Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 206 (2005) 227–251
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A coupled quadrilateral grid level set projection method applied to ink jet simulation

Jiun-Der Yu [a], Shinri Sakai [b], James Sethian [c,*,1]

[a] Epson Research and Development, Inc., 3145 Porter Drive, Suite 104, Palo Alto, CA 94304, USA
[b] Seiko Epson Corporation, TP Development Department, 80 Harashinden, Hirooka, Shiojiri-shi, Nagano-ken 399-0785, Japan
[c] Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720, USA

Received 21 June 2004; received in revised form 18 October 2004; accepted 6 December 2004
Available online 25 January 2005

Abstract

A coupled level set-projection method on quadrilateral grids is developed for piezoelectric ink jet simulations. The model is based on the Navier–Stokes equations for incompressible two-phase flows in the presence of surface tension and density jump across the interface separating ink and air, coupled to an electric circuit model which describes the driving mechanism behind the process, and a macroscopic contact model which describes the air–ink–wall dynamics. We simulate the axisymmetric flow on quadrilateral grids using a combination of second-order finite difference projection methods to solve the fluid equations and level set methods to track the air/ink interface. To improve the mass conservation performance of the coupled level set method, a bicubic interpolation is combined with the Fast Marching Method for level set re-initialization on quadrilateral grids. The numerical method is used to analyze the motion of the interface, droplet pinch off, formation of satellites, effect of nozzle geometry on droplet size and motion, and the dynamics for droplet landing. The simulations are faithful to the dimensions and physics of a particular class of inkjet devices.

1. Problem description and previous work

The goal of this work is to develop a numerical simulation tool for fluid flow phenomena associated with ink jet printers. The physical goal is to analyze the motion of the boundary, pinch off of droplets, formation

* Corresponding author. Tel.: +1 510 642 2721; fax: +1 510 642 8204.
E-mail address: sethian@math.berkeley.edu (J. Sethian).
[1] This author was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, US Department of Energy, under Contract Number DE-AC03-76SF00098, and the Division of Mathematical Sciences of the National Science Foundation.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.12.012

of satellites, and the effect of nozzle geometry on ink ejection size and motion. In order to do so, the underlying algorithms should be able to faithfully discretize non-rectangular geometries, accurately capture two-phase flows through an axisymmetric nozzle, handle complicated topological change of ink droplets, conserve mass to a good approximation, and couple to external models which simulate the ink cartridge, supply channel, pressure chamber, and piezoelectric actuator.

This paper is the second of two papers on coupling level set methods to projection methods for ink jet simulations. In previous work [32], an obstacle cell method was used together with a first order in time, second order in space projection method with first order reinitialization scheme to study ink jet dynamics near the printhead. In this paper, we present a coupled level set projection method on quadrilateral grids. The new components of this work include:

- We develop projection methods on quadrilateral grids to faithfully discretize the body geometry. We provide relevant equations of motion and details on how to transform these equations from the physical space to the computational space in Section 2.
- We develop a second order accurate in space and time scheme in this quadrilateral body-fitted setting.
- We derive a transformed viscosity term for a quadrilateral grid in an axisymmetric coordinate system.
- We extend Chopp's [9] bicubic reinitialization scheme to quadrilateral meshes to provide a combined bicubic interpolation and Fast Marching Method for reinitialization, and demonstrate its increased fidelity for mass conservation.
- We extend our simulations beyond the near printhead field and examine the full cycle from ejection, through bubble breakup, on through to droplet landing, and provide detailed comparison with experimental results.

In Fig. 1, we show the typical structure of an ink jet nozzle; the actual geometry is axisymmetric and is not drawn to scale. Ink is stored in a bath reservoir (cartridge), and driven through the nozzle in response to a dynamic pressure at the lower boundary. The dynamics of incompressible flow through the nozzle, coupled to surface tension effects along the ink–air interface and boundary conditions along the wall,

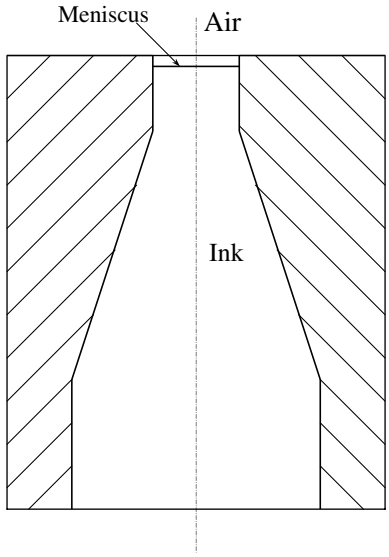

Fig. 1. The cross section view of an ink jet nozzle.

act to determine the shape of the interface as it moves. A negative pressure at the lower boundary induces a backflow which causes a bubble to pinch off.

Our model assumes the axisymmetric Navier–Stokes equations for two-phase immiscible incompressible flow with surface tension and density jumps across an infinitely thin immiscible interface separating ink and air, each with constant viscosity and density. A contact model is devised to capture air–ink–wall dynamics.

Several different numerical simulations of the ink jet process have been performed in recent years, see, for example, Aleinov et al. [1], Sou et al. [26], and Yu et al. [32]. In our approach, we employ second-order finite difference projection methods to solve the fluid flow equations, level set methods to capture the motion of the fluid interface, and body-fitted quadrilateral grids to faithfully discretize the nozzle geometry. A large number of background references for this approach are given in [32]; here, we briefly mention the original paper on projection methods for incompressible flow by Chorin [10], second-order Godunov-type improvements by Bell et al. [6], the finite-element approximate projection by Almgren et al. [2], and the extension of these techniques to quadrilateral grids (see, for example, Bell et al. [5]) and to moving quadrilateral grids (see Trebotich and Colella [31]). On the interface tracking side, level set methods, introduced in Osher and Sethian [16], rely in part on the theory of curve and surface evolution given in Sethian [19,20] and on the link between front propagation and hyperbolic conservation laws discussed in Sethian [21]; these techniques recast interface motion as a time-dependent Eulerian initial value partial differential equation. For details about projection methods and their coupling to level set methods, see Almgren et al. [2,4], Bell et al. [6], Bell and Marcus [7], Chang et al. [8], Chorin [10], Puckett et al. [17], Sussman and Smereka [28], Sussman et al. [27,29], and Zhu and Sethian [33].

The application of finite difference techniques to complex geometry must account for boundary conditions. Methods include the obstacle cell [13] method, use of a boundary-embedded Cartesian grid, or a body-fitted quadrilateral grid. The method of obstacle cell results in the staircase pattern at the slant part of the nozzle wall [32]. The boundary-embedded Cartesian grid is very powerful for simulations of combustion or atmospheric flows, however it leads to the "cut cell" problem in which very tiny boundary cells are generated; here, flux-redistribution has to be performed. The major advantage of body-fitted quadrilateral grids for ink jet simulations is that the nozzle wall is a grid line and, hence, it facilitates the implementation of contact models. For details about the boundary-embedded Cartesian grid and body-fitted quadrilateral grid, see Almgren [3] and references cited there.

Several subsections describing ink jet dynamics, piezoelectric performance, and contact models are abbreviated versions taken from [32]; we refer the interested reader to that work for further details.

2. Level set formulation

2.1. Governing equations

The two-phase flow consists of fluid #1 (ink) and #2 (air) whose densities and viscosities are denoted respectively by ρ_i, μ_i, $i = 1, 2$. The governing equations consist of the continuity equation and Navier–Stokes equations:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\rho(\phi)}\nabla p + \frac{1}{\rho(\phi)Re}\nabla \cdot (2\mu(\phi)\mathscr{D}) - \frac{1}{\rho(\phi)We}\kappa(\phi)\delta(\phi)\nabla\phi. \tag{2}$$

In the above,

$$\mathscr{D} = \tfrac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}}] \tag{3}$$

is the rate of deformation tensor. The density ratio, viscosity ratio, Reynolds number, and Weber number are defined by

$$\begin{aligned}
\rho(\phi) &= \begin{cases} 1 & \text{if } \phi \geqslant 0, \\ \rho_2/\rho_1 & \text{if } \phi < 0, \end{cases} \\
\mu(\phi) &= \begin{cases} 1 & \text{if } \phi \geqslant 0, \\ \mu_2/\mu_1 & \text{if } \phi < 0, \end{cases} \\
Re &= \frac{\rho_1 UL}{\mu_1}, \\
We &= \frac{\rho_1 U^2 L}{\sigma},
\end{aligned} \tag{4}$$

where U is a velocity scale, L is a length scale, and σ is the surface tension coefficient.

Since the interface moves with the fluid, the evolution of the level set is governed by

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0. \tag{5}$$

We choose this form because the interface moves advectively.

Since Eqs. (1) and (2) are expressed in terms of the vector notation, they assume the same form in Cartesian coordinates and axisymmetric coordinates.

2.2. Equations on quadrilateral meshes

We are interested in computing in reasonably complex geometries, in which rectangular grids may not work well. Fig. 7 in Yu et al. [32] shows the ladder case pattern at the shrinking part of the nozzle as a result of the rectangular finite difference grid. Instead, we prefer to build finite differences schemes on a body-fitted geometry generated by a quadrilateral grid; in addition, we wish to extend previous work to axisymmetric cases. To do so, we reformulate the governing equations (obtained by following [5,31]), with some additional care taken to transform the viscosity and surface tension terms to the quadrilateral axisymmetric setting.

Suppose we have a continuous transformation Φ which maps the grid points in a computational space $\Xi = (\xi,\eta)$ to the physical space $X = (r,z)$ (see Fig. 2):

$$X = \Phi(\Xi). \tag{6}$$

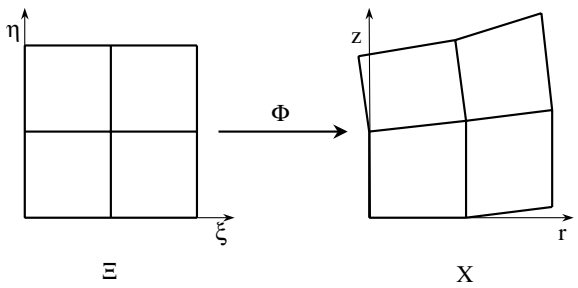

Fig. 2. Coordinate transformation.

