INL/EXT-17-43108
Revision 1

Light Water Reactor Sustainability Program

Preliminary Study on the Suitability of a Second-Order Reconstructed Discontinuous Galerkin Method for RELAP-7 Thermal-Hydraulic Modeling

September 2017

DOE Office of Nuclear Energy

INL/EXT-17-43108
Revision 1

Light Water Reactor Sustainability Program

Preliminary Study on the Suitability of a Second-Order Reconstructed Discontinuous Galerkin Method for RELAP-7 Thermal-Hydraulic Modeling

Yidong Xia
Joshua Hansel
Ray A. Berry
David Andrs
Richard C. Martineau

September 2017

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov/lwrs

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Abstract

This document presents a preliminary study on the suitability of a second-order reconstructed discontinuous Galerkin (rDG) method for RELAP-7 thermal-hydraulic modeling. The document begins with a brief description of the governing equations for compressible, two-phase vapor and liquid flow, with a presentation of the seven-equation formulation details. A comparative study between the second-order rDG method and the RELAP-7's finite element method (FEM) with a entropy viscosity method (EVM) based numerical stabilization scheme (namely FEM-EVM) over a series of benchmark test problems is demonstrated. The intent for this suite of test problems is to provide baseline comparison data that demonstrate the performance of 1) the rDG solution and 2) the RELAP-7's FEM-EVM solution (with RELAP-7 code version dated August 15, 2017), on problems from single- to specific, limited two-phase flows. For all the test problems in this document, the rDG solutions were obtained with a second-order, two-step, explicit strong stability preserving Runge-Kutta time integration method. The computational results clearly indicate that the performance of the rDG method is superior to that of the RELAP-7's FEM-EVM method in all the test problems presented. Therefore, as far as the test problems in this document are considered, the second-order rDG method is recommended as an improved solution method option for RELAP-7.

Acknowledgment

The first author expresses appreciation to Dr. Ling Zou at the Idaho National Laboratory for many helpful, instructive, and fruitful discussions about two-phase flow models and nuclear thermal-hydraulics modeling.

Contents

Figures

Tables

1 Introduction

1.1 About RELAP-7

The RELAP-7 (Reactor Excursion and Leak Analysis Program) code is a next-generation nuclear reactor system safety analysis code being developed at the Idaho National Laboratory (INL). The code is based on a modern scientific software development framework MOOSE (Multi-Physics Object Oriented Simulation Environment) [3, 4] being developed at INL. The overall design goal of RELAP-7 is to take advantage of the previous thirty years of advancements in computer architecture, software design, numerical methods and physical models. The end result is expected to be a reactor systems analysis capability that retains and improves upon RELAP5's [5] and TRACE's [6] abilities and extends the analysis capability for all reactor system simulation scenarios.

The RELAP-7 project, which was started from 2012, is expected to become the main reactor systems simulation toolkit for the Light Water Reactor Sustainability (LWRS) program's Risk Informed Safety Margin Characterization (RISMC) effort and the next-generation toolkit in the RELAP reactor safety/systems analysis application series. The key to the success of RELAP-7 lies in the simultaneous advancement of physical models, numerical methods, and software design, while maintaining a solid user perspective. Physical models include both partial differential equations (PDEs) and ordinary differential equations (ODEs) and experiment-based closure models. RELAP-7 utilizes well-posed governing equations for two-phase flow, which can be strictly verified in a modern verification and validation effort. RELAP-7 uses modern computational methods, which allow implicit time integration, second-order schemes in both time and space, and strongly coupled multi-physics.

MOOSE is a development and runtime framework developed at INL for solving engineering problems computationally in a well planned and coordinated way. By using the MOOSE framework, the RELAP-7 code is developed by following the same modern software design paradigms used in other MOOSE development efforts. The code is easily coupled with other codes. MOOSE provides various options of numerical integration methods and mesh management for parallel computing.

Because RELAP-7 is an ongoing development effort, its foundational theories and numerical solution methods will evolve with periodic updates to keep it current with the state of development, implementation, and model revisions. A complete description of the technical details of the RELAP-7 code can be found in the RELAP-7 Theory Manual [2].

1.2 Objective of this Work

The well-posed, hyperbolic, two-phase flow equation system of RELAP-7 is adopted from Ref. [1] wherein the discrete equation method (DEM) was utilized with a finite volume method to prove the model's solution feasibility. Over the past six years of the RELAP-7 code development, however, the continuous Galerkin finite element method (commonly denoted as "FEM") has been employed as the numerical solution method for solving these governing equations. FEM has been a straightforward solution method option for thermal-hydraulic modeling in RELAP-7, largely because of the finite element modeling framework of MOOSE upon which the RELAP-7 code has been built. The FEM approach has advantages when multi-physics coupling is considered in a unified code development infrastructure. For example, the multi-physics and multi-dimensional analysis capabilities, such as radiation transport and fuel performance, are conveniently

obtained by coupling the current RELAP-7 code and other MOOSE-based applications (through MOOSE).

Despite the advantages of FEM in multi-physics coupling, however, the achievement of a viable, accurate and ultimately robust numerical stabilization scheme for FEM has been a challenge. The recent development of a novel numerical stabilization scheme for FEM — the entropy viscosity method (EVM) for the non-equilibrium, seven-equation, two-phase compressible flow model in RELAP-7 [7–11], has significantly advanced the state of the art for stabilized FEM methods, but has not achieved the desired robustness. An outstanding issue associated to the EVM method for RELAP-7 thermal-hydraulic modeling is that the performance of the EVM method depends critically on the tuning of its parameters. Thus the EVM method in RELAP-7 works well given parameters empirically tuned for certain benchmark problems, e.g. the water faucet and phase separation problems [11]. However, RELAP-7's FEM with the EVM stabilization has struggled for some problems that involve more complex physical phenomena of practical interest in nuclear engineering. Therefore, with its current robustness limitations, it is unclear whether the EVM method should be the final choice of stabilization method for RELAP-7.

Meanwhile, a second-order finite volume (FV) method variant from the class of reconstructed discontinuous Galerkin (rDG) methods [12–14], namely rDG(P0P1) [15], has been recently considered as an alternative solution method option for RELAP-7 thermal-hydraulic modeling. The FV method [16] is one of the most robust solution methods in the field of computational fluid dynamics (CFD), having been successfully employed in a wide range of research and industry CFD codes. As far as the class of seven-equation, two-phase compressible flow models are concerned, the FV method has been the most used solution method in an abundance of relevant literature, e.g. Refs. [17–25]. The FV method (and thus the rDG(P0P1) method) was not an option for RELAP-7 at its beginning, because of the missing FV option in the underlying MOOSE framework. Because of the recent MOOSE capability extension effort, the rDG(P0P1) method is now possible in MOOSE, e.g. see the example of rDG(P0P1)-based, multi-dimensional, compressible flow simulation code developed based on MOOSE [26]. In the rDG(P0P1) method, P0 indicates that a piecewise constant monomial is used to represent a DG solution, and P1 represents a reconstructed linear polynomial solution that is used to compute the fluxes across the cell boundaries.

The objective of this effort is to present a preliminary study of the suitability of the rDG(P0P1) method for RELAP-7 stabilized thermal-hydraulic modeling. For quick proof of concept, a stand-alone rDG code has been developed in this work. The rest of the document is organized as follows. First, the governing equations for the compressible two-phase vapor and liquid flow, with a presentation of the seven-equation formulation details are briefly presented in Section 2. Next, the rDG(P0P1) method is briefly described in Section 3. A comparative study between the rDG(P0P1) method and RELAP-7's FEM-EVM method over a series of benchmark test problems is demonstrated in Section 4. The intent for this suite of test problems is to provide baseline comparison data that demonstrate the performance of 1) the rDG(P0P1) solution and 2) RELAP-7's FEM-EVM solution (with the RELAP-7 code version dated August 15, 2017), on problems from single- to specific, limited two-phase flows. For all the test problems in this document, the rDG(P0P1) solutions were obtained with a second-order, two-step, explicit strong stability preserving Runge-Kutta (SSP-RK) time integration method [27], while a first-order, one-step, backward Euler implicit time integration method was used in the RELAP-7's FEM-EVM solution. Finally, conclusions are given in Section 5.

2 Seven-Equation Two-Phase Model

2.1 Governing Equations

The one-dimensional, seven-equation two-phase flow model with variable cross-section flow area employed in the RELAP-7 program [1, 2] can be described by the equations for the conservation of mass, momentum and total energy for each phase (vapor and liquid), respectively, plus a volume fraction evolution equation for the vapor phase. If we assume an unsteady, fully non-equilibrium state between the two phases, and ignore the friction, boiling, and convective heat transfer originated of the wall as well as the inter-phase friction, the governing equations can then be expressed with the following seven equations:

$$\frac{\partial \alpha_g A}{\partial t} + u_{int} A \frac{\partial \alpha_g}{\partial x} = A\mu(p_g - p_l) + \frac{\Gamma_{int,g} A_{int} A}{\rho_{int}} \tag{1}$$

$$\frac{\partial (\alpha\rho)_g A}{\partial t} + \frac{\partial (\alpha\rho u)_g A}{\partial x} = \Gamma_{int,g} A_{int} A \tag{2}$$

$$\begin{aligned} \frac{\partial (\alpha\rho u)_g A}{\partial t} + \frac{\partial \alpha_g (\rho u^2 + p)_g A}{\partial x} &= p_{int} A \frac{\partial \alpha_g}{\partial x} + p_g \alpha_g \frac{\partial A}{\partial x} + A\lambda(u_l - u_g) + \Gamma_{int,g} A_{int} u_{int} A \\ &\quad + \alpha_g \rho_g A \vec{g} \cdot \vec{n} \end{aligned} \tag{3}$$

$$\begin{aligned} \frac{\partial (\alpha\rho E)_g A}{\partial t} + \frac{\partial \alpha_g (\rho E + p)_g u_g A}{\partial x} &= p_{int} u_{int} A \frac{\partial \alpha_g}{\partial x} - \bar{p}_{int} A\mu(p_g - p_l) + \bar{u}_{int} A\lambda(u_l - u_g) \\ &\quad - \Gamma_{int,g} A_{int} \left(\frac{p_{int}}{\rho_{int}} - H_{g,int}\right) A + A_{int} h_{conv,g} (T_{int} - T_g) A \\ &\quad + \alpha_g \rho_g u_g A \vec{g} \cdot \vec{n} \end{aligned} \tag{4}$$

$$\frac{\partial (\alpha\rho)_l A}{\partial t} + \frac{\partial (\alpha\rho u)_l A}{\partial x} = -\Gamma_{int,g} A_{int} A \tag{5}$$

$$\begin{aligned} \frac{\partial (\alpha\rho u)_l A}{\partial t} + \frac{\partial \alpha_l (\rho u^2 + p)_l A}{\partial x} &= p_{int} A \frac{\partial \alpha_l}{\partial x} + p_l \alpha_l \frac{\partial A}{\partial x} + A\lambda(u_g - u_l) - \Gamma_{int,g} A_{int} u_{int} A \\ &\quad + \alpha_l \rho_l A \vec{g} \cdot \vec{n} \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{\partial (\alpha\rho E)_l A}{\partial t} + \frac{\partial \alpha_l (\rho E + p)_l u_l A}{\partial x} &= p_{int} u_{int} A \frac{\partial \alpha_l}{\partial x} - \bar{p}_{int} A\mu(p_l - p_g) + \bar{u}_{int} A\lambda(u_g - u_l) \\ &\quad + \Gamma_{int,g} A_{int} \left(\frac{p_{int}}{\rho_{int}} - H_{l,int}\right) A + A_{int} h_{conv,l} (T_{int} - T_l) A \\ &\quad + \alpha_l \rho_l u_l A \vec{g} \cdot \vec{n} \end{aligned} \tag{7}$$

where the subscripts g and l denote the physical quantities of the gaseous and liquid phase, respectively. The "gaseous state" is generic, thus the two-phase model discussed in this document can be applied to both the two-phase air-water and vapor-water scenarios. The solution state vector is

$$\left[\alpha_g A, \quad (\alpha\rho)_g A, \quad (\alpha\rho u)_g A, \quad (\alpha\rho E)_g A, \quad (\alpha\rho)_l A, \quad (\alpha\rho u)_l A, \quad (\alpha\rho E)_l A\right]^T,$$

where α_k, ρ_k, u_k, and E_k ($k = g$ or l) are the phasic volume fraction, density, velocity, and specific total energy, respectively, with the relation $\alpha_l = 1 - \alpha_g$. $A = A(x)$ is the variational cross-section area function. The rest of the physical variables in Eqs. (1) – (7) will be given next.

2.2 Velocity and Pressure Relaxation Terms

The interfacial pressure p_{int} is given as

$$p_{int} = \bar{p}_{int} - sgn(\frac{\partial \alpha_g}{\partial x}) \frac{Z_l Z_g}{Z_l + Z_g}(u_l - u_g) \tag{8}$$

where the mixture interfacial pressure $\bar{p}_{int}$ is given as

$$\bar{p}_{int} = \frac{Z_l p_g + Z_g p_l}{Z_l + Z_g}. \tag{9}$$

where $Z_g = \rho_g c_g$ and $Z_l = \rho_l c_l$, with c_g and c_l the speed of sound in the gaseous and liquid phase, respectively.

The interfacial velocity u_{int} is given as

$$u_{int} = \bar{u}_{int} - sgn(\frac{\partial \alpha_g}{\partial x}) \frac{p_l - p_g}{Z_l + Z_g} \tag{10}$$

where the mixture interfacial velocity $\bar{u}_{int}$ is given as

$$\bar{u}_{int} = \frac{Z_l u_l + Z_g u_g}{Z_l + Z_g}. \tag{11}$$

The velocity relaxation coefficient λ is given as

$$\lambda = \frac{1}{2} \mu Z_l Z_g \tag{12}$$

where μ is the pressure relaxation coefficient and given as

$$\mu = \frac{A_{int}}{Z_l + Z_g} \tag{13}$$

For simplicity, the interfacial area A_{int} can be defined as a polynomial function of α_g, e.g.

$$A_{int} = A_{int,max}[6.75\alpha_g^2(1 - \alpha_g)] \tag{14}$$

as used in [1], where $A_{int,max}$ is a user-input coefficient. RELAP-7 also allows another simple polynomial function [2]

$$A_{int} = a\alpha_l^2 + b\alpha_l + c \tag{15}$$

where $a = 4(A_{int,min} - A_{int,max})$, $b = -a$, and $c = A_{int,min}$, and $A_{int,min}$ and $A_{int,max}$ are user-input coefficients. A common choice for $A_{int,min}$ in RELAP-7 is $A_{int}^{min} = 0$, with which Eq. (15) can then be rearranged as

$$A_{int} = A_{int,max}[4\alpha_g(1 - \alpha_g)] \tag{16}$$

where $A_{int,max} = 1700$ is a common choice.

2.3 Inter-phase Mass and Energy Transfer Terms

The inter-phase mass transfer is given by

$$\begin{aligned}\Gamma_{int,g} &= \frac{h_{conv,l}(T_l - T_{int}) + h_{conv,g}(T_g - T_{int})}{h_{g,int} - h_{l,int}} \\ &= \frac{h_{conv,l}(T_l - T_{int}) + h_{conv,g}(T_g - T_{int})}{L_v(T_{int})}\end{aligned} \tag{17}$$

where the inter-phase temperature $T_{int} = T_{sat}(\bar{p}_{int})$ is the saturation temperature for a given pressure $\bar{p}_{int}$, and the inter-phase enthalpy for the gaseous and liquid phase $h_{g,int} = h_{g,sat}(T_{int})$ and $h_{l,int} = h_{l,sat}(T_{int})$, respectively. Instead of using closure models, $h_{conv,g}$ and $h_{conv,l}$ are treated as user-input coefficients in this work, with default choices to be $h_{conv,g} = 5,000$ and $h_{conv,l} = 50,000$, respectively.

To calculate T_{int}, the procedure in Refs. [1, 28] is adopted, as is the stiffened gas equation of state (SGEOS) used in the present work. This procedure can be very accurate provided the two reference states are picked sufficiently close to represent the experimental saturation curves as locally quasi-linear. Restrictions can occur near the critical point. But away from this point, wide ranges of temperatures and pressures can be considered [28]. For the SGEOS, the Gibbs free enthalpy of the gaseous and liquid phase is

$$\begin{aligned}g(\bar{p}_g, T_{int}) &= (\gamma_g c_{v,g} - q'_g)T_{int} - c_{v,g}T_{int}\ln\frac{(T_g)^{\gamma_g}}{(p_g + p_{\infty,g})^{\gamma_g - 1}} + q_g \\ g(\bar{p}_l, T_{int}) &= (\gamma_l c_{v,l} - q'_l)T_{int} - c_{v,l}T_{int}\ln\frac{(T_l)^{\gamma_l}}{(p_l + p_{\infty,l})^{\gamma_l - 1}} + q_l\end{aligned} \tag{18}$$

respectively. At thermodynamic equilibrium at the interface, the two phasic Gibbs free enthalpies are equal, i.e. $g(\bar{p}_g, T_{int}) = g(\bar{p}_l, T_{int})$. This implies:

$$\ln(\bar{p}_{int} + p_{\infty,g}) = a + \frac{b}{T_{int}} + c\ln(T_{int}) + d\ln(\bar{p}_{int} + p_{\infty,l}) \tag{19}$$

where

$$a = \frac{c_{p,l} - c_{p,g} + q'_g - q'_l}{c_{p,g} - c_{v,g}}, \quad b = \frac{q_l - q_g}{c_{p,g} - c_{v,g}}, \quad c = \frac{c_{p,g} - c_{p,l}}{c_{p,g} - c_{v,g}}, \quad d = \frac{c_{p,l} - c_{v,l}}{c_{p,g} - c_{v,g}} \tag{20}$$

Eq. (19) is nonlinear, but can be used to compute the theoretical curve $T_{sat}(p)$. using simple Newton iterative numerical procedure. With the $T_{sat}(p)$ determined, the heat of vaporization is calculated as

$$L_v(T_{sat}) = h_{g,int} - h_{l,int} \tag{21}$$

where $h_{g,int} = h_{g,sat}(T_{int})$ and $h_{l,int} = h_{l,sat}(T_{int})$ can be calculated through SGEOS, respectively. The toal enthalpy for the gaseous and liquid phase is

$$H_{g,int} = h_{g,int} + \frac{1}{2}u_{int}^2 \quad \text{and} \quad H_{l,int} = h_{l,int} + \frac{1}{2}u_{int}^2 \tag{22}$$

respectively. The interfacial density

$$\rho_{int} = \rho_{l,sat}(\bar{p}_{int}) \tag{23}$$

is also calculated through SGEOS.

3 A Second-Order Reconstructed Discontinous Galerkin Method

3.1 Overview

The class of reconstructed discontinuous Galerkin (rDG) methods, termed as PnPm schemes, were originally introduced by Dumbser et al. [29–31], where Pn indicates that a piecewise polynomial of degree of n is used to represent a discontinuous Galerkin (DG) solution, and Pm represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes. The PnPm schemes are designed to enhance the accuracy of the DG method by increasing the order of the underlying polynomial solution. The beauty of the PnPm schemes is that they provide a unified formulation for both the finite volume (FV) and DG methods, and contain both the classical cell-centered FV and standard DG methods as two special cases of PnPm schemes. When $n = 0$, i.e., a piecewise constant polynomial is used to represent a numerical solution, P0Pm is nothing but the classical high-order FV schemes, where a polynomial solution of degree m ($m \geq 1$) is reconstructed from a piecewise constant solution. When $m = n$, the reconstruction reduces to the identity operator, and the PnPn scheme yields a standard DG method. Many variants of the original rDG methods have been developed in recent years, for example, the hybrid HWENO+DG schemes by Balsara et al. [32], the least-squares reconstruction-based DG schemes by Luo et al. [12, 13, 33], and the class of Green-Gauss reconstruction-based hybrid DG/FV schemes by Zhang et al. [34, 35]. All of these schemes are able to improve the spatial accuracy of the underlying DG methods without significant extra cost in storage and computing time.

This work considers a second-order FV method variant from the PnPn schemes, namely rDG(P0P1) [15], as an alternative solution method for RELAP-7 thermal-hydraulic modeling. In this method, P0 indicates that a piecewise constant monomial is used to represent a DG solution, and P1 represents a reconstructed linear polynomial solution that is used to compute the fluxes across the cell boundaries. As it was mentioned in Section 1, the FV method (and thus rDG(P0P1)) was not an option for RELAP-7 at its beginning, because of the missing capability in the underlying MOOSE framework. Because of recent MOOSE capability extensions, a generalized design of the rDG(P0P1) method has been implemented in MOOSE. Based on the rDG(P0P1) implementation in MOOSE, an rDG(P0P1)-based, multi-dimensional (1D, 2D and 3D), compressible flow simulation code, BIGHORN, has been developed [26]. The similar design can be applied to RELAP-7. The rest of this section will briefly describe the rDG(P0P1) method for single-phase compressible flows in the 1D domain with variable cross-section area. Extension of this method to coupled two-phase gaseous and liquid flows based on the seven-equation model will be discussed in the future work.

3.2 Discretization in Space

By neglecting the velocity and pressure relaxation, inter-phase mass and energy transfer, and external force terms in the seven-equation system of Eqs. (1) – (7), the gaseous and liquid phases become decoupled, and consequently the system can be reduced to a system of Euler-like equations for each individual phase [36]:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} - \mathbf{S} = 0, \tag{24}$$

where the volume fraction evolution equation for the gaseous phase as well as the subscripts for denoting the phases is dropped. In Eq. (24), the conserved state vector $\mathbf{U}$ and flux vector $\mathbf{F}$ become $[\rho A, \rho uA, \rho EA]^T$ and $\left[\rho uA, \rho u^2 A, (\rho E + p)uA\right]^T$, respectively. The remaining source term $\mathbf{S}$ is equal to $[0, p(\partial A/\partial x), 0]^T$.

