Solving Implicit Multi-Mesh Flow and Conjugate Heat Transfer Problems with RELAP-7

M&C 2013

Ling Zou
John Peterson
Haihua Zhao
Hongbin Zhang
David Andrs
Richard Martineau

May 2013

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint should not be cited or reproduced without permission of the author. This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party’s use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. The views expressed in this paper are not necessarily those of the United States Government or the sponsoring agency.
**SOLVING IMPLICIT MULTI-MESH FLOW AND CONJUGATE HEAT TRANSFER PROBLEMS WITH RELAP-7**

Ling Zou, John Peterson, Haihua Zhao,
Hongbin Zhang, David Andrs, and Richard Martineau  
Idaho National Laboratory  
2525 Freemont Ave., Idaho Falls, ID 83415  
Ling.Zou@inl.gov; JW.Peterson@inl.gov; Haihua.Zhao@inl.gov;  
Hongbin.Zhang@inl.gov; david.andrs@inl.gov; richard.martineau@inl.gov

**ABSTRACT**

The fully implicit simulation capability of RELAP-7 to solve multi-mesh flow and conjugate heat transfer problems for reactor system safety analysis is presented. Compared to general single-mesh simulations, the reactor system safety analysis-type of code has unique challenges due to its highly simplified, interconnected, one-dimensional, and zero-dimensional flow network describing multiple physics with significantly different time and length scales. To use the Jacobian-free Newton Krylov-type of solver, preconditioning is generally required for the Krylov method. The uniqueness of the reactor safety analysis-type of code in treating the interconnected flow network and conjugate heat transfer also introduces challenges in providing preconditioning matrix. Typical flow and conjugate heat transfer problems involved in reactor safety analysis using RELAP-7, as well as the special treatment on the preconditioning matrix are presented in detail.

*Key Words: JFNK, MOOSE, reactor safety analysis code*

1. **INTRODUCTION**

In system analysis codes, thermal hydraulics is one of the most important physics in determining the system response. Fluids flow and conjugate heat transfer are of great interest in reactor safety analysis because they are essential to predict heat transfer between the fuel and coolant in the reactor core and structure materials and fluids in steam generators. In existing reactor safety analysis codes (such as RELAP5 [1]), fluids flow and conjugate heat transfer were achieved by using an operator-splitting type of loose coupling. Unfortunately, this type of loose coupling is a less desirable method, because it generally is less accurate, especially when different physics are strongly coupled and a fast transient is present. Therefore, for future system analysis codes such as RELAP-7 [2], using fully coupled method to solve the fluids flow and conjugate heat transfer problem is a desired approach. RELAP-7, a MOOSE-based [3] application, is the next generation system safety analysis code being developed at Idaho National Laboratory.

By using the MOOSE framework, the full coupling between different physics in reactor safety analysis has been achieved through the Jacobian-free Newton Krylov (JFNK) method. The JFNK method is a multi-level approach, because it includes the outer Newton’s iterations and inner Krylov subspace methods in solving large linear systems. In solving large-scale problems, the concept of ‘Jacobian-free’ is proposed, because deriving and assembling such large Jacobian matrices could be difficult and expensive. However, in most applications, the Krylov subspace methods require preconditioning to be efficient. There are different ways in providing the preconditioning matrix (for example, approximated inexact Jacobian, mutli-grid approaches and
physics-based preconditioning). An extensive review of JFNK methods and its applications is available in reference [4].

Although the JFNK method has been widely used in different applications, there are only a few such applications in solving reactor safety analysis problems [5, 6, 7, 8]. It should be noted that, in general, a system analysis code has unique challenges in terms of spatial discretization of the system (simulation domain) when compared to general single-mesh type of multi-physics simulation. System analysis-type of codes tends to deal with smaller problems (in terms of the number of degrees of freedom) and the problems tend to have sparse matrices with more scattered entries due to heterogeneous meshes. Additionally, the different physics involved in reactor safety analysis are typically more complicated, with wider range of time and spatial scales.