The Jacobian and the transformation matrix (metric) are defined by

$$J = g \det \nabla_{\Xi} \Phi = g \det \begin{pmatrix} r_\xi & r_\eta \\ z_\xi & z_\eta \end{pmatrix},$$
$$\boldsymbol{T} = g^{-1} J [\nabla_{\Xi} \Phi]^{-1} = \begin{pmatrix} z_\eta & -r_\eta \\ -z_\xi & r_\xi \end{pmatrix}, \tag{7}$$

where $g = 2\pi r$ for the axisymmetric coordinate system. For the convenience of future discussion, we also define the transformed convection velocity as

$$\bar{\boldsymbol{u}} = g \boldsymbol{T} \boldsymbol{u}. \tag{8}$$

The above definitions for axisymmetric coordinate systems can be easily extended to the two-dimensional Cartesian system by the following substitution

$$r \to x, \quad z \to y, \quad g \to 1. \tag{9}$$

For our ink jet simulations, a body-fitted quadrilateral grid in the physical space is shown in Fig. 3. The corresponding uniform square grid system in the computational space is shown in Fig. 4. We first note that the grid lines in the physical space are not orthogonal. An almost orthogonal quadrilateral system would facilitate the implementation of boundary conditions and contact models. Nonetheless, the grid in Fig. 3 is easy to generate, and adequately (see below) handles the boundary conditions. The algorithms developed below work for any rectangular grid systems; for our ink jet simulations, we use the uniform square grid shown in Fig. 4 in the computational space.

In the following subsections, we show the derivation of the transformed viscosity term and surface tension term in the computational space.

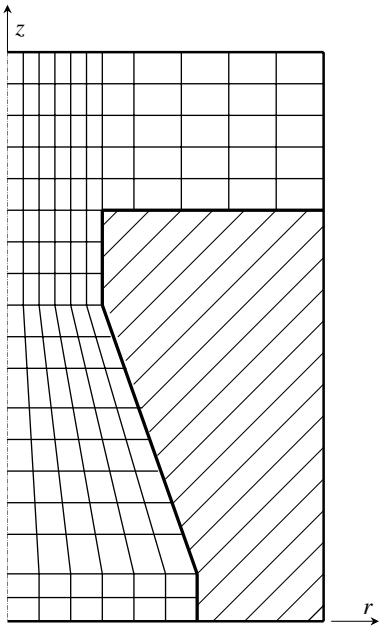

Fig. 3. A boundary-fitted quadrilateral grid for ink jet simulation.

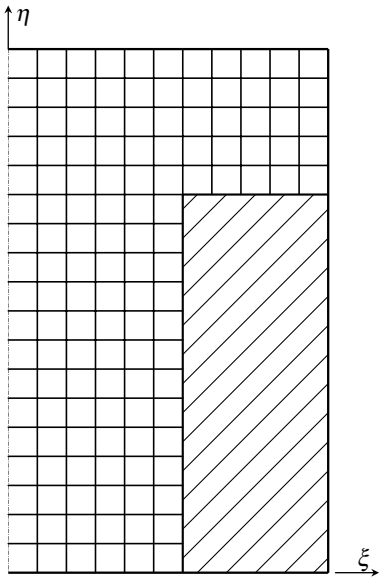

Fig. 4. The uniform square grid in the computational space.

2.2.1. Axisymmetric quadrilateral viscosity term

The complexity of the transformed viscosity term comes from the θ-related tensor component of the velocity gradient, which in the diadic form is

$$\nabla \boldsymbol{u} = (u_r \boldsymbol{e}_r \boldsymbol{e}_r + v_z \boldsymbol{e}_z \boldsymbol{e}_z + u_z \boldsymbol{e}_z \boldsymbol{e}_r + v_r \boldsymbol{e}_r \boldsymbol{e}_z) + \frac{u}{r} \boldsymbol{e}_\theta \boldsymbol{e}_\theta. \tag{10}$$

The r and z-related tensor components can be transformed by following Bell et al. [5]

$$\nabla \boldsymbol{u} - \frac{u}{r} \boldsymbol{e}_\theta \boldsymbol{e}_\theta = g J^{-1} \boldsymbol{T}^{\mathrm{T}} \nabla_\Xi \boldsymbol{u}. \tag{11}$$

The θ-related component turns out to not need any sort of special transformation, since we have that

$$\nabla \cdot \left(\mu(\phi) \frac{u}{r} \boldsymbol{e}_\theta \boldsymbol{e}_\theta \right) = -\frac{u}{r^2} \boldsymbol{e}_r. \tag{12}$$

By combining (10)–(12) and the definition of the rate of deformation tensor, we obtain the viscosity term on quadrilateral grids

$$\frac{1}{\rho(\phi) Re} \nabla \cdot (2\mu(\phi) \mathscr{D}) = \frac{1}{\rho(\phi) Re} \left\{ J^{-1} \nabla_\Xi \cdot [g^2 J^{-1} \mu(\phi) (\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}} \nabla_\Xi \boldsymbol{u} + (\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}} \nabla_\Xi \boldsymbol{u})^{\mathrm{T}})] - \frac{2\mu(\phi) u}{r^2} \boldsymbol{e}_r \right\}. \tag{13}$$

2.2.2. Axisymmetric quadrilateral curvature and surface tension terms

Since

$$\kappa(\phi) = \nabla \cdot \left(\frac{\nabla \phi}{| \nabla \phi |} \right) = J^{-1} \nabla_\Xi \cdot \left(g \boldsymbol{T} \frac{\boldsymbol{T}^{\mathrm{T}} \nabla_\Xi \phi}{| \boldsymbol{T}^{\mathrm{T}} \nabla_\Xi \phi |} \right), \tag{14}$$

the transformed surface tension term is easily obtained:

$$-\frac{g\delta(\phi)}{J^2\rho(\phi)We}\nabla_{\Xi}\cdot\left(g\boldsymbol{T}\frac{\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi}{|\,\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi\,|}\right)(\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi). \tag{15}$$

Combining (13)–(15), we obtain the transformed governing equations

$$\begin{aligned}
&\frac{\partial\boldsymbol{u}}{\partial t}+J^{-1}(\bar{\boldsymbol{u}}\cdot\nabla_{\Xi})\boldsymbol{u}=-\frac{1}{\rho(\phi)J}g\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}p+(\text{Viscosity term})+(\text{Surface tension}),\\
&\nabla_{\Xi}\cdot\bar{\boldsymbol{u}}=0,\\
&\frac{\partial\phi}{\partial t}+J^{-1}\bar{\boldsymbol{u}}\cdot\nabla_{\Xi}\phi=0,
\end{aligned} \tag{16}$$

where the viscosity term is given in (13) and the surface tension term is given in (15).

We make several comments here. First, Eqs. (13)–(16) are derived for a quadrilateral grid in the axisymmetric coordinate system; however, they can be used for two-dimensional flow problems if one neglects the last term in (13) and uses the substitution (9). Second, one can easily check Eq. (13) by reducing the computational space $\Xi=(\xi,\eta)$ to the physical space $X=(r,z)$. For this case, the transformation matrix reduces to the identity matrix and the Jacobian to g. Third, we note that ∇_{Ξ} and $\nabla_{\Xi}\cdot$ are "matrix operators" while ∇ and $\nabla\cdot$ are vector operators. When a vector operator is put in front of a vector quantity, it not only "operates" on the magnitude of the vector quantity but also on the direction. Here, we apply the matrix operators ∇_{Ξ} and $\nabla_{\Xi}\cdot$ to scalars or matrices, and hence the "direction" is not relevant. Fourth, the quadrilateral-grid algorithm is defined in terms of the Jacobian J and metric $\boldsymbol{T}$. The elements in the metric and the Jacobian are calculated using appropriate grid point locations. However, the algorithm does not explicitly use the continuous mapping Φ. Only the grid point locations in the physical space are needed.

2.3. Boundary conditions and contact model

On solid walls, we assume that both the normal and tangential components of the velocity vanish (this must be amended at the triple point). At both inflow and outflow, our formulation allows us to prescribe either the velocity

$$\boldsymbol{u}=\boldsymbol{u}^{\mathrm{BC}} \tag{17}$$

or the pressure boundary condition

$$p=p^{\mathrm{BC}},\quad \frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}=0, \tag{18}$$

where $\boldsymbol{n}$ denotes the unit normal to the inflow or outflow boundary.

To numerically simulate the ejection of ink droplets, one needs to prescribe a velocity or pressure at the inflow to the nozzle. However, only the input voltage to the piezoelectric actuator is known. The equivalent circuit model by Sakai [18] is employed to handle the problem. The equivalent circuit, which includes the effect of ink cartridge, supply channel, vibration plate, and piezoelectric actuator, simulates the ink velocity and pressure at nozzle inflow with a given dynamic voltage. By solving the equivalent circuit and the flow equations in turn, one simulates a real ink jet. A typical driving voltage pattern and a typical inflow pressure are as shown in Figs. 5 and 6. The driving voltage is such that the ink is first pulled back, pushed and fired, and then pulled back to get ready for the next jetting. The inflow pressure shown in Fig. 6 reflects the reaction of a typical nozzle-ink channel–actuator–cartridge system to the applied voltage. The pressure pattern contains a higher frequency signal. It is basically the fundamental natural frequency of the system, which is five to six times higher than the driving voltage frequency in this case.

At the triple point, where air and ink meet at the solid wall, we adopt the slipping contact line model discussed in detail in [32]. The contact angle θ is the angle made by the air–liquid interface and the solid, mea-

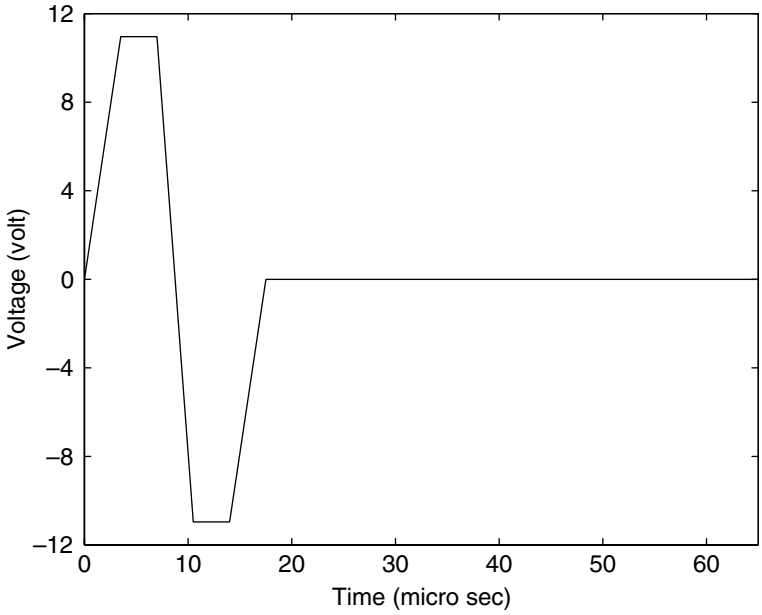

Fig. 5. Typical ink jet driving voltage.

