Available online at www.sciencedirect.com
ScienceDirect

Journal of Computational Physics 221 (2007) 837–853

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

A Level Set Method for vaporizing two-phase flows

Sébastien Tanguy [a], Thibaut Ménard [b], Alain Berlemont [b,*]

[a] *LEMTA CNRS UMR – 7563, 2, avenue de la forêt de Haye, BP 160, 54504 Vandoeuvre lès Nancy, France*
[b] *UMR6614-CORIA, Technopôle du Madrillet, BP 12, Avenue de l'Université, 76801 Saint-Etienne-du-Rouvray Cedex, France*

Received 21 December 2005; received in revised form 22 June 2006; accepted 3 July 2006
Available online 15 September 2006

Abstract

Development and applications of numerical methods devoted to reactive interface simulations are presented. Emphasis is put on vaporization, where numerical difficulties arise in imposing accurate jump conditions for heat and mass transfers. We use both the Level Set Method and the Ghost Fluid Method to capture the interface motion accurately and to handle suitable jump conditions. A local vaporization mass flow rate per unit of surface area is defined and Stefan flow is involved in the process. Specific care has been devoted to the extension of discontinuous variables across the interface to populate ghost cells, in order to avoid parasitic currents and numerical diffusion across the interface. A projection method is set up to impose both the velocity field continuity and a divergence-free condition for the extended velocity field across the interface. The d^2 law is verified in the numerical simulations of the vaporization of an isolated static drop. Results are then presented for a water droplet moving in air. Vapor mass fraction and temperature fields inside and outside the droplet are presented.

Keywords: Level Set; Ghost Fluid Method; Two-phase flows; Reactive interface; Vaporization; Phase change; Jump conditions

1. Introduction

Computing interface motion in a multiphase incompressible flow is a wide topic of research and several approaches can be used. The Volume of Fluid method [12], the Level Set Method [15,24] and front tracking methods [26] are the most common numerical strategies used to predict interface motion. Whatever the chosen method, a specific approach is needed to impose appropriate jump conditions across the interface for pressure, density and viscosity. In [21] three different formulations are proposed for the Navier–Stokes equations for two-phase flows with jump conditions, namely the "Whole-Domain Formulation", the "Whole-Domain Conservation Law Form" and the "Jump Condition Form"; the first one is a non-conservative Navier–Stokes formulation and the second is a conservative Navier–Stokes formulation. But in both formulations, jump conditions are smoothed around the interface, smearing out discontinuous terms. The third formulation is much more satisfactory as it allows a sharp interface representation; specific numerical methods have been

[*] Corresponding author. Tel.: +33 2 32 95 36 17; fax: +33 2 32 91 04 85.
E-mail address: Alain.Berlemont@coria.fr (A. Berlemont).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.07.003

developed to take into account jump conditions, with a level set method and a front tracking method respectively [11,18]. These approaches not only avoid the introduction of a fictitious interface thickness, but are also suitable for providing a more accurate discretization of discontinuous terms, reducing parasitic currents and improving the resolution of the pressure jump condition.

In recent years, new investigations have been developed for computing reactive interfaces such as boiling flows with low density ratios [10,23,28], premixed flames [14,19,20] or Stefan problems [7,8]. The main difficulty in computing reactive interfaces is the accurate discretization of new jump conditions. For example, between two reacting phases of different densities, a velocity jump condition must be imposed to satisfy mass conservation. Moreover, three different velocities can be defined on the interface, namely the reacted phase velocity, the unreacted phase velocity and the reaction speed.

The Ghost Fluid Method [6] is a very powerful numerical technique to account accurately for various jump conditions; it has been successfully applied for inert multiphase incompressible flow [11,13]. Extending that approach, Nguyen et al. [14] developed a ghost fluid method for propagating a flame front in an incompressible fluid. They proposed an extension of the velocities on each side of the interface with no smearing out of any quantities across the interface. In that framework, powerful numerical tools [7,13] have been developed to impose both jump conditions and boundary conditions on the interface. Recently, impressive results have been obtained by Wang et al. [27] for the numerical simulation of heterogeneous propellant combustion with a Ghost Fluid Method. However, none of this work deals with liquid vaporization. We observe that [7,11,13,14] studies are quite suitable in some cases for handling vaporization jump conditions, but improvements are still required; we thus propose to extend these numerical methods in order to study this challenging task. We first present a simplified model for a 2D-axisymmetric static drop, where the surrounding Stefan flow is assumed to be irrotational. This first step enables us to validate the numerical methods which are defined for heat and mass transfers across the interface: the d^2 law is verified in the numerical simulations of the vaporization of an isolated static drop. Starting from this first approach, an extended model is then developed for moving droplets: the Navier–Stokes equations are solved, using Nguyen method [14], in order to describe the drop motion and the vapor flow around the drop. We observe that the original method [14] is accurate for a low density ratio, as is the case in flame front propagations, but it can be strongly affected by parasitic currents for a high density ratio. We overcome this difficulty by improving the extension of the velocity field in the liquid phase, by setting a divergence-free velocity extension.

2. Formalism

We consider incompressible flows and we assume that the fluid's physical properties are constant in space and time. The liquid phase is mono-component and the gas phase is different from the liquid vapor. Two phases are considered, the liquid phase (subscript l) and the gas phase (subscript g). The jump operator [·] across the interface Γ is defined as follows:

$$[A]_\Gamma = A_\mathrm{l} - A_\mathrm{g}$$

2.1. Governing equations and interfacial conditions for heat and mass transfers

Liquid vaporization depends strongly on the temperature and the concentration of the different species. We thus have to solve the following conservation equations to predict the temperature and the species mass fraction fields:

$$\frac{\partial T}{\partial t} + (\vec{V} \cdot \nabla) T = \frac{\nabla . (\lambda \nabla T)}{\rho C_p} \tag{1}$$

$$\frac{\partial Y}{\partial t} + (\vec{V} \cdot \nabla) Y = \frac{\nabla \cdot (\rho D_\mathrm{m} \nabla Y)}{\rho} \tag{2}$$

where T is the temperature, Y is the mass fraction, $\vec{V}$ is the velocity vector, λ is the thermal conductivity, ρ is the density, C_p is the specific heat at constant pressure and D_m is the mass diffusion coefficient.

