Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 200 (2004) 104–132
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Implicitly balanced solution of the two-phase flow equations coupled to nonlinear heat conduction

V.A. Mousseau *

Los Alamos National Laboratory, Mail Stop B216, Los Alamos, NM 87545, USA

Received 18 November 2003; received in revised form 23 March 2004; accepted 25 March 2004
Available online 27 April 2004

Abstract

This paper presents the solution of the two-phase flow equations coupled to nonlinear heat conduction using the Jacobian-free Newton–Krylov (JFNK) method which employs a physics-based preconditioner. Computer simulations will demonstrate that the implicitly balanced solution obtained from the JFNK method is more accurate than traditional approaches that employ operator splitting and linearizing. Results will also indicate that by employing a physics-based preconditioner the implicitly balanced solution can provide a more accurate solution for the same amount of computer time compared to the traditional approach for solving these equations. Finally, convergence plots will show that as the transient time lengthens, the implicitly balanced solution can maintain this higher level of accuracy at much larger time steps.

Keywords: Two-phase flow; Nonlinear; Implicit; Preconditioning

1. Introduction

Two-phase flow has a variety of meanings in different areas. To narrow the scope of this paper, the focus will be placed on a system of water and steam flowing through a channel which is attached to a heat conducting solid slab. The slab is coupled to the two-phase flow in the channel through convective heat transfer and is heated by a source (Q) which is on the opposite side of the slab from the two-phase fluid flow (see Fig. 1). Fig. 1 shows the water flow on the left (bubbles in the water indicating two-phase flow), with a fixed velocity inflow boundary condition ($V = V_0$) and a fixed pressure outflow boundary condition ($P = P_0$). The slab is insulated on the top an bottom ($\frac{\partial T}{\partial x} = 0$) and has a heat source on the right side which has a cosine profile from the top to bottom (this is a simplified form of the power profile from a fission reaction).

* Tel.: +1-505-665-5891; fax: +1-505-667-3726.
E-mail address: vmss@lanl.gov (V.A. Mousseau).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.03.009

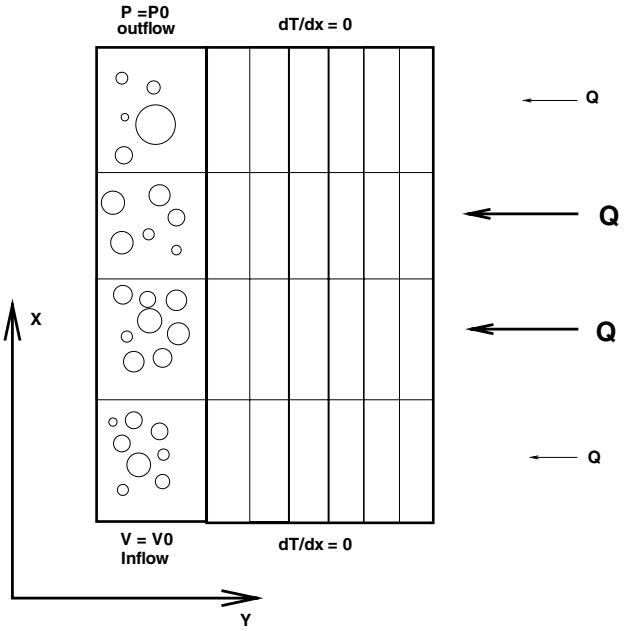

Fig. 1. Problem schematic.

This system is a very simplified version of a nuclear reactor. The heat source (Q) comes from nuclear fission and is conducted through a metallic structure to be removed by the water-steam convection. There are large computer codes (such as, TRAC-M [1] and RELAP5 [2,3] and their progeny) that were designed to simulate the two-phase flow, heat conduction, and nuclear fission reaction in a nuclear reactor. The numerical methods in these codes are based on using operator splitting to separate the nuclear reaction, the heat conduction and the fluid flow into three independent algorithms. In addition, the semi-implicit solution method for the fluid flow equations has further linearizations and operator splitting to be described in detail in Appendix A. The basic numerical methods in these codes are still very similar to the codes that were originally written in the late 1970s and early 1980s. Because of the small memory and very slow computational speed of computers in the 1980s, these simplifications were required to make the simulations tractable.

Efforts were quickly undertaken after these codes were written to increase the level of implicitness in both TRAC-M with SETS [4] and RELAP5 with its Nearly Implicit [5]. Both of these efforts met with limited success at creating a time integration technique that could step over the material Courant Friedrichs and Lewy (CFL) stability limit caused by the semi-implicit integration technique [6].

Since the 1980s, when the six equation two-phase flow model was originally developed for nuclear reactor simulation, there has been significant work on improving the equations to allow them to more accurately model the physics of two-phase flow. This work includes the incorporation of Lagrangian bubble and droplet fields (see [7,8] and references within) to more accurately model the interactions between the two phases. Additional work includes modeling interfacial area transport (see [9,10] and references within) with partial differential equations. Both of these approaches are attempts to add physical length and times scales to the closure relations of the equation set. In two phase flow, closure relations are used to account for physics that is not resolved by the discrete equations (similar to turbulence models).

In addition to work on the two-phase flow physics, new approaches have focused on improving the spatial accuracy (see [11–13] and references within) of the numerical models. The goal of this work is to replace the upwind differencing methods employed by TRAC-M and RELAP5 and is often based on knowledge of the characteristics of the partial differential equations [14]. These types of numerical advection schemes require the partial differential equations to be well posed (see [15,16] and references within) since the advection scheme is based on the eigenvalues of the equations. It should be noted that the two-phase flow equations that are the physical model in TRAC-M and RELAP5 are not well posed. Because of the numerical viscosity induced by the first order upwind advection scheme employed in the discretization of these equations, the discrete solution of these ill posed equations does not exhibit problems until very fine spatial nodalizations are used. Since the spatial discretizations employed in this paper are relatively coarse, the ill posed nature of the equations does not effect the results of this study.

This background is not meant to be exhaustive but simply a starting place for the reader to locate other work related to improving the accuracy of physics and spatial discretization employed by TRAC-M and RELAP5. However, in this paper the spatial discretization and the form of the equations will be identical to RELAP5. This paper will focus on the improved temporal accuracy obtained from implicitly balanced solutions employing second order in time nonlinear solvers.

Employing nonlinear solvers for the two-phase equations and related nonlinearly coupled physics has been a recent area of research by other authors (see [17–21]). In this paper the nonlinear solver will be the Jacobian-free Newton–Krylov (JFNK) method [22–24]. This method provides an accurate approach to couple nonlinear physics without incurring the error of operator splitting and linearizations [25]. Nonlinear solution techniques have been very successfully employed for solving steady state problems [18,19], but their computational cost have traditionally prevented them from being used for time accurate transients. The key to using the nonlinear solver for a time accurate transient is the hybrid technique that is referred to as ''physics-based'' preconditioning [26,27]. The key idea behind physics-based preconditioning is to recognize that a fast but inaccurate solution method can be coupled to an accurate solution method to provide a hybrid which is both fast and accurate.

As opposed to the other methods of increasing the implicitness of the solution of the two-phase flow equations [4,5], this approach uses the operator split semi-implicit (OSSI) method that is the basic solution algorithm of the current production two-phase system codes [1–3]. This traditional method, which has problems with stability and accuracy, is used to provide a solution which is close to the correct answer. Given a good estimate of the solution, the JFNK method, which is stable and accurate, quickly converges to the correct solution with only a small amount of computational work.

Using this hybrid approach has three main advantages. First, it takes advantage of the 20 years of investment to optimize the OSSI method. Second, if the solution algorithm contains both the OSSI and JFNK methods, either one can be employed where appropriate. For example, because of the high optimization of the OSSI method it is a better algorithm for scoping studies where accuracy is not critical. This fast method can be used to determine which simulations should be run with JFNK to obtain accurate solutions. In addition, there are some transients that are very fast and require small time steps to resolve the physical time scales. For these transients the OSSI method may provide adequate accuracy. For transients that require high accuracy (such as using a computer simulation to extend or fill-out an experimental data base) the JFNK method is an obvious choice. Also for transients that have very slow physical time scales (or steady state) the stability and accuracy of the JFNK method make it an obvious choice. The third and final advantage is that this hybrid approach provides a mechanism for coupling different physics [28–30] within the same simulation. Because the operator split approaches can be used as physics-based preconditioners, existing algorithms can be employed to provide preconditioners for implicitly balanced solutions of complicated nonlinearly coupled multi-physics, multi-length, and multi-time scale simulations. For example one could use an existing 3-D computational fluid dynamics (CFD) algorithm and an existing 1-D two-phase flow algorithm to provide a coupled solution that includes part of the domain in 1-D from one

algorithm and the rest of the domain in 3-D. These two algorithm could then be coupled by using the existing algorithms as physics-based preconditioners for an implicitly balanced solution for the two coupled systems. This is precisely what is done for coupling the 1-D fluid flow to the 2-D heat conduction in this paper.

The work in this paper is an extension of work previously done by other authors. Using Newton's method to solve a simplified form of the two-phase flow equations has been employed by Toumi et al. [17]. Downar and co-workers [20] employed a Jacobian-free Newton's method to solve a transient which couples a production two-phase flow code, TRAC-M [1], and a 3-D neutron kinetics code, PARCS [31]. Kastanya [18,19] used Newton's method to solve steady state problems with a simplified two-phase flow model coupled to neutron diffusion. Frepoli et al. [21] used Newton's method to solve a eight equation model for transient two-phase flow. The work in this paper differs from these previous works in three ways. First, a six equation model for two-phase flow is solved with an implicitly balanced solution. Second, this two-phase flow model is coupled to nonlinear heat conduction in a single nonlinear system to solve transient problems. Finally, the idea of using the OSSI solution method to precondition these equations to create a fast and accurate hybrid technique is new.

The rest of this paper has the following layout. Section 2 presents the physical model of the one-dimensional two-phase flow equations coupled to the two-dimensional nonlinear heat conduction equation. In Section 3, the discretized version of the physical model is presented which includes both the spatial and temporal differencing employed. Section 4 describes the Jacobian-free Newton–Krylov solution technique in terms of the discrete equations. In Section 5 the operator split semi-implicit algorithm is presented and it is shown how this algorithm can be employed both as a solution technique and as a physics-based pre-conditioner. Section 6 presents results for two test problems to demonstrate the ability of this hybrid approach and Section 7 presents conclusions and future work.

2. Physical model problem

The equations presented in this section are taken directly from the RELAP5 computer code manual [2,3]. Once again it is noted that these equations are known to be ill-posed [15,16] and no attempt has been made to address this problem in this study. The equation set is the standard one-dimensional, six-equation, single pressure model with coupling to a two-dimensional nonlinear heat conduction problem. As opposed to the simpler four equation model, in this equation set both phases have their own mass, momentum, and energy conservation equations. Notationally the subscripts "g", "f", "s", and "w" indicate variables associated with the gas (vapor) phase, the fluid (liquid) phase, a saturation value, and a variable associated with the solid wall, respectively. The first equation in this model is the conservation of volume.

$$\alpha_g + \alpha_f = 1 \tag{1}$$

In Eq. (1), α_g is the gas volume fraction and α_f is the fluid volume fraction. This equation is strictly enforced and in implementation the variable α_f is replaced with $1 - \alpha_g$. It is interesting to note that the constraint that $\alpha_g \in [0, 1]$ is not implemented directly in this equation set. In the JFNK method, this constraint has been added to the Newton update but in the OSSI method this constraint has to be enforced throughout the algorithm. Given the conservation of volume equation one can now write the conservation of mass equation for the gas phase,

$$\frac{\partial \alpha_g \rho_g}{\partial t} + \frac{\partial \alpha_g \rho_g v_g}{\partial x} = \Gamma_g, \tag{2}$$

and the conservation of mass in the fluid phase,

$$\frac{\partial \alpha_f \rho_f}{\partial t} + \frac{\partial \alpha_f \rho_f v_f}{\partial x} = -\Gamma_g. \tag{3}$$

In these equations ρ is density, v is velocity, and Γ_g is the interfacial mass transfer given by

$$\Gamma_g = -\frac{H_{ig} a_i (T_s - T_g) + H_{if} a_i (T_s - T_f)}{h_g^* - h_f^*}. \tag{4}$$

Here H_{ig} and H_{if} are the heat transfer coefficients between the separate phases and the interface (which is assumed to be in saturation). The variable a_i represents the interfacial area and T is temperature. The phasic mass transfer enthalpies (h_g^*, h_f^*) are calculated from the phasic enthalpies (h_g, h_f) and the phasic saturation enthalpies (h_{gs}, h_{fs}) by

$$h_g^* = \begin{cases} h_{gs} & \text{if } \Gamma_g > 0, \\ h_g & \text{otherwise,} \end{cases} \tag{5}$$

and

$$h_f^* = \begin{cases} h_f & \text{if } \Gamma_g > 0, \\ h_{fs} & \text{otherwise.} \end{cases} \tag{6}$$

It is important to note that mass transfer is equal in magnitude and opposite in sign between the gas and fluid phases and that these equations only apply to straight pipes like the one shown in Fig. 1. The equations in RELAP5 include terms that allow the area of the pipe to change as a function of the length.