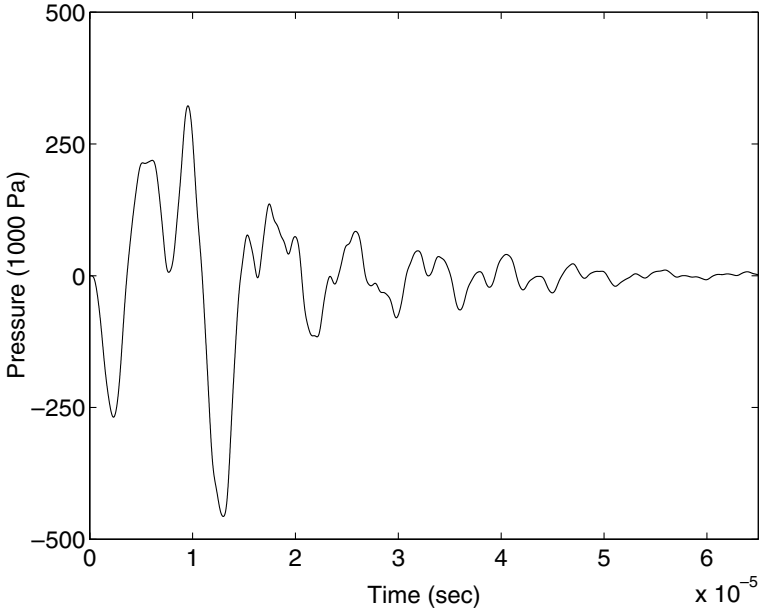

Fig. 6. Typical inflow pressure.

sured from the side of the liquid by approaching the contact line (i.e. the triple point) as close as possible. The advancing critical contact angle θ_a and receding critical contact angle θ_r are the maximum and minimum contact angles for the triple point to stay. The velocity v_B is the tangential velocity of the point on the inter-

face at 0.5 μm from the triple point. The triple point is allowed to move toward the air side if $\theta \geqslant \theta_a$ and $v_B > 0$. The triple point is allowed to move toward the liquid side if $\theta \leqslant \theta_r$ and $v_B < 0$. If the triple point is not allowed to move, the boundary condition at the solid wall is the no-slip condition. If the triple point is allowed to move, the no-slip condition in a close vicinity of the triple point is switched to the free slip condition. For Epson's dye-based ink and print head nozzle wall, θ_a and θ_r are about 70° and 20°.

3. Numerical algorithms

In this paper, the superscript n (or $n+1$) denotes the time step, i.e. $u^n = u(t = t^n)$ and so on. Suppose we have quantities $\boldsymbol{u}^n$, $p^{n-1/2}$, ϕ^n. The purpose is to obtain $\boldsymbol{u}^{n+1}$, $p^{n+1/2}$, ϕ^{n+1} from the governing equations (16). Note that the pressure is retarded in time (by half a time step) in the following coupled second-order level set projection scheme.

3.1. Temporal discretization

The boundary condition on the nozzle wall is given from the contact model. The inflow pressure $p^{n+1/2}$ is given by the equivalent circuit.

3.1.1. Level set update

The level set is updated first by

$$\phi^{n+1} = \phi^n - \frac{\Delta t}{J}[\bar{\boldsymbol{u}} \cdot \nabla_\Xi \phi]^{n+1/2}. \tag{19}$$

The algorithm for the evaluation of the time-centered advection term $[\bar{\boldsymbol{u}} \cdot \nabla_\Xi \phi]^{n+1/2}$ will be explained later in this paper. Once ϕ^{n+1} is obtained, we compute $\phi^{n+1/2}$ by

$$\phi^{n+1/2} = \tfrac{1}{2}(\phi^n + \phi^{n+1}). \tag{20}$$

3.1.2. Semi-implicit algorithm for Navier–Stokes equations

Following [29,7,31], the temporal discretization is 2nd-order explicit for the advection term and semi-implicit for the viscosity term. In this scheme, the preliminary velocity $\boldsymbol{u}^*$ is first solved from the Navier–Stokes equations by

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + J^{-1}[(\bar{\boldsymbol{u}} \cdot \nabla_\Xi)\boldsymbol{u}]^{n+1/2} = -\frac{g}{\rho(\phi^{n+1/2})J}\boldsymbol{T}^{\mathrm{T}}\nabla_\Xi p^{n-1/2} + (\text{Viscosity term})^{n+*} + (\text{Surface tension})^{n+1/2}, \tag{21}$$

where

$$(\text{Viscosity})^{n+*} = \frac{1}{\rho(\phi^{n+1/2})Re}\left\{J^{-1}\nabla_\Xi \cdot [g^2 J^{-1}\mu(\phi^{n+1/2})(\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_\Xi \boldsymbol{u}^{n+*} + (\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_\Xi \boldsymbol{u}^{n+*})^{\mathrm{T}})] - \frac{2\mu(\phi^{n+1/2})u^{n+*}}{r^2}\boldsymbol{e}_r\right\} \tag{22}$$

and

$$\boldsymbol{u}^{n+*} = \tfrac{1}{2}(\boldsymbol{u}^n + \boldsymbol{u}^*). \tag{23}$$

Since the preliminary velocity $\boldsymbol{u}^*$ appears at both sides of (21), we have to invert the viscosity term to solve for $\boldsymbol{u}^*$ in each time step.

We apply a second-order explicit Godunov scheme for the advection term and the central difference for the viscosity term in (21), which will be explained later. We note that we use the time-centered level set $\phi^{n+1/2}$, which is obtained explicitly, for the evaluation of the viscosity term, and hence the viscosity term is not truly semi-implicit.

3.1.3. Projection for $\boldsymbol{u}^{n+1}$

In order to project the whole velocity and to obtain the whole pressure, the preliminary velocity is replaced by

$$\boldsymbol{u}^* \leftarrow \boldsymbol{u}^* + \frac{g\Delta t}{\rho(\phi^{n+1/2})J}\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}p^{n-1/2}. \tag{24}$$

Since the FEM projection is used in our work, the regular projection equation is used

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\rho(\phi^{n+1/2})}\nabla p^{n+1/2}. \tag{25}$$

Taking the divergence and noting that $\nabla\cdot\boldsymbol{u}^{n+1} = 0$, we have

$$\nabla\cdot\boldsymbol{u}^* = \nabla\cdot\left(\frac{\Delta t}{\rho(\phi^{n+1/2})}\nabla p^{n+1/2}\right). \tag{26}$$

The projection equation is elliptic. It reduces to a Poisson's equation if the density ratio $\rho(\phi^{n+1/2})$ is a constant. To facilitate code implementation, the following finite element formulation is used

$$\int_{\Omega}\boldsymbol{u}^*\cdot\nabla\psi\,\mathrm{d}\boldsymbol{x} = \int_{\Omega}\frac{\Delta t}{\rho(\phi^{n+1/2})}\nabla p^{n+1/2}\cdot\nabla\psi\,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_1}\psi\boldsymbol{u}^{\mathrm{BC}}\cdot\boldsymbol{n}\,\mathrm{d}S, \tag{27}$$

where ψ is the weight function, Γ_1 denotes all the boundary with prescribed inflow or outflow velocity $\boldsymbol{u}^{\mathrm{BC}}$. It is easy to verify using the divergence theorem that (27) implies (26) and the boundary condition at Γ_1

$$\frac{\Delta t}{\rho(\phi^{n+1/2})}\frac{\partial p^{n+1}}{\partial n} = (\boldsymbol{u}^* - \boldsymbol{u}^{\mathrm{BC}})\cdot\boldsymbol{n}. \tag{28}$$

In this work, the weight function is chosen to be piecewise bilinear and the velocity $\boldsymbol{u}^*$ is taken as constant in each cell. After the pressure $p^{n+1/2}$ is solved from Eq. (27), the velocity field $\boldsymbol{u}^{n+1}$ can be obtained by (25).

In our ink jet simulation, only the inflow pressure and outflow pressure are given. There is no prescribed inflow or outflow velocity. Hence the second term on the right hand side of (27) vanishes.

We employ the multigrid-preconditioned conjugate gradient method (MGPCG) to solve the linear system resulting from the finite element projection and that from the MAC projection (see Section 3.2.2). The preconditioner is a single multigrid V-cycle. The multicolor Gauss–Seidel relaxation is used as the smoother at each level except the bottom level, where the conjugate gradient method is used.

3.2. Spatial discretization

3.2.1. The quadrilateral grid

As shown in Fig. 7, the velocity components $\boldsymbol{u}^n_{i,j}$ and the level set $\phi^n_{i,j}$ are located at cell centers, and the pressure $p^n_{i,j}$ at grid points. The time-centered edge velocities and level set (i.e., the "predictors"), such as $\boldsymbol{u}^{n+1/2}_{i+1/2,j}$, $\phi^{n+1/2}_{i+1/2,j}$, are at the middle point of each edge.

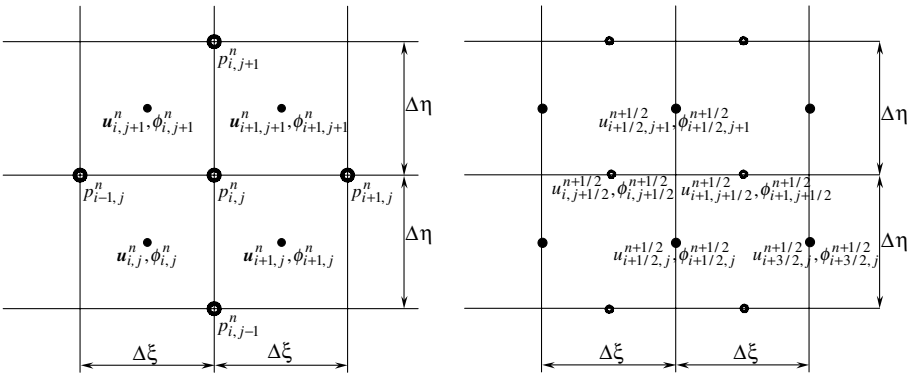

Fig. 7. Location of variables.

