library.

EXAMPLE

If a program reads its data set library from logical unit 9 change 9 in M and insert after the instruction CALL CIP (NT), the instruction M = NT (6).

THE RULES TO BE FOLLOWED TO WRITE TRANSFER ROUTINES

As it has been pointed out a TRANSFER routine must be able to read some models from the programs D.P., to elaborate them and finally to write the resulting model on the programs D.P. In this sense the TRANSFER routines are to be considered as INTERFACES between programs of the sequence. As for the actual elaboration of data it is left to the user of Caronte to write the convenient statements depending on the particular application.
We will here describe the rules to be followed for reading and writing models on the programs D.P.

1. The name of the TRANSFER (up to 8 alphanumeric characters) will be specified at link-editor time (see below). The first executable instruction of the transfer must be CALL CIP (NT).

2. To each input model to be looked for on the programs data pool (up to six) the following statements must be written.

```
    DO 100 I = 1,5

    100 IDEVEC (I) = NT (26 + 6 x (K-1)+I)
```

K being the order number of the input model

EXAMPLE

```
    K = 1   2   3                   N = 0
        3 + 5 + 8  → 1              CALL SEARCH (IDEVEC, N)
                                    READ (99) ....
                                    ....
```

3. For the output model to be written on the programs D.P. the following statements must be written

```
DO  101 I = 1,5

101 IDEVEC (I) = NT (20 + I)

N = - 1

CALL SEARCH (IDEVEC, N)
WRITE (99) .....

N = - 2

CALL SEARCH (IDEVEC, N)
```

4. The routine SEARCH is to be added to the TRANSFER
5. To the TRANSFER, an assembler routine called CIP, is to be added, which relates the system to the TRANSFER (see below). The possibility to perform comparison between the output model just obtained during the elaboration, and the corresponding model obtained in a previous iteration the rules to be followed are here described. At the very beginning insert the statement

```
 EQUIVALENCE (NAPOLI, TOL)

 NAPOLI = NT (11)

NT (11) = 0
```

Remember now that the comparison will be actually executed whenever TOL is different from 0. Insert than the statements

```
DO 101 1 = 1,4
101 IDEVEC (I) = NT (20 + I)
IF (TOL. E Q. 0.) GO TO 2000

N = 0
CALL SEARCH (IDEVEC, N)
READ (99) -            { Read from the programs D.P. the
                       { model obtained in the previous
- ....                 { iteration

- { compare the prescribed data of the actual model and the
  { previous iteration model. Suppose that the maximum dif-
  { ference found is DIFF.

IF (DIFF. GT. TOL) GO TO 2000
NT (11) = 1
GO TO 2001

2000 CONTINUE
N = - 1
CALL SEARCH (IDEVEC, N)
WRITE (99)    ..         { write on the program D.P. the
                         { output model

N = - 2
CALL SEARCH (IDEVEC, N)

2001 RETURN
```

6. It is to be noted that the Transfer routines acting as interface between two programs for which the output of the first is exactly the input of the second one, must just change the LABEL preceeding the data. In such a case the user must not trouble himself in writing down such Transfer (SELF TRANSFER) since an unique routine to the purpose (STR) has been prepared and is available with the other programs (see below).

SETTING UP CARONTE LIBRARIES

The library of Caronte is a partitioned data set constitued by executable modules. Every module is identified by an up to 8 alphanumeric characters word. Every program or transfer is a module constitued by a main and subroutines having each a name up to six or eight alphanumeric characters, respectively in Fortran language or in Assembler. The main routine constitues the entry point of the module.

Here in follows the list of the Caronte modules :

1. NUCLEUS, defined by the name NUCLEUS, being the part of Caronte resident in fast memory ;

2. PROCESSOR, identified by the name PROCESSOR being the part of Caronte called in fast memory by the Nucleus before the execution of the programs ;

3. EXECUTOR, identified by the name EXECUTOR called in fast memory by the NUCLEUS, can reside in fast memory alternatively or with the programs.

The complete Caronte package (besides the reports) is constituted by all the above described modules (*) (in symbolic and / or binary language), plus the routine CIP (in symbolic or binary language) which must be inserted in any new program or transfer, plus the SEARCH routine, plus two separate program: TVC 0 RR (in symbolic or binary language) able to set up on disk all the correspondence model tables together with its input cards MACR0 (in symbolic or binary language) able to set up on disk all the macro developments together with its input cards, plus a sample problem of a Caronte run.

It will be now described in detail how to set up on disk a Caronte library, being a partitioned DATA SET. The following operations must be executed:

1. Reserve the convenient space on a disk unit;
2. (Compile) (**), linkedit and insert in the DATA SET the modules: NUCLEUS, PROCESSOR, EXECUTOR, PROGRAMS and TRANSFER; each identified by its pertinent name - (NUCLEUS, PROCESSOR , EXECUTOR)

(*) For the list of programs and transfer distributed in the Caronte package and related to nuclear reactor design problems, see Part IV and V.

(**) The input cards concern the program and transfer distributed in the Caronte package.

The job control cards apt to perform the work indicated in 1 and 2 are reported in annexe II.

SETTING UP OF THE CARONTE TABLES.

The way to use the program TVCORR able to prepare an sequential DATA SET with jointers containing the model tables, will be herein described. The input cards to the program TVCORR are divided in two logical groups :

a) cards describing the models treated by the programs and their relative position in side the program ;

b) card describing the combination of models. Every group a) or b) constitutes a "row" of the table.

We will here describe first groups a) and b) of input cards. In annexe II the exact format of the input cards is shown.

a) The cards are of three kinds :

CI card, with the module name of the considered program (up to 8 characters).

SI cards, one for each programs model.
This card contains the number 1 to 25 which describes the relative position of the model in the program, the number of the model, and an indication specifying whether the model is to be considered input program model, or output or both ;

ET cards, indicating the end of the table "row".

b) The cards are of three kinds :

CI cards, like in a) description.

SS card, containing the actual identification number of the output model, the number of the input models of a transfer (*), the actual identification number, of the input models eventually the rule of elaboration, the module name of the considered transfer (up to 8 characters) ;

ET card, indicating the end of the table "row".

A run of a TVCORR program can be related to just one of the above described options.

MACRO LANGUAGE

As it has pointed out in the part II, there is the possibility in Caronte system to use a macrolanguage vary flexible and mnemonic to specify in the control input the transfer of data among the programs.

More precisely the macro-instruction is of this type :

$C - \beta$ KEY WORD FROM $C_1 - \beta_1, C_2 - \beta_2, \ldots, C_n - \beta_n$

where C are programs and β are appearence number.

(*) It is to be noted that the self transfer must not be inserted into the tables.

The development of this instruction into the normal control input instruction of Caronte is of the following kind: first of all one macro instruction is converted in one or more normal micro instruction (each corresponding to a input model of the program $C-\beta$). For each of these micro instructions the input program model is obtained by one or more transfer programs (indicated, if necessary, by parenthesis), elaborating output models of the $C_1-\beta_1$ programs.

EXAMPLE 1

$C-\beta$ KEY WØRD FRØM $C_1-\beta_1$, $C_2-\beta_2$

can be developed into

$C-\beta$ (3) FRØM $C_1-\beta_1(1)$, $C_2-\beta_2(2)$

where the numbers between parenthesis are model numbers.

In such a case there must exists the interface program which elaborates the models 1 and 2 to generate model 3:

model 1 + model 2 → model 3

The model 3 is input model of the program $C-\beta$.

EXAMPLE 2

$C-\beta$ KEY WØRD FRØM $C_1-\beta_1$, $C_2-\beta_2$, $C_3-\beta_3$

can be developed into:

a) $C-\beta$ (1) FRØM ($C_1-\beta_1(3)$, $C_3-\beta_3(2)$), $C_2-\beta_2(5)$

b) $C-\beta$ (3) FRØM $C_3-\beta_3(1)$, $C_2-\beta_2(8)$

In such a case, in expansion a) the programs $C_1-\beta_1$ and $C_3-\beta_3$ are closed in parenthesis. This means that there must exists an interface program which elaborates models 3 and 2 to generate a model X and an interface program which elaborates model X and model 5 to generate a model 1 (input model for $C-\beta$ program). Expansion b) is described above in EXAMPLE 1.

The key words, being constitued by an up to 72 characters word including blancks, or commas) identifies in an unique way the development of the macro instruction.

Possible key words to be used in the macro language can be as an example

DIAMETER OF THE SPHERE, OR CYLINDER

MULTIPLICATION FACTOR OF ISPRA 1 REACTOR

As it has been said, a certain number of key words and corresponding developments has already been set up for the particular field of reactor design (see report V). However there exists the possibility for the programmer to set up new developments (corresponding to new keyword) and to insert them in Caronte.

All the KEY WORDS and corresponding developments constitue a data set on direct access device and his management is carried out by an utility program (input and operative instructions see appendix MACRO).

PROGRAMS DATAPOOL

This datapool is always defined on disks storage and is devoted to contain out put and input models of each program and transfer at each execution time.

If a program or transfer request input models, the EXECUTOR transfers input models from Caronte datapool to programs datapool, starting from the begining of the data pool before each program or transfer execution.

Programs and transfers write on this data pool their out put models after input models and EXECUTOR transfers these models from programs data pool to Caronte data pool.

The input and out put models of the programs or transfers are costituted by groups of records preceeded and identified by a label, consisting in two records each of three words being :

1. * * * *

2. }
3. } Name of program or transfer

4. Appearence number

5. Model number

6. Indicator o = input model, 1 = output model

The last record of the program datapool contains three words being STORAGE END.

The programs data pool must contain variable records not blodhed, maximum lenght of 800 bytes.

Annexe I

```
//         EXEC  PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//RESIDENT DD VOLUME=SER=EURSY0,DISP=SHR,UNIT=2301
//CTECNTR1 DD VOLUME=SER=EURSY6,DISP=SHR,UNIT=2314
//CTELIBET DD DSNAME=CTE.CODE.LIBET,UNIT=2314,                         C
//             SPACE=(CYL,(20,5,5)),DISP=(,KEEP,KEEP),                 C
//             VOLUME=(PRIVATE,SER=EURSYE)
//SYSIN  DD *
         CATLG  DSNAME=CTE.CODE.LIBET,VOL=2314=EURSYE
```

Annexe II

```
//INSERT   EXEC  PGM=IEWL,PARM='MAP'
//SYSPRINT DD SYSOUT=A
//SYSLIB  DD  DSNAME=SYS1.FORTLIB,DISP=SHR
//        DD  DSNAME=SYS1.SSPLIB,DISP=SHR
//        DD  DSNAME=SYS1.EURLIB,DISP=SHR
//SYSUT1 DD  UNIT=2314,SPACE=(CYL,(10)),VOLUME=SER=EURSYE
//PRVALB DD  DISP=SHR,DSNAME=CTE.CODE.LIBET
//SYSLMOD DD DISP=MOD,DSNAME=CTE.CODE.LIBET
//SOURCE  DD DSNAME=+LOADSET,DISP=(OLD,DELETE)
//SYSLIN  DD DDNAME=SYSIN
//SYSIN  DD *
        INCLUDE SOURCE
        ENTRY ********
        NAME  XXXXXXXX
/*
```

******** IS ENTRY POINT OF MODULE (maximum 8 caracters lenght)

XXXXXXXX IS NAME OF MODULE (" " ")

Part IV

THE APPLICATION TO THE FIELD OF NUCLEAR REACTOR DESIGN.

PART IV - The Application to the Field of Nuclear Reactor Design

INTRODUCTION

In this paper it will be described the particular application of the system Caronte to a group of programs selected in the field of nuclear reactor design.