In contrast to the work in [17–19] which employs a four equation two phase model that has a single mixture momentum equation, this physical model has a conservation of momentum equation for the gas phase,

$$\alpha_g \rho_g \frac{\partial v_g}{\partial t} + \alpha_g \rho_g v_g \frac{\partial v_g}{\partial x} + \alpha_g \frac{\partial P}{\partial x} - \alpha_g \rho_g \mathbf{g} = -F_{wg} a_{wg} (\alpha_g \rho_g)^2 |v_g| v_g - a_i \mathrm{FI} |v_g - v_f| (v_g - v_f) - \Gamma_g (v_i - v_g), \tag{7}$$

and a conservation of momentum equation for the fluid phase,

$$\alpha_f \rho_f \frac{\partial v_f}{\partial t} + \alpha_f \rho_f v_f \frac{\partial v_f}{\partial x} + \alpha_f \frac{\partial P}{\partial x} - \alpha_f \rho_f \mathbf{g} = -F_{wf} a_{wf} (\alpha_f \rho_f)^2 |v_f| v_f + a_i \mathrm{FI} |v_g - v_f| (v_g - v_f) + \Gamma_g (v_i - v_f). \tag{8}$$

Here P represents pressure and $\mathbf{g}$ is the gravity vector. The momentum losses due to wall and interfacial friction are given in terms of a phasic wall area (a_{wg}, a_{wf}), phasic wall friction coefficients (F_{wg}, F_{wf}), and an interfacial friction coefficient (FI). Note that the pressure gradient is partitioned between phases, the wall friction is a momentum sink, and the interfacial friction has equal magnitude and opposite sign between the two phases. The terms which include Γ_g account for the momentum lost or gained by the new mass appearing at the interfacial velocity v_i which is assumed to be the average of the two phasic velocities (this is simpler than how v_i is computed in RELAP5 [2,3]). Another notable difference with the RELAP5 momentum equations is the omission of the virtual mass terms.

In contrast to the four equation model which assumes thermal equilibrium between the two phases, this model has two energy equations. The first for conservation of energy in the gas phase,

$$\frac{\partial \alpha_g \rho_g U_g}{\partial t} + \frac{\partial \alpha_g \rho_g U_g v_g}{\partial x} + P \frac{\partial \alpha_g}{\partial t} + P \frac{\partial \alpha_g v_g}{\partial x} = H_{wg} a_{wg} (T_w - T_g) + H_{ig} a_i (T_s - T_g) + \Gamma_g h_g^*, \tag{9}$$

and the second for conservation of energy in the fluid phase,

$$\frac{\partial \alpha_f \rho_f U_f}{\partial t} + \frac{\partial \alpha_f \rho_f U_f v_f}{\partial x} + P\frac{\partial \alpha_f}{\partial t} + P\frac{\partial \alpha_f v_f}{\partial x} = H_{wf} a_{wf}(T_w - T_f) + H_{if} a_i (T_s - T_f) - \Gamma_g h_f^*. \tag{10}$$

Here, (U_g, U_f) are the phasic specific internal energies and (H_{wg}, H_{wf}) are the phasic wall heat transfer coefficients. The RELAP5 equations include an additional wall heat transfer term (that has been omitted here) that uses $(T_w - T_s)$ as the driving potential. The physical model of this study is closed out by including a two-dimensional nonlinear heat conduction equation to represent conservation of energy in the solid wall,

$$\frac{\partial e_w}{\partial t} - \frac{\partial}{\partial x}\kappa_x \frac{\partial T_w}{\partial x} - \frac{\partial}{\partial y}\kappa_y \frac{\partial T_w}{\partial y} = -\left[H_{wg} a_{wg}(T_w - T_g) + H_{wf} a_{wf}(T_w - T_f)\right] + Q_{nw}. \tag{11}$$

Here, (κ_x, κ_y) are the thermal conductivities in the x and y direction respectively, Q_{nw} is the source term on the side opposite the side of the wall contacting the fluid (see Fig. 1) which represents the energy imparted into the wall from nuclear fission. The variable e_w is the wall energy which is computed from

$$e_w = \int_{T_0}^{T_w} \rho_w C_p \, dT_w. \tag{12}$$

The variable C_p represents the specific heat of the wall. Note that in Eq. (11) the energy exchange with the fluid and vapor is equal in magnitude and opposite in sign between the wall and the two phasic energy equations. Additionally a_{wg} and a_{wf} have nonzero values only in cells that are adjacent to the fluid and Q_{nw} is nonzero only in the cells adjacent to the opposite edge from the fluid (see Fig. 1). For details on the equation of state for the fluid and solid see Appendix B. For the form of the nuclear fission energy source (Q_{nw}) see Appendix C.

3. Discrete equations

This section will present the discrete equations that are solved computationally. The discretization is a finite volume discretization on a staggered mesh. On this mesh the velocities are located at the cell faces, and the volume fractions, energies, and pressures are located at cell centers. The two-dimensional nonlinear solid wall heat conduction is also modeled with a finite volume discretization. This is different than the node based discretization employed by RELAP5 [2,3]. This choice of discretization has led to different boundary conditions than the RELAP5 code uses (i.e., the energy exchange on the boundaries is represented as source terms in the finite volume discretization whereas it is represented by fluxes in the node based discretization).

To write the finite volume form of the equations one has to define two volumes, V_f which is the volume of the fluid cell, and V_w which is the volume of the wall cell. These are given by

$$V_f = \Delta x \times \Delta y_f, \tag{13}$$

$$V_w = \Delta x \times \Delta y_w, \tag{14}$$

where Δy_f is the width of the fluid control volume and Δy_w is the width of the wall control volume.

The spatial discretization for the six two-phase flow equations matches that of RELAP5 and the nomenclature of a dot above a variable $(\dot{(\alpha_g \rho_g)}^n_{i+\frac{1}{2}})$ indicates that the variable is discretized with a first order upwind discretization. An over-bar on a variable $(\overline{\alpha_g^n \rho_g^n})$ indicates that the variable has been arithmetically averaged to either the cell center indicated by a subscript "i" or a cell face indicated by a subscript "$i+\frac{1}{2}$".

The temporal differencing is second order in time and is based on the Crank–Nicolson differencing scheme. The terms not in the time derivative are included as a weighted average (weighting variable is Θ) of the new time (superscript $n+1$) and old time (superscript n) variables. The choice of $\Theta = \frac{1}{2}$ yields the second order in time Crank–Nicolson method. Similarly $\Theta = 1$ yields implicit (backward Euler)

differencing and $\Theta = 0$ yields explicit (forward Euler) differencing. Further discussion of the higher order in time integration method will be addressed at the end of Section 4.

To simplify nomenclature the two continuity equations and the two energy equations have an implied subscript of i since they are cell centered equations. The two momentum equations have an implied subscript of $i + \frac{1}{2}$ since they are cell face equations. One can now write the residual of the conservation of mass equation for the gas phase (rescg) as,

$$\text{rescg} = \frac{V_\text{f}}{\Delta t}\left[\alpha_\text{g}^{n+1}\rho_\text{g}^{n+1} - \alpha_\text{g}^{n}\rho_\text{g}^{n}\right] + (1-\Theta)F_\text{cg}^{n} + \Theta F_\text{cg}^{n+1}, \tag{15}$$

where,

$$F_\text{cg}^{n+1} = \Delta y_\text{f}\left[(\alpha_\text{g}\dot{\rho}_\text{g})_{i+\frac{1}{2}}^{n+1} v_{\text{g},i+\frac{1}{2}}^{n+1} - (\alpha_\text{g}\dot{\rho}_\text{g})_{i-\frac{1}{2}}^{n+1} v_{\text{g},i-\frac{1}{2}}^{n+1}\right] - V_\text{f}\Gamma_\text{g}^{n+1}. \tag{16}$$

Similarly the residual of conservation of mass in the fluid phase (rescf) is given by

$$\text{rescf} = \frac{V_\text{f}}{\Delta t}\left[\alpha_\text{f}^{n+1}\rho_\text{f}^{n+1} - \alpha_\text{f}^{n}\rho_\text{f}^{n}\right] + (1-\Theta)F_\text{cf}^{n} + \Theta F_\text{cf}^{n+1}, \tag{17}$$

where,

$$F_\text{cf}^{n+1} = \Delta y_\text{f}\left[(\alpha_\text{f}\dot{\rho}_\text{f})_{i+\frac{1}{2}}^{n+1} v_{\text{f},i+\frac{1}{2}}^{n+1} - (\alpha_\text{f}\dot{\rho}_\text{f})_{i-\frac{1}{2}}^{n+1} v_{\text{f},i-\frac{1}{2}}^{n+1}\right] + V_\text{f}\Gamma_\text{g}^{n+1}. \tag{18}$$

The other two cell centered equations are the residual of conservation of energy in the gas phase (reseg) which is given by

$$\text{reseg} = \frac{V_\text{f}}{\Delta t}\left[\alpha_\text{g}^{n+1}\rho_\text{g}^{n+1}U_\text{g}^{n+1} - \alpha_\text{g}^{n}\rho_\text{g}^{n}U_\text{g}^{n}\right] + \frac{V_\text{f}P^{n+\frac{1}{2}}}{\Delta t}\left[\alpha_\text{g}^{n+1} - \alpha_\text{g}^{n}\right] + (1-\Theta)F_\text{eg}^{n} + \Theta F_\text{eg}^{n+1}, \tag{19}$$

where,

$$\begin{aligned} F_\text{eg}^{n+1} = {} & \Delta y_\text{f}\left[\left\{(\alpha_\text{g}\dot{\rho}_\text{g}U_\text{g})^{n+1}v_\text{g}^{n+1}\right\}_{i+\frac{1}{2}} - \left\{(\alpha_\text{g}\dot{\rho}_\text{g}U_\text{g})^{n+1}v_\text{g}^{n+1}\right\}_{i-\frac{1}{2}}\right] + \Delta y_\text{f}P^{n+1}\left[\left\{(\dot{\alpha}_\text{g})^{n+1}v_\text{g}^{n+1}\right\}_{i+\frac{1}{2}}\right. \\ & \left. - \left\{(\dot{\alpha}_\text{g})^{n+1}v_\text{g}^{n+1}\right\}_{i-\frac{1}{2}}\right] - V_\text{f}H_\text{wg}^{n+1}a_\text{wg}^{n+1}(T_\text{w}^{n+1} - T_\text{g}^{n+1}) - V_\text{f}H_\text{ig}^{n+1}a_\text{i}^{n+1}(T_\text{s}^{n+1} - T_\text{g}^{n+1}) - V_\text{f}\left(\Gamma_\text{g}h_\text{g}^{*}\right)^{n+1}. \end{aligned} \tag{20}$$

The discrete form for the residual of conservation of energy in the fluid phase (resef) is

$$\text{resef} = \frac{V_\text{f}}{\Delta t}\left[\alpha_\text{f}^{n+1}\rho_\text{f}^{n+1}U_\text{f}^{n+1} - \alpha_\text{f}^{n}\rho_\text{f}^{n}U_\text{f}^{n}\right] + \frac{V_\text{f}P^{n+\frac{1}{2}}}{\Delta t}\left[\alpha_\text{f}^{n+1} - \alpha_\text{f}^{n}\right] + (1-\Theta)F_\text{ef}^{n} + \Theta F_\text{ef}^{n+1}, \tag{21}$$

where,

$$\begin{aligned} F_\text{ef}^{n+1} = {} & \Delta y_\text{f}\left[\left\{(\alpha_\text{f}\dot{\rho}_\text{f}U_\text{f})^{n+1}v_\text{f}^{n+1}\right\}_{i+\frac{1}{2}} - \left\{(\alpha_\text{f}\dot{\rho}_\text{f}U_\text{f})^{n+1}v_\text{f}^{n+1}\right\}_{i-\frac{1}{2}}\right] + \Delta y_\text{f}P^{n+1}\left[\left\{(\dot{\alpha}_\text{f})^{n+1}v_\text{f}^{n+1}\right\}_{i+\frac{1}{2}}\right. \\ & \left. - \left\{(\dot{\alpha}_\text{f})^{n+1}v_\text{f}^{n+1}\right\}_{i-\frac{1}{2}}\right] - V_\text{f}H_\text{wf}^{n+1}a_\text{wf}^{n+1}(T_\text{w}^{n+1} - T_\text{f}^{n+1}) - V_\text{f}H_\text{if}^{n+1}a_\text{i}^{n+1}(T_\text{s}^{n+1} - T_\text{f}^{n+1}) + V_\text{f}\left(\Gamma_\text{g}h_\text{f}^{*}\right)^{n+1}. \end{aligned} \tag{22}$$

