Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIII, págs. 3227-3236 (artículo completo)
Graciela Bertolino, Mariano Cantero, Mario Storti y Federico Teruel (Eds.)
San Carlos de Bariloche, 23-26 Setiembre 2014

ARCANE AND STORM DATABASE FOR THE INTEGRATION OF DIFFERENT CALCULATION TOOLS

Eduardo A. Villarino[a]

[a] *Nuclear Engineering Deparment. INVAP SE. Av. Luis Piedrabuena 4950, San Carlos de Bariloche,
Rio Negro Argentina, http://www.invap.com.ar
men@invap.com.ar*

Keywords: Neutronic Calculation, Cell Code, Core Code, calculation code databases

Abstract. INVAP calculation line and codes are in continuous development using the improvements available in the computational systems such as the increase in the memory and storage capacity and reduction of computational time. These enhancements allow the development of innovative methods for reactor calculations, that allows to include not only more detailed theories and highly accurate numerical methods, but also more prediction capabilities and additional engineering information to perform the engineering analysis of the system. The INVAP's propietary calculation line (*I.Mochi 2010*) integrates different tools with interdisciplinary or multi engineering information, improving the analysis of the calculated systems. It is based in CONDOR and CITVAP codes, but several additional tools can be used like, graphical or text based, pre and post processors to simplify the input preparation or output analysis. These computational tools allow the user to perform reactor physics calculations with a solid understanding of the set of characteristics that constitute an accurate model taking into account all the engineering aspects of the system under analysis.

As far as the integration between codes increases significantly the amount of multi engineering data available and requires a proper utilization of them by the user or analyst, a uniform interface is required to ease the proper utilization of these data and give to the analysts a solid understanding of the reactor design. To success with this uniform interface, INVAP develops three powerful computational tools: a) STORM, a hierarchy database structure for the calculation codes, b) X-Storm, a GUI utility to see the contents of the databases and c) ARCANE, a program to operate, combine and process all the data available in STORM format databases. This work describes these 3 computational tools in the frame of the INVAP calculation line, presenting its main characteristics and its potential capabilities.

1 INTRODUCTION

The variables needed in reactor physics analysis depend on the interaction of the neutrons with matter. The properties of matter are basically given by the nuclear XS and engineering data. The engineering data depends on operational conditions (for example temperature, density, cooling conditions, etc) and the nuclear XS depends basically on energy and temperature, and also in the specific engineering data like crystal and molecular structure of the materials. To obtain the neutron distribution in a reactor core (through the neutron flux) the transport equation needs to be solved. For this purpose different methods and tools are usually used. The neutronic flux depends on Position, Angle, Energy and Time, thus the solution of the flux in the whole reactor taking into account all the details of system on these variables requires a large amount of computational resources.

The INVAP calculation line (*Mochi I, 2011*) is divided in three steps: Nuclear data Library generation, Cell calculation and Core calculation. Besides it is capable to integrate the INVAP proprietary deterministic codes with third parties codes. Figure 1 shows an scheme of the INVAP calculation line, where it can be seen that several codes are incorporated, where the integration is oriented to ease the user handling of the huge amount of data involved.