The rDG(P0P1) method is used to discretize Eq. (24) in space. In 1D, the computational domain Ω is divided by a set of non-overlapping line segment cells Ω_i. In each cell, the integral form of the governing equations is required to be satisfied,

$$\int_{\Omega_i} \frac{\partial \mathbf{U}}{\partial t}\, \mathrm{d}V + \int_{\Omega_i} \nabla \cdot \mathbf{F}\, \mathrm{d}V - \int_{\Omega_i} \mathbf{S}\, \mathrm{d}V = 0. \tag{25}$$

The conserved variable vector $\mathbf{U}_i$ are taken to be the unknowns and defined as cell-average solution variables by $\mathbf{U}_i(t) = \frac{1}{V_i}\int_{\Omega_i} \mathbf{U}(\mathbf{x},t)\, \mathrm{d}V$, where V_i is the length of cell Ω_i. The following equations can then be derived using the divergence theorem,

$$V_i \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} + \sum_{j\in N_i} \int_{\Gamma_{ij}} \mathbf{F}_{ij}\cdot\mathbf{n}_{ij}\, \mathrm{d}\Gamma + \sum_{\Gamma_{ib}\in\partial\Omega} \int_{\Gamma_{ib}} \mathbf{F}_{ib}\cdot\mathbf{n}_{ib}\, \mathrm{d}\Gamma - V_i\mathbf{S}_i = 0, \tag{26}$$

where $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ denotes an interior common face between cell Ω_i and Ω_j, $\Gamma_{ib} = \partial\Omega_i \cap \partial\Omega$ denotes a face on the boundary of Ω; and $\mathbf{n}_{ij}$ and $\mathbf{n}_{ib}$ are the unit vectors normal to face Γ_{ij} and Γ_{ib}, respectively. For each cell Ω_i, N_i represents a set of neighboring cells, Ω_j, sharing a common face, Γ_{ij}. In the 1D domain, $j = i-1$ or $i+1$. Because the numerical solution is discontinuous across the cell interfaces, the interface fluxes are not uniquely defined. The flux function $\mathbf{F}_{ij}\cdot\mathbf{n}_{ij}$ that appears in the second term of Eq. (26) can be replaced by a numerical flux function $\mathcal{H}(\mathbf{U}_{ij},\mathbf{U}_{ji},\mathbf{n}_{ij})$, where $\mathbf{U}_{ij}$ and $\mathbf{U}_{ji}$ are the conserved state vector at the left and right side of the cell interface ($i<j$). In the case of P0 solution, the solution in each cell is assumed to be constant in space. On any interior face, Γ_{ij}, the two states are simply $\mathbf{U}_{ij} = \mathbf{U}_i$ and $\mathbf{U}_{ji} = \mathbf{U}_j$. In order to guarantee consistency and conservation, $\mathcal{H}(\mathbf{U}_{ij},\mathbf{U}_{ji},\mathbf{n}_{ij})$ is required to satisfy $\mathcal{H}(\mathbf{U}_{ij},\mathbf{U}_{ji},\mathbf{n}_{ij}) = \mathbf{F}_i\cdot\mathbf{n}_{ij}$, and $\mathcal{H}(\mathbf{U}_{ij},\mathbf{U}_{ji},\mathbf{n}_{ij}) = -\mathcal{H}(\mathbf{U}_{ji},\mathbf{U}_{ij},\mathbf{n}_{ij})$. Similarly, the flux function on the domain boundary $\mathbf{F}_{ib}\cdot\mathbf{n}_{ib}$ should be determined by $\mathcal{H}(\mathbf{U}_{ib},\mathbf{U}_b,\mathbf{n}_{ib})$ with the use of appropriate boundary conditions satisfying the characteristic theory [37, 38].

Finally, the boundary integration in Eq. (26) is approximated using the one-point quadrature rule at the midpoint of the face, and the semi-discrete form of the equations is written as

$$V_i \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} + \sum_{j\in N_i} \mathcal{H}(\mathbf{U}_{ij},\mathbf{U}_{ji},\mathbf{n}_{ij}) + \sum_{\Gamma_{ib}\in\partial\Omega} \mathcal{H}_b(\mathbf{U}_{ib},\mathbf{U}_b,\mathbf{n}_{ib}) - V_i\mathbf{S}_i = 0, \tag{27}$$

where the flux function is approximated using Batten's HLLC (Harten-Lax-van Leer-Contact) approximate Riemann solver [39] in this work. This HLLC scheme has the following properties: 1) exact preservation of isolated contact and shear waves, 2) positivity-preserving of scalar quantity, and 3) enforcement of entropy condition.

By assembling contributions from all the cells and moving all the non-time derivative terms of Eq. (27) to the right-hand side, a set of ordinary differential equations (ODEs) governing the evolution of the discrete solution in time can be written as

$$\boldsymbol{M}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = -\mathbf{R}(\mathbf{U}), \tag{28}$$

where $\boldsymbol{M}$ is the mass matrix, $\mathbf{U}$ is the global vector of the degrees of freedom, and $\mathbf{R}$ is the residual vector. $\boldsymbol{M}$ has a block diagonal structure that couples the degrees of freedom of the unknown vector associated to $\mathbf{U}_i$ only within Ω_i. Consequently, the inverse of $\boldsymbol{M}$ can be computed considering one cell at a time and stored in advance. A multi-stage (up to three stages), explicit strong stability preserving Runge-Kutta time integration method [27] can be used to advance the solution of Eq. (28) in time.

3.3 Piecewise Linear Reconstruction

To achieve spatial accuracy higher than the first order, the solution in each cell needs to be reconstructed. The simplest approach is piecewise linear reconstruction, which can be performed on the conserved variables, primitive variables such as $\mathbf{Q} = [\rho, u, p]^T$, or characteristic variables. Though the conserved variables satisfy the conservation of the reconstructed solution, they are not as robust as the primitive variables for ensuring the positivity of density. The characteristic variables lead to more accurate schemes, but at a higher computational cost. This work adopts the primitive variables in the solution reconstruction process. From now we denote $\bar{\mathbf{Q}}_i$ as the underlying P0 primitive variables in each cell, and $\mathbf{Q}_i$ the P1 linear polynomial primitive variables to be reconstructed. Following a reconstruction process, the two reconstructed states $\mathbf{Q}_{ij}$ and $\mathbf{Q}_{ji}$ at the common cell face Γ_{ij} can be obtained to compute the inviscid fluxes $\mathcal{H}(\mathbf{Q}_{ij}, \mathbf{Q}_{ji}, \mathbf{n}_{ij})$. By assuming the gradient of $\mathbf{Q}_i$ at the center of cell Ω_i, the reconstructed solution in Ω_i is $\mathbf{Q}(x) = \bar{\mathbf{Q}}_i + (x - x_i) \cdot \nabla \mathbf{Q}_i$. The two states can be obtained at the face Γ_{ij} (i.e., $x = x_{ij}$):

$$\mathbf{Q}_{ij} = \bar{\mathbf{Q}}_i + (x_{ij} - x_i) \cdot \nabla \mathbf{Q}_i \quad \text{and} \quad \mathbf{Q}_{ji} = \bar{\mathbf{Q}}_j + (x_{ij} - x_j) \cdot \nabla \mathbf{Q}_j,$$

where $x_{ij} = x_{i+1/2}$ when $j = i + 1$. In order to ensure monotone solutions, a limiter function needs to be calculated in each cell. With a limiter function applied, the limited reconstructed values can be written as

$$\mathbf{Q}_{ij} = \bar{\mathbf{Q}}_i + \phi_i (x_{ij} - x_i) \cdot \nabla \mathbf{Q}_i \quad \text{and} \quad \mathbf{Q}_{ji} = \bar{\mathbf{Q}}_j + \phi_j (x_{ij} - x_j) \cdot \nabla \mathbf{Q}_j.$$

Since $\mathbf{Q}$ has several components, the limiter function is computed for each component. On multi-dimensional unstructured meshes, popular limiters include the min-max limiter of Barth & Jespersen [40] and Venkatakrishnan's limiter [41].

The reconstruction methods mentioned above are dimension-agnostic, meaning they can be applied on one- and multi-dimensional unstructured meshes of arbitrary shapes. However, on one- and multi-dimensional structured meshes, the total variation diminishing (TVD) type slope limiters such as the minmod, superbee and MC slope limiters [16] are preferred over their dimension-agnostic counterparts, where they are usually more robust for obtaining high-resolution monotonic solutions. One choice of slope in one dimension that gives second-order accuracy for smooth solutions while still satisfying the TVD property is the *minmod slope limiter*:

$$\phi_i \nabla Q_i = \text{minmod}\left(\frac{Q_i - Q_{i-1}}{\Delta x}, \frac{Q_{i+1} - Q_i}{\Delta x} \right) \tag{29}$$

where the minmod function of two arguments is defined by

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0 & \text{if } ab \leq 0. \end{cases} \tag{30}$$

If a and b have the same sign, then this selects the one that is smaller in modulus, else it returns zero. Rather than defining the slope on the ith cell by always using the downwind difference (which would give the LaxWendroff method), or by always using the upwind difference (which would give the BeamWarming method), the minmod method compares the two slopes and chooses the one that is smaller in magnitude. If the two slopes have different sign, then the value Q_i must be a local maximum or minimum, and it is easy to check in this case that we must set $\phi_i = 0$ in order to satisfy the TVD condition. The minmod method does a fairly good job of maintaining good accuracy in the smooth hump and also sharp discontinuities in the square wave, with no oscillations.

4 Benchmark Test Problems

Verification testing is part of our software quality control process and ensures that the developed code is solving problems of interest to RELAP-7 [11], while meeting the necessary design requirements. This section identifies a number of verification benchmark problems of interest to RELAP-7, and summarizes the solutions of the second-order reconstructed discontinuous Galerkin (rDG(P0P1)) code to the problems. The tests were organized by the methods and physics to enable a quick survey of each component in the code. Each test has a subsection describing why the test is included and how to set it up (§Problem Setup), and numerical results and remarks (§Results and Remarks).

The rDG(P0P1) code is written in Fortran 90/95, and compiled with the GNU Fortran (6.2.0) compiler with optimization level 3. All of the rDG(P0P1) results were obtained by running the code in serial computing on a MacBook Pro (2013) with a 2.6 GHz Intel Core i7 processor.

The version of RELAP-7 code used for solving the test problems corresponds to RELAP-7's Git version control commit number **e71678ebf30f9bfc401000a6a07d37be9336ad1e** as of **August 15, 2017**.

Definition of the few notations that appear in the figures in the section is given as below:

Method	Denotation
rDG(P0P0)	first-order rDG method
rDG(P0P1)	second-order rDG method with minmod slope limiter
RELAP-7 FEM	second-order FEM method without any numerical stabilization scheme
RELAP-7 FEM-EVM(0)	second-order FEM method with constant entropy viscosity stabilization
RELAP-7 FEM-EVM(1)	second-order FEM method with variable entropy viscosity stabilization

The stiffened-gas equation of state (SGEOS) is used within its range of applicability for calculating the thermodynamic properties of fluids under consideration. Although SGEOS is not as physically accurate as the steam and water equation of state based on the latest IAPWS Guideline [42], the SGEOS for the simulation problems in this document is sufficient, as the focus in the present study is the viability and robustness of the numerical discretization methods under consideration.

The SGEOS parameters used in single-phase ideal gas flow problems are listed in Table 1, where the parameters are non-dimensionalized so that the results can be conveniently compared with available analytical solution data in literature.

The SGEOS parameters used for a single-phase liquid/vapor convergent-divergent nozzle problem in Section 4.4 were taken from Ref. [1], as shown in Table 2. Though not used for two-phase calculations in Ref. [1] due to a manuscript error, those parameters were still used in Section 4.4, as the results can be readily verified by comparing with those in Ref. [1] for single-phase calculations.

The SGEOS parameters used for a co-existing two-phase liquid and vapor flow simulation problem in Section 4.5. were taken from RELAP-7 Theory Manual [2], and are shown in Table 3. A T_{sat} curve can be established with these parameters, as shown in Fig. 1.

Table 1. Stiffened-gas equation of state non-dimensionalized parameters for ideal gas.

Gas	γ	p_∞	q	q'	c_v
Air	1.4	0	0	0	$1/\gamma(\gamma-1)$

Table 2. Stiffened gas equation of state parameters for water and its vapor, from [1].

Water	γ	p_∞ (Pa)	q ($\mathrm{J\cdot kg^{-1}}$)	q' ($\mathrm{J\cdot kg^{-1}\cdot K^{-1}}$)	c_v ($\mathrm{J\cdot kg^{-1}\cdot K^{-1}}$)
Liquid phase	2.04	8.5×10^8	-1151×10^3	0	2069
Vapor phase	1.34	0	1968×10^3	-23×10^3	1265

Table 3. Stiffened gas equation of state parameters for water and its vapor, from [2].

Water	γ	p_∞ (Pa)	q ($\mathrm{J\cdot kg^{-1}}$)	q' ($\mathrm{J\cdot kg^{-1}\cdot K^{-1}}$)	c_v ($\mathrm{J\cdot kg^{-1}\cdot K^{-1}}$)
Liquid phase	2.35	10^9	-1167×10^3	0	1816
Vapor phase	1.43	0	2020×10^3	-23×10^3	1040

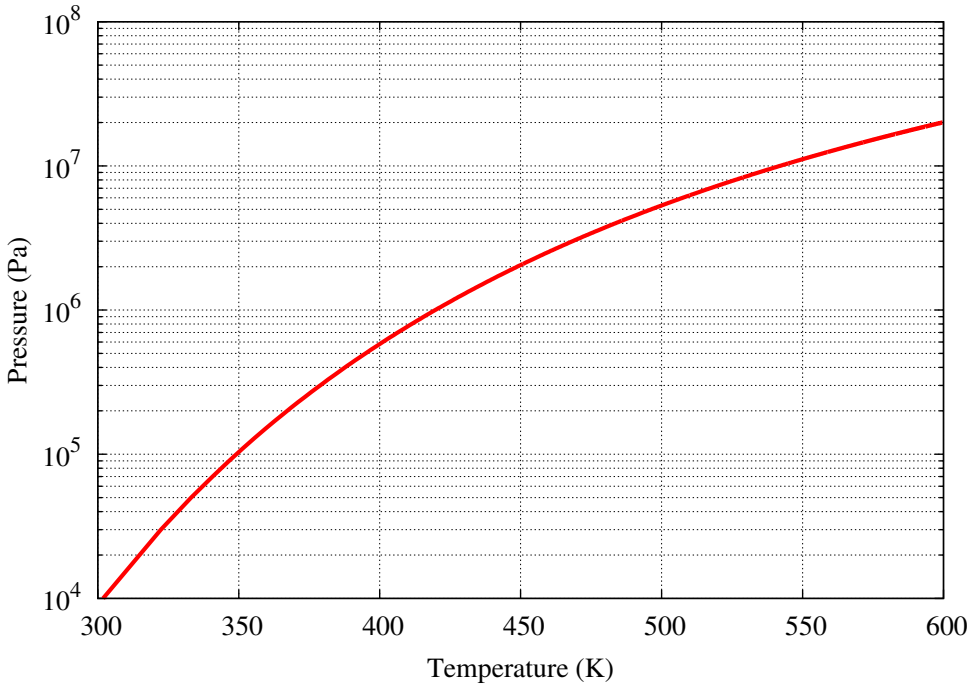

Figure 1. A $T_{sat}(p)$ curve calculated with Eq. (19) given the input pressure ranging from 10^3 to 2×10^7 Pa.

4.1 Lax Shock Tube

4.1.1 Problem Setup

The Lax shock tube [43] is a Riemann problem used as a standard test problem in computational hydrodynamics. This problem is very suitable to be used for investigating the robustness of a spatial discretization scheme, as the shock wave involved in this problem is much stronger than that in the Sod shock tube problem, and thus is more challenging to solve. The set of SGEOS parameters in Table 1 was used for ideal gas in this test. The computational domain is bounded between $x = 0$ and 1, and contains 400 equi-distant elements. The free outflow boundary condition is imposed at both the left and right ends. The initial conditions are described as: $\rho = 0.445$ $u = 0.698$, $p = 3.528$ for $0 \leq x < 0.5$, and $\rho = 0.5$, $u = 0$, $p = 0.571$ for $0.5 < x \leq 1$. The simulation was started at $t = 0$, and terminated at $t = 0.15$. In the rDG simulations, a fixed timestep size of $dt = 1 \times 10^{-4}$ was used, and 1500 timesteps were used to complete each simulation.

4.1.2 Results and Remarks

The computed solution profiles for density, specific internal energy, velocity, Mach number, pressure and temperature are plotted in Figs. 2 - 7, respectively, and compared with analytical solutions to this problem. Remarks are given below.

- RELAP-7's FEM-EVM(1) delivered incorrect solution in the rarefaction region as well as near the left and right ends. Source of error might be from the treatment of free outflow boundary condition under the context of FEM-EVM(1) within RELAP-7.
- rDG(P0P1) delivered the best overall solution among the methods tested in this problem. rDG(P0P0), as first-order accurate and monotonic, delivered a more diffusive solution than rDG(P0P1).
- RELAP-7's FEM-EVM(0) delivered a much more diffusive solution than rDG(P0P0). In that regard, RELAP-7's FEM-EVM(0) is considered much less spatially accurate than the first-order rDG(P0P0).
- In comparison with rDG(P0P1), RELAP-7's FEM-EVM(1) exhibited similar accuracy in resolving the contact discontinuity and shock front. However, RELAP-7's FEM-EVM(1) rendered an oscillatory solution in the vicinity of shock front.

Finally, the CPU time measured in the simulations is listed in the following table.

Method	CPU time (seconds)
RELAP-7 FEM-EVM(0)	**185.347**
RELAP-7 FEM-EVM(1)	**185.347**
rDG(P0P0)	**0.253**
rDG(P0P1)	**0.351**

The CPU time indicates that rDG methods with explicit time stepping are an advantage for solving transient gas dynamic problems with wave propagation containing strong gradients.

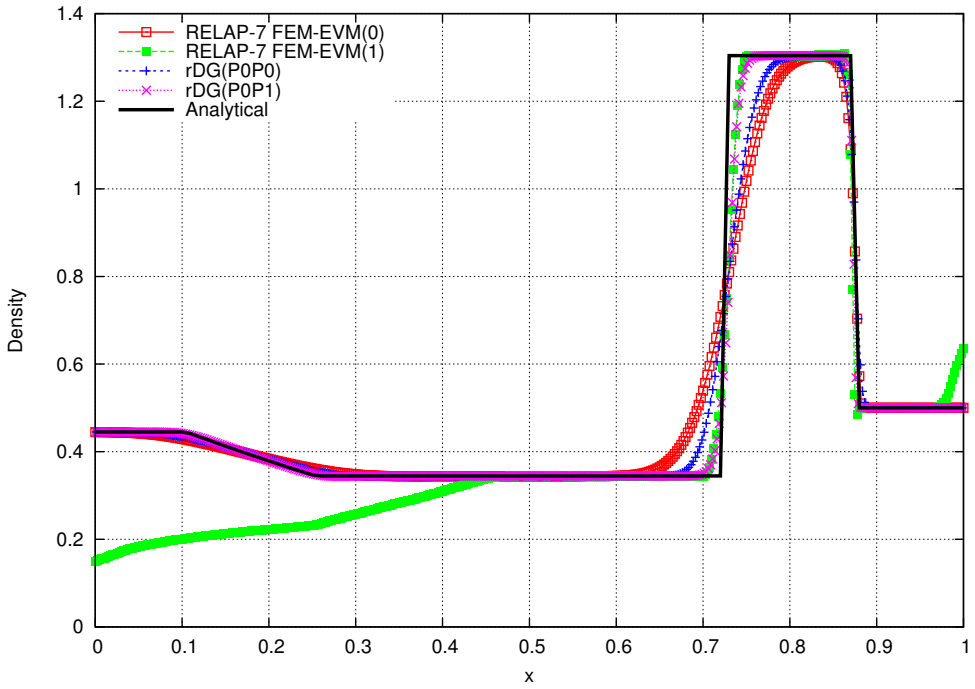

Figure 2. Computed density profiles at $t = 0.15$ for the Lax shock tube.

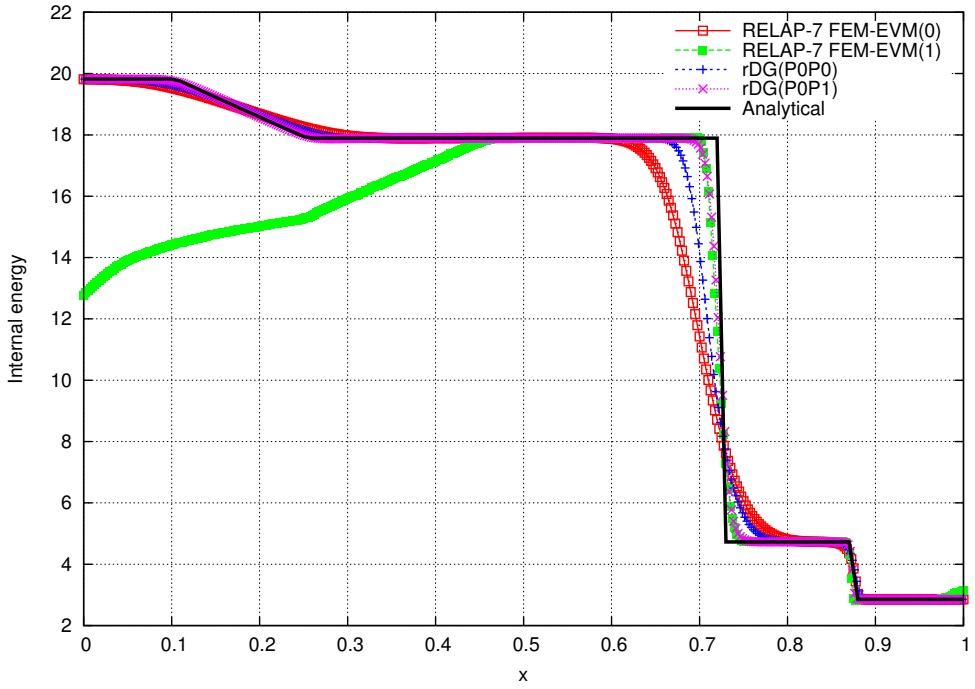

Figure 3. Computed specific internal energy profiles at $t = 0.15$ for the Lax shock tube.

