

CFD for Microfluidics: A Workflow for Setting Up the Simulation of Microfluidic Devices

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

philipp.ebner@jku.at, robert.wille@tum.de

<https://www.cda.cit.tum.de/research/microfluidics>

Abstract—Microfluidics is an emerging technology that is expected to revolutionize biochemical experiments and reduce the need for unwieldy laboratory equipment. Motivated by that, the design automation community has spent considerable efforts and proposed numerous methods on automating the design process for corresponding microfluidic devices. For many of them, however, it often remains unclear whether the (automatically) generated design indeed works as intended and/or satisfies its purpose. Simulations, e.g., based on *Computational Fluid Dynamics* (CFD) can help here as they allow for studying the behavior of microfluidic devices without the need for actual fabrication. However, the setup and configuration of CFD simulations is time-consuming and requires extensive expertise—some of the reasons why it is hardly utilized in the design automation community yet. In this work, we propose a workflow that improves upon this state of the art by automating intermediate steps and highlighting the parameters that are relevant to the specific use case. We demonstrate the feasibility of the proposed workflow for fluid mixing in channel-based microfluidics.

Index Terms—Microfluidics, Simulation, Computational Fluid Dynamics, OpenFOAM

I. INTRODUCTION

Microfluidics is an emerging technology that deals with the manipulation of fluids at micro- to picoliter-scales [1]. By carrying out analytic operations on a single chip, microfluidic devices are able to replace bulky and expensive laboratory equipment. Therefore, such devices are known as *Labs-on-a-Chip* (LoCs), which enable users to conduct a wide variety of biochemical experiments and have found widespread usage in medicine, biology, and chemistry [2]–[4]. Due to their tiny feature size and capacity to process small amounts of reagents, microfluidic devices are attractive candidates for point-of-care tests and other comparable diagnostics [5].

With the increasing complexity of LoCs, the design of corresponding microfluidic devices or components has evolved into a rather challenging task. Motivated by that, researchers and engineers from the design automation community put considerable efforts into the development of methods aimed at automating this design process. Over time, this resulted in numerous design tools for the various microfluidic platforms (see, e.g., [6]–[15]). Unfortunately, the resulting tools have often not been tested in practice, and for many of them, it remains unclear whether the (automatically) generated design indeed works as intended and/or satisfies its purpose. While fabricating prototypes and conducting actual experiments on the resulting device is certainly one way to verify the feasibility

of a generated design, actual fabrication is costly and comes with the downside of time-consuming iteration cycles [16].

Fortunately, simulations, e.g., based on *Computational Fluid Dynamics* (CFD, [17]), provide a viable alternative. CFD simulation software such as *OpenFOAM* [18] or *COMSOL* [19] is capable of carrying out numerical calculations in order to predict the behavior of fluids by employing certain physical simulation models. Such simulations take a variety of parameters into account (e.g., the geometric shape of the considered device or the physical properties of the fluids involved) and, hence, often allow for a good assessment, or at least close approximation, of whether the considered use case does what it is supposed to do.

However, most of this work focuses on applications such as aerospace engineering [20], heat exchangers [21], or coating processes [22]. While some related work on CFD simulation for microfluidics exists [23], it often focuses on isolated parts of microfluidic devices (e.g., T-junctions [24] or droplet traps [25]) or certain locally restricted behaviors of fluids on a microscale level (e.g., the flow regime in channel bends [26]). As of now, there is no *systematic* workflow for the setup and configuration of CFD simulations which would allow for validating the design of microfluidic devices—certainly some of the reasons why the potential of those simulations has been hardly utilized in the design automation community thus far.

The main reasons for the absence of such a workflow mostly stem from the following three issues:

- 1) *Broad Expertise Needed*: The CFD simulation setup is comprehensive and requires expertise in CFD and experience with certain software frameworks—effectively creating high entry barriers.
- 2) *No Generic Solution*: There is no established one-for-all approach to CFD in microfluidics that readily supports and can be applied to many use cases of the domain.
- 3) *Tedious Tool Access*: Many CFD frameworks are commercial and/or feature their own interfaces that do not interconnect very well.

