



Contents lists available at ScienceDirect

# Journal of Computational and Applied Mathematics

journal homepage: [www.elsevier.com/locate/cam](http://www.elsevier.com/locate/cam)

## A conservative pressure-correction scheme for transient simulations of reacting flows

P. Rauwoens\*, J. Vierendeels, E. Dick, B. Merci<sup>1</sup>

Ghent University - UGent, Department of Flow, Heat and Combustion Mechs., Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium

### ARTICLE INFO

#### Article history:

Received 16 July 2008

Received in revised form 26 March 2009

#### MSC:

65N12

65N22

76D05

76M12

76R99

#### Keywords:

Low Mach

Pressure-correction

Non-premixed combustion

Conservation

Stability

### ABSTRACT

An algorithm is presented for numerical simulations of time-dependent low Mach number variable density flows with an arbitrary amount of scalar transport equations and a complex equation of state. The pressure-correction type algorithm is based on a segregated solution formalism. It is conservative and guarantees stable results, regardless of the difference in density between neighboring cells. Furthermore, states are predicted which exactly match the equation of state. In the one-dimensional example, considering non-premixed flames, a simplified flamesheet model is used to describe the combustion of fuel and oxidizer. We demonstrate that the predicted states exactly correspond to the equation of state. We illustrate the accuracy improvement due to higher order formulation and demonstrate grid convergence.

© 2009 Elsevier B.V. All rights reserved.

### 1. Introduction

Recently, large eddy simulations (LES) have become increasingly important in turbulent combustion simulations. They require time-accurate solutions. Common pressure-correction schemes, developed for incompressible flows, can no longer be used due to lack of stability [1,2]. Instabilities arise when the density variations, due to e.g. temperature variations, are too large. In [3], we showed that for a non-reacting single-fluid ideal-gas flow, a constraint for the velocity field can be formulated, such that the solution is stable. Later [4], the propositions of [3] were extended towards non-premixed combustion, making use of the mixture fraction as a conserved variable. A pressure-correction algorithm was obtained which (1) conserves mass, (2) conserves fuel elements' mass and (3) is stable and robust, without the need for (unphysical) rescaling factors. The latter is important for LES, where time accuracy of the transient simulations needs to be respected. The introduction of rescaling factors in the algorithm, for stabilization of the standard pressure-correction algorithms, corrupts this time accuracy.

The three above mentioned properties are naturally obtained for a linear equation of state, as for single-fluid ideal-gas and two-fluid inert mixing, with the pressure projection methods of [5,6]. When applied on more general equations of state, as in reacting flows, it is claimed in [7] that these three properties cannot be fulfilled at the same time. We disagree with this statement.

\* Corresponding author.

E-mail addresses: [pieter.rauwoens@ugent.be](mailto:pieter.rauwoens@ugent.be) (P. Rauwoens), [jan.vierendeels@ugent.be](mailto:jan.vierendeels@ugent.be) (J. Vierendeels), [erik.dick@ugent.be](mailto:erik.dick@ugent.be) (E. Dick), [bart.merci@ugent.be](mailto:bart.merci@ugent.be) (B. Merci).

<sup>1</sup> Postdoctoral fellow of the *Fund of Scientific Research Flanders (FWO Vlaanderen)*.

In the present paper, we repeat some key features of the algorithm, described in [4] and discuss a formulation for an arbitrary number of scalar equations and a general equation of state. We show that all scalars are conserved, mass is conserved and the equation of state is fulfilled in an exact manner. The algorithm is applied to a severe test case for stability, involving non-premixed combustion, serving as an illustrative example for more general flows. The flow is one-dimensional. An initial step in mixture fraction is convected by a constant velocity  $U$ . A Burke–Schumann flamesheet chemistry model is used to describe the combustion. Higher order monotone discretization schemes are compared. A grid refinement study shows that the discretization error converges to zero, still preserving the exact correspondence to the equation of state.

## 2. Governing equations

The basic equations are the Navier–Stokes equations for the flow field, with  $N$  additional scalar equations. The conservation of mass, momentum and scalars, with summation convention, reads:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} \quad (2)$$

$$\frac{\partial \rho y_\alpha}{\partial t} + \frac{\partial \rho u_i y_\alpha}{\partial x_i} = RHS_\alpha \quad (3)$$

with  $\alpha = 1, \dots, N$ . In these equations,  $\rho$  denotes the density,  $u$  the velocity,  $p$  the pressure and  $y_\alpha$  a generic scalar. The molecular viscous stress tensor  $\tau_{ij}$  is given by

$$\tau_{ij} = \mu \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] \quad (4)$$

with  $\mu$  the viscosity and  $\delta_{ij}$  the Kronecker delta.

