5 Simulations in Microfluidics
Goran Goranović and Henrik Bruus

5.1 Introduction

This chapter shows by example how one can use computer simulations in designing and testing of microfluidic systems. The fabrication of lab-on-chip systems is very complex and requires interdisciplinary research and development. Moreover, the production time in a cleanroom facility of, say, a silicon-based lab-on-a-chip system can be several weeks or even months. Numerical simulations of any given chip design is therefore extremely useful. Not only can they provide a more complete understanding of the fundamental physical and chemical processes of the entire lab-on-a-chip, but they can also be used to develop optimal designs and to minimize the risk of wasting expensive production time on a flawed design.

It is by no means possible to cover or even mention all relevant aspects of simulations in microfluidics in a short overview. Therefore, descriptive explanations will prevail over details in the attempt to create a helpful, “zeroth-order approximation” picture in the mind of the reader.

The structure of this chapter itself reflects how one can deal with simulations by working through various phases: analyzing the fundamental physics and chemistry of a given lab-on-a-chip, choosing the right software and hardware, setting up a simulation, analyzing the sources for numerical errors and uncertainties, and finally evaluating the precision and accuracy of the result. Indeed, the importance of the last point cannot be overrated. It is nearly always possible to generate a result, but is it correct and does it describe the actual device?

5.2 Analyzing physical aspects

When designing new microfluidic devices the designer must have a proper understanding of the physical aspects of the problem. Some of these are:

**Dimensions:** A microfluidic system is a network of fluidic channels and other components such as valves and pumps. It typically occupies an area of one square centimeter, with the typical channel widths on the order of 100 µm. For comparison, a human hair has a thickness of approximately 50 µm and the diameter of red blood cells are about 7 µm. From a practical point of view even smaller dimensions in a lab-on-a-chip are not necessarily advantageous, since the requirement for easier handling as well as measurements can impose some restrictions. However, if the goal is to develop systems for single molecule detection, the sub-micrometer is unavoidable, and one enters the new and exciting field of nanofluidics.

**Geometry:** The basic component in a microfluidic network is a channel. Main features include length, cross-section and surface properties such as
roughness. After a channel is made in a substrate it is covered with a bonded lid that can be made from a different material.

Long channels are usually needed if a reaction of several mixed chemicals is to occur (in laminar flow regime diffusion is the main mixing mechanism and it takes time). Also, when a chemical compound consisting of several components needs to be analyzed, usually by electrical separation techniques, the separation channel is made longer in order to allow for the proper separation of the components. Since microchips are confined to a small area, the extra length is achieved by meander or spiral structures, (Figure 5.1). Such channel bending, however, induces dispersion of a chemical species since molecules travel different distances inside the turn (race-track effect, Figure 5.2). The dispersion lowers the concentration resulting in reduced resolution, i.e. larger overlaps of the concentration peaks, and diminished detection signals. Another common source of the geometrical dispersion are interconnections, such as tubings, between a chip and external liquid reservoirs.

[Figures 5.1, 5.2]

Channels can have different cross-sections depending on the material in which they are embedded. In silicon, the typical profiles are rectangular, or actually slightly trapezoid due to underetching. In polymers, laser beams can produce rectangular, triangular and Gaussian-like shapes. Circular cross-sections as in capillaries are also encountered. In addition, channels can have various degrees of surface roughness depending on the fabrication process. Geometrical features are important since they determine the characteristic resistances of a channel. The hydraulic resistance, a concept used in the uniform flow regime, relates an applied pressure to the corresponding flow rate, while the electrical resistance of a channel filled with a conducting liquid relates an applied voltage to the passing current.

Surface: Chemical groups at the surface of channel walls, such as silanol (SiOH) groups in glass, can react with the ions in an electrolyte solution and create very thin polarized layer. This electric double layer, or Debye layer, has a length scale of 1-10 nm and is responsible for electroosmotic flow (EOF) in an applied electric field. Surface double-layers are characterized by the zeta potential. Different surfaces, such as the lid and the bottom of a channel, can have different zeta potentials. The characteristic EOF velocity depends on the zeta potential through electroosmotic mobility which is largely an empirical value.

Fluid properties: The channels are filled with liquids which can have different properties such as density, viscosity, electrical and thermal conductivity, diffusion coefficients, surface tension etc. In many cases there are two types of liquids in the system – a buffer and a sample. The buffer is usually the transporting liquid resisting the changes in pH value which can for example be induced close to electrodes or in chemical reactions. A sample is a liquid of interest which needs to be analyzed. Both buffer and sample are usually placed on the chip through separate reservoirs, with the buffer finally surrounding the sample. A sample confined to a specific region is called a sample plug. In
separation science, the sample consists of several chemical species such as a positively charged, a negatively charged and a neutral. As mentioned above, the sample plug lowers its initial concentration by dispersing through the system. In addition to the geometrical constrictions, dispersion is caused by velocity profiles, such as the parabolic in the pressure driven flows, and by diffusion.

Fluids such as blood or gels are so-called non-Newtonian liquids, for which the viscosity is not constant but is a function of the local shear stress. In [3] a simple power-law model for the blood viscosity is treated and a variety of other models are summarized. The resulting flow profile is a blunted one rather than the usual parabolic one.

Heating: Joule heating is generated when an electrical current flows through a channel. In some applications such heating may destroy biological samples. It may also induce changes in viscosity and thus affect the velocity profile. Moreover, due to enhanced the Brownian motion, heating may increase the dispersion of a sample plug since it. Heating may also increase the mobility of the ions in electric double layers leading to enhanced flow rates in electroosmotic flows, [5]. Finally, heating will stimulate production of bubbles of the gases dissolved in the liquid. The bubbles are usually unwanted and can block the flow if getting stuck in narrow contractions.

For a given application only some phenomena need to be taken into account. In many microfluidic simulations heating is often neglected in the first considerations.

5.3 Choosing software and hardware

When choosing a commercial CFD package, there are basically two guidelines: its capability (including the company’s support) and its price.

The companies usually give one-month free test trial so that a user can get a feeling for the tool. Because the tools are rather comprehensive, one month is often not enough time. Therefore, it is a good practice to think of an instructive, fairly simple test case (different from the tutorials) that can be easily checked in the literature. If possible, several software tools ought to be evaluated on the same problem so that direct comparison can be achieved.

The software companies distinguish between academic licenses (a few thousands of euros), aimed for universities, and the commercial licenses (several tens of thousands of euros), aimed for companies. Some companies give special inexpensive annual academic licenses (a couple of thousands of euros) which are extremely attractive. These can be renewed every year with an even smaller amount.

Of all CFD packages available today on the market two will be discussed more thoroughly since they have been used to produce the simulations in this chapter.

CFD-ACE+ version 6.6: This software is a comprehensive Finite-Volume tool with an intuitive interface [1]. The original version was designed for macroscopic CFD problems (including turbulence), but later, very extensive
modules have been added to describe a more general range of problems including those of microtechnology. At present, there are altogether about 20 different modules. In microfluidics, specific combinations of flow, heat transfer, chemistry, free surface, electric, magnetic and bio-chemistry modules are used to tackle the phenomena such as pressure driven flows, transport of numerous chemical species (e.g. electrophoresis), electroosmosis, bubble behavior, binding of bio-molecules, etc.

The program has excellent grid generation capabilities supporting both structured and unstructured grids. It has built in scripting techniques around the Python language which makes optimization of geometry as well as parametric runs (i.e. running consecutive simulations while changing one or more parameters) very easy to handle.

