Efficient Simulation of Droplet Merging in Channel-based Microfluidic Devices

Gerold Fink*, Florina Costamoling*, Philipp Ebner* and Robert Wille†‡
*Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
†Chair for Design Automation, Technical University of Munich, Germany
‡Software Competence Center Hagenberg GmbH, Austria
{gerold.fink, florina.costamoling, philipp.ebner}@jku.at, robert.wille@tum.de
https://www.cda.cit.tum.de/research/microfluidics

***Abstract*—Channel-based microfluidic devices, often in the form of so-called *Lab-on-a-Chip* (LoC), have a broad range of applications in domains such as biology, chemistry, medicine, etc. Many of these applications rely on merging of droplets, e.g., in order to trigger some kind of reaction inside the droplets. However, the design process of LoCs is, in general, still in its infancy and mostly relies on simplifications, assumptions, as well as the expertise of the designer – making this process rather error-prone and frequently resulting in a "trial-and-error" approach. Simulation tools can help in this regard. While *Computational Fluid Dynamics* (CFD) tools can simulate the merging of droplets, their complex setup and computational efforts limit their applicability to rather small components and do not allow simulations of larger microfluidic devices. Instead, considerations on the so-called *one-dimensional model* (1D model) offer a more abstract and, hence, computationally much faster simulation. However, currently there are no simulators based on the 1D model available that support the merging of droplets – severely restricting the applicability of such simulators. In this work, we address this problem by proposing a concept for droplet merging based on the 1D-model and implementing these ideas on top of an already existing 1D-simulator. The resulting simulator (which is made publicly available as part of the *Munich Microfluidics Toolkit* (MMFT)) eventually allows for the efficient simulation of channel-based microfluidic devices where droplet merging is an essential part.**

I. Introduction

Microfluidics is an emerging technology and allows to shrink bulky and expensive laboratory equipment to a single microfluidic chip, which are therefore often called *Lab-on-a-Chip* (LoC, [1], [2], [3]). This miniaturization allows to integrate, automate, and parallelize experiments and comes with many advantages, such as reduced sample and reagent volume, higher throughput, shorter reaction times, etc. Especially channel-based microfluidic devices, where droplets are transported through closed micro-channels by a so-called continuous phase, have a broad range of applications in domains such as biology, chemistry, medicine, etc. [4], [5].

Additionally, a lot of these devices rely on the merging of droplets, which is a crucial operation for controlling droplets and reactions [6]. More precisely, droplet merging is needed to initiate and terminate a reaction, as well as for droplet dilution and reagent dosing [7], [8], [9]. The process is particularly important for the miniaturization of multi-phase reaction and separation processes [10] and is also used in single-cell pairing, nano-particle synthesis, and multi-step multiplexed biochemistry [7].

Despite these broad applications, the design process of LoCs is still in its infancy and frequently done by hand. For this, designers rely on simplifications, assumptions, and their experience, which makes this task rather error-prone and often results in a "trial-and-error" approach in which the suitability of a design gets often not tested before an actual fabrication of the corresponding device is available. This frequently causes many and rather long debugging loops since possible errors are detected rather late in the design and fabrication process. Simulation tools can help here, as they allow for validating a design in early stages. Corresponding simulators can be utilized for design automation tasks, early design explorations, and for validating the functionality of a design prior to fabrication.

However, currently no simulation tool exists that allows to *efficiently* simulate the merging of droplets, which makes the design process of such devices very tedious and slow. Although *Computational Fluid Dynamics* (CFD) tools (such as OpenFOAM [11], COMSOL Multiphysics [12], or Ansys [13]) allow to simulate the merging of droplets, they are also very costly in terms of computational power and require complex setups [14]. Hence, they are rather limited, for example, when it comes to simulations of larger microfluidic devices or when quick design explorations should be made [15].

On the other hand, simulators exist [16], [17], [18], [19] that exploit the so-called one-dimensional (1D) analysis model [20], which is a rather abstract model but has the benefit that it is computationally fast. As a result, these simulators are also applicable for larger microfluidic networks and do not require complex setups, which is handy during quick design explorations. Unfortunately, none of these 1D-simulators support the merging of droplets thus far and only allow to determine the positions and velocities of droplets. In this work, we address this problem, by proposing a method that abstracts the complex droplet merging process [21] and incorporates it into the 1D-model. This abstraction might not be as accurate as CFD simulations (or the real physical world), but is sufficient enough when no detail view of droplet merging is required anyway. Hence, this finally allows to efficiently simulate droplet merging and can greatly support designers during the design process of such microfluidic devices.