For a general fluid at low Mach number, a general equation of state can be formulated, expressing that the state variables  $\rho$  and  $y_\alpha$  are not independent:

$$\mathcal{G}(\rho, y_1, \dots, y_N) = 0 \quad \text{or} \quad \mathcal{H}(\rho, \rho y_1, \dots, \rho y_N) = 0. \quad (5)$$

In the simulation examples, a non-premixed reacting fluid is considered, governed by the Navier–Stokes equations (1) and (2), and one additional scalar equation for mixture fraction  $\xi$  (e.g. [8]):

$$\frac{\partial \rho \xi}{\partial t} + \frac{\partial \rho u_i \xi}{\partial x_i} = \frac{\partial}{\partial x_i} \left( \rho D \frac{\partial \xi}{\partial x_i} \right), \quad (6)$$

with  $D$  the species diffusivity. The mixture fraction is a non-dimensional variable, equal to 1 in pure fuel and 0 in pure oxidizer and describes the mixing between the two components (fuel and oxidizer). Since the equation for mixture fraction is based on chemical elements, rather than on chemical species, there is no source term in the equation.

One can also write an equation for static enthalpy, which is the sum of formation enthalpy and sensible enthalpy:  $h = h^0 + c_p(T - T_{\text{ref}})$ . Under the commonly made assumptions in turbulent flow calculations of unity Lewis number (species diffusivity is equal to thermal diffusivity), in the absence of radiation and neglecting minor diffusion effects, such as Soret and Dufour effects, the enthalpy equation reads:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = \frac{\partial}{\partial x_i} \left( \rho D \frac{\partial h}{\partial x_i} \right) - \frac{dp_0}{dt}. \quad (7)$$

Unless the reactions take place in a closed environment, the thermodynamic pressure  $p_0$  is constant in space and time, and the enthalpy equation is similar to the equation for mixture fraction. Because, for a normalized enthalpy, the same boundary conditions apply as for the mixture fraction equation, Eq. (7) is superfluous and enthalpy can be algebraically derived from mixture fraction.

The equation of state is based on the ideal-gas law:  $\rho = p_0/(RT)$ , with  $R$  the gas constant in J/(kg K). The temperature follows from the definition of enthalpy and thus depends entirely on mixture fraction in case of constant  $p_0$ . We adopt a simple chemistry model, assuming irreversible and infinitely fast chemistry (Burke–Schumann flamesheet model [9]). In that case, fuel and oxidizer cannot be found together at the same place and only three species exist: fuel, oxidizer and products. The equation of state consists of two branches, on the lean and rich side of stoichiometry. The lowest density is found at stoichiometry, where temperatures are the highest. If we further assume, for simplicity, equal molecular weights for fuel, oxidizer and products, the equation of state results in a piecewise linear relationship  $\rho = \mathcal{H}_c(\rho \xi)$ :

$$\begin{aligned} \rho &= \rho_0 + \frac{\rho_{st} - \rho_0}{\rho_{st} \xi_{st}} \rho \xi \quad \text{for } \xi \leq \xi_{st} \\ \rho &= \rho_0 + \frac{\rho_{st} - \rho_0}{\rho_0 - \rho_{st} \xi_{st}} (\rho_0 - \rho \xi) \quad \text{for } \xi \geq \xi_{st}, \end{aligned} \quad (8)$$