In this work, we propose a workflow that facilitates the use of CFD in microfluidics by targeting these issues. The goal is to provide an easy setup for prototypical microfluidic use cases and, therefore, remove the high entry barriers of CFD simulations—also allowing direct input from high-level design automation solutions to these simulations [6]–[15]. This is achieved by automating repetitive tasks and restricting the

configurations to the necessary parameters that are relevant to a particular use case. To this end, intuitive *Graphical User Interfaces* (GUIs) are provided which support the end-user in specifying the geometric shape of the microfluidic device as well as defining the physical properties and other important parameters.

In order to demonstrate the feasibility of the proposed workflow, we considered the simulation of a channel-based microfluidic device for fluid mixing as a proof of concept. More precisely, we set up a CFD simulation for a so-called *gradient generator* [27]–[29] that is able to generate different mixing ratios of fluids.

The remainder of this work is structured as follows: Section II describes the process of setting up CFD simulations for microfluidics. Then, Section III identifies the shortcomings of the state-of-the-art approach. In Section IV, the proposed workflow is described in detail. Afterwards, Section V shows the results that can be obtained with this workflow, discusses its advantages, and outlines how it can be generalized to other use cases. Finally, the paper is concluded in Section VI.

II. SETTING UP CFD SIMULATIONS FOR MICROFLUIDICS

In this section, we describe the process of setting up a CFD simulation for microfluidic devices. First, we elaborate on mesh generation (using the *SALOME* platform [30], which provides various meshing algorithms). Afterwards, we give an overview of the simulation setup itself (using the tool *OpenFOAM* [18], which is an open-source CFD simulation software). As a representative case for a microfluidic device, we consider a so-called gradient generator [27]–[29], i.e., a device that generates specific mixing ratios of fluids. More precisely, we describe the setup for simulating the design of a gradient generator with two inlets and three outlets as sketched in Fig. 1. This is a design which could have been (automatically) generated by a corresponding design automation tool (such as [14]), but for which no automatic support for simulation exists yet.

A. Mesh Generation

The first step of every simulation setup is the generation of a suitable mesh of the considered microfluidic design that can be used for physical simulation. The corresponding process usually involves the following five major steps:

1) *Two-dimensional Device Design*: The first task is the specification of the geometric shape of the microfluidic device in two dimensions. Using, e.g., *SALOME*'s GUI, this can be done by graphically specifying vertices and line segment connections. Care must be taken to produce a valid design that features no holes and fully specifies the shape.

Example II.1. Fig. 1a shows a picture of the two-dimensional design of the gradient generator with two inlets and three outlets.

2) *Transformation to Three Dimensions*: The two-dimensional sketch of the device needs to be converted to a three-dimensional object. Since we assume upright walls, the two-dimensional shape can be simply extruded, i.e., this is a straightforward task.

Example II.2. The extruded, three-dimensional gradient generator is depicted in Fig. 1b.