The numerical tools include a variety of boundary conditions, different discretization schemes etc. and even parallel processing is possible. There are numerous viewing capabilities of the results. One of the strengths of the tool are very extensive manuals in which everything from numerics to physical governing equations is described in detail. In particular, limitations of the models are discussed. The staff delivers good professional support, and the company is open for collaboration and offers annual academic licenses.

Simulations often require long run times. Fortunately, a new solver can be selected to speed up convergence. The solver is the particular algorithm used to obtain the numerical solution (see Sections 5.4.2 and 5.5). Extra attention needs to be put on properly setting the numerical parameters, and this requires a knowledgeable user. This, however, is beneficial in the long run since the user has the control over things. The program can be operated on Windows, Unix or Linux. It uses around 320 MB of hard disk space to be installed (standard package). More on specifics of the package can be found in [2,3] where bubble motion in microchannels and non-Newtonian microflows are considered.

**CoventorWare™, version 2001.3**: A software package specifically oriented to microtechnology (to both micro-electro-mechanical systems (MEMS) and microfluidics) [4]. CoventorWare™ has no turbulence module, which is probably unnecessary in microfluidics anyway. Another characteristic feature is that the grid generation emulates the manufacturing process, which automatically guides the user in making realistic designs. The possible designs practically have two-dimensional character so that more complicated structures and grids need to be designed in an external program such as I-DEAS® and subsequently imported.

The CoventorWare™ package contains many different types of modules. Three are of particular importance in microfluidics (parts of Analyzer™):

- MemCFD covers general CFD tasks (steady, transient, compressible and incompressible laminar flows; suitable for pumps and valves);
- NetFlow covers pressure, diffusion, electrophoresis, electroosmosis and chemical transport of up to four species;
- SwitchSim has very convenient cyclic voltage boundary conditions useful when studying time dependent separations on a chip. The results can be visualized in a variety of ways.
A strong point of the program is that the solvers are rather fast (NetFlow is using another CFD program called Fluent®, which is well tested). Also, different parametric runs can be easily implemented. Another attractive feature is that the system level modeling, based on the equivalent circuit theory, for more complicated fluidic networks is on the way to be incorporated in a user-friendly manner. The downsides are that in many aspects CoventorWare™ is like a black box, without sufficient explanation in the manuals of the working mechanisms, numerical or theoretical. Also, complex interfacing of the external solvers can bring about some compatibility errors. However, the support staff is exceptionally professional and all requests and questions are answered in detailed and timely manner. An additional point is that the user interface is somewhat non-standard compared to other CFD programs. CoventorWare™ runs on Windows and Unix. It uses around 380 MB of hard disk space to be installed and requires at least 512 RAM memory to properly run, which is rather demanding. For a thorough overview of the program with the excellent examples of electroosmotic pumps and dispersion phenomena please see [5,6].

To conclude: CFD-ACE+ and CoventorWare™ are generally good and useful with some specific advantages and limitations. In the end, it is the knowledge of a user that makes a difference. Apart from being prone to make more frequent mistakes, untrained users cannot exploit such extensive software to the full extent.

The standard mathematics programs, such as MATLAB®, [7], and MATHEMATICA®, [8], are useful supplements in computing analytical (exact) solutions of equations for comparison with approximate CFD simulations (the latter often do not model all aspects of interest).

Hardware: The simulations presented later in this chapter were done on PCs operating with Windows 2000. The computers had the following specifications:

- processor: Intel Pentium III processor 933 MHz
- memory: 512 MB
- video: Intel 82815 Graphics controller (AGP) 4MB of memory
- hard drive: 40 GB
- other: 17" monitor, 3-button mouse, a CD writer (for large simulations)

5.4 The core elements of typical CFD software

Any CFD package typically consists of a pre-processor, a solver, and a post-processor. In the following these different elements will briefly be described.

5.4.1 Pre-processors

A pre-processor is the interface between the user and the numerical solver. Several factors must be considered before the solver can be started.

Geometry: When solving CFD problems the first step is to define the computational domain, i.e. the region in which the fluid equations are being solved. Using symmetry arguments the region to be treated can often be
reduced significantly thus cutting down on the computational time. Therefore, the computational domain should be the minimum region needed to be simulated (Figure 5.3). In case of three-dimensional (3D) structures it is a good idea to simulate first similar and less time demanding two-dimensional (2D) problems until all the parameters are properly set. Some programs offer parametric specification of geometry which makes the complicated designs easily altered.

[Figure 5.3]

Grid generation (meshing): In order to obtain appropriate results the computational domain needs to be divided in a number of cells by use of a grid or mesh. In general, finer grids yield more accurate results, but also increase the computational time immensely. Thus it is very important to generate an efficient grid.

There are two most common types of grids: structured and unstructured. The first one is more regular in the sense that neighboring cells can be easily enumerated (Figure 5.4). They usually include four-sided polygons such as rectangles (2D), or bricks and prisms (3D). A well structured grid include high orthogonality, low skewness and low to moderate aspect ratios of the cells. An unstructured grid can be more easily adapted to irregular geometries since it includes triangles in 2D and tetrahedra in 3D. Its generation and implementation, however, requires more complicated algorithms and can lead to increased numerical diffusion e.g. in shear regions, [10]. Numerical diffusion is accumulation of numerical round-off errors due to the discretization of the differential equations (see Section 5.5). A hybrid grid combining both structured and unstructured grid is shown in (Figure 5.5).

A solution should be independent of the grid. This can be checked by monitoring the solution as a function of grid refinement. Often the grid is refined successively by a factor of two until the change in calculated quantities is smallen than a user-defined tolerance. The regions where large gradients of velocity or concentration are to be expected require more resolved grid. In particular, the area around sharp corners needs to be well resolved. The grid aspects are discussed further in the test cases.

[Figures. 5.4, 5.5]

Model: The model of the physical, electrical and bio/chemical processes to be simulated needs to be defined with care. Sometimes a simple model with clear physical implications may lead to better understanding of the resulting CFD simulations than a model of unnecessary complixity.

Fluid parameters Parameters such as viscosity, diffusion constant (Table 5.1), and electroosmotic mobility need to be specified. The relevant data can be extracted from the software databases or can be found in literature, one good source being [11]. However, sources in the literature do not cover the ever increasing number of materials used in new applications. For example, the diffusion coefficient of a DNA strand depends on its length and the liquid environment and obviously cannot be arbitrarily given. New sets of experiments
might be needed to provide the correct input data, so a place to look are scientific publications dealing with the subject.

[Table 5.1]

**Boundary conditions (BCs):** The boundary conditions are specified on the surfaces that define the computational domain. Typical boundary conditions are velocity, pressure, wall effects (EOF), voltage and so forth. A more detailed overview of the BCs is given in the following section.

**Initial conditions (ICs):** The ICs are the values of the variables at the initial time. When searching for a steady state solution, a good guess for the ICs resonable close to the solution is often needed to shorten the computational time and to avoid divergence. When solving time-dependent problems, the late-time solutions of one simulation may serve as ICs for subsequent simulations.

**Solver settings:** Numerical parameters are adjusted so that the solvers can solve the problem at hand most efficiently. These settings include type of solver, numerical schemes, number of iterations, convergence criteria, relaxation parameters etc. A more detailed description along with some definitions is given in the following section as well.

