2002 RELAP5 Users Seminar
Park City, Utah
September 4-6, 2002

SAND2002-2440C

Coupling of MELCOR to Other Codes under an Executive Program using PVM Message Exchange

R. K. Cole, Jr.
Modeling and Analysis Department 6415
Sandia National Laboratories*
P. O. Box 5800
Albuquerque, NM 87185-0739

Abstract

Sandia National Laboratories is continuing to develop the MELCOR simulation code for the U.S. Nuclear Regulatory Commission (USNRC). MELCOR is currently the USNRC's tool of choice for integrated analyses of the progression of severe accidents in light water reactors.

The capabilities of MELCOR have recently been extended to allow it to participate in the coupling of multiple tasks using PVM message-passing software under an executive program developed at the Idaho National Engineering & Environmental Laboratory (INEEL). The coupled tasks can include one or more calculations performed by MELCOR, by RELAP5-3D©, and/or by other codes compatible with the executive program. One possible application is the use of MELCOR rather than CONTAIN as a containment module coupled to RELAP5-3D©. MELCOR's capabilities in application to containment response are quite comparable to those of CONTAIN, and MELCOR continues to be under active development, whereas CONTAIN has been placed in maintenance mode.

The executive program controls the exchange of data between—and the generation of output by—all coupled tasks. All variables to be exchanged are defined as part of input to the executive program, in the "native language" of each task sending or receiving data. The interface in MELCOR is based on an extension of the existing "control function" utility.

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

1. BACKGROUND

In many applications of current interest, the computational domain can be divided into two or more sub-domains with relatively limited coupling. If the boundary conditions can be matched, separate calculations can then be performed for each sub-domain. These calculations can be performed either by different computer codes chosen to best represent the important phenomena in each sub-domain, or by separate instances of the same code.

One such application involves the use of the CONTAIN code [1] as a containment model with RELAP5 [2] or RELAP5-3D© [3]. Although CONTAIN is no longer under active development, the MELCOR code [4] provides an alternative. In addition, because MELCOR models a wider range of phenomena than does CONTAIN, the possibilities for its use in coupled calculations are greater.

The major challenges for coupled calculations are (1) implementing the exchange of boundary condition data between codes with data structures that may be very different and (2) coordinating the advancement of the coupled system. An executive program [5] was developed at INEEL to handle these issues in an extremely flexible way, using the Parallel Virtual Machine (PVM) message exchange protocol [6].

We have extended the capabilities of MELCOR to permit it to participate in the coupling of multiple tasks under this executive, thus allowing it to be used as a containment module with RELAP5-3D. Other applications are clearly possible.

1.1 The Executive Program

The executive program reads a simple input file. Based on the contents of that file, it configures a "virtual machine" containing one or more host processors and starts "pvm daemons" on each of them, to mediate the message exchange. It then spawns the requested tasks and sets up, through PVM messages, the definition of all variables to be exchanged between these tasks. Finally, it coordinates the advancement through time. This includes calling for the exchange of variables directly between sibling tasks, controlling the generation of output (in particular, of plot and restart dumps), and shutting down the virtual machine on termination. During setup and advancement, the executive also monitors the system for high-level errors, and initiates a graceful termination of all tasks if necessary.

For each spawned task, setup information from the parent process (the executive) defines any and all messages to be exchanged between sibling tasks. The definition of each message includes a unique "message identifier", the direction of transfer (send or receive), the task identifier ("tid") of the other process, and the contents of the message. Control messages issued by the executive during the advancement phase trigger the exchange of variables, the generation of output, and the termination of calculations.

Within a given process, the contents of each message are defined in its own “native language.” This eliminates any need for further information about any sibling task. In particular, there is no need to know anything about its internal data structure. “Incoming” variables simply arrive on schedule, packed into messages from designated tids, with message identifiers that define the contents. “Outgoing” variables are similarly packed into messages and sent out to designated tids.

Because of the need to diagnose errors and the difficulty of debugging problems in coupled tasks, there is an elaborate system of returned “handshakes” to confirm receipt of all messages. This applies to setup and control messages from the executive and to messages containing variables exchanged between siblings. If there is an error involving any message, or if an individual task fails, all processes will be shut down as gracefully as possible. Failure to receive an expected message (setup, control, data exchange, or confirmation of receipt of a message sent) within a specified maximum wait time is treated as an error.

The executive program allows for “synchronous” and “asynchronous” exchange of data between tasks. For “synchronous” exchange, the processes are forced to advance using identical time steps, with data exchange on every time step. For “asynchronous” exchange, advancement times are matched and data exchanged only at predefined time intervals. Our MELCOR implementation permits only “asynchronous” coupling.

In addition, the executive program allows for other types, including “semi-implicit” coupling. The latter option involves exchange of derivative information (used to deal with problems of numerical stability). Our MELCOR implementation permits only “explicit” coupling.