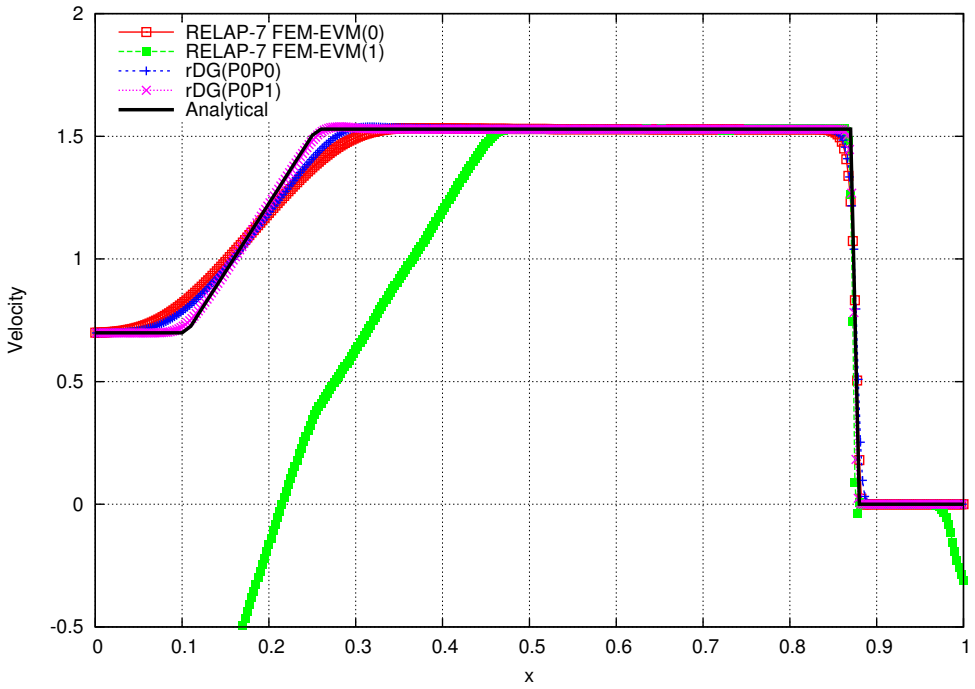

Figure 4. Computed velocity profiles at $t = 0.15$ for the Lax shock tube.

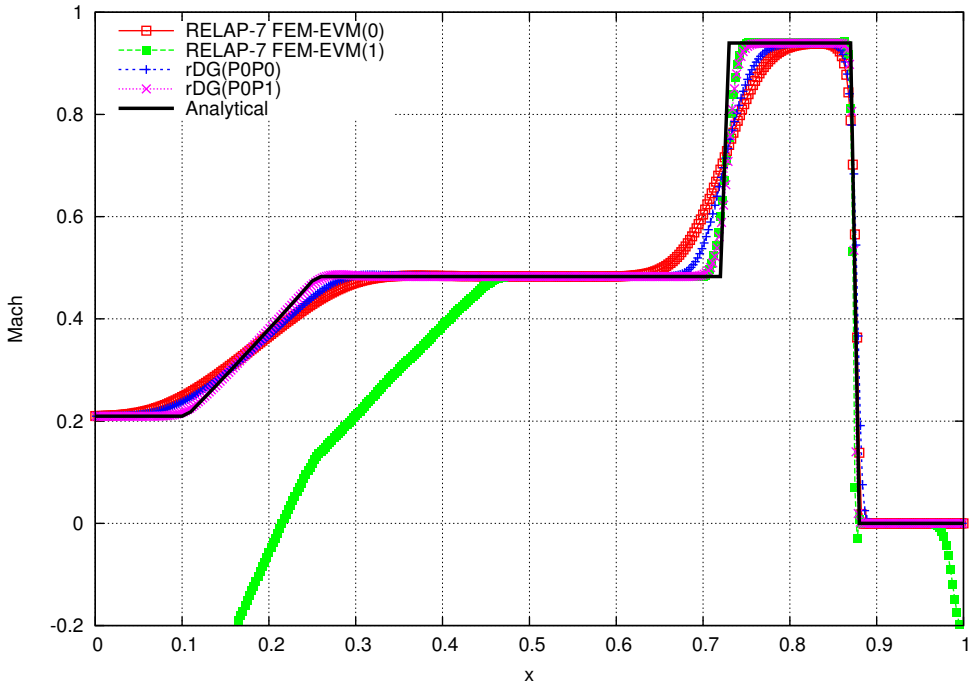

Figure 5. Computed Mach number profiles at $t = 0.15$ for the Lax shock tube.

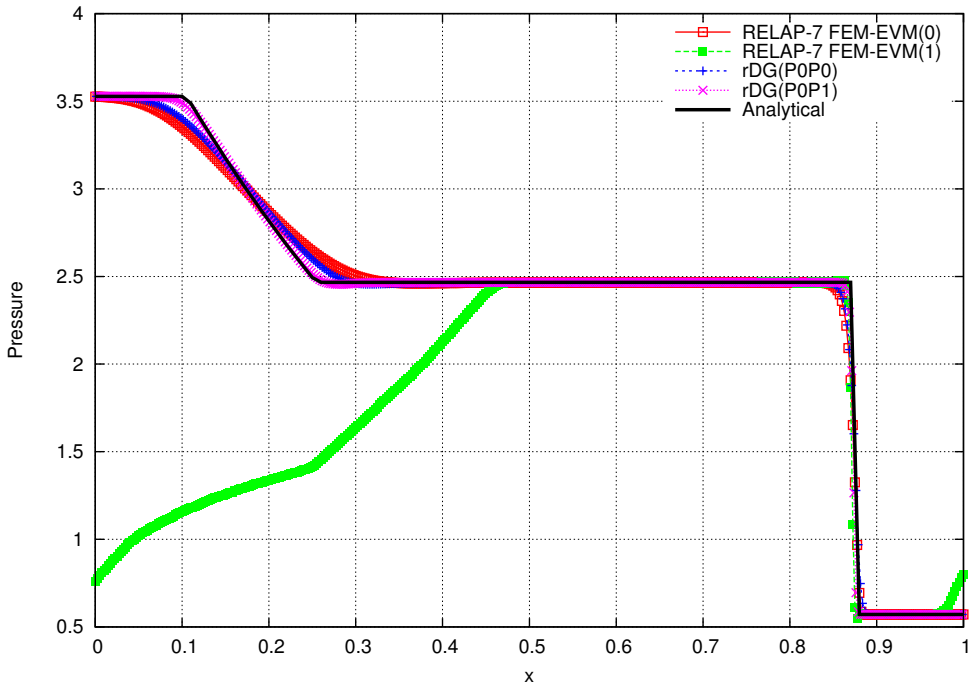

Figure 6. Computed pressure profiles at $t = 0.15$ for the Lax shock tube.

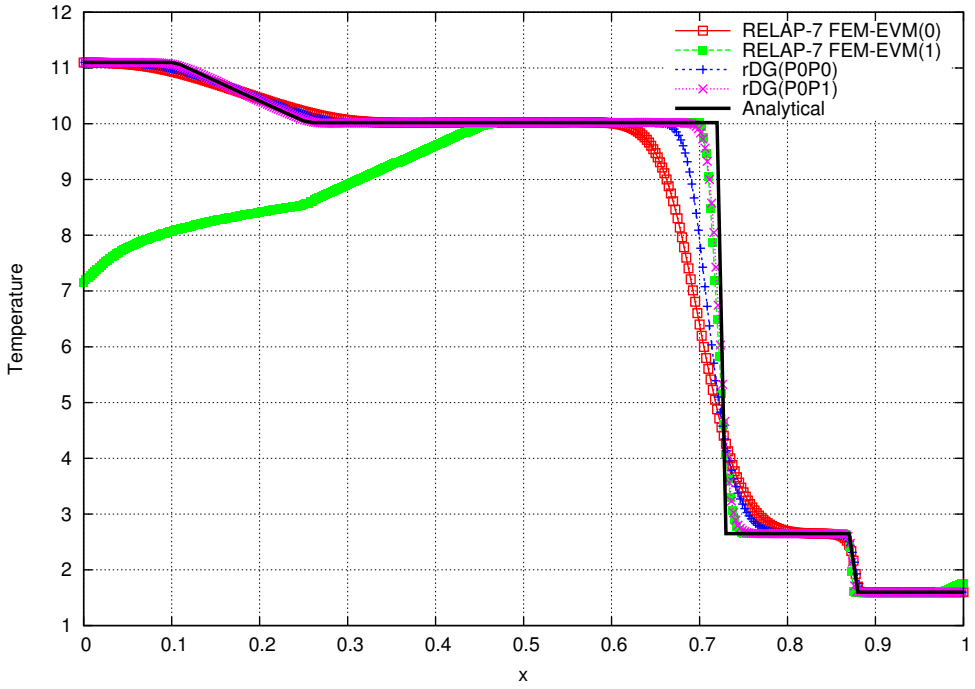

Figure 7. Computed temperature profiles at $t = 0.15$ for the Lax shock tube.

4.2 Vapor Shock Tube

This test consists of a single-phase vapor shock tube in a pipe bounded between $x = 0$ and 2 m [11, 44, 45]. The 1-D pipe was discretized with 400 uniform elements, and the wall boundary conditions were applied to the left and right ends. A diaphragm located at $x = 1$ m separated a high pressure chamber ($P = 2$ MPa and $T = 486.5$ K) and a low pressure chamber ($P = 1$ MPa and $T = 453.1$ K). The vapor properties were computed from the SGEOS with parameters in Table 3. The solution results for this problem recently reported in Ref. [11] used the IAPWS-95 Spline Based Table Lookup approach [42] to accurately compute the vapor properties. At $t = 0$ s, the diaphragm is removed, and the vapor initially at rest starts to develop contact, rarefaction, and shock waves. In the present test, the simulation was started at $t = 0$ s, and terminated at $t = 0.001$ s. In the rDG simulations, a fixed timestep size of $dt = 1 \times 10^{-6}$ s was used, and 1000 timesteps were used to complete each simulation.

4.2.1 Results and Remarks

The computed rDG solution profiles for density, specific internal energy, velocity, Mach number, pressure and temperature are plotted in Figs. 8 – 13, respectively, and compared with the RELAP-7's FEM-EVM solutions to this problem. Remarks are given below.

- rDG(P0P1) delivered the best overall solution among the methods tested in this problem. No oscillation is found at all in the rDG(P0P1) solution. rDG(P0P0) delivered a solution that is more diffusive than rDG(P0P1).
- RELAP-7's FEM-EVM(1) solution rendered non-physical spurious oscillations in the vicinities of rarefaction, contact discontinuities and shock wave. Alternative parameters in the EVM(1) method may reduce, but probably not completely eliminate those oscillations.
- RELAP-7's FEM-EVM(0) delivered a much more diffusive solution than rDG(P0P0). In that regard, RELAP-7's FEM-EVM(0) is considered much less spatially accurate than the first-order rDG(P0P0).

Finally, the CPU time measured in the simulations is listed in the following table.

Method	CPU time (seconds)
RELAP-7 FEM-EVM(0)	**12.551**
RELAP-7 FEM-EVM(1)	**17.272**
rDG(P0P0)	**0.164**
rDG(P0P1)	**0.229**

The CPU time indicates that rDG methods with explicit time stepping are in significant advantage in terms of computing time in this test problem.

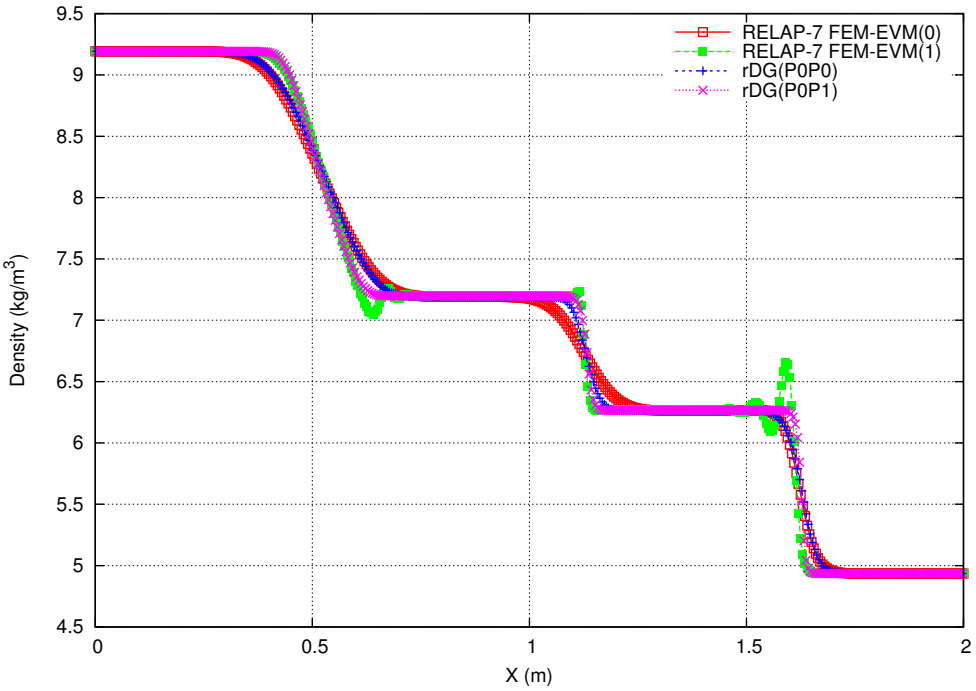

Figure 8. Computed density profiles at $t = 0.001$ for the vapor shock tube.

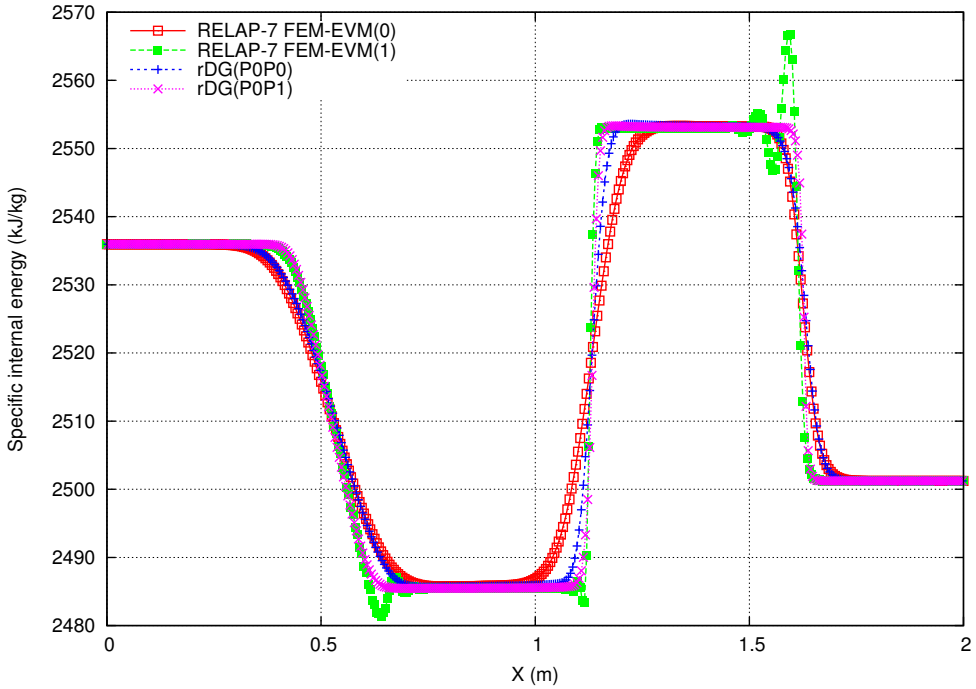

Figure 9. Computed specific internal energy profiles at $t = 0.001$ for the vapor shock tube.

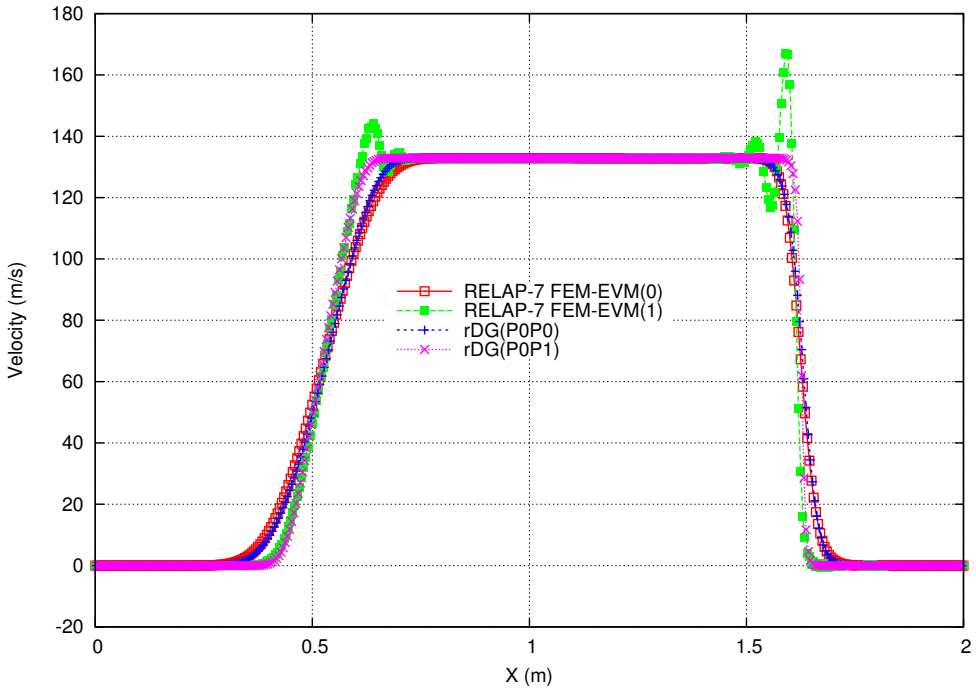

Figure 10. Computed velocity profiles at $t = 0.001$ for the vapor shock tube.

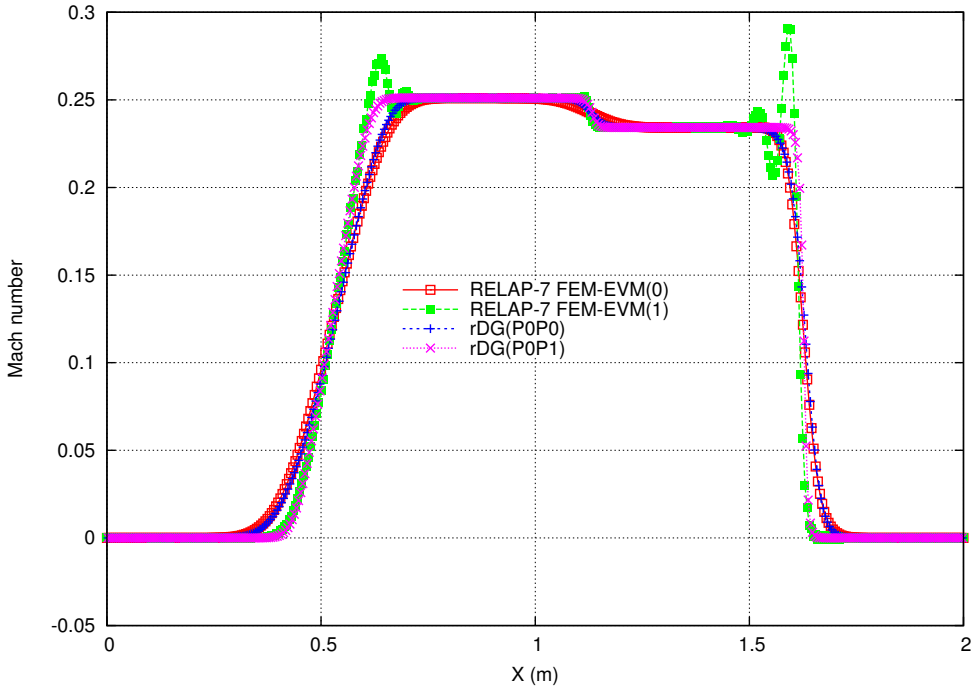

Figure 11. Computed Mach number profiles at $t = 0.001$ for the vapor shock tube.

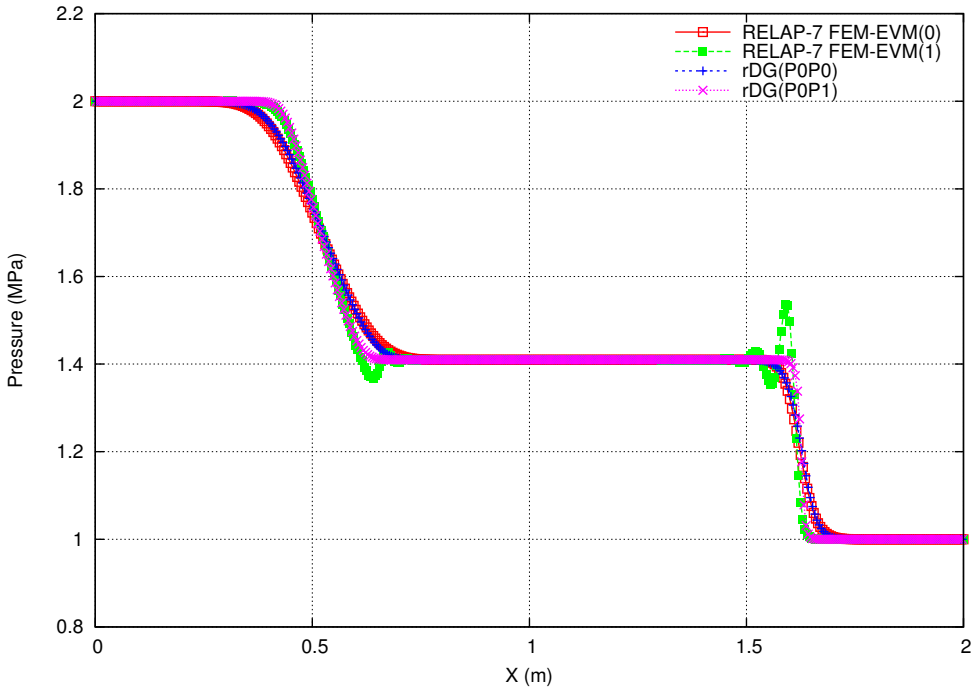

Figure 12. Computed pressure profiles at $t = 0.001$ for the vapor shock tube.