### 5.4.2 Solvers

Solvers are numerical algorithms intended to solve the governing equations. The following two methods are most common in commercial CFD programs.

**Finite Volume Method (FVM):** The Finite Volume Method is the most widely used CFD technique. It is based on the concept of transport equations (Figure 5.6). The fluid domain is divided into finite volumes, and the governing equations are integrated over these volumes. The integrated equation is approximated with various finite difference methods such as the central difference and the upwind scheme. The numerical schemes convert the integral equations into a system of algebraic equations. These equations are then solved by an iterative method. Specific solver properties are discussed in Sec. 5.5.2.

[Figure 5.6]

**Finite Element Method (FEM):** The FEM method makes use of simple piecewise functions, such as linear or quadratic defined on small elements, to approximate the exact solution. A residual is defined to measure how far the approximate solution is from fulfilling the governing equations. In the so-called Galerkin method, in an iterative procedure these residuals are then minimized by using them to correct the approximate solution [12].

FVM is mostly used for fluid dynamics while FEM is more suitable for mechanical stress analysis and simulations of MEMS. The FEM method uses usually unstructured grid and the requirement for the grid quality is less stringent compared to FVM. On the other hand, FEM requires more computer memory than FVM on structured grids with same number of nodes [13].
5.4.3 Post-processors

A post-processor is used to visualize results. Some commonly used visualization techniques are:

**Translation/Rotation/Scaling:** Computational domain can be easily manipulated, probed and viewed from many sides that could otherwise, as in experiments, be hardly accessible.

**Vector field:** Plots the velocity field with vectors in different colors and lengths depending on the velocity.

**Streamlines:** A streamline is a tangent to the velocity field at a fixed time.

**Streaklines:** A streakline is also a tangent to the velocity field, but not at a fixed time. The streaklines follow the particles as when a dye is injected into the flow. In a steady state flow, the stream- and streaklines are identical.

**Volume visualization:** Scalar variables such as pressure, electrical potential or concentration may be visualized using colors. Today’s packages offer animations of transient files as well as exporting of reliable numerical data which can be manipulated in external programs for further evaluation. Detailed visualization capabilities are one of the strong points of CFD techniques.

5.5 Important numerical settings

Before undertaking a large simulation project it is crucial to have a simulation logbook. Even the most logical settings may be forgotten after a short while. The logbook should contain information about the grid generation, boundary conditions and especially numerical parameters.

5.5.1 Boundary Conditions

The boundary conditions are necessary to define the problem and thus to obtain the solutions. Surface boundary conditions are applied to so-called patches, as in CoventorWare™ [4]. A patch is a surface section on the model, e.g., a cube has six patches. In 2D problems a patch is a line. Here are briefly described boundary conditions commonly used in EOF simulations such as EOF pumps, [5].

**Pressure:** Specifies pressure on a patch.

**Voltage:** Specifies voltage on a patch.

**Inlet/outlet (default):** Specifies zero velocity gradients in the normal direction of a patch. Hence one must be careful not to place the inlet/outlet in a non-uniform region which would give spurious results.
Symmetry: Symmetry is invoked in order to save computational time. For example, two symmetry planes can reduce the number of cells by a factor four. The symmetry BC assumes that the gradients perpendicular to the symmetry plane or line are zero.

Wall: The wall BC simply sets the velocity on a patch to zero.

Velocity: Specification of a velocity on a patch. If needed, velocity profiles can also be specified.

EOF mobility: The commercial packages assume the Debye layer to be infinitely thin. This is valid if the Debye length is much smaller than typical channel dimensions. In case of an EOF-pump, a typical Debye length is less than 10 nm, a factor thousand or more smaller than the channel dimensions. Therefore, assigning the EOF mobility on a patch is a good approximation. However, if the Debye length is comparable with channel dimensions some corrected velocity boundary conditions must be applied [14].

5.5.2 Solver Settings

When running software such as CFD-ACE+, containing many numerical subroutines and parameters, the user must choose the proper setting in order to run the given simulation optimally. Fine tuning is important since it influences the solution convergence and can speed up solving of equations. If the parameters are not set and monitored properly, simulation will result in wrong solutions that can appear correct. In the following the most relevant solver settings are discussed.

Numerical discretization schemes are applied to discretize the partial differential equations in space and time in order to compute a variable at a new position or a new time [12], [25]. However, the numerical discretization is not unique, and the following list points out strengths and weaknesses of the most commonly used schemes. Examples of spatial discretization schemes are:

- Central differencing schemes are applicable for diffusion-dominated low-Reynolds-number problems or when the grid cells are small. They are not suitable for general flow problems involving convection due to the lack of transportiveness (direction of flow is not properly manifested). For high convection-to-diffusion ratios, they yield oscillations or wiggles in the solution.
- Upwind differencing schemes are very stable and generally applicable but there are problems with numerical (false) diffusion.
- Hybrid schemes are widely used as they combine the central and upwind differencing schemes to take the advantage of both. However, they do produce some numerical diffusion.
- Higher order schemes are more accurate i.e. minimize numerical diffusion, but at the same time less stable resulting in wiggles (so-called over- and undershoots).

Time discretization needs to be applied for time-dependent flows. Generally, smaller time steps are needed in order to achieve more accurate results.
Determination of proper time steps need to be done in connection with the specific problem keeping in mind all relevant physical time scales.

For discretized transport problems, the so-called Courant-Friedrich-Levy (CFL) number determines how many grid cells a fluid element passes during a time step, \([10]\). Stability of simulations can be controlled by the CFL number, \([13]\). For incompressible flows, the CFL number is defined by \(\text{CFL} = \frac{v \Delta t}{\Delta x}\), where \(v\) is the local velocity, \(\Delta t\) is a time step, and \(\Delta x\) a local cell size, \([13]\]. Knowing the velocity, the grid spacing and the upper limit of CFL number (equal or less than 1) one can get a feeling for correct order of the time step.

Three widely used temporal discretization schemes are:

- Explicit schemes that use only values of a variable from the previous time step to calculate the value at a new time. It is not generally recommended since it severely restricts maximum allowable time-step.
- The Crank-Nicolson scheme that combines the values both from the previous and at the new time step. It is usually used in conjunction with spatial central differencing scheme. It has a second-order accuracy.
- Implicit schemes that use values from the surrounding nodes at the new time step. It has a first-step accuracy. However, since it is unconditionally stable and robust, the implicit method is most recommended for general flows.

Solver choice: After the governing equations are integrated and discretized, they are converted into a system of algebraic equations. The matrix of this system is then being solved by iterative methods ensuring correct linkage between the pressure and the velocity field. Two possible types of equation solvers are:

- Algebraic-Multi-Grid (AMG) solver comes faster to the average solution by using a hierarchy of grids, involving both fine and coarse grids, all the way down to two-cell grid, to solve the equations. As opposed to geometric multigrid solver, an algebraic solver manipulates matrices rather than geometry. It is fast for both structured and fully-unstructured grids. It is particularly good for low- Reynolds-number regimes where convection is low, such as in microfluidics.
- Conjugate-Gradient-Squared (CGS) solver with Preconditioning solves the system matrix in a more straightforward way, easier to implement, but at the expense of longer computational time. Matrix equations are treated as a minimization problem and preconditioning is added to accelerate convergence rate, \([15]\). The solver is slower than AMG when a large number \(N\) of grid cells is involved, since it takes \(N\) steps to arrive at the final solution.

