TECHNISCHE UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:
Methoden und Werkzeuge für die Nutzung
paralleler Rechnerarchitekturen

Fluid Structure Interaction: 3D Numerical Simulation and Visualization of a Micropump

Hans-Joachim Bungartz, Anton Frank,
Florian Meier, Tilman Neunhoeffer,
Stefan Schulte

TUM-I9710
SFB-Bericht Nr. 342/06/97 A
April 97

TUM–INFO–04-I9710-350/1.–FI

Anforderungen an: Prof. Dr. A. Bode
Sprecher SFB 342
Institut für Informatik
Technische Universität München
D-80290 München, Germany

Druck: Fakultät für Informatik der
Technischen Universität München

Fluid Structure Interaction: 3 D Numerical Simulation and Visualization of a Micropump

Hans-Joachim Bungartz, Anton Frank, Florian Meier, Tilman Neunhoeffer, and Stefan Schulte
Institut für Informatik der Technischen Universität München
D - 80290 München, Germany

Summary

The numerical treatment of coupled problems has turned out to be one of the principal challenges in numerical simulation. Due to the increasing demands for accuracy, the interactions of different physical phenomena can no longer be neglected or simplified, but have to be taken into account in detail. On the other hand, because of the variety of applications and due to the fact that modeling such a kind of interaction always requires a lot of experience from different scientific areas, the problem specific modeling and simulation of a coupled problem very often seem to be a too time-consuming process. Therefore, modular concepts which allow to profit from existing subproblem models or codes, resp., and to integrate those into an efficient coupling algorithm have to be developed.

In this paper, we study a modular approach for the 3 D numerical simulation of the fluid structure interaction in a valve-driven micropump that has been designed for medical and environmental applications. For the structural part, we use ADINA and MSC/NASTRAN. The fluid flow simulation is done with the MAC-based Navier-Stokes solver NaSt3D which has been developed at our department. The coupling itself is realized via an outer iterative scheme.

1 Introduction

Together with the increasing possibilities of numerical simulation in science and engineering, the demands for both accuracy of the calculations and complexity of the problems to be tackled grow as well. More and more, not only single devices, but whole systems consisting of several components and involving different physical phenomena are in the centre of interest. This can be seen, especially, in the area of microsystem technology [2, 22], where microminiaturized sensors and actuators are based on physical effects like fluid mechanics, structural mechanics, heat transfer, or electromagnetics, e. g. Furthermore, here, aspects of scaling can lead to a dominance of surface effects (which are often responsible for interactions) over volume dependent effects. Finally, the high integration density in microsystems often results in unintentional cross effects (thermal strains in smart sensors or cross talks between different electric conductors, e. g.). Therefore, the numerical simulation of such interactions or couplings gains more and more in importance, and its focus has shifted from problems like reservoir dam interaction or reactor safety to microtechnology [2, 8, 22, 23, 25, 27–29, 33, 34].

In this paper, we study parts of a valve-driven micropump [33, 34], i. e. the fluid structure interaction which occurs at the valve flaps and which is responsible for their opening and closing. In order to avoid a completely new and comprehensive modeling of the micropump, we apply a modular solution strategy, the so-called *partitioned solution* [6, 7], which allows us to profit from existing simulation codes for both the fluid and the structural part [4, 20]. For such an approach, the crucial task is the choice of an appropriate outer iteration that links the different codes and finally leads to a solution of the coupled problem. At the moment, the quasi-stationary iteration scheme is based on Gauß-Seidel-type or fix point methods.

Whereas the structural part is not too hard in this case and can be solved by almost any code for structural mechanics, the simulation of the fluid flow requires a code that can efficiently deal with varying geometries. However, due to the small dimensions, we do not encounter turbulence. So far, all experiments have been done with the codes NaSt2D/NaSt3D [9] which solve the incompressible Navier-Stokes equations with a marker-and-cell discretization scheme.

The remainder of this article is organized as follows. In Section 2, we give a short overview of coupled problems and their numerical solution, and we present and discuss the modular approach of the partitioned solution. Section 3 provides the most important properties of our model of the valve-driven micropump. Due to its importance, the fluid code is presented in detail in Section 4. In Section 5, finally, we discuss some aspects concerning the implementation of a modular coupled simulation, we describe the application of the modular solution strategy to this special case of fluid structure interaction, and we show some results and visualizations of numerical experiments.

2 Coupled Problems and their Numerical Solution

The variety of examples for coupled problems from all kinds of areas of application (cf. Figure 1) has led to a very broad interpretation of the term coupled problems and to the development of quite different techniques for their numerical treatment [13, 15, 19, 35, 36]. A first kind of a unifying approach was done by Zienkiewicz in [35], where he defines a coupled problem to be a problem with bidirectional interactions of (usually different) physical effects without any possibility of an independent solution of a subproblem on its respective domain. Based on this definition, Zienkiewicz distinguishes between two classes of coupled problems: those with totally or partially overlapping domains and those with non-overlapping domains. In the first case, the coupling takes place via differential equations that, usually, refer to different underlying physical phenomena. The second class, however, is characterized by a coupling that occurs on the domain interfaces, i. e. via boundary conditions. Here, both problems with different underlying physics or different variables (fluid structure interaction, e. g.) and problems with identical physics and variables (structure structure interaction, e. g.) exist.

Concerning the modeling and numerical simulation of coupled

components / kind of coupling	structural dynamics	heat conduction	fluid dynamics	electromagnetics
structural dynamics		- resonators (thermally induced high frequency vibrations) - thermal actuators	- accelerometers - gyroscopes - valves, pumpes	- sensors with capacitive detection - electrostatic and piezoelectric actuators
heat conduction	thermo-elasticity h→s: material law s→h: energy dissipation		- microcooler - flow- and thermal sensors	- thermopiles (electrical heating with integrated resistors) - thermal actuators (pump actuation, resonators)
fluid dynamics	f→s: surface forces s→f: - geometry - boundary cond.	coupled by the energy equation		- sensors and pumps working with the electrohydrodynamic principle
electro-magnetics	e→s: surface and volume forces s→e: - geometry - moving conductor	e→h: electromagnetic loss h→e: material properties	electrohydrodynamics e→s: surface and volume forces s→e: - geometry - moving conductor	

Figure 1 Examples of interactions and corresponding devices

problems, two principal approaches have been established. A *joint* or *simultaneous* solution strategy combines the models of the different subproblems (either on the continuous or on the discrete level) via the respective coupling conditions in order to get a comprehensive model for the coupled problem, which is afterwards tackled numerically or solved, resp., as a whole (see [8, 30], e. g.). In contrast to that, the *partitioned* solution strategy [2, 6, 7, 23] is based on a strictly modular approach. There is no joint model, neither continuous nor discrete, but the coupled problem is solved by an *outer iteration* for the coupling and by almost arbitrary inner solution processes for each single subproblem (cf. Figure 2). The coupling is realized via changing boundary conditions, geometries, or parameters after each step of the outer iteration. Concerning the outer iteration, SOR-type methods are the state of the art [2, 4, 6, 7, 20]. However, the convergence behaviour of such methods can not be guaranteed in general [31]. Therefore, more robust iteration schemes have to be developed. Here, techniques for both linear [12] and nonlinear systems [17, 21] can be the starting point. As a first step, cg- and GMRES-based

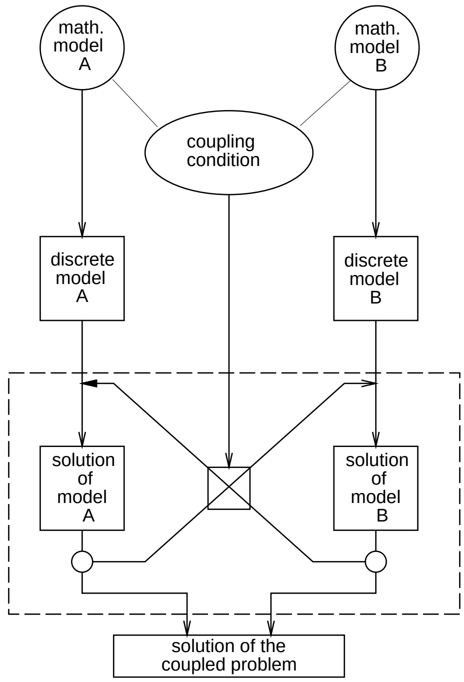

Figure 2 Partitioned solution

methods have been studied in [1]. Obviously, the main advantage of the partitioned solution scheme is its flexibility concerning both the problems to be solved and the codes to be used.

