Modeling Electrokinetic Injection in a Micronit X8050 Cross Microchannel Using COMSOL Multiphysics

Charles Zhou

December 18, 2025

Abstract

This report presents a detailed multiphysics model of electrokinetic injection (EKI) in a Micronit X8050 glass microfluidic cross channel using COMSOL Multiphysics. A two-dimensional depth-averaged representation of the Micronit X8050 geometry is constructed and used to simulate the pinching and dispensing steps of a standard electrokinetic injection protocol for a fluorescein tracer. The model couples Electric Currents (EC), Creeping Flow (Stokes, SPF), and Transport of Diluted Species (TDS). Stationary studies compute electric fields and electroosmotic velocities for each voltage configuration, followed by time-dependent transport simulations. The model captures direction-dependent flow during pinching and dispensing, which is critical for defining inlet/outlet species conditions and for using voltage ratios reported in the literature.

Simulation results demonstrate that plug velocity is determined jointly by EOF, electrophoresis, and convective contributions, while plug broadening arises from a combination of diffusion and field-induced stretching. This modeling framework also provides a foundation for future extraction of unknown parameters such as wall zeta potential, pressure differences, and effective diffusivity by matching COMSOL results to experiment in subsequent work.

1 Introduction

Electrokinetic injection (EKI) is a widely used method for introducing analyte plugs into microfluidic channels without mechanical valves or moving parts. By applying controlled electric fields across a cross-shaped microchannel network, a narrow sample zone can be pinched, injected, and transported downstream for subsequent electrophoretic separation or analysis. The technique is central to capillary electrophoresis (CE) and many microchip electrophoresis platforms.

In the Micronit X8050 glass microchannel used in this work, three short channels intersect a single long channel, forming a symmetric cross. The injection procedure consists of:

- A **pinching step**, where high transverse fields compress the sample stream into a narrow core.
- A **dispensing step**, where fields are reconfigured to pull the sample into the long channel.

Understanding the interplay of electroosmosis, electrophoresis, convection, and diffusion during this process is essential for producing a well-defined injected plug and for interpreting downstream fluorescence data. In this work, COMSOL Multiphysics is used to model the coupled electric fields, electroosmotic flow, and Nernst–Planck species transport during a canonical pinching–dispensing sequence in the Micronit X8050 cross channel driven by prescribed voltage waveforms.

The immediate goal of the model is to resolve how these transport mechanisms combine to set the shape, velocity, and broadening of a fluorescein plug in the vicinity of the cross intersection. The present report focuses on simulation results in a depth-averaged two-dimensional representation of the device. The model is intended to serve as a foundation for quantitative comparison with fluorescein injection experiments in the Pennathur Lab, in order to extract key parameters such as zeta potential, pressure and diffusivity.

Within this scope, the specific objectives are:

1. Construct a coupled Electric Currents–Creeping Flow–Transport of Diluted Species model of the Micronit X8050 geometry with electroosmotic slip at the walls.

2. Simulate the electric potential, electroosmotic velocity field, and fluorescein concentration during the pinching and dispensing steps for a fixed set of LabSmith HVS voltage waveforms.

3. Extract simple plug metrics—center position, full width at half maximum (FWHM), and plug velocity—from a zoomed-in region of the long channel surrounding the cross intersection.

2 Background

Electrokinetic transport in aqueous microchannels is governed by several coupled physical phenomena. The choice of field ratios and the qualitative description of injection physics follow Bharadwaj et al. [1], while the electrokinetic theory summarized here is standard and can be found in Probstein [2].

2.1 Electroosmosis

The negatively charged glass walls generate an electric double layer that produces a pluglike electroosmotic flow when an electric field is applied. The Helmholtz–Smoluchowski slip condition gives:

$$\mathbf{u}_{\mathrm{EO}} = -\frac{\varepsilon\zeta}{\mu}\nabla_t V, \tag{1}$$

where ζ is the wall zeta potential, ε is the dielectric permittivity, and μ is the fluid viscosity. Here $\nabla_t V$ denotes the tangential gradient of the electric potential along the wall.

2.2 Electrophoresis

Charged analytes experience a drift velocity in the electric field:

$$\mathbf{u}_{\mathrm{EP}} = \mu_e z F \nabla V, \tag{2}$$

where μ_e is the electrophoretic mobility, z is the charge number, and F is Faraday's constant. Depending on the sign of z, electrophoresis can either augment or oppose the electroosmotic flow.

2.3 Convection and Diffusion

Convection transports the analyte according to the local velocity field,

$$\mathbf{u} = \mathbf{u}_{\mathrm{EO}}, \tag{3}$$

which is nearly plug-like in the channel interior but may contain transverse gradients near the walls. Diffusion acts to smooth concentration gradients according to Fick's law:

$$\mathbf{J}_{\mathrm{diff}} = -D \nabla c, \tag{4}$$

where D is the molecular diffusivity and c is the analyte concentration.

2.4 Nernst–Planck Transport

All contributions combine to produce the total species flux:

$$\mathbf{J} = -D \nabla c - \mu_e z F c \nabla V + c\mathbf{u}, \tag{5}$$

which leads to the Nernst–Planck equation for species conservation:

$$\frac{\partial c}{\partial t} + \nabla \cdot \mathbf{J} = 0. \tag{6}$$

2.5 Injection Physics

During the pinching step, opposing transverse flows in the short arms of the cross sharpen the analyte band at the intersection. During dispensing, axial fields in the long channel accelerate the plug downstream, where dispersion mechanisms cause the full width at half maximum (FWHM) to increase. Correct field orientation and voltage ratios are essential for reproducing observed plug trajectories and shapes.

3 Governing Equations

The COMSOL model couples three physics interfaces: Electric Currents (ec), Creeping Flow (spf), and Transport of Diluted Species (tds). The governing equations below follow the forms reported in the COMSOL auto-generated report for this model.

3.1 Electric Currents (ec)

In the stationary conduction limit, COMSOL solves current conservation in the electrolyte with purely conductive current:

$$\nabla \cdot \mathbf{J} = 0, \tag{7}$$

$$\mathbf{J} = \sigma_b \mathbf{E}, \tag{8}$$

$$\mathbf{E} = -\nabla V, \tag{9}$$

where $\mathbf{J}$ is the current density, σ_b is the buffer conductivity (from the `10 mM Sodium Borate` material), and V is the electric potential. Displacement currents are neglected in this stationary DC model, consistent with the `Current Conservation 1` node in the COMSOL report.

3.2 Creeping Flow (spf)

The flow is modeled using the incompressible Stokes (creeping-flow) approximation. COMSOL writes the momentum balance in terms of the Cauchy stress tensor,

$$\mathbf{0} = \nabla \cdot (-p\mathbf{I} + \mathbf{K}) + \mathbf{F}, \tag{10}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{11}$$

$$\mathbf{K} = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right), \tag{12}$$

where $\mathbf{u}$ is the velocity field, p is pressure, μ is the dynamic viscosity, and $\mathbf{F}$ is any body-force term (zero here). These equations correspond to the Creeping Flow interface with *Neglect inertial term (Stokes flow)* and *Incompressible flow* settings.

Electroosmotic flow is imposed via an electroosmotic slip boundary condition on all solid walls:

$$\mathbf{u}_t = \mathbf{u}_{\mathrm{EO}} = -\frac{\varepsilon_r \varepsilon_0 \zeta}{\mu} \mathbf{E}_t, \tag{13}$$

where ζ is the (negative) wall zeta potential, ε_r is the relative permittivity of water, ε_0 is the vacuum permittivity, and $\mathbf{E}_t$ is the tangential electric field taken from the Electric Currents interface.

