Microdynamics versus Macrodynamics – An Interdisciplinary Student Project

Dr. Gunter Bischof, Joanneum University of Applied Sciences
Ms. Annette Casey B.A., University of Applied Sciences FH JOANNEUM, Graz, Austria

Annette Casey is a faculty member of the Institute of Automotive Engineering at the University of Applied Sciences FH JOANNEUM, Graz, where she has been teaching undergraduate English for Specific Purposes (ESP) courses for the past fifteen years. After graduating from Dublin City University with a B.A. (Hons.) in Applied Languages, she taught at several schools in Austria, before taking up a three-year appointment as an exchange lecturer at the Department of English and American Studies at the University of Klagenfurt, Austria. She has also worked extensively as a freelance language trainer at other third level institutions and in industry. Her research interests include ESP, Engineering Education, Project-based Learning, Materials Development and Educational Research Methods.

Dr. Emilia Andreeva-Moschen, Bombardier Transportation Austria GmbH

Since 2013 Bombardier Transportation Austria GmbH Director Drives Engineering Since 2013 External lecturer at the Technical University Vienna Sensors and Actuators

2012 – 1997 FH JOANNEUM 2006 – 2012 Head of Department of Automotive and Railway Engineering: 2006 – 2012 Head of accredited test laboratory (accredited to EN ISO/IEC 17025). 2008 – 2012 Head of Budget Committee 2008 – 2012 Member of the working group responsible for quality at the UAS. 2006 – 2012 Head of Board of Trustees of Department of Automotive and Railway Engineering. 2006 – 2012 Founder and Coordinator of the Field of Competence ”Measurement Technology” 2006 – 2012 Founder and Coordinator of the Field of Competence Engineering Education” 1998 – 2005 University’s representative in the national project ”Women in Technic” 1997 – 2006 Professor of ”Measurement Technology, Signal Analysis and Informatics in the Department of Automotive and Railway Engineering

1994 – 1997 EVK, Graz (Automation, Measurement Technology) Project leader, Product management and marketing, Engineering

1991 – 1994 CPA/Steinklauber, Graz (Automation, Control) Project management, engineering

1989 – 1991 Institute of Microprocessor Technology in Sofia Software and hardware development

EDUCATION

2001 – 2005 PhD at the Technical University Graz, graduated with distinction 1985 – 1989 Degree in Technical Journalism at the TU Sofia, graduated with distinction 1984 – 1989 Degree in Electrics and Electronics, specialization in medical electronics at the TU Sofia, graduated with distinction 1983 – 1984 Professional education at the Commercial Academy in Sofia, sales assistant certificate, passed with distinction 1982 - 1983 Professional education at ”Orbita” Tourist Agency, certified translator and tour guide (German and Russian), passed with distinction 1979 – 1984 German High School, passed with distinction Further education: numerous trainings, workshops and distance study programs in personnel management, controlling, economics, quality management, etc.

OTHER QUALIFICATIONS AND POSITIONS 2012 Member of the curatorial of the Siemens Railway Award since 2007 Member of the curatorial of the Magna Excellence Engineering Award 2008 Member of the development team of department of Industrial Management 2005 Member of the development team of department of Industrial Electronic and Technology Management 2002, 2007 Expert at the European Commission 2003, 2004 Visiting lecturer at TU Sofia since 1988 Translator

Languages: German (mother tongue level), English (business fluent), Bulgarian (mother tongue), Russian (good)

Microdynamics versus Macrodynamics
An Interdisciplinary Student Project

Abstract

Starting from their freshman year our students are involved in project work within the framework of project-based learning. Software projects complementary to the standard curriculum exemplify the applicability of the just learned methods and mathematical algorithms, thus increasing the students' attentiveness and their appreciation for the contents of teaching.

In this paper, an interdisciplinary project for sophomore students – the development of computer programs for the simulation and visualization of two-dimensional incompressible fluid flows – is presented. Two different approaches for problem solving could be chosen by the students – the numerical solution of the Navier-Stokes equation within the finite difference method, and the simulation of the flow field with a lattice-gas cellular automaton.

The projects, in general, and the visual perception of the flow fields derived from their self-made computer programs, in particular, have led to a deeper appreciation of mathematics, computer programming and fluid dynamics than solely teacher-centered instruction could convey.

Introduction

Project-based learning (PBL) has proved to be an appropriate method to demonstrate the interaction of computer sciences, mathematics and technology in the engineering disciplines. The establishment of interdisciplinary student projects within the framework of the Information Systems and Programming courses in the first and second year of our Automotive Engineering degree program turned out to be a particularly suitable first steps into a whole range of complexities of PBL[1]. The students are offered a variety of project proposals at the beginning of the semester and choose their projects according to their interests and skills. An important factor contributing to the acceptance and success is the usefulness and applicability of the projects. Students are highly motivated by tasks that stem from real engineering problems arising from their field of study[2].

One of those projects, the development of computer programs for the simulation and visualization of two-dimensional incompressible fluid flows, is presented in this paper.

The behavior of a viscous incompressible fluid is governed by the simplified Navier-Stokes equations, a coupled system of nonlinear partial differential equations. While the numerical solution of linear partial differential equations is part of the standard Engineering Mathematics curriculum, the nonlinearity of the problem made it necessary to offer supplementary lectures in order to bridge the students' knowledge gap. Three sophomore student teams in their third semester of study took up the challenge to simulate and visualize the flow of incompressible fluids within the finite difference method.