Besides the construction of the outer iteration scheme, there are the tasks of developing a suitable variable transformation at domain interfaces (in the case of non-matching discretizations, e. g.) and of the construction of error indicators and estimators for the overall procedure in order to be able to decide which subproblems have to be solved more exactly. Furthermore, the communication between the different solvers at the interfaces as well as the control of the overall procedure have to be organized, and an efficient implementation of a coupling environment has to be designed. A first and crucial task, however, is the identification and separation of appropriate mathematical models for the single effects and for the respective couplings. To this end, we define a coupled problem to consist of n subproblems $\mathcal{P}_i$, $1 \leq i \leq n$,

$$L_i[\Pi_i, \mathrm{K}_i](u_i) \;=\; 0 \qquad \text{on} \;\; \Omega_i, \tag{2.1}$$

and of m coupling conditions $\mathcal{C}_k$, $1 \leq k \leq m$,

$$C_k(u_1,...,u_n,\mathrm{K}_1,...,\mathrm{K}_n) \;=\; 0 \qquad \text{on} \;\; \Gamma_k. \tag{2.2}$$

Here, L_i denotes the operator of subproblem i, u_i the corresponding unknowns, and Ω_i the underlying domain. The operator L_i depends on two sets of parameters, the coupling ones (K_i) and the non-coupling ones (Π_i). The coupling equations (2.2) combine either the subproblem variables and the coupling parameters (parameter coupling: thermo-mechanical problems, e. g.) or the subproblem variables and changes in the underlying geometry (geometry coupling: fluid structure interaction, e. g.). The coupling domains or interfaces Γ_k are subsets of the union of the domains Ω_i of the involved subproblems $\mathcal{P}_i$. In the outer iteration scheme for the solution of the coupled problem, the subproblem equations (2.1) are solved by suitable available codes, whereas the coupling equations (2.2) have to be discretized and solved separately.

Finally, note that such a formulation opens the way to graph theoretical investigations, starting from a graph for the coupled problem where the nodes represent the subproblems $\mathcal{P}_i$ and the edges denote the couplings $\mathcal{C}_k$. Obviously, if n_k subproblems are involved in coupling $\mathcal{C}_k$, then $\mathcal{C}_k$ results in a complete subgraph of the respective n_k subproblems. If there are both uni- and bidirectional interactions, we have to work with directed graphs, if only bidirectional couplings occur, non-directed graphs are sufficient. A detailed analysis of a coupled problem's graph can give additional information concerning the complexity of the problem (are there cycles, is the graph reducible?) and, thus, some hints concerning the appropriate iterative scheme. Furthermore, in order to be able to choose a suitable and efficient outer iteration for a given coupled problem, the couplings have to be characterized further. Up to now, there exists no general concept of how to define and to determine weak or strong couplings and of how to distinguish between high-frequency couplings that need a very fine discretization of C_k and low-frequency couplings for which a few (coarse) grid points are sufficient. Those considerations are in the centre of our present work in this area.

3 The Micropump

Until now, several variants of micropumps have been developed differing mainly in the actuation unit, which is responsible for the fluid to flow, and in the regulation unit, which determines the flow's direction. For such pumps, the field of application ranges from medical drug infiltration systems [25] to chemical analysis in environmental technology. As we are primarily interested in fluid structure interactions, we decided to study a valve-driven micropump with passive check valves. Here, the interaction is crucial for the operation of the pump. Furthermore, there are a lot of experimental investigations and results [33, 34].

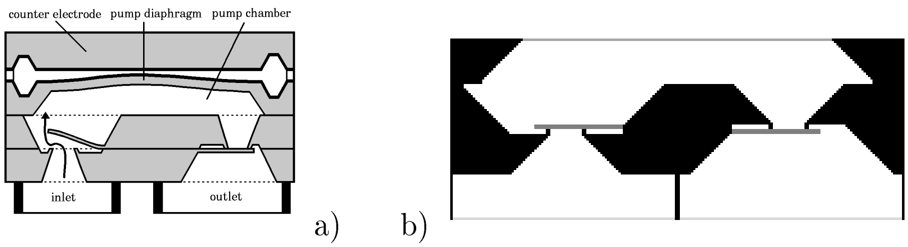

Figure 3 (a) Sketch of the structure of a valve-driven micropump [34], (b) geometric model used for the numerical simulation

The principal structure of a valve-driven micropump is shown in Figure 3. The inlet and the outlet (usually connected to the surrounding fluidic system via flexible hoses) are placed on the lower left-hand and on the lower right-hand side, resp. The upper part forms the actuation chamber where the fluid flow is induced by the oscillation of a diaphragm actuated by an external force (electrostatically, e. g.). When the membrane moves outward, the resulting low pressure causes the fluid to flow into the chamber. Moving inward, the membrane extrudes the fluid. In normal operation mode, the valves are bent in a way that regulates the flow from the inlet channel into the pump chamber and from the pump chamber into the outlet channel. The dimensions and some characteristic physical parameters of the pump described in [34] are itemized in Table 1.

Since our approach bases on the use of separate codes for the different physical disciplines, we have to consider the peculiarities of

fluid domain	
dimension	$7 \times 7 \times 2$ mm^3
material	ethanol
viscosity	$1.2 \cdot 10^{-3}$ Pa s
density	$7.89 \cdot 10^3$ kg/m^3

valve flaps	
dimension	$1.7 \times 1.0 \times 0.01$ mm^3
material	silicon
Young's modulus	$1.7 \cdot 10^{11}$ Pa
Poisson ratio	2.79

Table 1 Physical parameters of the fluid mechanical and the structural mechanical problem

each subproblem in order to select appropriate solvers. In our case, the elastomechanical problem is quite simple, since the dimension of this problem can be reduced due to the small gauge of the flaps. Thus, for a 3 D simulation, only the 2 D plate equation has to be solved, and in the reduced 2 D model of the pump, only the 1 D beam equation has to be considered. Therefore, almost any (commercial) code can be applied. In order to show the modularity of our approach, we chose ADINA for the 2 D computations and MSC/NASTRAN for the 3 D case.

The fluid flow is characterized by the use of ethanol, by the pump's small dimensions, and by the low occurring velocities. Thus, the fluid is viscous and incompressible, and the flow is laminar and transient. As an important requirement, the fluid solver should be able to deal with varying and moving obstacles in an efficient way. I. e., there should not be a significant overhead due to a repeated grid generation, e. g. Therefore, we chose the fluid code NaSt2D/NaSt3D, which will be described in detail in the following section.

4 The Fluid Model and Code

The fluid flow is described by the incompressible, transient Navier-Stokes equations

$$\rho\left(\frac{\partial \vec{v}}{\partial t} + \nabla(\vec{v} \otimes \vec{v})\right) = -\nabla p + \mu \triangle \vec{v} + \rho \vec{g}\,,^{1} \qquad (4.3)$$

$$\operatorname{div} \vec{v} = 0, \qquad (4.4)$$

where $\vec{v}$ denotes the velocity, p the pressure, $\vec{g}$ external forces (gravity, e. g.), ρ the constant density, and μ the dynamic viscosity of the fluid. Moreover, initial and boundary conditions such as no-slip, inflow, or outflow must be supplied.

Our CFD code NaSt2D/NaSt3D uses the marker-and-cell (MAC) scheme introduced in [14]. It is based on a finite difference discretization on a staggered equidistant grid. Although this scheme is very simple, it is quite flexible for describing complicated and moving fluid domains. For the discretization, the fluid domain is embedded into a rectangular basic domain, and the cells of this basic domain are flagged whether they are fluid cells, obstacle cells, or even empty cells for free boundary problems (see Figure 4). For the type of a cell, it is decisive in which domain the midpoint of the cell is situated.

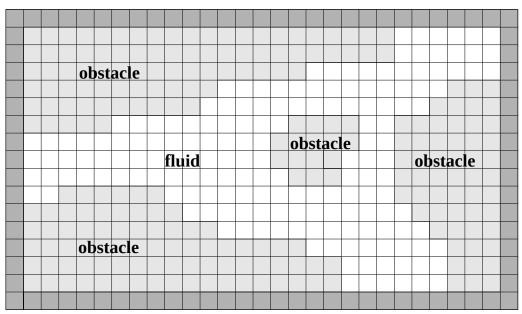

Figure 4 Embedding of the fluid domain into a rectangular domain

The time derivative in (4.3) is discretized by a simple explicit Euler step. In the time step $t_n \to t_{n+1}$, we first compute an intermediate velocity field $\vec{v}^*$ by means of the momentum equation (4.3) and by neglecting the pressure p:

$$\vec{v}^* = \vec{v}^{(n)} + \delta t \left(\frac{\mu}{\rho} \triangle \vec{v}^{(n)} - \nabla(\vec{v}^{(n)} \otimes \vec{v}^{(n)}) + \vec{g} \right). \tag{4.5}$$

[1] $\vec{v} \otimes \vec{v}$ denotes the tensor product of two vectors which gives a matrix with elements $(\vec{v} \otimes \vec{v})_{i,j} := v_i v_j$.

