Computers and Fluids 157 (2017) 253–275

Contents lists available at ScienceDirect

Computers and Fluids

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

The dual information preserving method for stiff reacting flows

Li Liu [a,b], Yiqing Shen [a,b,*], Shengping Liu [a,b], Ming Yu [c]

[a] LHD, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China
[c] Institute of Applied Physics and Computational Mathematics, Beijing 100094, China

ARTICLE INFO

Article history:
Received 31 March 2017
Revised 29 August 2017
Accepted 3 September 2017
Available online 4 September 2017

Keywords:
Stiff reacting flow
Dual information preserving method
Numerical perturbation method
Shock-capturing scheme

ABSTRACT

In this paper, we construct a new numerical method to solve the reactive Euler equations to cure the numerical stiffness problem. The species mass equations are first decoupled from the reactive Euler equations, and then they are further fractionated into the convection step and the reaction step. In the convection step, by introducing two kinds of Lagrangian points (cell-point and particle-point), a dual information preserving (DIP) method is proposed to resolve the convection characteristics. In this new method, the information (including the transport value and the relative coordinates to the center of the current cell) of the cell-point and that of the particle-point are updated according to the velocity field. The information of the cell-point in a cell can effectively restrict the incorrect reaction activation caused by the numerical dissipation, while the information of the particle-point can help to preserve the sharp shock front once the strong shock waves are formed. Hence, by using the DIP method, the spurious numerical propagation phenomenon in stiff reacting flows is effectively eliminated. In addition, a numerical perturbation method is also developed to solve the fractional reaction step (ODE equation) to improve the stability and efficiency. A series of numerical examples are presented to validate the accuracy and robustness of the new method.

1. Introduction

In simulating problems governed by the reactive Euler equations, such as combustion and high-speed chemical reaction, the difference between the timescales of reaction and convection, which limits both the time step and grid spacing, may cause the numerical stiffness problems, for example, the spurious numerical propagation phenomenon of the shock waves in flow fields [1–3]. In order to attenuate the influence of the limited time step, the implicit time method or fractional step method is usually used to calculate the reaction ODE equations. However, if the mesh is not fine enough, the time method cannot remove the incorrect reaction activation caused by the spatial discretization, especially in the flows with shock waves. This is because the numerical dissipation introduced to capture shocks smears the shock front and also leads to the reaction activation in incorrect cells. Although the applications of high order shock capturing schemes can effectively reduce the numerical dissipation and sharpen the discontinuity, the incorrect reaction activation and spurious propagation may still occur.

Wang et al. [4] gave a comprehensive overview of the last two decades of efforts contributed to overcome the spurious numerical phenomenon. Since there is difference between timescales of the convection and the reaction, the fractional step method is usually used to solve the reactive Euler equations. Bao and Jin [3,5,6] developed a random projection method in the reaction terms to capture the detonations, but the assumption of a priori stiff source limits the application of this method. Zhang et al. [7] proposed an equilibrium state method (ESM) by using appropriate equilibrium states to activate the stiff source terms. The main defect of the ESM in applications is that it is difficult to determine the equilibrium states, especially in a complex chemical system. Based on the idea of Harten ENO subcell resolution method [8], Chang [9] developed a finite volume ENO method in the convection step, while Wang et al. [4,10] proposed high order finite difference methods with subcell resolution reconstructing the reaction step. However, as pointed out by Yee et al. [11], the subcell resolution method and its nonlinear filter counterparts [12] can delay the onset of the wrong speed of propagation for the stiff detonation problem with coarse grids and moderate stiff source terms, but this kind of method has additional spurious behavior as the grid is refined and the stiffness is further increased.

Ideally, the shock wave front can be regarded as an interface, hence, the interface tracking methods, such as the level set

* Corresponding author at: LHD, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China.
E-mail address: yqshen@imech.ac.cn (Y. Shen).

http://dx.doi.org/10.1016/j.compfluid.2017.09.001
0045-7930/

method, the VOF method and the front tracking method, have been used in the premixed combustion with the instantaneous flame viewed as an infinitely thin interface between fresh and burned gases [13,14], and also used in alleviating the nonphysical phenomena [15] in the simple two-phase detonations by tracking the inert shock as an interface. However, since the general chemical dynamic model is composed of multi-species and finite rate reactions, there are continuous reacting regions other than a traditional two-phase interface, hence, these interface tracking methods mentioned above cannot solve the stiff problem generated in general reacting flows well.

For solving the interface/free surface fluid flow problems, the marker and cell (MAC) method is regarded as the basis of the interfacial tracking techniques [16]. The essence of the MAC method is the Lagrangian virtual marker particles and the cells defined on an Eulerian grid. Marker particles, often as many as 16 per cell, are moved from their coordinates at time t_n to their new coordinates at time t_{n+1} according to the newly computed velocity u at the cell center. The cell classification is updated every time step using information provided by the virtual Lagrangian mesh constituted by the marker particles. The MAC method has been applied to interface/free surface flow problems successfully [17–19]. The main advantages of the MAC method are that it eliminates all logic problems associated with interfaces and readily extended to three-dimension. However, because a large number of particle coordinates must be stored, the storage increases significantly. Another limitation in the MAC method as well as in the level set and the VOF methods is that it is difficult to extend to the case that the interface (free surface) is generated by the flow itself, such as the shock waves and the chemical reaction.

Recent years, the Lagrangian–Eulerian (LE) approaches with the combination of Lagrangian particles and the Eulerian background grids have attracted great attention in solving the convection-diffusion problems [20–24]. The LE method takes advantage of appropriate operator splitting techniques to solve different aspects of the physical model with most suitable Lagrangian or Eulerian formalism [24]. Shipilova et al. [25] applied a LE method (the particle transform method) to solve the convection-diffusion-reaction problems, numerical results showed that the PTM can avoid the numerical oscillation even for a very sparse grid. So far, there is no attempt to use the LE approaches to deal with the spurious numerical propagation phenomenon generated in simulating the reacting flows.

In this paper, by introducing two kinds of Lagrangian points, we propose the dual information preserving method to cure the spurious numerical propagation in the chemical reacting flows. In this method, the information includes the transport value and the relative coordinates to the center of the Eulerian cell containing the point. The species mass fraction equations are first decoupled from the reactive Euler equations, and then they are further fractionated into the convection step and reaction step. In the species convection step, one Lagrangian particle-point is introduced in each cell at the beginning of the whole computation, and one Lagrangian cell-point is introduced in each cell at the beginning of each time step. All the particle-points are tracked in the whole computation, and the information of the cell-point is determined as: if there are particle-points in current cell, the information is updated by averaging all the particle-points' information; else if there are cell-points entered, the information is updated by averaging all the entered cell-points' information; otherwise a new cell-point is set at the cell center and its transport value is obtained by interpolating its neighboring cell-points' values. The information of the cell-point in a cell can effectively restrict the incorrect reaction activation possibly caused by the numerical dissipation, while the information of the particle-point can help to preserve the sharp shock front once the strong shock waves are formed. Hence, the new method can effectively eliminate the spurious numerical propagation phenomenon in the stiff reacting flows. Different from the MAC method, the new method does not need cell classification and has only two times of the cell number's points to be stored. As it contains information on two kinds of Lagrangian points, we call the method as dual information preserving (DIP) method. In addition, by multiplying a power-series of the time step to the explicit Euler scheme, a numerical perturbation method is developed to solve the fractional reaction step (ODE equation) to improve the stability and efficiency.

This paper is organized as follows. In Section 2, we briefly introduce the decoupling method for solving the reactive Euler equations. In Section 3, a dual information preserving method is proposed to solve the convection step of species mass fraction equations. In Section 4, a numerical perturbation method is developed to solve the fractional reaction step, and analysis of stability and numerical examples are also presented. A series of examples, including one- and two-dimensional problems, simplified reaction and multi-species reaction models, are given to validate the accuracy and robustness of the new method in Section 5. Conclusions are shown in Section 6.

2. Spurious reaction activation and the decoupling method for reactive Euler equations

The one-dimensional governing equations of reacting flows without consideration of heat conduction and viscosity are the reactive Euler equations given as

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=S, \tag{2.1}$$

where

$$U=\begin{pmatrix}\rho\\ \rho u\\ E\\ \rho z_1\\ \rho z_2\\ \vdots\\ \rho z_{ns-1}\end{pmatrix}, F=\begin{pmatrix}\rho u\\ \rho u^2+p\\ u(E+p)\\ \rho u z_1\\ \rho u z_2\\ \vdots\\ \rho u z_{ns-1}\end{pmatrix}, S=\begin{pmatrix}0\\ 0\\ 0\\ \omega_1\\ \omega_2\\ \vdots\\ \omega_{ns-1}\end{pmatrix},$$

ns is the number of reaction species, z_i and ω_i are the mass fraction and the production rate of the ith species. The mass fraction of the nsth species is given by

$$z_{ns}=1-\sum_{i=1}^{ns-1} z_i.$$

And the pressure is

$$p=(\gamma-1)\left(E-\frac{1}{2}\rho u^2-\rho\sum_{i=1}^{ns} q_i z_i\right),$$

where q_i is the formation enthalpy of species i and the ratio of specific heats γ is independent of temperature in this paper.

In the shock-dominant reaction flows, such as the detonation flows, the reaction follows the shock with a process of reaction heat release. Obviously, if a shock-capturing method is used, the numerical dissipation of the reaction is unavoidable. Especially if the computational grid is not fine enough, then the numerical dissipation results in a relatively thick reaction region with large temperature variation, and hence the reaction may be activated at a wrong location. In order to avoid the presence of the wrong reaction, a natural idea is to deal with the shock and the reaction, separately.

Usually, a fractioned method for Eq. (2.1) is as following,

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=0, \tag{2.2}$$

$$\frac{dU}{dt}=S, \tag{2.3}$$

Actually, the fractioned method can handle some stiff problems caused by the difference between the time scales of convection and reaction. However, the method hardly alleviates the wrong activation caused by the numerical dissipation, because the mass fractions of species and also the temperature have been contaminated by solving the Eq. (2.2). Hence, we propose a different split method. In the new method, Eq. (2.1) is decoupled as

$$\begin{cases}\dfrac{\partial U_1}{\partial t}+\dfrac{\partial F_1}{\partial x}=0,\\ \dfrac{\partial U_2}{\partial t}+\dfrac{\partial F_2}{\partial x}=S_1,\end{cases} \tag{2.4}$$

where,

$$U_1=\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix}, F_1=\begin{pmatrix}\rho u\\ \rho u^2+p\\ u(E+p)\end{pmatrix},$$

$$U_2=\begin{pmatrix}\rho z_1\\ \rho z_2\\ \vdots\\ \rho z_{ns-1}\end{pmatrix}, F_2=\begin{pmatrix}\rho u z_1\\ \rho u z_2\\ \vdots\\ \rho u z_{ns-1}\end{pmatrix}, S_1=\begin{pmatrix}\omega_1\\ \omega_2\\ \vdots\\ \omega_{ns-1}\end{pmatrix}.$$

For the second equation of Eq. (2.4), there is

$$Z\frac{\partial \rho}{\partial t}+Z\frac{\partial \rho u}{\partial x}+\rho\frac{\partial Z}{\partial t}+\rho u\frac{\partial Z}{\partial x}=S_1, \tag{2.5}$$

where, $Z=(z_1,\quad z_2,\quad \cdots,\quad z_{ns-1})^T$.

Applying the total mass equation, Eq. (2.5) is written as

$$\frac{\partial Z}{\partial t}+u\frac{\partial Z}{\partial x}=S_e, \tag{2.6}$$

where, $S_e=S_1/\rho=(\omega_1/\rho,\quad \omega_2/\rho,\quad \cdots,\quad \omega_{ns-1}/\rho)^T$.

Similar to the fractioned method, Eq. (2.6) is further split into two parts, i.e., the convection equation of mass fraction (2.8) and the reaction equation (2.9). Hence, the splitting method results in the following equations,

$$\begin{cases}\dfrac{\partial U_1}{\partial t}+\dfrac{\partial F_1}{\partial x}=0, & (2.7)\\ \dfrac{\partial Z}{\partial t}+u\dfrac{\partial Z}{\partial x}=0, & (2.8)\\ \dfrac{dZ}{dt}=S_e. & (2.9)\end{cases}$$

Obviously, Eq. (2.7) is the Euler equations, and can be solved by using the conventional numerical methods. In this paper, the Lax–Friedrichs flux splitting, the fifth-order weighted essentially non-oscillatory (WENO) scheme [26] and the fourth-order Runge–Kutta scheme [27] are used for the spatial discretization and temporal discretization, respectively. Eq. (2.8) is only related to the velocity field and the distribution of the mass fraction at last time level, and hence a new method (DIP), which can effectively avoid the spurious numerical propagation phenomenon in stiff reacting flows, is developed to solve Eq. (2.8) in the next section. Eq. (2.9) is an ordinary differential equation, and a numerical perturbation (NP) is also developed to solve it to improve the stability in this paper.

The framework for solving the whole governing equations is shown in Fig. 1. The variables $\hat{U}_1$ and $\hat{Z}$ are the intermediate values. E, A and R denote the operators for solving the Euler equations (2.7), the convection equation (2.8) and the reaction equation (2.9) on the time interval $[t_n, t_{n+1}]$, respectively.

3. Dual information preserving method

In this section, a new method is proposed to cure the stiff problem caused by the numerical dissipation generated by the strong shock capturing method used in the chemical reacting flows.

The two dimensional convection equation

$$\frac{\partial z}{\partial t}+u(z)\frac{\partial z}{\partial x}+v(z)\frac{\partial z}{\partial y}=0, \tag{3.1}$$

is solved by the Dual Information Preserving method to be described in the following. Before going into detail, there are two comments for this method:

Dual information: means the information on two different kinds of Lagrangian points (cell-points and particle-points).

Information: includes the transport value and the relative coordinates to the center of the Eulerian cell containing the point.

The main idea of the dual information preserving method (DIP) method is described as below. For simplicity, the particle-point is taken as an example to illustrate the description and formula of DIP method, and one can obtained those of the cell-point by omitting the subscript p if there is no further explanation. The detailed algorithm is given in Appendix A.

As preparations, at the beginning of the computation, one Lagrangian particle-point and one Lagrangian cell-point are introduced for each Eulerian cell. The position of the Lagrangian point is at the center of the Eulerian cell and the transport value is the initial value.

For example, for the particle-points, the relative coordinates and the transport value are set as

$$\begin{cases}X_p(i,j)=0,\\ Y_p(i,j)=0,\\ \bar{z}_p(i,j)=z(i,j).\end{cases}$$

Note that, since the cell-points are transported only in each time step, while the particle-points are tracked in the whole computation, the index (ip, jp) is actually $(ip(i,j), jp(i,j))$, and at the beginning

$$\begin{cases}ip(i,j)=i,\\ jp(i,j)=j.\end{cases}$$

In the procedure of DIP method, the value of $(ip(i,j), jp(i,j))$ also denotes which cell the initial particle-point (i,j) is located in now.

(1) Get the velocities of the points (particle-point and cell-point) by using an interpolation (in this paper only the 1D linear interpolation is used) of the velocities of the current cell and its neighbour cell.

For example, for a particle-point (i,j) (i.e., initially set at (i,j)), the formula for calculating velocities are

$$\begin{cases}\bar{u}_p(i,j)=(1-|X_p|)u(ip,jp)+|X_p|u(ip+s_x,jp),\\ \bar{v}_p(i,j)=(1-|Y_p|)v(ip,jp)+|Y_p|v(ip,jp+s_y),\end{cases}$$

where,

$$\begin{cases}s_x=\mathrm{sign}(X_p(ip,jp)),\\ s_y=\mathrm{sign}(Y_p(ip,jp)).\end{cases}$$

The velocity interpolation of a cell-point in x-direction is showed in Fig. 2a.

(2) Update the position of two kinds of points by using the corresponding velocities and the time step Δt.

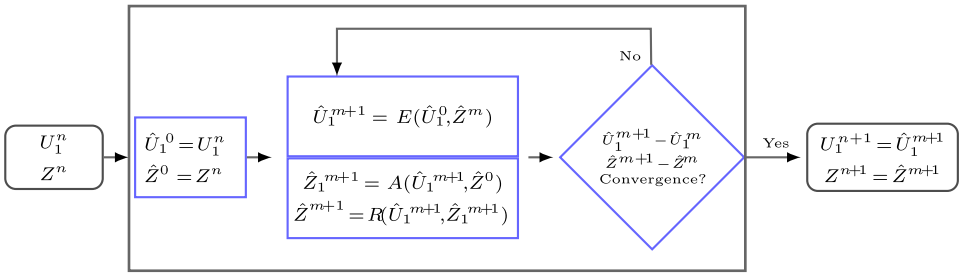

Fig. 1. The framework of the solving process for the decoupling method.

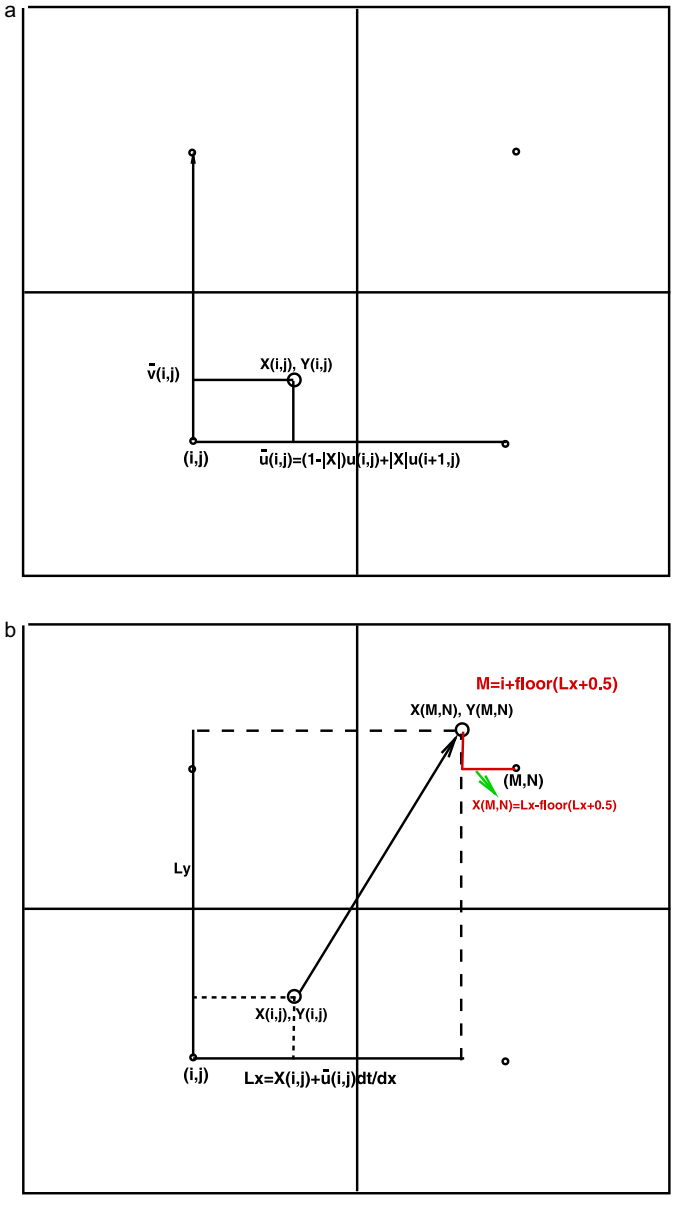

Fig. 2. (a) Sketch of the velocity interpolation in x-direction of cell-point (i, j). (b) Sketch of the movement process of cell-point (i, j).