The remainder of this work is structured as follows: First, we provide the background for 1D-simulations and the corresponding model in Sec. II, which acts as the basis for our proposed approach. In Sec. III we propose the general idea of how to implement our approach and discuss the resulting challenges that need to be tackled. The actual implementation of the method is reviewed in more detail in Sec. IV. Finally, the approach gets validated in Sec. V, before the work is concluded in Sec. VI.

II. Background

In this section, we discuss the one-dimensional (1D) analysis model in more detail and also briefly review the working principles of the 1D-simulator that will act as the basis for the simulator proposed in this work. The resulting simulator is available as an open-source implementation of the MMFT

Droplet Simulator [22], which is part of the *Munich Microfluidics Toolkit* (MMFT) and can be freely accessed under https://github.com/cda-tum/mmft-droplet-simulator.

A. 1D Analysis Model

The 1D analysis model is an ideal choice for efficient simulations of scenarios where a fully developed, laminar, viscous, and incompressible flow occurs [23], i.e., in scenarios with low Reynolds numbers (which is almost always the case in channel-based microfluidic devices). In such scenarios, the relationship inside a channel can be described by *Hagen-Poiseuille's law* [24], [20], i.e.,

$$\Delta P = Q\ R, \tag{1}$$

where Q is the volumetric flow rate, ΔP the pressure drop along the channel, and R the hydrodynamic resistance of the channel. The value of the resistance depends on the dimensions of the channel (i.e., length l, width w, and height h) as well as the dynamic viscosity of the continuous phase μ_c and can be computed (for rectangular channels with $h/w < 1$) by [25]

$$R = 12\left[1 - \frac{192\,h}{\pi^5\,w}\tanh\left(\frac{\pi\,w}{2\,h}\right)\right]^{-1}\frac{\mu_c\,l}{w\,h^3}\ . \tag{2}$$

Additionally, droplets increase the resistance of the channel they are currently in by an amount that is directly proportional to the volume (or length l_d) of the droplet, i.e.,

$$R_d = b\ 12\left[1 - \frac{192\,h}{\pi^5\,w}\tanh\left(\frac{\pi\,w}{2\,h}\right)\right]^{-1}\frac{\mu_c l_d}{w\,h^3}\ . \tag{3}$$

Here, he parameter b represents a factor that indicates how much the droplet increases the resistance of the channel segment it occupies and lies around $2 - 5$ according to [26].

Moreover, Hagen-Poiseuille's law in the microfluidic domain is equivalent to Ohm's law in the electrical domain. This allows to convert a microfluidic network to an equivalent electrical network, which is accomplished by mapping channels to electrical resistances, flow rate pumps to current sources, and pressure pumps to voltage sources. Furthermore, additional droplets in the network are easily considered by increasing the corresponding channel resistances according to Eq. (3).

Having such an equivalent electrical network finally allows to compute the pressure drops and flow rates for each channel by applying well-known methods from the electrical domain. This acts as basis for the 1D-simulator reviewed next.

B. 1D-Simulator

Since the equivalent electrical network allows to compute the flow state of a microfluidic network (i.e., the pressure drops and flow rates of the channels), it can be used to predict the position and movement of droplets. The 1D-simulator does this by an event-driven approach. More precisely, it assumes that the flow state is constant during time and only changes when a new event is triggered. As a result, the movement of the droplets can be predicted between these events. Figure 1 shows a snapshot of such a simulation of a simple network, where the red and blue arrows indicate inlets (at which the droplets get injected) and the black circle an outlet.

Basically, the working principles of the simulator can be broken down into the following six steps:

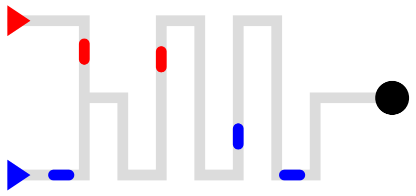


Fig. 1: Snapshot of a 1D-simulation

1) *Initialization:* In this step, the simulator captures the user defined inputs (e.g., the microfluidic network itself, the dimensions of channels, fluid properties, droplet injections, etc.) and generates an initial state, which acts as a starting point for the simulation. After that, the following steps are performed in a loop.
2) *Current flow state:* By converting the microfluidic network into its equivalent electrical network, the simulator is able to compute the current flow state of the network, i.e., pressure drops and flow rates of the channels. With this flow state, it is possible to determine the movements of the droplets until an event occurs.
3) *Next event:* An event is basically an incident that changes the current flow state and can be triggered, e.g., by injecting a droplet, when a droplet switches channels, or when a droplet leaves the network. Therefore, the simulator collects all these events (that could possible be triggered next) inside an event list. Afterwards, the event that will happen next in time is selected as the next event.
4) *Move droplets:* By knowing the time at which the next event will happen, the simulator moves all the droplets accordingly in time with the help of the current flow state from Step 2.
5) *Perform Event:* After the movement of the droplets is conducted, the simulator eventually performs the corresponding event, e.g., changing the channel of a droplet. Due to the changes of the performed event, the flow state is now invalid and, thus, it must be updated in the next iteration at Step 2.
6) *Termination condition:* The simulation stops if a termination condition is reached (e.g., all droplets left the network); otherwise, the simulator continues with Step 2.

