Numerical computation of local vapor-liquid interface shape and heat transfer near steady contact line on heated surface

Huy Raymond Quach and Vladimir Ajaev

Southern Methodist University

Abstract. We develop efficient and robust numerical procedures for computing the local interface shape and heat flux near a steady contact line on a heated surface. The approach is based on reduction of the full system of coupled equations for liquid flow and heat transfer to a single ordinary differential equation for interface position. Results are obtained for a number of values of non-dimensional parameters. Common features of the solutions include high values of the evaporative mass flux and curvature close to the apparent contact line, as well as strong capillary pressure gradients that drive the local flow.

1. Introduction

Constrained vapor-liquid interfaces are encountered in microfluidic devices that use expanding vapor bubbles for actuation or pumping, as well as in a micro heat pipe. The latter is a closed device that can be used to remove the excess heat from working electronic devices by means of liquid vaporization and subsequent vapor flow away from the hot region. The cycle of evaporation and re-condensation at the vapor-liquid interface is continuous since the capillary pressure gradients pump liquid back into the hot region.

The key difficulties in mathematical modeling of constrained interfaces have to do with description of contact lines, i.e. the lines where the interface comes into contact with the constraining walls. The contact line model used in the present work is based on the ideas from the pioneering works of Derjaguin and Zorin [1], Potash and Wayner [2], and Moosman and Homsy [3]. They described contact lines as transition regions between macroscopic meniscii and miscroscopic adsorbed films. The latter is stable at a finite thickness due to action of van der Waals forces which effectively prevent complete dry-out of the heated solid surface. The value of the adsorbed film thickness can be computed based on equilibrium thermodynamics.

Numerical methods for describing evolution of vapor-liquid interfaces are rather well developed, as discussed e.g. in [4]. However, in order to apply these methods for studies of constrained vapor bubbles, one has to consider local physical effects near the contact line where the interface comes into contact with the solid wall. An important step in this direction has been taken in [5] where the shape of the interface near the contact line has been expressed in terms of solution of an ordinary differential equation. Since the focus of that study was on the details of the physical model, only sample computations of the interface shape for a narrow range of physical parameters have been discussed. The issues of optimization of the numerical method and its robustness for a wider range of parameters have not been discussed in sufficient detail. The efficient and robust methods for solving this equation are the subject of the present report. We note that while most of our discussion is based on [5], the numerical method is also applicable to other versions of the physical models describing vapor-liquid interface shapes near the contact line, e.g. the model developed by Morris [6].

2. Equation for interface

The mathematical model described in [5] is based on solving the Stokes flow equations coupled with heat transfer in the liquid. Interfacial boundary conditions include capillary pressure jump, zero shear stress, as well as mass and energy balances across the vapor-liquid interface. In addition, a non-equilibrium condition is formulated in [5] that relates local phase change temperature to the mass flux and pressure jump across the interface. All dynamical processes in the vapor are neglected. This system of equations and boundary conditions is rather difficult to solve numerically, so an asymptotic method was developed in [5] that takes advantage of the small value of the capillary number $C=\mu U/\sigma$, where μ is the liquid viscosity, U is the characteristic flow velocity, σ is the surface tension. As a result, the description of vapor-liquid interface near contact line is reduced to solving a boundary value problem for a single variable $\hat{h}$ in terms of local non-dimensional wall temperature T_w :

$$\frac{\delta\hat{h}_{\hat{x}\hat{x}} + \delta\varepsilon\hat{h}^{-3} - T_w + 1}{\hat{K} + \hat{h}} = \frac{1}{3}\left[\hat{h}^3(\hat{h}_{\hat{x}\hat{x}} + \varepsilon\hat{h}^{-3})_{\hat{x}}\right]_{\hat{x}}$$

$$\hat{h}_{\hat{x}\hat{x}} \to 1 + (\pi/4)^{1/2} \qquad \hat{x} \to \infty$$

$$\hat{h}_{\hat{x}}, \hat{h}_{\hat{x}\hat{x}} \to 0 \quad \hat{x} \to -\infty$$

$$\hat{h} \to \left(\frac{\delta\varepsilon}{T_w - 1}\right)^{1/3} \qquad \hat{x} \to -\infty$$

In order to solve this boundary value problem for interface position it is convenient to convert the fourth-order equation into a system of first-order differential equations. However, since the values of the parameters δ and ε are typically small, causing problems for numerical schemes, the values of $\hat{h}$ in [5] were first rescaled according to

$$\hat{h} = \frac{(\delta\varepsilon)^{\frac{1}{3}}}{(T_w - 1)^{\frac{1}{3}}} H$$

Also, x must be rescaled to δX .