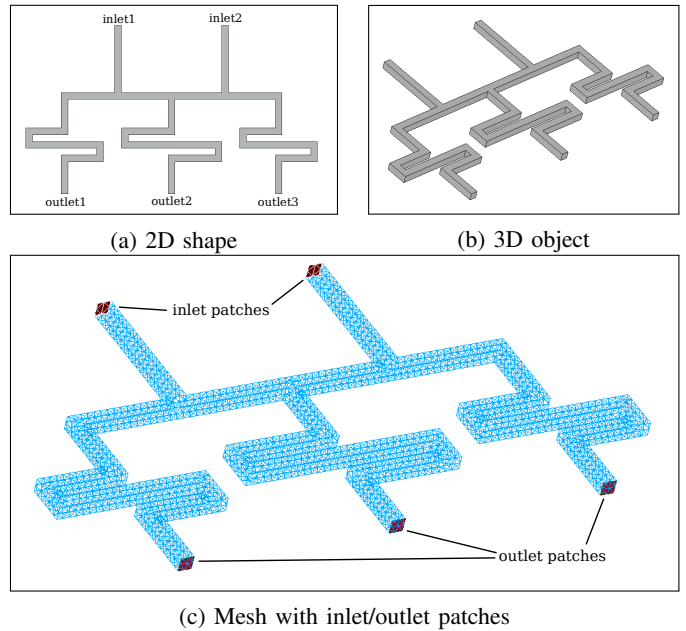


Fig. 1: Setting up a simulation for a gradient generator

3) *Mesh Generation*: The geometric domain then needs to be partitioned into smaller cells that can be used efficiently for CFD simulations. There are a variety of mesh topologies, including hexahedral and tetrahedral meshes, which have six and four vertices in each cell, respectively. Tetrahedral meshes offer several advantages over other mesh types and are thus frequently employed in CFD simulations [31]—leading to a tetrahedral mesh. To this end, an appropriate meshing algorithm must be selected and configured. *SALOME* aggregates a variety of meshing algorithms (most of which are commercial) for various applications. Another critical factor is mesh quality. The meshing algorithm's parameters must be adjusted appropriately (by so-called hypotheses) in order to produce an adequate mesh with well-formed tetrahedrons. Typical hypotheses include a specified maximum cell volume or a certain local length (maximum edge length of cells) [32], [33].

Example II.3. To generate the mesh for the given geometry, we use the *NETGEN* [32] algorithm in this example. For hypotheses, we used a local length criterion of $50\ \mu\text{m}$. The resulting mesh is shown in Fig. 1c.

4) *Inlet/Outlet Identification*: With the geometry of the considered device complete, next, the parts of the boundary have to be identified which constitute inlets or outlets of the device. These parts are called patches.

Example II.4. Recall that, in the considered example, we have two inlets (*inlet1*, *inlet2*) and three outlets (*outlet1*, *outlet2*, *outlet3*). Therefore, we uniquely identify the cross sections of these inlets and outlets as patches (illustrated in Fig. 1c). The remaining boundaries are solid walls. In *SALOME*, this has to be manually selected for each surface and assigned to a group, which is later used in the configuration files of the CFD simulation.

5) *Export*: Finally, the generated mesh is exported so that it can be fed into the CFD simulator. In the case considered here, exporting the mesh in the *UNV* file format is chosen as this

format is also supported by *OpenFOAM* and, therefore, can be directly imported into it.

B. Simulation Setup

Having the mesh of the gradient generator, the actual simulation can be set up. To this end, the following steps have to be carried out:

1) *Mesh Import*: First, the mesh needs to be imported. *OpenFOAM* can work with multiple mesh formats but internally uses its own representation. Fortunately, *OpenFOAM* offers a command-line utility that enables importing *UNV*-files.

2) *Initial Field Values*: Field and boundary values need to be set for the first time step of the simulation, i.e., each cell in the mesh must be assigned values for the first time step. These values may include, e.g., the velocity of the fluids, their hydrostatic pressures, or the α -values indicating the mixing ratio of fluids in the cell. Additionally, boundary conditions have to be specified for patches and walls. The configuration is done in *OpenFOAM*'s own input format, consisting of so-called dictionary files.

Example II.5. For the considered example, the following values have to be set: For the interior, we assume uniform fields at the initial timestep, i.e., we set all values to zero. For the inlets, a fixed value of 10 mm/s is used for the velocity field, a zero-gradient condition is applied to the pressure field, and fixed values for α are introduced, such that *inlet1* and *inlet2* each introduce one fluid phase, with $\alpha = 0$ and $\alpha = 1$, respectively. At the outlets, a zero-gradient condition is applied to the velocity field as well as the α , while the pressure field is set to a fixed value of zero. For the remaining channel walls, we apply a no-slip boundary condition for the velocity field and a zero-gradient condition for the pressure and α fields.

3) *Phase Definitions*: Another important matter is the specification of the physical properties of both fluid phases. For the use case considered here, we restrict the simulation to a laminar flow regime. For each fluid phase, we specify a Newtonian transport model. Additionally, we have to specify the kinematic viscosity ν and density ρ , as well as diffusivity.

4) *Choice of Solvers and Simulation Control*: CFD tools such as *OpenFOAM* typically incorporate different and complementary methods such as the *Finite Volume Method* (FVM) and the *Finite Element Method* (FEM) [18]. Accordingly, the user needs to select and carefully configure them. In the example considered here, we choose *OpenFOAM*'s *twoLiquidMixingFoam* as the general solver, whereas solving schemes for several derivatives have to be configured as well. Finally, we have to specify how the simulation is carried out. This includes start/end time, limits for the Courant number and delta time, as well as formatting options for simulation output.