We suppose the reader familiar with the possibilities offered by the system Caronte. In particular we suppose the knowledge of the parts I and II of the Caronte series, namely :

I - Caronte : The Euratom modular calculation system
II - Caronte : Users manual

In the following paragraphs the selected programs are listed and described, and for each of them the choice of the PROGRAM MODELS is discussed. Then it follows a description of the transfer routines and the macro language for the particular field of application. Finally a complete example will be examined concerning the K_{eff} and fluxes calculation of ISPRA-1 reactor.

Selected Programs

At the moment just a small group of 16 programs has been modified and adapted in order to be controlled by Caronte.

It is foreseen that such a group will be highly increased in the next future, following the wishes of the users. However the selected programs cover a rather large range of fields, and by means of them a complete multigroup calculation of K_{eff} and fluxes of an heterogeneous reactor can be performed.

Let's now list and briefly describe the programs adapted to the system.

GAM2

This is the IBM 360/65 version of the GAM-2 part of GGC-2 (1) (for IBM 7090) for the calculation of fast neutron spectra and associated multigroup cross sections.
As for the 7090 library tape, it has been converted to a 360/65 tape by the use of an utility program called LUCIA, which is capable through the use of convenient masks, to identify integers and floating point caracters, and which converts them properly.
It has to be noted that one decimal digit is cut off in the conversion, but the results of a same run are not modified much than some % and only in the case of resonance calculations.
The alphanumeric caracters are not identified by LUCIA, and were converted by a special program.

GATHER2

This is the IBM 360/65 version of the GATHER-2 part of GG2-2 (for IBM 7090) for the calculation of thermal neutron spectra and associated multigroup cross sections. The 360/65 version is exactly equivalent to the 7090 version. The original library tape were converted to 360/65 just as described for GAM2 ..

COMBINE

This is a quite small program having just the aim to permit to the user of Caronte to specify the characteristics of the resulting cross section library to be obtained properly mixing the two libraries respectively obtained in the fast and thermal region by the cross sections calculation programs (i.e. Gam2 , Gather2). In annexe I the input informations to be supplied to COMBINE are fully described.

DTF4

This is the 360/65 version of the original DTF4 (2) program, solving, by the method of discrete ordinates, the multigroup one dimensional Boltzmann transport equation. Many programs errors have been eliminated in respect to the original version. Moreover a library tape has been set up to be used by DTF4 containing the 16, 18, 24, 25, 6 groups cross section reported by Bell, Devaney, et al. (3). The dimensions for floating and integers data have been brought to A(50.000), IA(1000).

EQUIPS 3

This is 360/65 version of the program EQUIPOISE - 3 (4), a two dimensional, two group, neutron diffusion program for the IBM 7090 computer. In the 360 version two blanck cards at the end of the problem have to be inserted.

GAZE

This is the IBM 360 version of the original Fortran GAZE program(5)(xxx). No modifications were introduced in the conversion.

GAD

This is the IBM 360/65 version of the original Fortran 2 GAD program (6). No modifications were introduced in the conversion.

FIODOR

This is described in reference (7)

THERMITE

This is described in reference (8)

GATHET

This is described in reference (9)

P28

This is a small program which does not require any input and punches in output cross section cards in GAD format.

ANISN

A one dimension 1 discrete ordinates transport code with anisotropic scattering - ward W. Engle, Jr.

DOT

A txo dimension 1 discrete ordinates transport code.

CONDOR

A few-group two dimensional program for the evaluation of water reactor long term reactivity changes - Emilio Salina.

SQUIRREL

A one dimensional few-group diffusion - depletion code which includes the effects of local power and water density E. Salina.

TRITON

A multi-group diffusion - depletion program in three dimensions. A. Daneri, G. Maggioni, E. Salina.

Selected Programs Models

For each of the programs being selected, a number of Models has been chosen, each Model containing the smallest collection of logically related information that can be considered a coherent whole. It must be pointed out that a lot of arbitrarities is associated to such choice. Anyway is rather easy either to add new Program Models to the existing ones, or to completely redefine the entire group of Models.

As a general rule, the information which one wants read from the DATA POOL must be left blank in the corresponding positions of the input card deck. Whenever a complete card is correspondingly to be left blank, it can be omitted.

Let'us now examine in detail the selected program Models for each of the considered programs.

The list of the Models is reported in the following table :

GATHER2 PROGRAM

PROGRAM MODELS

Model	Nr.	1	Slowing down SOURCES	←
Model	Nr.	2	Thermal SELF SHIELDING factors per broad group and per isotope	↔
Model	Nr.	10	Thermal CROSS SECTIONS per broad group and per isotope	→
Model	Nr.	11	Number of BROAD GROUPS	→
Model	Nr.	25	List of ISOTOPES and CONCENTRATIONS	→
Model	Nr.	17	χ SPECTRUM per broad group	→
Model	Nr.	16	ENERGY and VELOCITY LIMITS per broad groups	→

} CARONTE
DATA POOL

DTF4 PROGRAM

PROGRAM MODELS

Model Nr. 14	Total number of BROAD GROUPS in the total energy range	←
Model Nr. 8	Selection specification concerning the ISOTOPES to be considered in the total range, and CARLSON CONSTANTS for the library specification	←
Model Nr. 19	χ SPECTRUM, total range	←
Model Nr. 18	ENERGY and VELOCITY LIMITS per broad group, total range	←
Model Nr. 9	MICROSCOPIC CROSS SECTIONS, total range, following CARLSON scheme	←
Model Nr. 12	MATERIAL SPECIFICATION in different zones	→
Model Nr. 13	VOLUMES and TOTAL FLUXES per zone and per broad group	→
Model Nr. 27	FLUXES per point and group as input to the program	←
Model Nr. 29	FLUXES per point and group as output of the program	→

CARONTE
DATA POOL

Note that the sequence of reading cross sections by DTF4 is the following:
1) isotopes from cards, 2) isotopes from library tape, 3) isotope from D.P.

EQUIPS3 PROGRAM

PROGRAM MODELS

Model Nr. 23 MACROSCOPIC CROSS SECTION library for composition 1 ←
Model Nr. 23 MACROSCOPIC CROSS SECTION library for composition 2 ←
. . . .
. . . .
Model Nr. 23 MACROSCOPIC CROSS SECTION library for composition n ←

} CARONTE DATA POOL

It is to be noted that when "normal" compositions and "control" regions are contemporaneously presents, one has to call 1, 2, 3...N, the normal compositions, N+1, N+2...M the control regions, and give the corresponding cards just for these last ones.

GAZE PROGRAM

PROGRAM MODELS

Model Nr. 26	MICROSCOPIC CROSS SECTIONS, total range, following GAZE scheme	←	CARONTE DATA POOL
Model Nr. 30	FLUXES per point and group as input to the program	←	
Model Nr. 31	FLUXES per point and group as output of the program	→	
Model Nr. 19	χ SPECTRUM, total range, region 1	←	
Model Nr. 19	. . . region 2	←	
.	. . . .	←	
.	. . . .	←	

When set 1 is taken from the D.P., note that the input data consisting of the volume ratios per isotope must be intended multiplied by the isotopes densities.

GAM2 PROGRAM

PROGRAM MODELS

Model Nr. 1	Slowing down SOURCES to thermal range programs	→	CARONTE DATA POOL
Model Nr. 15	Slowing down SOURCES PER ISOTOPE to thermal range programs	→	
Model Nr. 3	Fast and epithermal CROSS SECTIONS per broad group and per isotope	→	
Model Nr. 4	Number of BROAD GROUPS	→	
Model Nr. 24	List of ISOTOPES and CONCENTRATIONS	→	
Model Nr. 7	χ SPECTRUM per broad group	→	
Model Nr. 5	ENERGY and VELOCITY LIMITS per broad group	→	

COMBINE PROGRAM

PROGRAM MODEL

Model Nr. 8 Selection specifications concerning the ISOTOPES to be considered in the total range, and CARLSON CONSTANTS for the library specification } → CARONTE DATA POOL

FIODOR PROGRAM

PROGRAM MODEL

Model Nr. 26 MICROSCOPIC CROSS SECTIONS, total range, following GAZE scheme → CARONTE DATA POOL

THERMITE PROGRAM

PROGRAM MODELS

Model Nr. 14	Total number of BROAD GROUPS in the total energy range	←	CARONTE DATA POOL
Model Nr. 8	Selection specification concerning the ISOTOPES to be considered in the total range, and CARLSON CONSTANTS for the library specification	←	
Model Nr. 9	MICROSCOPIC CROSS SECTIONS, total range, following CARLSON scheme	←	

GATHET PROGRAM

PROGRAM MODELS

Model Nr. 15	Slowing down SOURCES PER ISOTOPE	←	CARONTE DATA POOL
Model Nr. 10	Thermal CROSS SECTIONS per broad group and per isotope	→	
Model Nr. 11	Number of BROAD GROUPS	→	
Model Nr. 25	List of ISOTOPES and CONCENTRATIONS	→	
Model Nr. 17	χ SPECTRUM per broad group	→	
Model Nr. 16	ENERGY and VELOCITY LIMITS per broad groups	→	

PROGRAM MODELS

Model Nr. 28 MICROSCOPIC CROSS SECTIONS, total range, following GAD scheme ← CARONTE DATA POOL

Note that the self shielding indicators, NSHIELD, are given in sequence (one per isotope) with FORMAT 18I4.

P 28 PROGRAM

PROGRAM MODELS

Model Nr. 28 MICROSCOPIC CROSS SECTIONS, total range, following GAD scheme ← CARONTE DATA POOL

CONDOR PROGRAM

PROGRAM MODELS

Model Nr. 32 MICROSCOPIC CROSS SECTIONS, total range, following CONDOR scheme ← CARONTE DATA POOL

SQUIRELL PROGRAM

PROGRAM MODELS

Model Nr. 32 MICROSCOPIC CROSS SECTIONS, total range, following CONDOR scheme ← CARONTE DATA POOL

ANISN PROGRAM

The subdivision of models of this program is the same of DTF4 program.

DOT PROGRAM

The subdivision of models of this program is the same of ANISM program.

SETS-MODELS correspondence

The scheme which follows clarify the correspondences among the models, and their relative position in the programs.

GAM2	MODELS
1	1
2	15
3	3
4	4
6	24
7	7
8	5

GATHER2	
1	1
2	2
3	10
4	11
6	25
7	17
8	16

COMBINE	
1	8

DTF4 - ANISN	- DOT
1	14
2	8
3	19
4	18
5	9
6	12
7	13
8	27
9	29

TRITON	
1	33

CONDOR-SQUIRELL	MODELS
1	32

EQUIPS3	
1	23
2	23
.	.
.	.
N	23

GAZE	
1	26
2	30
3	31
4	19
5	19
.	.
.	.
N	19

GAD	
1	28

FIODOR	
1	26

THERMITE	
1	14
2	8
4	9

GATHET	
1	15
3	10
4	11
6	25
7	17
8	16

In the following tables for each model it will be specified respectively the logical meaning and the record format.