For example, for a particle-point (i, j), the relative coordinates are changed to

$$\begin{cases} L_x = X_p(i,j) + \bar{u}_p(i,j)\Delta t/\Delta x, \\ L_y = Y_p(i,j) + \bar{v}_p(i,j)\Delta t/\Delta y. \end{cases}$$

Hence its cell is changed to

$$\begin{cases} ip(i,j) = ip(i,j) + \text{floor}(L_x + 0.5), \\ jp(i,j) = jp(i,j) + \text{floor}(L_y + 0.5), \end{cases}$$

with the new relative coordinates

$$\begin{cases} X_p(i,j) = L_x - \text{floor}(L_x + 0.5), \\ Y_p(i,j) = L_y - \text{floor}(L_y + 0.5). \end{cases}$$

The formula of a cell-point in x-direction is illustrated in Fig. 2b.

It should be noted that, for the cell-points, a temporary index (M, N) is used to indicate the cell-point (i, j) moves to the Eulerian cell (M, N).

$$\begin{cases} M = i + \text{floor}(L_x + 0.5), \\ N = j + \text{floor}(L_y + 0.5). \end{cases}$$

In this step, a marker is needed to record the Eulerian cell contains which kind of points (or not any point) for updating the information in the next step.

(3) Update the information (both position and value) of cell-points in three cases:

(i) first of all, if there are particle-points in the current cell, the information is updated by averaging (in this paper, the arithmetical average is used) all the particle-points' information;

(ii) else if there are cell-points in the cell, the information is updated by averaging all the entered cell-points' information;

(iii) otherwise, there is no any kind of points, a new cell-point is set at the cell center as $X(i, j) = 0$ and $Y(i, j) = 0$, and its transport value is obtained by weighting the values of its neighbouring cell-points (if there are),

$$\bar{z}(i,j) = \sum_{i_1,j_1} \left[\omega_{i_1,j_1} \bar{z}(i+i_1, j+j_1)\right] / \sum_{i_1,j_1} \omega_{i_1,j_1},$$

where, $i_1 = -1, 1$ and $j_1 = -1, 1$. And the weight is inversely proportional to the distance between the cell-point $(i + i_1, j + j_1)$ and the center of the cell (i, j), $\omega_{i_1,j_1} = 1/L_{i_1,j_1}$. From the definition of the relative coordinate, it is easy to find the distance as

$$L_{i_1,j_1} = \sqrt{(X(i+i_1, j+j_1) + i_1)^2 + (Y(i+i_1, j+j_1) + j_1)^2}.$$

Specially, for an inlet Eulerian cell, if it has neither particle-point nor cell-point, only a cell-point is reset, and the transport value of the new cell-point is the prescribed value in the inlet.

(4) The transport value of the cell-point is used as the current cell's value, $z(i, j) = \bar{z}(i, j)$.

Clearly, there is only one cell-point in each cell after a time step, and the particle-points are tracked and their transport values are preserved all the time except those moved out of the computation domain. The information of the cell-point in a cell can effectively restrict the incorrect reaction activation possibly caused by the numerical dissipation, while the information of the particle-point can help to preserve the sharp shock front once the strong shock waves are formed.

Note that, in the detailed algorithm of the DIP method given in Appendix A, for convenience in coding, the steps are not strictly followed the above steps (1)–(4), and the order of dealing with cell-points and particle-points shows that if there are both cell-points and particle-points in one cell, then the information is updated only by those particle-points.

It is worthy to point out that, the splitting method Eqs. (2.7)–(2.9) is like Godunov splitting method, hence it has only first-order time accuracy. However, as remarked by Crandall and Majda [28], the second-order Strang splitting method generates even more dispersion near discontinuities than the first-order accurate algorithm, so the first-order splitting method with iteration is implemented for the DIP method in this paper. In addition, the movement of two kinds of points belongs to Lagrangian motion, hence, the accuracy of the relative coordinates is dependent on the accuracy of the velocities solved from the convective equation (2.7) and the linear interpolation of velocities, and hence it is second-order accurate. Finally, the step (4) is only a simple approximation, how to construct a high order formula is still worthy to study for the DIP method.

3.1. Numerical test for the DIP method

In this subsection, we test the capability of the DIP method in discontinuities/interface cases.

Example 3.1. The linear equation

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=0,$$

with the following initial condition is solved as the first example to test the DIP algorithm.

$$u_0(x)=\begin{cases}\frac{1}{6}(G(x,\beta,z-\delta)+G(x,\beta,z+\delta)+4G(x,\beta,z)), & -0.8\leqslant x\leqslant -0.6,\\ 1, & -0.4\leqslant x\leqslant -0.2,\\ 1-|10(x-0.1)|, & 0\leqslant x\leqslant 0.2,\\ \frac{1}{6}(F(x,\alpha,a-\delta)+F(x,\alpha,a+\delta)+4F(x,\alpha,a)), & 0.4\leqslant x\leqslant 0.6,\\ 0, & \text{otherwise},\end{cases} \tag{3.2}$$

and

$$G(x,\beta,z)=\exp(-\beta(x-z)^2),$$

$$F(x,\gamma,a)=\sqrt{\max(1-\alpha^2(x-a^2),0)}.$$

where $a=0.5$, $z=-0.7$, $\delta=0.05$ and $\beta=\log 2/36\delta^2$. The grid number is $N=200$. Fig. 3 gives the comparison of the results of the fifth-order WENO scheme and the DIP method. For the linear equation, as the velocity is constant, the DIP method is a Lagrangian point tracking method, and hence there is no numerical dissipation introduced in the propagation process. The solution obtained by the DIP method is the exact solution.

Example 3.2. The inviscid Burgers equation is calculated as the second example.

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=0,\ u_0(x)=\sin(\pi x),\ 0\leq x\leq 2. \tag{3.3}$$

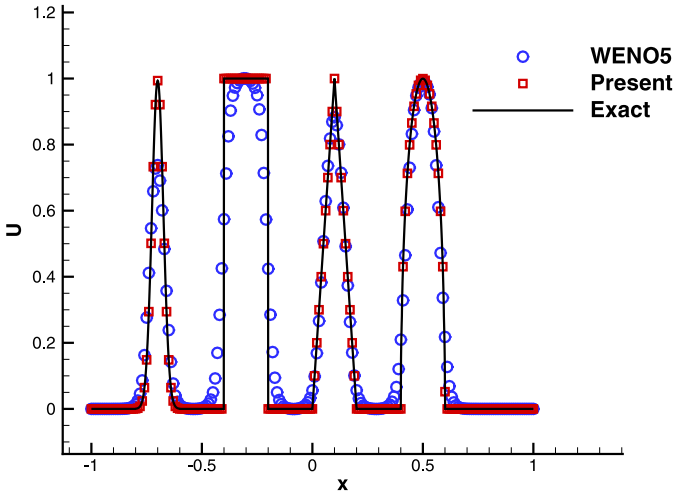

Fig. 3. Numerical solution of Example 3.1, T = 6, CFL = 0.6.

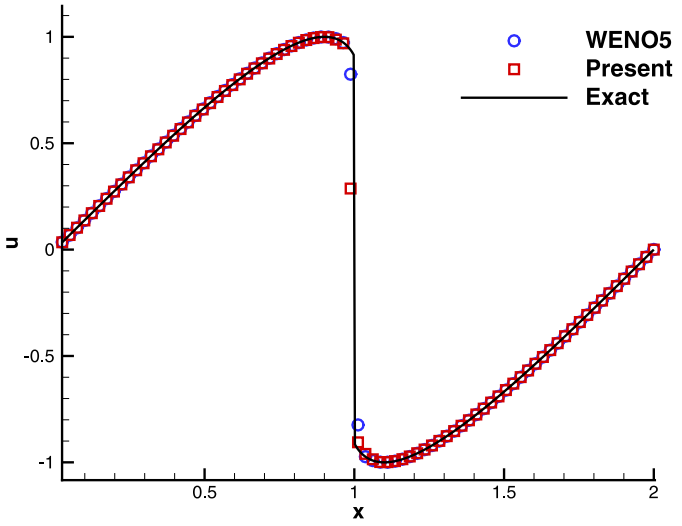

Fig. 4. Numerical solution of Example 3.2, T = 0.4, CFL = 0.6.

Fig. 4 shows the results at $T=0.4$ with $N=200$. It can be seen that, for nonlinear convection equation, even if the initial condition is a smooth solution, the discontinuity is generated with time advancing. The numerical results show that the DIP method can capture this kind of discontinuity well.

Example 3.3. Zalesak's disk [29] is a classical example to exam the capability of interface-tracking of a method [7]. The governing equation is a 2D scalar equation

$$\frac{\partial u}{\partial t}+v_x\frac{\partial u}{\partial x}+v_y\frac{\partial u}{\partial y}=0. \tag{3.4}$$

The velocity field is taken as

$$\begin{cases}v_x(x,y) &= 2\pi y,\\ v_y(x,y) &= -2\pi x,\end{cases}$$

and the initial conditions are

$$u(x,y)=\begin{cases}0, & \sqrt{x^2+y^2}>0.4,\\ 0, & 0.4<y<0.6 \text{ and } x>0.5,\\ 1, & \text{else}.\end{cases}$$

The computation domain is $[0,1]\times[0,1]$. Fig. 5 shows the results at the time ($T_1=0$, $T_2=0.25$, $T_3=0.5$, $T_4=0.75$, $T_5=1$ and $T_6=20$) with $N=200\times 200$. The DIP method can keep the shape of the disk well.

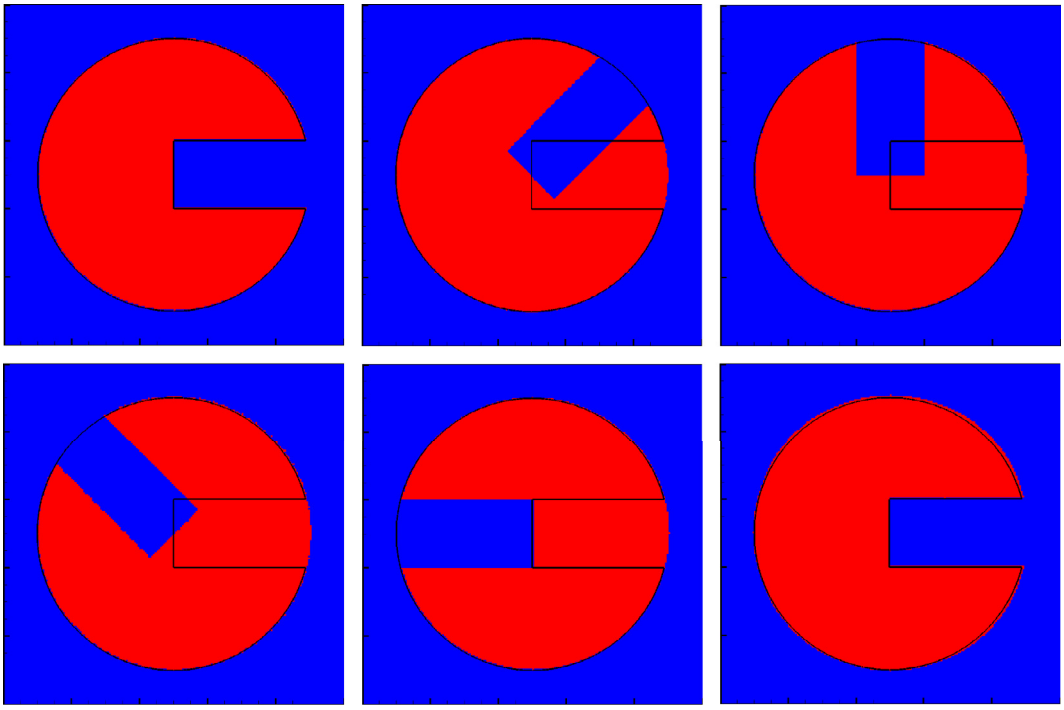

Fig. 5. Numerical results of Example 3.3 with $N = 200 \times 200$ at different time ($T_1 = 0$, $T_2 = 0.25$, $T_3 = 0.5$, $T_4 = 0.75$, $T_5 = 1$ and $T_6 = 20$). The exact initial value is given by the black line.

Example 3.4. Using the same Eq. (3.4), another 2D interfacial problem is calculated [30,31]. At the initial time, a circle with the radius of 0.2 is located at

$$u(x,y) = \begin{cases} 1, & \sqrt{(x-0.5\pi)^2 + (y-0.7)^2} \leqslant 0.2, \\ 0, & \text{else.} \end{cases}$$

The velocity field is taken as

$$\begin{cases} v_x(x,y) &= \cos(x-0.5\pi)\sin(y-0.5\pi), \\ v_y(x,y) &= -\sin(x-0.5\pi)\cos(y-0.5\pi). \end{cases}$$

The computation domain is $[0,\ \pi] \times [0,\ \pi]$. The interface is stretched up to time $T = t/2$, and then is brought back to its initial configuration at time $T = t$. Fig. 6 shows the results at the time $t = 2\pi$ and $t = 8\pi$ with $N = 200 \times 200$. The shape of the circle at the time $T = t$ is in agreement well with the initial shape even after a long time.

4. Numerical perturbation method for reactive ordinary differential equations

The NP method was first proposed by Gao and co-workers [32,33] to solve the convective-diffusion equations. The main idea of constructing the algorithm is as follows: the coefficient of the convective derivative in the basic discretization schemes (the first-order upwind scheme, the second-order central scheme) are reconstructed as a power-series of grid intervals; using the convective-diffusion equation itself, the high order mathematical relation is obtained; by eliminating truncated error terms in the modified differential equation of the reconstructed scheme, the coefficients in the power-series are determined and finally the numerical perturbation algorithms are obtained. In the fractional method, the reaction step forms a set of ordinary differential equations (ODE). In this section, we construct several efficient schemes for solving the reaction ODE based on the idea of numerical perturbation (NP).

4.1. The numerical perturbation schemes

Usually, the ODE equation is given as

$$\frac{dx}{dt} = f(t,x), x(0) = x_0, (x \in \mathbb{R}^s, t \leqslant 0). \tag{4.1}$$

The first-order explicit Euler scheme

$$x_{n+1} - x_n = \Delta t f(t, x_n), \tag{4.2}$$

is taken as the basic discretization scheme for the numerical perturbation. Applying Taylor expansion, we get the modified differential equation of Eq. (4.2) as

$$\frac{dx}{dt} = f(t,x) - \frac{1}{2}\Delta t \frac{d^2x}{dt^2} - O(\Delta t^2). \tag{4.3}$$

Similar to the construction of the numerical perturbation method for convective diffusion equation [32,34], a perturbation polynomial p is used to multiply the left of Eq. (4.2), i.e.,

$$p(x_{n+1} - x_n) = \Delta t f(t, x_n), \tag{4.4}$$

where the polynomial p is defined as

$$p = 1 + \sum_{i=1}^{\infty} a_i \Delta t^i. \tag{4.5}$$

Substituting Eq. (4.5) into Eq. (4.4) and using Taylor expansion, we get

$$\frac{dx}{dt} = f(t,x) - \left(\frac{1}{2}\frac{d^2x}{dt^2} + a_1\frac{dx}{dt}\right)\Delta t - \left(\frac{1}{6}\frac{d^3x}{dt^3} + \frac{a_1}{2}\frac{d^2x}{dt^2} + a_2\frac{dx}{dt}\right)\Delta t^2 + O\left(\Delta t^4\right). \tag{4.6}$$