III. CHALLENGES

Following the steps reviewed in the previous section, a setup for a simulation of a microfluidic device (in this case, the design of a gradient generator) results. This setup can directly be employed in simulation tools such as *OpenFOAM* and can be used, e.g., to validate whether the design works as intended and/or fulfills its purpose. However, although this procedure is well-established and self-evident to any experienced CFD engineer, it is hardly applied in the design automation community yet. In fact, most of the design automation solutions proposed for microfluidics in recent years do *not* utilize the

power of simulation in order to validate the generated designs. This is mainly caused by the following challenges:

1) *Broad Expertise Needed*: First and foremost, designers need substantial expertise in CFD simulations. Choosing appropriate solvers, as reviewed in Section II-B, is a non-trivial task. Even small mistakes or oversights may lead to incorrect or error-prone simulations. Moreover, experience with design automation tools and corresponding meshing algorithms is required in order to generate a suitable mesh for simulation. Finally, knowledge of microfluidics is essential in order to select reasonable boundary conditions, numerical values for fluids, etc. While that expertise is usually available to microfluidic engineers that focus on fabricating corresponding devices, it is often not available to the engineers developing corresponding design automation solutions. Accordingly, further support that guides those engineers through the respective steps is key for closing the gap between design automation and simulation.

2) *No Generic Solution*: Even in a perfect setting where design automation and CFD expertise are covered, the configuration of even a single simulation case is rather time-consuming. All the steps reviewed in Sections II-A and II-B have to be carried out for *each and every* simulation case. While common tools are built to be powerful and flexible, engineers working in the domain of microfluidics are typically interested in only a few distinct use cases, for which no one-for-all approach exists. Developing the necessary expertise to successfully operate these tools is often too time-consuming and, therefore, not feasible.

3) *Tedious Tool Access*: For design and fabrication, a number of established tools are used for parts of the process. However, many of these software programs are commercial and thus not easily accessible to a broad audience. Furthermore, access to these tools is tedious, especially for intermediate steps in the design and simulation process, since almost every piece of software uses different input and output formats. Software libraries that are able to read and write these formats are not always readily available. For example, the configuration files for *OpenFOAM* (cf. Section II-B) are written in a unique format with little library support in common programming languages.

IV. PROPOSED WORKFLOW

The reviews and discussions of the current state of the art as done in the previous sections clearly show that design automation engineers need support in setting up CFD simulations for microfluidic devices (empowering them to validate the results generated by their methods). In this work, we are proposing a workflow that provides this support. The proposed workflow combines and automates repetitive tasks as well as provides *Graphical User Interfaces* (GUIs) to simplify steps such as specifying the geometry and boundary, generating meshes, specifying inlets and outlets, determining physical values, etc. In this section, we describe the general ideas of the proposed workflow as well as the resulting steps in detail—using the gradient generator as a representative use case for a proof-of-concept implementation. Afterwards, Section V discusses the feasibility and benefits of the proposed solution and the extension of the workflow for other use cases.

A. General Ideas

The goal of the presented workflow is to provide an easy and fast way for engineers to conduct CFD simulations for