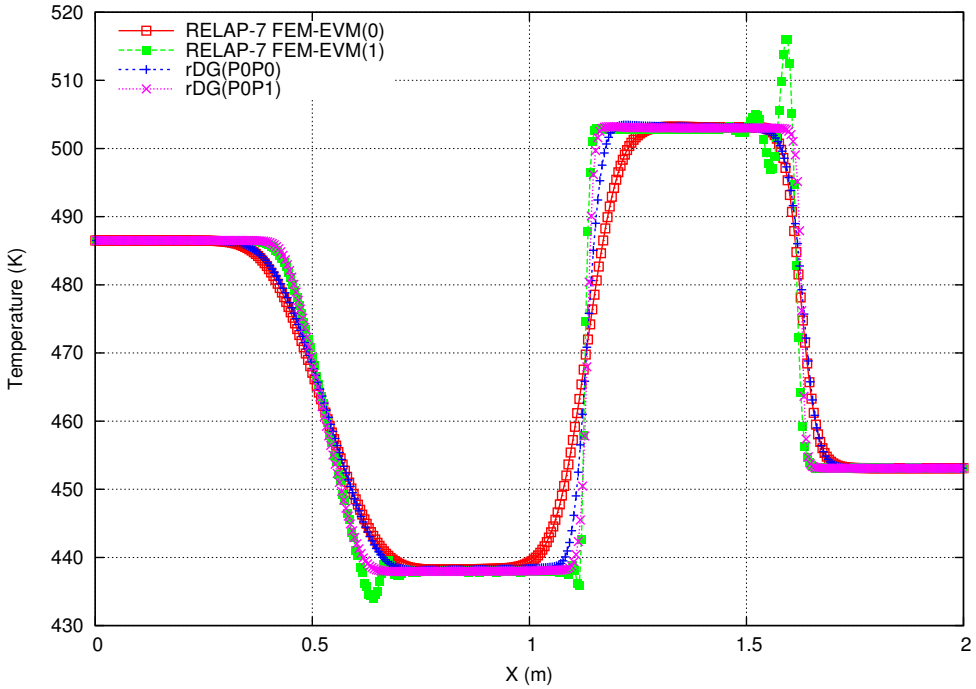

Figure 13. Computed temperature profiles at $t = 0.001$ for the vapor shock tube.

4.3 Liquid Water Shock Tube

Again we draw a test problem from Ref. [11] to which RELAP-7 has recently been applied. This problem presents a numerical solution of a liquid water shock tube in a 1-D pipe bounded between $x = 0$ and 10 m. The pipe was discretized with 400 uniform elements. The initial conditions consisted of two chambers separated by a diaphragm. The left chamber was filled with high pressure water ($P = 10$ MPa), and the right chamber was filled with low pressure water ($P = 0.1$ MPa). Wall boundary conditions were set to the left and right ends of the pipe. The water is initially at rest and at the same temperature $T = 300K$ in both chambers. Unlike in Ref. [11], where the IAPWS-95 Spline Based Table Lookup approach [42] was used to compute the vapor properties, the liquid properties in the present test are computed from the SGEOS with parameters in Table 3. At $t = 0$ s, the diaphragm is removed and the flow develops. In the present test, the simulation was started at $t = 0$ s, and terminated at $t = 0.001$ s. In the rDG simulations, a fixed timestep size of $dt = 1 \times 10^{-5}$ s was used, and 100 timesteps were used to complete each simulation.

4.3.1 Results and Remarks

The computed solution profiles for density, specific internal energy, velocity, Mach number, pressure and temperature are plotted in Figs. 14 – 19, respectively, and compared with analytical solutions to this problem. Remarks are given below.

- rDG(P0P1) delivered the best solution among the methods tested in this problem. No oscillation is found at all in the rDG(P0P1) solution. rDG(P0P0) delivered a solution that is more diffusive than rDG(P0P1) in the vicinities of the shock waves. In particular, the contact discontinuity at $x = 5$ m was captured by rDG(P0P1) and rDG(P0P0) within only two elements.
- RELAP-7's FEM-EVM(1) solution rendered non-physical spurious oscillations in the vicinities of contact discontinuities, shock waves, and even in regions which should not be disturbed by shock waves yet. Again, alternative parameters in the EVM(1) method may reduce, but probably not completely eliminate those oscillations.
- RELAP-7's FEM-EVM(0) delivered an overly diffusive solution in this problem, especially near the contact discontinuity at $x = 5$ m.

Finally, the CPU time measured in the simulations is listed in the following table.

Method	CPU time (seconds)
RELAP-7 FEM-EVM(0)	**11.628**
RELAP-7 FEM-EVM(1)	**11.941**
rDG(P0P0)	**0.018**
rDG(P0P1)	**0.024**

The CPU time indicates that rDG methods with explicit time stepping give a markedly reduced computing time for solving transient gas dynamic problems with wave propagation containing strong gradients.

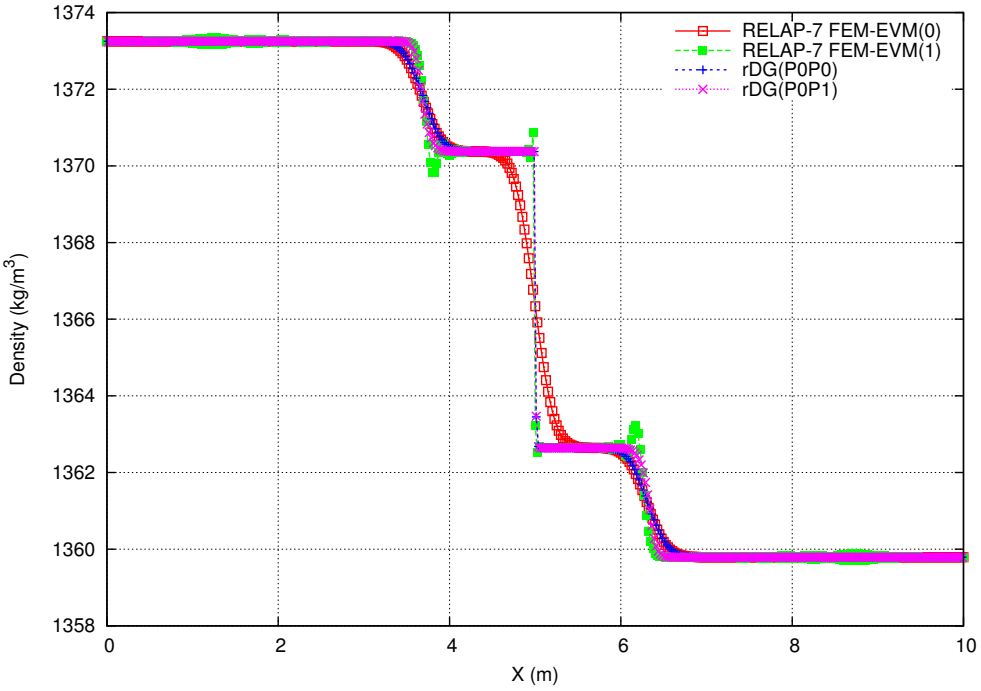

Figure 14. Computed density profiles at $t = 0.001$ for the liquid water shock tube.

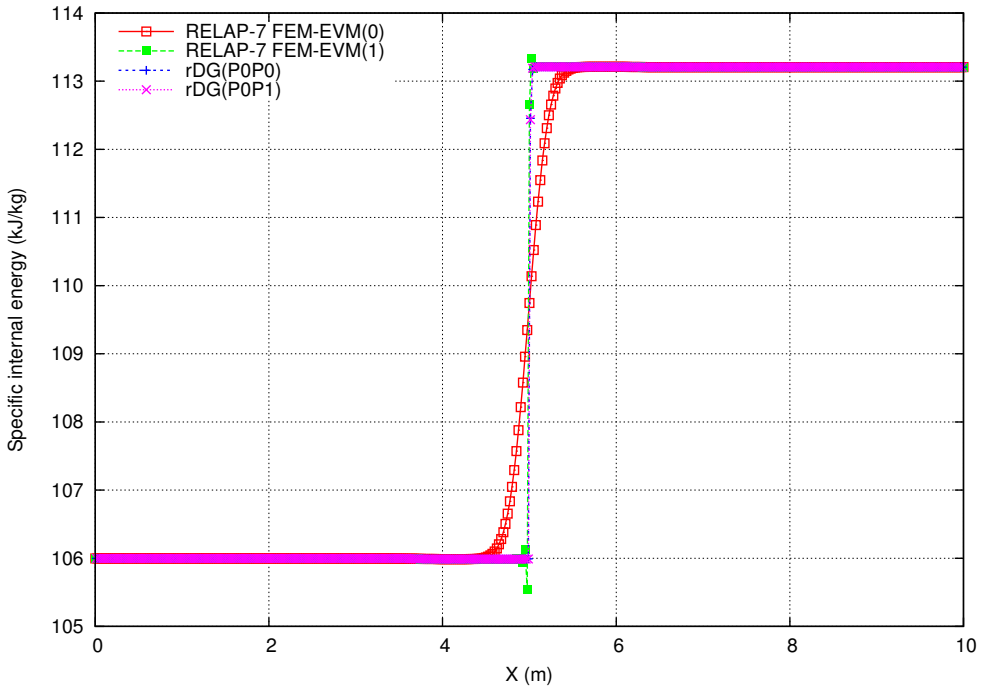

Figure 15. Computed specific internal energy profiles at $t = 0.001$ for the liquid water shock tube.

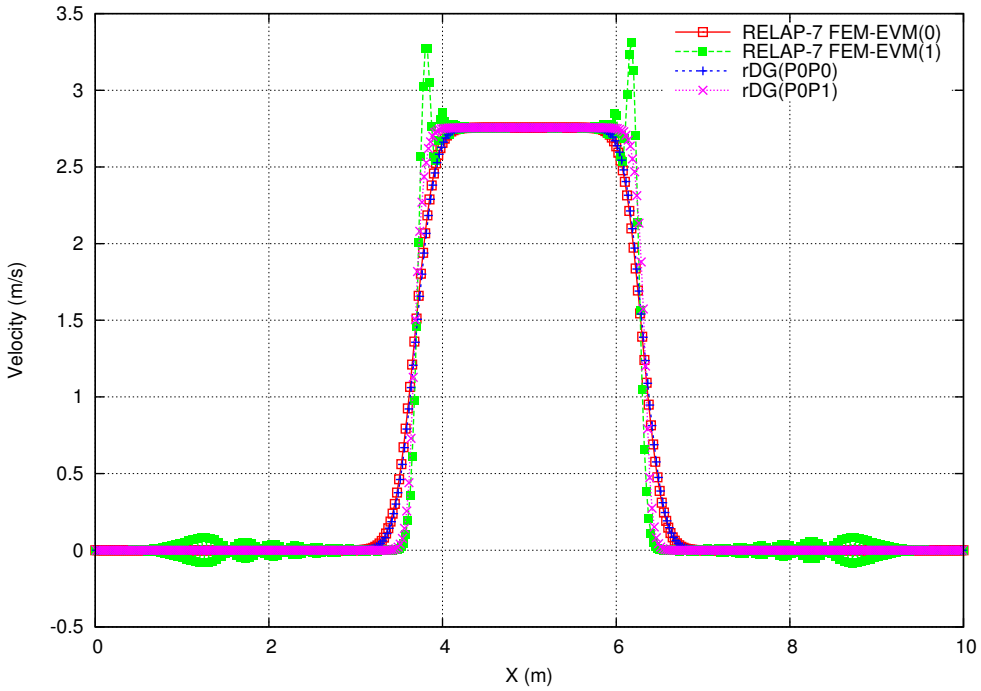

Figure 16. Computed velocity profiles at $t = 0.001$ for the liquid water shock tube.

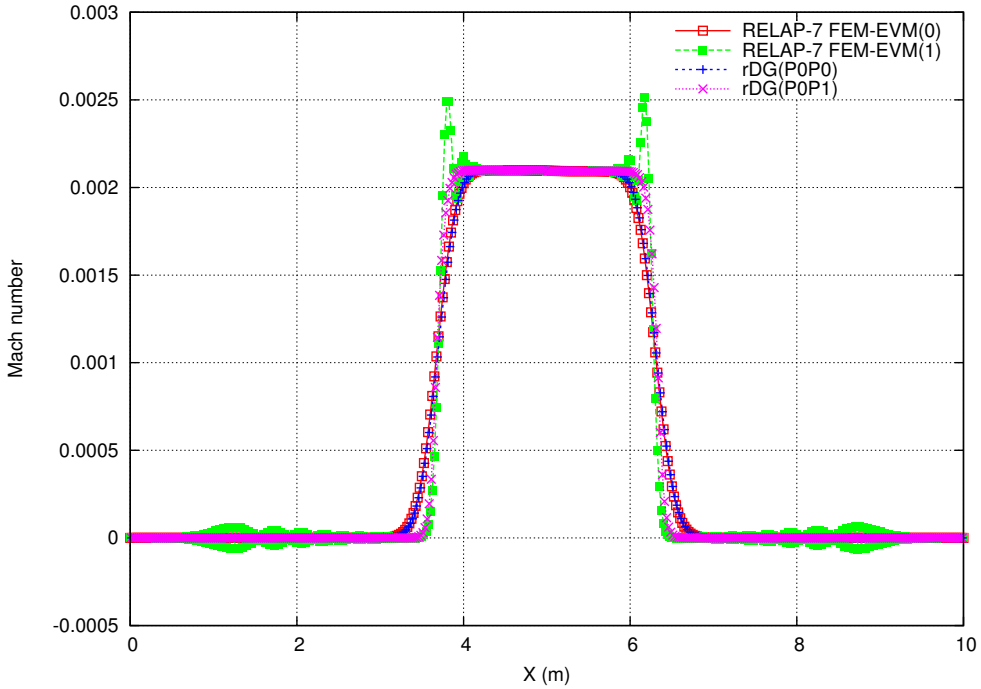

Figure 17. Computed Mach number profiles at $t = 0.001$ for the liquid water shock tube.

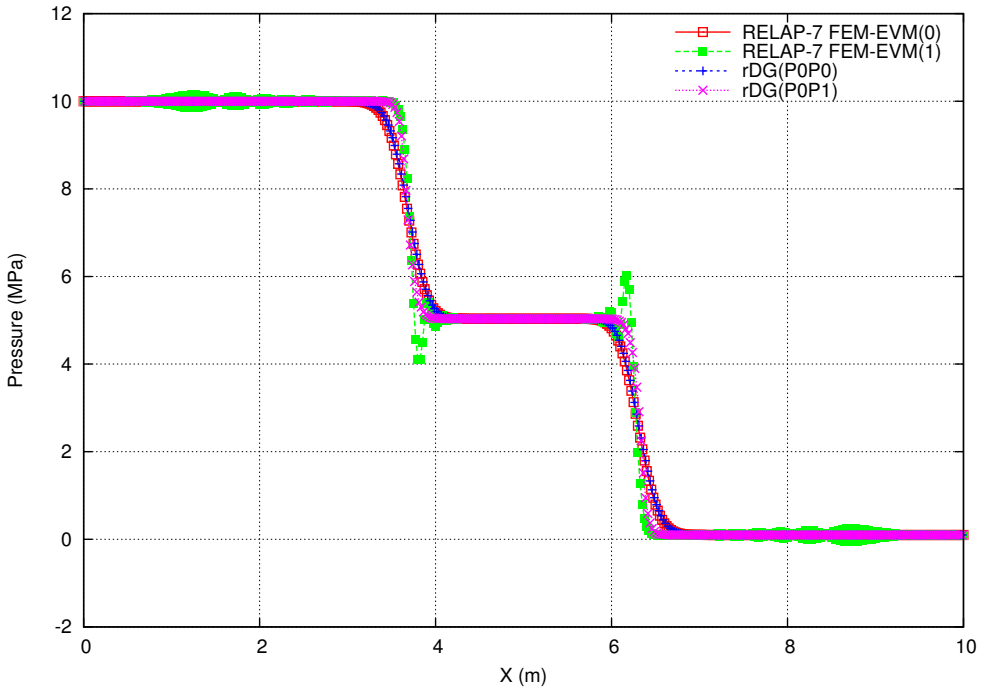

Figure 18. Computed pressure profiles at $t = 0.001$ for the liquid water shock tube.

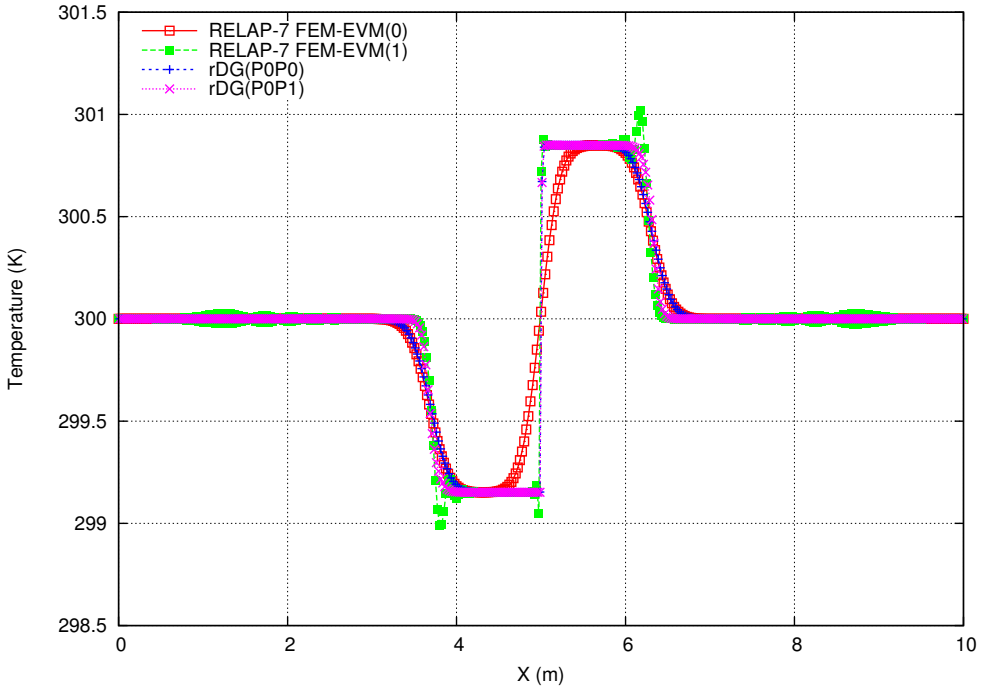

Figure 19. Computed temperature profiles at $t = 0.001$ for the liquid water shock tube.

4.4 Convergent–Divergent Nozzle

4.4.1 Problem Setup

From Ref. [1] we take a problem that describes a steady-state flow of vapor/liquid water in a convergent–divergent nozzle. This problem allows for some special-case, two-phase flow solutions that only the seven-equation model can be tested on, but which offer very good testing of several two-phase modeling fundamentals. The computational domain is bounded between $x = 0$ and 1 m, and the cross-section area of the nozzle can be mathematically described as $A(x) = 1 + \frac{1}{2}\cos(2\pi x)$ m. The stagnation inlet boundary condition was imposed at the left end with stagnation pressure $p_0 = 1$ MPa and stagnation temperature $T_0 = 453$ K, and the static pressure outlet boundary condition was imposed at the right end with static pressure $p_{out} = 0.5$ MPa. For the purpose of preliminary investigation, two-phase coupling through relaxation was turned off in this test. In this case, two simultaneous single-phase problems result with the phases sharing the same flow domain. The analytical solutions for each phase are known. The SGEOS parameters in Table 2 were used. A mesh consisting of 400 elements uniformly distributed in the domain was used. The initial conditions: $p = 0.5$ MPa, $T = 453$ K, and $u = 0$ in the whole domain, were used in the tests of vapor and liquid water, respectively. Each simulation was started at $t = 0$, and run with sufficient timesteps to reach steady state.

4.4.2 Results and Remarks

The computed solution profiles for vapor density, specific internal energy, velocity, Mach number, pressure and temperature are plotted in Figs. 20 – 25, respectively. Remarks are given below.

- The location of shock predicted by RELAP's FEM-EVM(1) does not appear correct. With reference to Ref. [1], it can be confirmed that the other three methods, rDG(P0P0), rDG(P0P1) and RELAP-7's FEM-EVM(0), delivered correct prediction of the shock location.
- Oscillations in the RELAP-7 FEM-EVM(1) solutions were observed near the right boundary.
- rDG(P0P1) delivered the most accurate and oscillation-free shock resolution, where **the shock was captured within only two elements**. rDG(P0P0) also delivered excellent shock resolution, where the shock was captured within two elements.
- RELAP-7's FEM-EVM(0) has been **significantly more diffusive than the first-order accurate rDG(P0P0)**.

In addition, the histories of L^2 norm of rDG(P0P0) solution variable residual in terms of timesteps and CPU time are plotted in Fig. 26 and 27, respectively. Those for rDG(P0P1) are plotted in Fig. 28 and 29, respectively. The L^2 norm of rDG solution variable residual is given as

$$\| U_i^n - U_i^{n-1} \|_2 = \sqrt{\sum_{i=1}^{Nelem} (U_i^n - U_i^{n-1})^2}$$

where U is a solution variable (e.g. ρ, ρu, or ρE), $Nelem$ the number of elements in the domain, and n the current timestep. Remarks are given below.

- The rDG(P0P0) vapor density residual dropped toward **machine zero** with sufficient timesteps; see Fig. 26. Indeed, a first-order spatial scheme is expected to achieve such convergence in steady-state problems. Moreover, it took only **5 seconds in CPU time** for rDG(P0P0) density residual to drop **14 orders of magnitude** in explicit two-stage TVDRK stepping; see Fig. 27.
- rDG(P0P1) with a TVD slope limiter (e.g. minmod) usually cannot make solutions converge to machine zero. Nevertheless, steady-state solutions can still be reached at a higher level of residual norm. Moreover, rDG(P0P1) ran **fast**, as only **3 seconds in CPU time** were needed in this test; see Fig. 29.

The computed solution profiles for liquid density, specific internal energy, velocity, Mach number, pressure and temperature are plotted in Figs. 30 – 35, respectively. Remarks are given below.

- In this smooth problem, RELAP-7's FEM-EVM(0) and rDG(P0P0) appeared almost **unusable**, as the results by these two schemes were severely diffused. Only significant amount of mesh refinement could improve their resolution in this problem, which would become drastically more expensive in terms of computing time, and perhaps practically unaffordable in system-scale simulations.
- The results of rDG(P0P1) and RELAP-7's FEM-EVM(1) agreed well with each other.

The histories of L^2 norm of rDG(P0P0) liquid solution variable residual in terms of timesteps and CPU time are plotted in Fig. 36 and 37, respectively. Those for rDG(P0P1) are plotted in Fig. 38 and 39, respectively. Remarks are given below.