For comparison, another student team that has already collected some experience in a preceding project[17] tackled the same problem with a cellular automaton approach. The

microscopic interactions in such lattice-gas cellular automata lead to the same form of macroscopic equations for the simulation of fluid flow. These models are different from the classical fluid dynamics computations based on the discretization of partial differential equations. They consider artificial micro-worlds of particles located on lattices with interactions that conserve mass and momentum. From their microdynamic Boolean rules macrodynamic equations can be obtained, which reduce in suitable limits to the incompressible Navier-Stokes equations[3].

For a better comparability of the results of the macrodynamic and the microdynamic approach, a standard example in fluid dynamics - the viscous flow driven by the tangential motion of the top boundary in a rectangular cavity - was to be simulated by all participating student teams.

The dynamics of incompressible fluids

The fundamental equations of fluid dynamics are based on the universal laws of conservation of mass, momentum and energy. The continuity equation results from applying the conservation of mass. The conservation of momentum is basically Newton's second law applied to a fluid passing through a control volume, and yields the Navier-Stokes equations. The conservation of energy eventually resembles the first law of thermodynamics. A simplification of the flow equations is obtained when considering the flow of an incompressible Newtonian fluid. Usually only liquids are regarded as incompressible fluids because they require high pressure to compress them appreciably. In many applications, however, it is possible to consider also gaseous media to be incompressible. In the case of our atmosphere the incompressible flow assumption typically holds well at low Mach numbers up to about 0.3. A similar restriction applies to lattice-gas models, which must be run at moderate Mach numbers to remain incompressible, and to avoid spurious high-order nonlinear terms.

The behavior of a viscous incompressible fluid is governed by the simplified Navier-Stokes equation, which can be written as

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} = -\frac{1}{\rho}\nabla p + \nu\,\Delta \mathbf{v} \tag{1}$$

and by the continuity equation (under the assumption of incompressibility):

$$\nabla \cdot \mathbf{v} = 0\,, \tag{2}$$

where **v** is the flow velocity, p the pressure, ρ the constant mass density, and ν the kinematic viscosity[4]. The Navier-Stokes equation is a coupled system of nonlinear partial differential equations, which excludes attaining an analytical solution except for a few cases. In general, numerical methods are required to simulate the behavior of real fluids.

Numerical Solution of the Navier-Stokes Equation

Finite Difference Method

The first numerical model for solving the incompressible Navier-Stokes equations was the Marker-and-Cell (MAC) method developed by *Harlow and Welch*[5]. In their model, the

incompressible flow field is discretized into a forward time finite difference form. By enforcing zero divergence of the velocity field (2) at both previous and current time steps, the pressure is solved in an iterative way. With this updated pressure, the velocity of the current time step can be derived. One branch of the later developed finite difference solvers, the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE), follows a similar idea. This method was first proposed by *Patankar and Spalding*[6] and is based on a pressure correction technique. In the project considered here the students were recommended to use SIMPLE and to follow the approach outlined in *Griebel et al.*[7], as briefly summarized below.

The starting point of the derivation of the finite difference equations is the conservative form of the Navier-Stokes equation for viscous incompressible flow in two-dimensional Cartesian coordinates. The momentum equations (1) are

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}+\frac{\partial p}{\partial x}=\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+g_x \tag{3}$$

$$\frac{\partial v}{\partial t}+\frac{\partial v^2}{\partial y}+\frac{\partial uv}{\partial x}+\frac{\partial p}{\partial y}=\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+g_y \tag{4}$$

and the continuity equation (2) is

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0. \tag{5}$$

The x and y coordinates are horizontal and vertical, respectively, and the corresponding components of the velocity are u and v. The pressure p is in fact the ratio of the real pressure and the constant density ρ. The body force, if considered in the simulations, is designated by the constant acceleration components g_x and g_y.

The finite difference approximation to the equations corresponds to an Eulerian mesh of cells with dimensions δx and δy covering the computation region. The cells are numbered by indices i and j which count cell center positions along the horizontal and vertical directions, respectively. For each cell the pressure averages $p_{i,j}$ are centered and the $u_{i,j}$ and $v_{i,j}$ averages are staggered in space by half a grid cell as shown in Figure 1. The advantage of such a staggered grid is the convenience to assure the divergence-free property of the numerical velocity field[7]. The forward difference form of the gradient of the velocity field (6) can thus be regarded as a central-difference estimate with half the step-size:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j}=\frac{u_{i,j}-u_{i-1,j}}{\delta x},\quad \left(\frac{\partial u}{\partial y}\right)_{i,j}=\frac{v_{i,j}-v_{i,j-1}}{\delta y}\quad . \tag{6}$$

