Improvements Multi-deck and Strip Features

Dr. George Mesina

International RELAP5-3D User Group Meeting

Date: Aug 13, 2015

Overview

- Multi-deck capability enhancement
- Strip conformance for MBINARY
- Data Management Improvement

Restart, Plot and Strip Improvements

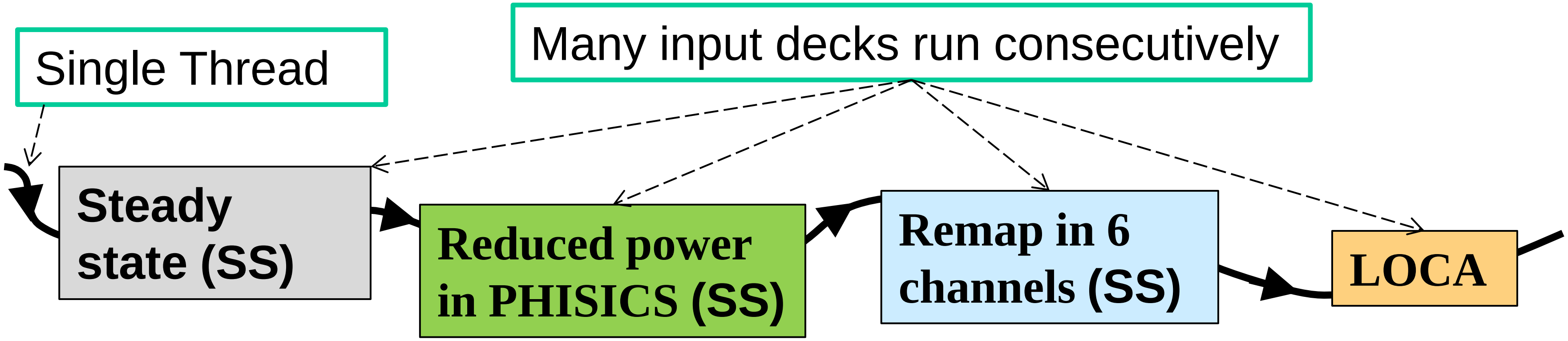
- The restart-plot file was split a decade ago
- Some issues arose because of the split
- Interest in the RELAP5-3D multi-deck capability has increased
    - Some issues have been identified
    - Some expansion is requested

Multi-deck Input File

- RELAP5-3D has a multi-deck input file capability
  - Can run two similar or unrelated decks from one input file
  - Used in standard installation test suite (3dflow, 3dflown, edstack)
    - E.G. 3dflow – two decks with 9 input cases each
- Some uses:
  - For parameter studies where each deck creates output files named for parameter value
  - Strip large plot files to desired variables with second deck. Useful for internet transmission
  - Time balancing in batch environments.
    - Group short-duration decks in one file

Multi-deck Input File

- PHISICS project needs Multi-deck capability
  - Must run a sequence of RELAP5-3D problems on a single thread of an INL supercluster computer
  - Each successive input deck is modified in a different way
  - Each sequence varies a set of parameters of interest
  - All RELAP5-3D runs must occur on same thread
  - Ensured if RELAP5-3D runs all decks without stopping
  - Cannot use input deck cases because all data gets re-initialized


Single Thread

Many input decks run consecutively

Steady state (SS)

Reduced power in PHISICS (SS)

Remap in 6 channels (SS)