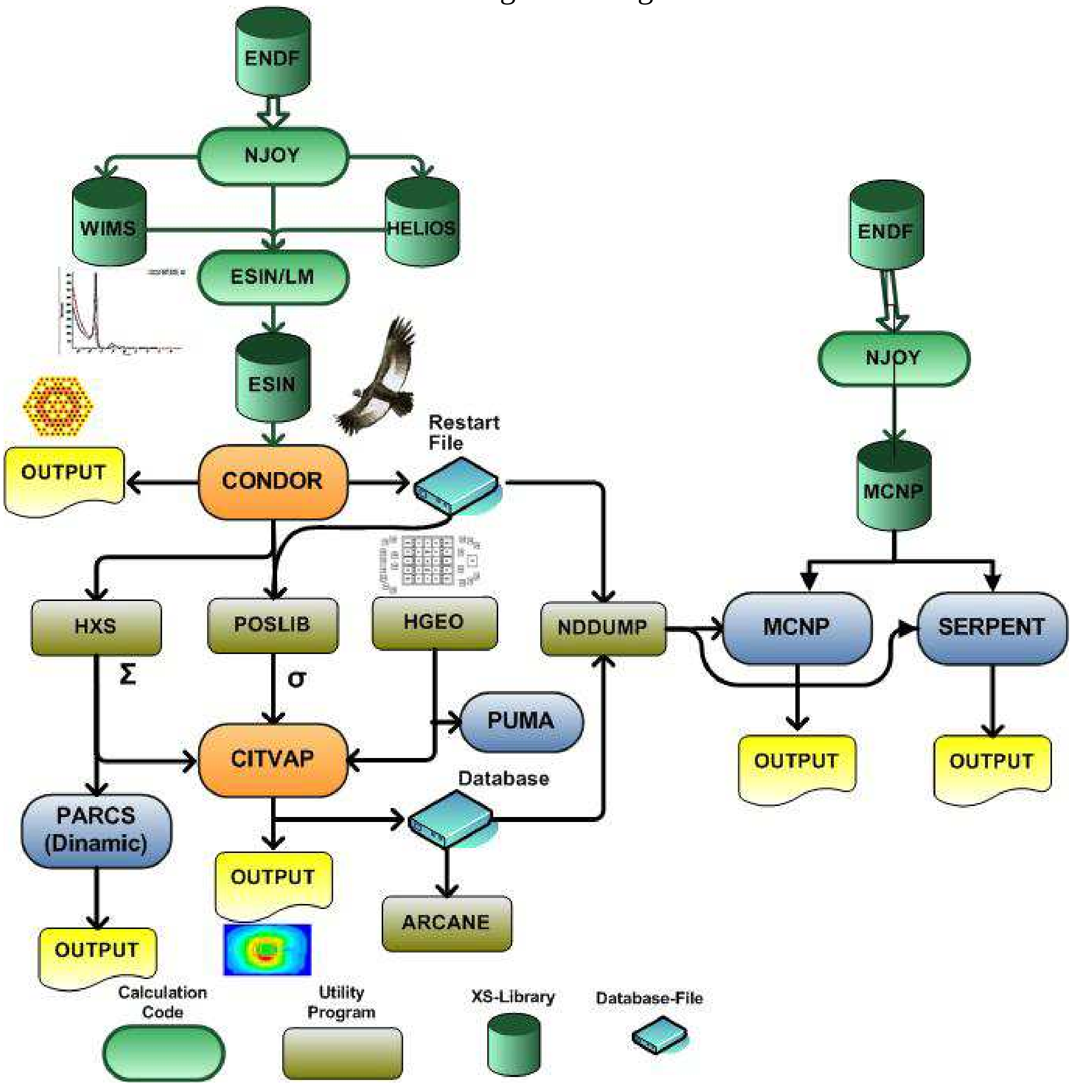


Figure 1: INVAP Calculation Line

Several nuclear and engineering variables with different engineering dependence are generated or needed by the different programs. To properly use these data by each code and by the user or analyst, INVAP develops a uniform database interface with the following concepts:

- To ease the proper utilization of these data.
- To mitigate the user dependence on the computational models.
- To provide to the analysts tools for a solid understanding of the reactor design.

To success with these concepts, INVAP developed the following computational tools:

- STORM: A hierarchy database structure for the calculation codes.
- X-Storm: A GUI utility to see the contents of the databases.
- ARCANE: A program designed to operate with the data available in a database.

2 STORM DATABASE

STORM (**STOR**age **M**ethod) is a hierarchy database structure used by computational calculation codes with the purpose to standardize the binary files. These files are used as interface between programs or with the user through graphical post processors.

The database is a direct access binary file where the data saved can be managed through its name and it is divided in three categories:

- Database Header: Administrative information of the database. Like version, creation date, time and program.
- Directories: They are used to provide a hierarchy structure to the database. They work as directories in a file system.
- Arrays: They properly contain the data or information saved in the database. They work as a file in a file system. These arrays contain the array data and they have a header identifying the data type and its shape. Two special array types are also managed: files and links. The first one is used to save the contents of a file in the database, and the second one is used to dynamically link another STORM database.