This intermediate velocity field is then corrected by the pressure gradient to obtain the velocity at time step t_{n+1}:

$$\vec{v}^{\,(n+1)} = \vec{v}^{\,*} - \frac{\delta t}{\rho} \nabla p^{(n+1)}. \tag{4.6}$$

Finally, to satisfy the continuity equation (4.4) for $\vec{v}^{\,(n+1)}$, the pressure $p^{(n+1)}$ has to fulfil

$$\triangle p^{(n+1)} = \frac{\rho}{\delta t} \nabla \cdot \vec{v}^{\,*}. \tag{4.7}$$

This Poisson equation is solved by successive over relaxation (SOR). At the boundary of the fluid domain, we set $\vec{v}^{\,*} := \vec{v}^{\,(n+1)}$, such that homogeneous Neumann boundary conditions $\partial p^{(n+1)}/\partial n = 0$ follow from multiplying (4.6) with the normal $\vec{n}$ at the boundary.

The discretization in space is done by central differences for the continuity equation (4.4) and by second order derivatives in the momentum equation (4.3), whereas the convective part is discretized by flux blending, i. e. a mixture of central and upwind differences. According to the staggered grid scheme, the continuity equation (4.4) is discretized at the cell centers, and the momentum equation is discretized at the midpoints of the cell faces (see [10] for details).

Moreover, some stability conditions on the step sizes δt in time and $\delta x, \delta y, \delta z$ in space must be satisfied, namely the Courant-Friedrichs-Lewy (CFL) conditions

$$\frac{2\delta t\, \mu}{\rho} < \left(\frac{1}{dx^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2} \right)^{-1},$$

$$|u_{max}|\, \delta t < \delta x, \qquad |v_{max}|\, \delta t < \delta y, \qquad |w_{max}|\, \delta t < \delta z,$$

where $|u_{max}|$, $|v_{max}|$, and $|w_{max}|$ denote the maximal absolute values of the velocity components u, v, and w, $\vec{v} = (u,v,w)^T$, occurring in the grid.

As boundary values, we need the velocity values at the boundary faces, i. e. at the faces between fluid cells and non-fluid cells, and at faces between two non-fluid cells adjacent to fluid cells. In the first version of NaSt2D/NaSt3D, we approximated the boundary of the fluid domain by a surface composed of cell faces, i. e. we set the velocity values at faces between fluid and non-fluid cells to zero for no-slip walls, for example. This method can be considered as a constant interpolation of the boundary values from the exact boundary to the boundary faces, which leads to an approximation order of $O(\delta x)$.

This simple scheme enables us to simulate fluid flow through complicated geometries like porous media (see Figure 5), the transport of chemical pollution (see Figure 6), temperature-driven flow (see Figure 7), and free boundary problems like the free surface flow over a backward facing step (see Figure 8).

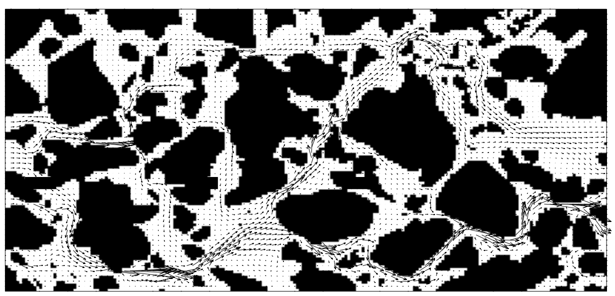

Figure 5 Flow through a porous medium (velocity field)

But for problems with moving obstacles like the valve flap, e. g., in order to avoid too small cells, the accuracy of our scheme has to be improved, because we can only simulate a discrete movement of the obstacles, i. e. we can only decide whether an obstacle moves a whole cell or is at rest.

Therefore, in [3] a better approximation of the boundary values was implemented which is based on the GENSMAC scheme [26]. Here, interpolation techniques are used to get the values on boundary faces from values at the exact boundary and nearest velocity values at faces between two fluid cells. Using central differences to discretize the convective terms of the momentum equations, an approximation order of $O(\delta x^2)$ results [3]. To describe the location of an obstacle, it is possible to use analytical functions, one- or two-dimensional

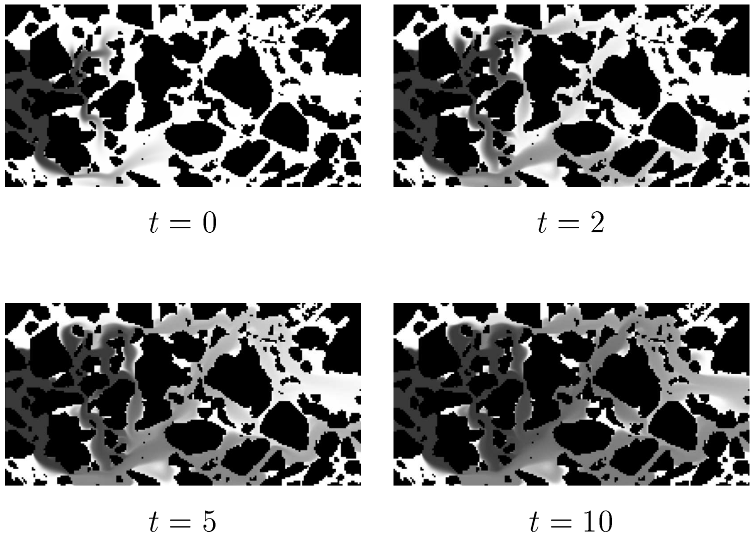

Figure 6 Pollution transport through a porous medium (time dependent concentrations, velocity field shown in Figure 5)

meshes in 2 D or 3 D, respectively, or volume of fluid techniques [16, 32].

Besides this extension concerning a better approximation of the boundary, some other work was done with NaSt2D/NaSt3D. In [11], an algebraic multigrid solver was implemented to solve the pressure equation and also the momentum equations in a semi-implicit time discretization of the momentum equations, the code was parallelized in two [10] and three space dimensions [18], and we are working on fluid flow problems with phase change, namely the solidification of an undercooled melt.

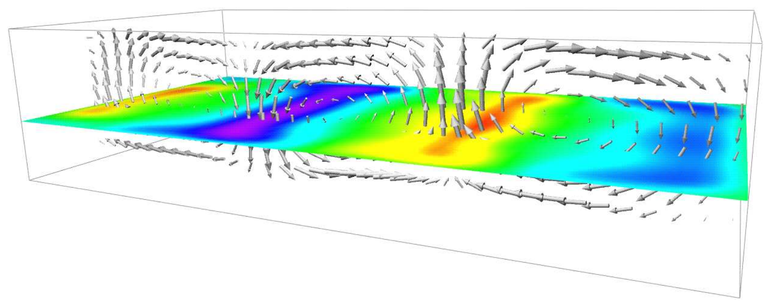

Figure 7 Rayleigh-Bénard flow: Flow driven by a hot bottom and a cold top wall (velocity field in a vertical, temperature distribution in a horizontal plane)

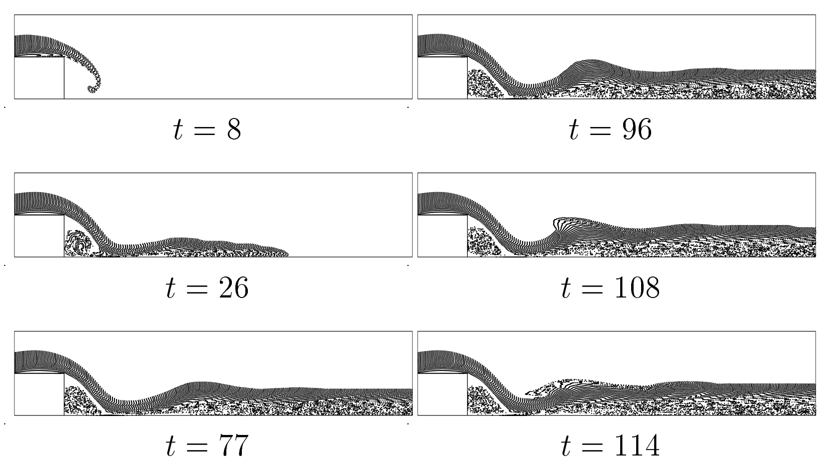

Figure 8 Free boundary flow over a backward facing step

5 Implementation, Experiments, and Visualization

The principle of the *partitioned solution* puts us in a position to avoid the full implementation of problem specific software by using a modular structure. Hence, we gain a better efficiency concerning flexibility and, therefore, modeling time and cost. Furthermore, the use of existing codes for the solution of the subproblems and for the visualization permits an adaptation to new problems and new developments without great efforts. First approaches in developing a modular software environment for the solution of coupled problems were carried out in the late 1970s [5], followed by the MEMCAD initiative [24], e. g. Increasing computing capacities and improved numerical algorithms add interesting new features. A combination with modern paradigms in computer science like parallel computing, client server systems, distribution over networks, and the usage of remote resources is quite promising. However, this requires to master heterogeneous computer, network, and software environments.

