Nuclear Engineering and Technology 54 (2022) 49–60

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original article

Preconditioned Jacobian-free Newton–Krylov fully implicit high order WENO schemes and flux limiter methods for two-phase flow models

Xiafeng Zhou [a, *], Changming Zhong [a], Zhongchun Li [b, **], Fu Li [c]

[a] Department of Nuclear Engineering and Technology, School of Energy and Power Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China
[b] Nuclear Power Institute of China, Chengdu, 610041, China
[c] Key Laboratory of Advanced Reactor Engineering and Safety of Ministry of Education, Institute of Nuclear and New Energy Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 26 October 2020
Received in revised form
22 June 2021
Accepted 16 July 2021
Available online 17 July 2021

Keywords:
Jacobian-free Newton-Krylov
WENO and flux limiter schemes
Finite-difference-based preconditioning acceleration
Fully-implicit two-fluid models

ABSTRACT

Motivated by the high-resolution properties of high-order Weighted Essentially Non-Oscillatory (WENO) and flux limiter (FL) for steep-gradient problems and the robust convergence of Jacobian-free Newton-Krylov (JFNK) methods for nonlinear systems, the preconditioned JFNK fully implicit high-order WENO and FL schemes are proposed to solve the transient two-phase two-fluid models. Specially, the second-order fully-implicit BDF2 is used for the temporal operator and then the third-order WENO schemes and various flux limiters can be adopted to discrete the spatial operator. For the sake of the generalization of the finite-difference-based preconditioning acceleration methods and the excellent convergence to solve the complicated and various operational conditions, the random vector instead of the initial condition is skillfully chosen as the solving variables to obtain better sparsity pattern or more positions of non-zero elements in this paper. Finally, the WENO_JFNK and FL_JFNK codes are developed and then the two-phase steep-gradient problem, phase appearance/disappearance problem, U-tube problem and linear advection problem are tested to analyze the convergence, computational cost and efficiency in detailed. Numerical results show that WENO_JFNK and FL_JFNK can significantly reduce numerical diffusion and obtain better solutions than traditional methods. WENO_JFNK gives more stable and accurate solutions than FL_JFNK for the test problems and the proposed finite-difference-based preconditioning acceleration methods based on the random vector can significantly improve the convergence speed and efficiency.

1. Introduction

The robust and accurate numerical simulation for the two-phase flow model is significantly important in the design and safety analysis of nuclear reactors systems. However, for most reactor safety analysis code such as RETRAN [1], TRAC [2], RELAP5 [3] and TRACE [4] and so on, the semi-implicit numerical operator splitting iterative methods and the low order (first-order) temporal and spatial discretization schemes are widely adopted to solve the two-phase problems, which introduce the material Courante-Friedrichse-Lewy (CFL) stability limit of the time step size and excessive numerical dissipation. Therefore, the fully implicit and high-fidelity numerical solutions with strong numerical stability and high efficiency are being developed in the new generation of reactor safety analysis and thermal hydraulic code such as RELAP7 [5] and CATHARE-3 [6].

To improve the numerical discrete accuracy, Bertolotto et al. [7] implemented the high-order QUICKEST scheme and ULTIMATE limiter into TRACE to greatly reduce the errors and improve the spatial accuracy of the solute convection equation using the semi-implicit methods, but the computational time can be greatly increased by the high-order methods. Wang et al. [8] applied some 2nd-order flux limiter schemes including the MUSCL, Van Leer, OSPRE, Van Albada and so on to the mass and energy equation in TRACE, which also indicates that the high-resolution schemes can effectively reduce the numerical diffusion, however, the restrictive time-step size limit was needed for Wang's work because of the semi-implicit methods. To allow the large time step and ensure the

* Corresponding author.
** Corresponding author.
E-mail address: zhouxiafeng@hust.edu.cn (X. Zhou).

https://doi.org/10.1016/j.net.2021.07.022

strong numerical stability, Frepoli et al. [9] studied the fully-implicit two-phase three-filed flow model using the Newton method and the relative large time steps can be allowed even on very fine spatial grids which could not be considered for a semi-implicit scheme. Mousseau et al. [10] discussed the physics-based preconditioned Jacobian-free Newton Krylov (JFNK) method for solving the two-phase flow coupled to the heat conduction models. The simulations demonstrate that a fully implicit solution derived from JFNK method can obtain higher numerical accuracy and efficiency than the conventional semi-implicit schemes at the much larger time steps. However, both Frepoli and Mousseau adopted the first order upwind scheme for the spatial discretization.

To ensure the high-order accuracy, the large time step allowed and the strong numerical stability simultaneously, Zou et al. [11–14] applied the symmetric Van Albada limiter for the spatial discretization, the second-order fully-implicit backward differentiation formula (BDF2) for the temporal discretization and JFNK method for nonlinear systems to solve the two-fluid and drift-flux two phase model and the numerical accuracy were presented for the test problems, however, the preconditioned acceleration method and effects for JFNK could not be analyzed in detailed. Ashrafizadeh et al. [15] adopted the Advection Upstream Splitting Method+ (AUSM+) scheme to spatial discretization, and the Crank–Nicolson and backward Euler schemes for the temporal discretization. And then the Newton-Krylov (NK), JFNK, and explicit method are discussed and compared with each other for two-fluid two-phase flow simulation. Hu [16] and Saleem and Kozlowski [17] also combined the second-order Roe type spatial discretization and JFNK method to solve two-fluid six-equation model. In addition, a preconditioner is very important to improve the computational efficiency of JFNK and reduce the Krylov iterative number, some SIMPLE-based preconditioners and matrix decomposition methods were also tried to solve the two phase flow [18,19].

To further reduce potential numerical instabilities of the flux limiter in choosing the unreasonable flux limiter functions and develop the high-efficiency preconditioned methods of JFNK with higher-order discretization in order to obtain high-resolution and fast convergence rate for two-phase two-fluid model, in this work the high-order Weighted Essentially Non-Oscillatory (WENO) schemes are firstly adopted. Then preconditioned JFNK fully implicit high-order WENO schemes and flux limiter (FL) methods are successfully developed to solve the two-phase two-fluid model. The finite-difference-based preconditioning acceleration methods are successfully proposed by choosing the random vector/number instead of the initial condition as the solving variables to predetermine the full sparsity pattern or all the positions of non-zero elements as well as possible in this paper.

The remainder of the paper is organized as follows. Section 2 gives a brief description of the two-phase two-fluid model in this paper. Section 3 presents the fully implicit discretization of the two-phase flow governing equations with the WENO and FL schemes for spatial discretization and BDF2 schemes for temporal discretization. Section 4 focused on the preconditioning acceleration methods for JFNK. In Section 5, numerical results are analyzed and discussed to test the performance of the WENO_JFNK and FL_JFNK. Finally, some conclusions are summarized in Section 6.

2. Two-phase two-fluid model

In this paper, the primary purpose is to demonstrate the advantages of JFNK fully implicit high-order WENO schemes and FL methods to solve the two-phase two-fluid model and to focus on the two-phase hydrodynamic problems. Therefore, the governing equations of the two mass equations and two momentum equations are only considered and given as Eqs. (1)–(4).

$$\frac{\partial(\alpha_g\rho_g)}{\partial t}+\frac{\partial(\alpha_g\rho_g u_g)}{\partial x}=0 \tag{1}$$

$$\frac{\partial(\alpha_l\rho_l)}{\partial t}+\frac{\partial(\alpha_l\rho_l u_l)}{\partial x}=0 \tag{2}$$

$$\frac{\partial(\alpha_g\rho_g u_g)}{\partial t}+\frac{\partial\left(\alpha_g\rho_g u_g^2\right)}{\partial x}=-f_{ig}+f_{gg}-\alpha_g\frac{\partial p}{\partial x} \tag{3}$$

$$\frac{\partial(\alpha_l\rho_l u_l)}{\partial t}+\frac{\partial\left(\alpha_l\rho_l u_l^2\right)}{\partial x}=-f_{il}+f_{gl}-\alpha_l\frac{\partial p}{\partial x} \tag{4}$$

where

α_g = volume fraction of the gas phase (void fraction)
$\alpha_l = 1-\alpha_g$ volume fraction of the liquid phase
u_g, u_l = gas phase velocity and liquid phase velocity, respectively
ρ_g, ρ_l = gas phase density and liquid phase density, respectively
p = pressure
f_{gg}, f_{gl} = gas phase gravity and liquid phase gravity, respectively
f_{ig}, f_{il} = the friction between the phases

The densities are defined as the function of pressure for both phases, and the interfacial drag model can be used in Städtke's work [12,20]. Therefore, the $\rho_g, \rho_l, f_{ig}, f_{il}$ and f_{gg}, f_{gl} are modeled by the following Eqs. (5)–(7).