The transformation $X = \Phi(\Xi)$ is such that the grid in the computational space is composed of unit squares, i.e. the grid in the computational space has $\Delta\xi = \Delta\eta = 1$, and the quadrilateral grid in the physical space is body-fitted. The coordinates of cell centers in the physical space are

$$\tfrac{1}{4}(X_{i-1/2,j-1/2} + X_{i-1/2,j+1/2} + X_{i+1/2,j-1/2} + X_{i+1/2,j+1/2}). \tag{29}$$

Differences of the grid points are used to define the elements of the transformation matrix at cell centers. As an example, we have that

$$(X_\xi)_{i,j} = \tfrac{1}{2}(X_{i+1/2,j+1/2} - X_{i-1/2,j+1/2} + X_{i+1/2,j-1/2} - X_{i-1/2,j-1/2}). \tag{30}$$

3.2.2. The advection term

The algorithm for the advection terms is based on the unsplit, second-order Godunov type upwind method introduced by Colella [12], see also Colella [11]. It is a cell-centered predictor-corrector scheme. In the predictor step, we extrapolate the velocity and level set in space and time to obtain their cell edge values at $t^{n+1/2}$. In the corrector step, we compute the Godunov upwind fluxes which are then differenced to obtain an approximation to the advection terms.

Predictor: For the predictor, we use Taylor's series to extrapolate the velocity and level set at t^n to obtain their cell edge values at $t^{n+1/2}$. The partial derivative with respect to time in the extrapolation is substituted by the Navier–Stokes equations or by the level set convection equation. There are two extrapolated velocities and level sets for each cell edge. For example, for the cell edge between cells i, j and $i+1, j$, one can extrapolate from the left and have

$$\boldsymbol{u}^{n+1/2,L}_{i+1/2,j} = \boldsymbol{u}^n_{i,j} + \frac{1}{2}\boldsymbol{u}^n_{\xi,i,j} + \frac{\Delta t}{2}\boldsymbol{u}^n_{t,i,j} = \boldsymbol{u}^n_{i,j} + \left(\frac{1}{2} - \frac{\Delta t}{2J_{i,j}}\bar{u}^n_{i,j}\right)\boldsymbol{u}^n_{\xi,i,j} - \frac{\Delta t}{2J_{i,j}}(\bar{v}\boldsymbol{u}_\eta)^n_{i,j} + \frac{\Delta t}{2}\boldsymbol{F}^n_{i,j}, \tag{31}$$

where

$$\boldsymbol{F}^n_{i,j} = \left\{ -\frac{g}{\rho(\phi)J}\nabla_\Xi p + (\text{Viscosity term}) + (\text{Surface tension}) \right\}^n_{i,j} \tag{32}$$

and extrapolate from the right to produce

$$\begin{aligned}\boldsymbol{u}^{n+1/2,R}_{i+1/2,j} &= \boldsymbol{u}^n_{i+1,j} - \frac{1}{2}\boldsymbol{u}^n_{\xi,i+1,j} + \frac{\Delta t}{2}\boldsymbol{u}^n_{t,i+1,j} \\ &= \boldsymbol{u}^n_{i+1,j} - \left(\frac{1}{2} + \frac{\Delta t}{2J_{i+1,j}}\bar{u}^n_{i+1,j}\right)\boldsymbol{u}^n_{\xi,i+1,j} - \frac{\Delta t}{2J_{i+1,j}}(\bar{v}\boldsymbol{u}_\eta)^n_{i+1,j} + \frac{\Delta t}{2}\boldsymbol{F}^n_{i+1,j}.\end{aligned} \tag{33}$$

We use the monotonicity-limited 2nd-order central difference for the evaluation of the normal slopes, i.e. $\boldsymbol{u}^n_{\xi,i,j}$ and $\boldsymbol{u}^n_{\xi,i+1,j}$ in this case. The limiting is done on each component of the velocity at t^n separately. The transverse derivative terms $(\bar{v}\boldsymbol{u}_\eta)^n_{i,j}$ and $(\bar{v}\boldsymbol{u}_\eta)^n_{i+1,j}$ are evaluated by the simple upwind difference.

Corrector: As was noted by Bell et al. [5], $\bar{u}^{n+1/2}_{i+1/2,j}\boldsymbol{u}^{n+1/2}_{i+1/2,j}$ represents the flux of $\boldsymbol{u}^{n+1/2}$ through cell edge $i+1/2,j$, if we let

$$(X_\eta)_{i+1/2,j} = X_{i+1/2,j+1/2} - X_{i+1/2,j-1/2} \tag{34}$$

in the calculation of the transformed convection velocity $\bar{u}^{n+1/2}_{i+1/2,j}$. Similarly, $\bar{v}^{n+1/2}_{i,j+1/2}\boldsymbol{u}^{n+1/2}_{i,j+1/2}$ is the flux of $\boldsymbol{u}^{n+1/2}$ through cell edge $i,j+1/2$, if

$$(X_\xi)_{i,j+1/2} = X_{i+1/2,j+1/2} - X_{i-1/2,j+1/2}. \tag{35}$$

These suggest the finite volume-type differencing for the corrector given by Bell et al. [5]. The obtained cell edge convection velocities are, in general, not divergence-free. In this work, we use an intermediate MAC projection to make all the normal advection velocities divergence-free. Suppose q is a function which is smooth enough and $\boldsymbol{u}^{\mathrm{e}}$ are the edge convection velocities. We want

$$\boldsymbol{u}^{\mathrm{e}} - \frac{1}{\rho(\phi^n)}\nabla q \tag{36}$$

to be divergence-free. By taking the divergence of (36), we have, in the physical space,

$$\nabla\cdot\left(\frac{1}{\rho(\phi^n)}\nabla q\right) = \nabla\cdot\boldsymbol{u}^{\mathrm{e}}. \tag{37}$$

In the computational space, the above MAC projection equation is transformed as

$$\nabla_\Xi\cdot\bar{\boldsymbol{u}}^{\mathrm{e}} = \nabla_\Xi\cdot\left(\frac{g^2}{\rho(\phi^n)J}\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_\Xi q\right). \tag{38}$$

The boundary conditions for q are similar to those for the pressure. At the inflow or outflow, if a pressure is given, we use

$$q = \Delta t\Delta p^{\mathrm{BC}}, \tag{39}$$

where Δp^{BC} is the increment of the boundary pressure. If a velocity is prescribed, we have

$$\frac{\partial q}{\partial \boldsymbol{n}} = \boldsymbol{n}\cdot[\boldsymbol{u}^{\mathrm{e}} - \boldsymbol{u}^{\mathrm{BC}}(t^{n+1/2})]. \tag{40}$$

The discretization of (38) can be easily done by a finite volume type differencing. We observe that $\bar{u}_{i+1/2,j}$ (or $\bar{v}_{i+1/2,j}$) is a flux through the edge $i+1/2, j$ (or $i, j+1/2$) if (34) (or (35)) is used to define the metric at the mid point of cell edges. Hence (38) is discretized as

$$\bar{u}_{i+1/2,j} - \bar{u}_{i-1/2,j} + \bar{v}_{i,j+1/2} - \bar{v}_{i,j-1/2} = \mathscr{M}_{i+1/2,j} - \mathscr{M}_{i-1/2,j} + \mathscr{M}_{i,j+1/2} - \mathscr{M}_{i,j-1/2}, \tag{41}$$

where

$$\mathscr{M}_{i+1/2,j} = \left(\frac{g^2}{\rho(\phi^n)J}\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_\Xi q\right)_{i+1/2,j} \tag{42}$$

with $\mathscr{M}_{i-1/2,j}$, $\mathscr{M}_{i,j+1/2}$, and $\mathscr{M}_{i,j-1/2}$ defined similarly. In (42), the Jacobian $J_{i+1/2,j}$ and the metric $\boldsymbol{T}_{i+1/2,j}$ are calculated using (34). The gradient is by central difference.

After q is solved, we replace the edge advective velocities by

$$\bar{\boldsymbol{u}}^{\mathrm{e}} \leftarrow \bar{\boldsymbol{u}}^{\mathrm{e}} - \frac{g^2}{\rho(\phi^n)J}\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_\Xi q. \tag{43}$$

We have two comments here. First, for ink jet simulations in this work, the inflow pressure calculated by the equivalent circuit is used to drive the coupled level set projection code at each time step. The code solves the governing equations and feeds back the ink flow rate to the equivalent circuit, which then calculates a new inflow pressure for the next time step. Since pressure instead of velocity is given at the inflow, the velocity predictors at the inflow and outflow are simply extrapolated from the interior; they are also not upwinded in the corrector step. This dilemma does not go away if the inflow velocity is prescribed by the equivalent circuit. Since the actuator first pulls the ink back and then pushes to fire a droplet, the inflow velocity is negative at the beginning of each droplet ejection. The upwind direction is at outflow, where no velocity information is available.

Second, since geometry scales considered in this work are very small, the time step is dominated by the viscosity constraint (see 3.3). That is to say, both Δt and Δp^{BC} are proportional to the square of mesh size. Hence $\Delta t \Delta p^{\mathrm{BC}}$ is negligible and can be replaced by a homogeneous boundary condition in the MAC projection.

3.2.3. The viscosity term

The discretization of the viscosity term (13) is done using standard central differences. For example, at cell i, j on the computational grid, the first part at the right hand side of (13) is discretized as

$$J^{-1}\nabla_{\Xi}\cdot[g^2J^{-1}\mu(\phi)(\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\boldsymbol{u})] = \frac{1}{J_{i,j}}(\mathscr{L}_{i+1/2,j}-\mathscr{L}_{i-1/2,j}+\mathscr{L}_{i,j+1/2}-\mathscr{L}_{i,j-1/2}), \tag{44}$$

where

$$\mathscr{L}_{i+1/2,j} = [g^2J^{-1}\mu(\phi)(\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\boldsymbol{u})]_{i+1/2,j} \tag{45}$$

with $\mathscr{L}_{i-1/2,j}$, $\mathscr{L}_{i,j+1/2}$, and $\mathscr{L}_{i,j-1/2}$ defined similarly. In (45), the Jacobian, transformation matrix, relative viscosity, and velocity gradient at cell edge are calculated by averaging or central difference, namely

$$\begin{aligned}
&J_{i+1/2,j} = (J_{i,j}+J_{i+1,j})/2, \quad \boldsymbol{T}_{i+1/2,j} = (\boldsymbol{T}_{i,j}+\boldsymbol{T}_{i+1,j})/2,\\
&(\mu(\phi))_{i+1/2,j} = (\mu(\phi_{i+1,j})+\mu(\phi_{i,j}))/2,\\
&(\nabla_{\Xi}\boldsymbol{u})_{i+1/2,j} = \begin{bmatrix} u_{i+1,j}-u_{i,j} & v_{i+1,j}-v_{i,j} \\ \frac{u_{i+1,j+1}-u_{i+1,j-1}+u_{i,j+1}-u_{i,j-1}}{4} & \frac{v_{i+1,j+1}-v_{i+1,j-1}+v_{i,j+1}-v_{i,j-1}}{4} \end{bmatrix}.
\end{aligned} \tag{46}$$

3.2.4. The surface tension term

The surface tension term is also discretized using standard central differences. For example, the divergence in (15) is discretized as

$$\nabla_{\Xi}\cdot\left(g\boldsymbol{T}\frac{\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi}{|\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi|}\right) = \mathscr{K}_{i+1/2,j}-\mathscr{K}_{i-1/2,j}+\mathscr{K}_{i,j+1/2}-\mathscr{K}_{i,j-1/2}, \tag{47}$$

where

$$\mathscr{K}_{i+1/2,j} = \left(g\boldsymbol{T}\frac{\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi}{|\boldsymbol{T}^{\mathrm{T}}\nabla_{\Xi}\phi|}\right)_{i+1/2,j} \tag{48}$$

with $\mathscr{K}_{i-1/2,j}$, $\mathscr{K}_{i,j+1/2}$, and $\mathscr{K}_{i,j-1/2}$ defined similarly. Again, the velocity gradient and metric at cell edge are calculated by central difference or by averaging.