When reactive interfaces are considered, jump conditions must be added to respect energy conservation [4,7,27] and mass conservation [2,4,27] across the interface:

$$h_{\mathrm{lg}}\dot{\omega} - [\lambda \nabla T \cdot \vec{N}]_{\Gamma} = 0 \tag{3}$$

$$\dot{\omega} Y_{\mathrm{l}}^{\Gamma} + [\rho D_{\mathrm{m}} \nabla Y \cdot \vec{N}]_{\Gamma} = \dot{\omega} Y_{\mathrm{g}}^{\Gamma} \tag{4}$$

where $\dot{\omega}$ is the vaporization rate (local vaporization mass flow rate per unit surface), h_{lg} is the latent heat of vaporization and $\vec{N}$ is the vector normal to the interface. Noting that the gradients of the species mass fraction are zero in a mono-component liquid, Eq. (4) becomes:

$$\dot{\omega} = \frac{\rho_{\mathrm{g}} D_{\mathrm{m}} \nabla Y \cdot \vec{N}|_{\mathrm{g}}^{\Gamma}}{1 - Y_{\mathrm{vap}}^{\Gamma}} \tag{5}$$

On the interface, two values can be defined for the species mass fraction: (1) on the liquid side and (2) on the gas side Y_{g}^{Γ} the vapor concentration, depending on the saturated vapor pressure. Using the Clausius–Clapeyron relation:

$$p_{\mathrm{vap}}^{\Gamma} = p_{\mathrm{atm}} \exp\left(-\frac{h_{\mathrm{lg}} m_{\mathrm{vap}}}{R}\left(\frac{1}{T^{\Gamma}} - \frac{1}{T^{\mathrm{B}}}\right)\right) \tag{6}$$

$$Y_{\mathrm{vap}}^{\Gamma} = \frac{p_{\mathrm{vap}}^{\Gamma} m_{\mathrm{vap}}}{(p_{\mathrm{atm}} - p_{\mathrm{vap}}^{\Gamma}) m_{\mathrm{g}} + p_{\mathrm{vap}}^{\Gamma} m_{\mathrm{vap}}} \tag{7}$$

where p_{atm} is the ambient pressure in the gaseous domain, $p_{\mathrm{vap}}^{\Gamma}$ is the saturated vapor pressure on the interface, m_{vap} is the molar mass of the vapor, R is the perfect gas constant, T^{B} is the liquid boiling temperature for the gaseous ambient pressure condition, T^{Γ} is the interface temperature and m_{g} is the molar mass of the ambient gas. We observe that solving the species mass fraction equation is only required in the gas phase, using a Dirichlet boundary condition on the interface, given by Eq. (7).

2.2. Interface reaction speed and fluid velocity jump

The mass conservation condition across the interface reads:

$$\dot{\omega} = \rho_{\mathrm{l}}(\vec{V}_{\mathrm{S}} - \vec{V}_{\mathrm{l}}) \cdot \vec{N} = \rho_{\mathrm{g}}(\vec{V}_{\mathrm{S}} - \vec{V}_{\mathrm{g}}) \cdot \vec{N} \tag{8}$$

where the interface velocity $\vec{V}_{\mathrm{S}}$ is defined as the sum of the liquid phase velocity and the surface regression speed due to vaporization $\vec{V}_{\mathrm{int}}$:

$$\vec{V}_{\mathrm{s}} = \vec{V}_{\mathrm{l}} + \vec{V}_{\mathrm{int}} \tag{9}$$

We can then write:

$$\vec{V}_{\mathrm{int}} = \frac{\dot{\omega}}{\rho_r} \vec{N} \tag{10}$$

Moreover, the following velocity jump condition can be derived from Eqs. (8) and (9) to respect mass conservation across the interface [4,14]:

$$[\vec{V}]_{\Gamma} = \dot{\omega} \left[\frac{1}{\rho}\right]_{\Gamma} \vec{N} \tag{11}$$

2.3. Computing the vapor flow around a static drop

A first model is developed for the vaporization of a static drop. The velocity jump condition equation (11) induces a vapor flow around the drop, known as the Stefan Flow. Assuming that the vapor flow around a static drop is irrotational, the velocity can be derived from a velocity potential:

$$\vec{V}_\mathrm{g} = \nabla\varphi \tag{12}$$

Assuming that the flow is incompressible, we can solve the following equation to compute the velocity potential:

$$\Delta\varphi = 0 \tag{13}$$

with the following jump condition to respect mass conservation across the interface:

$$[\nabla\varphi \cdot \vec{N}]_\Gamma = \dot{\omega}\left[\frac{1}{\rho}\right]_\Gamma \tag{14}$$

2.4. Computing the fluid flow inside and outside a moving drop

Fluid velocity fields can be computed using Navier–Stokes equations, with a divergence-free condition in the gas and the liquid domains:

$$\frac{\partial\vec{V}}{\partial t} + (\vec{V}\cdot\nabla)\vec{V} = -\frac{\nabla p}{\rho} + \frac{\nabla.(2\mu D)}{\rho} \tag{15}$$

$$\nabla\cdot\vec{V} = 0 \tag{16}$$

where p is the fluid pressure, μ the dynamic viscosity and D is the viscous deformation tensor. We apply appropriate jump conditions, as is usual for an inert interface:

$$[p]_\Gamma = \sigma\kappa(\phi) + 2[\mu]_\Gamma(\nabla u\cdot\vec{N}, \nabla v\cdot\vec{N})\cdot\vec{N} - \dot{\omega}^2\left[\frac{1}{\rho}\right]_\Gamma \tag{17}$$

$$[\rho]_\Gamma = \rho_\mathrm{l} - \rho_\mathrm{g} \tag{18}$$

$$[\mu]_\Gamma = \mu_\mathrm{l} - \mu_\mathrm{g} \tag{19}$$

where σ is the liquid surface tension and κ the interface curvature.

But, due to the interface reaction, one more jump condition for the fluid velocity is required, and given by Eq. (11).

2.5. Level Set Methods

The interface is numerically defined as the zero level curve of a level set function ϕ. This function evolves with the interface reaction speed and is transported by the liquid velocity at the interface using the following relation:

$$\frac{\partial\phi}{\partial t} + \left(\vec{V}_\mathrm{l} + \frac{\dot{\omega}}{\rho_\mathrm{l}}\vec{N}\right)\cdot\nabla\phi = 0 \tag{20}$$

One feature of Level Set Methods is to transport, over the whole domain, a function which is physically meaningful only on the interface. The continuity of the interface velocity must be satisfied in both phases in the interface neighborhood, and it is important that the function ϕ stays continuous and well-resolved to keep the geometrical properties of Level Set Methods. This requirement can be achieved by solving a redistancing equation which forces ϕ to be the signed distance to the interface for every time step, without changing the zero level curve location.

$$\frac{\partial d}{\partial\tau} = \mathrm{sign}(\phi)(1 - |\nabla d|) \tag{21}$$

Eq. (21) is iterated on a few steps for a fictitious time τ, and it converges to a signed distance function in the whole domain [24].

Thus, geometrical interface properties, such the normal vector and the curvature κ, can be easily and accurately computed with a standard second order central finite difference scheme:

$$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|} \quad \kappa(\phi) = -\nabla\cdot\vec{N} \tag{22}$$

It is well known that numerical computation of Eqs. (20) and (21) can generate mass loss in under-resolved regions. This is the main drawback of level set methods, but grid resolution in the test cases presented is fine enough to ensure that this loss is negligible (<1%).

3. Numerical methods

We now describe the numerical methods used to solve the above equations. The spatial derivatives are evaluated with a fifth order Weighted Essentially Non Oscillatory (WENO 5) scheme [9] for convective terms and a second order central scheme for diffusive terms. Temporal derivatives are first order explicit for convective terms and implicit for diffusive terms. The symmetric linear systems are solved with the Conjugate Gradient Preconditioned method.

