ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method · Nashville, Tennessee · April 19–23, 2015, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2015)

INITIAL 1-D SINGLE PHASE LIQUID TRANSIENT VERIFICATION OF COBRA-TF

Chris Dances and Dr. Maria Avramova
Department of Mechanical and Nuclear Engineering
The Pennsylvania State University
137 Reber Building, University Park, PA, 16802, USA
cad39@psu.edu; mna109@psu.edu

Dr. Vince Mousseau
Computer Science Research Institute
Sandia National Laboratories
1450 Innovation Parkway, Albuquerque, NM 87123, USA
vamouss@sandia.gov

ABSTRACT

Abstract . . .

Key Words: List no more than five key words

1 INTRODUCTION

For the past several decades, the primary focus in nuclear engineering within the United States has been focused on light water reactors (LWR). Commercially, all nuclear reactors are either boiling water reactors (BWR) or pressurized water reactors (PWR). Correct computation of the thermal hydraulics within the reactor core leads to effi- cient design and accuracy in the safety analysis. A popular subchannel code for modelling the hydrodynamics with in the reactor core is COBRA-TF. This FORTRAN based code solves 8 conservation equations for liquid, entrained droplet, and vapor phases in 3-D dimmensions [1]. The conservation equations analytically reduce into a pressure matrix in a semi-implicit method with rod temperatures solved for explicitly. Because the physics are integrated into the numerical solution, the equations must be linear and the solution method semi-implicit. With a residual formulation, greater flexibility and control over the numerical solution is possible. COBRA-TF was originally written in FORTRAN 77, but over the years has been partially updated to newer versions of Fortran.

2 COBRA-TF

The finite volume structure in COBRA-TF in figure 1 is for a one-dimmensional channel in the axial direction with n number of cells. The first and last cells at 0 and n + 1 are ghost cells and act as the boundary conditions for the problem. Pressure, enthalpy, and density are averaged over the cell volume and are located at the center of the cell. Mass flow rate and velocity are located at the faces in between cells. The cells are represented with an index i, and the faces with indexes of $i + \frac{1}{2}$ or $i - \frac{1}{2}$. This project will initially focus on this 1-D configuration. Usually the code is 3-D, with channels connecting to each other in two more dimmensions.

2.1 Software Quality Assurance

Software quality assurance is a set of tools and procedures that helps ensure that the software is reliable. COBRA-TF is managed by Github Repository setup and maintained by CASL. An exensive test matrix is run before each major push to ensure that the code meets the specificed requirements. The test matrix consists of unit tests, code coverage runs, validation problems, and validation problems. The code documentation consists of a theory manual, a users manual, a developers manual, and a validation manual. Further work might involve using autodocumentation tools to keep an up to date developers manual. This paper will be the begining of a verification manual, integrating this verificaiton problem directly into the test matrix.

2.2 1-D Single Phase Liquid Conservation Equations

The thermal hydraulics of a LWR core is an important part of nuclear reactor desigin. COBRA-TF solves 8 conservation equations for liquid, entrained droplet, and vapor phases of water boiling within the rod structure of a LWR reactor core [1]. Currently, the conservation equations analytically reduce into a pressure matrix in a semi-implicit method with rod temperatures solved for explicitly. This work involves representing the 1-D single phase liquid conservation equations and calculated

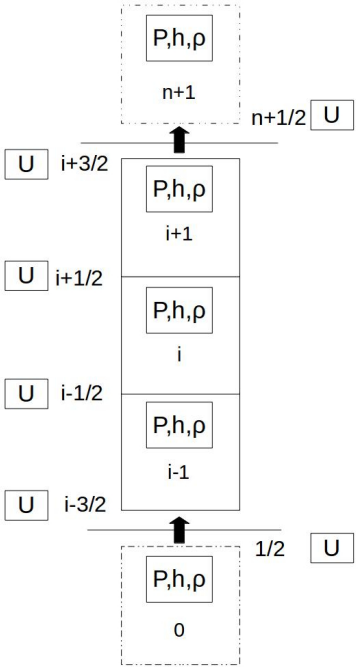

Figure 1. The finite volume structure for COBRA-TF

variables in a residual formulation. The full jacobian matrix can then be built numerically, and can then either be reduced to a pressure matrix or solved directly.

