

COMPLETION REPORT FOR THE COUPLED TRAC-M/PARCS CODE

R. Matthew Miller, Thomas J. Downar

School of Nuclear Engineering
Purdue University
W. Lafayette, IN 47907-1290

submitted to:

Division of Systems Technology
Office of Regulatory Research
Nuclear Regulatory Commission
Washington, D.C. 20555-001

Performed Under Subcontract to SCIENTECH, Inc.
Contract SC96-1026-001

August 1999

I. Introduction

The coupled TRAC-M/PARCS code utilizes a General Interface, which manages the mapping of property data and solution variables between thermal-hydraulics and spatial kinetics codes. To meet the requirements of this General Interface⁽¹⁾, separate data map routines for both TRAC-M and PARCS are utilized⁽²⁾⁽³⁾. The TRAC-M-Specific Data Map Routine (TDMR) functions as a secondary interface between TRAC-M and the General Interface, while the PARCS-Specific Data Map Routine (PDMR) functions as a secondary interface between PARCS and the General Interface. In this design, TRAC-M, the General Interface, and PARCS are executed as separate processes and communicate with each other through the use of message-passing protocols in the Parallel Virtual Machine (PVM) package.

The following section, Section II, describes a few of the important features of the coupled TRAC-M/PARCS code. Section III provides instructions on the installation procedure, and Section IV describes both the input requirements and the execution procedure of the TRAC-M/PARCS code. Section V discusses results of a coupled execution using the OECD PWR MSLB Benchmark model. Section VI contains a description of some improvements which could be made to the coupled TRAC-M/PARCS code in the future.

II. Features of the TRAC-M/PARCS Code

The coupled TRAC-M/PARCS code utilizes an internal integration scheme in which the solution of the system and core thermal-hydraulics is obtained by TRAC-M and only the spatial kinetics solution is obtained by PARCS. In this scheme, PARCS utilizes the thermal-hydraulics solution data (e.g., moderator temperatures/densities and fuel temperatures) calculated by TRAC-M to incorporate appropriate feedback effects into the cross-sections. Likewise, TRAC-M takes the space-dependent powers calculated by PARCS and solves for the heat conduction in the core heat structures.

The temporal coupling of TRAC-M and PARCS is explicit in nature, and the two codes are locked into the same time step. For this implementation, the TRAC-M solution leads the PARCS solution by one time step. Specifically, the advancement of the time step begins with TRAC-M obtaining the solution to the hydrodynamic and heat conduction field equations using the power from the previous time step. The property data obtained from this solution is then sent to PARCS and the power at the current time step is computed.

TRAC-M and PARCS maintain control of different aspects of the calculation procedure. The following subsections outline these controls for each code.

II.A. TRAC-M Control

As mentioned, TRAC-M and PARCS are locked into the same time step. For this implementation, the time step size is computed in TRAC-M and passed to PARCS through the inter-

face. In addition, the normal termination of the coupled TRAC-M/PARCS code is controlled by TRAC-M. Edits to both the TRAC-M and PARCS restart files are synchronized, allowing the user to perform a restart calculation with the coupled TRAC-M/PARCS code.

For reactor control, TRAC-M manages the control rod scram logic and communicates the corresponding trip signal to PARCS. In addition, the boron transport calculation is performed by TRAC-M, and the resulting space-dependent boron concentrations are sent to PARCS.

II.B. PARCS Control

When PARCS is executed within the framework of the coupled code, the thermal-hydraulic solution of PARCS is bypassed. PARCS then maintains control of all other features in the code except for those mentioned in Section II.B (i.e., the control rod scram and boron perturbation logic of PARCS is bypassed). However, PARCS controls a boron criticality search within the framework of the coupled TRAC-M/PARCS code if it is specified in the PARCS input. The critical boron concentration calculated by PARCS will be reflected throughout the thermal-hydraulic system on the time step immediately following.