3.1. Solving mass fraction equation

As has been previously mentioned (Section 2.1), to describe species mass fraction around the vaporizing liquid interface, we solve Eq. (2) with a prescribed Dirichlet boundary condition on the interface equation (7). A symmetric linear system is thus obtained from:

$$Y^{n+1} - \Delta t\nabla\cdot(D_{\mathrm{m}}\nabla Y^{n+1}) = Y^n - \Delta t(\vec{V}\cdot\nabla)Y^n \tag{23}$$

For nodes close to the interface, the Dirichlet condition is applied using the second order scheme developed by [7] for Stefan problems. Vapor mass fraction at the interface is obtained with Eqs. (6) and (7). To determine the temperature on the interface, the extended liquid temperature T_i^{ghost} in the gaseous domain (described in Section 3.3) is used to find the liquid temperature value closest to the location considered. The interface temperature is then obtained (in 1D and assuming for example that $\phi_{i+1} > 0$ and $\phi_i < 0$) by

$$T_{\mathrm{int}} = \frac{T_{i+1}|\phi_i| + T_i^{\mathrm{ghost}}|\phi_{i+1}|}{|\phi_i| + |\phi_{i+1}|} \tag{24}$$

That scheme can be generalised in two and three dimensions. When T_{int} is estimated by averaging of the liquid temperature and the gas temperature, instead of the liquid temperature and the extended liquid temperature in the gas, unrealistic interface temperature can be observed.

3.2. Solving heat equation

Energy conservation equation is expressed using the temperature variable. The temperature is continuous across the interface. But when the liquid is vaporizing, the thermal flux is discontinuous due to the latent heat from the phase change. We thus solve a symmetric linear system deduced from:

$$\rho C_p T^{n+1} - \Delta t\nabla\cdot(k\nabla T^{n+1}) = \rho C_p T^n - \rho C_p\Delta t(\vec{V}\cdot\nabla)T^n \tag{25}$$

and we must account for the jump condition equation (3) on the thermal flux. In [13], the authors set up a method for solving the Poisson equation with jump conditions on the solution and/or on the solution's normal derivative, and with variable coefficients (k is not continuous) across the interface. This method can be easily adapted to solve Eq. (1) with the jump condition equation (3).

3.3. High order extension for computing convective terms

Let us consider a variable θ which is discontinuous across the interface, and/or its normal derivative θ_n also discontinuous across the interface (temperature or mass fraction fields for example). When deriving such variables with a high order scheme in the neighborhood of the interface, we use points on each side of the interface. In order to avoid smearing out temperature and mass fraction fields when computing explicit convective terms, we must extrapolate each one of these fields on the other side of the interface in order to respect the continuity and normal derivative continuity of these variables. That can be achieved using high order extrapolation in the normal direction as proposed by [1]. The author extended numerical algorithms from [6,16] to an arbitrary order, imposing continuity for the variables and the normal derivative of the variables. Then the extrapolated values populate ghost values in the ghost domain, and are used to compute the convective derivative. We use here the following two-step iterative algorithm from [1]:

$$\frac{\partial\theta_n}{\partial t} = \pm(\vec{N}\cdot\nabla)\theta_n \tag{26}$$

$$\frac{\partial\theta}{\partial t} = \pm((\vec{N}\cdot\nabla)\theta - \theta_n) \tag{27}$$

where θ_n stands for:

$$\theta_n = (\vec{N}\cdot\nabla)\theta \tag{28}$$

We can note that this algorithm has been used in [8] to improve the numerical accuracy of the method, developed by the same authors in [7] for computing Stefan problems.

In the present study, we extrapolate the liquid temperature field (in the gas), the gas temperature field (in the liquid) and the vapor mass fraction field (in the liquid). By improving numerical accuracy of convective terms in the interface neighborhood, this method enables us to carry out simulations involving high temperature gradients between the drop and the gas, and avoid artificial heating of the liquid phase due to numerical diffusion. Moreover, such extrapolation for mass fraction fields enables us to compute the vaporization rate on each side of interface continuously using Eq. (5). It is well designed both for the transport of the interface with a continuous reaction speed in the interface neighborhood, and for computing explicit convective derivatives without smearing out discontinuous variables. This last consideration is particularly important as an artificial heating of the liquid phase can lead to results, such as liquid temperature greater than boiling temperature, that are not physically meaningful.

For the discretization of Eqs. (26)–(28), we use a first order upwind scheme for spatial derivatives and a first order explicit scheme for temporal derivatives.

3.4. Computing Stefan flow for a static drop

To compute the Stefan flow around the static liquid phase we solve Eq. (13) with appropriate jump conditions equation (14). We use standard a second order central difference scheme for derivative discretization and we apply the numerical methods set up by [13] to take into account jump conditions on normal derivatives, as previously presented for the energy equation. Note that high order extrapolation is used to define ghost values of the velocity potential in the liquid phase, as described in the previous section. Knowing the velocity field extension in the liquid phase, we are thus able to compute convective terms in the gas phase.

3.5. Solving Navier–Stokes equations with appropriate jump conditions

A first order projection method is used for temporal integration of Navier–Stokes equations [17]. Spatial discretization is carried out on a standard MAC grid, where the pressure and the level set function are defined on the cell centers (i,j) whereas velocity components are expressed on a staggered grid, namely $u_{i+1/2,j}$, $v_{i,j+1/2}$.

Starting from [11] for the discretization method for pressure and density discontinuities, Nguyen [14] proposed an extension to take into account a reactive interface between two incompressible immiscible fluids with different densities. In the first step we ran preliminary tests and observed that although the discretization proposed by [14] was well adapted to studying flame front propagation with a low density ratio, it led to unrealistic mass predictions for liquid–gas interface computations with a high density ratio. In particular, it appears that the liquid phase velocity field extension is not fully satisfactory. Indeed, as pointed out by [25], the extension must the respect divergence-free condition in the whole domain, even in the interface neighborhood. Otherwise, the interface is transported by a liquid velocity field which does not respect mass conservation, leading to erroneous predictions of the interface motion. We now therefore present an improvement of the liquid velocity field extension to preserve the divergence-free condition on the interface.

Let us consider a velocity field which is discontinuous across a given interface, with a jump condition given by Eq. (11). Using the Ghost Fluid Method [6], we have to populate ghost cells on each side of the interface, corresponding to an extension of the gaseous/liquid velocity field in the liquid/gaseous domains respectively. This can be obtained with the following strategy [14]:

$$\vec{V}_{\mathrm{l}}^{\mathrm{ghost}} = \vec{V}_{\mathrm{g}} - \dot{\omega}\left[\frac{1}{\rho}\right]_{\Gamma}\vec{N} \tag{29}$$

$$\vec{V}_{\mathrm{g}}^{\mathrm{ghost}} = \vec{V}_{\mathrm{l}} + \dot{\omega}\left[\frac{1}{\rho}\right]_{\Gamma}\vec{N} \tag{30}$$

As has been previously pointed out, such an extension imposes velocity field continuity across the interface but the divergence-free condition is not satisfied. We observed that this failure generates important parasitic currents in the velocity field, in particular in the liquid domain. On the other hand, the gas velocity field extension is not so critical, as the interface is only transported with the liquid velocity field. We thus decided to keep the gaseous velocity field extension, and we used the following projection method for solving Navier–Stokes equations and to extend the liquid velocity field:

1. Compute an intermediate velocity field $\vec{V}_{\mathrm{l}}^{*}$ and $\vec{V}_{\mathrm{g}}^{*}$:

$$\begin{aligned} \vec{V}_{\mathrm{l}}^{*} &= \vec{V}_{\mathrm{l}}^{*} - \Delta t\left((\vec{V}_{\mathrm{l}}^{n}\cdot\nabla)\vec{V}_{\mathrm{l}}^{n} - \frac{\mu_{\mathrm{l}}\Delta\vec{V}_{\mathrm{l}}^{n}}{\rho_{\mathrm{l}}}\right) \\ \vec{V}_{\mathrm{g}}^{*} &= \vec{V}_{\mathrm{g}}^{n} - \Delta t\left((\vec{V}_{\mathrm{g}}^{n}\cdot\nabla)\vec{V}_{\mathrm{g}}^{n} - \frac{\mu_{\mathrm{g}}\Delta\vec{V}_{\mathrm{g}}^{n}}{\rho_{\mathrm{g}}}\right) \end{aligned} \tag{31}$$

Note that when the interface crosses a mesh cell, the viscosity is computed following numerical procedures from [11,13].