The two momentum equations are located at the cell face so their implied subscript is $i + \frac{1}{2}$. The residual for conservation of momentum in the gas phase (resmg) is

$$\text{resmg} = \frac{V_f(\overline{\alpha_g^{n+\frac{1}{2}}\rho_g^{n+\frac{1}{2}}})}{\Delta t}(v_g^{n+1} - v_g^n) + (1-\Theta)F_{mg}^n + \Theta F_{mg}^{n+1}, \tag{23}$$

where,

$$\begin{aligned} F_{mg}^{n+1} &= \Delta y_f(\overline{\alpha_g^{n+1}\rho_g^{n+1}})v_g^{n+1}(\dot{v_{g,i+1}}^{n+1} - \dot{v_{g,i}}^{n+1}) + \Delta y_f\overline{\alpha_g^{n+1}}\left(P_{i+1}^{n+1} - P_i^{n+1}\right) - V_f(\overline{\alpha_g^{n+1}\rho_g^{n+1}})\mathbf{g} \\ &\quad + V_f(F_{wg}^{n+1}a_{wg}^{n+1})(\overline{\alpha_g^{n+1}\rho_g^{n+1}})^2|v_g^{n+1}|v_g^{n+1} + V_f\overline{\Gamma_g^{n+1}}(v_i^{n+1} - v_g^{n+1}) \\ &\quad + V_f(\overline{a_i^{n+1}}\mathrm{FI}^{n+1})|v_g^{n+1} - v_f^{n+1}|(v_g^{n+1} - v_f^{n+1}). \end{aligned} \tag{24}$$

The discrete form of the residual of conservation of momentum in the fluid phase (resmf) is

$$\text{resmf} = \frac{V_f(\overline{\alpha_f^{n+\frac{1}{2}}\rho_f^{n+\frac{1}{2}}})}{\Delta t}(v_f^{n+1} - v_f^n) + (1-\Theta)F_{mf}^n + \Theta F_{mf}^{n+1}, \tag{25}$$

where

$$\begin{aligned} F_{mf}^{n+1} &= \Delta y_f(\overline{\alpha_f^{n+1}\rho_f^{n+1}})v_f^{n+1}(\dot{v_{f,i+1}}^{n+1} - \dot{v_{f,i}}^{n+1}) + \Delta y_f\overline{\alpha_f^{n+1}}\left(P_{i+1}^{n+1} - P_i^{n+1}\right) - V_f(\overline{\alpha_f^{n+1}\rho_f^{n+1}})\mathbf{g} \\ &\quad + V_f(F_{wf}^{n+1}a_{wf}^{n+1})(\overline{\alpha_f^{n+1}\rho_f^{n+1}})^2|v_f^{n+1}|v_f^{n+1} - V_f\overline{\Gamma_g^{n+1}}(v_i^{n+1} - v_f^{n+1}) \\ &\quad - V_f(\overline{a_i^{n+1}}\mathrm{FI}^{n+1})|v_g^{n+1} - v_f^{n+1}|(v_g^{n+1} - v_f^{n+1}). \end{aligned} \tag{26}$$

The six one-dimensional equations that describe the two-phase fluid flow are now complete. The last equation is the two-dimensional nonlinear heat conduction which describes the conservation of energy in the wall. In this equation the two implied subscripts are "i" and "j". The residual for the conservation of energy in the wall (resew) is given in discrete form by

$$\text{resew} = \frac{V_w}{\Delta t}\left[e_w^{n+1} - e_w^n\right] + (1-\Theta)F_{ew}^n + \Theta F_{ew}^{n+1}, \tag{27}$$

where,

$$\begin{aligned} F_{ew}^{n+1} &= -\frac{\Delta y_w}{\Delta x}\left\{\kappa_{x,i+\frac{1}{2}}^{n+1}\left[T_{w,i+1}^{n+1} - T_{w,i}^{n+1}\right] - \kappa_{x,i-\frac{1}{2}}^{n+1}\left[T_{w,i}^{n+1} - T_{w,i-1}^{n+1}\right]\right\} - \frac{\Delta x}{\Delta y_w}\left\{\kappa_{y,j+\frac{1}{2}}^{n+1}\left[T_{w,j+1}^{n+1} - T_{w,j}^{n+1}\right]\right. \\ &\quad \left. - \kappa_{y,j-\frac{1}{2}}^{n+1}\left[T_{w,j}^{n+1} - T_{w,j-1}^{n+1}\right]\right\} + V_w H_{wg}^{n+1}a_{wg}^{n+1}(T_{w,1,j}^{n+1} - T_{g,j}^{n+1}) + V_w H_{wf}^{n+1}a_{wf}^{n+1}(T_{w,1,j}^{n+1} - T_{f,j}^{n+1}) - V_w Q_{nw}^{n+1}. \end{aligned} \tag{28}$$

It should be noted that the areas (a_{wg} and a_{wf}) are zero except when the wall control volume is adjacent to the fluid and the source term Q_{nw} is only nonzero on the edge of the wall on the opposite side of the fluid (see Fig. 1).

If we define nx to be the number of control volumes in the x direction and ny to be the number of control volumes in the y direction then the total number of unknowns N for the discrete equations are

$$N = (6 \times nx) + (nx \times ny). \tag{29}$$

This gives a nonlinear system of N equations and N unknowns where nx copies of Eqs. (15), (17), (19), (21), (23), and (25) are solved for $\alpha_{g,i}, P_i, U_{g,i}, U_{f,i}, v_{g,i+\frac{1}{2}}, v_{f,i+\frac{1}{2}}$ for $i \in [1,2,\ldots,nx]$ and $nx \times ny$ copies of Eq. (27) are solved for $T_{w,i,j}$ for $i \in [1,2,\ldots,nx]$ and $j \in [1,2,\ldots,ny]$.

4. Jacobian-free Newton–Krylov

Given the description of the nonlinear system in the previous section, the solution of these residual equations will be presented here. A modified form of Newton's method called physics-based preconditioned Jacobian-free Newton–Krylov is used to solve the discrete nonlinear equations. Newton's method is designed to solve nonlinear systems of the form,

$$\mathbf{res}(\mathbf{x}) = 0, \tag{30}$$

where $\mathbf{res} = [\mathrm{rescg}, \mathrm{rescf}, \mathrm{reseg}, \mathrm{resef}, \mathrm{resmg}, \mathrm{resmf}, \mathrm{resew}]^{\mathrm{T}}$ (Eqs. (15), (17), (19), (21), (23), (25), and (27)) and $\mathbf{x} = [\alpha_{\mathrm{g}}, P, U_{\mathrm{g}}, U_{\mathrm{f}}, v_{\mathrm{g}}, v_{\mathrm{f}}, T_{\mathrm{w}}]^{\mathrm{T}}$. Newton's method solves this nonlinear system iteratively by solving a sequence of linear problems defined by

$$\mathbf{J}^k \, \delta\mathbf{x}^k = -\mathbf{res}(\mathbf{x}^k). \tag{31}$$

The matrix $\mathbf{J}$ is the Jacobian matrix, the superscript k is the Newton iteration and $\delta\mathbf{x}^k$ is the update vector. The (i, j) element of the Jacobian matrix is the derivative of the ith equation with respect to the jth variable or in equation form,

$$\mathbf{J}_{i,j} = \frac{\partial \mathbf{res}_i}{\partial \mathbf{x}_j}. \tag{32}$$

Eq. (31) is solved for $\delta\mathbf{x}^k$ and the new Newton iteration value for $\mathbf{x}$ is then computed from,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + d \, \delta\mathbf{x}^k. \tag{33}$$

The damping parameter, $d \in [0, 1]$, is computed to keep the components of $\mathbf{x}$ in physically realizable space (i.e., $\alpha_{\mathrm{g}} \in [0, 1]$, $P > 0$, $U_{\mathrm{g}} > 0$, $U_{\mathrm{f}} > 0$, and $T_{\mathrm{w}} > 0$). Note that the same damping value is applied to all of the updates. The damping of the update vector can be interpreted geometrically as preserving the update vectors direction but shortening its length. This damping of the Newton update vector $\delta\mathbf{x}$ increases the robustness of Newton's method. An additional constraint that was not included in this work is that $U_{\mathrm{g}} \geqslant U_{\mathrm{f}}$. This constraint becomes more important for applications near the critical point of water where the gas and fluid state become equivalent.

This iteration on $\mathbf{x}$ is continued until the nonlinear residual $\mathbf{res}$ is small. Here small is defined by

$$\|\mathbf{res}(\mathbf{x}^k)\|_2 < \mathrm{tol}\|\mathbf{res}(\mathbf{x}^0)\|_2, \tag{34}$$

where $\mathrm{tol} = 1.0 \times 10^{-8}$ is the nonlinear tolerance.

If the Jacobian matrix $\mathbf{J}$ (Eq. (32)) is computed analytically, and the linear system (Eq. (31)) is solved exactly, then this is simply the traditional Newton's method for the solution of a nonlinear system of equations. In the remainder of this section, modifications to the basic Newton algorithm to improve its efficiency and storage are presented.

The first modification is referred to as an Inexact Newton's method [32]. The word inexact refers to the accuracy of the solution of the linear system not the accuracy of the nonlinear iteration. The nonlinear tolerance given by Eq. (34) is identical for both the exact and inexact implementations. The basic idea behind an inexact Newton's method is only to solve the linear system to a tight tolerance when the added accuracy effects the convergence of the Newton iteration. This is accomplished by making the convergence of the linear residual proportional to the nonlinear residual or in equation form,

$$\|\mathbf{J}^k \, \delta\mathbf{x}_m^k + \mathbf{res}(\mathbf{x}^k)\|_2 < \eta\|\mathbf{res}(\mathbf{x}^k)\|_2. \tag{35}$$

In these equations, the subscript m refers to the "mth" iteration of the linear solver and $\eta = 1.0 \times 10^{-3}$ is a linear tolerance.

The linear solver used in this study is the Arnoldi based GMRES [33] iterative Krylov solver. The Krylov solver constructs the "mth" iteration from,

$$\delta\mathbf{x}_m^k = \delta\mathbf{x}_0^k + a_0\mathbf{r}_0 + a_1\mathbf{J}\mathbf{r}_0 + a_2\mathbf{J}^2\mathbf{r}_0 + \cdots + a_m\mathbf{J}^m\mathbf{r}_0, \tag{36}$$

where $\mathbf{r}_0 = \mathbf{J}^k\,\delta\mathbf{x}_0^k + \mathbf{res}(\mathbf{x}^k)$ and $\delta\mathbf{x}_0^k$ is the initial guess at the linear solution. It is important to note that the Jacobian matrix only shows up as the product of the Jacobian matrix and a vector in Eq. (36). Therefore, if the action of the Jacobian matrix can be approximated, the Jacobian matrix itself is never required for the Krylov solution. Fortunately, the action of the Jacobian matrix [34] can be approximated by

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{res}(\mathbf{x} + \epsilon\mathbf{v}) - \mathbf{res}(\mathbf{x})}{\epsilon}, \tag{37}$$

where,

$$\epsilon = \frac{\sum_{i=1}^{N} bx_i}{N\|\mathbf{v}\|_2}, \tag{38}$$

and $b = 1.0 \times 10^{-8}$, N is the number of unknowns, and $\mathbf{v}$ is a Krylov vector (i.e. $\mathbf{v} \in (\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, \mathbf{J}^2\mathbf{r}_0, \ldots)$). If one employs the Jacobian-free approximation (Eq. (37)), then the work associated with forming the Jacobian matrix (Eq. (32)) and the storage for the Jacobian matrix can be eliminated. If the number of Krylov iterations is small (see [35] for a precise definition of what is meant by small) this approximation saves both CPU time and storage for the nonlinear iteration. The optimization of the linear iteration will be addressed next.

