

Supplementary Material for
Toolbox for the Design of Optimized Microfluidic Components

David R. Mott*, Peter B. Howell, Jr. †, Joel Golden†,
Carolyn R. Kaplan*, Frances S. Ligler†, Elaine S. Oran*,
Naval Research Laboratory
Washington, DC 20375

Appendix: Lagrangian Advection Approach

Lagrangian techniques for passive scalars require the construction of the streamline or pathline for a massless particle; this path is defined by $d\mathbf{x}/dt = \mathbf{u}$, where \mathbf{x} is the coordinate location and \mathbf{u} is the velocity vector. A tremendous body of work describes Lagrangian advection methods, and the field continues to expand through improvements in the methods and intriguing applications.¹⁻⁷ In this appendix, we describe an algorithm for calculating the path of a passive particle through an established velocity field. The relative merits of tracing particle paths backwards from a component exit rather than forward from the component entrance is also explored.

Advection Routine for Calculating Streamlines

The advection routine used in the current work is very fast because this construction of the pathline does not require numerical integration: the velocity field is assumed to vary linearly in each computational cell between the interface-normal velocities provided by TINY3D,⁸ so given the downstream point where a particle left a cell, the code can calculate directly the point at which the particle entered the cell (or vice-versa). This algorithm is described in detail by Mott et al.⁹ The case for tracking particles as they

* Laboratory for Computational Physics and Fluid Dynamics

† Center for Biomolecular Science and Engineering

move downstream will be presented, and the procedure for tracing streamlines upstream is identical if the velocity field $\mathbf{U}(x,y,z)$ is replaced with $-\mathbf{U}$.

For example, consider a particle that enters a cell at x_i , and define Δx_p as the distance this particle must traverse to cross the downstream x interface. Assume that $u(x)$ varies linearly in x between u_i at x_i and $u_f = u_i + \Delta u_p$ at that downstream interface at $x_i + \Delta x_p$. Then the time required to reach that interface is

$$t_x = \frac{\Delta x_p}{\Delta u_p} \log \left(1 + \frac{\Delta u_p}{u_i} \right) \quad (\text{A1})$$

We calculate the exit times t_y and t_z for the other coordinate directions as well, and $t_{\text{exit}} = \min(t_x, t_y, t_z)$ indicates when the particle will leave the cell. The x location at t_{exit} relative to the point the particle entered the cell is

$$x(t_{\text{exit}}) = \frac{u_i}{\Delta u_p} \left[\exp \left(\frac{\Delta u_p t_{\text{exit}}}{\Delta x_p} \right) - 1 \right] \Delta x_p \quad (\text{A2})$$

Therefore, we calculate the new location of the particle as it exits the cell without numerically integrating the trajectory – the exit point is provided by the above equations in closed form. This procedure can be used to project points downstream to their final destinations or upstream to their points of origin. This algorithm removes the time step constraint imposed to ensure stability and accuracy when integrating the particle path using a more traditional approach. This approach also eliminates “lost” particles that are reported by some authors. These lost particles either intersect a wall or are trapped in paths that are prohibitively expensive to calculate and are therefore abandoned before the particle exits the geometry. Lost particles would result in “holes” in the advection map where data would be unavailable for interpolating the particle coordinates, so alternate schemes that do not guarantee that every particle path is successfully integrated cannot be used to define an advection map.

Forward-Tracking versus Backtracking Particle Paths

Figure A1 repeats the channel with the single groove shown in Fig. 1 of the main text, and Figure A2 compares two advection methods for specifying the dye distribution in plane 3 using discrete particles. In Fig. A2a, an equally-spaced distribution of 64 x 22 particles is introduced at the inflow plane 1 of Fig. A1 and tracked downstream to plane 3. Each particle's color is determined at the start of the calculation by where it was released (i.e., the left half produces dark particles, and the right half produces light particles), and the calculation of its path determines its final location in Fig. A2a. In Fig. A2b, a uniform grid of particles is placed in the outflow plane 3, but their colors are not specified at the beginning of the calculation. The streamline intersecting each particle is traced upstream to determine where each particle intersects plane 1. Particles that intersect plane 1 in the left half of the channel are specified dark, and those that intersect plane 1 in the right half of the channel are light. In both cases, the algorithm described above is used to determine the particle paths. Large gaps appear in Fig. A2a near the floor of the channel where the fluid experiences the most deformation. This sparse coverage limits the information available about the species concentrations in this area. In contrast, a uniform distribution is ensured in Fig. 2b since the particle locations are specified at this plane and their paths backtracked to obtain the entrance location and the color.