Clearly, if the second term in the right hand side of Eq. (4.6) becomes zero,

$$\frac{1}{2}\frac{d^2x}{dt^2} + a_1\frac{dx}{dt} = 0, \tag{4.7}$$

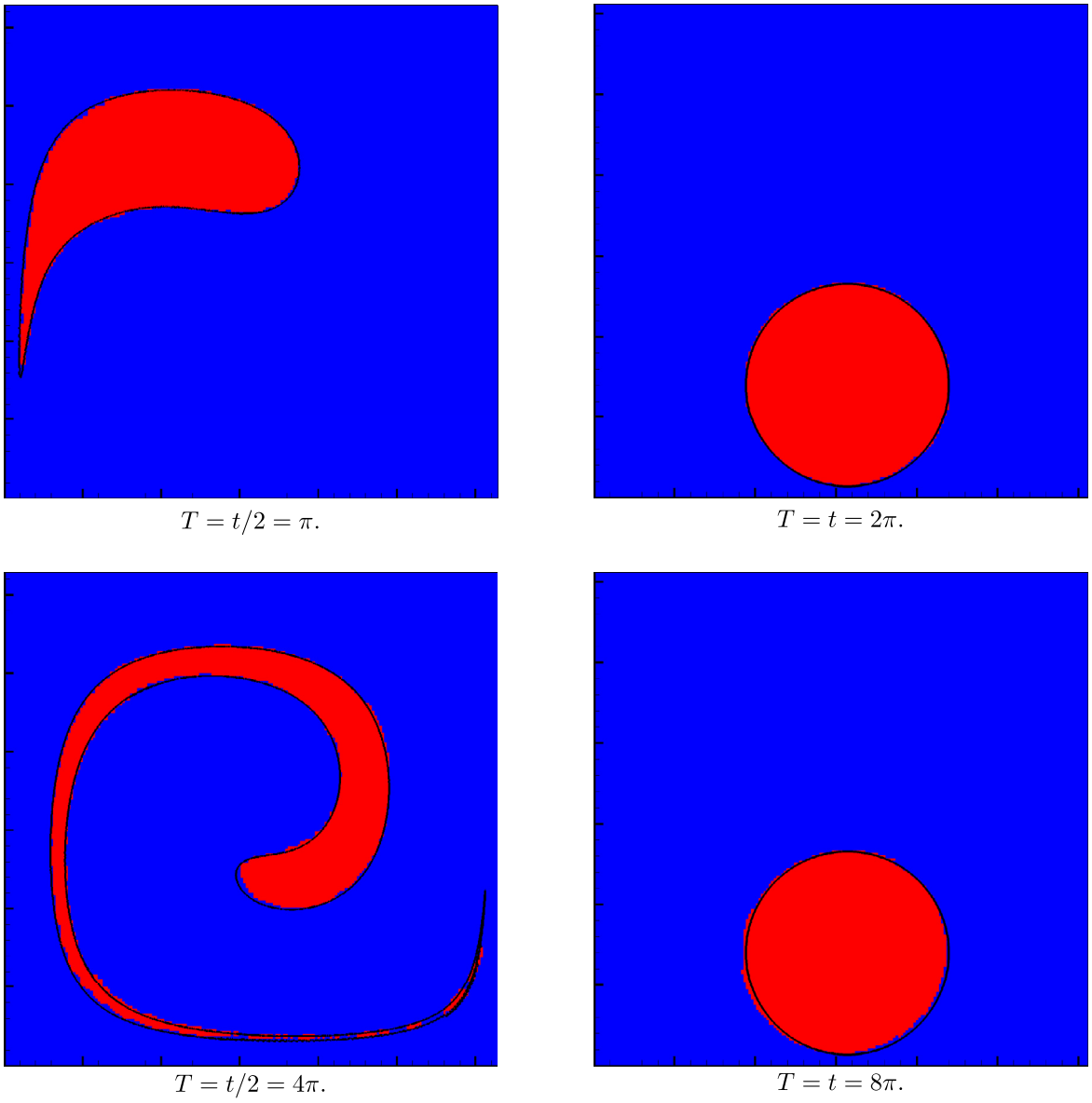

Fig. 6. Numerical results of Example 3.4, $N = 200 \times 200$. The exact value is given by the black line.

then the scheme (4.4) has second-order accuracy. Similarly, we can get higher order schemes by eliminating more terms of Eq. (4.6). Since all derivatives can be calculated by using Eq. (4.1)

$$\frac{dx}{dt} = f, \frac{d^2x}{dt^2} = f'_t + f'_x f, \cdots,$$

if a certain order of the truncation error is expected, we can get the coefficients $a_1, a_2, \cdots$ as

$$a_1 = -\frac{f'_t + f'_x f}{2f},$$

$$a_2 = \frac{-2f(f''_{tt} + 2f''_{tx}f + f'_x f'_t + f'_x f'_x f + f''_{xx}f^2) + 3(f'_t + f'_x f)^2}{12f^2},$$

$\cdots$

Usually, a better way to approximate $a_1, a_2, \cdots$ is using finite difference scheme of numerical solution. Specially, if f is the function only respecting to x, these coefficients a_i have simple formulas as follows,

$$a_1 = -\frac{f'}{2}, \quad a_2 = \frac{1}{12}f'^2 - \frac{1}{6}f''f, \quad \cdots. \tag{4.8}$$

For expressing conveniently, a Nth-order NP scheme for the Eq. (4.1) can be written as

$$x_{n+1} = x_n + \Delta x f(t, x_n)/p_N, \tag{4.9}$$

and the perturbation polynomial for the Nth-order NP scheme is denoted as

$$p_N = 1 + \sum_{i=1}^{N-1} a_i \Delta t^i.$$

In addition, we construct a transformed function to replace the original perturbation polynomial to improve the stability of the third-order NP(3NP) scheme. The function can be expressed as

$$\overline{p}_3 = \frac{1 + b_1\Delta t + b_2\Delta t^2}{1 - b_2\Delta t}, \tag{4.10}$$

Requiring $\overline{p}_3$ to be a second-order approximation of p_3 we get

$$b_1 = a_1 - \frac{a_2}{a_1 + 1}, \quad b_2 = \frac{a_2}{a_1 + 1}. \tag{4.11}$$

The new third-order transformed NP (3TNP) scheme with the transformed function $\overline{p}_3$ has third-order accuracy, but its stability

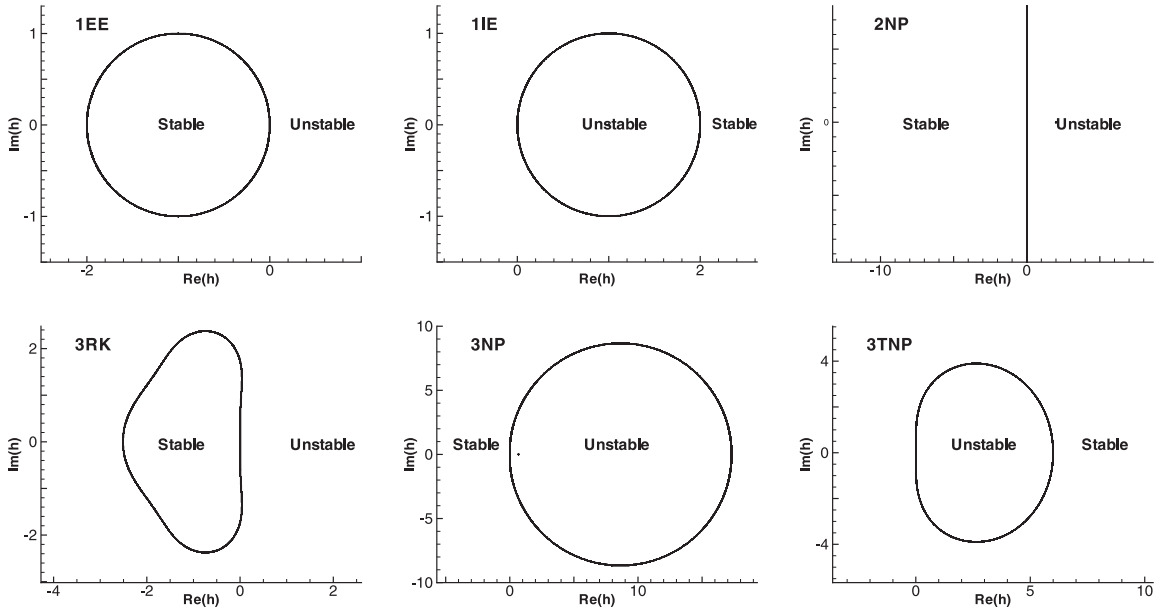

Fig. 7. Region of the stability in complex h plane for different schemes.

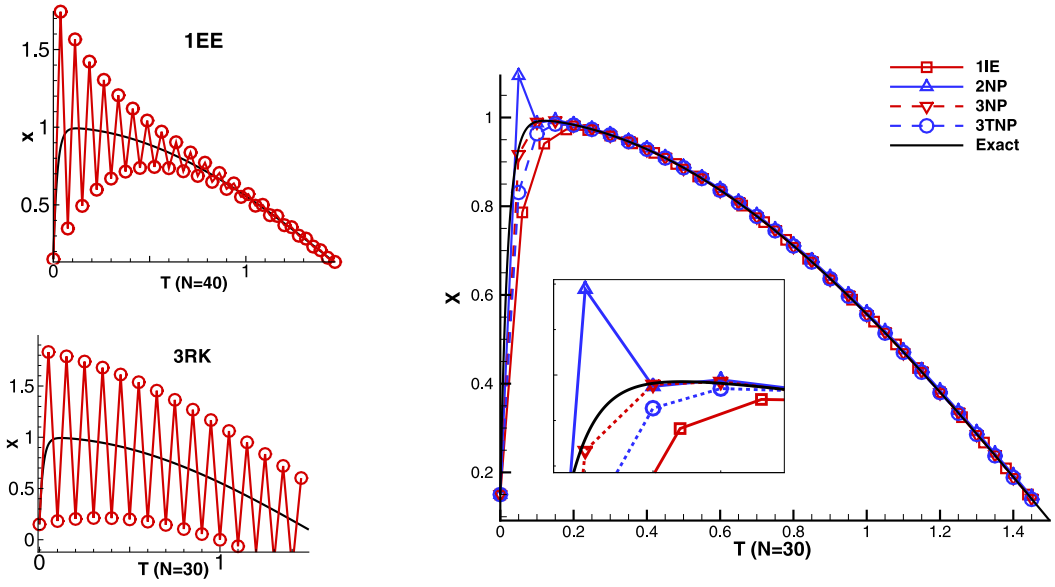

Fig. 8. Numerical results of Example 4.1. Exact solution with $N = 3000$.

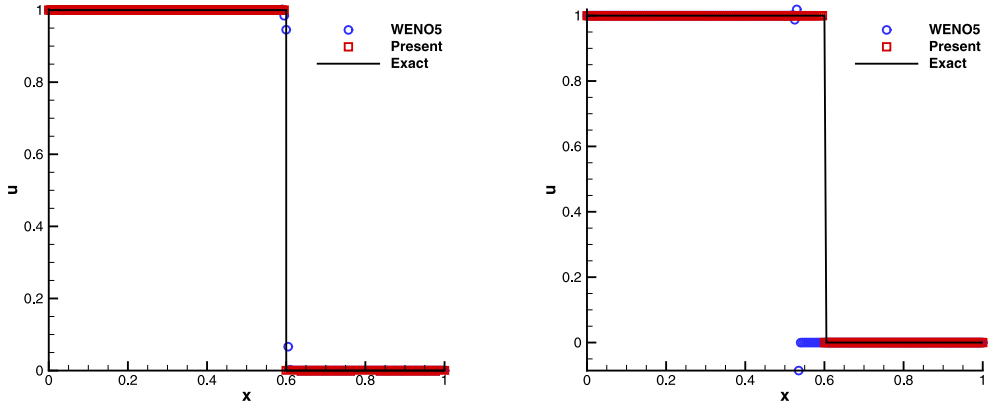

Fig. 9. The numerical results of Example 5.1, $t = 0.3$. Left:the non-stiff case; right: the stiff case.

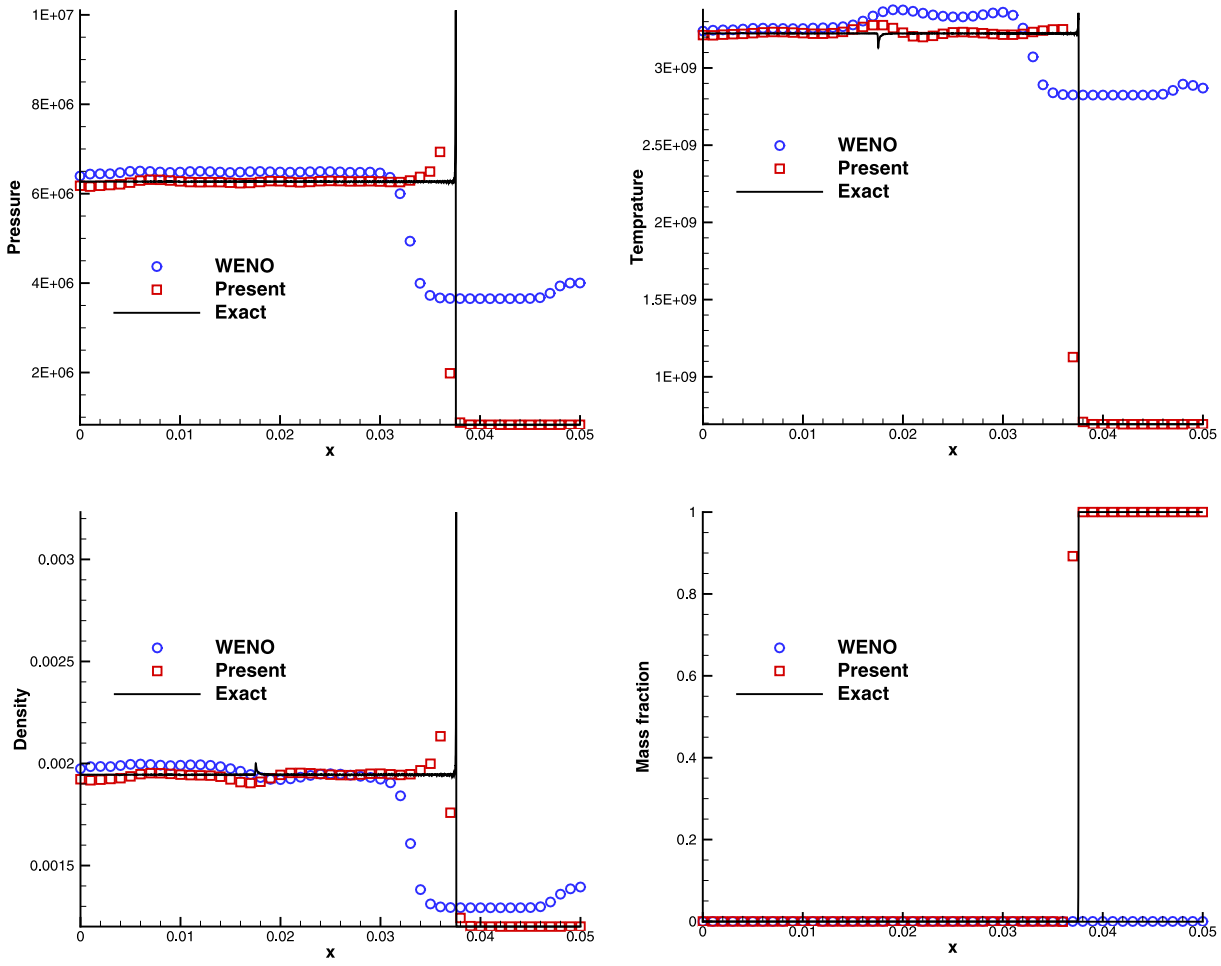

Fig. 10. Numerical results of Example 5.2, $T = 3 \times 10^{-7}$, $N = 50$.

region is larger than the 3NP scheme. The analysis and comparison will be given in the next subsection.

4.2. The stability analysis of the NP schemes

The stability is necessary and important for a scheme to solve the system of ODE equations with stiffness [35]. Generally, the scalar equation

$$x' = qx, Re(q) < 0, \tag{4.12}$$

is used to study the stability. For a scheme, the solution of Eq. (4.12) can be expressed as

$$x_{n+1} = E(h)x_n, \tag{4.13}$$

where $h = q\Delta t$. The A-stability was proposed and used to analyze a numerical scheme in Refs. [36–38].

Two definitions for A-stability are given in Ref. [38]:

Definition 1 (A-stable)**.** A scheme is A-stable, in the sense of Dahlquist [36], if $|E(h)| < 1$ for all complex h with negative real part.

Definition 2 (Strong A-stable)**.** A scheme is strongly A-stable, if it is A-stable and $\lim_{Re(h)\to-\infty} |E(h)| = 0$.

In order to show the performance of the stability of NP schemes, several conventional schemes, include the first-order explicit Euler scheme (1EE), the first-order implicit Euler scheme (1IE), the second-order linearized implicit Euler scheme (2LIE) and the third-order explicit Runge–Kutta scheme (3RK), are analyzed and compared.

(1) The first-order explicit Euler scheme

$$x_{n+1} - x_n = \Delta t f(t, x_n), \tag{4.14}$$

and

$$E^{1EE}(h) = 1 + h. \tag{4.15}$$

(2) The first-order implicit Euler scheme

$$x_{n+1} - x_n = \Delta t f(t, x_{n+1}), \tag{4.16}$$

and

$$E^{1IE}(h) = \frac{1}{1-h}. \tag{4.17}$$

(3) The second-order linearized trapezoidal method [4,7]

$$x_{n+1} - x_n = \frac{\Delta t f(t, x_n)}{1 - 1/2\Delta t f'_x(t, x_n)}, \tag{4.18}$$

and

$$E^{2LIE}(h) = \frac{1 + \frac{1}{2}h}{1 - \frac{1}{2}h}. \tag{4.19}$$

Notice that, it has the same form as the 2NP scheme.

(4) The third-order explicit Runge–Kutta scheme

$$x_{n+1} = x_n + \frac{1}{4}k_1 + \frac{3}{4}k_3,$$
$$k_1 = \Delta t f(t_n, x_n),$$

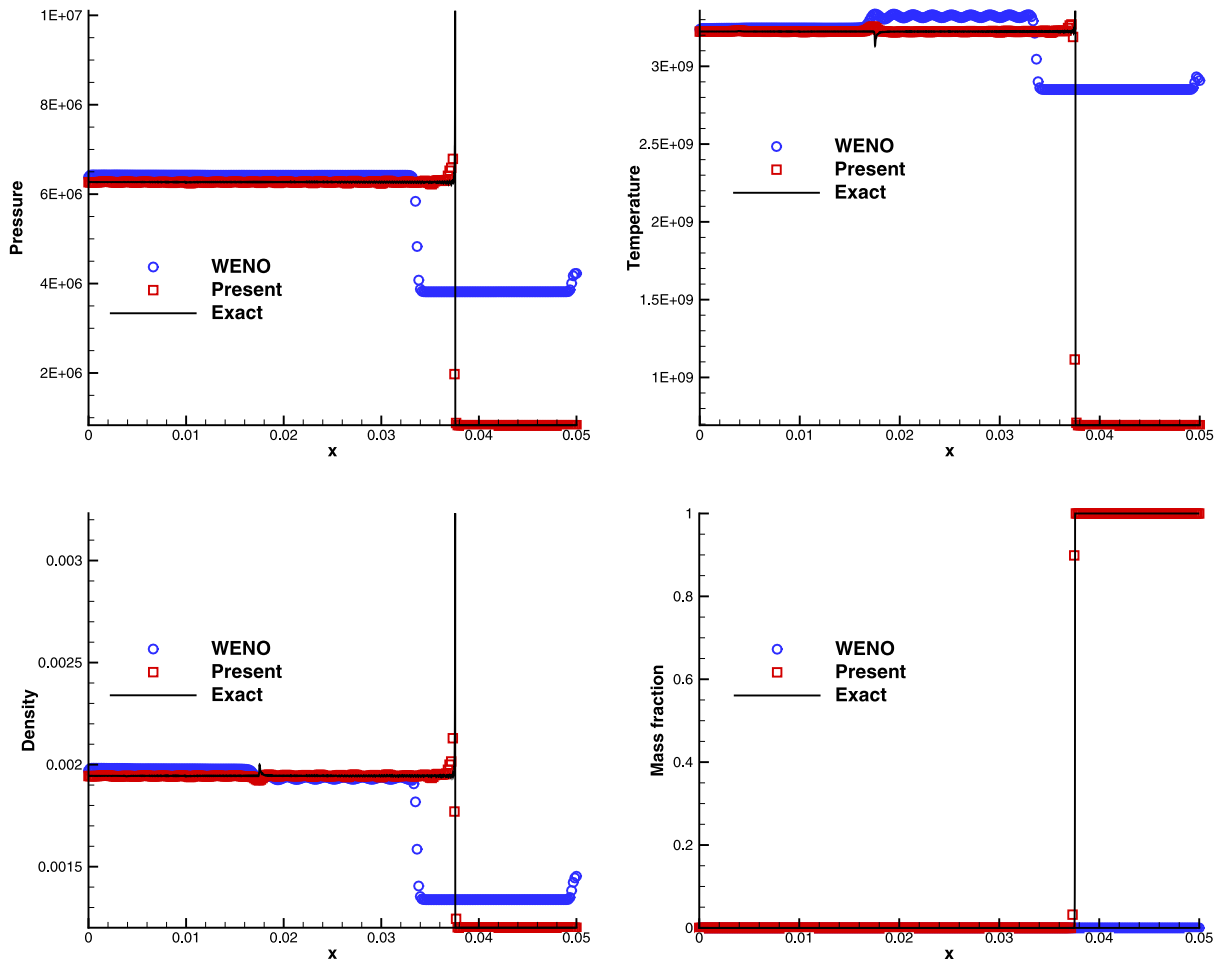

Fig. 11. Numerical results of Example 5.2, $T = 3 \times 10^{-7}$, $N = 300$.

