Lab on a Chip

Accepted Manuscript

ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/loc

ARTICLE TYPE www. rsc.org/xxxxxx | XXXXXXXX

Micropillar sequence designs for fundamental inertial flow transformations

Daniel Stoecklein,[†a] Chueh-Yu Wu,[†b] Keegan Owsley[‡b], Yu Xie,[‡a] Dino Di Carlo,[*b] and Baskar Ganapathysubramanian[*a]

DOI: 10.1039/b000000x

The ability to control the shape of a flow in a passive microfluidic device enables potential applications in chemical reaction control, particle separations, and complex material fabrication. Recent work has demonstrated the concept of sculpting fluid streams in a microchannel using a set of pillars or other structures that individually deform a flow in a predictable pre-computed manner. These individual pillars are then placed in a defined sequence within the channel to yield the composition of the individual flow deformations - and ultimately complex user defined flow shapes. In this way, an elegant mathematical operation can yield the final flow shape for a sequence without experiment or additional numerical simulation. Although these approaches allow for programming complex flow shapes without understanding the detailed fluid mechanics, the design of an arbitrary flow shape of interest remains difficult, requiring significant design iteration. The development of intuitive basic operations (i.e. higher level functions that consist of combinations of obstacles) that act on the flow field to create a basis for more complex transformations would be useful in systematically achieving a desired flow shape. Here, we show eight transformations that could serve as a partial basis for more complex transformations. We initially used an in-house, freely available custom software (uFlow) which allowed us to arrive at these transformations that include making a fluid stream concave and convex, tilting, stretching, splitting, adding a vertex, shifting, and encapsulating another flow stream. The pillar sequences corresponding to these transformations were subsequently fabricated and optically analyzed using confocal imaging - yielding close agreement with uFlow-predicted shapes. We performed topological analysis on each transformation, characterizing potential sequences leading to these outputs and trends associated with changing diameter and placement of the pillars. We classify operations into four sets of sequence-building concatenations: stacking, recursion, mirroring, and shaping. The developed basis should help in the design of microfluidic systems that have a phenomenal variety of applications, such as optofluidic lensing, enhanced heat transfer, or new polymer fiber design.

1 Introduction

The ability to control the cross-sectional shape of a fluid stream is enabling for a variety of microfluidic applications, from optimizing mixing for reactions and heat transfer[1] to developing tunable optical component such as fluid lenses[2], to fabricating shaped polymer fibers[3]. The most common methods by which fluid parcels have been diverted across a channel cross-section to shape a stream make use of asymmetric structured channels (e.g. grooves, chevrons) in Stokes flow[4] or curved channels with finite inertia[1,5,6]. Grooved channels have been widely used to mix flows[7] and more recently combinations of grooves have been combined to shape flows[8] in a programmable manner.

Fluid inertia also allows for the cross-stream motion of fluid parcels in curving or structured channels. Inertial fluid flow ($1 < Re < 100$, where the Reynolds number, $Re = \rho U D_H/\mu$, is the ratio of inertial to viscous forces in the flow, with fluid density ρ, viscosity μ, downstream velocity U, and hydraulic diameter D_H) has recently seen increased interest in the microfluidics community, with various strategies for manipulating particles emerging from equilibrium inertial focusing effects and the action of inertial lift forces on particles[6,9]. Fluid streams within a channel can also be deformed by the presence of particles[10], curvature[1], or channel structure[11]. Beebe et al employed a serpentine channel design which efficiently used chaotic advection for mixing fluids[5]. This most common approach used Dean flows, which are induced secondary flows through curved channels, to deform the flow field and increase interfacial area for mixing. Dean flow can also act in superposition with inertial lift forces to reduce the number of stable particle focusing positions[6]. A remarkable example of fluid flow engineering via Dean flow was the creation of a tunable optofluidic lens[2]. But Dean flow is limited in its ability to precisely deform a particular region of the fluid - that is, the entire flow cross-section is manipulated. Particle-induced convection and Dean flow-based schemes -

[a] Department of Mechanical Engineering, Iowa State University, Ames, Iowa.
[b] Department of Bioengineering, University of California, Los Angeles, California.
[†]These authors contributed equally to this work.
[‡]These authors contributed equally to this work.
[*]Corresponding author

while useful for mixing - are difficult to adapt to arbitrary flow sculpting. However, recent developments in geometry-induced secondary flows under finite inertia conditions show promise for novel strategies in microfluid engineering with local flow stream perturbations [11].

The inertial flow around a cylindrical pillar yields secondary flows within a channel cross-section that are localized to the pillar location. The flow deformation around a simple geometry in this inertial regime has been only recently investigated in microfluidic systems, due to the general assumption of Stokes flow creating fore-aft symmetry in flow deformations around an object [6]. Experimental work has made use of flow separation and recirculations that form downstream of obstacles, where the fore-aft symmetry is broken in inertial flow around a geometry placed in the cross section of the fluid stream [1,12]. More recently, microscale cylinders (pillars) have been used to induce secondary flow (separate from the flow separation) and this has been used in tandem with inertial focusing in order to reposition particle streams at high-throughput [13]. A single cylindrical pillar at a variety of lateral positions in a microchannel has been found to induce a set of local fundamental deformations in inertial flow around each pillar [11]. One notable aspect of the inertial flow deformation past a micropillar is the saturation of the turning motion of the flow within a limited distance downstream ($<\sim$4 pillar diameters), which leads to the idea of 'programmability' using pre-computed advection maps for each pillar that are sequentially combined.