Because GMRES stores all of the previous Krylov vectors, it is necessary to keep the number of iterations small to prevent the storage and CPU time from becoming prohibitive. This minimization of the Krylov iterations can be accomplished by right preconditioning the linear system,

$$\mathbf{J}^k\mathbf{P}^{-1}\mathbf{P}\,\delta\mathbf{x}^k = -\mathbf{res}(\mathbf{x}^k), \tag{39}$$

where $\mathbf{P}^{-1}$ approximates $\mathbf{J}^{-1}$. The traditional approach to preconditioning is to construct $\mathbf{P} \approx \mathbf{J}$ and then approximately compute $\mathbf{P}^{-1}$ in a fast manner. The physics-based preconditioning approach is based on the observation that $\mathbf{J}^{-1}$ is really a linearized time step and therefore $\mathbf{P}^{-1}$ can be any time step algorithm. The time advancement procedure employed in this study as the physics-based preconditioner is the Operator Split, Semi-Implicit (OSSI) solution method employed by RELAP5.

It is important to note that there is no dependency between the Crank–Nicolson time integration and the JFNK method. The JFNK method only depends on the evaluation of the residuals. Other higher order in time temporal integration methods (such as, Runge–Kutta, Adams–Bashford, and Backward difference formulas) could just as easily been incorporated into the residuals to achieve higher temporal accuracy.

5. Using an OSSI solution as a preconditioner

This section is divided into two parts; first an overview of using physics-based preconditioning and second a description of the specifics of using the OSSI method as a physics-based preconditioner.

5.1. Overview

To demonstrate the OSSI solver as a preconditioner, one needs to write the OSSI solution method in the same form as Eq. (31),

$$\mathbf{P}\,\delta\mathbf{x} = -\mathbf{res}. \tag{40}$$

Written as a solver, $\mathbf{P}^{-1}$ is a linear operator which maps a residual vector into an update vector,

$$\delta\mathbf{x} = -\mathbf{P}^{-1}\mathbf{res}. \tag{41}$$

Traditionally, the symbol $\mathbf{P}^{-1}$ represents the inverse of the matrix $\mathbf{P}$. In this paper this nomenclature has been extended to include any linear operator that computes an update vector, $\delta\mathbf{x}$, from a residual, **res** (details are given in Appendix A). Following is a discussion of the implementation of the physics-based preconditioner into the JFNK solution.

In the implementation of right preconditioning (Eq. (39)) a new vector ($\mathbf{y}$) is defined by

$$\mathbf{y}^k = \mathbf{P}\,\delta\mathbf{x}^k. \tag{42}$$

Using Eq. (42), the preconditioned system is given by

$$\mathbf{J}^k\mathbf{P}^{-1}\mathbf{y}^k = -\mathbf{res}(\mathbf{x}^k). \tag{43}$$

One solves Eq. (43) for $\mathbf{y}^k$ and then computes $\delta\mathbf{x}^k$ from,

$$\delta\mathbf{x}^k = \mathbf{P}^{-1}\mathbf{y}^k. \tag{44}$$

It is important to make a few observations here. First, when using preconditioning, one never needs to define the linear operator $\mathbf{P}$, since it is never required. Second, in the OSSI solution, a residual is given and an update vector is computed by the linear process $\mathbf{P}^{-1}$ (see Eq. (41)). When the OSSI method is employed as a preconditioner, it is given a Krylov vector $[\mathbf{v} \in (\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, \mathbf{J}^2\mathbf{r}_0, \ldots)]$ and it computes the preconditioned vector by applying the linear process which represents the OSSI time advancement, $\mathbf{P}^{-1}$. Therefore, the identical software that computes a time step advancement when OSSI is used as a solver is also used to implement OSSI as a preconditioner. The only change is when OSSI is a solver the right-hand-side is a residual vector and when OSSI is a preconditioner the right-hand-side is a Krylov vector.

There are additional details associated with the Jacobian-free implementation. The preconditioned version of the Jacobian-free approximation (Eq. (37)) is computed from,

(1) Given a vector $\mathbf{y}$, compute $\mathbf{z} = \mathbf{P}^{-1}\mathbf{y}$.
(2) From $\mathbf{z}$ then compute $\mathbf{J}\mathbf{z}$ using

$$\mathbf{J}\mathbf{z} \approx \frac{\mathbf{res}(\mathbf{x} + \epsilon\mathbf{z}) - \mathbf{res}(\mathbf{x})}{\epsilon}. \tag{45}$$

Therefore one can solve the right preconditioned system,

$$\mathbf{J}^k\mathbf{P}^{-1}\mathbf{P}\,\delta\mathbf{x}^k = -\mathbf{res}(\mathbf{x}^k), \tag{46}$$

without defining $\mathbf{P}$ (but knowing how to compute the action of $\mathbf{P}^{-1}$ on a vector) and without computing the Jacobian matrix, $\mathbf{J}$ (but knowing how to compute the action of $\mathbf{J}$ on a vector).

Therefore, given an existing time stepping algorithm which computes an update from a residual, the fully coupled nonlinear system can be solved using the physics-based preconditioned Jacobian-free Newton–

Krylov method without any knowledge of the Jacobian matrix but simply knowing how to compute the residuals of the equations.

This approach should be contrasted to the more traditional preconditioning approach where the matrices **J** and **P** are computed element by element. These two very large matrices are then passed to a linear solver package which computes the update. The formation and storage of **J** and **P** have made implementation of Newton's method difficult for these types of equations sets. But, now when one employs the physics-based Jacobian-free Newton–Krylov method, the only storage that is required (besides storing the Krylov vectors required by GMRES) is the storage of the small matrices used in the operator split semi-implicit solution.

Additionally since the OSSI method is used as a preconditioner, it does not effect the accuracy or stability of the JFNK method. The preconditioner simply effects the amount of CPU time to achieve the levels of accuracy set by the input numbers tol in Eq. (34) and η in Eq. (35). The stability is insured by the fact that all variables have a new time $(n+1)$ contribution in the solution (Eqs. (15), (17), (19), (21), (23), (25), and (27)) as long as $\Theta \geqslant \frac{1}{2}$.

5.2. Equations

In this section the equations that are solved in the OSSI method are presented. These equations include a variety of linearizations and operator splitting that allow the 1-D two-phase flow equations and the 2-D nonlinear heat conduction equations to be solved by only inverting tridiagonal matrices. The residuals in these equations are the same as in the JFNK method but here they are evaluated with old time (time level n) variables. It should be noted that when the time level $n+1$ terms in the residuals (Eqs. (15), (17), (19), (21), (23), (25), and (27)) are replaced with time level n terms, the time derivative contributions cancel out and the results are identical for any value of $\Theta \in [0,1]$. This evaluation of the residuals with old time variables is signified in the following equations as a superscript n on the residual (e.g., rescg^n).

From the linearized equation of state, given in Appendix B, the updates for temperature and density used in the following equations are given by

$$\delta T_s = \left(\frac{\partial T}{\partial P}\right)_0 \delta P, \tag{47}$$

$$\delta T_g = \left(\frac{\partial T}{\partial P}\right)_0 \delta P + \left(\frac{\partial T_g}{\partial U_g}\right)_0 \delta U_g, \tag{48}$$

$$\delta T_f = \left(\frac{\partial T}{\partial P}\right)_0 \delta P + \left(\frac{\partial T_f}{\partial U_f}\right)_0 \delta U_f, \tag{49}$$

$$\delta \rho_g = \left(\frac{\partial \rho_g}{\partial P}\right)_0 \delta P + \left(\frac{\partial \rho_g}{\partial U_g}\right)_0 \delta U_g, \tag{50}$$

$$\delta \rho_f = \left(\frac{\partial \rho_f}{\partial P}\right)_0 \delta P + \left(\frac{\partial \rho_f}{\partial U_f}\right)_0 \delta U_f. \tag{51}$$

It is important to note that because of the linearized equation of state employed for this study, the above approximations are exact. In a more realistic equation of state these linearizations employed by the OSSI solution method are partially responsible for loss or gain in mass of the system.

The linearization of the equation of state is one of the sources of mass error in the OSSI solution algorithm. Another source, which can be significantly larger, is the enforcement of the constraint that $\alpha_g \in [0,1]$. In the implicitly balanced solution this constraint is enforced by the choice of d in Eq. (33). In the OSSI solution, enforcement of this constraint causes mass to be added or removed from the simulation. The standard strategy is to add or remove the mass if it is "small" but to lower the time step if it is large. The danger of this strategy is if the error is "small" but it is always adding (or removing) mass for many time steps. The result of summing this error over a large number of time steps can lead to catastrophic mass errors that are on the same order as the total mass in the system being modeled.

Because the implicitly balanced solution uses the real equation of state (not the linearized form in Eqs. (47)–(51)) there is no mass error associated with this approximation employed by the OSSI solution. In addition, because the damping parameter, d, in Eq. (33) is chosen to keep $\alpha_g \in [0,1]$ this second source of mass error in the OSSI solution has also been removed.

Another source of errors in the OSSI algorithm comes from the evaluation of the heat transfer and friction coefficients at time level n. Because these coefficients are constant in this study (instead of a highly nonlinear function of all of the state variables which is physically more realistic) the error associated with evaluating the coefficients with old time state values is zero. In the more realistic case of nonlinear heat transfer and friction coefficients this error can be very large.

The linearized equations solved in the OSSI solution are presented below. Recall that the conservation of mass and energy are cell centered equations (subscript "i"), the momentum equations are cell face equations (subscript "$i+\frac{1}{2}$"), and the wall energy equation is two-dimensional cell centered (subscript "i,j"). As before, the implied subscripts are ignored for clarity of presentation.

The semi-implicit conservation of mass equations in the gas and fluid phase are

$$\frac{V_f}{\Delta t}\left[\alpha_g^n \delta\rho_g + \rho_g^n \delta\alpha_g\right] + \Delta y_f\left[(\alpha_g\dot{\rho}_g)^n_{i+\frac{1}{2}}\delta v_{g,i+\frac{1}{2}} - (\alpha_g\dot{\rho}_g)^n_{i-\frac{1}{2}}\delta v_{g,i-\frac{1}{2}}\right] + \left(\frac{V_f}{h_g^* - h_f^*}\right)^n\left[H_{ig}^n a_i^n(\delta T_s - \delta T_g) + H_{if}^n a_i^n(\delta T_s - \delta T_f)\right] = -\mathrm{rescg}^n, \tag{52}$$

and

$$\frac{V_f}{\Delta t}\left[\alpha_f^n \delta\rho_f - \rho_f^n \delta\alpha_g\right] + \Delta y_f\left[(\alpha_f\dot{\rho}_f)^n_{i+\frac{1}{2}}\delta v_{f,i+\frac{1}{2}} - (\alpha_f\dot{\rho}_f)^n_{i-\frac{1}{2}}\delta v_{f,i-\frac{1}{2}}\right] - \left(\frac{V_f}{h_g^* - h_f^*}\right)^n\left[H_{ig}^n a_i^n(\delta T_s - \delta T_g) + H_{if}^n a_i^n(\delta T_s - \delta T_f)\right] = -\mathrm{rescf}^n. \tag{53}$$

The update terms (δ's) are implicit, and all other terms are explicit. For stability the velocities in the advection terms are implicit. It is important to note that the interphase mass exchange terms have also been made implicit since the condensation and vaporization rates induce fast time scales into the equations. From an accuracy point of view it is important to note that the coefficients of the phase change terms are evaluated at time level "n".

The semi-implicit conservation of momentum equations for the gas and fluid phase are given by

$$\frac{V_f(\overline{\alpha_g\rho_g})^n}{\Delta t}\delta v_g + \Delta y_f\overline{\alpha_g}^n(\delta P_{i+1} - \delta P_i) + V_f(F_{wg}a_{wg}|v_g|)^n\left[\overline{(\alpha_g\rho_g)}^n\right]^2\delta v_g + V_f\overline{\Gamma}_g^n(\delta v_i - \delta v_g) + V_f(\overline{a_i}\mathrm{FI})^n|v_g^n - v_f^n|(\delta v_g - \delta v_f) = -\mathrm{resmg}^n, \tag{54}$$

and

$$\frac{V_f(\overline{\alpha_f\rho_f})^n}{\Delta t}\delta v_f + \Delta y_f \overline{\alpha_f}^n(\delta P_{i+1} - \delta P_i) + V_f(F_{wf}a_{wf}|v_f|)^n\left[\overline{(\alpha_f\rho_f)}^n\right]^2\delta v_f - V_f\overline{\Gamma}^n_g(\delta v_i - \delta v_f) + V_f(\overline{a_i FI})^n|v^n_g - v^n_f|(\delta v_g - \delta v_f) = -\text{resmf}^n. \quad (55)$$

In these equations it is important to note that the advection and buoyancy terms are purely explicit so their entire contribution comes from the residuals resmg^n and resmf^n. The pressure is implicit to allow the semi-implicit stability time step to be larger than the sound speed Courant stability limit. In these momentum equations, the frictional terms have been made implicit since they introduce fast time scales into the solution. But again, their coefficients are at old time.