The single phase Euler partial differential equations for mass (1), momentum (2), and energy (3) coresond to the unknown variables density ρ, velocity u, pressure P, and enthalpy h. The first terms in each of the equations are temporal terms. The rest of the terms are steady state spatial terms.

$$\frac{\partial \rho}{\partial t} + \bigtriangledown \rho u = 0 \tag{1}$$

$$\frac{\partial \rho u}{\partial t} + \bigtriangledown \rho u^2 + \bigtriangledown P - \rho g = 0 \tag{2}$$

$$\frac{\partial \rho h}{\partial t} - \frac{\partial P}{\partial t} + \bigtriangledown (\rho u h) = 0 \tag{3}$$

2.3 Modified Equation Analysis

The original mass balance equation can be re-written to look like equation 4. Using upwinding, the finite difference can be written to look like equation 5. A second order Taylor series approximation can be used for ρ_i^{n+1} and ρ_{i-1}^n as shown in equations 6 and 7 respectively. The higher

order terms ($O(\Delta x^2, \Delta t^2)$) are not taken into account for this approximation. The Taylor series approximations can then be substituted into 5 to yield 8. This is the beginning of the modified equation analysis. The goal will be to isolate the original PDE and define the truncation error.

$$\frac{\partial \rho}{\partial t} + U_0 \frac{\partial \rho}{\partial x} = 0 \tag{4}$$

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + U_0 \frac{\rho_i^n - \rho_{i-1}^n}{\Delta x} = 0 \tag{5}$$

$$\rho_i^{n+1} = \rho_i^n + \frac{\partial \rho}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t^2 + O(\Delta t^3) \tag{6}$$

$$\rho_{i-1}^n = \rho_i^n - \frac{\partial \rho}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}\Delta x^2 + O(\Delta x^3) \tag{7}$$

The lengthy equation 8 can be reduced to equation 9 since the ρ_i^n terms subtract out and the Δt and Δx terms in the denominator cancel out. This reduced equation can the be re-written into equation 10, with the original PDE followed by the truncation terms. Notice how the terms on the right are dependent on both the numerical spacing Δt and Δx, but also on the second derivatives of density with respect to space and time.

$$\frac{\left(\rho_i^n + \frac{\partial \rho}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t^2\right) - \rho_i^n}{\Delta t} + U_0 \frac{\rho_i^n - \left(\rho_i^n - \frac{\partial \rho}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}\Delta x^2\right)}{\Delta x} + O(\Delta x^2, \Delta t^2) = 0 \tag{8}$$

$$\frac{\partial \rho}{\partial t} + \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t + U_0\left(\frac{\partial \rho}{\partial x} - \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}\Delta x\right) + O(\Delta x^2, \Delta t^2) = 0 \tag{9}$$

$$\frac{\partial \rho}{\partial t} + U_0 \frac{\partial \rho}{\partial x} + \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t - U_0 \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}\Delta x + O(\Delta x^2, \Delta t^2) = 0 \tag{10}$$

Before we can procede, we need to take the derivative of the original PDE with respect to space and time as shown in equations 11 and 12 respectively. These two derivatives can substitute into each other using the common term $\frac{\partial^2 \rho}{\partial x \partial t}$. The second derivatives of density with respect to space and time are therefore related by the velocity squared as shown by equation 13.

$$\frac{\partial^2 \rho}{\partial t^2} + U_0 \frac{\partial^2 \rho}{\partial x \partial t} = 0 \tag{11}$$

$$\frac{\partial^2 \rho}{\partial t \partial x} + U_0 \frac{\partial^2 \rho}{\partial x^2} = 0 \tag{12}$$

$$\frac{\partial^2 \rho}{\partial t^2} = U_0^2 \frac{\partial^2 \rho}{\partial x^2} \tag{13}$$

