PU/NE-00-28

DEPLETOR:

A Depletion Code for PARCS

User/Programmer's Manual

Yunlin Xu
Taek Kyum Kim
Thomas J. Downar

Purdue University
School of Nuclear Engineering

December,2000

Abstract

The depletion capability was added to the PARCS code by developing an external depletion code, DEPLETOR, with the following five functions:

1) read in the macroscopic cross sections from PMAXS, the XS file prepared by the HELIOS interface code GENPXS
2) calculate assembly wise macroscopic cross sections as a function of the assembly burnup,
3) transfer the assembly wise macroscopic cross sections to PARCS,
4) obtain the node wise fluxes from PARCS,
5) calculate assembly wise burnup increment at each step based on the assembly fluxes

In order to minimize the changes to the PARCS code, DEPLETOR was developed and will be maintained as a separate code. All information transfer between PARCS and DEPLETOR is performed using the standard message passing interface software PVM. This manual describes the methodologies employed in DEPLETOR, the structure of DEPLETOR and PARCS coupling, the input data for DEPLETOR, and provides the results of the benchmark problems to verify and validate the code performance.

Abstract ---------- i
List of Tables ---------- iv
List of Figures ---------- v
List of Appendices -------
---------- vi

1. Introduction

2. Coupled code system

3. **Depletion Methods in DEPLETOR**
3.1 Methods for calculating burnup distribution.
3.2 Methods for interpolating XS for a given burnup using the tabular XS set
3.3 Methods for calculating XS and derivatives at reference state points
3.3.1 No branch state case
3.3.1 One branch state case
3.3.1 Two branch states case

4. DEPLETOR Code Programming

5. Input Card Description

6. Changes in PARCS Input File for Depletion

7. Benchmark Problems
7.1. Introduction
7.2. Benchmark Results

List of Tables

Table 1. Content of initial Buffer (P2DIR(1))
Table 2. Content of depletion Buffer (P2DIR(2))
Table 3. Content of nuetronic Buffer (P2DIR(3))
Table 4. Content of initial T/H Buffer (P2DIR(4))
Table 5. The function and calling relationship and location of subroutines in DEPLETOR
Table 6. Description of Input cards for DEPLETOR

List of Figures

Figure 1. The overview of the depletion procedure with PARCS code ---------- 2
Figure 2. The Main structure of the Coupled code system
Figure 3. The calling sequence of subroutines in DEPLETOR
Figure 4. The calling sequence of subroutines called by READXS

List of Appendices

Appendix A. Record of Modifactions in PARCS for Depletion---------- A1

Appendix B Benchmark Problems B1

1.0 Introduction

This manual describes **DEPLETOR**, the external depletion code for **PARCS**. PARCS is a code for predicting the transient behavior of light water reactor cores. PARCS solves the steady-state and time-dependent neutron diffusion equation in three-dimensional geometry to obtain the neutron flux distribution. Because PARCS solves the eigenvalue problem and is capable of performing criticality searches, it can provide the neutron flux distribution required to perform core depletion analysis. Instead of adding the subroutines necessary to perform depletion analysis directly to the PARCS code, a separate depletion code DEPLETOR was developed, thus minimizing the modifications required in PARCS. The general scheme for coupling DEPLETOR and PARCS is shown in Figure 1. DEPLETOR generates the cross-sections at each burnup state and transfers them to PARCS. PARCS then solves the diffusion equations with the given macroscopic cross sections and transfers the resulting flux distribution to DEPLETOR.