2. Compute the right hand side for the Poisson equation:

$$\begin{aligned} &\text{If } \phi > 0: \quad f = \nabla\cdot\vec{V}_{\mathrm{l}}^{*} \\ &\text{Else if } \phi < 0: \quad f = \nabla\cdot\vec{V}_{\mathrm{g}}^{*} \end{aligned} \tag{32}$$

3. Compute the pressure field by solving the Poisson equation with a numerical strategy from [11] to account for pressure and density jumps:

$$\nabla\cdot\left(\frac{\nabla p^{n+1}}{\rho^{n+1}}\right) = \frac{f}{\Delta t} \tag{33}$$

4. Compute the real divergence-free velocity field following [11] and [14]:

$$\begin{aligned} &\text{If } \phi > 0: \quad \vec{V}_{\mathrm{l}}^{n+1} = \vec{V}_{\mathrm{l}}^{*} - \Delta t\frac{\nabla p^{n+1}}{\rho^{n+1}} \\ &\text{Else if } \phi < 0: \quad \vec{V}_{\mathrm{g}}^{n+1} = \vec{V}_{\mathrm{g}}^{*} - \Delta t\frac{\nabla p^{n+1}}{\rho^{n+1}} \end{aligned} \tag{34}$$

The above procedure is similar to [14] method. The next step is to apply the velocity field extension before a new time step. Our extension methodology for the liquid velocity field is now described.

5. Compute a ghost pressure p^{ghost} for the liquid velocity field in the gaseous domain:

$$\nabla\cdot\left(\frac{\nabla p^{\mathrm{ghost}}}{\rho^{n+1}}\right) = \frac{\nabla\cdot\vec{V}_{\mathrm{l}}^{*}}{\Delta t} \tag{35}$$

6. Compute a liquid velocity field extension $\vec{V}_{\mathrm{l}}^{\mathrm{ghost}}$ in the gaseous domain:

$$\vec{V}_{\mathrm{l}}^{\mathrm{ghost}} = \vec{V}_{\mathrm{l}}^{*} - \Delta t\frac{\nabla p^{\mathrm{ghost}}}{\rho^{n+1}} \tag{36}$$

7. Define a new velocity field $\vec{W}$:

$$\begin{aligned} &\text{If } \phi > 0: \quad \vec{W} = \vec{V}_1^{n+1} \\ &\text{Else if } \phi < 0: \quad \vec{W} = \vec{V}_1^{\text{ghost}} \end{aligned} \tag{37}$$

The resulting $\vec{W}$ vector is still not divergence free at the interface, and one more step is necessary to enforce this condition.

8. Following the Hodge decomposition, project on divergence-free space to construct a divergence-free liquid velocity field in the whole domain by solving the following Poisson equation for a potential function ψ:

$$\Delta\psi = \nabla \cdot \vec{W} \tag{38}$$

We then get the solution:

$$\vec{V}_1^{n+1} = \vec{W} - \nabla\psi \tag{39}$$

9. Determine an gas velocity field extension $\vec{V}_{\text{g}}^{\text{ghost}}$ for the next time step:

$$\vec{V}_{\text{g}}^{\text{ghost}} = \vec{V}_1 - \dot{\omega}\left[\frac{1}{\rho}\right]_{\Gamma}\vec{N} \tag{40}$$

We can observe that, instead of applying this quite complex methodology, we could only apply step eight on Eq. (30) for the liquid velocity field extension from [14]. However we observed that with such a simplification in the procedure, the efficiency of the projection method is not sufficient to impose the divergence-free condition.

3.6. Temporal and spatial numerical accuracy

We use a classical time-step restriction to respect the CFL condition for convective terms, viscous terms and surface tension terms as proposed in [11]. Heat and mass transfers do not impose further time-step restriction, as implicit schemes are used for heat diffusion and mass diffusion. Since the lowest order of the method used to impose pressure, density, viscosity and heat flux jumps is of the first order, the overall accuracy of the method may not be more than first order.

4. Numerical results

The following numerical values for physical variables are used in the results presented. They are close to water properties for the liquid phase and air properties for the gas phase (Table 1).

In the first test case (Section 4.1), we consider a static vaporizing drop. Thus the flow is computed only in the gas phase using Eq. (13) and the corresponding jump condition (14), and the interface motion is only driven by the regression speed of vaporization. Then Eq. (20) becomes:

$$\frac{\partial\phi}{\partial t} + \frac{\dot{\omega}}{\rho_r}\vec{N} \cdot \nabla\phi = 0 \tag{41}$$

Simulations in Section 4.1 are carried out in a 2D-axisymmetric configuration. The formulation used here for static drops enables us to run the computations with a much longer time-step than when the Navier–Stokes solver is used, since no drastic time-step restriction on surface tension is required. Thus, less computer time is needed for the convergence study carried out in order to check that our method correctly takes into account

Table 1
Physical properties for the liquid phase and the gas phase

	ρ (kg m^{-3})	μ (kg m^{-1} s^{-1})	λ (W m^{-1} K^{-1})	C_{p} (J kg^{-1} K^{-1})	M (kg mol^{-1})	h_{lg} (J kg^{-1})	D_{m} (s^{-1})	T_{B} (K)	σ (N m^{-1})
Gas	1.226	1.78×10^{-5}	0.046	1000	0.029	2.3×10^{6}			
Liquid	1000	1.137×10^{-3}	0.6	4180	0.018		2×10^{-5}	373	0.070

heat and mass transfer across the interface. In Sections 4.1 and 4.2, simulations are stopped when the drop diameter is equal to half its initial diameter. In this section gravity is neglected.

4.1. Vaporization of an axisymmetric static droplet (low temperature)

In this first numerical test case, the initial drop radius is $R_G = 150$ μm, the initial drop temperature is 353 K and two initial gas temperatures are considered, $T_g = 373$ K and 573 K. The length of the domain in the x-direction is $lx = 4R_G$ and in the y-direction $ly = 2lx$. Computations are carried out with two mesh sizes (64×128) and (128×256). We use three Dirichlet boundary condition for temperature, species mass fraction and potential function φ, namely $T_{\text{boundary}} = 373$ K or 573 K, $Y_{\text{boundary}} = 0$, $\varphi_{\text{boundary}} = 0$. On the symmetry axis Neumann symmetric boundary conditions are used. A more physical test case would be to study drop vaporization in an infinite medium but that would imply quite a large distance between the boundary and the droplet interface, in order to be free from the influence of boundary conditions. Here, as we are considering the method from a numerical point of view, we keep the computational domain quite small.

On Fig. 1(a) and (b), we present the temperature field and the mass fraction field around the interface at $t = 0.4$ s on the finest grid (128×256) and for $T_g = 373$ K; we observe that a spherical symmetry property around the drop is obtained in the simulations and no artificial heating of the droplet occurs. We then present in Fig. 2 the time evolution of the reduced square diameter (until $d/d_0 = 0.5$) for the finest grid resolution. We observe that a d^2 law is perfectly verified for both gas temperatures. The two different grid resolutions are compared in Fig. 3 for the two gas temperatures: the convergence of the method is satisfactory. We then present in Fig. 4 the temperature profile in the drop and in the gas for two different times ($t = 0.01$ s and $t = 0.15$ s). We observe that the drop reaches an equilibrium temperature as expected, and Fig. 4 also clearly shows that the numerical procedure for handling the temperature gradient jump on the interface is efficient.

4.2. 2D drop vaporization with a constant vaporization speed

This second test case illustrates the efficiency of the velocity extension methodology described in Section 3.5 for solving Navier–Stokes equations with a velocity jump condition. In this theoretical test case, we consider a static drop with a constant vaporization speed, without any coupling with heat and mass transfer. Thus only Eqs. (15), (16), (20) and (21) are solved with the jump conditions (11), (17), (18) and (19). The drop is initially static, and we use free boundary conditions everywhere. The exact solution of the problem is obvious: the drop

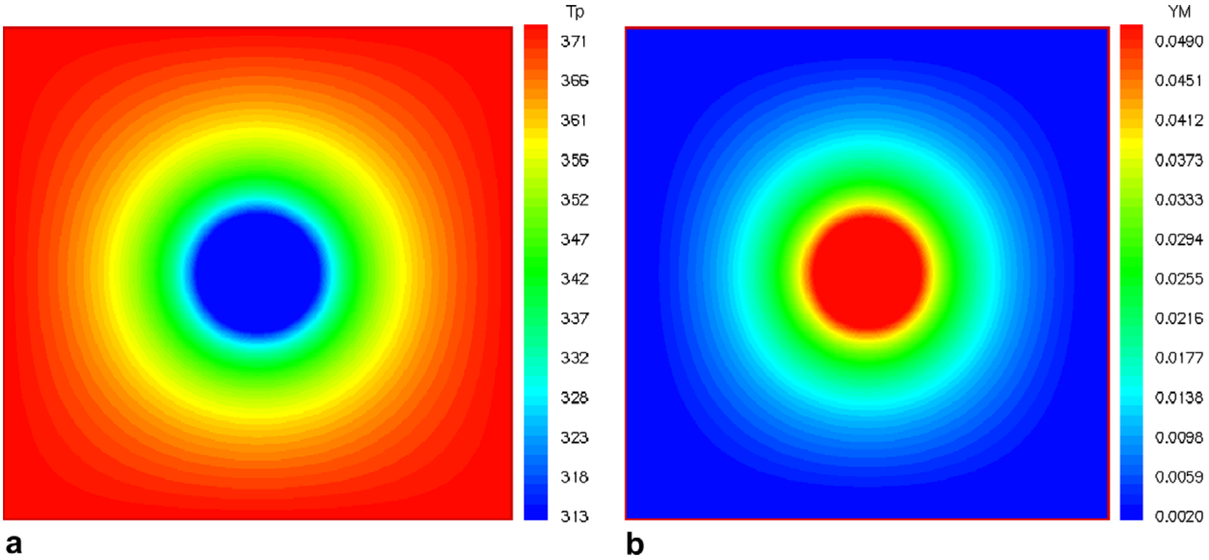

Fig. 1. (a) Temperature field and (b) vapor mass fraction field ($t = 0.4$ s).

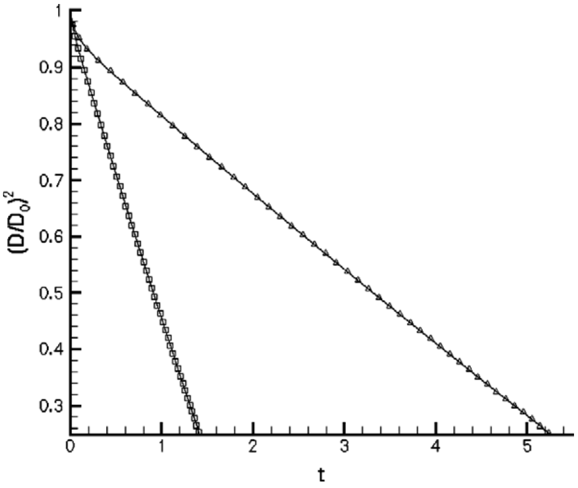

Fig. 2. Normalized square diameter versus time (s): (△) $T_g = 373$ K; (□) $T_g = 573$ K; grid size is 128 × 256.

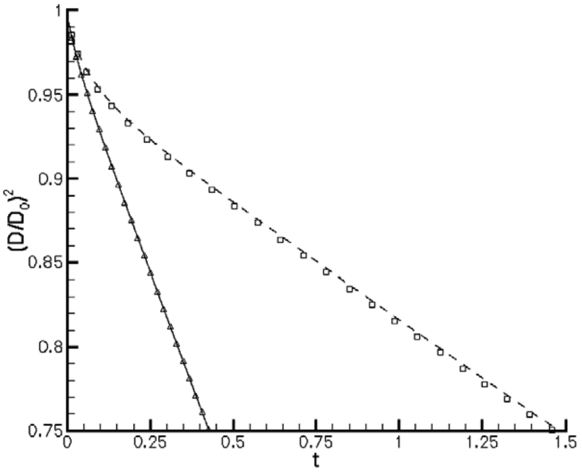

Fig. 3. Normalized square diameter versus time (s): $T_g = 573$ K, (△) grid size 64 × 128, (—) grid size 128 × 256; $T_g = 373$ K, (□) grid size 64 × 128, (- - -) grid size 128 × 256.

stays static and vaporizes linearly. Thus we can compute the temporal mass variation of the vaporizing drop and we can compare it with the exact solution to the problem. We performed numerical tests in a 2D configuration on a 128 × 128 grid; the initial drop radius is $R_G = 200$ μm, the length of the domain in the x-direction is $lx = 6R_G$ and in the y-direction $ly = lx$.

The same physical properties as above are kept. The vaporization speed $|\vec{V}_{int}|$ is equal to 0.001 m s^{-1}. The temporal mass variation is known exactly and reads:

$$M_{th}(t) = \rho_l \pi (R_G - V_{int} t)^2 \tag{42}$$

Whereas such a test case seems very simple, only accurate methods that avoid the development of parasitic currents, can succeed. In particular, good predictions of the global temporal mass evolution are only possible when the liquid extension velocity exhibits the divergence-free property. In order to point out that requirement, we draw in Fig. 5 the exact solution, the results with Nguyen et al. [14] method (without divergence-free), and our results. The time of vaporization is limited to 1 ms. The divergence-free property clearly appears to be of high influence on the mass conservation, as 10% of the mass is lost after 1 ms when that property in not involved in the simulations. We observe in Fig. 6 for a longer time of vaporization that results obtained

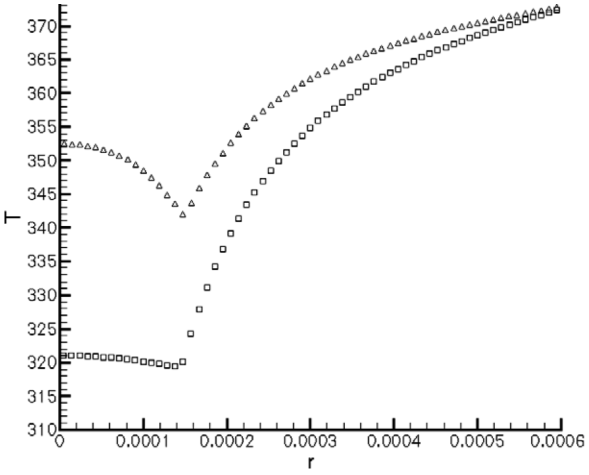

Fig. 4. Temperature profiles: (△) $t = 0.01$ s, (□) $t = 0.15$ s.

with our method for the temporal mass evolution are the same as the exact solution. Moreover, we observe that the drop shape remains circular. We draw, at the same time $t = 0.001$ s, the real velocity field in Fig. 7, the gas velocity field and its extension in the liquid domain (Fig. 8), and the liquid velocity field and its extension in the gas domain (Fig. 9). As previously mentioned, the good agreement between the simulations and the exact solution is strongly linked to the accurate extension of these velocities. We clearly observe that the parasitic currents in the liquid phase and its extension in the gas are 1000 times smaller than the real liquid velocity field.

In order to go further in the convergence study, the loss of mass between the exact solution and the numerical results is given in Table 2. The relative error E to the exact solution is given for three different grid resolutions, 32×32, 64×64 and 128×128, at three different times. We observed that the order of the method is close to 2. Note that the vaporization rate is fixed, and the energy equation and the mass fraction equation are not involved in these calculations.

4.3. Displacement of a vaporizing drop due to wall boundary conditions

This theoretical test case enables us to check that the Stefan flow computed in the gas phase is correctly coupled with the liquid flow through the procedure described in Section 3.5. Wall boundary conditions are imposed on the left boundary, the right boundary and the bottom boundary, whereas free boundary condi-

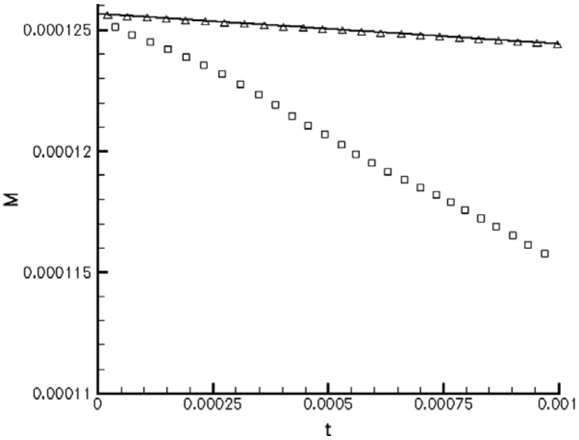

Fig. 5. Mass evolution of the drop: (△) level set simulation (128×128), (—) exact solution, (□) without divergence-free property.

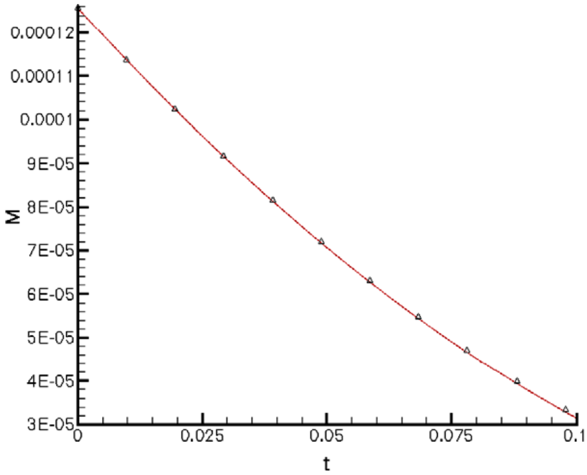

Fig. 6. Mass evolution of the drop: (△) level set simulation (128 × 128), (—) exact solution.

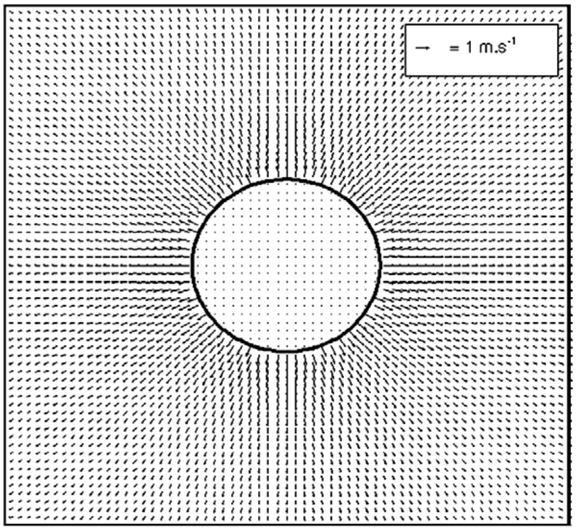

Fig. 7. Real velocity field ($t = 0.001$ s).

tions are assumed on the top of the domain. Numerical tests are performed in a 2D configuration with 64×128 grid resolution. The drop radius is initially equal to $R_G = 200$ μm, the length domain in the x-direction is $lx = 6R_G$ and in the y-direction $ly = 2lx$. The drop is initially centered on $(lx/2, ly/4)$.

The same physical properties for the fluid and speed vaporization are kept ($|\vec{V}_{\mathrm{int}}|$ is still equal to 0.001 m s^{-1}). Although we do not know the exact solution to the problem, intuitive arguments lead us to conclude that the drop should have an upward motion due to interaction between the wall boundary conditions and the ejected vapor flow that should push the drop up to the top boundary. The results of our computations are presented in Fig. 10 for three different times ($t = 0, 0.01, 0.02$). The results clearly show that the numerical method used to impose a normal velocity jump, presented in Section 3.5, is physically significant; the coupling between the Stefan flow and the liquid flow appears very satisfactory.

4.4. Vaporization of a moving droplet

In this section, we provide test cases in order to illustrate the ability of the method to describe the vaporization of a moving deformable drop. In particular, using low dissipative algorithms described in Section 3.3,

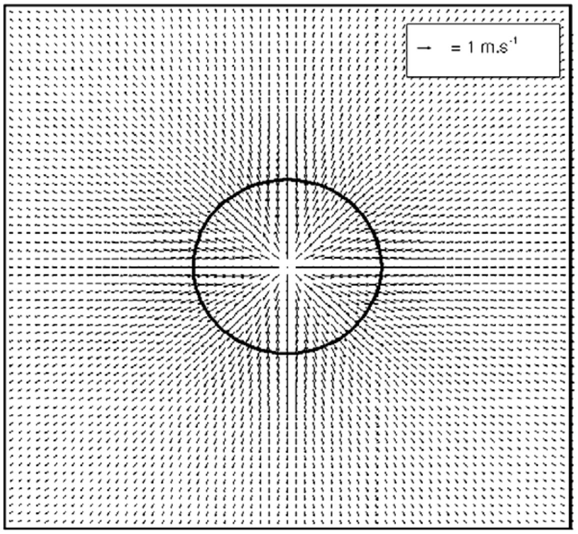

Fig. 8. Gas velocity field and its extension in the liquid domain ($t = 0.001$ s).

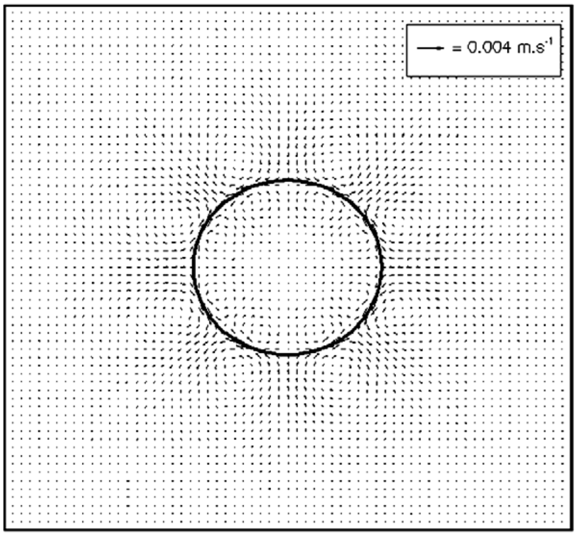

Fig. 9. Liquid velocity field and its extension in the gaseous domain ($t = 0.001$ s).

Table 2
Rate of convergence for three different times

Grid	E ($t = 0.001$) (%)	Order	E ($t = 0.002$) (%)	Order	E ($t = 0.003$) (%)	Order
32 × 32	0.73	–	0.70	–	0.63	–
64 × 64	0.20	1.9	0.13	2.3	0.11	2.4
128 × 128	0.056	1.9	0.03	2.1	0.03	1.9

simulations can be carried out with high drop velocity and strong interfacial thermal gradient. We consider a 2D axisymmetric water drop with radius is $R_G = 300$ μm, initial drop temperature equal to 323 K, initial drop velocity $V_G = 1$ m s^{-1} and initial gas temperature equal to 873 K. On the boundary of the domain we use the Dirichlet boundary condition for temperature, species mass fractions and pressure: $T_{\text{boundary}} = 873$ K, $Y_{\text{boundary}} = 0$, $P_{\text{boundary}} = 0$. Neumann symmetric boundary conditions are used on the symmetry axis. The

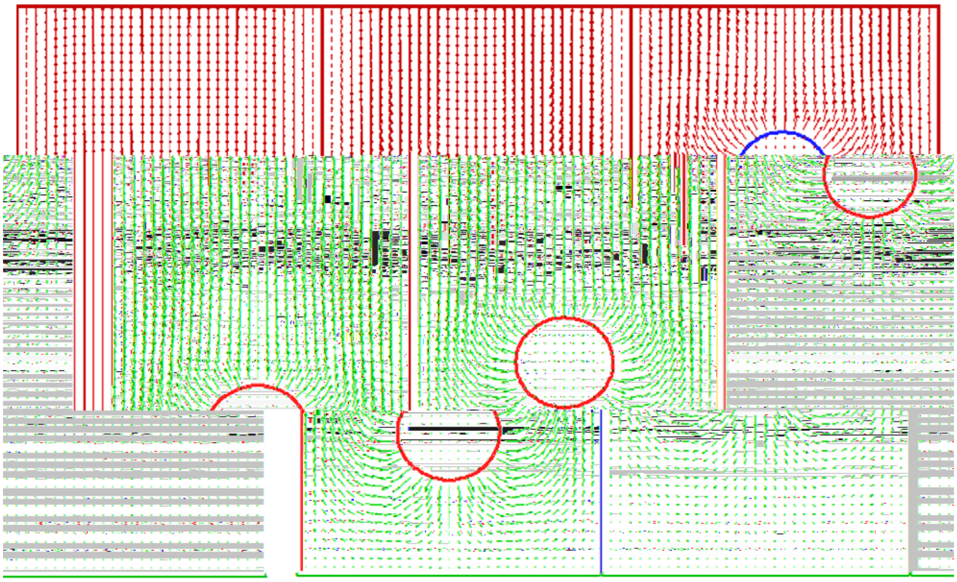

Fig. 10. Droplet motion induced by Stefan Flow and wall boundary conditions Physical time between pictures is 0.01 s.

length of the domain in the x-direction is $lx = 4R_G$ and in the y-direction $ly = 8lx$. Computations are carried out on a 64×512 grid size. An initialization step is performed before starting the computation in order to get smooth initial conditions for temperature and mass fraction. During this preliminary step, the drop is considered as static and only diffusive terms are involved. Fig. 11 shows the mass fraction field evolution for different times. The coupling of a general formalism with accurate and robust numerical methods enables us to capture quite subtle effects, such as the non-homogeneous vaporization rate around the drop presented in Fig. 12: we observe that the vaporization rate is much larger in front of the drop, where vapor mass fraction gradients are stronger, than behind the drop, in the vapor plume where the vapor mass fraction is more concentrated and thus induces lower vapor mass fraction gradients. In Fig. 13, we present the temperature field evolution for different times, and Fig. 14 shows the ability of the method to describe spatial temperature variation inside

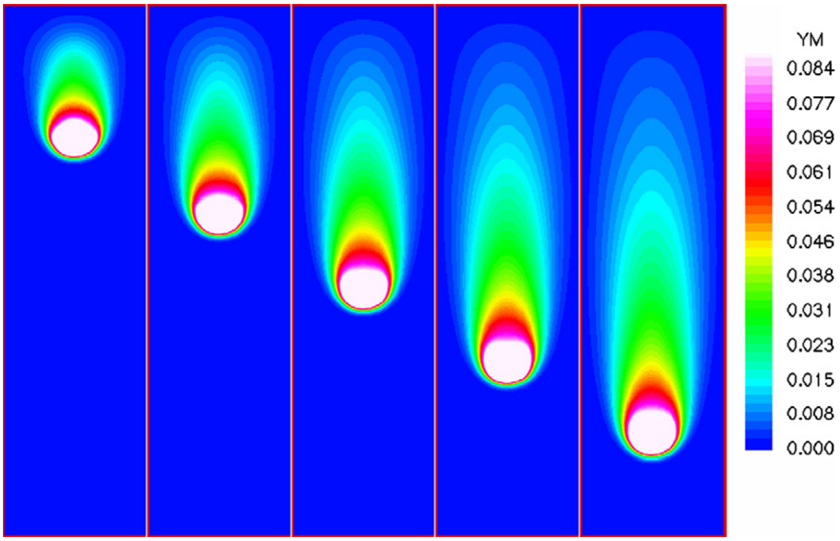

Fig. 11. Temporal evolution of the vapor mass fraction field Physical time between pictures is 0.0015 s.

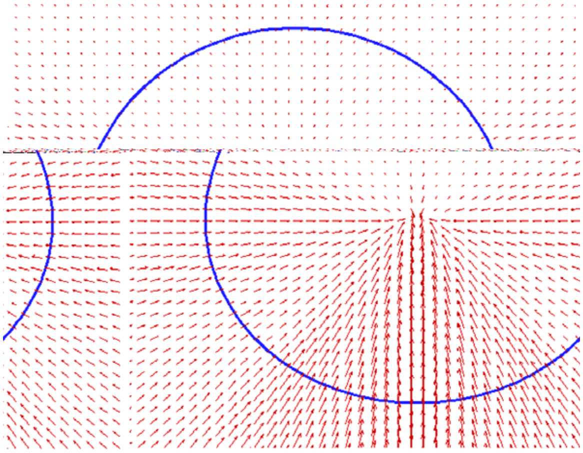

Fig. 12. Non-homogeneous vaporization vector and its extension around the drop $t = 0.0075$ s.

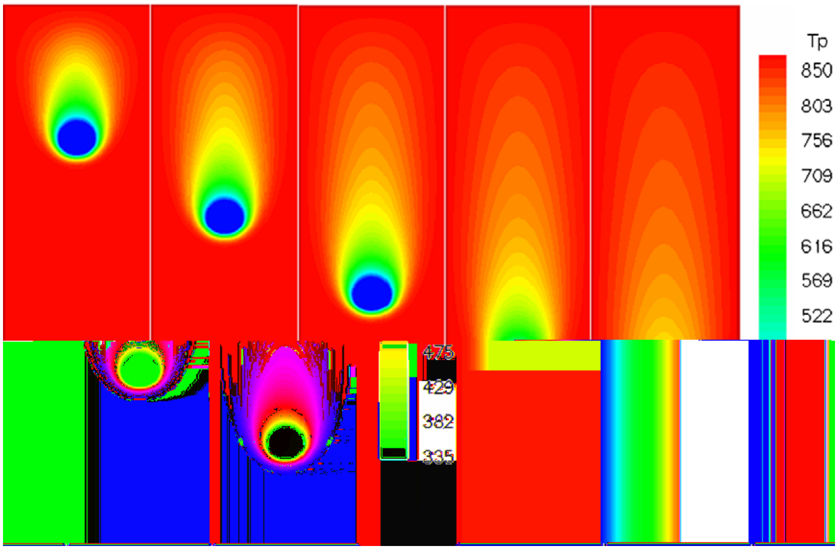

Fig. 13. Temporal evolution of the temperature field Physical time between pictures is 0.0015 s.

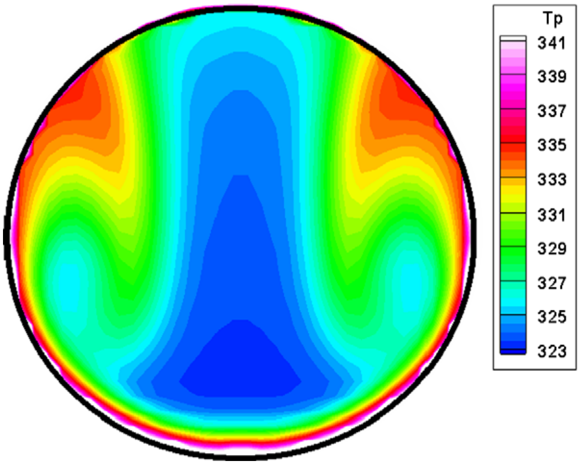

Fig. 14. Temperature field inside the drop $t = 0.0075$ s.

the drop. In particular, the development of a vortex inside the drop due to the viscosity jump between the liquid phase and the gas phase can lead to singular spatial temperature distribution inside the drop [3,5,22], as is shown in Fig. 14. Moreover, we observe in Figs. 11 and 13 that the drop shape is no longer spherical and this shows the importance of defining the local vaporization rate per unit of surface area.

5. Conclusions

The developments and applications of numerical tools for coupling interface tracking methods with reactive flows are presented, in particular for liquid–gas vaporizing flows. This challenging task is carried out using the level set method for interface tracking associated with the ghost fluid method to impose accurate jump conditions across the interface.

Specific care is taken with the accurate extension of discontinuous variables to minimize numerical diffusion across the interface for mass fraction and temperature field, to avoid artificial heating of the droplet. Moreover, in order to compute the Stefan flow around the drop accurately, we have developed an original divergence-free velocity extension of the liquid velocity field in the gas phase. Various numerical examples show that the method is accurate, robust and able to capture physical effects. The main interests of the method are its ability to handle phase change with heat and mass transfer and to avoid unrealistic velocity field developments that could affect numerical solutions. Moreover extension to 3D configuration is straightforward.

Future work will deal with a more complete modeling of vaporization, by taking into account gas density variations due to thermal and mixing effects, multi-component liquids and interaction between vaporization and combustion. In particular, future investigations will emphasize on the validations of numerical results against experimental data.

Acknowledgments

This research is part of the joint CNRS-ONERA ASTRA program on "Experimental and simulation methods for multi-component sprays". We thank also Ms. D. Moscato for improving the English.

References

[1] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2003) 349–355.

[2] E. Bassano, Numerical simulation of thermo-solutal-capillary migration of a dissolving drop in a cavity, Int. J. Numer. Methods Fluids 41 (2003) 765–788.

[3] A. Berlemont, M.S. Grancher, G. Gouesbet, Heat and mass transfer coupling between vaporizing droplets and turbulence using a Lagrangian approach, Int. J. Heat Mass Transfer 38 (16) (1995) 3023–3034.

[4] X. Calimez, Simulation à petite échelle par une méthode V.O.F. d'écoulements diphasiques réactifs, Ph.D. thesis, Ecole Centrale Paris, EM2C, 1998.

[5] C.H. Chiang, M.S. Raju, W.A. Sirignano, Numerical analysis of convecting, vaporizing fuel droplet with variable properties, Int. J. Heat Mass Transfer 35 (5) (1992) 1307–1324.

[6] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (The Ghost Fluid Method), J. Comput. Phys. 152 (1999) 457–492.

[7] F. Gibou, R. Fedkiw, L.T. Chieng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.

[8] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2003) 577–601.

[9] G.S. Jiang, C.W. Shu, Efficient implementation of weighted essentially non-oscillatory schemes, J. Comput. Phys. 126 (1996) 202–228.

[10] D. Juric, G. Tryggvason, Computations of boiling flows, Int. J. Multiphase Flow 24 (3) (1998) 387–410.

[11] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.

[12] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.

[13] X.-D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.

[14] D. Nguyen, R. Fedkiw, M. Kang, A boundary condition capturing method for incompressible flame discontinuities, J. Comput. Phys. 172 (2001) 71–98.

[15] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[16] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.

[17] R. Peyret, T.D. Taylor, Computational Method for Fluid Flow, Springer-Verlag, 1983.

[18] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (6) (1999) 775–793.

[19] J. Qian, G. Tryggvason, C.K. Law, A front tracking method for the motion of premixed flames, J. Comput. Phys. 144 (1998) 52–69.

[20] C.W. Rhee, L. Talbot, J.A. Sethian, Dynamical behaviour of a premixed turbulent open V-flame, J. Fluid Mech. 300 (1995) 87–115.

[21] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.

[22] W.A. Sirignano, Fluid Dynamics and Transport of Droplets and Sprays, Cambridge University Press, 1999.

[23] G. Son, V.K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, J. Heat Transfer 120 (1998).

[24] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.

[25] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (2003) 110–136.

[26] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[27] X. Wang, T.L. Jackson, L. Massa, Numerical simulation of heterogeneous propellant combustion by a level set method, Combust. Theory Modell. 8 (2004) 227–254.

[28] S. Welch, J. Wilson, A volume of fluid based method for fluid flows with phase change, J. Comput. Phys. 160 (2000) 662–682.