Figure 2 shows an example of the scheme of the STORM structure.

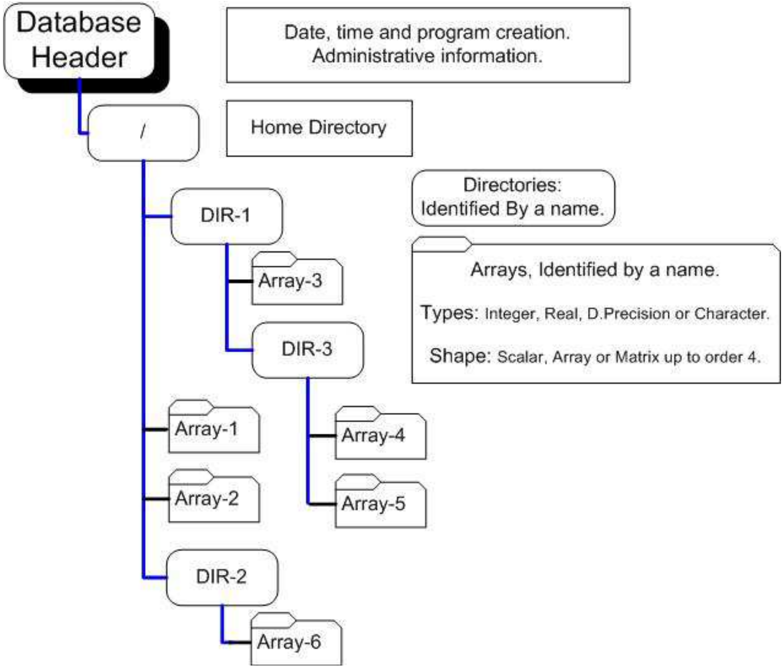

Database Header
Date, time and program creation.
Administrative information.
/
Home Directory
DIR-1
Directories:
Identified By a name.
Array-3
Arrays, Identified by a name.
Types: Integer, Real, D.Precision or Character.
Shape: Scalar, Array or Matrix up to order 4.
DIR-3
Array-1
Array-4
Array-2
Array-5
DIR-2
Array-6

Figure 2: Example of the STORM structure.

To access to the STORM databases a FORTRAN 95 library interface was developed. The modules of this library can be divided in the following categories:

- Database administration.
- Directory administration.
- Array administration.
- Utilities.

2.1 Database Administration

The modules available in this category are for database handling and they have the following functions:

- Creating.
- Opening and closing.
- Testing.
- Getting information.
- Formatting and Un-formatting.

2.2 Directory Administration

The modules available in this category are for directory handling and they have the following functions:

- Creating.
- Changing directories.
- Testing.
- Getting directory information.
- Getting list of files with customizable criteria.
- Push and pop functions.