To handle heterogeneity, we designed an open environment able to control many quite different modules cooperating to solve coupled problems. With the separate tools, we are able to take advantage of distributed computing within a local area network without great efforts. The single modules can be assigned to five units (see Figure 9). The *input unit* collects and assembles the input data supplied by the user or by preceding software. The results are provided by the *output unit* for further processing. The *couple unit* is responsible for data maintenance as well as for the interaction and data exchange over the interfaces to the other units. It passes the appropriately converted data to the *tool unit*, which performs the required task (problem solving, visualization, e. g.). The couple unit is driven by the *control unit* according to the coupling algorithm. This algorithm is written in a script-like *couple language*, so that it can be easily modified.

The main problem in designing such a polymorphic environment is the definition of appropriate interfaces between the different units, especially for the integration of commercial software tools into the environment. Here, due to the prevalent unavailability of the source

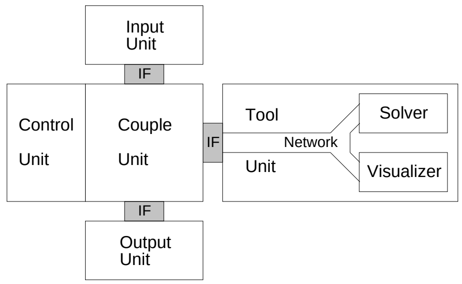

Figure 9 The five units of the modular system

code, manipulations therein are mostly impossible. For the combination of different data formats used in the various tools, it seems to be most appropriate to choose a unified representation of the parameters, the variables, and the geometry. In the prototype version, the data base in the centre of the system (i. e. the couple unit) consisting of the full data necessary for the simulation is realized as a file in tagged file format called *Coupled System Data Format (CSDF)*. As part of the CSDF, the geometry is represented as a structured quadrilateral grid or hexahedral cell structure, resp., to be equipped with a data structure that efficiently supports variations of the geometry. The physical parameters are transformed into standardized units (possibly in usage of reference values). Data necessary for visualization is also included. Hence, the visualization can be accomplished while or after processing the data, and an analysis or even an interactive modification of the problem parameters become possible.

According to our iterative coupling strategy, the mode of operation for the solution of a coupled problem consisting of n subproblems $\mathcal{P}_i$, $1 \le i \le n$, with the help of the respective solvers S_i has the following structure:

- collection and assembly of data with conversion into CSDF
- loop (outer iteration):
  - extraction and preparation of data for solver S_i, $1 \le i \le n$
  - solution of problem $\mathcal{P}_i$, $1 \le i \le n$
  - conversion and integration of results into CSDF
  - solution of coupling equations C_k, $1 \le k \le m$

- output of data in appropriate format for further processing

Now, let us turn to our example. In terms of the notation introduced in Section 2, the operators of the subproblems describing the fluid structure interaction are defined as follows:

- $L_1\,[\Pi_1, \{\tilde{p}\}]\,(u) := S[\Pi_1](u) - \tilde{p}$ in the fluid domain,
- $L_2\,[\Pi_2, \{\tilde{u}\}](p, v) := F_{\tilde{u}}[\Pi_2](p, v) - w_0$ in the flaps,
- $C_1(p, \tilde{p})$ extracting the load in the structural model from the pressure field in the fluid model,
- $C_2(u, \tilde{u})$ modifying the geometry of the fluid model according to the deflection of the structure.

Here, u represents the displacement of the flaps as the unknown for the structural problem and $\tilde{u}$ the displacement as a parameter of the fluid problem. p and v stand for the pressure and velocity of the fluid, resp., and $\tilde{p}$ for the load affecting the flaps. S denotes the model for structural mechanics, Π_1 the set of inner parameters for S containing boundary conditions and the Young's modulus, e. g. Finally, $F_{\tilde{u}}$ describes the model for fluid dynamics (depending on the displacement), Π_2 the set of inner parameters for $F_{\tilde{u}}$ like the viscosity and density, e. g., and w_0 the boundary conditions for the fluid flow.

As this system is not separable, we examined approaches like Block-Gauß-Seidel (i. e. alternate solution of the two subproblems) and Block-Jacobi iterations (i. e. concurrent solution of the subproblems). A close inspection of the Block-Jacobi scheme shows that it results in two complete Gauß-Seidel schemes not interacting with each other [20].

Using the Block-Gauß-Seidel scheme, the loop for the outer iteration in the mode of operation described above has to be adjusted accordingly. For the micropump, we start with both flaps closed and a given pressure and velocity distribution. The pressure difference at the flaps is regarded as a load impacting them and, therefore, is used as an input for the structural mechanics solver, which provides us with a new shape of the flaps' geometry. This new geometry is now passed to the fluid dynamics solver that calculates new pressure and velocity values according to the changes in geometry and boundary conditions, and so on. This procedure was first applied to a rudimentary 2 D model of a pump unit (see Figure 10 (left)) [20]. Afterwards, an improved model with a more realistic geometry in two and three dimensions (see Figure 10 (right)) and with parameters according to micromechanics was developed [4].

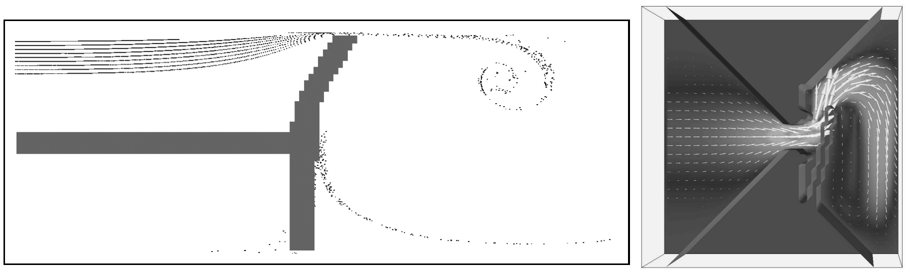

Figure 10 Draft 2 D geometry of a pump (left) and improved 3 D geometry of an inlet valve (right)

To model the time behaviour, we started with the interaction of a stationary fluid flow and a time independent flap displacement. In order to realize a transient simulation, a variation of the global boundary conditions (i. e. at the outer boundaries of the overall geometry) and the application of one outer iteration has to take place in every time step. Since the effects of these different modifications get mixed up that way, a closer survey of the particular influences is prevented. Therefore, we decided to pursue a quasi-stationary strategy first by introducing certain points at which the global boundary conditions are reconfigured. For each so-called *reconf point*, we apply a certain number of outer iterations (with fixed global boundary conditions), in order to mute the effects of occurring oscillations in the flap displacement. Due to the altered flap geometry, the Navier-Stokes equations have to be solved in each outer iteration. To accomplish this task, a varying amount of time steps with several SOR iterations for the pressure equation in the fluid solver is necessary. Hence, we get a hierarchy of four nested loops:

- inner iterations: solution of the pressure equation
- time steps: solution of the Navier-Stokes equations
- outer iterations: coupling of the interacting subproblems

- reconf points: modification of the global boundary conditions

Concerning the global boundary conditions, the influence of the actuation unit is modelled by velocity boundary conditions, while the changes of the geometry at the diaphragm are neglected, since its maximum expansion is only about 1/50 of the maximum flap displacement. The fluidic system connected to the pump unit is modelled by pressure boundary conditions for the global counter pressure [33]. The pressure declension between outlet and inlet has to be considered, since the behaviour of the pump depends on its value. Because of a still quite coarse approximation of the geometry, some details like the deflection of the flaps in reverse direction as well as their inertia are not yet taken into account.

As mentioned above, we use NaSt2D/NaSt3D for the solution of the fluid problem. For structural mechanics, a solver for the beam equation or ADINA are used for the 2 D case and MSC/NASTRAN for the 3 D case. The tools used for visualization are IDL for 2 D and IRIS EXPLORER for 3 D. Our visualizations focus on the fluid flow, since we are mainly interested in the behaviour of the fluid inside the pump chamber and in the pump's throughput. The representation of the corresponding quantities is done by various standard techniques as provided by the visualization packages including isolines/isosurfaces, vector fields, colour slices, and particle tracing (path- and streaklines, e. g.). The evolution in time is visualized by a series of single frames created at each reconf point with the techniques mentioned above that are combined to an animation sequence.