The semi-implicit conservation of energy equations in the gas and fluid phase are

$$\frac{V_f}{\Delta t}\left[\alpha^n_g U^n_g\delta\rho_g + (\rho^n_g U^n_g + P^n)\delta\alpha_g + \alpha^n_g\rho^n_g\delta U_g\right] + \Delta y_f\left[\left\{(\alpha_g\rho_g\dot{U}_g) + P_i(\dot{\alpha_g})\right\}^n_{i+\frac{1}{2}}\delta v_{g,i+\frac{1}{2}} - \left\{(\alpha_g\rho_g\dot{U}_g) + P_i(\dot{\alpha_g})\right\}^n_{i-\frac{1}{2}}\delta v_{g,i-\frac{1}{2}}\right] - V_f H^n_{ig}a^n_i(\delta T_s - \delta T_g) + \left(\frac{V_f h^*_g}{h^*_g - h^*_f}\right)^n\left[H^n_{ig}a^n_i(\delta T_s - \delta T_g) + H^n_{if}a^n_i(\delta T_s - \delta T_f)\right] = -\text{reseg}^n, \quad (56)$$

and

$$\frac{V_f}{\Delta t}\left[\alpha^n_f U^n_f\delta\rho_f - (\rho^n_f U^n_f + P^n)\delta\alpha_g + \alpha^n_f\rho^n_f\delta U_f\right] + \Delta y_f\left[\left\{(\alpha_f\rho_f\dot{U}_f) + P_i(\dot{\alpha_f})\right\}^n_{i+\frac{1}{2}}\delta v_{f,i+\frac{1}{2}} - \left\{(\alpha_f\rho_f\dot{U}_f) + P_i(\dot{\alpha_f})\right\}^n_{i-\frac{1}{2}}\delta v_{f,i-\frac{1}{2}}\right] - V_f H^n_{if}a^n_i(\delta T_s - \delta T_f) - \left(\frac{V_f h^*_f}{h^*_g - h^*_f}\right)^n\left[H^n_{ig}a^n_i(\delta T_s - \delta T_g) + H^n_{if}a^n_i(\delta T_s - \delta T_f)\right] = -\text{resef}^n. \quad (57)$$

In these equations the velocity terms in the advection of energy are implicit for stability and again the energy exchange between equations has been made implicit. It should be noted that the energy exchange with the wall is explicit and is therefore included in the residuals reseg and resef. It should be noted that it is possible in the OSSI solution to make the wall-fluid energy exchange implicit [2,3] but all results presented in this paper for the OSSI solution use explicit wall heat transfer to the fluid.

To describe the wall heat transfer solution used in the OSSI solution, two new variables are defined,

$$\delta T'_w = T^{n+1/2} - T^n, \quad (58)$$

$$\delta T''_w = T^{n+1} - T^{n+1/2}, \quad (59)$$

from which on can write,

$$\delta T_w = \delta T'_w + \delta T''_w = T^{n+1} - T^n. \quad (60)$$

With these new definitions the heat conduction equation in the wall now follows.

$$\frac{V_w\rho_w C^n_p}{\Delta t/2}(\delta T') - \frac{\Delta y_w}{\Delta x}\left[\kappa^n_{x,i+1/2}(\delta T'_{i+1} - \delta T') - \kappa^n_{x,i-1/2}(\delta T' - \delta T'_{i-1})\right] + V_w H^n_{wg}a^n_{wg}(\delta T'_{1,j}) + V_w H^n_{wf}a^n_{wf}(\delta T'_{1,j}) = -\text{resew}^n \quad (61)$$

$$\frac{V_w \rho_w C_p^n}{\Delta t/2}(\delta T'') - \frac{\Delta x}{\Delta y_w}\left[\kappa^n_{y,j+1/2}(\delta T''_{j+1} - \delta T'') - \kappa^n_{y,j-1/2}(\delta T'' - \delta T''_{j-1})\right] + V_w H^n_{wg} a^n_{wg}(\delta T''_{1,j}) + V_w H^n_{wf} a^n_{wf}(\delta T''_{1,j}) = -\text{resew}^{n+\frac{1}{2}} \quad (62)$$

In the OSSI solution, Eq. (61) is solved for $\delta T'$, and then the half time step temperatures are computed from $T^{n+\frac{1}{2}} = T^n + \delta T'$. The half time residual ($\text{resew}^{n+\frac{1}{2}}$) is computed from $T^{n+\frac{1}{2}}$ and then Eq. (62) is solved for $\delta T''$. Finally the total update is computed from $\delta T = \delta T' + \delta T''$. It should be noted that in RELAP5 equations (61) and (62) are solved for $T^{n+1/2}$ and T^{n+1} instead of $\delta T'$ and $\delta T''$.

Because the dominant flow direction of energy is in the y direction, the physics-based preconditioner only solves Eq. (62) for a full time step instead of a half time step. The details that describe how these equations are solved in the OSSI method are presented in Appendix A.

6. Results

This section presents results from two idealized representations of nuclear reactor transients. The first transient is a simplified model of a reactor SCRAM. In a SCRAM, the control rods are quickly inserted into the reactor to slow down the nuclear fission reaction. The second transient will be a simplified version of a rod ejection. In a rod ejection accident, the reactor is at low power and one control rod is removed bringing the power level up.

The input variables that describe the geometry, the state of the fluid, and the state of the pipe are given in Appendices B and C. Therefore the results section will only include the information that is unique to the transient being studied.

Three versions of each transient will be presented. The difference between the transients will be the wall thickness. The three transients will include a one inch, a two inch, and a four inch wall. The power input, and the fluid conditions will be the same in each of these versions. This means that all of the fast time scales associated with condensation, vaporization, and interfacial and wall friction are the same in all three versions. By modifying the thickness of the wall (note that $\Delta y_w = 2.5 \times 10^{-3}$ m is a constant and $ny = 10$ for the $1''$ version, $ny = 20$ in the $2''$ version, and $ny = 40$ in the $4''$ version) the slow time scale, associated with the wall's gain or loss of energy, is made longer as the wall is made thicker.

6.1. SCRAM

Results for the three versions of the SCRAM problem are presented in this section. In a real SCRAM the reactor power is lowered over the time that it takes for the control rods to be inserted into the reactor core. When the control rods are inserted, the reactor still produces a small amount of energy due to the decay heat of the fuel. Since this simplified model does not contain neutron kinetics, this transient will be modeled by instantaneously setting the power input on the right side of the wall to zero. When the power addition is removed the wall begins to cool.

To compress the two-dimensional temperature data from the wall into a single point that can be plotted as a function of time, a new variable will be defined (which is important to nuclear reactor safety) which will be called the peak clad temperature. The peak clad temperature will be defined as the maximum temperature of the wall cells that are adjacent to the fluid (the first column of cells on the left side of the wall in Fig. 1). Fig. 2 presents the peak clad temperature for all three versions of the SCRAM problem (see Table 1). The transients are stopped ($t = t_f$) when the peak clad temperature falls within 20% of the no power temperature of 590 K. In this figure one sees that the time scale of the transient slows as the wall is made thicker.

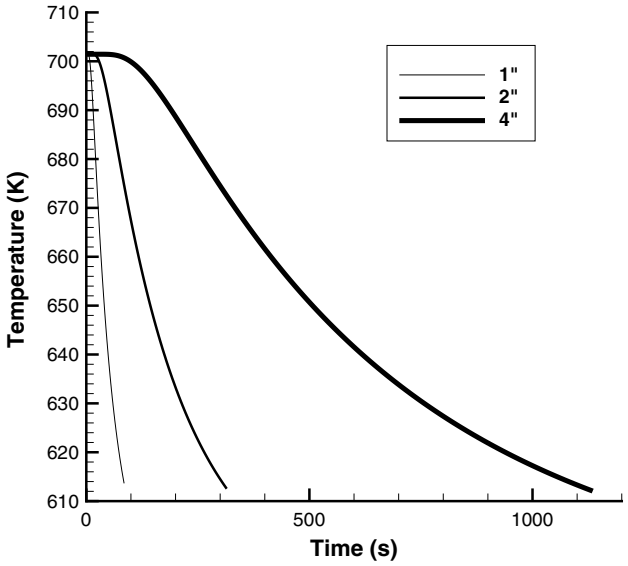

Fig. 2. Peak clad temperature vs. time for SCRAM.

Table 1
SCRAM input

Wall	nx	ny	L_x (m)	L_y (m)	t_f (s)
$1''$	40	10	5.0	0.025	85
$2''$	40	20	5.0	0.050	315
$4''$	40	40	5.0	0.100	1135

Before presenting results about accuracy a short discussion of error needs to occur. Because of the complexity of this nonlinearly coupled system of equations, there is no exact solution for the test problems. For this paper the "exact" solution will be a second-order in time Newton–Krylov solution run at a time step ten times smaller than the smallest data presented on the plot. From this "exact" solution the vapor velocity is extracted. The vapor velocity was chosen since it is one of the most sensitive variables. The "error" is then computed from the following equation.

$$\text{error} = \sqrt{\sum_{i=1}^{nx} (v_{g,i} - v_{g,i}^{e})^2}, \tag{63}$$

here v_g^e is the velocity from the "exact" solution.

The variable that will be used to represent time discretization in the following plots will be the CFL number. If one defines

$$v_{\mathbf{max}} = \max(v_{\mathbf{g},i}, v_{\mathbf{f},i}) : \quad i \in [1, 2, \ldots, nx], \tag{64}$$

then the CFL number is given by

$$\text{CFL} = \frac{v_{\mathbf{max}} \Delta t}{\Delta x}. \tag{65}$$

Fig. 3 presents a time convergence study for all three versions of the SCRAM problem for both the OSSI method and the JFNK method. The OSSI data is represented by circles and the larger the circle the thicker the wall. The JFNK data is represented by squares and the larger the square the thicker the wall. To see plots of the OSSI and JFNK solutions converging to the same solution under time step convergence see Fig. 8 and Fig. 9 in Appendix D.

There are three observations to be made about Fig. 3. The first deals with the three OSSI data points that have an error two orders of magnitude larger than any of the other data. This large error is caused by oscillations in the solution that are discussed further in Appendix D. The second is that the error for the OSSI data is significantly larger than for the JFNK method. For the OSSI solution to obtain the same level of error as the JFNK method running at time steps well above the material Courant stability limit, the OSSI method has to take time steps with CFL numbers approximately 1.0×10^{-2}. Finally as the wall thickness increases, the accuracy of the JFNK method increases while the accuracy of the OSSI method appears independent of the wall thickness. All three observations agree with the analysis presented in [25] which indicates that the implicitly balanced solution (JFNK) has truncation error terms that are related to the dynamical (slow) time scale of the problem where the OSSI solution has truncation error terms associated with the normal mode (fast) time scales. This test problem was constructed to have identical normal mode time scales (condensation, vaporization, friction) but a large variance in the dynamical (wall energy loss) time scales.

Similar to Fig. 3, Fig. 4 has error on the "y" axis, but the "x" axis is now CPU time instead of CFL number. This plot which compares error and CPU time will be referred to as an "efficacy" plot. This plot is an efficient way to compare two different algorithms. A horizontal line on this plot determines which method is fastest for a given level of accuracy. A vertical line on this plot indicates which method is more accurate for a given amount of CPU time.