LOCA

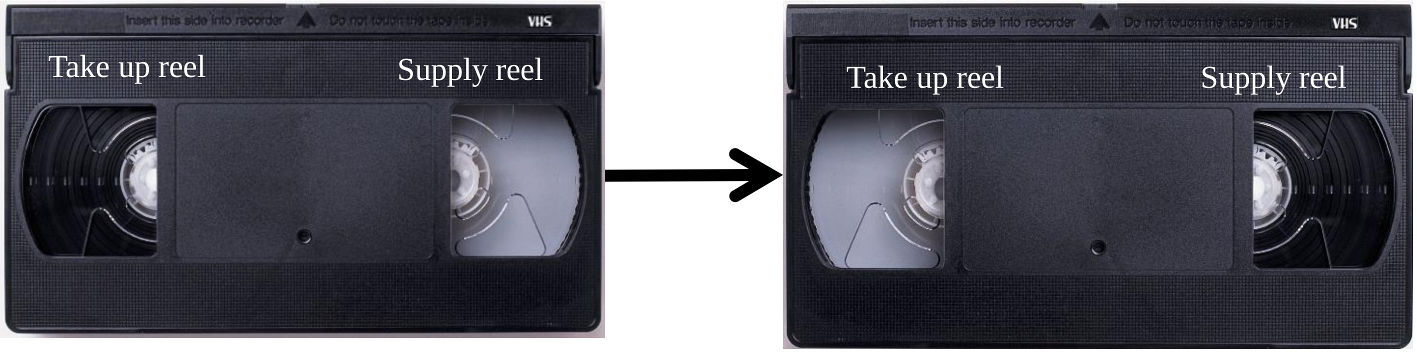
Multi-deck Input File

First Problem – UP15025

- On running a base case with one deck and restarting it with another, could not read the plot file
- Error message indicated the code could not read the plot file
  - This despite correctly recognizing the plot file format as ASCII at first
- Examination showed RELAP5-3D could not read the file because, when opened, it was already at the end of file

Multi-deck Input File

- Solution is to rewind it
  - Can be done when closing the file
    - Old code versions always rewound restart-plot file on closing
    - Split of restart and plot file failed to add rewind for plot file
  - Can be done by using open statement with proper keyword
    - (This keyword did not work with my Fortran compiler)
  - Can be done when opening with rewind statement
- The third solution may be safest
- The first was implemented for the benefit of plotting packages that may need it


Insert this side into recorder
Do not touch the tape inside
VHS
Take up reel
Supply reel
Insert this side into recorder
Do not touch the tape inside
VHS
Take up reel
Supply reel

Multi-deck Input File

Second Problem – UP 15024

- The PHISICS parameter study requires the RELAP5-3D perform several restarts, each changing the input model in a different way
- A multi-deck with a **base case**, **restart**, and **second restart** was tested, it failed
- The error was first recorded in kinetics input, but when a second test case was built, it failed in IHTMOD

The source of the trouble:

- The startup of the second deck (1st restart) writes an initial restart dump immediately after the dump of the user request
- This creates two dumps for one advancement number
- This creates issues reading the second restart

After first restart run

Time = 0.0
Time = 1.0
Time = 2.0
Time = 5.0
Time = 5.0
Time = 10.0

Multi-deck Input File

- Note, this problem does not occur on a second restart with separate runs and separate decks

Solution – Part 1

- When restarting, be certain there are not two records at the same time
- Backup to the beginning of the specified restart dump before writing the first restart dump of the restart run
  - This initial dump is at the same time as restart
  - It will overwrite the original dump at that time
- This was implemented in version 4.3.2, tested, works

Solution – Part 2

- Backfit into the code version 4.1.3 in use with PHISICS

After first restart run

Time = 0.0
Time = 1.0
Time = 2.0
Time = 5.0
Time = 10.0

Conform Strip to Plot File Format for MBINARY

- An exhaustive study by ANSALDO machine dependent format strip files showed
  - Three different strip formats in the last 20 years
  - Plot files have real(4) values, while strip has real(8)
  - Strip values cannot be more accurate than plot values
    - Makes sense to have real(4) on strip file
  - Loses backward compatibility
    - Backward compatibility does not extend very far
- User request made to simplify strip format by having just one for both plotting and stripping