- The rDG(P0P0) liquid density residual dropped toward **machine zero** with sufficient timesteps; see Fig. 36. It took **200 seconds in CPU time** for rDG(P0P0) liquid density residual to drop **15 orders of magnitude** in explicit two-stage TVDRK stepping; see Fig. 27.
- Using fully compressible formulations, rDG(P0P1) with explicit time stepping is known to be disadvantaged for low-Mach water flow problems. The effective dissipation is wrong and gives inaccurate results. Another issue is convergence stall, in which much more timesteps are actually required for rDG(P0P1) to reach converged solution even after the residual curves appear to flat out; see Fig. 38.
- An implicit time integration scheme should help rDG(P0P1) reach steady state significantly faster than its explicit counterpart in terms of CPU time; see Fig. 39. Low-Mach Riemann solvers are also available now to ameliorate this effect.

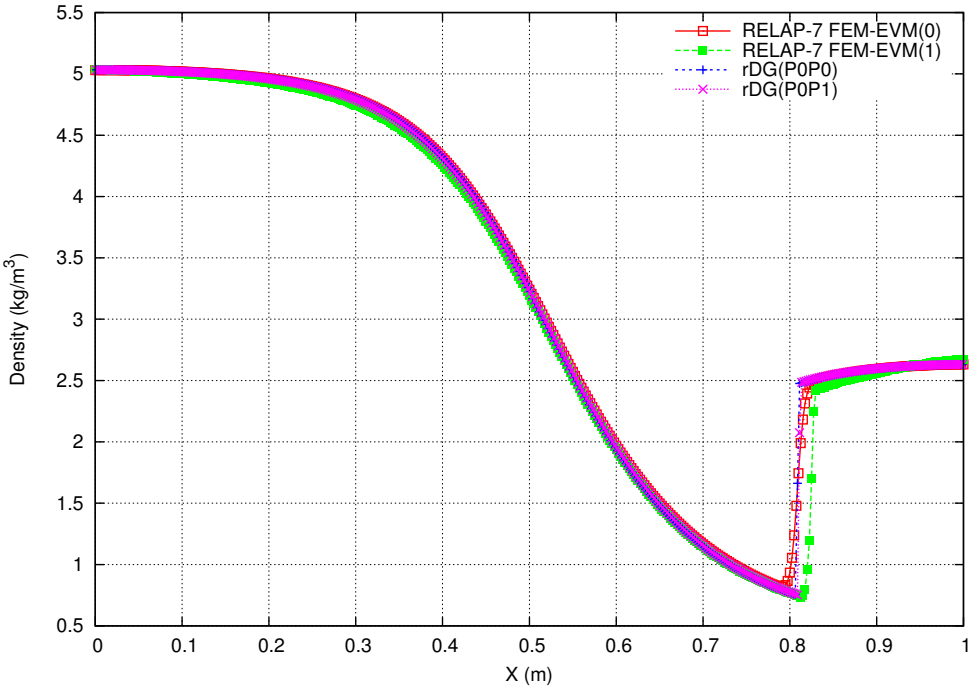

Figure 20. Steady-state density profiles for vapor in a convergent–divergent nozzle.

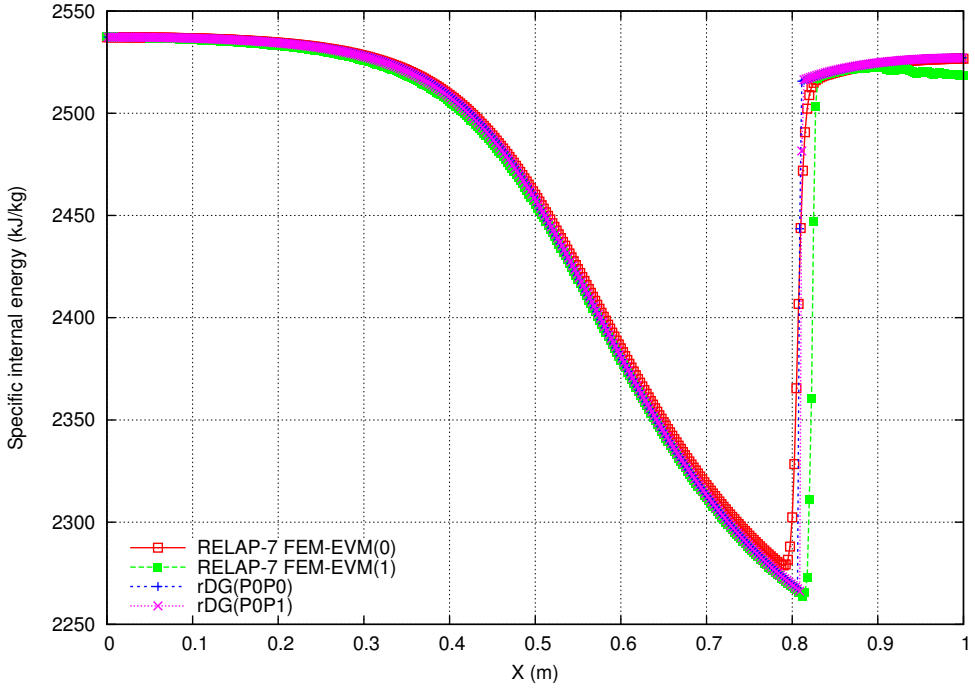

Figure 21. Steady-state specific internal energy profiles for vapor in a convergent–divergent nozzle.

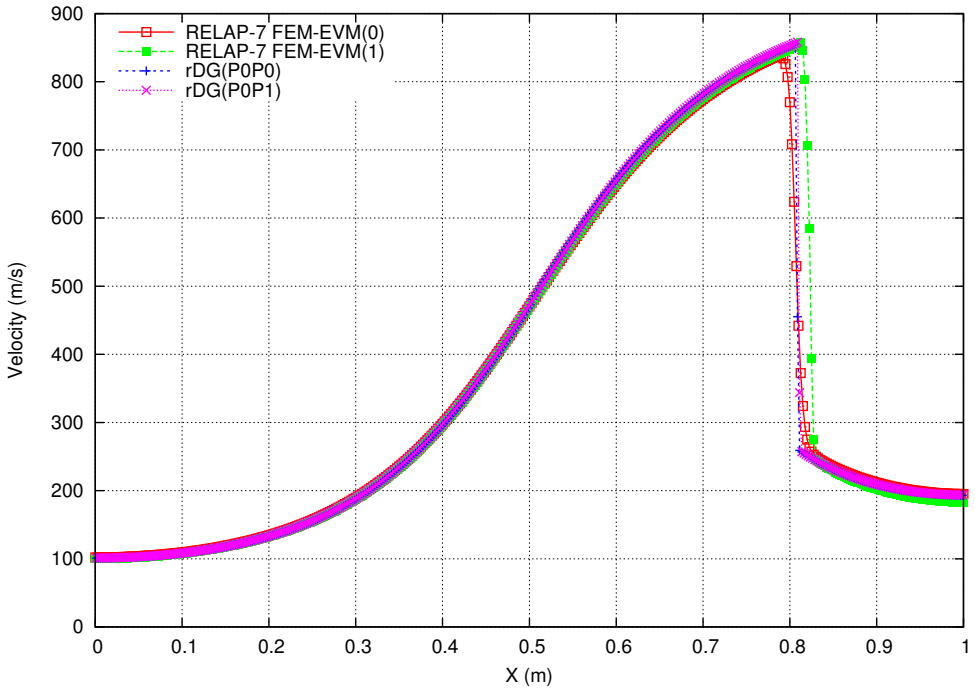

Figure 22. Steady-state velocity profiles for vapor in a convergent–divergent nozzle.

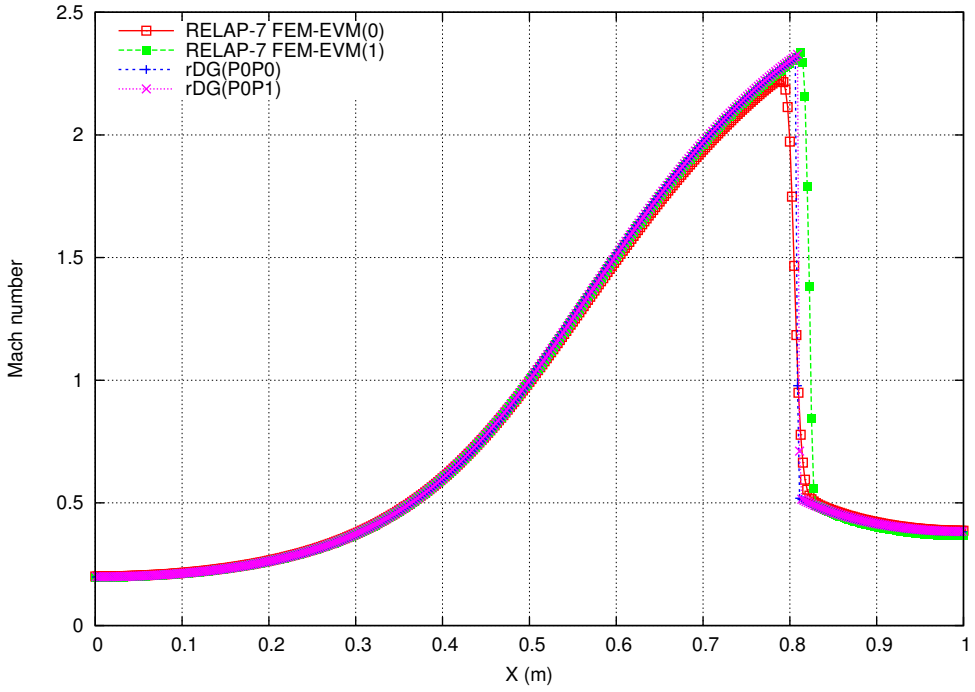

Figure 23. Steady-state Mach number profiles for vapor in a convergent–divergent nozzle.

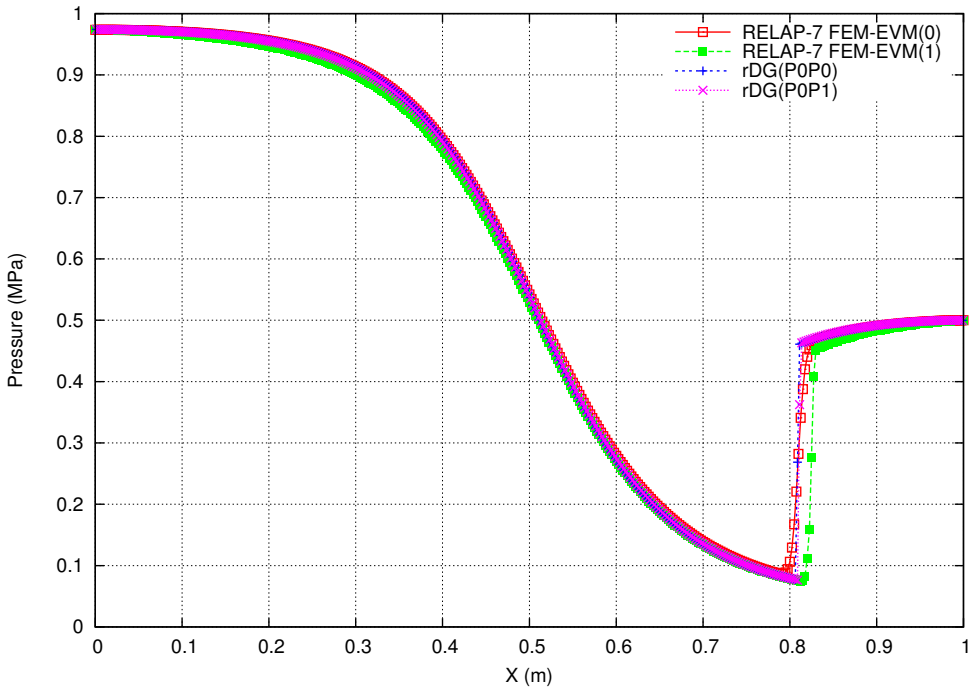

Figure 24. Steady-state pressure profiles for vapor in a convergent–divergent nozzle.

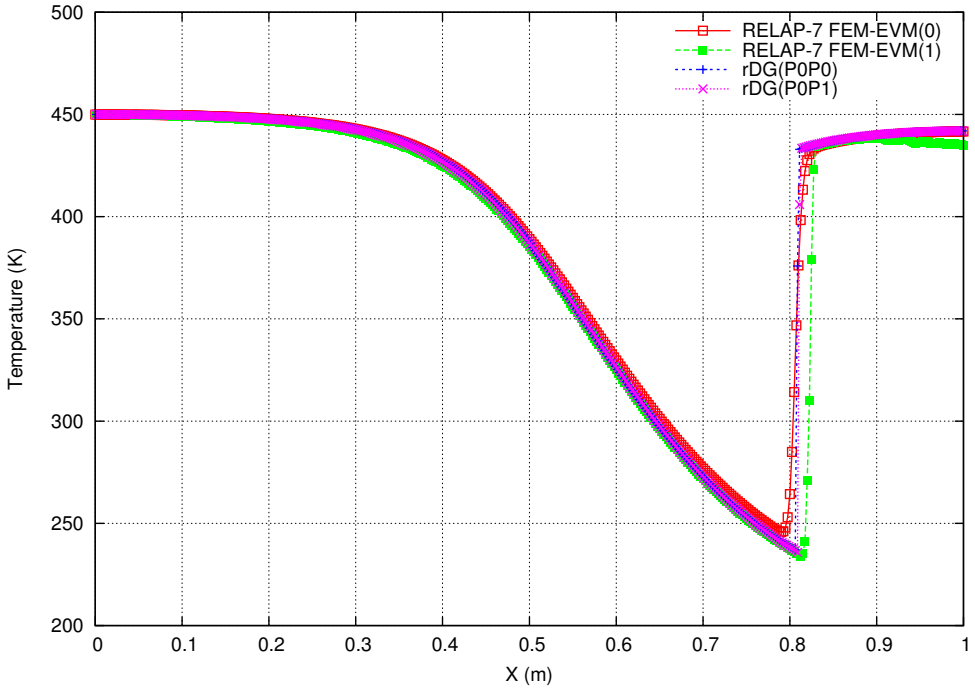

Figure 25. Steady-state temperature profiles for vapor in a convergent–divergent nozzle.

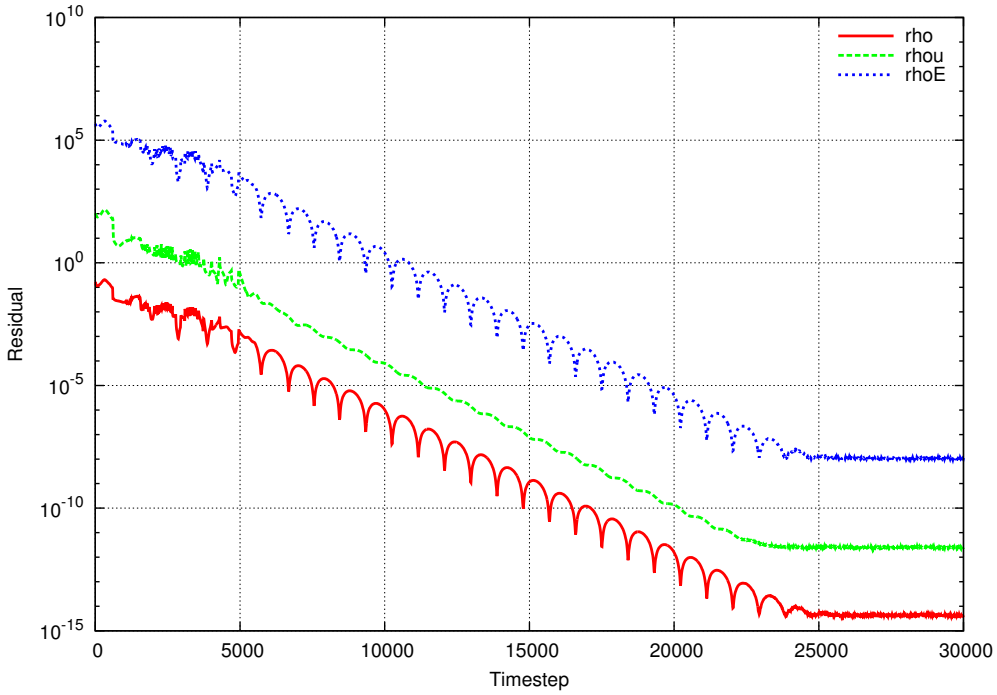

Figure 26. rDG(P0P0) solution residual versus timesteps for vapor in a convergent–divergent nozzle.

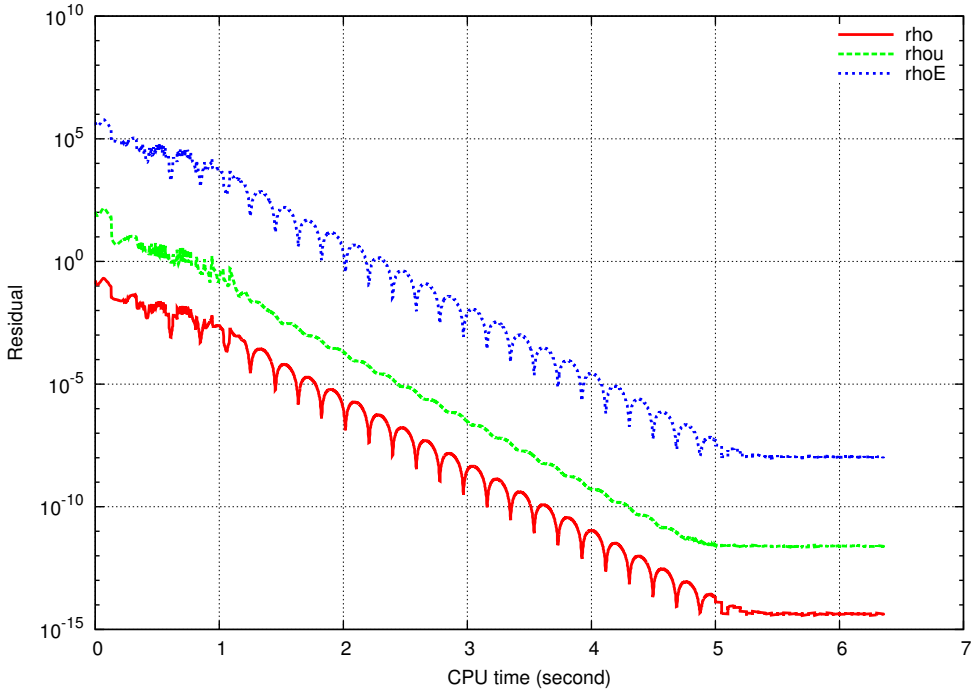

Figure 27. rDG(P0P0) solution residual versus CPU time for vapor in a convergent–divergent nozzle.

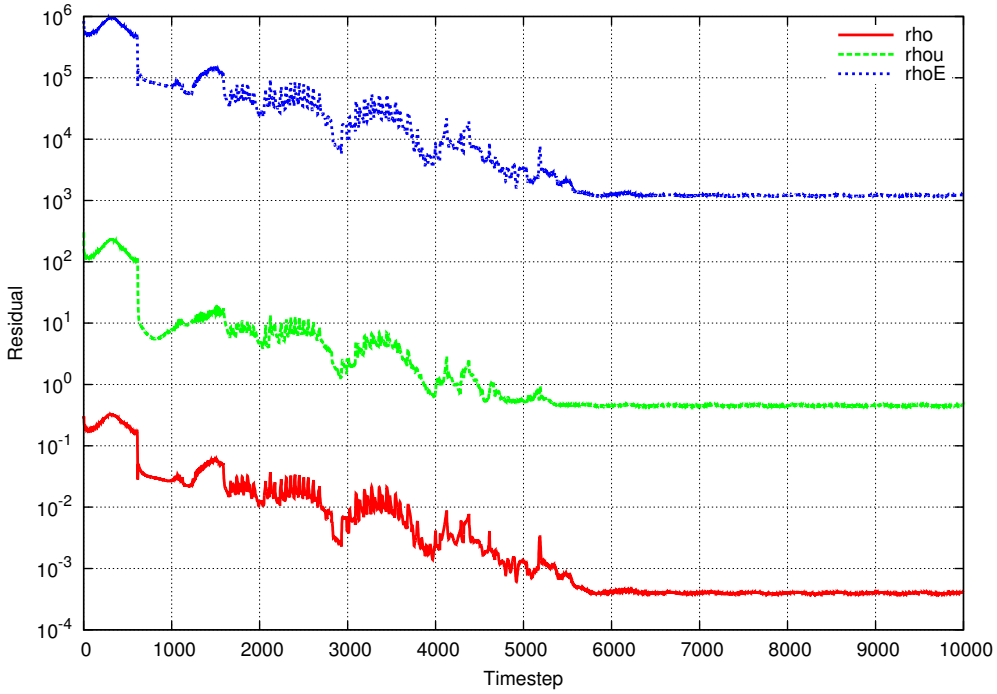

Figure 28. rDG(P0P1) solution residual versus timesteps for vapor in a convergent–divergent nozzle.

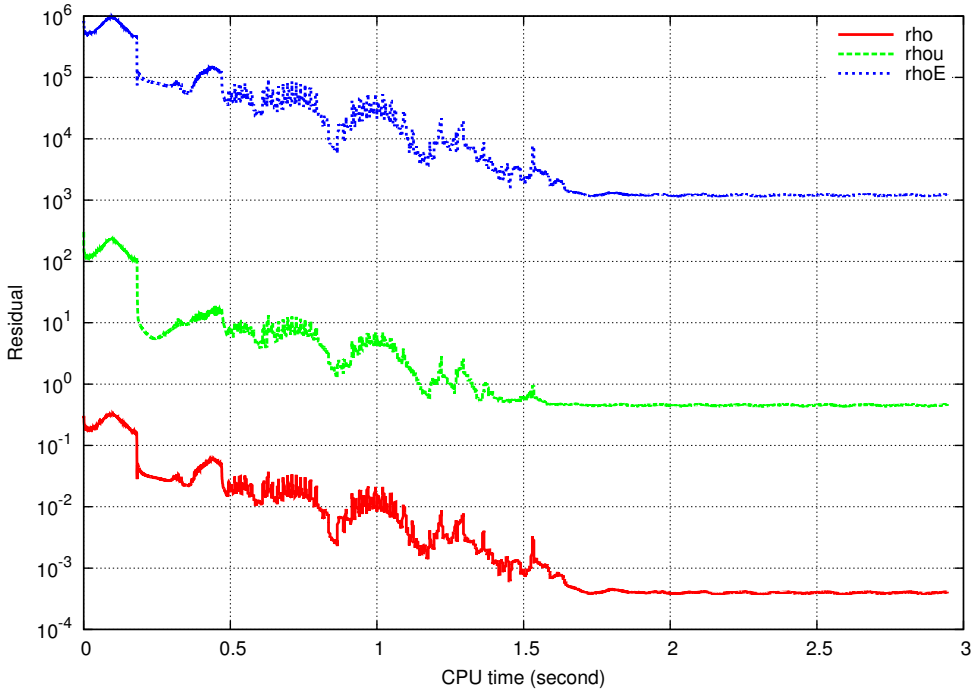

Figure 29. rDG(P0P1) solution residual versus CPU time for vapor in a convergent–divergent nozzle.