MODELS

1 - Slowing down SOURCES to thermal range programs
4 - Number of BROAD GROUPS in fast and epithermal range
5 - ENERGY and VELOCITY LIMITS per broad group in fast and epithermal range
7 - χ SPECTRUM per broad group in fast and epithermal range
24 - List of ISOTOPES and CONCENTRATIONS, considered in the programs treating fast and epithermal range
3 - Fast and epithermal CROSS SECTIONS per broad group and per isotope

---

11 - Number of BROAD GROUPS in thermal range
2 - Thermal SELF SHIELDING factors per broad group and per isotope
16 - ENERGY and VELOCITY LIMITS per broad group in thermal range
17 - χ SPECTRUM per broad group in thermal range
25 - List of ISOTOPES and CONCENTRATIONS, considered in the programs treating thermal range
10 - Thermal CROSS SECTIONS per broad group and per isotope

---

14 - Total number of BROAD GROUPS in the total energy range
18 - ENERGY and VELOCITY LIMITS per broad group, total range
19 - χ SPECTRUM, total range
8 - Selection specifications concerning the ISOTOPES to be considered in the total range, and CARLSON CONSTANTS for the library specifications
6 - List of ISOTOPES and CONCENTRATIONS, considered in the total range
9 - MICROSCOPIC CROSS SECTIONS, total range, following CARLSON scheme
20 - MACROSCOPIC CROSS SECTIONS, total range, following CARLSON scheme
23 - Like-20-but containing DIFFUSION COEFFICIENT
26 - MICROSCOPIC CROSS SECTIONS, total range, following GAZE scheme
28 - MICROSCOPIC CROSS SECTIONS, total range, following GAD scheme

12 - MATERIAL SPECIFICATION in different zones

13 - VOLUMES and TOTAL FLUXES per zone and per broad group

27 - FLUXES per point and group as input to a program

29 - FLUXES per point and group as output of a program

30 - FLUXES per point and group as input to a program

31 - FLUXES per point and group as output of a program

32 - MICROSCOPIC CROSS SECTIONS, total range, following CONDOR scheme

33 - MICROSCOPIC CROSS SECTIONS, total range, following TRITON scheme

MODEL 1 Slowing down SOURCES to thermal range programs

RECORD = Number of fine groups of a thermal cross section program (i.e. Gather) (NF) ; P_0(fast 1, fast 2, .. fast NF) ; DUMMY

RECORD = Number of fine groups of a thermal cross section program (i.e. Gather) (NF) ; P_1(fast 1, fast 2, ... fast NF) ; DUMMY

MODEL 2 Thermal SELF SHIELDING factors per broad thermal group and per isotope

RECORD = Number of isotope considered in thermal range (NGATHER) ; total number of broad groups (NB) ; DUMMY

RECORD = I(1) ; I(2) ; ... I(NGATHER) ; DUMMY ; DUMMY

Note that I(i) =
1 indicates that there exists a S.S.F. vector for isotope i
0 indicates that there does not exist a S.S.F. vector for isotope i

For every selected isotope:

RECORD = S.S.F. (1) ; S.S.F. (2) ... S.S.F. (NB) ; DUMMY ; DUMMY

Note that group 1 corresponds to lowest energy

MODEL 3 FAST CROSS SECTIONS per broad group and per isotope

RECORD = Number of broad fast groups (NBF) ; number of isotopes considered in fast range

For every isotope:

RECORD = indicator
0 the n, 2n reaction is not present
1 the n, 2n reaction is present

RECORD = $P_0(1\to1)$; $P_0(1\to2)$; ... $P_0(1\to NBF)$; $P_0(1\to thermal)$; $P_0(2\to2)$; $P_0(2\to3)$,...... $P_0(2\to NBF)$; $P_0(2\to thermal)$, ...

RECORD = $P_1(1\to1)$; $P_1(1\to2)$,.... $P_1(1\to NBF)$; $P_1(1\to thermal)$; $P_1(2\to2)$; $P_1(2\to3)$,...... $P_1(2\to NBF)$; $P_1(2\to thermal)$,...

RECORD = n,2n($1\to1$); n,2n($1\to2$),... n,2n($1\to NBF$); n,2n($1\to thermal$); n,2n($2\to2$); n,2n($2\to3$),... n,2n($2\to NBF$), n,2n($2\to thermal$),..

RECORD = total($1\to1$); total($1\to2$),... total($1\to NBF$); total($1\to thermal$) total($2\to2$); total($2\to3$),... total($2\to NBF$); total($2\to thermal$)

RECORD = number of considered reactions ; number indicating the total vector length ; twelve words for the type of reaction; $\sigma(1)$; $\sigma(2)$,... $\sigma(NBF)$

Note that the type of considered reactions may be :

ABSORPTION; SCATTER; TRANSPORT; TOTAL; NU; FISSION; NU*FISSION; CAPTURE; OUT-SCATTER; N.P; N, GAMMA; N, ALFA

MODEL 4 Number of FAST BROAD GROUPS

RECORD = Number of fast broad group (NBF); DUMMY; DUMMY

MODEL 5 ENERGY and VELOCITY LIMITS per broad FAST group

RECORD = Number of broad fast groups (NBF); $v(1)$; $V(2)$... V (NBF); E(1); E(2)......E(NBF)

Note that v is given in m/sec and E is given in eV.

MODEL 6 List of ISOTOPES and CONCENTRATIONS, considered in the total range

RECORD = number of isotopes in the total range (NFINTO); CONC (1); CONC(2)..... CONC(NFINTO)

Note that the isotopes for which there exist anisotropy are conted twice.

Note that the concentration must be intended as density * volume fraction.

MODEL 7 SPECTRUM per broad FAST group

RECORD = Number of broad fast groups (NBF); $\chi(1)$; $\chi(2)$... χ(NBF)

MODEL 8 Selection specification concerning the ISOTOPES to be considered in the total range, and CARLSON CONSTANTS for the library specifications.

RECORD = Number of isotopes considered in the fast region (NGAM); IV(1);IV(2) ... IV(NGAM), Number of isotopes considered in the thermal region (NGATHER) ; KV(1); KV(2).....KV(NGATHER),(*) Number of selected isotopes in the total range, considering that an isotope with anisotropy is conted twice (NFINTO); three words containing the three CARLSON constants.

Note that IV, KV
- 0 not selected for total range
- 1,2,3 selected and associated in order

Note that LV
- 0 isotope without anisotropy
- 1 isotope with anisotropy

(*) Number of selected isotopes in the total range (NVERO): LV(1); LV(2);....; LV(NVER0).

for each isotope: RECORD = VETT(1); VETT(2).... VETT(20), NMØD, MATRIX

Note that VETT(1), VETT(20) are twenty alphanumerical words for the isotope identification.

NMØD
0 a thermal transfer matrix is not considered
1 a thermal transfer matrix is considered

MATRIX
0 a fast transfer matrix is considered
1 a fast transfer matrix is not considered

MODEL 9 MICROSCOPIC CROSS SECTION, total range, following CARLSON scheme

RECORD = Number of istopes in total range, where isotopes with anisotropy are counted twice (NFINTO); Number of groups in total range (NB) ; three words being the three Carlson constants.

To be repeat NB NFINTO times: RECORD = (σ_f), σ_a, $\nu\sigma_f$, σ , σ_{up}...., σ_{gg}, σ_{down}.....

Note that the presence or not of σ_f depends from the value of the first Carlson constant (3 or 4).

Note that the length of the record is equal to the third Carlson constant.

For each isotope: RECORD = VETT(1), VETT(2)....VETT(20)

Note that VETT(1), VETT(20) are twenty alphanumerical words for the isotope identification.

MODEL 10 THERMAL CROSS SECTIONS per broad group and per isotope

RECORD = Number of broad thermal groups (NBT) ; Number of isotopes considered in the thermal range ; DUMMY

For each isotope:

RECORD = Number of mass (or other identification) of the isotope (NMASS); NFIS; NKERN; $\sigma_a(1)$;$\sigma_{sc}(1)$; $\sigma_{tot}(1)$; $\sigma_{trold}(1)$; $\sigma_a(2)$; $\sigma_{sc}(2)$; $\sigma_{tot}(2)$; $\sigma_{trold}(2)$....

RECORD = $\sigma_f(1)$; $\nu\sigma_f(1)$; $\sigma_c(1)$; $\sigma_f(2)$; $\nu\sigma_f(2)$; $\sigma_c(2)$.....
(IF NFIS=1)

RECORD = $P_{oup}(1)$; $P_{oup}(2)$....$P_{1up}(1)$; $P_{1up}(2)$... $\sigma_{out}(1)$;$\sigma_{out}(2)$..
(IF NKERN=1) $P_o(1 \rightarrow 1)$; $P_1(1 \rightarrow 1)$; $P_o(2 \rightarrow 1)$...$P_1(2 \rightarrow 1)$...$P_o(1 \rightarrow 2)$; $P_1(1 \rightarrow 2)$; $P_o(2 \rightarrow 2)$; $P_1(2 \rightarrow 2)$........

MODEL 11 Number of THERMAL BROAD GROUPS

RECORD = Number of thermal broad group (NBT); DUMMY; DUMMY

MODEL 12 MATERIAL SPECIFICATION in different zones

RECORD = MS;MT;IZM;MAT(1);MAT(2).....MAT(IZM)

RECORD (IF MS≠0) = MIX1(1);MIX1(2)...MIX1(MS);MIX2(1);MIX2(2)...... MIX2(MS);AMIX3(1)

Note that MT is the number of materials including mixtures, IZM is the number of zones, MAT(i) is the material in region i, where a minus sign indicates anisotropy. As for MIX1, MIX2, AMIX3 an example will classify their meaning. Suppose that material 5 is made up of elements 3 and 4 with concentrations C1 and C2, and material 6 is made up of element 2 with concentrations C_3, then it will be MS = 7 and

MIX1(1)	MIX1(2)	MIX1(3)	MIX1(4)	MIX1(5)	MIX1(6)	MIX1(7)
5	5	5	5	6	6	6
MIX2(1)	MIX2(2)	MIX2(3)	MIX2(4)	MIX2(5)	MIX2(6)	MIX2(7)
0	3	4	0	0	2	0
AMIX3(1)	AMIX3(2)	AMIX3(3)	AMIX3(4)	AMIX3(5)	AMIX3(6)	AMIX3(7)
0	C_1	C_2	1	0	C_3	1

In the case of anisotropy suppose material 5 is made up of elements 3 and 5, where 3 is the isotropic part of an element, whose anisotropic part is 4. The material 6, being the anisotropic part of material 5, will be made up of element 4.

MIX1(1)	MIX1(2)	MIX1(3)	MIX1(4)	MIX1(5)	MIX1(6)	MIX1(7)
5	5	5	5	6	6	6
MIX2(1)	MIX2(2)	MIX2(3)	MIX2(4)	MIX2(5)	MIX2(6)	MIX2(7)
0	3	5	0	0	4	0
AMIX3(1)	AMIX3(2)	AMIX3(3)	AMIX3(4)	AMIX3(5)	AMIX3(6)	AMIX3(7)
0	C_1	C_2	1	0	C_1	1

MODEL 13 VOLUMES and TOTAL FLUXES per zone and per broad group

RECORD = Number of total broad groups (NB); number of zones (IZM); VOLUME(1);VOLUME(2)..........VOLUME(IZM)

For each group { RECORD = FLUX(1);FLUX(2)........FLUX(IZM);DUMMY,DUMMY

MODEL 14 Total number of BROAD GROUPS in the total energy range

RECORD = Total number of broad groups (NB); DUMMY; DUMMY

MODEL 15

MODEL 16 VELOCITY and ENERGY LIMITS per broad group in THERMAL range

RECORD = Number of broad thermal groups (NBT);V(1);V(2)... V(NBT);E(1);E(2)......E(NBT)

Note Note that V is given in m/sec and E is given in eV.

MODEL 17 χ SPECTRUM per broad group in THERMAL range

RECORD = Number of broad thermal groups (NBT); $\chi(1)$; $\chi(2)$.... χ (NBT).

MODEL 18 VELOCITY and ENERGY LIMITS per broad group, total range

RECORD = Total number of broad groups (NB);V(1);V(2)...V(NB); E(1);E(2)....E(NB)

Note that V is given in m/sec and E is given in eV.