III. Installing TRAC-M/PARCS

(1) `type: perl Install`

The PERL script will execute, and prompt the user for whether PVM is already installed on the system. If it is not, the script will install the PVM package on the machine. Next, the script will ask whether XMGR is installed on the system already. If not, the script will install it. The next question makes sure that the machine architecture as defined by the PVM function `pvmgetarch` is correct. The user is prompted for the directory containing the XMGR library `libacegr_np.a`, which is required for real-time plotting of PARCS data. (It should be noted that for `res30.nrc.gov`, this should be set to `‘/local2/local/xmgr/lib/’`). Finally, the script asks for the installation directory, i.e., the directory in which to install the coupled TRAC-M/PARCS code. After all questions have been answered, the script will install the General Interface, PARCS, and TRAC-M for the specific machine architecture.

(2) Decide whether the QATP should be run.

Once installation of the codes is complete, the script provides the option of automatically running the test plan as defined in the QATP document⁽⁴⁾. In order to run the QATP, the PVM daemon must first be running on the system. The easiest way is to get to the PVM console by typing (in a separate window): `pvmd &` The script will prompt the user to start PVM whether it is running or not and wait until any key is pressed. The QATP then executes, and results are saved in the `TRAC-M/run/QATP/CalcExp/Output/` and `TRAC-M/run/QATP/FuncEval/Output/` subdirectories for comparison with expected output.

The following executables should be used when running the coupled TRAC-M/PARCS code:

```
GI:      <install_dir>/GI/nrc-v1.02/run/<machine_architecture>/gi.x
PARCS:   <install_dir>/PARCS/nrc-v1.04/bld/<machine_architecture>/parcs.x
TRAC-M:  <install_dir>/TRAC-M/run/tracm3dk.x
```

IV. Building Input and Running TRAC-M/PARCS

The following subsections provide detail on both setting up the input decks and running the coupled TRAC-M/PARCS code.

IV.A. TRAC-M Input

In order to couple a TRAC-M calculation with PARCS, the only additional input requirements are two additional NAMELIST variables, `itdmr` and `itmrp`. By default, these values tell TRAC-M that the current calculation is stand-alone, i.e., not coupled with PARCS. By defining `itdmr=1` in the TRAC-M input file, the hooks to the TDMR are activated during the calculation automatically. If this variable is either not present or set to 0 in the TRAC-M input file, the calculation will not be coupled with PARCS. The other variable, `itmrp`, defines the verbosity with which the TDMR will operate during the calculation. The default value is 0, which allows only the vital edits regarding the operation of the TDMR to be printed to the screen and to the output files. By setting `itmrp=1` in the input file, the TDMR will print additional information during the calculation. This option was provided for debugging purposes only, and is not recommended for normal use.

All power profile options defined in the TRAC-M input are ignored, with the exception of the radial power distribution within the fuel rod, which is required for properly distributing the power sent from PARCS within each fuel rod. Additionally, the material definition of each fuel rod should be specified in order to accurately provide PARCS with the necessary fuel temperature data. More specifically, the user should make sure a gas gap is defined in the heat structures coupled to PARCS, otherwise the results may be inaccurate.

IV.B. PARCS Input

PARCS was previously coupled to RELAP5, and, as such, all the required input changes have already been made to PARCS. PARCS version 1.04, however, has the additional feature of coupled to either RELAP5 or TRAC-M using the same executable. The `EXT_TH` card in the PARCS input file has an additional (optional) input. By inserting the following card into the PARCS input file, the user is specifying that the current run is coupled, uses the mapping input table file named 'MAPTAB', and is coupled to TRAC-M:

```
EXT_TH      MAPTAB      TRAC
```

Similarly, by using the following line, the user is coupling to RELAP5:

```
EXT_TH      MAPTAB      RELAP
```

If the code is not identified in the PARCS input file, PARCS defaults to coupling with RELAP5. The mapping input table is briefly described in the next section (the filename specified can be changed to the user's preference, so long as the file exists).

