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 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 variables in a residual formulation. This residual formulation should allow for easier and more in depth verification analysis. This paper details the intitial comparison of the residual formulation to the original code.

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 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.

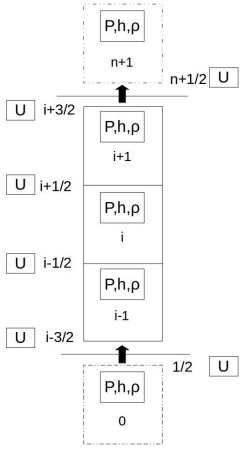

Figure 1. The finite volume structure for COBRA-TF

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 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 (4), (5), (6), and (7). 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 (8).

$$\delta P_i = P_i^{n+1} - P_i^k \tag{4}$$

$$\delta h_i = h_i^{n+1} - h_i^k \tag{5}$$

$$\delta u_{i+\frac{1}{2}} = u_{i+\frac{1}{2}}^{n+1} - u_{i+\frac{1}{2}}^k \tag{6}$$

$$\delta \rho_i = \rho_i^{n+1} - \rho_i^k \tag{7}$$

$$J\delta X = -F(X) \tag{8}$$

The Jacobian matrix is defined in equation (9) 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 (10) 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.

$$J_{i,j} = \frac{\partial F_j(X)}{\partial X_i} \tag{9}$$

$$J_{i,j} \approx \frac{F_j(X_i + \epsilon) - F_j(X)}{\epsilon} \tag{10}$$

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.

2.4 Modified Equation Analysis

The original mass balance equation can be re-written to look like equation 11. Using upwinding, the finite difference can be written to look like equation 12. A second order Taylor series approximation can be used for ρ_i^{n+1} and ρ_{i-1}^n as shown in equations 13 and 14 respectively. The higher order terms ($O(\Delta x^2, \Delta t^2)$) are not taken into account for this approximation. The Taylor

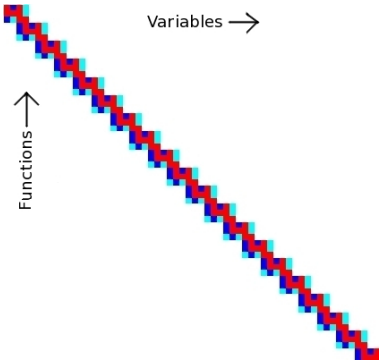

Figure 2. Strucutre of the jacobian matrix for single phase liquid

series approximations can then be substituted into 12 to yield 15. This is the beginning of the modified equation analysis. The goal will be to isolate the original PDE and define the truncation error.

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

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + U_0 \frac{\rho_i^n - \rho_{i-1}^n}{\Delta x} = 0 \tag{12}$$

$$\rho_i^{n+1} = \rho_i^n + \frac{\partial \rho}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t^2 + O(\Delta t^3) \tag{13}$$

$$\rho_{i-1}^n = \rho_i^n - \frac{\partial \rho}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}\Delta x^2 + O(\Delta x^3) \tag{14}$$

The lengthy equation 15 can be reduced to equation 16 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 17, 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.

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

$$\frac{\partial \rho}{\partial t} + \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t + U_0\left(\frac{\partial \rho}{\partial x} - \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}\Delta x\right) + O(\Delta x^2, \Delta t^2) = 0 \tag{16}$$

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

Before we can procede, we need to take the derivative of the original PDE with respect to space and time as shown in equations 18 and 19 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 20.

$$\frac{\partial^2 \rho}{\partial t^2} + U_0 \frac{\partial^2 \rho}{\partial x \partial t} = 0 \tag{18}$$

$$\frac{\partial^2 \rho}{\partial t \partial x} + U_0 \frac{\partial^2 \rho}{\partial x^2} = 0 \tag{19}$$

$$\frac{\partial^2 \rho}{\partial t^2} = U_0^2 \frac{\partial^2 \rho}{\partial x^2} \tag{20}$$

This relationship can then be substituted back into equation 17, which can be reduced to equation 22 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.