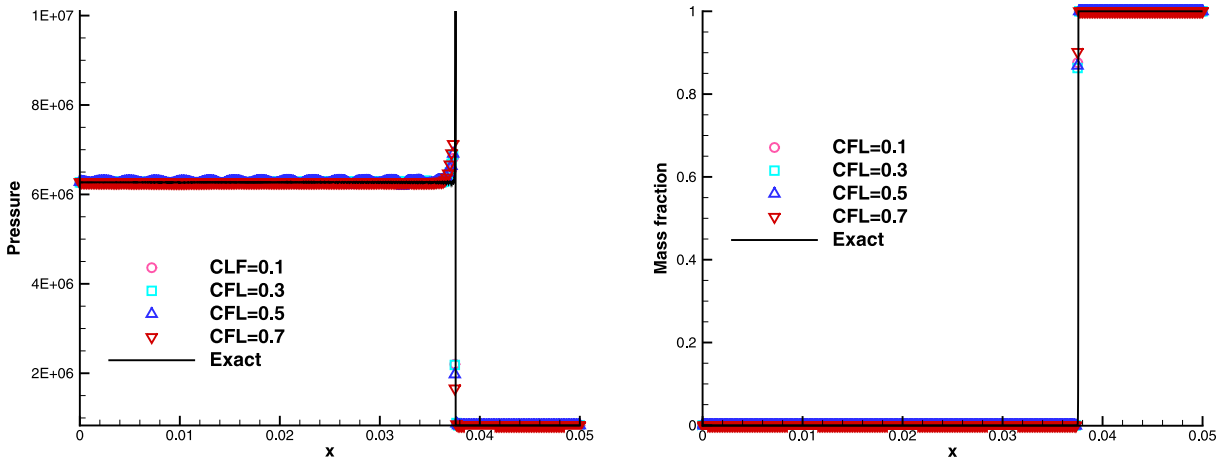

Fig. 12. Numerical results of Example 5.2 with $100/\epsilon$, $T = 3 \times 10^{-7}$, $N = 300$.

$$k_2 = \Delta t f\left(t_n + \frac{1}{3}\Delta t, x_n + \frac{1}{3}k_1\right),$$
$$k_3 = \Delta t f\left(t_n + \frac{2}{3}\Delta t, x_n + \frac{2}{3}k_2\right), \tag{4.20}$$

and

$$E^{3\mathrm{RK}}(h) = 1 + h + \frac{1}{2}h^2 + \frac{1}{6}h^3. \tag{4.21}$$

For Eq. (4.12), it's easy to find the perturbation coefficients as $a_1 = -\frac{1}{2}q$, $a_2 = \frac{1}{12}q^2$, $\cdots$. Hence, the functions $E(h)$ for the NP schemes are

$$E^{2\mathrm{NP}}(h) = \frac{1 + \frac{1}{2}h}{1 - \frac{1}{2}h},$$
$$E^{3\mathrm{NP}}(h) = \frac{1 + \frac{1}{2}h + \frac{1}{12}h^2}{1 - \frac{1}{2}h + \frac{1}{12}h^2},$$
$$E^{3\mathrm{TNP}}(h) = \frac{1 + \frac{1}{3}h}{1 - \frac{2}{3}h + \frac{1}{6}h^2}.$$

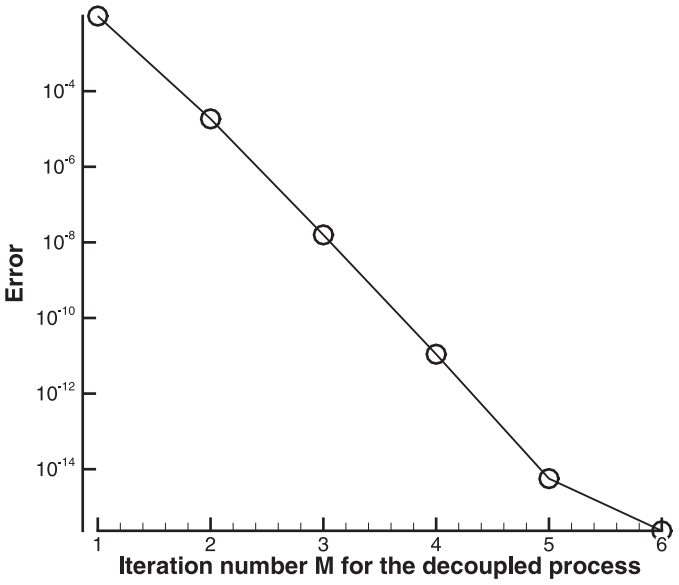

Fig. 13. Convergence history of Example 5.2.

Fig. 7 gives the stable region for different schemes in the complex h plane. It shows that the first-order implicit, the second-order perturbation, the third-order NP and the transformed third-order NP schemes are A-stable. The transformed third-order NP scheme has a larger stable region than its counterpart, moreover, only this scheme and the first-order implicit Euler scheme are strongly A-stable schemes.

4.3. Numerical comparison of different schemes

Before being used to solve chemical equations, the stability and accuracy of perturbation schemes are tested and compared with other schemes used in ODEs.

Example 4.1.

$$\frac{dx}{dt} = f(t,x) = -50(x - \cos t). \tag{4.22}$$

This equation is calculated by Hairer and Wanner [37] and used as the first case in this section. The perturbation coefficients in Eq. (4.8) are

$$a_1 = 25 - \frac{\sin t_n}{2(x_n - \cos t_n)}, \quad a_2 = \frac{1}{4}\left(\frac{\sin t_n}{x - \cos t_n} - 50\right)^2 - \frac{\cos t_n - 50 \sin t_n}{6(x - \cos t_n)}.$$

Fig. 8 shows the results of different schemes. We can see the 1EE scheme and the 3RK schemes are not stable to solve Eq. (4.22). While the 2NP scheme has one point overshot. The 3NP, the 3TNP and the 1IE schemes are stable. It also can be seen that the 3NP and the 3TNP are more accuracy than the 1IE scheme. It should be noted that, although the implicit Euler scheme can get a stable solution, they need iteration in every time step.

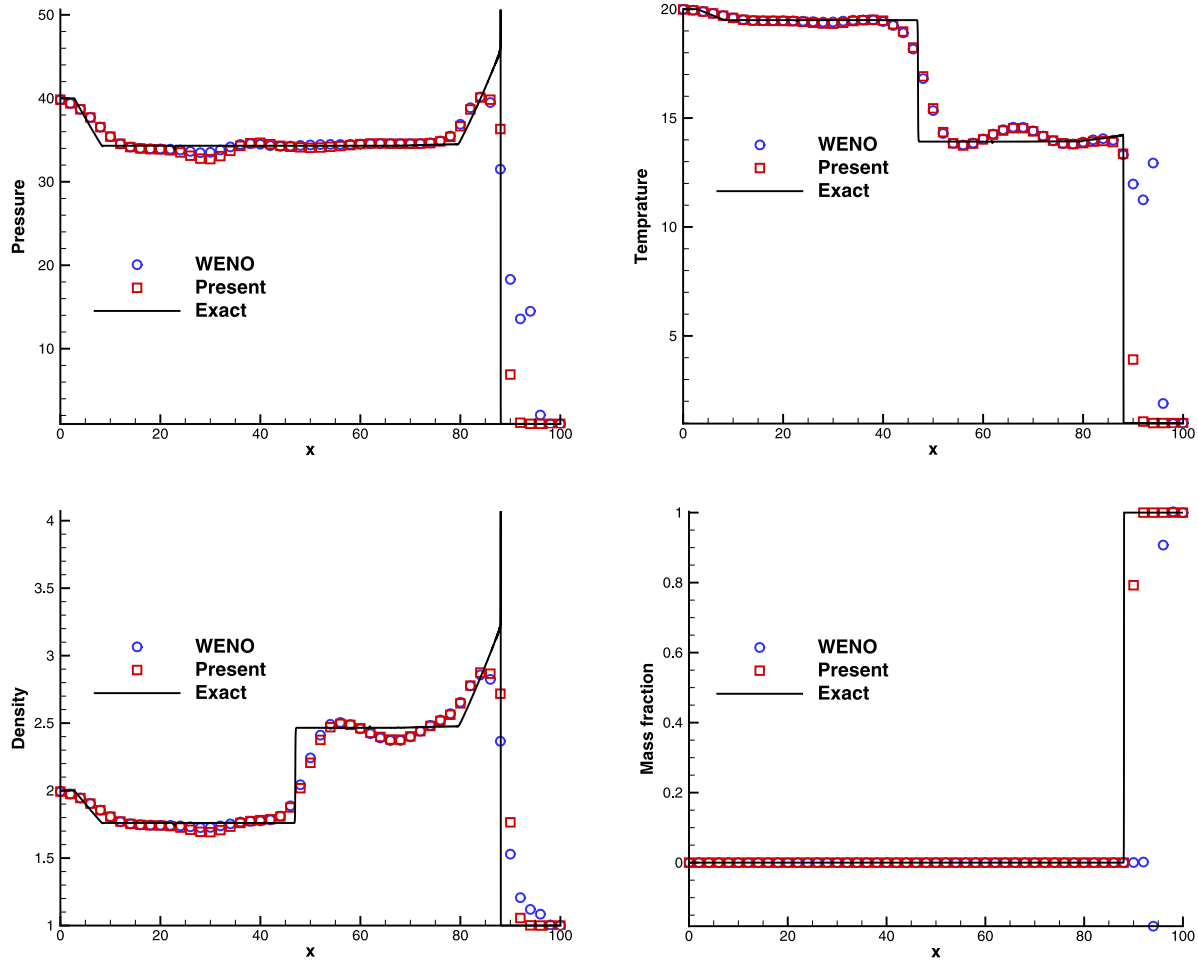

Fig. 14. Numerical results of Example 5.3, $T = 8$, $N = 50$.

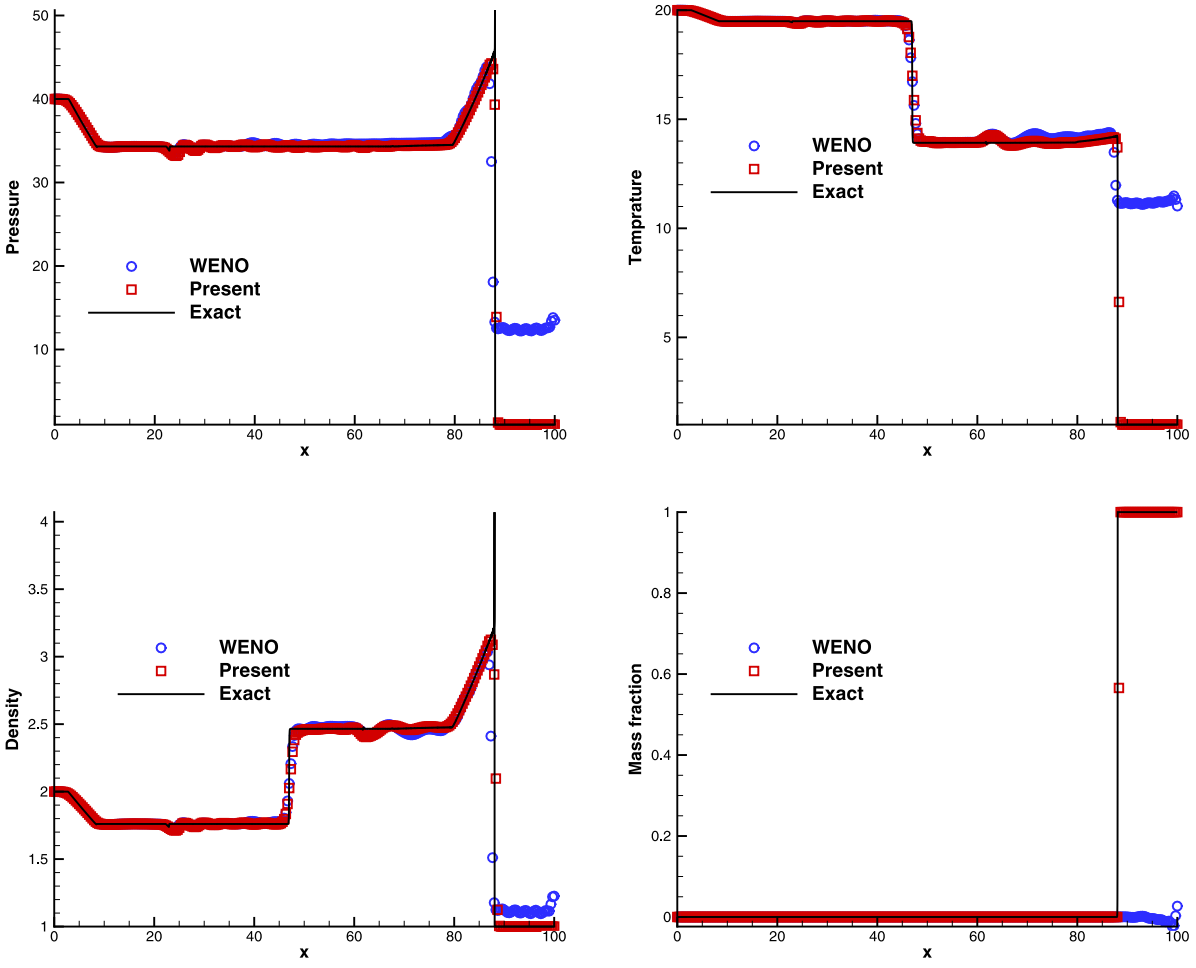

Fig. 15. Numerical results of Example 5.3 with $100/\epsilon$, $T = 8$, $N = 300$.

Example 4.2. The equation

$$\frac{dx}{dt} = f(t,x) = -x^3, x_0 = 1, t = [0,1], \tag{4.23}$$

is calculated as the second case. In this case, the analytic solution is

$$x = \frac{1}{\sqrt{2t+1}}.$$

The perturbation coefficients of the second-order and the third-order schemes are

$$a_1 = \frac{3x_n^2}{2}, a_2 = -\frac{x_n^4}{4}.$$

Table 1 gives the errors and accuracy orders of different schemes. It shows that the second-order, the third-order and the transformed NP schemes can get their theoretical accuracy orders. The errors of the third-order NP and transformed NP schemes are lower than the third-order Runge–Kutta schemes, though all of them are third-order accuracy.

Table 1
The accuracy of different schemes used in Example 4.2.

Scheme	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1IE	20	7.7149d-3	–	8.7737d-3	–
	40	3.9062d-3	0.98	4.4879d-3	0.97
	80	1.9656d-3	0.99	2.2712d-3	0.98
	160	9.8594d-4	1.00	1.1426d-3	0.99
	320	4.9377d-4	1.00	5.7304d-4	1.00
3RK	20	1.3696d-5	–	1.7438d-5	–
	40	1.6161d-6	3.08	2.0646d-6	3.08
	80	1.9601d-7	3.04	2.5119d-7	3.04
	160	2.4128d-8	3.02	3.0964d-8	3.02
	320	2.9924d-9	3.01	3.8433d-9	3.01
2NP	20	5.0683d-5	–	6.0346d-5	–
	40	1.2338d-5	2.04	1.4812d-5	2.03
	80	3.0414d-6	2.02	3.6679d-6	2.01
	160	7.5487d-7	2.01	9.1226d-7	2.01
	320	1.8803d-7	2.01	2.2750d-7	2.00
3NP	20	1.9980d-6	–	2.5388d-6	–
	40	2.3999d-7	3.06	3.0629d-7	3.05
	80	2.9376d-8	3.03	3.7628d-8	3.03
	160	3.6325d-9	3.03	4.6607d-9	3.01
	320	4.5143d-10	3.01	5.7972d-10	3.01
3TNP	20	1.8060d-6	–	2.2866d-6	–
	40	2.1766d-7	3.06	2.7702d-7	3.05
	80	2.6685d-8	3.03	3.4085d-8	3.02
	160	3.3025d-9	3.01	4.2255d-9	3.01
	320	4.1057d-10	3.01	5.2582d-10	3.01

5. Applications in the reactive Euler equations

In this section we apply the methods proposed in Sections 2–4 to solve various reactive problems.