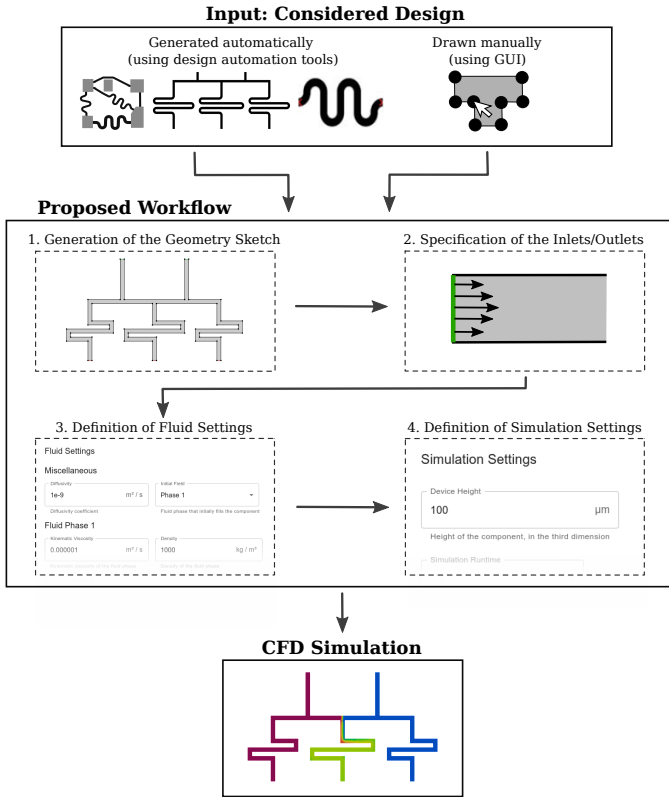


Fig. 2: Proposed workflow

their designs without requiring an extensive background in the subject. To achieve this, we introduce GUIs that guide the user through the specification of the necessary parameters and abstract away the actual CFD configuration by internally using a template case.

The basic idea and major steps of the workflow are illustrated in Fig. 2. The main input for the workflow is the considered design, which is either automatically provided by high-level design automation tools such as [6]–[15] or can be manually drawn by the user using a GUI provided by the proposed workflow. Based on that, the first step of the proposed flow is the generation of the geometric sketch of the design, i.e., the two-dimensional representation of the considered design, including the geometric shape and boundaries in terms of a mesh. Then, the user needs to specify the inlets and outlets of the microfluidic device, determining properties such as the introduced fluid phases and inlet velocities. Afterwards, the user is asked to define the fluid phases, i.e., their physical properties. Finally, the general simulation settings need to be defined, e.g., the runtime of the simulation. The final output is a fully configured simulation case that can be run with *OpenFOAM*.

In the following, a more detailed description of the steps of the proposed workflow is provided.

B. Geometry Sketch

The first task is the specification of the geometric shape of the microfluidic device. A (two-dimensional) sketch of the geometric shape may be generated by high-level design automation, e.g., solutions such as the tools described in [6]–[15]. Therefore, it is easy to incorporate external design automation tools into the proposed workflow. On the other

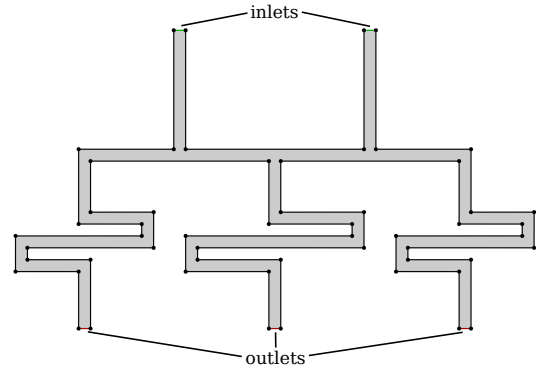


Fig. 3: Sketch of the gradient generator device

hand, the presented GUI includes an interface that allows for simply drawing the device using a pointer or specifying the coordinates of the points. Hence, the two-dimensional representation of the gradient generator can also be drawn manually. An example of that is depicted in Fig. 3.

The actual generation of the mesh is taken off the user’s responsibility by using the external meshing library *Tetgen* [33]. To this end, the two-dimensional sketch can be automatically converted to *Tetgen*’s input format (a three-dimensional piecewise linear complex). Since the considered microfluidic devices have upright walls (e.g., channels with rectangular cross-sections), it is sufficient to simply extrude the two-dimensional design along the third dimension. No additional configuration has to be done by the user, except for specifying a mesh quality parameter (a maximum for mesh cell volumes, cf. Section IV-E). The output is a three-dimensional tetrahedral mesh that is automatically converted to the *OpenFOAM* mesh format.

C. Inlet/Outlet Settings

In the next step, the parts of the device that implement inlets and outlets are defined. In the geometry sketch, the device’s inlets and outlets can be specified by simply clicking on the appropriate boundary edges. To this end, a GUI/context menu as shown in Fig. 4 is provided, which enables the user to edit the properties of each edge (depicted in Fig. 4). Here, the user may choose between a regular channel wall, an inlet, or an outlet. While outlets require no further configuration, each inlet introduces a fluid phase at a certain velocity (in $\mu\text{m/s}$), which can be specified in the same context menu. Additionally, the two fluid phases that are introduced at each inlet can be specified. There is no need for further configuration; the choice of boundary conditions is done automatically and, therefore, this step is substantially simplified for the user.