3.3 Transport of Diluted Species (tds)

Species transport is governed by the Nernst–Planck equation with convection and migration in the electric field enabled. COMSOL implements this as

$$\frac{\partial c}{\partial t} + \nabla \cdot \mathbf{J}_c + \mathbf{u} \cdot \nabla c = 0, \tag{14}$$

with the total diffusive–migrative flux

$$\mathbf{J}_c = -D \nabla c - z_c \, u_{m,F} \, c \, \nabla V, \tag{15}$$

where c is the fluorescein concentration, D is its diffusion coefficient, z_c is the charge number ($z_c = -1$ for fluorescein), and $u_{m,F}$ is the ionic mobility given by the Nernst–Einstein relation used in COMSOL,

$$u_{m,F} = \frac{D}{RT}. \tag{16}$$

The velocity field $\mathbf{u}$ in Eq. (14) is taken directly from the solution of the Stokes equations (`Velocity field (spf)` selection in the `Fluid 1` node), and the electric potential V is taken from the Electric Currents interface.

Equations (14)–(15) are algebraically equivalent to writing the Nernst–Planck equation in conservative form,

$$\frac{\partial c}{\partial t} + \nabla \cdot \left(-D\nabla c - z_c \frac{DF}{RT} c \nabla V + c\mathbf{u} \right) = 0, \tag{17}$$

which is the form often used in analytical treatments of electrokinetic transport.

4 Modeling Assumptions and Numerical Setup

4.1 Physical and Modeling Assumptions

The COMSOL model makes a number of simplifying assumptions in order to capture the dominant electrokinetic physics while keeping the problem computationally tractable:

- **Depth-averaged 2D model:** The true X8050 channels are three-dimensional with a rectangular cross-section. Here they are represented by a two-dimensional planar cross with an out-of-plane thickness $d_z = 0.02\,\mathrm{mm}$ to approximate the channel depth. This assumes that fields and concentrations are approximately uniform over the depth.
- **Electroneutral, isothermal electrolyte:** The bulk buffer is treated as electroneutral with constant temperature $T = 293.15\,\mathrm{K}$ and uniform properties (σ_b, D, ε_r, μ). Joule heating and temperature-dependent property changes are neglected.
- **Thin electric double layer:** The Helmholtz–Smoluchowski slip condition is applied on all glass walls with a uniform zeta potential $\zeta = -40\,\mathrm{mV}$. This assumes a thin electric double layer relative to the channel height and linear electroosmotic mobility.
- **Single, non-interacting tracer:** Fluorescein is modeled as a single dilute species with constant diffusivity D and charge number $z_c = -1$. Buffer ions and space-charge effects are not resolved explicitly.
- **Quasi-static electric fields:** The Electric Currents interface solves a stationary conduction problem for each voltage configuration. Displacement currents, capacitive charging, and frequency-dependent effects are neglected.
- **No surface reactions or adsorption:** Wall adsorption, chemical reactions, and electrokinetic dispersion associated with surface conduction are ignored.
- **No imposed pressure gradients:** No external pressure-driven flow is applied; the flow is entirely electroosmotic, as set by the slip boundary condition.

4.2 Mesh and Convergence

Resolving the steep electric field, velocity gradients, and concentration fronts at the cross intersection requires a locally refined spatial discretization. Initial simulations were performed using the physics-controlled *Finer*, *Extra fine*, and *Extremely fine* mesh settings. Although these meshes produced qualitatively correct field distributions, the downstream location of the injected plug was not numerically stable: the position of a tracked reference concentration point varied between mesh levels, indicating a lack of mesh independence.

To address this, the mesh was selectively refined only in the regions where high gradients occur and where the plug shape is extracted—specifically, the pinching junction and the first millimeter of the east (long) channel. The remainder of the device retained a coarser discretization to minimize computational cost. Figure 1 shows the final mesh with the user-defined bounding box over which refinement was applied.

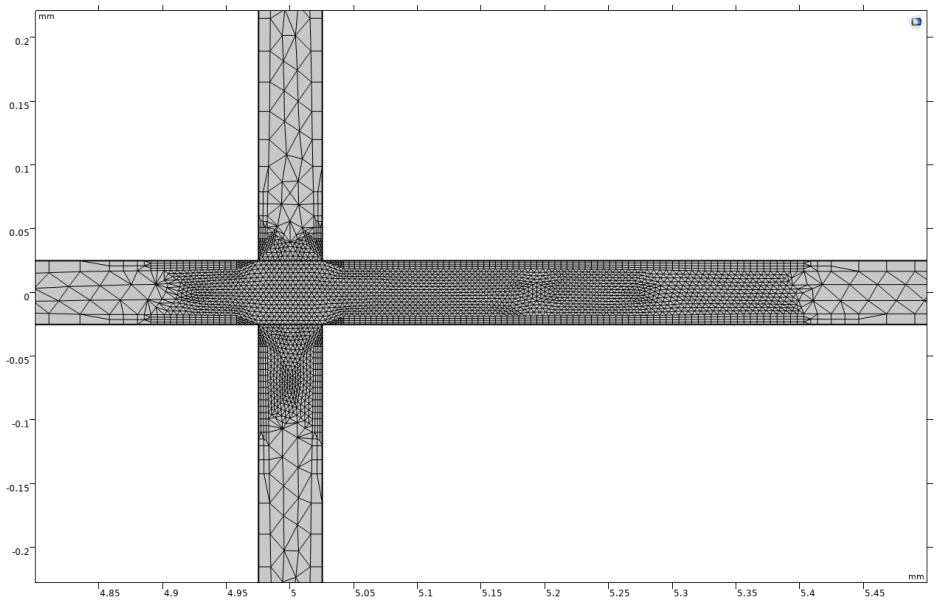


Figure 1: Final mesh used in this study, with local refinement applied only in the pinching intersection and the initial downstream channel segment where plug formation and transport are analyzed.

Mesh convergence was evaluated by tracking the x-position at which the fluorescein concentration reached $c = 0.0014\ \mathrm{mol\,m^{-3}}$ at $t = 19.9$ s. This concentration level lies near the leading edge of the injected plug and is sufficiently sensitive to changes in spatial resolution. The reference point position was extracted for each mesh level and plotted against the corresponding total number of degrees of freedom (DOF). As shown in Figure 2, coarse and physics-controlled meshes did not produce a consistent position. After doubling the mesh resolution within the bounding box, the tracked point location stabilized, and an additional refinement by a factor of three produced no meaningful change, remaining within a 1.15% difference.

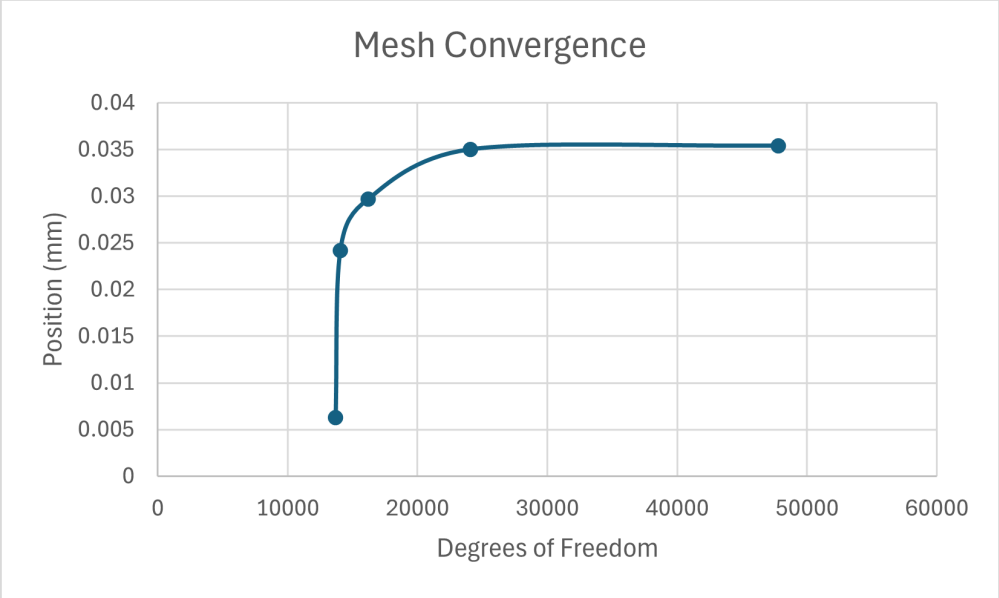


Figure 2: Mesh convergence plot showing the downstream x-position of the tracked reference concentration point ($c = 0.0014\,\mathrm{mol\,m^{-3}}$ at $t = 19.9\,\mathrm{s}$) versus total degrees of freedom (DOF). Convergence is achieved when the mesh is refined by a factor of two in the regions of interest.

