

# ELECTRONIC SUPPLEMENTARY INFORMATION



$$Error = \frac{C_{num} - C_{anal}}{C_{anal}}$$

Supplementary Material (ESI) for Lab on a Chip This journal is © The Royal Society of Chemistry 2007



Supplemental Figure 2. Numerical simulation of transport phenomena along analog microchannel after 1 minute of focusing. 0.1mM fluorescein-Na in 900mM Tris-Borate buffer is simulated. (a) Geometrical profile of microchannel (L = 16mm, d = 18µm). (b) Electric potential profile: 1200V applied at x = 0, x=L grounded. (c) Pressure profile: P = 0 assumed at both ends. (d) Temperature profile due to Joule heating:  $T = 25^{\circ}$ C assumed at both ends. (e) Resulting velocity profiles of: bulk fluid,  $u_{bulk} = u_{eof} + u_{pdf}$  (dashed line),  $u_{ep}$  (dashed-dot line) and  $u_{net} = u_{bulk} + u_{ep}$  (solid line). (f) Normalized concentration profile.

#### **Further Details of Numerical Simulation**

The following details the steps used in formulating the Matlab numerical model used in this work. The model is a fundamental transient finite difference scheme solving the governing equations given in the manuscript. The simulations were conducted using a graphical user interface, which can be seen in Supplemental Figure 3.

### Calculating $\phi_i$

After establishing the flow parameters, boundary values, and initial conditions, the code first solves for the constant current, *I*, through the channel. *I* is found iteratively using the *fzero* command. The function calculates  $\phi$  at each point, then compares  $\phi$  (N) with the specified  $\phi$  (L). The function *fzero* finds the point at which the error = zero. We make an initial guess of:

$$I_{guess} = \frac{L[\phi(L) - \phi(0)]}{\sum_{i=1}^{N} \frac{1}{\sigma_i A_i}}$$

After the program has converged on a solution for I, equation (12) can be used to solve for the change in potential at each point:

$$\frac{d\phi_i}{dx} = \frac{I}{\sigma_i A_i}$$

The potential at each point along the channel is then calculated using:

$$\phi_i = \phi_{i-1} + \frac{d\phi_{i-1}}{dx}$$

Note: In calculating first and second derivatives at point i, a central finite difference scheme was used throughout the model, i.e.

$$\frac{d\phi_i}{dx} = \frac{(\phi_{i+1} - \phi_{i-1})}{2dx} \qquad \qquad \frac{d^2\phi_i}{dx^2} = \frac{(\phi_{i+1} - 2\phi_i + \phi_{i-1})}{(dx)^2}$$

#### Calculating P<sub>i</sub>

The next step is to solve for the constant mass flow rate, uA, through the channel. Equation (2) is calculated at the channel inlet to provide an initial guess for  $u_{bulk}$ . The *fzero* command is used to iteratively solve for the mass flow rate using equation (1). The function solves for *P* at each point, and runs until the error between P(N) and the imposed P(L) is zero. Once the mass flux is known, the bulk velocity at each point,  $u_{bulk,i}$ , can be easily calculated.

Equation (2) is then used to solve for the pressure gradient at each point.

Supplementary Material (ESI) for Lab on a Chip This journal is © The Royal Society of Chemistry 2007

$$dP_i = \frac{-32\eta_i dx}{d_{h,i}^2} \left( u_{bulk,i} - \left(\frac{\varepsilon_0 \varepsilon_i \zeta_i}{\eta_i}\right) \frac{d\phi_i}{dx} \right)$$

The pressure at each point is calculated as

$$P_i = P_{i-1} + \frac{dP_{i-1}}{dx}$$

## Calculating $T_i$ and $c_i$

To calculate the temperature and concentration at each point in the channel, we used the Crank-Nicolson scheme, an inherently stable implicit formulation. Equations (7) and (13) are each discretized and solved at each time step. As an example of this method, we will walk through the process for solving for the concentration,  $c_i$ .

$$A_{i}\left(\frac{c_{i}^{n+1}-c_{i}^{n}}{dt}\right) = A_{i}D_{eff,i}\left(\frac{c_{i+1}-2c_{i}+c_{i-1}}{dx^{2}}\right) + \left(A_{i}\frac{dD_{eff,i}}{dx} + D_{eff,i}\frac{dA_{i}}{dx} + A_{i}\mu_{ep,i}\frac{d\phi_{i}}{dx} - A_{i}u_{bulk,i}\right)\left(\frac{c_{i+1}-c_{i-1}}{2dx}\right) + \cdots + \left(A_{i}\frac{d\mu_{ep,i}}{dx}\frac{d\phi_{i}}{dx} + \mu_{ep,i}\frac{dA_{i}}{dx}\frac{d\phi_{i}}{dx} + A_{i}\mu_{ep,i}\frac{d^{2}\phi_{i}}{dx^{2}} - A_{i}\frac{du_{bulk,i}}{dx} - u_{bulk,i}\frac{dA_{i}}{dx}\right)c_{i}$$