$$\frac{\partial \rho}{\partial t} + U_0 \frac{\partial \rho}{\partial x} - \frac{1}{2}\left(\Delta x U_0 \frac{\partial^2 \rho}{\partial x^2} - U_0^2 \frac{\partial^2 \rho}{\partial x^2}\Delta t\right) + O(\Delta x^2, \Delta t^2) = 0 \tag{21}$$

$$\frac{\partial \rho}{\partial t} + U_0 \frac{\partial \rho}{\partial x} - \frac{\Delta x U_0}{2}\frac{\partial^2 \rho}{\partial x^2}\left(1 - CFL\right) + O(\Delta x^2, \Delta t^2) = 0 \tag{22}$$

Modified eqauation analysis can be applied to the energy balance equation presented in equation 23. 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 24 is similar in form to the result for the mass balance equation 22.

$$\frac{\partial h}{\partial t} - \frac{1}{\rho}\frac{\partial P}{\partial t} + U_0 \frac{\partial h}{\partial x} = 0 \tag{23}$$

$$\frac{\partial h}{\partial t} - \frac{1}{\rho}\frac{\partial P}{\partial t} + U_0 \frac{\partial h}{\partial x} - \frac{\Delta x U_0}{2}\frac{\partial^2 h}{\partial x^2}(1 - CFL) = 0 \tag{24}$$

3 PROBLEM SETUP

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

The verication problem is defined as a single horizontal channel problem the base parameters listed in table I. The problem will have a fixed channel area and perimeter across the entire height of the channel with no grid spacers. The velocity and pressure are assumed to be constant, but small fluctuations may occur due to coding mistakes or nuermical noise. The channel geometry and operating conditions are taken to be approximate a standard PWR. The inlet of the channel has a constant velocity with a fluctuating enthalpy that corresponds to a 37.5 °C temperature change.

Table I. Problem Parameters

Parameter	Symbol	Value	Unit
Axial Height	H	3.6586	m
Channel Area	A_{ch}	4.94E-005	m^2
Wetted Perimeter	P_w	1.49E-002	m
Velocity	V_o	7.35	$\frac{m}{s}$
Pressure	P_o	155.00	bar
Temperature 1	T_1	289.500	°C
Temperature 2	T_2	327.00	°C
Enthalpy 1	h_1	1281.55	$\frac{kJ}{kg}$
Enthalpy 2	h_2	1497.21	$\frac{kJ}{kg}$
Mass Flow Rate 1	$\dot{m}_1$	0.2713	$\frac{kg}{s}$
Mass Flow Rate 2	$\dot{m}_2$	0.2399	$\frac{kg}{s}$
Final Time	t_f	2.00	sec
Wave Frequency	ω	1.00	Hz

The lookup table to vary the inlet enthalpy h and inlet mass flow rate $\dot{m}$ use smooth trigonometric functions given in equations 25 and 26. The equations assume constant axial spacing Δx and

time step size Δt where i and j are the spatial and temporal index respectively. These equations should also behave as the known solutions throughout the entire domain of the problem. The enthalpy and mass flow rate should vary proportionally to the density as to create an isokinetic boundary condition at the inlet. However this is dependent on the steam tables used. A python script was used to generate the data tables according to trignometric equations using lookup tables that mimic the IAPWS-97 steam tables used by the code [3].

$$h(i,j) = \frac{1}{2}\left((h_1 + h_2) + (h_1 - h_2)cos\left(\omega\left(j\Delta t + \frac{i\Delta x}{V_o}\right)\right)\right) \tag{25}$$

$$\dot{m}(i,j) = \frac{1}{2}\left((\dot{m}_1 + \dot{m}_2) + (\dot{m}_1 - \dot{m}_2)cos\left(\omega\left(j\Delta t + \frac{i\Delta x}{V_o}\right)\right)\right) \tag{26}$$

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 but this effect is small in the figure.