IV.C. Mapping Input

The input file which is read by the PDMR may be named anything, as long as the filename is specified on the EXT_TH card in the PARCS input file. It is recommended that this file exist in the same directory from which PARCS is being executed. Four card options are allowed: "TABLE1", "TABLE2", "TRIP" and "DOPL". These input cards are preceded by a "%" in the first column (e.g., %TABLE1), and comment lines are preceded by a "*" in the first column (e.g., * comment). The lines following an input card contain free format data relating to that card, where the input processor assumes that if the first column does not contain either "%" or "*", then the line being read corresponds to actual data. Subsequent lines of data will continue to be processed for the previously-read input card until a new card is input. For example, if card "TABLE1" is read, the input processor will assume that the lines which follow correspond to this card until another card is detected, such as "TABLE2". It should be noted that it is not required that the user input only one TABLE1 card and only one TABLE2 card. If these cards are entered multiple times, the input processor will simply append the data that follows to the data which have already been read. However, for the "TRIP" and "DOPL" cards, only one line of data needs to be read following these cards. Thus, if multiple "TRIP" or "DOPL" cards are entered, the input processor will simply overwrite the previously read data with the new data.

A description of the mapping input table format is provided in Section II.A of the addendum to the PDMR SQA documents⁽⁵⁾. The mapping input file used for the OECD PWR MSLB Benchmark calculation is provided with this package at the following location:

```
<install_dir>/PARCS/nrc-v1.04/run/mslb/193hs/maptab-linw.map
```

IV.D. Running TRAC-M/PARCS

Prior to executing the coupled TRAC-M/PARCS code, the PVM daemon, which will manage the data communication, must be started by typing: `pvmd &`

Once the PVM daemon has been started, the execution of the coupled TRAC-M/PARCS code can proceed by initiating three processes: TRAC-M, General Interface, and PARCS. There are two techniques for accomplishing this task. The first involves opening three windows and performing the following:

window 1:

- (1) `cd` to the TRAC-M directory where the input file exists.
- (2) copy the input file to `tracin`
- (3) type: `tracm3dk.x`

window 2:

- (1) `cd` to the GI directory where the executable file exists.
- (2) type: `gi.x`

window 3:

- (1) `cd` to the PARCS directory where the input file exists.
- (2) type: `parcs.x < input_file`

The second technique involves utilizing a shell script, which would look similar to:

```
#!/bin/sh
cd GI_directory
gi.x > gi.screen &
cd PARCS_directory
parcs.x < input_file > parcs.screen &
cd TRAC_directory
cp input_file tracin
tracm3dk.x > trac.screen
```

It should be noted that following the completion of TRAC-M/PARCS, the PVM daemon will continue to operate in the background and the user can perform additional TRAC-M/PARCS executions without starting the daemon. If no further coupled TRAC-M/PARCS executions will be performed, the PVM daemon should be killed. This can be done by typing: `pvm`, which will bring up the PVM console. In the console, type: `halt` This will stop the daemon and return the user to the command prompt. At this point it may be necessary to remove temporary PVM files by typing: `rm /tmp/pvm*`

For further information on PVM, visit the PVM web site:

http://www.epm.ornl.gov/pvm/pvm_home.html

V. Coupled Results

The OECD PWR MSLB Benchmark exercise 2 model is used as the basis for testing the coupled TRAC-M/PARCS code. The TRAC-M model was built according to the specifications provided in the benchmark. The model was made to be nearly identical to a RELAP5 model based on the same benchmark for purposes of comparison. The same PARCS input file was used for executing with both RELAP5 and TRAC-M, providing a solution which should differ only

due to the differences in the thermal-hydraulic solution. Both results have been submitted to the benchmark providers as the final solutions to exercise two based on the RELAP5/PARCS and TRAC-M/PARCS coupled code systems. Figure 1 shows a comparison of the transient results.