1.2 The MELCOR Program

Although MELCOR was conceived in 1982 as a tool for Probabilistic Risk Assessment (PRA), the modeling approach has always been more mechanistic than parametric and, over the years, MELCOR has evolved as a flexible, mechanistic modeling tool, with increasingly best-estimate capabilities.

For example, MELCOR models two-fluid hydrodynamics, using a full six-equation set and equations of state for water and noncondensible gases. It also models heat transfer by convection and/or radiation to boundary structures, including one-dimensional conduction within the structure, condensation and evaporation at its surfaces, and ablation.

Additional models treat the performance of engineered safety features, core response and degradation, ex-vessel debris/concrete interactions, radionuclide release, and vapor and aerosol transport. The most recent code release is version 1.8.5; further development is continuing, with consolidation of knowledge gained from experiments and from other code projects.

2. IMPLEMENTATION OF COUPLING IN MELCOR

“MELCOR” is really a pair of programs, MELGEN and MELCOR. MELGEN is run first to read an input file containing a detailed description of problem nodalization and initial conditions. If no errors are encountered, it writes the initial dump in a restart file. The actual advancement of a calculation is then performed by MELCOR, starting from either the initial restart dump written by MELGEN or from some later point that has been saved in the restart file.

Thus, only MELCOR has been coupled to the executive. This allows problem input to be constructed and tested in MELGEN before coupling is attempted. The implementation also allows the running of preliminary “shakedown” MELCOR in standalone mode, with constant values for all boundary conditions that are destined to be defined by a coupled task.

2.1 MELCOR “Control Functions,” and Exchange of Data with Other Tasks

Data calculated and sent out by one task are ordinarily used to define sources or other boundary conditions in the receiving task. MELCOR’s existing control function utility provides a natural interface for coupled data, and allows a user interface that is consistent with existing MELCOR practice. An added benefit of this approach is that only minimal changes are required to the MELCOR source code beyond those needed to acknowledge the existence of the executive.

MELCOR allows the definition of “control functions” as part of MELGEN input. Each such function is associated with a user-defined function number, and its definition includes the type of function (its functional form), and the elements of the time-dependent MELCOR database that serve as arguments. Linear scaling of the values of the arguments and of the evaluated function is also allowed.

The list of function types includes mathematical and fortran functions such as MAX, ABS, and EXP; logical comparisons such as .GT. or .EQ.; numerical integrals and derivatives; table look-ups; and others. There is an extensive list of variables that are available for use as control function arguments. Any of these can be referenced in user input by a mnemonic name. Examples include “CVH-P.107”, the pressure in control volume 107; “FL-VELLIQ.342”, the velocity of the liquid in flow path 342; and HS-TEMP.1071004, the temperature of node 04 of structure 10710. In each case, the leading characters in the name identify the “package” (major computational module) in MELCOR that owns the data.

Each function is evaluated on every MELCOR time step, based on the current value(s) of its argument(s). The value of the function becomes part of the database, and is accessible for use in other models. Any such usage involves only reference to the function number as part of the input for the model. One common use is in the definition of sources or other boundary conditions. For example, mass and energy sources or sinks in control volumes, velocities in flow paths, and surface temperatures or heat fluxes on structure surfaces may all be specified in this way.

2.1.1 Sent data

It appears that all variables calculated by MELCOR whose values might be required as boundary conditions in a coupled task are already available as control function arguments. (There would be no great difficulty in extending the list of available arguments, if this should be desired.)

For outgoing message, the new interface simply assembles data for all variables to be included by using the existing internal protocol for evaluating control function arguments. The values are then packed into appropriate PVM messages and sent to the tid associated with the associated receiving task.

2.1.2 Received data

We believe that all sources and boundary conditions that one might wish to define in a MELCOR calculation may be specified in MELGEN input through reference to control functions. (If exceptions are found or additional flexibility is desired, it would be a straightforward task to add the capability to link additional variables to control functions.)

Therefore, incoming data are used to define control function values, and we have defined a new type of control function, ‘PVM-MSG’, to implement the connection. It allows no arguments, and requires specification of an initial value. The initial value can be changed only through PVM messages from another process. Scaling and limits can be applied to the raw value received, which can be useful for such things as conversion of units. MELGEN input might look like

```
CF8000300    'Liq Lev, CV200' PVM-MSG 0 1.0 0.0 * Need not be “1.0 0.0”
CF8000301    -15.0     * initial value (m) from MELGEN output, cntmnt task
CF8000302    . . .     * optional limits record
```

with control function 80003 used to define a liquid level in the receiving calculation.

2.2 Data Exchange, Time Advancement, and Output Control

MELCOR responds to requests from the executive for exchange of variables, and for the generation of text edits, plot dumps and restart dumps, at precisely specified times. This requires the advancement hit these times exactly. The capability to match one or more pre-specified times is already available through MELCOR input, using EXACTTIME input records. This capability was simply extended to add the time of the next "appointment" with the executive to the list.

The executive program calls for exchange of variables between spawned tasks at the initial time. We have implemented the interface in such a way that this first exchange takes place *before* the initial edit in MELCOR. Thus, while the edit from MELGEN shows the user-defined initial values of PVM-MSG control functions, the first MELCOR edit shows the initial values actually passed from sibling tasks.