Based on this analysis, the refinement-by-2 mesh was selected for all reported simulations. This mesh represents the minimum discretization that produces numerically stable plug metrics without unnecessary computational expense. The mesh-independence verification ensures that the predicted plug center position, velocity, and broadening originate from the model physics rather than from discretization artifacts.

4.3 Solver Configuration and Time Stepping

Each study step uses COMSOL's fully coupled solver:

- **Stationary steps (pinching/dispensing fields):** Direct linear solver (e.g., `MUMPS`) with tolerance factor of 1 and automatic scaling.
- **Time-dependent steps:** Generalized-α time integrator with automatic time stepping. The pinching phase is simulated for $t \in [0, 20\,\mathrm{s}]$ and the dispensing phase for $t \in [20\,\mathrm{s}, 25\,\mathrm{s}]$.
- **Coupling strategy:** Within each time step, the electric potential V is solved and passed to the Creeping Flow interface to update the electroosmotic slip velocity. The resulting velocity field $\mathbf{u}$ is then used in the Nernst–Planck species transport equation.

The maximum time step during dispensing was limited to 0.1 seconds to resolve plug motion and deformation.

5 Geometry

The Micronit X8050 cross microchannel has the following dimensions:

- Channel width: 0.05 mm
- Channel depth: 0.02 mm
- Length of North, West, and South channels: 5 mm
- Length of East (long) channel: 80 mm

A two-dimensional depth-averaged model is employed. The out-of-plane thickness is set to $d_z = 0.02\,\text{mm}$ to approximate the channel depth.

All post-processed simulation figures shown in this report are further restricted to a 2 mm span of the long channel centered on the cross intersection. This zoomed field of view improves visualization of the plug shape and evolution in the region of primary interest.

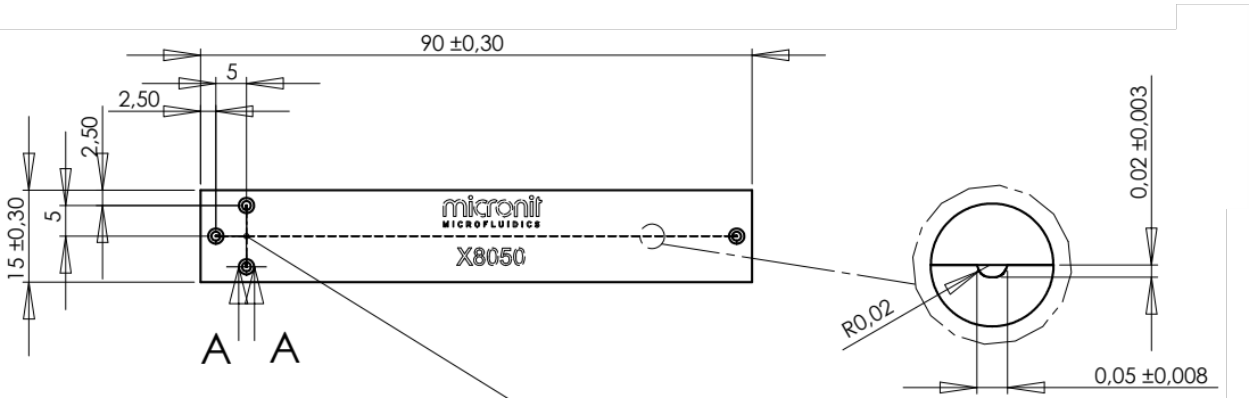


Figure 3: Schematic of the Micronit X8050 cross-channel geometry used for simulation.

6 Physics Setup and Boundary Conditions

6.1 Electric Currents

The South reservoir is grounded. Voltages at the North, East, and West reservoirs follow optimized ratios from literature for standard pinching and dispensing.

6.1.1 Pinching Step Voltages

- $V_{\text{north}} = 377.66\,\text{V}$
- $V_{\text{east}} = 1500\,\text{V}$
- $V_{\text{west}} = 343.09\,\text{V}$
- $V_{\text{south}} = 0$ (ground)

6.1.2 Dispensing Step Voltages

- $V_{\text{north}} = 0$
- $V_{\text{east}} = -1500\,\text{V}$
- $V_{\text{west}} = 272.21\,\text{V}$
- $V_{\text{south}} = 0$ (ground)

6.2 Species Boundary Conditions

- Initial concentration in the sample channel boundary: $c_0 = 0.005\,\text{mol}\,\text{m}^{-3}$.
- No-flux boundaries on walls.
- Inflow or outflow boundaries depend on active flow direction (pinching vs. dispensing).

Correct flow direction during pinching and dispensing is essential for properly identifying which reservoirs act as species inflow and outflow.

6.3 Material Properties

- Conductivity: $\sigma_b = 1638\,\mu\text{S}\,\text{cm}^{-1}$
- Zeta potential: $\zeta = -40\,\text{mV}$
- Diffusivity: $D = 3.5 \times 10^{-10}\ \text{m}^2\,\text{s}^{-1}$
- Electrophoretic charge number: $z = -1$
- Temperature: $T = 293.15\,\text{K}$

7 Study Sequence

The model uses a five-step study:

1. **Stationary: Pinching** — compute electric field and Stokes flow for pinching voltages.
2. **Time Dependent: Pinching** — simulate 20 s of pinching.
3. **Stationary: Dispensing** — compute electric field and Stokes flow for dispensing voltages.
4. **Time Dependent: Dispensing** — simulate 5 s of downstream plug transport.
5. **Derived Values / Combine Solutions** — collect results into a continuous time series.

The mesh is user-controlled: pre-defined as extremely fine with a manually refined section (by a factor of 2) on the area of interest.

8 Results

The following results are cropped to a 0.6 mm window around the cross-channel intersection to enable clearer visualization.

8.1 Electric Field and Flow Patterns

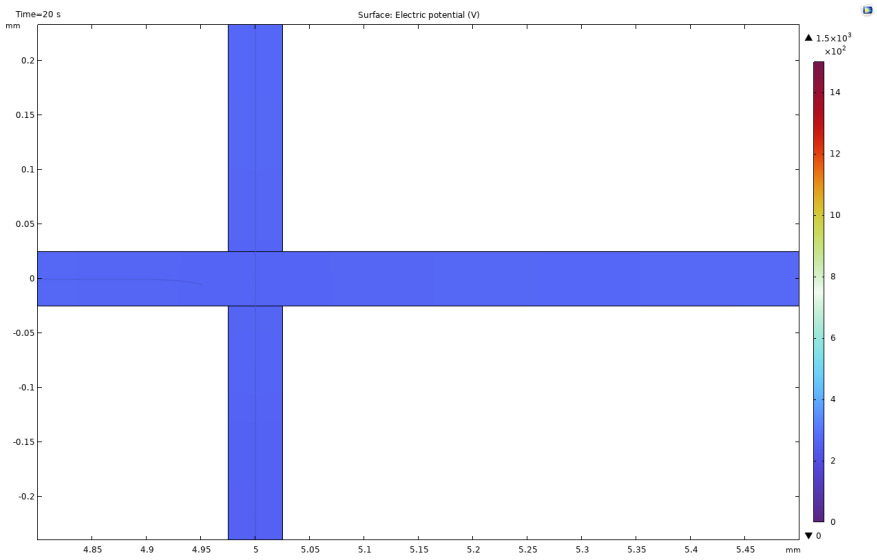


Figure 4: Electric potential distribution during pinching.