This plot shows that for the one inch wall simulation, which has a "fast" dynamical time scale, the OSSI method can provide a faster solution which is less accurate. For scoping studies this lower level of accuracy may not be as important, so the OSSI method may be a very good choice. However, for the four inch problem with a slow dynamical time scale, the JFNK method provides a more accurate solution for the

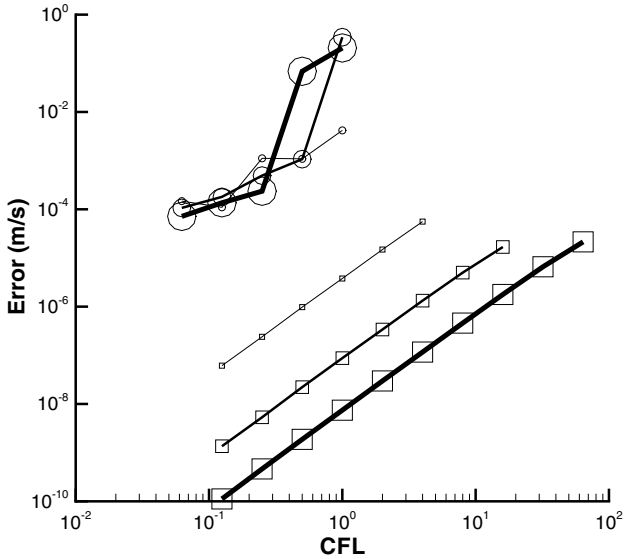

Fig. 3. Error vs. CFL for SCRAM: circles OSSI, squares JFNK.

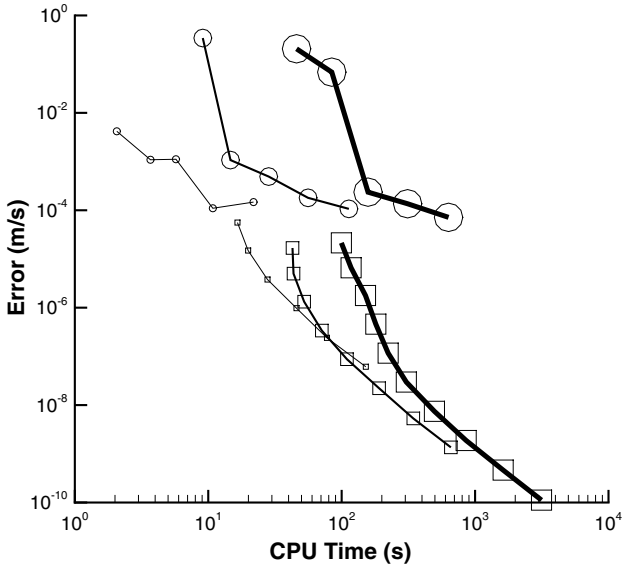

Fig. 4. Error vs. CPU time for SCRAM: circles OSSI, squares JFNK.

same amount of CPU time compared to any OSSI data point with an acceptable (non-oscillatory) solution. For problems with slow dynamical time scales the implicitly balanced solution is clearly the better choice.

There are three observations to be made based on these two algorithms and the data presented so far. The first is that because of the linearized equation of state and the constant friction and heat transfer coefficients the error in the OSSI method has been minimized. Since the friction and heat transfer coefficients are constant (instead of nonlinear functions of the state variables), the time lagging of these coefficients from time level $n+1$ to time level n does not introduce any error. Also, the OSSI solution method requires that the densities and temperatures be represented as linear functions of pressure and energies. Since the equation of state is linear, there is no error in representing density and temperature as a linear function of pressure and energy. A real equation of state or nonlinear friction and heat transfer coefficients would increase the error of the OSSI method. Second, since the same fluid flow algorithm is used both as the OSSI solver and the physics-based preconditioner (note the physics-based preconditioner uses 1-D heat conduction instead of the 2-D heat conduction in the OSSI solver) the optimization of this algorithm has a positive impact on reducing the CPU time of both solution methods. The final observation is that since the physics-based preconditioned implicitly balanced solution contains both solution methods, one can switch between the OSSI and JFNK solutions based on the required level of accuracy and the dynamical time scale of the problem.

6.2. Rod ejection

In a rod ejection transient the reactor power level has been brought down due to the insertion of the control rods into the nuclear reactor core. The transient is initiated by one of the control rods being "ejected" from the reactor core. When this occurs, the fission reaction rate increases in the area where the control rod has been removed. This results in a very rapid rise in the power level of the reactor. For this test problem it will be assumed that there is only one control rod in the reactor and when it is removed, the

reactor comes up to full power. Again, since there is no neutron kinetics model in the simulation, the power level increase will be modeled as an instantaneous increase at time zero.

Fig. 5 shows the peak clad temperature as a function of time for the three versions of the rod ejection problem (see Table 2). For these test problems the transient is terminated when the peak clad temperature reaches approximately 80% of its maximum value of 700 K. Due to the symmetry of the two test problems these time scales are almost identical to the SCRAM test problem (see Table 1).

Fig. 6 presents a time convergence plot for the rod ejection test problem. This plot is very similar to the SCRAM problem in Fig. 3. Again, the implicitly balanced solution has a decrease in error as the dynamical time scale of the problem is increased.

In Fig. 7 the efficacy plot is similar to that in Fig. 4. Here one can see that the JFNK method provides a more accurate solution than the OSSI method. For the 4″ wall problem, with a slow dynamical time scale, the implicitly balanced method provides a highly accurate solution for the same amount of CPU time as the OSSI method.

6.3. Preconditioner effectiveness

This section will focus on the symbiotic relationship between the implicitly balanced solution algorithm and the physics-based preconditioner which employs the more traditional OSSI method. The OSSI physics-based preconditioner provides a very fast estimate of the problem solution. The implicitly balanced JFNK

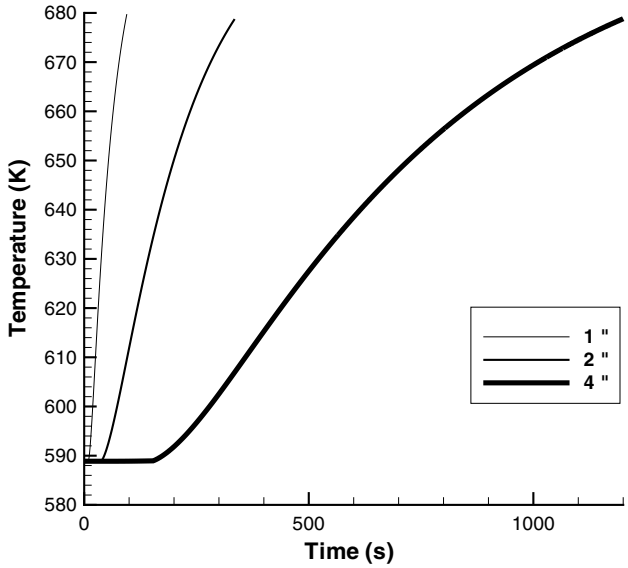

Fig. 5. Peak clad temperature vs. time for rod ejection.

Table 2
Rod ejection input

Wall	*nx*	*ny*	L_x (m)	L_y (m)	t_f (s)
1″	40	10	5.0	0.025	95
2″	40	20	5.0	0.050	335
4″	40	40	5.0	0.100	1205

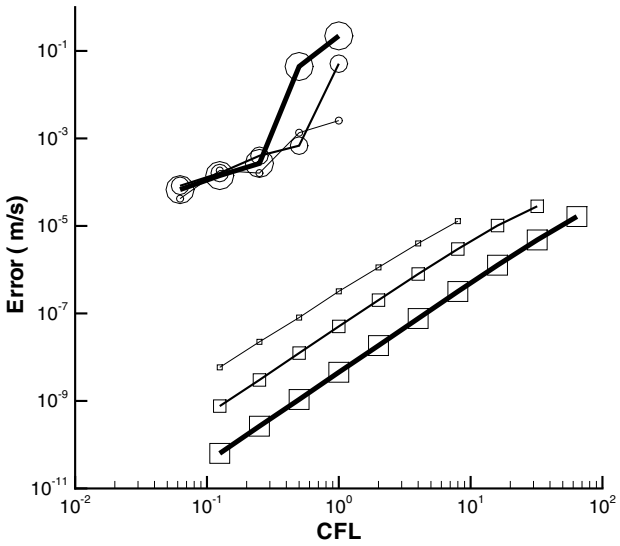

Fig. 6. Error vs. CFL for rod ejection: circles OSSI, squares JFNK.

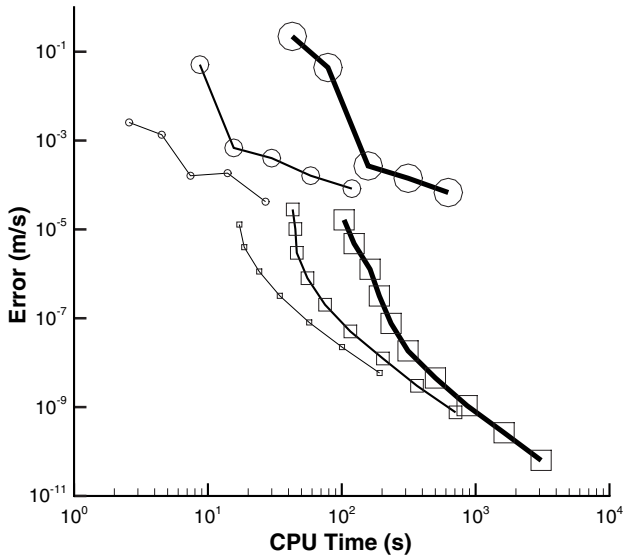

Fig. 7. Error vs. CPU time for rod ejection: circles OSSI, squares JFNK.

method then takes this approximate solution and eliminates the errors associated with the fast time scales providing a high level of accuracy. The combination of the two algorithms results in a fast and accurate solution method.

To precondition a large linear system, one needs to cluster the eigenvalues of the matrix and to improve the condition number of the matrix by lowering the spread in the size of the eigenvalues. The key to physics-based preconditioning is to recognize that the eigenvalues of the matrix are related to physical time scales in the simulation. The fast time scales (related to the small eigenvalues) are associated with interphase mass,

momentum, and energy transfer as well as energy and momentum transfer and energy transfer with the wall. It is important to note that except for energy transfer with the wall, the OSSI method has all of the other fast time scales implicit. This implicitness in the fast time scales in the physics-based preconditioner seems to reduce the number of Krylov iterations in a very efficient manner.

Another important advantage of the physics-based preconditioner is that the linear systems in the preconditioner are significantly smaller than the Jacobian matrix. The Jacobian matrix for this model has $nx(ny+6)$ unknowns. The physics-based preconditioner preconditions this system by solving nx systems of size ny for the heat conduction and one system of size nx for the two-phase fluid flow (see Appendix A for details). In addition, all of these preconditioner matrices are tridiagonal which means that they can be solved exactly in three times the number of unknowns worth of work. Thus the cost for preconditioning the Jacobian matrix is $O(3nx(ny+1))$. For comparison, a direct solve of the Jacobian matrix would be $O([nx(ny+6)]^3)$. Table 3 shows the discretization and the number of unknowns for the different variations of wall thickness.

Table 4 shows the effect of the preconditioner on the solution algorithm for the 1″ SCRAM problem. The first data presented is the average number of Krylov iterations per time step $(\frac{\text{GMRES}}{\Delta t})$ which represents the amount of linear solver work required to solve the nonlinear system per time step. The second data is the maximum number of Krylov vectors that are needed to be stored by the GMRES solver (Max GMRES). The maximum number of Krylov vectors that need to be stored is the maximum number of Krylov iterations per Newton iteration over all of the Newton iterations and time steps in the simulation. It should be noted that the size of a Krylov vector is equal to the total number of unknowns in the problem. It should also be noted that the GMRES algorithm can be restarted to minimize the amount of Krylov vectors that need to be stored but this modification of the GMRES algorithm has a negative impact on its convergence rate [36]. The third data presented is the number of seconds of computer time for the simulation (CPU). These three pieces of data are presented for both the preconditioned and unpreconditioned solutions for three different CFL numbers.