This relationship can then be substituted back into equation 10, which can be reduced to equation 15 after igonoring the higher order terms. The error depends on the CFL number, the axial spacing, and the second order derivative of density with respect to space. This derivative is what gives the error the characterisitcs of diffusion. When the CFL number is less than one, the error term is negative and the diffusion is dampening. When the CFL number is greater than one, the error term becomes positive, and the accumulation of the error destabilizes the solution.

$$\frac{\partial \rho}{\partial t} + U_0 \frac{\partial \rho}{\partial x} - \frac{1}{2}\left(\Delta x U_0 \frac{\partial^2 \rho}{\partial x^2} - U_0^2 \frac{\partial^2 \rho}{\partial x^2} \Delta t\right) + O(\Delta x^2, \Delta t^2) = 0 \tag{14}$$

$$\frac{\partial \rho}{\partial t} + U_0 \frac{\partial \rho}{\partial x} - \frac{\Delta x U_0}{2} \frac{\partial^2 \rho}{\partial x^2} (1 - CFL) + O(\Delta x^2, \Delta t^2) = 0 \tag{15}$$

Modified eqauation analysis can be applied to the energy balance equation presented in equation 16. The energy equation is presented in a form where the momentum equation was substituted in as zero and then divided through by density. The result presented in equation 17 is similar in form to the result for the mass balance equation 15.

$$\frac{\partial h}{\partial t} - \frac{1}{\rho} \frac{\partial P}{\partial t} + U_0 \frac{\partial h}{\partial x} = 0 \tag{16}$$

$$\frac{\partial h}{\partial t} - \frac{1}{\rho} \frac{\partial P}{\partial t} + U_0 \frac{\partial h}{\partial x} - \frac{\Delta x U_0}{2} \frac{\partial^2 h}{\partial x^2} (1 - CFL) = 0 \tag{17}$$

2.4 Residual Formulation and Jacobian Construction

A residual is simply the difference between the value at some future time $n + 1$ and the value at the current iteration k. This can be applied to desired variables as shown in equations (18), (19),

(20), and (21). Residuals can also be applied to the conservation equations by substituting the definition of the residual variables into the conservation equations. This will effectively change any variables evaluated at $n+1$ to k. Each cell will have three residual variables and three residual equations. For the entire solution, we will then have a residual variable array δX, and a residual function array $F(X)$ which defines a linear system as seen in equation (22).

$$\delta P_i = P_i^{n+1} - P_i^k \tag{18}$$

$$\delta h_i = h_i^{n+1} - h_i^k \tag{19}$$

$$\delta u_{i+\frac{1}{2}} = u_{i+\frac{1}{2}}^{n+1} - u_{i+\frac{1}{2}}^k \tag{20}$$

$$\delta \rho_i = \rho_i^{n+1} - \rho_i^k \tag{21}$$

$$J\delta X = -F(X) \tag{22}$$

The Jacobian matrix is defined in equation (23) as the derivative of each response of the function F_j with respect to each variable X_i. The derivative can be calculated numerically as shown by equation (24) where ϵ is a small numerical value. For COBRA-TF the equations are linear, and this numerical approximation of the Jacobian matrix is exact. This should produce the same jacobian matrix that COBRA-TF currently generates analytically.

$$J_{i,j} = \frac{\partial F_j(X)}{\partial X_i} \tag{23}$$

$$J_{i,j} \approx \frac{F_j(X_i + \epsilon) - F_j(X)}{\epsilon} \tag{24}$$

To build the jacobian matrix, an object oriented class was created that contains three arrays. An array that points to the residual functions, an array that points to the position within a target variable arrray, and an array that has the index that the function is to be evaluated at. These lists can be appended to in any order, but have to be appended all at the same time so that variables and functions must correspond with each other. Then to construct the jacobian matrix, the residual function and residual variable arrays can each be looped over to numerically build the jacobian matrix as seen in figure 2.

