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

A Mass and Energy Conserving Form of Explicit Coupling For Use With RELAP5-3D©

W. L. Weaver
Idaho National Engineering and Environmental Laboratory
P.O. Box 1625
2525 North Fremont Ave.
Idaho Falls, ID 83415-3880

Abstract

A mass and energy conserving form of explicit coupling has been developed for use with the RELAP5-3D© computer program. The original implementation of explicit coupling did not conserve mass or energy although the mass and energy conservation errors could by mimimized by the use of short explicit coupling intervals at the expense of increased execution time for the coupled computation. The newly developed form of explicit coupling conserves mass and energy in that the mass and energy that leaves the computational domain of one code through the coupling junctions between the computational domains of the two coupled codes enters the computational domain of the other code. This was never true for the original implementation of explicit coupling. This new explicit coupling algorithm allows for the use of larger coupling intervals that reduce the communication overhead of a coupled computation.

1.0 Background

Several previous papers (Martin, 1995; Aumiller, 2001; Weaver, 2002) have described the methodology by which the RELAP5-3D© computer program (RELAP5-3D, 2002) may be coupled to another computer code either explicitly or semi-implicitly. The first two papers (Martin, 1995 ; Aumiller, 2001) describe how RELAP5-3D© was coupled explicitly to both another instance of RELAP5-3D© and to another thermal-hydraulic analysis code. The paper by Weaver (2002) describes the methodology by which RELAP5-3D© can be coupled to another thermal-hydraulic code using a semi-implicit coupling methodology. The coupling between RELAP5-3D© and other codes was accomplished using the PVM (Parallel Virtual Machine) message passing software developed at Oak Ridge National Laboratory (Geist, 1993). In the previously developed explicit coupling algorithm, the PVM executive program (Weaver, 2001) directs the coupled codes to advance in parallel using fixed boundary conditions for each coupling interval. If both codes use pressure boundary conditions, the flow rates of mass and energy in the

coupling junctions computed by one code will, in general, not be the same as the flow rates of mass and energy computed by the other code. This is because both codes solve the momentum equations at the coupling locations and there is no guarantee that the two codes solve the momentum equations in the same way. Even if one code uses pressure boundary conditions to solve the momentum equations at the coupling location and the other code uses the phase velocities in the coupling junction computed by first code as its boundary conditions, mass and energy will not be conserved. This is because the mass and energy flow rates are instantaneous values, not average values over the coupling interval, and because the two codes are one coupling interval out of phase.

Mass and energy can be conserved in explicit coupling if both codes use the same mass and energy flow rates in the coupling junctions to advance their solutions. Both codes could solve for these mass and energy flow rates, but the codes would need to be modified to solve the momentum equations in the coupling junctions in the same way. An easier way to ensure that both codes use the same mass and energy flow rates in the coupling junctions is for one code to compute the flow rates in the coupling junctions and to use these instantaneous values to compute the average values over the coupling interval. Then the other code uses these average flow rates of mass and energy in the coupling junctions to advance its solution. This means that the second code must wait for the first code to compute the average flow rates of mass and energy before advancing its solution. This guarantees that the total amount of mass and energy that leaves the computational domain of the first code enters the computational domain of the second code. This also changes the explicit coupling algorithm from *parallel* advancement of the two coupled codes to *sequential* advancement of the two coupled codes. Because the codes are advanced one after the other, the mass and energy conserving form of explicit coupling will be called sequential explicit coupling to distinguish it from the previously implemented explicit coupling that will be designated as parallel explicit coupling.

2.0 Sequence of Events in a Sequentially Coupled Computation

A coupled computation can be divided into two phases, that are the input and initialization phase of the computation and the transient simulation phase of the coupled computation. A coupled computation is initiated and controlled by the PVM executive program (Weaver, 2001).