Since each system state (droplet positions, flow state, etc.) is stored at every time step, the paths of the droplets can be easily obtained, which eventually allows to capture the behavior of a microfluidic network. The advantages of such an event-driven approach compared to CFD simulations are the much lower computational requirements and the less complex setup (at the cost of reduced accuracy), which allows to simulate even large microfluidic networks in negligible runtime.

III. General Idea and Resulting Challenges

Many channel-based LoCs and microfluidic devices depend on the merging of droplets during their operation, in order to initiate or terminate reactions as well as prepare customized dilutions of reagents [6], [7], [10]. For example, in [9] the success of a method depends on the correct contact time of two reactants, or in [8] droplet merging is used for a reaction scheme of sol-gel reactions. At the same time, the design process for these applications and devices is, in general, still in its infancy. Designs are mostly created by hand while relying on assumptions, simplifications, and the experience of the designer. Simulation tools can help in this regard, as they can be utilized for design automation tools, allow for quick design explorations, or enable the designer to quickly validate the behavior of a microfluidic device. Unfortunately, no efficient simulation method covering droplet merging in channel-based microfluidic devices exists yet. In this section, we briefly discuss the state-of-the-art and provide a general idea how this unfortunate situation can be addressed. Afterwards, we discuss the remaining challenges that need to be overcome to realize this idea.

A. General Idea

According to [21], the merging of droplets is a complex four-stage process: (1) the droplets approach each other, (2) the film of the continuous phase is pushed away, (3) the remaining film breaches, and (4) the actual merging of the droplets happens. Being able to (efficiently) simulate this behavior is key for designing and validating the functionality of microfluidic devices in which droplet merging is an essential part. Unfortunately, all simulation approaches available thus far suffer from severe drawbacks that prevent an efficient consideration of droplet merging in channel-based devices.

More precisely, the 1D-simulator (as reviewed in the previous section) is currently only able to simulate the positions and velocities of the droplets, but does not allow to simulate the droplet merging process in such a detail since the underlying 1D-model is rather abstract. On the other hand, CFD simulation tools such as OpenFOAM [11], COMSOL Multiphysics [12], or Ansys [13] are available that can handle such complex effects. But they are rather limited due to their high computational effort and complex setup [14]. Hence, they can be used for smaller components but are not suited when larger microfluidic devices should be simulated.

In this work, we investigate how to overcome these disadvantages. To this end, we aim at exploring the strength of both approaches while mitigating their weaknesses. More precisely, the 1D-model can make assumptions that certain effects will happen in specific scenarios, which allows to consider these scenarios in a more abstract way and, thus, allows to incorporate it into the 1D-model. Droplet merging is exactly such a scenario, where the simulator can just assume that droplets will merge and mix when their boundaries touch each other. Furthermore, it is often not even necessary to simulate the merging of droplets in such a great detail, but it would be sufficient to know when and where two droplets merge as well as their combined volume or resulting concentration value. This abstraction might not be fully accurate compared to CFD simulations (or the real physical world), but when no detailed view of droplet merging is required anyway, this can greatly benefit from the advantages of the 1D-simulator (i.e., fast computational and setup time even for large microfluidic networks).

Hence, in the following, we discuss how this idea can be integrated into existing 1D-simulation tools and what challenges need to be addressed to realize droplet merging as sketched before under the restrictions of the 1D-model used thus far.

B. Resulting Challenges

Currently, the 1D-model and the simulation approaches based on it do not describe/implement droplet merging at all. In fact, droplet merging is not even considered as a particular action and completely ignored during 1D-simulations thus far.

Example 1. *Let's consider the two droplets d_2 and d_3 (highlighted in green and red, respectively) shown in Fig. 2. As the striped area indicates, the two droplets actually overlap. In reality, this would cause both droplets to merge. However, since this is not covered at all in current 1D-models, the two droplets are handled as different entities that do not interact with each other. If they would move into two different channels again, they would just be separated like they never touched each other. This, of course, makes the current simulation physically inaccurate.*

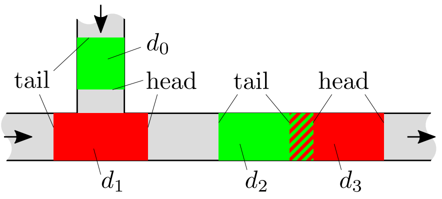


Fig. 2: Current problems during droplet merging

At best, this "approach" can be used to warn the user that the current simulation unveils a possible droplet merging, which, however, cannot be covered properly. For use cases where the designed device does not intend for any droplet merging (or where droplet merging should explicitly be prohibited), this is sufficient. But for all the applications mentioned above, for which droplet merging is key, the current 1D simulators are not applicable.