In order to be able to solve the resulting equations, one has to find nonzero values to use for derivatives on the left boundary, since $\hat{h}$ approaches a known constant, h_{af} , as x approaches negative infinity [5]. Let us re-write the rescaled variable H as $1 + H_1$, H_1 being very small. After substituting this expression for H into the equation, we find an approximate solution for H_1 that gives expression from which the two conditions for derivatives can be derived. Once the appropriate substitution is made and the equation is linearized, the following result is obtained for the small correction to scaled thickness H_1 as a function of the scaled variable X **:**

$$\frac{H_{1xx}}{\delta} - \frac{3H_1(T_w - 1)^{\frac{4}{3}}}{(\delta\varepsilon)^{\frac{1}{3}}} = \frac{\varepsilon K + 3\delta^{-\frac{2}{3}}\varepsilon^{\frac{1}{3}}}{3(T_w - 1)} \times \left[\frac{H_{1xx}}{\delta} - \frac{3H_1(T_w - 1)^{\frac{4}{3}}}{(\delta\varepsilon)^{\frac{1}{3}}} \right]_{xx}$$

Solve this linearized ODE for H_1 gives the necessary information about behavior of derivatives at negative infinity. Using these boundary conditions, and the rescaled variable H , Ajaev & Homsy [5] converted the differential equation into a system of first-order equations and use numerical methods to solve the equation. However, despite this rescaling and significant amount of preliminary analytical work associated with imposing proper boundary conditions, the resulting Fortran code was not sufficiently robust to obtain results over all practically relevant values of non-dimensional parameters.

Therefore, in this report we take a different approach by attempting to solve the original boundary value problem without any rescaling. In *Matlab*, we are able to find a numerical solution to the equation without rescaling $\hat{h}$, through use of the boundary value solver function, *bvp4c*. Prior to inputting the equation into the solver, all we need to do is to convert it into a system of first-order equations. We introduce

$$y_1 = h', \quad y_2 = h'', \quad y_3 = h''', \quad y_4 = h''''$$

and then write the system in terms of new variables:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= y_4 \end{aligned}$$

$$y_4' = \frac{3}{y_1^{\,3}}\left[\frac{\delta y_3 + \delta\varepsilon y_1^{\,-3} - T_w + 1}{K + y_1} - y_1^{\,2} y_2 y_4 - \frac{\varepsilon y_2^2}{y_1^{\,2}} - \frac{\varepsilon y_3}{y_1}\right]$$

In the code we use the following notation:

δ = *lam*
ε = *ep*

We also note that the asymptotic analysis of the equation for *h* close to the value of the adsorbed film thickness shows the behavior in the form

$$h \sim h_{af}(1 + \zeta e^{\mu x} + \zeta C e^{\nu x})$$

where

$$\mu = \left(\frac{3}{h_{af}(K + h_{af})}\right)^{1/2} \qquad \nu = \frac{(3\varepsilon)^{1/2}}{h_{af}^2}$$

and the values of the parameter C is found from the numerical solution based on the boundary conditions at the left end of the computational domain. The value of ζ is fixed for a given run, but we vary it to verify that the solution is independent of it: the only change in the solution is shift in the x direction, which gives the same physical result since the original physical problem has translational invariance. However, we note that of ζ has to be small for the asymptotic analysis to be valid; in most simulations below we choose ζ=0.01.

3. Numerical results and discussion

To input this into the solver, we create a function function that accepts an array of expressions, with each value in the array representing an expression in the system of equations. The first value in the array is the expression for y_1' (which should be y_2), the next value y_2' and so forth. Next, we create another function function that accepts an array of expressions for the boundary values. We use *ya* to represent the left bound and *yb* for the right bound. For example, to represent h approaching h_{af} on the left side we would input *ya – haf*. Finally, we need an initial guess for the solution of each variable in the system of equations and a mesh on which the function will calculate the solution. Both of these have a fairly significant impact upon the solution calculated by the solver. In order to output the solution, the most efficient way is to plot the solution calculated by the solver onto a graph. The solver outputs a numerical solution to the original variable $\hat{h}$**.** The parameter C is determined as part of the solution, so the total number of boundary conditions for this fourth-order equation is five.