The default settings for these solvers rarely need to be changed. For thorough treatment of numerical solvers see \([1]\), \([15]\), \([25]\).

Residuals: The governing equations are solved iteratively until convergence is achieved. This means the solution does not considerably change from one iteration to another. The residuals can be understood as deviations between two iterative values. The convergence of a solution can be evaluated by monitoring the residuals during the iteration process (Figure 5.7). The rate of change of a residual is usually plotted as function of iteration count. In a time dependent simulation the development of the residuals is presented at every time step. Generally, a solution is considered converged if the residuals drop by
four to five orders of magnitude. However, in some cases of bubble simulations, where a special Volume-of-Fluid method for two-phase flows is used, convergence by more than two orders of magnitude is hardly achieved [1].

[Figure 5.7]

**Maximum iterations**: Although problem dependent, the maximum iterations should at least be set to 40 or 50, and then adjusted according to the residuals.

**Relaxation parameter**: When it comes to the practical simulations, this particular setting causes the most annoyance among users so it is worth the effort to explain it in a more detail. Depending on the value of the relaxation parameter the terms over- and under-relaxation are used in literature, but we will use the shorter term ‘relaxation’ as it appears in the software.

The previously mentioned pressure-velocity linkage or coupling is somewhat tricky and the iterative algorithms need to be devised in a smart way. Namely, in the case of incompressible viscous flow, there is no separate equation for pressure; it only appears through the gradient term in the momentum equations. The SIMPLE algorithm (Semi-Implicit Method for Pressure-Linked Equations), [16], reconstructs the pressure and velocity fields from an initial, guessed pressure. First, approximate velocities are calculated from the discretized momentum equations that include this initial pressure. Next, using the continuity equation a pressure correction is obtained, which is then used back to correct the pressure and velocities. The discretized equations for scalar variables such as temperature are then being solved. The procedure is repeated until all variables converge, [12].

The relaxation parameters are used to constrain the change in a variable from iteration to iteration, preventing divergence of the solution. For example, the pressure correction might be too big of a change especially in the beginning where the initial pressure is far away from the final solution. Therefore, the idea is to relax the solution \( \hat{S}_n \) obtained by iterating the previous solution \( S_{n-1} \). The relaxed solution \( S_n \) is given by the expression \( S_n = \alpha \hat{S}_n + (1 - \alpha) S_{n-1} \), where \( \alpha \) is a relaxation parameter between 0 and 1. Basically all variables (velocity, pressure, temperature etc.) are being relaxed in the iterative process. Too big values of \( \alpha \) cause large changes between iterations leading to numerical instabilities (oscillations) and finally divergence. On the other hand, too small values of \( \alpha \) can cause extremely slow convergence [12]. The fine tuning is important since there is a minimum in number of iterations needed to reach convergence for a specific value of the relaxation parameter [1].

The present version of CFD-ACE+ includes the specification of two relaxation parameters which can be somewhat confusing. The above defined parameter is called linear relaxation parameter and is set to 1 by default. Decreasing the value adds stability and slows the convergence. Since the program uses SIMPLEC (SIMPLE Consistent) algorithm, [17], it is recommended that this value is left intact and is only altered in cases of severe divergence. In some

---

1 J. Maruszewski and Z. J. Chen of CFD Research Corporation are acknowledged for their input.
cases of skewed grids, [18], linear relaxation parameter for pressure was set to 0.8.\(^2\)

Usually altered inertial relaxation parameter acts oppositely, i.e. its decreased value speeds up convergence at the cost of lower stability. It is important to recognize this difference since users tend to ever increase this parameter, which can actually take values greater than 1. One proven method is to modify it by systematically lowering its default value until the convergence deteriorates or divergent behavior appears. Then one can be sure to be close to the mentioned minimum of iterations. In many cases such as heat transfer the inertial relaxation parameter must be lowered several orders of magnitude. The code allows minimum value of 10^-10.

Relaxation parameter acts like a time step, [25]. The elliptic equations, such as the Laplace equation for the potential, do not have time dependence and the variables can have the steepest change or largest “time step" from one iteration to another i.e. the inertial relaxation parameter can be as small as possible (theoretically zero). As pointed, velocity is calculated from a guessed pressure that can be far away from the final pressure; thus, the changes in both the pressure and the velocity between the iterations need to be gradual to avoid divergence – a certain optimum relaxation value needs to be used. Generally, a larger inertial relaxation (smaller “time step”) is needed in cases where convective terms are significant.

Here is a concrete summary about the inertial relaxation parameter for most common microfluidic variables (confirmed by experience):

- For velocity, the values around 0.1 result both in fast convergence and good stability. In CFD-ACE+ the default value is 0.2.

- The pressure-correction inertial relaxation parameter is by default 0.2 and is adequate for many simulations. Alternatively, it can be set to the smallest possible (10^-10, corresponding to the largest “time-step"), which is especially valid in the case of low Reynolds numbers; if necessary, the pressure regulation, as for the highly skewed grids, can then be done through the linear relaxation parameter as explained above. One of the reasons for using the smallest values of the pressure-correction parameter is that the increased value of this parameter keeps the code from conserving mass at every iteration, [19].

- In case of the electrical potential, the elliptic nature of the Laplace or the Poisson equations allows the parameter to be the smallest possible to speed-up the convergence.

- For species transport, since convection is present, a certain relaxation parameter is necessary. The default value of 0.05 is usually sufficient.

- For the heat transfer simulations, the relaxation parameter for enthalpy should be as small as possible.

---

\(^2\) At the same time the pressure-correction relaxation parameter, introduced further in the text, was set to the minimal value 10^-10.
5.6 Errors and Uncertainties

It would be useful now, after describing the various stages of the simulation process, to give a summary of the involved errors and uncertainties. In CFD the following categorization can be made, [13]:

Model uncertainty: real flow vs. exact solution of modeled equations

Discretization or numerical error: exact solution vs. numerical solution of the equations (numerical diffusion enters this category).

Iteration or convergence error: fully converged vs. not fully converged solutions

Round-off errors: parameter values which are below machine accuracy

Application uncertainties: lack of available data (precise geometry, specific BCs, diffusion coefficients etc.)

User errors: mistakes and carelessness of the user

Code errors: bugs in software

5.7 Interpretation and Evaluation

The correct interpretation of obtained numerical results is one of the most important issues because of the complexity of simulations. In particular, the numerous visualizing capabilities of CFD programs, especially nice colors, can give a misleading sense of having obtained the correct result when this may not be the case. So how do we know to what extent the simulation results represent reality?

The ultimate test of CFD results are experiments. In microfluidic systems it is rather difficult to make detailed measurements since the systems are small. However, one of the experimental techniques for flow measurements, namely Particle-Image-Velocimetry (PIV), has been adapted for microfluidics and simulations can indeed be directly tested against suitable experiments, [20]. The idea behind the method is to reconstruct the flow velocity from the velocity of small fluorescent particles (tracers) placed inside the flow. In simple language, the particles are illuminated by a light source and their position recorded in two consecutive instants by a CCD camera. From the known positions and the time delay between the two instants, the particle velocities and consequently the flow velocity can then be determined.