Figure 5: Electric potential distribution during dispersion.

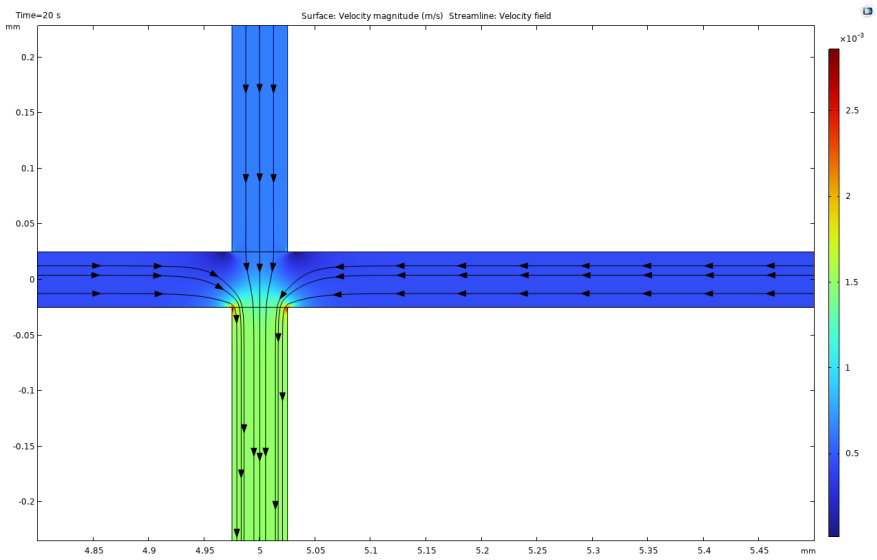


Figure 6: Electroosmotic flow field for the pinching configuration.

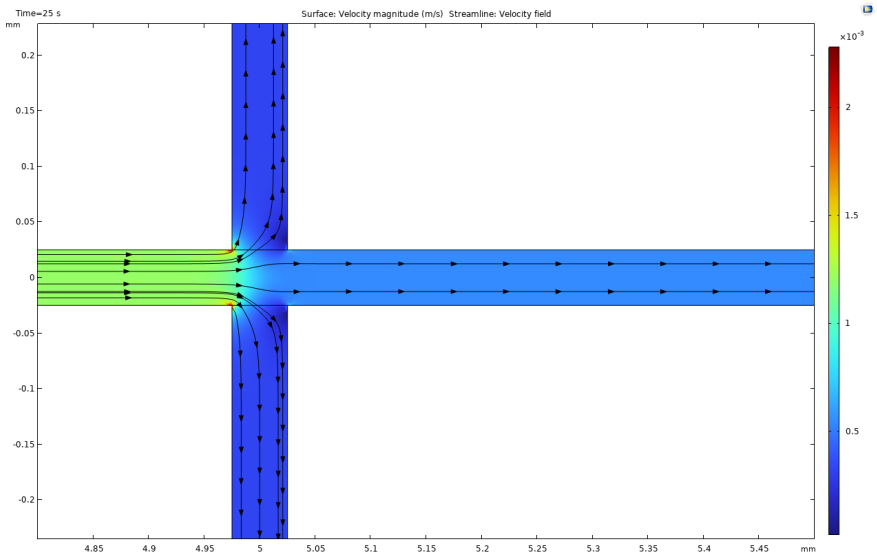


Figure 7: Electroosmotic flow field during dispersion.

8.2 Plug Concentration Snapshots

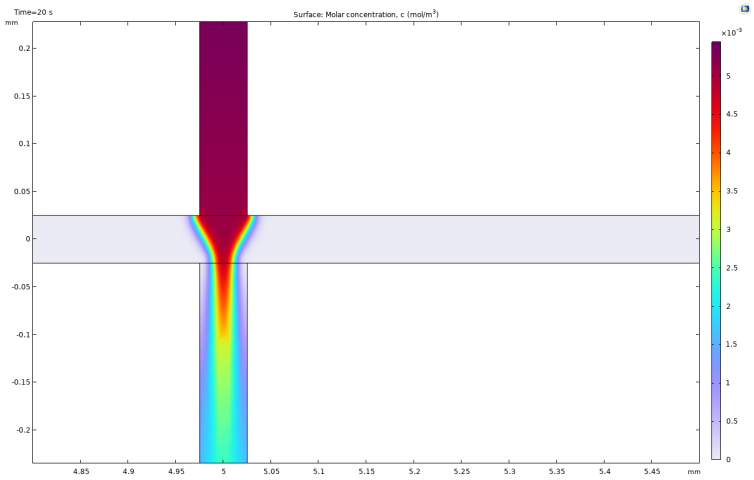
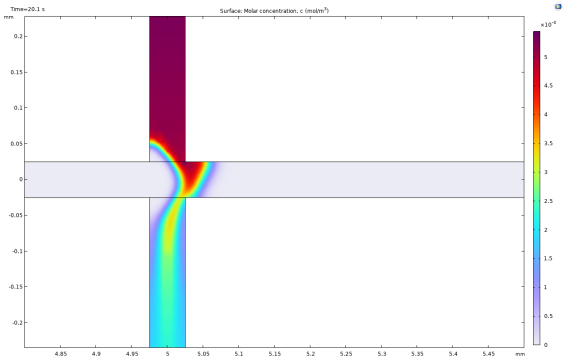
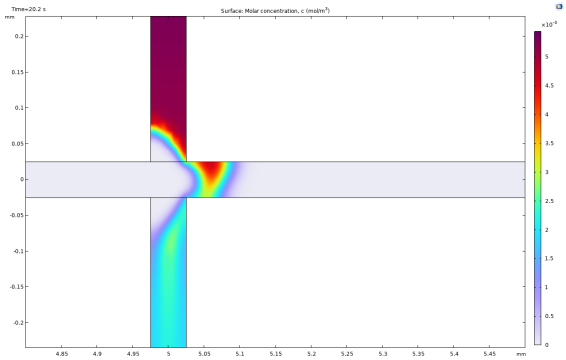


Figure 8: Concentration of fluorescein during pinching at $t = 20\,\mathrm{s}$.

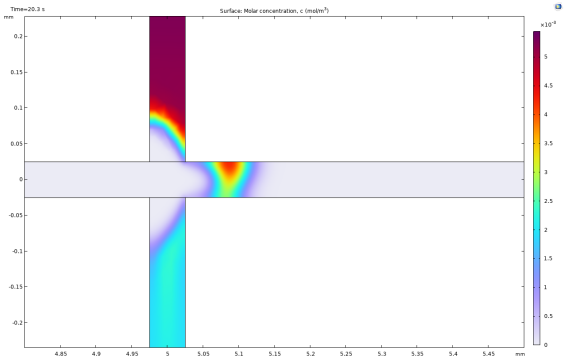
8.3 Time Progression of the Plug



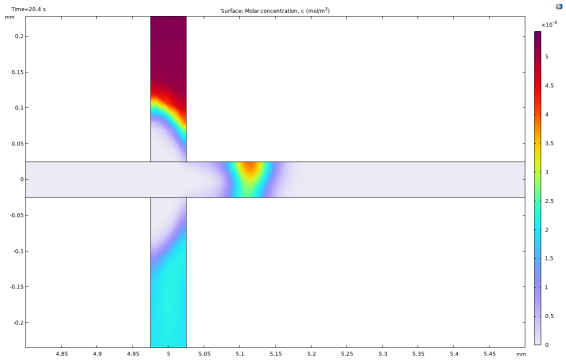
(a) $t = 20.1\,\mathrm{s}$



(b) $t = 20.2\,\mathrm{s}$



(c) $t = 20.3\,\mathrm{s}$



(d) $t = 20.4\,\mathrm{s}$

Figure 9: Time progression of the plug profile at 0.1 s intervals starting during dispersion.