Using the idea presented in the previous section, this limitation can be removed. In fact, whenever two droplets start to overlap, a corresponding merge process can be triggered. The current representation of droplets in existing models/simulators, however, still pose a challenge. More precisely, the 1D-simulator is currently restricted to work with droplets that have two boundaries, i.e., a head and a tail, as shown in Fig. 2, which has the advantage that this approach is rather simple and also computationally fast. However, this two-boundary approach has serious drawbacks when it comes to droplet merging, as the following example shows.

Example 2. *Let's consider Fig. 2 again; this time, with the two droplets d_0 and d_1. As indicated, droplet d_0 will flow into droplet d_1, and they would eventually merge. If this happens, the resulting (merged) droplet would consist of three boundaries – a two-boundary approach as used thus far is not sufficient anymore.*

Taking this into consideration, realizing the idea proposed above does not only require the addition of a dedicated handling of droplet merging, but also a new multi-boundary representation of droplets.

Then, a droplet merging process can be triggered whenever two boundaries of droplets touch each other. Therefore, it has to be known when and where the boundaries meet, i.e., some kind of detection process is needed. To stay in compliance with the event-driven nature of current 1D-simulators, this detection process should also be implemented by events. That is, a so-called merge-event should be created (and added to the event list) every time two boundaries would meet.

When such a merge-event is triggered, the actual merging of the droplets happens. This is done by removing the corresponding droplets from the network and "inserting" a new droplet at the same place instead. Additionally, it can also happen that droplets are merged, that have different fluids with different parameters. Here, the parameters of the mixed fluid should be derived from the parameters of the other two droplet fluids and, by this, emulate the mixing of the two fluids.

Overall, implementing these ideas on top of an existing simulator should allow the coverage of droplet merging based on 1D-models and, by this, providing an efficient simulation approach for channel-based microfluidic devices heavily rely-

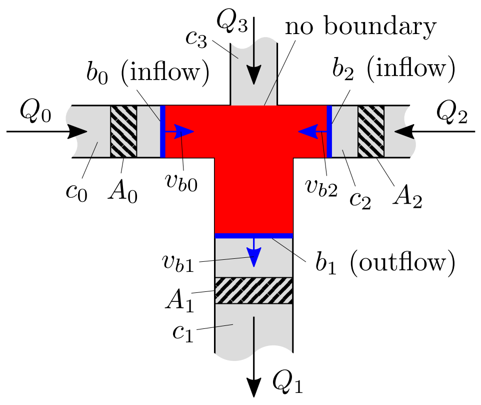


Fig. 3: Multi-boundary approach

ing on that. How to accomplish this is described in more detail in the next section.

IV. Implementation

In order to realize the ideas proposed above, basically three main parts have to be implemented: (1) the multi-boundary approach, (2) merge-events, and (3) the actual merging of droplets. In this section, we describe each of these parts individually and discuss corresponding implementations in more detail.

A. Multi-Boundary Approach

As already mentioned, a droplet is currently represented inside the simulator by two boundaries (namely a head and a tail), which is not sufficient enough when droplet merging should be implemented, and, thus, a multi-boundary approach is needed. In this approach, a droplet is represented inside the 1D-simulator by the channel positions of all boundaries as well as additional channels that are fully occupied by the droplet (this case can happen when channels are shorter than the actual droplet length). With this information, the simulator is able to decide at which channels additional droplet resistances are needed, which finally allows to compute the flow rates inside each channel as described in Sec. II.

In order to predict the movement of a droplet, the flow rate of each boundary is to be determined. Here, it is important that all flow rates are in balance, i.e., boundaries that move away from the droplet center must shift the same volume in a certain time interval as boundaries that move towards the droplet center. If this would not be the case, then the droplet volume would not stay consistent. Hence, some kind of average droplet flow rate is needed that represents the volume shift per time interval of the droplet.

For the two-boundary approach, this is simple, because it is merely the average of the channel flow rates where the two boundaries reside, i.e., the head and the tail move with the same flow rate and guarantee a consistent droplet volume. However, for the multi-boundary approach, the computation of the droplet movement is more complex since there is, in general, no average droplet flow rate which is shared across all boundaries. This means that each boundary has its own flow rate, which does not necessarily have to match with the corresponding channel flow rate. In order to still ensure a consistent droplet volume, a concept for the movement of the droplet is introduced, which involves the following steps.