In reactor safety analysis codes, the complicated physical components are highly simplified as interconnected, one-dimensional, zero-dimensional, and very few two and three-dimensional components. This kind of approach has been proved to be reasonable and effective, because reactor safety analysis codes generally require fast running speed, and the results focused more on the system response rather than local details. One-dimensional components generally are responsible for capturing local changes of physics. For example, core channels are designed to simulate the conjugate heat transfer between fuel rods and their surrounding coolant. Zero-dimensional components are designed to simulate components with very complicated geometries and physics (such as reactor core upper/lower plenums or steam separator and dryer in boiling water reactors) or to simulate connection components (such as pumps and elbows). The later connection capability of zero-dimensional components serves a critical role in system analysis codes (i.e., connecting all components to form a network to simulate the reactor system). This major difference, in contrast to single-mesh simulations, introduces unique challenges in the treatment of coupling between different meshes and physics. In the case of RELAP-7 implementation, the couplings between different meshes are realized in the form of node-to-node or cell-to-cell information transfer. In addition, the problem tends to be relatively small-scaled when compared to general multi-physics coupling simulations (e.g., computational fluid dynamics simulations), which could affect selection of the solution algorithms.

In this paper, the multi-mesh and multi-physics approach implemented in RELAP-7 in solving reactor safety thermal-hydraulics problems, such as flow and conjugate heat transfer, is introduced. Basic physics equations, several one-dimensional and zero-dimensional components for general reactor safety analysis, special treatment of Jacobian terms, preconditioning, comparison between simulation results and analytical solutions, and study on numerical convergence are presented.

2. MODEL DESCRIPTIONS

2.1. Flow Equation System

Initial efforts have been spent to simulate single-phase fluid dynamics in order to demonstrate the capability of RELAP-7 [2]. One-dimensional pipes are basic components for predicting the fluid flow and wall heat transfer involved in reactor safety analysis. The system of equations of
the one-dimensional, compressible, single-phase fluid flow with wall friction and wall heat transfer in the conservative form is given by Equation (1):

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + S = 0$$  \hspace{1cm} (1)

in which, $U$ is the conservative variables vector; and $F(U)$ and $S$ are advection flux and source terms, respectively. For single-phase flow, the conservative variables include density ($\rho$), momentum ($\rho u$), and total energy ($\rho E$), which correspond to continuity, momentum, and energy equations, respectively. Those conservative variables and corresponding advection flux terms are both shown in Equation (2):

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix}; \quad F(U) = \begin{bmatrix} \rho \\ \rho u^2 + p \\ \rho u H \end{bmatrix}; \quad S = \begin{bmatrix} 0 \\ f \frac{\rho}{2D_h} |u|u - \rho g x \\ H_w a_w (T_f - T_w) + u \left( f \frac{\rho}{2D_h} |u|u - \rho g x \right) \end{bmatrix}$$  \hspace{1cm} (2)

in which $p$ is pressure; $H$ is the total enthalpy, and $H = E + p/\rho$. The source term in the momentum equation consists of two contributions: wall friction and gravity terms, as shown in Equation (2), in which $u$ is velocity; $D_h$ is the hydraulic diameter of a pipe; $f$ is the non-dimensional wall friction factor; and $g_x$ is the gravity component in the pipe direction. The source term in the energy equation also consists of two contributions: the wall heat transfer part and the external force power part, as shown in Equation (2), in which $H_w$ is the wall heat transfer coefficient; $a_w$ is the heat transfer wall area density per unit fluid volume; and $T_w$ is the wall temperature.

Spatial discretization is implemented by using a Galerkin type of finite element method (i.e., the partial differential equations are multiplied by test functions and then integrated over the whole domain):

$$\int PDE \cdot \phi d\Omega = 0$$  \hspace{1cm} (3)

in which, $\phi$ is called the test function and Equation (3) is called the weak form. The solution of a hyperbolic equations system (such as Equation (1)) generally has spurious oscillations for the Galerkin type of finite element method and requires stabilizers to avoid these oscillations in the solution. The normally used stabilizer schemes include artificial viscosities and the streamline upwind Petrov Galerkin (SUPG) method [9]. In this study, the SUPG method was successfully used as the stabilizer for the fluid equations.