Appropriate flags are set in response to requests from the executive for text, plot, or restart output, and MELCOR input (or internal logic) may result in generation of additional output. A final restart dump and text edit will be generated at the end of a calculation, and we have chosen to generate a corresponding plot point.

2.3 Other Input Options

Two potentially useful options are available, and can be invoked through MELCOR input on a new PVMOPT input record.

One directs that a detailed dialog describing message exchanges (expected and actual) during execution of MELCOR be included in the diagnostic file. It is intended as an aid to debugging and, because the volume of messages is large, should only be invoked for short checkout runs.

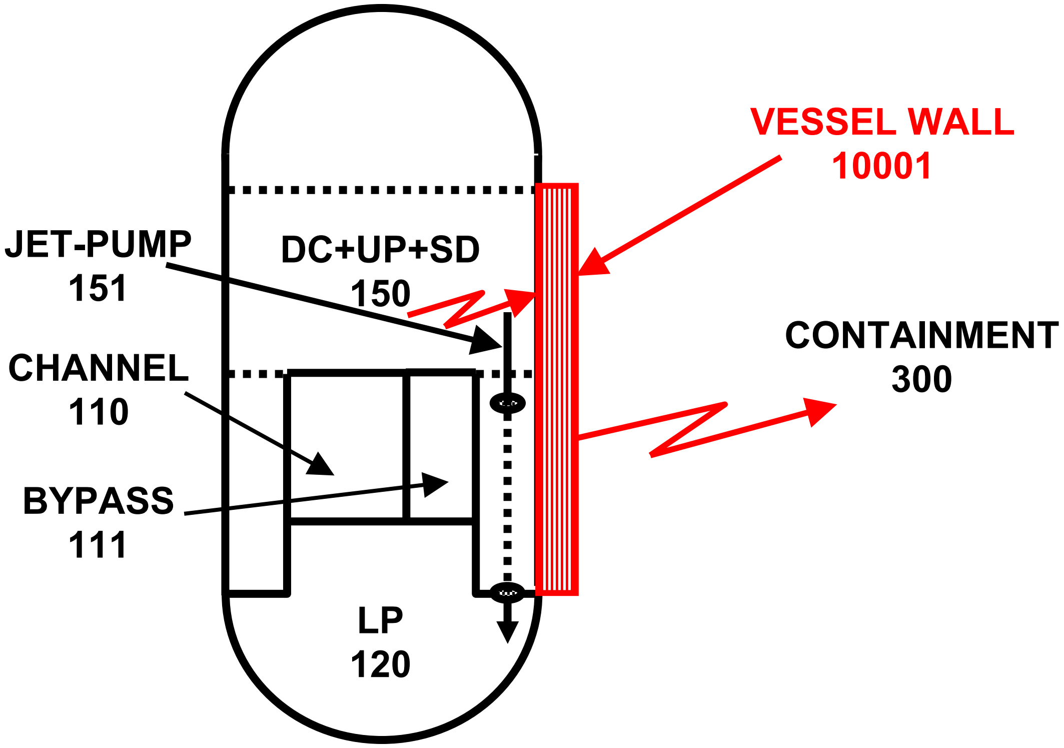
The second controls the encoding to be used for data exchanged in PVM messages. By default, "native" encoding is used, but the option of "XDR" encoding is available for possible use on heterogeneous systems.