In Figure 1 we have plotted the solution calculated by *bvp4c* using *[x^2+h_{af}; 2*x; 2; 0]* for the guess values, *0 to 2.5* as the interval. The values of the physical parameters are identical to those in [5]. Fig 1a shows the profiles of the interface and its second derivative; fig1b illustrates scaled mass flux versus the coordinate. The solution for $\hat{h}$ is identical to the one arrived by Ajaev and Homsy [5] on page 186. In order to show that this is indeed the physical solution we have checked that it is independent from the interval length and the parameter ζ .

a. b.

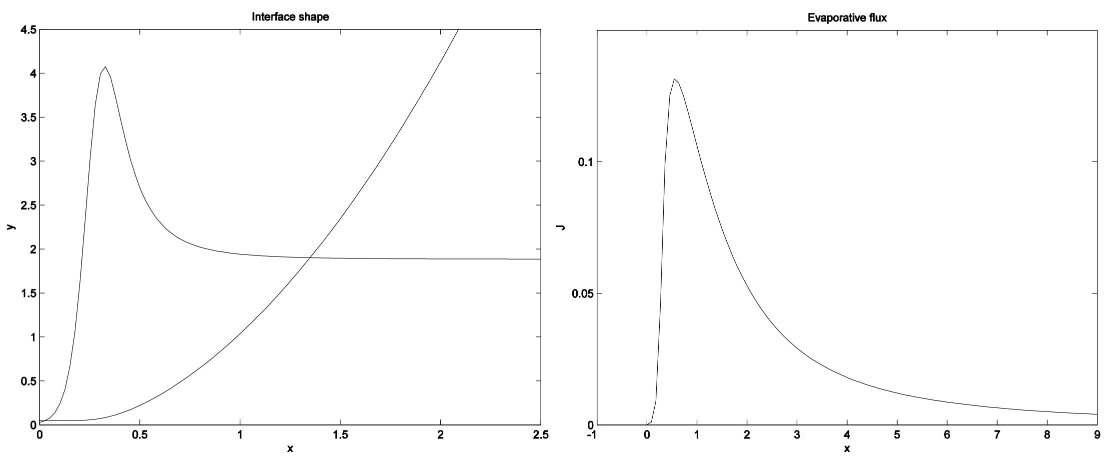

FIG.1. Solutions for interface profile and mass flux corresponding to [5]

The shooting method used in [5] to find the solutions illustrated in Fig. 1 above fails to give accurate results for the values of ε below .001, which makes it impossible to describe local interface shapes for a number of practically relevant situations that involve small values of the scaled adsorbed film thickness. The present numerical approach is free from this limitation, as illustrated in Fig. 2. It shows the scaled flux for small values of ε and δ motivated by practically relevant physical parameters. Qualitative behavior of the flux is essentially the same as in Fig. 1b, but the maximum value, which characterizes the intensity of evaporation, is significantly higher.