With appropriate inter-pillar spacing, pillars placed sequentially in a microfluidic channel can be used to sculpt the fluid into complex new configurations with little new computation [11]. Useful subsequences can be labelled as functions, which can act on fluids alone or concatenate in order to tailor the flow as desired. These 'pillar programs' can open up an extensive library of associated net deformations for future microfluidic applications, such as using shifted fluid streams to improve heat transfer from local hotspots, creating new methods of cell sorting in bio-medical diagnostic 'lab-on-a-chip' devices, enhancing reactions at the interface between fluids by increasing surface area of streams, or changing the cross-section shape of polymer precursors for novel material design [3] .

The complicated phase space on offer per micropillar was initially described through dimensional analysis using three non-dimensional groups: the Reynolds number, the channel aspect ratio h/w, and normalized pillar diameter D/w, where h, w, and D are the channel height, width, and pillar diameter. A dataset of 12,400 transformations with varying pillar diameter, offset, channel height, and flow Reynolds number was generated through distributed HPC resources by Diaz-Montes et al [14]. Changing the pillar position and diameter was observed to have a tremendous effect on the fluid transformation with rich variety of shapes, but the vast number of permutations for sequences from this large dataset made for a daunting phase space of pillar programs. Therefore, a reduced, discretized framework was created for a preliminary foray into understanding what is possible with pillar programming. In order to quickly explore the potential operations allowed by this new method of microfluid engineering, we introduce a lightweight program (uFlow, www.biomicrofluidics.com/software.php) that uses a database of high precision 3D simulations based on single-pillar deformations. We first utilize this easily extensible program to develop intuition with the most fundamental transformations (one to three pillars per sequence), which we eventually combine to design more complex, hierarchically designed transformations. This facilitates rapid investigation of pillar programs without requiring access to high performance computational power. A set of eight fluid transformations, chosen for potential applications in microfluidics, are initially identified as target transformations. This first set of functions helps to illustrate the impact of pillar programming, in addition to signaling the potential complexity of a more complete library of programs. Upon obtaining these novel transformations through this software exploration, we validated the designs by fabricating microfluidic devices and conducting confocal imaging. This work presents specific sequences used to create the target operations, along with some of the basic intuition, tools, and analysis that emerged from the process of finding and understanding these operations.

2 Objective

8 fundamental transformations were identified to be potentially useful in microfluidic applications, and are illustrated in figure 1. Unless specified otherwise, all transformations will be acting on a fluid region of interest ⅕ the width of the channel, and spanning the entire height (figure 1(b)).

- '*Make Concave*' (figure 1(a-i)) and '*Make Convex*' (figure 1(a-ii)) give curvature to the fluid interface for potential microscale optofluidic applications, or as a basic tool for hierarchical design by collapsing or expanding the fluid at different sections.
- '*Tilt*' (figure 1(a-iii)) is among the most basic transformations, and could be utilized for minor deformations in hierarchical design, or as a simple folding mechanism for enhanced mixing.
- '*Stretch*' (figure 1(a-iv)) flattens the fluid region, and its ability to collapse the whole shape could be useful in program design, as many transformations will have weaker effects on the fluid elements near the microfluid channel walls. Control over the thickness of the fluid via *stretch* will aid more complex fiber design.

