Computers and Fluids 213 (2020) 104718

Contents lists available at ScienceDirect

Computers and Fluids

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

A quasi-direct numerical simulation solver for compressible reacting flows

Tao Li[a], Jiaying Pan[b], Fanfu Kong[c], Baopeng Xu[c,*], Xiaohan Wang[a,*]

[a] Guangzhou Institute of Energy Conversion, Chinese Academy of Sciences, Guangzhou 510640, China
[b] State Key Laboratory of Engines, Tianjin University, Tianjin 300072, China
[c] School of Energy and Power Engineering, Dalian University of Technology, Dalian 116024, China

ARTICLE INFO

Article history:
Received 2 April 2020
Revised 1 July 2020
Accepted 13 August 2020
Available online 18 September 2020

Keywords:
q-DNS
Compressible reacting flows
OpenFOAM®
Chemical diffusion

ABSTRACT

A new quasi-direct numerical simulation (q-DNS) solver named *combustionFoam*, capable of simulating reacting flows at arbitrary Mach number with a detailed chemical reaction mechanism, has been developed. This solver is implemented based on *reactingFoam*-solver from OpenFOAM-v7. The main improvements are a mixture-averaged formula transport model and ability to simulate reacting flows at arbitrary Mach number. The transport model is implemented by developing an data exchange interface to couple OpenFOAM® with Cantera. To damped nonphysical oscillations near shocks, a hybrid KT/KNP approach derived by Kraposhin et al. [18] is adopted. This solver was validated by two test cases: (1) a low speed steady two dimensional non-premixed counterflow diffusion flame; (2) a supersonic transient one dimensional detonation waves. Comparisons of the results with data from other solvers and literatures are carried out and good agreements are obtained.

1. Introduction

Direct numerical simulation (DNS) of reacting flows can resolve flow structures and chemical reactions at all length scales and time scales. Therefore, DNS can obtain much more information than experimental methods. DNS can help us to understand physical phenomena, develop and verify turbulence models and turbulent combustion interaction models.

Due to the usability of high order schemes, DNS codes usually are implemented within spectral method (such as *Channelflow* [12]) or within finite difference method (FDM). Several DNS codes with FDM for reacting flows simulation are reported in literatures, such as *SENGA* [14], *S3D* [5], *SMC* [9] and *Pencil Code* [2]. In addition, there is a type of code belonging to the high order finite volume method, such as *OpenCFD-Comb* [11], *AMLSDC* algorithm [8] and *AMROC* [24]. Higher order schemes have smaller truncation errors, but are also more costly.

In general, the accuracy of finite volume method is greatly limited by the midpoint interpolation in the calculation of flux. However, Komen et al. [17] studied turbulent flows in a circular tube using a second-order finite volume method in OpenFOAM®. Their results show that using sufficient grid resolution could also obtain high quality average and fluctuating velocity fields. Because of the abandonment of the higher-order schemes, such methods are called quasi-direct numerical simulation (q-DNS). In addition, Axtmann and Rist [1] and Liu et al. [19] also used OpenFOAM® to conduct q-DNS simulations, further validated the feasibility of using OpenFOAM® for q-DNS. Several other q-DNS type codes used for combustion simulations are reported in lectures, such as *ASURF* [6], *laminarSMOKE* [7], and *reactingFoam-SCI* [23].

As far as we are aware, including the codes mentioned above, most of them are in-house or part in-house type code, except *AMROC* and *Pencil Code*. Considering the need for further development, the implementation of models such as moving boundaries, liquid fuel, tabulation of dynamic adaptive chemistry (TDAC), radiation, and surface catalytic effect etc. in AMROC or Pencil Code is enormous. Since OpenFOAM® is one of the most popular open source framework for computational fluid dynamics (CFD) simulation. Community developers have developed a plenty of highly versatile and scalable libraries and it is easy to extend simulation from one dimensional to three dimensional. Hence, in this work we implement our new solver in the OpenFOAM® framework (Version 7).

The current work based on the famous *reactingFoam*-solver for reacting flows in OpenFOAM®. The governing equations solved by *reactingFoam*-solver is based on Schmidt number and Lewis number are equal to one. Under this assumption, mass diffusion and

* Corresponding authors.
E-mail addresses: lit@dlut.edu.cn (T. Li), baopengxu@dlut.edu.cn (B. Xu), wangxh@ms.giec.ac.cn (X. Wang).

https://doi.org/10.1016/j.compfluid.2020.104718
0045-7930/

momentum diffusion cannot be predicted well. Therefore, the laminar flame propagation speed is predicted with discrepancy that cannot be ignored. Therefore, a detailed transport model is needed for DNS. Only limited attempts in this aspect have been made, such as [3,23]. Despite the promising results presented in articles, the development platform is too old or the search for source code implementation fails. In current study, the transport models from Cantera (Version 2.3) [13] are linked with thermo-physical models of OpenFOAM® to calculate the mass diffusion coefficient of each species in mixture-averaged formula, dynamic viscosity and thermal conductivity of the gas mixture. Besides, *reactingFoam*-solver predicted a non-physical temperature oscillation for pure gas mixing process (see Section 4.1). It also cannot deal with discontinuities in flow fields. Therefore, we use a hybrid scheme proposed by Kraposhin et al. [18], which is capable of simulating flows at arbitrary Mach number and the nonphysical oscillations are damped better than the original *reactingFoam*-solver.

The structures of this paper are as follows: in Section 2 we discuss the governing equations solved in *combustionFoam*-solver. The differences between *combustionFoam*-solver and *reactingFoam*-solver are emphasized. In Section 3 we discussthe transport model interface, discretization schemes and algorithm used by *combustionFoam*-solver. In Section 4 we present numerical results to show its ability to solve reacting flows with multi-species diffusion and shocks. Finally some conclusions are summarized in Section 5.

2. Governing equations

The compressible Navier-Stokes equations, equation of state (EoS) and transport model used in this work are described in this section. The governing equations has been solved using the open source toolbox OpenFOAM® (version 7) [21]. Some of the key features are:

- An ideal gas mixture with zero bulk viscosity assumption is adopted.
- The mixture averaged transport model is employed.
- Soret effect, Dufour effect and radiant effect are neglected.

The governing equations are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{U}\right) = 0 \tag{1}$$

$$\frac{\partial \left(\rho \vec{U}\right)}{\partial t} + \nabla \cdot \left(\rho \vec{U}\vec{U}\right) - \nabla \cdot \bar{\bar{\tau}} = -\nabla p \tag{2}$$

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot \left(\rho Y_i \vec{U}\right) + \nabla \cdot \left(\rho Y_i \vec{\boldsymbol{V}}_i\right) = \dot{w}_i \tag{3}$$

$$\frac{\partial \left(\rho h_s\right)}{\partial t} + \nabla \cdot \left(\rho h_s \vec{U}\right) + \frac{\partial \rho K}{\partial t} + \nabla \cdot \left(\rho K \vec{U}\right) = -\nabla \cdot \vec{\boldsymbol{q}} + \frac{\partial p}{\partial t} + \bar{\bar{\tau}} : \nabla \vec{U} + \dot{Q}_r \tag{4}$$

where t is time, ρ is density, $\vec{U} = (u, v, w)$ is velocity vector, p is pressure, $Y_i, \vec{\boldsymbol{V}}_i, \dot{w}_i$ are the mass fraction, diffusion velocity vector and net reaction rate of the ith specie respectively. h_s denotes sensible enthalpy, K denotes kinematic energy, $\dot{Q}_r$ denotes net heat production rate, $\vec{\boldsymbol{q}}$ denotes heat flux.