A spatially averaging central-difference scheme for the discretization of the convective terms

$$\left(\frac{\partial (uv)}{\partial y}\right)_{i,j}=\frac{1}{\delta y}\left(\frac{(v_{i,j}+v_{i+1,j})}{2}\frac{(u_{i,j}+u_{i,j+1})}{2}-\frac{(v_{i,j-1}+v_{i+1,j-1})}{2}\frac{(u_{i,j-1}+u_{i,j})}{2}\right) \tag{7}$$

and

$$\left(\frac{\partial(u^2)}{\partial x}\right)_{i,j} = \frac{1}{\delta x}\left(\left(\frac{u_{i,j}+u_{i+1,j}}{2}\right)^2 - \left(\frac{u_{i-1,j}+u_{i,j}}{2}\right)^2\right) \tag{8}$$

is used. The approximation of the Laplacian of the velocity field can be calculated in a straightforward manner by the second-order central difference scheme

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\delta x)^2}. \tag{9}$$

Similar equations can be found for the other velocity component. The gradient of the pressure field is estimated within the same approximation as the gradient of the velocity field:

$$\left(\frac{\partial p}{\partial x}\right)_{i,j} = \frac{p_{i+1,j} - p_{i,j}}{\delta x}, \quad \left(\frac{\partial p}{\partial y}\right)_{i,j} = \frac{p_{i,j+1} - p_{i,j}}{\delta y} \tag{10}$$

In addition to the space index subscripts i and j a superscript (n) is used to number the time cycle. The forward difference form of the acceleration in x-direction is therefore

$$\left(\frac{\partial u}{\partial t}\right)^{(n+1)}_{i,j} = \frac{u^{(n+1)}_{i,j} - u^{(n)}_{i,j}}{\delta t}. \tag{11}$$

With these definitions the finite difference approximations to (3) and (4) can be written in the following form

$$u^{(n+1)} = \underbrace{u^{(n)} + \delta t\left[\nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x\right]}_{F} - \delta t \frac{\partial p}{\partial x} \tag{12}$$

$$v^{(n+1)} = \underbrace{v^{(n)} + \delta t\left[\nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial(v^2)}{\partial y} - \frac{\partial(uv)}{\partial x} + g_y\right]}_{G} - \delta t \frac{\partial p}{\partial y}, \tag{13}$$

and with the abbreviations

$$F_{i,j} = u_{i,j} + \delta t\left[\nu\left(\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} + \left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j}\right) - \left(\frac{\partial(u^2)}{\partial x}\right)_{i,j} - \left(\frac{\partial(uv)}{\partial y}\right)_{i,j} + g_x\right] \tag{14}$$

$$G_{i,j} = v_{i,j} + \delta t\left[\nu\left(\left(\frac{\partial^2 v}{\partial x^2}\right)_{i,j} + \left(\frac{\partial^2 v}{\partial y^2}\right)_{i,j}\right) - \left(\frac{\partial(v^2)}{\partial y}\right)_{i,j} - \left(\frac{\partial(uv)}{\partial x}\right)_{i,j} + g_y\right] \tag{15}$$

the discretization in time of the momentum equations reads:

$$u^{(n+1)} = F^{(n)} - \delta t \frac{\partial p^{(n+1)}}{\partial x} \tag{16}$$

$$v^{(n+1)} = G^{(n)} - \delta t \frac{\partial p^{(n+1)}}{\partial y}. \tag{17}$$

The solution of this couple of equations is not straightforward because an explicit equation for the pressure is not available. This kind of discretization is explicit in velocity, but implicit in pressure. One of the most common approaches is to derive an equation for the pressure by taking the divergence of the momentum equation and by substituting it in the continuity equation (5). This requires the solution of the Poisson equation (18), which may be accomplished by a relaxation technique like SOR or any other suitable procedure.

$$\frac{\partial^2 p^{(n+1)}}{\partial x^2}+\frac{\partial^2 p^{(n+1)}}{\partial y^2}=\frac{1}{\delta t}\left(\frac{\partial F^{(n)}}{\partial x}+\frac{\partial G^{(n)}}{\partial y}\right) \tag{18}$$

In SIMPLE the time step is limited by a CFL condition [8]. A save condition is that if $u_{\max}$ and $v_{\max}$ are the greatest components of velocity for any cell, then

$$\delta t<\frac{\nu}{2}\left(\frac{1}{(\delta x)^2}+\frac{1}{(\delta y)^2}\right)^{-1} \text{ and } \delta t<\frac{\delta x}{|u_{\max}|} \text{ and } \delta t<\frac{\delta y}{|v_{\max}|}. \tag{19}$$

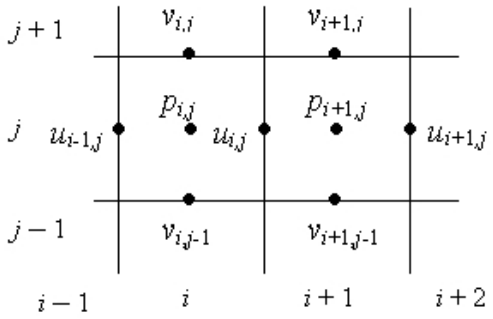

Figure 1: Field variable value placement about a computational cell. Velocities are defined at cell boundaries while pressures are defined at cell centers.

The implementation of wall boundary conditions (no-slip tangential and zero normal velocity) in a staggered grid requires the use of exterior cells. A vertical wall, for instance, does not pass through vertical-velocity mesh points; the calculation has to make use of the values of v lying just outside of the wall. For a no-slip wall the boundary condition requires the velocity v' beyond the wall to be $v' = -v$. Analogous boundary conditions are applied at a horizontal wall. Boundary conditions are also needed for the solution of the pressure equation (18). While it is not necessary to calculate the changes in normal velocities for points lying on the wall, the boundary conditions for p must be consistent with the identical vanishing of that velocity.