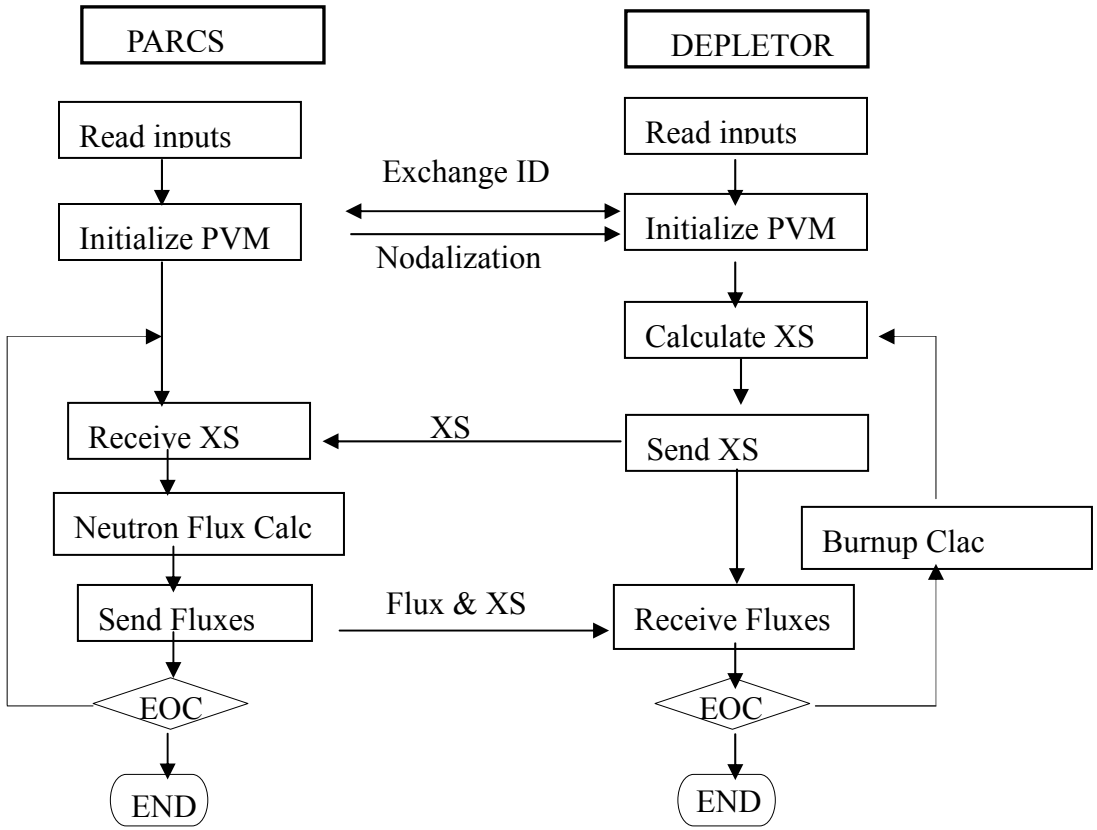


Fig 1. Schematic of the depletion procedure with DEPLETOR/PARCS codes

2. Coupled code system

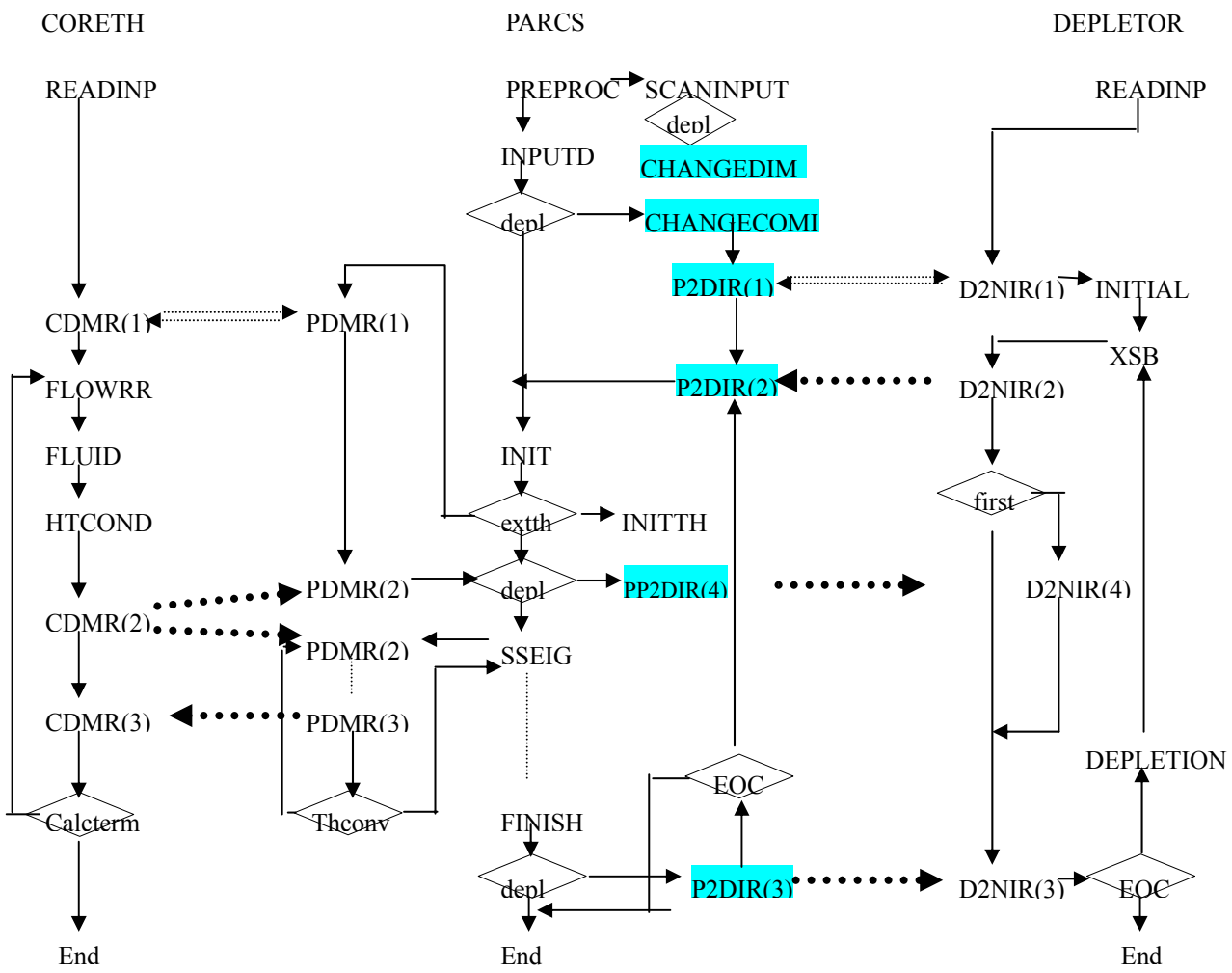
The data transfer between PARCS and DEPLETOR is performed using the message passing software PVM. The primary structure of the coupled PARCS/DEPLETOR code system is shown in Fig.2. In this structure, PARCS exchanges information with DEPLETOR at four times during each burnup step. The content of the buffers and the PARCS files involved in the are listed in Tables 1 to 4 for each of the four transfer times during a burnup step.