$$\rho_g(p)=\rho_{g,0}+k_g\%(p-p_0),\ \rho_l(p)=\rho_{l,0}+k_l\%(p-p_0) \tag{5}$$

$$f_{ig}=\frac{1}{8}C_i(u_g-u_l)|u_g-u_l|, f_{il}=-\frac{1}{8}C_i(u_g-u_l)|u_g-u_l|\ C_i=\frac{1}{8}C_d a^{int}\rho_m \tag{6}$$

$$f_{gg}=\alpha_g\rho_g g_x,\ f_{gl}=\alpha_l\rho_l g_x, \tag{7}$$

where

$\rho_{l,0}$ = initial liquid density ($\rho_{l,0}=0.5kg/m^3$)
$\rho_{g,0}$ = initial gas phase density ($\rho_{g,0}=1\times10^3 kg/m^3$)
p_0 = reference pressure ($p_0=10^5 Pa$)
k_l = compressibility of the liquid phase ($k_l=1\times10^{-7}(Kg/m^3)/Pa$)
k_g = compressibility of the gas phase ($k_g=1\times10^{-6}(Kg/m^3)/Pa$)
C_i = interfacial drag coefficient
C_d = drag coefficient ($C_d=0.44$ in this work)
a^{int} = interfacial area per unit volume ($a^{int}=\frac{3\alpha_g(1-\alpha_g)}{r_p}$)
ρ_m = density of the continuous phase ($\rho_m=\alpha_g\rho_g+(1-\alpha_g)\rho_l$)
r_p = particle size ($r_p=0.5\times10^{-3}$)
g_x = gravity

3. Fully implicit high order numerical discretization

The two-fluid model described in Section 2 is used as the governing equation to simulate the two-phase flow in the tube. This section introduces the fully implicit high order temporal and spatial discretization using the BDF2, WENO and FL schemes, which

determine the fully implicit residual equations of WENO_JFNK and FL_JFNK methods for the two-phase flow model in this paper.

3.1. Fully implicit discretization

In this paper, the staggered grid-based finite volume method is used so that the vector variables (velocity) are arranged at the edge of the grid, and the scalar variables (pressure and void fraction) are arranged at the center of the grid. The fully implicit discrete schemes of the two-fluid governing equation can be broadly expressed as following:

$$\frac{m_1(f_kV)_j^{n+1} - m_2(f_kV)_j^{n} + m_3(f_kV)_j^{n-1}}{\Delta t} + (\langle f_k\rangle u_kA)_{j+\frac{1}{2}}^{n+1} - (\langle f_k\rangle u_kA)_{j-\frac{1}{2}}^{n+1} - R_{kj}^{n+1} = 0 \tag{8}$$

where

superscripts $n+1$ and n, $n-1$ = different time points
subscripts k = gas phase or liquid phase ($k = g, l$)
subscripts j = grid spatial index
m_1, m_2, m_3 = implicit time discretization scheme coefficients
Δt = time step size
V = volume of grid
A = area of grid
u_k = gas phase velocity or liquid phase velocity, respectively ($k = g, l$)
f_k = scalar variable or vector variables shown in Table 1
$\langle f_k\rangle = f_k$ that is discreted at the edge of control volume $j+\frac{1}{2}$ and $j-\frac{1}{2}$
R_{kj}^{n+1} = the remainder of the governing equation including f_i, f_g and p in the current time step shown in Table 1.

In this paper, two implicit time discretization schemes are used, one is the first-order backward differencing (BDF1 $m_1 = 1, m_2 = 1, m_3 = 0$), and the other is the second-order backward differencing (BDF2 $m_1 = 1.5, m_2 = 2, m_3 = 0.5$).

3.2. FL schemes

The flux limiter (FL) schemes can give a result as close as possible to the second-order central scheme in smooth regions and as sufficient as possible to avoid the appearance of spurious oscillation if the flux limiter functions can be reasonably chosen. By limiting the slope of interpolation based on the values at the center points of the adjacent control volumes, the mathematical expression of $\langle f_k\rangle$ at the interface of control volume [15] can be presented as follows:

$$\langle f_k\rangle_{j+\frac{1}{2}}^{n+1} = \begin{cases} (f_k)_j^{n+1} + \phi(r_{j+\frac{1}{2}}^{n+1})\ \dfrac{(f_k)_{j+1}^{n+1} - (f_k)_j^{n+1}}{2} & \text{if } (u_k)_{j+\frac{1}{2}}^{n+1} \geq 0 \\ (f_k)_{j+1}^{n+1} - \phi(r_{j+\frac{1}{2}}^{n+1})\ \dfrac{(f_k)_{j+1}^{n+1} - (f_k)_j^{n+1}}{2} & \text{if } (u_k)_{j+\frac{1}{2}}^{n+1} < 0 \end{cases} \tag{9}$$

where

$$r_{j+\frac{1}{2}}^{n+1} = \frac{1}{(f_k)_{j+1}^{n+1} - (f_k)_j^{n+1}} \begin{cases} \dfrac{\Delta x_j + \Delta x_{j+1}}{\Delta x_j + \Delta x_{j-1}} \left[(f_k)_j^{n+1} - (f_k)_{j-1}^{n+1}\right] & \text{if } (u_k)_{j+\frac{1}{2}}^{n+1} \geq 0 \\ \dfrac{\Delta x_j + \Delta x_{j+1}}{\Delta x_{j+1} + \Delta x_{j+2}} \left[(f_k)_{j+2}^{n+1} - (f_k)_{j+1}^{n+1}\right] & \text{if } (u_k)_{j+\frac{1}{2}}^{n+1} < 0 \end{cases} \tag{10}$$

$\phi(r_{j+\frac{1}{2}}^{n+1})$ is the slope limiter and Δx_j is the grid length. $r_{j+\frac{1}{2}}^{n+1}$ is the smoothness parameter to evaluate the change rate of the local gradients. Some slope limiter methods have been shown in Table 2 and more flux limiter can be read in Ref [21,22].

3.3. WENO schemes

WENO schemes have been widely developed to obtain a higher accuracy than ENO type of flux limiter in smooth regions while more effectively avoid the appearance of spurious oscillation [23]. The main idea of WENO scheme is to use the weighted combination of all the existing ENO stencils rather than choosing one specific ENO stencil to make numerical properties as good as possible. Based on the theory of WENO schemes, the WENO schemes can be expressed as Eq. (11) for the interface value.

$$\langle f_k\rangle_{j+\frac{1}{2}}^{n+1} = \begin{cases} \langle f_k\rangle_{j+\frac{1}{2}}^{n+1,-} = \sum_{r=0}^{k-1} \omega_{i+1/2,r}^{n+1,-} (f_k)_{i+1/2,r}^{n+1,-} & \text{if } (u_k)_{j+\frac{1}{2}}^{n+1} \geq 0 \\ \langle f_k\rangle_{j+\frac{1}{2}}^{n+1,+} = \sum_{r=0}^{k-1} \omega_{i+1/2,r}^{n+1,+} (f_k)_{i+1/2,r}^{n+1,+} & \text{if } (u_k)_{j+\frac{1}{2}}^{n+1} < 0 \end{cases} \tag{11}$$

The weight factors $\omega_{j+1/2,r}^{n+1,-}$ and $\omega_{j+1/2,r}^{n+1,+}$ can be given by:

$$\omega_{j+1/2,r}^{n+1,-} = \frac{\tilde{\omega}_{j+1/2,r}^{n+1,-}}{\sum_{r=0}^{k-1} \tilde{\omega}_{j+1/2,r}^{n+1,-}} \cdots \omega_{j+1/2,r}^{n+1,+} = \frac{\tilde{\omega}_{j+1/2,r}^{n+1,+}}{\sum_{r=0}^{k-1} \tilde{\omega}_{j+1/2,r}^{n+1,+}}$$
$$\tilde{\omega}_{j+1/2,r}^{n+1,-} = \frac{C_r^{n+1,-}}{\left(\varepsilon + IS_{j,r}^{n+1,-}\right)^2} \text{ and } \tilde{\omega}_{j+1/2,r}^{n+1,+} = \frac{C_r^{n+1,+}}{\left(\varepsilon + IS_{j,r}^{n+1,+}\right)^2} \tag{12}$$

where $(f_k)_{j+1/2,r}^{n+1,-}$ and $(f_k)_{j+1/2,r}^{n+1,+}$ are the interpolation terms used for

the construction of the polynomial $\langle f_k\rangle_{j+\frac{1}{2}}^{n+1,-}$ and $\langle f_k\rangle_{j+\frac{1}{2}}^{n+1,+}$. The $C_r^{n+1}(\sum C_r^{n+1}=1)$ is a polynomial coefficient and ε is a very small value to ensure that the denominator $\tilde{\omega}_{j+1/2,r}^{n+1,\pm}$ in Eq. (12) is never zero (for example $\varepsilon = 10^{-6}$). $IS_{j,r}$ is a measure of smoothness of interpolation polynomials and $\langle f_k\rangle_{j+\frac{1}{2}}^{n+1}$ can achieve the highest $(2k-1)$ order accuracy in smooth regions. In this work, the WENO3 scheme ($k=2$) are adopted for WENO_JFNK methods and the specific expression of $\langle f_k\rangle_{j+1/2,r}^{n+1,\pm}$, C_r^{n+1} and $IS_{j,r}$ can be shown in Table 3.

