Nuclear Research Institute Řež plc

www.ujv.cz

MELCOR post-processing using open source tools — 2012 update

Petr Vokáč, vok@ujv.cz

NRI Řež plc

4th EMUG meeting, 16/17 April 2012, Cologne

Overview

- Fortran tools :
  - `readptf.exe`
  - `tranptf.exe`
  - implementation&testing
- Python tools :
  - Post-processing
  - Pre-processing
  - Installation
- Example
- Contents of CD prepared for $^{4\text{th}}$EMUG

This presentation is an update of talks given at :

- CSARP/MCAP 2009
- EMUG 2011 (presentation can be downloaded from sacre.web.psi.ch)

and it is focused on new features.

Fortran tools: `readptf.exe`

`readptf.exe` is a command line tool for data retrieval from the MELCOR binary plot file.

New features:

- data of control function argument variables added by MELCOR command plot*nnn* can be retrieved, e.g.: COR-EBND-RAT-RADIAL.ia /c/
- data for more than one variable can be output to single dataset, e.g.:
  `readptf.exe` MELPTF "CVH-TVAP.100 CVH-TSAT(P).100"
- ambiguity in meaning of array index is avoided by trailing dot, e.g.:
  CVH-MASS.1 will return total fluid mass in CVH volume 1 (if it exists)
  CVH-MASS.1. will return pool mass for all CVH volumes in any case
- check for variable index change in the plot file

Fortran tools: `tranptf.exe`

`tranptf.exe` is a command line tool which allows to:

- merge several plot files to single one or
- extract selected data to new smaller plot file:
  - selected time interval
  - selected variables (not tested yet)

Single command line argument: configuration file name.

Configuration file is in the Fortran name list format.

Fortran tools: `tranptf.exe` (2)

Example input

list of `tranptf.conf`:

```
&INPUT
smelptfi(1) = "../f01/MELPTF"
sendtime(1) = "1.317311E+04"
smelptfi(2) = "../f02/MELPTF"
sendtime(2) = "1.38388E+04"
smelptfi(3) = "../f03/MELPTF"
sendtime(3) = "1.42182E+04"
smelptfi(4) = "../f04/MELPTF"
smelptfo = "MELPTF"
/
```

Example output:

```
tranptf.exe tranptf.conf
------------ Input check ------------------------
Input file  1 : ../f01/MELPTF       1.317311E+04
Input file  2 : ../f02/MELPTF       1.38388E+04
Input file  3 : ../f03/MELPTF       1.42182E+04
Input file  4 : ../f04/MELPTF
Output file : MELPTF
Records per dot :   1
------------ Input check end --------------------
Processing file: ../f01/MELPTF :
.........................................xxxx
Processing file: ../f02/MELPTF :
.........................................xxxxxxx
Processing file: ../f03/MELPTF :
..............xxx
Processing file: ../f04/MELPTF :
.........................................
tranptf finished
```

Fortran tools : implementation&testing

both codes were :

- written in Fortran 90 (name list input, modules)
- compiled using GNU make
- tested with compilers : Intel Fortran 11.1 ($^{\text{effective use of HD cache in Linux,}}_{\text{better performance}}$),
  GNU gfortran 4.5.3,
  g95 version 0.92
- tested in : Linux (64-bit), Mac OS X (64-bit) with macports,
  Windows (g95 for Windows, cygwin gfortran, cygwin make)

`readptf.exe` was tested with plot files of MELCOR 1.8.5, 1.8.6 and 2.1
(2.1 just with distributed sample cases)

`tranptf.exe` was tested just with 1.8.6 plot files

Python tools: post-processing

- `browseptf.py` — simple GUI for `readptf.exe` and GNUplot
- state snapshots using PyX
  - `cor-volf.py` — core degradation state using cor-volf-$xxx.n$ variable and water levels in adjacent CVH volumes;
    `colors.py` plots color key
  - `cor-tlh.py` — temperature color map of the lower head wall and molten pool or debris in the lower plenum
  - `cvh-liql.py` — liquid level in CVH volumes
  - `cvh-t.py` — temperature color map of CVH volumes (gas) and HS
  - `cor-tz.py` — axial profiles of COR parameters
- miscellaneous calculations with data retrieved using `readptf.exe`