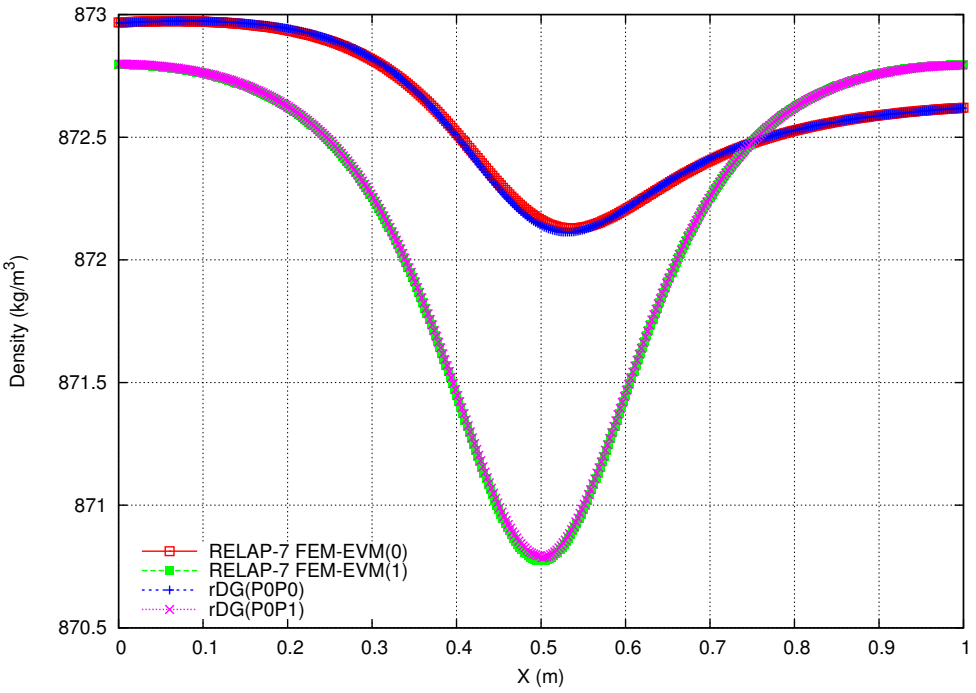

Figure 30. Steady-state density profiles for liquid water in a convergent–divergent nozzle.

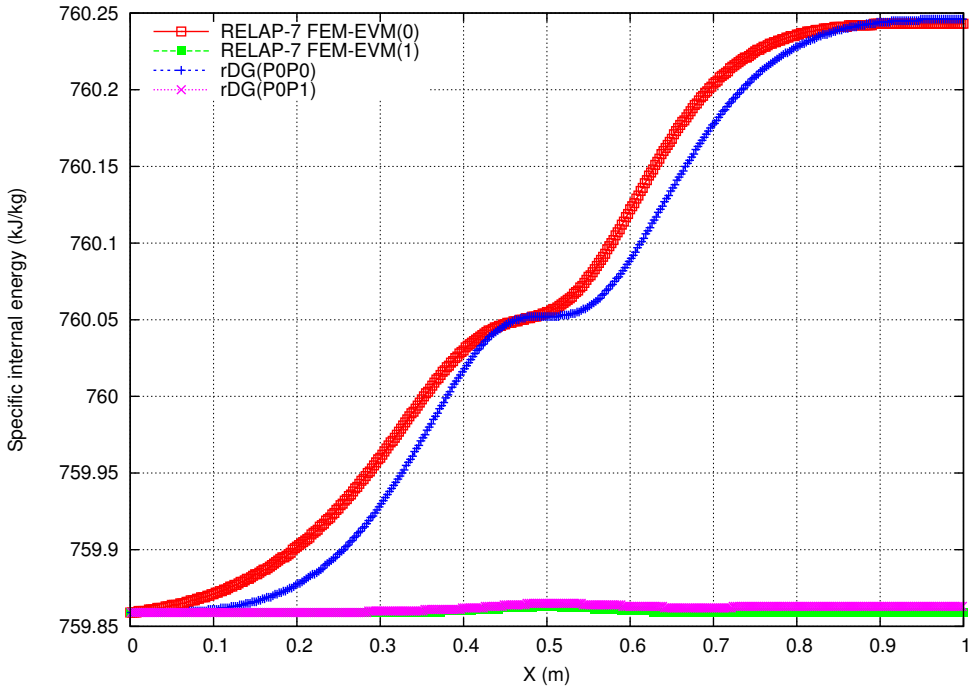

Figure 31. Steady-state specific internal energy profiles for liquid water in a convergent–divergent nozzle.

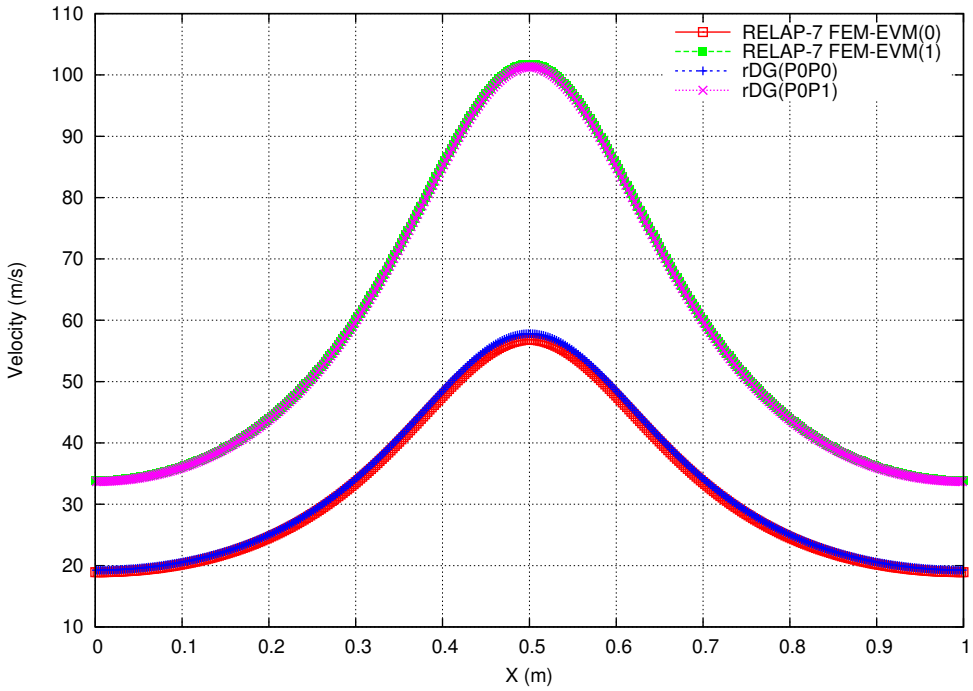

Figure 32. Steady-state velocity profiles for liquid water in a convergent–divergent nozzle.

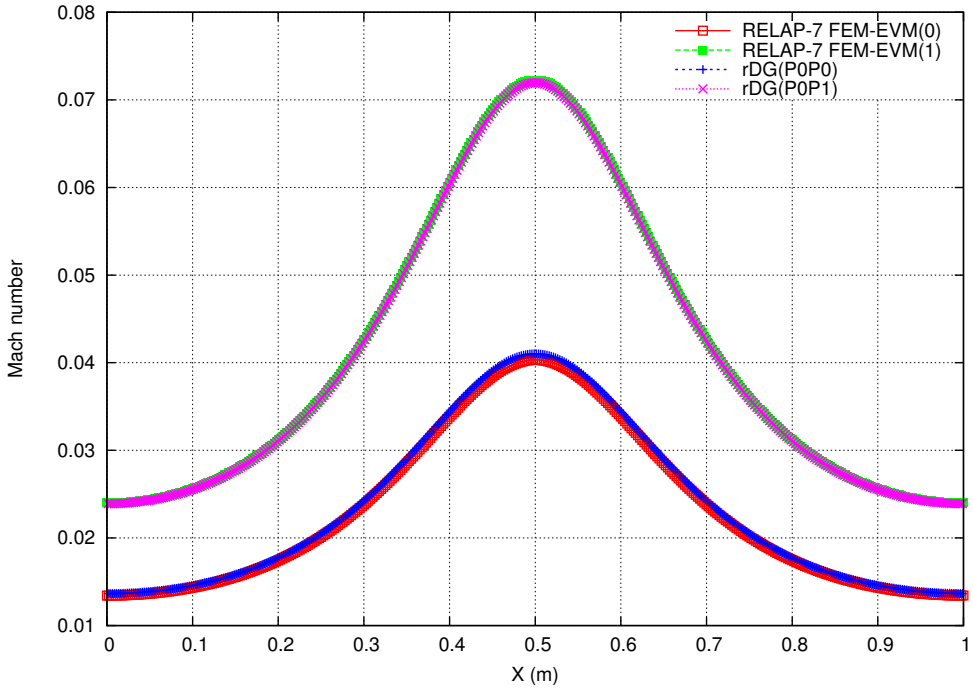

Figure 33. Steady-state Mach number profiles for liquid water in a convergent–divergent nozzle.

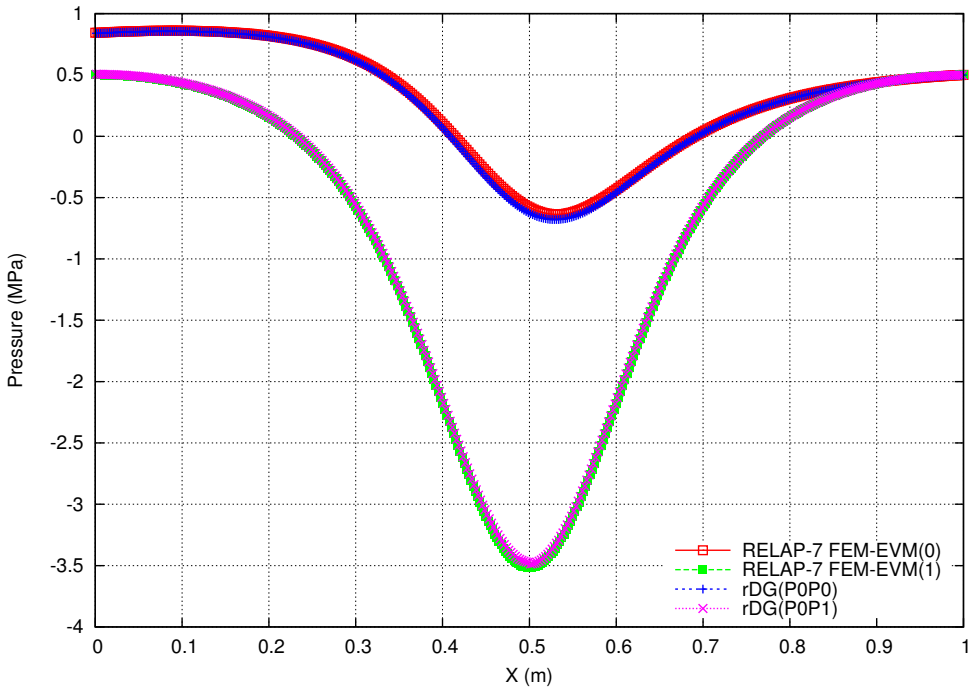

Figure 34. Steady-state pressure profiles for liquid water in a convergent–divergent nozzle.

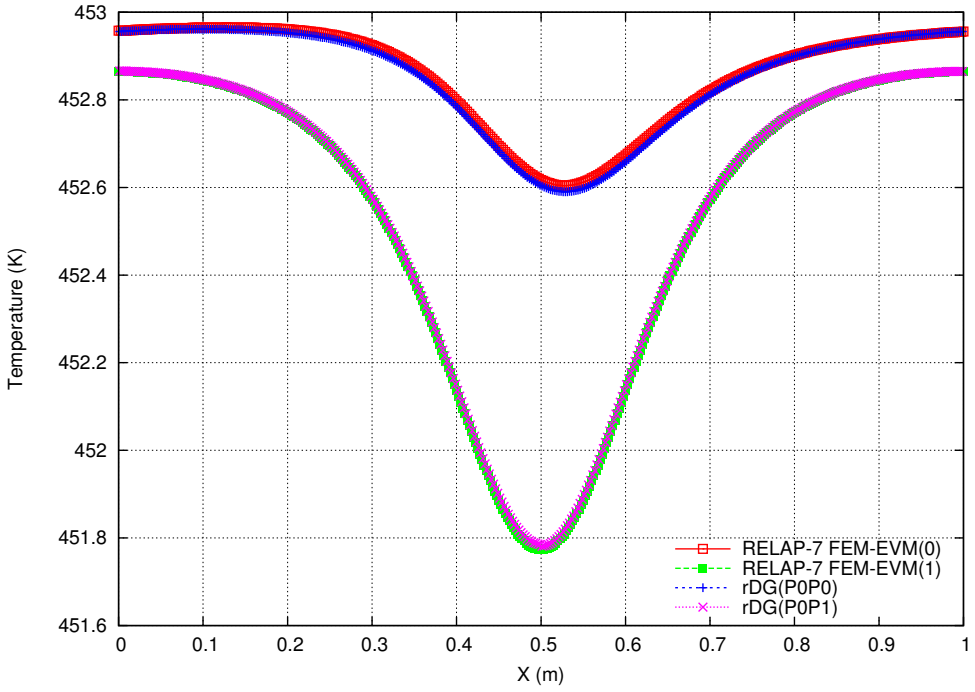

Figure 35. Steady-state temperature profiles for liquid water in a convergent–divergent nozzle.

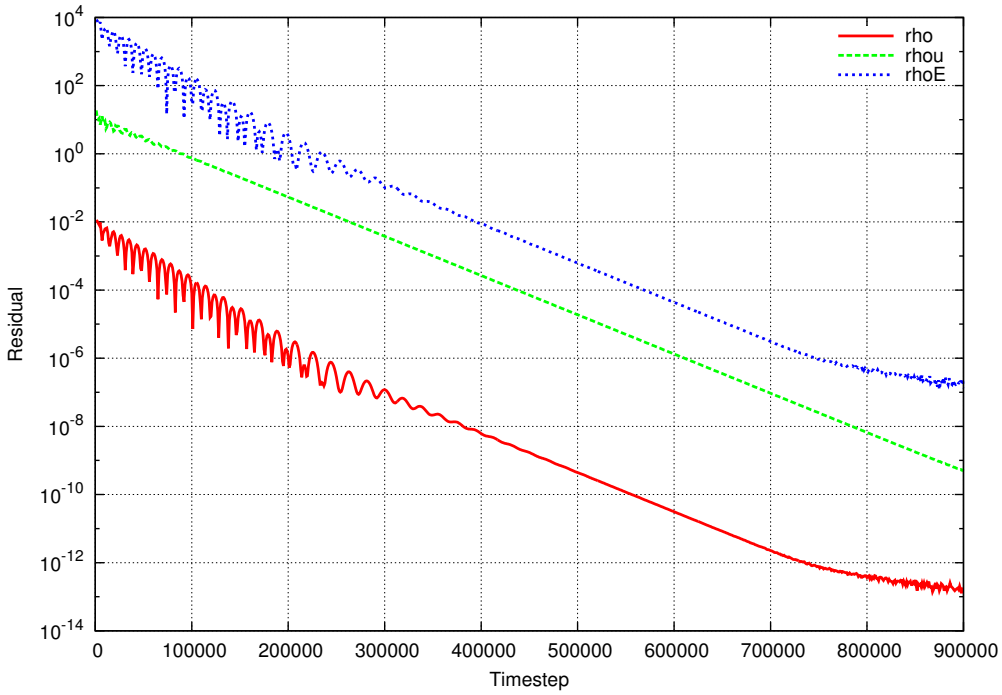

Figure 36. rDG(P0P0) solution residual versus timesteps for water in a convergent–divergent nozzle.

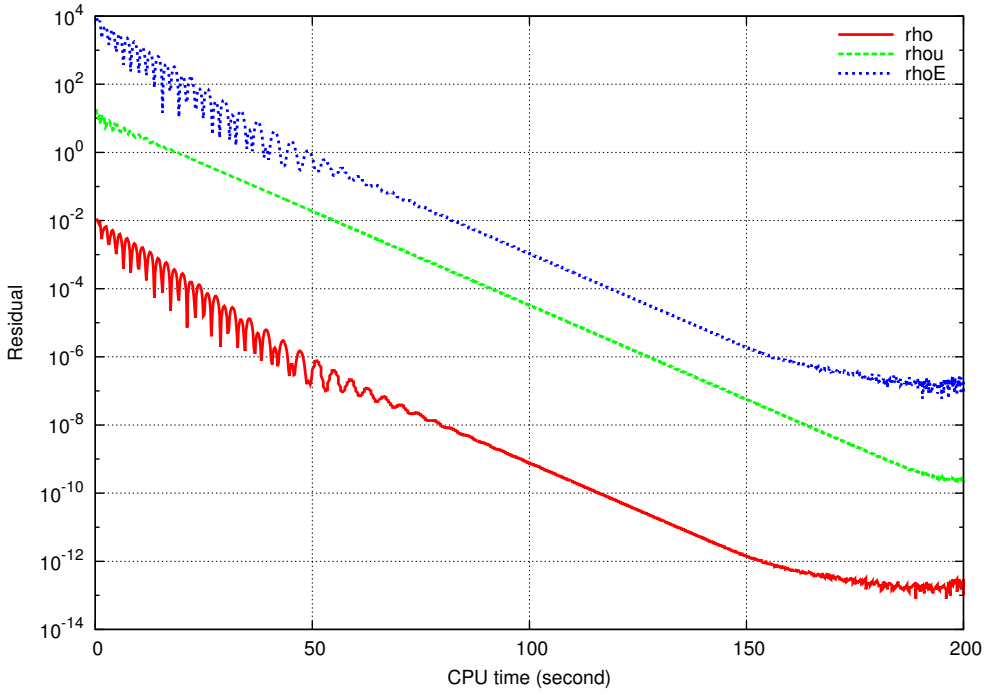

Figure 37. rDG(P0P0) solution residual versus CPU time for water in a convergent–divergent nozzle.

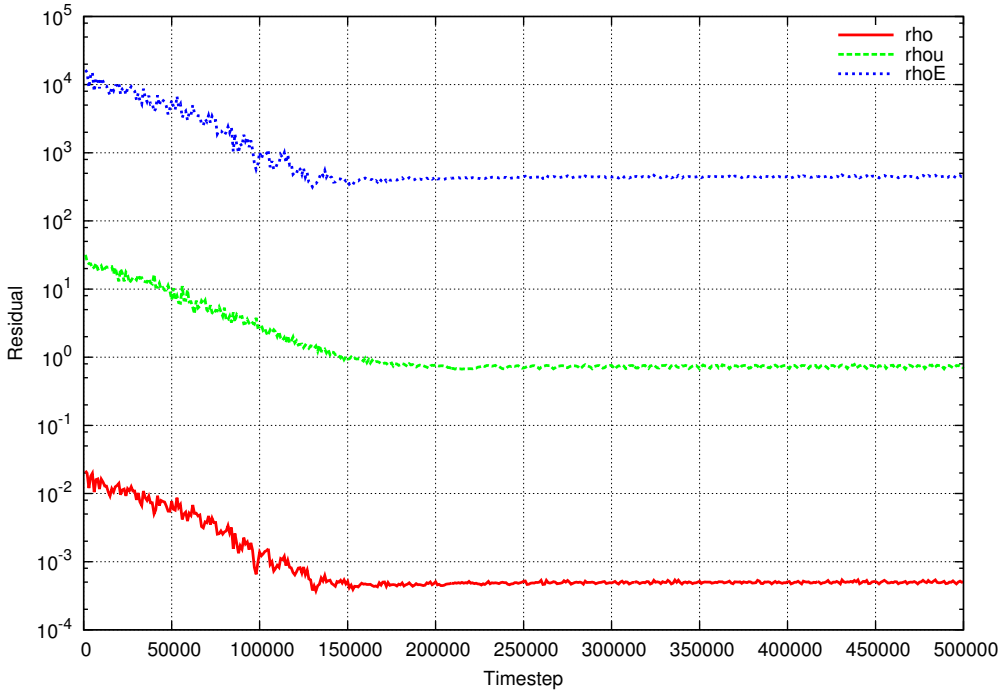

Figure 38. rDG(P0P1) solution residual versus timesteps for water in a convergent–divergent nozzle.

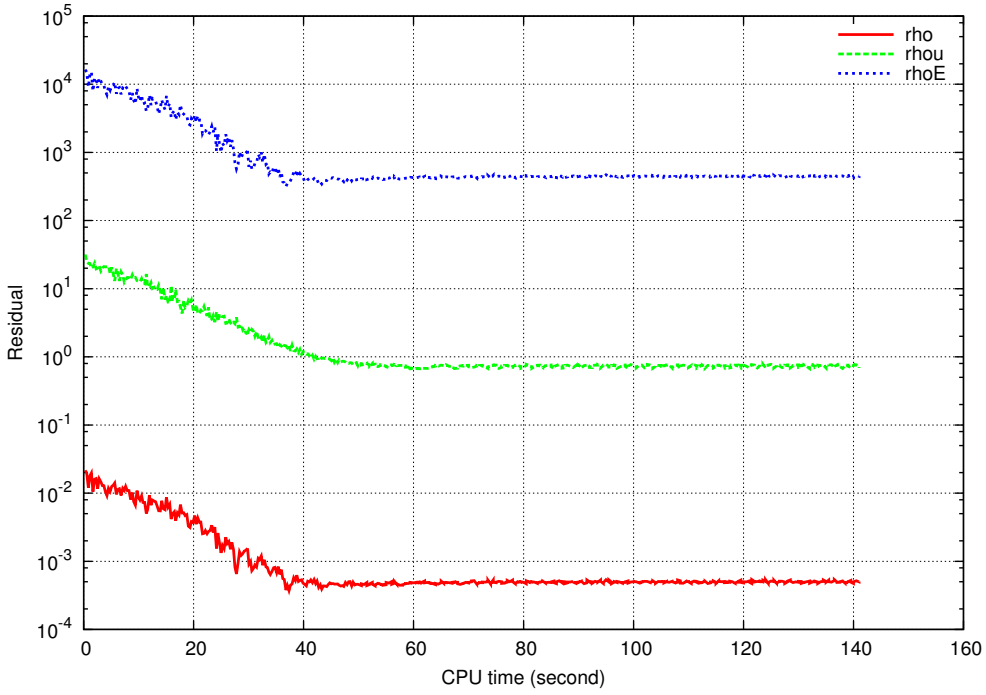

Figure 39. rDG(P0P1) solution residual versus CPU time for water in a convergent–divergent nozzle.