3. An Example of MELCOR-to-MELCOR Coupling

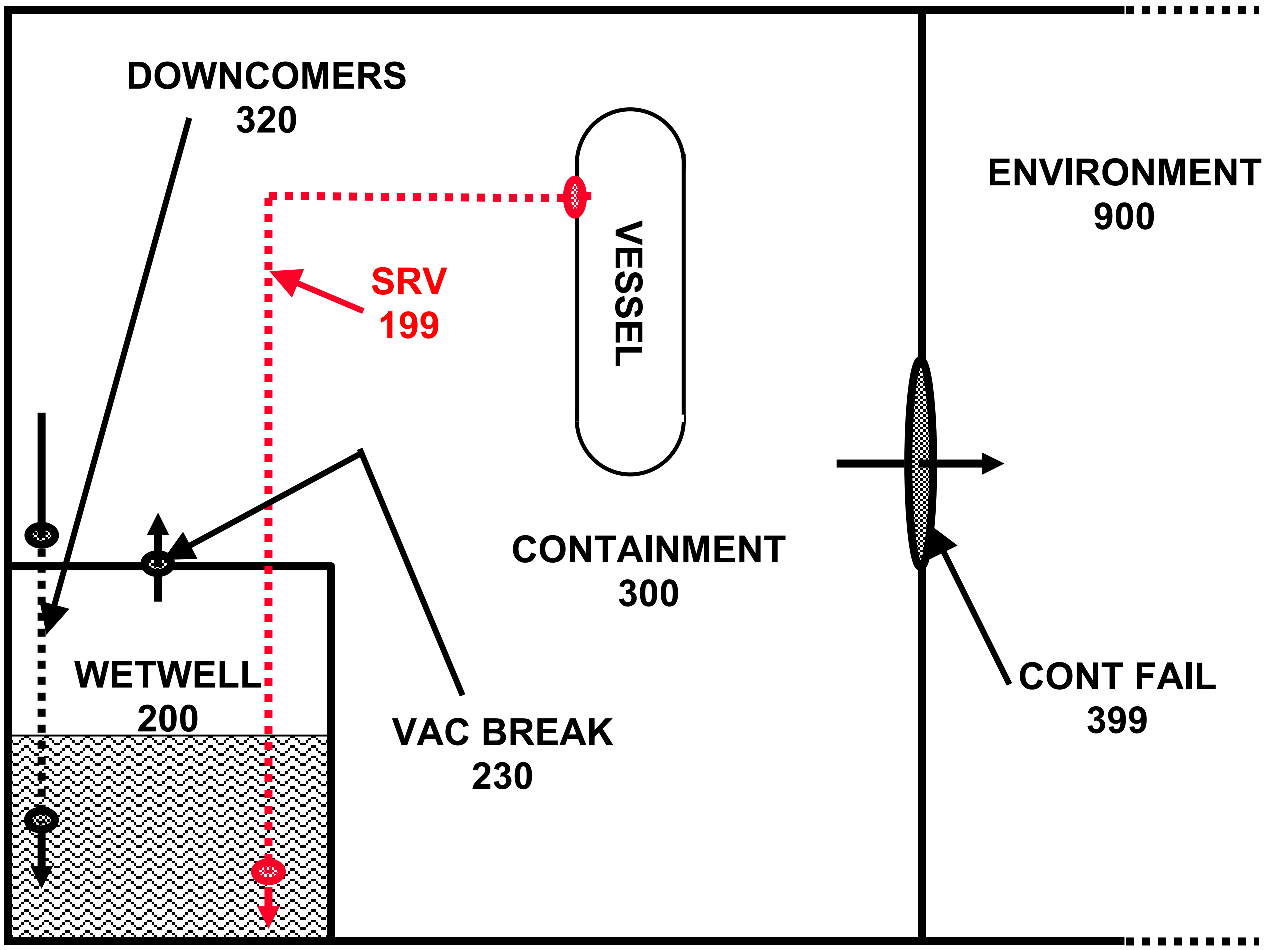
This example is based, in part, on the example calculation that is distributed with MELCOR. The scenario involves the loss of all water sources in a BWR after shutdown, without depressurization. In the full simulation, decay heat boils away the water, uncovering the core. Oxidation of the exposed Zircaloy then generates hydrogen within the vessel. It is actually Exercise 4 from the 2001 MELCOR Workshop [7] which, in the interest of simplicity, does not employ the core response (COR) or radionuclide behavior (RN) packages, and terminates the calculation before the vessel fails. (In this simplified version, a heat source is used to simulate the decay heat and mass/energy sources to simulate the oxidation of Zircaloy and the production of hydrogen.)

The nodalization is exceedingly simple. The nuclear steam system (NSS) contains four control volumes and the containment two, while a seventh volume models the environment as shown in the following two figures. The following figures identify the control volumes and most of the flow paths in the example calculation. (Core inlet and outlet flow paths are not shown.) There are also thirteen heat structures, but only the vessel wall is shown, as it is the only one that will be involved in coupling the two calculations.

Our test case splits the simulation into two sections, corresponding to the two parts of the nodalization, and referred to as "vessel" and "cntmnt", respectively. This separation seems quite natural, and the two parts are coupled only by flow through the Safety Relief Valve (SRV) and by heat loss through the vessel.



Nodalization of Nuclear Steam System ("vessel")



Balance of Nodalization ("cntmnt")

3.1 Vessel Heat Loss

We chose to model the vessel wall, HS10001, within the "vessel" simulation. The full conduction equation is solved there, including appropriate boundary conditions on the interior surface. The outer boundary condition in "vessel" is a specified heat flux, with a value calculated by "cntmnt" and used to define the value of a PVM-MSG control function in "vessel."

The surface temperature of the vessel in "cntmnt" is defined to match the temperature calculated by "vessel" for the outside of the wall, and used to define the value of a PVM-MSG control function in "cntmnt," and the heat flux from that surface is calculated based on conditions in containment. (In fact, MELCOR does not currently allow direct specification of a temperature boundary condition for a heat structure surface in communication with a volume. It is therefore necessary to use the data from “vessel” to define the temperature of the *inner* surface of a very thin “vestigial” vessel with negligible thermal resistance or thermal mass.)