MODEL 19 χ SPECTRUM per broad group, total range

RECORD = Number of total broad groups (NB); $\chi(1)$; $\chi(2)$... $\chi(NB)$

MODEL 20 MACROSCOPIC CROSS SECTIONS, total range, following CARLSON scheme

RECORD = 1; Number of groups in total range (NB); three words being the three Carlson constants

NB times { RECORD = (σ_f); σ_a; $\nu\sigma_f$; σ_t; σ_{up}.....; σ_{gg}; σ_{down}...

Note that the presence or not of σ_f depends from the value of the first Carlson constant (3 or 4).

Note that the lenght of the record is equal to the third Carlson constant.

MODEL 21

MODEL 22

MODEL 23 MACROSCOPIC CROSS SECTIONS, total range, following CARLSON scheme, containing DIFFUSION COEFFICIENTS

RECORD = 1; Number of groups in total range (NB); three words being the three Carlson constants

NB times

RECORD = D; (σ_f); σ_a; $\nu\sigma_f$; σ_t; σ_{up}.....σ_{gg}....σ_{down}

Note that the presence or not of σ_f depends from the value of the first Carlson constant (3 or 4)

Note that the lenght of the record is equal to the third Carlson constant +1.

MODEL 24 List of ISOTOPES and CONCENTRATIONS, considered in the programs treating FAST range

RECORD = Number of isotopes (NGAM);CONC(1);CONC(2)...CONC(NGAM)

Note that the concentration must be intended density * volume fractio.

MODEL 25 List of ISOTOPES and CONCENTRATIONS, considered in the programs treating THERMAL range

RECORD = Number of isotopes (NGATHER);CONC(1);CONC(2)..CONC(NGATHER)

Note that the concentrations must be intended density * volume fractio.

MODEL 26 MICROSCOPIC CROSS SECTIONS, total range, following GAZE scheme

RECORD = 1 WORD = TAPE
RECORD = 2 WORD = GAZE, PROBLEM TITLE

RECORD = Number of isotopes (NFINTO) ; Number of total broad groups (NB); DUMMY

For each isotope

RECORD = Number of words in **this**record = **2** (begin with 1001) isotope ordering number; **(begin with 1001); 1.**

RECORD = Number of words in the record

$\sigma_c(1);\sigma_c(2)...\sigma_f(1);\sigma_f(2)...\sigma_{tr}(1),\sigma_{tr}(2)....$

* $\nu(1);\nu(2).....\sigma(\overset{1}{2}\to1),\sigma(2\to1),\sigma(3\to1)...(1\to2)(2\to2)(3\to2)...$

For each isotope

<u>RECORD</u> = VETT(1);VETT(2).....VETT(20)

Note that VETT(1);VETT(20) are twenty alphanumeric words for the isotope identification.

<u>RECORD</u> = 2, WORD=GAZE, WORD = EØF

<u>MODEL 27</u> <u>FLUXES</u> per point and group as input to a program

<u>RECORD</u> = Number of total broad groups (NB); Number of mesh (NP);NP NB; ρ (1,1); ρ(2,1);... ρ(NP,1); ρ(1,2); (2,2)....... ρ(NP,2);.... ρ(1,NB); ρ(2,NB)...ρ (NP,NB)

<u>MODEL 28</u> <u>MICROSCOPIC CROSS SECTIONS</u>, total range, following <u>GAD</u> scheme

<u>RECORD</u> = Number of isotopes (NFINTO); Number of total broad groups (NB) ; Number of fast broad groups (NBF) ; Number of thermal broad groups (NBT)

For each isotope

<u>RECORD</u> = VETT(1);VETT(2)....VETT(20), NMOD, MATRIX

Note that VETT(1), VETT(20) are twenty alphanumerical words for the isotope identification.

NMOD
- 0 a thermal transfer matrix is not considered
- 1 a thermal transfer matrix is considered

MATRIX
- 0 a fast transfer matrix is considered
- 1 a fast transfer matrix is not considered

<u>RECORD</u> = $\nu\sigma_f(1);\sigma_{tr}(1);\sigma_a(1);\sigma(1\rightarrow2);\nu(1);\nu\sigma_f(2);\sigma_{tr}(2)\sigma_a(2);$
$\sigma(2\rightarrow3)$........$\nu\sigma_f$(NB);σ_a(NB);DUMMY,ν(NB)

IF MATRIX = 0

RECORD σ (1→2); σ(1→3)...... σ(1→ NBF)
RECORD σ (2→3); σ(2→4)...... σ(2→ NBF)
. . . .
RECORD σ (NBF-1→ NBF)

RECORD σ(1→NBF+1); σ(1→NBF+2)... σ(1→NBF+NBT)
RECORD σ(2→NBF+1); σ(2→NBF+2)... σ(2→NBF+NBT)
. . . .
RECORD σ(NBF,NBT →NBF+1); σ(NBF+NBT →NBF+2); ...
σ(NBF+NBT → NBF+NBT)

MODEL 29 FLUXES per point and group as output of a program

RECORD = Number of total broad groups (NB); Number of mesh **interval** (NP)NPxNB;$\rho(1,1)$;$\rho(2.1)$...$\rho(NP,1)$; $\rho(1,2)$;$\rho(2,2)$....$\rho(NP,2)$....$\rho(1,NB)$;$\rho(2,NB)$........$\rho(NP,NB)$

MODEL 30 FLUXES per point and group as input to a program

RECORD = Number of total broad groups (NB); Number of mesh points (NP);$\rho(1,1)$;$\rho(1,2)$;...$\rho(1,NB)$; $\rho(2,1)$; $\rho(2,2)$... $\rho(2,NB)$;... $\rho(NP,NB)$.

MODEL 31 FLUXES per point and group as output of a program

RECORD = Number of total broad groups (NB); Number of mesh points (NP); $\rho(1,1)$; $\rho(1,2)$;... $\rho(1,NB)$; $\rho(2,1)$; $\rho(2,2)$... $\rho(2,NB)$;... $\rho(NP, NB)$.

MODEL 32 MICROSCOPIC CROSS SECTIONS (FULL RANGE)

For each isotope non fissile

RECORD = Name, σtr (1), σabs (1), σrem (1);
σtr (2), σabs (2), σrem (2);
. . . .
σtr (NG),σ abs (NG), σrem(NG)

For each isotope fissile

RECORD = Name, σtr (1), σabs (1), σrem (1),σ f (1),$\nu\sigma$ f (1),
. . . .
σtr (NG), σabs (NG), σrem (NG), σf (NG), $\nu\sigma$f (NG)

Where name is the isotope name (2 words)
NG is number of groups

TRANSFER ROUTINES

The transfer routines which have been prepared are listed in the following:

TRANSFER PROGRAMS

TRANSFER PROGRAMS

13+12+8 → 2
4+11 → 14
5+16 → 18
7+17 → 19
24+25+8 → 6
3+10+8+5+16 → 9
9+9 → 9
9+6 → 20
20(RULE-1,RULE-2) → 23
9 → 26
8+9 → 28
29 → 27
31 → 30

The operations performed by these transfer routines are generally self-explenatory.
The RULES-1 and 2 appering in the transfer 20 → 23 are related to the particular way to calculate the diffusion coefficient D starting from Σ_a, Σ_{tr}

LAW 1 $$D = \frac{1}{3\,\Sigma_{tr}}$$

LAW 2 $$D == \frac{1}{3\,\Sigma_{tr}\,\left(1 - \frac{4}{5}\,\frac{\Sigma_a}{\Sigma_{tr}}\right)}$$

The transfer programs 29 → 27 and 31 → 30 are in fact "no operation" transfers. Some comments are required for the TRANSFER 13+12+8 → 2. In this transfer a part is contained by which the self-shieldings just calculated are compared, if an explicit indication to do so is given by CARONTE to the self-shieldings calculated in a previous iteration of a loop.
The memorization on the Caronte D.P. of the new calculated self-shielding is performed only when the difference is greater than a prescribed

value (supplied by the Caronte user). In any case a message is sent back to Caronte stating whether loop exit condition was satisfied or not.
Such TEST parts of the TRANSFER routines are rather standard and can be inserted whenever desired following the wishes of the user.

Note that in all the transfer routines the maximum permitted numbers of groups and isotopes are respectively 100 and 50.

MACRO LANGUAGE

The macro language instructions related to the considered field of applications and already inserted into the system are listed in the following:

MACRO-LANGUAGE INSTRUCTIONS

	$C-\alpha$	SOURCE from	$C_1-\beta_1$
	↓		↓
Example	GATHER2		GAM2

	$C-\alpha$	SELF-SHIELDING from	$C_1-\beta_1$,	$C_2-\beta_2$
	↓		↓	↓
Example	GATHER2		DTF4	COMBINE

optional (Not accepted for Thermite)

	$C-\alpha$	ONE BLOCK TRANSPORT MICROSCOPIC LIBRARY, CHI, V from		
		$C_1-\beta_1$,	$C_2-\beta_2$,	$C_3-\beta_3$
	↓	↓	↓	↓
Example	DTF4 THERMITE ANISN	GAM2	GATHER2 GATHET	COMBINE

optinal (Not accepted for Thermite)

	$C-\alpha$	TWO BLOCKS TRANSPORT MICROSCOPIC LIBRARY, CHI, V from					
		$C_1-\beta_1$,	$C_2-\beta_2$,	$C_3-\beta_3$,	$C_4-\beta_4$,	$C_5-\beta_5$,	$C_6-\beta_6$
	↓	↓	↓	↓	↓	↓	↓
Example	DTF4 THERMITE ANISN	GAM2-1	GATHER2-1 GATHET-1	COMBINE-1	GAM2-2	GATHER2-2 GATHET-2	COMBINE-2

	$C-\alpha$	ONE BLOCK DIFFUSIONE MICROSCOPIC LIBRARY from		
		$C_1-\beta_1$,	$C_2-\beta_2$,	$C_3-\beta_3$
	↓	↓	↓	↓
Example	GAZE	GAM2	GATHER2 GATHET	COMBINE

	$C-\alpha$	TWO BLOCKS DIFFUSION MICROSCOPIC LIBRARY from					
		$C_1-\beta_1$,	$C_2-\beta_2$,	$C_3-\beta_3$,	$C_4-\beta_4$,	$C_5-\beta_5$,	$C_6-\beta_6$
	↓	↓	↓	↓	↓	↓	↓
Example	GAZE	GAM2-1	GATHER2-1 GATHET-1	COMBINE-1	GAM2-2	GATHER2-2 GATHET-2	COMBINE-2

	C- α	ONE BLOCK BURN-UP MICROSCOPIC LIBRARY from					
	↓	$C_1-\beta_1$,	$C_2-\beta_2$,	$C_3-\beta_3$			
		↓	↓	↓			
Example	GAD,P28 CONDOR	GAM2	GATHER2	COMBINE			
	C- α	TWO BLOCKS BURN-UP MICROSCOPIC LIBRARY from					
	↓	$C_1-\beta_1$,	$C_2-\beta_2$,	$C_3-\beta_3$,	$C_4-\beta_4$,	$C_5-\beta_5$,	$C_6-\beta_6$
		↓	↓	↓	↓	↓	↓
Example	GAD, P 28 CONDOR	GAM2-1	GATHER2-1 GATHET-1	COMBINE-1	GAM2-2	GATHER2-2 GATHET-2	COMBINE-2
	C- α	MACROSCOPIC LIBRARY COMPOSITIONS (1,2,3....) from					
	↓	$C_1-\beta_1$,	$C_2-\beta_2$,,	$C_3-\beta_3$			
		↓	↓	↓			
Example	EQUIPS3	GAM2	GATHER2 GATHET	COMBINE			