The Crank-Nicolson scheme calculates the right-hand side as the average of the concentration changes from *time* = n to *time* = n+1. For simplicity, we will denote:

$$\alpha_{i} = \frac{D_{eff,i}dt}{2dx^{2}}$$

$$\beta_{i} = \left(A_{i}\frac{dD_{eff,i}}{dx} + D_{eff,i}\frac{dA_{i}}{dx} + A_{i}\mu_{ep,i}\frac{d\phi_{i}}{dx} - A_{i}\mu_{bulk,i}\right)\frac{dt}{4A_{i}dx}$$

$$\gamma_{i} = \left(A_{i}\frac{d\mu_{ep,i}}{dx}\frac{d\phi_{i}}{dx} + \mu_{ep,i}\frac{dA_{i}}{dx}\frac{d\phi_{i}}{dx} + A_{i}\mu_{ep,i}\frac{d^{2}\phi_{i}}{dx^{2}} - A_{i}\frac{du_{bulk,i}}{dx} - u_{bulk,i}\frac{dA_{i}}{dx}\right)\frac{dt}{2A_{i}}$$

The formulation then becomes:

$$c_{i}^{n+1} - c_{i}^{n} = \alpha_{i} \left[ \left( c_{i-1}^{n+1} - 2c_{i}^{n+1} + c_{i+1}^{n+1} \right) + \left( c_{i-1}^{n} - 2c_{i}^{n} + c_{i+1}^{n} \right) \right] + \beta_{i} \left[ \left( c_{i+1}^{n+1} - c_{i-1}^{n+1} \right) + \left( c_{i+1}^{n} - c_{i-1}^{n} \right) \right] + \gamma_{i} \left( c_{i}^{n+1} + c_{i}^{n} \right)$$
  
Or,

$$c_{i-1}^{n+1}(-\alpha_i + \beta_i) + c_i^{n+1}(1 + 2\alpha_i - \gamma_i) + c_{i+1}^{n+1}(-\alpha_i - \beta_i) = c_{i-1}^n(\alpha_i - \beta_i) + c_i^n(1 - 2\alpha_i + \gamma_i) + c_{i+1}^n(\alpha_i + \beta_i)$$

# Supplementary Material (ESI) for Lab on a Chip This journal is © The Royal Society of Chemistry 2007

The left-hand side of this equation is represented as a product of a vector and tridiagonal matrix. The right hand side is kept as a vector. The formulation is then written in matrix form as:

$$\begin{bmatrix} 1+2\alpha_{1}-\gamma_{1} & -\alpha_{1}-\beta_{1} & 0 & \cdots & 0\\ -\alpha_{2}+\beta_{2} & 1+2\alpha_{2}-\gamma_{2} & -\alpha_{2}-\beta_{2} & 0 & 0\\ 0 & -\alpha_{3}+\beta_{3} & 1+2\alpha_{3}-\gamma_{3} & \ddots & 0\\ \vdots & 0 & \ddots & \ddots & -\alpha_{N-1}-\beta_{N-1}\\ 0 & 0 & 0 & -\alpha_{N}+\beta_{N} & 1+2\alpha_{N}-\gamma_{N} \end{bmatrix} \begin{bmatrix} c_{1}^{n+1}\\ c_{2}^{n}\\ c_{2}^{n+1}\\ \vdots\\ c_{n}^{n+1}\\ c_{n}^{n+1}\\ \vdots\\ c_{n}^{n+1}\\ c_{n}^{n+1}\\ c_{n}^{n}\\ c_{n}^{n+1}\\ c_{n}^{n}\\ c_{n}^{n}$$

The model first calculates the  $\alpha$ ,  $\beta$ , and  $\gamma$  values at each point, then establishes the tridiagonal matrix using the *spdiags* function. The next step is to establish the right-hand side vector using the known concentration values at the current time step, n. The concentration at the subsequent time step, n+1, is then calculated as:

$$c^{n+1} = \mathbf{R} \cdot A^{-1}$$

In order to smooth out the oscillations in the transient solutions, the following moving average was employed:

$$c_i^{n+1} = (2c_i^n + 2c_i^{n+1})/4$$

The time step, *dt*, is calculated using the following stability recommendation.

$$dt \cong \frac{N(dx)^2}{2D_{eff,\max}}$$

The same basic structure is used to calculate the temperature using equation (13).

Supplementary Material (ESI) for Lab on a Chip This journal is © The Royal Society of Chemistry 2007



Supplemental Figure 3. Screen shot of Matlab graphical user interface (GUI) used to run simulations.