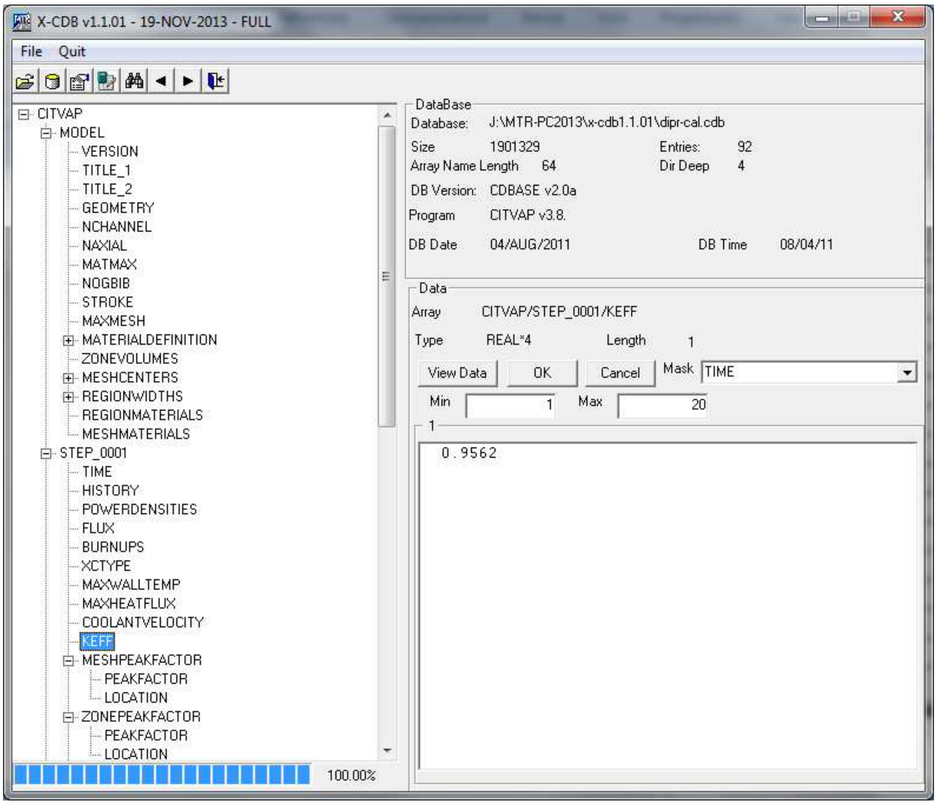
2.3 Array Administration

The modules available in this category are for array handling and they have the following functions:

- Saving: To save only the objet to be saved and its name is required. All the other information is automatically handled by the STORM library (whether it is real, integer, o double precision and whether it is a scalar, array or n-order matrix, etc).
- Reading.
- Testing.
- Getting information.
- Renaming (It can also be used to rename directories).
- Deleting (It can also be used to delete directories).

3 X-STORM UTILITY

X-Storm is a GUI utility to see the contents of the STORM databases. It allows an easy way to navigate between the database structures and has a giving to the analyst a friendly and quick way to see all database data and its general information. Besides, this utility allows the ASCII formatting of the database for further analysis. As an example, the Figure 3 and Figure 4 show present two different ways to navigate between the saved data.



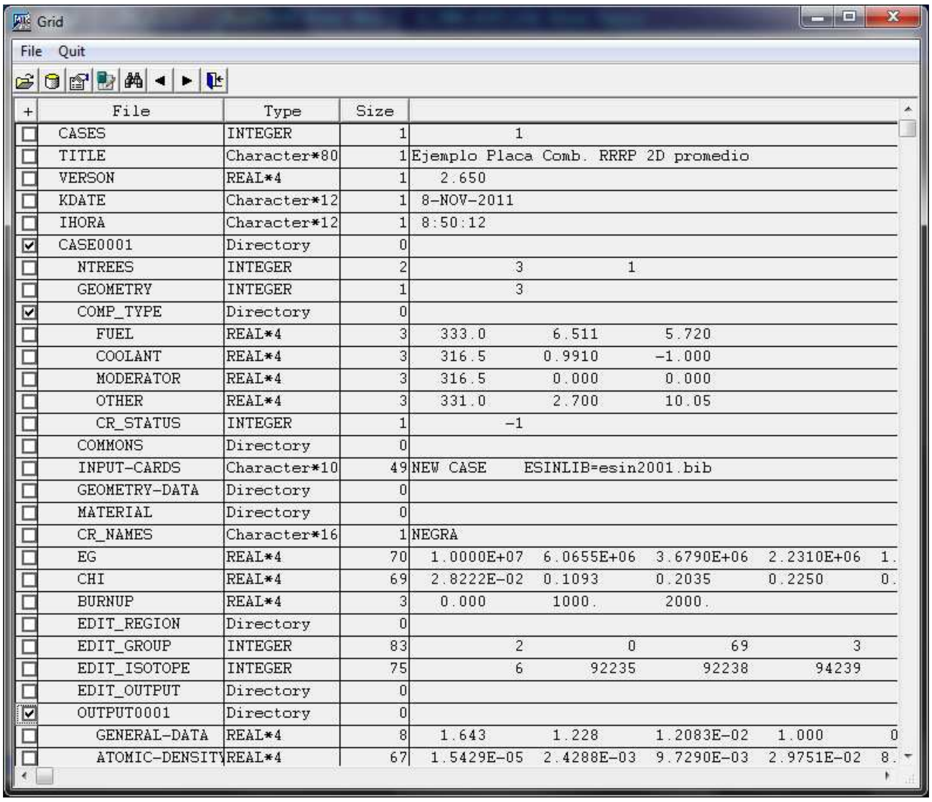
```
CITVAP
  MODEL
    VERSION
    TITLE_1
    TITLE_2
    GEOMETRY
    NCHANNEL
    NAXIAL
    MATMAX
    NOGBIB
    STROKE
    MAXMESH
  + MATERIALDEFINITION
    ZONEVOLUMES
  + MESHCENTERS
  + REGIONWIDTHS
    REGIONMATERIALS
    MESHMATERIALS
  STEP_0001
    TIME
    HISTORY
    POWERDENSITIES
    FLUX
    BURNUPS
    XCTYPE
    MAXWALLTEMP
    MAXHEATFLUX
    COOLANTVELOCITY
    KEFF
    MESHPEAKFACTOR
      PEAKFACTOR
      LOCATION
    ZONEPEAKFACTOR
      PEAKFACTOR
      LOCATION
```