4.5 Thermodynamic Equilibrium of Co-Existing Vapor and Liquid Water

4.5.1 Problem Setup

This problem describes thermodynamic equilibrium of vapor and liquid co-existing in a pipe with constant cross-section area of $A = 1$. The initial conditions are: (α_g, p_g, u_g, T_g) = (α_l, p_l, u_l, T_l) = (0.5, 0.5 MPa, 0 m/s, 453 K) in the whole domain bounded between $x = 0$ and 1. The SGEOS parameters in Table 3 were used. The solid wall boundary conditions were imposed at the left and right end. In Eqs. (1) - (7), the terms that involve the phasic velocities (i.e. u_g and u_l), interphase velocity (i.e. u_{int} and $\bar{u}_{int}$), and variable spatial gradients (i.e. $\partial\alpha_g/\partial x$, $\partial\alpha_l/\partial x$, $\partial p_g/\partial x$, $\partial p_l/\partial x$, and $\partial A/\partial x$) should always remain zero in this problem, and thus can be eliminated, resulting in the following *zero-th order system of partial differential equations* (a system of ordinary differential equations):

$$\frac{\partial \alpha_g A}{\partial t} = A\mu(p_g - p_l) + \frac{\Gamma_{int,g} A_{int} A}{\rho_{int}} \tag{31}$$

$$\frac{\partial \alpha_g \rho_g A}{\partial t} = \Gamma_{int,g} A_{int} A \tag{32}$$

$$\frac{\partial \alpha_g \rho_g u_g A}{\partial t} = 0 \tag{33}$$

$$\frac{\partial \alpha_g \rho_g E_g A}{\partial t} = -\bar{p}_{int} A\mu(p_g - p_l) - \Gamma_{int,g} A_{int}\left(\frac{p_{int}}{\rho_{int}} - H_{g,int}\right)A + A_{int} h_{conv,g}(T_{int} - T_g)A \tag{34}$$

$$\frac{\partial \alpha_l \rho_l A}{\partial t} = -\Gamma_{int,g} A_{int} A \tag{35}$$

$$\frac{\partial \alpha_l \rho_l u_l A}{\partial t} = 0 \tag{36}$$

$$\frac{\partial \alpha_l \rho_l E_l A}{\partial t} = -\bar{p}_{int} A\mu(p_l - p_g) + \Gamma_{int,g} A_{int}\left(\frac{p_{int}}{\rho_{int}} - H_{l,int}\right)A + A_{int} h_{conv,l}(T_{int} - T_l)A \tag{37}$$

In theory it requires only 1 elment to solve Eqs. (31) - (37), as the solution should remain constant regardless of the location. However, we used 400 elements to solve Eqs. (1) - (7), in order to check if the computed solution would vary between the elements.

4.5.2 Results and Remarks

In this test problem, the variable residuals were driven below 10^{-14} in the rDG(P0P0) and rDG(P0P1) simulations, while in RELAP-7 EVM(0), EVM(1) and FEM(1) simulations, the lowest possible level of variable residuals was around 10^{-9}. The computed solution profiles for phasic volume fraction, density, specific internal energy, velocity, pressure and temperature are plotted in Figs. 40 – 51, respectively. Remarks are given below.

- Overall, the solutions computed by the rDG and RELAP-7 agreed well. Maximum relative error of the solutions between the two classes of methods was around 0.01% and 0.001%. In that regard, we can conclude that the interface mass and energy transfer terms implemented in the rDG code are able to produce the same solution as those of RELAP-7.
- The velocities computed by rDG(P0P0) and rDG(P0P1) were uniformly zero, as shown in Figs. 46 and

47. The reconstructed gradients of the variables in rDG(P0P1) were also zero. Thus rDG(P0P1) was numerically identical to rDG(P0P0) in this test, and they resulted in numerically the same solutions.

- The velocities computed by RELAP-7's FEM-EVM(0), FEM-EVM(1) and FEM were not uniformly zero, as shown in Figs. 46 and 47. Their maximum magnitudes (10^{-11}) were however small enough to not be a concern. It is usually not possible to obtain absolute zero velocities using an implicit solver.

- As shown in Fig. 45, the liquid-phase specific internal energy profile computed by RELAP-7's FEM-EVM(0) does not visually match those by RELAP-7's FEM-EVM(1) and FEM. Disagreement between the RELAP-7's EVM(0), EVM(1), and FEM(1) solution profiles is also visually observed in the vapor-phase density in Fig. 42, vapor- and liquid-phase pressures in Figs. 48 and 43, respectively. The contributions of the dissipation terms in RELAP-7 involve combinations of terms proportional to gradients of a number of variables. Whenever gradients exist for those variables, the dissipation terms are possibly non-zero. These effects can be amplified for large dissipation coefficients, such as those sometimes encountered for the low-order dissipation scheme, possibly producing a slightly different solution than for the no-dissipation case.

- As shown in Fig. 48, the vapor pressures obtained by RELAP-7's FEM-EVM(0), FEM-EVM(1) and FEM do not closely match each other. Such disagreement is also observed in the liquid pressures obtained by RELAP-7's FEM-EVM(0), FEM-EVM(1) and FEM, as shown in Fig. 49.

In conclusion, this test problem, though it looks trivial, is important to verify the implementation and convergence with regard to the interface mass and energy transfer terms. An implicit solver is necessary for the best solution speed in zero-speed problems like this. However on the other side, the implicit solver in RELAP-7 may need to be reviewed to overcome a few known defficiencies for solving such problems.

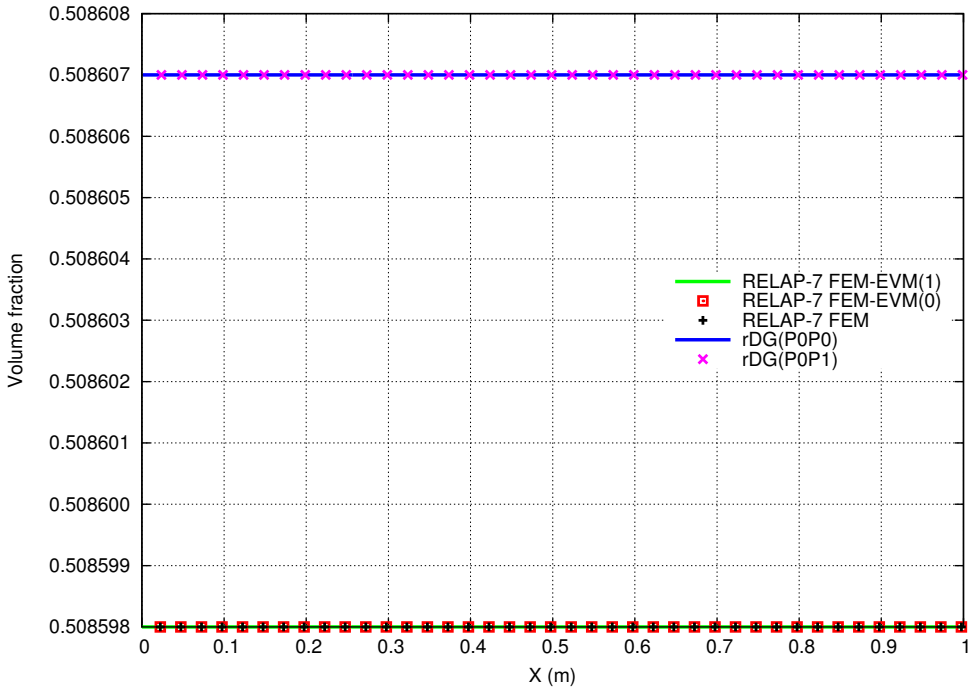

Figure 40. Volume fraction profiles for vapor at equilibrium.

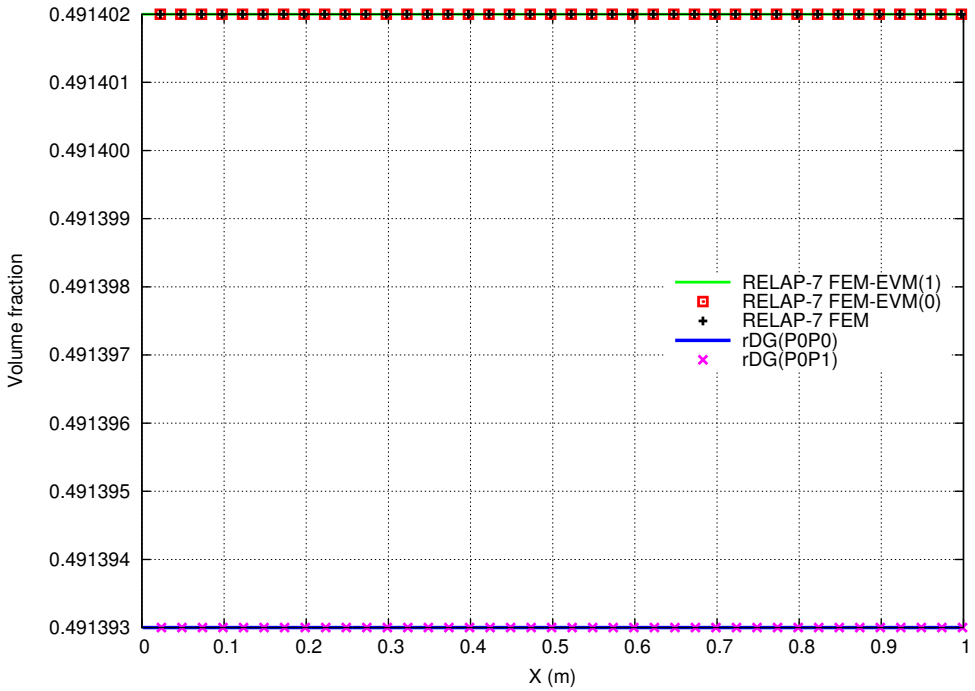

Figure 41. Volume fraction profiles for liquid at equilibrium.

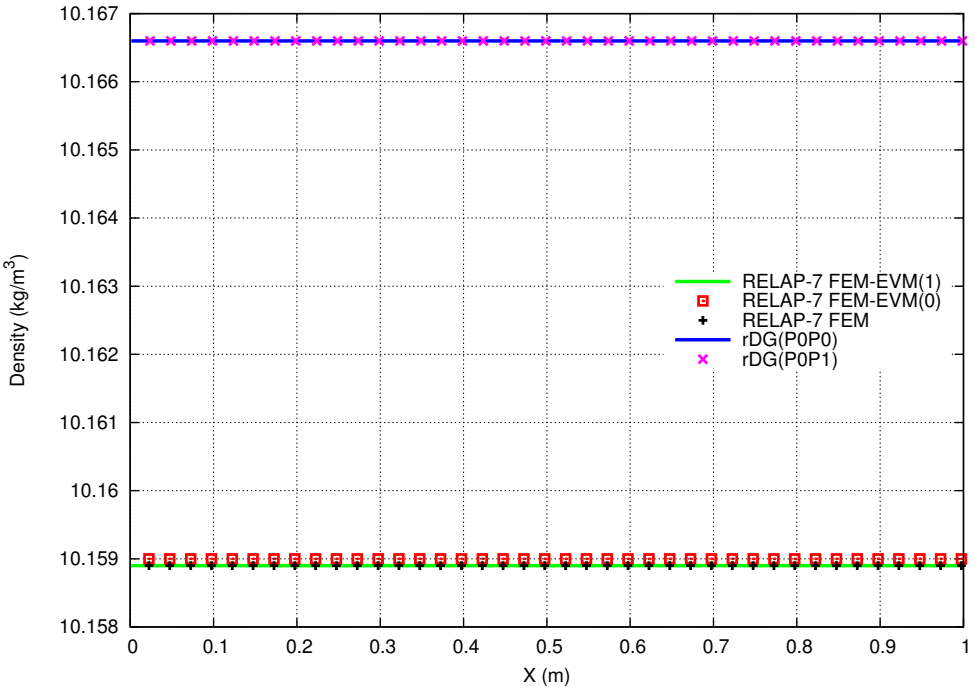

Figure 42. Density profiles for vapor at equilibrium.

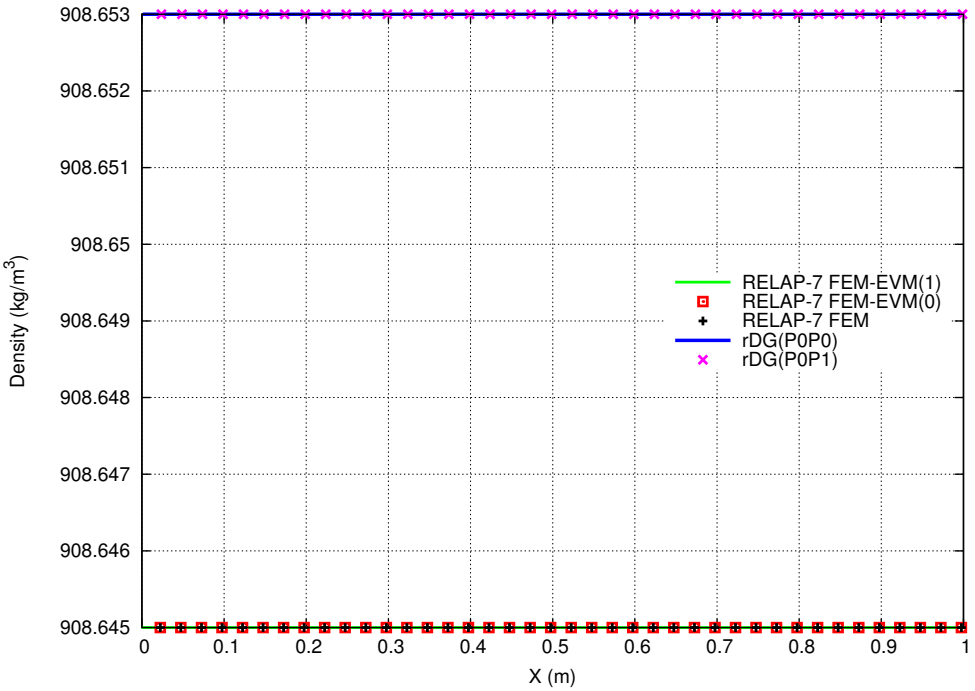

Figure 43. Density profiles for liquid at equilibrium.

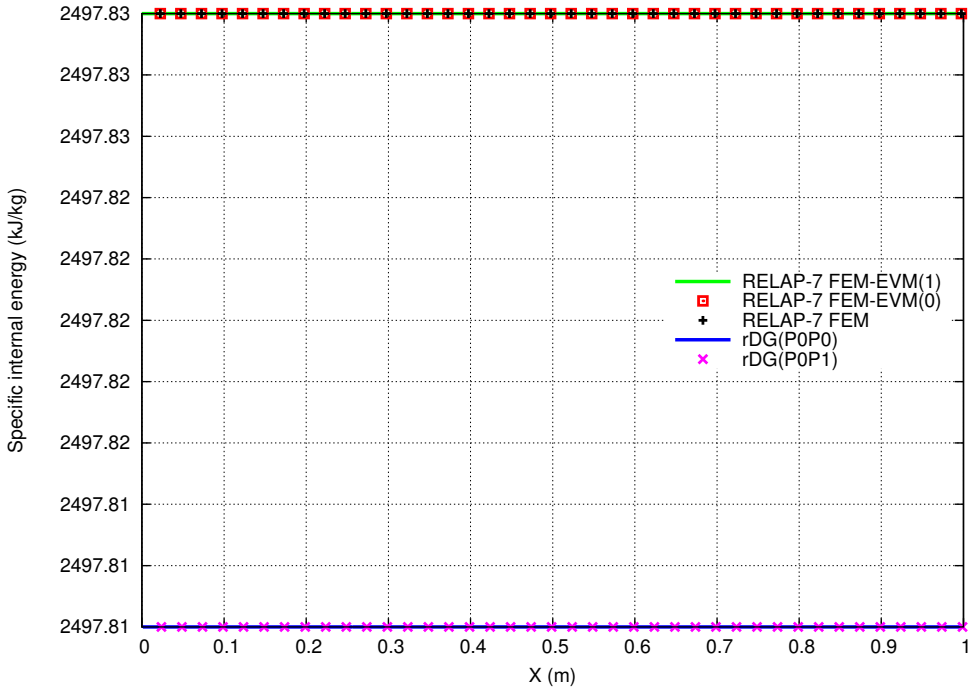

Figure 44. Specific internal energy profiles for vapor at equilibrium.

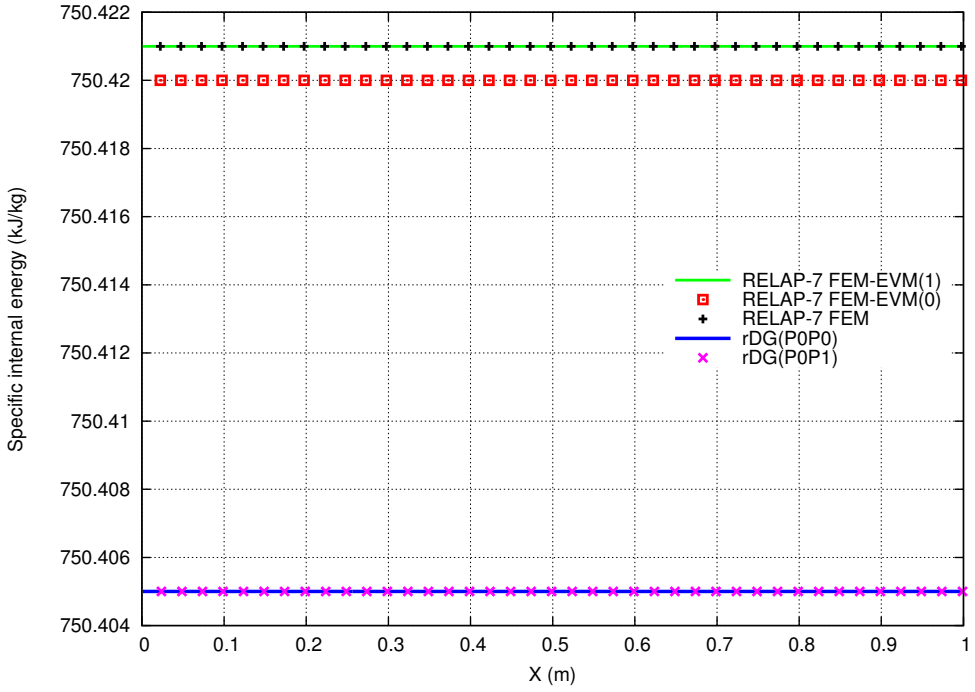

Figure 45. Specific internal energy profiles for liquid at equilibrium.

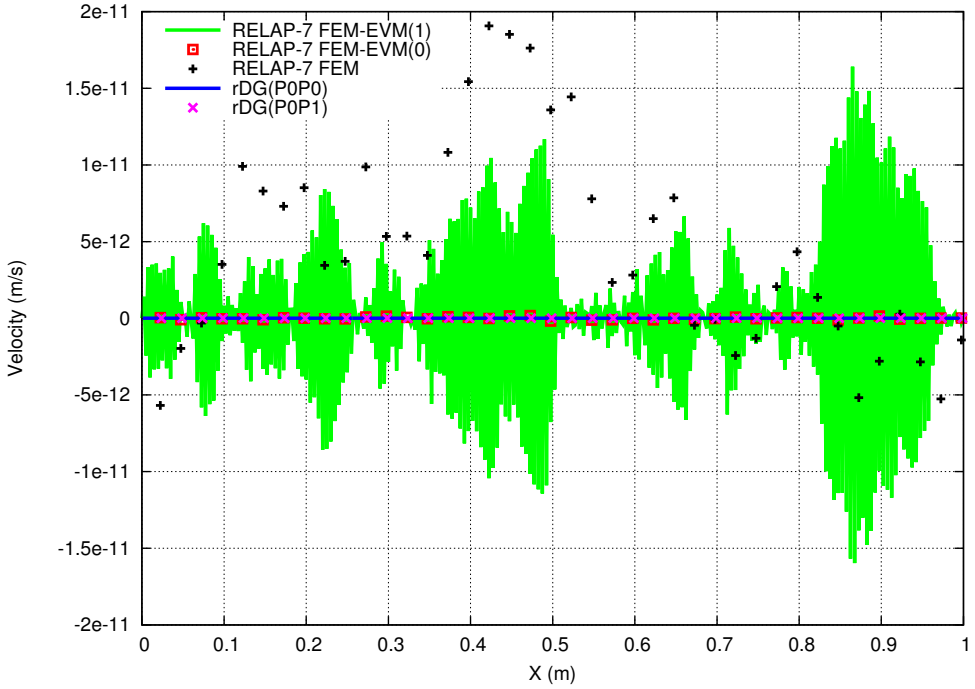

Figure 46. Velocity profiles for vapor at equilibrium.