3.3. Interface thickness and time step

Because of the numerical difficulty caused by the Dirac delta function and by the sharp change of ρ and μ across the free surface, the Heaviside and Dirac delta functions are replaced by smoothed functions (see Sussman et al. [27] and Yu et al. [32]). The interface thickness is 2ϵ, where the parameter ϵ is related to the mesh size by $\epsilon = \alpha/2(\Delta r + \Delta z)$. The thickness of the interface reduces as we refine the mesh. Nonetheless, for any numerically practical choices, the interface will necessarily have some smearing. For the convergence study and simulations in Sections 5.1–5.3, α is set to be 2. For the droplet ejection and landing simulation in Section 5.4, $\alpha = 3$.

The time step constraint is determined by the CFL condition, surface tension, viscosity, and total acceleration. The constraint is given by (3.36) in Yu et al. [32], where $\boldsymbol{F}^n$ is defined by (32) for this work.

4. Level set re-initialization

To correctly capture the interface and accurately obtain the surface tension term, the level set needs to be maintained as a signed distance function to the interface. We re-initialize the level set as signed distance every few time steps using the combined bicubic interpolation and Fast Marching Method. The idea was proposed by Chopp [9]. We modify and extend it to quadrilateral grids.

4.1. Bicubic interpolation

For the central cell in Fig. 8, we would like to construct a local bicubic interpolation function $f(r,z)$ from which we can accurately solve for the zero level. A bicubic function is

$$f(r,z) = \sum_{m=0}^{3}\sum_{n=0}^{3} a_{m,n} r^m z^n \tag{49}$$

which satisfies

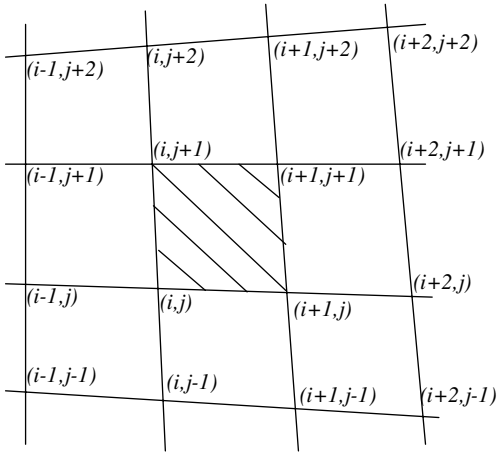

Fig. 8. Grids for level set.

$$
\begin{aligned}
&f_{k,l} = [f(r,z)]_{k,l} = \phi_{k,l},\\
&\left(\frac{\partial f}{\partial r}\right)_{k,l} = \left(\frac{\partial \phi}{\partial r}\right)_{k,l},\\
&\left(\frac{\partial f}{\partial z}\right)_{k,l} = \left(\frac{\partial \phi}{\partial z}\right)_{k,l},\\
&\left(\frac{\partial^2 f}{\partial r\,\partial z}\right)_{k,l} = \left(\frac{\partial^2 \phi}{\partial r\,\partial z}\right)_{k,l},
\end{aligned}
\tag{50}
$$

where

$$
k = i, i+1, \quad l = j, j+1. \tag{51}
$$

Note that there are 16 coefficients to decide in the bicubic interpolation function f and there are exactly 16 conditions in (50).

To obtain the values of $(\frac{\partial \phi}{\partial r})_{k,l}$, $(\frac{\partial \phi}{\partial z})_{k,l}$, and $(\frac{\partial^2 \phi}{\partial r\,\partial z})_{k,l}$ on the quadrilateral mesh, one has to use the metric matrix and the Jacobian associated with the mesh

$$
\left(\frac{\partial \phi}{\partial r}\ \frac{\partial \phi}{\partial z}\right) = J^{-1}\boldsymbol{T}^{\mathrm{T}}\begin{pmatrix}\frac{\partial \phi}{\partial \xi}\\ \frac{\partial \phi}{\partial \eta}\end{pmatrix}. \tag{52}
$$

So, for example, at the grid point (i,j) in Fig. 8, the first-order derivatives are

$$
\left(\frac{\partial \phi}{\partial r}\ \frac{\partial \phi}{\partial z}\right)_{i,j} = J_{i,j}^{-1}\boldsymbol{T}_{i,j}^{\mathrm{T}}\begin{pmatrix}\frac{\partial \phi}{\partial \xi}\\ \frac{\partial \phi}{\partial \eta}\end{pmatrix}_{i,j}, \tag{53}
$$

where

$$
\begin{aligned}
&\left(\frac{\partial \phi}{\partial \xi}\right)_{i,j} = (\phi_{i+1,j} - \phi_{i-1,j})/2,\\
&\left(\frac{\partial \phi}{\partial \eta}\right)_{i,j} = (\phi_{i,j+1} - \phi_{i,j-1})/2,\\
&\boldsymbol{T}_{i,j}^{\mathrm{T}} = \begin{pmatrix} z_{,\eta} & -z_{,\xi}\\ -r_{,\eta} & r_{,\xi}\end{pmatrix}_{i,j},\\
&J_{i,j} = (r_{,\xi}z_{,\eta} - r_{,\eta}z_{,\xi})_{i,j}.
\end{aligned}
\tag{54}
$$

The second-order derivative is

$$
\frac{\partial^2 \phi}{\partial r\,\partial z} = (\phi_{,r})_{,\xi}\xi_{,z} + (\phi_{,r})_{,\eta}\eta_{,z} = \phi_{,\xi\xi}\xi_{,r}\xi_{,z} + \phi_{,\xi\eta}(\xi_{,z}\eta_{,r} + \xi_{,r}\eta_{,z}) + \phi_{,\eta\eta}\eta_{,r}\eta_{,z}, \tag{55}
$$

where, at (i,j),

$$
\begin{aligned}
&\phi_{,\xi\xi} = \phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j},\\
&\phi_{,\eta\eta} = \phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1},\\
&\phi_{,\xi\eta} = (\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i+1,j-1} + \phi_{i-1,j-1})/4.
\end{aligned}
\tag{56}
$$

For the level set re-initialization using bicubic interpolation, one has to interpolate in all the cells that the zero level passes through and then solve for the new level set values for the four nodes of each of those cells. If the hatched cell contains a segment of the zero level, one just does the interpolation using (49)–(56). The remaining job is to solve the new distance for (i,j), $(i+1,j)$, $(i,j+1)$, $(i+1,j+1)$.

Let $\boldsymbol{x}^0$ be a node of the hatched cell where we would like to compute the distance to the zero level. Let $\boldsymbol{y}$ be a point on the zero level nearest to $\boldsymbol{x}^0$. We have

$$f(\boldsymbol{y}) = 0, \quad \nabla f(\boldsymbol{y}) \times (\boldsymbol{x}^0 - \boldsymbol{y}) = 0. \tag{57}$$

To solve (57), the standard Newton's method can be used. Chopp [9] suggested the following alternative:

$$\begin{aligned}
&\boldsymbol{\delta}_1 = -f(\boldsymbol{x}^k)\frac{\nabla f(\boldsymbol{x}^k)}{\nabla f(\boldsymbol{x}^k)\cdot\nabla f(\boldsymbol{x}^k)},\\
&\boldsymbol{x}^{k+1/2} = \boldsymbol{x}^k + \boldsymbol{\delta}_1,\\
&\boldsymbol{\delta}_2 = (\boldsymbol{x}^0 - \boldsymbol{x}^k) - \frac{(\boldsymbol{x}^0 - \boldsymbol{x}^k)\cdot\nabla f(\boldsymbol{x}^k)}{\nabla f(\boldsymbol{x}^k)\cdot\nabla f(\boldsymbol{x}^k)}\nabla f(\boldsymbol{x}^k),\\
&\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1/2} + \boldsymbol{\delta}_2,
\end{aligned} \tag{58}$$

where $k = 0,1,2,\ldots$ The iteration is continued until

$$|\boldsymbol{\delta}_1|^2 + |\boldsymbol{\delta}_2|^2 < 10^{-6}\Delta r\Delta z. \tag{59}$$

The obtained closest point $\boldsymbol{y}$ should lie in the quadrilateral cell defined by nodes (i,j), $(i+1,j)$, $(i,j+1)$, $(i+1,j+1)$. If not, the obtained distance value is disregarded.

4.2. Fast Marching Method

The Fast Marching Method (see [22]) is a Dijkstra-like finite difference scheme for computing the solution to the Eikonal equation. It is an $O(N \log N)$ method, where N is the number of points in the computational domain, and has been used in a wide variety of applications (see [23] for a review of some of the applications, as well as [30] for a related Dijkstra-like method and [25] for extensions of these approaches to more general static Hamilton–Jacobi equations.)

In particular, Fast Marching Methods have been extended to triangular grids [15,24]. To be able to use a version of the triangular Fast Marching Method, we change our quadrilateral grid to a triangular grid as in Fig. 9, where all the triangles are acute triangles. To start the Fast Marching Method to complete the reini-

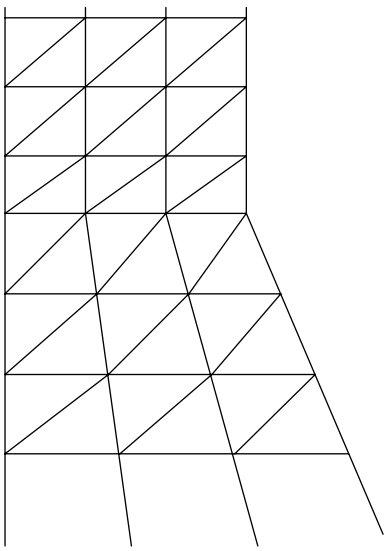

Fig. 9. The triangular grid for fast marching.

tialization, all nodes whose distance values are calculated by the bicubic interpolation are placed in the set of accepted points. Details of the triangular Fast Marching Method can be found in [15,24].