The effect is that “vessel” sends “cntmnt” an outer boundary temperature and receives an outer boundary heat flux in return. The executive input includes (in the section where messages are defined) the records

```
    vessel  sends      cntmnt   . . .   HS-TEMP.1000105
    cntmnt  receives   vessel   . . .   CFVALU.80008
    . . .
    cntmnt  sends      vessel   HS-QFLUX-ATMS-R.10001  . . .
    vessel  receives   cntmnt   CFVALU.80001           . . .
```

During the setup process for the coupled run, the executive directs “vessel” to send the outer surface temperature, HS-TEMP.1000105 to “cntmnt,” where it is received as the value of PVM-MSG control function 80008 and used to impose the temperature boundary condition. It similarly directs “cntmnt” to send the outer surface heat flux, HS-QFLUX-ATMS-R.10001, to “vessel,” where it is received as the value of PVM-MSG control function 80001 and used to impose the heat flux boundary condition.

3.2 SRV Flow

Coupling the mass and energy flow through the SRV (flow path 199) presents a somewhat more subtle problem. One approach would be to use the wetwell pressure calculated by “cntmnt” as a back pressure in “vessel’, and introduce the resulting mass and energy calculated by “vessel” as sources in “cntmnt.” This is inadequate for several reasons:

1. These sources would be introduced directly into the atmosphere of the target volume (CV200). The effects of cooling and condensation during passage through the wetwell suppression pool would be completely missed;

2. Sources in control volumes are not included in calculating the characteristic velocities on which Reynolds numbers used in heat transfer correlations are based, and

3. If reverse flow of fluid, from “cntmnt” to “vessel,” were to occur, its composition and enthalpy could not be correctly represented.

An alternative approach—and one that we favor—is to include the connecting flow path in both sub-domains, each connecting to a boundary volume for which all properties are defined to match those in the other sub-domain. Either task can then calculate the flow through the path (in terms of the velocity and, when a valve is involved, the fraction of the area that is open), and impose it on the flow path in the other task. This will capture the effects of the suppression pool and include the flow in Reynolds number calculations. Further, the appropriate donor properties would be used for all flows, regardless of the flow direction.

This is easily implemented in a MELCOR task. MELGEN allows definition of “time-specified volumes” with all necessary thermodynamic properties defined by control functions. It also allows definition of “time dependent flow paths” with the flow velocities defined by control functions (under the assumption of no slip). Similar capabilities are available in RELAP5, described as “time dependent volumes” and “time dependent junctions.

We chose to calculate the SRV opening area and flow velocity (in FL199) in the “vessel” simulation, and impose these on a duplicate flow path in “cntmnt.” The executive input includes the records

```
vessel  sends      cntmnt   CFVALU.199     FL-VELVAP.199
cntmnt  receives   vessel   CFVALU.80006   CFVALU.80007
```

During the setup process for the coupled run, the executive directs “vessel” to send the open fraction for the SRV, `CFVALU.199`, and the velocity of fluid passing through it, `FL-VELVAP.199`, to “cntmnt.” These are received as the values of control functions 80006 and 80007, respectively, and are used to control a “slave” valve and to define the flow velocity.

We knew that no reverse flow would occur in the example calculation. Therefore, “vessel” used a simplified boundary condition for the volume representing the containment end of the SRV flow path. Only the pressure was matched, set equal to the local pressure (surface pressure plus fluid head) at the exit of the SRV line in volume 200 as calculated by “cntmnt.”

As actually run, the pressure, pool surface elevation, and pool density in “cntmnt” were transmitted to “vessel.” The executive input includes the records

```
cntmnt  sends      vessel   . . .    CVH-P.200
vessel  receives   cntmnt   . . .    CFVALU.80002
. . .
cntmnt  sends      vessel   CVH-LIQLEV.200   CVH-RHOP.200
vessel  receives   cntmnt   CFVALU.80003     CFVALU.80004
```

The received variables were then used to evaluate the local pressure as a control function in the “vessel” task. We could equally well have evaluated the local pressure as a control function in “cntmnt” and sent that single variable to “vessel.”

In contrast, a full copy of the donor volume (CV150) was included in the "cntmnt" task, with all properties "slaved" to those calculated in "vessel." This required consideration of many more variables, and the executive input includes the records

```
vessel  sends     cntmnt  CVH-P.150       CVH-TVAP.150    CVH-PPART.3.150
cntmnt  receives  vessel  CFVALU.80001    CFVALU.80002    CFVALU.80003
vessel  sends     cntmnt  CVH-X.6.150     CVH-X.4.150
cntmnt  receives  vessel  CFVALU.80004    CFVALU.80005
```