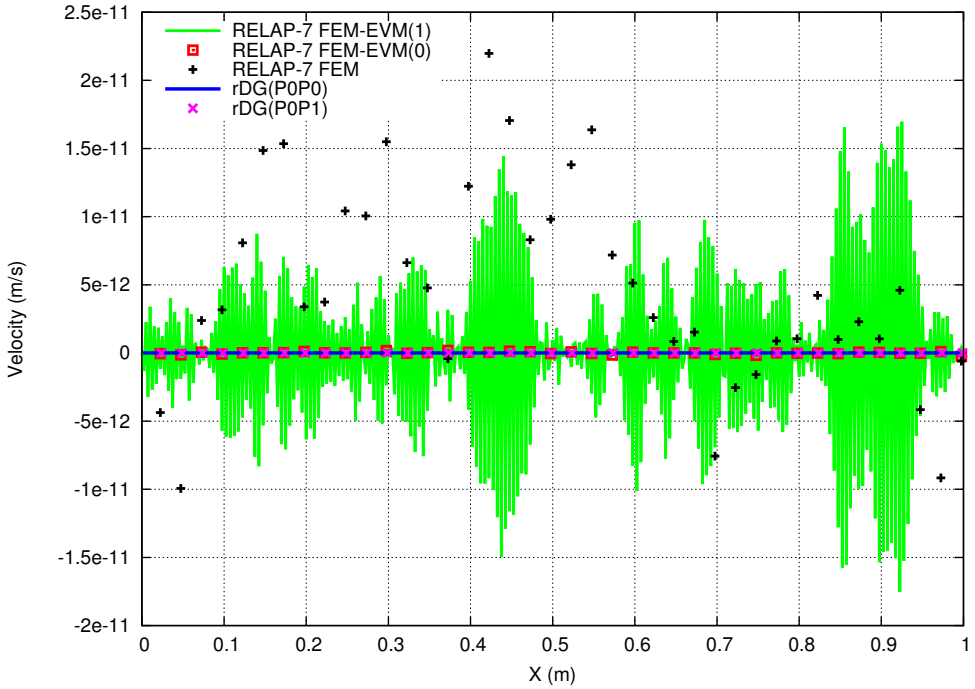

Figure 47. Velocity profiles for liquid at equilibrium.

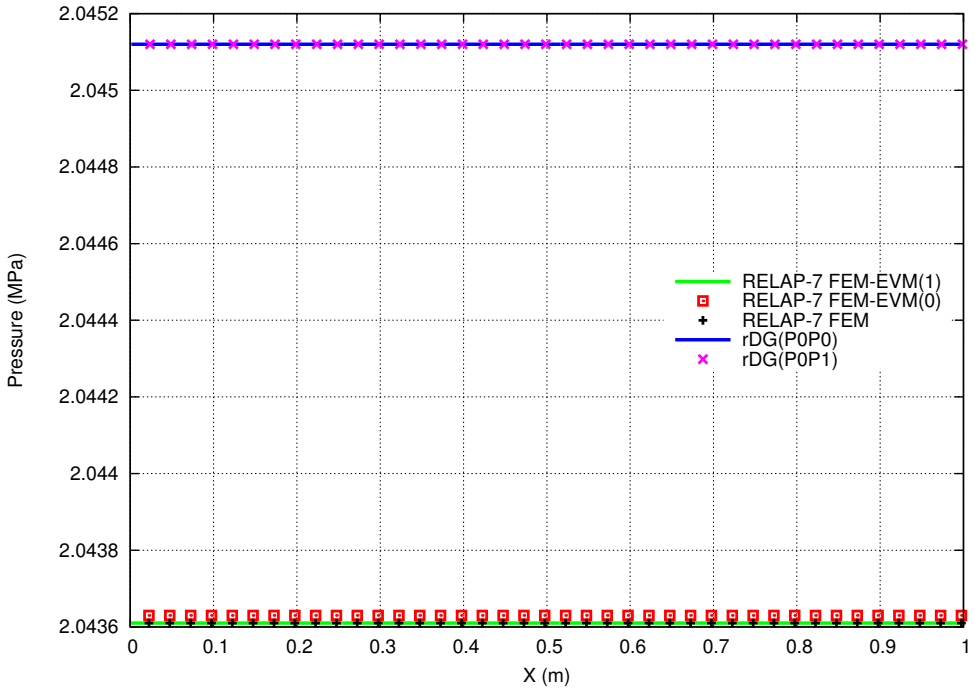

Figure 48. Pressure profiles for vapor at equilibrium.

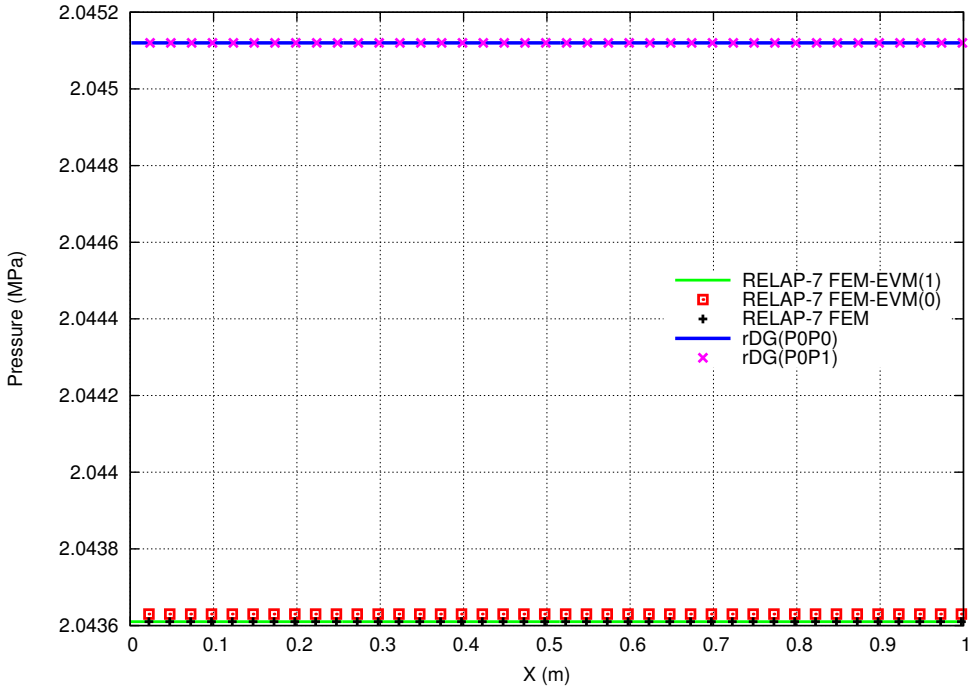

Figure 49. Pressure profiles for liquid at equilibrium.

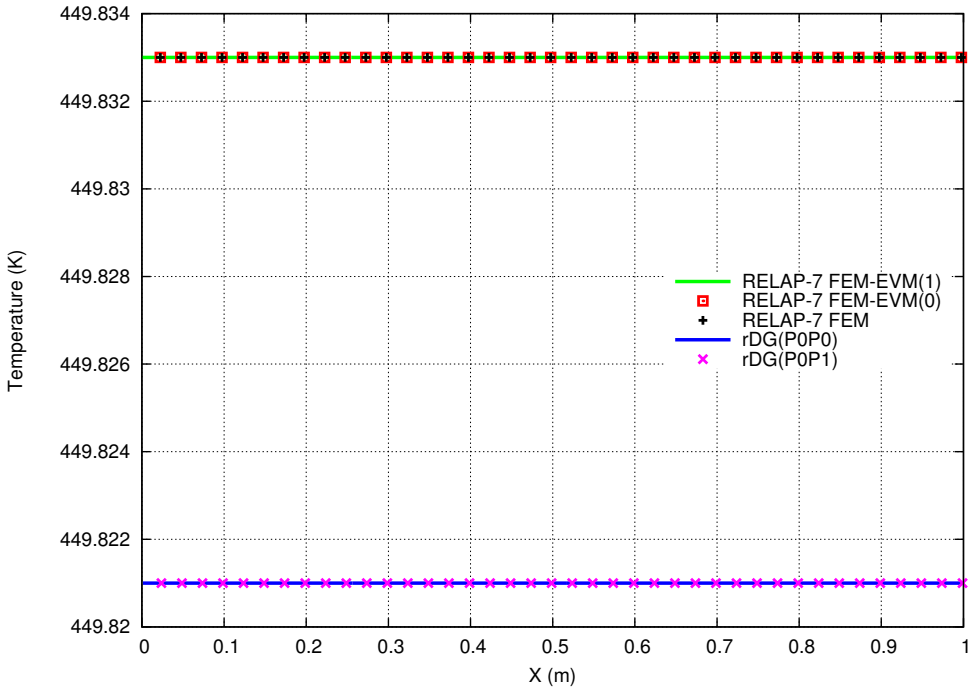

Figure 50. Temperature profiles for vapor at equilibrium.

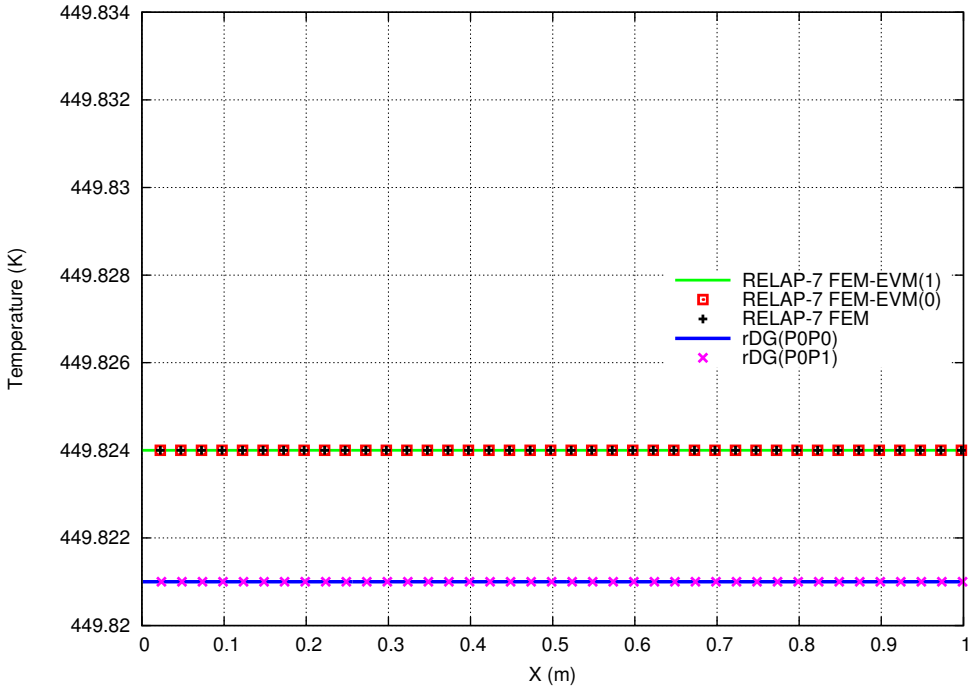

Figure 51. Temperature profiles for liquid at equilibrium.

5 Conclusions

5.1 About the Testing Results

A comparative test case study between a second-order reconstructed discontinuous Galerkin (rDG(P0P1)) method and RELAP-7's finite element method (FEM) with an entropy viscosity method (EVM) based numerical stabilization scheme over a series of benchmark test problems has been presented in this document. The computational results clearly indicate that the performance of the rDG(P0P1) method is superior to that of RELAP-7's FEM method with the use of the EVM stabilization method in all the test problems presented. We emphasize that the same default parameters in the EVM method were used in all the RELAP-7's FEM-EVM simulations in this document. Although additional efforts at parameter tuning in each test problem might improve the RELAP-7's FEM-EVM simulation results, this approach of parameter tuning could be burdensome or challenging for any RELAP-7 user without prior experience or guidance. In comparison, the TVD-type slope limiter in the rDG(P0P1) method is parameter-free, and mathematically guarantees numerical stability at the vicinity of large solution variable gradients. Nevertheless, the current state of the art of rDG methods for the seven-equation, two-phase compressible flow models over a broad spectrum of problems is not expected to be ideal. For example, the recent approximate Riemann solver for the seven-equation model of Furfaro and Saurel [46] should be implemented along with a suitable low-Mach Riemann solver. Above all, as far as the test problems in this document are considered, the second-order rDG(P0P1) method is recommended as a better solution method option for RELAP-7.

5.2 Outlook of the Future of RELAP-7

The achievement of a spatially and temporally accurate, numerically stable and ultimately robust numerical solution method within the MOOSE framework environment for the RELAP-7 thermal-hydraulic modeling component has been one of the major technical challenges over the past six years of the RELAP-7 code development. A numerically stable solution method serves as the cornerstone of the overall design of a thermal-hydraulic modeling code. The resolution of other challenges such as the development, verification and validation of new physical models will not be possible in the absence of a numerically stable solution method. In the case of RELAP-7, a robust numerical stabilization scheme will aid in the resolution of some other challenges. If the rDG(P0P1) method is accepted in the follow-up development of RELAP-7, a series of other related technical issues could be resolved. For example, the fully-implicit solver currently used in RELAP-7 is not an optimal option for solving flow problems that involve wave propagation containing drastic physical variable gradients. In those particular problems, the allowable physical time step size is usually very small, where the use of a fully-implicit time integration solver can become significantly slower than a fully-explicit or a semi-implicit method. As shown in this document, an explicit method properly implemented can be hundreds or thousands of times faster in terms of CPU time. Thus it is recommended that the follow-up development of RELAP-7 include the development of such numerical methods as well as the implementation of the same schemes in an optimal way in the MOOSE framework.

References

[1] Ray A Berry, Richard Saurel, and Olivier LeMetayer. The discrete equation method (DEM) for fully compressible, two-phase flows in ducts of spatially varying cross-section. *Nuclear Engineering and Design*, 240(11):3797–3818, 2010.

[2] Ray A. Berry, Ling Zou, Haihua Zhao, Hongbin Zhang, John W. Peterson, Richard C. Martineau, Samet Y. Kadioglu, and David Andrs. RELAP-7 Theory Manual. Technical Report INL/EXT-14-31366 (Revision 2), Idaho National Laboratory, Idaho Falls, Idaho, United States, March 2016.

[3] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandie. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.

[4] Derek R Gaston, Cody J Permann, John W Peterson, Andrew E Slaughter, David Andrš, Yaqi Wang, Michael P Short, Danielle M Perez, Michael R Tonks, Javier Ortensi, et al. Physics-based multiscale coupling for full core nuclear reactor simulation. *Annals of Nuclear Energy*, 84:45–54, 2015.

[5] R. L. Moore, S. M. Sloan, R. R. Schultz, and G. E. Wilson. RELAP5/MOD3 Code Manual, Volume 1: Code Structure, System Models and Solution Methods. Technical Report NUREG/CR-5335, INEL-95/0174, Idaho National Laboratory, Idaho Falls, ID, United States, June 1995.

[6] S Bajorek et al. TRACE V5.0 Theory Manual, Field Equations, Solution Methods and Physical Models. Technical report, United States Nuclear Regulatory Commission, 2008.

[7] R. A. Berry, M. O. Delchini, and J. Ragusa. RELAP-7 Numerical Stabilization: Entropy Viscosity Method. Technical Report INL/EXT-14/32352, Idaho National Laboratory, Idaho Falls, ID, United States, June 2014.

[8] Marc Delchini. *Extension of the Entropy Viscosity Method to the Multi-D Euler Equations and the Seven-Equation Two-Phase Model*. PhD thesis, Texas A&M University, 2014.

[9] Marc O Delchini, Jean C Ragusa, and Ray A Berry. Entropy-based viscous regularization for the multi-dimensional Euler equations in low-Mach and transonic flows. *Computers & Fluids*, 118:225–244, 2015.

[10] Marc O Delchini, Jean C Ragusa, and Ray A Berry. Viscous regularization for the non-equilibrium seven-equation two-phase flow model. *Journal of Scientific Computing*, 69(2):764–804, 2016.

[11] Marc O Delchini, Jean C Ragusa, and Ray A Berry. Simulations of single-and two-phase shock tubes and gravity-driven wave problems with the RELAP-7 nuclear reactor system analysis code. *Nuclear Engineering and Design*, 319:106–116, 2017.

[12] Hong Luo, Luqing Luo, Robert Nourgaliev, Vincent A Mousseau, and Nam Dinh. A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids. *Journal of Computational Physics*, 229(19):6961–6978, 2010.

[13] Hong Luo, Yidong Xia, Seth Spiegel, Robert Nourgaliev, and Zonglin Jiang. A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction for compressible flows on tetrahedral grids. *Journal of Computational Physics*, 236:477–492, 2013.

[14] Yidong Xia, Jialin Lou, Hong Luo, Jack Edwards, and Frank Mueller. OpenACC acceleration of an unstructured CFD solver based on a reconstructed discontinuous Galerkin method for compressible flows. *International Journal for Numerical Methods in Fluids*, 78(3):123–139, 2015.

[15] Yidong Xia, Xiaodong Liu, and Hong Luo. A finite volume method based on a WENO reconstruction for compressible flows on hybrid grids. In *52nd AIAA Aerospace Sciences Meeting*, number 2014–0939, National Harbor, Maryland, United States, January 2014.

[16] Randall J LeVeque. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[17] Richard Saurel and Rémi Abgrall. A multiphase Godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics*, 150(2):425–467, 1999.

[18] Richard Saurel and Rémi Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21(3):1115–1145, 1999.

[19] Richard Saurel and Olivier Le Métayer. A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation. *Journal of Fluid Mechanics*, 431:239–271, 2001.

[20] Nikolai Andrianov and Gerald Warnecke. The Riemann problem for the Baer–Nunziato two-phase flow model. *Journal of Computational Physics*, 195(2):434–464, 2004.

[21] Li Qiang, Feng Jian-Hu, Cai Ti-min, and Hu Chun-bo. Difference scheme for two-phase flow. *Applied Mathematics and Mechanics*, 25(5):536–545, 2004.

[22] Donald W Schwendeman, Christopher W Wahle, and Ashwana K Kapila. The riemann problem and a high-resolution godunov method for a model of compressible two-phase flow. *Journal of Computational Physics*, 212(2):490–526, 2006.

[23] Vincent Deledicque and Miltiadis V Papalexandris. An exact Riemann solver for compressible two-phase flow models containing non-conservative products. *Journal of Computational Physics*, 222(1):217–245, 2007.

[24] SA Tokareva and Eleuterio F Toro. HLLC-type Riemann solver for the Baer–Nunziato equations of compressible two-phase flow. *Journal of Computational Physics*, 229(10):3573–3604, 2010.

[25] Shan Liang, Wei Liu, and Li Yuan. Solving seven-equation model for compressible two-phase flow using multiple GPUs. *Computers & Fluids*, 99:156–171, 2014.

[26] Yidong Xia, David Andrs, and Richard C. Martineau. BIGHORN Computational Fluid Dynamics Theory, Methodology, and Code Verification & Validation Benchmark Problems. Technical Report INL/EXT-16-39214, Idaho National Laboratory, Idaho Falls, Idaho, United States, August 2016.

[27] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation of the American Mathematical Society*, 67(221):73–85, 1998.

[28] O. Le Métayer, J. Massoni, and R. Saurel. Elaborating equations of state of a liquid and its vapor for two-phase flow models. *International Journal of Thermal Sciences*, 43:265–276, 2004. (In French).

[29] Michael Dumbser, Dinshaw S Balsara, Eleuterio F Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209–8253, 2008.

[30] Michael Dumbser and Olindo Zanotti. Very high order $P_N P_M$ schemes on unstructured meshes for the resistive relativistic MHD equations. *Journal of Computational Physics*, 228(18):6991–7006, 2009.

[31] Michael Dumbser. Arbitrary high order $P_N P_M$ schemes on unstructured meshes for the compressible Navier–Stokes equations. *Computers & Fluids*, 39(1):60–76, 2010.

[32] Dinshaw S Balsara, Christoph Altmann, Claus-Dieter Munz, and Michael Dumbser. A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG+HWENO schemes. *Journal of Computational Physics*, 226(1):586–620, 2007.

[33] Hong Luo, Luqing Luo, Amjad Ali, Robert Nourgaliev, and Chunpei Cai. A parallel, reconstructed discontinuous Galerkin method for the compressible flows on arbitrary grids. *Communications in Computational Physics*, 9(2):363–389, 2011.

[34] Laiping Zhang, Liu Wei, He Lixin, Deng Xiaogang, and Zhang Hanxin. A class of hybrid DG/FV methods for conservation laws I: Basic formulation and one-dimensional systems. *Journal of Computational Physics*, 231(4):1081–1103, 2012.

[35] Laiping Zhang, Liu Wei, He Lixin, Deng Xiaogang, and Zhang Hanxin. A class of hybrid dg/fv methods for conservation laws ii: Two-dimensional cases. *Journal of Computational Physics*, 231(4):1104–1120, 2012.

[36] Ray A. Berry. A Logical Progression of Steps for Implementation and Testing of the 7-equation, Two-Phase Model into a Computational Framework. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Sun Valley, Idaho, United States, May 2013. American Nuclear Society, LaGrange Park, Illinois.

[37] Jan-Reneé Carlson. Inflow/Outflow Boundary Conditions with Application to FUN3D. Technical Report NASA/TM-2011-217181, NASA Langley Research Center, Hampton, Virginia, United States, 2011.

[38] Gianmarco Mengaldo, Daniele De Grazia, J Peiro, Antony Farrington, F Witherden, PE Vincent, and SJ Sherwin. A Guide to the Implementation of Boundary Conditions in Compact High-Order Methods for Compressible Aerodynamics. In *7th AIAA Theoretical Fluid Mechanics Conference*, AIAA Paper No. 2014-2923, Atlanta, Georgia, United States, June 2014. American Institute of Aeronautics and Astronautics.

[39] P Batten, MA Leschziner, and UC Goldberg. Average-state Jacobians and implicit methods for compressible viscous and turbulent flows. *Journal of Computational Physics*, 137(1):38–78, 1997.

[40] Timothy J Barth and Dennis C Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Sciences Meeting*, AIAA Paper No. 89-0366, Reno, Nevada, United States, January 1989. American Institute of Aeronautics and Astronautics.

[41] Venkat Venkatakrishnan. Convergence to steady state solutions of the euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118(1):120–130, 1995.

[42] M. Kunick, R. A. Berry, R. C. Martineau, H-J Kretzschmar, and U. Gampe. Application of the new IAPWS Guideline on the fast and accurate calculation of steam and water properties with the Spline-Based Table Look-Up Method (SBTL) in RELAP-7. *Kerntechnik*, 82(3):264–279, 2017.

[43] Peter D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1):159–193, 1954.

[44] I Tiselj, G Černe, A Horvat, J Gale, I Parzer, M Giot, JM Seynhaeve, B Kucienska, and H Lemonnier. WAHA3 Code Manual, WAHA Loads Project Deliverable D10, 2004.

[45] Lukasz Sokolowski and Zbigniew Koszela. RELAP5 capability to predict pressure wave propagation phenomena in single-and two-phase flow conditions. *Journal of Power Technologies*, 92(3):150, 2012.

[46] Damien Furfaro and Richard Saurel. A simple HLLC-type Riemann solver for compressible non-equilibrium two-phase flows. *Computers & Fluids*, 111:159–178, 2015.


INL
Idaho National Laboratory