5.1. Numerical examples for scalar problems

Example 5.1. Consider a scalar model problem [2]

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = -\mu u(u-0.5)(u-1). \tag{5.1}$$

Its initial condition is given as

$$u_0(x) = \begin{cases} 1, & x \leqslant 0.3, \\ 0, & x > 0.3. \end{cases}$$

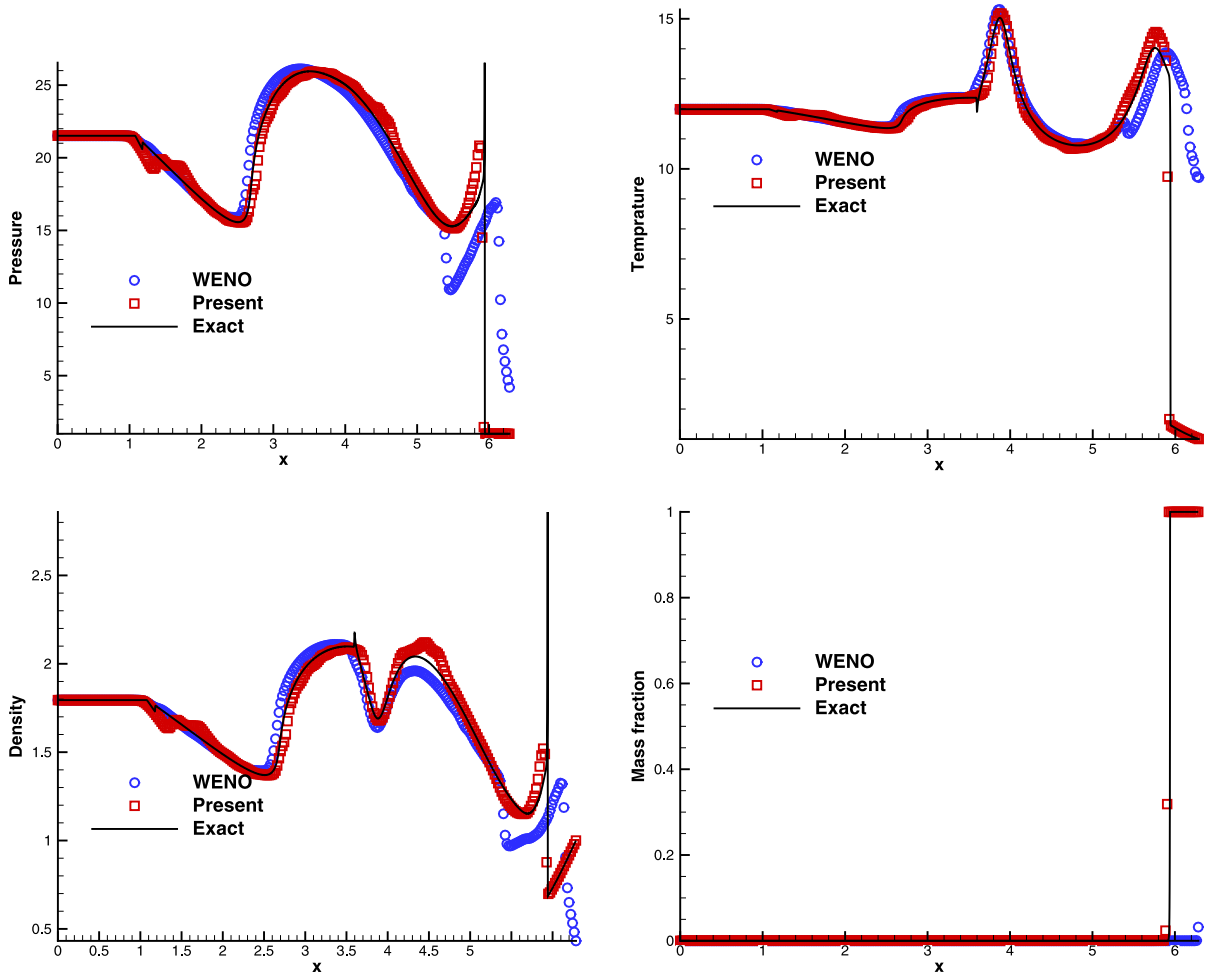

Fig. 16. Numerical results of Example 5.4, $T = \pi/2$, $N = 300$.

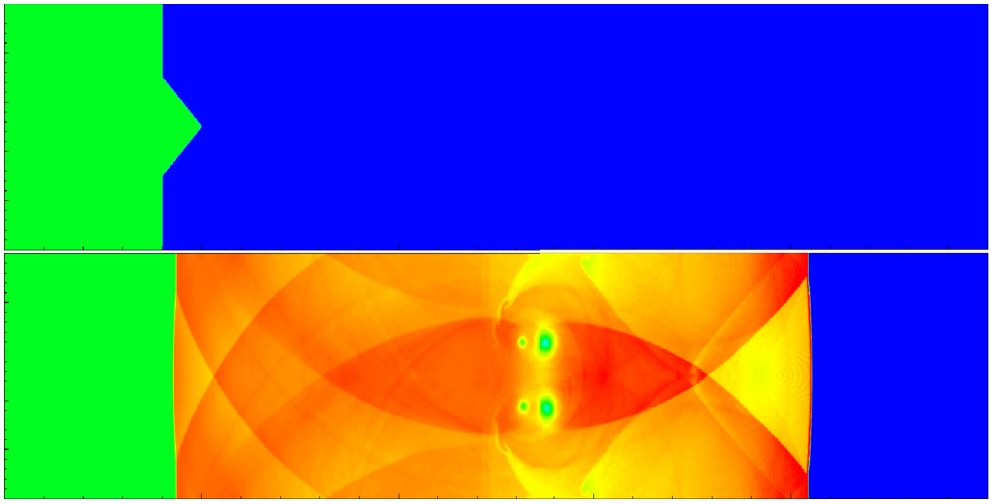

Fig. 17. Density contours results of Example 5.5 by the direct WENO method, $T = 0$ and $T = 1.4 \times 10^{-7}$, $N = 2000 \times 400$.

The exact solution is

$$u_0(x) = \begin{cases} 1, & x \leqslant t + 0.3, \\ 0, & x > t + 0.3. \end{cases}$$

The source term should always be zero theoretically. However, if μ in Eq. (5.1) is very large, the wrong numerical result may appear in the transition region without a suitable method. Using the fractional method, the convection step

$$A: \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, \quad t_n \leqslant t \leqslant t_{n+1},$$

is solved by the DIP method, and the reaction step

$$R: \frac{du}{dt} = f(u) = -\mu u(u - 0.5)(u - 1), \quad t_n \leqslant t \leqslant t_{n+1},$$

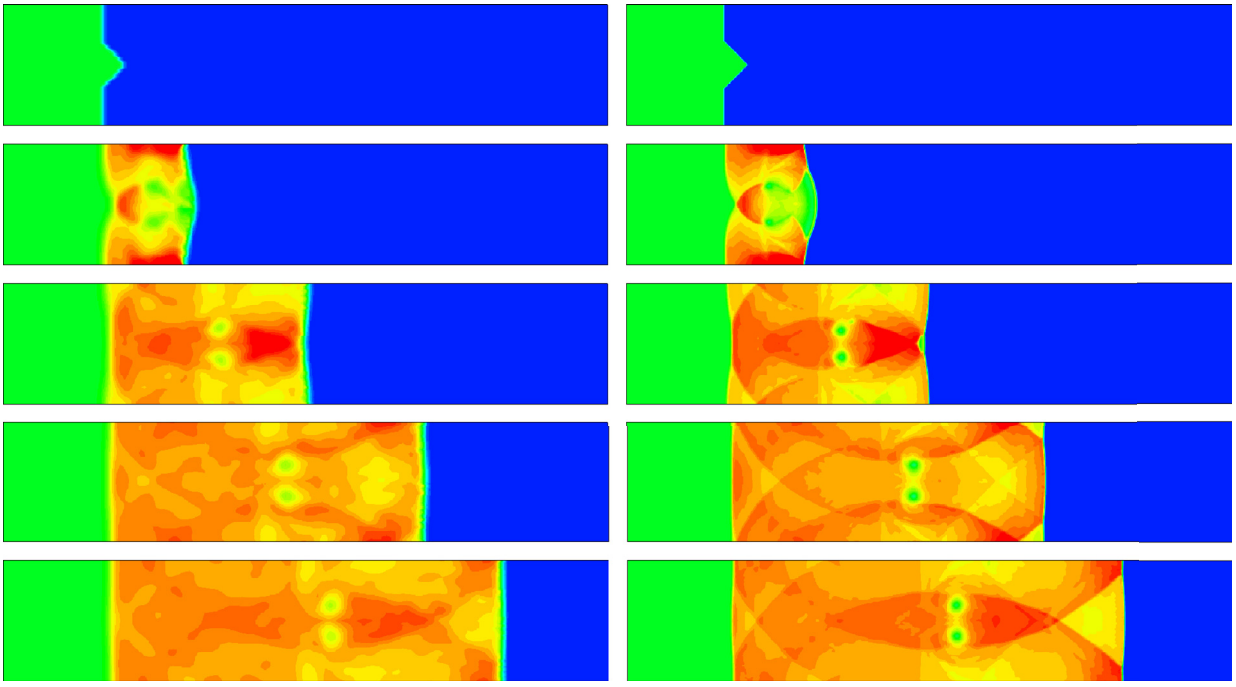

Fig. 18. Density contours of Example 5.5 at time $(0, 0.25 \times 10^{-7}, 0.7 \times 10^{-7}, 1.12 \times 10^{-7}$ and $1.4 \times 10^{-7})$. Left: $N = 150 \times 50$. Right: $N = 600 \times 200$. The present method.

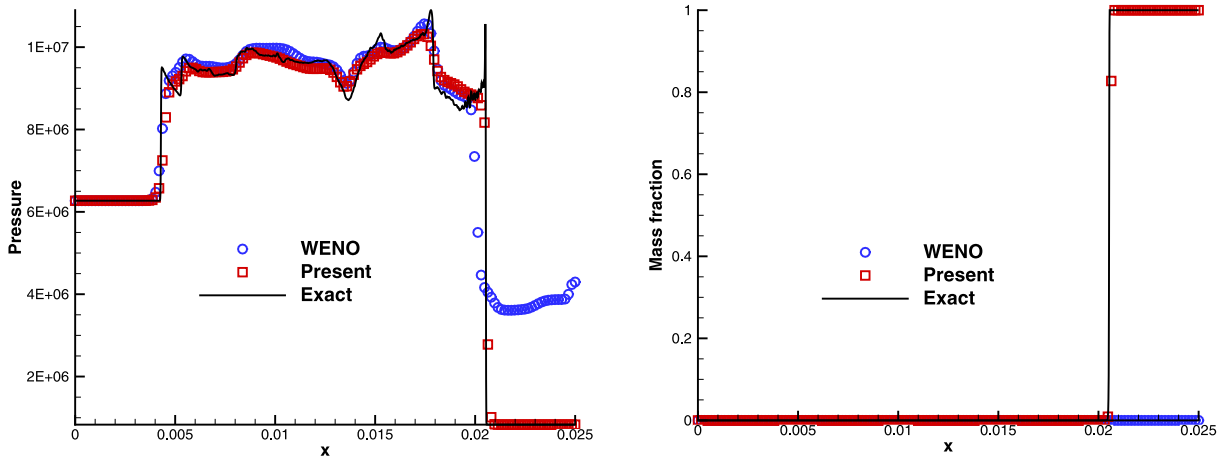

Fig. 19. The distributions on the line $y = 0.0025$ of Example 5.5, $T = 1.4 \times 10^{-7}$.

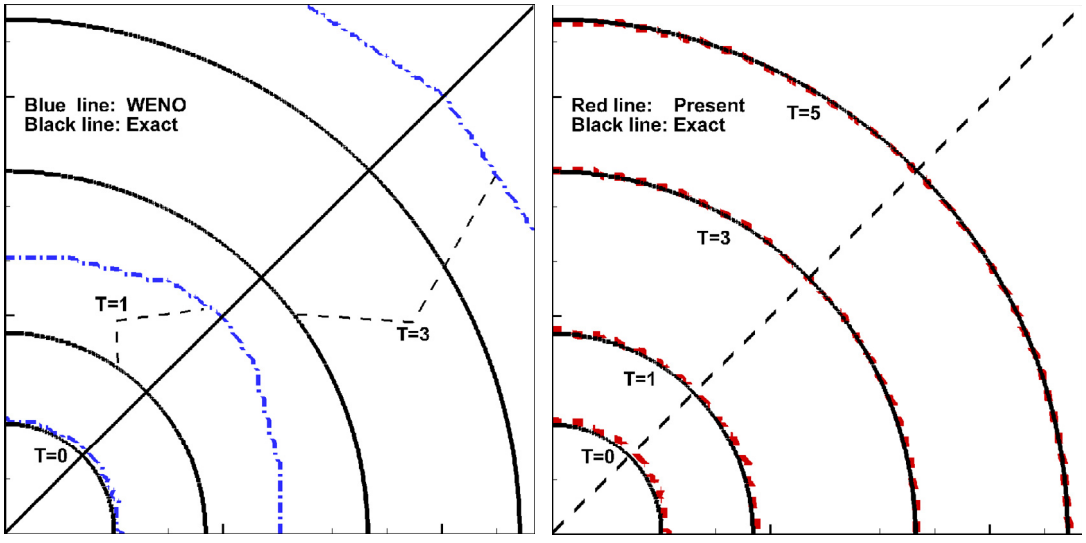

Fig. 20. Mass fraction $(z = 0.5)$ of Example 5.6.

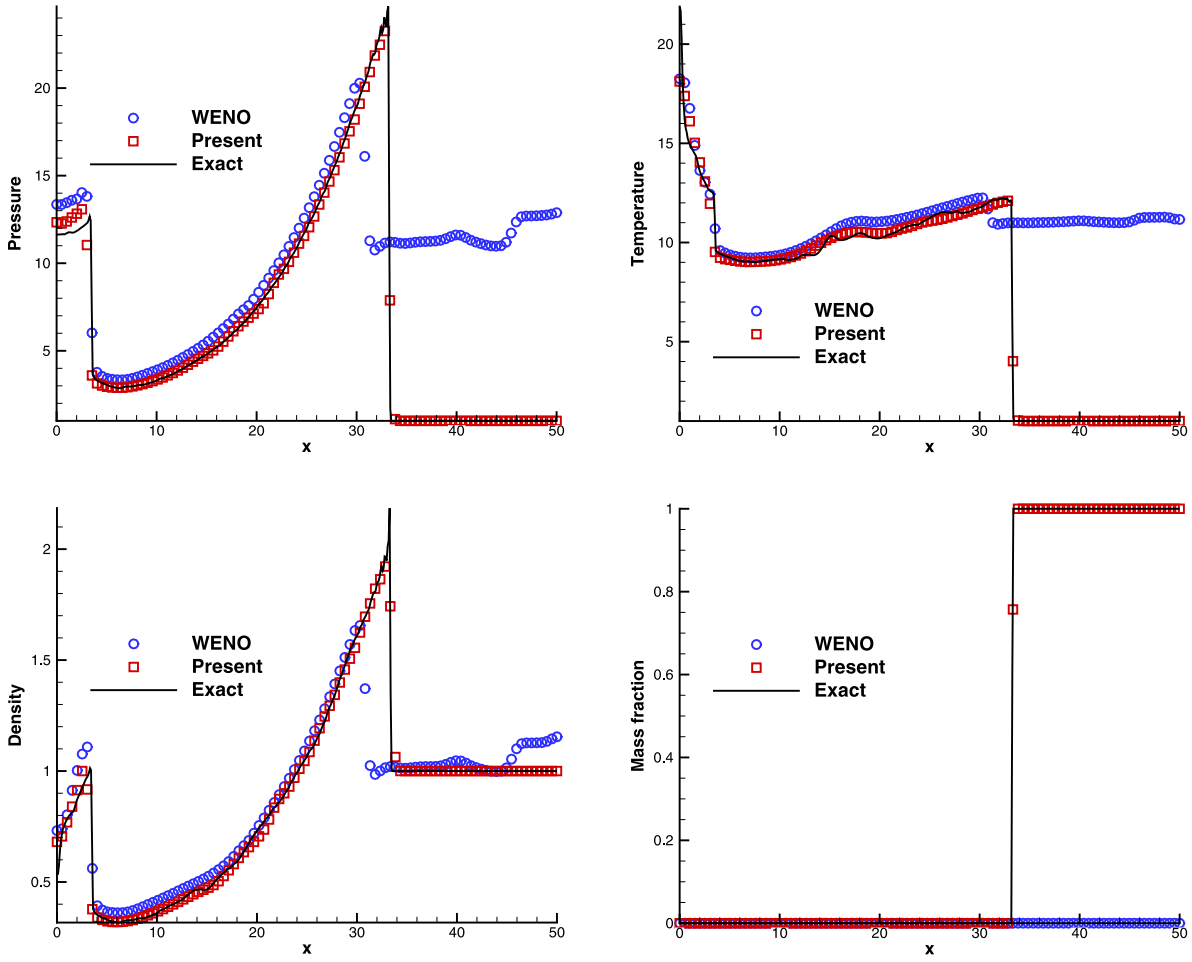

Fig. 21. Comparison of the numerical results on the diagonal line, Example 5.6, $T = 5$.

is solved by the 3TNP scheme. For this case, it is easy to find the first-order and the second-order derivatives of f for calculating the perturbation coefficients.

Notice that, the scheme for ODE equations mainly influences the stability of computation and the time step (the CFL number). Due to its high order and stability, only the third-order transformed NP scheme is used in this paper.

Fig. 9 gives the numerical results calculated by the present method (DIP) and the WENO method for the non-stiff case $\mu = 10$ and stiff case $\mu = 10,000$. We can see the present method can resolve Eq. (5.1) with both cases, while the result calculated by the WENO method for the stiff case has a spurious propagation phenomenon.

5.2. Simplified reactive Euler system

In this system, the reaction has only two states, burnt and un-burnt. Un-burnt gas converts to burnt gas via a single irreversible reaction. The governing equation is Eq. (2.1), its mass fraction is controlled by a scalar equation

$$\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} = s_1, \tag{5.2}$$

and the source term is

$$s_1 = -K(T)z.$$

The reaction rate K determines the stiffness and can be modeled by the Arrhenius law

$$K(T) = K_0 \exp\left(\frac{-T_{\text{ign}}}{T}\right),$$

or by the Heaviside law

$$K(T) = \begin{cases} 1/\epsilon, & T \geqslant T_{\text{ign}}, \\ 0, & T < T_{\text{ign}}, \end{cases}$$

where K_0 is the reaction rate constant, T_{ign} is the ignition temperature and ϵ is the reaction time.