In the case where experiments are not available, one powerful way is to compare the simulation with well established theoretical models such as the equivalent circuit theory. This theory describes a fluidic network as an equivalent electrical network by expressing a linear relationship between a pressure drop (equivalent to a voltage drop) and the corresponding flow rate (equivalent to a current), [5,21]. It is applicable to uniform incompressible flows, and is very useful for microfluidic systems operating in low-Reynolds-number
regimes. If the theoretical model agrees well with the simulations (in quantities such as pressure drop, or flow rate) there is a strong indication of having obtained correct results. Simulations, however, provide more details. The accuracy of the codes of the software packages need to be tested either against experiment, analytical results or some other form of approved references.

5.8 Illustrative Examples

5.8.1 Fully-developed flow in a circular capillary

In this test case the idea is to show several aspects of grid-dependency analysis. We will consider Hagen-Poiseuille flow i.e. a pressure driven flow in a circular capillary, for which analytical solution exists. The validity of grids will be determined by comparing the calculated velocities with the analytical solution.

Physics: When liquid from a large reservoir enters a straight tube of arbitrary but constant cross-section, the velocity profile develops from being flat at the entrance to become paraboloid after a certain length called the entrance length. For a circular pipe, the entrance length can be up to 100 diameters [22]. Beyond the entrance length the velocity field is parallel to the cylinder axis and its axial component \( u(r) \) depends only on the radial coordinate \( r \) according to the well-known expression

\[
    u(r) = - \left( \frac{G}{4 \mu} \right) (a^2 - r^2),
\]

where \( G \) is the pressure gradient, \( a \) the radius of the pipe, and \( \mu \) the viscosity of the liquid. We will consider the fully-developed flows described by the above equation. For microfluidic systems which are connected to external reservoirs by long tubes, this will be a good description [23].

Geometry: The problem is axially symmetric, so the computational domain can be reduced to a two-dimensional rectangle with four boundary conditions specified as inlet, outlet, wall and symmetry, (Figure 5.8). The geometry of the computational domain is written as a Python-language scripting file to ease the modifications of the dimensions or of the number of grid cells. For this case the following dimensions are used: \( r = 75 \mu m, L = 900 \mu m \). [Figure 5.8]

Grid: The velocity is changing in the \( r \)-direction (orthogonal to the flow), thus the grid resolution needs to be higher in that direction. Four different grids are depicted in (Figure 5.9). [Figure 5.9]

Physical and numerical settings: Since we are trying to solve a 3D case from the reduced 2D geometry, simulations need to be set as axisymmetric. The equations for axisymmetric problems differ from the equations for simple 2D problems: if a flow rate is specified at a 2D inlet, solver integrates the value over the circular cross-section to obtain the 3D value. The density and the

---

3 Both of the examples are being simulated with the CFD-ACE+ 6.6 software.
viscosity of water need to be selected. A desired pressure is assigned on the inlet and the zero pressure on the outlet. Choosing of the proper numerical parameters is governed largely by experience (see section 5.5.2) and by trial-and-error. Sometimes it is good to try the default values first and then make appropriate changes that yield the fastest convergence. The settings are summarized in (Table 5.2).

[Table 5.2]

Results:
The maximum velocities are calculated on each of the four grids. The program simulates the results in about five seconds. The velocity profiles from the coarsest and the finest grids are compared with the analytical solution in (Figure 5.10). The deviations between the analytical solution and the calculated values are summarized in (Table 5.3). The most refined grid in y-direction gives the best results deviating only 0.5%. It is important to notice the decrease in relative deviations. In most of the cases there will be no analytical solution for comparison and the grid dependency will rely on the decrease of the deviations from one grid to the next refined.

[Figure 5.10]

[Table 5.3]

The axial symmetry of the flow allows for a substantial reduction of the computational domain. If the flows are symmetric, but not axially symmetric, the simulations can be performed on half of the full computational domain, as in (Figure 5.3). When symmetry cannot be used, simulations need to be done on the full 3D domain. Although not necessary here, it would be instructive to highlight some grid aspects from simulations on a 3D Poiseuille problem, (Figure 5.11).

(Figure 5.12) shows three grids with different orthogonality properties used to mesh the complete circular cross-section of the pipe. The orthogonality of grid cells directly affects the accuracy of flux calculations, and therefore the accuracy of results. In cases such as simulations of gas bubbles, only orthogonal grids yield convergence. For the above dimensions, the number of grid cells in the 3D domain is on the order of 6000.

[Figures 5.11, 5.12]

All three grids reach the desired solution for the mean and the maximal velocity within a margin of 3% error, although the third grid gives the best results. The deviation is acceptable but larger than the above axisymmetric simulations. In order to get a better agreement the grids need to be even more refined which will significantly increase the simulation time. With the default solver settings the convergence in this 3D case was reached after 200 iterations in approximately 90 minutes.

5.8.2 Movement of a chemical plug in an electroosmotic flow for a 2D geometry.

In this test case the purpose is to show how numerical settings (time step, solvers, relaxation parameters) influence the results of time-dependent simulations. Both the incorrect and the correct results will be presented.
Physics: (Figure 5.13) shows a fluidic system used for separations, [24]. By timely manipulation of voltages between sample, buffer and waste reservoirs, a sample plug, consisting of several differently charged species, can be injected into the separation channel. The voltage drop generates an electroosmotic flow (EOF) which is governed by the zeta-potential at the walls. The EOF induces a downward bulk flow, and species from the plug, once they come into the separation channel, begin to separate due to the different velocities in the applied electric field. When species enter the horizontal part of the detection cell, they are confined in the much narrower and longer region. If a light beam is then passed through the region, as from a waveguide fabricated next to it, the species will absorb the light. Due to the longer absorption length, the detection signal in absorption measurements will be significantly enhanced. In making realistic simulations, the main concern is to correctly calculate the EOF i.e. to couple the electric field with the flow. The potential distribution is found by solving the Laplace equation using the voltage BCs. The electric field at the boundaries is found as the negative gradient of the potential, and by multiplying it with the value of the electroosmotic mobility we arrive at the flow velocity at the boundaries. The mobility can be set as a BC or it can be calculated from the zeta potential. Heating phenomena will not be taken into account.

[Figure 5.13]

Geometry: Simulations are performed on a 2D structure which includes only a part of the detection cell (Figure 5.14). This simplified set-up will reduce the computational domain while being sufficient to highlight the desired aspects of (im)proper settings. However, 3D simulations that include the entire geometry with the proper initial and boundary conditions are necessary in order to make the qualitatively correct simulations comparable with experiments. Variables such as flow resistance are governed by the 3D geometry.

The real device is made in silicon thus the edges are sharp. The separation channel is asymmetrically narrowing down to join the narrow detection channel. In this way dispersion of a plug entering the narrow part due to the race-track effect can be controlled. The two corner points which determine the junction between the wide and the narrow sections are parametrically specified for the optimization purpose. The channel widths \( w_{\text{sep}} = 120 \, \mu \text{m} \) and \( w_{\text{det}} = 30 \, \mu \text{m} \) used in the simulations are the same as in the real device.

In the CFD-ACE+ program, a special region within geometry needs to be specified for a sample plug. The undistorted plug, consisting of a positive and a negative species, is positioned at the beginning between the two lines parallel to the inlet, (Figure 5.14).

[Figure 5.14, 5.15]

Grid: A part of the grid is shown in (Figure 5.15). In order to make the grid orthogonal and well resolved around the corners, the geometry was divided in several segments. A larger number of cells in the separation channel produces smooth concentration profiles and allows better resolution of the concentration peaks. In the narrow region, the velocity profile is homogeneous and the overall
(integrated) concentration is important, so the cell density can be reduced. However, biasing is implemented since the adjacent cells need to be of comparable sizes. The sharp corners produce singularities and the results cannot be trusted in their vicinity.