	C-α	DIFFUSION MICROSCOPIC LIBRARY from	$C_1 - \beta_1$
	↓		↓
Example	GAZE		FIODOR
	C-α	<u>SOURCE PER ISOTOPE</u> from	$C_1 - \beta_1$
	↓		↓
Example	GATHET		GAM2

The developments of the macro instructions in normal input control instructions are given below:

C-α TWO BLOCKS TRANSPORT MICROSCOPIC LIBRARY, CHI, V
FRØM $C_1-\beta_1, C_2-\beta_2, C_3-\beta_3, C_4-\beta_4, C_5-\beta_5, C_6-\beta_6$
C-α (14) FRØM $C_1-\beta_1(4), C_2-\beta_2(11)$
C-α (8) FRØM $C_3-\beta_3(8)$
C-α (19) FRØM $C_1-\beta_1(7), C_2-\beta_2(17)$
C-α (18) FRØM $C_1-\beta_1(5), C_2-\beta_2(16)$
C-α (9) FRØM $[C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8), C_1-\beta_1(5),$
$C_2-\beta_2(16)], [C_4-\beta_4(3), C_5-\beta_5(10), C_6-\beta_6(8),$
$C_4-\beta_4(5), C_5-\beta_5(16)]$

It is valid if C has models 14,8,19,18,9. The transfers 4+11 → 14, 8 → 8, 7+17 → 19 5+16 → 18, 3+10+8+5+16 → 9, again 3+10+8+5+16 → 9, 9+9 → 9 will be used. An example of C is DTF4,

C-α ONE BLOCK TRANSPORT MICROSCOPIC LIBRARY FRØM $C_1-\beta_1, C_2-\beta_2,$
$C_3-\beta_3$
C-α (14) FRØM $C_1-\beta_1(4), C_2-\beta_2(11)$
C-α (8) FRØM $C_3-\beta_3(8)$
C-α (9) FRØM $C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8), C_1-\beta_1(5), C_2-\beta_2(16)$

It is valid if C has models 14,8,9. The transfers 4+11 → 14, 8 → 8, 3+10+8+5+16 9 will be used. An example of C is DTF4, or THERMITE.

C-α TWO BLOCKS TRANSPORT MICROSCOPIC LIBRARY FRØM $C_1-\beta_1, C_2-\beta_2,$
$C_3-\beta_3, C_4-\beta_4, C_5-\beta_5, C_6-\beta_6$
C-α (14) FRØM $C_1-\beta_1(4), C_2-\beta_2(11)$
C-α (8) FRØM $C_3-\beta_3(8)$
C-α (9) FRØM $\{[C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8), C_1-\beta_1(5),$
$C_2-\beta_2(16)], [C_4-\beta_4(3), C_5-\beta_5(10), C_6-\beta_6(8),$
$C_4-\beta_4(5), C_5-\beta_5(16)]\}$

It is valid if C has models 14;8,9. The transfers 4+11 → 14, 8 → 8, 3+10+8+5+16 → 9, again 3+10+8+5+16 → 9, 9+9 → 9 will be used. An example of C is DTF4, or THERMITE.

MACRO-LANGUAGE DEVELOPMENTS

C-2(3)=Model 3 of program C-2
C-2(5)=Model 5 pf program C-2.

The following relations are valid when just <u>one</u> model of C_1 can be obtained, starting from $C_1-\beta_1$ (N)

C-α <u>SOURCE</u> FRØM $C_1-\beta_1$
C-α(1) FRØM $C_1-\beta_{1(1)}$

It is valid if C has model 1. An example of C is GATHER2.

C-α SELF-SHIELDING FRØM $C_1-\beta_1, C_2-\beta_2$
C-α (2) FRØM $C_1-\beta_1(13)$, $C_1-\beta_1(12)$, $C_2-\beta_2(8)$

It is valid if C has model 2. The TR 13+12+8 → 2 will be used. An example of C is GATHER2.

C-α <u>ONE BLOCK TRANSPORT MICROSCOPIC LIBRARY, CHI, V</u>
FRØM $C_1-\beta_1, C_2-\beta_2, C_3-\beta_3$
C-α(14) FRØM $C_1-\beta_1(4), C_2-\beta_2(11)$
C-α(8) FRØM $C_3-\beta_3(8)$
C-α(19) FRØM $C_1-\beta_1(7), C_2-\beta_2(17)$
C-α(18) FRØM $C_1-\beta_1(5), C_2-\beta_2(16)$
C-α(9) FRØM $C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8), C_1-\beta_1(5)$,
$C_2-\beta_2(16)$

It is valid if C has models 14,8,19,18,9. The transfers 4+11 → 14, 8→8, 7+17 → 19, 5+16→18, 3+10+8+5+16 → 9 will be used. A example of C is DTF4.

C-α TWO BLOCKS BURNUP MICROSCOPIC LIBRARY

FRØM $C_1-\beta_1,C_2-\beta_2,C_3-\beta_3,C_4-\beta_4,C_5-\beta_5,C_6-\beta_6$

$$\begin{aligned} C\text{-}\alpha\ (28)\ \text{FRØM}\ & \left[C_3-\beta_3(8),C_6-\beta_6(8)\right], \Big\{ \Big[C_1-\beta_1(3),C_2-\beta_2(10),\\ & C_3-\beta_3(8),C_1-\beta_1(5),C_2-\beta_2(16)\Big], \Big[C_4-\beta_4(3),\\ & C_5-\beta_5(10),C_6-\beta_6(8),C_4-\beta_4(5),C_5-\beta_5(16)\Big]\Big\},\\ & C_1-\beta_1(5),C_2-\beta_2(16) \end{aligned}$$

It is valid if C has a model 28. The transfers 3+10+8+5+16→ 9, again 3+10+8+5+16→ 9, 9+9→ 9, 8+8→ 8, 8+9+5+16→ 28 will be used. An example of C is GAD, or P 28.

C-α DIFFUSION MICROSCOPIC LIBRARY
FRØM $C_1-\beta_1$

C-α (26) FRØM $C_1-\beta_1$ (26)

It is valid if C has a model 26. An example of C is GAZE.

C-α SOURCE PER ISOTOPE FRØM $C_1-\beta_1$

C-α (15) FRØM $C_1-\beta_1$ (15)

It is valid if C has a model 15. An example of C is GATHET.

C-α MACROSCOPIC LIBRARY COMPOSITIONS J(1,2,3..)
FRØM $C_1-\beta_1, C_2-\beta_2, C_3-\beta_3$
C-α (J(23,23,23..)) FRØM $\Big[\Big[C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8),$
$C_1-\beta_1(5), C_2-\beta_2(16)\Big], \Big[C_1-\beta_1(24),$
$C_2-\beta_2(25), C_3-\beta_3(8)\Big]\Big\}$

It is valid if C has models J (23, 23, 23 ..).
The transfers 3+10+8+5+16 → 9, 24+25+8 → 6, 9+6 → 20, 20 → 23 will be used. An example of C is EQUIPS3.

C-α ONE BLOCK DIFFUSION MICROSCOPIC LIBRARY
FRØM $C_1-\beta_1, C_2-\beta_2, C_3-\beta_3$
C-α (26) FRØM $\Big[C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8), C_1-\beta_1(5),$
$C_2-\beta_2(16)\Big]$

It is valid if C has model 26. The transfers 3+10+8+5+16 → 9, 9 → 25 will be used. An example of C is GAZE.

C-α TWO BLOCKS DIFFUSION MICROSCOPIC LIBRARY
FRØM $C_1-\beta_1, C_2-\beta_2, C_3-\beta_3, C_4-\beta_4, C_5-\beta_5, C_6-\beta_6$
C-α (26) FRØM $\Big[\Big[C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8), C_1-\beta_1(5),$
$C_2-\beta_2(16)\Big]\Big[C_4-\beta_4(3), C_5-\beta_5(10), C_6-\beta_6(8),$
$C_4-\beta_4(5), C_5-\beta_5(16)\Big]\Big]$

It is valid if C has model 26.
The transfers 3+10+8+5+16 → 9; again 3+10+8+5+16 → 9, 9+9 → 9, 9 → 26 will be used. An example of C is GAZE.

C-α ONE BLOCK BURN-UP MICROSCOPIC LIBRARY
FRØM $C_1-\beta_1, C_2-\beta_2, C_3-\beta_3$
C-α (28) FRØM $C_3-\beta_3(8), \Big[C_1-\beta_1(3), C_2-\beta_2(10), C_3-\beta_3(8),$
$C_1-\beta_1(5), C_2-\beta_2(16)\Big], C_1-\beta_1(5), C_2-\beta_2(16)$

It is valid if C has model 28.
The transfers 3+10+8+5+16 → 9, 8+9+5+16 → 28 will be used. An example of C is GAD, or P 28.

We note that quite easily it is possible to insert new macro instructions, or to modify existing ones. However in the present Caronte version the maximum number of allowable macro instructions is **not limited.**

A complete EXAMPLE OF A REACTOR CALCULATION

To illustrate the actual utilisation of the system Caronte as applied to the programs herein considered, an example has been chosen in which nearly all the possibility of the system Caronte are utilized, being the multiplication factor and fluxes calculation of the ISPRA-1 reactor.
A simplified scheme of the reactor is shown in fig. N. 1
To the purpose of determining the K_{eff} and fluxes of the reactor the following calculation scheme has been chosen:

1) Fast cross sections (one broad group) calculation for composition 1 by means of the program GAM2
2) The same for composition 2
3) The same for composition 2
4) Thermal cross sections (one broad group) calculation for composition 1, by means of the program GATHER2, which utilizes source terms coming from the corresponding GAM2
5) The same for composition 2
6) The same for composition 3. An initial guess is given in the self shielding factors input to GATHER2
7) Combine specifications for the cross sections libraries of composition 1
8) The same for composition 2
9) The same for composition 3
10) Calculation of the cell fluxes by means of DTF4 program, utilizing the two groups cross sections previously calculated for the composition 3
11) Thermal cross section (one broad group) calculation for composition 3 by means of GATHER2, the self shielding factors input to GATHER2 are derived by the previously executed transport calculation

11) Points 10 and 11 have to be repeated until self shielding factors are stabilized.
12) Two group calculation of K_{eff} and fluxes by the EQUIPS3 program, utilizing the calculated cross section library.

The job control cards for the present work are reported in annexe II. The input control instructions corresponding to the described scheme of computation are reported in annexe III.
The actual executed sequence of programs and transfer is given in annexe IV.
Finally the macro language which can be used at the place of the normal input control instructions, and which greatly reduces the user labor, is presented in annexe V.

The complete scheme of the deck is given in annexe VI.