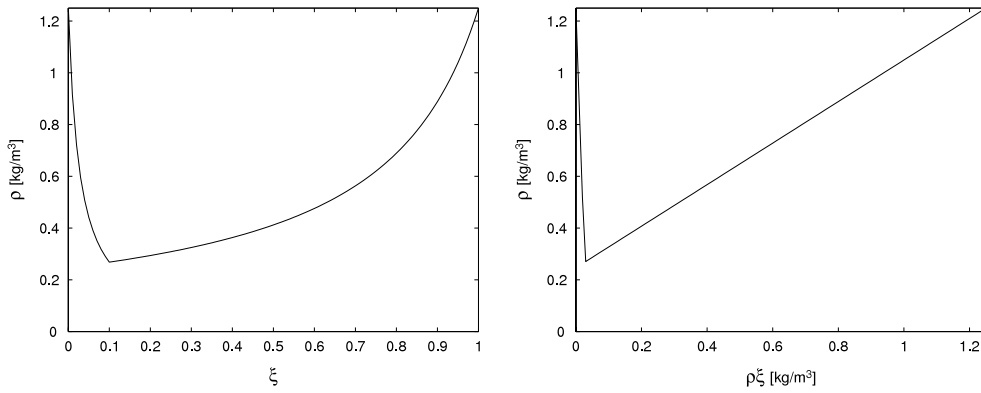


Fig. 1. Density as a function of mixture fraction ( $\xi$ ) and fuel elements' mass ( $\rho\xi$ ), if we assume all molecular weights to be equal.

depicted in Fig. 1. This equation of state is non-linear and non-differentiable in the stoichiometric point, and forms therefore a challenge for the algorithm. In the remainder, the following dimensional values will be used:  $\rho_0 = 1.25 \text{ kg/m}^3$ ,  $\rho_{st} = 0.27 \text{ kg/m}^3$ ,  $\xi_{st} = 0.1$ .

If the thermodynamic pressure is not constant, as is e.g. the case in internal combustion engines, the temperature is a function of enthalpy and mixture fraction, such that the equation of state has the form  $\rho = \mathcal{H}_c(\rho h, \rho\xi)$ , which is, under the same conditions and assumptions, a piecewise bilinear equation. More variables can be added, depending on the physics. For instance, if ignition or extinction needs to be modelled, more chemistry must be included, involving extra transport equations for *progress variables*. Also, especially in RANS turbulence modelling, turbulent fluctuations can be regarded as an extra scalar variable. The list of possible extra scalars is endless and shows the potential applicability of the presented algorithm for any scientific domain. In the examples below, the scalar equation for mixture fraction serves as an illustration for the general case.

### 3. The pressure-correction formalism

The pressure-correction formalism of [10–12] was originally developed for constant density flows. The general algorithm has proven to be accurate in these flows and no substantial problems are encountered there. The basic idea behind the pressure-correction strategy is to advance momentum in two steps.

First a prediction is made of velocity (or momentum), using the momentum equations with the pressure term evaluated at time level  $n$ :

$$(\rho u_j)^* = (\rho u_j)^n + \Delta t \left[ -\frac{\delta(\rho u_i u_j)^n}{\delta x_i} - \frac{\delta p^n}{\delta x_j} + \frac{\delta \tau_{ij}^n}{\delta x_i} \right]. \tag{9}$$

The  $\delta$ -notation is introduced to stress the fact that the derivatives are discrete.

Ultimately, the following equation, with the pressure, an acoustic term, evaluated implicitly at time level  $n + 1$ , must be satisfied:

$$(\rho u_j)^{n+1} = (\rho u_j)^n + \Delta t \left[ -\frac{\delta(\rho u_i u_j)^n}{\delta x_i} - \frac{\delta p^{n+1}}{\delta x_j} + \frac{\delta \tau_{ij}^n}{\delta x_i} \right].$$

The predicted field is corrected to give the velocity at the new time level

$$(\rho u_j)^{n+1} = (\rho u_j)^* + (\rho u_j)' \tag{10}$$

with the correction for the momentum  $(\rho u_j)'$  related to the correction for the pressure  $p' = p^{n+1} - p^n$ :

$$(\rho u_j)' = -\Delta t \frac{\delta p'}{\delta x_j}. \tag{11}$$

The correction for the pressure follows from inversion of an elliptic equation, based on a constraining equation for the velocity field at time level  $n + 1$ . In constant density flows, the continuity equation naturally imposes a constraint on the velocity field:

$$\frac{\delta(\rho u_i)^{n+1}}{\delta x_i} = 0, \tag{12}$$

so that the pressure equation in this case is:

$$\frac{\delta^2 p'}{\delta x_i^2} = -\frac{1}{\Delta t} \frac{\delta(\rho u_i)^*}{\delta x_i}. \quad (13)$$

In variable density flow, the constraining equation can be built in multiple ways and this forms the major difference between several pressure-correction formalisms. The key to obtain a pressure-correction algorithm that satisfies all prerequisites of conservation and state equation fulfillment, is to build a consistent constraining equation for the velocity (or momentum), as is done in the next section. This is a generalization of what was presented in [3].