2.2. Boundary Coupling between Two-Pipe Simulation

As explained in the previous sections, reactor system analysis codes are simplified as interconnected, one-dimensional and zero-dimensional components. Pipes and pumps are good examples of these one-dimensional and zero-dimensional components. The simulation of a single pipe flow could be simplified to solve the one-dimensional fluid equation system inside a pipe with proper boundary conditions. However, the simulation of pipe flow in two pipes driven by a pump requires an additional zero-dimensional component (such as an ideal pump) to deal with the special treatment on pipe boundaries. Figure 1 shows a schematic drawing of flow in two pipes driven by an ideal pump. For simplicity, an ideal pump is designed to maintain a given
amount of mass flow rate. Also, because the pumping process and the fluid are assumed to be ideal, there is no internal energy addition or loss to the fluid. The information exchange, or coupling, between these two pipes takes place across the ideal pump in the form of coupled boundary conditions. For the continuity equation, the same mass flow rate boundary condition is applied on both the outlet of Pipe (1) and the inlet of Pipe (2). For the momentum equation, similar to the continuity equation, a local momentum flux in the form of \((p + \rho u^2)\hat{n}\) is applied on both boundaries, in which \(\hat{n}\) is the local outward normal direction of the pipe end. However, the energy equation needs special treatment on both boundaries. For this specific case, assuming the flow is from Pipe (1) to Pipe (2), the inlet fluid internal energy of Pipe (2) is set to be equal to the outlet fluid internal energy of Pipe (1).

![Figure 1. Schematic drawing of pipe flow driven by an ideal pump.](image)

### 2.3. Conjugate Heat Transfer

Conjugate heat transfer in RELAP-7 is implemented by fully coupling the fluid equations, which only exist in fluid part, and heat conduction equations, which only exist in solid structures such as fuel rods. Figure 2 shows a schematic drawing of a typical conjugate heat transfer scenario in a reactor safety simulation (i.e., the heat exchange between a fuel rod and its surrounding coolant). As shown in Figure 2, the surrounding coolant is simulated as one-dimensional coolant channel. For simplicity, the fuel rod is divided into the same number of segments as that of the fluid channel. Each fuel rod segment is then simulated by a one-dimensional heat conduction problem with volumetric heat generation and convective heat transfer boundaries. The heat conduction equation is given as:

\[
\rho C_p \frac{\partial T_s}{\partial t} - \frac{\partial}{\partial x} \left( k \frac{\partial T_s}{\partial x} \right) - q''' = 0
\]  

(4)

in which, \(\rho, C_p,\) and \(k\) are density, specific heat capacity, and thermal conductivity of solid material, respectively. \(q'''\) is the volumetric energy density.

As shown in Equation (2), the energy exchange between the fuel and the coolant contribution is calculated as:

\[
H_w a_w (T_f - T_s)
\]  

(5)

in which, \(T_f\) is the nodal fluid temperature and \(T_s\) is the solid surface temperature. On the contrary, coupling between fluid and solid heat transfer poses a coupled boundary on the simulation of fuel heat conduction problem:

\[-k \frac{\partial T_s}{\partial x} = H_w (T_s - T_f)\]  

(6)

Similar to the fluid part, the same finite element method was used for the discretization of heat conduction equation in the solid part.
2.4. Solution Scheme