First, the boundaries are partitioned into inflow and outflow boundaries, where the distinction is made by the corresponding channel flow rates. If the channel flow rate moves the boundary towards the droplet center, then it is considered an inflow boundary; if the boundary moves away from the droplet center, it is considered an outflow boundary. Afterwards, the channel flow rates of all inflow and outflow boundaries are summed up as follows

$$Q_{\text{in}} = \sum_{i \in I_{\text{in}}} Q_i \qquad Q_{\text{out}} = \sum_{i \in I_{\text{out}}} Q_i \ , \tag{4}$$

where I_{in} and I_{out} are the sets of channel indices for the corresponding inflow and outflow boundaries, respectively.

Now an average flow rate can be computed by

$$\bar{Q} = \frac{Q_{\text{in}} + Q_{\text{out}}}{2} \ , \tag{5}$$

which is used in the next step to guarantee a consistent droplet volume. More precisely, the sum of all flow rates of the inflow/outflow boundaries must match $\bar{Q}$, which ensures that the volume shift of the inflow boundaries equals the volume shift of the outflow boundaries.

The final step is then to determine the flow rates of each inflow/outflow boundary. This is done by assigning each inflow/outflow boundary a portion of $\bar{Q}$, which is based on the ratio of the channel flow rate and the total inflow/outflow channel flow rate, i.e.,

$$Q_{\text{b},i} = \frac{Q_i}{Q_{\text{in}}} \, \bar{Q} \qquad\qquad i \in I_{\text{in}} \tag{6}$$

$$Q_{\text{b},i} = \frac{Q_i}{Q_{\text{out}}} \, \bar{Q} \qquad\qquad i \in I_{\text{out}} \ . \tag{7}$$

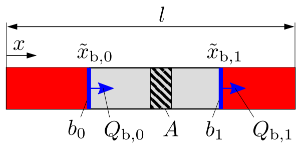
In other words, the higher the flow rate inside a channel, the higher the flow rate of the boundary.

Example 3. *Let's consider the droplet (red) in Fig. 3 with the three boundaries b_0, b_1, and b_2. Each channel c_i has its own flow rate Q_i and cross section A_i (i indicating the channel index), where we assume that the values for the channel flow rates are $Q_0 = 5\,\mathrm{nL\,s^{-1}}$, $Q_1 = 12\,\mathrm{nL\,s^{-1}}$, $Q_2 = 3\,\mathrm{nL\,s^{-1}}$, and $Q_3 = 2\,\mathrm{nL\,s^{-1}}$. Since Q_0 and Q_2 move the boundaries b_0 and b_2 towards the droplet center, they are considered as inflow boundaries, while b_1 is marked as an outflow boundary. Hence, the sets of indices for the inflow/outflow boundaries can be defined as $I_{in} = \{0, 2\}$ and $I_{out} = \{1\}$. As a result, the total channel flow rates of all inflow/outflow boundaries can be determined by $Q_{in} = 8\,\mathrm{nL\,s^{-1}}$ and $Q_{out} = 12\,\mathrm{nL\,s^{-1}}$, while the average flow rate yields $\bar{Q} = 10\,\mathrm{nL\,s^{-1}}$. With these values the boundary flow rates can now be computed by $Q_{b,0} = 6.25\,\mathrm{nL\,s^{-1}}$, $Q_{b,1} = 10\,\mathrm{nL\,s^{-1}}$, and $Q_{b,2} = 3.75\,\mathrm{nL\,s^{-1}}$. As it can be seen now, the flow rates of the inflow boundaries add up to the flow rates of the outflow boundaries, i.e., $Q_{b,0} + Q_{b,2} = Q_{b,1}$, which is critical for a consistent droplet volume.*

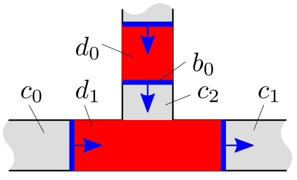
With the multi-boundary approach implemented in the simulator, the next section now deals with merge-events that allow to detect when and where boundaries meet.

B. Merge-Events

During the simulation, droplet merging should happen when two droplets touch each other. In compliance with the event-driven simulation process reviewed in Sec. II-B, the merging of droplets will also be implemented as so-called merge-events. There are two different events, where merging

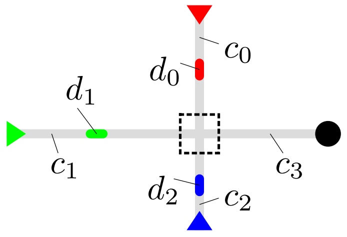


(a) Channel

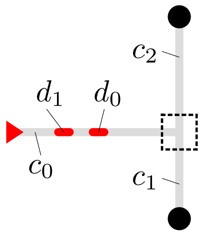


(b) Bifurcation

Fig. 4: Merge-events



(a) Network A



(b) Network B

Fig. 5: Merge scenarios

of droplets can occur, namely a *Channel-Merge Event* (CME) and a *Boundary-Merge Event* (BME).