Python tools : pre-processing

`inpdoc.py` is used for documentation typesetting

MELCOR input file should contain comments such as :

inp1.inc :

```
*en*cv100*this is control volume 100
cv10000 'cvh-100'
*en*cv200*this is control volume 200
cv20000 'cvh-200'
*en*fl100*this is flow path from volume 100 to 200
fl10000 'fl-100-200' 100 200
```

inp2.inc :

```
*en*hs10020*heat structure between 100 and 200
hs10020001 '100wall'
hs10020400 1 100 'INT' 0.1 0.1
hs10020600 1 200 'EXT' 0.1 0.1
```

Makefile:

```
out=example.tex
inc=\
inp1.inc \
inp2.inc

all: $(inc)
inpdoc.py $(inc) > $(out)
```

LaTeX output :

CVH - control volumes

cv100 cvh-100
this is control volume 100

cv200 cvh-200
this is control volume 200

FL - flow paths

fl100 fl-100-200 cv100→cv200
this is flow path from volume 100 to 200

HS - heat structures

hs10020 100wall cv100↔hs10020↔cv200
heat structure between 100 and 200

Python tools : installation

All scripts were tested with Python 2.7
(it should work also with 2.6 or 2.5, providing all the necessary packages are installed)

Additional Python packages needed (not always installed by default) :

- all scripts : subprocess module
  (it should be standard module but it is not installed by default on RedHat)
- `browseptf.py` : pygtk (and everything it depends on : pypango, pycairo . . .)
  and GNUplot
- scripts using PyX : pyx module, pyPdf or pdftk, LaTeX

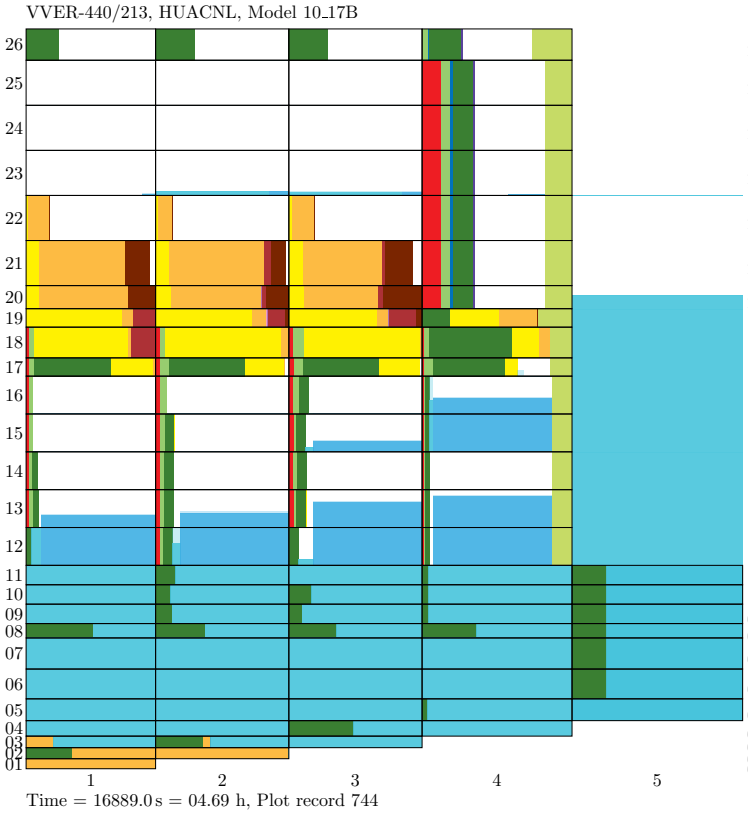
OS portability

- Linux — installation depends on distribution, no problems expected
- Mac OS X — use macports, do not forget to select active Python version
  (e.g. : `sudo port select python python27`)
- Windows — try cygwin (py-pyx is not there)
  or try “all in one” Python+GTK installer for Windows,
  see www.pygtk.org