5. Numerical examples and discussions

5.1. Convergence study

For a convergence check of the code for ink jet simulations, we considered a typical nozzle as in Fig. 1. The diameter is 25 μm at the opening and 49.5 μm at the bottom. The length of the nozzle opening part, where the diameter is 25 μm, is 25 μm. The slant part is 55 μm and the bottom part is 7.5 μm.

The inflow pressure is determined by an equivalent circuit, which simulates the ink in the supply channel considering the effect of the ink cartridge, supply channel, vibration plate, and PZT actuator for given dynamic voltage. We assumed that the input voltage is given by Fig. 5, where the peak voltages are ±10.96 V. The corresponding inflow pressure is as shown in Fig. 6. The outflow pressure at the top of the solution domain is set to zero.

The solution domain was chosen to be $\{(r,z)|0 \leqslant r \leqslant 32\ \mu\text{m}, 0 \leqslant z \leqslant 501\ \mu\text{m}\}$. The critical contact angles are 70° for advancing and 20° for receding. The initial meniscus is assumed to be flat and 2.5 μm under the nozzle opening.

For the purpose of normalization, we chose the nozzle opening diameter (25 μm) to be the length scale and 6 m/s to be the velocity scale. The normalized solution domain is hence $\{(r,z)|0 \leqslant r \leqslant 1.28, 0 \leqslant z \leqslant 20.04\}$. Since the density, viscosity, and surface tension of Epson's dye-based ink are approximately

$$\rho_1 = 1070\ \text{kg/m}^3, \quad \mu_1 = 3.7 \times 10^{-3}\ \text{kg/m s}, \quad \sigma = 0.032\ \text{kg/s}^2, \tag{60}$$

we have the following non-dimensional parameters

$$Re = 43.38, \quad We = 30.09. \tag{61}$$

The density and viscosity of air used in the study are

$$\rho_2 = 1.225\ \text{kg/m}^3, \quad \mu_2 = 1.77625 \times 10^{-3}\ \text{kg/m s}. \tag{62}$$

To check the convergence of our code, we list in Tables 1–3 the time of droplet pinch off from the meniscus, droplet head velocity, and droplet volume obtained from our code using various meshes. All of these listed data are dimensionless. It is seen that the 32×432 mesh does not conserve mass well. The level set re-initialization using bicubic interpolation tends to gain mass on such coarse mesh.

Table 1
The time to pinch off by various meshes

Mesh number	32×432	64×864	96×1296	128×1728
Time to pinch off	7.3020	7.5780	7.6251	7.6245

Table 2
Droplet head velocities by various meshes

Mesh number, t	32×432	64×864	96×1296	128×1728
7.986	1.3017	1.2804	1.2890	1.2913
8.994	1.2979	1.3147	1.3379	1.3467
10.002	1.3739	1.3840	1.4041	1.4092
10.986	1.3783	1.3437	1.3608	1.3628
12.000	1.3357	1.3045	1.3174	1.3217
13.002	1.3197	1.2986	1.3134	1.3199

Table 3
Droplet volumes by various meshes

Mesh number, t	32×432	64×864	96×1296	128×1728
7.986	0.7984	0.7867	0.7913	0.7861
8.490	0.7747	0.7881	0.7920	0.7863
8.994	0.7675	0.7896	0.7925	0.7866
9.498	0.7767	0.7908	0.7930	0.7868
10.002	0.7877	0.7922	0.7935	0.7870

5.2. Droplet ejection

Experimental results for the case considered are shown in Fig. 10. The nozzle is not shown; instead, we show the position of nozzle exit at the lower edge of each of the six images. Because the droplet is small, the droplet velocity is high, and the camera shutter speed is not infinitely fast, the images are a little blurred (We note that exposure time for each image is in the order of one microsecond). The major droplet at $t = 50$ μs is actually more like a sphere than it appears in the last image in Fig. 10. The figure shows that the ink droplet leaves the nozzle sometime between $t = 30$ μs and $t = 35$ μs. Since the pinch off from the meniscus usually happens somewhere inside the nozzle, we could not see it in the experiment. The average droplet size from the experiment is 12.928 pico liters.

To compare with the experiment, simulation results from $t = 25$ μs to $t = 50$ μs, using the 64×864 quadrilateral mesh, are plotted in Fig. 11. The simulation shows the droplet pinches off at $t = 32.83$ μs, its tail leaves the nozzle at $t = 33.50$ μs, and the droplet size is 12.292 pico liters. All of these are reasonable.

It is interesting to note that there is no visible difference between the quadrilateral grid result (Fig. 11) and the uniform rectangular grid/obstacle cell result (Fig. 14 of [32]). We hence confirm that, for the case considered, the latter approach is reasonably accurate even if it causes the ladder case pattern. With a 70° advancing critical angle and a 20° receding critical angle, the nozzle wall is easy to wet but difficult to un-wet. Although the contact point can move back and forth during droplet ejection, it never retreats to the slant part of the nozzle. The fluid right next to the slant wall is always very slow whether the quadrilateral grid or uniform rectangular grid is used and the ladder case pattern is not crucial in the simulation. If the nozzle wall is easy to un-wet, the ladder case pattern would most likely downgrade the accuracy of simulations.

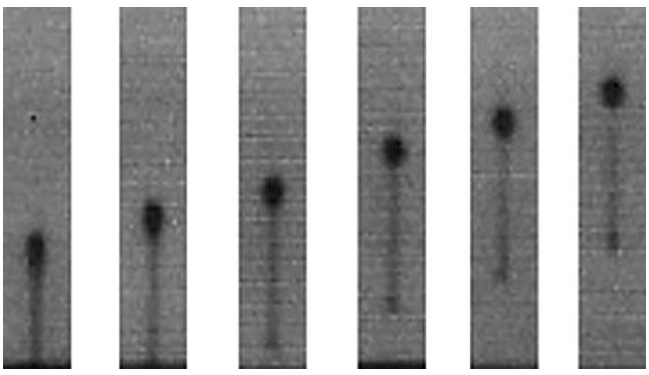

Fig. 10. Shape of the ink droplet at $t = 25$, 30, 35, 40, 45, 50 μs.

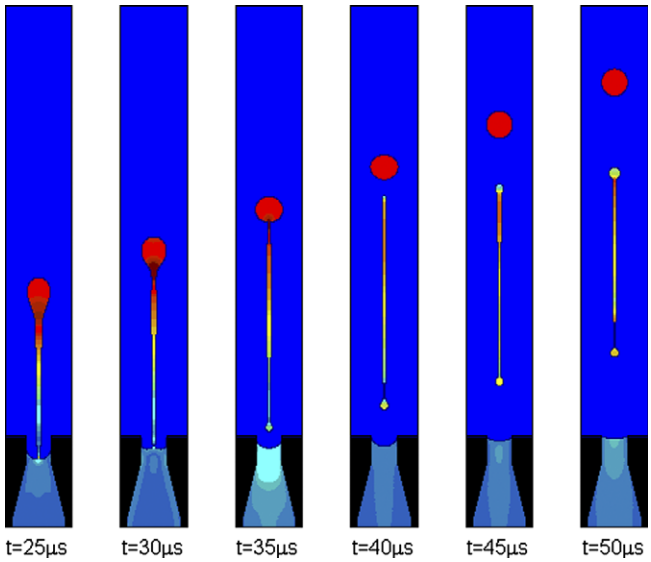

Fig. 11. Ejection of an ink droplet from simulation.

5.3. Droplet landing

As a second numerical example, the landing of ink droplets on substrates of various wetting properties was simulated. The ink droplet, flying at 7 m/s, is originally a sphere of diameter $d_0 = 25$ μm. Two substrates, hydrophobic (not easy to wet) and hydrophilic (easy to wet), were considered. The advancing and receding critical contact angles for the hydrophobic substrate are 60° and 90° while the angles for the hydrophilic substrate are 20° and 50°. A higher dynamic viscosity $\mu_1 = 4.5 \times 10^{-3}$ kg/m s is used in the simulation in order to compare with available experimental results. Otherwise the fluid properties are the same as the droplet ejection. Simulation results using 64×64 meshes are plotted in Fig. 12 and 13, where the moment that the droplet touches the substrates is denoted as $t = 0$ μs and the number in parentheses is the ratio of the maximum diameter to the original diameter.

We see from the results that the final shape of the droplet on substrate depends on the initial droplet landing velocity, substrate material properties, and fluid material properties. For our case of a 7 m/s landing ink droplet, the final shape, i.e. the height and width, strongly depends on the substrate property. In the hydrophobic case, the droplet shape remains basically unchanged from $t = 50$ μs to $t = 200$ μs. In the hydrophilic case, however, the droplet diameter increases from $1.75d_0$ to $1.93d_0$ in the same period. The final shape is hence wider and shorter in the hydrophilic case. We also see that the droplet shape is not strongly influenced by the wetting property of the substrates within the first 3 μs.

Experimental results of droplet landing [14] are in Fig. 14. These photos are side views so that the shallow valley at the center of the droplet at $t = 5$ μs cannot be seen. We see that, for the hydrophobic case, the droplet shape does not change between $t = 50$ μs and $t = 200$ μs. For the hydrophilic case, the diameter increases from $1.36d_0$ at $t = 50$ μs to $1.68d_0$ at $t = 200$ μs. Compared with experiments, our simulation results have a 15% larger diameter in both cases.