2.1 Input and Initialization Phase

The PVM executive program is executed by the user in a manner appropriate for the users operating system specifying the input file and the output file for the executive program as command line parameters (default input and output files are also defined). The executive program reads the first section of its input file, constructs a PVM hostfile, and starts the PVM daemon process on the several computational nodes in the virtual machine. Then the executive program spawns the several coupled processes on the one or more computational nodes. The coupled processes that are spawned read their respective input files, process the data contained in their input files, and then listen to receive messages from the executive process. After the executive process has spawned all of the coupled processes, it sends messages to each of the spawned

processes containing the data specifications for messages to send to and receive from the other coupled processes. Each spawned process proceeds with its own input and initialization after the coupling data specifications have been received from the executive process. The executive process listens to receive a message from each process describing its initialization status and its run status. Each coupled code sends its initialization status to the executive program at the end of its initialization process. This initialization status may be zero (initialization successful) or one (errors during input and initialization). They also send the executive program their run status, where zero denotes no transient to be executed because of input or initialization errors or because this run was for input checking only, or one, ready for transient simulation. The coupled computation is terminated if any of the coupled processes return an initialization error or return a zero run status. The executive program determines the global initialization and run status and broadcasts this status to all of the coupled processes.

2.2 Transient Computation Phase

Assuming that the initialization was successful for all of the coupled processes and that the run status indicated that all coupled processes are ready to perform a transient simulation, the executive program initiates the transient phase of the coupled computation. After the exchange of control information, the executive directs the coupled codes to advance through the first explicit coupling interval. At the beginning of the first advancement of an explicit coupling interval, one code (designated the ‘follower’ code) sends coupling volume conditions to the other code (designated the ‘leader’ code) and then waits to receive the average flow rates and the average properties in the coupling junctions. The leader code receives the coupling volume conditions from the follower code, proceeds to advance in time computing the conditions in the coupling junctions, holding the coupling volume conditions constant in time. The leader code computes the average flow rate of mass and energy passing through the coupling junctions and the average fluid properties in the coupling junctions during the coupling interval. At the end of the coupling interval, these average flow rates and average properties are transmitted to the follower code. The leader code then waits for messages from the executive program for instructions as to how to proceed (end transient or proceed to a new coupling interval). The follower code, having received the average flow rates and the average fluid properties in the coupling junctions, can now advance through the coupling interval. Once it has reached the end of the coupling interval, it communicates with the executive program as to how to proceed. At this point both codes have advanced to the end of the first coupling interval and are waiting to hear from the executive program as to how to proceed. The executive program will then direct the coupled codes to proceed through the coupling intervals until the transient has reached the end time, whereupon it terminates the coupled computation and shuts down the virtual machine. One point to emphasize is that the codes proceed asynchronously using different time step sizes to advance themselves through the coupling intervals. Synchronous coupling could be used but the coupling interval would be restricted to the global time step size thereby increasing the communication overhead for the computation.

3.0 Testing of Sequential Explicit Coupling

The sequential explicit coupling algorithm was tested using a sequential version of the explicitly coupled Edwards pipe problem (Aumiller, 2001) where RELAP5-3D was used to simulate both computational domains. Figure 1 shows nodalization diagrams for this test case along with the data flow. Three versions of this problem were executed. The first test case is the Edwards pipe problem as a single system. The second and third test cases computed the solution as a coupled problem, the second test case using parallel explicit coupling and the third test case using sequential explicit coupling. Control systems were added to the input decks for these test case to compute and store the initial mass in the pipe components and to compute and store the integral of the mass flow rate in the output junction for the pipe components. The accuracy of the simulations was assessed by comparing the actual initial mass in the system to the inferred initial mass in the system. The inferred initial mass in the system is the sum of the current mass in the system and the integral of the mass flow rate at the output junction for the system. The actual initial mass in the system for the two coupled test cases is the sum of the actual initial masses in the two coupled pipe components. The inferred initial system mass for the two coupled test cases is the sum of the current masses in the two pipe components plus the integral of the mass flow rate in the output junction of the downstream pipe component. The integral of the mass flow rate in the coupling junction in the coupled simulations is not used because this junction is an internal junction.