D. Fluid Settings

The proposed workflow covers the configuration of the fluid and mixing settings as the next step. To this end, a *Fluid Settings* form is provided (depicted in Fig. 5), which makes it possible to directly fill in the values of the physical parameters. The following parameters need to be specified by the user:

- *Diffusivity*: the diffusion coefficient in m^2/s
- *Initial field*: which one of the two fluid phases initially fills the device

Fig. 4: Inlet/outlet settings

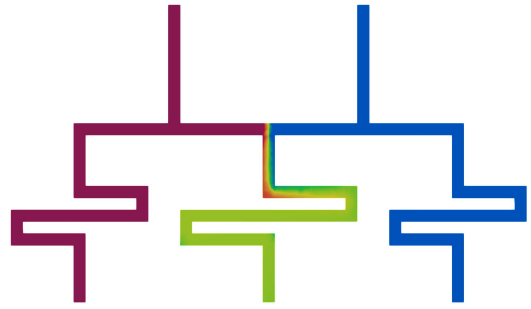


Fig. 7: Simulation result

Fig. 5: Fluid settings

Fig. 6: Simulation settings

Additionally, the physical properties of both fluid phases are specified. Each phase is characterized by two physical parameters:

- *Kinematic viscosity* in m^2/s
- *Density* in kg/m^3

E. Simulation Settings

Some settings are common to all simulations, e.g., determining the runtime of the simulation. To this end, the *Simulation Settings* dialogue form (illustrated in Fig. 6) supports the user in setting these parameters. More precisely, the following parameters are mandatory in order for the simulation to be run:

- *Device height*: the height of the device in the third dimension
- *Simulation time*: for how many seconds the simulation is running
- *Mesh cell max volume*: the maximum allowed volume of each cell in the mesh, as a means of mesh quality control

These general settings conclude the configuration of the simulation case for the gradient generator.

After all configurations are done, the ready-to-run simulation case can be generated by the simple click of a button. The simulation itself is then performed with a local installation of the *OpenFOAM* software.

V. RESULTS AND DISCUSSION

Using the workflow described in the previous section makes it easy to swiftly generate the simulation case for the considered microfluidic device. By providing a template case and intuitive GUIs for the important parameters, we remove the

need for expertise in CFD in general as well as experience with specific software such as *OpenFOAM*. Therefore, the proposed workflow is able to substantially reduce the effort needed for setting up CFD simulations and make them more accessible to the end-user.

For the considered use case (i.e., the simulation of the design of a gradient generator), the result of the correspondingly set up simulation is shown in Fig. 7. The complete simulation, with all timesteps included, could be used to determine the mixing ratios at the outlets and, thus, allows validation of whether the design indeed works as intended and/or satisfies its purpose.

Overall, the introduced proof of concept provides the following benefits compared to the current setup procedure for CFD simulations:

- The cumbersome CFD configuration is abstracted away by using a template case and requiring the user to only supply values that are relevant to the specific simulation case—not only simplifying the process but also reducing the probability of a faulty configuration.
- The time and effort needed to set up a CFD simulation are substantially reduced and, therefore, it becomes more feasible to conduct such simulations.
- The used software, especially *OpenFOAM*, is non-commercial and, therefore, accessible to a large audience.

Moreover, although the proposed workflow has been explicitly developed towards gradient generators (as a representative use case to provide a proof of concept), it can be easily extended to other use cases. In fact, the steps covering the mesh generation are rather general and can be equivalently used for numerous other use cases as well. Only the simulation case setup may require a (slight) adjustment. For this, however, just a revision of the used template (which is tailored to a specific use case) as well as some of the forms in the GUI need to be adapted.