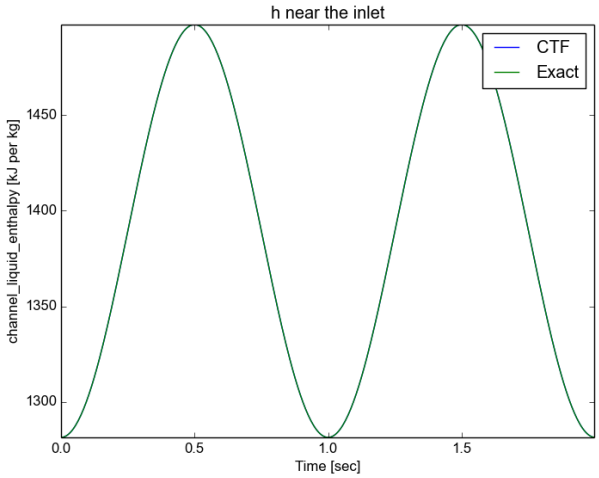

Figure 3. Enthalpy Near the Inlet and the Analytical Solution

For the original version of COBRA-TF there is a small discrepancy in the way the density is calculated at the inlet that causes the velocity to be non-consatnt. This is considered small for this problem and should not greatly affect the order of accuracy. The residual formulation was coded in such a way as to avoid this problem and has considerably less fluctation in the inlet velocity.

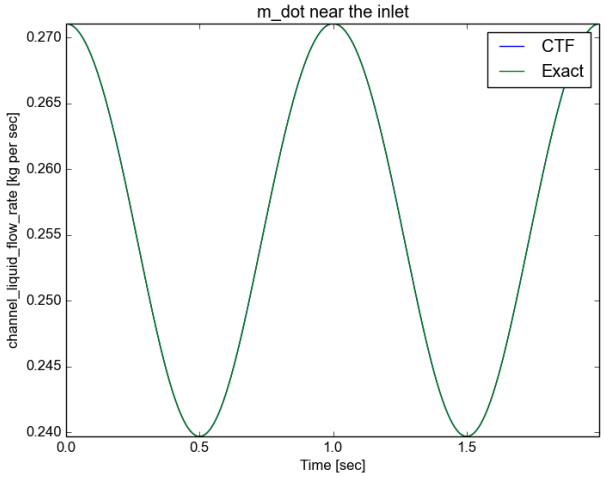

Figure 4. Density Near the Inlet and the Analytical Solution

3.2 Code Convergence

The current version of COBRA-TF uses global code convergance criteria across the inlet and the entire domain. These approximate the residauls for the equation. For this problem, the solid energy storate is zero since there are not any heat structures present. The original convergence criteria in COBRA-TF is shown in figure 5 and for the residual version in figure 5. Note how the residual version has zero mass storage and fluid energy terms. The other terms are balances across the inlet and outlet of the problem and follow the sinusodial nature of the transient.

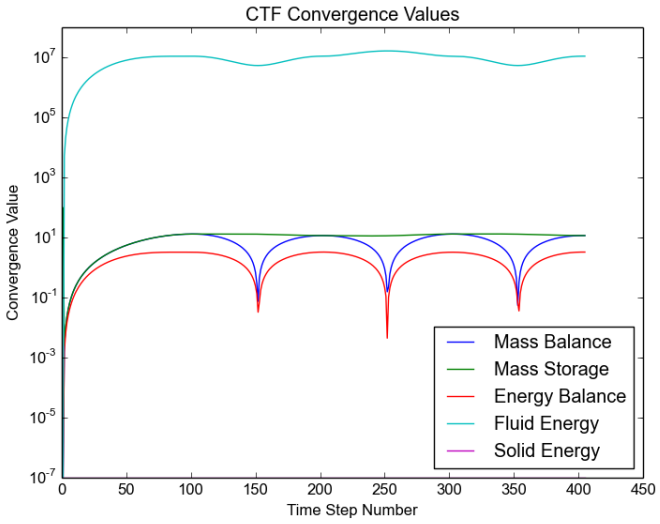

Figure 5. Code Convergence Criteria for the original version of COBRA-TF

The residual formulation prints out the sumnation of the residuals functions across the domain to an output file for eac time step and can be seen in figure7.