100.00%

DataBase
Database: J:\MTR-PC2013\x-cdb1.1.01\dipr-cal.cdb
Size 1901329 Entries: 92
Array Name Length 64 Dir Deep 4
DB Version: CDBASE v2.0a
Program CITVAP v3.8.
DB Date 04/AUG/2011 DB Time 08/04/11

Data
Array CITVAP/STEP_0001/KEFF
Type REAL*4 Length 1
Mask TIME
Min 1 Max 20
1
0.9562

Figure 3: Standard way to navigate between Storm data



+	File	Type	Size					
□	CASES	INTEGER	1	1				
□	TITLE	Character*80	1	Ejemplo Placa Comb. RRRP 2D promedio				
□	VERSON	REAL*4	1	2.650				
□	KDATE	Character*12	1	8-NOV-2011				
□	IHORA	Character*12	1	8:50:12				
☑	CASE0001	Directory	0					
□	NTREES	INTEGER	2	3	1			
□	GEOMETRY	INTEGER	1	3				
☑	COMP_TYPE	Directory	0					
□	FUEL	REAL*4	3	333.0	6.511	5.720		
□	COOLANT	REAL*4	3	316.5	0.9910	-1.000		
□	MODERATOR	REAL*4	3	316.5	0.000	0.000		
□	OTHER	REAL*4	3	331.0	2.700	10.05		
□	CR_STATUS	INTEGER	1	-1				
□	COMMONS	Directory	0					
□	INPUT-CARDS	Character*10	49	NEW CASE ESINLIB=esin2001.bib				
□	GEOMETRY-DATA	Directory	0					
□	MATERIAL	Directory	0					
□	CR_NAMES	Character*16	1	NEGRA				
□	EG	REAL*4	70	1.0000E+07	6.0655E+06	3.6790E+06	2.2310E+06	1.
□	CHI	REAL*4	69	2.8222E-02	0.1093	0.2035	0.2250	0.
□	BURNUP	REAL*4	3	0.000	1000.	2000.		
□	EDIT_REGION	Directory	0					
□	EDIT_GROUP	INTEGER	83	2	0	69	3	
□	EDIT_ISOTOPE	INTEGER	75	6	92235	92238	94239	
□	EDIT_OUTPUT	Directory	0					
☑	OUTPUT0001	Directory	0					
□	GENERAL-DATA	REAL*4	8	1.643	1.228	1.2083E-02	1.000	0
□	ATOMIC-DENSITY	REAL*4	67	1.5429E-05	2.4288E-03	9.7290E-03	2.9751E-02	8.