For example, in order to illustrate the adjustments needed to support another use case, let's consider the design of a droplet-based microfluidic device such as that considered in [25]. Here, the goal of CFD simulation is to study the behavior and movement of droplets in microfluidic channels. While, as said, the definition of the geometric shape and, thus, the meshing, work in the same way as for the gradient generator (and numerous other microfluidic devices for that matter), the simulation setup requires the following adjustments:

- 1) *Immiscibility*: Since the droplet phase and continuous phase do not mix in this case, a different solver has to be configured, and diffusion coefficients become obsolete.

- 2) *Droplet injection*: The droplets may be generated inside the devices (e.g., with a T-junction of both phases) during the simulation. Another possibility is inserting single droplets at specific locations in advance, which would require a modification of the GUI as well as an implementation that modifies the initial field. An application for the latter is the simulation of passive droplet switching concepts.

Both adjustments are rather straightforward and showcase that the proposed workflow has the potential to improve the setup of CFD simulations for a large number of microfluidic devices and use cases.

VI. CONCLUSION

In this work, we presented a workflow for an easy and straightforward setup of CFD simulation cases for microfluidic devices. The introduced workflow provides an intuitive GUI and restricts the necessary configuration parameters to the bare minimum, both of which make it substantially easier to setup CFD simulations. Through this, we have particularly reduced the effort to validate whether a given design (generated manually or by design automation methods) works as intended and/or satisfies its purpose. We demonstrated the feasibility of the proposed workflow using gradient generators as a proof of concept use case, but also discussed how easily the approach can be adjusted to support further microfluidic devices. We strongly believe that the workflow proposed in this work encourages the design automation community to validate the methods developed by them and, therefore, use the full potential of CFD simulations in microfluidic applications.

