ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2008, Vol. 37, No. 1

IMPLEMENTATION OF STRESS DEPENDENT BOUNDARY CONDITIONS IN FEM CODE FOR COUPLED PROBLEMS

Arnas Kačeniauskas

Laboratory of Parallel Computing, Vilnius Gediminas Technical University
Saulėtekio 11, Vilnius, LT-10223, Lithuania

Remigijus Kutas

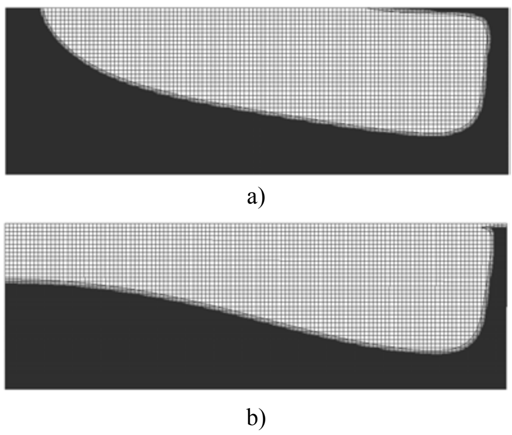
Computing Centre, Vilnius Gediminas Technical University
Saulėtekio 11, Vilnius, LT-10223, Lithuania

Abstract. The paper describes the implementation of stress dependent boundary conditions in the FEMTOOL code developed for numerical solution of coupled problems. Moving interface flows including breaking waves are considered. The performance of the numerical technique is validated by solving the dam break problem in the confined domain. The code development issues and the detailed investigation of parameters governing water separation from walls are presented.

1. Introduction

The progress in simulation of particular fields stimulated the development of numerical methods and computational technologies for coupled problems including multi-physical phenomena, complex flows and moving interfaces [5]. Various coupling mechanisms in a different context [6], such as flow-structure interaction [2], interfaces separating different fluids or gases [9], magneto-thermo-mechanical analysis [12], electromagnetic metal forming [17] are meant by the term "coupled analysis". Available numerical analyses are strongly dependent on the application.

Moving interface flows include a strong coupling between the interface propagation and dynamics of the continuum. Numerical methods advocated for solving moving interface problems might be classified into two categories: interface tracking techniques (ITT) and interface capturing techniques (ICT). In the first category of interface simulating methods, a moving interface is represented and tracked explicitly either by making it with special marker points, or by attaching it to a mesh surface. Various ITT [2, 15] for attaching the interface to a mesh surface were developed during the past decades using the finite element method (FEM). In the second category of interface simulating methods, either massless particles or an indicator function marks gas or fluid on either side of the interface. The marker-and-cell method [7], the volume of fluid method [8] and the level set method [16] are well known methods using the ICT idea and the Eulerian approach. The ICT require no geometry manipulations after the mesh is generated and can be applied to interfaces of a complex topology. The FEM is becoming increasingly popular in many fields of engineering, therefore, the demand for further investigation of the ICT and implementation in FEM codes is rapidly growing [9, 18].



a)

b)

Figure 1. Non-physical behaviour of the flow near walls: (a) inappropriate handling of the flow separating from the upper wall; (b) incorrect flow behaviour in the corner

Complex 3D flows often include waves hitting a wall or jets entering geometrically complex cavities

[4]. The developed software should accurately capture behaviour of the flow near walls or handle air bubbles near the corners. The application of the standard slip boundary conditions in conjunction with conventional numerical schemes sometimes can produce non-physical behaviour of the flow. Figure 1a illustrates inappropriate handling of the flow separating from the wall. Sometimes flow easily moves in the tangential direction of the wall, but the motion in the normal direction is heavily induced. The incorrect flow behaviour in the corner is shown in Figure 1b. Such problems can be resolved applying the stress dependent boundary conditions [3]. The above idea leads to a change in the type of boundary conditions for the velocity on the certain boundaries. Dirichlet-type boundary conditions are replaced by Neumann one, producing differences in the global finite element matrix [1]. The FEM code development and implementation issues are presented in this paper.