5.4. Droplet ejection and landing

For a third numerical example, we considered the ejection and landing of an ink droplet on a solid substrate. The substrate is located 0.8 mm from the nozzle exit, which is roughly the actual distance from the

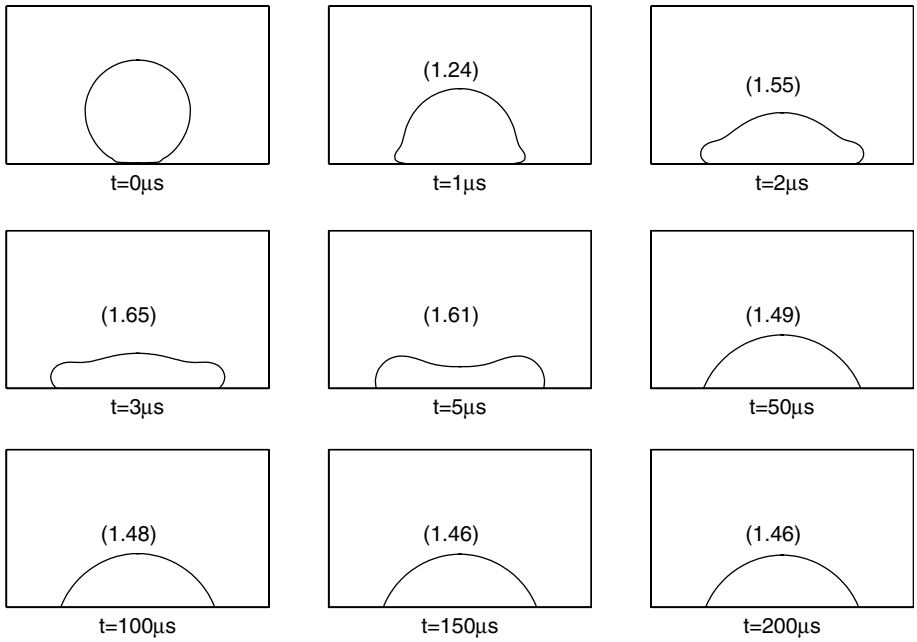

Fig. 12. The landing of an ink droplet on a hydrophobic substrate.

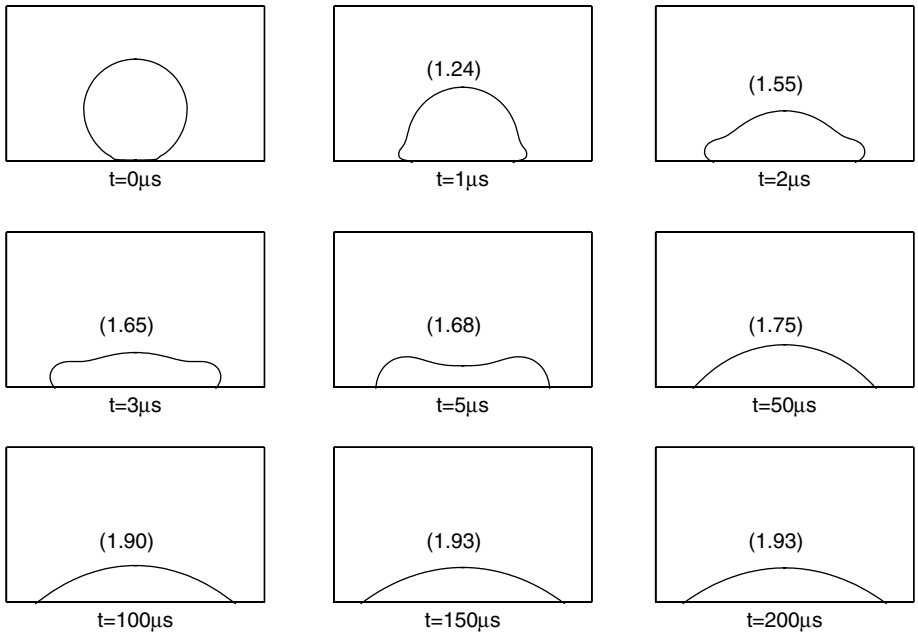

Fig. 13. The landing of an ink droplet on a hydrophilic substrate.

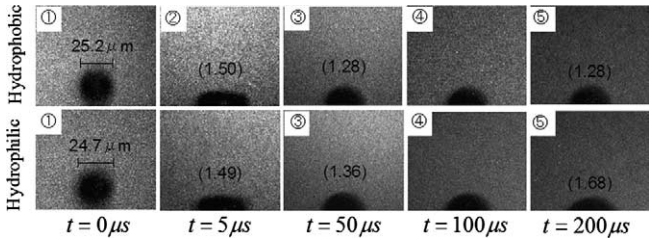

Fig. 14. The experimentally observed landing of ink droplets on two different substrates.

nozzle exit to the paper in an ink jet printer. We assumed that the material property of the substrate is the same as the nozzle wall, that is to say they have the same critical contact angles. A slightly smaller ink viscosity 3.5×10^{-3} kg/m s is used. The dynamic voltage is still like Fig. 5 except that the peak voltages are reduced to ±10 V so that the droplet velocity would be roughly 7 m/s. The voltage is kept at zero after 65 μs. Simulation results using a 128×3136 quadrilateral grid and a smearing of 3.5 cells ($\alpha = 3.5$) are plotted in Figs. 15–17. The simulation takes about 86,000 time steps to finish. On a Windows XP workstation with an Intel Xeon 2.8 GHz CPU and 1 GB 266 MHz ECC SDRAM, each time step takes about 9 s, for which 50% of the CPU time is spent on the finite element projection, 30% on the MAC projection, 7% on inverting the viscosity term.

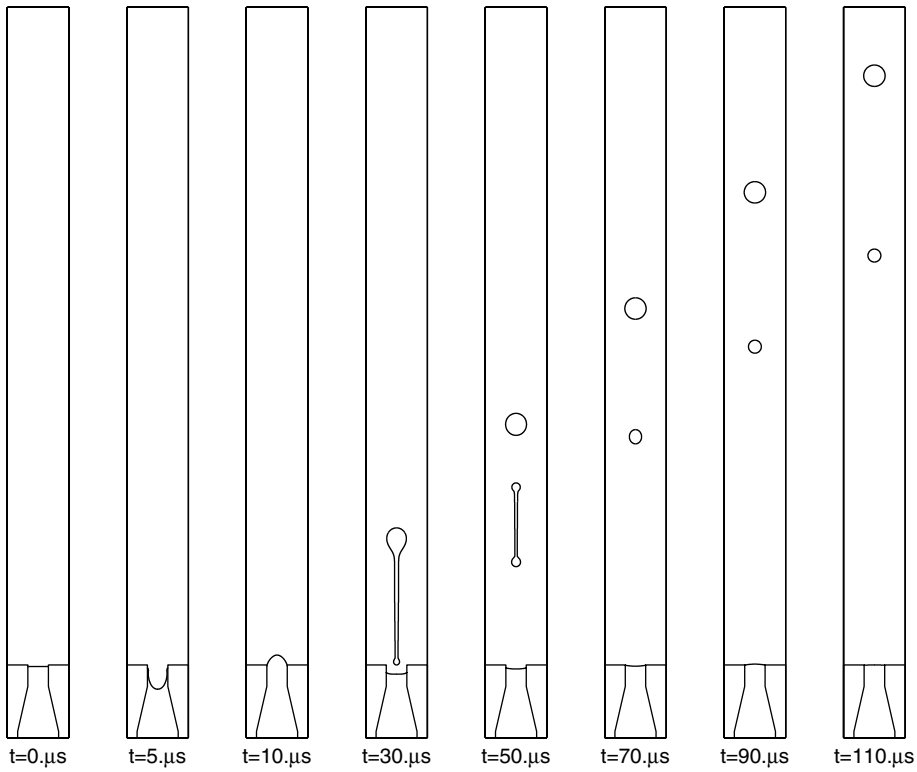

Fig. 15. An ink droplet fired toward an end substrate.

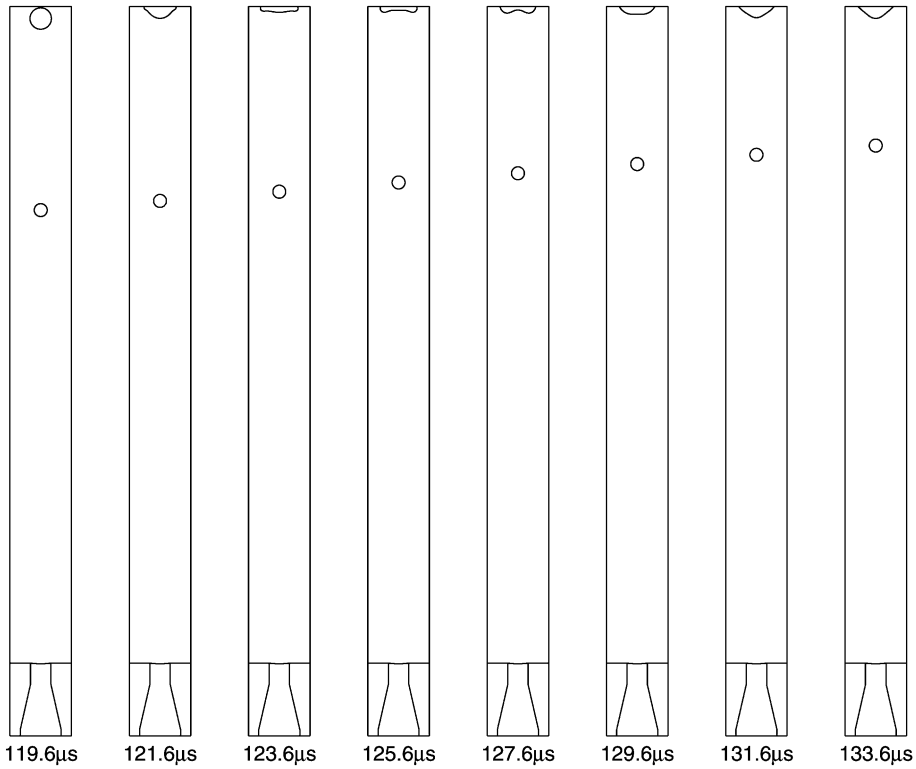

Fig. 16. Landing of the major ink droplet on the substrate.

We see that the droplet pinches off earlier than the previous case due to the slightly smaller ink viscosity and different peak voltage used in this simulation. The droplet tail is about to leave the nozzle exit at $t = 30$ μs. The long droplet separates into a bigger major droplet and a long satellite at about $t = 35$ μs (the moment is not shown), which is also earlier than the previous case. Due to the surface tension of ink, the satellite gradually changes to almost a spherical ball. The major droplet flies at 7.1 m/s while the satellite is slower at 5.6 m/s. The major droplet, which at the moment is 13.0 μm in radius (9.25 pico liters in volume), hits the end substrate at $t = 119.6$ μs. It splashes to the sides while some energy dissipates by viscosity and the rest is stored as surface tension potential. The latter part finally releases so that the center of the droplet rises, oscillates, and comes to a rest. The slower satellite, which is 7.87 μm in radius (2.04 pico liters) hits the major droplet on the substrate at about $t = 160$ μs. A tiny air bubble is formed during the merge.