Channel-Merge Event: In this scenario, the boundaries of different droplets (which might or might not merge) are inside a single channel, as shown in Fig. 4a, where the channel has a specific length l and cross section A. The positions of the boundaries at a certain time t can then be determined by

$$x_{\text{b},0}(t) = \tilde{x}_{\text{b},0} + \frac{Q_{\text{b},0}}{A}\, t \qquad x_{\text{b},1}(t) = \tilde{x}_{\text{b},1} + \frac{Q_{\text{b},1}}{A}\, t \;, \tag{8}$$

where $\tilde{x}_{\text{b},0}$ and $\tilde{x}_{\text{b},1}$ are the current boundary positions inside the channel. By combining these two equations, the merge time t_m and merge position x_m (i.e., when and where the boundaries meet) can then be derived by

$$t_\text{m} = \frac{\tilde{x}_{\text{b},1} - \tilde{x}_{\text{b},0}}{Q_{\text{b},0} - Q_{\text{b},1}}\, A \tag{9}$$

$$x_\text{m} = \tilde{x}_{\text{b},0} + \frac{Q_{\text{b},0}}{A}\, t_\text{m} = \tilde{x}_{\text{b},1} + \frac{Q_{\text{b},1}}{A}\, t_\text{m} \;. \tag{10}$$

Based on the values of t_m and x_m, a CME is created or not. More precisely, when the conditions $t_\text{m} \geq 0$, $t_\text{m} \neq \infty$, and $x_\text{m} \leq l$ are satisfied, then a CME is created and added to the event list (cf. Sec. II-B). Other values would indicate that both boundaries have the same velocity and would never meet ($t_\text{m} = \infty$), the boundary b_1 is faster than b_0 ($t_\text{m} < 0$), or the merging would happen outside of the channel ($x_\text{m} > l$).

Bifurcation-Merge Event: This scenario is shown in Fig. 4b and (may or may not) occur when two droplets meet, while one of them (droplet d_1) currently flows through a bifurcation. This event is tightly coupled with a so-called boundary-head event, which is triggered when a boundary switches the channels and currently acts as an outflow boundary (i.e., it is the head of a droplet). For example, if the droplet d_1 would not be present, then the boundary b_0 of the droplet d_0 would just flow into one of the channels c_0 or c_1. However, if a droplet is present inside the bifurcation, then this event would not happen, since the droplet d_0 would merge with the droplet d_1 and, thus, a BME must be performed instead. Implementation-wise, this is done the same way as described, i.e., before a boundary-head event is created, it is checked if a droplet is present at the corresponding bifurcation. If yes, then a BME is created instead and added to the event list.

However, while these events only consider when and where a merging of droplets happens, the simulator has to perform additional tasks to ensure that the droplets are actually merged, which is discussed next.

C. Merging of Droplets

Once a merge event (i.e., a CME or BME) is triggered as the next event in the event list and should be performed, the simulator conducts the following tasks: First a new droplet is created which contains all the boundaries (and fully occupied channels) of both droplets, i.e., it is "injected" at the same position as the two droplets that are going to merge. Important here is that the boundaries that triggered the merge event are not included since they are not present anymore due to the merging.

Additionally, when the two droplets consist of different fluids, then a new fluid has to be generated, which is a mixture of the two droplet fluids. Currently, this is done by averaging the fluid parameters based on the droplet volumes V_0 and V_1 as follows,

$$p_\text{new} = \frac{V_0\; p_0 + V_1\; p_1}{V_0 + V_1} \;, \tag{11}$$

where p stands for a specific fluid parameter like density, viscosity, concentration value, etc.[1]

Overall, the multi-boundary approach, the merge-events, and the actual merging of the droplets allow to implement the successful merging of droplets inside the 1D-simulator.

V. Validation

In order to demonstrate that the proposed approach indeed allows for an (efficient) simulation of channel-based microfluidic devices relying on droplet merging at the 1D-model, we evaluated and eventually validated the resulting simulator using the two networks shown in Fig. 5. These networks basically cover all different scenarios where droplet merging can happen, i.e., within a bifurcation or inside a channel. In the two figures, the colored arrows indicate an inlet where droplets get injected at the channel inputs, and each color stands for a different fluid with different parameters. Furthermore, the black circle represents an outlet (i.e., a place where droplets leave the network). The dashed square indicates the area where the droplet merging occurs.