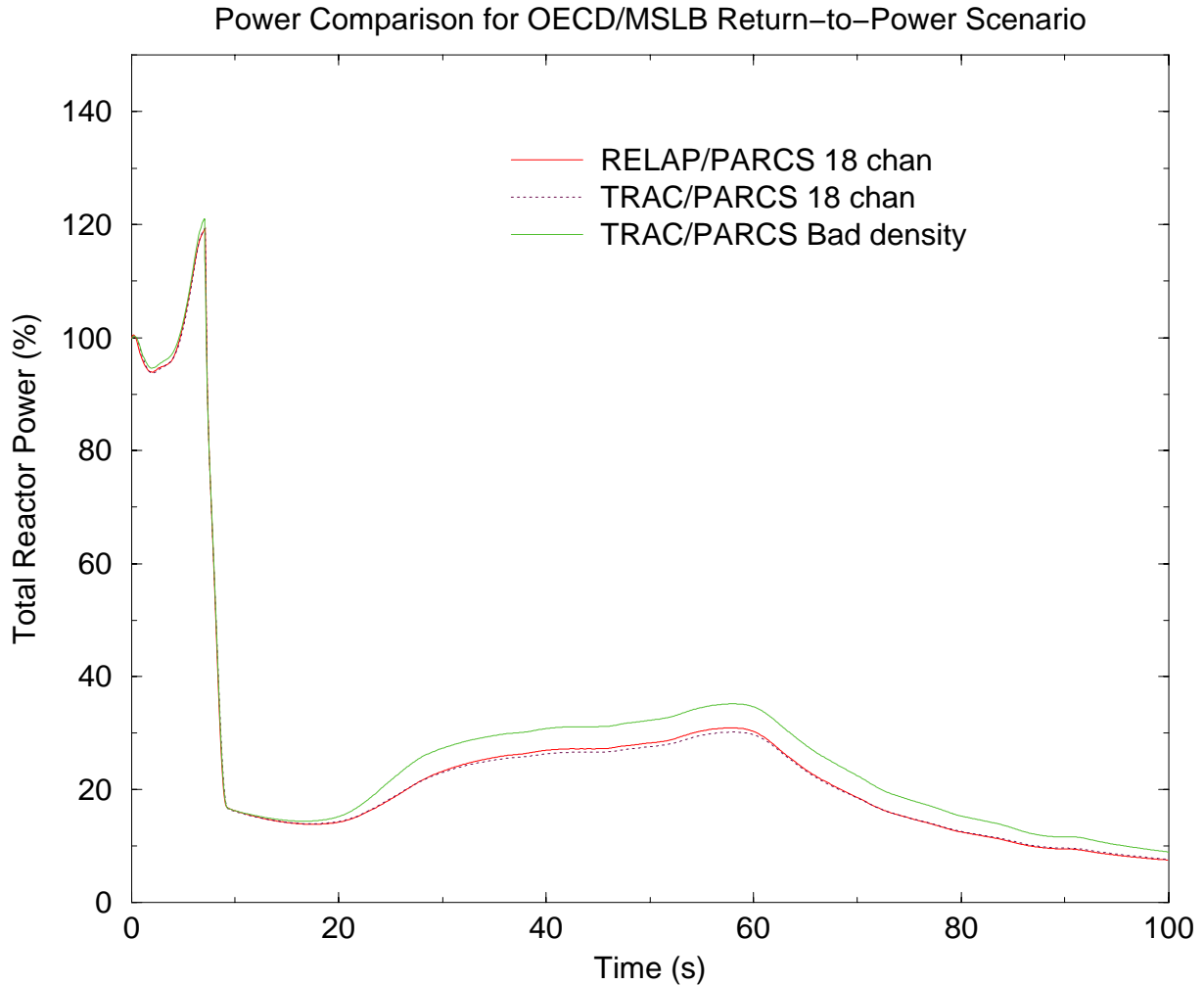


Figure 1: OECD PWR MSLB Benchmark Transient Calculation Results

There are two TRAC-M/PARCS solutions shown in Figure 1 because during the analysis of the results, it was found that the density correlation internal to TRAC-M is inaccurate at fluid conditions far from normal PWR operating conditions. The results labelled 'bad density' show the solution with an uncorrected density correlation, while the results labelled '18 chan' demonstrate the after correcting the density correlation, the results agree almost completely with those from RELAP5/PARCS. This result demonstrates the successful completion of TRAC-M with PARCS.