**Physical and numerical settings:** Three modules need to be switched on at the same time: flow, chemistry and electric.

- Fluid properties: in this system, the buffer is water and the sample consists of the positively charged rhodamine 110 ($C_{20}H_{15}ClN_{2}O_{3}$) and the negatively charged fluorescein ($C_{20}H_{12}O_{5}$). When simulating various chemical solutions, the program uses diluted approximation which means that the complicated interactions between and within chemical species are neglected. This is a reasonable assumption since a sample is usually strongly diluted in real experiments with typical concentrations around 1 nmol/L. Therefore, bulk properties such as viscosity, density and conductivity of both the buffer and the sample can be assumed to be the same. To simplify things, bulk properties of water were used. If, for example, the borate buffer $Na_2B_4O_7\cdot10H_2O$ was used, it would have the viscosity and the density of water but a different conductivity. The conductivity depends on the buffer concentration and can be experimentally measured.

In order to assign the specific properties such as charge, diffusion coefficient, concentration etc. both the buffer and the sample need to be “created” in the material database so that they could be invoked into different settings such as initial conditions. For the sample species in our case the diffusion coefficients were deduced from the diffusion constants of ions with the similar structure, [11]. The important properties are given in (Table 5.4).

- BCs: A voltage of 100 V is specified at the inlet and 0 V at the outlet. The zeta potential of -0.1 V was set at the walls. The boundary conditions are also summarized in (Table 5.4).

[Table 5.4]

- ICs: Initial conditions are particularly important in the time dependent simulations since things change with each time step. The sample, selected from the material database, need to be assigned to the plug region that was previously made in the geometry for that purpose. The buffer needs to be assigned to the rest of geometry.

Solver settings: Two sets of simulations were performed. The “bad” one include improper numerical parameters yielding unphysical results, and the “good” one with the corrected values. Both settings are summarized in (Table 5.5).

[Table 5.5]

Results: The movements of the positive and the negative species at different times are shown in (Figures 5.16 and 5.17) respectively. The positive species seems to be moving too fast so that the snap shots ($t=0.10 – 0.15$ s) barely resolve the concentration profile, a possible indication of a too big time step. In the case of the negative species, there is a clear unphysical distortion of the plug.

[Figure 5.16, 5.17]
The movement of the species is a combination of the three effects: the downward electroosmotic movement of the entire bulk fluid i.e. EOF, the electrophoretic motion of the charges towards the opposite electrodes, and the diffusion. For negative species the upward electrophoretic motion is opposing the EOF. We would expect that the overall motion is downwards, although less pronounced than for the positive species, since EOF usually prevails. From (Figure 5.17), it seems as though that the EOF velocity is not well developed. A possible reason for this could be that the time-step was set extremely small, which is unlikely since the distortion is too pronounced in each time step. A more likely cause is that the potential did not converge fast enough. Since it delivers a body force that drives the EOF, a converged potential is required to determine the proper velocities. Indeed, by checking the output shown in (Table 5.6), it is revealed that the residuals of the potential and the resulting velocity diminish by only two orders of magnitude, after 25 iterations per time step. The unphysical recirculation regions in the velocity can be noticed from the graph in (Figure 5.18 a).

[Table 5.6]

The remedy is to increase the number of iterations per step or choose a different (faster) solver. In addition, the relaxation parameter for the potential needs to be adjusted in order to speed up the convergence. The correct time step was estimated from the condition CFL=1 (The size of a grid-cell is ~3 µm, the EOF velocity is \( v \approx 0.006 \) m/s yielding \( t \approx 0.0005 \) s). However, the step \( t = 0.001 \) s from the first simulations was close enough so it was not changed. Proper convergence of both the electric potential and the velocity can be seen from (Table 5.6). The residuals drop ten and five orders of magnitude respectively. The resulting, physically correct velocity vectors are shown in (Figure 5.18 b).

[Figure 5.18a,b]

As a consequence of the proper settings, the movements of the species also appear physical (Figures 5.19 and 5.20). The positively charged rhodamine moves undistorted (plug-like) at the beginning. This is one of the advantages of the EOF over pressure driven flows: it produces an uniform velocity profile resulting in undistorted plug concentrations. That is why the distortions in (Figures 5.16 and 5.17) looked suspicious. Even though the time step was not altered, it now takes longer time for the species to enter the horizontal detection channel.

More drastic changes appear in the movement of the negatively charged fluorescein. It moves only slightly downwards but completely undistorted. After \( t = 0.35 \) s the two species are separated with concentration peaks lagging behind each other. This can be seen by comparing positions of the concentrations in (Figures 5.19 and 5.20).

[Figure 5.19 and 5.20]

If the concentration profiles of e.g. fluorescein from \( t = 0 \) s to \( t = 0.35 \) s are compared, there can be noticed an increase in the lighter, blurry region on both edges of the plug. This spread of the concentration profile is caused by diffusion whose effect is determined by the diffusion coefficient. Even if the diffusion coefficient is set to zero, a certain spreading will persist due to the numerical diffusion immanent to some discretization schemes. It needs to be
underlined that simulations are reliable only within the certainty of the built-in models and the input data. Thus, an approximate diffusion coefficient will further affect the accuracy of the output concentrations.

Corners represent discontinuities of boundary conditions and lead to singularities in numerical analysis. The results in the vicinity cannot be trusted. For a grid with a finite cell size, this singularity is smeared out over a region. The remedy is to make round corners or choose different boundary conditions to represent the flow; in case of EOF it can be velocity instead of the zeta-potential. A smeared region around a corner can be seen in the contour plot of the electrical field in y-direction, (Figure 5.21).

[Figure 5.21]

Evaluation: As already mentioned, realistic 3D simulations are needed for the appropriate comparison with the experiments. Then, the species concentrations integrated in the detection channel could be directly compared with the concentration-dependent signals from the absorbance measurements. Several things are worth noting when such a comparison is made.

- Integration basically means adding up the value of a variable from each of the grid cells of interest. The integrated signal will somewhat depend on the grid resolution. Usually, sharp discontinuous concentration peaks are a sign that the grid is not well resolved.
- CFD programs can display concentration values between certain specified limits. The lower limit for concentration needs to coincide with the detection limit of the experimental set-up. Therefore, only the concentrations that can be detected are to be displayed in simulations and integrated.
- The effects of numerical diffusion need to be estimated.

Once the simulations reproduce sufficiently closely the experimental results, they can be further applied to optimize geometries. In the above case, for example, the slope of the asymmetrical narrowing can be optimized in relation to the sample distortion.

5.8.3 Conclusion

The previous two examples hopefully highlight some of the procedures used in setting and interpreting simulation problems.

It is clear that using a CFD program requires physical and mathematical insight. Even the program itself may contain a considerable amount of errors. Furthermore, a special care should be taken when traditional numerical techniques are applied to new physical areas such as microfluidics. Therefore, it cannot be stressed enough how one should always be critical towards the results. Only through the comparison with experiments or proven theoretical models simulations can be truly confirmed.