In order to resemble real-world structures, the networks are asymmetrical, i.e., the channels have different lengths l and input flow rates Q as stated in Tab. I, but they share the same width $w = 100\,\mu\text{m}$ and height $h = 30\,\mu\text{m}$. Additionally, each droplet has its own volume V and concentration value C, as shown in Tab. II. Note that the concentration value is just a representative and can easily be replaced/extended by other parameters that the fluids might have and that are relevant when two fluids are mixed, e.g., a concentration of a specific reagent, a certain color of the fluid, etc.

In the first network, shown in Fig. 5a, a single droplet is injected at each of the three channel inputs. Then these droplets merge at the bifurcation, and finally flow towards the outlet. After simulating this network with the proposed approach, the resulting droplet (arriving at the outlet) has the volume and concentration value as stated in the last row of Tab. II. As it can be observed, the resulting droplet has the

[1]Note that possible effects from chemical reactions are not considered here.

TABLE I: Dimensions

	Network A				Network B		
	c_0	c_1	c_2	c_3	c_0	c_1	c_2
l [µm]	1000	1500	750	1000	1500	750	1000
Q [µL/min]	1.8	2.1	1.5	5.4	1.8	1.03	0.77

TABLE II: Droplets

	Network A		Network B	
Droplets	V [nL]	C [%]	V [nL]	C [%]
d_0	0.450	10	0.450	40
d_1	0.495	50	0.495	40
d_2	0.405	80		
d_{res}	1.350	45.67	0.945	40

combined volume of all three injected droplets and also has the mixed concentration value, according to Eq. (11).

The second network shown in Fig. 5b has a single input channel and two output channels, which means that the input flow rate gets divided into two smaller output flow rates. As a result, when a droplet reaches the bifurcation and its head boundary flows into one of the output channels, the whole droplet automatically gets slower, since an average flow rate is computed according to Eq. (5). Hence, when two droplets get injected at the input channel in quick succession, the second droplet d_1 will outrun the droplet d_0 at the bifurcation, i.e., the head boundary of droplet d_1 will touch the tail boundary of d_0 inside the channel c_0 and, thus, the droplets will merge. When simulating this scenario with the proposed approach, the droplets indeed merge as described and the resulting droplet flows towards the output channel c_1 (since it has the higher flow rate compared to c_2). While the volume of the resulting droplet is the sum of the injected droplets, the concentration value stays the same since the two droplets consist of the same fluid.

Overall, the results obtained by simulating the two networks (which, again, cover all scenarios in which droplet merging can happen) are as expected. Furthermore, all simulation runs can be conducted in negligible runtime (i.e., less than a second). This confirms that the approach proposed in this work indeed allows for accurately simulating droplet merging in channel-based microfluidic devices while still remaining at the highly abstract (and, hence, computationally efficient) 1D-model – satisfying the purpose and premise of this work.

VI. Conclusion

In this work, we proposed an approach that allows to simulate droplet merging in channel-based microfluidic devices in an efficient manner. This is accomplished by abstracting the complex process of droplet merging and incorporate it into the 1D-model. While this does not allow a detailed view of the actual merge process (as opposed to CFD simulations), it provides sufficient information on the nature of the resulting droplet, i.e., the position where the merging takes place, the resulting volume, etc. Especially for larger networks (where CFD simulations quickly render infeasible), the proposed approach benefits from the negligible runtime and easy setup. The resulting simulator is available as an open-source implementation of the MMFT Droplet Simulator [22], which is part of the *Munich Microfluidics Toolkit* (MMFT) and can be freely accessed under https://github.com/cda-tum/mmft-droplet-simulator.

Acknowledgments

This work has partially been supported by the FFG project AUTOMATE (project number: 890068) as well as by BMK, BMDW, and the State of Upper Austria in the frame of the COMET Programme managed by FFG.

References

[1] G. M. Whitesides, "The origins and the future of microfluidics," *Nature*, vol. 442, no. 7101, pp. 368–373, 2006.

[2] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, "Microfluidic Lab-on-a-Chip platforms: requirements, characteristics and applications," *Chemical Society Reviews*, vol. 39, no. 3, pp. 1153–1182, 2010.

[3] P. S. Dittrich and A. Manz, "Lab-on-a-chip: microfluidics in drug discovery," *Nature Reviews Drug Discovery*, vol. 5, no. 3, p. 210, 2006.

[4] H. Gu, M. H. Duits, and F. Mugele, "Droplets formation and merging in two-phase flow microfluidics," *International Journal of Molecular Sciences*, vol. 12, no. 4, pp. 2572–2597, 2011.

[5] S.-Y. Teh, R. Lin, L.-H. Hung, and A. P. Lee, "Droplet microfluidics," *Lab on a Chip*, vol. 8, pp. 198–220, 2008.