The Semi-Implicit Method for Pressure-Linked Equations allows the solution of the Navier-Stokes equation with an iterative procedure, which can be summed up as follows:

1. Set the boundary conditions
2. Choose δt according to (19)
3. Calculate the discretized equations (14) and (15) to compute the intermediate velocity field
4. Calculate the discretized right hand side of (18)
5. Solve the pressure equation (18) iteratively

6. Correct the velocities on the basis of the new pressure field in (16) and (17)
7. Update the boundary conditions
8. Repeat till $t = t_{end}$

Lattice-Gas Cellular Automata

The best-known way in which cellular automata were introduced was through work by *John von Neumann* and *Stanislaw Ulam* as an idealization of biological systems[9, 10], and can be regarded as mathematical idealizations of physical systems with discrete space and time and in which physical quantities can only take on a finite number of discrete values. A cellular automaton usually consists of a regular uniform lattice with a discrete variable at each site. The set of these variables completely specifies the state of the system. The temporal evolution of cellular automata takes place in discrete time steps, with the value of the variable at one site being affected by the values of the variables in its neighborhood. The variables at each site are updated simultaneously, based on the values of the variables of the adjacent lattice sites at the preceding time step, and according to a finite set of local rules[11].

Hardy, Pomeau and de Pazzis introduced the first lattice-gas cellular automaton model in 1973, which was later called the HPP model[12]. Their model is based on a two-dimensional square lattice and is of interest today mainly for historical reasons. The HPP model lacked rotational invariance, which made the model highly anisotropic, and it did not obey the desired hydrodynamic equations in the macroscopic limit. In 1986 *Frisch, Hasslacher and Pomeau*[13] showed that a lattice-gas cellular automata model over a lattice with a larger symmetry group than for the square lattice yields the incompressible Navier-Stokes equation in the macroscopic limit (FHP model).

The essential properties of the FHP model are[3]:

The underlying regular lattice shows hexagonal symmetry (see Figure 2). The nodes are linked to six nearest neighbors located all at the same distance with respect to the central node.

The vectors $\mathbf{c}_i$ linking nearest neighbor nodes are called lattice velocities (or lattice vectors)

$$\mathbf{c}_i = \left(\cos\left(\frac{\pi}{3} i\right), \sin\left(\frac{\pi}{3} i\right) \right),$$

with $|\mathbf{c}_i| = 1$ for all i (see Figure 2).

A cell is associated with each link at all nodes. The cells are either occupied or not. This exclusion principle is characteristic for all lattice-gas cellular automata.

The collisions are strictly local, i.e. only particles of a single cell are involved. The particles are of unit mass and indistinguishable.

The evolution in time proceeds by an alternation of collision and streaming (propagation).

The collision conserves both mass and momentum while changing the occupation of the cells.

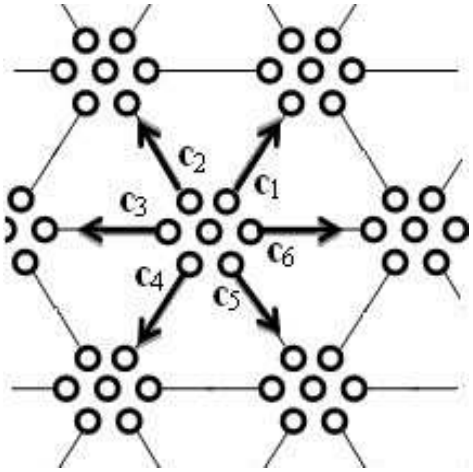

Figure 2: The triangular lattice of the FHP model shows hexagonal symmetry. The lattice velocities are represented by arrows. The circles at each node represent the seven channels corresponding to particles moving along the six directions of the triangular lattice and to the rest particle (center).

Several versions of the FHP model have been developed with the same geometrical structure but with different collision rules. The FHP-I model is a 6-bit model; each node has six channels, corresponding to the six directions of the triangular lattice $\mathbf{c}_i$. Due to their unit mass, the momentum $\mathbf{p}_i$ of each particle is equal to $\mathbf{c}_i$ and the kinetic energy is therefore equal to 0.5.

The evolution rule of the FHP model is at each time step δt a two-phase sequence: propagation and collision. During the propagation phase, a particle present at node $\mathbf{r}$ in channel i moves to node $\mathbf{r} + \mathbf{c}_i$, its nearest neighbor in the i-direction. During the collision phase, pairs of particles arriving at the same node from opposite directions undergo a binary collision with an output state rotated by + 60° or – 60° with probabilities p and $1 – p$ respectively, which adds some stochasticity to the FHP microdynamics. Most commonly p is chosen to be 0.5 (Figure 3).