To verify the mass conservation performance of the coupled level set projection method and the bicubic interpolation/fast marching re-initialization for such a lengthy simulation, we also calculate the droplet volume at different moments. At $t = 30$ μs, when the droplet tail is about to leave the nozzle, it is 11.083 pico liters. At $t = 119.6$ μs, when the major droplet is hitting the end substrate, the total volume (including the major droplet and the satellite) is 11.303 pico liters. The mass difference is 1.99%, which is reasonably good considering that the code completes more than 42,000 time steps and does more than 1000 level set re-initializations during this period.

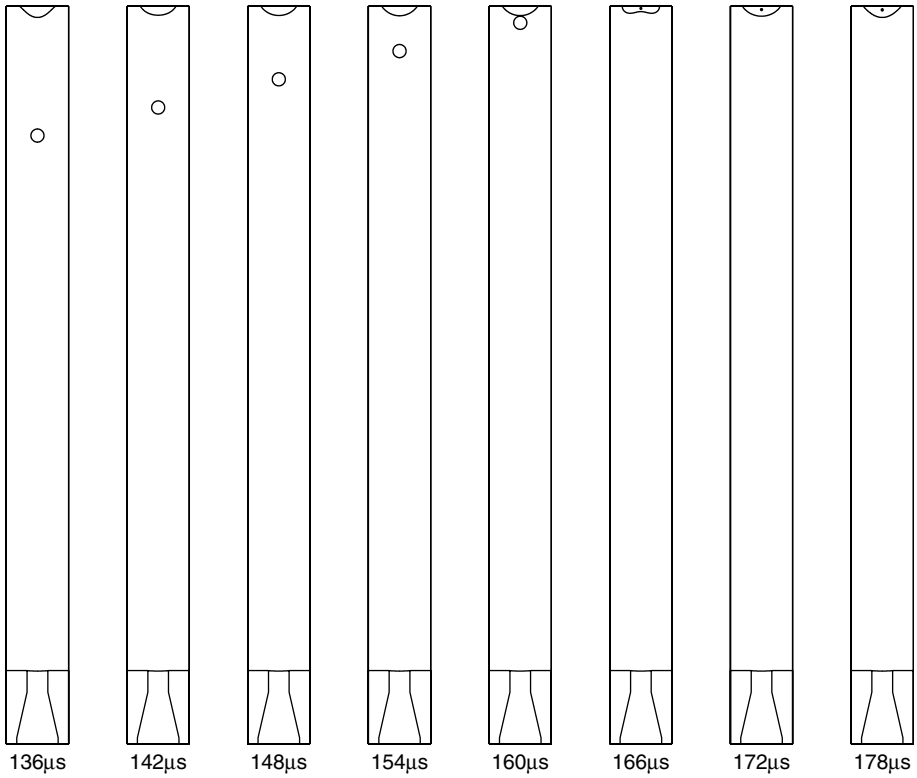

Fig. 17. The slower satellite droplet hits the major droplet on the substrate.

Finally it is interesting to see that the meniscus is not flat at the end of simulation $t = 178$ μs (see Fig. 18). The inflow pressure is $p = -160$ N/m^2. This negative pressure is balanced by the small surface tension so that the ink does not leak when the print head is not printing.

6. Analysis and future work

A coupled level set-projection method on body-fitted quadrilateral grids has been developed for piezoelectric ink jet simulations. To improve the mass conservation performance of the method, a bicubic interpolation is combined with the Fast Marching Method for level set re-initialization on quadrilateral grids. Numerical examples on code convergence, droplet ejection, and droplet landing are presented.

In all our simulations, we use a surface tension coefficient that has been statically measured, rather than dynamically. This leads to some error in our simulation. A dynamically determined coefficient will be larger, and will result in a shorter satellite tail; this will bring our simulation closer to experimental measurements. Nonetheless, our droplet ejection simulation is quite accurate, and is being used in engineering design.

Additional improvements will come from addressing two other issues. First, the nozzle wall does not have a perfect 90° corner at the opening. Thus, the nozzle geometry is under resolved unless a very large number of cells is used for the nozzle opening. Second, the slipping contact line model is not sophisticated

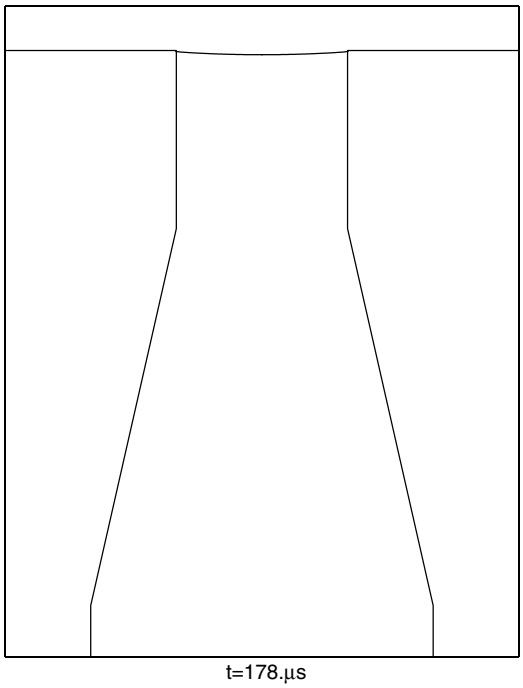

Fig. 18. The meniscus at the end of simulation.

enough. A better model should at least relate the slipping velocity to the difference between the contact angle and critical angles.

Acknowledgments

We would like to thank Dr. Ann Almgren, Dr. John Bell, and Prof. David Chopp for many valuable conversations. We also would like to thank Mr. Hashimoto, Mr. Ishida, and Dr. Shimoda for providing us the droplet landing photos.

References

[1] I.D. Aleinov, E.G. Puckett, M. Sussman, Formation of droplets in microscale jetting devices, in: Proceedings of ASME FEDSM'99, San Francisco, California, July 18–23, 1999.
[2] Ann S. Almgren, John B. Bell, William G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, SIAM J. Sci. Comput. 17 (2) (1996) 358–369.
[3] Ann S. Almgren, John B. Bell, Philip Colella, Tyler Marthaler, A Cartesian grid projection method for the incompressible Euler equations in complex geometries, SIAM J. Sci. Comput. 18 (5) (1997) 1289–1309.
[4] Ann S. Almgren, John B. Bell, William Y. Crutchfield, Approximate projection methods: Part I. Inviscid analysis, SIAM J. Sci. Comput. 22 (4) (2000) 1139–1159.
[5] John B. Bell, Jay M. Solomon, William G. Szymczak, A projection method for viscous incompressible flow on quadrilateral grids, AIAA J. 32 (10) (1994) 1961–1969.
[6] John B. Bell, Phillip Colella, Harland M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (2) (1989) 257–283.

[7] John B. Bell, Daniel L. Marcus, A second-order projection method for variable-density flows, J. Comput. Phys. 101 (1992) 334–348.
[8] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comput. Phys. 124 (1996) 449–464.
[9] David L. Chopp, Some improvements of the Fast Marching Method, SIAM J. Sci. Comput. 23 (1) (2001) 230–244.
[10] Alexandre J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[11] Phillip Colella, A direct Eulerian MUSCL scheme for gas dynamics, SIAM J. Sci. Stat. Comput. 6 (1) (1985) 104–117.
[12] Phillip Colella, Multidimensional upwind methods for hyperbolic conservation laws, SIAM J. Sci. Stat. Comput. 87 (1990) 171–200.
[13] M. Griebel, T. Dornseifer, T. Neunhoeffer, Numerical Simulation in Fluid Dynamics: A Practical Introduction, SIAM, 1998, pp. 45–49.
[14] T. Hashimoto, M. Ishida, T. Shimoda, Observation of ink droplet landing using high speed camera (in Japanese), Abstract of the 50th Spring Meeting of Japan Society of Applied Physics, 2003.
[15] R. Kimmel, James A. Sethian, Computing geodesic paths on manifolds, Proc. Nat. Acad. Sci. 95 (1998) 8431–8435.
[16] Stanley Osher, James A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[17] Elbridge G. Puckett, Ann S. Almgren, John B. Bell, Daniel L. Marcus, William J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[18] S. Sakai, Dynamics of piezoelectric inkjet printing systems, Proc. IS&T NIP 16 (2000) 15–20.
[19] James A. Sethian, An analysis of flame propagation, Ph.D. Dissertation, Department of Mathematics, University of California, Berkeley, CA, 1982.
[20] James A. Sethian, Curvature and the evolution of fronts, Commun. Math. Phys. 101 (1985) 487–499.
[21] James A. Sethian, in: P. Concus, R. Finn (Eds.), Numerical Methods for Propagating Fronts in Variational Methods for Free Surface Interfaces, Springer-Verlag, New York, 1987.
[22] James A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Nat. Acad. Sci. 93 (4) (1996) 1591–1595.
[23] James A. Sethian, Level set methods and Fast Marching Methods, second ed., Cambridge University Press, Cambridge. MA, 1999.
[24] James A. Sethian, A. Vladimirsky, Fast methods for the Eikonal and related Hamilton–Jacobi equations on unstructured meshes, Proc. Nat. Acad. Sci. 97 (May 23) (2000) 5699–5703.
[25] J.A. Sethian, A. Vladimirsky, Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms, SIAM Numer. Anal. 41 (1) (2003) 325–363.
[26] Akira Sou, Kosuke Sasai, Tsuyoshi Nakajima, Interface tracking simulation of ink jet formation by electrostatic force, in: Proceedings of ASME FEDSM'01, New Orleans, Louisiana, May 29–June 1, 2001.
[27] Mark Sussman, Peter Smereka, Stanley Osher, Axisymmetric free boundary problems, J. Comput. Phys. 114 (1994) 146–159.
[28] Mark Sussman, Peter Smereka, Axisymmetric free boundary problems, J. Fluid Mech. 341 (1997) 269–294.
[29] Mark Sussman, Ann S. Almgren, John B. Bell, Phillip Colella, Louis H. Howell, Michael L. Welcome, An adaptive level set approach for incompressible two-phase flow, J. Comput. Phys. 148 (1999) 81–124.
[30] J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Trans. Automatic Control 40 (1995) 1528–1538.
[31] David P. Trebotich, Phillip Colella, A projection method for incompressible viscous flow on moving quadrilateral grids, J. Comput. Phys. 166 (2001) 191–217.
[32] Jiun-Der Yu, Shinri Sakai, James A. Sethian, A coupled level set projection method applied to ink jet simulation, Interfaces and Free Boundaries 5 (2003) 459–482.
[33] Jingyi Zhu, James Sethian, Projection methods coupled to level set interface techniques, J. Comput. Phys. 102 (1992) 128–138.