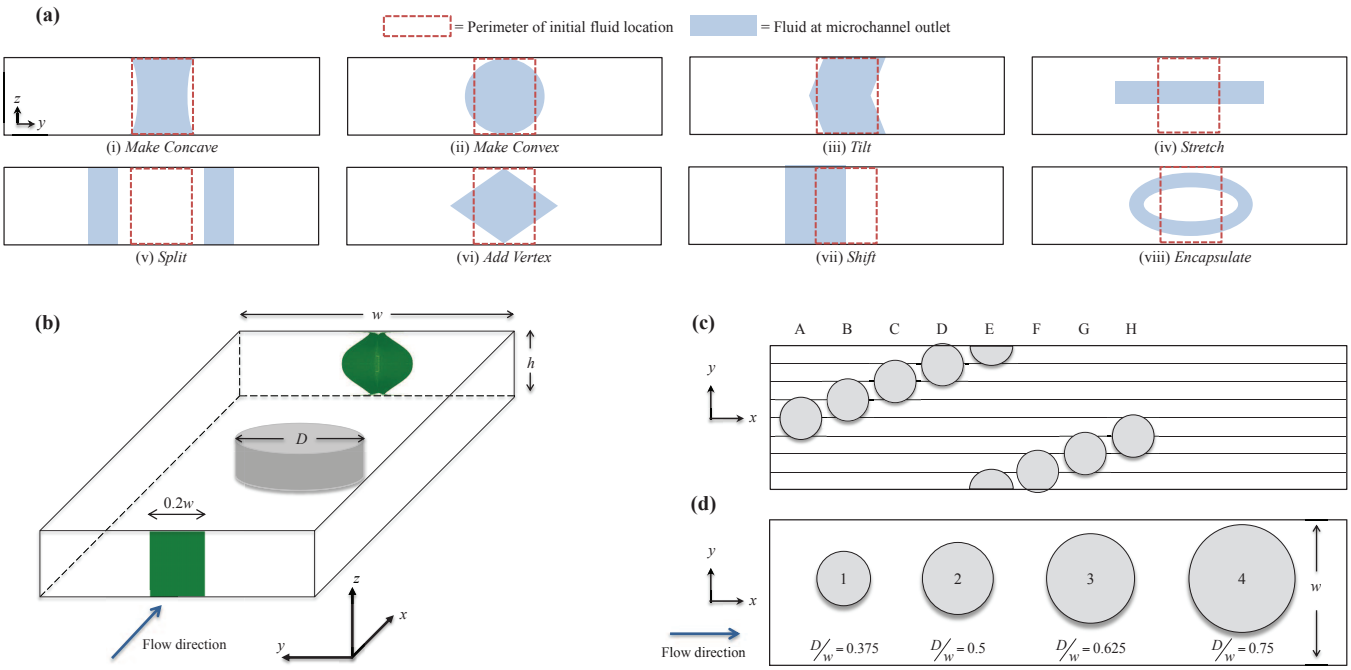


Fig. 1 (a) Objective transformations. (b) Schematic of microchannel, showing a cross-section of the fluid elements at the inlet (middle 20% being tracked), and a sample deformation induced by a single pillar at the center of the channel. (c) Discretized simulation scheme lettered indices for locations space $^1/_8 w$ (A-H), and (d) pillar diameters shown with indices 1-4, labelled with true pillar diameter and drawn to scale.

- '*Split*' (figure 1(a-v)) has applications for fiber design and could lead to simultaneous engineering of separate streams in the microfluid channel.
- '*Add Vertex*' (figure 1(a-vi)) stands apart from most of the transformations due to its sharp vertices at the midsection of the fluid. Such angles could help define a more robust interlocking structure, or hint to more complex polygonal transformations.
- '*Shift*' (figure 1(a-vii)) is a very powerful manipulation of the fluid flow, with potential to isolate and further engineer fluid regions of interest. A program that could laterally shift streams will be a boon to mixing, heat transfer, solution reactions, and other far-reaching applications.
- '*Encapsulate*' (figure 1(a-viii)) seeks to envelope a fluid stream with the primary, central stream. It could be used to create a polymer sheath, induce coaxial solution reactions, or for general mixing applications.

3 Method

3.1 Computational

The phase space for this initial study is limited to manipulating the non-dimensional pillar diameter, $^D/_w$, and lateral location, $^y/_w$. The position was defined by eight equally spaced offsets in a microchannel ($^h/_w = 0.25$), while pillar diameters were restricted to $^D/_w = 0.375$, $^D/_w = 0.5$, $^D/_w = 0.625$, $^D/_w = 0.75$. The velocity field for each pillar case was computed by solving the Navier-Stokes equations using the finite element method, incorporating streamline-upwind/Petrov-Galerkin (SUPG)[15] and pressure-stabilizing/Petrov-Galerkin (PSPG)[16] terms to stabilize the numerical solution. No-slip boundary conditions are applied on the lateral walls and the surface of the pillar. Due to symmetry of the flow field, the simulation is only performed on the top half of the channel with a mirror boundary condition of velocity components on the centerline of the channel. We assume that the fluid is incompressible and at a steady state, and that the velocity profile is fully developed at the inlet. The pressure drop between the inlet and outlet is fixed and is determined to match the flow rate. The inlet velocity profile and pressure drop are calculated with the method used in[17], where the inlet velocity profile is predicted by solving the 2D cross-section flow with a guessed pressure gradient from the Hagen-Poisseuille law, and then corrected by linearly adjusting the pressure gradient to give the correct flow rate. For each configuration, fluid parcels were tracked through their respective velocity field to produce an advection map, which describes the displacement of each parcel as it travels down the channel. Stream deformation is characterized by marking those fluid parcels that belong

to a stream of interest, and looking up the motion of those particles in the advection map. Advection maps are concatenated by using a parcel's deformed location to lookup the displacement in the following map, and summing displacements down the pillar sequence. The result is a new advection map comprising of the net effect of the entire pillar sequence.

3.2 uFlow

A lightweight utility "uFlow" was created (and is freely available online[¶]) that composes advection maps in real-time using the graphical processor (GPU) available on most modern consumer computing hardware. The software presents a simple graphical user interface that allows the user to place pillars into a straight channel and mark regions of the flow using a pallette of colors. Feedback is provided immediately in the form of cross-sectional views of the fluid between each pillar and at the end of the channel.

In order to provide realtime feedback, uFlow takes advantage of some properties of advection maps that make them amenable to the highly parallelized computing provided by the GPU. Briefly, the pre-computed advection maps for each pillar are loaded as OpenGL floating point textures, with x, y, and z displacements stored in the blue, red, and green channels, respectively. A render-to-texture framebuffer object of the same size as the advection maps is created to store the output. A custom fragment shader is executed for each "pixel" in the framebuffer, corresponding to a single fluid parcel in the advection map. The fragment shader performs two fast texture lookups using the GPU's texture fetching hardware, and combines two advection maps into a single map corresponding to the net parcel displacement. The result is stored into a texture, and can be used as the input advection map for a subsequent pillar. In this way, the net displacement for each parcel is accumulated as it travels down the channel.

For visualization, a custom vertex shader performs a texture lookup into the advection map, and uses this information to deform particle locations. The deformed particles are rendered onto the user's screen, and optionally exported to a high-resolution png. This entire loop occurs in realtime on typical laptop hardware.