Table 1 Content of Initial Buffer (P2DIR(1))

Variable in the buffer		Dimension of Variable	
Variables	Related file	Dimension	Related files
Errn	p2d_var.F		
fdbk, tran, ixesmopt,srchppm	cntl.h		
nxy, nz, nassy, nfuel, nfuelfa, nasyy	geom.h	mynxy,mynz,mynxya	param.h
ltola,ltolfa, nodef, volnode, nxfas, nxfae		nz,nxy,nfuel,nasyy	geom.h
Iasyfys,nasyfy,kfs,kfe	edit.h	mynya	
ncomp, ppmref,tmref,dmref,tfref	xsec.h	myncomp	
Nprec	kin.h	mynprec	
npr0, izpr0, iprcom0	deplete.h	mynpr0	deplete.h

Table 2 Content of Depletion Buffer (P2DIR(2))

Variable in the buffer		Dimension of Variable	
Variables	Related file	Dimension	Related files
Errd	p2d_var.F		
Dend	cntl.h		
sigtr, siga, signf, sigkf,	xsec.h	ng,mycomp	param.h
delcontr, delcona, delconnf,		ncomp	xsec.h
delconkf,sig12, delcon12			
dsigtr, dsiga, dsignf, dsigkf,			
sigf, delconf, dsigf			
Sigadf	adf.h		
Gammafp,sigxea, sigsma,	xesm.h		
delconxea, delconsma,dsigxea, dsigsma			
tvelo,tbeta	inptemp.h	mynprec	
Rambda	kin.h		



Note: The highlighted subroutines and statements are added in PARCS for DEPLETION

Fig 2. The Structure of the Coupled PARCS/DEPLETOR Code System

Table 3 Content of Neutronic Buffer (P2DIR(3))

Variable in the buffer		Dimension of Variable	
Variables	Related file	Dimension	Related files
Errn	p2d_var.F		
Effk, Phi	solvec.h	ng,mynxy,mynz,	param.h
Xskf	xsec.h	Mynxya,	
Crbdens	cntrod.h	nz	geom.h
Dcool	fbvar.h	Nchan mynzth,mynchan	thgeom.h

Table 4 Content of initial T/H Buffer Buffer (P2DIR(4))

Variable in the buffer		Dimension of Variable	
Variables	Related file	Dimension	Related files
Errn	p2d_var.F		
nchan,nzth	thgeom.h	mynzth,mynchan	param.h
ktokth(nz), ltochan(1:nxy)			

The modifications to the PARCS code are list in Appendix A, and the modifications to the DEPLETOR code will be described in detail in section 4.

3. Depletion Methods in DEPLETOR

The macroscopic Cross Section (XS) representation used in PARCS is fairly standard and represented mathematically as follows:

$$\Sigma(ppm,Tf,Tm,D)=\Sigma^{r}+\frac{\partial\Sigma}{\partial ppm}\Delta ppm+\frac{\partial\Sigma}{\partial\sqrt{Tf}}\Delta\sqrt{Tf}+\frac{\partial\Sigma}{\partial Tm}\Delta Tm+\frac{\partial\Sigma}{\partial D}\Delta D+\frac{1}{2}\frac{\partial^{2}\Sigma}{\partial D^{2}}(\Delta D)^{2} \quad (1)$$

Where
- Σ^{r} : Macroscopic XS at reference state
- ppm: soluble boron concentration (ppm)
- Tf: fuel temperature (K)
- Tm: moderator temperature (K)
- D: moderator density (g/cc)
- $\frac{\partial\Sigma}{\partial x}$: derivative of XS at reference state with respect to x, x =ppm, Tf,Tm,andD
- $\frac{\partial^{2}\Sigma}{\partial D^{2}}$: second derivative of XS at reference state with respect to D

In this formulation, PARCS requires node-wise XS and derivatives at a specified burnup state point. The data provided in the PMAXS cross section files are the tabular XS at specified base states, which are usually different from the node burnup states specified in PARCS. Therefore, it is necessary to employ some type of interpolation method with the PMAXS data to calculate the XS at the appropriate burnup state.