In momentum equation, $\bar{\bar{\tau}}$ is viscous stress tensor. According Stokes assumption, $\bar{\bar{\tau}}$ is given by

$$\bar{\bar{\tau}} = -\frac{2}{3}\mu(\nabla \cdot \vec{U})\bar{\bar{\boldsymbol{I}}} + \mu\left[\nabla \vec{U} + \left(\nabla \vec{U}\right)^T\right] \tag{5}$$

where μ is dynamic viscosity of mixture and $\bar{\bar{\boldsymbol{I}}}$ is unit tensor. μ is computed using the Wilke mixture rule

$$\mu = \sum_i \frac{\mu_i X_i}{\sum_j \Phi_{i,j} X_j}$$

Here μ_i is the viscosity of the ith specie, X_i is the mole fraction of the ith specie and

$$\Phi_{i,j} = \frac{\left[1 + \sqrt{\left(\frac{\mu_i}{\mu_j}\sqrt{\frac{M_j}{M_i}}\right)}\right]^2}{\sqrt{8}\sqrt{1 + M_i/M_j}}$$

where M_i is the mole mass of the ith specie.

In species transport equations, the diffusion velocity $\vec{\boldsymbol{V}}_i$ is modeled in the mixture averaged formula

$$\vec{\boldsymbol{V}}_i = -\frac{D_i}{X_i}\nabla X_i \quad \text{with} \quad D_i = \frac{1 - Y_i}{\sum_{j \neq i} X_j / \mathcal{D}_{ji}} \tag{6}$$

where D_i is the mixture-averaged diffusion coefficient between ith specie and the rest of the mixture. $\mathcal{D}_{ji}$ is binary diffusion coefficient between jth specie and ith specie, estimated according to Takahashi correlation [16]. To conserve the global mass, the transport equations for all species except the "inert specie" (i.e., N2) are solved. The "inert specie" is explicitly calculated by subtracting the sum of the remaining mass fractions from unity. Similarly, the diffusion velocity of the inert specie is explicitly calculated according $\sum_{i=1}^{N} \rho Y_i \vec{\boldsymbol{V}}_i = 0$ to ensure the net species diffusion flux is zero, where N is the number of species.

In energy equation, the heat flux $\vec{\boldsymbol{q}}$ is caused by two parts: heat conductive and mass diffusion of species, which is given by:

$$\vec{\boldsymbol{q}} = -\lambda \nabla T + \rho \sum_{i=1}^{N} h_{s,i} Y_i \vec{\boldsymbol{V}}_i \tag{7}$$

where λ is the thermal conductivity of mixture, T is temperature, $h_{s,i}$ is the sensible enthalpy of the ith specie. λ is computed from the following mixture rule:

$$\lambda = 0.5\left(\sum_i X_i \lambda_i + \frac{1}{\sum_i X_i / \lambda_i}\right)$$

Here λ_i is the viscosity of ith specie. All of the aforementioned transport coefficients of pure specie are computed by using expressions based on Chapman-Enskog theory [16].

Since temperature is not a prime variable used in this partial deference equations, it is convenient to cast temperature to enthalpy. This derivation starts from the expression of the enthalpy gradient for a multicomponent mixture, which is given by:

$$\nabla h = c_p(Y_k, T)\nabla T + \sum_{k=1}^{N} h_k(T)\nabla Y_k \tag{8}$$

Then the first term in Eq. (7) can be expressed as

$$\begin{aligned}\lambda \nabla T &= \frac{\lambda}{c_p}\left(\nabla h - \sum_{k=1}^{N} h_k \nabla Y_k\right) \\ &= \frac{\lambda}{c_p}\left(\nabla h_s + \sum_{k=1}^{N} h_{c,k}\nabla Y_k - \sum_{k=1}^{N} h_{c,k}\nabla Y_k - \sum_{k=1}^{N} h_{s,k}\nabla Y_k\right) \\ &= \frac{\lambda}{c_p}\left(\nabla h_s - \sum_{k=1}^{N} h_{s,k}\nabla Y_k\right)\end{aligned} \tag{9}$$

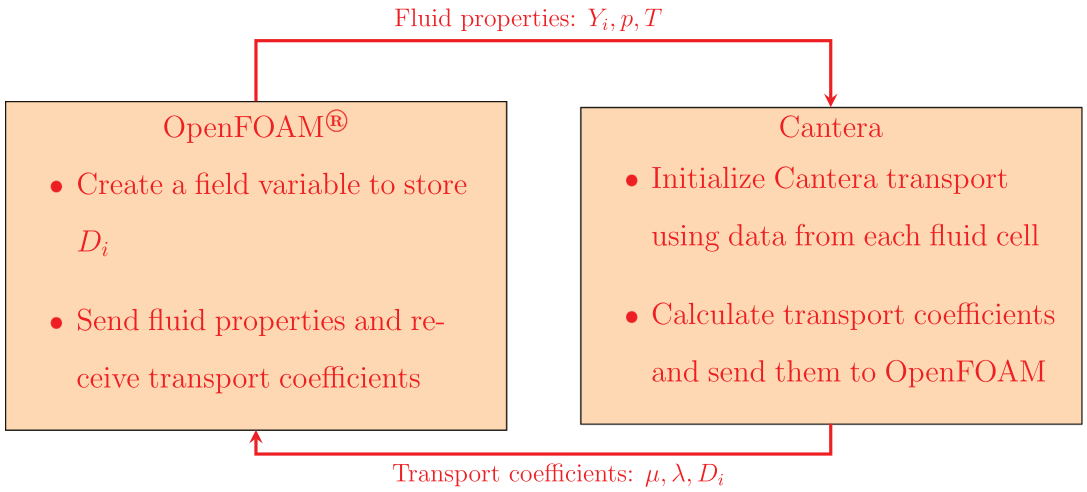

Fig. 1. Data exchange between OpenFOAM® and Cantera.

Finally, the transport equation for sensible enthalpy is given by

$$\begin{aligned}\frac{\partial(\rho h_s)}{\partial t} &+\nabla\cdot\left(\rho h_s\vec{U}\right)+\frac{\partial\rho K}{\partial t}+\nabla\cdot\left(\rho K\vec{U}\right)-\frac{\partial p}{\partial t}-\overline{\overline{\tau}}:\nabla\vec{U}-\dot{Q}_r\\ &=\nabla\cdot\left(\tfrac{\lambda}{c_p}\nabla h_s\right)-\underbrace{\sum_{k=1}^{N}\nabla\cdot\left(\frac{\lambda}{c_p}h_{s,k}\nabla Y_k\right)}_{\text{contribution due to casting from temperature to enthalpy}}\\ &-\underbrace{\sum_{k=1}^{N}\nabla\cdot\left(h_{s,k}\rho Y_k\vec{\boldsymbol{V}}_k\right)}_{\text{contribution from relative motion of components}}\end{aligned} \tag{10}$$

To closed the equations, the ideal gas assumption is adopted. According to Dalton's law, the total pressure exerted is equal to the sum of the partial pressures of the individual gases. The pressure of mixture can be defined as:

$$p=\sum_{i=1}^{N}p_i=\sum_{i=1}^{N}\rho_i R_i T=\sum_{i=1}^{N}\rho_i\frac{R_0}{M_i}T \tag{11}$$

where, $R_0=8.314J/(\text{mol}\cdot\text{K})$ is universal gas constant.