3.3 Fabrication

The microfluidic devices for verifying transformations were fabricated using conventional soft photolithography. The microchannels with sequence of pillars were designed to be 200 μm by 50 μm, and standard photolithography was utilized to produce corresponding mold from a silicon master spin-coated with KMPR 1050 (MicroChem Corp.). Sylgard 184 Elastomer Kit (Dow Corning Corporation) was used to replicate the polydimethylsiloxane (PDMS) devices from the mold. PDMS base was mixed with a curing agent with a ratio of 10:1, and poured into the molds placed in the petri dishes. The petri dishes were put in a vacuum to remove bubbles and then in an oven until fully cured. The PDMS devices were peeled from the mold and punched using a pin vise (Technical Innovations, Inc.) to create inlet and outlet holes. The PDMS device and a thin glass slide were activated by air plasma (Plasma Cleaner, Harrick Plasma) and bonded together to enclose the microchannel. To help visualizing the channel, Rhodamine B (Sigma-Aldrich) was infused into the channels to permeate the PDMS layer, and washed after one night.

[¶] www.biomicrofluidics.com/software.php

3.4 Confocal Imaging

The confocal images were obtained using a Leica inverted SP1 confocal microscope at the California NanoSystem Institute. For each transformation, the bonded microfluidic device was taped tightly on the stage of microscopy and its three inlets were connected to three syringes set on three separate syringe pumps (Harvard Apparatus PHD 2000) by PEEK tubing (Upchurch Scientific Product No. 1569). The middle syringe included fluorescent isothiocyanate dextran 500kDa (5 μM, Sigma-Aldrich) dissolved in deionized water to help with visualization of the deformed stream, and the other two were filled only with deionized water. The volume flow rate of each pump is proportional to area of the flow stream at the microchannel inlet, with a total volume flow rate of 150 μL/min. The images were taken perpendicular to the flow direction at a downstream location of at least four times the pillar diameter from the last pillar. Six images were taken and averaged to get a final image for each case to avoid random noise.

4 Results

Figure 2(a-h) shows pillar sequences found to engineer the fluid into each objective output (not shown to scale). There are two indices for each pillar (above the channel schematic and within the pillars). The naming convention for pillar program schematics is as follows: the lettered index above the figure represents the pillar's position in the channel, with eight possible positions (see figure 1(c)). The numbered index placed inside of the pillar denotes its diameter, with four numbers corresponding to four diameters: $1 = 0.375w$, $2 = 0.5w$, $3 = 0.625w$, and $4 = 0.75w$ (see figure 1(d)). $Re = 20$ for all conditions, and the channel aspect ratio $h/w = 0.25$.

We divide the pillar programs into two categories: fundamental and hierarchically designed transformations. The fundamental transformations are created using simple programs with three or fewer pillars, and can be leveraged in sequence

in order to create the more complex hierarchically designed transformations.

5 Discussion

5.1 Pillar Sequence Concatenation

The exploration of pillar sequences and a subsequent search for target transformations was enabled by an understanding of how transformations can be used in sequence to appropriately sculpt flow. Because of the deterministic mapping of fluid elements upstream to downstream of the pillar sequences, the sequences themselves can be 'concatenated' end to end to create more complex programs. We identified four classes of concatenation, presented in order of simplicity:

- stacking
- recursion
- mirroring
- shaping

Stacking operations simply repeat the location and diameter of the prior pillar exactly, as seen in figure 3(a). It is typically seen to create a more exaggerated transformation of the previous pillar. A recursive operation is like stacking, but instead of a single pillar, a subsequence of pillars is repeated, as seen in the first pillar program of figure 3(b-i). A sequence of two pillars at positions H and B is repeated, thereby eliciting a more pronounced vertex on the left side of the midsection. Mirroring builds on recursion, except each location of the repeated subsequence is "mirrored" across the microchannel's longitudinal centerline, as demonstrated in figure 3(b-ii). Mirroring can be useful creating symmetric transformations from asymmetric functions, allowing shapes to have more complex edges. A clear example of this is the overall *add vertex* transformation: mirroring is used to form vertices, while each pillar deformation is based on secondary flows that contain no obvious sharp angles. Finally, shaping is simply an arbitrary concatenation of any subsequence to a different one, provided it is not stacking, recursive, or mirroring. An example is seen in figure 3(b-iii), where the known effect of *make convex* is used to create the net shape of *add vertex*. Shaping is a critical tool for discovering new transformations by building on a foundation of fundamental operations, for the user can attempt to sculpt a fluid packet in a way defined by an operation already seen, thus further defining the hierarchy of pillar programming.

5.2 Analysis of *Make Concave*/*Make Convex*

Make concave is formed by a basis of two half-pillars at the channel wall of diameters $D/w = \{0.375, 0.5, 0.625\}$ (figure 4). A diameter of $0.75w$ induces sharp corners in the fluid geometry, and is not effective in creating a smooth concave shape. Changing the pillar diameter changes the curvature of the fluid element, with a varying radius of curvature across the middle ⅓ of the contour as plotted in figure 4(b). Note the change in behavior with pillar diameters larger than $D/w = 0.375$; the 'longer' sequences tended to force more fluid to the edges of the channel, and created a more linear contour at the convex shape's midsection, thus increasing the radius of curvature for the majority of the shape. While it seems that the largest space to work while maintaining a truly convex shape is with $D/w = 0.375$, mass diffusion effects may be magnified by difficulty manufacturing smaller diameter pillars. Figure 2(a) shows a poor reproduction of the numerical prediction even with the modest pillar size of $D/w = 0.625$, suggesting the need for incorporating a diffusion model into the computational framework.