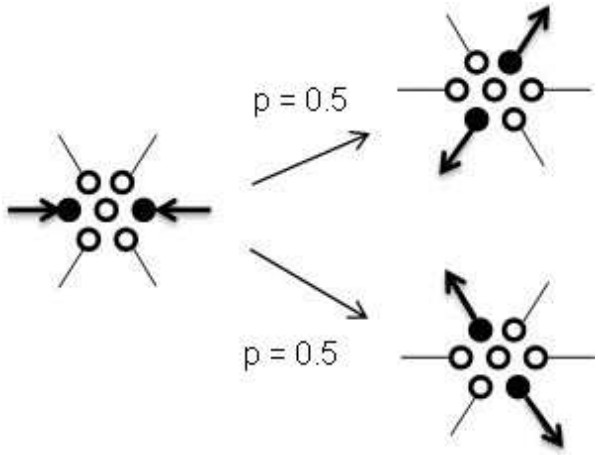

Figure 3: Example of a collision rule for the FHP model, reduced by symmetry. Filled circles denote occupied cells and open circles empty cells. In-state is shown on the left hand side, out-states on the right hand side.

Deterministic triple collisions were also included in the model to remove spurious symmetries which purely binary collisions would lead to. Another FHP variant, the 7-bit FHP-II model, includes the possibility of one rest particle per node, in addition to the six moving particles of FHP-I. Each node then has seven channels, corresponding to particles moving along the six directions of the triangular lattice and to the rest particle, which does not propagate to neighboring sites but is capable of colliding with other particles. FHP-I and FHP-II have the same microscopic properties; the essential difference lies in the number of possible effective collisions which is larger for FHP-II. A further variant of the FHP-II model is FHP-III where the collision rules are designed to include as many collisions as possible. It can be shown that 76 of the 128 possible states can undergo effective collision, which results in a collisional efficiency of about 59 % [14]. The corresponding macroscopic equations of the three FHP models have the same form and differ only in their viscosity coefficients.

The hexagonal FHP model was the first successful lattice-gas cellular automaton; its Boolean microdynamics reproduces macroscopic hydrodynamics. The Euler equation can be derived by a first order *Chapman-Enskog* expansion and the second order expansion yields the Navier-Stokes equation[3].

Implementation by the Students

Finite Difference Method

The lid-driven cavity is a well-known benchmark problem for the numerical simulation of viscous incompressible fluid flow. In this project, all student groups were dealing with a rectangular cavity consisting of three rigid walls with no-slip boundary conditions and a lid moving with a predefined constant tangential velocity. Although the simplicity of the geometry of the cavity and of the Dirichlet boundary conditions makes the problem easy to code, the implementation of the SIMPLE algorithm sketched above was a demanding task for the students.

In order to get an overview of the task the student teams sketched a 5 × 5 model of a staggered grid with paper and pencil (see Figure 4). This model has the minimum size when boundary conditions with exterior cells have to be implemented. A reference static pressure field as well as an initial velocity distribution was defined and a time step for the calculation was chosen according to (19). When the students tried the first computational step, i.e., the calculation of $F_{i,j}$ (14) and $G_{i,j}$ (15), they realized at first hand the absolute necessity of well-defined boundary conditions. Stokes flow theory is valid at the wall, i.e. the fluid adheres to the wall and moves with the wall velocity. As a consequence, the fluid velocity components equal the velocity of the wall, and the normal ($\mathbf{v}_n$) and tangential ($\mathbf{v}_t$) velocity components at an impermeable, non-moving wall are $\mathbf{v}_n = \mathbf{0}$ and $\mathbf{v}_t = \mathbf{v}_{wall} = \mathbf{0}$.

As the next step, the Poisson equation (18) was solved using the Maple computer algebra system. Pressure boundary conditions had to be determined in order to enable the calculation of the right hand side of (18). Figure 5 illustrates its calculation for $i = j = 2$ and time step $n = 1$ for the computational cell 10. With the thus modified pressure field the corrected velocities u (16) and v (17) of the next time step were calculated. In this way the whole SIMPLE procedure as described in the last chapter was reproduced manually. These calculations by hand made it easier for the students to implement the finite difference scheme in a C++ computer code.