VI. Future Improvements

The following improvements could be made to the coupled TRAC-M/PARCS code in the future:

VI.A. Logic for Control Rod Movement

Since TRAC-M is a systems code, it is natural to allow TRAC-M to control the rod positions based on user input. Scenarios such as power maneuvering and/or load following require that a more intelligent communication protocol for rod movement be defined than is currently implemented in TRAC-M/PARCS. Rod criticality searches driven by TRAC-M could also be made feasible by allowing control rod positions to pass from PARCS to TRAC-M. Work toward implementing this feature into TRAC-M/PARCS is already being performed, and will be provided with the next TRAC-M/PARCS transmittal.

VI.B. Support for BWR CHANNEL Components

TRAC-M is currently under development, and new features are being added continuously. Since the beginning of this project, support for BWR CHANNEL components has been added to TRAC-M, although this component is currently not supported by this version of TRAC-M/PARCS. Implementing support for this component allows the coupling of BWR models to PARCS, which will be required in the future. Like control rod movement logic, work is already underway to implement this feature into the next version of TRAC-M/PARCS.

VI.C. Mapping Input Generalization

In the current implementation, the user is required to physically determine the mapping between thermal-hydraulic cells/heat structures and neutronic nodes. This can become somewhat tedious for models which involve very fine spatial discretization. Thus, in the future, it might be beneficial to examine more automated techniques (e.g., using a graphical user interface) to produce the necessary mapping functions with minimal user input.

VI.D. Simultaneous Execution

In the current implementation, PVM barriers are utilized during the initialization of the TDMMR, General Interface, and PDMR, which synchronize the three processes such that the communication paths between the processes can be established. The design of these coding sections restricts the number of simultaneous TRAC-M/PARCS executions to one. More precisely, it is possible for many simultaneous jobs to be initiated, but there is no guarantee of safe execution. In addition, the use of the PVM barrier in this coding section limits the fault detection of the three processes because there is no time associated time limit for how long a process should wait on a barrier. Thus, implementing a more sophisticated barrier could allow for the safe execution of several simultaneous TRAC-M/PARCS jobs, and could provide better fault detection.

VII. Summary

The coupled TRAC-M/PARCS code functions as three separate processes (TRAC-M, the General Interface, and PARCS) and utilizes the PVM package to manage the transfer of data between the processes. An internal integration scheme is used for the spatial coupling, and the temporal coupling is explicit in nature.

The testing of the coupled code using the OECD PWR MSLB Benchmark exercise 2 model for TRAC-M and the appropriate spatial kinetics model compares well with results obtained with a nearly identical RELAP5/PARCS model, demonstrating the successful coupling of TRAC-M and PARCS.

This work has resulted in the coupling of TRAC-M with PARCS, thus providing TRAC-M with the capability to analyze transients using three-dimensional spatial kinetics. In the current design, there are certain limitations to the applicability of this coupled code, as discussed in Section VI. Nonetheless, most of the proposed modifications to remove these limitations would require minimal effort, and would provide a more robust coupled TRAC-M/PARCS code.

VIII. References

1. D. Barber and T. Downar, "Software Requirements Specification for the General Interface in the Coupled Code," Technical Report, PU/NE-98-8, Purdue University, (1998).
2. R. Miller, D. Barber, and T. Downar, "Software Requirements Specification for the TRAC-M-Specific Data Map Routine in the Coupled TRAC-M/PARCS Code," Technical Report, PU/NE-98-22, Purdue University, (1998).
3. D. Barber and T. Downar, "Software Requirements Specification for the PARCS-Specific Data Map Routine in the Coupled TRAC-M/PARCS Code," Technical Report, PU/NE-98-11, Purdue University, (1998).
4. R. Miller and T. Downar, "Qualification Assessment Test Plan for the TRAC-M-Specific Data Map Routine in the Coupled TRAC-M/PARCS Code," Technical Report, PU/NE-98-23, Purdue University, (1998).
5. R. Miller, D. Barber, and T. Downar, "Addendum to the Software Qualification Assessment Documents for the PARCS-Specific Data Map Routine," Technical Report, PU/NE-98-25, Purdue University, (1998).