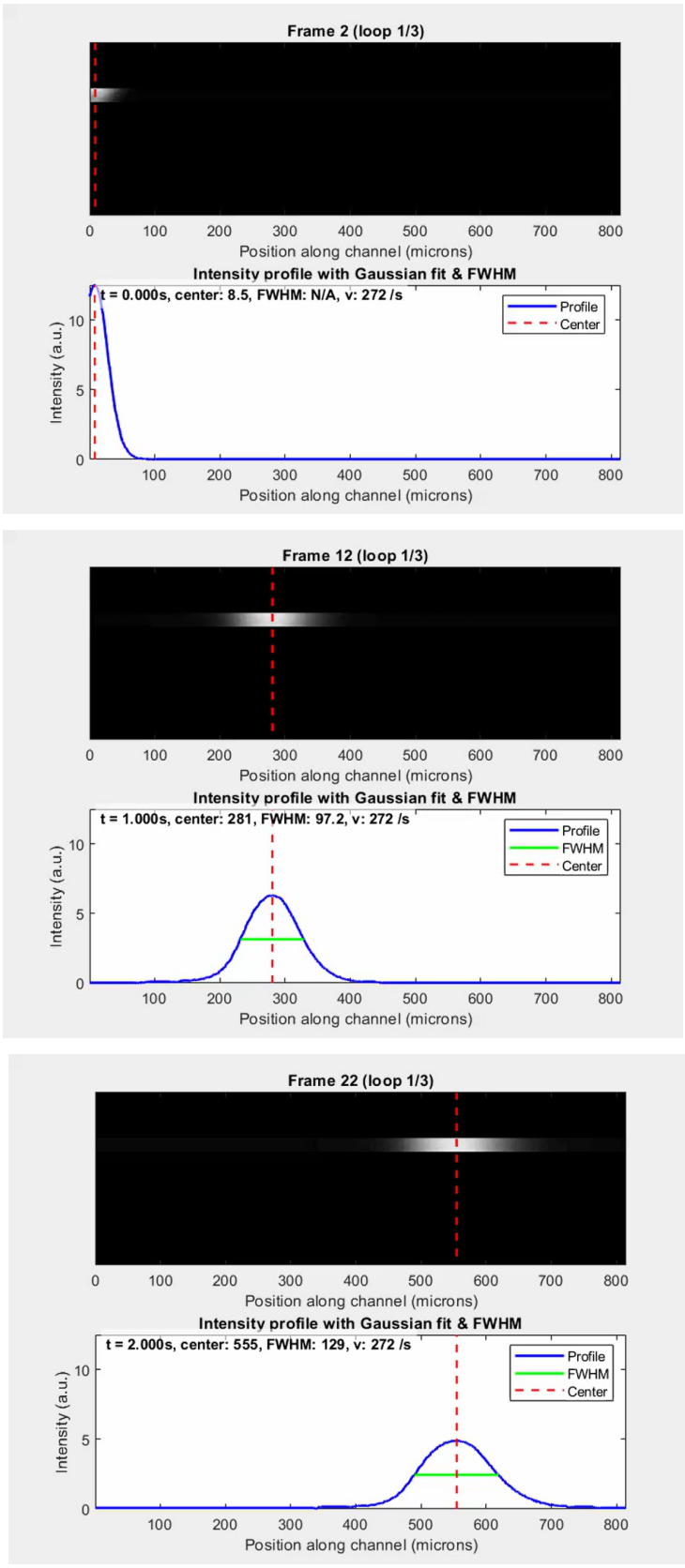


Figure 10: Progression of the electrokinetic plug over time, showing center position and intensity profile at three key frames.

8.4 Plug Center and FWHM

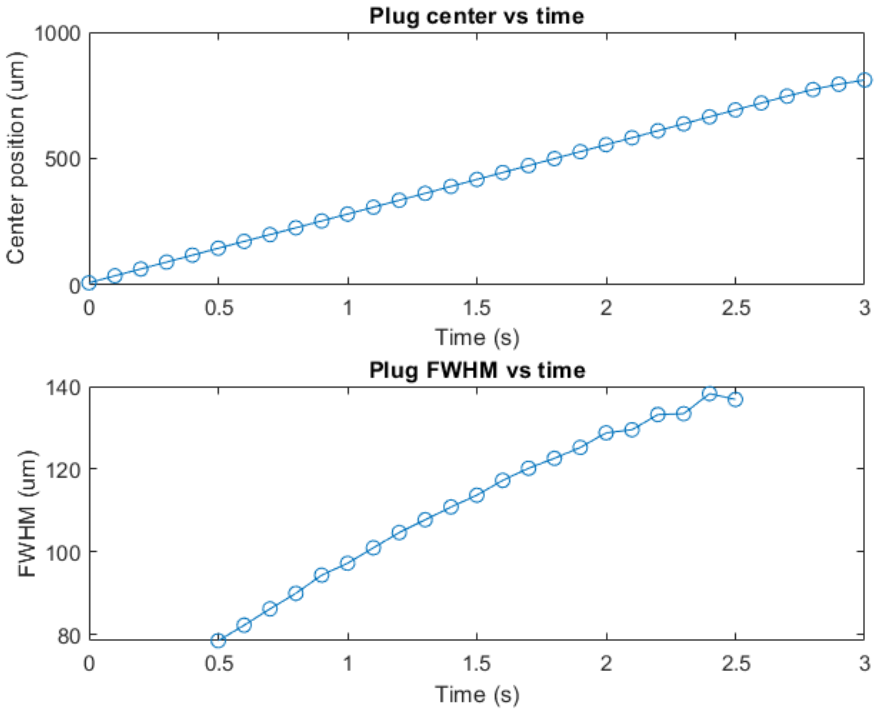


Figure 11: Plug center position (top) and FWHM (bottom) in the 815 µm east channel section after the cross intersection versus time.

8.5 Quantitative Plug Metrics

The figures above illustrate the qualitative evolution of the fluorescein plug as it is pinched and then dispensed into the long east channel. To quantify this behavior, a one-dimensional intensity profile along the long channel is extracted at each time point and fit with a Gaussian profile,

$$I(x,t) \approx I_0(t)\,\exp\left[-\frac{(x-x_c(t))^2}{2\sigma^2(t)}\right], \tag{18}$$

where $x_c(t)$ is the plug center position and $\sigma(t)$ is related to the full width at half maximum (FWHM) via $\mathrm{FWHM}(t) = 2\sqrt{2\ln 2}\,\sigma(t)$. From these fits we obtain:

- Initial injected plug width $\mathrm{FWHM}(t_{\mathrm{inj}}) \approx 80\,\mu\mathrm{m}$,
- Plug velocity in the long channel $u_{\mathrm{plug}} \approx 272\,\mu\mathrm{m/s}$,
- Broadening rate $\mathrm{d(FWHM)}/\mathrm{d}t \approx 30\,\mu\mathrm{m/s}$ over the time window shown in Fig. 11.

These quantitative metrics summarize plug behavior in the present simulation and will serve as baseline quantities for future experimental comparison and model refinement.

9 Discussion

9.1 Importance of Correct Flow Direction

Throughout model development, it became clear that accurately determining the direction of flow in each operational step is critical. Misidentifying flow paths leads to incorrect boundary assignments and propagates error into all subsequent analyses. Correct flow identification enables:

- proper designation of Transport of Diluted Species (TDS) inflow and outflow boundaries,
- correct identification of which reservoirs serve as analyte sources or sinks,
- application of voltage ratios consistent with literature protocols and chip operation.