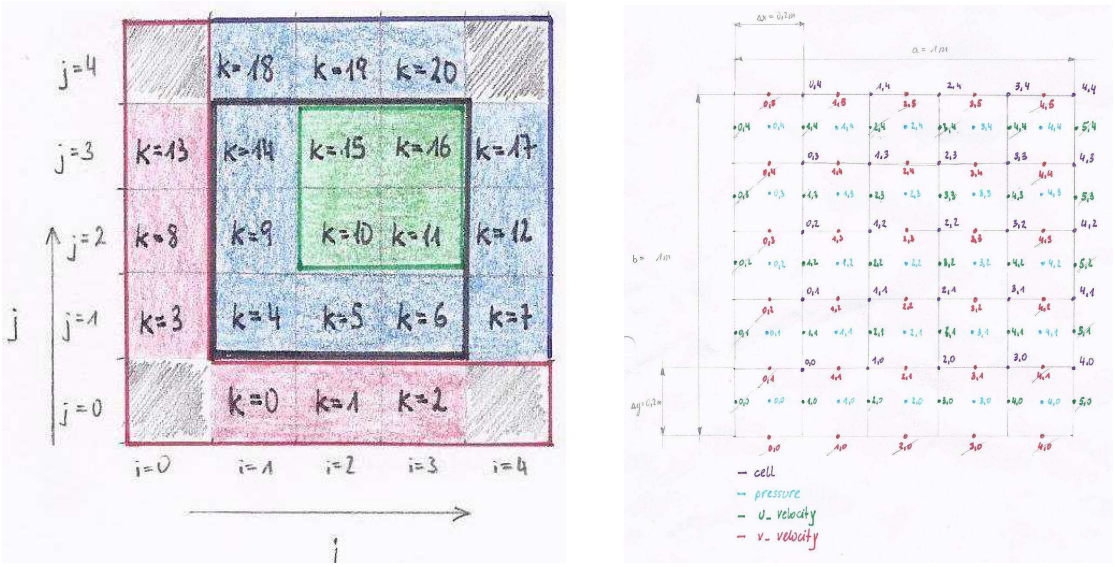

Figure 4: A 5 × 5 model of a staggered grid drafted by the students with paper and pencil[15].

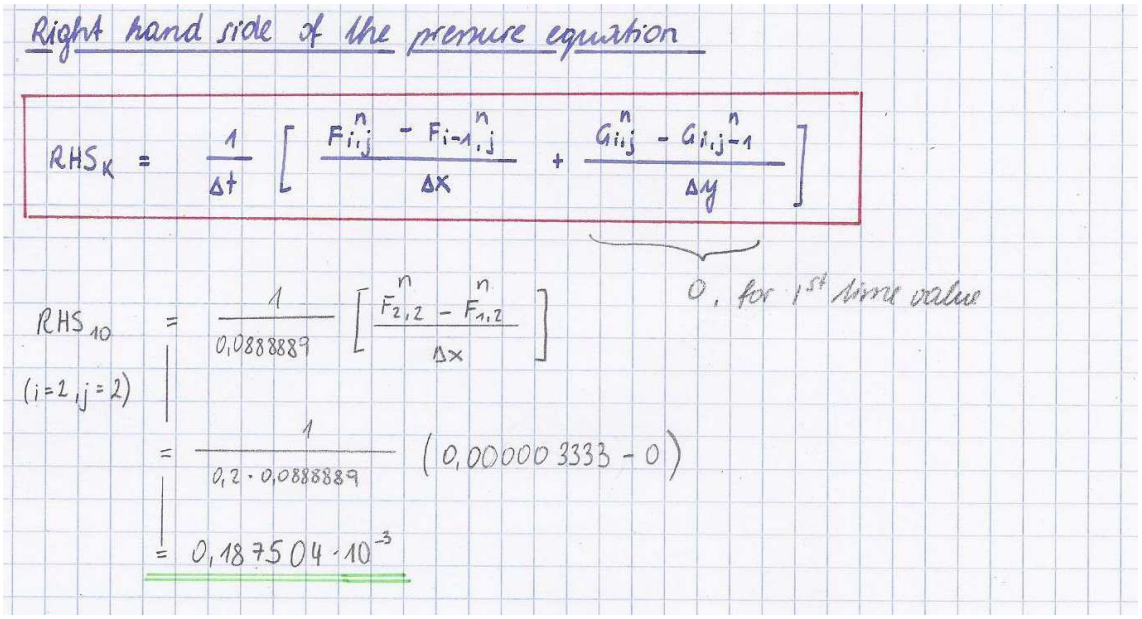

Figure 5: Manual calculation of the right-hand side of the pressure equation (18) for $i = j = 2$ and time step $n = 1$ for the computational cell 10 by the students[15].

In Figure 6 the result of a lid-driven cavity simulation is presented. The grid size is 25 × 25 and the grid points are illustrated by small disks. The resulting pressure field is represented by color or gray scale coding of these disks. The velocity of the fluid flow is indicated by an elongation of the respective disks in the direction of the flow and proportional to the speed. The flow field depicted in this figure has developed from a uniform initial field within about 15000 time steps.

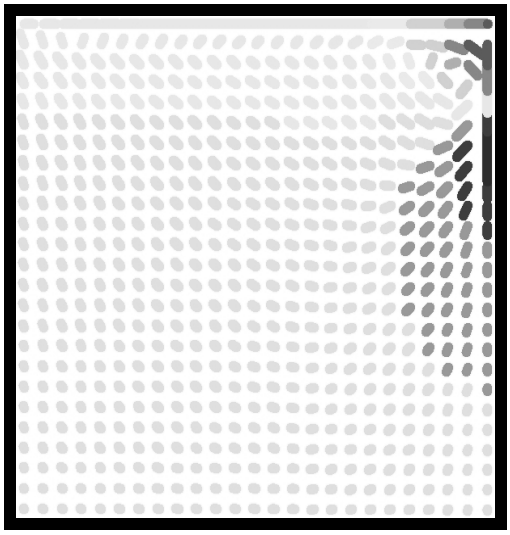

Figure 6: Illustration of the flow field in a driven cavity, as obtained by a finite difference simulation. The velocity field is represented by elongated grid points, the pressure field by gray scale values[16].

The calculation was terminated before a steady flow field could develop because the students struggled with stability problems of their code, which was most likely due to neglecting the upstream differencing method[3] or the incorrect implementation of boundary conditions.

Lattice-Gas Cellular Automata

When designing the computer program, the biggest hurdle for the students was to comprehend that in cellular automata solely information is handed over from node to node, and no physical movement of particles takes place[17]. As soon as this conceptual difficulty was overcome, the emphasis was shifted mainly to acquiring suitable programming techniques.