<table>
<thead>
<tr>
<th>Species</th>
<th>( D_{\text{mass}} ) (m²/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>( 0.890 \times 10^{-5} ) (self-diffusion)</td>
</tr>
<tr>
<td>Ethanol</td>
<td>( 1.24 \times 10^{-9} ) (in water)</td>
</tr>
<tr>
<td>Glucose</td>
<td>( 0.67 \times 10^{-9} ) (in water)</td>
</tr>
</tbody>
</table>
Sucrose $0.52 \times 10^{-9}$ (in water)
Urea $1.38 \times 10^{-9}$ (in water)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>$0.9982 \text{ kg/m}^3$ (water at 20 °C)</td>
</tr>
<tr>
<td>Viscosity (dynamic)</td>
<td>$1.002 \times 10^{-3} \text{ kg/ms}$ (water at 20 °C)</td>
</tr>
<tr>
<td>Inlet pressure</td>
<td>100 Pa</td>
</tr>
<tr>
<td>Outlet pressure</td>
<td>0 Pa</td>
</tr>
<tr>
<td>Temporal discretization scheme</td>
<td>None (steady state simulation)</td>
</tr>
<tr>
<td>Spatial discretization scheme</td>
<td>Upwind</td>
</tr>
<tr>
<td>Solver</td>
<td>Algebraic Multigrid (AMG)</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>100</td>
</tr>
<tr>
<td>Inertial relaxation</td>
<td></td>
</tr>
<tr>
<td>- Velocity</td>
<td>0.2</td>
</tr>
<tr>
<td>- Pressure correction</td>
<td>$10^{-10}$</td>
</tr>
<tr>
<td>Linear relaxation</td>
<td></td>
</tr>
<tr>
<td>- Pressure</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.3 Grid dependency analysis.

<table>
<thead>
<tr>
<th>Solution</th>
<th>$u_{\text{max}}$ m/s</th>
<th>Deviation in % (from analytical)</th>
<th>Deviation in % (from previous grid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>0.155938</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Grid 1 (coarse)</td>
<td>0.150493</td>
<td>3.49</td>
<td>0</td>
</tr>
<tr>
<td>Grid 2 (one cell)</td>
<td>0.154811</td>
<td>0.72</td>
<td>2.78</td>
</tr>
<tr>
<td>Grid 3 (fine)</td>
<td>0.155089</td>
<td>0.54</td>
<td>0.18</td>
</tr>
<tr>
<td>Grid 4 (biased)</td>
<td>0.154967</td>
<td>0.62</td>
<td>-0.08</td>
</tr>
</tbody>
</table>

Table 5.4 Property and BC settings for the test case 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>$1 \text{ kg/m}^3$ (water)</td>
</tr>
<tr>
<td>Viscosity (kinematic)</td>
<td>$10^{-6} \text{ m}^2/\text{s}$ (water)</td>
</tr>
<tr>
<td>Conductivity</td>
<td>$10^{-4} \text{ S/m}$ (typical value)</td>
</tr>
<tr>
<td>Diffusion coefficients</td>
<td>$10^{-9} \text{ m}^2/\text{s}$ (for both species)</td>
</tr>
<tr>
<td>Inlet voltage</td>
<td>100 V</td>
</tr>
<tr>
<td>Outlet voltage</td>
<td>0 V</td>
</tr>
<tr>
<td>Zeta-potential</td>
<td>-0.1 V</td>
</tr>
<tr>
<td>Debye thickness</td>
<td>$10^{-9} \text{ m}$</td>
</tr>
</tbody>
</table>

Table 5.5 Solver settings for the test case 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Improper</th>
<th>Proper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time discretization scheme</td>
<td>Crank-Nicolson with default blending (0.6)</td>
<td>Crank-Nicolson with default blending (0.6)</td>
</tr>
<tr>
<td>Time step</td>
<td>0.001 s</td>
<td>0.001 s</td>
</tr>
</tbody>
</table>
Spatial discretization scheme

<table>
<thead>
<tr>
<th>Solver</th>
<th>Upwind</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGS + Preconditioning with default settings</td>
<td>AMG with default settings</td>
</tr>
</tbody>
</table>

Iterations per step

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

Inertial relaxation:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity</td>
<td>0.2</td>
</tr>
<tr>
<td>Pressure correction</td>
<td>0.2</td>
</tr>
<tr>
<td>Voltage</td>
<td>0.0001</td>
</tr>
<tr>
<td>Species</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 5.6 Residuals of the electrical potential and the downward velocity in cases of improper and proper convergence. In the case of improper convergence, the electric potential drops by only three orders of magnitude.

<table>
<thead>
<tr>
<th>Improper convergence</th>
<th>Proper convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>El. pot</td>
</tr>
<tr>
<td>1</td>
<td>0.1853</td>
</tr>
<tr>
<td>2</td>
<td>0.005573</td>
</tr>
<tr>
<td>3</td>
<td>0.003157</td>
</tr>
<tr>
<td>4</td>
<td>0.001741</td>
</tr>
<tr>
<td>5</td>
<td>0.001252</td>
</tr>
<tr>
<td>6</td>
<td>0.001017</td>
</tr>
<tr>
<td>7</td>
<td>0.00096</td>
</tr>
<tr>
<td>8</td>
<td>0.000866</td>
</tr>
<tr>
<td>9</td>
<td>0.000794</td>
</tr>
<tr>
<td>10</td>
<td>0.000704</td>
</tr>
<tr>
<td>11</td>
<td>0.000623</td>
</tr>
<tr>
<td>12</td>
<td>0.000562</td>
</tr>
<tr>
<td>13</td>
<td>0.000509</td>
</tr>
<tr>
<td>14</td>
<td>0.000467</td>
</tr>
<tr>
<td>15</td>
<td>0.000419</td>
</tr>
<tr>
<td>16</td>
<td>0.000386</td>
</tr>
<tr>
<td>17</td>
<td>0.000352</td>
</tr>
<tr>
<td>18</td>
<td>0.000314</td>
</tr>
<tr>
<td>19</td>
<td>0.000291</td>
</tr>
<tr>
<td>20</td>
<td>0.000265</td>
</tr>
<tr>
<td>21</td>
<td>0.00024</td>
</tr>
<tr>
<td>22</td>
<td>0.000226</td>
</tr>
<tr>
<td>23</td>
<td>0.000203</td>
</tr>
<tr>
<td>24</td>
<td>0.000184</td>
</tr>
<tr>
<td>25</td>
<td>0.000164</td>
</tr>
</tbody>
</table>

[1] CFD-ACE+ version 6.6, (trademark of CFD Research Corporation)
CFD Research Corporation, Huntsville, AL, USA, 2001; software, manuals and support
http://www.cfdrc.com

http://www.mic.dtu.dk/research/MIFTS/publications/msc.htm

http://www.mic.dtu.dk/research/MIFTS/publications/msc.htm

http://www.coventor.com

http://www.mic.dtu.dk/research/MIFTS/publications/msc.htm

http://www.mic.dtu.dk/research/MIFTS/publications/msc.htm


http://www.afm.dtu.dk/Staff/jam/vocabulary/vocabulary.html


Figure 5.1 A mask of an integrated microsystem for detecting chemical reactions via chemiluminescence. Before entering the mixing region, a sample is passed through the long spiral for a prolonged reaction with immobilized enzyme. Then, a product, usually H₂O₂, is entering the mixing region and in catalyzed reaction with reagent produces light. Back-side photodiodes, used for the light detection, are not shown. The dimensions of the chip are 20 mm x 10 mm, and width of the enzyme reactor is 400 µm. Courtesy of A. M. Jørgensen.