In our experiments, we use a maximum resolution of 200×80 cells for the 2 D case and $120 \times 60 \times 20$ cells for 3 D with a minimum mesh width of 25 μm. Since obstacle cells that are only adjoining at their corners are prohibited, the thickness of the flaps has to be at least two cells. However, these oversized flap extensions appear to have a negligible influence on the fluid flow. Following [33], for the global boundary conditions the pressure difference between outlet and inlet is set to 20 hPa, and the maximum velocity at the membrane is adjusted according to the pump frequency. We assume that the periodic behaviour of the membrane causes a sinusoidal alteration of the velocity at this boundary. To simulate a pump frequency of 25 Hz, for example, we subdivide each period by 20 reconf points with five outer iterations in-between. Hence, the time steps have to add up to $4 \cdot 10^{-4}$ s. Results for this configuration are shown in Figures 11 and 12 for 2 D. In order to reduce the extensive computation times, a coarser discretization in space and time was chosen for the 3 D case shown in Figure 13. Due to heavy oscillations in the flap motion, the deflection was restricted to a maximum displacement of one cell per outer iteration. As we perform several outer iterations for each reconf point, a sufficient mobility is guaranteed, nevertheless. The deflection of the flaps' tips at each outer iteration is shown in Figure 14. Note the obviously different behaviour of the two flaps.

Figure 11 Snap shots of particles during the inlet and the outlet phase

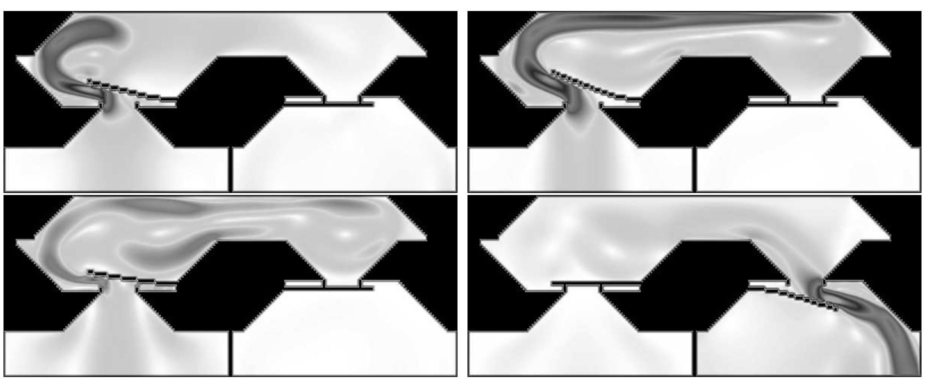

Figure 12 Four states in a period: absolute values of the velocity

The runtime for one outer iteration in the 2 D case takes about two minutes on a SGI Indy workstation (MIPS R4600PC, 100MHz). This leads to a total computing time of ten hours for three periods. The solution for the fluid problem consumes about 85 % of

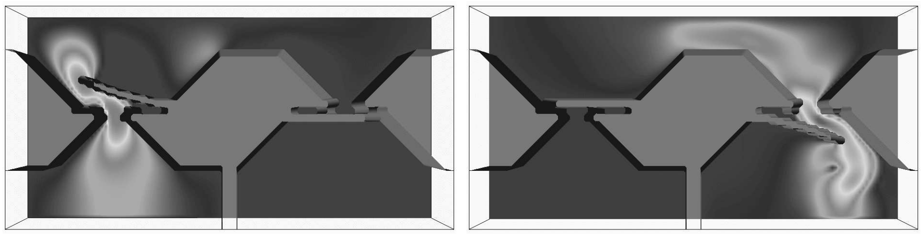

Figure 13 Absolute values of the velocity during the inlet and outlet phase (3 D)

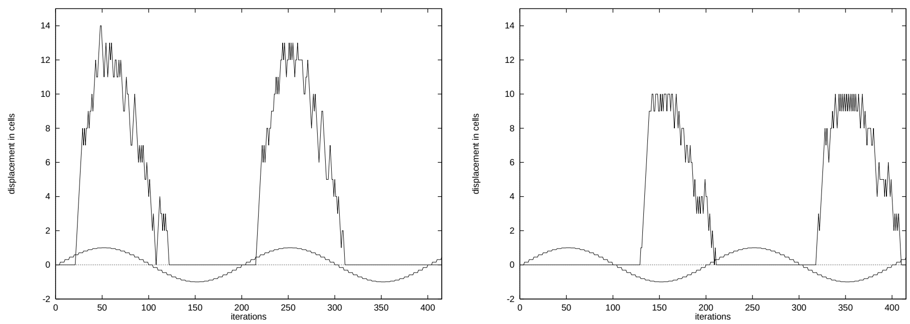

Figure 14 Displacement of the flap tips (1. and 2. period) at the inlet (left) and outlet (right) and progression of the velocity boundary conditions (normalized), simulating the oscillating membrane

the total computation time. As the structural mechanics problem is quite simple, the time for determining the new flap position is less than 5 %. The remaining 10 % are necessary for iteration procedures, coupling calculations, and data handling. Therefore, apart from the inevitable switch to a fully transient coupling, further optimizations have to concentrate on the solution of the fluid part. The improvements mentioned in Section 4 concerning a better approximation of the geometry, adaptivity of the solver, semi-implicit time discretization, and multigrid methods as well as parallelism are already examined and will be applied together with a usage of high performance computers in the near future. In spite of the strong imbalance in the complexity of the two subproblems, the examination of the fluid structure interaction in our example already shows the potential of the modular solution approach for the numerical simulation of coupled problems.

6 Concluding Remarks

In this article, we studied the fluid structure interaction of a valve-driven micropump. For the modular 3 D quasi-stationary numerical simulation of this coupled problem, we used the partitioned solution approach which is based on a separation of the solution of the subproblems (done by the structural solvers ADINA and NASTRAN and the fluid code NaSt3D) from the numerical treatment of the coupling equations in an outer iteration. The partitioned solution strategy reduces the modeling of a coupled problem to the modeling of the interaction itself, whereas, for the solution of each subproblem, existing mathematical models and codes can be used. First experiments with SOR-type outer iterations in the fluid structure case and with more general cg- and GMRES-based iterative schemes in the case of structure structure interaction show quite promising results.

For the future work, we will focus on a fully transient simulation and on the development of efficient and robust schemes for the outer iteration that allow to take into account the coupling characteristics of a given coupled problem.

Bibliography

[1] H.-J. Bungartz, R. Ebner, and S. Schulte, *Hierarchische Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik*, SFB-Report 342/05/97A, Institut für Informatik, TU München, (1997).

[2] H.-J. Bungartz and S. Schulte, *Coupled problems in microsystem technology*, in Numerical Treatment of Coupled Problems, vol. 51 of Notes on Numerical Fluid Mechanics, W. Hackbusch and G. Wittum, eds., Vieweg, 1995, pp. 11–24.

[3] M. Callies, *Verbesserte Randapproximation zur Strömungssimulation mit NaSt2D.* Fortgeschrittenenpraktikum, Institut für Informatik, TU München, 1997.

[4] U. Deisz, *Dreidimensionale numerische Simulation einer ventilgesteuerten Mikropumpe.* Diplomarbeit, Institut für Informatik, TU München, 1996.

[5] C. A. Felippa, *Architecture of a distributed analysis network for computational mechanics*, Computers & Structures, 13 (1981), pp. 405–413.

[6] C. A. Felippa and T. L. Geers, *Partitioned analysis for coupled mechanical systems*, Eng. Comput., 7 (1988), pp. 331–342.

[7] C. A. Felippa and K. C. Park, *Staggered transient analysis procedures for coupled mechanical systems*, Computer Methods in Applied Mechanics and Engineering, 24 (1980), pp. 61–111.

[8] H. Fujita and T. Ikoma, *Numerical determination of the electromechanical field for a micro servosystem*, Sensors and Actuators, A21-A23 (1990), pp. 215–218.

[9] M. Griebel, T. Dornseifer, and T. Neunhoeffer, *Numerische Simulation in der Strömungsmechanik – Eine praxisorientierte Einführung*, Vieweg, 1995.

[10] ——, *Numerical Simulation in Fluid Dynamics – a practical introduction*, SIAM, Philadelphia, 1997.

[11] M. Griebel, T. Neunhoeffer, and H. Regler, *Algebraic multigrid methods for the solution of the Navier-Stokes equations in complicated geometries*, Int. J. Num. Meth. Fluids, (to appear).

[12] W. Hackbusch, *Iterative Lösung großer schwachbesetzter Gleichungssysteme*, Teubner, Stuttgart, 1991.

[13] W. Hackbusch and G. Wittum, eds., *Numerical Treatment of Coupled Problems, vol. 51 of Notes on Numerical Fluid Mechanics*, Vieweg, 1995.

[14] F. Harlow and J. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, The Physics of Fluids, 8 (1965), pp. 2182–2189.

[15] E. Hinton, P. Bettess, and R. W. Lewis, eds., *Numerical Methods for Coupled Problems — Proceedings of the International Conference Held at the University College, Swansea, 7.–11. 9. 1981*, Pineridge Press, 1981.

[16] C. Hirt and B. Nichols, *Volume of fluid (VOF) method for the dynamics of free boundaries*, J. Comp. Phys., 39 (1981), pp. 201–225.