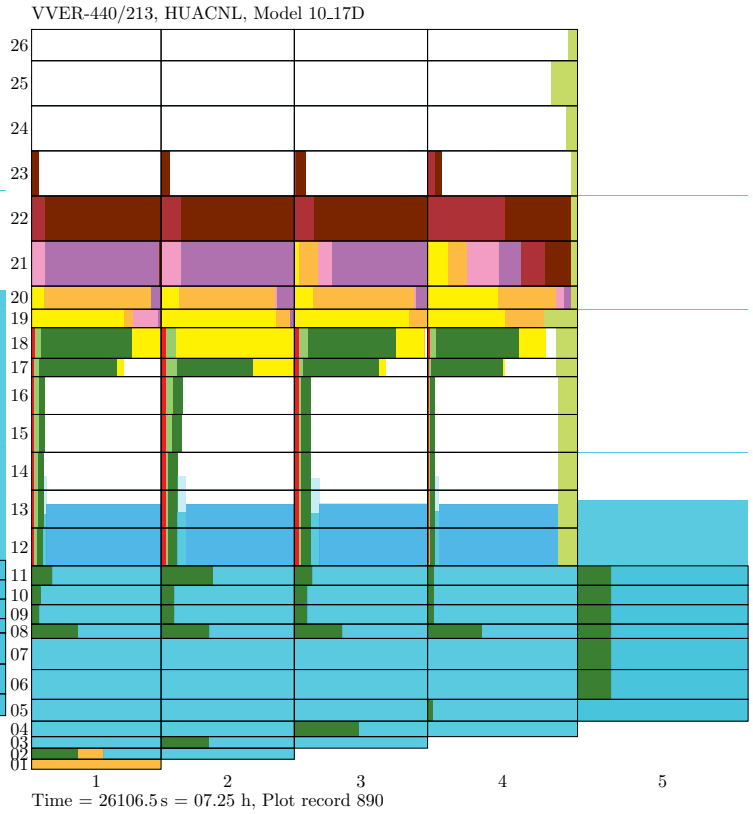
Python modules should be on `PYTHONPATH`, `readptf.exe` should be on `PATH`.

Example : VVER-440/213 core blockage sensitivity study

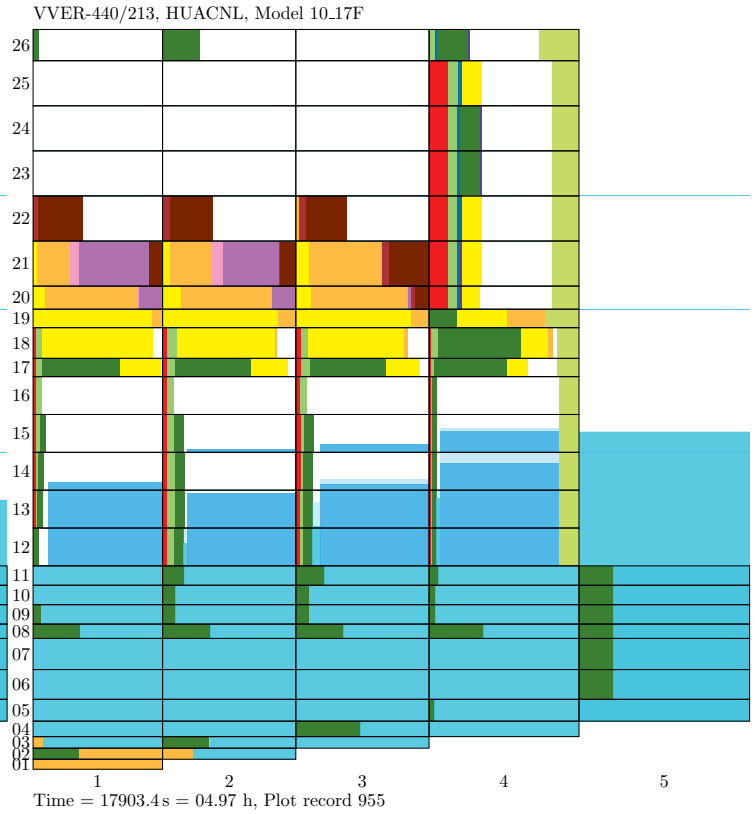
Flow blockage on all axial flow paths, time 4.69 h (just before the support plate failure)



No axial flow paths blockage 7.25 h ($\sim 2\frac{1}{2}$ h later !)



No blockage in the bypass of the peripheral ring 4.97 h ($\sim \frac{1}{4}$ h later)



Contents of CD prepared for 4th EMUG

- source files in Fortran and Python
- `readptf.exe` and `tranptf.exe` executables for Windows (g95)
- documentation in HTML format
- examples — evaluation of MELCOR 2.1 distributed sample cases

Nuclear Research Institute Řež plc

I would like to invite MELCOR users to use presented post-processing tools and participate in their further development

Thank you for your attention