Figure 4: Spreadsheet style to navigate between Storm data

4 ARCANE PROGRAM

ARCANE **(A**dvance **R**eactor **C**ore **AN**alysis **E**nvironment: enhancing the esoteric art of reactor design**)** is a post processing environment developed under FORTRAN 95 language by INVAP S.E. to perform detailed analysis of the results from codes that manage databases in STORM v2.0 format. This environment has its own language (based on FREFMT v3.3 free-format routines) that allows the user to process, operate and take decisions using data from databases in such format.

ARCANE can operate with any STORM database, but it is important to note that the current version of ARCANE code is oriented to analyze CITVAP results, thus several built-in functions are developed in such sense. Future versions of the code will have built-in functions for other codes databases (such as CONDOR) to ease its utilization.

Using this environment the user can easily:

- Manipulate and combine data from different codes like CITVAP, CONDOR, etc.
- Homogenize and condensate results.
- Evaluate design parameters and/or requirements for a given reactor using data from several databases, such as shutdown margins, discharge burnup, activity, etc.
- Systematize procedures for design criteria checking.
- Facilitate the documentation of CITVAP (and other INVAP codes) results.

The user can also access all variables from databases, thus the use of this environment is not restricted to post processing. In spite of this, the current ARCANE version includes several built-in functions, commands and automatically defined variables that are oriented to the post processing of CITVAP`s results.

The program runs in a command line mode, allowing to be run inside of CITVAP code and processing intermediate data during CITVAP calculations.

The input is composed by a combination of commands, variables and functions. In addition the user can define procedures, which are environment functions. The basic structure of the input is as follows:

- Database definition: It can be defined in the command line, using the DATABASE command (which customize the built-in function based in the program who generate the database), or using the CDB_OPEN command.
- User commands, global variable definitions, call procedures or functions, local loops, etc.
- END RUN command. After this command only procedures are allowed.
- User defined Procedures: Local variables can be defined, and similar structure as the input file is allowed. Procedures end with END procedure command.

The input structure allows the user to manipulate the data in the database through a series of simple functions, logical loops, flow control commands and user defined functions (procedures) and built-in commands. Besides, the already mentioned automatically defined variables (i.e. several database values and some mathematical constants) cannot be modified by user.

Moreover, the built-in commands included in ARCANE allows the user to operate with the database information in an easy, quick and powerful way. As an example, two important built-in commands will be discussed, presenting a simplified application;

- **Homogenization**: It is possible to homogenize in a very easy way any region of the reactor using the *`HOMOGENIZE`* command in the information stored in a database. This command allows the user to to homogenize per mesh, region, material or channel, includinglogical operators to combine these options. For example, we can obtain an homogenization scheme for a given fuel channel only for a couple of axial layers just defining an operator named as “*H_example*” as:

```
H_example=HOMOGENIZE(CHANNELS 49 / AND AXIAL_LAYERS 10:12)
```

Afterwards, This operator can be applied to any spatial dependent variable such as neutron flux, temperatures, power densities, burnup, etc to obtain the homogenizated value.

- **Condensing**: It is possible to condense in a very easy way using the *`CONDENSE`* command. This command allows the user to define a condensation scheme just identifying the upper group limit for each condensed group. For example, we can obtain a condensation scheme for three energy groups just defining an operator “C_example” as:

```
C_example=condense(1 3 5)
```

This operator can be applied to any energy dependent variable, for example a zone flux or mesh flux (if is available).

It can be seen that the combination of pre-defined operators, pre-defined variables, logical loops, flow control loops and user defined variables (global and local) allows the user to operate with all the information from a database with high versatility and intelligence. Using ARCANE, the potential applications of this are mainly limited by the user capability and imagination.