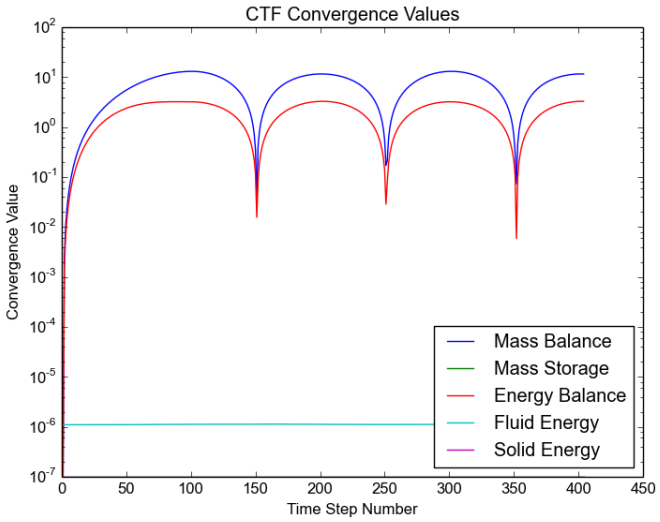

Figure 6. Code Convergence Criteria for the residual version of COBRA-TF

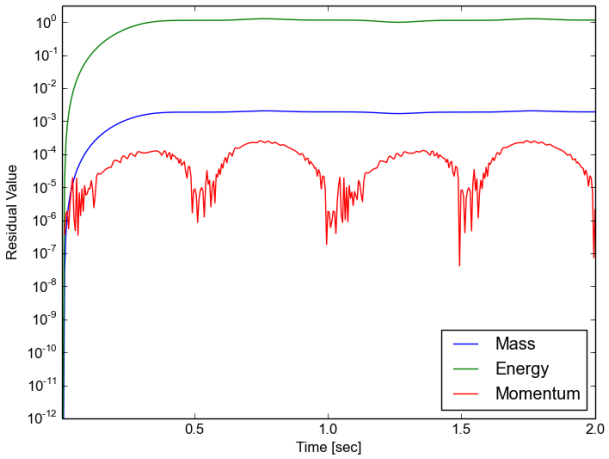

Figure 7. Sumnation of the residuals for the residual version of COBRA-TF

3.3 Error Quantification

4 RICHARDSON EXTRAPOLATION

Richardson extrapolation is . . .

It is used to compare that the error converges to zero with the correct order of accuracy in space and time. The order of accuracy needs to match the modified equation analysis.

These are the equations used . . .

4.1 Convergence of Error

Talk about how the error for all of the variables approaches zero, with a linear trend line.

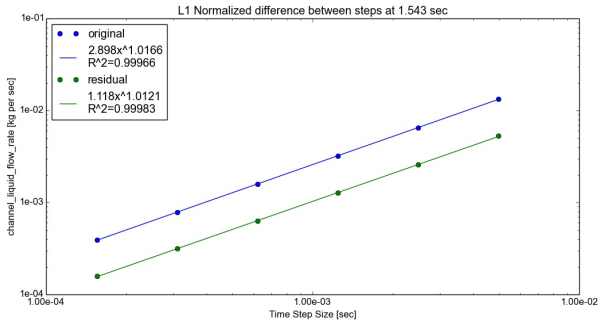

Figure 8. Difference Between Succssessive Temporal Refinements

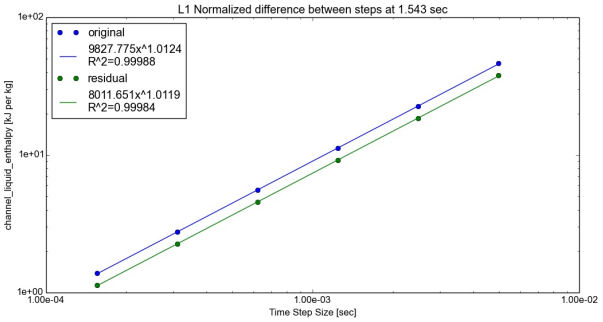

Figure 9. Difference Between Succssessive Temporal Refinements

Show how error behaves for different time and space sizes

4.2 Order of Accuracy

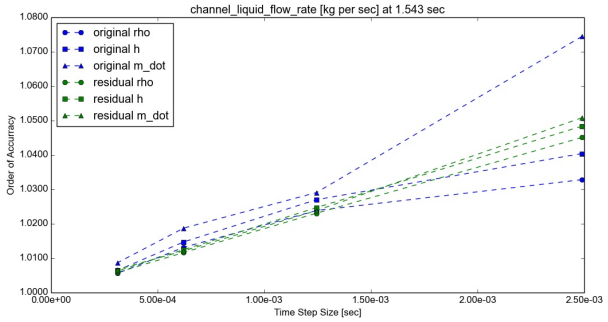

Figure 10. Temporal Order of Accuracy

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).

[3] IAPWS, "Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam," The International Association for the Properties of Water and Steam (IAPWS) (2007).