2. Governing equations

The laminar and Newtonian flow of viscous and incompressible fluids is described by the Navier-Stokes equations:

$$\rho\left(\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}\right)=\rho F_i+\frac{\partial \sigma_{ij}}{\partial x_j}, \quad (1)$$

$$\frac{\partial u_i}{\partial x_i}=0, \quad (2)$$

where u_i are the velocity components; ρ is the density; F_i are the gravity force components and σ_{ij} is stress tensor

$$\sigma_{ij}=-p\delta_{ij}+\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right), \quad (3)$$

where μ is dynamic viscosity coefficient; p is pressure and δ_{ij} is Kronecker delta.

The pseudo-concentration method [11, 19] is developed for moving interface flows using the Eulerian approach and the interface capturing idea. The pseudo-concentration function φ serves as a marker identifying fluids A and B. φ=1 for fluid A and φ=0 for fluid B. The evolution of the interface is governed by a time dependent convection equation:

$$\frac{\partial \varphi}{\partial t}+u_j\frac{\partial \varphi}{\partial x_j}=0. \quad (4)$$

The velocity u_j is obtained from the solution of the Navier-Stokes equations (1-3). The initial conditions defined on the entire solution domain should be prescribed for the equation (4). While the interface propagates at a correct velocity, the pseudo-concentration function becomes irregular after some period of time. An interface sharpening procedure [13] together with a simpler limiter is applied in order to preserve sharpness of the interface and satisfactory mass conservation.

The space-time Galerkin least squares finite element method [15] is applied as a general-purpose computational approach to solve the partial differential equations (1–4). Equal order bilinear shape functions are used for both the pressure and velocity components as well as for the pseudo-concentration function. The detailed description of variational formulation and stabilisation parameters can be found in the work [10].

3. Stress dependent boundary conditions

In the most cases the slip boundary conditions for velocity are prescribed on impermeable rigid walls:

$$u_i n_i = 0, \quad (5)$$

where n_i are components of a unit normal vector. The boundary conditions (5) are a usual choice of boundary conditions on rigid walls used for solving moving interface problems.

The discussed boundary conditions can not resolve problems of incompressible flows related with air bubbles or flow suddenly separating from the wall. The stress dependent boundary conditions [3] have been devised by researchers working on metal casting or mould filling problems. In practice, moulds are made of porous materials. Therefore, permeable walls are assumed and air can leave the mould without resistance. If the fluid does not push the wall, the open boundary conditions are prescribed:

$$\text{if } n_i\sigma_{ij}n_j > 0 \text{ then } \sigma_{ij}n_j = 0. \quad (6)$$

In case of dominant inertial forces and high Reynolds numbers, the viscous terms in boundary conditions (6) are negligible. This reduces (6) to a zero Dirichlet boundary conditions for the pressure and standard Newmann boundary conditions for velocities:

$$p=0, \quad (7)$$

$$n_j\frac{\partial u_i}{\partial x_j}=0. \quad (8)$$

On the other side, if fluid pushes against the wall then the slip boundary conditions are prescribed, that is zero normal velocities and zero tangent stress:

$$\text{if } n_i\sigma_{ij}n_j < 0 \text{ then } u_i n_i = 0 \text{ and } n_i\sigma_{ij}t_j = 0, \quad (9)$$

where t_j are components of a unit tangent vector. The condition of zero tangent stress can be replaced by one of the Newmann boundary conditions (8) in case of straight boundaries or high Reynolds numbers. The Newmann boundary conditions (8) are automatically satisfied in standard finite element formulations. Anyway, resulting set of boundary conditions (6, 9) or (5, 7-8) are more complex than standard Newmann or Dirichlet boundary conditions. Stress values should be accurately computed on the walls and the type of boundary conditions should be dynamically changed during simulation.

4. Implementation in FEM code

The stress dependent boundary conditions have been implemented in the FEMTOOL code [10] created in Swiss Federal Institute of Technology Zurich and developed in Laboratory of Parallel Computing of Vilnius Gediminas Technical University. FEMTOOL is a Finite Element Method Toolbox developed by using FORTRAN and C programming languages, which allows implementation of any partial differential equation (PDE) with minor expenses. Time dependent problems are solved using space-time finite elements. The order of shape functions is determined by input and is limited neither in space nor in time. The flexible and universal structure of the code makes FEMTOOL to be applicable to various PDEs of interest such as Poisson's equations, convection-diffusion equations, shallow water equations and Navier-Stokes equations.