[17] T. Kerkhoven and Y. Saad, *On acceleration methods for coupled nonlinear elliptic systems*, Numerische Mathematik, (1992), pp. 525–548.

[18] R. Kreissl and M. Rykaschewski, *Parallelisierung eines 3D-Codes zur numerischen Simulation nichtstationärer Strömungen und thermischer Effekte in inkompressiblen laminaren Medien.* Fortgeschrittenenpraktikum, Institut für Informatik, TU München, 1996.

[19] R. W. Lewis, P. Bettess, and E. Hinton, eds., *Numerical Methods in Coupled Systems*, John Wiley & Sons, 1984.

[20] F. Meier, *Numerische Simulation von Fluid-Struktur-Wechselwirkungen am Beispiel einer ventilgesteuerten Mikropumpe.* Diplomarbeit, Institut für Informatik, TU München, 1995.

[21] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, Inc., 1970.

[22] S. Schulte, *Simulation of cross coupled effects in physical sensors*, in MICRO SYSTEM Technologies' 94, 4th Int. Conf. on Micro Electro, Opto, Mechanical Systems and Components, Berlin, October 19-21, 1994, H. Reichl and A. Heuberger, eds., vde-verlag gmbh Berlin, Offenbach, 1994, pp. 833–842.

[23] S. Schulte, A. Maurer, and H.-J. Bungartz, *A modular solution approach for the simulation of coupled physical phenomena*, in Simulation and Design of Microsystems and Microstructures, R. A. Adey, A. Lahrmann, and C. Leßmöhlmann, eds., Computational Mechanics Publications, Southampton, 1995, pp. 201–210.

[24] S. Senturia, R. Harris, B. Johnson, S. Kim, and K. Nabors, *A computer-aided design system for microelectromechanical systems (MEMCAD)*, Journal of Microelectromechanical Systems, 1 (1992), pp. 3–13.

[25] W. Spencer, W. Corbett, L. Dominguez, and B. Shafer, *An electronically controlled piezoelectric insulin pump and valves*, IEEE Transactions On Sonics And Ultrasonics, SU-25 (1978), pp. 153–156.

[26] M. Tome and S. McKee, *GENSMAC: A computational marker and cell method for free surface flows in general domains*, J. Comp. Phys., 110 (1994), pp. 171–186.

[27] H.-P. TRAH, H. BAUMANN, C. DÖRING, H. GOEBEL, T. GRAUER, AND M. METTNER, *Micromachined valve with hydraulically actuated membrane subsequent to a thermoelectrically controlled bimorph cantilever*, Sensors and Actuators A, 39 (1993), pp. 169–176.

[28] T. TSCHAN, *Simulation, Design and Characterization of a Silicon Piezoresistive Accelerometer, Fabricated by a Bipolar-Compatible Industrial Process*, PhD thesis, University of Neuchatel, Switzerland, 1992.

[29] P. VOIGT, G. SCHRAG, AND G. WACHUTKA, *Micropump macromodel for standard circuit simulators using HDL-A*, in Proc. 10^{th} Europ. Conf. on Solid-State Transducers, R. Puers, ed., vol. 5, Timshel BVBA, Leuven, 1996, pp. C5–2.

[30] G. WACHUTKA, *Tailored modeling of miniaturized electrothermomechanical systems using thermodynamic methods*, in Micromechanical Systems, D. Cho, J. Peterson, A. Pisano, and C. Friedrich, eds., vol. 40, 1992.

[31] H. YIE, X. CAI, AND J. WHITE, *Convergence properties of relaxation versus the surface-Newton generalized-conjugate residual algorithm for self-consistent electromechanical analysis of 3-d micro-electro-mechanical structures*, in International Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits: NUPAD V, Honolulu, IEEE 0 7803 1867 6, 1994, pp. 137–140.

[32] D. L. YOUNGS, *Time-dependent multi-material flow with large fluid distortion*, in Numerical Methods for Fluid Dynamics, K. W. Morton and M. J. Baines, eds., 1982, pp. 273–285.

[33] R. ZENGERLE, *Mikro-Membranpumpen als Komponenten für Mikro-Fluidsysteme*, PhD thesis, Universität der Bundeswehr München, 1994.

[34] R. ZENGERLE, S. KLUGE, M. RICHTER, AND A. RICHTER, *A bidirectional silicon micropump.* To be published, 1995.

[35] O. C. ZIENKIEWICZ, *Coupled problems and their numerical solution*, in Numerical Methods in Coupled Systems, R. W. Lewis, P. Bettess, and E. Hinton, eds., John Wiley & Sons, 1984.

[36] O. C. ZIENKIEWICZ AND A. H. C. CHAN, *Coupled problems and their numerical solution*, in Advances in Computational Nonlinear Mechanics, L. S. Doltsinis, ed., Springer, Berlin, 1988, pp. 139–176.

SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

342/1/90 A	Robert Gold, Walter Vogler: Quality Criteria for Partial Order Semantics of Place/Transition-Nets, Januar 1990
342/2/90 A	Reinhard Fößmeier: Die Rolle der Lastverteilung bei der numerischen Parallelprogrammierung, Februar 1990
342/3/90 A	Klaus-Jörn Lange, Peter Rossmanith: Two Results on Unambiguous Circuits, Februar 1990
342/4/90 A	Michael Griebel: Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen Transformations-Mehrgitter-Methode
342/5/90 A	Reinhold Letz, Johann Schumann, Stephan Bayerl, Wolfgang Bibel: SETHEO: A High-Performance Theorem Prover
342/6/90 A	Johann Schumann, Reinhold Letz: PARTHEO: A High Performance Parallel Theorem Prover
342/7/90 A	Johann Schumann, Norbert Trapp, Martin van der Koelen: SETHEO/PARTHEO Users Manual
342/8/90 A	Christian Suttner, Wolfgang Ertel: Using Connectionist Networks for Guiding the Search of a Theorem Prover
342/9/90 A	Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Jaroslav Kremenek, Robert Lindhof, Thomas Ludwig, Peter Luksch, Thomas Treml: TOPSYS, Tools for Parallel Systems (Artikelsammlung)
342/10/90 A	Walter Vogler: Bisimulation and Action Refinement
342/11/90 A	Jörg Desel, Javier Esparza: Reachability in Reversible Free- Choice Systems
342/12/90 A	Rob van Glabbeek, Ursula Goltz: Equivalences and Refinement
342/13/90 A	Rob van Glabbeek: The Linear Time - Branching Time Spectrum
342/14/90 A	Johannes Bauer, Thomas Bemmerl, Thomas Treml: Leistungsanalyse von verteilten Beobachtungs- und Bewertungswerkzeugen
342/15/90 A	Peter Rossmanith: The Owner Concept for PRAMs
342/16/90 A	G. Böckle, S. Trosch: A Simulator for VLIW-Architectures
342/17/90 A	P. Slavkovsky, U. Rüde: Schnellere Berechnung klassischer Matrix-Multiplikationen
342/18/90 A	Christoph Zenger: Sparse Grids
342/19/90 A	Michael Griebel, Michael Schneider, Christoph Zenger: A combination technique for the solution of sparse grid problems

Reihe A

342/20/90 A	Michael Griebel: A Parallelizable and Vectorizable Multi- Level-Algorithm on Sparse Grids
342/21/90 A	V. Diekert, E. Ochmanski, K. Reinhardt: On confluent semi-commutations-decidability and complexity results
342/22/90 A	Manfred Broy, Claus Dendorfer: Functional Modelling of Operating System Structures by Timed Higher Order Stream Processing Functions
342/23/90 A	Rob van Glabbeek, Ursula Goltz: A Deadlock-sensitive Congruence for Action Refinement
342/24/90 A	Manfred Broy: On the Design and Verification of a Simple Distributed Spanning Tree Algorithm
342/25/90 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Peter Luksch, Roland Wismüller: TOPSYS - Tools for Parallel Systems (User's Overview and User's Manuals)
342/26/90 A	Thomas Bemmerl, Arndt Bode, Thomas Ludwig, Stefan Tritscher: MMK - Multiprocessor Multitasking Kernel (User's Guide and User's Reference Manual)
342/27/90 A	Wolfgang Ertel: Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems
342/28/90 A	Rob van Glabbeek, Frits Vaandrager: Modular Specification of Process Algebras
342/29/90 A	Rob van Glabbeek, Peter Weijland: Branching Time and Abstraction in Bisimulation Semantics
342/30/90 A	Michael Griebel: Parallel Multigrid Methods on Sparse Grids
342/31/90 A	Rolf Niedermeier, Peter Rossmanith: Unambiguous Simulations of Auxiliary Pushdown Automata and Circuits
342/32/90 A	Inga Niepel, Peter Rossmanith: Uniform Circuits and Exclusive Read PRAMs
342/33/90 A	Dr. Hermann Hellwagner: A Survey of Virtually Shared Memory Schemes
342/1/91 A	Walter Vogler: Is Partial Order Semantics Necessary for Action Refinement?
342/2/91 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Rainer Weber: Characterizing the Behaviour of Reactive Systems by Trace Sets
342/3/91 A	Ulrich Furbach, Christian Suttner, Bertram Fronhöfer: Massively Parallel Inference Systems
342/4/91 A	Rudolf Bayer: Non-deterministic Computing, Transactions and Recursive Atomicity
342/5/91 A	Robert Gold: Dataflow semantics for Petri nets
342/6/91 A	A. Heise; C. Dimitrovici: Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften
342/7/91 A	Walter Vogler: Asynchronous Communication of Petri Nets and the Refinement of Transitions