Note that the species and energy equations solved in *reactingFoam*-solver is simplified. Compared with the proposed solver, there are three simplifications for *reactingFoam*-solver:

- In energy equation, the contribution from viscous force is neglected which may be a significant heat source in compressible flows.
- The energy transport due to species diffusion is neglected. The Lewis number is assumed to equal to one, i.e. the thermal conductivity is derived from the diffusion coefficient :

$$Le=\frac{\lambda}{Dc_p\rho}=1\rightarrow\frac{\lambda}{c_p}=\rho D \tag{12}$$

  Then the heat flux term q is simplified as $\frac{\lambda}{c_p}\nabla h_s$
- In species equations, the specie mass diffusion is derived from momentum diffusion, which is expressed as $\rho Y_i\vec{\boldsymbol{V}}_i=\mu\nabla Y_i$. This is actually based on the assumption that the Schmidt number is one:

$$Sc=\frac{\mu}{\rho D}=1\rightarrow\rho D=\mu \tag{13}$$

3. Numerical method

3.1. Couple Cantera with OpenFOAM®

Multicomponent transport coefficients include viscosity coefficient μ, thermal conductivity coefficient λ and diffusion coefficient D_i. These coefficients can be calculated by molecular dynamics method, but are not trivial. Cantera [13] is an open-source suite of tools for problems involving chemical kinetics, thermodynamics, and transport processes. It is convenient to be used in applications written in C++. Therefore, we couple Cantera with OpenFOAM® to calculate these coefficients in the present work. Fig. 1 shows the data exchange between OpenFOAM® and Cantera. There are no variables for storing diffusion coefficients in the original OpenFOAM®'s thermo-physical libraries, therefore we create a new field variable to store D_i in each control volume first. OpenFOAM® send state parameters of the flow field to Cantera for initialization. Cantera calculate transport coefficients, as described in Section 2, and send them to OpenFOAM® back. More detailed description of when and where to exchange data during the loop of calculation is provided in subsection 3.3. The description of implementation is provided in Appendix A.

3.2. Schemes

The governing equations are approximated using finite volume method (FVM) in OpenFOAM®. The terms in governing equations can be classified as unsteady terms, convection terms, diffusion terms and source terms. Different terms are of different physical meanings and need to be treated with different numerical schemes. Many schemes are already available in the standard OpenFOAM® library. In this work, the volume integral of diffusion terms is converted to surface integral using Gauss theorem to ensure conservation. The source terms are approximated using midpoint integration rule.

The convection terms require special attention to simulate flows with shocks. The mass flux, ϕ, is the inner product of velocity times the density on cell faces.

$$\phi=\left(\rho\vec{U}\right)_f\cdot\vec{S}_f\approx\rho_f\vec{U}_f\cdot\vec{S}_f \tag{14}$$

where subscript f denotes variables at cell face and $\vec{S}_f$ is product of face normal on its area. The approximation in

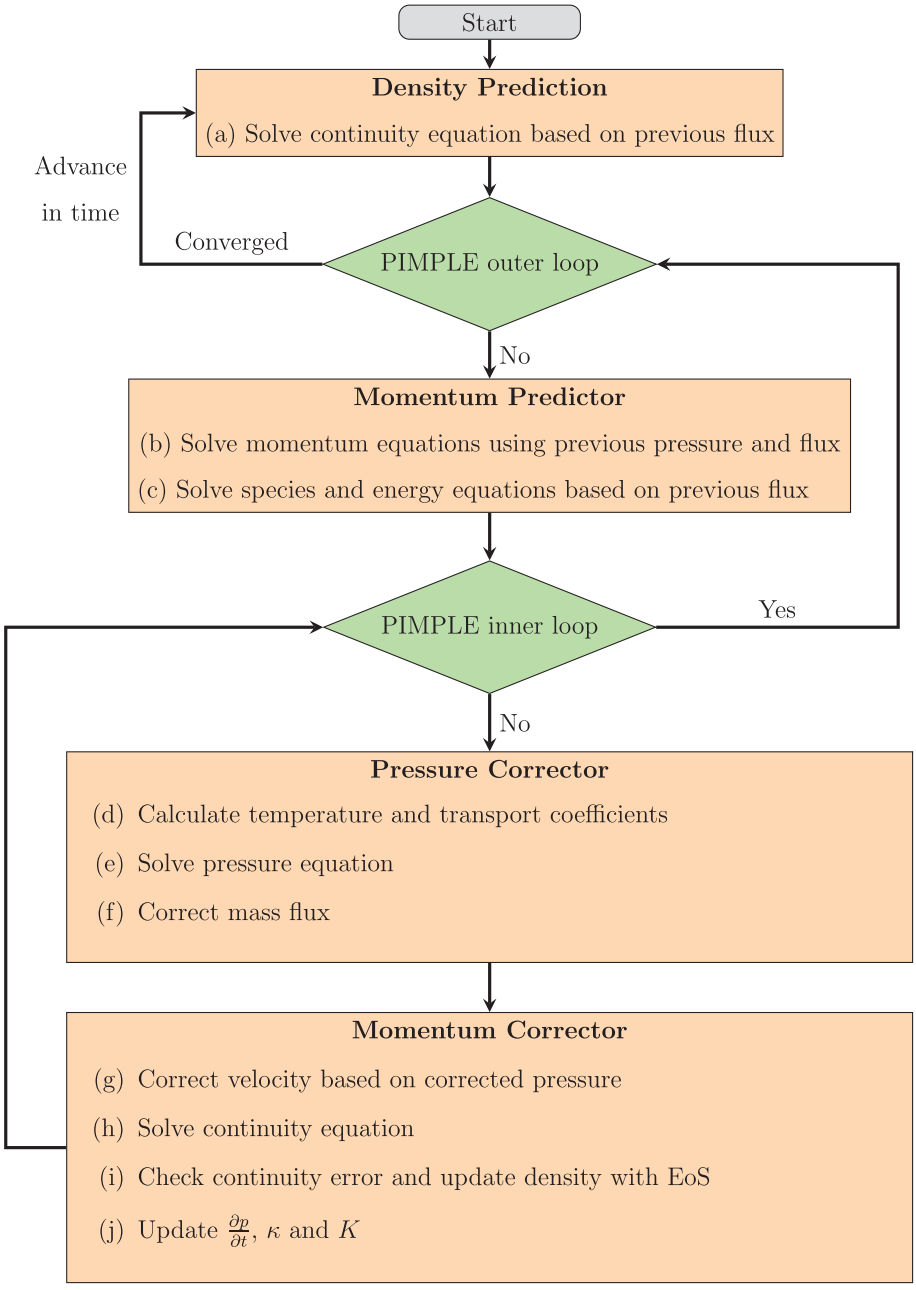

Fig. 2. Flowchart of the proposed solver.

Eq. (14) is used in *reactingFoam*-solver and commonly found in many other solvers. However, it is not appropriate when shocks exit in flow fields. The presence of shocks in high speed compressible flows requires numerical schemes that can capture shocks while avoiding spurious oscillations. To solve this problem, a hybrid approach derived by Kraposhin et al. [18] is implemented into OpenFOAM®, which use a blending function to automatically switch flux calculations between Eq. (14) and Kurganov-Tadmor/Kurganov-Noelle-Petrova (KT/KNP) scheme.

Considering the flux on the face between "owner" cell and "neighbor" cell, the fluxes are split into portion that flows into "owner" cell (inlet) and portion that flows out of the "owner" cell (outlet). The maximum value of inlet volume flux is zero, given by

$$F_{\text{inlet}} = \min\left(\min\left(\frac{(\rho\vec{U})_f^+ \cdot \vec{S}_f}{\rho_f^+} - c_f^+|\vec{S}_f|, \frac{(\rho\vec{U})_f^- \cdot \vec{S}_f}{\rho_f^-} - c_f^-|\vec{S}_f|\right), 0\right) \tag{15}$$

where c is local speed of sound, superscript + and - denote inlet and outlet values respectively. The minimum value of outlet volume flux is zero, given by

$$F_{\text{outlet}} = \max\left(\max\left(\frac{(\rho\vec{U})_f^+ \cdot \vec{S}_f}{\rho_f^+} + c_f^+|\vec{S}_f|, \frac{(\rho\vec{U})_f^- \cdot \vec{S}_f}{\rho_f^-} + c_f^-|\vec{S}_f|\right), 0\right) \tag{16}$$

The positive mass flux is a weighted value, given by

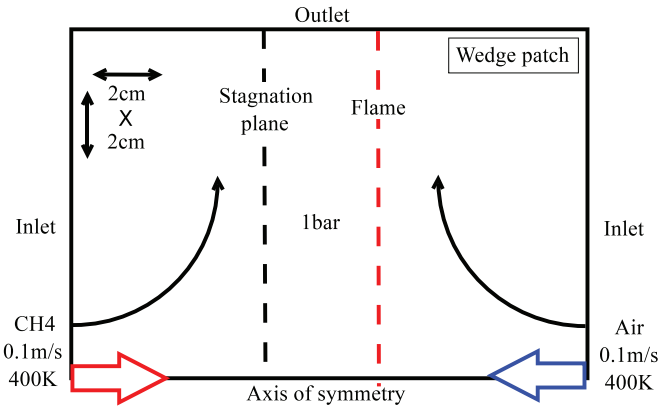

Fig. 3. Schematic of methane/air non-premixed counterflow flame.