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] E. Samiei, M. Tabrizian, and M. Hoorfar, "A review of digital microfluidics as portable platforms for lab-on-a-chip applications," *Lab on a Chip*, vol. 16, no. 13, pp. 2376–2396, 2016.
- [3] J. Potter, P. Brisk, and W. H. Grover, "Rapid development and optimization of paper microfluidic designs using software automation," *Analytica Chimica Acta*, vol. 1184, p. 338985, 2021.
- [4] 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.
- [5] J. R. Mejia-Salazar, K. Rodrigues Cruz, E. M. Materon Vasques, and O. Novais de Oliveira Jr, "Microfluidic point-of-care devices: New trends and future prospects for ehealth diagnostics," *Sensors*, vol. 20, no. 7, p. 1951, 2020.
- [6] G. Fink, M. Hamidović, W. Haselmayr, and R. Wille, "Automatic design of droplet-based microfluidic ring networks," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2020.
- [7] A. Grimmer, W. Haselmayr, A. Springer, and R. Wille, "Design of application-specific architectures for Networked Labs-on-Chips," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 1, pp. 193–202, 2018.
- [8] T.-M. Tseng, M. Li, D. N. Freitas, T. McAuley, B. Li, T.-Y. Ho, I. E. Araci, and U. Schlichtmann, "Columba 2.0: A co-layout synthesis tool for continuous-flow microfluidic biochips," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1588–1601, 2017.
- [9] W. H. Minhass, P. Pop, J. Madsen, and T.-Y. Ho, "Control synthesis for the flow-based microfluidic large-scale integration biochips," in *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2013, pp. 205–212.
- [10] J. Potter, W. Grover, and P. Brisk, "Design automation for paper microfluidics with passive flow substrates," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*, 2017, pp. 215–220.
- [11] Z. Zhong, Z. Li, K. Chakrabarty, T.-Y. Ho, and C.-Y. Lee, "Micro-electrode-dot-array digital microfluidic biochips: Technology, design automation, and test techniques," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 2, pp. 292–313, 2018.
- [12] J. McDaniel, W. H. Grover, and P. Brisk, "The case for semi-automated design of microfluidic very large scale integration (mvlsi) chips," in *Design, Automation and Test in Europe*, 2017, pp. 1793–1798.
- [13] A. Grimmer, P. Frank, P. Ebner, S. Häfner, A. Richter, and R. Wille, "Meander designer: Automatically generating meander channel designs," *Micromachines – Journal of Micro/Nano Sciences, Devices and Applications*, vol. 9, no. 12, 2018.
- [14] G. Fink, T. Mitteramskogler, M. A. Hintermüller, B. Jakoby, and R. Wille, "Automatic design of microfluidic gradient generators," *IEEE Access*, vol. 10, pp. 28 155–28 164, 2022.
- [15] G. Fink, P. Ebner, and R. Wille, "Comprehensive and accessible channel routing for microfluidic devices," in *Design, Automation and Test in Europe (DATE)*, 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. [Online]. Available: <http://dx.doi.org/10.1039/C8RA05531A>
- [17] D. Erickson, "Towards numerical prototyping of labs-on-chip: modeling for integrated microfluidic devices," *Microfluidics and Nanofluidics*, vol. 1, no. 4, pp. 301–318, 2005.
- [18] H. Jasak, "OpenFOAM: open source CFD in research and industry," *International Journal of Naval Architecture and Ocean Engineering*, vol. 1, no. 2, pp. 89–94, 2009.
- [19] N. Thorne, L. Flores-Olazo, R. Egoávil-Espejo, E. A. Vela, J. Noel, J. Valdivia-Silva, and D. van Noort, "Systematic review: Microfluidics and plasmodium," *Micromachines*, vol. 12, no. 10, p. 1245, 2021.
- [20] M. M. Bhatti, M. Marin, A. Zeeshan, and S. I. Abdelsalam, "Recent trends in computational fluid dynamics," *Frontiers in Physics*, vol. 8, p. 593111, 2020.
- [21] M. M. A. Bhutta, N. Hayat, M. H. Bashir, A. R. Khan, K. N. Ahmad, and S. Khan, "CFD applications in various heat exchangers design: A review," *Applied Thermal Engineering*, vol. 32, pp. 1–12, 2012.
- [22] K. Verma, L. Ayuso, and R. Wille, "Parallel simulation of electrophoretic deposition for industrial automotive applications," in *2018 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, 2018, pp. 468–475.
- [23] M. Takken and R. Wille, "Simulation of Pressure-Driven and Channel-Based Microfluidics on Different Abstract Levels: A Case Study," *MDPI Sensors*, 2022.
- [24] X. Chao, F. Xu, C. Yao, T. Liu, and G. Chen, "CFD simulation of internal flow and mixing within droplets in a t-junction microchannel," *Industrial & Engineering Chemistry Research*, vol. 60, no. 16, pp. 6038–6047, 2021.
- [25] G. Fink, P. Ebner, M. Hamidović, W. Haselmayr, and R. Wille, "Accurate and efficient simulation of microfluidic networks," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, pp. 85–90.
- [26] J. You, L. Flores, M. Packirisamy, and I. Stiharu, "Modeling the effect of channel bends on microfluidic flow," *IASME Trans*, vol. 1, no. 1, pp. 144–151, 2005.
- [27] Q. Shen, Q. Zhou, Z. Lu, and N. Zhang, "Generation of linear and parabolic concentration gradients by using a christmas tree-shaped microfluidic network," *Wuhan University Journal of Natural Sciences*, vol. 23, no. 3, pp. 244–250, 2018.
- [28] X. Wang, Z. Liu, and Y. Pang, "Concentration gradient generation methods based on microfluidic systems," *RSC advances*, vol. 7, no. 48, pp. 29 966–29 984, 2017.
- [29] T. Zhang, J. Meng, S. Li, C. Yu, J. Li, C. Wei, and S. Dai, "A microfluidic concentration gradient maker with tunable concentration profiles by changing feed flow rate ratios," *Micromachines*, vol. 11, no. 3, p. 284, 2020.
- [30] A. Ribes and C. Caremoli, "Salome platform component model for numerical simulation," in *31st annual international computer software and applications conference (COMPSAC 2007)*, vol. 2. IEEE, 2007, pp. 553–564.
- [31] M. W. Bern and P. E. Plassmann, "Mesh generation." *Handbook of computational geometry*, vol. 38, 2000.
- [32] J. Schöberl, "Netgen an advancing front 2d/3d-mesh generator based on abstract rules," *Computing and visualization in science*, vol. 1, no. 1, pp. 41–52, 1997.
- [33] S. Hang, "Tetgen, a delaunay-based quality tetrahedral mesh generator," *ACM Trans. Math. Softw.*, vol. 41, no. 2, p. 11, 2015.