As an example, users have already used Arcane in very different applications, for example:

- Verification of design criteria using design margins.
- Calculating Mo99 activities in different core cycles.
- Conversion of the CITVAP variables to be used by Paraview (http://www.paraview.org/) software (see Figure 5)
- Condensation of CITVAP variables to be used by other engineering-oriented language software, such as OCTAVE (http://www.gnu.org/software/octave/) in a straight forward way (see Figure 6).
- Getting average values and standard deviations from different CONDOR runs.
- Getting minimum and maximum values of a given variable in the whole range of operation, to properly define the range of application of this parameter.
- Analyze macroscopic XS in function of BURNUP.

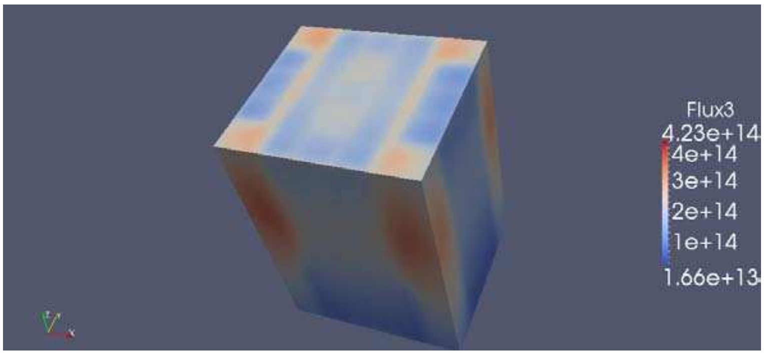


Figure 5: Example of Arcane conversion to Paraview program

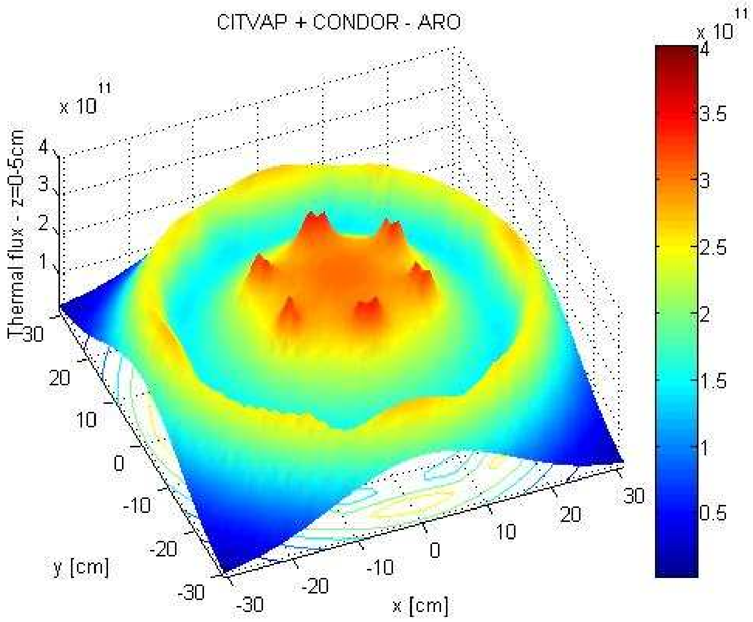


Figure 6: Example of Arcane conversion to use CONDOR-CITVAP results with OCTAVE

Two specific examples are given to show the potential capabilities of the ARCANE:

1) Calculation the maximum power peaking factor (including a 10% uncertainties) for all times steps calculated for the core, the Mo irradiation facilities and the fuel test loops.

```
DATABASE(RodCritical.cdb)
FPSFAM=0.0         ; Define Maximum
FPLOPM=0.0
FPMolM=0.0
SFAH=HOMOGENIZE(KEY "SFA") ; Define Homogenizations
LOPH=HOMOGENIZE(KEY "Loop")
MolH=HOMOGENIZE(KEY "Mol")
;
DO k=1:NSTEPS ; All time stesp
  SET_STEP(k)
  powall=MESHPOWER(SFAH)
  FPSFA=1.1*powall%MAX(1)/powall%AVG(1)
  if (FPSFA > FPSFAM) FPSFAM=FPSFA
;
  powall=MESHPOWER(LOPH)
  FPLOP=1.1*powall%MAX(1)/powall%AVG(1)
  if (FPLOP > FPLOPM) FPLOPM=FPLOP
;
  powall=meshpower(MolH)
  FPMol=1.1*powall%MAX(1)/powall%AVG(1)
  if (FPMol > FPMolM) FPMolM=FPMol
;
  PRINT TIME FPSFA FPLOP FPMOL
END DO
PRINT ""
PRINT "FP MAX" FPSFAM FPLOPM FPMolm
PRINT ""
IF ( FPLOPM > 1.5 .OR. FPMolM > 1.2 .OR. FPSFAM > 3.0) THEN
  PRINT "*********************************"
  PRINT "* DOES NOT FULFILL PPF CRITERIA *"
  PRINT "**********************************"
END IF
END RUN
END
```