Table 5 presents the raw data of Table 4 in a compressed form where the unpreconditioned numbers are divided by the preconditioned numbers. Table 5 shows that for a variety of CFL numbers the preconditioner lowers the number of Krylov iterations by roughly a factor of 45. This reduction of a factor of 45 only results in a CPU savings of a factor of 15, meaning that the construction and solution of the preconditioner costs about three times as much as a single Krylov iteration. One obvious way to reduce this preconditioning cost, that has not yet been implemented, would be to form the preconditioner once per Newton iteration and store the result. The current implementation forms the preconditioning matrix on

Table 3
Problem size for different wall thicknesses

Wall	nx	ny	1-D unknowns	2-D unknowns	Total unknowns
1″	40	10	240	400	640
2″	40	20	240	800	1040
4″	40	40	240	1600	1840

Table 4
1″ SCRAM linear solution

CFL	Preconditioned			No preconditioning		
	$\frac{\text{GMRES}}{\Delta t}$	Max GMRES	CPU (s)	$\frac{\text{GMRES}}{\Delta t}$	Max GMRES	CPU (s)
4	19	17	17	845	214	270
1	16	8	28	780	171	474
$\frac{1}{4}$	13	6	78	623	160	1218

Table 5
1″ SCRAM ratios

CFL	Ratio of no preconditioning to preconditioning		
	$\frac{\text{GMRES}}{\Delta t}$	Max GMRES	CPU (s)
4	45	13	16
1	48	21	17
$\frac{1}{4}$	49	27	16

Table 6
2″ SCRAM linear solution

CFL	Preconditioned			No Preconditioning		
	$\frac{\text{GMRES}}{\Delta t}$	Max GMRES	CPU (s)	$\frac{\text{GMRES}}{\Delta t}$	Max GMRES	CPU (s)
4	28	17	52	1012	212	938
1	17	9	109	879	173	2368
$\frac{1}{4}$	13	7	334	625	159	6350

Table 7
2″ SCRAM ratios

CFL	Ratio of no preconditioning to preconditioning		
	$\frac{\text{GMRES}}{\Delta t}$	Max GMRES	CPU (s)
4	36	12	18
1	52	19	22
$\frac{1}{4}$	49	23	18

every Krylov iteration. Table 5 also shows that the storage required for the Krylov solution is lowered by more than an order of magnitude when the preconditioner is used. It is important to note that the total storage required for the preconditioner linear solution is $3 \times nx$ (assuming that $ny \leqslant nx$). This should be compared to the storage of the Krylov linear solver which is (Max GMRES $\times\, nx(ny + 6)$).

Table 6 presents the same raw data for the 2″ SCRAM problem and Table 7 presents the compressed data for the same problem. Table 5 and Table 7 can be summarized in the following three bullets.

- The average number Krylov iteration per time step is reduced by a factor of 46.
- Employing the physics-based preconditioner lowers the average amount of storage required by the GMRES algorithm by a factor of 19.
- The average computer time per simulation is reduced by a factor of 18.

7. Conclusions and future work

Results have been presented for variations of two simulations which show that the implicitly balanced solution, obtained by employing a physics-based preconditioned version of the Jacobian-free Newton–Krylov algorithm, provides a significantly more accurate solution to the two-phase flow equations coupled to nonlinear heat conduction than the traditional approach based on operator splitting the conduction from the two-phase flow and then linearizing these split solutions. Efficacy results are presented that show for some versions of the test problems, the implicitly balanced solution can achieve this higher level of accuracy for the same amount of computer time. Finally, data is presented that indicates that the same level

of error can be obtained using larger time steps with the implicitly balanced solution when the dynamical time scale of the solution is slower.

Future work should include a detailed eigen analysis of the physics-based preconditioner to determine its effect on the eigen spectrum of the Jacobian matrix. Work should also be done to compare a traditional Newton's method, where the Jacobian matrix is formed and the preconditioner is based on an approximate inversion of the Jacobian matrix, to the OSSI and JFNK methods. By comparing to a traditional Newton method, one could determine how much the JFNK improvements over the OSSI method are from the physics-based preconditioner and the Jacobian-free implementation and how much of the improvements are from an implicitly balanced solution.

Acknowledgements

The author would like to thank Dana Knoll, John Reisner, Richard Riemke, Rex Shumway, and Glen Mortensen for their valuable feedback that greatly improved this paper. This work was carried out under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. W-7405-ENG-36 (LA-UR-03-8238).

Appendix A. Semi-implicit algorithm

This appendix will describe how the equations presented in Section 5 are solved in the Operator Split, Semi-Implicit (OSSI) algorithm. This same software that implements the OSSI solution procedure is used for both the physics-based preconditioner and the OSSI solution method. The equations are solved by the following eight steps.

(1) The first step in the OSSI algorithm is the operator splitting of the two-dimensional heat conduction. The heat conduction equations (61) and (62) are solved using a tridiagonal solution procedure. Note that the equations are tridiagonal because only the horizontal terms are implicit in Eq. (61) and only the vertical terms are implicit in Eq. (62). Eq. (61) is solved for $\delta T'$ and Eq. (62) is solved for $\delta T''$. The new time wall temperature to be used in the fluid equations is then computed from,

$$T^{n+1} = T^n + \delta T' + \delta T''. \tag{A.1}$$

It is important to note that the heat exchange term with the fluid is computed using the old time fluid temperatures, thus contributing an explicit component to the energy coupling between the fluid and the wall. The consequences of this choice will be demonstrated in Appendix D.

(2) The second step is to solve the momentum equations for their dependence on pressure (see Table 8). Note an "X" in Table 8 indicates a non-zero coefficient of the variable in the column head. To accomplish this one solves the two momentum equations (54) and (55) simultaneously for $\delta v_{g,i+1/2}$ and $\delta v_{f,i+1/2}$ in terms of δP_{i+1}, δP_i, $\text{resmg}^n_{i+1/2}$, and $\text{resmf}^n_{i+1/2}$.

(3) In the third step these two momentum equations and the five linearized state equations (47)–(51) are substituted into the two continuity equations (52) and (53) and two energy equations (56) and (57)

Table 8
Momentum 2×2

	$\delta v_{g,i+1/2}$	$\delta v_{f,i+1/2}$	δP_i	δP_{i+1}	r.h.s.
Momentum gas	X	X	X	X	$-\text{resmg}_{i+1/2}$
Momentum fluid	X	X	X	X	$-\text{resmf}_{i+1/2}$

Table 9
Pressure 4×4

	$\delta U_{g,i}$	$\delta U_{f,i}$	$\delta \alpha_{g,i}$	δP_{i-1}	δP_i	δP_{i+1}	r.h.s.
Energy gas	X	X	X	X	X	X	$-\text{reseg}_i + \text{resm}$
Energy fluid	X	X	X	X	X	X	$-\text{resef}_i + \text{resm}$
Mass gas	X	X	X	X	X	X	$-\text{rescg}_i + \text{resm}$
Mass fluid	X	X	X	X	X	X	$-\text{rescf}_i + \text{resm}$

to eliminate $\delta v_{g,i+1/2}$, $\delta v_{g,i-1/2}$, $\delta v_{f,i+1/2}$, $\delta v_{f,i-1/2}$, $\delta T_{s,i}$, $\delta T_{g,i}$, $\delta T_{f,i}$, $\delta \rho_{g,i}$, and $\delta \rho_{f,i}$. This results in four equations in terms of six unknowns ($\delta U_{g,i}$, $\delta U_{f,i}$, $\delta \alpha_{g,i}$, δP_{i-1}, δP_i, and δP_{i+1}) and eight residuals ($\text{resmg}^n_{i+1/2}$, $\text{resmg}^n_{i-1/2}$, $\text{resmf}^n_{i+1/2}$, $\text{resmf}^n_{i-1/2}$, rescg^n_i, rescf^n_i, reseg^n_i, and resef^n_i). To simplify nomenclature resm will be defined by

$$\text{resm} = \text{resmg}_{i+1/2} + \text{resmg}_{i-1/2} + \text{resmf}_{i+1/2} + \text{resmf}_{i-1/2}. \tag{A.2}$$

The results of these substitutions are represented in graphical form in Table 9.

(4) These four equations are row reduced by Gaussian elimination into upper triangular form. This results in a tridiagonal matrix for δP in terms of the eight residuals.
(5) This tridiagonal system is then solved for δP.
(6) These δP's are then back substituted into the block 4×4's to solve for $\delta \alpha_g$, δU_f, and δU_g.
(7) The δP's are then back substituted into the block 2×2's to solve for δv_g, and δv_f.
(8) The new time variables are then solved for by adding the δ's to the old time values and that is the end of a time step.

Appendix B. Equation of state

The specific heat at constant pressure is computed from

$$C_p = C_{pT_0} T_w + C_{p0}, \tag{B.1}$$

where $C_{pT_0} = 0.14$ J/kg and $C_{p0} = 470$ J/kg K. The thermal conductivities are computed from,

$$\kappa_x = \kappa_y = \kappa_{T_0} T_w + \kappa_0, \tag{B.2}$$

where $\kappa_{T_0} = 1.36 \times 10^{-2}$ W/m and $\kappa_0 = 11.5$ W/m K. These linearizations where computed from data in Incropera and Dewitt [37] for Stainless Steal AISI 304 (Table A.1 on page A5).

The following are the linearized equations of state employed in this study.

$$T_s = T_0 + \left(\frac{\partial T}{\partial P}\right)_0 (P - P_0), \tag{B.3}$$

$$h_{gs} = h_{gs0} + \left(\frac{\partial h_{gs}}{\partial P}\right)_0 (P - P_0), \tag{B.4}$$

$$h_{fs} = h_{fs0} + \left(\frac{\partial h_{fs}}{\partial P}\right)_0 (P - P_0), \tag{B.5}$$

$$T_g = T_0 + \left(\frac{\partial T}{\partial P}\right)_0 (P - P_0) + \left(\frac{\partial T_g}{\partial U_g}\right)_0 (U_g - U_{g0}), \tag{B.6}$$

$$T_f = T_0 + \left(\frac{\partial T}{\partial P}\right)_0 (P - P_0) + \left(\frac{\partial T_f}{\partial U_f}\right)_0 (U_f - U_{f0}), \tag{B.7}$$

$$\rho_g = \rho_{g0} + \left(\frac{\partial \rho_g}{\partial P}\right)_0 (P - P_0) + \left(\frac{\partial \rho_g}{\partial U_g}\right)_0 (U_g - U_{g0}), \tag{B.8}$$

$$\rho_f = \rho_{f0} + \left(\frac{\partial \rho_f}{\partial P}\right)_0 (P - P_0) + \left(\frac{\partial \rho_f}{\partial U_f}\right)_0 (U_f - U_{f0}). \tag{B.9}$$

These linearizations were derived from data in Van Wylen and Sonntag [38] (Table A.1.2 on page 618) around a point on the water-steam saturation line that corresponded to a pressure of 1.0×10^7 Pa.

The phasic enthalpies are computed from,

$$h_g = U_g + \frac{P}{\rho_g}, \tag{B.10}$$

$$h_f = U_f + \frac{P}{\rho_f}. \tag{B.11}$$

The linearization points and the derivatives are presented in Table 10.

Table 10
Equation of state values

Variable	Value
T_0	584.21 K
h_{gs0}	2.7247×10^6 J/kg
h_{fs0}	1.40756×10^6 J/kg
$\left(\frac{\partial T_g}{\partial U_g}\right)_0$	2.112×10^{-4} kg K/J
P_0	1.0×10^7 Pa
U_{g0}	2.544×10^6 J/kg
ρ_{g0}	5.547×10^1 kg/m^3
$\left(\frac{\partial \rho_g}{\partial U_g}\right)_0$	-7.766×10^{-5} kg^2/m^3 J
$\left(\frac{\partial \rho_f}{\partial P}\right)_0$	1.290×10^{-6} kg/m^3 Pa
$\left(\frac{\partial T}{\partial P}\right)_0$	7.375×10^{-6} K/Pa
$\left(\frac{\partial h_{gs}}{\partial P}\right)_0$	-1.825×10^{-2} J/kg Pa
$\left(\frac{\partial h_{fs}}{\partial P}\right)_0$	4.342×10^{-2} J/kg Pa
$\left(\frac{\partial T_f}{\partial U_f}\right)_0$	1.877×10^{-4} kg K/J
$\left(\frac{\partial \rho_g}{\partial P}\right)_0$	7.376×10^{-6} kg/m^3 Pa
U_{f0}	1.393×10^6 J/kg
ρ_{f0}	6.887×10^2 kg/m^3
$\left(\frac{\partial \rho_f}{\partial U_f}\right)_0$	-3.755×10^{-4} kg^2/m^3 J

Table 11
Input values

Variable	Value
nx	40
Δy_f	0.1963 m
$\mathbf{g}$	9.8 m/s^2
v_{g0}	0.1 m/s
H_{ig}	1.0×10^5 W/m^5 K
P_0	1.0×10^7 Pa
U_{g0}	2.544×10^6 J/kg
F_{wg}	1.0 kg^{-1}
H_{wg}	4.0×10^5 W/m^5 K
a_{wg}	1.0 m^2
a_i	1.0 m^2
tol	1.0×10^{-8}
b	1.0×10^{-8}
Δx	0.125 m
Δy_w	2.5×10^{-3} m
α_{g0}	0.5
v_{f0}	0.1 m/s
H_{if}	2.0×10^6 W/m^5 K
FI	1.0×10^6 kg/m^6
U_{f0}	1.393×10^6 J/kg
F_{wf}	100.0 kg^{-1}
H_{wf}	4.0×10^6 W/m^5 K
a_{wf}	1.0 m^2
ρ_w	7900.0 kg/m^3
η	1.0×10^{-3}

Appendix C. Inputs

The input values that are the same for both simulations are given in Table 11.
The shape of the energy source that represents the nuclear fissions is given by

$$Q_{nw}(x) = 5.0 \times 10^8 \cos(\pi(x - 2.5)/5) \text{ W/m}^4. \tag{C.1}$$

Appendix D. OSSI oscillations

In Fig. 3 of Section 6 it was noted that there were three data points that had errors that appeared to be about two orders of magnitude larger than was expected. This appendix will explain the cause of this discrepancy.

Fig. 8 shows the gas velocity as a function of distance down the pipe for four different CFL numbers and two integration schemes. The first observation is that on the scale of this plot the two lines that represent the JFNK solution at a CFL number of 64 (NK2 64), and the OSSI solution at a CFL number of 0.25 (SI 0.25) lay on top of each other. The other two lines (SI 1 and SI 0.5) represent the OSSI solution at CFL numbers of 1.0 and 0.5, respectively. Both of these CFL numbers have very large errors in Fig. 3. From Fig. 8 it is clear that there is an oscillation in the solution at a CFL number of 0.5 and that this oscillation is larger for the CFL number of 1.0.

Fig. 9 shows the same data as Fig. 8. This plot is an error plot where the "exact velocity" (NK2 1.25e−2) has been subtracted from the other four data sets. It should be noted that error is plotted on a logarithmic

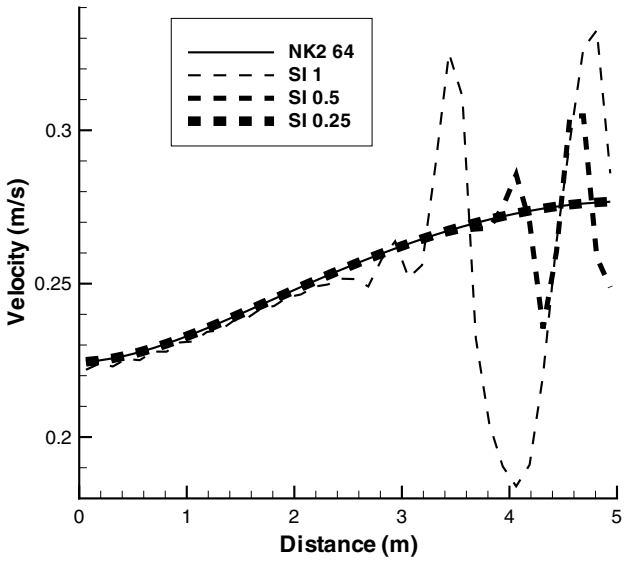

Fig. 8. Velocity vs. distance for 4″ thick SCRAM.

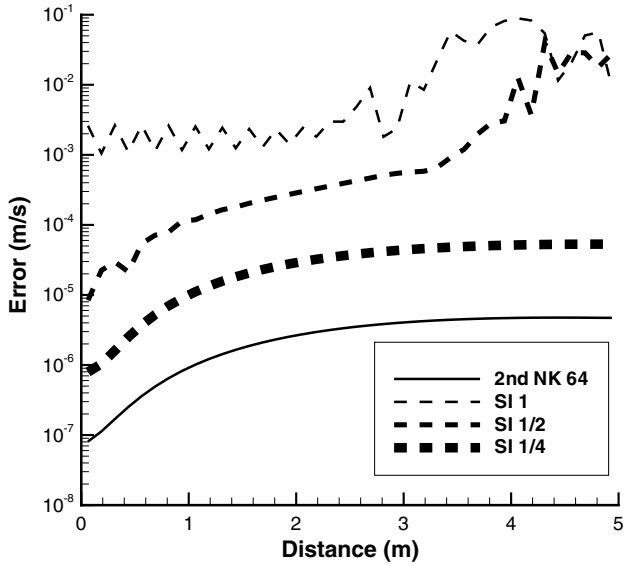

Fig. 9. Velocity error vs. distance for 4″ thick SCRAM.

axis. In this plot one can see that for the OSSI solution with a CFL number of 1.0 the error is large across the entire domain. For the CFL number of 0.5 the OSSI solution has a much smaller error for lower values of x (the pipe inflow) and larger errors for higher values of x (the pipe outflow). Once the CFL number drops to 0.25 the error is significantly smaller than for a CFL number of 0.5 but it is interesting to note that the JFNK solution is much more accurate even at a large CFL number of 64.

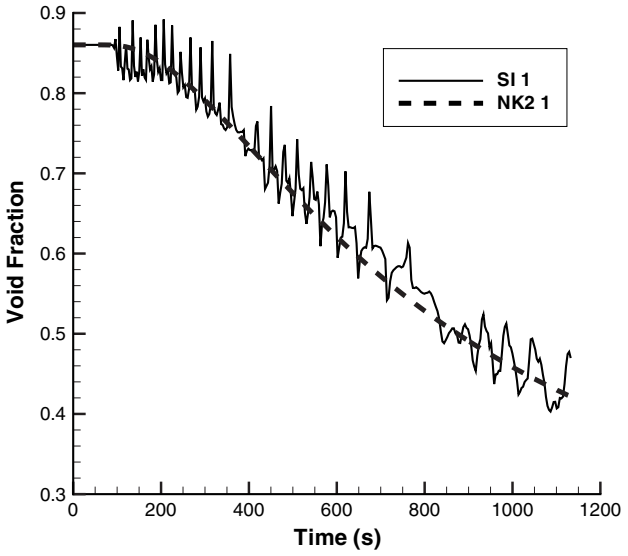

Fig. 10. Void fraction vs. time for 4″ thick SCRAM CFL = 1 semi-implicit.

To show the source of the error, the void fraction will be plotted at every tenth time step for the control volume closest to the outflow end ($x = 5$). In Fig. 10 one can see that the OSSI solution is oscillating (plotting more data points reveals that this oscillation frequency is on every time step). The JFNK solution is completely smooth at the same time step level. It appears that the oscillation in void fraction on every time step is caused by the fact that the phasic temperatures (T_f, T_g) in the fluid wall transfer terms in the liquid, gas, and wall energy equations (see Eqs. (56) and (57)) are explicit. This explicit temperature dependence causes the void fraction to oscillate since the energy transfer from the wall is first being over predicted, and the being under predicted on the next time step. This oscillation in void fraction then feeds throughout the rest of the solution.

References

[1] US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-001, TRAC-M/FORTRAN 90 (Version 3.0) Theory Manual, NUREG/CR-6724 Edition (April 2001).
[2] US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-001, RELAP5/MOD3.3 Code Manual volume I: Code Structure, System Models, and Solution Methods, NUREG/CR-5535 Edition (December 2001).
[3] Idaho National Engineering and Environmental Laboratory, RELAP5-3D Code Manual volume I: Code Structure, System Models, and Solution Methods, INEEL-EXT-98-00834 Rev. 2.0 Edition (July 2002).
[4] J. Mahaffy, A stability-enhancing two-step method for fluid flow calculations, J. Comput. Phys. 46 (1982) 329–341.
[5] J. Trapp, R. Riemke, A nearly-implicit hydrodynamic numerical scheme for two-phase flows, J. Comput. Phys. 66 (1986) 62–82.
[6] F. Harlow, A. Amsden, A numerical fluid dynamics calculation method for all flow speeds, J. Comput. Phys. 8 (1971) 197–231.
[7] J. Trapp, G. Mortensen, A discrete particle model for bubble-slug two-phase flows, J. Comput. Phys. 107 (1993) 367–377.
[8] H. Pokharna, M. Mori, V. Ransom, The particle fluid model and using Lagrangian representation in two-phase flow modeling, Nucl. Eng. Design 175 (1997) 59–69.
[9] M. Ishii, S. Kim, J. Uhle, Interfacial area transport equation: model development and benchmark experiments, Int. J. Heat Mass Transfer 45 (2002) 3111–3123.
[10] T. Hibiki, M. Ishii, Development of one-group interfacial area transport equation in bubbly flow systems, Int. J. Heat Mass Transfer 45 (2002) 2351–2372.
[11] I. Toumi, A. Kumbaro, An approximate linearized Riemann solver for a two-fluid model, J. Comput. Phys. 124 (1996) 286–300.

[12] I. Tiselj, S. Petelin, Modeling of two-phase flow with second-order accurate scheme, J. Comput. Phys. 136 (1997) 503–521.
[13] R. Nourgaliev, N. Dinh, T. Theofanous, A characteristics-based approach to the numerical solution of the two-fluid model, in: Proceedings of FEDSM03 4th ASME/JSME Joint Fluids engineering conference, 2003.
[14] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, Berlin, 1999.
[15] V. Ransom, D.L. Hicks, Hyperbolic two-pressure models for two-phase flow, J. Comput. Phys. 53 (1984) 124–151.
[16] T. Dinh, R. Nourgaliev, T. Theofanous, Understanding the ill-posed two-fluid model, in: The 10th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-10), 2003.
[17] I. Toumi, A. Bergeron, D. Gallo, E. Royer, D. Caruge, FLICA-4: a three-dimensional two-phase flow computer code with advanced numerical methods for nuclear applications, Nucl. Eng. Design 200 (2000) 139–155.
[18] D. Kastanya, Implementation of a Newton–Krylov iterative method to address strong non-linear feedback effects in FORMOSA-B BWR core simulator, Ph.D. thesis, North Carolina State University, 2002.
[19] D. Kastanya, P. Turinsky, Implementation of a Newton-BICGSTAB solver to treat the strong non-linearity in the FORMOSA-B boiling water reactor core simulator code, in: Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew, American Nuclear Society, 2003.
[20] J. Gan, Y. Xu, T. Downar, A matrix-free Newton method for coupled neutronics thermal-hydraulics reactor analysis, in: Nuclear mathematical and Computational Sciences: A century in Review, a century anew, American Nuclear Society, 2003.
[21] C. Frepoli, J. Mahaffy, K. Ohkawa, Notes on the implementation of a fully-implicit numerical scheme for a two-phase three-field flow model, Nucl. Eng. Design 225 (2003) 191–217.
[22] D. Knoll, D. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[23] T.F. Chan, K.R. Jackson, Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms, SIAM J. Sci. Stat. Comput. 5 (1984) 533–542.
[24] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Stat. Comput. 11 (1990) 450–481.
[25] D. Knoll, L. Chacon, L. Margolin, V. Mousseau, On balanced approximations for time integration of multiple time scale systems, J. Comput. Phys. 185 (2003) 583–611.
[26] D. Knoll, V. Mousseau, L. Chacon, J. Reisner, Jacobian-free Newton–Krylov methods for accurate time integration of stiff wave systems, Journal of Scientific Computing, in press.
[27] V. Mousseau, D. Knoll, J. Reisner, An implicit nonlinearly consistent method for the two-dimensional shallow-water equations with Coriolis force, Monthly Weather Rev. 130 (2002) 2611–2625.
[28] W. Weaver, E. Tomlinson, D. Aumiller, A generic semi-implicit coupling methodology for use in RELAP5-3D, Nucl. Eng. Design 211 (2002) 13–26.
[29] D. Aumiller, E. Tomlinson, R. Bauer, A coupled RELAP-3D/CFD methodology with a proof-of-principle calculation, Nucl. Eng. Design 205 (2001) 83–90.
[30] J. Gan, T. Downar, J. Mahaffy, J. Uhle, Parallel applications of the USNRC consolidated code, Proceedings of the Society of Photo-Optical Instrumentation Engineers 4528 (2001) 77–86.
[31] H. Joo, D. Barber, G. Jiang, T. Downar, PARCS: A multi-dimensional two-group reactor kinetics code based on the nonlinear analytic nodal method, Tech. Rep. PU/NE-98-26, Purdue University, 1998.
[32] R. Dembo, S. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Num. Anal. 19 (1982) 400–408.
[33] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symetric linear systems, SIAM J. Sci. and Stat. Comput. 7 (1986) 856.
[34] P. Brown, A. Hindmarsh, Matrix-Free methods for stiff systems of ODE's, SIAM J. Num. Anal. 23 (1986) 610–638.
[35] P. McHugh, D. Knoll, Comparison of standard and matrix-free implementations of several Newton–Krylov solvers, AIAA J. 32 (1994) 2394–2400.
[36] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.
[37] F. Incropera, D.P. Dewitt, Introduction to Heat Transfer, John Wiley & Sons, Inc, New York, 1990.
[38] G. VanWylen, R. Sonntag, Fundamentals of Classical Thermodynamics, John Wiley & Sons, New York, 1985.