3.4. Fully-implicit high-order discretized residuals for JFNK

Based on the above expression, the fully-implicit high-order discretized residuals of Eqs. (1)–(4) in this paper can be obtained as follows

$$Res_{\alpha,j}^{n+1} = V_j^{n+1}\frac{m_1(\alpha_g\rho_g)_j^{n+1} - m_2(\alpha_g\rho_g)_j^{n} + m_3(\alpha_g\rho_g)_j^{n-1}}{\Delta t} + \left(\langle\alpha_g\rho_g\rangle\right)_{j+\frac{1}{2}}^{n+1}(u_gA)_{j+\frac{1}{2}}^{n+1} - \left(\langle\alpha_g\rho_g\rangle\right)_{j-\frac{1}{2}}^{n+1}(u_gA)_{j-\frac{1}{2}}^{n+1} \tag{13}$$

$$Res_{p,j}^{n+1} = V_j^{n+1}\frac{m_1(\alpha_l\rho_l)_j^{n+1} - m_2(\alpha_l\rho_l)_j^{n} + m_3(\alpha_l\rho_l)_j^{n-1}}{\Delta t} + \left(\langle\alpha_l\rho_l\rangle\right)_{j+\frac{1}{2}}^{n+1}(u_lA)_{j+\frac{1}{2}}^{n+1} - \left(\langle\alpha_l\rho_l\rangle\right)_{j-\frac{1}{2}}^{n+1}(u_lA)_{j-\frac{1}{2}}^{n+1} \tag{14}$$

$$Res_{ug,j+\frac{1}{2}}^{n+1} = V_{j+\frac{1}{2}}^{n+1}\frac{m_1(\alpha_g\rho_g)_{j+\frac{1}{2}}^{n+1}(u_g)_{j+\frac{1}{2}}^{n+1} - m_2(\alpha_g\rho_g)_{j+\frac{1}{2}}^{n}(u_g)_{j+\frac{1}{2}}^{n} + m_3(\alpha_g\rho_g)_{j+\frac{1}{2}}^{n-1}(u_g)_{j+\frac{1}{2}}^{n-1}}{\Delta t} + \left(\langle\alpha_g\rho_g u_g\rangle\right)_{j+1}^{n+1}(u_gA)_{j+1}^{n+1} - \left(\langle\alpha_g\rho_g\rangle\right)_{j}^{n+1}(u_gA)_{j}^{n+1} + \left(-f_{ig}V + f_{gg}V\right)_{j+\frac{1}{2}}^{n+1} - (\alpha_gA)_{j+\frac{1}{2}}^{n+1}\left(P_{j+1}^{n+1} - P_j^{n+1}\right) \tag{15}$$

$$Res_{ul,j+\frac{1}{2}}^{n+1} = V_{j+\frac{1}{2}}^{n+1}\frac{m_1(\alpha_l\rho_l)_{j+\frac{1}{2}}^{n+1}(u_l)_{j+\frac{1}{2}}^{n+1} - m_2(\alpha_l\rho_l)_{j+\frac{1}{2}}^{n}(u_l)_{j+\frac{1}{2}}^{n} + m_3(\alpha_l\rho_l)_{j+\frac{1}{2}}^{n-1}(u_l)_{j+\frac{1}{2}}^{n-1}}{\Delta t} + \left(\langle\alpha_l\rho_l u_l\rangle\right)_{j+1}^{n+1}(u_lA)_{j+1}^{n+1} - \left(\langle\alpha_l\rho_l\rangle\right)_{j}^{n+1}(u_lA)_{j}^{n+1} + \left(-f_{il}V + f_{gl}V\right)_{j+\frac{1}{2}}^{n+1} - (\alpha_lA)_{j+\frac{1}{2}}^{n+1}\left(P_{j+1}^{n+1} - P_j^{n+1}\right) \tag{16}$$

where $Res_{\alpha,j}^{n+1}$, $Res_{p,j}^{n+1}$, $Res_{ug,j+\frac{1}{2}}^{n+1}$ and $Res_{ul,j+\frac{1}{2}}^{n+1}$ are the discrete residuals of void fraction α_g, pressure p, gas phase velocity u_g and liquid phase velocity u_l, respectively. It is straightforward to compute the nonlinear residual equations by subtracting the terms on the right hand side from the left hand side of the discrete equations. It is worth noting that the discrete center position of the momentum equation is located at the $j+\frac{1}{2}$. And then the fully implicit high order residual equations of JFNK can be rewritten in a vector form as shown in Eq. (17).

$$\mathrm{F}_{Res}\left(x^{n+1}\right) = \mathrm{F}_{Res}\begin{pmatrix}\alpha_j^{n+1}\\ p_j^{n+1}\\ u_{g,\,j+\frac{1}{2}}^{n+1}\\ u_{l,j+\frac{1}{2}}^{n+1}\end{pmatrix} = \begin{bmatrix}Res_{\alpha,j}^{n+1}\\ Res_{p,j}^{n+1}\\ Res_{ug,j+\frac{1}{2}}^{n+1}\\ Res_{ul,j+\frac{1}{2}}^{n+1}\end{bmatrix} = 0 \tag{17}$$

4. Preconditioning acceleration methods for JFNK

JFNK methods combine the Jacobian-free technique, Newton iteration and Krylov method to solve the complicated nonlinear systems. The basic framework can be expressed as

$$\begin{aligned} &J\left(x^{n+1,k}\right)\cdot P^{-1,k}\cdot\delta u^{n+1.k} = -\mathrm{F}_{Res}\left(x^{n+1,k}\right)\\ &\delta u^{n+1.k} = P^{k}\cdot\delta x^{n+1.k}\\ &x^{n+1,k+1} = x^{n+1,k} + P^{-1,k}\delta u^{n+1.k}\end{aligned} \tag{18}$$

where $J(x^{n+1,k})$ is the Jacobian matrix of the kth Newton step. The sparse linear systems Eq. (18) can be solved by the Krylov methods (for example, GMRES or BICGSTAB). Because only the product of matrix and vector is needed for the Krylov method, the product of the Jacobian matrix $J(x^{n+1,k})$ and the vector can be approximated by a finite difference of the residuals $\mathrm{F}_{Res}(x^{n+1})$ for JFNK methods and doesn't require to explicitly construct the Jacobian matrix $J(x^{n+1,k})$ in order to reduce the computational cost and memory for complicated problems:

$$J\left(x^{n+1,k}\right)\cdot\nu = \frac{\mathrm{F}_{Res}\left(x^{n+1,k}+\varepsilon\cdot\nu\right) - \mathrm{F}_{Res}\left(x^{n+1,k}\right)}{\varepsilon} \tag{19}$$

where ν is a basis vector of Krylov subspaces and ε is the

Table 1
$f_k$and $R_{k,j}^{n+1}$ for mass or momentum equations.

Governing equation	f_k	$R_{k,j}^{n+1}$
Mass equation	$\alpha_k\rho_k$	0
Momentum equation	$\alpha_k\rho_k u_k$	$(-f_{ik}V+f_{gk}V)^{n+1}_{j+\frac{1}{2}} - (\alpha_k A)^{n+1}_{j+\frac{1}{2}}(P^{n+1}_{j+1}-P^{n+1}_{j})$

perturbation parameter that is chosen and studied in many references [25,26].

Since most of the computational time for JFNK methods is spent in Krylov iteration, an efficient preconditioner P^k is very important to improve the convergence rate and the computational efficiency, which also has a great influence on the Newton iteration of JFNK. To obtain a good preconditioning effect, the preconditioner P^k in every Newton step should be a good approximation for Jacobian matrix $J(x^{n+1,k})$ and the matrix inversion $P^{-1,k}$ can be solved as easily as possible.

Though the full sparsity of the Jacobin matrix can be theoretically pre-determined by the mesh connectivity and the stencil of the schemes, for the sake of the generalization of the preconditioning acceleration methods to solve the complicated and various operational conditions of the two phase flow model, in this paper the finite difference methods [27] are proposed and adopted to obtain the positions (sparsity pattern) and the specific values of non-zero elements of Jacobin matrix by the following expression:

$$J_{i,j}\left(x^{n+1,k}\right)=\frac{\partial \mathrm{F}_{i,Res}}{x_j^{n+1,k}}=\frac{\mathrm{F}_{i,Res}\left(x^{n+1,k}+h_j\cdot e_j\right)-\mathrm{F}_{i,Res}\left(x^{n+1,k}\right)}{h_j} \tag{20}$$

where e_j is the column j of the identity matrix and h_j is the perturbation parameter. x is the variable and $F(x)$ is the residual. Specially, the positions and number of non-zero elements (sparsity pattern) of Jacobian matrix are firstly obtained as the Step one of the finite difference methods and then the specific values of the non-zero elements are computed as the Step two.

We find that maybe the finite difference methods cannot give the full sparsity pattern or the positions of all the non-zero elements of the Jacobin matrix if the variables x at initial condition are chosen to obtain the sparsity structure. More specifically, Because the sparsity pattern of the Jacobian matrix are numerically determined by the specific values of the residuals $F(x)$ based on the variable x using Eq. (20, the sparsity pattern obtained by the finite difference methods may be not equal to the theoretical and full sparsity structure.

To explain the above-mentioned point, we take the flux limiter of Van Leer scheme as an example. It can be observed from Eq. (9) that $\langle f_k\rangle_{j+\frac{1}{2}}^{n+1}$ should be connected with the variables $(f_k)_j^{n+1}$ and $(f_k)_{j+1}^{n+1}$ for $(u_k)_{j+\frac{1}{2}}^{n+1}\geq 0$, however, when $(f_k)_{j+1}^{n+1}$ is equal to $(f_k)_j^{n+1}$ or the slope limiter $\phi(r_{j+\frac{1}{2}}^{n+1})$ is zero for the initial conditions, $\langle f_k\rangle_{j+\frac{1}{2}}^{n+1}$ is

Table 2
FL schemes used in the FL_JFNK methods [21,22].

Discretization scheme	flux limiter function $\phi\left(r^{n+1}_{j+\frac{1}{2}}\right)$
FUD (First order upwind)	0
CD (Central difference)	1
VA (Van Albada)	$\left(r^{n+1}_{j+\frac{1}{2}}+\left(r^{n+1}_{j+\frac{1}{2}}\right)^2\right)\Big/\left(1+\left(r^{n+1}_{j+\frac{1}{2}}\right)^2\right)$
VL(Van Leer)	$\left(r^{n+1}_{j+\frac{1}{2}}+\left\|r^{n+1}_{j+\frac{1}{2}}\right\|\right)\Big/\left(1+\left\|r^{n+1}_{j+\frac{1}{2}}\right\|\right)$
ENO/Minmod	$max\left[0,min\left(1,r^{n+1}_{j+\frac{1}{2}}\right)\right]$

Table 3
WENO3 scheme for spatial discretization [24].

WENO3	Discretization scheme for spatial discretization
$(f_k)_{j+1/2,0}^{n+1,-}$	$(\Delta x_{j+1}(f_k)_j^{n+1})/(\Delta x_j+\Delta x_{j+1})+(\Delta x_j(f_k)_{j+1}^{n+1})/(\Delta x_j+\Delta x_{j+1})$
$(f_k)_{j+1/2,1}^{n+1,-}$	$(1+\Delta x_j/(\Delta x_j+\Delta x_{j-1}))(f_k)_j^{n+1}-(\Delta x_j/(\Delta x_j+\Delta x_{j-1}))(f_k)_{j-1}^{n+1}$
$(f_k)_{j+1/2,0}^{n+1,+}$	$(1+\Delta x_{j+1}/(\Delta x_{j+1}+\Delta x_{j+2}))(f_k)_{j+1}^{n+1}-0.5(\Delta x_{j+1}/(\Delta x_{j+1}+\Delta x_{j+2}))(f_k)_{j+2}^{n+1}$
$(f_k)_{j+1/2,1}^{n+1,+}$	$(\Delta x_j/(\Delta x_j+\Delta x_{j+1}))(f_k)_j^{n+1}+(\Delta x_{j+1}/(\Delta x_j+\Delta x_{j+1}))(f_k)_{j+1}^{n+1}$
$C_0^{n+1,-}$	$(\Delta x_{j-1}+\Delta x_j)/(\Delta x_{j-1}+\Delta x_j+\Delta x_{j+1})$
$C_1^{n+1,-}$	$(\Delta x_{j+1})/(\Delta x_{j-1}+\Delta x_j+\Delta x_{j+1})$
$C_0^{n+1,+}$	$(\Delta x_j)/(\Delta x_j+\Delta x_{j+1}+\Delta x_{j+2})$
$C_1^{n+1,+}$	$(\Delta x_{j+1}+\Delta x_{j+2})/(\Delta x_j+\Delta x_{j+1}+\Delta x_{j+2})$
$IS_{j,0}^{n+1,-}$	$4(\Delta x_j(f_k)_{j+1}^{n+1}/(\Delta x_j+\Delta x_{j+1})-(f_k)_j^{n+1})^2$
$IS_{j,1}^{n+1,-}$	$4(\Delta x_j(f_k)_j^{n+1}/(\Delta x_j+\Delta x_{j-1})-(f_k)_{j-1}^{n+1})^2$
$IS_{j,0}^{n+1,+}$	$4(\Delta x_{j+1}(f_k)_{j+2}^{n+1}/(\Delta x_{j+1}+\Delta x_{j+2})-(f_k)_{j+1}^{n+1})^2$
$IS_{j,1}^{n+1,+}$	$4(\Delta x_{j+1}(f_k)_{j+1}^{n+1}/(\Delta x_{j+1}+\Delta x_j)-(f_k)_j^{n+1})^2$

Table 4
Update frequency of sparsity structure of the preconditioners for JFNK.

Strategy/ Case	Calculation method of Sparsity pattern of Jacobian matrix using Eq. (20) (positions of non-zero elements, Step one)
A	Obtain the sparsity pattern by choosing the random vector as the variables once at the initial time (calculate the sparsity pattern once)
B	Obtain the sparsity pattern by choosing the initial condition as the variables once at the initial time (calculate the sparsity pattern once)
C	a) Obtain the sparsity pattern by choosing the initial condition as the variables for the first time at the initial time; b) Obtain the sparsity pattern by choosing the variables at the time t = Δt for the second time at the time t = Δt (calculate the sparsity pattern twice)
D	a) Obtain the sparsity pattern by choosing the initial condition as the variables for the first time at the initial time; b) Obtain the sparsity pattern by choosing the variables at the time t = Δt for the second time at the time t = Δt; c) Obtain the sparsity pattern by choosing the variables at the time t = 2Δt for the second time at the time t = 2Δt (calculate the sparsity pattern three times)

Table 5
Parameters for steep gradient problems.

Parameter	Value	Unit
Pipe length L	2	m
Gravity	0.0	m/s^2
$u_{l,init}$	1	m/s
$u_{g,init}$	1	m/s
α_{init}	$\begin{cases} 0.7-1.25\times\lvert x-0.6\rvert \ if 0.2\le x\le 0.6 \\ 0.7-1.25\times\lvert x-1.4\rvert \ if 1.0\le x\le 1.4 \\ 0.2 \quad otherwise \end{cases}$	–
P_{init}	10^5	Pa
α_{inlet}	0.2	–
$u_{l,inlet}$	1	m/s
$u_{g,inlet}$	1	m/s
P_{outlet}	10^5	Pa
N_{cell}	20/100/200	–
Δt	1×10^{-3}	s
$N_{timestep}$	200	–

only connected with the variables $(f_k)_j^{n+1}$ if the finite difference methods are adopted on the basis of the initial conditions. We cannot obtain the full sparsity pattern of $(f_k)_j^{n+1}$ and $(f_k)_{j+1}^{n+1}$. as well as possible if the sparsity pattern is constructed only once by choosing the initial conditions as the variables, which maybe result in the unreasonable preconditioners.

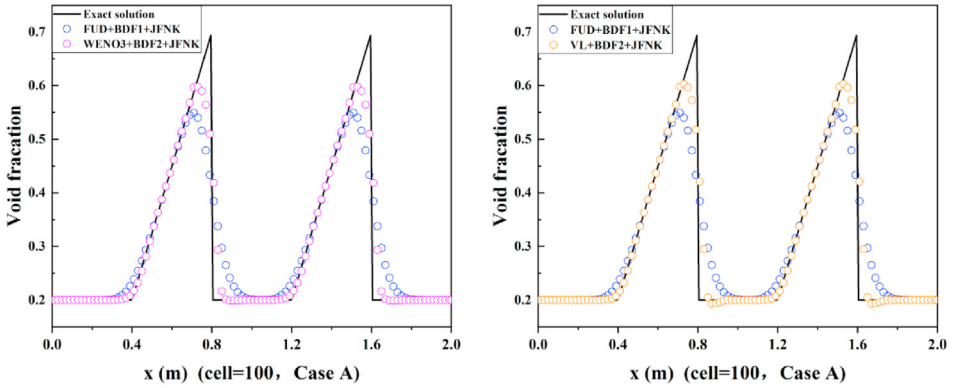
We take it for granted that the method of updating sparsity pattern several times with the change of the variables is adopted to get the full sparsity structure as well as possible as shown in Table 4 (Case C ~ D). However, more frequent update of Jacobian matrix will result in more computational cost. Therefore, to prevent the finite difference methods from getting the incomplete sparsity pattern of

Table 6
L-1 norm errors of two-phase flow peak problem.

Different methods	L-1 Norm		
	N_{cell} = 50	N_{cell} = 100	N_{cell} = 200
FUD + BDF1+JFNK	7.98×10^{-2}	4.62×10^{-2}	2.93×10^{-2}
VL + BDF2+JFNK	5.01×10^{-2}	1.47×10^{-2}	9.05×10^{-3}
WENO3+BDF2+JFNK	4.38×10^{-2}	1.40×10^{-2}	8.75×10^{-3}

the Jacobin matrix when the variable $(f_k)_{j+1}^{n+1}$ is equal to $(f_k)_j^{n+1}$ or the slope limiter $\phi\left(r_{j+\frac{1}{2}}^{n+1}\right)$ is zero on the basis of the initial conditions, the random number is chosen as the variables x to predetermine the full sparsity pattern or all the positions of non-zero elements as well as possible using the finite difference methods (Case A as shown in Table 4) because it is usually not possible for the random vector to make the variable $(f_k)_{j+1}^{n+1}$ be equal to $(f_k)_j^{n+1}$ and the slope limiter $\phi\left(r_{j+\frac{1}{2}}^{n+1}\right)$ be zero. After that, the sparsity structure obtained by using the random vector needs to be calculated only once to get better sparsity pattern or more positions of non-zero elements than that using the variables from the initial conditions, which avoids adopting the method of updating sparsity pattern several times in the original manuscript to get the full sparsity structure.

In addition, the ILU(k) matrix decomposition [28] is adopted to obtain the preconditioning matrix based on the sparsity of Jacobian matrix calculated by the finite difference methods. To analyze the proposed finite-difference-based preconditioning acceleration methods based on the random vector, Table 4 show the different finite-difference-based preconditioning strategies for the



(a) WENO3+BDF2+JFNK (b) VL+BDF2+JFNK

Fig. 1. Numerical results of void fraction for the steep-gradient two-phase problems for N_{cell} = 100 and Case A.

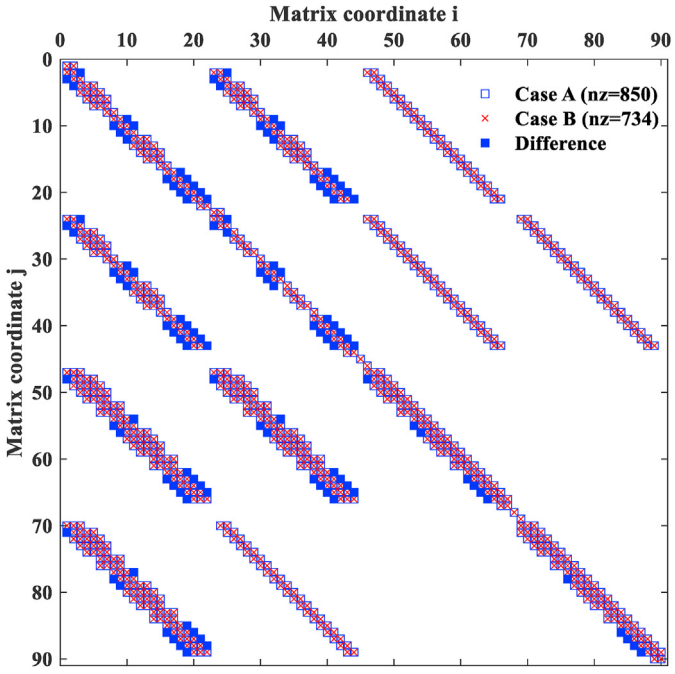


Fig. 2. Sparsity structure of Jacobian matrix for VL + BDF2+JFNK (N_{cell} = 20).

WENO_JFNK and FL_JFNK methods (Case A~D).

5. Numerical analysis and results

Based on the formulation and the algorithm presented in Sec.3–4, the code WENO_JFNK and FL_JFNK are developed to solve the two fluid models using the Fortran language. The steep-gradient problem, phase appearance/disappearance problem, U-tube problem and linear advection problem are tested to study the numerical properties of WENO_JFNK and FL_JFNK. Then proposed finite-difference-based preconditioning methods based on the random vector are analyzed for the different problems in detail. For the numerical results, FUD + BDF1+JFNK, VL + BDF2+JFNK and VA + BDF2+JFNK methods (FUD, VL and VA schemes for spatial discretization and BDF1 and BDF2 schemes for temporal discretization respectively) are chosen as the representatives of the code WENO_JFNK. WENO3+BDF2+JFNK methods (WENO3 schemes for spatial discretization and BDF2 schemes for temporal discretization respectively) is used for the code WENO_JFNK.

5.1. Steep-gradient two-phase flow problem

The steep gradient problem is often used to test the numerical tracking ability of nuclear reactor analysis code on relevant scalars such as void fraction distributions, which is difficult to get the accurate numerical solutions because of the unphysical numerical diffusion or oscillation for many numerical schemes.

The boundary conditions, initial profiles, mesh number, time step and other parameters for the steep gradient problems in this paper are presented in Table 5. Compared with the Zou's works [11], the different void fraction and larger time step are chosen in this paper and the ILU (k = 3) are chosen to obtain the preconditioner.

The numerical solutions of void fraction using the different methods including FUD + BDF1+JFNK,VL + BDF2+JFNK and WENO3+BDF2+JFNK methods are shown in Fig. 1 for $N_{cell} = 100$ and the preconditioning strategy of Case A. The L-1 error norms [11] of void fraction for the different methods are listed in Table 6 on the different mesh numbers ($N_{cell} = 50, 100, 200$) and can be defined as

$$\|L\|_1 = \Delta x \sum_{i=1}^{Ncell} \left|\alpha_g(x_i) - \alpha_{g,exact}\right| \tag{21}$$

where $\alpha_g(x_i)$ is the numerical solution, and $\alpha_{g,exact}$ is the exact solution, Δx is the cell size, N_{cell} is the number of cells.

It can be clearly observed from Fig. 1 and Table 6 that VL + BDF2+JFNK and WENO3+BDF2+JFNK have more accurate numerical solutions than FUD + BDF1+JFNK and significantly

Table 7
Convergence and CPU time for VL + BDF2+JFNK methods.

Different preconditioning Cases	Convergence criteria per time step	N_{cell}	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time (s)
A	10^{-6}	50	3.020	1.005	1.0764
		100	3.330	1.512	3.5412
		200	3.025	1.031	9.8124
	10^{-10}	50	4.185	2.705	1.6224
		100	6.430	2.592	6.1916
		200	4.665	2.429	16.2865
B	10^{-6}	50	3.275	2.032	1.0984
		100	3.455	2.229	3.6704
		200	3.500	2.606	9.9840
	10^{-10}	50	8.215	3.550	2.6052
		100	8.515	3.401	8.4864
		200	9.575	3.445	28.5950
C	10^{-6}	50	3.280	1.881	1.4664
		100	3.345	1.572	4.2448
		200	3.045	1.338	10.7001
	10^{-10}	50	7.120	3.082	2.9484
		100	8.230	2.865	9.1572
		200	6.650	2.541	23.2909
D	10^{-6}	50	3.335	1.964	1.5600
		100	3.825	2.229	4.6832
		200	3.590	1.969	10.7001
	10^{-10}	50	8.560	3.054	3.1796
		100	8.635	3.415	9.5460
		200	9.655	4.046	33.540

Table 8
Convergence and CPU time for WENO3+BDF2+JFNK methods.

Different preconditioning Cases	Convergence criteria per time step	N_{cell}	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time (s)
A	10^{-6}	50	2.650	1.234	1.2232
		100	2.620	1.006	2.5642
		200	2.625	1.213	8.2112
	10^{-10}	50	5.120	2.380	2.3402
		100	4.080	2.837	5.0544
		200	4.275	2.752	16.0681
B	10^{-6}	50	2.730	1.073	1.2436
		100	2.710	1.288	2.7768
		200	2.650	1.245	8.5377
	10^{-10}	50	5.700	2.521	2.5492
		100	4.240	3.100	5.5884
		200	4.330	2.808	16.6333
C	10^{-6}	50	2.685	1.279	1.3324
		100	2.690	1.273	2.8672
		200	2.645	1.227	8.8552
	10^{-10}	50	5.565	2.521	2.9681
		100	4.210	2.876	5.9438
		200	4.325	2.793	17.3665
D	10^{-6}	50	2.675	1.250	1.4701
		100	2.670	1.243	3.0642
		200	2.640	1.227	9.0748
	10^{-10}	50	5.295	2.606	3.2544
		100	4.150	2.810	6.3996
		200	4.295	2.796	18.2249

Table 9
Parameters for the phase disappearance/appearance problem.

Parameter	Value	Unit
Pipe length L	1	m
Gravity	0.0	m/s^2
$u_{l,init}$	1	m/s
$u_{g,init}$	1	m/s
α_{init}	$\begin{cases} 0.0 & if 0.2 \le x \le 0.4 \\ 1.0 & otherwise \end{cases}$	–
P_{init}	10^5	Pa
α_{inlet}	periodic	–
$u_{l,inlet}$	periodic	m/s
$u_{g,inlet}$	periodic	m/s
P_{outlet}	periodic	Pa
N_{cell}	200	–
Δt	1×10^{-3}	s
$N_{timestep}$	200	–

reduce the unphysical numerical diffusion of the low order FUD + BDF1 method, but the slight numerical oscillation can be observed at x = 0.8 m and x = 1.6 m for the VL + BDF2+JFNK method. However, WENO3+BDF2+JFNK can give the more stable and accurate solutions than VL + BDF2+JFNK for the tested steep gradient or discontinuous cases.

To analyze the different preconditioning strategies, the sparsity structure of the Jacobian matrix for VL + BDF2+JFNK methods with $N_{cell} = 20$ is presented in Fig. 2. The parameter nz means the number of non-zero elements of sparsity structure and the difference of sparsity structure between Case A and Case B is located in blue box. It can be seen that there is the significant difference between Case A and Case B. Case A can obtain more non-zero elements and better sparsity structure than Case B, which indicates the effectiveness and reasonability for the preconditioning strategy of Case A.

In addition, the convergence and CPU time of VL + BDF2+JFNK and WENO3+BDF2+JFNK methods are tested and analyzed on the different mesh numbers. The average Newton iterative number per time step, average GMRES iterative number per Newton step and CPU time are summarized in Tables 7 and 8, respectively. Numerical results show that the preconditioning acceleration strategy of Case A has the best convergence and highest efficiency for both VL + BDF2+JFNK and WENO3+BDF2+JFNK among the four preconditioning strategies/cases.

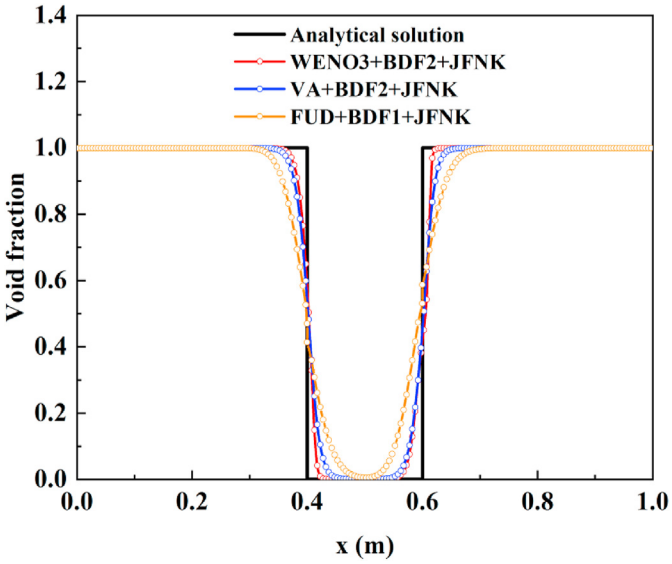


Fig. 3. Numerical results of void fraction using the FUD + BDF1+JFNK, VA + BDF2+JFNK and WENO3+BDF2+JFNK methods with $N_{cell} = 200$ for the phase disappearance/appearance two phase problem.

5.2. Phase appearance/disappearance two-phase flow problem

The phase disappearance/occurrence two-phase flow is often occurred in reactor thermal hydraulic analysis, which is also a very challenging problem due to the difficulty of capturing sharp discontinuities and the prone to non-physical oscillations [12,29]. The

Table 10
Convergence and CPU time for the phase disappearance/appearance two-phase problem using WENO3+BDF2+JFNK with $N_{cell} = 200$.

WENO3+BDF2+JFNK with the different preconditioning Cases	Convergence criteria per time step	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time	Number of non-zero elements
A	10^{-6}	3.27	2.26	11.15	10004
	10^{-10}	4.49	3.70	15.58	
B	10^{-6}	No convergence	_	_	7487 (First time)
	10^{-10}				
C	10^{-6}	3.30	2.26	11.70	9896 (Second time)
	10^{-10}	4.55	3.89	16.59	
D	10^{-6}	3.30	2.56	12.51	9861 (Third time)
	10^{-10}	4.55	4.06	17.27	

Table 11
Convergence and CPU time for the phase disappearance/appearance two-phase problem using VA + BDF2+JFNK with $N_{cell} = 200$.

VA + BDF2+JFNK with the different preconditioning Cases	Convergence criteria per time step	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time	Number of non-zero elements
A	10^{-6}	4.12	2.69	13.34	9977
	10^{-10}	5.58	3.83	19.94	
B	10^{-6}	No convergence	_	_	4706 (First time)
	10^{-10}				
C	10^{-6}	4.38	2.82	14.44	9595 (Second time)
	10^{-10}	6.05	4.16	21.56	
D	10^{-6}	4.89	2.70	14.84	9578 (Third time)
	10^{-10}	6.78	3.81	24.24	

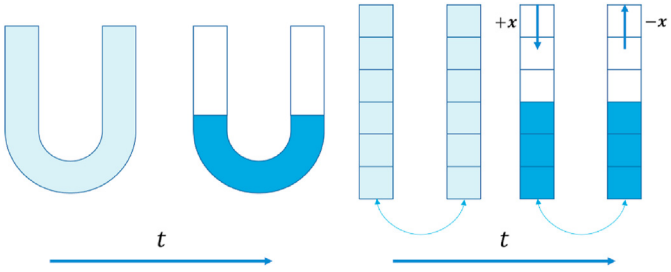


Fig. 4. Schematic of the U-tube two-phase flow problem.

parameters for this test problem are given in Table 9 and the ILU (k = 4) are chosen to obtain the preconditioner.

Fig. 3 shows the numerical results of the void fraction using the FUD + BDF1+JFNK, VA + BDF2+JFNK and WENO3+BDF2+JFNK methods with $N_{cell} = 200$ and the preconditioning strategy of Case A. The WENO3+BDF2+JFNK methods gives the best numerical solutions for the above-mentioned three methods and the FUD + BDF1+JFNK methods has the worst solutions. Then Tables 10 and 11 present the average Newton iterative number per time step, average GMRES iterative number per Newton step and CPU times of WENO3+BDF2+JFNK and VA + BDF2+JFNK with the different preconditioning strategies (Case A~Case D), respectively. As expected, the JFNK method with the preconditioning strategy of Case A shows the relatively robustness and high computational efficiency for the phase appearance/disappearance two-phase flow problem compared with other preconditioning strategies. However, there is no convergence for the preconditioning strategy of Case B. In addition, it can be observed from Tables 10 and 11 that WENO3+BDF2+JFNK unexpectedly shows higher computational efficiency and better convergence than VA + BDF2+JFNK for the tested phase appearance/disappearance two-phase flow problem because of the fewer GMRES iterative number for WENO3+BDF2+JFNK methods.

Table 12
Parameters for the U-tube two-phase flow problem.

Parameter	Value	Unit
Pipe length L	4.0	m
Gravity g	$\begin{cases} 9.81 & if x \le 2 \\ -9.81 & otherwise \end{cases}$	m/s^2
$u_{l,init}$	0	m/s
$u_{g,init}$	0	m/s
α_{init}	0.5	_
P_{init}	10^5	Pa
α_{inlet}	Non-penetrating wall	_
$u_{l,inlet}$	Non-penetrating wall	m/s
$u_{g,inlet}$	Non-penetrating wall	m/s
P_{outlet}	Non-penetrating wall	Pa
N_{cell}	50	_
Δt	1×10^{-2}	s
$N_{timestep}$	1000	_

5.3. U-tube two-phase flow problem

The typical U-tube two-phase flow test problem is originally proposed by Delhaye [30], which has spawned many similar two-phase numerical benchmark problems and is widely used in simulating two-phase flow interaction. In this manuscript, the U-tube two-phase flow problem is a hybrid one of the typical U-tube manometer problem and the sedimentation problem [12,13]. At the same time, the semicircle part at the bottom of the U-tube is simplified by the two interconnected straight pipes. The gravity field is equal to the constant g and does not change along the semicircle pipe, which is similar to the simplification of U-tube problem as shown in Refs. [12,13].

The schematic of the U-tube two-phase flow problem is shown in Fig. 4 and the parameters are summarized in Table 12. A straight tube with a different gravitational constant G in different directions is used to approximate the U-tube two-phase flow problem. Due to the gravity effect, the uniformly mixed two-phase flow can be eventually separated so that the lighter gas phase is located at the top of the tube and the heavier liquid phase at the bottom of the tube. Since many researchers have studied the results of FL methods, herein we focus on the results of the WENO scheme and the efficiency of the proposed preconditioned acceleration methods.

Figs. 5–7 show the transient numerical results of the void fraction

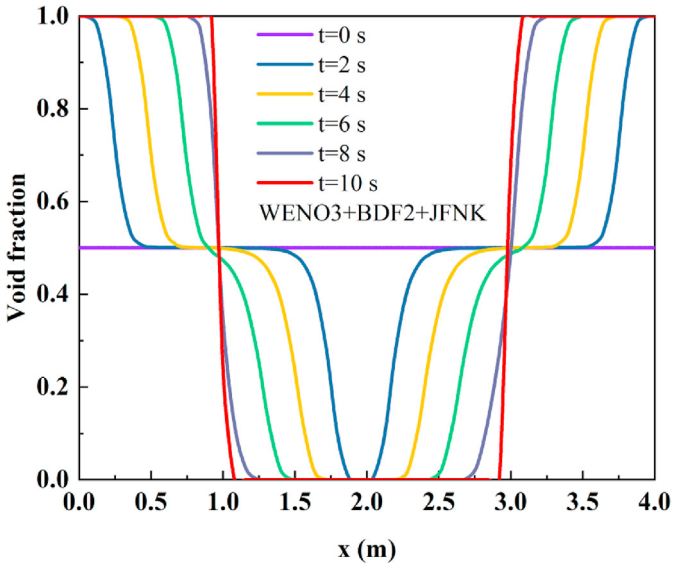


Fig. 5. Numerical results of void fraction distribution using the WENO3+BDF2+JFNK method for the U-tube two-phase flow problems.

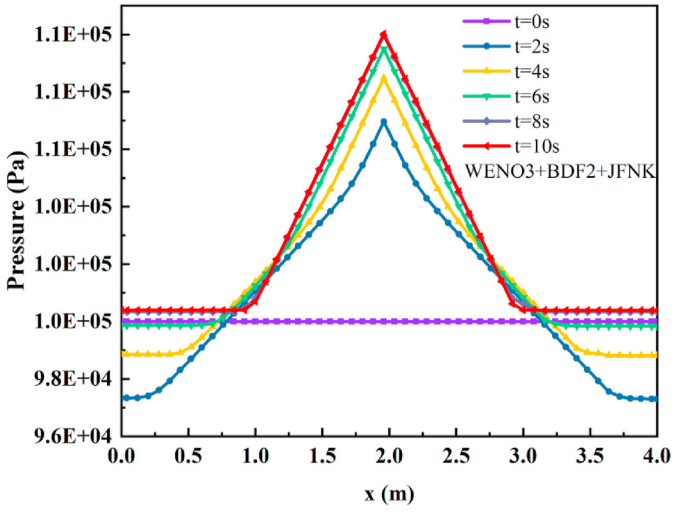


Fig. 6. Numerical results of pressure distribution using the WENO3+BDF2+JFNK method for the U-tube two-phase flow problems.

distribution, pressure distribution and liquid velocity distribution using the high order WENO3+BDF2+JFNK method at the different time, respectively. These numerical solutions shows the good symmetry. The gas-liquid interface can be clearly observed and finally moves to the middle position of the U-tube height at about 10s.

The average Newton iterative number per time step, average GMRES iterative number per Newton step and total CPU time of the WENO3+BDF2+JFNK method with the different preconditioning strategies for the U-tube two-phase flow test problem are summarized in Table 13 to further verify the effectiveness and good convergence behavior of the preconditioning acceleration method of Case A.

5.4. Linear advection problem

The linear advection problem is chosen to further analyze the numerical accuracy and preconditioning efficiency of FL_JFNK and WENO_JFNK codes. The sine distribution and peak in space will be always kept with time for the problem herein. The parameters are summarized in Table 14, and the ILU (k = 3) are chosen to obtain the preconditioner.

Numerical results of the void fraction are presented in Fig. 8 and Table 15 using FUD + BDF1+JFNK, VA + BDF2+JFNK and WENO3+BDF2+JFNK methods with N_{cell} = 200. WENO3+BDF2+

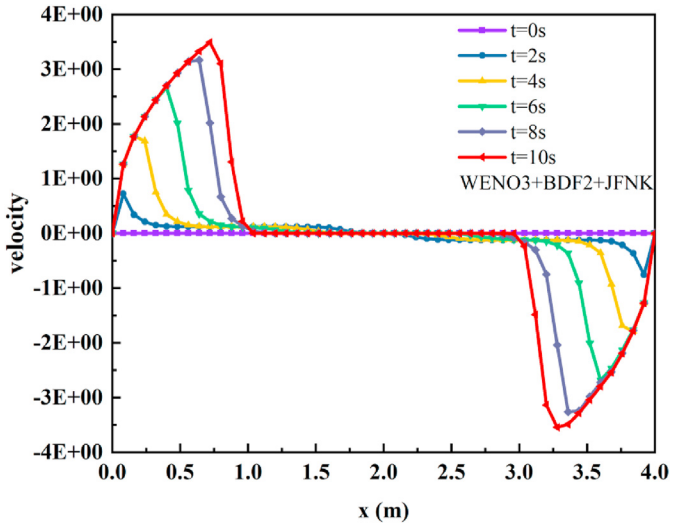


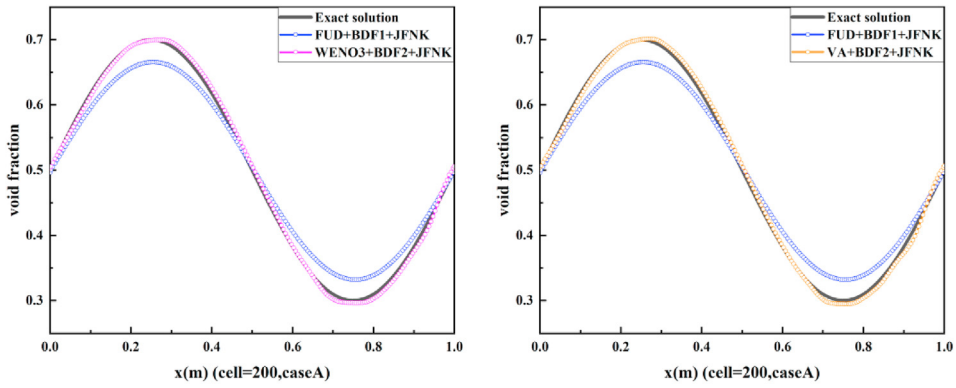
Fig. 7. Numerical results of liquid velocity distribution using the WENO3+BDF2+JFNK method for the U-tube two-phase flow problems.

Table 13
Convergence and CPU time for the U-tube two-phase flow problems with N_{cell} = 50.

WENO3+BDF2+JFNK with the different preconditioning Cases	Convergence criteria per time step	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time	Number of non-zero elements
A	10^{-6}	3.90	5.04	4.36	2265
	10^{-10}	4.68	6.89	5.19	
B	10^{-6}	No convergence	_	_	1602
	10^{-10}				
C	10^{-6}	3.89	5.08	4.51	2262
	10^{-10}	4.69	6.92	5.37	
D	10^{-6}	3.92	5.05	4.76	2260
	10^{-10}	4.70	6.92	5.67	

Table 14
Parameters for the linear advection problem.

Parameter	Value	Unit
Pipe length L	1	m
Gravity	0.0	m/s^2
$u_{l.init}$	1	m/s
$u_{g.init}$	1	m/s
α_{init}	$0.5 + 0.2\sin\left(\frac{2\pi x}{L}\right)$	–
P_{init}	10^5	Pa
α_{inlet}	periodic	–
$u_{l.inlet}$	periodic	$m/_s$
$u_{g.inlet}$	periodic	$m/_s$
P_{outlet}	periodic	P_a
N_{cell}	200	–
Δt	5×10^{-3}	s
Pipe length L	200	m



(c) WENO3+BDF2+JFNK (d) VA+BDF2+JFNK

Fig. 8. Numerical results of void fraction for the linear advection problems for $N_{cell} = 200$ and Case A.

Table 15
L-1 norm errors of two-phase flow linear advection problem.

Different methods	L-1 Norm ($N_{cell} = 200$)
FUD + BDF1+JFNK	2.13×10^{-2}
VL + BDF2+JFNK	5.54×10^{-3}
WENO3+BDF2+JFNK	5.19×10^{-3}

JFNK and VA + BDF2+JFNK methods can obtain better solutions than the FUD + BDF1+JFNK method. And WENO3+BDF2+JFNK can give the more smooth and accurate solutions than VA + BDF2+JFNK for the continuous cases.

Tables 16 and 17 present the numerical properties and efficiency of the different preconditioning strategies (Case A~Case D). Since the solving-variables x has a sine distribution in space, it is usually impossible to make the variable $(f_k)_{j+1}^{n+1}$ be equal to $(f_k)_j^{n+1}$ and the slope limiter $\phi\left(r_{j+\frac{1}{2}}^{n+1}\right)$ be zero in Section 4, Therefore, the initial conditions and profiles can give the similar sparsity pattern with the random vector chosen as the solving-variables x from the number of non-zero elements in Tables 16 and 17, which can further verify the importance of choosing the appropriate solving-variables and the rationality and generalization of choosing the random vector to pre-determine the better full sparsity pattern.

6. Conclusion

This paper presents the preconditioned JFNK fully implicit high-order WENO schemes (WENO_JFNK) and FL methods (FL_JFNK) to solve the transient two-phase two-fluid model. The proposed finite-difference-based preconditioning acceleration methods for WENO_JFNK and FL_JFNK choose the random vector/number instead of the initial condition as the solving variables to pre-determine the full sparsity pattern or all the positions of non-zero elements as well as possible.

Numerical results show that both WENO_JFNK and FL_JFNK can

Table 16
Convergence and CPU time for the linear advection problem using WENO3+BDF2+JFNK with $N_{cell} = 200$.

WENO3+BDF2+JFNK with the different preconditioning Cases	Convergence criteria per time step	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time	Number of non-zero elements
A	10^{-6}	4.89	4.30	15.11	9401
	10^{-10}	6.48	5.44	18.42	
B	10^{-6}	4.91	4.35	15.69	9385
	10^{-10}	6.53	5.48	19.08	
C	10^{-6}	4.93	4.33	16.42	9392
	10^{-10}	6.51	5.45	19.75	
D	10^{-6}	4.90	4.31	16.62	9395
	10^{-10}	6.49	5.44	20.45	

Table 17
Convergence and CPU time for the linear advection problem using VA + BDF2+JFNK with $N_{cell} = 200$.

VA + BDF2+JFNK with the different preconditioning Cases	Convergence criteria per time step	Average Newton number per time step	Average GMRES iterative number per Newton step	CPU time	Number of non-zero elements
A	10^{-6}	4.14	3.46	11.42	9367
	10^{-10}	5.81	4.56	16.23	
B	10^{-6}	4.19	3.69	11.77	9349
	10^{-10}	6.16	4.68	17.42	
C	10^{-6}	4.11	3.62	11.97	9354
	10^{-10}	5.83	4.66	18.86	
D	10^{-6}	4.19	3.68	12.06	9356
	10^{-10}	5.89	4.64	19.40	

significantly reduce numerical diffusion and obtain better solutions than low order FUD + BDF1+JFNK methods. But the slight numerical oscillation can be observed for the FL_JFNK method in solving the steep gradient two-phase problem. However, the WENO_JFNK method has no visible numerical oscillation and gives the more stable and accurate solutions than FL_JFNK for these three test problems shown in this paper by using the weighted combination of all the existing ENO stencils rather than choosing one specific ENO type of flux limiter to make numerical properties as good as possible.

In addition, the good convergence behavior and efficiency of the proposed finite-difference-based preconditioning acceleration methods are also verified for both the WENO_JFNK and FL_JFNK methods by choosing the random vector instead of the initial condition as the solving variables to pre-determine the better full sparsity pattern or get more positions of non-zero elements as well as possible. Further studies need to focus on extending the WENO_JFNK method and the proposed preconditioning acceleration to solve the more realistic and complicated two-phase flow problems.

Data availability

This data used to support the findings of this study are available from the corresponding author upon request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (12005073),the Project of Nuclear Power Technology Innovation Center of Science Technology and Industry for National Defense (HDLCXZX-2021-HD-033, HDLCXZX-2019-ZH-26), Science and Technology on Reactor System Design Technology Laboratory Foundation Project (HT-KFKT-10-2018005).

References

[1] J.H. McFadden, et al., RETRAN-02: a Program for Transient Thermal-Hydraulic Analysis of Complex Fluid Systems, Volume 1: Theory and Numerics (Revision 2), EPRI-NP 1850-R.2-V.1, Electric Power Research Institute, Palo Alto, Calif, 1984.
[2] NUREG/CR-6724, TRAC-M/FORTRAN 90 (Version 3.0) Theory Manual, U.S.Nuclear Regulatory Commission, 2001.
[3] NUREG/CR-5535, RELAP5/MOD3.3 Code Manual Volume I, U.S.Nuclear Regulatory Commission, 2001.
[4] ML120060218, TRACE V5.0 Theory Manual, U.S. Nuclear Regulatory Commission. USNRC, 2010.
[5] INL/EXT-14-31366 (Revision 2), RELAP-7 Theory Manual, U.S. Idaho National Laboratory, 2016.
[6] P. Emonot, A. Souyri, J.L. Gandrille, F. Barre, CATHARE- 3: a new system code for thermal-hydraulics in the context of the NEPTUNE project, Nucl. Eng. Des. 241 (2011) 4476–4481.
[7] D. Bertolotto, A. Manera, Improvement of the one-dimensional dissolved-solute convection equation using the QUICKEST–ULTIMATE algorithm, Nucl. Eng. Des. 241 (2011) 245–256.
[8] D. Wang, J.H. Mahaffy, J. Staudenmeier, C.G. Thurston, Implementation and assessment of high-resolution numerical methods in TRACE, Nucl. Eng. Des. 263 (2013) 327–341.
[9] C. Frepoli, J.H. Mahaffy, K. Ohkawa, Notes on the implementation of a fully-implicit numerical scheme for a two-phase three-field flow model, Nucl. Eng. Des. 225 (2003) 191–217.
[10] V.A. Mousseau, Implicitly balanced solution of the two-phase flow equations coupled to nonlinear heat conduction, J. Comput. Phys. 200 (2004) 104–132.
[11] L. Zou, H.H. Zhao, H.B. Zhang, Applications of high-resolution spatial discretization scheme and Jacobian-free Newton–Krylov method in two-phase flow problems, Ann. Nucl. Energy 83 (2015) 101–107.
[12] L. Zou, H.H. Zhao, H.B. Zhang, Solving phase appearance/disappearance two-phase flow problems with high resolution staggered grid and fully implicit schemes by the Jacobian-free Newton–Krylov Method, Comput. Fluid 129 (2016) 179–188.
[13] L. Zou, H.H. Zhao, H.B. Zhang, Implicitly solving phase appearance and disappearance problems using two-fluid six-equation model, Prog. Nucl. Energy 88 (2016) 198–210.
[14] L. Zou, H.H. Zhao, H.B. Zhang, Application of high-order numerical schemes and Newton-Krylov method to two-phase drift-flux model, Prog. Nucl. Energy 100 (2017) 427–428.
[15] A. Ashrafizadeh, C.B. Devaud, N.U. Aydemir2, A Jacobian-free Newton–Krylov method for thermalhydraulics simulations, Int. J. Numer. Methods Fluid. 77 (2015) 590–615.
[16] G.J. Hu, T. Kozlowski, Application of implicit Roe-type scheme and Jacobian-free Newton-Krylov method to two-phase flow problems, Ann. Nucl. Energy 119 (2018) 180–190.
[17] R.A.A. Saleem, T. Kozlowski, Development of accurate and stable two-phase two-fluid model solver, in: International Congress on Advances in Nuclear Power Plants, ICAPP 2014, American Nuclear Society, 2014.
[18] Lian Hu, Deqi Chen, Yanping Huang, et al., JFNK method with a physics-based preconditioner for the fully implicit solution of one-dimensional drift-flux model in boiling two-phase flow, Appl. Therm. Eng. 116 (2017) 610–622.
[19] A. Hajizadeh, H. Kazeminejad, S. Talebi, A new numerical method for solution of boiling flow using combination of SIMPLE and Jacobian-free Newton-Krylov algorithms, Prog. Nucl. Energy 95 (2017) 48–60.
[20] H. Städtke, Gasdynamic Aspects of Two-phase Flow: Hyperbolicity, Wave Propagation Phenomena, and Related Numerical Methods, Wiley-VCH, 2006.
[21] A. Prosperetti, G. Tryggvason, Computational Methods for Multiphase Flow, Cambridge university press, 2009.
[22] R.J. LeVeque, Numerical Methods for Conservation Laws, 1992, ISBN 3-7643-2464-3.
[23] C.W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, Advanced numerical approximation of nonlinear hyperbolic equations (1998) 325–432. Springer, Berlin, Heidelberg.
[24] J. Smit, M.V.S. Annaland, J.A.M. Kuipers, Grid adaptation with WENO schemes for non-uniform grids to solve convection-dominated partial differential equations, Chem. Eng. Sci. 60 (2005) 2609–2619.
[25] D. Knoll, D. Keyes, Jacobian-Free Newton-Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[26] Y.L. Xu, A Matrix Free Newton/Krylov Method for Coupling Complex Multi-Physics Subsystems, Purdue University, 2004.
[27] Behzad R. Ahrabi, Dimitri J. Mavriplis, A scalable solution strategy for high-order stabilized finite-element solvers using an implicit line preconditioner, Comput. Methods Appl. Mech. Eng. 341 (2018) 956–984.
[28] R.C. Mittal, A. HAl-Kurdi, An efficient method for constructing an ILU preconditioner for solving large sparse nonsymmetric linear systems by the GMRES method, Comput. Math. Appl. 45 (2003) 1757–1772.
[29] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[30] G.F. Hewitt, J.M. Delhaye, N. Zuber, Numerical Benchmark Test, Volume 3 of Multiphase Science and Technology, Wiley & Sons Ltd, 1987.