2) Calculation the maximum power generating in a Fuel Element, but considering the contribution of the local PPF calculated by CONDOR. This sample shows the integration between codes, in a very easy way.

```
DATABASE(carem.cdb)
;
MAXPWR_T=0.0
PRINT (file="ppf.out" FORMAT='A') "Time  Maximum power  At Zone "
;
DO NT=1:NSTEPS                  ; All time steps
  SET_STEP(NT)
  PWR=POWERDENSITIES()
  MAXPWR=0.0
  DO NZ=1:NZONES                ; Loop over zones.
    NOMBRE=KEY_NAME(NZ)
    FILE=NOMBRE//".cdu"
    PPFL=1.0
    CALL READ_CONDOR(PPFL)
    PWRL=PWR(NZ)*PPFL
    IF(PWRL > MAXPWR) THEN
      MAXPWR=PWRL
      MAXNZ=NZ
    END IF
  END DO
  PRINT (FORMAT='F7.2,F10.3,I9') TIME MAXPWR MAXNZ
  IF(MAXPWR > MAXPWR_T) THEN
    MAXPWR_T=MAXPWR
    TIMEMX=TIME
  END IF
END DO
PRINT (FORMAT='A') "-------------------------"
PRINT (FORMAT='A,F10.3,A,F7.2') "Max.Power" MAXPWR_T " at time " TIMEMX
END RUN
;
; This procedure get the maximum PPF from Condor Calculation.
;
PROCEDURE READ_CONDOR(PPFC)
PPFC=1.0
NCDU=CDB_OPEN(FILE)
IF(NCDU < 0) RETURN
;
CDB_DIR(NCDU,'/CASE0001')
BU=CDB_READ(NCDU,'BURNUP')
BUX={0.0,BU} ;  Agrega el primer paso 0 sin Xe.
NBU=LEN(BUX)
DO N=1:NBU
  STEP='OUTPUT'//CONVERT(N,'(I4.4)')
  CDB_DIR(NCDU,STEP)
  GD=CDB_READ(NCDU,'GENERAL-DATA')
  IF(PPFC < GD(4)) PPFC=GD(4)
  CDB_DIR(NCDU,'..')
END DO
CDB_CLOSE(NCDU)
END PROCEDURE
END
```

5 CONCLUSIONS

INVAP calculation line and codes are in continuous development in order to give to the user powerful and friendly tools to face the design and analysis of reactor cores. The newly development of the STORM database structure enhances the integration between all programs and also represents a key issue in the development of the INVAP calculation line from the maintenance and upgrades of the codes point of view.

Besides, a significant improvement was achieved from the user point of view with the GUI X-Storm utility, allowing to properly understand the main characteristics of the reactor design.

Finally, the development of the ARCANE program and its capabilities allows the user to operate with any parameter of the reactor using a high level language oriented to reactor design. This gives the user a very powerful environment for the calculation and analysis of the reactor that allows the manipulation of all data included in databases from different codes. The near future applications of this environment are mainly limited by the user´s imagination, where the systematization of procedures to check design parameters, the optimization of design and the operation between databases arise as very promising ones.

REFERENCES

Mochi I., INVAP's Nuclear Calculation System, *Science and Technology of Nuclear Installations (Volume 2011, Article ID 215363)*, 2011.