#### 4. Algorithmic strategy for general incompressible fluid

The starting point are the conservation equations (1)–(3). We assume that the right hand side of the transport equations, containing diffusive and source terms, is discretized conservatively. If a first order time stepping is used, the continuity and scalar equations are discretized in time as:

$$\rho^{n+1} = \rho^n - \Delta t \frac{\delta(\rho u_i)^n}{\delta x_i} \quad (14)$$

$$(\rho y_\alpha)^{n+1} = (\rho y_\alpha)^n - \Delta t \frac{\delta(\rho u_i y_\alpha)^n}{\delta x_i} + \Delta t \text{RHS}_\alpha^n, \quad (15)$$

which provides the values of density and scalars at the new time level. As explained in Section 3, the velocity is determined in two steps. The predicted value follows from Eq. (9), whereas the ultimate value at time level  $n+1$  follows from a constraining equation. A constraint for  $u_i^{n+1}$  is found by combination of (14) and (15), shifted to the next time level:

$$\rho^{n+2} = \rho^{n+1} - \Delta t \frac{\delta(\rho u_i)^{n+1}}{\delta x_i} \quad (16)$$

$$(\rho y_\alpha)^{n+2} = (\rho y_\alpha)^{n+1} - \Delta t \frac{\delta(\rho u_i y_\alpha)^{n+1}}{\delta x_i} + \Delta t \text{RHS}_\alpha^{n+1}. \quad (17)$$

The constraint is now formulated by requiring that the equation of state is fulfilled at every time level, in particular at time level  $n+2$ :

$$\mathcal{H}(\rho^{n+2}, (\rho y_1)^{n+2}, \dots, (\rho y_N)^{n+2}) = 0, \quad (18)$$

which yields, after inserting (16) and (17) a non-linear equation in  $u_i^{n+1}$  and ultimately in  $p'$ .

To the best of the authors' knowledge, this strategy has not been followed before. Researchers always searched for constraining equations, built on the analytical differential equations. This yields an algorithm that does not satisfy all the requirements we impose. A reason for this might be historical: the first algorithms were developed to obtain a steady state solution, where no benefit is found in a discrete construction of the constraint. When time accuracy becomes important, as it is nowadays, algorithms with better properties are required.

#### 5. Iterative scheme for reacting flows

As shown in the previous section, a non-linear equation for pressure is obtained. In this section, we show how to solve the equation in an efficient manner for the case of non-premixed combustion. For clarity, the equations are written in a 1D configuration, where the subscript  $i$  refers to a grid point.

In this case, one scalar, namely the mixture fraction, is present. The constraining equation can now be written, using a chemical operator  $\mathcal{H}_C$ , defined as  $\rho = \mathcal{H}_C(\rho\xi)$  according to (8):

$$\mathcal{H}(\rho_i^{n+2}, (\rho\xi)_i^{n+2}) = 0 \quad \text{or} \quad \rho_i^{n+2} = \mathcal{H}_C((\rho\xi)_i^{n+2}). \quad (19)$$

For ease of notation, we introduce a new variable, fuel elements mass  $f$ , defined as  $f = \rho\xi$ . Obeying constraint (19), requires the discrete evaluation of mass and fuel elements mass conservation:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x} \left( \rho_R^n u_{i+\frac{1}{2}}^n - \rho_L^n u_{i-\frac{1}{2}}^n \right) \quad (20)$$

$$f_i^{n+1} = f_i^n - \frac{\Delta t}{\Delta x} \left( f_R^n u_{i+\frac{1}{2}}^n - f_L^n u_{i-\frac{1}{2}}^n \right) - \frac{\Delta t}{\Delta x} \left( J_{i+\frac{1}{2}}^n - J_{i-\frac{1}{2}}^n \right), \quad (21)$$

with  $J_{i+\frac{1}{2}}^n = -(\rho D)_{i+\frac{1}{2}}^n \frac{\xi_{i+1}^n - \xi_i^n}{\Delta x}$ .  $L$  and  $R$  indicate the extrapolated values at the left and right face of the control volume. For a first order upwind scheme, with positive values for velocity, this means  $\phi_L = \phi_{i-1}$  and  $\phi_R = \phi_i$ .

A predicted value for the velocity is obtained, using the momentum equation,

$$(\rho u)_i^* = (\rho u)_i^n - \frac{\Delta t}{\Delta x} \left( (\rho u)_R^n u_{i+\frac{1}{2}}^n - (\rho u)_L^n u_{i-\frac{1}{2}}^n \right) - \frac{\Delta t}{\Delta x} \left( p_{i+\frac{1}{2}}^n - p_{i-\frac{1}{2}}^n \right) + \Delta t \left( \frac{\delta \tau^n}{\delta x} \right)_i, \tag{22}$$

with  $u_i^* = \frac{(\rho u)_i^*}{\rho_i^{n+1}}$ . The values for velocity,  $u_i^{n+1} = u_i^* + u_i'$  and pressure  $p_i^{n+1} = p_i^n + p_i'$  are related through  $u_{i+\frac{1}{2}}' = -\Delta t \frac{1}{\rho_{i+\frac{1}{2}}^{n+1}} \frac{p_{i+1}' - p_i'}{\Delta x}$ .

The constraint is now formulated by requiring  $\rho^{n+2} = \mathcal{H}_C(f^{n+2})$ , or

$$\rho_i^{n+1} - \Delta t \frac{\rho_R^{n+1} u_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1} u_{i-\frac{1}{2}}^{n+1}}{\Delta x} = \mathcal{H}_C \left( f_i^{n+1} - \Delta t \frac{f_R^{n+1} u_{i+\frac{1}{2}}^{n+1} - f_L^{n+1} u_{i-\frac{1}{2}}^{n+1}}{\Delta x} - \Delta t \frac{J_{i+\frac{1}{2}}^{n+1} - J_{i-\frac{1}{2}}^{n+1}}{\Delta x} \right). \tag{23}$$

Inserting  $u^{n+1} = u^* + u'$ , yields

$$\rho^* + \rho' = \mathcal{H}_C(f^* + f'), \tag{24}$$

with  $\rho^* = \rho_i^{n+1} - \Delta t \frac{\rho_R^{n+1} u_{i+\frac{1}{2}}^* - \rho_L^{n+1} u_{i-\frac{1}{2}}^*}{\Delta x}$ ,  $f^* = f_i^{n+1} - \Delta t \frac{f_R^{n+1} u_{i+\frac{1}{2}}^* - f_L^{n+1} u_{i-\frac{1}{2}}^*}{\Delta x} + \Delta t \frac{J_{i+\frac{1}{2}}^n - J_{i-\frac{1}{2}}^n}{\Delta x}$ ,  $\rho' = -\Delta t \frac{\rho_R^{n+1} u_{i+\frac{1}{2}}' - \rho_L^{n+1} u_{i-\frac{1}{2}}'}{\Delta x}$ ,  $f' = -\Delta t \frac{f_R^{n+1} u_{i+\frac{1}{2}}' - f_L^{n+1} u_{i-\frac{1}{2}}'}{\Delta x}$ . (24) can be linearized around  $f^*$ :

$$\rho^* + \rho' = \mathcal{H}_C(f^*) + \frac{d\mathcal{H}_C}{df}(f^*)f'. \tag{25}$$

Note that the linearization is a Newton procedure to solve the non-linear equation. During the iterative procedure, ultimately the non-linear equation is solved. So, the linearization has no effect at all on the overall accuracy. In system notation, (25) reads:

$$\left( A - \frac{d\mathcal{H}_C}{df}(f^*)B \right) \underline{p}' = RHS, \tag{26}$$

with  $RHS = \mathcal{H}_C(f^*) - \rho^*$ . The dimension of all vectors is the number of grid nodes. Matrices  $A$  and  $B$  change during iteration, since the matrices are composed of extrapolated values of density and fuel elements mass, which depend on the sign of the unknown velocity  $u^{n+1}$ . The same holds for the RHS, which value also depends on the sign of  $u^{n+1}$ . This influence is only secondary, so that a minor assumption can be introduced, still preserving the consistency of the algorithm: calculating the extrapolated values of  $\rho$  and  $f$ , based on the sign of  $u^*$ , instead of  $u^{n+1}$ , matrices  $A$  and  $B$  and vector RHS only need one calculation per time step, saving computing time. The monotonicity of the spatial discretization is then in principle no longer guaranteed and values for mixture fraction outside  $[0, 1]$  could be reached, but we have not encountered problems in this sense so far.

### 6. Results and discussion

The above described pressure-correction scheme is now applied to a 1D contact discontinuity. A 1D channel is filled with fuel on one side, and oxidizer on the other side. The densities of fuel and oxidizer are equal here, in order to illustrate the effects as clearly as possible. The simulations are done in a time-accurate manner, with constant CFL-number,  $CFL = 0.9$ , based on the maximum velocity in the domain.

The test case is purely convective: viscosity and species diffusivity are set to zero ( $\rho D = 0$ ). A velocity of 1 m/s is enforced at the inlet. So the exact velocity is 1 m/s in the entire domain. The initial conditions and the result after 10 time steps on a grid with spacing  $\Delta x = 1$  m are shown in Fig. 2. Ideally, the velocity field should remain constant in the entire domain. However, due to numerical diffusion, by the upwind discretization, a reaction zone is formed, resulting in a zone with lower density. Since mass conservation is imposed, the flow must accelerate towards the outlet, which is the case (Fig. 2 top right). Spatial discretization without dissipation would resolve this artefact, but yields non-monotonic results, which is not wanted in case of a bounded variable, such as density or mixture fraction. A stable solution is found, also at later times. Fig. 3 reveals that all the states in the domain obey the equation of state in an exact manner, as it should be.

Note that, even with a highly diffusive numerical discretization, a high accuracy is achieved in predicting a correct behaviour, according to the equation of state. This is due to the fact that the equation of state is a purely algebraic relationship, and thus independent of discretization in space and time. The presented algorithm can be applied with any discretization. As an illustration, we perform the 1D test case also with a higher order scheme. For reasons of monotonicity, higher order accuracy in space is achieved, using a TVD scheme. Two limiter functions are chosen: Roe's superbee flux limiter and the minmod limiter. For comparison reasons, the first order upwind scheme is also retained. Higher order accuracy in

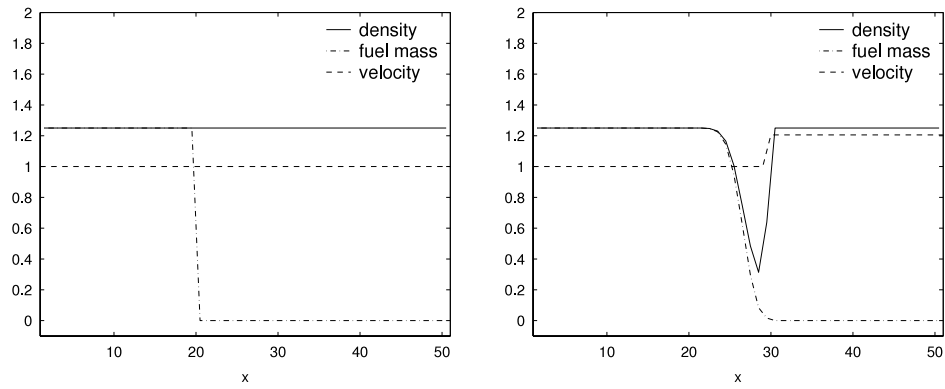


Fig. 2. Density, fuel elements' mass and velocity: initial field (left) and state after 10 time steps (right) for pure convection of fuel and oxidizer in a straight channel.

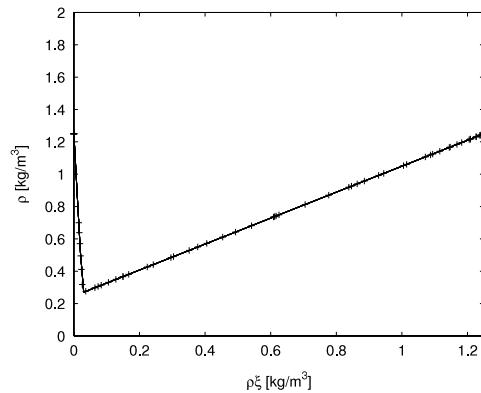


Fig. 3. Predicted states of density vs. fuel elements' mass during the first 10 time steps for pure convection of fuel and oxidizer in a straight channel: results exactly correspond to the equation of state (full line).

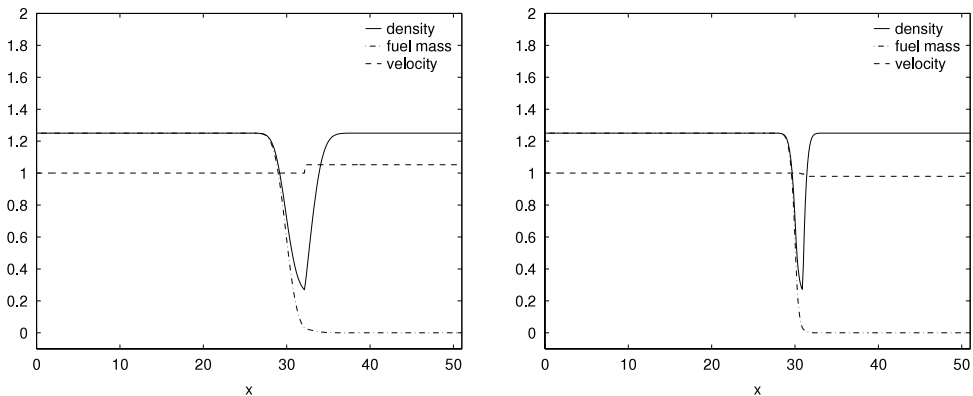
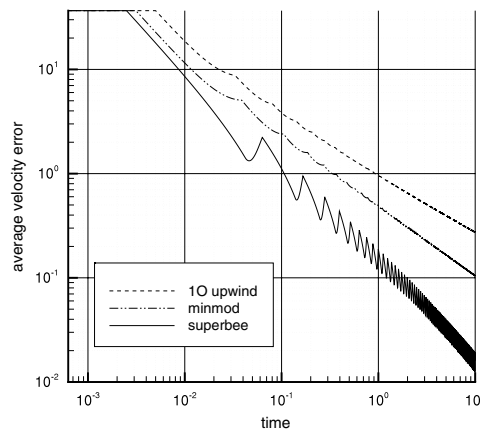


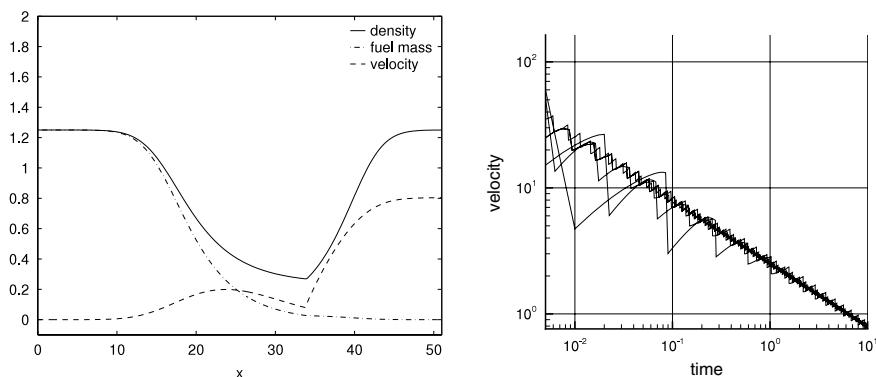
Fig. 4. Density, fuel elements' mass and velocity after 10 s: comparison of a first order upwind scheme (left) with a TVD scheme with minmod limiter (right) for pure convection of fuel and oxidizer in a straight channel.

time is adopted, using a low-storage Runge–Kutta scheme with 4 stages, using standard coefficients, with an evaluation of the pressure at each stage. The tests were performed using a constant time step of  $\Delta t = 0.005$  s on a grid with spacing  $\Delta x = 0.125$  m. Obviously, the representation of the fronts ameliorates (Fig. 4), with the same accuracy in predicting the equation of state correctly.

For a purely convective test case, there is no absolute length scale. The only non-dimensional parameter is the CFL-number, relating the time step ( $\Delta t$ ) and the grid spacing ( $\Delta x$ ):  $CFL = u_{inlet} \Delta t / \Delta x$ . For a reference grid with grid spacing  $\Delta x_{ref}$ , the result after  $t$  seconds is obtained using  $n_{ref}$  time steps:  $t = n_{ref} \Delta t_{ref}$ . The result at the same simulated time on a different grid, with the same, constant CFL, can be found using  $t = n_1 \Delta t_1$  with  $n_1 / n_{ref} = \Delta t_{ref} / \Delta t_1 = \Delta x_{ref} / \Delta x_1$ . As such, there exists a one-to-one relation between simulated time on the reference grid and grid refinement. A longer simulated



**Fig. 5.** Time-averaged exit velocity error for the first order upwind scheme and second order TVD scheme with minmod and superbee limiter function. Demonstration of grid convergence.



**Fig. 6.** Left: density, fuel elements' mass and velocity after 10 s with a TVD scheme with superbee limiter for pure diffusion of fuel and oxidizer in a straight channel. Right: exit velocity as a function of time for subsequently finer grids.

time corresponds to a finer grid. Fig. 5 shows that the method converges. The depicted error is based on the time-averaged exit velocity since instantaneous errors on the velocity field, resulting from the numerical discretization, oscillate in time.

To examine the influence of the diffusive term in (6), we use the same 1D test case. The initial profiles for density and mixture fraction profiles are taken from Fig. 2 (left). The velocity is set to zero at the inlet, but now diffusion is allowed with  $\rho D = 1$  Pa.s. All other settings are the same as in the purely convective problem described above. The results after 10 s with the higher order TVD scheme with superbee limiter are depicted in Fig. 6. The initially sharp front in mixture fraction is relaxed because of diffusion. This diffusion results in mixing between fuel and oxidizer, so reaction can take place. Hence, the lower values of density at the reaction zone. The behaviour of the velocity field can be explained as follows: first, due to the reaction, the flow accelerates towards the outlet. Second, due to diffusion between the stoichiometric mixture and fuel (or oxidizer), mass must be transported by convection from the high density side (fuel/oxidizer) to the low density side (stoichiometric). Again the equation of state is exactly fulfilled. Also, grid refinement results in more accurate predictions, as can be observed in Fig. 6 (right).

## 7. Conclusion

In this paper, we applied a novel pressure-correction algorithm, capable of dealing with an arbitrary amount of scalar conservation equations and a generic equation of state in transient simulations of variable density flows. The pressure-correction algorithm is characterized by a non-linear equation for the pressure, necessary to obtain conservation of mass and scalars, together with exact fulfillment of the equation of state. The algorithm was shown to remain stable, even for interfaces with high density ratios. An efficient implementation in the case of non-premixed combustion, with mixture fraction as a conserved variable, was described. We showed that the predicted states exactly match the equation of state.

## References

- [1] F. Nicoud, Conservative high order finite difference schemes for low-Mach number flows, *J. Comput. Phys.* 158 (2000) 71–97.
- [2] B. Van't Hof, J.H.M. Ten Thije Boonkkamp, R.M.M. Mattheij, Pressure-correction for laminar combustion simulations, *Combust. Sci. Technol.* 149 (1999) 201–223.

- [3] P. Rauwoens, K. Nerinckx, J. Vierendeels, B. Merci, Numerical issues for unsteady turbulent non-premixed combustion simulations, in: ECCOMAS Thematic Conference on Computational Combustion, Lisbon, Portugal, 2005.
- [4] P. Rauwoens, J. Vierendeels, B. Merci, A stable pressure-correction scheme for variable density flows involving non-premixed combustion, *Int. J. Numer. Methods Fluids* 56 (8) (2008) 1465–1471.
- [5] J.B. Bell, D.L. Marcus, A second order projection method for variable density flows, *J. Comput. Phys.* 101 (1992) 334–348.
- [6] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howel, M.L. Welcome, A conservative adaptive projection method for variable density incompressible Navier–Stokes equations, *J. Comput. Phys.* 142 (1998) 1–46.
- [7] M.S. Day, J.B. Bell, Numerical simulation of laminar reacting flows with complex chemistry, *Combust. Theory Model.* 4 (2000) 535–556.
- [8] T. Poinso, D. Veynante, *Theoretical and Numerical Combustion*, Edwards, Philadelphia, PA, 2001.
- [9] S.P. Burke, T.E.W. Schumann, Diffusion flames, *Ind. Eng. Chem.* 20 (1928) 998–1005.
- [10] A.J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comp.* 22 (1968) 745–762.
- [11] A.J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, *Math. Comp.* 23 (1969) 342–353.
- [12] F.H. Harlow, J.E. Welsch, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface, *Phys. Fluids* 8 (1965) 2182–2189.