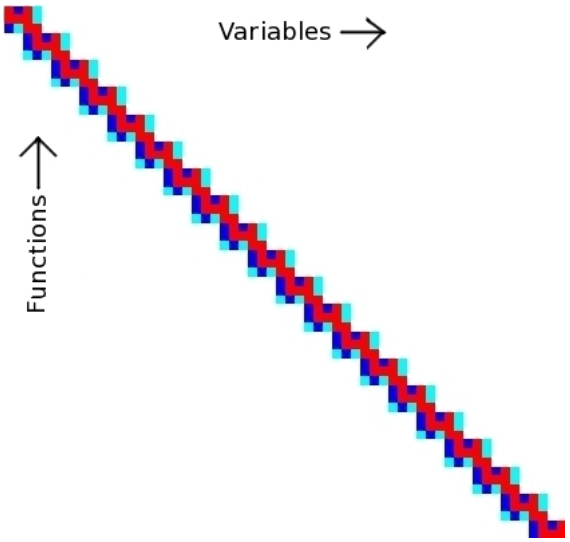

Figure 2. Strucutre of the jacobian matrix for single phase liquid

3 CODE VERIFICATION

Code verification is the set of procedures set in place to ensure that the code was written properly. The procedures can use the following as code verification criteria from least to most rigorous are expert judgement, error quantification, consistencey / convergence, and order of accuruacy [2]. For this work the method of known solutions will be used to initially quantify the error at for a particular problem setup. Next a Richardson Extrapolation will be used to check for convegence and order of accruacy of the error in space and time.

3.1 Isokinetic Sine Wave Advection Problem Setup

Needs a figure and a table of parameters. Geometry and reference conditions should be for a PWR.

The problem is to transiently vary the inlet enthalpy h and inlet mass flow rate $\dot{m}$ using a smooth trigonometric function so as to keep velocity constant throughout the solution. Using a cosine, the analytical solution for a variable Y at time index j and space index i, where Y_1 is the initial value, Y_2 is the minimum value of the wave, and P is the period of the wave. The time step size dt, axial mesh size dx, and velocity V_o are assumed constant. If $V_o * j * dt > i * x$, then this equation doesn't apply and the value should just equal the initial value Y_1.

$$Y(i,j) = \frac{1}{2}\left((Y_1 + Y_2) + (Y_1 - Y_2)cos\left(\frac{2\pi}{P}\left(j * dt + \frac{i * dx}{V_o}\right)\right)\right) \tag{25}$$

This analytical solution can be applied to mass flow rate $\dot{m}$, density ρ, liquid enthalpy h, and liquid temperature T. Since velocity and pressure are cosntant, mass flow rate will be proportional

to density, and enthlapy will be proportional to temperature.

3.2 Input Verification

For the inlet condition, a transient data table was generated for enthalpy and mass flow rate and applied at the inlet node. The comparison between the data table and the output in CTF are shown for enthalpy and mass flow rate in figures 3 and 4 respectively. The CTF output was read from hdf5 data files at each point in time, wich omitted the actual ghost cell where these values were applied. The CTF values are located at the nearest node to the inlet, and therefore will be slightly out of phase to the exact values in the figure. This difference is more notable for smaller mesh sizes.