For a realistic representation of a flow field the cellular automaton needs considerably more lattice points than the finite difference approach. Therefore, a higher computation speed is desirable, which can best be realized with multi-spin coding. Instead of specifying a whole variable to the state of each lattice point a single bit is all that is needed. This is because the exclusion principle makes it possible to describe the state of the cells of the lattice-gas cellular automata by Boolean arrays $n_i(\mathbf{r},t)$, which are simply $n_i(\mathbf{r},t) = 1$ if the cell i is occupied, and 0 if the cell i is empty ($\mathbf{r}$ and t indicate the discrete points in space and time). A seven-bit variable is then sufficient to carry all the information at one site. The collision phase is processed by a routine that refers to a collision look-up table, which selects the out-state corresponding to the in-state for each site, and uses the one indicated by a random variable.

For a proper illustration of the flow field, macroscopic quantities like density and velocity have to be derived from the array $n_i(\mathbf{r},t)$. The calculation of these quantities for individual sites is subject to statistical noise, which should be attenuated by space averaging. Thus, both mass density and momentum density are obtained from the mean occupation numbers N_i of the lattice-gas cellular automaton, which are calculated by averaging over a user-defined number of neighboring nodes:

$$N_i(\mathbf{r},t) = \langle n_i(\mathbf{r},t) \rangle . \qquad (20)$$

These mean occupation numbers are then used to calculate the mass

$$\rho(\mathbf{r},t) = \sum_i N_i(\mathbf{r},t) \tag{21}$$

and momentum density

$$\mathbf{j}(\mathbf{r},t) = \sum_i N_i(\mathbf{r},t)\mathbf{c}_i . \tag{22}$$

The momentum density of the flow field of a driven cavity consisting of 512 × 513 lattice points is illustrated in Figure 7. Such a simulation typically runs through some hundreds of thousands of time steps. The momentum density of the field is represented by lines that are attached to a user-defined number of lattice points. The FHP model yielded under the given boundary conditions a steady solution with counter rotating vortices.

Flows with small velocities are usually laminar and tend to become turbulent at higher velocities. In addition, the transition from laminar to turbulent flows does also depend on the characteristic length of the geometry and on the kinematic viscosity v. From these parameters the Reynolds number $\mathrm{Re} = |\mathbf{v}|\cdot L/v$ is formed. All flows of the same type but with different values of $|\mathbf{v}|$, L and v are described by one and the same non-dimensional solution if their Reynolds numbers are equal. The dynamic similarity, as measured by the Reynolds number, provides the link between flows in the real world where length is measured in meters and the simulation of these flows with cellular automata over a lattice with unit grid length and unit lattice speed. In these models the viscosity is a dimensionless quantity.

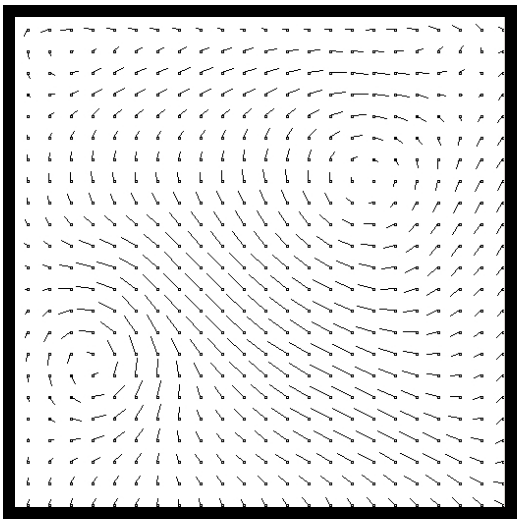

Figure 7: Illustration of the flow field in a driven cavity as obtained by the cellular automaton simulation. The momentum density field is represented by lines attached to a predefined number of lattice points[18].

As a rule of thumb, the viscosity coefficient decreases in a lattice-gas cellular automaton simulation with increasing number of collisions, which is largest for FHP-III (59 %) and

reduces to about 8 % for FHP-I. This leads to a smaller kinematic viscosity for FHP-III than for FHP-II and FHP-I, which allows simulations at a higher Reynolds number[14].

The cellular automaton team implemented several additional options going way beyond the minimum requirements, such as the simulation of flow around predefined obstacles or around objects that can be manually created with the help of a paint function. This was possible thanks to the particularly simple realization of boundary conditions in cellular automata.

Project progression

The students were introduced to the project at a kick-off meeting with the supervisors early in the semester where the supervisors presented the task. In addition, they received detailed information on the scope of the project, the timetable and deadlines and the evaluation criteria.

During the semester the students went through the following stages[1]:

1. Researching the topics
2. Finding the technical and mathematical solutions
3. Acquiring the relevant background knowledge and skills
4. Designing and programming the software
5. Documenting the process from research to development and finally to output
6. Reflecting on project management, team work and the performance of individual team members in the form of a brief written appraisal
7. Handing in software and documentation on a pre-defined date

The role of the project advisors, as subject experts, was to guide the students through these stages and give them the tools necessary for finding the missing pieces of, what for them was often a great puzzle, at least in the early stages.