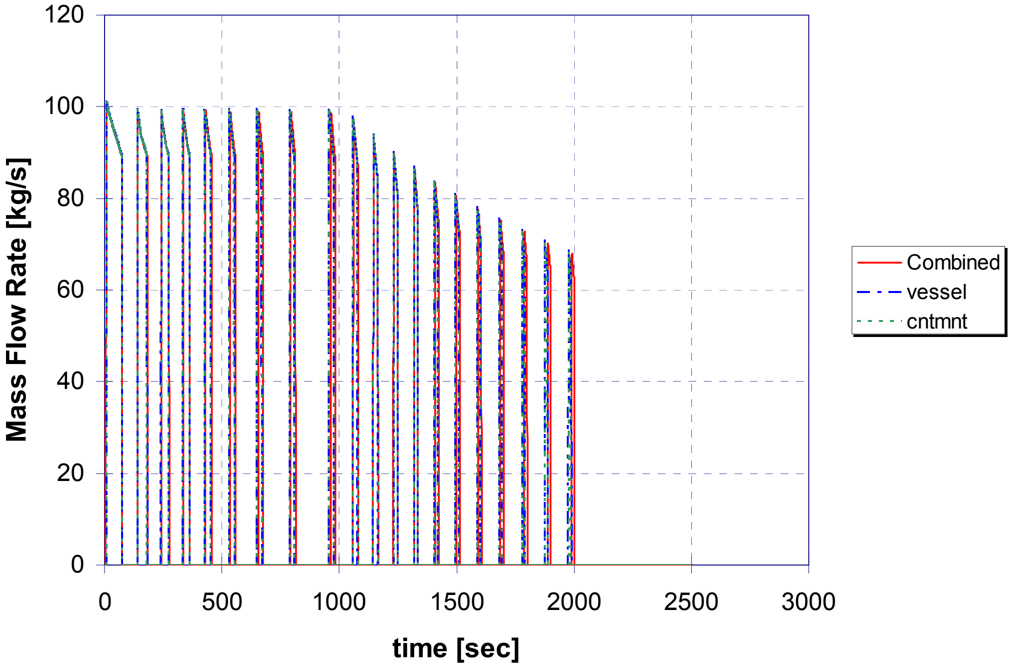
During the setup process for the coupled MELCOR run, the executive directs "vessel" to send the nominal pressure, CVH-P.150, the atmosphere temperature, CVH-TVAP.150, the partial pressure of water vapor, CVH-PPART.150, and the mole fractions of H_2 and N_2, CVH-X.6.150 and CVH-X.6.150, in CV200 to "vessel. These values are received as the values of PVM-MSG control functions 80001, through 80005, respectively. They are used to define the time-dependent properties of the duplicated CV150 in "cntmnt."

3.3 Comparison of Results for Split and Integral Calculations

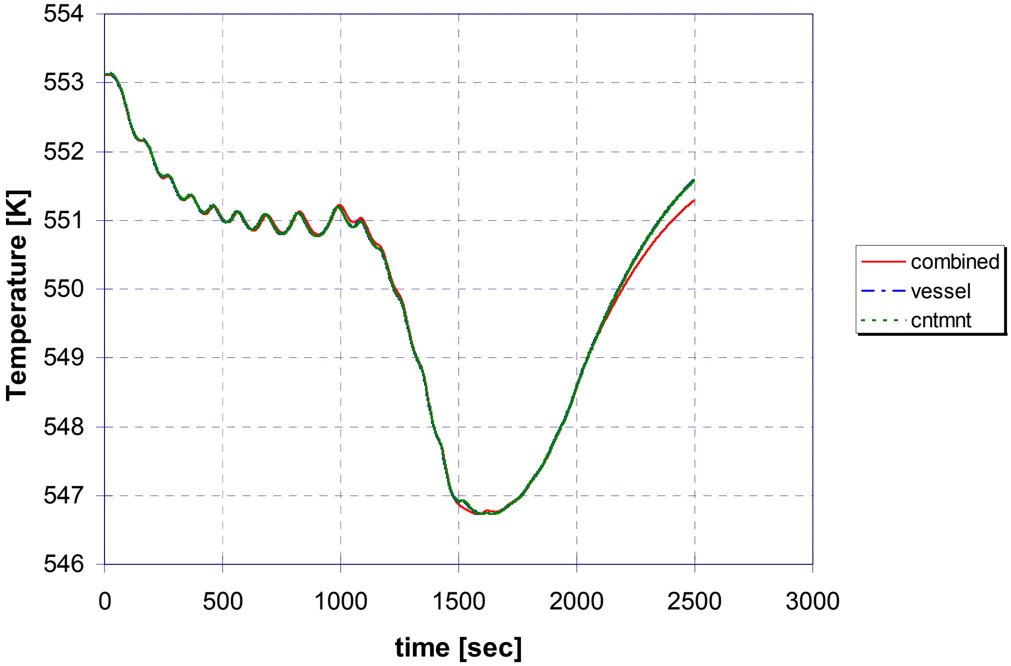
The following plots present some results for the split calculation, compared to those for the original integral (combined) one. The variables compared are the total mass flow through the SRV, the temperature of the outer surface of the vessel, and the pressure in the wetwell.

The first calculation employed a maximum time step of 1.0 seconds, with data exchanged every 2.0 seconds. This is only marginally adequate for resolution of SRV flows, because (after the first few activations) the valve is open for only about 10 seconds at a time. Despite this, agreement is quite good for the mass flow and vessel temperature; the most noticeable discrepancy is a slight shift in timing. Differences in the wetwell pressure are more pronounced; the essential difference is that the combined calculation predicted three hydrogen deflagrations in that volume, but the split calculation predicted only two.

