A DYNAMIC SIMULATOR FOR NUCLEAR POWER PLANTS (DSNP)

by

D. Saphier

Components Technology Division

U of C - AUA - USERDA

NOTICE
This report was prepared as an account of work sponsored by the United States Government for the United States and the United States Atomic Energy Commission. It is the property of the United States Government and is loaned to your organization. It and its contents are not to be distributed outside your organization without the express written approval of the Atomic Energy Commission. It is to be returned to the Atomic Energy Commission when requested to do so. It is to be destroyed when requested to do so. It is to be kept confidential and its contents are not to be disclosed to the public or to any other person without the express written approval of the Atomic Energy Commission.

BASE TECHNOLOGY

MASTER

NOTICE

January 1976

THE ARGONNE NATIONAL LABORATORY has been introduced from the only to permit the broadest possible availability.

1976
1977

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	3
I. Introduction	4
II. Dynamic Simulation of a Nuclear Power Plant	5
III. DSNP Requirements - A User's Viewpoint	7
IV. The DSNP Block Oriented Structure	9
V. The DSNP Library	13
VI. Data Handling	13
VII. DSNP Documentation	20
VIII. The Level Structure of the DSNP Modules	24
IX. The DSNP Simulation Language	26
X. Conclusions and Recommended Future Development	28
REFERENCES	32

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Steps Involved in Development of a Simulation Program	6
2.	Schematic Diagram of an IMFBR	10
3.	The DSNP Block Diagram of Modules	11
4a & 4b.	DSNP Basic Component Modules Library	14&15
4c & 4d.	DSNP Service Modules	16&17
5.	The Data Handling by DSNP	18
6.	Input Data File and Sorted-out Data File or a Simple Core Simulation	21
7.	Multiple Use of a DSNP Module and the Corresponding Modular Data Insertion	22
8.	Flowchart of the DSNP DOCUMENT GENERATOR Operation	23
9.	The Level Structure of the DSNP Modules	25
10.	Flowchart of the DSNP Execution Steps	29

LIST OF TABLES

<u>No.</u>	<u>Title</u>	<u>Page</u>
I	Requirements for a Dynamic Simulator for Nuclear Power Plants (DSNP)	8
II	A DSNP Program to Simulate Core Neutronics and Thermodynamics with Temperature Feedback	12
III	Input Data for a DSNP Program	19
IV	The DSNP Language	27

ABSTRACT

A new simulation language DSNP (Dynamic Simulator for Nuclear Power Plants) is being developed. It is a simple block oriented simulation language with an extensive library of component and auxiliary modules. Each module is a self-contained unit of a part of a physical component to be found in nuclear power plants. Each module will be available in four levels of sophistication, the fourth being a user supplied model. A module can be included in the simulation by a single statement. The precompiler translates DSNP statements into FORTRAN statements, takes care of the module parameters and the inter-modular communication blocks, prepares proper data files and I/O statements and searches the various libraries for the appropriate component modules. The documentation is computerized and all the necessary information for a particular module can be retrieved by a special document generator. The DSNP will be a flexible tool which will allow dynamic simulations to be performed on a large variety of nuclear power plants or specific components of these plants.

I. INTRODUCTION

Several organizations have developed computer programs to simulate nuclear power plants, or specific components of the power plants, such as the reactor core, the heat exchanger, or any other element. Some examples are the WARD "DEMO" Code⁽¹⁾, the GE Fore II Code,⁽²⁾ the APDA simulation for Enrico Fermi heat removal system,⁽³⁾ the ANL program to simulate heat removal from the CRBR core⁽⁴⁾ and many other programs written during the past 25 years. Almost all of these programs have been developed for a particular system, for a particular problem and with a specific set of transients inducing perturbations. As a result, these programs are rather inflexible and can be applied only to a particular system and to a limited set of situations. Whenever a specific power plant or part of a plant has to be simulated and no existing program covers this requirement exactly, a new simulation program is developed. The total present and past investments to develop these programs is enormous.

It is interesting to note, however, that most of the simulations are built from a limited number of similar modules. The mathematical models are often identical or very similar. The main differences among the programs are the ways by which the different modules are interconnected, the degree of sophistication of each module, the method and flexibility of the data handling, the method of solution, the computer for which the program is applicable and the many numerical data built into the program which makes it very difficult to adapt to any other system.