Figure 5.2 The race-track effect in a turn in an electrokinetically driven flow. The concentration contours of a sample are displayed. The inside track has shorter path as well as higher electrical field inducing faster movement of the inner molecules, therefore enhancing sample dispersion.
Figure 5.3 The computational domain of a three-dimensional channel laser-machined in plastic. Due to the symmetry, only half of the full domain needs to be simulated. The geometry includes the characteristic Gaussian-like cross-section and the perpendicular ripples which can arise from the fabrication process. The contours show the axial velocity profile, [9]. Performed on CFD-ACE+ 6.6.

Figure 5.4 An example of a structured grid (top view) used in simulations of electroosmotic pumps, [5]. Of the three different cell sizes, the finest are used in the narrow channels to resolve velocity profiles. In the upper left part large cells are adjacent to small cells, and computations cannot be trusted near the junctions. Experience has shown that integrated quantities such as flow rate and pressure are less susceptible on the grid quality than velocity profiles. Made in CoventorWare™ 2001.3. Courtesy of A. Brask.
Figure 5.5 An example of a hybrid grid, combining a structured and an unstructured part. The domain presents a quarter of a two-dimensional plate with a hole in the middle. The region near the hole is meshed regularly and farther away the grid becomes unstructured. The grid is denser in the regions of expected stresses. Made in CFD-ACE+ 6.6.

Figure 5.6 Description of the transport equation used in the Finite Volume Method.

\[
\frac{\text{Rate of change}}{\text{Net flux of}} \begin{array}{c} \text{of } \phi \text{ in the control} \\ \text{volume with} \\ \text{respect to time} \end{array} = \begin{array}{c} \text{Net flux of} \\ \phi \text{ due to} \\ \text{convection into} \\ \text{the control volume} \end{array} + \begin{array}{c} \text{Net flux of} \\ \phi \text{ due to} \\ \text{diffusion into the} \\ \text{control volume} \end{array} + \begin{array}{c} \text{Net rate of creation} \\ \phi \text{ inside the} \\ \text{control volume} \end{array}
\]

Figure 5.7 A typical residual graph of a simple two-dimensional flow problem. The residuals of the two velocity components and the pressure are plotted on a log scale against the number of iterations. The drop of more than four orders in magnitude can be seen for all three variables. From CFD-ACE+ 6.6.
Figure 5.8 Schematic view of a straight circular capillary of radius $a$. The dashed region is the reduced two-dimensional computational domain with the assigned boundary conditions. In a time dependent simulations, a velocity profile develops through entrance length to become a paraboloid.

Figure 5.9 Four different grids of the two-dimensional computational domain sketched in Figure 5.8, with length $L=900$ um and radius $a=75$ um. a) A coarse grid with only four cells in the vertical direction where changes of velocity appear. The grid does not resolve velocity well enough as seen in Figure 5.10. There is no need for so many grid cells in the horizontal direction since there are no velocity gradients in the direction of flow. b) A grid with better resolution in the vertical direction (10 cells) and only one cell in the direction of flow. It gives better results than the coarse grid, Table 5.3. c) A fine grid with 14 cells in the vertical direction resolving the velocity profile very accurately, Figure 5.10. d) A biased grid with a better resolution of the so-called boundary layer close to the wall where the gradients are the largest. It gives good results, although slightly less accurate, for the maximum velocity calculated at the symmetry line. The biasing needs to be done close to the regions of the biggest changes.
Figure 5.10 Theoretical and simulated velocity profiles. Note the four straight line segments of the coarse grid, from each of the four cells.

Figure 5.11 A full 3D simulation of a flow in a circular tube of length $L$ and radius $a$. The grids contain around 6000 cells as opposed to 80 for the 2D axi-symmetric case in Figure 5.9. Courtesy of M. J. Jensen.

Figure 5.12 The three grids used in the 3D Poiseuille flow a) Regions with bad orthogonality at the edges b) Good orthogonality at the edges and good aspect ratios. c) Good orthogonality and enhanced wall resolution. Courtesy of M. J. Jensen.
a) Figure 5.13 a) System of channels for a lab-on-chip used for chemical separations and absorption measurements, [24]. The separation channel of length $L=3.5$ cm comes into the detection cell (enlarged) consisting of a narrow, long horizontal part and two wave-guides. The geometry provides longer absorption length and subsequently the enhanced detection signal. The width of the separation and detection channel is $120 \, \mu$m and $30 \, \mu$m, respectively. The uniform depth of the channels is $12 \, \mu$m. Courtesy of N. J. Petersen. b) A photo of the fabricated system. Courtesy of K. B. Mogensen.

b) Figure 5.14 The computational domain representing a part of the detection cell. The widths of the channels as well as the length of the horizontal channel are the same as in real device while the vertical part comprises only $400 \, \mu$m. The geometry is a two-dimensional. The lines can be seen of the several structured domains which can be separately meshed to achieve better control over the grid resolution. Simulations performed on CFD-ACE+ 6.6.
Figure 5.15 A part of the grid used in simulations. The cells in the middle of the vertical channel are fairly orthogonal. Orthogonality of the cells is not high close to the upper right corner. The area around the two corner points in the narrow junction are well resolved. The biasing can be noted in the uniform horizontal channel.

Please note that the layer close to the walls is not well resolved, especially in the right part of the vertical separation channel. This means that there will be a large layer where velocity drops from a uniform EOF value (in the middle of the channel) to zero (at the walls). In reality, this is the Debye layer which is very thin compared to the channel dimensions. In simulations it is not possible to completely resolve the layer; yet, a better resolution e.g., with more cells and biasing close to the walls, would yield a more accurate (smooth) velocity profile.
Figure 5.16 a)-d) Incorrect movement of the positive species in the electroosmotic flow. Concentration contours are shown and four snap shots taken at 0.01 s, 0.05s, 0.1 s and 0.15 s. Unrealistic distortion at the edges can be noticed as well as the faded concentration profile, unresolved in time.

Figure 5.17 a)-d) Incorrect movement of the negative species in the electroosmotic flow. Highly distorted concentration profiles are indication of a badly converged velocity, which in turn is linked to the electric potential.

Figure 5.18 a)-b) Velocity fields (vectors and contours) a) Non-converged velocity field with clear unphysical re-circulation areas, caused by a bad convergence of the electric potential (Table 5.6) 
b) Smooth, uniform velocity profile as a result of a good convergence of the potential.
Figure 5.19 a)-h) The correct movement of the positive species. The properly converged, uniform velocity results in the undistorted plug. It now requires more time for plug to enter the detection channel. As plug moves downwards, it diffuses in both backward and forward direction. The two lighter regions broadening from the edges of the plug indicate the concentration spread due to the diffusion. If carefully noted from the vertical concentration scale, the concentration maximum has been re-set at each time step so that the same colors represent different concentration values. This is done to highlight the shape of the moving plug which would otherwise be hardly visible.
Figure 5.20 a)-d) The correct movement of the negative species. The four consecutive snap-shots indicate the expected downhill motion. It is less pronounced compared to the positive species due to the opposing electrophoretic effect. Comparing the Figures 5.19d and 5.20g, we see that at t=0.30 s two plugs are completely separated.

Figure 5.21 Contours of the y-component of the electric field. The corner effects are visible yielding large electric fields in the vicinity (dark spot), where the results cannot be trusted.