The distinction between these two approaches is critical to the current design approach. Each advection map stores a distribution of points in the outflow plane and the points upstream that lie on the streamlines passing through these points. To determine the inflow coordinate that corresponds to an arbitrary outflow point, the map must interpolate using the points that are stored. This interpolation would be erratic and inaccurate in areas that are poorly populated with points, such as near the channel floor in Fig. A2a. Therefore, the advection maps are constructed using backtracking from specified points in the outflow plane so that a uniform distribution of points is available for the interpolation.

References

1. C.W. Hirt, A. A. Amsden, and J.L. Cook. Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds. *Journal of Computational Physics* 14 (3) (1974) 227-253.

2. J.K. Dukowicz and J.R. Baumgardner. Incremental Remapping as a Transport/Advection Algorithm, *Journal of Computational Physics* 160 (2000) 318-335.
3. A. Staniforth and J. Cote, Semi-Lagrangian Integration Schemes for Atmospheric Models – A Review. *Monthly Weather Review* 119 (1991), 2206 – 2223.
4. A. Staniforth, A. White, and N. Wood. Analysis of semi-Lagrangian trajectory computations, *Q.J.R. Meteorol. Soc.* 129 (2003), 2065-2085.
5. K. Shabazi, M. Paraschivoiu, and J. Mostaghimi. Second order accurate volume tracking based on remapping for triangular meshes. *Journal of Computational Physics* 188 (2003) 100-122.
6. P. Bochev and M. Shashkov. Constrained interpolation (remap) of divergence-free fields. *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 511-530.
7. W.H. Lipscomb and E.C. Hunke. Modeling Sea Ice Transport Using Incremental Remapping. *Monthly Weather Review* 132 (2004), 1341-1354.
8. D.R. Mott, C.R. Kaplan, and E.S. Oran. A Robust Solver for Incompressible Flow on Cartesian Grids with Colocated Variables," *NRL Technical Memorandum Report NRL/MR/6404-05-8858*, 2005.
9. D.R. Mott, P.B. Howell, Jr., J.P. Golden, C.R. Kaplan, F.S. Ligler, E.S. Oran. "A Lagrangian Advection Routine Applied to Microfluidic Component Design," *AIAA paper 2006-1086*, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2006.