By using the JFNK method, in RELAP-7, different physics are fully coupled and the single unknown vector is solved iteratively. As discussed previously, a Krylov-type of method generally requires preconditioning to be efficient and effective. In this work, an approximation to the exact Jacobian is provided to the solver as a preconditioning matrix. However, the level of approximation is difficult to define and, in a lot of cases, it could be problem dependent. It generally is encouraged to include, at least, diagonal terms in the preconditioning matrix. For example, the transient term, which sits exactly on the matrix diagonals, is required. For one-dimensional problems, a tridiagonal Jacobian matrix is, in general, formed during the spatial discretization process and is encouraged to be included in the preconditioning matrix. Other than those diagonal and tridiagonal terms, the approximation level of preconditioning matrix to the exact Jacobian could be problem dependent, which, in turn, could depend on the nonlinearity of the problem (i.e., how strong the coupling is between different physics or variables). Sometimes, experience plays an important role in determining the preconditioning matrix. The following subsections discuss the detailed implementation of approximated Jacobians for different cases.

2.4.1. Single Pipe Fluid Flow Case

As discussed in the previous paragraph, diagonal and tridiagonal terms for a one-dimensional problem is normally desired. However, for a non-linear problem (such as a flow equation system), the off-diagonal Jacobian terms are encouraged to represent the strong coupling between different fluid variables. The nonlinearity could be clearer from both advection and source terms if Equation (2) is re-written in nonlinear variable form such as in Equation (7):

\[ U \equiv \begin{bmatrix} U_0 \\ U_1 \\ U_2 \end{bmatrix} : F(U) = \begin{bmatrix} \frac{U_1}{U_0} \\ \frac{U_1^2}{U_0} + p \\ \frac{U_1 U_2}{U_0} + \frac{U_1 p}{U_0} \end{bmatrix} : S = \begin{bmatrix} 0 \\ \frac{1}{2D_h} |U_1| \frac{U_1}{U_0} - U_0 g_x \\ H_w a_w (T_f - T_w) + \frac{U_1}{U_0} \left( \frac{1}{2D_h} |U_1| \frac{U_1}{U_0} - U_0 g_x \right) \end{bmatrix} \]  

(7)

The couplings between those flow equation variables are either in the form of direct coupling, (such as \( U_1 \) shown as mass flux in the continuity equation) or in the form of indirect coupling (such as that due to pressure and temperature dependence over those primary variables \( (U_i) \)).
matrix of the exact Jacobian for such a simple flow equation system is shown in Figure 3. Those tridiagonal terms represent the nodal and thus physics connectivity of the one-dimensional flow problem. Those non-tridiagonal terms represent the coupling between different flow equation variables. For example, the row-1, column-3 element is the Jacobian matrix represents that $\rho u$ is coupled as the flux term in the continuity equation.

![Figure 3. Jacobian matrix for a single pipe flow problem (with lines representing non-zero entries in the matrix)](image1)

![Figure 4. Jacobian matrix for two pipes, with an ideal pump flow problem (with lines and squares representing non-zero entries in the matrix)](image2)

2.4.2. Two Pipes with an Ideal Pump Case

Adding an ideal pump in order to connect two pipes and to drive the flow does not add an additional degrees of freedom into the nonlinear system. However, it does treat the coupled boundary conditions between those two pipes as explained in Section 2.2. As shown in Figure 4,
the Jacobian matrix of this problem looks almost exactly the same as that of flow in a single pipe (as shown in Figure 3). In Figure 4, each variable vector is divided into two sections, representing a two pipes simulation. Also, an important difference between those two Jacobian matrices is shown as squares in Figure 4, which represent the coupling between the two ends of these two pipes or, in other words, how the information propagates from the Pipe (1) outlet to the Pipe (2) inlet.