Reihe A

342/8/91 A	Walter Vogler: Generalized OM-Bisimulation
342/9/91 A	Christoph Zenger, Klaus Hallatschek: Fouriertransformation auf dünnen Gittern mit hierarchischen Basen
342/10/91 A	Erwin Loibl, Hans Obermaier, Markus Pawlowski: Towards Parallelism in a Relational Database System
342/11/91 A	Michael Werner: Implementierung von Algorithmen zur Kompaktifizierung von Programmen für VLIW-Architekturen
342/12/91 A	Reiner Müller: Implementierung von Algorithmen zur Optimierung von Schleifen mit Hilfe von Software-Pipelining Techniken
342/13/91 A	Sally Baker, Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Udo Graf, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Rainer Knödlseder, Jaroslav Kremenek, Siegfried Langenbuch, Robert Lindhof, Thomas Ludwig, Peter Luksch, Roy Milner, Bernhard Ries, Thomas Treml: TOPSYS - Tools for Parallel Systems (Artikelsammlung); 2., erweiterte Auflage
342/14/91 A	Michael Griebel: The combination technique for the sparse grid solution of PDE's on multiprocessor machines
342/15/91 A	Thomas F. Gritzner, Manfred Broy: A Link Between Process Algebras and Abstract Relation Algebras?
342/16/91 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Thomas Treml, Roland Wismüller: The Design and Implementation of TOPSYS
342/17/91 A	Ulrich Furbach: Answers for disjunctive logic programs
342/18/91 A	Ulrich Furbach: Splitting as a source of parallelism in disjunctive logic programs
342/19/91 A	Gerhard W. Zumbusch: Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme
342/20/91 A	M. Jobmann, J. Schumann: Modelling and Performance Analysis of a Parallel Theorem Prover
342/21/91 A	Hans-Joachim Bungartz: An Adaptive Poisson Solver Using Hierarchical Bases and Sparse Grids
342/22/91 A	Wolfgang Ertel, Theodor Gemenis, Johann M. Ph. Schumann, Christian B. Suttner, Rainer Weber, Zongyan Qiu: Formalisms and Languages for Specifying Parallel Inference Systems
342/23/91 A	Astrid Kiehn: Local and Global Causes
342/24/91 A	Johann M.Ph. Schumann: Parallelization of Inference Systems by using an Abstract Machine
342/25/91 A	Eike Jessen: Speedup Analysis by Hierarchical Load Decomposition
342/26/91 A	Thomas F. Gritzner: A Simple Toy Example of a Distributed System: On the Design of a Connecting Switch
342/27/91 A	Thomas Schnekenburger, Andreas Weininger, Michael Friedrich: Introduction to the Parallel and Distributed Programming Language ParMod-C

Reihe A

342/28/91 A	Claus Dendorfer: Funktionale Modellierung eines Postsystems
342/29/91 A	Michael Griebel: Multilevel algorithms considered as iterative methods on indefinite systems
342/30/91 A	W. Reisig: Parallel Composition of Liveness
342/31/91 A	Thomas Bemmerl, Christian Kasperbauer, Martin Mairandres, Bernhard Ries: Programming Tools for Distributed Multiprocessor Computing Environments
342/32/91 A	Frank Leßke: On constructive specifications of abstract data types using temporal logic
342/1/92 A	L. Kanal, C.B. Suttner (Editors): Informal Proceedings of the Workshop on Parallel Processing for AI
342/2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS
342/2-2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS - Revised Version (erschienen im Januar 1993)
342/3/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/4/92 A	Claus Dendorfer, Rainer Weber: Development and Implementation of a Communication Protocol - An Exercise in FOCUS
342/5/92 A	Michael Friedrich: Sprachmittel und Werkzeuge zur Unterstüt- zung paralleler und verteilter Programmierung
342/6/92 A	Thomas F. Gritzner: The Action Graph Model as a Link between Abstract Relation Algebras and Process-Algebraic Specifications
342/7/92 A	Sergei Gorlatch: Parallel Program Development for a Recursive Numerical Algorithm: a Case Study
342/8/92 A	Henning Spruth, Georg Sigl, Frank Johannes: Parallel Algorithms for Slicing Based Final Placement
342/9/92 A	Herbert Bauer, Christian Sporrer, Thomas Krodel: On Distributed Logic Simulation Using Time Warp
342/10/92 A	H. Bungartz, M. Griebel, U. Rüde: Extrapolation, Combination and Sparse Grid Techniques for Elliptic Boundary Value Problems
342/11/92 A	M. Griebel, W. Huber, U. Rüde, T. Störtkuhl: The Combination Technique for Parallel Sparse-Grid-Preconditioning and -Solution of PDEs on Multiprocessor Machines and Workstation Networks
342/12/92 A	Rolf Niedermeier, Peter Rossmanith: Optimal Parallel Algorithms for Computing Recursively Defined Functions
342/13/92 A	Rainer Weber: Eine Methodik für die formale Anforderungsspezifkation verteilter Systeme
342/14/92 A	Michael Griebel: Grid– and point–oriented multilevel algorithms

Reihe A

342/15/92 A	M. Griebel, C. Zenger, S. Zimmer: Improved multilevel algorithms for full and sparse grid problems
342/16/92 A	J. Desel, D. Gomm, E. Kindler, B. Paech, R. Walter: Bausteine eines kompositionalen Beweiskalküls für netzmodellierte Systeme
342/17/92 A	Frank Dederichs: Transformation verteilter Systeme: Von applikativen zu prozeduralen Darstellungen
342/18/92 A	Andreas Listl, Markus Pawlowski: Parallel Cache Management of a RDBMS
342/19/92 A	Erwin Loibl, Markus Pawlowski, Christian Roth: PART: A Parallel Relational Toolbox as Basis for the Optimization and Interpretation of Parallel Queries
342/20/92 A	Jörg Desel, Wolfgang Reisig: The Synthesis Problem of Petri Nets
342/21/92 A	Robert Balder, Christoph Zenger: The d-dimensional Helmholtz equation on sparse Grids
342/22/92 A	Ilko Michler: Neuronale Netzwerk-Paradigmen zum Erlernen von Heuristiken
342/23/92 A	Wolfgang Reisig: Elements of a Temporal Logic. Coping with Concurrency
342/24/92 A	T. Störtkuhl, Chr. Zenger, S. Zimmer: An asymptotic solution for the singularity at the angular point of the lid driven cavity
342/25/92 A	Ekkart Kindler: Invariants, Compositionality and Substitution
342/26/92 A	Thomas Bonk, Ulrich Rüde: Performance Analysis and Optimization of Numerically Intensive Programs
342/1/93 A	M. Griebel, V. Thurner: The Efficient Solution of Fluid Dynamics Problems by the Combination Technique
342/2/93 A	Ketil Stølen, Frank Dederichs, Rainer Weber: Assumption / Commit ment Rules for Networks of Asynchronously Communicating Agents
342/3/93 A	Thomas Schnekenburger: A Definition of Efficiency of Parallel Programs in Multi-Tasking Environments
342/4/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: A Proof of Convergence for the Combination Technique for the Laplace Equation Using Tools of Symbolic Computation
342/5/93 A	Manfred Kunde, Rolf Niedermeier, Peter Rossmanith: Faster Sorting and Routing on Grids with Diagonals
342/6/93 A	Michael Griebel, Peter Oswald: Remarks on the Abstract Theory of Additive and Multiplicative Schwarz Algorithms
342/7/93 A	Christian Sporrer, Herbert Bauer: Corolla Partitioning for Distributed Logic Simulation of VLSI Circuits
342/8/93 A	Herbert Bauer, Christian Sporrer: Reducing Rollback Overhead in Time-Warp Based Distributed Simulation with Optimized Incremental State Saving