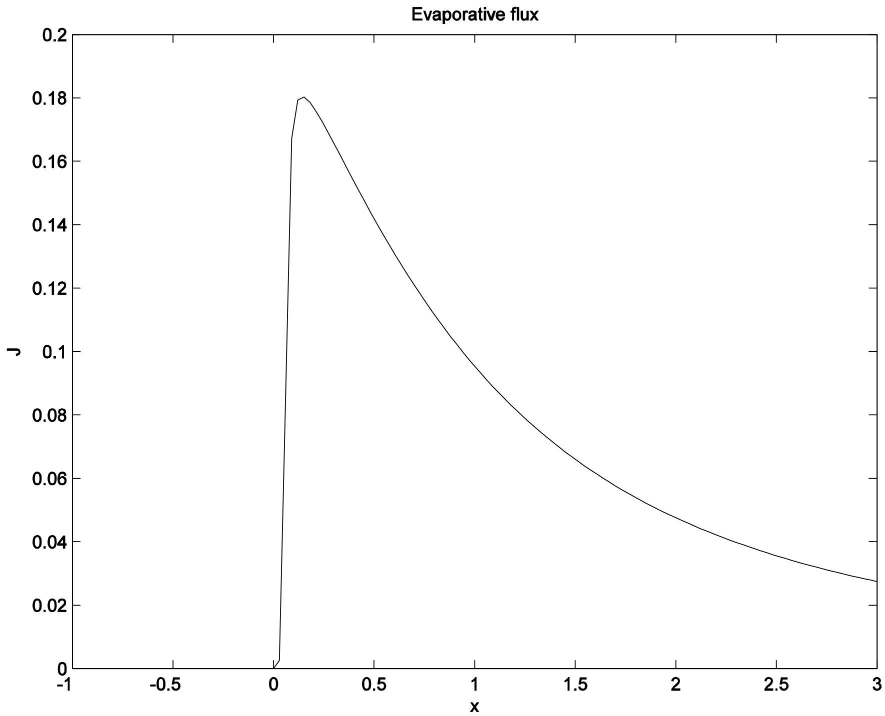

FIG 2. Solution for scaled evaporative flux at δ=0.005, ε=0.000025, and all other physical parameters same as in [5].

4. Conclusions

The use of *bvp4c* in *Matlab* is a robust method for finding numerical solutions to boundary value problems. This function allows us to specify a mesh and interval in order to compute the solution according to our needs. Furthermore, this flexibility allows for experimenting to see which solution is the best. Most importantly, the function is quick, accurate, and relatively simple to code, making it an efficient approach to numerical problems. Important features of the solution include rapid change of curvature and a well-defined maximum of the evaporative flux in the region of the vapor-liquid interface near the adsorbed film.

References

1. B.V. Derjaguin, A.M.Zorin, Proc. 2nd Int. Congress Surf. Activity, vol 2, 145 (1957)
2. M. Potash and P.C.Wayner, *Int. J. Heat Mass Transfer* **15**, 1851 (1972)
3. S.Moosman and G.M.Homsy, *J. Coll. Interface Sci.* **73,** 212 (1980)
4. A. Mukherjee and S.G. Kandlikar, *Microfluidics Nanofluidics* **1,** 137 (2005)
5. V.S.Ajaev, G.M.Homsy , *J. Coll. Interface Sci.* **244,** 180 (2001)
6. S.J.S.Morris, *J. Fluid Mech.* **432,** 1 (2001)

Appendix

Matlab code

```
function mybvp

st = cputime;

global Tw K lam ep haf
Tw = 1.4;
K = 2.0;
lam = 10^(-2);
ep = 10^(-4);
haf = ((lam*ep)/(Tw - 1))^(1/3);

xmin = 0;
xmax = 2.5;
pc=0.01;

solinit = bvpinit(linspace(xmin,xmax,100),@myinit,pc);
sol = bvp4c(@myode,@mybc,solinit);

xint = linspace(xmin,xmax);
Sxint = deval(sol,xint);
figure(1);
xint = linspace(0,2.5);
plot(xint,[Sxint(1,:);Sxint(3,:)]);
axis([0 2.5 0 4.5]);
title('Graph');
xlabel('x');
ylabel('h');
xint = linspace(0,10);
h = Sxint(1,:);
hxx = Sxint(3,:);
Jint = (-lam.*hxx - lam.*ep.*(h.^(-3)) + Tw - 1)./(K + h);
figure(2); plot(xint,Jint); axis([-1 9 0 0.15]); ylabel('J');
cputime - st

% ------------------------------------------------------------------------

function dydx = myode(x,y,pc)
global Tw K lam ep haf
T = Tw - 1;
T1 = 1/T^(1/3);
dydx = [        y(2)
                y(3)
```

```
               y(4)
               (3/(y(1)^3))*(((lam*y(3) + lam*ep*y(1)^(-3) - Tw + 1)/(K + y(1))) -
y(1)^2*y(2)*(y(4)-((3*ep*y(2))/y(1)^4)) - ((4*ep*y(2)*y(2))/y(1)^2) + (ep*y(3)/y(1)))];

% --------------------------------------------------------------------------

function res = mybc(ya,yb,pc)
global Tw K lam ep haf
ko = 1 + (pi/4)^(1/2);
mu=(3./((K+haf)*haf^3))^(1/2);
nu=(1./(haf^2))*(3.*ep)^(1/2);
zeta=0.01;
res = [ ya(1)-haf-pc*haf-haf*zeta; ya(2)-mu*pc*haf-nu*haf*zeta; ya(3)-mu^2*pc*haf-
nu^2*haf*zeta; ya(4)- mu^3*pc*haf-nu^3*haf*zeta;  yb(3)-ko ];

% --------------------------------------------------------------------------

function yinit = myinit(x)
global Tw K lam ep haf
yinit = [   x^2+haf; 2*x; 2; 0  ];
```