Make convex is a qualitative counterpart in output and program to *make concave*, and primarily uses a single pillar with diameters $D/w = \{0.375, 0.5, 0.625\}$. $D/w = 0.75$ results in an irregular hexagon at the output and therefore not satisfying the target operation requirements. Tunable convex and concave fluid shapes could provide a straight-channel alternative to Dean flow for the application of optofluidics, in addition to novel polymer shapes, or fluid focusing techniques. The average change in radius of curvature per number of pillars of similar diameter used is shown in figure 4(d). Again, it should be noted that the curvature shown is an average of the curvature for the middle ⅓ of the contour, as the profile is parabolic in nature, and becomes linear near the edge.

5.3 Analysis of *Stretch*/*Split*

The *stretch* transformation is accomplished by placing pillars on the center position of the channel. Though larger diameters can stretch the fluid element more rapidly - therefore requiring fewer pillars for a desired output - it is possible to overshoot the goal of a stretching deformation and move into the *split* transformation. *Stretch* is another example of how a transformation can be found through different pillar sequences. Figure 5(a) shows how a 5 pillar sequence with pillars of $D/w = 0.5$ (figure 5(a-i)) can be effectively recreated using a 3 pillar sequence with pillars of $D/w = 0.625$ (figure 5(a-ii)). Attempting to create this same transformation via 3 pillar sequence with $D/w = 0.75$ results in the beginning of the *split* transformation (figure 5(a-iii)), but still meets the goal of the *stretch* transformation. The variety of sequences for similar transformations shows the richness of the phase space, in addition to emphasizing the future need for numerically optimizing for minimal pillar sequences in order to mitigate diffusion effects. Figure 5(b) illustrates the effect of increasing the pillar diameter on the change in fluid width, per number of pillars in the sequence. The *split* transformation (figure 2(e)) is an exam-

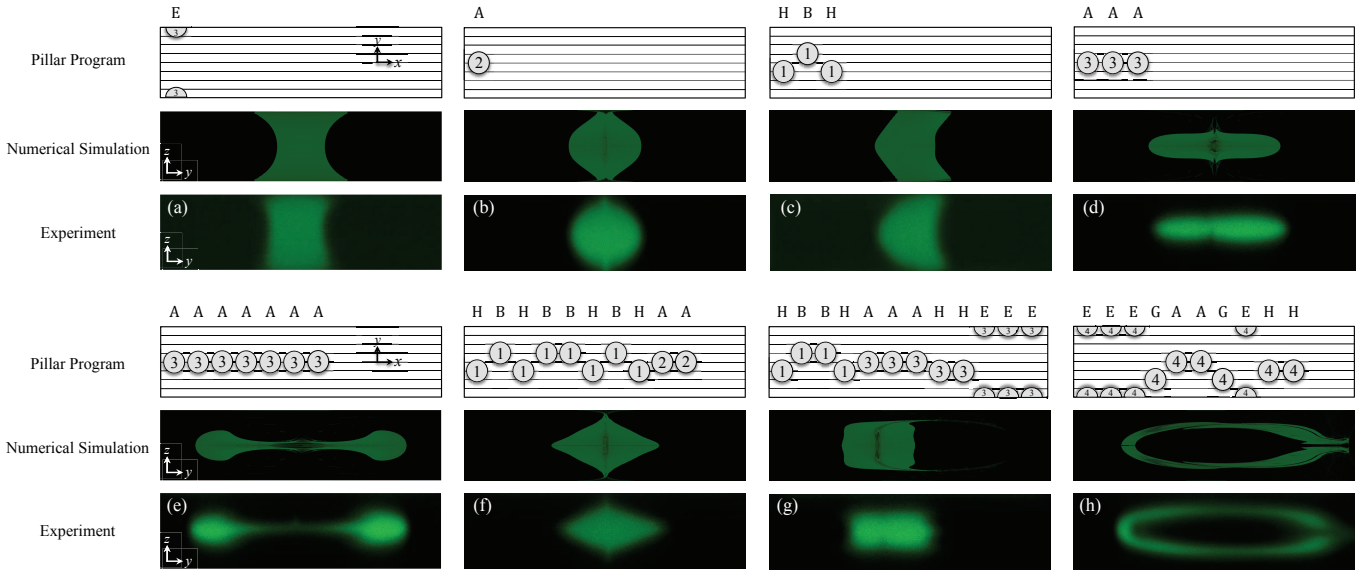


Fig. 2 Fundamental transformations '*Make Concave*' (a), '*Make Convex*' (b), '*Tilt*' (c), '*Stretch*' (d), and Hierarchical transformations '*Split*' (e), '*Add Vertex*' (f), '*Shift*' (g) and '*Encapsulate*' (h). The figures show the pillar sequence schematic on top, the numerical prediction as an output from uFlow in the middle, and the experimental validation on the bottom.

ple of how pillar program recursion can produce an entirely new operation. Placing at least seven pillars of $D/w = 0.625$ results in a separation of the fluid element into two roughly circular elements with diminishing tails on either side of the channel. Adding an eighth pillar begins to distort the separated elements until they begin to wrap back into the center of the channel, beginning a potential mixing scheme.

5.4 Analysis of *Tilt*

The *tilt* program shown in figure 2(c) creates a 38° tilt, with an internal angle of 104°. *Tilt* forms the basis for several hierarchical operations, but there are very few ways to create *tilt* with the reduced framework. The sequence uses pillar diameters of $D/w = 0.375$, but moving to $D/w = 0.5$ removes the vertex in the transformation. Similarly, moving outward in lateral placement also removes the vertex. This motivates the future analysis of a larger dataset with fine-grained pillar diameters and locations, which would allow for sensitivity analysis to determine the degree of error allowed in manufacturing pillar sequences.