BIBLIOGRAPHY

[1] GGG-II - A Program for Using the GAM-II and GATHER-II Spectrum Codes in Preparing Multigroup Cross-Section Input on Punched Cards for the GAZE, GAZED, DSN, GAPLSN, 2DXY, TDC, GAMBLE, FEVER, and GAD Codes. - C.V. Smith, H.A. Vieweg - GA-4436

[2] DTF-1V, FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering, K.D. Lathrop - LA-3373

[3] "Los Alamos group Average Cross-Sections" - Bell, Devaney, Hansen, Mills, Roach - LAMS-2941

[4] EQUIPOISE 3: A Two-Dimensional, Two-Group, Neutron Diffusion Code for the IBM-7090 Computer - T.B. Folwer, M.L. Tobias - ORNL-3199

[5] GAZE-2: A One-Dimensional, Multigroup, Neutron Diffusion Theory Code for the IBM-7090 - S.R. Lenihan - GA-3152

[6] GAD: An Infinite Medium Multigroup Burnup Code for Fuel Cycle Analysis F.W. Todt - GA-4150

[7] IGOR e FIODOR: Utilizzo d'un atlante di sezioni d'urto a 26 gruppi che tiene conto dei coefficienti di autoschermaggio delle risonanze a varie temperature - P. Azzoni, P. Poli - ARV (66)1 - CNEN-BOLOGNA-ITALY

[8] Thermite - to be issued

[9] Gathet - to be issued

ANNEXE 1

INPUT COMBINE

CARDS	COLUMN	FORMAT	DESCRIPTION	
1	1-5	I5	Number of isotopes of fast range	
1	6-10	I5	Number of isotopes of thermal range	
1	11-15	I5	The actual number of selected isotopes to be considered in the total energy range.	
2	1-5,6-10,..	14I5	Selection indicator (fast range)(*)	
3	1-5,6-10,..	14I5	Selection indicator (thermal range)(*)	
4	1-5,6-10,..	14I5	Isotropy indicator, 0 means that anisotropy is not considered 1 means that anisotropy (linear) is considered	
5	1-5	I5	First Carlson constant, if equal 3 sigma absorption is the first word, if equal 4 sigma fission is the first word.	
5		I5	Second Carlson constant.	
5		I5	Third Carlson constant.	
6	1-3,4-6,..	20A3	Alfanumeric Nuclide Identification	only useful in connection with GAD, P28 programs)
6	61-64	I4	indicator 0 a thermal transfer matrix is not to be considered 1 a thermal transfer matrix is to be considered	
6	65-70	I4	indicator 0 a fast transfer matrix is to be considered 1 a fast transfer matrix is not to be considered	

(*) Two equal numbers indicate that the two isotopes have to be combined. A zero indicates that the isotope is not to be selected.

As example suppose we have to forme a cross section library from two separate libraries in thermal and fast energy range; i.e.

Fast energy range	Thermal energy range	RESULTANT LIBRARY
1 H	H500° 1	H500°
2 Pb	H600° 1	H600°
2 Mg	Mg 2	Mg
3 U35	U35 3	Mg+Pb
4 U38	U38 4	U35
	Cd 0	U38

In this case we have to generate the following Input cards. No anisotropy is considered.

CARD						
1	5	6	5			
2	1	2	2	3	4	
3	1	1	2	3	4	0
4	0	0	0	0	0	
5	first Carlson constant			second Carlson constant		third Carlson constant
6	H500	1	0			
6	H600	1	0			
6	Mg+Pb	0	0			
6	U35	0	0			
6	U38	0	0			

INPUT MACRO

This appendice describes input cards for program which create a data set on disk containing the macro definition and relative development.

CARD	NAME	FORMAT	COLUMN	DESCRIPTION
1	MACRO(I),I=1,18	18A48	1-72	Macro name specification (Keyword)
2	NC0I	I5	1-5	Number of program generating input model elaborated by macro development
2	NM0DG	I5	6-10	Number of output models generated by macro
3	NTRAS	I5	1-5	Number of transfer program into which macro is developped
4	MODOUT(I)	I5	1-5	Output model number of transfer I starting from the last transfer
4	NM0DIN(I)	I5	6-10	Input models number of transfer I starting from the last transfer
5	IC0D(J)	I5	6-10	Appareance number of the program which generate model J, set zero if model J is generated by transfer like in intermediate model described above
5	MODEL(J)	I5	1-5	INPUT model number (note that model number can be an intermediate model like in the case in which is output of a transfer and input to another transfer)

Card 5 must be repeated NM0DIN(I) times

Card 4 must be repeated NTRAS times

Example:

Let us consider the case of macro name:

A-1 ONE BLOCK DIFFUSION MICROSCOPIC LIBRARY FROM B-1, C-1, D-1

When A is input program B,C,D are output program.
This macro has the following expansion:

A-1(26) From B-1(3), C-1(10), D-1(8), B-1(5), C-1(16)

The transfer 3+10+8+5+16 → 9 and 9 → 26 are used.

The input card are the following:

CARD 1 is of type: ONE BLOCK DIFFUSION MICROSCOPIC LIBRARY
" 2: NC0I = 3 and NM0DG = 1
" 3: NTRAS = 2
" 4: MODOUT(1) = 26 and NMODIN(1) = 1
" 5: ICOD(1) = 0 and MODEL(1)=9
" 6: MOD OUT(2) = 9 and NMODIN(2) = 5
" 7: ICOD(1) = 1 MODEL(1) = 3
" 8: " (2) = 2 " (2) = 10
" 9: " (3) = 3 " (3) = 8
" 10: " (4) = 1 " (4) = 5
" 11: " (5) = 2 " (5) = 16

INPUT TVCORR

Each card must begin with .CTE

First card contains two words, first is .CTE, second is LD in a format described above:
in column 1-4, .CTE, in column 7-8, LD.

The other cards are of two different types: one for programs and other for interface programs.

In particular input for programs is composed of five kinds of cards in the following sequence:

1) card CI
2) card SI
3) card DSL
4) card DSS
5) card ET

For interface programs the input cards are of four kinds in the following sequence

1) card CI
2) card SS
3) card DSS
4) card ET

In particular

Card	Column	Format	Description
CI	1-4	A4	.CTE
	7-8	A2	CI
	9-16	2A4	Name of program or interface
	17-20	I4	Length in K bytes of the program or interface

Card	Column	Format	Description
SI	1-4	A4	.CTE
	7-8	A2	SI
	17-20	I4	Sequence number of the model between the models treated by the program
	21-24	I4	Model number
	25-28	I4	Indicator 1 = input 2 = output 3 = I/O
	Repeat this card for each model		
DSL	1-4	A4	.CTE
	6-8	A3	DSL
	17-20	I4	Logical unit from which the program reads data set library
	21-24	I4	Number of buffers related to data set library.
	25-28	I4	Lenght in bytes of each buffer.
	Repeat this card for each logical unit		
DSS	1-4	A4	.CTE
	6-8	A3	DSS
	17-20	I4	Logical unit used by the program or interface like a data set scratch
	21-24	I4	Number of buffers related to data set scratch
	25-28	I4	Length in bytes of each buffer
	Repeat this card for each logical unit		
ET	1-4	A4	.CTE
	7-8	A2	ET
SS	1-4	A4	.CTE
	7-8	A2	SS
	17-20	I4	Output model of the interface program
	21-24	I4	Number of input models
	25-28	I4	Input models
	29-32	I4	as many as specified in
	....	I4	column 21-24

CARONTE input

The Caronte input is composed of three different sections, the first always present concern the sequence of programs to be executed, the second (can also not be present) concern the informations of transfer of data with or without elaboration between programs. The third (can also not be present) concern the modification which can be apported to DD cards at the execution time and the creation of a private data pool.

First section

The first card of this section can be BEGIN, RESTART or CONTINUATION card

The BEGIN card have the following format:

column 1-4	column 7-11
.CTE	BEGIN

and signifies that a completely new sequence of programs begins to be executed.

The RESTART card have the following format:

column 1-4	column 7-13	column 16-23	column 24-27
.CTE	RESTART	program name or interface	appareance number

and signifies that of the sequence of programs specified in the SQ card (see beloww) the program executed first is that which appears after the program or interface specified in RESTART card

The CONTINUATION card have the following format:

column 1-4	column 7-18
.CTE	CONTINUATION

and signifies that or it will be executed a sequence of programs with transfer of data between them and pre-existent private data pool or a new data set is added to a private data pool or both.

The second card of this section is a SQ card.
This card must begin with word SQ followed by the names of the programs to be executed with their appareance number preceeded by minus sign. The format is the following:

column 1-5	column 6	column 7-72
SQ	$\neq 0$ for continuation card	names of the programs to be executed

EXAMPLE

```
.CTE   BEGIN
SQ     A-1, B-1, C-1, 2 (D-1, E-1), (F-1, G-1)
```

this example siginifies that are executed the programs A,B,C, than are executed D-1, E-1, D-2, E-2, and then F-1, G-1 in loop.

Section two

The first card of this section have have the following format;

colum 1-4	column 7-10	column 12-23
.CTE	DATA	TRANSMISSION

The cards specifying transfer of data between programs can be of type "macro" or "micro".
The type "micro" is described in the following example:

```
A-1(3) FROM B-1(4), C-1(5)
```

and signifies that model 3 of program A appareance number 1 is obtained from model 4 of program B appareance number 1 and model 5 of program C appareance number 1.

The type "macro" is the following:

```
A-1 KEYWORD FROM B-1, C-1
```

and signifies that a transfer of data specified in KEYWORD (see MACRO LANGUAGE OF PART II - USER'S GENERAL MANUAL) is to be executed between progams B appareance number 1, C appareance number 1 and A appareance number 1.

For the programs in loop the cards specifying the transfer between programs must be specified twice.

Example:

```
.CTE  BEGIN
  SQ  A-1, B-1, (C-1, B-2)
.CTE DATA TRANSMISSION
     C-1 KEYWORD FROM A-1, B-1
   L C-1 KEYWORD FROM A-1, B-2
```

The card C-1 KEYWORD FROM A-1, B-1
concern transmission of data between programs A-1, B-1 and C-1 the first time that are executed.

The card L C-1 KEYWORD FROM A-1, B-2
concern transmission of data between programs A-1, B-2 and C-1 when C is executed in loop.

Third section

The first card of this section have the following format:

column 1-4	column 7-10	column 12-14	column 16-28
.CTE	DATA	SET	SPECIFICATION

Three types of cards can be present in this section:

1) card which change DDNAME of a program of the sequence
2) card which change the number of buffers and buffers length of a logical unit used by a program of the sequence
3) card for the creation of a private data pool

The card of type one have the following format:

A-1 USES DDNAME FTXXF001 INSTEAD OF FTYYF001

This card must be present when two or more programs of the sequence use the same logical unit. In this case it is to be changed the logical unit of one or more programs in order to permit that each program uses a different logical unit. Naturally the modification is to be intended to refer to that logical units that have been parametrized in the programs. The new parameter specified in this type of card is transmitted to the program by the system Caronte. The card of type two have the following format:

A-1 FOR LOGICAL UNIT 9 USES 3 BUFFERS EACH 1000 BYTES LONG

This card must be present when the number of buffers and their length used by a logical unit are different from that specified in corrispondence tables.
The cards of type three have the following format:

CREATE A PRIVATE DATA POOL

This card is present when a private data pool is to be created.

Summary description of Caronte system

The Caronte package is costitued by five programs:

1) Nucleus
2) Processor
3) Executor
4) TVCORR
5) MACRO

Nucleus, Processor, Executor constitute the System Catonte; in particular the Nucleus always resides fast memory and give control first to Processor to analyse user's input and then to Executor to execute the sequence of programs and transfers determined by the Processor. The Processor process user data: is organized in overlay structure and in variable dimensions; all matrix and vectors used and constructed by the program are grouped in a vector dimensioned 10000 words and the dimension of each vector and matrix above mentioned is determined by K parameter.

Example:

```
DIMENSION A (10000)

K1=1
K2=3000
K3=7000

CALL B (A(K1),A(K3))
......
......
```

In particular the Processor treats eight tables:

1) occupies 228 bytes for each program of the sequence considering different two or more programs with different appareance number. Actually is diemnsioned for 20 programs;

2) occupies 320 bytes for each program of the sequence. Two or more programs with different appareance number are considered only one time. Actually are considered 20 programs;

3) occupies 50 bytes for each transfer of library. Actually are considered maximum of 100 transfer

4) occupies 132 bytes for each transfer executed during a sequence of programs. Actually are considered 50 transfer

5) This table is at variable dimensions. Each table occupies (2 NSET + 4) word where NSET is a maximum number of models treated by the program. During a run exists a table for each program and transfer executed. Actually for this pool of tables are considered 1000 words.