Conform Strip to Plot File Format for MBINARY

- UP 15026: Match strip format to plot format
  - For machine dependent files only
  - So that legacy software can work properly
- Background
  - RELAP5-3D reads performs a contextual read of a plot file for 2 reasons
    - To position it at a restart time or advancement
      - Done in subroutine plotOpen of PLOTMOD
    - To strip it
      - Done in subroutine stripplot
  - All writes to machine dependent plot file done by subroutines contained in module PLOTMOD

Conform Mbinary Strip to Plot File Format

- Three PLOTMOD subroutines responsible for all plot file writing
- 5 kinds of records written on all plot/strip files
- Each begins with a keyword or symbol

key	Subroutine	Description
=	writePlotFileHdr	Title
plotinf	writePlotChanNames	Plot information about number of channels
plotalf	writePlotChanNames	Alphanumeric names of plot channels. Identify quantity to be plotted
plotnum	writePlotChanNames	Index in the array indicated by plotalf. Zero if it is a scalar quantity
plotrec	writePlotData	Record of floating point values of the variable given by the ordered pair (plotalf, plotnum)

Conform Mbinary Strip to Plot File Format

- Machine dependent binary also has 6th kind of record
  - One of these precedes every other record
  - A numerical record used to parity-check the file

- All writes to strip file made by subroutine **stripplot**, *not* **PLOTMOD**
  - This is a maintenance issue
  - It allowed the disparity to arise

- Strategy – use same write statements for both strip and plot files
  - Only for Machine Dependent format, not the others
- Requires new subroutine of machine dependent binary write statements

Conform Mbinary Strip to Plot File Format

- New subroutine named psWriteMB
  - stands for Plot/Strip Write Machine-dependent Binary
- Call arguments include
  - Unit number – plot file or strip file
  - Channel arrays – plot or strip
  - Data arrays – plot or strip
- Has numerous checks against incorrect calls
- Replaced all writes in writePlotFileHdr, writePlotChanNames, writePlotData and stripplot

Conform Mbinary Strip to Plot File Format

- Testing
  - Rename a machine-dependent binary strip file as a plot file
  - This file must open as a plot file and be successfully stripped
- Uses standard installation test case edMmbin.i
  - Copies edMmbin.st to stripmbstr.plt
  - Runs stripmbstr.i to successfully create stripmbstr.st
- Included in standard installation set

- NOTE: Doing this for other file formats might be worth consideration
  - ASCII, CSV

Data Management Improvement

- Many errors are created by insufficient handling of data between the point where it is created and its first usage.
  - Many user problems have resulted from this
- Examples:
  - memory leaks
  - hanging of a computer during parallel operations
  - destruction of data.
- When data is first allocated, it should be initialized.
  - A feature of FORTRAN 2003 allows this through a keyword in allocate statements.
- Pointers should be nullified upon creation.
- A systematic approach to handling data better from the outset is underway for RELAP5-3D.

Data Management Improvement - Rules

- 1. Array and pointer allocation must be tested before it is exercised
  - a. The size must make sense (positive and not overlarge)
  - b. The array or pointer must not already be allocated
  - c. For derived types, check the allocation status before proceeding to allocate components

- 2. Memory must be initialized whenever it is allocated one of two ways:
  - a. With a loop, or
  - b. With the FORTRAN 2003 “source” keyword on the allocate statement.

Data Management Improvement - Rules

- 3. Pointers must be nullified or assigned immediately upon creation.

- 4. Arrays and pointers must be deallocated before the run terminates.
  - a. The allocation status of arrays must be tested before deallocating.
  - b. The association status of pointers must be tested before deallocating.
  - c. Derived types must be deallocated from bottom up to prevent memory leaks.

Data Management Improvement - Implementation

- Two separate modules of allocation and deallocation routines for arrays have been written
  - In ALLOMOD, the call sequences are simple
  - ALLOCMOD adds call parameters that pinpoint the location of the call and provide an error message in case of failure

- Progress
  - All JUNMOD and VOLMOD, and modules beginning with the letters A-J have been upgraded to conform to the rules