5.5 Analysis of *Add Vertex*

Different programs were found to arrive at the *add vertex* transformation, but the sequence shown in figure 3(b-iii) contains the sharpest vertices. There are two primary components to this sequence: the creation of an irregular decagon with vertices protruding at the channel centerline, and the use of the *make convex* program to complete the final shape. The irregular decagon is formed by mirroring a modified 38° *tilt* sequence that adds a vertex to one side of the fluid structure, which results in two vertices symmetric about the channel's y-axis, as shown in figure 3(b-ii). The *make convex* program, 2 pillars with $D/w = 0.5$, is concatenated onto the previous eight pillars to finish the transformation. An alternate route is to decrease the mirrored sequence from four to two pillars. This creates a 5 pillar program, but alters the output edges to become more smooth and convex (see figure 6(b)).

5.6 Analysis of *Shift*

The initial strategy for shifting a fluid element was to begin with a *stretch* program and then force the fluid to either side of the channel with pillars to the left or right of the centerline. Thin 'tails' are formed in the most basic *shift* sequences, as seen in figure 6(a-i). These tails are a natural consequence of the secondary flow at the center of the channel, which - coupled with the no-slip boundary condition - results in an incomplete pinching effect at the wall/fluid interface. To marginalize this effect, the *add vertex* program was used to 'pre-treat' the collapsed fluid element, therefore minimizing the fluid at the wall/fluid interface that would end up creating the tails. Several methods were found for the *shift* transformation, with lateral migration of the central column varying from 10% to 20% of the full channel width (see figure 6(a-ii)), and varying degrees of 'squareness' in the final output. The basic transformation as originally outlined seeks to preserve the shape at the

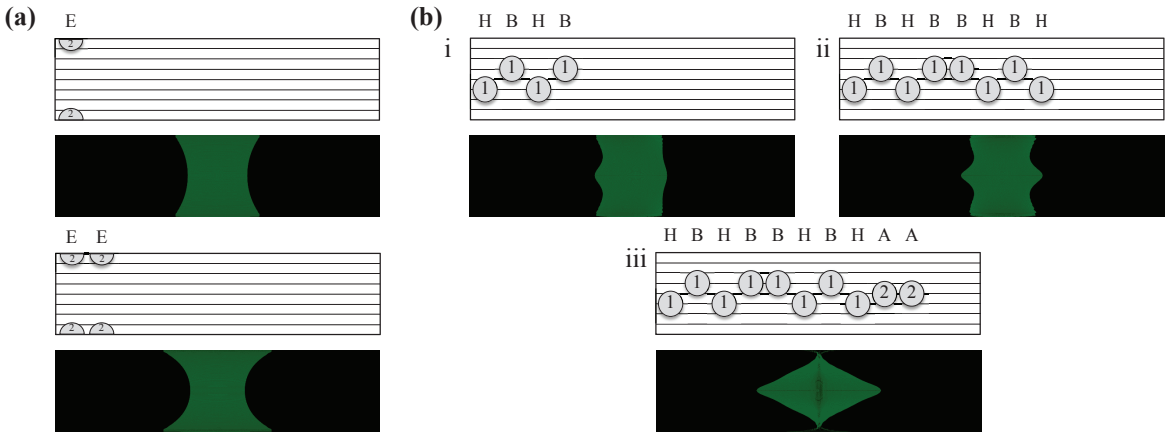


Fig. 3 (a) Stacking demonstration with *make concave*. (b) Creation of *add vertex* via recursion (i), mirroring (ii), and shaping (iii).