Since electrokinetic injection relies on sequential voltage-driven flow redirection, a misunderstanding of flow direction fundamentally compromises both the model's predictive accuracy and its relevance to experiment.

9.2 Mechanisms Governing Plug Velocity

The plug velocity arises from the superposition of three transport mechanisms:

- **Electroosmotic flow (EOF)**, which dominates and defines the bulk motion of the buffer,
- **Electrophoresis**, which can accelerate or retard transport depending on the analyte charge, and
- **Convective contributions** arising from channel geometry and flow redirection at the junction.

These contributions combine to yield the net plug velocity:

$$u_{\text{plug}} = u_{\text{EO}} + u_{\text{EP}} + u_{\text{conv}}.$$

Because EOF depends on the wall zeta potential, comparing simulated and experimental plug velocities provides a pathway to extract an effective zeta potential for a given device or buffer composition. This forms an important link between phenomenological chip behavior and underlying interfacial physics.

9.3 Plug Broadening Mechanisms

The observed increase in full width at half maximum (FWHM) aligns with the expected downstream spreading of the analyte plug. Broadening is primarily governed by molecular diffusion, although asymmetric electric fields near the intersection can also distort the

plug shape. When compared with time-resolved experimental measurements, these broadening trends allow estimation of the species diffusivity, providing an additional quantitative validation metric.

Importantly, the plug profile contains more information than just its width. Subtle deviations in plug symmetry, curvature, or centroid displacement can be traced to pressure imbalances between reservoirs or inconsistencies in hydrostatic head across the wells. In this way, simulated and experimental plug shapes can be compared not only to extract material properties, but also to diagnose experimental artifacts and unintended pressure-driven flows—quantities that are otherwise difficult to measure directly.

9.4 Model as a Quantitative Baseline

Although this report does not yet include direct experimental comparison, the present model establishes a consistent and physically grounded baseline for electrokinetic injection in the Micronit X8050 cross-channel platform. The extracted plug center, FWHM evolution, and velocity field visualizations together provide a reproducible reference against which future measurements, parameter sweeps, or device modifications can be evaluated. As experimental datasets become available, this framework will support rigorous extraction of zeta potential, diffusivity, and pressure-driven error sources, enabling progressively tighter coupling between simulation and experiment.

10 Model Limitations and Future Work

The present model captures the dominant electrokinetic physics of EKI in the X8050 cross channel, but it also makes several simplifying assumptions that limit its quantitative accuracy and generality.

10.1 Model Limitations

Key limitations include:

- The use of a depth-averaged 2D approximation, which neglects vertical variations in electric field and flow near the top and bottom walls.
- Neglect of Joule heating and temperature-dependent property changes, which can become significant at higher voltages or in low-conductivity buffers.
- Assumption of a uniform, constant zeta potential, whereas real glass surfaces may exhibit spatial variability and time-dependent conditioning effects.
- Omission of surface reactions, adsorption, and electrokinetic dispersion that can alter plug shape over long distances.
- Limited exploration of parameter space; only a single set of voltages and material properties is fully analyzed in this report.

10.2 Future Work

Several extensions would increase the predictive power and usefulness of the model:

- Systematically explore how plug metrics such as center position, FWHM, and injected volume vary with buffer conductivity, voltage amplitudes, and pulse durations using the existing model as a baseline.
- Calibrate ζ and effective D by fitting simulation results to future experimental plug center and FWHM data, using appropriate error metrics.
- Extend the model to 3D to capture cross-sectional variations in electric field and flow and to assess the validity of the depth-averaged approximation.
- Incorporate Joule heating and temperature-dependent material properties to study high-field or long-duration injections.
- Couple the EKI injection model to downstream separation models, enabling end-to-end optimization of injection plus electrophoretic separation.

Together, these enhancements would further align the COMSOL model with ongoing electrokinetic injection studies in the Micronit X8050 platform and with future LabSmith HVS experiments in the Pennathur Lab.

11 Conclusion

This work establishes a multiphysics modeling framework that captures the essential physics governing electrokinetic injection in the Micronit X8050 glass microchannel platform. By coupling electric fields, electroosmotic slip, Stokes flow, and Nernst–Planck species transport within a depth-averaged two-dimensional geometry, the model reproduces the key qualitative features of the pinching–dispensing sequence and quantifies the resulting plug trajectory, velocity, and broadening behavior. The extracted plug metrics—center position, full width at half maximum, and transport velocity—form a coherent set of descriptors that connect device operation to underlying electrokinetic mechanisms.

Beyond reproducing the canonical injection process, the model provides a principled foundation for experimental interpretation. Because EOF-driven transport is sensitive to interfacial properties and subtle hydrostatic imbalances, simulated plug shapes and trajectories offer a route to infer otherwise inaccessible quantities such as wall zeta potential, molecular diffusivity, or unintended pressure offsets between wells. In this way, the present framework transforms the injected plug from a mere visual observable into a diagnostic signal that encodes both material parameters and experimental artifacts.

Although this report focuses exclusively on simulation, the modeling infrastructure developed here is designed for direct integration with future fluorescence measurements in the Pennathur Lab. By fitting simulated plug metrics to experimental traces, the model can be iteratively refined into a quantitative tool for parameter extraction, protocol optimization, and ultimately, predictive control of electrokinetic injection conditions. As such, it serves

not only as a description of the phenomena but as a starting point for data-driven calibration and operation of LabSmith HVS-driven microfluidic systems.

In summary, this work demonstrates that a carefully constructed COMSOL model can resolve the interplay of electroosmosis, electrophoresis, convection, and diffusion in a complex microchannel junction and provides the baseline necessary for transitioning from qualitative visualization to quantitative inference. Its value will increase as experimental datasets become available, enabling a unified, simulation–experiment workflow for characterizing and improving electrokinetic injection in the Micronit X8050 platform.

References

[1] R. Bharadwaj, J. G. Santiago, and B. Mohammadi, "Design and optimization of on-chip capillary electrophoresis," *Electrophoresis*, vol. 23, no. 16, pp. 2729–2744, 2002.

[2] R. F. Probstein, *Physicochemical Hydrodynamics: An Introduction*, 2nd ed. John Wiley & Sons, New York, 2003.

Acknowledgements

I gratefully acknowledge the support of the **Pennathur Lab** for providing experimental data and guidance throughout this project.

Appendix

Table of Contents

cross_injection_voltages.m

Compute reservoir voltages for cross-channel injection given geometry and electric-field **magnitude** ratios (from the paper) and user-defined field directions (signs).

```
clear; clc;
```

---- USER INPUTS ----

Channel lengths from junction to each well (meters)

```
Ln = 5e-3;    % sample well (north)
Ls = 5e-3;    % sample waste (south)
Lw = 5e-3;    % buffer well (west)
Le = 80e-3;    % buffer waste (east)

Vmax = 1500;  % desired maximum magnitude on any electrode (V)

% Field magnitude ratios En/|Es|, Ee/|Es|, Ew/|Es| from the paper
FIELD_RATIOS.pinch    = [0.42, 0.29, 0.29];   % [En/|Es|, Ee/|Es|, Ew/|Es|]
FIELD_RATIOS.reverse  = [2.32, 0.66, 0.66];
FIELD_RATIOS.dispense = [1.00, 0.43, 2.43];

% field direction (sign) for each arm: [sign(En) sign(Es) sign(Ee) sign(Ew)]
% Choose signs so they match desired flow directions.
SIGN.pinch    = [+1, -1, +1, +1];
SIGN.reverse  = [-1, +1, +1, +1];
SIGN.dispense = [+1, +1, -1, +1];
```