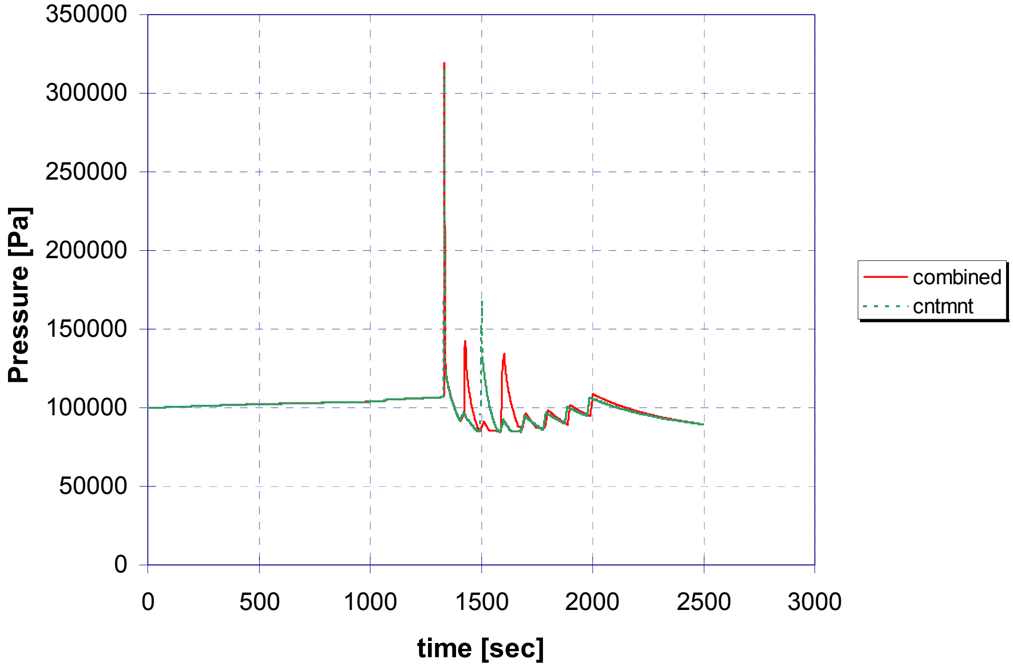
When the calculation was rerun with a reduced maximum time step and data exchanged every 0.1 seconds, the agreement between the split calculation and the combined was markedly better. All comparisons were improved, but the change was particularly apparent in the prediction of hydrogen deflagrations and their effect on wetwell pressure, where the results became almost identical.



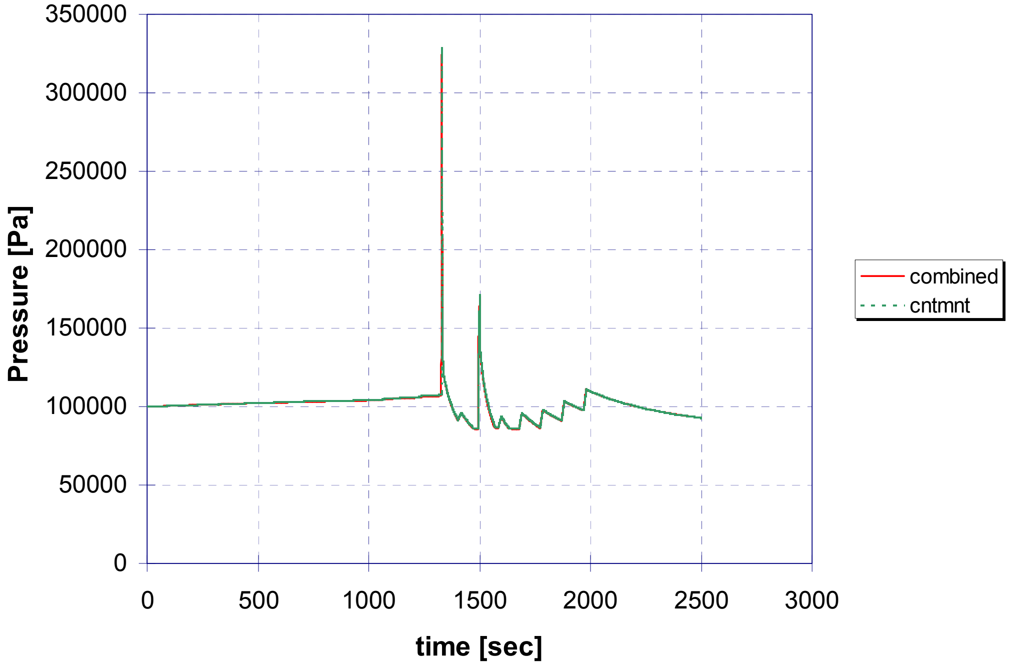
SRV Mass Flow, Data Exchange every 2.0 seconds



Vessel Outer Wall Temperature, Data Exchange every 2.0 seconds



Wetwell Pressure, Data Exchange every 2.0 seconds



Wetwell Pressure, Data Exchange every 0.1 seconds

4. DIFFERING EQUATIONS OF STATE

In coupling two codes that use different equations of state (EOS) for a given material, either the temperature or the enthalpy can be preserved, but not both. (The only exception is the trivial case where the *only* difference is the reference point.) If the specific heats underlying (or derivable from) the two versions of the EOS are very different, there is no acceptable resolution. If differences are “small”, as we believe they are between MELCOR and RELAP5, there seems little basis for arguing that preserving temperature or enthalpy is “better,” and the choice can be based on other considerations.