Figure 2 shows the results of the three simulations. Four curves are shown in Figure 2. The curve with no symbols is the actual initial mass in the system. The curve with the diamond symbol is the inferred initial mass from the computation as a single system. This curve shows that there is a small mass increase in the system over the duration of the simulation (the simulation was for 500 time steps using a coupling time interval of 0.0001 s). The curve with the circle symbols is the inferred initial mass in the system computed using parallel explicit coupling. The mass error for this simulation is larger than the error in the computation as a single system. This larger mass error for parallel explicit coupling is an expected result. The last curve with the square symbols is the inferred initial system mass computed using sequential explicit coupling. The mass error is much smaller than the mass error for parallel explicit coupling and is also an expected result. Finally, the mass error for sequential explicit coupling is slightly smaller that the mass error for the computation as a single system. It is impossible to determine a priori whether the mass error in the sequential explicit coupling should be larger or smaller that the mass error in the computation as a single system. One would expect that the solution as a single system should be more accurate (i.e., less error) than the computation as an explicitly coupled system. However, the differences between the two solution algorithms have compensating errors which makes it difficult to determine whether the mass error will increase or decrease. In the computation as a single system, the upstream pressure used in the computation of the velocities in the junction in the middle of the system (i.e., the junction that is the coupling junction in the coupled simulation) is a new time value, whereas it is an old time value in the coupled simulation. The use of the new time value should make the solution more accurate. However, the enthalpy flow rate in the coupling junction in the single system is non-conservative where the enthalpy flow rate out of the upstream volume uses the pressure in the upstream volume and the enthalpy flow rate into the downstream volume uses the pressure in the downstream volume. This means that the enthalpy flow rate out of the upstream volume is not equal to the enthalpy flow rate into the downstream volume. This feature

of the computation as a single system should make its solution less accurate. In the sequential explicit solution, these two 'errors' are reversed. The upstream pressure used to compute the phasic velocities in the coupling junction is an old time value, which should make the solution less accurate. However, the enthalpy flux through the coupling junction is in conservative form, which should decrease the error in the coupled solution. Thus the error in the two solutions is the result of compensating errors and the magnitude of the error in each of the solutions is the difference between the magnitudes of the two effects thus making it difficult (or impossible) to determine which solution should be more accurate. The fact remains that the error in sequential explicit coupling is far less than the error in parallel explicit coupling.

A second set of test cases were executed to investigate the effect of the size of the coupling interval on the accuracy of the explicit coupling algorithms, both parallel and sequential. The first set of three test cases used a coupling interval of 0.0001s. This value is the same as the maximum time step size specified for the two coupled codes. The second set of test cases used a coupling interval of 0.001s, a value 10 times larger than that used in the first test case, while the maximum time step was held constant at 0.0001s. This combination of maximum time step size for the coupled processes and the longer coupling interval guaranteed that there would be at least 10 time step advancements per coupling interval. The sequential explicitly coupled test case executed to completion whereas the parallel explicitly coupled test case failed almost immediately. The coupling interval for the parallel test case was reduced to factors of 2 and 5 times the maximum time step size, but these test cases also failed immediately. Figure 3 shows the estimated initial system mass for the two sequential explicitly coupled runs. The mass error in the two cases is virtually identical even though the coupling intervals vary by a factor of 10. Figure 4 shows an expanded view of the mass flow rate in the coupling junction for the leader and follower processes for the test case using the longer coupling interval. Figure 4 shows that the mass flow rate in the follower process is an averaged value and that the average tracks the instantaneous values in the leader process.

4.0 Summary

A mass and energy conserving form of explicit coupling has been developed, implemented in RELAP5-3D, and tested. The new sequestial coupling algorithm has been tested to determine if it performs as expected. The results of the tests show that the model is performing as expected.

Acknowledgement

Work supported by the U. S. Department of Energy, under DOE Idaho Field Office Contract No. DE-AC07-99ID13727.

References

1 Aumiller, D. L., Tomlinson, E. T., Bauer, R. C., 2001. "A Coupled RELAP5-3D/CFD Methodology with Proof-of-Principle Calculation," Nuclear Engineering and Design, Vol.

205, pp. 83-90.

2 Geist, A. et. al., 1993. "PVM (Parallel Virtual Machine) User's Guide and Reference Manual," Oak Ridge National Laboratory, ORNL/TM-12187.