In the present study a Dynamic Simulator for Nuclear Power plants (DSNP) is being developed. It is believed that this simulator or "simulation language" will reduce significantly the effort necessary to simulate a power plant or a part of it.

The basic ingredients of the DSNP are a simple simulation language, an extensive library of nuclear power plant component modules, a precompiler which builds from the DSNP and the modular library, a FORTRAN program, a document generator, a documentation library of the DSNP modules which is stored in specific data files to be retrieved upon request, several data files for flexible input and output, and a set of instructions to generate user supplied modules.

II. DYNAMIC SIMULATION OF A NUCLEAR POWER PLANT

When a dynamic simulation program for a nuclear power plant is being developed, several well identified steps are involved as shown schematically in Fig. 1. One usually starts by preparing a schematic diagram of the system. The purpose of this step is to identify the physical components of the system to be simulated. Next, modules described by a set of mathematical equations that can be developed as independent units are identified. A physical component might have more than one independent module. The reactor core for example, might have a kinetics module, a thermodynamic module, a hydrodynamic module, and a feedback reactivity module. In the third step, a flowchart showing the interconnection between the various modules is prepared. This flowchart coincides with the computer program or the simulation flowchart. The three steps described above can be considered as part I or as stage one in the development of the simulation program. Approximately 5% to 10% of the project manpower time is usually invested in this stage. The second stage or part II involves four distinct steps in a closed loop. First, the mathematical models for the various modules are developed based on a set of assumptions and approximations or based on data provided by the manufacturer of the component. Next, the mathematical relations are transformed into a computer program usually having a subroutine for each module in addition to the various data handling routines

Fig. 1: Steps Involved in Development of a Simulation Program

and management sub-programs. The third step involves program debugging which is one of the most tedious and time consuming steps. In the fourth step, after the first results from the simulation program are obtained, the results have to be verified by comparing them to measured values or to results obtained with other well proven programs. If any significant discrepancies are observed, either the mathematical model or the solution method have to be changed and the whole of stage two is repeated until satisfactory results are obtained.

The purpose in developing the DSNP is to eliminate part II of the simulation process and to restrict the effort to the 5% to 10% of part I. Going from the computer flowchart to a program in DSNP is simple and straightforward as shown in the following sections.

III. DSNP REQUIREMENTS -- A USER'S VIEWPOINT

In order for the DSNP to be useful and easily applicable for a variety of nuclear power plants and parts thereof, the requirements summarized in Table I should be met. The DSNP must be a simple block oriented simulation language that can be learned in a short time such as half of a day. The transfer from the computer flowchart to the DSNP program should be straightforward requiring only a minimal effort.

The library of DSNP modules must include an extensive number of component modules so that LMFBR, PWR, BWR, GCR could be simulated. These modules must have different levels of sophistication so that simulations for different goals can be performed. The data handling must be simple and straightforward. Unformatted and easily comprehensible methods should be used for the input data.

The documentation is a vital part of any computer program or language. A simple method to retrieve the information required to comprehend a module and to

Table I: Requirements for a Dynamic Simulator for
Nuclear Power Plants (DSNP)

- (A) A SIMPLE BLOCK ORIENTED SIMULATION LANGUAGE
- (B) SUFFICIENT LIBRARY OF POWER PLANT COMPONENT MODULES TO SET UP A NUCLEAR POWER PLANT SIMULATION
- (C) EASY DATA HANDLING
- (D) EASILY COMPREHENDIBLE AND USABLE DOCUMENTATION
- (E) FEATURES TO PERMIT SIMULATIONS AT DIFFERENT LEVELS OF SOPHISTICATION AND PARTIAL CONFIGURATIONS
- (F) PROVISIONS TO USE FORTRAN AS A PART OF THE SIMULATION LANGUAGE
- (G) A SET OF STANDARD PROCEDURES FOR DEVELOPMENT OF USER SUPPLIED COMPONENT MODULES

receive the information required for its inclusion in the simulation program is to be developed. The DSNP must have provisions to include FORTRAN statements as part of the program. In addition, a set of standard procedures should be defined to allow for the development of user supplied modules, and their inclusion into the DSNP library.