---- MAIN: compute voltages for each step ----

```
steps = fieldnames(FIELD_RATIOS);

for k = 1:numel(steps)
    stepName = steps{k};
    ratios   = FIELD_RATIOS.(stepName);       % [En/|Es|, Ee/|Es|, Ew/|Es|]
    rn = ratios(1);
    re = ratios(2);
    rw = ratios(3);

    sgn  = SIGN.(stepName);                   % [sn, ss, se, sw]
    sn = sgn(1);  ss = sgn(2);  se = sgn(3);  sw = sgn(4);

    % 1) Nominal *magnitude* of Es
```

```
    Es_mag = 1.0;                  % |Es|
    Es0    = ss * Es_mag;          % signed Es in S arm

    % 2) Set Vs = 0 (ground) and find junction potential Vc:
    %    Es0 = (Vs - Vc)/Ls  =>  Vc = Vs - Es0*Ls
    Vs0 = 0.0;
    Vc0 = Vs0 - Es0 * Ls;

    % 3) Fields in other arms with ratios and signs
    En0 = sn * rn * Es_mag;
    Ee0 = se * re * Es_mag;
    Ew0 = sw * rw * Es_mag;

    % 4) Reservoir voltages from E = (Vres - Vc)/L (same definition as
before)
    Vn0 = Vc0 + En0 * Ln;   % sample well
    Ve0 = Vc0 + Ee0 * Le;   % buffer waste
    Vw0 = Vc0 + Ew0 * Lw;   % buffer well

    % 5) Scale all voltages so max |V| = Vmax
    Vvec0 = [Vn0, Vs0, Vw0, Ve0];
    scale = Vmax / max(abs(Vvec0));
    Vn = Vn0 * scale;
    Vs = Vs0 * scale;   % stays 0
    Vw = Vw0 * scale;
    Ve = Ve0 * scale;
    Es = Es0 * scale;   % final signed Es in S arm (V/m)

    % Print results
    fprintf('\n=== %s STEP ===\n', upper(stepName));
    fprintf('Vn (sample well)   = %7.2f V\n', Vn);
    fprintf('Vs (sample waste)  = %7.2f V (ground)\n', Vs);
    fprintf('Vw (buffer well)   = %7.2f V\n', Vw);
    fprintf('Ve (buffer waste)  = %7.2f V\n', Ve);
    fprintf('Es (field in S arm)= %7.3f V/m\n', Es);
end
```

Published with MATLAB® R2025a

Table of Contents

analyzePlugVideo.m

Analyze grayscale EK plug videos.

```
clear; clc; close all;
```

=== User Input / Basic Settings ==========

```
[vidFile, vidPath] = uigetfile({'*.mp4;*.avi;*.mov;*.m4v','Video Files'}, ...
    'Select plug video');
if isequal(vidFile,0)
    error('No video selected.');
end
videoFullPath = fullfile(vidPath, vidFile);

% Playback options
playbackSlowFactor = 1;   % >1 = slower playback; 3 means 3x slower than
real-time
numLoops           = 1;   % number of times to loop over trimmed segment

% Field-of-view length (physical) along the channel (x direction)
fieldOfViewLength = input('Enter field-of-view length along the channel
(microns): ');
```

=== Load Video and Basic Info ==========

```
vObj = VideoReader(videoFullPath);
frameRate = vObj.FrameRate;
numFrames = floor(vObj.Duration * frameRate);
```

```
fprintf('Frame rate: %.2f fps, Approx. total frames: %d\n', frameRate,
numFrames);

% Sample frame for size and trimming preview
sampleFrame = readGrayFrame(vObj, 1);
[frameH, frameW] = size(sampleFrame);
```

=== Interactive Trimming ==========

```
[startFrame, endFrame] = interactiveTrim(vObj, numFrames);
numAnalyzedFrames      = endFrame - startFrame + 1;

frameIndices = (startFrame:endFrame).';
timeVec      = (frameIndices - startFrame) / frameRate;  % t = 0 at
startFrame
```

=== Preallocate Results ==========

```
centerPhys    = nan(numAnalyzedFrames,1);  % center position (physical units)
centerPix     = nan(numAnalyzedFrames,1);  % center position (pixels,
continuous)
FWHMphys      = nan(numAnalyzedFrames,1);  % FWHM in physical units
FWHMpix       = nan(numAnalyzedFrames,1);  % FWHM in pixels
profileMax    = nan(numAnalyzedFrames,1);  % peak intensity after baseline
removal

% Pixel -> physical mapping for x-axis
xPix  = (1:frameW).';  % column indices
xPhys = (xPix - 0.5) * (fieldOfViewLength / frameW);
```

=== Main Analysis Loop ==========

```
for k = 1:numAnalyzedFrames
    frameIdx = frameIndices(k);

    % Read grayscale frame (double)
    frame = readGrayFrame(vObj, frameIdx);

    % Collapse to 1D profile: average over rows
```

```
    profile = mean(frame, 1).';                    % (frameW x 1)

    % Simple baseline removal: shift so minimum is zero
    profileClean = profile - min(profile);
    y = profileClean;
    x = xPhys;

    % If almost flat, skip
    if max(y) <= 0
        continue;
    end

    % Fit a Gaussian without offset: y = A * exp(-(x - x0)^2 / (2*sigma^2))
    [A, x0, sigma, ok] = fitGaussian1D_simple(x, y);

    if ~ok
        % Skip nonsense fits
        continue;
    end

    % Store center and FWHM
    centerPhys(k) = x0;
    centerPix(k)  = (x0 / fieldOfViewLength) * frameW + 0.5;
    FWHMphys(k)   = 2 * sqrt(2*log(2)) * abs(sigma);
    FWHMpix(k)    = FWHMphys(k) * frameW / fieldOfViewLength;
    profileMax(k) = max(y);
end
```

=== Mask FWHM at beginning and end (first/last 0.5 s) ==========

```
% *** Only use FWHM where plug is fully in view: t in [0.5, T-0.5]
Ttotal = timeVec(end);
if Ttotal > 1
    maskFwhm = (timeVec >= 0.5) & (timeVec <= Ttotal - 0.5);
else
    maskFwhm = true(size(timeVec));  % if super short clip, don't mask
end
FWHMphysPlot = FWHMphys;
FWHMpixPlot  = FWHMpix;
FWHMphysPlot(~maskFwhm) = NaN;
FWHMpixPlot(~maskFwhm)  = NaN;
```

=== Compute Velocity from Center vs Time ==========

```
validIdx = ~isnan(centerPhys);
if nnz(validIdx) < 2
    warning('Not enough valid center points to compute velocity.');
    plugVelocity = NaN;
```

```
else
    p = polyfit(timeVec(validIdx), centerPhys(validIdx), 1);
    plugVelocity = p(1); % physical units per second
    fprintf('Estimated plug velocity: %.4g (um/s)\n', plugVelocity);
end

% Fixed intensity axis limit (use max over valid frames)
globalMaxIntensity = max(profileMax(validIdx));
if isempty(globalMaxIntensity) || globalMaxIntensity <= 0
    globalMaxIntensity = 1;
end
```

=== Synchronized Visualization: Video + Profile ==========

```
hFig = figure('Name','Plug Analysis: Video + Intensity Profile');
colormap gray;

for loopIdx = 1:numLoops
    for k = 1:numAnalyzedFrames
        frameIdx = frameIndices(k);

        if ~ishandle(hFig)
            break;
        end

        % Read frame
        frame = readGrayFrame(vObj, frameIdx);
        frameSm = frame;  % raw grayscale

        % 1D profile, same processing as in analysis
        profile = mean(frameSm, 1).';
        profileClean = profile - min(profile);
        y = profileClean;

        clf(hFig);

        % --- Top: Video frame with center overlay ---
        subplot(2,1,1);
        % *** Use xPhys so x-axis matches bottom plot width
        imagesc(xPhys, 1:frameH, frameSm);
        set(gca,'YDir','reverse');  % keep image upright
        axis tight;
        title(sprintf('Frame %d (loop %d/%d)', frameIdx, loopIdx, numLoops));
        xlabel('Position along channel (microns)');
        yticks([]);  % hide y ticks for video

        hold on;
        if ~isnan(centerPhys(k))
            xC = centerPhys(k);  % *** center in physical units
            plot([xC xC], [1 frameH], 'r--', 'LineWidth', 1.5);
        end
```