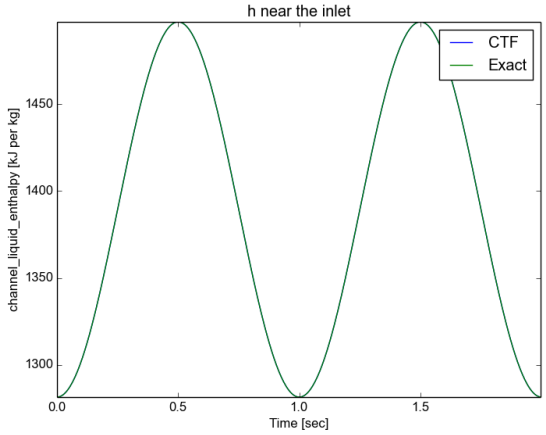

Figure 3. Enthalpy near the inlet and the analytical solution

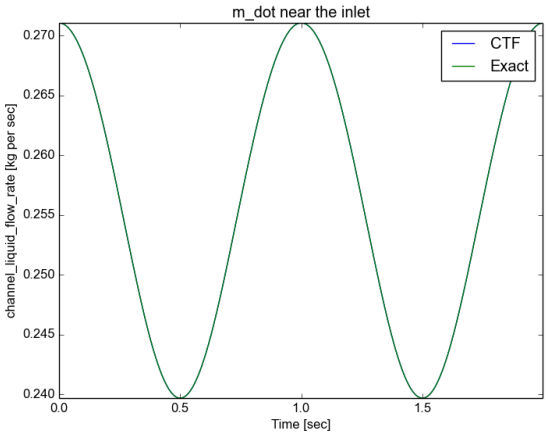

Figure 4. Mass Flow rate near the inlet and the analytical solution

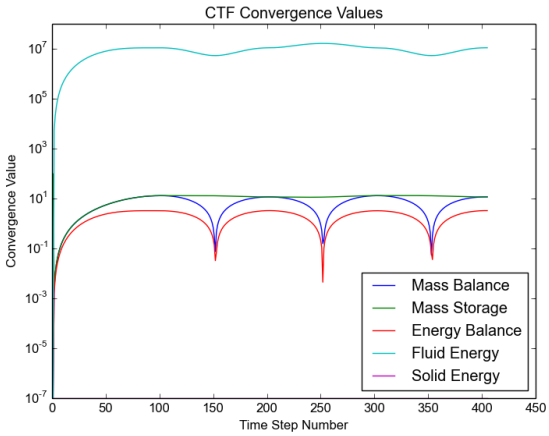

Figure 5. Code Convergence Criteria for the original version of COBRA-TF

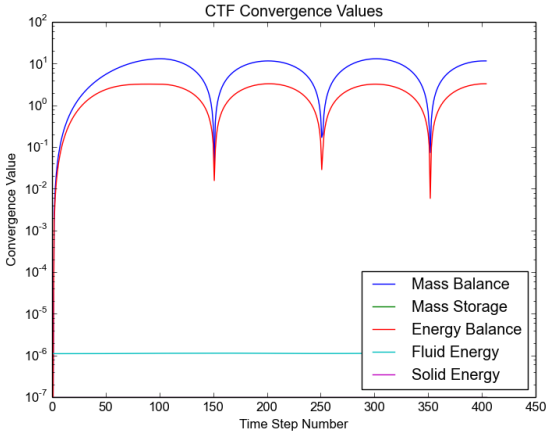

Figure 6. Code Convergence Criteria for the residual version of COBRA-TF

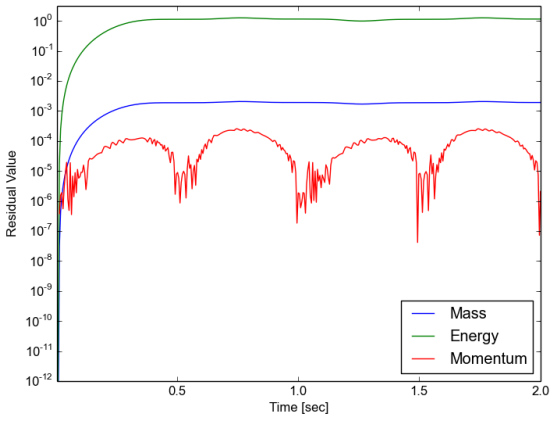

Figure 7. Sumnation of the residuals for the residual version of COBRA-TF

3.3 Code Convergence

4 TEMPORAL VERIFICATION

Both COBRA-TF and residual version semi-implicit method, residual fully implicit method with and without linear EOS.

4.1 Problem Setup

This compares the exact solution to the COBRA-TF Solution Methods

4.1.1 Error Quantification

Show qualatative plot and demonstrate that the error is quantified for a single time step and space discretization

A figure and a table of l1 normalized error

4.2 Convergence of Error

Show how error behaves for different time and space sizes

4.3 Order of Accuracy

5 CONCLUSIONS

Present your summary and conclusions here.

6 ACKNOWLEDGMENTS

Dr. Vince Mosseau, Dr. Maria Avramova, Dr. Kostadin Ivanov, and Nathan Porter.

7 REFERENCES

[1] R. K. Salko, "CTF Theory Manual," The Pennsylvania State University (2014).

[2] K. P. Knupp, *Verification of Computer Codes in Computational Science and Engineering*, Chapman and Hall/CRC, Boca Raton, FL (2003).