The following sections describe how each of the user requirements, summarized in Table I, are met.

IV. THE DSNP BLOCK ORIENTED STRUCTURE

To see how requirement A from Table I is met, part one of the simulation process outlined in Fig. 1 will be repeated here. A schematic diagram of an LMFBR plant is shown in Fig. 2. This diagram identifies the components to be included in a particular simulation. In this example, however, only the simulation for the reactor component will be developed, and only the neutronics and thermodynamics will be included, excluding the coolant hydrodynamics. In Fig. 3 the block diagram of the modules to be included in the simulation is shown. It is seen that in this particular case, the reactor component is simulated by using nine modules from the DSNP library. The approximate flow of information between the various modules is also shown in the diagram. This block diagram is transformed into a DSNP program in a straightforward manner, as shown in Table II. The core of the program to simulate the reactor neutronics with thermodynamics, with temperature feedback, and with a simple control module, has nine simple statements. Additional modules for I/O, time step and convergency control, and additional commands to control the program flow will be included. However, the simplicity of the approach is clearly visible from this example.

Fig. 2: Schematic Diagram of an LMFBR

Fig. 3: The DSNP Block Diagram of Modules

Table II: A DSNP Program to Simulate Core Neutronics and
Thermodynamics with Temperature Feedback

CNTRL1
FDBEK1
NEUTR2
GAMAR2
LPLEN1
POWER3
CORTP3
BLNTP2
UPLEN1

V. THE DSNP LIBRARY

An integral part of the DSNP is its module library. At present the modules to be included are shown in Figs. 4a through 4d. There are thirty basic component modules and twenty-three service modules. Each of the basic component modules describes a distinct part of a component usually present in a nuclear power plant such as NEUTR for neutron kinetics, BLNTP for radial blanket thermodynamics, etc. The DSNP service modules include frequently used elements VALVC for a control valve, STEAM for steam tables or functions, SODEN to calculate the sodium density as a function of temperature, etc. The proper communication between the various modules is organized by the precompiler that translates the DSNP language into a FORTRAN statement and produces the data blocks for inter-modular communications.

VI. DATA HANDLING

The handling of the input data by DSNP is shown schematically in Fig. 5. There are one or more basic default data files for each of the library modules requiring any input data. The user has the option to use any of these default files or replace any of the data by new values. The list of data required by each module, as well as its default value from any of its chosen data files, can be produced by the document generator as shown in the next section and in Fig. 8.

The new data for the specific run and the basic default data file are combined to produce the data file for the DSNP simulation. A new optional default file can be produced to store these data for a future simulation run. The DSNP uses the combined data file and the DSNP data library, which includes steam tables, material properties or other constants which are independent of the simulation properties. The DSNP has an option to dump all the initial values to an