![Figure 5. Jacobian matrix for conjugate heat transfer problem (with lines and squares representing non-zero entries in the matrix)](image)

### 2.4.3. Conjugate Heat Transfer Case

The physics of the conjugate heat transfer are straightforward (see Section 2.3). However, it is a typical multi-physics and multi-mesh coupling problem; therefore, the Jacobian terms exhibit a more complex form. As shown in Equations (5) and (6), the conjugate heat transfer is a two-way coupling between the fluid energy equation and the solid heat conduction equation. It should be noted that the coupling takes place only on the fluid-solid interface (i.e., between a fluid node and its adjacent solid surface node). The Jacobian term representing the effect of a solid surface temperature perturbation on fluid energy equation is straightforward and is in the following form:

\[
\frac{\partial}{\partial T_s} \int \phi H_w a_w (T_f - T_s) d\Omega
\]

(8)

The Jacobian term representing the effect of fluid equation variables perturbations on the solid heat conduction equation is more complicated due to fluid temperature dependence on those fluid primary variables \( (U \text{ in Equation (2) or (7))} \), as shown in Equation (9):

\[
\frac{\partial}{\partial U_i} [H_w (T_s - T_f)] = \frac{\partial}{\partial T_f} [H_w (T_s - T_f)] \frac{\partial T_f}{\partial U_i} \quad i = 0, 1, 2
\]

(9)
To summarize, the Jacobian matrix for a multi-physics and multi-mesh conjugate heat transfer problem is shown in Figure 5. In Figure 5, red squares represent the conjugate heat transfer coupling between fluid variables and the solid temperature variable (on surface only). For simplicity, the problem shown in Figure 5 assumes that the pipe is discretized into three elements (four nodes).

3. CASE STUDY AND RESULTS

In this section, a case study of conjugate heat transfer in a counter-current heat exchanger is presented. Both the simulation results and analytical solutions are presented and compared. The comparison between these two serves as an initial verification effort, because a more strict verification and validation process is necessary in the future. A numerical study also was done to investigate the effect of a preconditioning matrix approximation of the exact Jacobian matrix level on the numerical convergence.

This first part presents the simulation results of a simplified counter current flow heat exchanger (as shown in Figure 6) of which the flows in the primary pipe and in the secondary pipe are in opposite directions. The inlet temperatures are 400 K and 300 K for the primary pipe and the secondary pipe, respectively. The inlet velocities are both set to be 1 m/s on both sides. Both the primary and secondary pipes are given the same flow area, $7.85398 \times 10^{-5}$ m$^2$, and the same hydraulic diameter of 0.02 m. Heat transfer coefficients in both pipes are set to be identical at $1.0 \times 10^4$ W/m$^2$-K. The heating wall area density per unit fluid volume, $a_w$, in both pipes is the same, 400 m$^{-1}$. An analytical fluid temperature difference between the primary and the secondary sides for this problem could be obtained from Equation (10):

\[
\ln \left( \frac{\Delta T(x)}{\Delta T(0)} \right) = -\mu k_e a_w A_{pipe} x
\]

in which $\Delta T$ is the fluid temperature difference between the primary and secondary sides; and $\mu$ is a parameter related to the mass flow rates, $\dot{m}$, and fluid specific heat capacities, $C_p$, in both sides, which is defined as:

\[
\mu = \frac{1}{\dot{m}_1 C_{p,1}} - \frac{1}{\dot{m}_2 C_{p,2}}
\]

Figure 6. Schematic drawing of a counter current heat exchanger.

The effective heat transfer coefficient between the primary and the secondary side, $k_e$, is defined as:
in which, \( h_1 \) and \( h_2 \) are heat transfer coefficients in primary and secondary sides, respectively; and \( \delta_s \) and \( k_s \) are the thickness and thermal conductivity of the plate between primary and secondary sides, respectively. Figure 7 shows the comparison of the fluid temperature differences between the primary and the secondary sides along the heat exchanger pipe. The comparison indicates that the simulation results agree well with the analytical solution, with a maximum relative error around \( 7 \times 10^{-4} \).

A further numerical study was performed to investigate the effect of a preconditioning matrix on numerical convergence. For a same problem, several simulations were performed, with different time steps and different preconditioning matrix options. These different preconditioning matrix options represent different levels of approximation to the exact Jacobian matrix, including (1) a full exact Jacobian matrix as depicted in Figure 5; (2) a full exact Jacobian matrix, without all conjugate heat transfer terms (depicted as discrete squares in Figure 5); and (3) only tridiagonal terms of the full Jacobian matrix.