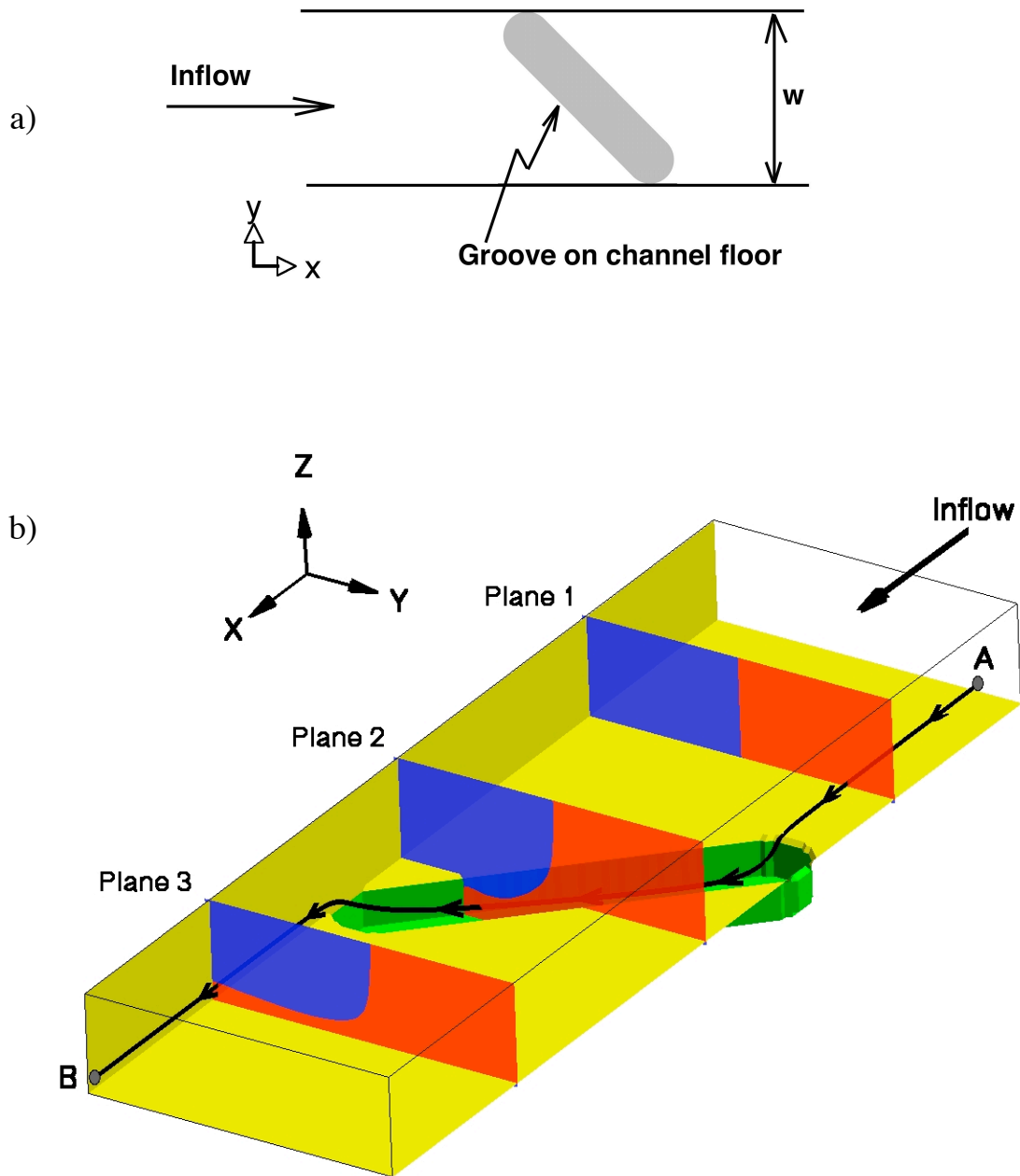


Figure A1: Microchannel with single diagonal groove cut into the bottom. a) top view of a microchannel with a single diagonal groove (gray). Flow enters the channel from the left. b) Perspective plot showing the distribution of two species (dark and light) at three stations and the streamline between points A and B.

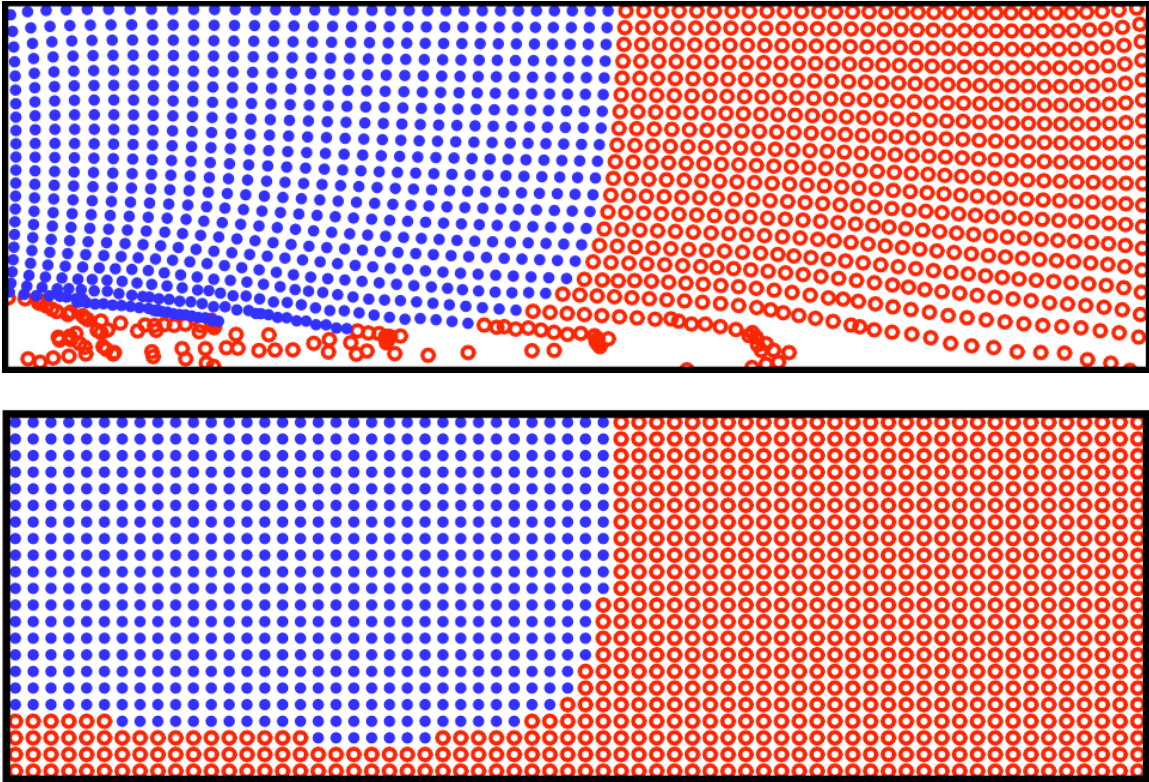


Figure A2: Particle distributions exiting the component at plane 3 in Fig. A1. Filled symbols show particles that entered the component in the left half of the channel, and open symbols represent particles that entered the component on the right. a) 64×22 particles that enter uniformly spaced at plane 1 in Fig. 1 and move downstream through the component b) 64×22 particles uniformly placed in the exit plane 3; the symbol type is chosen by tracking each particle's path upstream through the component to determine its point of origin.