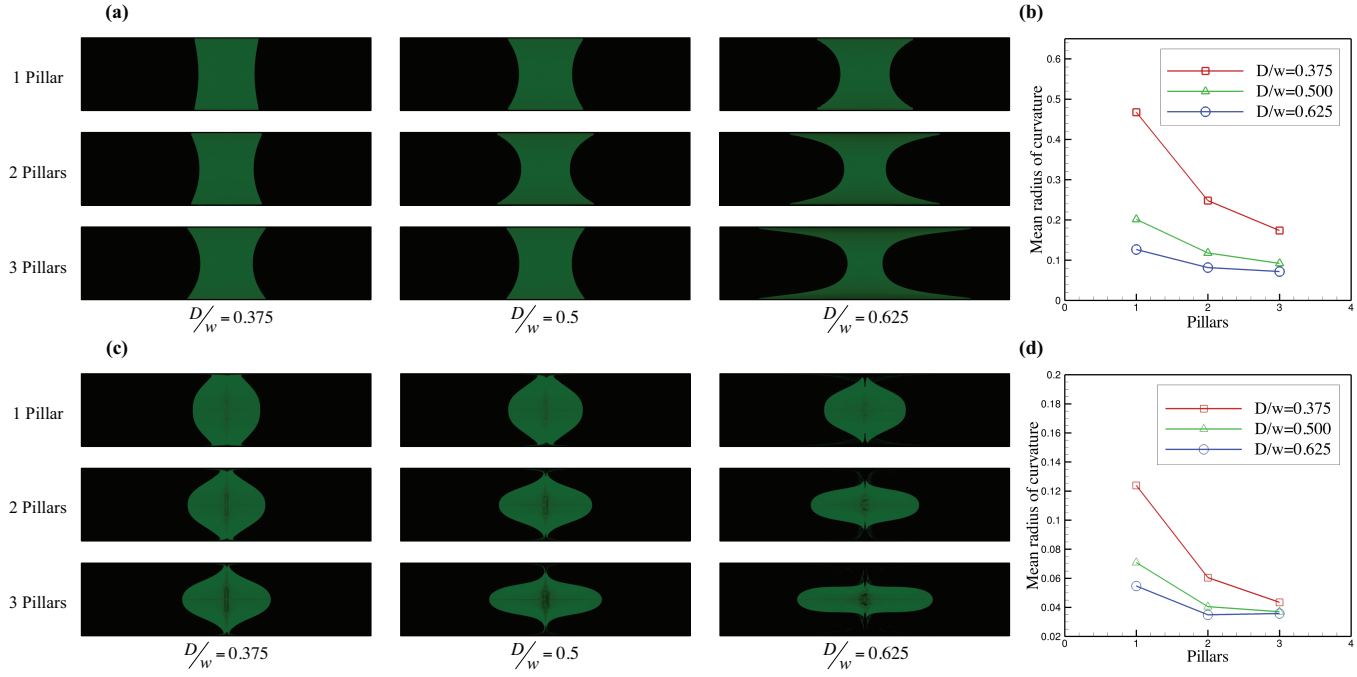


Fig. 4 Illustrations of (a) *make concave* and (c) *make convex* per pillar type through 3 pillars, and changes in mean radius of curvature vs. number of similar pillars in sequence for (b) the *make concave* and (d) *make convex* operations.

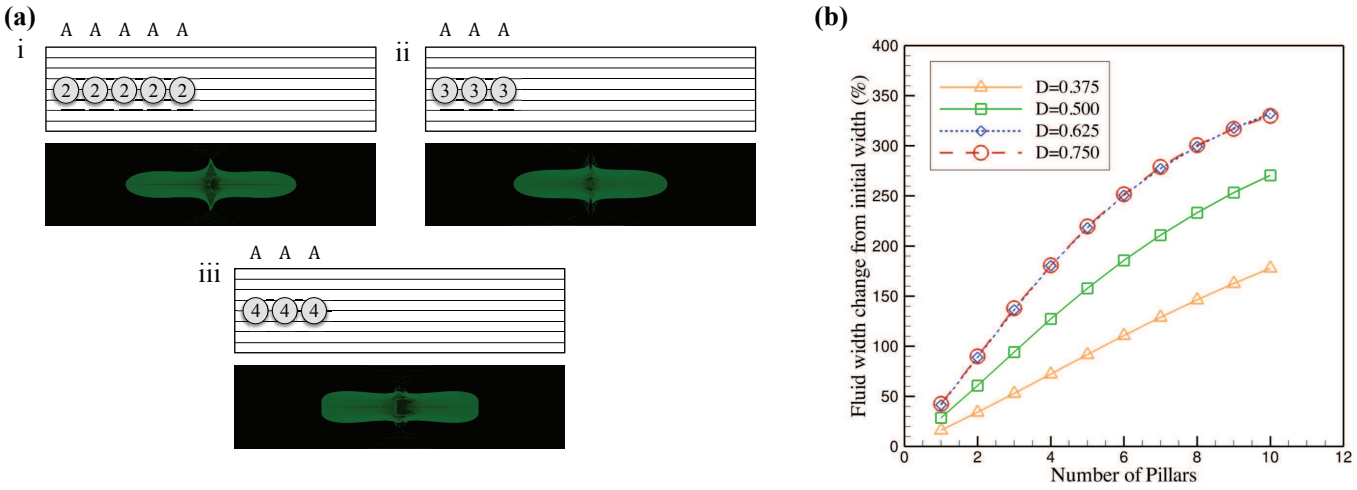


Fig. 5 (a) Multiple routes to similar *stretch* transformations. Note that three pillars of $^{D}/_{w} = 0.625$ achieves roughly the same output as five pillars of $^{D}/_{w} = 0.5$. Moving to a $^{D}/_{w} = 0.75$ begins the *split* transformation sooner, in addition to flattened ends. (b) (Please view in color) The effect of changing diameter and increasing numbers of central pillars on the stretched width of the central fluid element. Note that $^{D}/_{w} = 0.75$ and $D = 0.625$ are functionally identical for this effect.

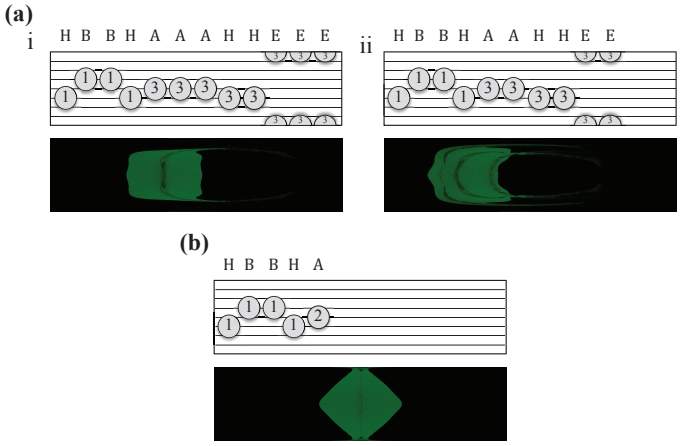


Fig. 6 (a) 10% *shift* (i) and 20% *shift* (ii). Note that the program for 20% *shift* is concatenated onto the 10% sequence, for a total of 22 pillars. (b) Alternative 5 pillar approach to *add vertex*, demonstrating the same recursion/mirroring/shaping methods as the 10 pillar version.

microchannel inlet, and the programs shown demonstrate the best outcome to this effect.