6) This table occupies 4 words for each program of the sequence. Actually are considered 400 words.

7) This table occupies 3 words for each model treated during execution of a sequence. Actually are considered 300 words.

8) This table occupies 4 words for each model to which we refer during a execution of a CONTINUATION or for each model already treated during execution of a RESTART.

The Executor is loaded in fast memory by Nucleus, determines the code or transfer to be executed next, prepare input of programs on logical unit 5, if any, in case of loop load in fast memory all programs or transfer concerning loop, if sufficient memory is available; can reside itself with programs in fast memory but, if there is not sufficient memory available, is deleted by Nucleus and reloaded after execution of the program.

PART V

USER'S MANUAL FOR THE FIELD OF NUCLEAR REACTOR DESIGN

Index

Part V User's manual for the field of nuclear reactor design

Introduction

It will hereby described the way of using the system Caronte in connection with the Library of programs adapted to the system at ISPRA-CETIS, and through the macro language instructions correspondingly set up.

The reader is supposed to know PART I of the present report.

Sequence of programs

The programs at the moment adapted to the system Caronte and inserted in the corresponding program libraries, are given in the following list, where, near the usual name of the program, it is reported the Caronte name of the program, which has to be used in the Caronte input cards.

USUAL NAME	CARONTE NAME	
GAM-2	GAM2	
GATHER-2	GATHER2	
COMBINE	COMBINE	- (see annexe I for the input description of Combine)
DTF-4	DTF4	- (Note that the sequence of reading cross-sections by DTF4 is the following: 1) isotopes from cards - 2) isotope from library tape 3) isotope from D.P.
EQUIPOISE-3	EQUIPS3	When normal and control regions are both present, one has to call 1,2,3... N the normal regions, N+1, N+2...M the control regions, and give the corresponding cards just for these last ones.
GAZE-2	GAZE	When cross sections are taken from the D.P. note that the input data consisting of the volume per isotope must be intended multiplied by the isotope density
GAD	GAD	Note that the self shielfing indicators NSHIELD, are given in sequence (one per isotope) with Format 18 I4.
FIØDØR	FIØDØR	
P28	P28	(This a samll program which just punchs out the cross sections library cards in the format required by GAD)
GATHET	GATHET	
THERMITE	THERMITE	
ANISN	ANISN	
DØT	DØT	

The programs just listed can be used indipendently or with transmission of data among them. A "run" of Caronte consists in the execution of a sequence of programs (with the possibility of loop).

A run produces in general a DATA POOL stored on tape or disk. There is the possibility to utilize the previously constructed D.P. for successives runs.

This can be the case when a problem is not treated in an unique run due to any kind of interruption (time, lines, power break off). Besides, when the solution of a given problem requires the utilization of results stored in the D.P. by previous problems treated also months before.

When a new problem requiring no previous D.P. is to be treated, the first input control card is a BEGIN card. When a new problem requiring a previous D.P. is dealt with, the firsts input control card is a CONTINUATION card. For both the BEGIN and CONTINUATION cases, a restart after a whatever type of interruption requires as first card a RESTART card, containing the indications pointed out after the last completely executed program of the sequence. In the case of RESTART the input deck is not to be changed at all, except in its first RESTART CARD which substitutes the previous BEGIN or CONTINUATION card.
To the purpose of defining the sequence of the programs to be executed the user must indicate, separated by commas, each program by its alphanumeric name, followed by a minus sign and a number specifying the ordering of appearance of a same program in in the sequence (APPEARANCE NUMBER).

It is to be noted that in the case of CONTINUATION, the appearance number of a program must be greater than the maximum appearance number stored in a label of D.P., for the same program.

In the sequence specification is forseen the possibility of a loop between two or more programs. The loop exit condition must be specified by the user (see below).

As for the appearance number in the case of loop, it is to be noted that any program in the loop is identified by an unique appearance number.

EXAMPLE 1 (BEGIN problem)

```
GAM2-1, GATHER2-1, COMBINE-1, DTF4-1, COMBINE-2,
DTF4-2
```

EXAMPLE 2 (CONTINUATION problem) (supposing that the DATA POOL contains the results of the previous example)

COMBINE-3, DTF4-3

- In the case of one or more programs repeated periodically, a given number of times, there is the possibility to indicate briefly as pointed out in examples 3 and 4.

EXAMPLE 3 - (BEGIN problem)

3 * (GAM2-1, GATHER2-1), GAM2-4

EXAMPLE 4 - (CONTINUATION problem)

3 * (GAM2-5), GAM2-8

- In the case of loop the names of the corresponding programs must be included in parenthesis:

EXAMPLE 5

(GATHER2-1, DTF4-1), GATHER2-2

- The sequence of the programs must be actually indicated in the second card, called SQ card.

SET transmission

The possibility of SET transmission among programs is indicated in Table I. To each key word (being in fact an ensemble of words) one program is associated at the left (c) and one or more programs are associated at the right (C_1, C_2...).
The actual programs which the user can utilise at the place of C or C_1, C_2... are indicated below.

It must be noted that in the case of more than one RULE for the transfer program the indication "USING RULE"..." has to preceed the word "from". Analogously, in the case of loop the indication "UNTIL DEVIATION EQUAL... has to preceed the word "from". In the case of both presence of RULE and loop the order of the two indications to be inserted before "from" isimmaterial.

At the moment being, however, the RULE possibility is admitted only for the keyword MACROSCOPIC LIBRARY COMPOSITION . In such a case RULE 1 means that Diffusion coefficients are calculated from $1/\Sigma_{tr}$, while RULE 2 indicated a more sophisticated expression.

Besides at the moment the loops is admitted only in correlation with the key word SELF SHIELDING.

Note that the maximum number of permissible energy groups and isotopes are respectively 100 and 50.

A complete EXAMPLE OF A REACTOR CALCULATION

To illustrate the actual utilisation of the system Caronte as applied to the programs herein considered, an example has been chosen in which nearly all the possibility of the system Caronte are utilized, being the multiplication factor and fluxes calculation of the ISPRA-1 reactor.

A simplified scheme of the reactor is shown in fig. N.1
To the purpose of determining the K_{eff} and fluxes of the reactor the following calculation scheme has been chosen;

1) Fast cross sections (one broad group) calculation for composition 1 by means of the program GAM2
2) The same for composition 2
3) The same for composition 2
4) Thermal cross sections (one broad group) calculation for composition 1, by means of the program GATHER2, which utilizes source terms coming from the corresponding GAM2
5) The same for composition 2
6) The same for composition 3. An initial guess is given in the self shielding factors input to GATHER2
7) Combine specifications for the cross section libraries of composition 1
8) The same for composition 2
9) The same for composition 3
10) Calculation of the cell fluxes by means of DTF4 program, utilizing the two groups cross sections previously calculated for the composition 3
11) Thermal cross section (one broad group) calculation for composition 3 by means of GATHER2, the self shielding factors input to GATHER2 are derived by the previously executed transport calculation
    Points 10 and 11 have to be repeated until self shielding factors are stabilized
12) Two group calculation of K_{eff} and fluxes by the EQUIPS3 program, utilizing the calculated cross section library.

The job control cards for the present work are reported in annexe II.

The input control instructions corresponding to the described scheme of computation are reported in annexe III.

The complete deck for a Caronte run

At this point suppose the user has to prepare the complete input deck for a Caronte run.

First of all the job control cards have to be set up. The list of them is reported in annexe I.

Then the control input cards follow, specifying the programs sequence, and the set transmission required.

Finally the input cards for the programs indicated in the sequence must follw: it must be noted that the data which will be supplied to a program by the D.P. have not to be included. In particular the cards containing data coming all from D.P. must be omitted, and the cards containing some data coming from the D.P. must have the corresponding fields left blank.

Each of such program input deck must be preceded by a card containing the name of the program (alphanumeric characters and appearance number).

For the complete arrangement of the input deck see annexe IV..

TABLE I

MACRO-LANGUAGE INSTRUCTIONS

C SOURCE from C_1

C=GATHER2 C_1=GAM2

C SELF-SHIELDING from C_1,C_2

C=GATHER2 C_1=DTF4 C_2=COMBINE

---

optional(not accepted for THERMITE)

C ONE BLOCK TRANSPORT MICROSCOPIC LIBRARY, CHI, V from C_1,C_2,C_3

C=DTF4,THERMITE C_1=GAM2 C_2=GATHER2,GATHET C_3=COMBINE

---

optinal(not accepted for THERMITE)

C TWO BLOCKS TRANSPORT MICROSCOPIC LIBRARY, CHI, V from C_1,C_2,C_3,C_4,C_5,C_6

C=DTF4,THERMITE C_1=GAM2 C_2=GATHER2,GATHET C_3=COMBINE
C_4=GAM2 C_5=GATHER2,GATHET C_6=COMBINE

C TWO BLOCKS DIFFUSION MICROSCOPIC LIBRARY from C_1,C_2,C_3,C_4,C_5,C_6

C=GAZE C_1=GAM2 C_2=GATHER2,GATHET C_3=COMBINE
C_4=GAM2 C_5=GATHER2,GATHET C_6=COMBINE

C ONE BLOCK BURN-UP MICROSCOPIC LIBRARY from C_1,C_2,C_3

C=GAD,P28 C_1=GAM2 C_2=GATHER2,GATHET C_3=COMBINE

C TWO BLOCKS BURN-UP MICROSCOPIC LIBRARY from C_1,C_2,C_3,C_4,C_5,C_6

C=GAD,P28 C_1=GAM2 C_2=GATHER2,GATHET C_3=COMBINE
C_4=GAM2 C_5=GATHER2,GATHET C_6=COMBINE

C MACROSCOPIC LIBRARY COMPOSITIONS (1,2,3.....) from C_1,C_2,C_3

C=EQUIPS3 C_1=GAM2 C_2=GATHER2,GATHET C_3=COMBINE

C DIFFUSION MICROSCOPIC LIBRARY from C_1

C=GAZE C_1=FI0D0R

C SOURCE PER ISOTOPE from C_1

C=GATHER C_1=GAM2

BIBLIOGRAPHY

[1] GGC-II - A Program for Using the GAM-II and GATHER-II Spectrum Codes in Preparing Multigroup Cross-Section Input on Punched Cards for the GAZE, GAZED, DNS, GAPLSN, 2DXY, TDC, GAMBLE, FEVER, and MD Codes. - C.V. Smith, H.A. Vieweg - GA-4436

[2] DTF-IV, a FORTRAN-IV for Solving the Multigroup Transport Equation with Anisotropic Scattering. K.D. Lathrop - LA-3373

[3] Bell, Devaney, Hansen, Mills, Roach -"Los Alamos group averaged cross sections" - LAMS 2941

[4] 2DF - Argonne Code Center, Abstract 173

[5] EQUIPOSE 3: A two-Dimensional, Two-Group, Neutron Diffusion Code for the IBM-7090 Computer - T.B. Folwer, M.L. Tobias - ORNL-3199

[6] GAZE-2: A One-Dimensional, Multigroup, Neutron Diffusion Theory Code for the IBM-7090 - S.R. Lenihan - GA-3152

[7] GAD: An Infinite Medium Multigroup Burnup Code for Fuel Cycle Analysis F.W. Todt - GA-4150

[8] IGOR e FIODOR: Utilizzo di un atlante di sezini d'urto a 26 gruppi che tiene conto dei coefficienti di auto-schermaggio delle risonanze a varie temperature P. Azzoni - P. Poli ARV 66(1) - CNEN-BOLOGNA (ITALY)

[9] THERMITE to be issued

[10] GATHET to be issued

ANNEXE I

INPUT COMBINE

CARDS	COLUMN	FORMAT	DESCRIPTION	
1	1-5	15	Number of isotope of fast range	
1	6-10	15	Number of isotopes of thermal range	
1	11-15	15	The actual number of selectred isotoped to be considered in the to total energy range.	
2	1-5,6-10,..	1415	Selection indicator (fast range)(*)	
3	1-5,6-10,..	1415	Selector indicator (thermal range)(*)	
4	1-5,6-10,..	1415	Isotropy indicator, 0 means that anisotropy is not considered 1 means that anisotropy (linear) is considered	
5	1-5	15	First Carlson constant, if equal 3 sigma absorption is the first word, if equal 4 sigma fission is the first word.	
5		15	Second Carlson constant.	
5		15	Third Carlson constant.	
6	1-3,4-6,..	20A3	Alfanumeric nuclide identification	
6	61-64	14	idicator 0 a thermal transfer matrix is not to be considered 1 a thermal transfer matrix is to be considered	only useful in connection with GAD programs P 28
6	65-70	14	indicator 0 a fast transfer matrix is to be considered 1 a fast transfer matrix is not be considered	

(*) Two equal numbers indicate that the two isotopes have to be combined. A zero indicates that the isotope is not to be selected.