3 Martin, R. P., 1995. "RELAP5/MOD3 Code Coupling Model," Nuclear Safety, Vol. 36, No. 2, pp. 290-299.

4 RELAP5-3D, 2002. "RELAP5-3D Code Manuals, Volumes I, II, IV, and V," Idaho National Engineering and Environmental Laboratory, INEEL-EXT-98-00834, Revision 2.0. (See also the RELAP5-3D home page at http://www.inel.gov/relap5)

5 Weaver, W. L., Tomlinson, E. T., Aumiller, D. L, 2002, "A Generic Semi-Implicit Coupling Methodology for Use in RELAP5-3D", Nuclear Engineering and Design, 211, pp. 13-26.

6 Weaver, W. L., Tomlinson, E. T., Aumiller, D. L., 2001, "An Executive Program for Use With RELAP5-3D", Proceedings of the RELAP5-3D International Users Seminar, Sun Valley, ID, Sept. 5-7, 2001.

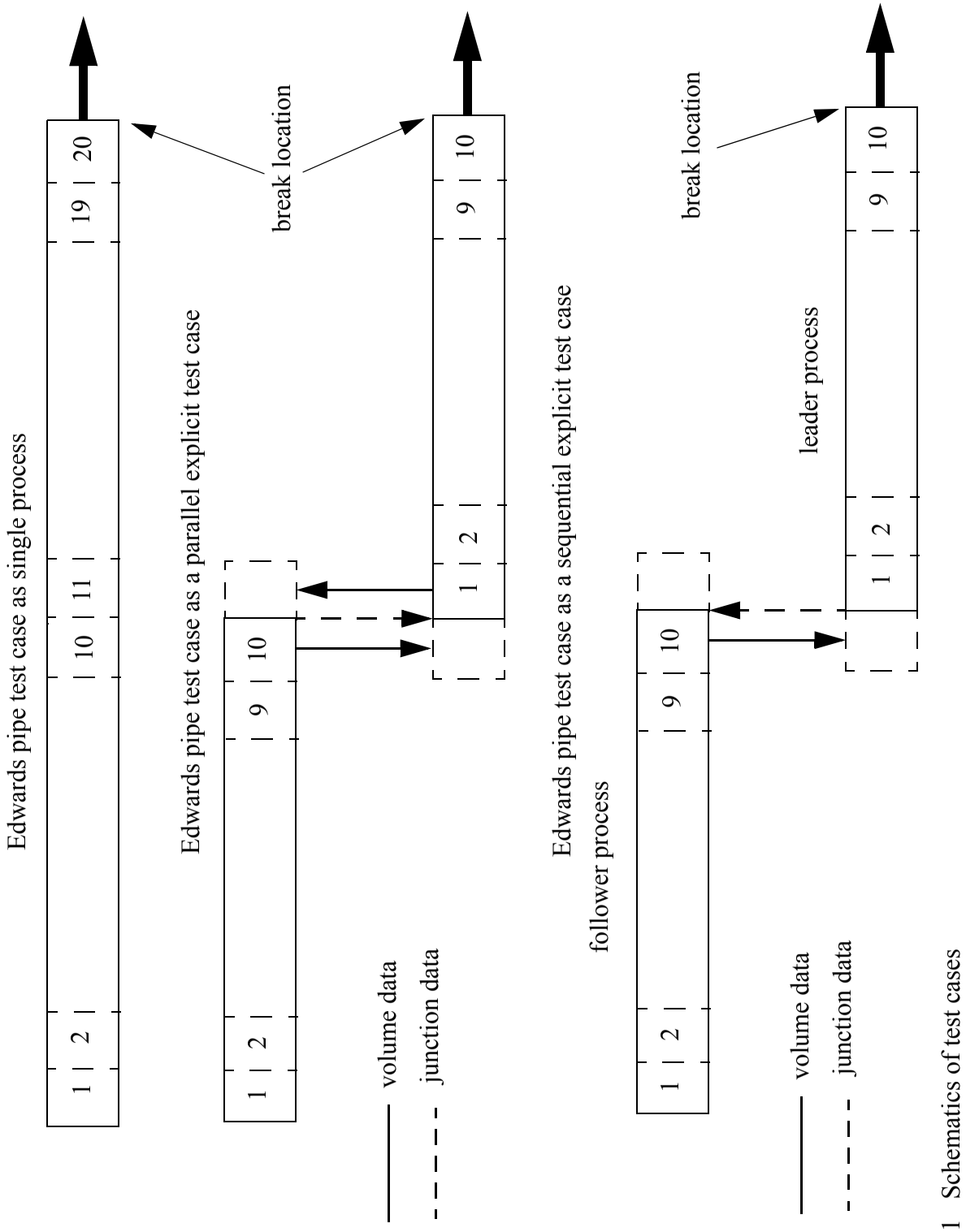

Figure 1 Schematics of test cases

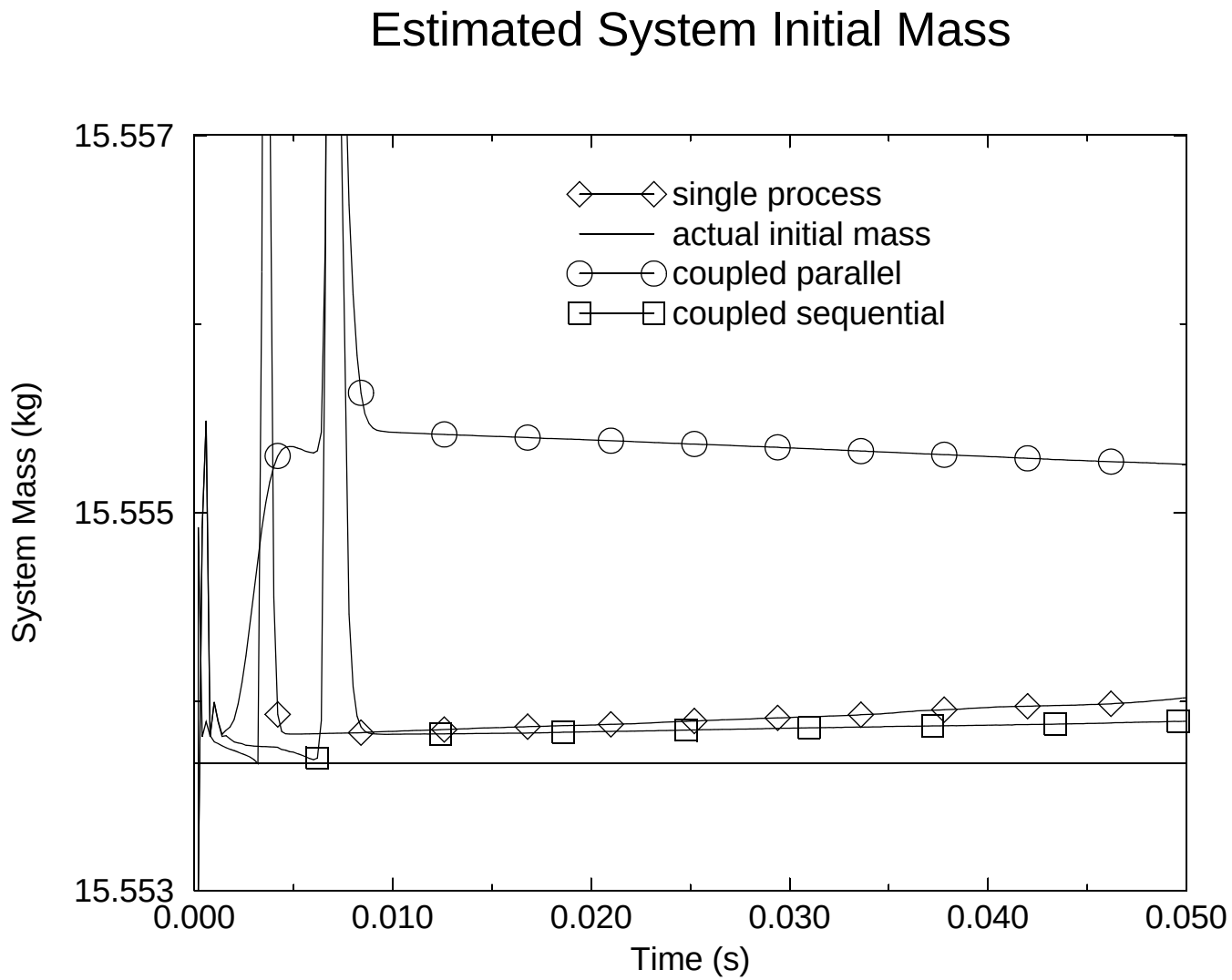

Figure 2 Estimated initial system mass for different test cases

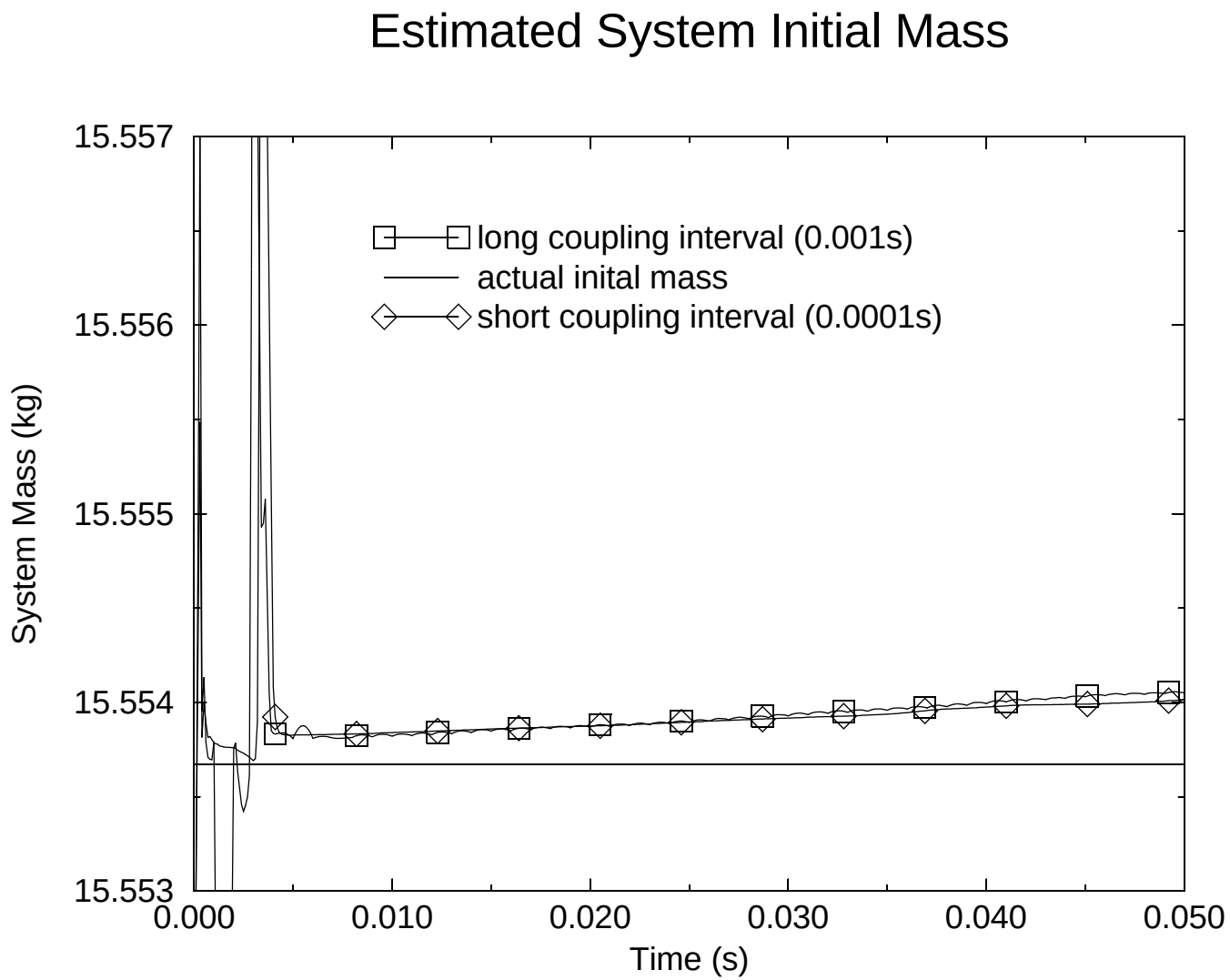

Figure 3 Estimated system inital masses at different explicit coupling interval sizes