The flow chart of the DOJOB subroutine, governing main computations, is shown in Figure 2. The conventional FEM routines present in the initial FEMTOOL code are plotted by the solid lines. The dashed lines show new blocs of routines and supplementary conditions added to the structure of the code in order to implement the stress dependent boundary conditions (6, 9). The TIME LOOP serves as the main loop for the time-dependent problems. The NON-LINEAR LOOP is used for iterative solution of non-linear PDEs. Subroutine CONVERGENCE checks the convergence of non-linear loop and sets some governing parameters for the time loop. The OLD TIME LOOP, which has been removed from the code, is plotted by the point lines.

The active approach enables the user to implement his routines into existing software without deep knowledge of the internal structure of the FEMTOOL library provided by developers. The user routines are implemented as include files *.inc within the framework of a specific toolbox, therefore, new finite elements for a coupled PDEs can be implemented with just a few lines of code. Routine GLOBALBUILD, filling the global FE matrix and the global right hand side vector, includes three main subroutines that user can modify implementing his PDEs and finite elements. In INITFKT user fills initial guess of the non-linear solution. TIMEFKT is devoted for the time-dependent boundary conditions. User subroutine LFE MATRIX fills the local FE matrix and the local right hand side vector. These three subroutines provide for users sufficient possibilities to implement any standard time-dependent PDE including non-linear terms and any user finite element. Other user subroutines USERFKT can be employed for the implementation of the advanced features such as the coupled problems or very specific data processing. Subroutine USERFKT1 is usually placed before the time loop in order to provide for users ability to perform some preparatory computations. USERFKT2 provides possibility to process results before the output at the end of the nonlinear iteration, while USERFKT3 performs some user specific computations at the end of a time step. Subroutine USERFKT4 is devoted for any specific post-processing at the end of computations.

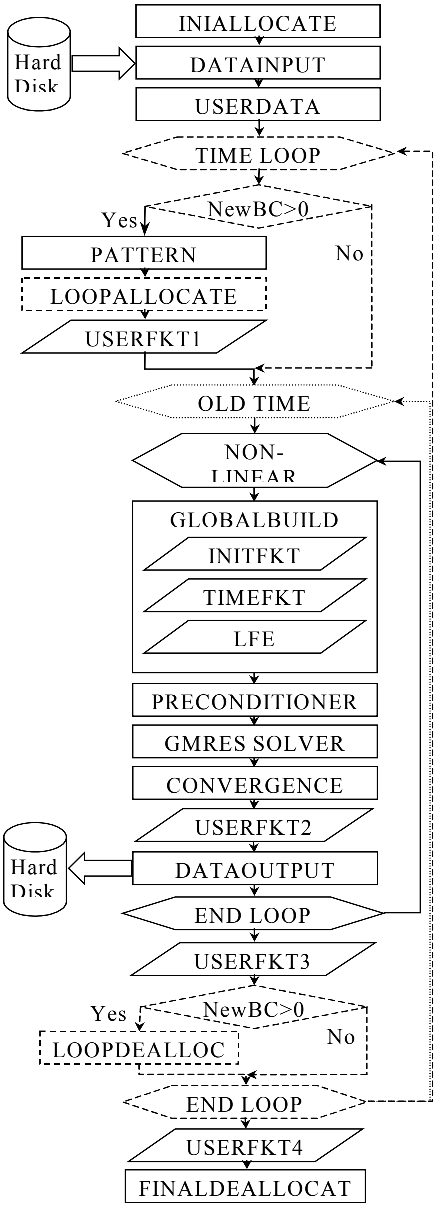


Figure 2. Flowchart of DOJOB routine

Remaining routines of the code is grouped around the time loop. Subroutine DATAINPUT reads the main data arrays from the file. Subroutine USERDATA is devoted for special user arrays that can be used for transferring the data between coupled problems solved iteratively. Subroutine DATAOUTPUT writes the results of the current time step or non-linear iteration to the result file. The block of routines INIALLOCATE reads governing parameters and dynamically allocates a memory for main data arrays. Subroutine PATTERN prepares the pattern arrays describing the structure of the global FE matrix. The memory for the global FE matrix, the right hand side vector and other arrays used by the GMRES SOLVER and PRECONDITIONER can be allocated by LOOPALLOCATE, when subroutine PATTERN have performed its work. Subroutines LOOPDEALLOCATE and FINALDEALLOCATE de-allocate the memory.