As example suppose we have to forme a cross section library from two separate libraries in thermal and fast energy range; i.e.

Fast energy range	Thermal energy range	RESULTANT LIBRARY
1 H	H500° 1	H500°
2 Pb	H600° 1	H600°
2 Mg	Mg 2	Mg
3 U35	U35 3	Mg+Pb
4 U38	U38 4	U35
	Cd 0	U38

In this case we have to generate the following Input cards. No anisotropy is considered.

CARD						
1	5	6	5			
2	1	2	2	3	4	
3	1	1	2	3	4	0
4	0	0	0	0	0	
5	first Carlson constant			second Carlson constant		third Carlson constant
6	H500	1	0			
6	H600	1	0			
6	Mg+Pb	0	0			
6	U35	0	0			
6	U38	0	0			

Annexe II

```
//JOBLIB   DD DSNAME=CTE.CODE.LIBET,DISP=SHR
           EXEC PGM=NUCLEUS
//FT02F001   DD UNIT=SYSSQ,SPACE=(CYL,(5,1))
//FT03F001   DD UNIT=SYSSQ,SPACE=(CYL,(5,1))
//FT04F001   DD UNIT=SYSSQ,SPACE=(CYL,(5,1))
//FT05F001   DD UNIT=SYSSQ,SPACE=(CYL,(5,1))
//              DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//FT06F001   DD SYSOUT=A
//FT07F001   DD SYSOUT=B
//FT08F001   DD UNIT=SYSSQ,SPACE=(CYL,(5,1))
//FT99F001   DD UNIT=SYSSQ,SPACE=(CYL,(2,1)),DCB=(,RECFM=VS,BLKSIZE=800)
//CODLIBDD   DD DISP=SHR,DSNAME=CTE.CODE.LIBET
//GO.TVCORRSP DD DSN=CTE.CODE.TVCORR,DISP=SHR
//GO.PTVCORRS DD DSN=CTE.CODE.PTVCORR,DISP=SHR
//GO.TABMACRO DD DSN=CTE.CODE.MACRO,DISP=SHR
//GO.PNTMACRO DD DSN=CTE.CODE.PMACRO,DISP=SHR
//GO.INPUTCR  DD DDNAME=SYSIN
//SYSPRINT  DD SYSOUT=A
//GO.PCGETAB  DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
//GO.PROGDATA DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
//GO.WRITENT  DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
//GO.WRITEPNT DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
```

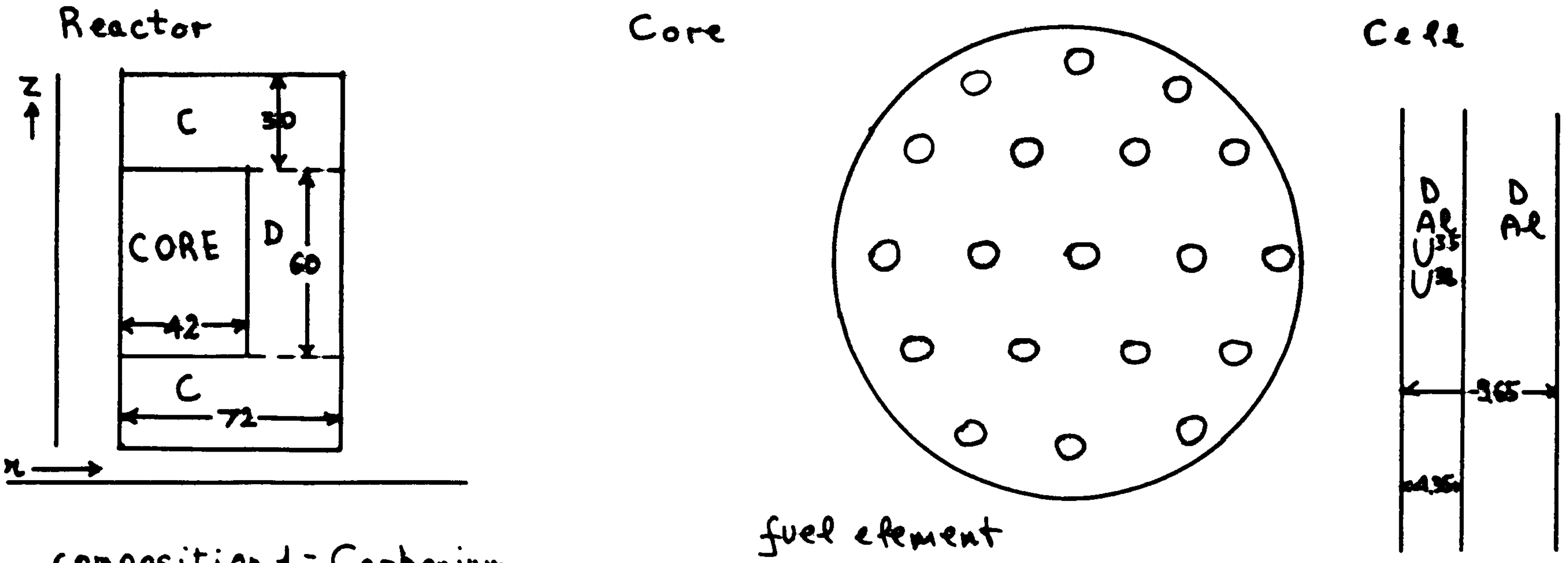
```
//DATAPOOL   DD UNIT=SYSSQ,SPACE=(CYL,(2,2))
//GO.PRIVDATA  DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
//LIBCODDD    DD DISP=SHR,DSNAME=CTE.CODE.LABET
//FT08F001    DD DSNAME=CTE.CODE.GAFASTLB,DISP=SHR
//FT10F001    DD DSNAME=CTE.CODE.GATAERLB,DISP=SHR
//GO.SYSIN   DD *
```

Annexe III

```
.CTE  BEGIN
 SQ   3*(GAM2-1,GATHER2-1,COMBINE-1),(DTF4-1,GATHER2-4),EQUIPS3-1
.CTE  DATA TRANSMISSION
      GATHER2-1 SOURCE FROM GAM2-1
      GATHER2-2 SOURCE FROM GAM2-2
      GATHER2-3 SOURCE FROM GAM2-3
      DTF4-1 ONE BLOCK TRANSPORT MICROSCOPIC LIBRARY,CHI,V
     CFROM GAM2-3,GATHER2-3,COMBINE-3
      GATHER2-4 SOURCE FROM GAM2-3
      GATHER2-4 SELF SHIELDING FROM DTF4-1,COMBINE-3
    L DTF4-1 ONE BLOCK TRANSPORT MICROSCOPIC LIBRARY,CHI,V
     CFROM GAM2-3,GATHER2-4,COMBINE-3
    L GATHER2-4 SOURCE FROM GAM2-3
    L GATHER2-4 SELF SHIELDING FROM DTF4-1,COMBINE-3,UNTIL DEVIATION
     CEQUAL 0.1
      EQUIPS3-1 MACROSCOPIC LIBRARY COMPOSITION1,USING RULE 2,FROM
     CGAM2-1,GATHER2-1,COMBINE-1
      EQUIPS3-1 MACROSCOPIC LIBRARY COMPOSITION2,USING RULE 2,FROM
     CGAM2-2,GATHER2-2,COMBINE-2
      EQUIPS3-1 MACROSCOPIC LIBRARY COMPOSITION3,USING RULE 3,FROM
     CGAM2-3,GATHER2-3,COMBINE-3
.CTE  DATA SET SPECIFICATION
      GATHER2-1 USES DDNAME FT10F001 INSTEAD OF FT09F001
```

```
       GATHER2-2 USES DDNAME FT10F001 INSTEAD OF FT09F001
       GATHER2-3 USES DDNAME FT10F001 INSTEAD OF FT09F001
       GATHER2-4 USES DDNAME FT10F001 INSTEAD OF FT09F001
.CTE   END
```

ISPRA 1 - REACTOR SCHEME



composition 1 = Carbonium

composition 2 = Deuterim (bound in D_2O)

composition 3 = Core

$C = 0{,}08 \cdot 10^{24}$ nuclei/cm^3

$D = 0{,}066 \cdot 10^{24}$ nuclei/cm^3

fuel element

$N_D = 0{,}0375 \cdot 10^{24}$ nuclei/cm^3

$N_{Al} = 0{,}0252 \cdot 10^{24}$ nuclei/cm^3

$N_u^{35} = 0{,}0000356 \cdot 10^{24}$ nuclei/cm^3

$N_u^{38} = 0{,}000382 \cdot 10^{24}$ nuclei/cm^3

moderator

$N_D = 0{,}0653 \cdot 10^{24}$ nuclei/cm^3

$N_{Al} = 0{,}00075 \cdot 10^{24}$ nuclei/cm^3

FIG. 1

To disseminate knowledge is to disseminate prosperity — I mean general prosperity and not individual riches — and with prosperity disappears the greater part of the evil which is our heritage from darker times.

Alfred Nobel

SALES OFFICES

All reports published by the Commission of the European Communities are on sale at the offices listed below, at the prices given on the back of the front cover. When ordering, specify clearly the EUR number and the title of the report which are shown on the front cover.

OFFICE FOR OFFICIAL PUBLICATIONS OF THE EUROPEAN COMMUNITIES
P.O. Box 1003 - Luxembourg 1
(Compte chèque postal Nº 191-90)

BELGIQUE — BELGIË
MONITEUR BELGE
Rue de Louvain, 40-42 - B-1000 Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42 - B-1000 Brussel

DEUTSCHLAND
VERLAG BUNDESANZEIGER
Postfach 108 006 - D-5 Köln 1

FRANCE
SERVICE DE VENTE EN FRANCE
DES PUBLICATIONS DES
COMMUNAUTÉS EUROPÉENNES
rue Desaix, 26 - F-75 Paris 15e

ITALIA
LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - I-00198 Roma

LUXEMBOURG
OFFICE DES
PUBLICATIONS OFFICIELLES DES
COMMUNAUTÉS EUROPÉENNES
Case Postale 1003 - Luxembourg 1

NEDERLAND
STAATSDRUKKERIJ
en UITGEVERIJBEDRIJF
Christoffel Plantijnstraat - Den Haag

UNITED KINGDOM
H. M. STATIONERY OFFICE
P.O. Box 569 - London S.E.1

Commission of the
European Communities
D.G. XIII - C.I.D.
29, rue Aldringen
Luxembourg