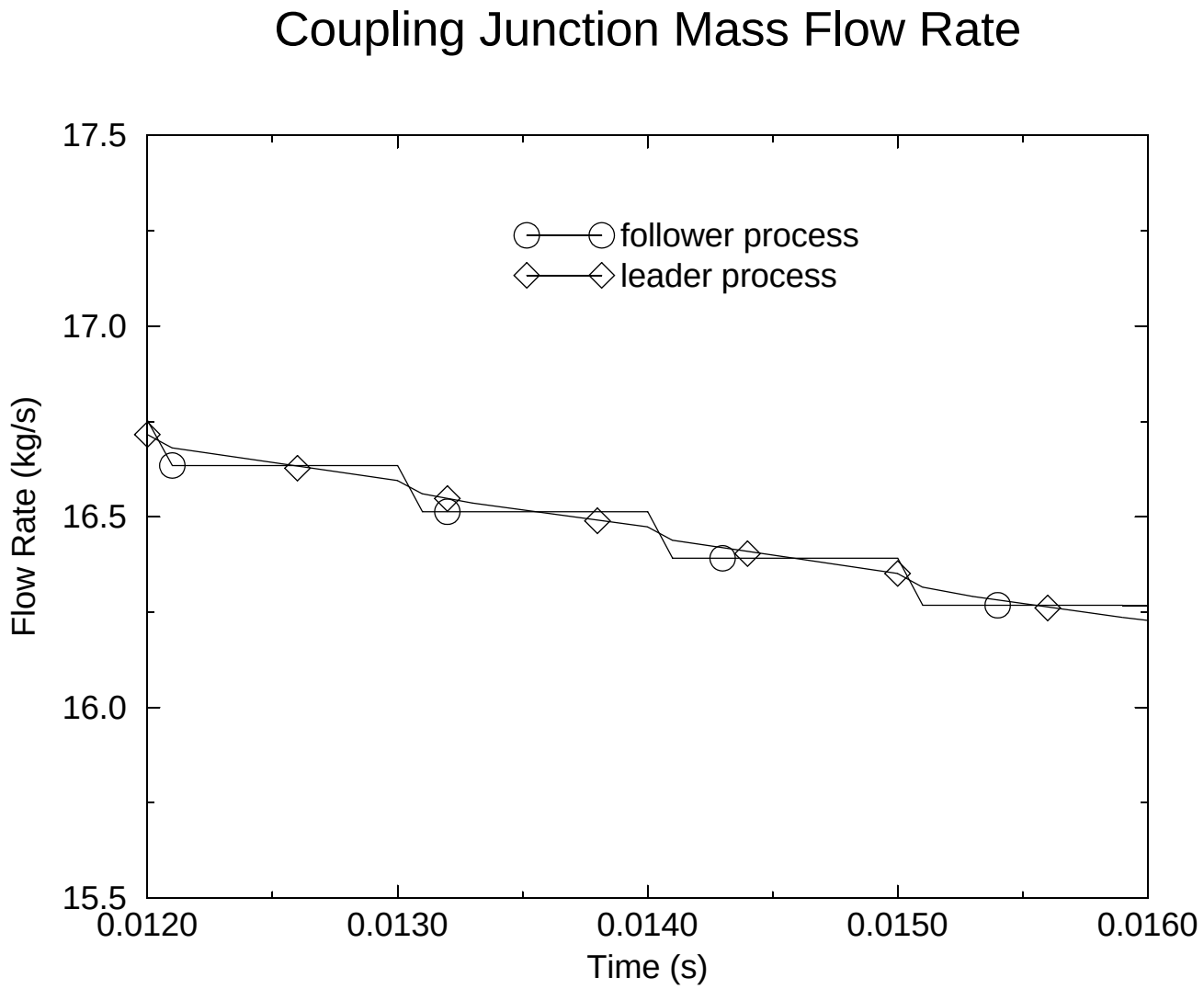

Figure 4 Coupling junction mass flow rate in different processes