The results of numerical convergence rate, in terms of total number of residual evaluations, are summarized in Table I. It is clear that simulations using the full exact Jacobian matrix as preconditioning matrix (option 1) took the least number of residual evaluations and, not surprisingly, simulations using the tridiagonal matrix (option 3) took the most number of residual evaluations. Simulations using the option 2 preconditioning matrix, took about 25% more residual evaluations than those using option 1. A clear trend could be found: the higher approximation level of the preconditioning matrix to the exact Jacobian matrix, the better the convergence is.

However, it raises a question about the closeness of preconditioning matrix to the exact Jacobian matrix. From the observation, it would be the most efficient way to have a full exact Jacobian matrix provided as the preconditioning matrix for Krylov’s method. However, this is not an optimized option for two reasons: (1) an exact Jacobian matrix could be difficult and expensive to obtain for most applications, and (2) if an exact Jacobian matrix could be obtained, a direct Newton’s method rather than the Jacobian-free version of Newton’s method could be directly applied (the direct Newton’s method is also shown in line 1 of Table I for the purpose of

\[
k_e = \frac{1}{\frac{1}{h_1} + \frac{\delta_s}{k_s} + \frac{1}{h_2}}
\]
comparison). At this point, it circles back to the same question, if the approximated Jacobian matrix is provided as the preconditioning matrix, how close should the approximation matrix be? This could depend on the specific problem; therefore, a deep understanding of the problem is required. In this sense, a physics-based preconditioning method (such as the semi-implicit pressure-corrected ICE method [10]) might be a more reasonable approach. This needs further investigation in the future.

Table 1. Total number of residual evaluations with different time steps and preconditioning matrix options, with a total number of time steps equaling 10.

<table>
<thead>
<tr>
<th>Simulation Options</th>
<th>Time step size (s)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0×10⁻³</td>
<td>5.0×10⁻³</td>
<td>1.0×10⁻²</td>
</tr>
<tr>
<td>Newton’s Method</td>
<td>62</td>
<td>63</td>
<td>101</td>
</tr>
<tr>
<td>Full Jacobian</td>
<td>126</td>
<td>184</td>
<td>229</td>
</tr>
<tr>
<td>Without Conjugate Heat Transfer Terms</td>
<td>156</td>
<td>229</td>
<td>302</td>
</tr>
<tr>
<td>Tridiagonal Jacobian</td>
<td>1,129</td>
<td>2,417</td>
<td>3,544</td>
</tr>
</tbody>
</table>

4. CONCLUSIONS

In this paper, the fully implicit method is introduced, in solving multi-mesh flow and conjugate heat transfer problems in reactor safety analysis simulations with RELAP-7. In reactor safety analysis codes, the complicated components are highly simplified as interconnected one-dimensional and zero-dimensional components. This major difference, when compared to general single-mesh simulations, introduces uniqueness and challenges in the treatment of coupling between heterogeneous meshes and different physics.

Two important physics generally present in reactor safety analysis (i.e., flow and conjugate heat transfer) have been discussed. The different cases of (1) single physics on the same mesh, such as fluid flow in a pipe; (2) single physics coupling between meshes, such as flow in two pipes driven by an ideal pump; and (3) multi-physics coupling between multi-meshes, such as conjugate heat transfer in a counter-current heat exchanger, are discussed in detail. The simulation results of the counter-current flow heat exchanger study case are discussed and compared to the analytical solutions. The comparison shows that the simulation results agree with analytical solutions very well. A further numerical study also was performed to investigate the preconditioning matrix’s effect on numerical convergence rate. For this study, the results show that the closer the preconditioning matrix is to the exact Jacobian matrix, the better the convergence behavior. However, it raises questions about how one should approximate the preconditioning matrix to the exact Jacobian matrix and when one should use the physics-based preconditioning method. These questions warrant further studies in the future.
ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy, under Department of Energy Idaho Operations Office Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

REFERENCES