6 Conclusions

We are able to identify a set of fundamental fluid transformations using rationally chosen sequences of pillars. This is made possible by quickly prototyping different pillar sequences in silico, and identifying those sequences that have the desired effect on the fluid shape. The identified transformations have potential applications in optofluidics, shaped fiber and particle fabrication, and mixing. The transformations also provide some intuition on the types of fluid deformations that can be performed using pillars in a channel.

The framework utilized on this study, although applied here only to pillars positioned in a straight channel, can be used to analyze the flow deformation resulting from the concatenation of arbitrary geometries, provided that the fluid is allowed to fully develop in between. Other geometries for consideration include channel expansion regions, curved segments, walls, asymmetric pillars, and channel bifurcations. The analysis can also be expanded to include a range of Reynolds numbers and channel aspect ratios, which were held fixed for this study.

Presently, the effect of diffusion on the fluid is ignored. Large numbers of pillars - and therefore longer channels - along with more complex fluid deformations will enhance mass diffusion further downstream, resulting in a blurred departure from uFlow's predicted transformation. Accordingly, mass diffusion will limit the size of structures that can be formed using this method, though diffusion of species across streamlines may be important for many applications. The current framework can be extended to account for diffusion by applying a simple local diffusion model to a fluid parcel as it traverses the channel. Techniques for enabling this analysis in real-time are under investigation.

Although the present study only follows fluid parcels as they traverse the channel, it is possible in principle to apply the same analysis to finite-sized particles suspended in the fluid flow, which may cross streamlines. Although it is more ex-

pensive to compute advection maps for finite-sized particles, this computation is performed offline, and does not affect the speed of the analysis.

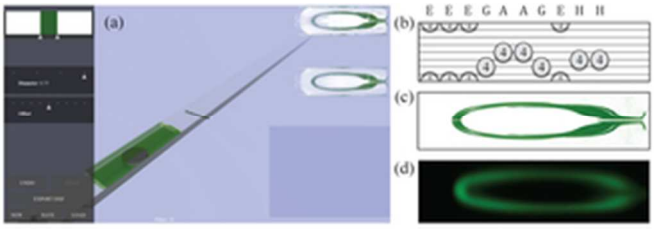
Building off this initial work, future designers of continuous flow fluidic systems should be able to achieve a desired output behavior without significant rounds of experimental trial and error, paving the way for systems of increased complexity to address biomedical, materials fabrication, and industrial heat and mass transport problems.

7 Acknowledgements

This research is supported in part by the National Science Foundation through XSEDE resources provided by TACC under grant number TG-CTS110007, and supported in part by NSF-1306866, NSF-1307550, and NSF-1149365.

References

1 A. P. Sudarsan and V. M. Ugaz, *Proceedings of the National Academy of Sciences of the United States*, 2006, **103**, 7228.
2 X. Mao, J. R. Waldeisen, B. K. Juluri and T. J. Huang, *Lab Chip*, 2007, **7**, 1303–1308.
3 J. K. Nunes, C.-Y. Wu, H. Amini, K. Owsley, D. Di Carlo and H. A. Stone, *Advanced Materials*, 2014, n/a–n/a.
4 A. D. Stroock, S. K. W. Dertinger, A. Ajdari, I. Mezic, H. A. Stone and G. M. Whitesides, *Science*, 2002, **295**, 647–651.
5 R. H. Liu, M. A. Stremler, K. V. Sharp, M. G. Olsen, J. G. Santiago, R. J. Adrian, H. Aref and D. J. Beebe, *Microelectromechanical Systems Journal of*, 2002, **9**, 190–197.
6 D. Di Carlo, *Lab on a Chip*, 2009, **9**, 3038–3046.
7 P. B. Howell, Jr., D. R. Mott, S. Fertig, C. R. Kaplan, J. P. Golden, E. S. Oran and F. S. Ligler, *Lab Chip*, 2005, **5**, 524–530.
8 D. A. Boyd, A. R. Shields, P. B. Howell and F. S. Ligler, *Lab Chip*, 2013, **13**, 3105–3110.
9 G. Segré and A. Silberberg, *Nature*, 1961, **189**, 209.
10 H. Amini, E. Sollier, W. M. Weaver and D. Di Carlo, *Proceedings of the National Academy of Sciences*, 2012, **109**, 11593–11598.
11 H. Amini, E. Sollier, M. Masaeli, Y. Xie, B. Ganapathysubramanian, H. a. Stone and D. Di Carlo, *Nature communications*, 2013, **4**, 1826.
12 D. Chiu, *Analytical and Bioanalytical Chemistry*, 2007, **387**, 17–20.
13 A. J. Chung, D. Pulido, J. C. Oka, H. Amini, M. Masaeli and D. Di Carlo, *Lab Chip*, 2013, **13**, 2942–2949.
14 J. Diaz-Montes, Y. Xie, I. Rodero, J. Zola, B. Ganapathysubramanian and M. Parashar, *Computing in Science Engineering*, 2014, **PP**, 1–1.
15 A. N. Brooks and T. J. Hughes, *Computer methods in applied mechanics and engineering*, 1982, **32**, 199–259.
16 T. E. Tezduyar, S. Mittal, S. Ray and R. Shih, *Computer Methods in Applied Mechanics and Engineering*, 1992, **95**, 221–242.
17 R. Jaeger, J. Ren, Y. Xie, S. Sundararajan, M. Olsen and B. Ganapathysubramanian, *Applied Physics Letters*, 2012, **101**, 184102.



We develop a user-friendly program to identify fundamental fluid transformations in inertial fluid flow using micropillars, with experimental validation.
27x9mm (300 x 300 DPI)