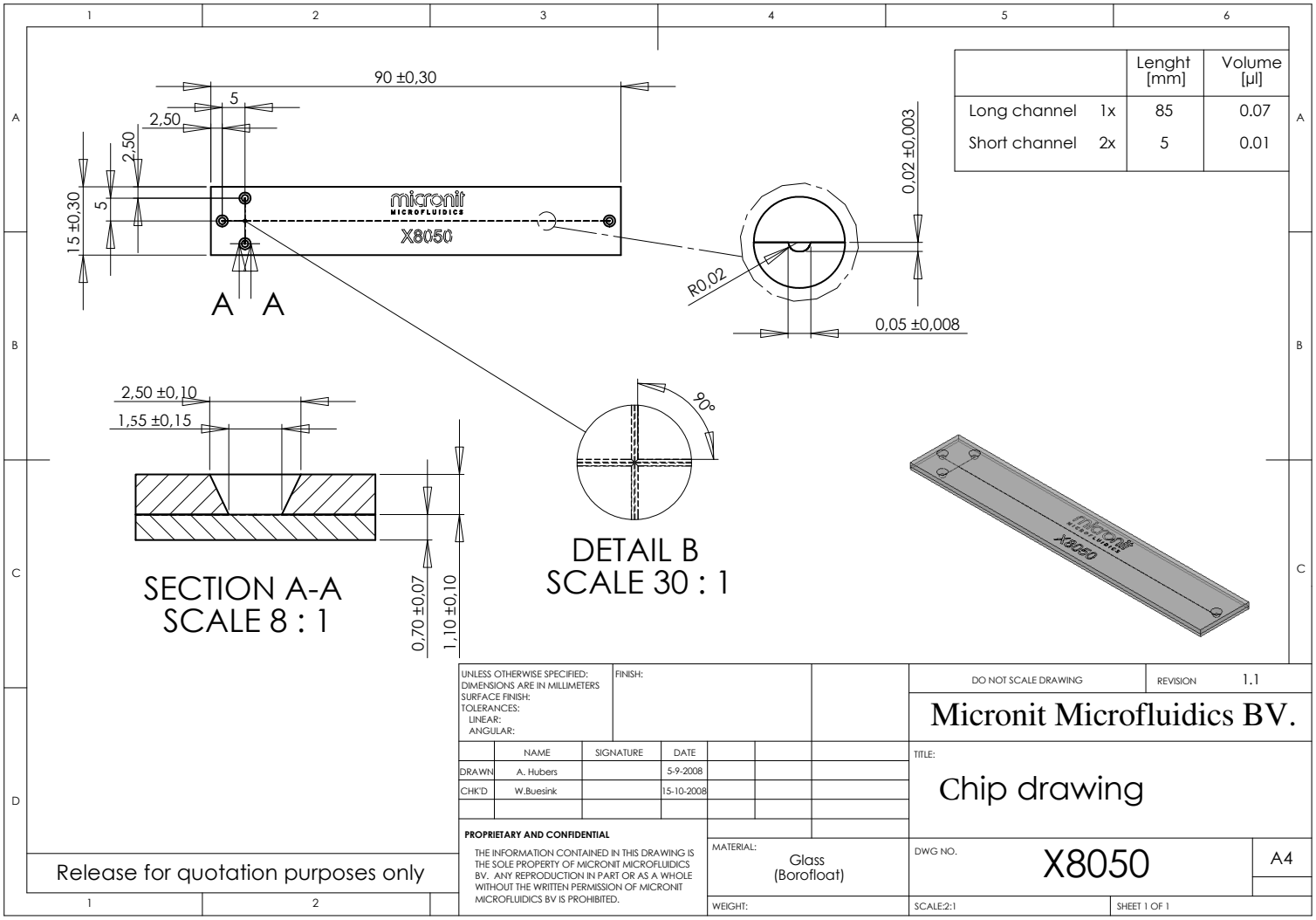
```
hold off;

% --- Bottom: Intensity profile with Gaussian fit + FWHM ---
subplot(2,1,2);
plot(xPhys, y, 'b-', 'LineWidth', 1.5);
hold on;
xlabel('Position along channel (microns)');
ylabel('Intensity (a.u.)');
title('Intensity profile with Gaussian fit & FWHM');
xlim([xPhys(1) xPhys(end)]);
ylim([0 globalMaxIntensity]);

% *** Only draw FWHM where maskFwhm is true
if ~isnan(centerPhys(k)) && ~isnan(FWHMphysPlot(k))
    xC = centerPhys(k);
    fwhmVal = FWHMphysPlot(k);
    halfMax = max(y) / 2;

    % FWHM band: horizontal segment at half max, +/- FWHM/2
    leftX  = xC - fwhmVal/2;
    rightX = xC + fwhmVal/2;
    plot([leftX rightX], [halfMax halfMax], 'g-', 'LineWidth', 1.5);

    % Center line
    plot([xC xC], [0 globalMaxIntensity], 'r--', 'LineWidth', 1.2);

    legend({'Profile','FWHM','Center'}, 'Location','best');
elseif ~isnan(centerPhys(k))
    % Draw just center line when FWHM masked
    xC = centerPhys(k);
    plot([xC xC], [0 globalMaxIntensity], 'r--', 'LineWidth', 1.2);
    legend({'Profile','Center'}, 'Location','best');
end

% Live text annotation
tNow  = timeVec(k);
if isnan(plugVelocity)
    vStr = 'v: N/A';
else
    vStr = sprintf('v: %.3g /s', plugVelocity);
end

if isnan(centerPhys(k))
    cStr = 'center: N/A';
else
    cStr = sprintf('center: %.3g', centerPhys(k));
end

% *** Show FWHM as N/A when masked
if isnan(FWHMphysPlot(k))
    wStr = 'FWHM: N/A';
else
    wStr = sprintf('FWHM: %.3g', FWHMphysPlot(k));
end
```

```
        txt = sprintf('t = %.3fs, %s, %s, %s', tNow, cStr, wStr, vStr);
        text(0.02, 0.95, txt, 'Units','normalized', 'Color','k', ...
             'FontSize',9, 'FontWeight','bold', 'BackgroundColor',[1 1 1
0.6]);

        hold off;
        drawnow;

        pause((1/frameRate) * playbackSlowFactor);
    end

    if ~ishandle(hFig)
        break;
    end
end
```

=== Summary Plots ==========

```
figure('Name','Summary: Center and FWHM vs Time');
subplot(2,1,1);
plot(timeVec, centerPhys, 'o-');
xlabel('Time (s)');
ylabel('Center position (um)');
title('Plug center vs time');

subplot(2,1,2);
% *** Use masked FWHM here so endpoints don't plot
plot(timeVec, FWHMphysPlot, 'o-');
xlabel('Time (s)');
ylabel('FWHM (um)');
xlim([0 timeVec(end)]);
title('Plug FWHM vs time');
```

Published with MATLAB® R2025a



		Lenght [mm]	Volume [µl]
Long channel	1x	85	0.07
Short channel	2x	5	0.01

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DO NOT SCALE DRAWING | REVISION 1.1

Micronit Microfluidics BV.

	NAME	SIGNATURE	DATE
DRAWN	A. Hubers		5-9-2008
CHK'D	W.Buesink		15-10-2008

TITLE:

Chip drawing

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF MICRONIT MICROFLUIDICS BV. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF MICRONIT MICROFLUIDICS BV IS PROHIBITED.

MATERIAL: Glass (Borofloat)

DWG NO. X8050

A4

WEIGHT:

SCALE:2:1

SHEET 1 OF 1

Release for quotation purposes only