342/9/93 A	Peter Slavkovsky: The Visibility Problem for Single-Valued Surface (z = f(x,y)): The Analysis and the Parallelization of Algorithms
342/10/93 A	Ulrich Rüde: Multilevel, Extrapolation, and Sparse Grid Methods
342/11/93 A	Hans Regler, Ulrich Rüde: Layout Optimization with Algebraic Multigrid Methods
342/12/93 A	Dieter Barnard, Angelika Mader: Model Checking for the Modal Mu-Calculus using Gauß Elimination
342/13/93 A	Christoph Pflaum, Ulrich Rüde: Gauß' Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids
342/14/93 A	Christoph Pflaum: Convergence of the Combination Technique for the Finite Element Solution of Poisson's Equation
342/15/93 A	Michael Luby, Wolfgang Ertel: Optimal Parallelization of Las Vegas Algorithms
342/16/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: Pointwise Convergence of the Combination Technique for Laplace's Equation
342/17/93 A	Georg Stellner, Matthias Schumann, Stefan Lamberts, Thomas Ludwig, Arndt Bode, Martin Kiehl und Rainer Mehlhorn: Developing Multicomputer Applications on Networks of Workstations Using NXLib
342/18/93 A	Max Fuchs, Ketil Stølen: Development of a Distributed Min/Max Component
342/19/93 A	Johann K. Obermaier: Recovery and Transaction Management in Write-optimized Database Systems
342/20/93 A	Sergej Gorlatch: Deriving Efficient Parallel Programs by Systemating Coarsing Specification Parallelism
342/01/94 A	Reiner Hüttl, Michael Schneider: Parallel Adaptive Numerical Simulation
342/02/94 A	Henning Spruth, Frank Johannes: Parallel Routing of VLSI Circuits Based on Net Independency
342/03/94 A	Henning Spruth, Frank Johannes, Kurt Antreich: PHIroute: A Parallel Hierarchical Sea-of-Gates Router
342/04/94 A	Martin Kiehl, Rainer Mehlhorn, Matthias Schumann: Parallel Multiple Shooting for Optimal Control Problems Under NX/2
342/05/94 A	Christian Suttner, Christoph Goller, Peter Krauss, Klaus-Jörn Lange, Ludwig Thomas, Thomas Schnekenburger: Heuristic Optimization of Parallel Computations
342/06/94 A	Andreas Listl: Using Subpages for Cache Coherency Control in Parallel Database Systems
342/07/94 A	Manfred Broy, Ketil Stølen: Specification and Refinement of Finite Dataflow Networks - a Relational Approach
342/08/94 A	Katharina Spies: Funktionale Spezifikation eines Kommunikationsprotokolls

Reihe A

342/09/94 A Peter A. Krauss: Applying a New Search Space Partitioning Method to Parallel Test Generation for Sequential Circuits
342/10/94 A Manfred Broy: A Functional Rephrasing of the Assumption/Commitment Specification Style
342/11/94 A Eckhardt Holz, Ketil Stølen: An Attempt to Embed a Restricted Version of SDL as a Target Language in Focus
342/12/94 A Christoph Pflaum: A Multi-Level-Algorithm for the Finite-Element-Solution of General Second Order Elliptic Differential Equations on Adaptive Sparse Grids
342/13/94 A Manfred Broy, Max Fuchs, Thomas F. Gritzner, Bernhard Schätz, Katharina Spies, Ketil Stølen: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/14/94 A Maximilian Fuchs: Technologieabhängigkeit von Spezifikationen digitaler Hardware
342/15/94 A M. Griebel, P. Oswald: Tensor Product Type Subspace Splittings And Multilevel Iterative Methods For Anisotropic Problems
342/16/94 A Gheorghe Ştefănescu: Algebra of Flownomials
342/17/94 A Ketil Stølen: A Refinement Relation Supporting the Transition from Unbounded to Bounded Communication Buffers
342/18/94 A Michael Griebel, Tilman Neuhoeffer: A Domain-Oriented Multilevel Algorithm-Implementation and Parallelization
342/19/94 A Michael Griebel, Walter Huber: Turbulence Simulation on Sparse Grids Using the Combination Method
342/20/94 A Johann Schumann: Using the Theorem Prover SETHEO for verifying the development of a Communication Protocol in FOCUS - A Case Study -
342/01/95 A Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law
342/03/95 A Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space
342/04/95 A Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes
342/05/95 A Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems
342/06/95 A Maximilian Fuchs: Formal Design of a Model-N Counter
342/07/95 A Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology
342/08/95 A Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies
342/09/95 A Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness

Reihe A

342/10/95 A	Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design
342/11/95 A	Thomas Schnekenburger: The ALDY Load Distribution System
342/12/95 A	Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm
342/13/95 A	Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming
342/14/95 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks
342/15/95 A	Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs
342/16/95 A	Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik
342/17/95 A	Georg Stellner: Using CoCheck on a Network of Workstations
342/18/95 A	Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications
342/19/95 A	Thomas Schnekenburger: Integration of Load Distribution into ParMod-C
342/20/95 A	Ketil Stølen: Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication
342/21/95 A	Andreas Listl, Giannis Bozas: Performance Gains Using Subpages for Cache Coherency Control
342/22/95 A	Volker Heun, Ernst W. Mayr: Embedding Graphs with Bounded Treewidth into Optimal Hypercubes
342/23/95 A	Petr Jančar, Javier Esparza: Deciding Finiteness of Petri Nets up to Bisimulation
342/24/95 A	M. Jung, U. Rüde: Implicit Extrapolation Methods for Variable Coefficient Problems
342/01/96 A	Michael Griebel, Tilman Neunhoeffer, Hans Regler: Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries
342/02/96 A	Thomas Grauschopf, Michael Griebel, Hans Regler: Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic-Multigrid Coarsening for Second Order Elliptic PDEs
342/03/96 A	Volker Heun, Ernst W. Mayr: Optimal Dynamic Edge-Disjoint Embeddings of Complete Binary Trees into Hypercubes
342/04/96 A	Thomas Huckle: Efficient Computation of Sparse Approximate Inverses
342/05/96 A	Thomas Ludwig, Roland Wismüller, Vaidy Sunderam, Arndt Bode: OMIS — On-line Monitoring Interface Specification
342/06/96 A	Ekkart Kindler: A Compositional Partial Order Semantics for Petri Net Components

Reihe A

342/07/96 A Richard Mayr: Some Results on Basic Parallel Processes
342/08/96 A Ralph Radermacher, Frank Weimer: INSEL Syntax-Bericht
342/09/96 A P.P. Spies, C. Eckert, M. Lange, D. Marek, R. Radermacher, F. Weimer, H.-M. Windisch: Sprachkonzepte zur Konstruktion verteilter Systeme
342/10/96 A Stefan Lamberts, Thomas Ludwig, Christian Röder, Arndt Bode: PFSLib – A File System for Parallel Programming Environments
342/11/96 A Manfred Broy, Gheorghe Ştefănescu: The Algebra of Stream Processing Functions
342/12/96 A Javier Esparza: Reachability in Live and Safe Free-Choice Petri Nets is NP-complete
342/13/96 A Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Many-to-Many Data-flow Networks
342/14/96 A Giannis Bozas, Michael Jaedicke, Andreas Listl, Bernhard Mitschang, Angelika Reiser, Stephan Zimmermann: On Transforming a Sequential SQL-DBMS into a Parallel One: First Results and Experiences of the MIDAS Project
342/15/96 A Richard Mayr: A Tableau System for Model Checking Petri Nets with a Fragment of the Linear Time μ -Calculus
342/16/96 A Ursula Hinkel, Katharina Spies: Anleitung zur Spezifikation von mobilen, dynamischen Focus-Netzen
342/17/96 A Richard Mayr: Model Checking PA-Processes
342/18/96 A Michaela Huhn: Put your Model Checker on Diet: Verification on Local States
342/01/97 A Tobias Müller, Stefan Lamberts, Ursula Maier, Georg Stellner: Evaluierung der Leistungsfähigkeit eines ATM-Netzes mit parallelen Programmierbibliotheken
342/02/97 A Hans-Joachim Bungartz, Thomas Dornseifer: Sparse Grids: Recent Developments For Elliptic Partial Differential Equations
342/03/97 A Bernhard Mitschang: Technologie für Parallele Datenbanken — Bericht zum Workshop
342/04/97 A Manfred Broy: Abstract Semantics of Synchronous Languages: The Example Esterel
342/05/97 A Hans-Joachim Bungartz, Ralf Ebner, Stefan Schulte: Hierarchische Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik
342/06/97 A Hans-Joachim Bungartz, Anton Frank, Florian Meier, Tilman Neunhoeffer, Stefan Schulte: Fluid Structure Interaction: 3 D Numerical Simulation and Visualization of a Micropump

SFB 342 : Methoden und Werkzeuge für die Nutz ung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B	Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B	Jörg Desel: On Abstraction of Nets
342/3/90 B	Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B	Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B	Barbara Paech1: Concurrency as a Modality
342/2/91 B	Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier- Toolbox -Anwenderbeschreibung
342/3/91 B	Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B	Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B	Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B	Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B	W. Reisig: Concurrent Temporal Logic
342/1/92 B	Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support
	Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B	Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B	Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung
342/2/93 B	Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B	Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS