Pressure-Driven Spatiotemporal Control of the Laminar Flow Interface in a Microfluidic Network

Supplementary Materials

Brandon Kuczenski, Philip R. LeDuc*, and William C. Messner*

Department of Mechanical Engineering

Carnegie Mellon University

February 7, 2007

* Corresponding Authors: prleduc@cmu.edu, bmessner@andrew.cmu.edu

Modeling Fluid Interface Position

Positioning the fluid interface requires a model which relates hydrostatic pressure to interface position. We show that interface position depends only on two dimensionless parameters. The first parameter describes the macroscopic characteristics of the microfluidic network, and the second represents the instantaneous difference between the two forcing pressures.

Under the assumptions of fluid and channel incompressibility, the position of the interface between two fluids along the channel's width is dependent on the instantaneous rates of fluid flow into the inlets. Flow rate and forcing pressure can be related via the equations governing laminar flow through an enclosed pipe. For a microfluidic network having uniform cross-sectional geometry, with two confluent input branches, each of length L_{in}, which converge to form one output branch of length L_{out}, the forcing pressures can be related to the flow rates as follows:

$$p_1 \quad = \quad cQ_1L_{in} + c(Q_1 + Q_2)L_{out}$$

$$p_2 \quad = \quad cQ_2L_{in} + c(Q_1 + Q_2)L_{out} \tag{1}$$

where p_1 and p_2 are the forcing pressures measured relative to the pressure at the outlet of the microfluidic device (generally atmospheric pressure); Q_1 and Q_2 are the rates of fluid flow through the two inlet channels; and c is a constant representing physical properties of the fluid and the channels. Equations 1 can be recombined to express the ratio of inlet flows ϕ in terms of the forcing pressures and channel geometry.

$$\phi = \frac{Q_1}{Q_2} = \frac{p_1L_{in} + (p_1 - p_2)L_{out}}{p_2L_{in} + (p_2 - p_1)L_{out}} \tag{2}$$

The interface position can be determined from ϕ via a model for the velocity profile across the channel. For a given interface position, Q_1 and Q_2 can be obtained by dividing the channel cross-section into two regions on either side of the interface and integrating the velocity profile across each region. The velocity profile model we used was a numerical approximation to the exact solution to the Poisson equation for forced laminar flow through a rectangular pipe.[1]

When flow through either inlet channel becomes zero, the ratio ϕ becomes zero or infinity. This occurs when the ratio of pressures reaches a critical value, namely when $p_1/p_2 = (L_{in} + L_{out})/L_{out}$ or $p_1/p_2 = L_{out}/(L_{in} + L_{out})$. Physically, these conditions correspond to the forcing pressure at one inlet equalling the pressure at the point of confluence, halting flow into that inlet. If the pressure ratio exceeds these values, the result is backflow up one channel or the other.

By choosing the relative lengths of the inlet and outlet channels, the critical values can be adjusted, changing the sensitivity of the interface position to reservoir pressure. Lengthening the inlet channels relative to the outlet channel increases robustness of the interface position to regulator disturbances at the expense of requiring a larger range of operating pressure.

Rewriting Equations 1 and 2 in terms of Π produces the following result:

$$\Pi \triangleq \frac{p_1 - p_2}{p_1 + p_2} = \left(\frac{\phi - 1}{\phi + 1}\right) \frac{L_{in}}{L_{in} + 2L_{out}} \tag{3}$$

This relates interface position to macroscopic channel geometry and to a dimensionless parameter

Π, the normalized difference in forcing pressures between the two channels. For a fixed geometry, ϕ depends only on Π and manifests no time dependence. Thus the interface position at any instant depends only on Π according to this theory. This model was used during experimentation to determine the predicted interface position, given pressure sensor readings and channel geometry. Our observations were in agreement with the theory.

Determining Interface Position from Fluid Flow Ratios

The formulation given in Equation 3 does not provide a direct method for computing interface position, only the ratio of fluid flows, ϕ. The interface position must be determined from fluid flow ratios via integration of the velocity profile. In the simple case of a parabolic flow profile, the interface position can be found by computing the root of a cubic polynomial; however, in the case that a more accurate velocity profile is used, such as the approximation to the exact solution,[1] numerical techniques must be employed to determine the interface position.

Our method employed a numerical implementation of this solution in Matlab®. We have included the source code for two functions, `zphi.m` and `zphi_inv.m`, which perform these computations.

The function `zphi`

`zphi.m` computes the volume under the surface described by the numerical solution to the velocity profile for flow through a rectangular pipe. It accepts two arguments, a vector of interface positions between -1 and 1, where -1 corresponds to the 'left' wall of the channel and 1 corresponds to the 'right' wall. It also accepts an optional aspect ratio of channel width over height. If the aspect ratio is omitted, a default value of 1 is assumed.

The output of `zphi` is a vector of fractional volumes that lie between the left wall of the channel and each of the interface positions supplied.

```
% Example of usage -- zphi.m
>> zphi(0) % volume fraction left of channel centerline
ans =
    0.5000
```

```
>> zphi([-0.3 0.3 0.6]) % usage with multiple interface positions
ans =
    0.2893    0.7107    0.8883
>> zphi([-0.3 0.3 0.6] , 10) % aspect ratio for a wide channel
ans =
    0.3399    0.6601    0.8201
>>
```

The function zphi_inv

zphi_inv.m uses the Matlab function fzero to find zero crossings in order to invert the function zphi. It also accepts two arguments: a vector of volumetric flows and an optional aspect ratio. The vector of volumetric flows should be a sequence of volumes occupied by parallel laminar streams, progressing from the left to the right edge of the channel. Below, the function is used to invert the calculations shown above:

```
>> zphi_inv([ 1 1]) % equal flows
ans =
    -2.0073e-16
>> % that is nearly zero
>> zphi_inv([ .2893,  .7107-.2893,   .8883-.7107,   1-.8883])
ans =
   -0.3000    0.3000    0.5999
>> zphi_inv([ .3399,  .6601-.3399,  .8201-.6601,  1-.8201] , 10)
ans =
   -0.3000    0.3000    0.5999
>>
```

Code Listings

```
function phi=zphi(x,as,n)
% function phi=zphi(x,as)
% this function evaluates an approximation of the normalized volume
% under the curve that represents the velocity profile of a laminar
% flow stream in a rectangular channel.
%
% The user supplies a boundary position, or a vector of positions,
% and the function computes the fraction of area 'left' of that
% position.
%
% x=-1 corresponds to the 'left' edge of the channel; x=1 corresponds
% to the right edge.  The function returns the area between -1 and
% the x supplied by the user.
%
% For incompressible laminar flow of fluids with equal densities and
% viscosities, the boundary can represent the laminar flow interface
% between the two streams, and the fraction of areas under this
% curve is equal to the ratio of the volumetric flow rate in the
% 'left'-hand stream to the total flow rate.
%
% For channel aspect ratios less than 1 (width over height), the
% velocity profile along the width dimension is approximately
% parabolic, and a parabolic approximation is used.  This is also
% used when the aspect ratio is unspecified.
%
% For wide-aspect-ratio channels, the function computes a 2-term
% approximation of the exact [hyperbolic] solution for the
% velocity profile, as solved in Brody 1996.  Because the
% terms have leading factors of (2*n+1)^-4, successive terms
% are negligible after n>1 for small aspect ratios.  If the
% function is called with a third argument N, the function will
% compute N terms.
%
% Works for vector x but only scalar as.


if nargin<3
  n=2;
end

if nargin<2
  as=1;
end

x(find(x<-1))=-1;
x(find(x>1))=1;

if isempty(as) | as<1
  phi= (3/4)*(x+1) - (1/4)*(x.^3 + 1);
else
  p=zeros(size(x));
  P=zeros(size(x));
  for i=0:n
    t=2*i+1;
    p=p + (t^-4)*(x+1 - 2*(sinh(t*pi*as*x/2) + sinh(t*pi*as/2)) ...
                / (pi*as*t*cosh(t*pi*as/2)));
    P=P + 2*(t^-4)*(1 - 2*(tanh(t*pi*as/2)) / (pi*as*t) );
  end
  phi=p./P;
end
```

```
function x=zphi_inv(phi,as,n)
% function x=zphi_inv(phi,as)
% this function performs the inverse computation of zphi.m.  Where zphi
% requests a boundary position and provides the fraction of volume on
% one side of the interface, zphi_inv expects a vector of volumetric
% flow rates and provides a series of boundary positions.
%
% The user can also supply a channel aspect ratio, but if that is not
% supplied the default value is 1.
%
% If the user supplies a third argument N, then the numerical
% approximation will include N terms (default is 2, which is accurate
% to about 0.1%).
%
% This function simply wraps fzero around zphi.
%

if nargin<3
  n=2;
end
if nargin<2
  as=1;
end

phi=phi/sum(phi); %normalize

x=-1;
for i=1:length(phi)-1
  x(i+1)=fzero(@(t) zphi(t,as,n)-sum(phi(1:i)),x(i));
end
if length(x)>1
  x=x(2:end);
else
  x=1;
end
```

References

[1] J. P. Brody, P. Yager, R. E. Goldstein, and R. H. Austin, *Biophys. J.*, 1996, **71**, 3430–3441.