Example 5.2. The first example is an ozone decomposition Chapman–Jouguet (C-J) detonation, which has been computed and discussed in [1,4,5,39]. The Arrhenius source term is used with the following parameter values

$$(\gamma, q_0, K_0, T_{\text{ign}}) = (1.4, 0.5196 \times 10^{10}, 0.5825 \times 10^{10}, 0.1155 \times 10^{10}).$$

The initial values are piecewise constants with burnt gas on the left-hand side and un-burnt gas on the right-hand side, given as

$$(\rho, u, p, z) = \begin{cases} (\rho_b, u_b, p_b, 0), & x \leqslant 0.005, \\ (\rho_0, u_0, p_0, 1), & x > 0.005, \end{cases}$$

where $\rho_0 = 1.201 \times 10^{-3}$, $p_0 = 8.321 \times 10^5$ and $u_0 = 0$. The states of the C-J initial burnt gas are obtained by [1,40–42]

$$\begin{aligned} p_b &= -b + (b^2 - c)^{1/2}, \\ \rho_b &= \rho_u[p_b(\gamma+1) - p_u]/(\gamma p_b), \\ S_{cj} &= [\rho_0 u_0 + (\gamma p_b \rho_b)^{1/2}]/\rho_0, \\ u_b &= S_{cj} - (\gamma p_b/\rho_b)^{1/2}, \\ b &= -p_u - \rho_u q_0(\gamma - 1), \\ c &= p_u^2 + 2(\gamma-1)p_u\rho_u q_0/(\gamma+1), \end{aligned} \tag{5.3}$$

where S_{cj} is the speed of the detonation front.

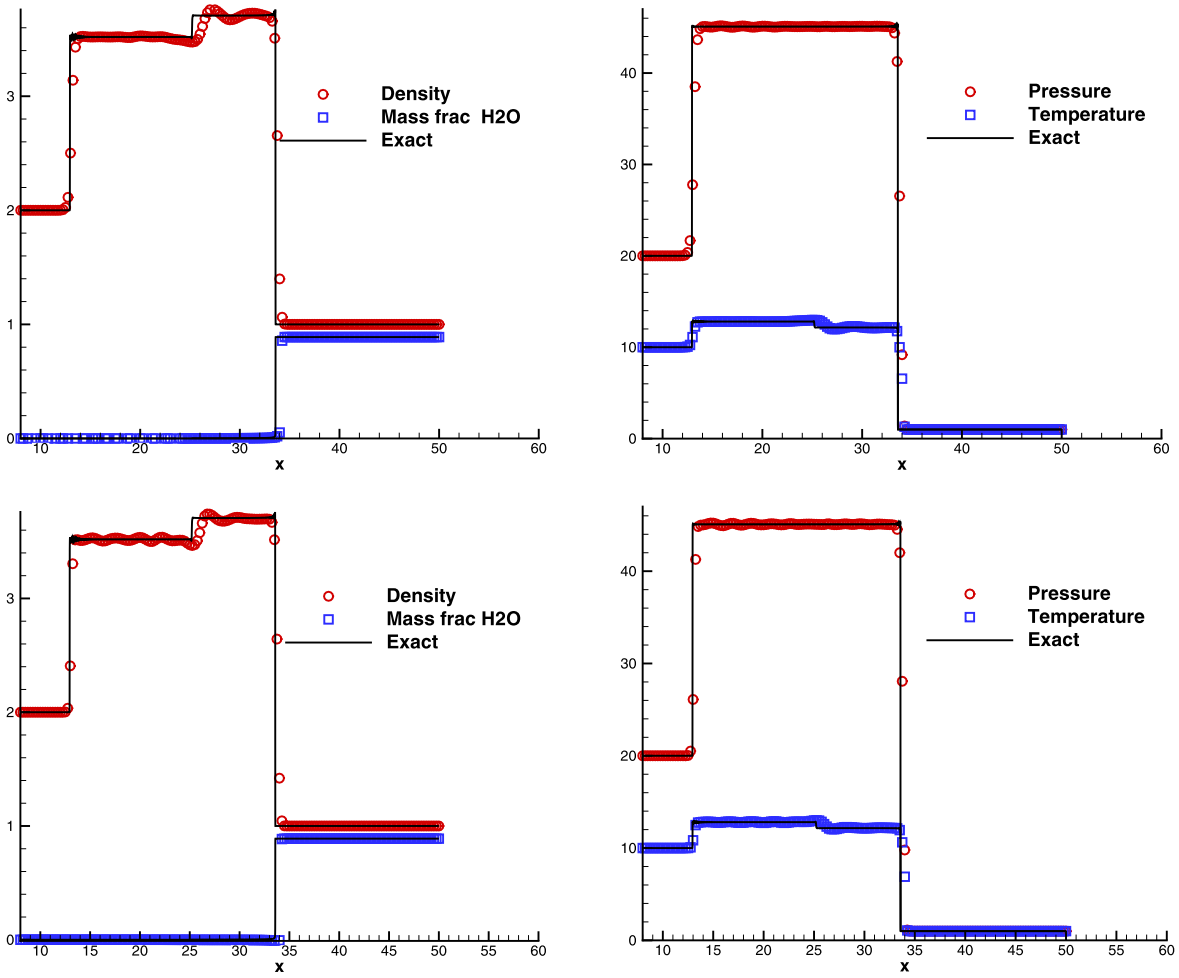

Fig. 22. Numerical results of Example 5.7, $T = 4$ and $N = 200$. Top: the direct WENO method; bottom: the present method.

This problem is solved on the interval [0, 0.05]. The "exact" solutions are obtained by using the direct WENO method in a fine mesh with the size $\Delta x = 5 \times 10^{-6}$ (i.e., $N = 10,000$), which is suggested to resolve the reaction scale [1,5,39]. In this paper, "the direct WENO method" means using the fifth-order WENO scheme [26] and the fourth-order Runge–Kutta method [27] to discretize the spatial derivatives and temporal derivatives in the homogeneous Euler equations (Eq. (2.1) without source term) in the conventional fractional step method, and the 3TNP method is also used to solve the reaction equations. The results obtained by the direct WENO method is symboled as "WENO" in all figures.

The solutions at the time $t = 3 \times 10^{-7}$ with two meshes of $N = 50$ and $N = 300$ are displayed in Figs. 10 and 11, respectively. They show that, the present method can capture the correct profile of detonation wave even the coarse mesh $N = 50$ is used, while applying the direct WENO method, a spurious weak detonation appears ahead of the detonation wave.

Kotov et al. [43] showed that the spurious behavior is influenced significantly by the CFL number, and increasing the stiffness coefficient may generate large derivation for predicting the shock location. To test the influence of the CFL on the new method, we change K_0 to $100K_0$. Three different meshes of $N = 50, N = 100$ and $N = 300$ are used. Numerical results show that the influence of the CFL number on the new method can be neglected. Fig. 12 gives the pressure and the mass fraction distribution of $N = 300$ obtained by the new method.

In this example, we also test the convergence of the iteration in the decoupling process, which is given in the framework of the solving process in Fig. 1. The error is measured by

$$\text{error} = \max_{1 \le i \le N} |u_i^{n,m+1} - u_i^{n,m}|.$$

A convergence history given in Fig. 13 shows that the iteration process introduced in the algorithm is useful to decrease the splitting error. However, although the solution is converged only after a few steps, the computational cost is equivalent to the multiples of the iteration number. Considering that the max error is small enough after one step and always occurs in the detonation region, hence, for the other examples in this paper, a direct computation of Eqs. (2.7)–(2.9) without iteration is implemented.

Example 5.3. In this example, we considered a detonation case with more complex waves [4,6,7,44]. And the Heaviside model with following parameters are used,

$$(\gamma, q_0, 1/\epsilon, T_{\text{ign}}) = (1.2, 50, 230.75, 3).$$

The initial conditions of this example are

$$(\rho, u, p, z) = \begin{cases} (2, 4, 40, 0), & x \leqslant 10, \\ (3.64282, 6.2489.54.8244, 0), & 10 < x \leqslant 20, \\ (1, 0, 1, 1), & x > 20. \end{cases}$$

The solution contains a right moving strong detonation, a right moving rarefaction wave, a right moving contact discontinuity and a left moving rarefaction wave. The "exact" solution is obtained with $N = 10,000$. The results at the final time $T = 8$ with $N = 50$ and $N = 300$ (with $100/\epsilon$) are plotted in Figs. 14 and 15. It can be seen that the DIP method can capture different structures well,

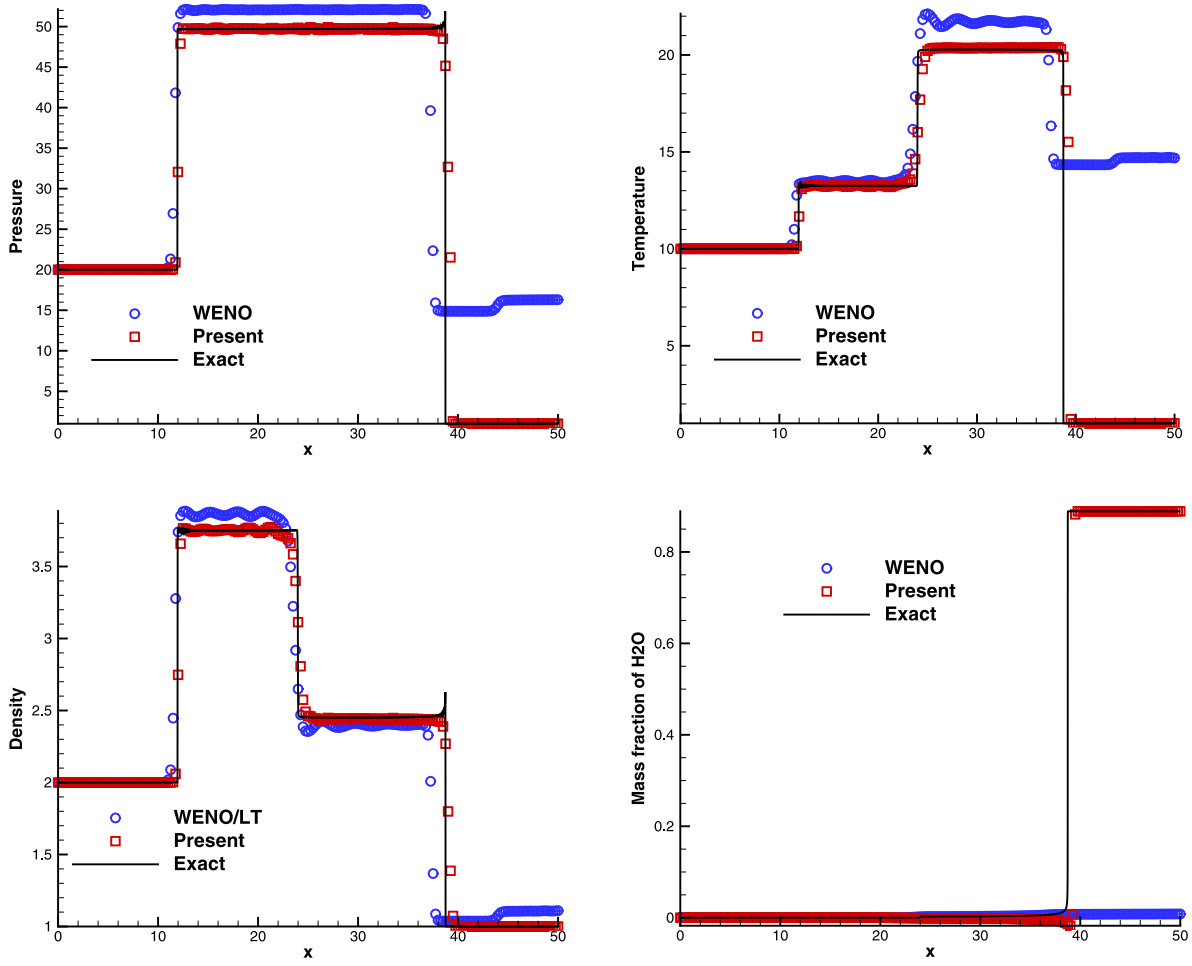

Fig. 23. Numerical results of Example 5.7 with $k_{H_2} = 300$, $T = 4$, $N = 200$.

even in the case with more complex waves and more serious stiffness, while the direct WENO method still cannot obtain the correct location of detonation front. We notice that, the wrong propagation is out of computation region in the right boundary and we use an outflow boundary type here.

Example 5.4. The last one-dimensional problem in this subsection involves a collision with an oscillatory profile [4,5]. The parameters for the Heaviside model are

$$(\gamma, q_0, 1/\epsilon, T_{\text{ign}}) = (1.2, 50, 1000, 3).$$

And the initial conditions are given as

$$(\rho, u, p, z) = \begin{cases} (1.79463, 3.0151, 21.53134, 0), & x \leqslant \frac{\pi}{2}, \\ (3.64282, 6.2489.54.8244, 0), & x > \frac{\pi}{2}. \end{cases}$$

Similarly, the results of the direct WENO method with $N = 10,000$ are taken as the "exact" solution. Fig. 16 gives the comparison at $t = \pi/2$ with the mesh $N = 300$. It shows that the present method can not only capture the shocks well, but also obtains good resolution in smooth regions.

Example 5.5. This is a two-dimensional detonation problem in which a moving detonation wave travels from left to right in a rectangular channel [4,5,7]. In this example, the Arrhenius source model is used and the parameters γ, q_0, K_0, T_{ign} are the same as those in Example 5.2. The initial conditions are given as

$$(\rho, u, v, p, z) = \begin{cases} (\rho_b, u_b, 0, p_b, 0), & x \leqslant \xi(y), \\ (1.201 \times 10^{-3}, 0, 0, 8.321 \times 10^5, 1), & x > \xi(y), \end{cases}$$

$$\xi(y) = \begin{cases} 0.004, & |y - 0.0025| \geqslant 0.001, \\ 0.005 - |y - 0.0025|, & |y - 0.0025| < 0.001. \end{cases}$$

where ρ_b, u_b and p_b are also calculated by Eq. (5.3).

The "exact" solution is computed by the direct WENO method with $N = 2000 \times 400$. Fig. 17 gives the density contours at the beginning and the final time ($t = 1.4 \times 10^{-7}$) with the fine mesh. Fig. 18 displays the density contours at time ($T = 0$, $T_1 = 0.28 \times 10^{-7}$, $T_2 = 0.7 \times 10^{-7}$, $T_3 = 1.12 \times 10^{-7}$ and $T_4 = 1.4 \times 10^{-7}$) with two meshes of $N = 150 \times 50$ and $N = 600 \times 200$. It can be seen that, the shock locations captured with different meshes agree well, and the flow structures are also resolved well even with coarse mesh. The distributions of $N = 600 \times 200$ on the line of $y = 0.0025$ are also given in Fig. 19, which shows that, the detonation wave computed by the present method has a good agreement with the reference solution, while the direct WENO method generates unphysical results similar to those in one-dimensional examples.

Example 5.6. This is another two-dimensional detonation wave problem taken from Ref.[6,45]. The following parameters are used for modeling the Heaviside source term,

$$(\gamma, q_0, K_0, T_{\text{ign}}) = (1.2, 50, 1000, 2).$$

The initial conditions are given as

$$(\rho, u, v, p, z) = \begin{cases} (1.79463, 10x/r, 10y/r, 21.53134, 0), & r \leqslant 10, \\ (1, 0, 0, 1, 1), & r > 10, \end{cases}$$

where

$$r = \sqrt{x^2 + y^2}.$$

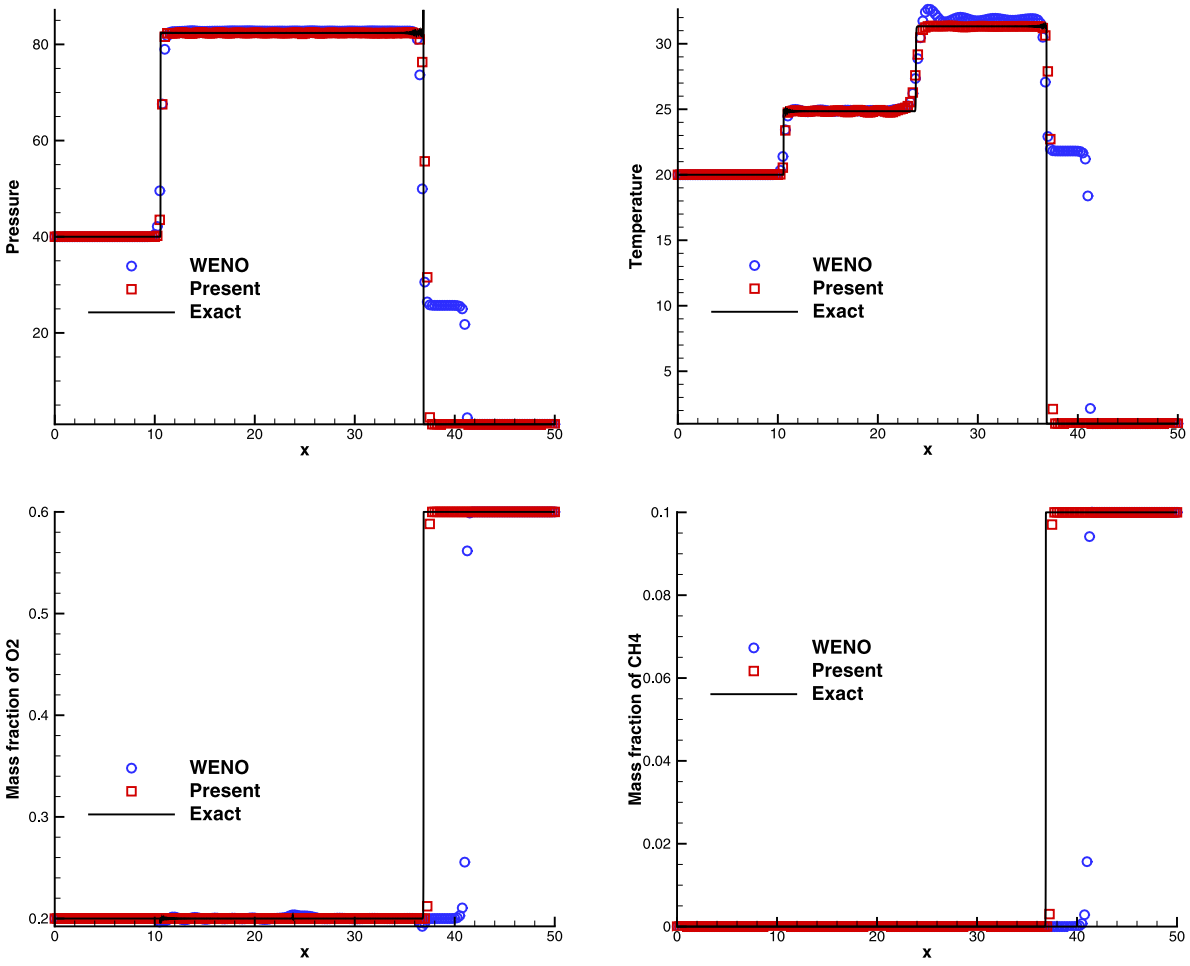

Fig. 24. Numerical results of Example 5.8, $T = 3$, $N = 200$.