The mathematical competencies needed to cope with the task were initially slightly beyond the students' skills. Having realized the gap in their knowledge of mathematical and programming methods, the students were eager to bridge it and asked for additional lectures. It was of utmost importance to design the lectures in such a way that the students' demands were satisfied, which led to an increased attentiveness as well as interaction and feedback during the lectures. Nevertheless, it should be stated that in the author's opinion a level of complexity has been reached within this project, which is only acceptable if excellent students can be recruited.

Conclusions

Within our project-based learning framework, sophomore students in their third semester of study developed computer programs for the simulation of a lid-driven cavity flow. Three groups of three or four students worked simultaneously and competitively on the implementation of a finite difference solution of the Navier-Stokes equation. Another team of three students approached the same problem by using a lattice-gas cellular automaton. All teams were advised to use VB.net for the Graphical User Interface and a C++ Dynamic Link

Library for the mathematical algorithms. The students were supported by their advisors with the essential information in regular meetings and, if necessary, by additional lectures. Initially, the students felt more comfortable with the macrodynamic model of the finite difference scheme than with the microdynamic cellular automaton model. However, when the conceptual difficulties were overcome the lattice-gas cellular automaton team made further and faster progress than the teams that tried to implement SIMPLE. The discretization of the partial differential equations was more difficult to handle for the students, not least because of the issues of their software's stability.

In summary, it can be said that the solution of the Navier-Stokes equation with the finite difference method allowed the students to get a deeper insight into numerical methods for the solution of partial differential equations, but caused them to invest considerable effort for a modest success.

The cellular automaton approach, on the other hand, needed essentially good programming skills while the mathematical requirements faded into the background. However, the results are rewarding with a convincing effort to performance ratio.

Acknowledgment

The authors would like to thank their students Peter Apolloner, Martin Brandstätter, Dieter Farthofer, Manuel Froihofer, Gerhard Hofer, Günther Holzer, Benjamin Huber, Robert Kalcher, Patrick Köllner, Christopher Lieb, Daniel Christoph Maier, Martin Schwarz, and Katja Wegerer for their high motivation in their project work, their creativity, and their excellent performance.

Bibliography

1. E. Bratschitsch, A. Casey, G. Bischof, and D. Rubeša, "3-phase multi subject project based learning as a didactical method in automotive engineering studies", Proceedings of the ASEE Annual Conference and Exposition, Honolulu, HI, June 24-27, 2007
2. G. Bischof, "Acoustic imaging of sound sources - a junior year student research project", 38th Annual Frontiers in Education Conference, Saratoga Springs, NY, Oct 22-25, 2008
3. D. Wolf-Gladrow, "Lattice gas cellular automata and lattice Boltzmann models: an introduction", Lecture notes in mathematics 1725, Springer, 2000
4. J. C. Tennehill, D. A. Anderson, and R. H. Pletcher, "Computational fluid mechanics and heat transfer", 2nd ed., PA: Taylor & Francis, 1997
5. F. H. Harlow and J. Eddie Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface", Phys. Fluids (American Institute of Physics) 8 (12), pp. 2182-2189, 1965
6. S. V. Patankar, "Numerical heat transfer and fluid flow" Taylor & Francis, 1980
7. M. Griebel, T. Dornseifer, and T. Neunhoeffer, "Numerische Simulation in der Strömungsmechanik" (in German), Vieweg, 1995
8. R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics", IBM Journal of Research and Development 11 (2), pp. 215-234, March 1967 [Translation of "Über die partiellen Differenzengleichungen der mathematischen Physik" (in German), Mathematische Annalen 100 (1), pp. 32-74, 1928]
9. J. von Neumann, "Theory of self-reproducing automata", edited by A. W. Burks, University of Illinois Press, Urbana, 1966
10. S. Ulam, "Some ideas and prospects in biomathematics", Ann. Rev. Bio. 12, pp. 255-257, 1974
11. S. Wolfram, "Statistical mechanics of cellular automata", Rev. Mod. Phys. 55 (3), pp. 601-644, 1983
12. J. Hardy, Y. Pomeau, and O. de Pazzis, "Time evolution of a two-dimensional model system: Invariant states and time correlation functions", J. Math. Phys. 14, pp. 1746-1759, 1973

13. U. Frisch, B. Hasslacher, and Y. Pomeau, "Lattice-gas automata for the Navier-Stokes equation", Phys. Rev. Lett. 56 (14), pp. 1505-1508, 1986
14. J.-P. Rivet and J. P. Boon, “Lattice gas hydrodynamics”, Cambridge University Press, 2001
15. G. Holzer, R. Kalcher, P. Köllner, and D. Maier, CFD solution for 2D incompressible fluids, Technical Report, FH Joanneum (2012)
16. C. Lieb, M. Froihofer, and M. Brandstätter, "Incompressible fluid simulation", Technical Report, FH Joanneum University of Applied Sciences, Graz, Austria, 2012
17. G. Bischof and C. Steinmann, “Fluid dynamics simulation using cellular automata” Proceedings of the ASEE Annual Conference and Exposition, San Antonio, TX, June 10-13, 2012
18. P. Apolloner, G. Hofer, and B. Huber, “CA - fluid simulation”, Technical Report, FH Joanneum University of Applied Sciences, Graz, Austria, 2012