4.1 Preserving Temperatures

Preservation of temperature is easy, unambiguous, and requires no information about the actual differences in the EOS. This approach was used for the MELCOR-to-MELCOR example in this paper, where SRV flows were transmitted by defining a donor volume and flow path in the “cntmnt” task, with properties “slaved” to values calculated in “vessel.” The properties reproduced were pressure, temperature, composition, and flow velocity.

4.2 Preserving Enthalpy

If one is willing to stipulate that matching reference point enthalpies is adequate, there is no problem in transmitting mass flows and specific enthalpies. No specific modifications to the coding of any coupled task are necessary, since the control function capability in MELCOR will handle the necessary offsets for data transmitted in either direction.

Sources and boundary conditions in MELCOR are imposed using values defined by control functions, and MELCOR receives data from a sibling process “X” (e.g. RELAP5) as the value of a control function. However, these need not be the *same* control functions.

Suppose that the mass and enthalpy flows are received from “X” as $CF_{M,X}$ and $CF_{H,X}$, and that the specific enthalpy at some reference point “0” is $h_{0,X}$. If the corresponding specific enthalpy in MELCOR is $h_{0,MELCOR}$, all that is necessary is to define one additional control function, as part of MELGEN input,

$$CF_{H,MELCOR} = CF_{H,X} + \dot{M}_X \left(h_{0,MELCOR} - h_{0,X} \right) ,$$

and use it in MELCOR (rather than $\dot{H}_X$) as the enthalpy associated with the mass flow $\dot{M}_X$.

Variables are sent to other tasks as the values of control function arguments. The mass and enthalpy flows in a MELCOR flow path are available as control function arguments.

Because the value of any control function is itself available as a control function argument, an intermediate control function,

$$CF_{H,X} = \dot{H}_X + \dot{M}_{MELCOR}\left(h_{0,X} - h_{0,MELCOR}\right) ,$$

can be constructed and sent to “X” (rather than $\dot{H}_{MELCOR}$) for use as the enthalpy associated with the mass flow $\dot{M}_{MELCOR}$.

5. SUMMARY AND CONCLUSIONS

MELCOR has been modified to run coupled calculations under an executive program, with the PVM protocol used for information exchange, and the code has been successfully tested. This paper describes a calculation in which two MELCOR tasks are coupled. A companion paper describes one in which a MELCOR task is coupled to a RELAP5-3D$^{©}$ task. These and other tests show that the code performs as intended, and that (with sufficiently frequent exchange of data) a calculation split into multiple tasks can give excellent agreement with a combined calculation.

The modified version of MELCOR can be built with or without linking to PVM libraries. If the libraries are linked, MELCOR can be run under the executive or in standalone mode. We expect this capability to be included with the next release of MELCOR.

References

1. K. K. Murata, et al., “Code Manual for CONTAIN 2.0: A Computer Code for Nuclear Reactor Containment Analysis,” NUREG/CR-6533, SAND97-1735, Sandia National Laboratories, Albuquerque, New Mexico, 1997. Early applications used CONTAIN 1.1, NUREG/CR-5026, SAND87-2309, 1990.

2. K. E. Carlson, *et al.*, “RELAP5/MOD3 Code Manual,” NUREG/CR-5535, EGG-2596, EG&G Idaho, Inc., 1990.

3. “RELAP5-3D Code Manuals,” Idaho National Engineering and Environment Laboratory, INEEL-EXT-98-00834, Revision 1b, 1999.

4. R. O. Gauntt *et al.*, “MELCOR Computer Code Manuals,” NUREG/CR-6119, Vols. 1 and 2, Rev. 2, SAND2000-2417/1, Sandia National Laboratories, Albuquerque, New Mexico, December 2000.

5. W. L. Weaver, E. T. Tomlinson, and D. L. Aumiller, “An Executive Program for use with RELAP5-3D©,” B-T-3394, 2001 RELAP5 Users Seminar, Sun Valley, Idaho, September 5-8 2001. This paper contains additional references.

6. Al Geist *et al.*, “PVM 3 User’s Guide and Reference Manual,” ORNL/TM-12187, Oak Ridge National Laboratory, Oak Ridge, Tennessee, September 1994.

7. “Fifth MELCOR Users’ Workshop,” MELCOR project Staff, Bethesda, Maryland, May 10-15, 2001. Available at http://melcor.sandia.gov.