This problem represents a radial symmetrical detonation wave moving in a rectangular region. The "exact" solution is computed by the direct WENO method with a fine mesh of $N = 1000 \times 500$. The detonation front with the mesh $N = 200 \times 100$ at time $T_1 = 0$, $T_2 = 1$, $T_3 = 3$ and $T_4 = 5$ are shown in Fig. 20. We can see the present method can capture the location of the detonation front accurately. Fig. 21 compares the results at the time $T = 5$. The present method obtained the same discontinuity location for pressure, density, temperature and mass fraction, but using the direct WENO method, the mass fraction displays a different behavior due to the stiffness, in addition, the distributions of pressure, density and temperature are also distorted compared to the "exact" solution.

5.3. Multi-species reactive Euler system

For multi-species reactive Euler equations without heat conduction and viscosity, the decoupled species equations are in the form of

$$\frac{\partial Z}{\partial t} + u\frac{\partial Z}{\partial x} = S_e,$$

where

$$Z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{ns-1} \end{pmatrix}, S_e = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{ns-1} \end{pmatrix},$$

with the source terms given as

$$s_i = \frac{W_i}{\rho}\sum_{k=1}^{nr}(\mu''_{i,k} - \mu'_{i,k})K_k \prod_{j}^{ns}\left(\frac{\rho z_j}{W_j}\right)^{\mu'_{j,k}},$$

where nr is the number of reactions

$$z_{ns} = 1 - \sum_{i=1}^{ns-1} z_i.$$

And the pressure is given by

$$p = (\gamma - 1)\left(e - \frac{1}{2}\rho u^2 - q_1\rho z_1 - q_2\rho z_2 - \cdots - q_{ns}\rho z_{ns}\right).$$

The temperature is defined as $T = p/\rho$. The reaction rate of the irreversible chemical reaction K_i determines the stiffness of the problem and is expressed in the Heaviside form

$$K_i(T) = \begin{cases} 1/\epsilon_i, & T \geqslant T_{\text{ign}}, \\ 0, & T < T_{\text{ign}}. \end{cases} \quad i = 1, 2, \cdots, nr$$

The transformed third-order perturbation scheme is

$$z_{i,j}^{n+1} = z_{i,j}^{n} + \frac{1}{\bar{p}_i}\Delta t s_i(z_{i,j}^{n}), \quad i = 1, 2, \cdots, ns-1$$

and

$$\bar{p}_i = \frac{1 + b_{i,1}\Delta t + b_{i,2}\Delta t^2}{1 - b_{i,2}\Delta t},$$

where

$$b_{i,1} = a_{i,1} - \frac{a_{i,2}}{a_{i,1}+1}, \quad b_{i,2} = \frac{a_{i,2}}{a_{i,1}+1},$$

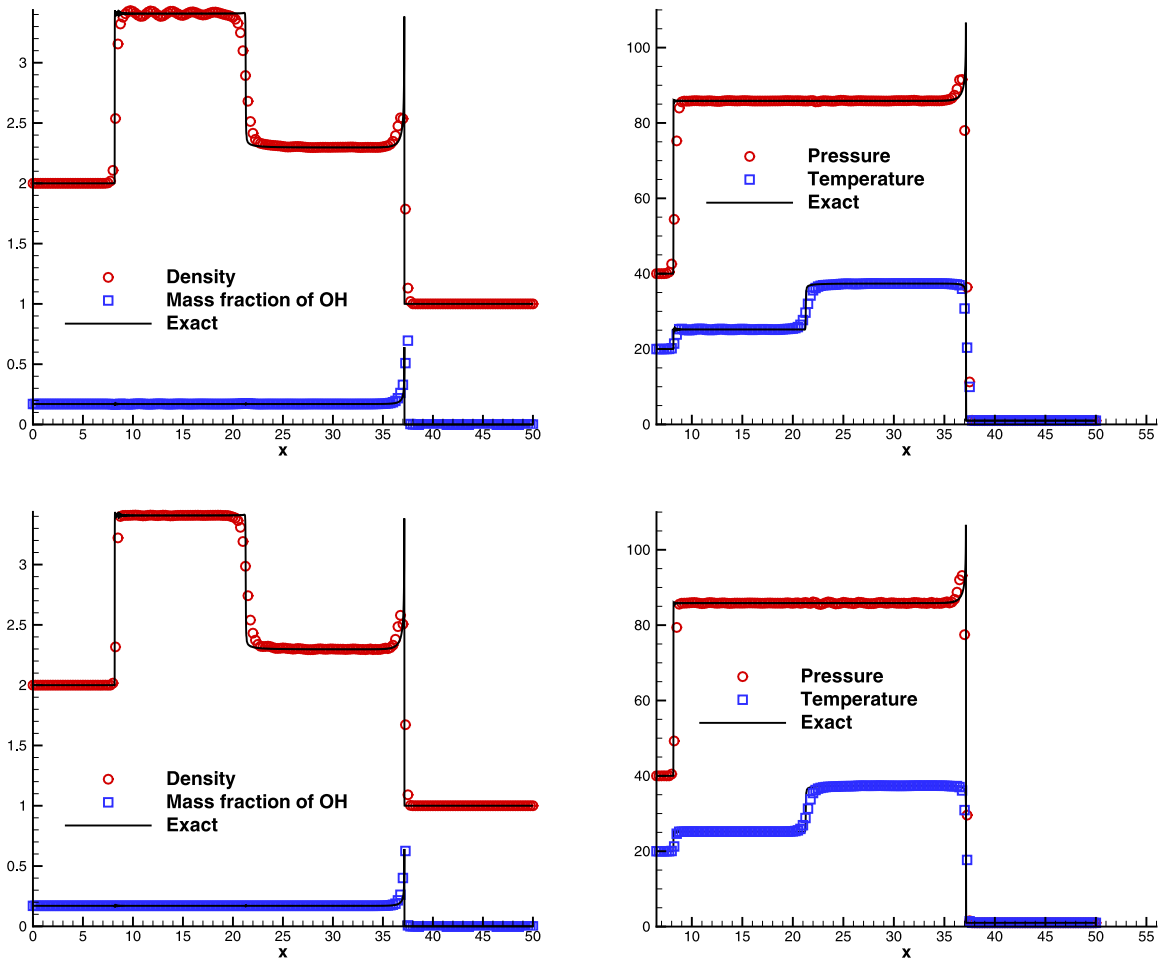

Fig. 25. Numerical results of Example 5.9, $T = 3$, $N = 200$. Top: the direct WENO method; bottom: the present method.

and

$$a_{i,1} = -\frac{1}{2}\sum_{j=1}^{ns}\frac{\partial s_i}{\partial z_j}s_j/s_i,$$

$$a_{i,2} = -\frac{1}{6}\sum_{j=1}^{ns}\sum_{k=1}^{ns}\left(\frac{\partial^2 s_i}{\partial z_j \partial z_k}s_j s_k + \frac{\partial s_i}{\partial z_j}\frac{\partial s_j}{\partial z_k}s_k\right)/s_i + a_{i,1}^2.$$

In this paper, a numerical approximation is used

$$\frac{\partial s_i}{\partial z_j} = \frac{s_i(z_j + \Delta z) - s_i(z_j)}{\Delta z},$$

where Δz is a small value compared to z_j, and taken as

$$\Delta z = \begin{cases} z_j/100, & z_j \neq 0, \\ 0.001, & z_j = 0. \end{cases}$$

Example 5.7. The first multi-species example is taken from [3,7], it uses a simple reaction model

$$2H_2 + O_2 \rightarrow 2H_2O.$$

The parameters for the Heaviside source term are

$$(\gamma, T_{\text{ign}}, 1/\epsilon, q_{H_2}, q_{O_2}, q_{H_2O}, W_{H_2}, W_{O_2}, W_{H_2O}) = (1.4, 2, 10^6, 100, 0, 0, 2, 32, 18).$$

And the initial values are given as following

$$(\rho, u, p, z_{H_2O}, z_{O_2}, z_{H_2O}) = \begin{cases} (2, 8, 20, 0, 0, 1), & 0 \leqslant x \leqslant 2.5, \\ (1, 0, 1, 1/9, 8/9, 0), & 2.5 < x \leqslant 50. \end{cases}$$

The "exact" solution is computed by the direct WENO method with a fine mesh of $N = 10,000$. The results of $N = 200$ at $T = 4$ are plotted in Fig. 22. It shows that both the methods can capture the correct propagation wave. However, if the value of q_{H_2} changes from 100 to 300, as used in [7], the results plotted in Fig. 23 shows that the WENO method cannot maintain the correct propagation speed, while the present method still performs well.

Example 5.8. This example has been studied in [3], its reaction model is

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O.$$

The following parameters are used for modeling the Heaviside source term,

$$(\gamma, T_{\text{ign}}, 1/\epsilon, q_{CH_4}, q_{O_2}, q_{CO_2}, q_{H_2O}) = (1.4, 2, 500, 100, 0, 0),$$

$$(W_{CH_4}, W_{O_2}, W_{CO_2}, W_{H_2O}) = (16, 32, 44, 18).$$

The initial conditions are

$$(\rho, u, p, z_{CH_4}, z_{O_2}, z_{CO_2}, z_{H_2O}) = \begin{cases} (2, 10, 40, 0.325, 0, 0, 0.675), & x \leqslant 2.5, \\ (1, 0, 1, 0.1, 0.4, 0.6, 0), & x > 2.5. \end{cases}$$

The solution of this problem consists of a detonation wave, followed by a contact discontinuity and a shock. The "exact" solution is computed with a fine mesh of $N = 10,000$. The results of $N = 200$ at $T = 3$ are displayed in Fig. 24. It can be seen that, the solution obtained by the present method is in agreement well

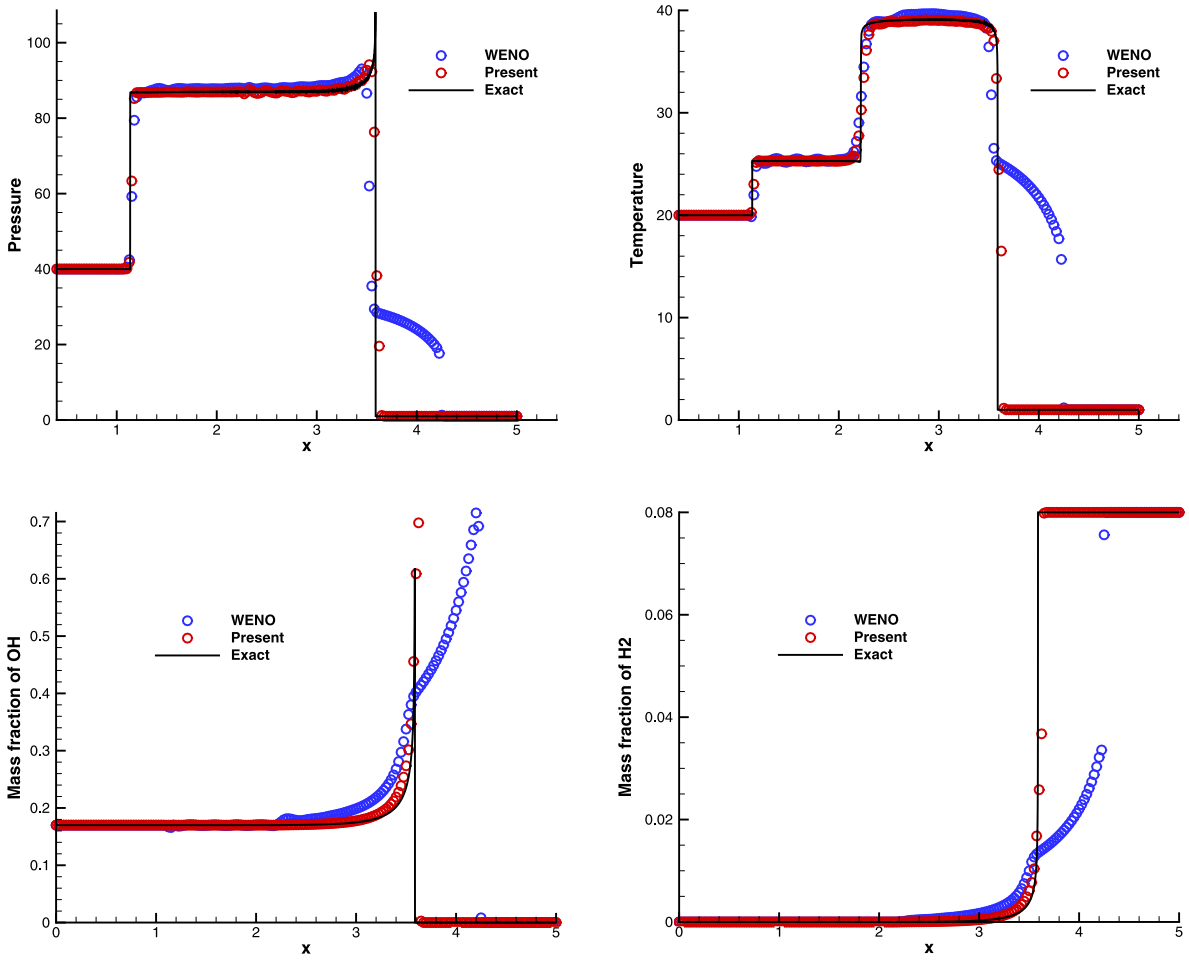

Fig. 26. Numerical results of Example 5.9 with $T_{\text{ign}} = 1.5$ and $q_{\text{H}_2} = -50$. $T = 3$, $N = 200$.

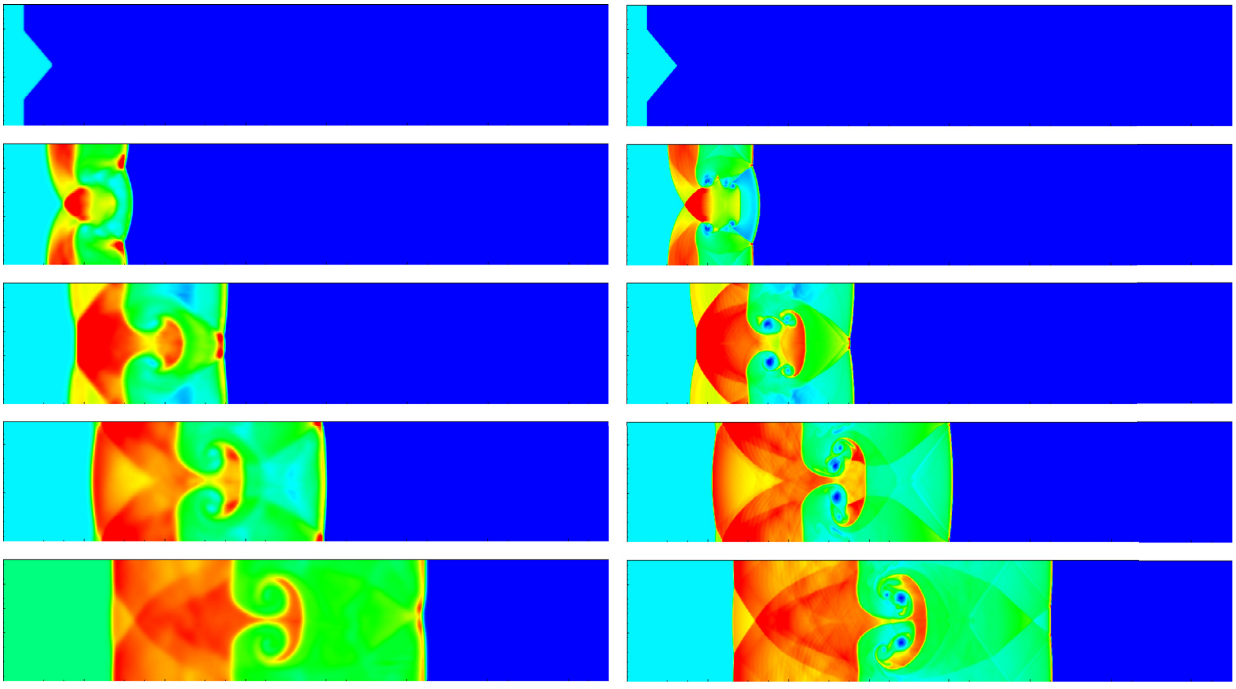

Fig. 27. Density contours of Example 5.10 at $T_1 = 0, T_2 = 2, T_3 = 4, T_4 = 6$ and $T_5 = 8$. Left: the present method, $N = 300 \times 50$. Right: the direct WENO method, $N = 1500 \times 250$.

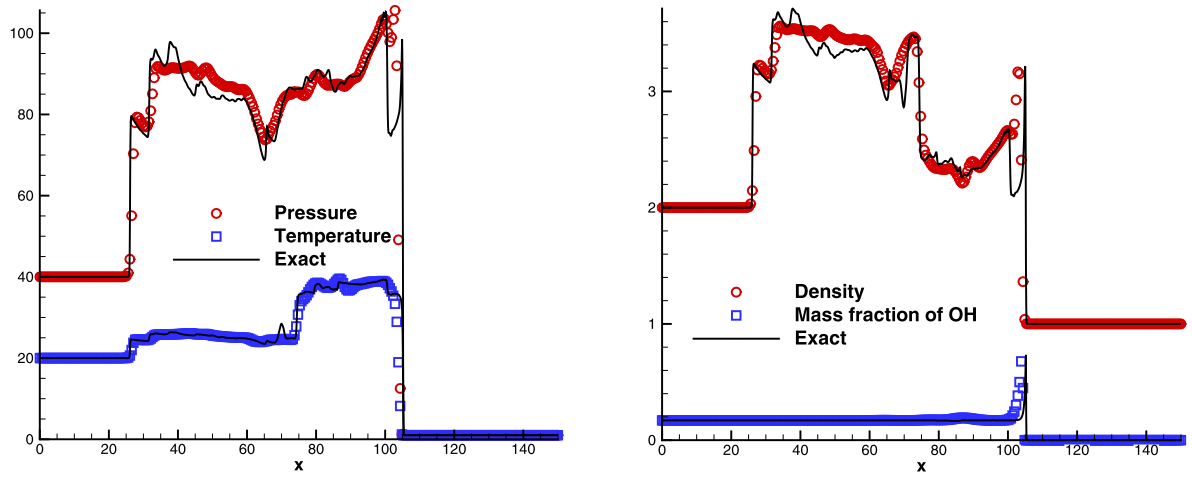

Fig. 28. Distribution on the line of $y = 12.5$ for Example 5.10 at $T = 8$, $N = 300 \times 50$.

with the reference solution, while a deviated propagation speed is obtained by the direct WENO method.

Example 5.9. The last one-dimensional multi-species example is taken from [3,7]. The reaction model consists of five species and two reactions

$$H_2 + O_2 \rightarrow 2OH, \qquad 2OH + H_2 \rightarrow 2H_2O.$$

The species N_2 is used as a catalyst. All parameters given in [3] are

$$(\gamma, T_{\text{ign}}, 1/\epsilon_1, 1/\epsilon_2) = (1.4, 2, 10, 10^5, 2 \times 10^4),$$

$$(q_{H_2}, q_{O_2}, q_{OH}, q_{H_2O}, q_{N_2}) = (0, 0, -20, -100, 0),$$

$$(W_{H_2}, W_{O_2}, W_{OH}, W_{H_2O}, W_{N_2}) = (2, 32, 17, 18, 28).$$

And the initial conditions are

$$(\rho, u, p, z_{H_2}, z_{O_2}, z_{OH}, z_{H_2O}, z_{N_2}) = \begin{cases} (2, 10, 40, 0, 0, 0.17, 0.63, 0.2), & x \leqslant 0.5, \\ (1, 0, 1, 0.08, 0.72, 0, 0, 0.2), & x > 0.5. \end{cases}$$

The computation domain is [0, 50]. The "exact" solution is obtained with a fine mesh of $N = 10,000$. Fig. 25 gives the results of pressure, temperature, mass fractions of O_2 and OH. It shows that, with the parameters given by Ref. [3], both the two methods can get reasonable results, though the direct WENO method generates oscillation more clear than the present method.

The study showed that [7], with a smaller ignition temperature, the first reaction equation is easier to be activated, and hence the stiffness increased. Fig. 26 shows the results with $T_{\text{ign}} = 1.5$ and $q_{H_2} = -50$. For this case, the direct WENO method generates a bifurcating wave pattern and a faster propagation speed similar to the results in [7], while the present method can capture all waves with correct speeds.

Example 5.10. The last case is a two-dimensional example, which has been studied in [3]. The source terms are calculated as those in Example 5.9. The initial conditions are

$$(\rho, u, v, p, z_{H_2}, z_{O_2}, z_{OH}, z_{H_2O}, z_{N_2}) = \begin{cases} (2, 10, 0, 40, 0, 0, 0.17, 0.63, 0.2), & x \leqslant \xi(y), \\ (1, 0, 1, 0.08, 0.72, 0, 0, 0.2), & x > \xi(y), \end{cases}$$

where,

$$\xi(y) = \begin{cases} 12.5 - |y - 12.5|, & |y - 12.5| \geqslant 7.5, \\ 5, & |y - 0.0025| < 7.5. \end{cases}$$

The compute domain is $[0, 150] \times [0, 25]$. Fig. 27 shows the evolution of the detonation wave at time $T_1 = 0, T_2 = 2, T_3 = 4, T_4 = 6$ and $T_5 = 8$ with a mesh $N = 300 \times 50$. In order to compare, the results simulated by the direct WENO method with a fine mesh of $N = 1500 \times 250$ are also plotted. This figure shows the present method can resolve all the structures, even using a coarse mesh. Fig. 28 gives the numerical comparison on the line of $y = 12.5$. The results obtained by the new method agree well with the "exact" solution.

6. Conclusions

The dual information preserving method is firstly proposed to cure the numerical stiff problem generated in simulating the reacting flows. First, the species mass fraction equations are decoupled from the reactive Euler equations, and then they are further fractionated into the convection step and reaction step. The DIP method is actually proposed to deal with the species convection step. Two kinds of Lagrangian points are introduced, one is limited in each Eulerian cell, and another one is tracked in the whole computation domain. Each kind of points has the same number of the Eulerian cells (grids). The information of the cell-point in a cell can effectively restrict the incorrect reaction activation possibly caused by the numerical dissipation, while the information of the particle-point can help to preserve the sharp shock front once the strong shock waves formed. Hence, by using the DIP method, the spurious numerical propagation phenomenon in stiff reacting flows is effectively eliminated.

In this paper, the numerical perturbation (NP) methods are also developed to solve the fractional reaction step (ODE equations). The NP schemes show several advantages, such as no need of iteration, high order accuracy and large stable region.

A series of numerical examples are used to demonstrate the reliability and robustness of the new methods.

Acknowledgement

This research work was supported by NSAF U1530145, 2016YFA0401200, NSFC No. 11272325 and 11272324.

Appendix A. The algorithm of the DIP method for solving the convection equations

As preparations, at the beginning of the computation,

DO $i = 0, NX$
  DO $j = 0, NY$
    $\begin{cases} X(i,j) = 0 \\ Y(i,j) = 0 \end{cases}$, $\begin{cases} X_p(i,j) = 0 \\ Y_p(i,j) = 0 \end{cases}$, $\begin{cases} ip(i,j) = i \\ jp(i,j) = j \end{cases}$, $\begin{cases} \bar{z}(i,j) = z(i,j) \\ \bar{z}_p(i,j) = z(i,j) \end{cases}$
  ENDDO
ENDDO

DO $it = 1, NT$! Time-Stepping

Solve the decoupled Euler equations (2.7) to get u, v, $\cdots$

Following is the DIP algorithm for the convection equations (2.8).

(1) Initial values

$Mrk(i,j) = 0, S_1(i,j) = 0, S_2(i,j) = 0$

(2) Calculate the cell-point

DO $i = 0, NX$
  DO $j = 0, NY$
    $s_x = \text{sign}(X(i,j)), s_y = \text{sign}(Y(i,j))$
    $\begin{cases} \bar{u}(i,j) = & (1-|X|)u(i,j) & +|X|u(i+s_x,j) \\ \bar{v}(i,j) = & (1-|Y|)v(i,j) & +|Y|v(i,j+s_y) \end{cases}$! Interpolate velocity according to its location
    $\begin{cases} L_x = X(i,j) + \bar{u}(i,j)\Delta t/\Delta x \\ L_y = Y(i,j) + \bar{v}(i,j)\Delta t/\Delta y \end{cases}$
    $\begin{cases} M = i + \text{floor}(L_x + 0.5), \\ N = j + \text{floor}(L_y + 0.5). \end{cases}$! (i,j) moves to cell (M,N)
    $\begin{cases} X(i,j) = L_x - \text{floor}(L_x + 0.5) \\ Y(i,j) = L_y - \text{floor}(L_y + 0.5) \end{cases}$! The relative location in (M,N)
    IF $(0 \leqslant M \leqslant NX$ and $0 \leqslant N \leqslant NY)$ Then
      $Mrk(M,N) = 1$
      $S_1(M,N) = S_1(M,N) + 1$
      $\begin{cases} \bar{X}(M,N) = & \{[S_1(M,N)-1]\bar{X}(M,N) & +X(i,j)\}/S_1(M,N) \\ \bar{Y}(M,N) = & \{[S_1(M,N)-1]\bar{Y}(M,N) & +Y(i,j)\}/S_1(M,N) \\ \bar{z}(M,N) = & \{[S_1(M,N)-1]\bar{z}(M,N) & +\bar{z}(i,j)\}/S_1(M,N) \end{cases}$! Average information
    ENDIF
  END DO
END DO
$\begin{cases} X(:,:) = \bar{X}(:,:) \\ Y(:,:) = \bar{Y}(:,:) \end{cases}$

(3) Calculate the particle-point and update the cell-point(ip, jp)

DO $i = 0, NX$
  DO $j = 0, NY$
    $s_x = \text{sign}(X_p(i,j)), s_y = \text{sign}(Y_p(i,j))$
    $\begin{cases} \bar{u}_p(i,j) = & (1-|X_p|)u(ip,jp) & +|X_p|u(ip+s_x,J) \\ \bar{v}_p(i,j) = & (1-|Y_p|)v(ip,jp) & +|Y_p|v(ip,jp+s_y) \end{cases}$! Interpolate velocity according to its position
    $\begin{cases} L_x = X_p(i,j) + \bar{u}_p(i,j)\Delta t/\Delta x \\ L_y = Y_p(i,j) + \bar{v}_p(i,j)\Delta t/\Delta y \end{cases}$
    $\begin{cases} ip(i,j) = ip(i,j) + \text{floor}(L_x + 0.5) \\ jp(i,j) = jp(i,j) + \text{floor}(L_y + 0.5) \end{cases}$! Particle point (i,j) moves to cell (ip,jp)
    $\begin{cases} X_p(i,j) = L_x - \text{floor}(L_x + 0.5) \\ Y_p(i,j) = L_y - \text{floor}(L_y + 0.5) \end{cases}$! The relative location in (ip,jp)
    IF $(0 \leqslant ip \leqslant NX$ and $0 \leqslant jp \leqslant NY)$ Then
      $Mrk(ip,jp) = 2$
      $S_2(ip,jp) = S_2(ip,jp) + 1$
      $\begin{cases} \bar{X}(ip,jp) = & \{[S_2(ip,jp)-1]\bar{X}(ip,jp) & +X_p(i,j)\}/S_2(ip,jp) \\ \bar{Y}(ip,jp) = & \{[S_2(ip,jp)-1]\bar{Y}(ip,jp) & +Y_p(i,j)\}/S_2(ip,jp) \\ \bar{z}(ip,ip) = & \{[S_2(ip,jp)-1]\bar{z}(ip,jp) & +\bar{z}_p(i,j)\}/S_2(ip,jp) \end{cases}$! Update cell-point (ip,jp)'s information by averaging all entered particle-points' information
    END IF
  END DO
END DO
$\begin{cases} X(:,:) = \bar{X}(:,:) \\ Y(:,:) = \bar{Y}(:,:) \end{cases}$

(4) If there is not any point in the cell (i,j), i.e, $Mrk(i,j) = 0$, then a new cell-point is set at the cell center and its transport value is obtained by averaging its neighboring cell-points' values.

DO $i = 0, NX$
  DO $j = 0, NY$
    IF $Mrk(i,j) = 0$ then
      $\bar{z}(i,j) = 0, S_1 = 0$
      $\begin{cases} X(i,j) = 0 \\ Y(i,j) = 0 \end{cases}$
      DO $i_1 = -1, 1, 2$
        DO $j_1 = -1, 1, 2$
          IF $Mrk(i+i_1, j+j_1) > 0$ THEN
            $L = \sqrt{(X(i+i_1,j+j_1)+i_1)^2 + (Y(i+i_1,j+j_1)+j_1)^2}$
            $S_1 = S_1 + 1/L$
            $\bar{z}(i,j) = \bar{z}(i,j) + \bar{z}(i+i_1,j+j_1)/L$
          END IF
        END DO
      END DO
      $\bar{z}(i,j) = \bar{z}(i,j)/S_1$
    END IF
  END DO
END DO

Solve the reaction equations (2.9) to get new z and $\bar{z}_p$ by using $\bar{z}$ and $\bar{z}_p$ to calculate the source terms, respectively.

END DO

References

[1] Colella P, Majda A, Roytburd V. Theoretical and numerical structure for reacting shock waves. SIAM J Sci Stat Comput 1986;7(4):1059–80.
[2] LeVeque RJ, Yee HC. A study of numerical methods for hyperbolic conservation laws with stiff source terms. J Comput Phys 1990;86(1):187–210.
[3] Bao W, Jin S. The random projection method for stiff multispecies detonation capturing. J Comput Phys 2002;178(1):37–57.
[4] Wang W, Shu C-W, Yee H, Sjögreen B. High order finite difference methods with subcell resolution for advection equations with stiff source terms. J Comput Phys 2012;231(1):190–214.
[5] Bao W, Jin S. The random projection method for hyperbolic conservation laws with stiff reaction terms. J Comput Phys 2000;163(1):216–48.
[6] Bao W, Jin S. The random projection method for stiff detonation capturing. SIAM J Sci Comput 2001;23(3):1000–26.
[7] Zhang B, Liu H, Chen F, Wang JH. The equilibrium state method for hyperbolic conservation laws with stiff reaction terms. J Comput Phys 2014;263:151–76.
[8] Harten A. Eno schemes with subcell resolution. J Comput Phys 1989;83(1):148–84.
[9] Chang S-H. On the application of subcell resolution to conservation laws with stiff source terms. Technical Memorandum 102384, November. NASA; 1989.
[10] Wang W, Shu C-W, Yee H, Kotov DV, Sjögreen B. High order finite difference methods with subcell resolution for stiff multispecies discontinuity capturing. Commun Comput Phys 2015;17(02):317–36.
[11] Yee H, Kotov DV, Wang W, Shu C-W. Spurious behavior of shock-capturing methods by the fractional step approach: problems containing stiff source terms and discontinuities. J Comput Phys 2013;241:266–91.
[12] Yee H, Sjögreen B. High order filter methods for wide range of compressible flow speeds. In: Spectral and high order methods for partial differential equations. Springer; 2011. p. 327–37.
[13] Poinsot T, Echekki T, Mungal M. A study of the laminar flame tip and implications for premixed turbulent combustion. Combust Sci Technol 1992;81(1–3):45–73.
[14] Moureau V, Fiorina B, Pitsch H. A level set formulation for premixed combustion les considering the turbulent flame structure. Combust Flame 2009;156(4):801–12.
[15] Nguyen D, Gibou F, Fedkiw R. A fully conservative ghost fluid method and stiff detonation waves. 12th international detonation symposium, San Diego, CA; 2002.
[16] McKee S, Tomé M, Ferreira V, Cuminato J, Castelo A, Sousa F, et al. The mac method. Comput Fluids 2008;37(8):907–30.
[17] De Sousa F, Mangiavacchi N, Nonato L, Castelo A, Tomé M, Ferreira V, et al. A front-tracking/front-capturing method for the simulation of 3d multi-fluid flows with free surfaces. J Comput Phys 2004;198(2):469–99.
[18] Tomé M, Cuminato J, Mangiavacchi N, McKee S, et al. Gensmac3d: a numerical method for solving unsteady three-dimensional free surface flows. Int J Numer Methods Fluids 2001;37(7):747–96.
[19] Yoon H, Park I, Lee Y, Jeong J. An unstructured smac algorithm for thermal non-equilibrium two-phase flows. Int Commun Heat Mass Transf 2009;36(1):16–24.
[20] Smolianski A, Shipilova O, Haario H. A fast high-resolution algorithm for linear convection problems: particle transport method. Int J Numer Methods Eng 2007;70(6):655–84.
[21] Moresi L, Dufour F, Mühlhaus H-B. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. J Comput Phys 2003;184(2):476–97.
[22] Dong L, Wang B. Trajectory-tracking scheme in Lagrangian form for solving linear advection problems: preliminary tests. Mon Weather Rev 2012;140(2):650–63.
[23] Liu L, Becerra M. An efficient semi-Lagrangian algorithm for simulation of corona discharges: the position-state separation method. IEEE Trans Plasma Sci 2015;PP:1–10.
[24] Oliveira B, Afonso JC, Zlotnik S. A Lagrangian–Eulerian finite element algorithm for advection–diffusion–reaction problems with phase change. Comput Methods Appl Mech Eng 2016;300:375–401.
[25] Shipilova O, Haario H, Smolianski A. Particle transport method for convection problems with reaction and diffusion. Int J Numer Methods Fluids 2007;54(10):1215–38.
[26] Jiang G-S, Shu C-W. Efficient implementation of weighted eno schemes. J Comput Phys 1996;126(1):202–28.
[27] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77(2):439–71.
[28] Crandall M, Majda A. The method of fractional steps for conservation laws. Numerische Mathematik 1980;34(3):285–314.
[29] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. J Comput Phys 1979;31(3):335–62.
[30] Olsson E, Kreiss G. A conservative level set method for two phase flow. J Comput Phys 2005;210(1):225–46.
[31] Aulisa E, Manservisi S, Scardovelli R. A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows. J Comput Phys 2003;188(2):611–39.
[32] Gao Z. Advances in perturbation finite difference (pfd) method. Adv Mech 2000;30:200–15.
[33] Yu C, Gao Z, Sheu TW. Development of a symplectic and phase error reducing perturbation finite-difference advection scheme. Numer Heat Transf Part B 2016;70(2):136–51.
[34] Gao Z. Numerical perturbation algorithm and its cfd schemes. Adv Mech 2010;40:607–33.
[35] Shampine LF. Numerical solution of ordinary differential equations, 4. CRC Press; 1994.
[36] Dahlquist GG. A special stability problem for linear multistep methods. BIT Numer Math 1963;3(1):27–43.
[37] Hairer E, Wanner G. Stiff differential equations solved by Radau methods. J Comput Appl Math 1999;111(1):93–111.
[38] Chipman F. A-stable Runge–Kutta processes. BIT Numer Math 1971;11(4):384–8.
[39] Ben-Artzi M. The generalized Riemann problem for reactive flows. J Comput Phys 1989;81(1):70–101.
[40] Chorin AJ. Random choice methods with applications to reacting gas flow. J Comput Phys 1977;25(3):253–72.
[41] Colella P. Glimm's method for gas dynamics. SIAM J Sci Stat Comput 1982;3(1):76–110.
[42] Courant R, Friedrichs KO. Supersonic flow and shock waves, 21. Springer Science & Business Media; 1999.
[43] Kotov D, Yee H, Wang W, Shu C. On spurious numerics in solving reactive equations. In: Proceedings of the ASTRONUM-2012, The Big Island, Hawaii; 2012. p. 24–8.
[44] Hwang P, Fedkiw R, Merriman B, Aslam T, Karagozian A, Osher S. Numerical resolution of pulsating detonation waves. Combust Theor Model 2000;4:217–40.
[45] Helzel C, Leveque RJ, Warnecke G. A modified fractional step method for the accurate approximation of detonation waves. SIAM J Sci Comput 2000;22(4):1489–510.