In general, the conventional FEM codes do not include the stress dependent boundary conditions. The main difficulty is related with dynamic change of the structure of the global finite element matrix in the time loop. The application of the boundary conditions (7-8) yield two degrees of freedom for velocity components at a boundary node. The boundary condition (5) yields one degree of freedom for pressure and one for the tangent velocity component. Moreover, the computation of stresses includes accurate handling of normal and tangent vectors on boundaries forming corners and other complicated shapes.

The initial structure of FEMTOOL does not allow the direct implementation of the considered boundary conditions by users without changing the main code provided as "the black box". The user subroutine TIMEFKT is devoted for computation of time dependent values of boundary conditions, but it cannot change the global FE matrix structure. The stress computations STRESS and the update of boundary conditions BCUPDATE can be naturally implemented in the user subroutine USERFKT3 without changing the whole library. However, the memory for the global matrix and the solver should be allocated and de-allocated in the time loop (LOOPALLOCATE and LOOPDEALLOCATE) in order to consider these changes. Subroutine PATTERN should also be called after every change of the boundary condition type. The indicator NewBC set up in USERFKT3 governs the necessary calls of appropriate routines. The preparation of information for stress computations should be performed after the call to PATTERN, therefore, it is done in the user routine USERFKT1. The initial solution can also be updated in the user routine INITFKT, but some governing parameters are not conveniently accessed in GLOBALBUILD, inside the non-linear loop. Therefore, the update of the initial solution considering the global changes are also performed in the user routine USERFKT1, in the time loop. The whole USERFKT1 can be switched off if the user does not need the stress boundary conditions. The extension of the time loop and additional conditions governed by indicator NewBC can be considered as the main changes of the basic code structure.

The solution of the coupled problems, including moving interfaces, is illustrated in Figure 3. The parameters of the main problem, governing by the Navier-Stokes equations, are set up at the beginning of computations. Considering specified parameters DOJOB calls the appropriate user routines. The parameters of the second problem governed by the convection equation are set up in the user routine USERFKT3A at the end of each time step. The subroutine DOJOB governed by these parameters is called recursively. The names of user subroutines like USERFKT3$(prob2) includes the name of the second problem, therefore, the conflict does not happen. The interface sharpening is performed in the subroutine REINIT called from the USERFKT3conv. All computations of the stress boundary conditions are performed in user routines USERFKT1 and USERFKT3B. The stress computations are separated from the solution of the second problem in order to have more structured code. Therefore, two different user routines USERFKT3A and USERFKT3B appear instead of the former USERFKT3. The user routines USERFKT2 and USERFKT4 are not employed. The modified structure of the code shows, that development of the software modelling breaking waves presents challenges to all scientists and software developers.

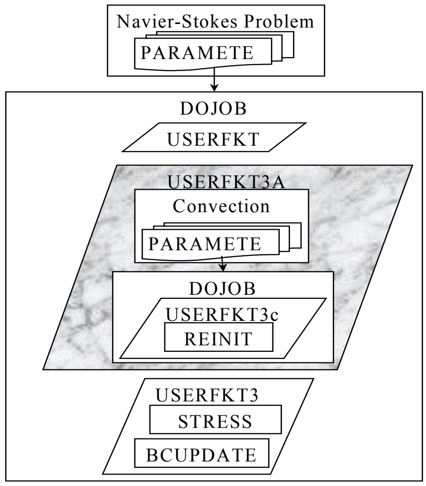


Figure 3. Flow chart of the FEMTOOL solving the coupled problem with stress dependent boundary conditions