1: NEUTR

CALCULATE NEUTRONIC POWER

2: GAMAR

DECAY HEAT GENERATED BY RADIATION

3: CNTRL

CALCULATES REACTIVITY GENERATED BY EXTERNAL CONTROL

4: SAFTY

REACTIVITY GENERATED BY THE SAFETY SYSTEM

5: FDBEK

REACTIVITY GENERATED BY TEMPERATURE AND DENSITY CHANGES

6: POWER

CALCULATES POWER DISTRIBUTION INTO VARIOUS CORE SECTIONS

7: CORTP

CORE THERMODYNAMIC CALCULATION

8: BLNTP

RADIAL BLANKET THERMODYNAMICS

9: LPLEN

LOWER PLENUM MIXING AND THERMODYNAMICS

10: UPLEN

UPPER PLENUM MIXING AND THERMODYNAMICS

11: GASCV

COVER GAS SYSTEM

12: PPUMP

PRIMARY PUMP

13: SPUMP

SECONDARY PUMP

14: PHYDR

PRIMARY HYDRAULICS

15: SHYDR

SECONDARY HYDRAULICS

Fig. 4a: DSNP Basic Component Modules Library

16: MHDPM
MAGNETO-
HYDRODYNAMIC
PUMP

17: IHXTR
INTERMEDIATE
HEAT EXCHANGER
THERMODYNAMICS

18: HEATR
HEATER SECTION
HEAT TRANSFER

19: BOILR
BOILER SECTION
HEAT TRANSFER

20: SUPER
SUPERHEATER
SECTION HEAT
TRANSFER

21: DRUMT
STEAM DRUM
THERMODYNAMICS

22: DRUMH
STEAM DRUM HY-
DRAULICS

23: FDWTR
FEEDWATER FLOW
AND PUMP

24: WPUMP
RECIRCULATION
WATER PUMP

25: XFLOW
HYDRAULICS OF
THE STEAM GEN-
ERATOR

26: STFLW
STEAM FLOW
THROUGH THE
SUPER HEATER
TO TURBINE

27: TURBN
TURBINE

28: CNDNS
CONDENSER

29: FDWHR
FEEDWATER
HEATER

30: GENER
GENERATOR

Fig. 4b: DSNP Basic Component Modules Library

1: TIMER
TIMER STEP CON-
TROL ROUTINE

2: PIPED
PIPE AS A PURE
DELAY

3: PIPET
PIPE WITH DELAY
AND HEAT TRANS-
FER

4: PIPEH
PIPE WITH DELAY
AND HYDRAULICS

5: PIPTH
DETAILED PIPE
SIMULATION

6: VALVA
EQUAL PERCENT-
TAGE VALVE

7: VALVB
PROPORTIONAL
VALVE

8: VALVC
CONTROL VALVE

9: STEAM
STEAM TABLES
OR FUNCTIONS

10: HACOR
HEAT TRANSFER
COEFFICIENT IN
CORE

11: HAIHX
HEAT TRANSFER
COEFFICIENTS
FOR IHX

12: HAHTR
HEAT TRANSFER
COEFFICIENTS
FOR HEATER
ECONOMISER

13: HABOL
HEAT TRANSFER
COEFFICIENTS
FOR BOILER

14: HASUP
HEAT TRANSFER
COEFFICIENTS
FOR SUPERHEATER

15: SODEN
SODIUM DENSITY

Fig. 4c: DSNP Service Modules

16: FRICT
FRICTION FAC-
TOR CORRELA-
TION

17: INTEG
SEVERAL INTE-
GRATION ROU-
TINES

18: DUTIC
A ROUTINE THAT
CHOOSES THE DUTY
CYCLE TO BE
SIMULATED

19: SPLOT
STORES DATA FOR
SUBSEQUENT PLOT-
TING AND PRE-
PARES PRINTPL

20: PLOTR
PERFORMS GRAPH-
IC PLOTTING
FROM FILE

21: TPRIN
PRINTS SHORT
INFORMATION
PER TIME STEP

22: MPRIN
PRINTS DE-
TAILED IN-
FORMATION

23: INPUT
HANDLES INPUT
DATA FILES

24: RUPTR
PERFORMS A
PIPE RUPTURE

25: PRVLV
PERTURBS THE
FLOW BY CHANG-
ING VALVE
OPENING

26: PRTMP
CAUSES ABRUPT
TEMPERATURE
CHANGES IN THE
FLUID

Fig. 4d: DSNP Service Modules

Fig. 5: The Data Handling by DSNP

IC file or to dump the value of all its variables if the execution of the program is terminated before the simulation is completed. The actual input data mode for a program such as developed in Table II is given below in Table III.

Table III: Input Data for a DSNP Program

```
TIMER---INTIM = 0, FINTIM = 100, DELT = 0.01, .....
FDBEK---X = 3.6, Y(1) = 1, Y(2) = 10, Z = 25 * 3.4,.....
NEUTR---.....
GAMAR---.....
POWER---.....
UPLN---.....
LPLEN---TIN = 520, WIN = L20, .....
CORTP---
TPRIN---A, B, C, X, Y, Z, TIME, Z(18), Z(1)
SPLOT---
SPLOT---
SPLOT---
PLOT---
PLOT---
PLOT---
```

The data are included in the program by identifying the name of the module to which the data are to be supplied at the beginning of the record, and then, after three spaces, the name of a variable or a vector and its value or values respectively. The data can be inserted in any order even in the middle of the program. The input data given in Table III includes alphanumeric information for the TPRIN module. This is a list of variables that will be printed at pre-selected time steps. A complete list of the variables and the vectors with appropriate identification is available from the document generator. Additional

output modules are SPLOT and PLOTR which will produce graphic output and printed plot output. An actual example of an input file and the way by which it is sorted out for a simple core simulation is shown in Fig. 6.

If a module is used in a simulation several times, like for example pipe and valve modules, the data should be supplied the same number of times the module is inserted in the DSNP. An example for multiple use of a module is shown in Fig. 7. As shown in the figure, the module is considered an operator acting on a specific vector of data, hence different data will produce different simulation models.

VII. DSNP DOCUMENTATION

Except for a short introductory brochure explaining the DSNP language and its use, all of the DSNP documentation is stored in computer files. It can be retrieved by the DSNP DOCUMENT GENERATOR shown schematically in Fig. 8. To obtain the documentation for a specific set of modules used in a simulation program, the first version of the program or the list of modules to be used, is submitted to the document generator. The generator will search the input stream for module names. It will then search the Modular Documentation Library for the appropriate text or the Modular Data List for the data requested for the module.

The DSNP Document Generator will produce three documents for each module.

1. A write up describing the specific module and the appropriate equation, and as an option, the module program listing; 2. an alphabetical listing of all the variables used in the module and their identification; and 3. an alphabetical listing of input data for the particular module with values chosen from a predetermined default file.

There are several advantages to this method of documentation. During the development stage of the DSNP, the documentation can be kept up-to-date at all

CNDNS	NTEMP=1,	00000010
CNTRL	EXREC(1)=1.001, EXREC(2)=-0.001, TIEXR(1)=1.0, TIEXR(2)=10., IEXR=1, 00000020	
CNTRL	EXREC(3)=0.0, TIEXR(3)=100.,	00000025
CORTP	TEMP=110	00000030
DRUMH	NTEMP=240	00000040
DRUMT	NTEMP=290,	00000050
MAINP	IEDIT(1)=1, IEDIT(5)=1, LETM=2,	00000060
NEUTR	EPS 1=1.E-4, EPS 2=1.E-5,	00000070
TIMER	FINTIM=20.0, DTMAX=0.5	00000080
TPRIN	TIME, X, Y, AVRAHA, PNI, PGAM	00000100
TPRIN	DELT, PNP, SUMC, RAK, REAK, RKCEN, RKFB, RKSF,	00000110
GAMAR	IGAM=4, GAM=1.E-5, B.E-3, 0.01, 0.03, 0.0, BGM=.27,.21,.16,.36, 0.0,	00000120
/*		00000130

SORTED INPUT DATA FILE

MAINP	IEDIT(1)=1, IEDIT(5)=1, LETM=2,
TIMER	FINTIM=20.0, DTMAX=0.5
CNTRL	EXREC(1)=0.001, EXREC(2)=-0.001, TIEXR(1)=1.0, TIEXR(2)=10., IEXR=1,
CNTRL	EXREC(3)=0.0, TIEXR(3)=100.,
NEUTR	EPS 1=1.E-4, EPS 2=1.E-5,
GAMAR	IGAM=4, GAM=1.E-5, B.E-3, 0.01, 0.03, 0.0, BGM=.27,.21,.16,.36, 0.0,
CORTP	TEMP=110
DRUMH	NTEMP=240
DRUMT	NTEMP=290,
CNDNS	NTEMP=1,

SUBROUTINE NUMBER:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
#OF INPUT CARDS :	1	1	2	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

AVRAHAM IS MORE THEN 6 CHARACTERS AND WILL NOT BE PRINTED

WILL ATTEMPT TO PRINT : TIME , X , Y , PNI , PGAM ,
 WILL ATTEMPT TO PRINT : DELT , PNP , SUMC , RAK , REAK , RKCEN , RKFB ,

Fig. 6: Input Data File and Sorted-out Data File for a Simple Core Simulation

DSNP - PROGRAM

 PIPED1
 VALVA1
 PIPED1
 VALVA1

 PIPED1
 VALVA1

DSNP - DATA

PIPED---1, A = 0.5, L = 100,....
 PIPED---2, A = 0.7, L = 22,.....
 PIPED---3, A = 2.5, L = 83,.....

 VALVA---1,
 VALVA---2,

A MODULE AS AN OPERATOR

Fig. 7: Multiple Use of a DSNP Module and the Corresponding Modular Data Insertion

Fig. 8: Flowchart of the DSNP DOCUMENT GENERATOR Operation

times. A text editing program is used to update the proper text file whenever a change in a DSNP module is made. The updating of a text file is a much simpler process than a new edition of a printed document. For the user, it will be much easier to comprehend and use a module because only the relevant information for the module he wants to use will be supplied together with the appropriate variables and data listing for this module. The documents for each module will be self-contained. Presently, the documents accompanying large programs are usually very cumbersome and, more often than not, they are out of date.

VIII. THE LEVEL STRUCTURE OF THE DSNP MODULES

In designing an LMFBR, different types of simulation are requested at various stages of plant development. The following is a list of some of these simulations:

- 1 - Detailed Core Dynamics
- 2 - Detailed Heat Exchanger Dynamics
- 3 - Total Power Plant Dynamics
- 4 - Control System Design Checkout
- 5 - Effects of Changing or Optimizing a Particular Parameter

To perform the above simulations, different levels of sophistication for each component included in the simulation are required. Therefore, for each basic module there will be four levels of sophistication as shown in Fig. 9. Level - 1 is the simplest module, usually a lumped parameter module with many simplifying assumptions. Level - 3 will be the most advanced model available and will comprise usually a detailed description of the component with either a multinode approach or the digital solution of the actual partial differential equation describing the appropriate system. Level - 2 is an intermediate level module

PROGR1 - SIMPLE USUALLY LUMPED PARAMETER MODEL
PROGR2 - INTERMEDIATE LEVEL MODEL
PROGR3 - SOPHISTICATED MATHEMATICAL MODEL
PROGR4 - USER SUPPLIED MODULE

Fig. 9: The Level Structure of the DSNP Modules

between Level - 1 and Level - 3. Level - 4 is a user supplied model that can be used by the DSNP. A set of standard procedures is supplied according to which the user can develop his own module if the particular situation so requires. The name of a module is composed by a five character module designation name "PROGR" and a fixed number of 1 to 4 according to the module level. Some examples with appropriate level indicators are given in Table II.

A DSNP program can use modules of different levels in the same simulation. As an example, Fig. 9 shows a case in which a detailed simulation of the Intermediate Heat Exchanger (IHE) is required. A Level - 3 IHE component module is therefore chosen. For the components immediate to the IHE, the Level - 2 of sophistication was chosen, while for the more distant elements, Level - 1 is chosen. In this case, modules far away from the investigated component were excluded from the simulation. This method allows for very detailed simulation models while keeping the computer time within reasonable limits.

IX. THE DSNP SIMULATION LANGUAGE

The basic list of the DSNP statements is summarized in Table IV. The additional statements are any FORTRAN statements. The PROGR'I' statement starting in column 1 will be translated by the precompiler into a CALL statement where PROGR stands for the module name and 'I' for the appropriate level of sophistication. The number of statements is 4 times the number of available modules. The precompiler will insert the proper parameters, if required, and will handle the appropriate COMMON blocks.

The FORTRAN statements and comments are not changed by the precompiler and are left in their place. However a test is performed to avoid erroneous variable naming.

Table IV: The DSNP Language

PROGR'I' INDICATES THE MODULE TO BE CHOSEN, 'I' INDICATES LEVEL

PROGR---* INDICATES DATA SUPPLIED TO THE APPROPRIATE MODULE

_ INDICATES THAT THE STATEMENT IS A FORTRAN
STATEMENT

TPRIN--- INDICATES A LIST OF VARIABLES TO BE PRINTED

SPLIT--- INDICATES A LIST OF VARIABLES TO BE PRINT PLOTTED

PLOTR--- INDICATES A LIST OF VARIABLES TO BE PLOTTED ON A
GRAPHIC OUTPUT MEDIA

CNVRG_III_ X_EPS_NIT IF REQUESTED WILL CHECK CONVERGENCE OF
THE VARIABLE X TO EPS AND IF NOT ACHIEVED RETURN
CONTROL TO III (100-999). NIT ITERATIONS ARE
PERMITTED

C_ INDICATES A COMMENT

* ' ' IS A SPACE

The module name, followed by three spaces, indicates a data record and can be placed anywhere, TPRIN, SPLOT, and PLOTR will be translated to appropriate WRITE statements or will use separate modules for graphic output.

The last statement in Table IV is the CNVRG statement. This statement will be converted to appropriate FORTRAN statements to seek convergence for the variable X to a value EPS. If convergence is not achieved, control is returned to the III labeled FORTRAN statement. The number of iterations is limited by the user.

A flowchart of the various execution steps of DSNP is shown in Fig. 10. It is seen that the precompiler searches the input stream and prepares a FORTRAN MAIN program. It will sort out the various data records wherever encountered and prepare a DSNP data file. It will search the libraries for the basic DSNP modules, the auxiliary modules, and the user supplied modules and generate finally, a DSNP FORTRAN program. This program uses the appropriate data to produce the requested results.

X. CONCLUSIONS AND RECOMMENDED FUTURE DEVELOPMENT

The DSNP is in the initial stage of development. The basic framework of the system has been developed. The details of the anticipated final system are summarized in this report. A simplified working version of the program is available which has the proposed data handling capability and a Level - 1 library including a kinetic module, some thermodynamic and reactivity modules and a few service modules. The rest of the modules in the library are empty dummy elements. The DSNP presently allows me to perform kinetic calculation with simple temperature feedback and different modes of perturbation.

The DSNP will be developed in three phases. In phase I, the basic structure

Fig. 10: Flowchart of the DSNP Execution Steps

of the DSNP system will be further developed. The language, the precompiler, the intermodular communications, the method of integration of the various component, as well as timer control philosophy will be developed. Exact definitions of the component modules, and the rules for their development will be set down. The Level - 1 and the Level - 4 libraries will then be prepared according to the specified set of rules. The document generator and the Level - 1 documentation library will be prepared and the whole system integrated and tested on a variety of nuclear power plant simulation problems. The first DSNP package will then be distributed to prospective users.

During phase I, the emphasis is on the structural development of the DSNP system. For phase II, equal emphasis will be placed on the DSNP structural improvement and the development of the Level - 2 library and documentation. From the experience gained by debugging phase I DSNP and from the feedback obtained from DSNP users, the system structure and its various controlling and integrating modules will be optimized. The Level - 2 library will require a significant increase in programming effort and in the development of the appropriate intermediate mathematical models for the different component modules. The mathematical models of the various component developed by WARD in their DEMO⁽¹⁾ code, is an example of the type of sophistication sought for the Level - 2 library.

Phase III of the project will be almost completely devoted to the development of sophisticated mathematical models of the various components, incorporation of these models into the Level - 3 library and preparation of the appropriate data libraries and documentation modules.

The development of Level - 1 and Level - 2 libraries is considered straightforward. Large numbers of tested models and programs can be found in the technical literature and they will be evaluated and appropriately incorporated into the DSNP modular library. The main areas of difficulty requiring a significant

amount of research and development and ingenious solutions are the structure of the DSNP precompiler, the methods of incorporating the various component modules into a single program and the communication between the modules, the development of the master times control module allowing different modules to proceed with different time steps, the definition of the modules, the development of the rules for preparing the library modules and finally the development the Level - 3 modular library which will require detailed solutions of many difficult modeling problems not well documented at present. Even though a significant effort is required for the development of the DSNP, it is believed that the benefits from reducing the cost of future simulation projects for the nuclear research and development community, for the utilities, and for the nuclear industry merit allocation of the required funds and manpower.

REFERENCES

1. LMFBR Demonstration Plant Simulation Model "Demo", WARD-O-0005 Rev 3 (1975).
2. J.N.FDX "Fore II A Computational Program for the Analysis of Steady - State and Transient Reactor Performance", GEAP-5273.
3. A. Hosler, T. LeMarechal, "A Digital Program for the Dynamic Simulation of the Heat Removal System Including the Steam Generators for the Enrico Fermi Reactor", APDA-266 (1971).
4. D. Saphier, L. W. Kirsch, and T. P. Mulcahey, "A Reactor Core Model for a Fast Breeder Power Plant Simulation," Proc. of Power Plant Dynamics Control and Testing Symposium, Knoxville, TN (1975); see also ANL-CT-75-23 (1975).