3.1 Formulation for Calculating the Burnup Distribution.

The burnup distribution is calculated using the fluxes provided by PARCS as follows:

$$\Delta B_{i}=\Delta B_{c}\frac{P_{i}}{G_{i}}/\frac{P_{c}}{G_{c}} \quad (2)$$

where: i : ith depletion region, one region is one Z-direction node of a assembly,

ΔB_i : burnup increase of ith region,

ΔB_c : Core average burnup increment in one step, specified in DEPLETOR input,

G_i : the heavy metal loading in ith region,

G_c : total heavy metal loading in the core($=\sum G_i$),

P_i : power in ith region,

P_c : Total power in core ($=\sum P_i$).

G_i, P_i can be calculated as following:

$$G_i = \rho_i \sum_{j \in i} V_j \tag{3}$$

$$P_i = \sum_{j \in i} V_j \left[\sum_{ig} \left(\Phi_{ig,j} \times \kappa\Sigma_{f\,ig,j} \right) \right] \tag{4}$$

where: j: jth neutronic node in PARCS,

ig: igth energy group,

V_j: volume of jth node, given by PARCS,

ρ_i : heavy metal density in ith region, provided in PMAXS,

$\Phi, \kappa\Sigma_f$: Fluxes and fission energy XS, given by PARCS.

3.2 Formulations for Interpolating XS for a Specified burnup Using a Tabular XS Set

Linear interpolation is used between 2 burnup points as:

$$\Sigma(B_i) = \Sigma(B_n)\frac{B_{n+1} - B_i}{B_{n+1} - B_n} + \Sigma(B_{n+1})\frac{B_i - B_n}{B_{n+1} - B_n} \tag{5}$$

where: Σ represents the node XS and derivatives at the reference state
B_n, B_{n+1} are assembly burnups of two XS sets in PMAX.

3.3 Formulations for Calculating XS and Derivatives at Reference States

The base state in PMAX can be different from the reference state specified in the PARCS input deck. The XS at each reference state is calculated with the following formula:

$$\Sigma^r = \Sigma_0 + \Delta\Sigma_{ppm} + \Delta\Sigma_{Tf} + \Delta\Sigma_{Tm} + \Delta\Sigma_D \tag{6}$$

where : Σ_0 : XS at base state

$\Delta\Sigma_{ppm}, \Delta\Sigma_{Tf}, \Delta\Sigma_{Tm}, \Delta\Sigma_D$: the XS difference between reference states specified in PARCS input deck and the base state due to the difference in *ppm, Tf, Tm*, and *D* respectively.

In DEPLETOR, each independent variable (e.g. *ppm, Tf, Tm*, or *D*), is treated separately. Here, x is used to represent each independent variable. The formulation then becomes:

$$\Sigma(x) = \Sigma_0 + \Delta\Sigma_x + \frac{\partial\Sigma}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2\Sigma}{\partial x^2}(\Delta x)^2 \tag{7}$$

(Note: The second order derivative is only evaluated for x=*D*)

3.3.1 For the case with no branch:

$$\Delta\Sigma_x = \frac{\partial\Sigma}{\partial x} = \frac{1}{2}\frac{\partial^2\Sigma}{\partial x^2} = 0 \tag{8}$$

3.3.2 For the case with one branch:

$$\Delta\Sigma_x = (x_r - x_0)d_1, \quad \frac{\partial\Sigma}{\partial x} = d_1, \quad \frac{1}{2}\frac{\partial^2\Sigma}{\partial x^2} = 0 \tag{9}$$

Where : x_r : the reference state

x_0 : the base state

x_1 : the first branch state

d_1 : the average derivative between x_0 and x_1

3.3.3 for the case with two branches:

$x = D$:

$$\Delta\Sigma_x = (x_r - x_0)\frac{d_1 \times (x_2 - x_r) - d_2 \times (x_1 - x_r)}{x_2 - x_1}$$

$$\frac{\partial\Sigma}{\partial x} = \frac{d_1 \times (x_2 + x_0 - 2x_r) - d_2 \times (x_1 + x_0 - 2x_r)}{x_2 - x_1} \tag{10}$$

$$\frac{1}{2}\frac{\partial^2\Sigma}{\partial x^2} = \frac{d_2 - d_1}{x_2 - x_1}$$

Where : x_2 : the second branch state

d_2 : the average derivative between x_0 and x_2

$x \neq D$:

$$\Sigma^r = \Sigma_0 - \frac{d_2 \times (x_2 - x_0) \times (x_1 - x_r) - d_1 \times (x_1 - x_0) \times (x_2 - x_r)}{x_2 - x_1}$$

$$\frac{\partial\Sigma}{\partial x} = \frac{d_2 \times (x_2 - x_0) - d_1 \times (x_1 - x_0)}{x_2 - x_1}$$

3.3.4 for the case with three branches: ($x = D$ only)

$$\Delta\Sigma_x = -\frac{d_1 \times (x_1 - x_0)(x_2 - x_r)(x_3 - x_r)}{(x_1 - x_3)(x_2 - x_1)} - \frac{d_2 \times (x_2 - x_0)(x_3 - x_r)(x_1 - x_r)}{(x_2 - x_1)(x_3 - x_2)} - \frac{d_3 \times (x_3 - x_0)(x_1 - x_r)(x_2 - x_r)}{(x_3 - x_2)(x_1 - x_3)}$$

$$\frac{\partial\Sigma}{\partial x} = \frac{d_1 \times (x_1 - x_0)(x_2 + x_3 - 2x_r)}{(x_1 - x_3)(x_2 - x_1)} + \frac{d_2 \times (x_2 - x_0)(x_3 + x_1 - 2x_r)}{(x_2 - x_1)(x_3 - x_2)} + \frac{d_3 \times (x_3 - x_0)(x_1 + x_2 - 2x_r)}{(x_3 - x_2)(x_1 - x_3)}$$

$$\frac{1}{2}\frac{\partial^2\Sigma}{\partial x^2} = -\frac{d_1(x_1 - x_0)}{(x_1 - x_3)(x_2 - x_1)} - \frac{d_2(x_2 - x_0)}{(x_2 - x_1)(x_3 - x_2)} - \frac{d_3(x_3 - x_0)}{(x_3 - x_2)(x_1 - x_3)}$$

4. DEPLETOR Code Programming

The DEPLETOR code is written in FORTRAN-90. DEPLETOR has the following five functions:

1) read in the macroscopic cross sections from PMAXS, the XS file prepared by the HELIOS interface code GENPXS
2) calculate assembly wise macroscopic cross sections as a function of the assembly burnup,
3) transfer the assembly wise macroscopic cross sections to PARCS,
4) obtain the node wise fluxes from PARCS,
5) calculate assembly wise burnup increment at each step based on the assembly fluxes

The calling sequence of subroutines in DEPLETOR is shown in Fig. 3 and the calling sequence of subroutines called by the subroutine READXS is shown in Fig. 4. The function and calling relationship and location of subroutines in DEPLETOR is shown in Table 5. The function of the subroutines called by READXS are indicated by the names of subroutines and therefore are not listed in the Table. All of those subroutined, except 'errmsg' are in the file named 'read_pmaxs_file.f90', the subroutine 'errmsg'is in 'errmsg.f90'.

There are 30 files in DEPLETOR, the list of names is as following:

1)dparam.h 2)dcntl.h 3)dbls.F
4)burnup_var.F 5)dneutronic_var.F 6)dreadinp_var.F 7)interpxs_var.F
8)xsec_var.F
9)deplet.F 10)deplmain.F 11)allocnva.F 12)initial.F 13)interpxs.F
14)outpburnup.F 15)readinp.F 16)readxs.F 17)xsb.F
18)d2n-var.F 19)d2nir.F 20)d2n-initc.F 21)d2n-recv.F 22)d2n-send.F
23)d2n-recvTHnodal.F
24)read_pmaxs_file.f90 25)File_data.f90 26)assembly_data.f90 27)branch_data.f90
28)common_data.f90 29)parameters.f90 30)errmsg.f90

Table 5. Programming Information of Subroutines in Depletor

Routine	function	Called by	Call	Location
Main routine	Main structure of DEPLETOR		Readinp, READXS, D2NIR, Initial, Xsb Depletion	Deplmain.F
Readinp	Read input data from input file	Main		
READXS	Read XS from PMAX	Main	read_file_identification read_Pmaxs_file read_Pmaxs_file	Readxs.F
D2NIR	Exchange message with PARCS there are for entries D2NIR(1) D2NIR(2) D2NIR(3) D2NIR(4)	Main	 D2NIR_Init D2NIR_send D2NIR_recv D2NIR_recvTHnodal	D2nir.F
Initial	Initialize variables and get initial burnup	Main	Allocov, volhml Initburnup	Initial.F
Xsb	Main subroutine for Calculate XS for each region base on burnup distribution	Main	Xsbreg	Xsb.F
Depletion	Calculate burnup depend on the fluxes given by PARCS	Main	Outpburnup	Deplet.F
D2NIR_Init	Initialize PVM, build connection with PARCS, and get nodalization information for PARCS	D2NIR(1)	D2NIR_Obtain_Ids D2NIR_Buf_Init	D2nir_init.F
D2NIR_send	Send depletion buffer to PARCS	D2NIR(2)		D2nir_send.F
D2NIR_recv	Receive nutronic buffer from PARCS	D2NIR(3)		D2nir_recv.F
D2NIR_recv THnodal	Receive initial TH buffer from PARCS	D2NIR(3)	AllocTHv unpackTHnodal	D2NIR_rec vTHnodal.F
AllocTHv	Allocate the variables related to TH	D2NIR_rec vTHnodal		D2NIR_rec vTHnodal.F
unpackTHn odal	Unpack TH nodaliztion data	D2NIR_rec vTHnodal		D2NIR_rec vTHnodal.F
D2NIR_Obtai n_Ids	Get process ID, join PVM group, get ID of PARCS, and send ID to PARCS	D2NIR_Ini t		D2nir_init.F
D2NIR_Bu f_Init	get nodalization information for PARCS	D2NIR_Ini t		D2nir_init.F
Allocov	Allocate variables	Initial		Initial.F
Volhml	Get volume and heavy metal loading for each region	Initial		Initial.F
Initburnup	Initialize burnup distribution by read from restart file or set to zeros.	Initial	outpburnup	Initial.F
outpburnup	Initialize burnup distribution by read	Initburnup ,		Outpburnup.

	from restart file or set to zeros.	depletion		F
xsbreg	Calculate XS for one region base on it's burnup	Xsb	Basexs, d1xs d2xs, contxs	Xsb.F
Basexs	Get initial XS for reference state	Xsb	Getxs interpxs	Xsb.F
d1xs	Modify reference XS, calculate derivatives for one branch case	Xsb	Getxs interpxs	Xsb.F
d2xs	Modify reference XS, calculate derivatives for two branch case	Xsb	Getxs interpxs	Xsb.F
contxs	Get initial XS for control rod	Xsb	Getxs interpxs	Xsb.F
Getxs	Get XS when the burnup of one region happen to be same as one burnup point in the tabular XS	Basexs, d1xs, d2xs, contxs		Interpxs.F
interpxs	interpolate XS when the burnup of one region are not same as one burnup point in the tabular XS	Basexs, d1xs, d2xs, contxs		Interpxs.F

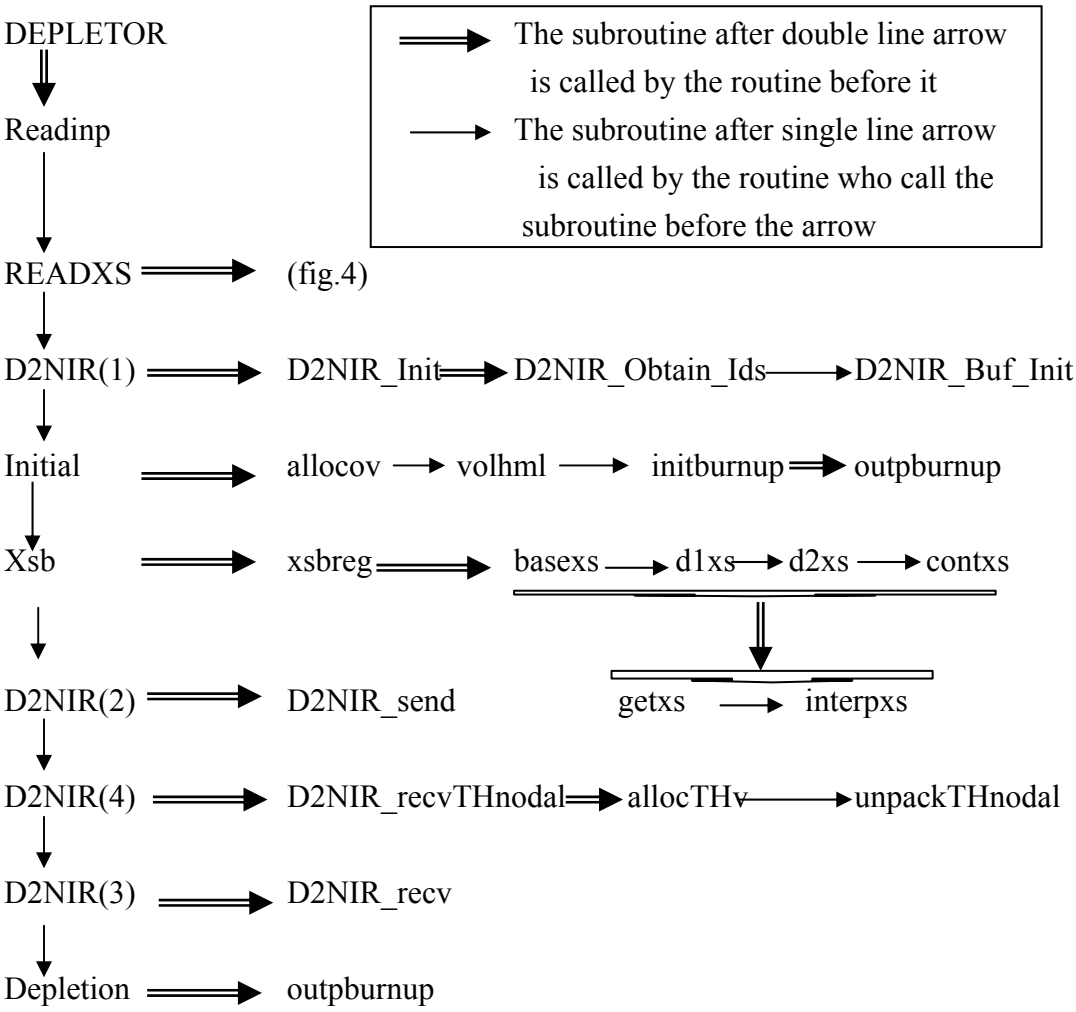


Fig. 3 The Calling Sequence of Subroutines in DEPLETOR

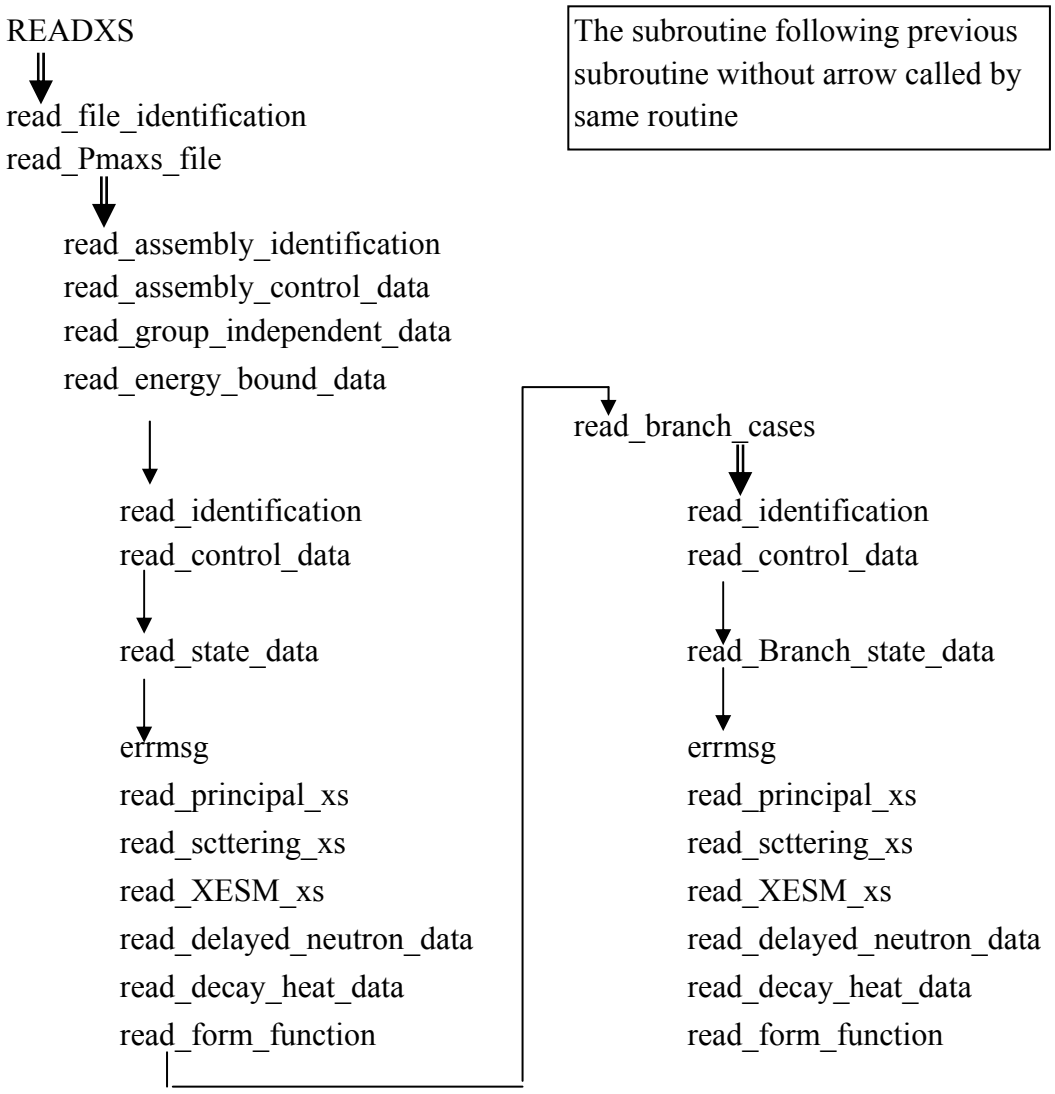


Fig. 4 The Calling Sequence of Subroutines Called by READXS

5. DEPLETOR Input Card Description

The input file of the DEPLETOR code consists of different types of input cards. Each card, leads with the card name as a key word. The field values after the card name are specified in free format in the same line as card name. The cards after 'END_INPT' will be not read.

Table 6. Description of input cards for DEPLETOR

Card Name	Field	Default	Description
CASEID	CASEID	None	Provides an alphanumeric case ID to which the three character extension are attached for generating output files
{A,F}	Power	None	Total power (MW) for calculated mode.
INPHST_F	InpHst_F	None	Name of Input file contains history information
	Format	0	0/1/2: binary/Ascii/ENTREE
{A,I,I/	RPs	1	Will restart from RPS points
(I)}	ResP(i), i=1,RPs		Restart from these points in file InpHst_F
DEPLET_S	Steps	0	DEPLETION STEPS in this calculation
{I,/ (I)}	Days(i), i=1,Steps		Days for each step
XS_FILES	XS_Fs	None	number of XS files
	$\begin{pmatrix} N \\ name(N) \\ form(N) \end{pmatrix}, N = 1, XS\_Fs$		PMAXS FILE INDEX
{I/			PMAXS FILE NAME
(I,A,I/)}			PMAXS FORM
DATA_OPT	LADF	F	T/F Tranfer/not ADF to PARCS
	LCDF	F	T/F Tranfer/not CDF to PARCS
{15L}	LLPF	F	T/F Tranfer/not LPF to PARCS
	LPFF	F	T/F Tranfer/not PFF to PARCS
	LGFF	F	T/F Tranfer/not GFF to PARCS
	LDET	F	T/F Tranfer/not DET to PARCS
	LXES	F	T/F Tranfer/not XES to PARCS
	LBET	F	T/F Tranfer/not BET to PARCS
	LAMB	F	T/F Tranfer/not LAMBDA to PARCS
	LSPD	F	T/F Tranfer/not SPDto PARCS
	LDEC	F	T/F Tranfer/not DEC s to PARCS
	LSTA	F	T/F Tranfer/not state values to PARCS
	CPOW	T	T/F normalize /not power distribution
	PPOW	T	T/F print /not power distribution
	PHST	T	T/F print /not History distribution
	PXSS	F	T/F print /not cross section
INPSTA_F	InpSta	none	Name of Input file contains instantaneous States
{A,I}	Format	0	0/1: binary/Ascii

BRANCH	DMD	0	Uniform Perturbations of Dm, Soluble Boron, Tf, Tm
	DSB	0	
{4F}	DTF	0	
	DTM	0	
CRBANK	Ncrb	0	CR banks
I2/	Mstep	0	Total steps from full inserted to withdrawn
(I)	Ncr(i),I=1,ncrb		CRs in each bank
RECRBANK	RECR	F	Rearrange/not CR banks
L,I/	Ncrbo	none	Old cr banks
(I)	icrbo(i),I=1,ncrb		Old bank index for each new bank
PTCRBANK	PTCR	F	Perturb/not cr positions
{L/(I)}	dcrp(i),I=1,ncrb		Perturb steps for each bank
EXCLBLAN	Tbla	0	The top/bottom layers of blanket you want to exclude
{I,I}	Bbla	0	from average burnup calculation
SATURATI	Rhols	0.737	Saturated water and steam densities.
{F,F}	Rhogs	0.03753	For estimate core average void fraction
END_INPT			End of input

6. Change in PARCS Input File for Depletion

There are two changes required in the PARCS input file in order to activate depletion:

1) Add a depletion cards in **CNTL** block as follows:

Card type	**Field**	**Default**	**Description**
DEPLETION	Depl	False	Involve depletion or not

Example: DEPLETION T

2) The 'REF_COND' card in the **XSEC** block is added when performing depletion. No other XSEC cards are required for PARCS when performing depletion.

7. Benchmark

7.1. Introduction

In order to validate the capability of PARCS for depletion and to verify the PMAXS format, several benchmark problems were developed to test the following specific functions:

1) Turning-on the reading of the PMAXS file,
2) Turning-on the Xe/Sm Equilibrium/Transient procedure,
3) Turning-on the depletion procedure,
4) Turning-on the T/H routines,
5) Turning-on the criticality search procedure.

Because the depletion code has many functions, it is only possible to validate the depletion code by developing and performing problems to test each individual function separately. Therefore, a step by step approach was developed from the lowest level to the highest level functions of the depletion procedure in order to systematically validate the depletion procedure. In **Appendix B**, benchmark problems are described which were designed to test the various depletion functions. In these benchmarks, HELIOS and the KAERI Depletion code MASTER serve as the reference codes. The MASTER (Multi-purpose Analyzer for steady state and transient effects of Reactor) is a light water rector core design code, developed by KAERI. The MASTER code has the capability to deplete with the cross sections of HELIOS and has been well validated with several benchmark and actual reactor core simulations. Because the depletion code of PARCS also uses the cross sections from HELIOS, this provides a suitable reference code for benchmarking purposes.

7.2. Results of Benchmarks

The depletion benchmark problems and input descriptions are provided in detail in Appendix B. Only the key results of the benchmarks will be summarized in this section.

Problem P212 is 2-dimensional PWR problem.
The following functionalities are validated with this problem:

1) Validate the proper reading of the PMAXS fileCheck the depletion routine of PARCS

2) Validate the Xe/Sm cross sections
3) Validate the Xe/Sm routine of PARCS
4) Validate the Critical Boron concentration search

The differences in the k-inf are within 70pcm. The differences of critical Boron concentrations are within 25ppm.

Problem P303 is a 3-dimensional PWR problems. This problem is used to validate the 3D calculation procedure. The trends in the k-infinite vs burnup from PARCS – DEPLETOR and MASTER are similar.

Problem B301 is a 3-dimensional BWR problem. Because the T/H feedback involves two phase flow, this problem is performed by coupling with the external T/H code, RELAP5. Therefore the capabilities of depleting a BWR core and coupling with an external T/H code are validated.