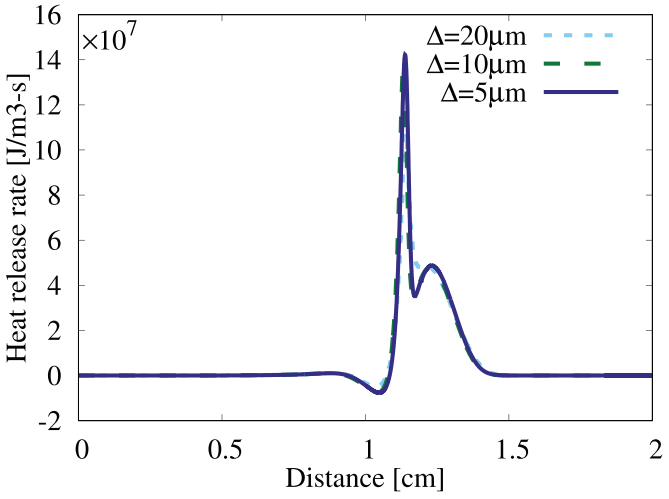

Fig. 4. Heat release rate along axis of symmetry with different grid resolution. Δ denotes the minimum mesh size in difference cases.

$$\phi^{+} = \rho^{+}\left(\alpha^{+} \cdot \frac{\left(\rho\vec{U}\right)_f^{+} \cdot \vec{S}_f}{\rho_f^{+}} - \alpha^{+} \cdot F_{\text{inlet}}\right) \tag{17}$$

and the negative mass flux is

$$\phi^{-} = \rho^{-}\left(\alpha^{-} \cdot \frac{\left(\rho\vec{U}\right)_f^{-} \cdot \vec{S}_f}{\rho_f^{-}} + \alpha^{+} \cdot F_{\text{inlet}}\right) \tag{18}$$

where

$$\alpha^{+} = \frac{F_{\text{outlet}}}{F_{\text{outlet}} - F_{\text{inlet}}}, \quad \alpha^{-} = \frac{-F_{\text{inlet}}}{F_{\text{outlet}} - F_{\text{inlet}}} \tag{19}$$

The mass flux is derived from combine ϕ^{+} with ϕ^{-} through a blending function κ_f

$$\phi \approx \left(\phi^{+} + (1-\kappa_f)\phi^{-}\right) + \kappa_f\phi^{-} \tag{20}$$

where

$$\kappa_f = \min\left(\frac{|\vec{U}_f \Delta x_f|}{c_f^2 \Delta t}, 1\right) \tag{21}$$

To calculate variable values with superscript + and - , we use vanLeer limiter. All schemes used in this work are listed in Table 1, where SnGrad means component of gradient normal to the cell face and default is a keyword in OpenFOAM® which means all items, except for the ones that are listed.

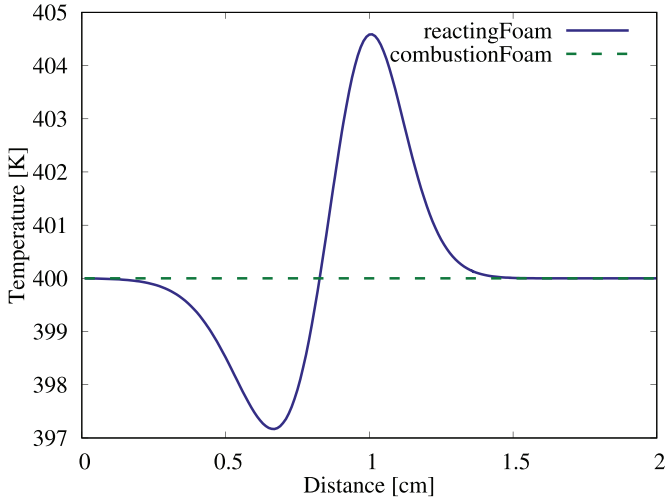

Fig. 5. Temperature profile along axis of symmetry for pure gas mixing process.

Table 1
Schemes used in this work.

	Term	Scheme
Time	Default	Euler
Gradient	Default	Gauss linear
Divergence	$\nabla\cdot\left(\mu\left(\nabla\vec{U}\right)^{T} - \frac{2}{3}\left(\nabla\cdot\vec{U}\right)\bar{\bar{I}}\right)$	Gauss linear
	$\nabla\cdot\left(\frac{\lambda}{c_p}h_{s,k}\nabla Y_k\right)$	Gauss linear
	$\nabla\cdot\left(h_{s,k}\rho Y_k\vec{\mathbf{V}}_k\right)$	Gauss linear
	Default	Gauss vanLeer
Laplacian	Default	Gauss linear corrected
Interpolation	Default	linear
SnGrad	Default	Corrected

3.3. Algorithm

To solve the governing equations of compressible flows, the following methods can be adopted:

- Explicit method: such as explicit Godunov-type methods, etc.
- Coupling method: all equations are coupled into a large linear system and are solved simultaneously.
- Operator splitting method: such as PISO, SIMPLE, PIMPLE, etc.

The explicit method is inefficient for low speed flow and coupling method consume more memory and time. Most solvers in OpenFOAM® adopt operator splitting method.

The algorithm implemented in this work are based on the classical PIMPLE algorithm, which is widely used in OpenFOAM®. PIMPLE algorithm is a pressure-correction method. The coupling of pressure-velocity-density is handled in a segregated method, i.e. while solving for one variable, other variables are treated as known. To improve the stability of the solver, the original PIMPLE algorithm was modified. The flowchart of the modified algorithm are shown in Fig. 2.

To obtain the solution at new time level, several outer iterations are performed. The density is updated by solving continuity equation based on previous mass flux (explicit) first (step (a)). Then the momentum equations are solved to obtain predicted velocities (step (b)). The velocities do not satisfy the mass conservation equations since they are based on 'old' pressure and 'old' mass flux. So several inner iterations are performed to correct them. For compressible flows, mass flux depend on both density and velocity. Hence, it is critical to correct both density and velocity. In the in-

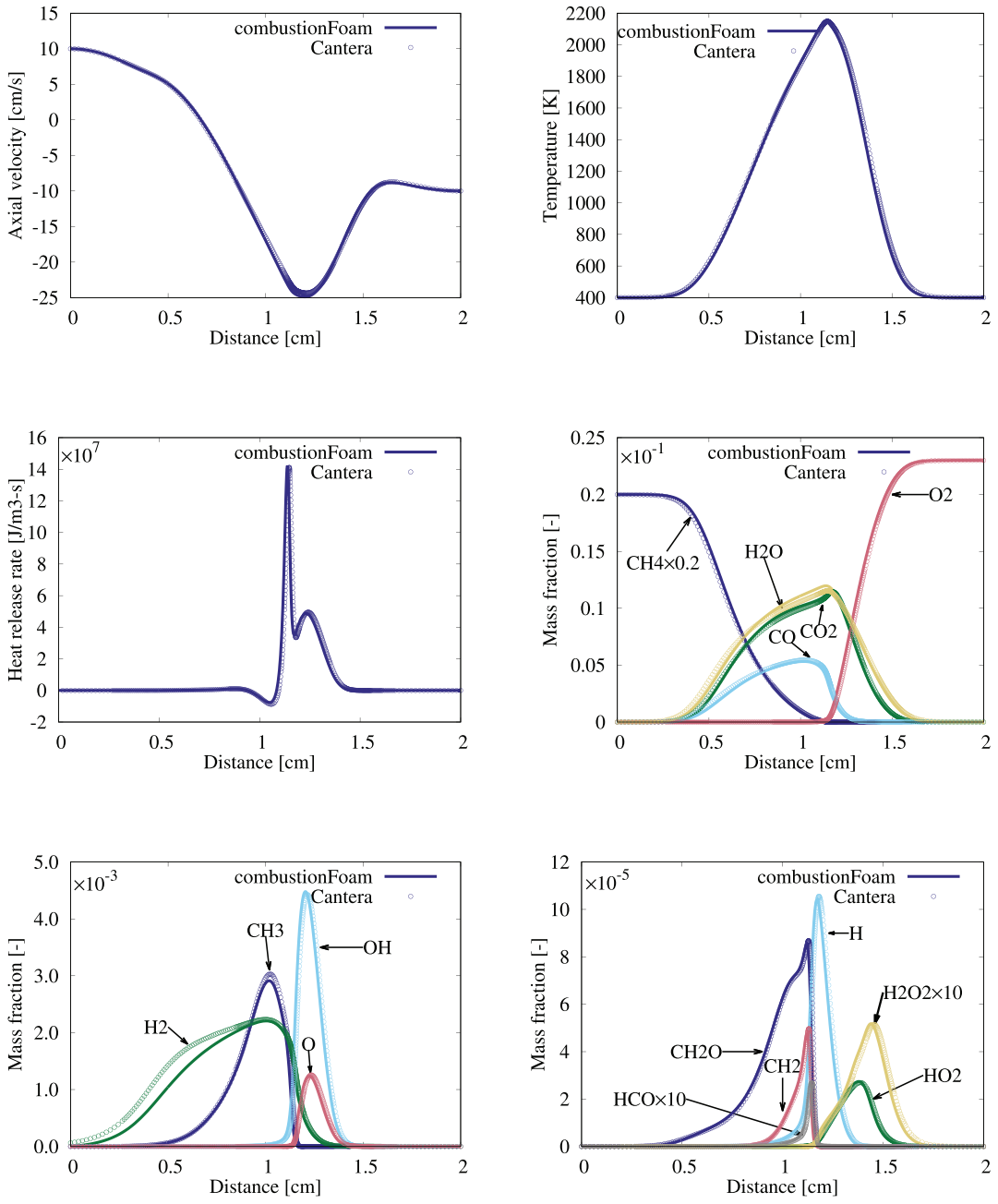

Fig. 6. Axial velocity, temperature, heat release rate and species mass fraction profiles obtained from OpenFOAM® (lines) and from Cantera(circles).

ner iterations, species and energy equations are solved first to find intermediate species and sensible enthalpy (step (c)). Thereafter, the temperature and transport coefficients are updated based on these intermediate species and sensible enthalpy (step (d)). Then a pressure equation derived from continuity and momentum equations (the predicted velocity is multiplied by ρ and inserted into the continuity equation) is solved to find a intermediate pressure (step (e)), which is

$$\frac{\partial \psi p}{\partial t} + \nabla \cdot (\psi \mathbf{HbyA}_{\mathrm{P}} \cdot p) - \nabla \cdot \left(\frac{\rho}{\mathbf{A}_{\mathrm{P}}} \nabla p \right) = 0 \tag{22}$$

where $\psi = \rho / p$ is compressibility, $\mathbf{HbyA}_{\mathrm{P}}$ is predicted velocity, $\mathbf{A}_{\mathrm{P}}$ is diagonal matrix of the velocity equation. To maintain consistency among the approximations used in continuity and momentum equations, the pressure equation is derived from the discrete

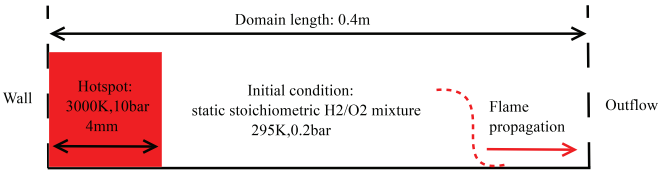

Fig. 7. Schematic of a transient one dimensional H2/O2 detonation.

momentum and continuity equations rather than by approximating a pressure equation directly. Using this intermediate pressure, the mass flux ϕ is corrected by $\phi = (\mathbf{HbyA}_{\mathrm{P}})_f - \left(\frac{\rho}{\mathbf{A}_{\mathrm{P}}} \nabla p \right)_f$, where the

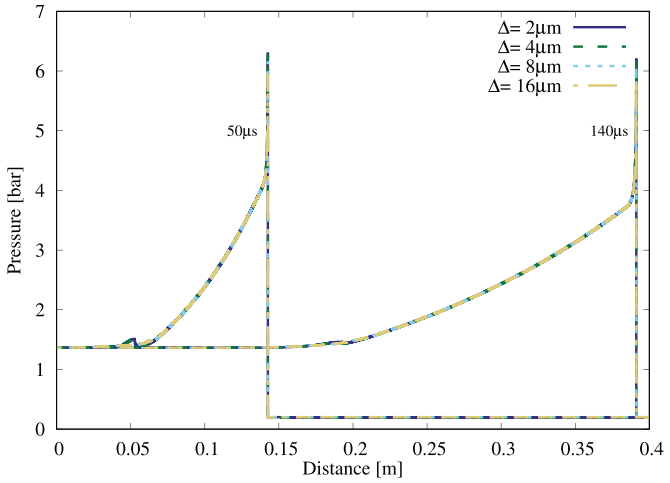

Fig. 8. Pressure profiles with different grid resolution. Δ denotes the minimum mesh size in difference cases.

subscript f means variables on face of control volume.(step (f)). The the velocity is corrected by $\vec{U} = \mathbf{HbyA}_\mathrm{P} - \frac{1}{\mathbf{A}_\mathrm{P}}\nabla p$ (step (g)). Note that the density can be updated by solving continuity equation (step (h)) or by EoS. The density should be the same either way. Therefore, error test is conducted by comparing the density difference between the two (step (i)). Finally, $\frac{\partial p}{\partial t}$, κ and K is updated based on new variables (step (j)). For most of the flow problems, two or three inner loops can already get sufficiently accurate solutions, and there is no need for more corrections. The outer loops are controled by user-defined residuals. More details of PIMPLE algorithm are recommend to refer to Ferziger and Perić [10].

4. Validation and discussion

Generally, a systematically verification study should be conducted to verified every terms in governing equations one by one, including chemical reaction source terms (zero dimensional ODE solver), convection terms and diffusion terms. However, one benefit from developing new solvers based on OpenFOAM® is that many validations have already been done, such as, a series of zero dimensional ignition delay test for different fuels (H2, CH4, nC7H16, iC8H18) have already been conducted in OpenFOAM®'s tutorials cases using SEULEX ODE solver [22]. The results obtained from OpenFOAM® are compared with data from Chemkin and good agreement are obtained. Therefore, we only chose two canonical test cases to verify the following aspects of the code:

- Multiple species transport models. The viscous coefficient of the mixture, mass diffusion coefficients of species into mixture and thermal conductivity of the mixture are calculated by Cantera. It is important to verify that all these transport coefficients are calculated correctly. This is done by comparing fields of a non-premixed counterflow flame, where diffusion is critical.
- Applicability to supersonic reacting flows. One feature of *combustionFoam*-solver is that it can simulate reacting flows at arbitrary Mach number. Hence, an onset and propagation of detonation wave simulation is conducted and compared with data obtained from *ASURF* and literature.

4.1. Steady two dimensional non-premixed counterflow flame

In this case, a steady methane/Air counterflow flame is simulated to validate the diffusion flame structures. Schematic of the test case is shown in Fig. 3. The computation domain is a section of cylinder and size is 2cm × 2cm. Methane is injected from the left side and air is injected from the right side. Both inlet velocity and temperature are 0.1m/s and 400K respectively. The operating pressure is set as 1bar. For the purpose of saving computing time, the chemical kinetic mechanism developed by Kee et al. [15] is adopted here, which involves 17 species and 58 reactions. Note that the results obtained in Cartesian coordinate system are different with that in cylindrical coordinate system. All simulations in this work are conducted in cylindrical coordinate system.

To get better resolution near the flame front, the mesh for cases solved by *combustionFoam*-solver is divided into three parts. In the axis direction, a refined uniform discretization is adopted within 1 cm to 1.3 cm from the fuel inlet and a gradient discretization is adopted in other regions. A gradient discretization is adopted along the radial direction. Firstly, the grid independence is verified. This is done by comparing heat release rate along axis of symmetry, as shown in Fig. 4, since heat release rate is very sensitive to grid resolution. It is seen that the mesh with a minimum size of 20 μm does not resolve the second heat release rate peaks. Comparing the amplitude and position of heat release rate, the results have converged at mesh with a minimum size of 5 μm.

The initial internal temperature is set as 400K first. Thus this case simulates a pure mixing process of methane/air. This case is calculated using both *reactingFoam*-solver and *combustionFoam*-solver. The schemes and algorithm used in *reactingFoam*-solver are identical to the ones using in OpenFOAM®'s tutorial case *counterFlowFlame2D_GRI*. The temperature profile along axis of symmetry is shown in Fig. 5, where we found that there is a non-physical discrepancy when using *reactingFOAM*-solver. This is due to ignoring the sensible enthalpy introduced from relative motion of components in *reactingFOAM*-solver (see Eq. (10)). This

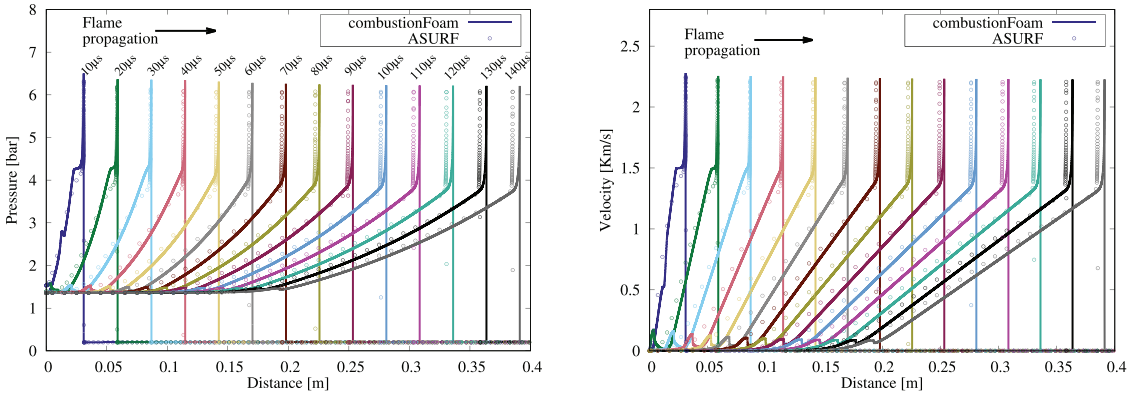

Fig. 9. Basic characteristics of flow fields at different time for stoichiometric H2/O2 mixture detonation wave.

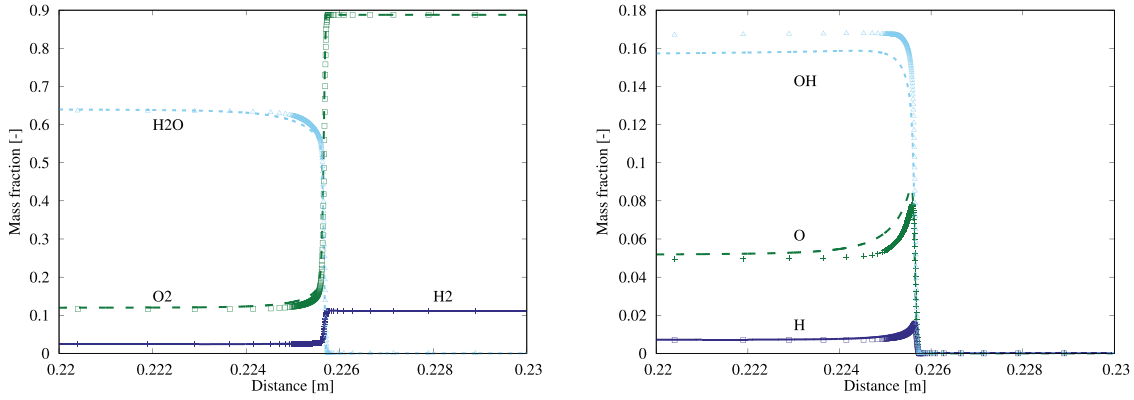

Fig. 10. Mass fraction of species at 80μs. Lines denotes *combustionFoam*-solver results and points denotes *ASURF* results.

discrepancy may also occurs in combustion simulations. Similar results have also been reported in literature [25]. The governing equations along the axis of symmetry can be reduced to a single dimension normal to the flame. The reduced governing equations can be solved using Cantera. Hence, an opposed-flow methane/air diffusion flame simulation using the same transport model, same chemical mechanism, same boundary conditions and same domain size is conducted in Cantera. An adaptive mesh refinement technique is used in Cantera. The initial total grid number is set as 50. Additional points will be added if the ratio of the spacing, value slope and curve exceeds 2, 0.03 and 0.08 respectively. Such that the minimum mesh size in Cantera is approximately $5\mu m$ which is the same with simulations in *combustionFoam*-solver.

Fig. 6 compares axial velocity, temperature, heat released rate and mass fractions of all species along the axis of symmetry with the results obtained by Cantera. According to the order of magnitude, the mass fraction of components is divided into three groups. In general, the comparison is seen to be very favorable, demonstrating the success of OpenFOAM-Cantera coupling. The maximum axial velocity and stagnation location are well predicted in the velocity configuration. The temperature profile also shows good agreements. Two heat release peaks and heat absorption region are well captured as well. It is also seen that most components show good level of quantitative predictability. The maximum discrepancy appears in the mass fraction of H2. This is due to the chemical effect rather than diffusion effect, since hydrogen atom possesses greater diffusivity than hydrogen, but shows excellent agreement.

4.2. Transient one dimensional detonation

In this case, an onset and propagation of detonation waves are conducted. The schematic of the test case is shown in Fig. 7. The computation domain is a 0.4m long one-dimensional pipe. The left end of the pipe is closed and the right end is open. The pipe is filled with a static stoichiometric H2/O2 mixture initially. A small hotspot of high-temperature and high-pressure gas is patched at the left side to ignite the mixture. Simulations were performed by using both *combustionFoam*-solver and *ASURF*. A grid independence study is conducted by comparing pressure profiles with different uniform grid resolution as shown in Fig. 8. It is seen that the results have converged at $\Delta = 4\mu$m. For cases solving by *ASURF*, an adaptive mesh is used (grid level is 9, root grid size is $\Delta x_0 = 1mm$ and minimum grid size is $\Delta x_8 \approx 4\mu m$). The chemical kinetic mechanism developed by Burke et al. [4] is adopted in both solvers, which involves 13 species and 27 reactions. This mechanism was previously validated against high-pressure conditions.

Table 2
Comparing variables at CJ point and von Neumann spike point.

Data source	D_{CJ} (m/s)	P_{CJ} (bar)	T_{CJ} (K)	u_{vN} (m/s)	P_{vN} (bar)	T_{vN} (K)
ASURF	2718.8	3.4	3369	2203.0	6.1	1789
combustionFoam	2760.8	3.5	3381	2225.8	6.2	1774
Schultz and Shepherd [20]	2753.2	3.6	3391	2252.7	6.3	1682

Initial conditions at high temperature and pressure introduce a shock wave at first. The high temperature and high pressure region behind the wave ignites the mixture to form a detonation wave. At this point, the detonation is in an overdriven state. The propagation velocity of the overdriven detonation wave is higher than CJ (Chapman-Jouguet) detonation velocity. Therefore, the front of detonation wave soon catches up with the front shock wave, and the pressure drops gradually to form a stable detonation wave. After that, although the von Neumann spike pressure fluctuate, it generally reached a stable state, maintaining around 6bar.

Fig. 9 shows the computed pressure and velocity profiles at different times using both *combustionFoam*-solver and *ASURF*. It is seen that the leading shock wave and the chemical reaction zone following it constitute the front of detonation wave. And then is the rarefaction wave region, where the gas velocity decreases gradually. Finally, transition to the stable zone, where the speed is zero and the pressure remains constant. This result agrees well with the self-similar solution of CJ model. The main difference is that the detonation wave structure cannot be described by CJ model. It is also seen that there is a velocity wave in stable region (see combustionFoam results), which is caused by the ignition energy. However, this velocity waves attenuated quickly for the results of *ASURF*. This is due to numerical dissipation, since the base mesh used in *ASURF* is too coarse for DNS. This point is also demonstrate in Fig. 8 for the pressure profiles obtained by different grid resolution. Therefore, results from *combustionFoam*-solver are more convincing than that from *ASURF*. It can also be concluded that the mesh refinement or coarsening criteria need special attention for DNS using adaptive mesh refinement technique.

Variables at CJ point and von Neumann spike point are compared in Table 2. The CJ detonation velocity, D_{CJ}, can be fitted by time-distance curve. The CJ point is identified by CJ condition, $D_{CJ} = u_{CJ} + a_{CJ}$, where u_{CJ} and a_{CJ} denote the gas particle velocity of the detonation product and the acoustic speed in the detonation

products behind the detonation front, respectively. In general, the results calculated by *combustionFoam*-solver are pretty close to that by *ASURF*-solver and in literature [20]. Considering the gradient of the variable is very large, the results are acceptable.

The advantage of simulations with detailed chemical reactions is that it can obtain not only the evolution of pressure, velocity, density and temperature, but also the concentration changes of each component involved in the chemical reaction, that is, the release process of chemical energy. Fig. 10 compares the main component concentration distribution at 80 μs between *combustionFoam*-solver and *ASURF*. As can be seen from the Fig. 9 and Table 2, the detonation velocity estimated by *ASURF* is slightly lower. Therefore, the data from *ASURF* are shifted 3.4mm for the convenience of components comparison. It is seen that the chemical reaction is mainly carried out in the region between the leading shock front and the CJ surface. H2 and O2 are rapidly consumed to generate H2O. O and H rapidly reaches a peak and then decays rapidly, then reaches a stable state. The chemical reaction does not finish completely at CJ surface. As the temperature decreases slightly in the rarefaction wave region, the chemical equilibrium moves in the direction of the generated water. The comparison shows that the results agree well both qualitatively and quantitatively.

5. Conclusions

This paper has presented a quasi-direct numerical simulation solver for compressible reacting flows with detailed chemical reactions. This solver is named *combustionFoam* and developed based on *reactingFoam*-solver in OpenFOAM®. OpenFOAM® is a famous open-source framework for CFD, but its ability for DNS of compressible reacting flows, especially when there are shocks in the flow field, is not satisfactory. In this paper, *reactingFoam*-solver and some other solvers used for DNS of reacting flows are reviewed. The improvements of *combustionFoam*-solver can be summarized as follows: (1) introducing a mixture-averaged formula transport model by coupling OpenFOAM® with Cantera; (2) capable of simulating flows at arbitrary Mach number without noteworthy changes in numerical implementation. Two test cases are carried out to validate the performance of *combustionFoam*-solver, including a low speed steady two dimensional non-premixed counterflow flame and a supersonic transient one dimensional denotation waves. Good agreements are obtained, demonstrating it is applicable to DNS of reacting flows. The present study also suggests that special attention should be paid to the selection of refine/coarse criteria when using adaptive grids. In the future, *combustionFoam*-solver will be used to study the experimental scale multi-physical interaction problems in industrial devices.

Declaration of Competing Interest

No conflict of interest exists.

CRediT authorship contribution statement

Tao Li: Conceptualization, Methodology, Software, Writing - original draft. **Jiaying Pan:** Methodology, Validation, Resources. **Fanfu Kong:** Visualization, Investigation. **Baopeng Xu:** Conceptualization, Writing - review & editing. **Xiaohan Wang:** Conceptualization, Project administration, Writing - review & editing.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 51976219), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA 21060102) and the National Key R&D Program of China (2018YFB1501500).

Appendix A. Class diagram for Cantera and OpenFOAM® coupling

The main classes diagram for coupling OpenFOAM® with Cantera are shown in Fig. A.11. Two new classes named *canteraWrapper* and *canterThermo* are defined. *canteraWrapper* inherits from *Cantera::IdealGasMix* and depends on *Cantera::Transport*. *Cantera::Transport* and *Cantera::IdealGasMix* are two classes defined in Cantera. They implement multicomponent transport properties for gas mixture. *canteraWrapper* is used to wrap Cantera. *canterThermo* depends on *canteraWrapper* and it also defines member functions to initialize gas mixture and calculate transport coefficients. The name of member functions in these classes are self-explanatory.

Since all functions and data we needed are defined in *canteraThermo*, the next thing is to link it with OpenFOAM®'s thermophysical libraries. The main classes of thermo-physical models in OpenFOAM® for combustion inherit from *basicSpecieMixture*, which is a class for mixtures consisting a number of molecular species. *multiComponentMixture* is a specialization of *basicSpecieMixture* which defines the mass fraction weighted average algorithm and provides an interface to get the raw specie thermodynamic data. Detailed transport coefficients of species are calculated through an object of *PtrList* $<$ *ThermoType* $>$ in the original thermo-physical libraries of OpenFOAM®, where *ThermoType* is a template parameter. The limitations of original thermo-physical libraries are explained in Section 2. To calculate transport model coefficients using Cantera, new private data members named *cMixture, Dimix* and new public member functions named *getCMixutre, calculateD* are added in this class definition. *cMixture* is an object of *canteraThermo*. All transport model coefficients calculated by Cantera are accessed through this attribute. *getCMixutre* is the interface to access it. Transport coefficients are variables of *volScalarField* type and declared in the *BasicThermo* class defination, where *BasicThermo* is a template parameter of *heThermo*. Since there are no variables to store diffusion coefficients of species in OpenFOAM®, *Dimix* is added in *multiComponentMixture* to store diffusion coefficients of species. *calculateD* is a member function to initialize or update *Dimix*.

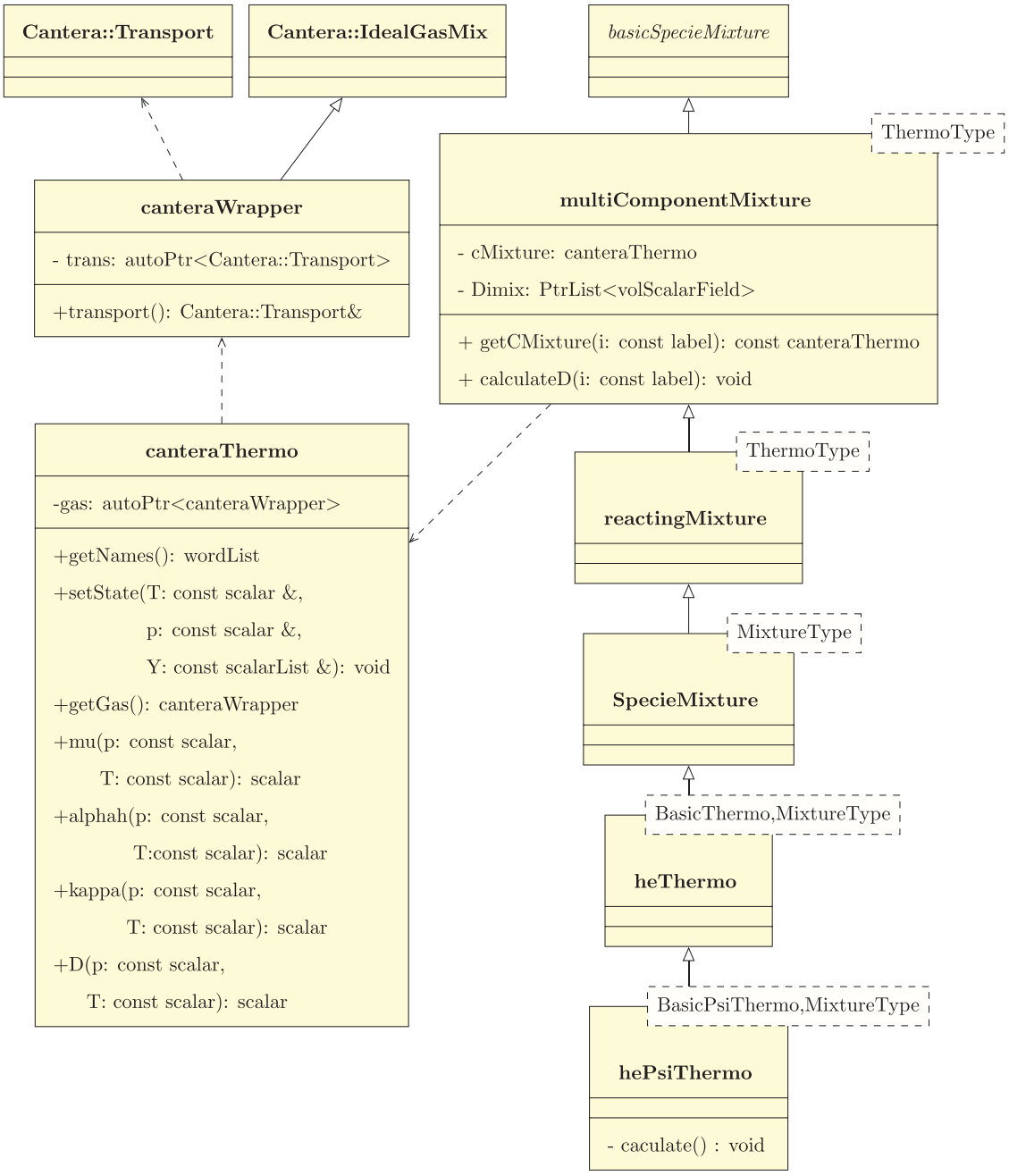

Fig. A.11. Class diagram for Cantera and OpenFOAM coupling. Normal arrow denotes inheritance and dotted arrow denotes dependency.

References

[1] Axtmann G, Rist U. Scalability of openFOAM with large eddy simulations and DNS on high-performance systems. In: High performance computing in science and engineering, vol. 16. Springer; 2016. p. 413–24.

[2] Babkovskaia N, Haugen N, Brandenburg A. A high-order public domain code for direct numerical simulations of turbulent combustion. J Comput Phys 2011;230(1):1–12.

[3] Bockhorn IH. Implementation and validation of a solver for direct numerical simulations of turbulent reacting flows in openFOAM. Karlsruhe Institute of Technology; 2012. Ph.d. thesis.

[4] Burke MP, Chaos M, Ju Y, Dryer FL, Klippenstein SJ. Comprehensive H2/O2 kinetic model for high-pressure combustion. Int J Chem Kinet 2012;44(7):444–74.

[5] Chen JH, Choudhary A, De Supinski B, DeVries M, Hawkes ER, Klasky S, et al. Terascale direct numerical simulations of turbulent combustion using S3D. Comput Sci Discov 2009;2(1):015001.

[6] Chen Z. Studies on the initiation, propagation, and extinction of premixed flames. Princeton University; 2009.

[7] Cuoci A, Frassoldati A, Faravelli T, Ranzi E. A computational tool for the detailed kinetic modeling of laminar flames: application to C2H4/CH4 coflow flames. Combust Flame 2013;160(5):870–86.

[8] Emmett M, Motheau E, Zhang W, Minion M, Bell JB. A fourth-order adaptive mesh refinement algorithm for the multicomponent, reacting compressible Navier–Stokes equations. Combust Theor Model 2019;23(4).

[9] Emmett M, Zhang W, Bell JB. High-order algorithms for compressible reacting flow with complex chemistry. Combust Theor Model 2014;18(3):361–87.

[10] Ferziger JH, Perić M. Computational methods for fluid dynamics, vol. 3. Springer; 2002.

[11] Fu Y, Yu C, Yan Z, Li X. DNS analysis of the effects of combustion on turbulence in a supersonic H2/air jet flow. Aerosp Sci Technol 2019;93:105362.

[12] Gibson J.F.. Channelflow. 2018. http://channelflow.org.

[13] Goodwin D.G., Speth R.L., Moffat H.K., Weber B.W.. Cantera: an object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. 2017. https://www.cantera.org, version 2.3.0.

[14] Jenkins KW, Cant RS. Direct numerical simulation of turbulent flame kernels. In: Recent advances in DNS and LES. Springer; 1999. p. 191–202.

[15] Kee R, Grcar J, Smooke M, Miller J. A fortran program for modeling steady laminar one-dimensional premixed flames. Sandia National Laboratories; 1985. report no. sand85-8240.

[16] Kee RJ, Coltrin ME, Glarborg P, Zhu H. Chemically reacting flow: theory and practice. 2nd ed. John Wiley and Sons; 2017.
[17] Komen E, Shams A, Camilo L, Koren B. Quasi-DNS capabilities of openFOAM for different mesh types. Comput Fluids 2014;96:87–104.
[18] Kraposhin M, Bovtrikova A, Strijhak S. Adaptation of Kurganov-Tadmor numerical scheme for applying in combination with the PISO method in numerical simulation of flows in a wide range of mach numbers. Procedia Comput Sci 2015;66:43–52.
[19] Liu Q, Gómez F, Perez J, Theofilis V. Instability and sensitivity analysis of flows using openFOAM®. Chin J Aeronaut 2016;29(2):316–25.
[20] Schultz E., Shepherd J.. Validation of detailed reaction mechanisms for detonation simulation. 2000.
[21] The OpenFOAM Foundation Ltd,. 2019. OpenFOAM: The open source computational fluid dynamics (CFD) toolbox. Version 7.
[22] Wanner G, Hairer E. Solving ordinary differential equations II. Springer Berlin Heidelberg; 1996.
[23] Yang Q, Zhao P, Ge H. reactingFOAM-SCI: an open source CFD platform for reacting flow simulation. Comput Fluids 2019.
[24] Ziegler JL, Deiterding R, Shepherd JE, Pullin D. An adaptive high-order hybrid scheme for compressive, viscous flows with detailed chemistry. J Comput Phys 2011;230(20):7598–630.
[25] Gopala Krishna Moorthy V K, Kampili M, Kelm S, Arul Prakash K, Allelein H-J. Development and verification of a multi-species gas transport solver. 14th OpenFOAM Workshop 2019.