[6] G. Christopher, J. Bergstein, N. End, M. Poon, C. Nguyen, and S. L. Anna, "Coalescence and splitting of confined droplets at microfluidic junctions," *Lab on a Chip*, vol. 9, no. 8, pp. 1102–1109, 2009.

[7] A. Sattari, P. Hanafizadeh, and M. Hoorfar, "Multiphase flow in microfluidics: From droplets and bubbles to the encapsulated structures," *Advances in Colloid and Interface Science*, vol. 282, p. 102208, 2020.

[8] V. Chokkalingam, B. Weidenhof, M. Krämer, W. F. Maier, S. Herminghaus, and R. Seemann, "Optimized droplet-based microfluidics scheme for sol–gel reactions," *Lab on a Chip*, vol. 10, no. 13, pp. 1700–1705, 2010.

[9] K. Liu, H. Ding, Y. Chen, and X.-Z. Zhao, "Droplet-based synthetic method using microflow focusing and droplet fusion," *Microfluidics and Nanofluidics*, vol. 3, no. 2, pp. 239–243, 2007.

[10] K. Wang, Y. Lu, L. Yang, and G. Luo, "Microdroplet coalescences at microchannel junctions with different collision angles," *AIChE Journal*, vol. 59, no. 2, pp. 643–649, 2013.

[11] C. J. Greenshields, "Openfoam user guide," *OpenFOAM Foundation Ltd, version*, vol. 3, no. 1, 2015.

[12] C. Multiphysics, "Comsol multiphysics user guide (version 4.3 a)," *COMSOL, AB*, pp. 39–40, 2012.

[13] I. Ansys, "Ansys fluent theory guide," *Canonsburg, PA*, p. 794, 2011.

[14] P. Ebner and R. Wille, "CFD for Microfluidics: A Workflow for Setting Up the Simulation of Microfluidic Devices," in *Euromicro Conference on Digital System Design (DSD)*, 2023.

[15] M. Takken and R. Wille, "Simulation of Pressure-Driven and Channel-Based Microfluidics on Different Abstract Levels: A Case Study," *MDPI Sensors*, 2022.

[16] A. Grimmer, X. Chen, M. Hamidović, W. Haselmayr, C. L. Ren, and R. Wille, "Simulation before fabrication: a case study on the utilization of simulators for the design of droplet microfluidic networks," *RSC Advances*, vol. 8, pp. 34 733–34 742, 2018.

[17] A. Grimmer, M. Hamidović, W. Haselmayr, and R. Wille, "Advanced simulation of droplet microfluidics," *Journal on Emerging Technologies in Computing Systems*, 2019.

[18] A. Biral, D. Zordan, and A. Zanella, "Modeling, simulation and experimentation of droplet-based microfluidic networks," *Trans. on Molecular, Biological, and Multi-scale Communications*, vol. 1, no. 2, pp. 122–134, 2015.

[19] G. Fink, P. Ebner, M. Hamidovic, W. Haselmayr, and R. Wille, "Accurate and Efficient Simulation of Microfluidic Networks," in *Asia and South Pacific Design Automation Conference*, 2021.

[20] K. W. Oh, K. Lee, B. Ahn, and E. P. Furlani, "Design of pressure-driven microfluidic networks using electric circuit analogy," *Lab on a Chip*, vol. 12, no. 3, pp. 515–545, 2012.

[21] C.-X. Zhao and A. P. Middelberg, "Two-phase microfluidic flows," *Chemical Engineering Science*, vol. 66, no. 7, pp. 1394–1411, 2011.

[22] G. Fink, F. Costamoling, and R. Wille, "MMFT Droplet Simulator: Efficient Simulation of Droplet-based Microfluidic Devices," *Software Impacts*, 2022.

[23] F. Jousse, G. Lian, R. Janes, and J. Melrose, "Compact model for multi-phase liquid–liquid flows in micro-fluidic devices," *Lab on a Chip*, vol. 5, no. 6, pp. 646–656, 2005.

[24] H. Bruus, *Theoretical microfluidics*. Oxford university press Oxford, 2008, vol. 18.

[25] M. J. Fuerstman, A. Lai, M. E. Thurlow, S. S. Shevkoplyas, H. A. Stone, and G. M. Whitesides, "The pressure drop along rectangular microchannels containing bubbles," *Lab on a Chip*, vol. 7, no. 11, pp. 1479–1489, 2007.

[26] T. Glawdel and C. L. Ren, "Global network design for robust operation of microfluidic droplet generators with pressure-driven flow," *Microfluidics and Nanofluidics*, vol. 13, no. 3, pp. 469–480, 2012.