5. Numerical results and discussions

The dam break flow in a confined domain [3] has been simulated in order to validate the developed code and the implementation of the stress dependent boundary conditions. The geometry of the solution domain is shown in Figure 4. A rectangular cavity with dimensions 0.09m×0.03m is considered (a=0.015m). At initial time t=0.0s, water is confined in the left half of the cavity. Later it is subject of vertical gravity (g=9.81m/s^2) and free to move. The density of water is ρ_A=1000kg/m^3, the dynamic viscosity coefficient is μ_A=0.01kg/(m·s). The density of air is taken to be ρ_B=1kg/m^3 and the dynamic viscosity coefficient is μ_B=0.0001kg/(m·s). The slip boundary conditions (5) have been applied to the bottom and sides of the reservoir. The stress dependent boundary conditions (6, 9) have been prescribed on the upper wall. The 120×40 and 240×80 finite element meshes are employed for computations. The investigated time interval t=[0.0; 0.3]s is divided to 300 time steps. The size of the time step Δt=0.001s for the 120×40 finite element mesh. In case of 240×180 finite element mesh, the number of time steps is equal to 600. Respectively, the size of time step is Δt=0.0005s.

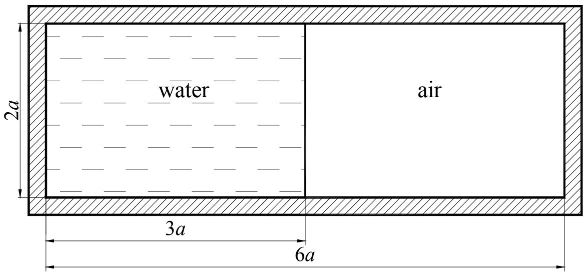


Figure 4. Geometry of dam break flow in a confined domain

Figure 5 illustrates time evolution of the moving interface. The gravity causes the water column on the left of the reservoir to seek the lowest possible level of potential energy. Thus, the column will collapse covering the bottom of the reservoir. The initial stages of the flow are dominated by inertia forces. Therefore, on such a large scale the effect of surface tension forces is unimportant. Figure 5b shows the successful separation of the water from the upper wall. The slip boundary conditions (5) has been automatically changed to the open boundary conditions (7-8), evaluating positive stress values on the wall. Figures 5c, 5d, 5e illustrate very complex behaviour of the flow in the right corner at the top wall. The complexity of the interface shape occurring in the different stages of breaking wave phenomena can be easily captured using the applied numerical technique. When t=0.3s, the backward moving wave has folded over and a small amount of air is trapped (Figure 5f). In experiments [14], this air is present in the form of small bubbles. The current methodology has been derived for sharp interfaces, therefore, the mesh needs significant refinement to a resolution smaller than the bubble size. The validation of the obtained numerical results by the quantitative comparison with experimental measurements can be found in the works [11, 13].

The most complex behaviour of the flow is observed in the right corner at the top wall. The slip boundary conditions (5) should be immediately prescribed when water reaches the top wall in order to prevent the mass loss. The prescribed boundary conditions remain until the water fills the corner and starts propagate in the tangent direction of the top wall. When t=0.125s, the stress values becomes positive in several nodes of the upper boundary. Thus, the slip boundary conditions (5) should be changed to (7-8) on the part of the upper boundary. The dynamic change of boundary conditions produces numerical oscillations that can significantly influence the stress distribution on the boundary. Figure 6 shows the developed algorithm. Usually *push* values lower than zero are considered in order to avoid frequent change of boundary conditions producing oscillations. Two different conditions are employed in the algorithm in order to prescribe different values for *phi1* and *phi2*. Numerical experiments have showed that *phi2*-*phi1*=0.1 helps to postpone early water separation from the wall and to prevent undesirable oscillations. The best performance has been achieved by using two sets of numerical parameters: *push*=–5.0, *phi1*=0.7, *phi2*=0.8 and *push*=–5.0, *phi1*=0.6, *phi2*=0.7. Larger *phi* values can cause non-physical slow down of the separation process. Figure 7 illustrates the undesirable numerical oscillations and the proper handling of water separation from the upper wall.

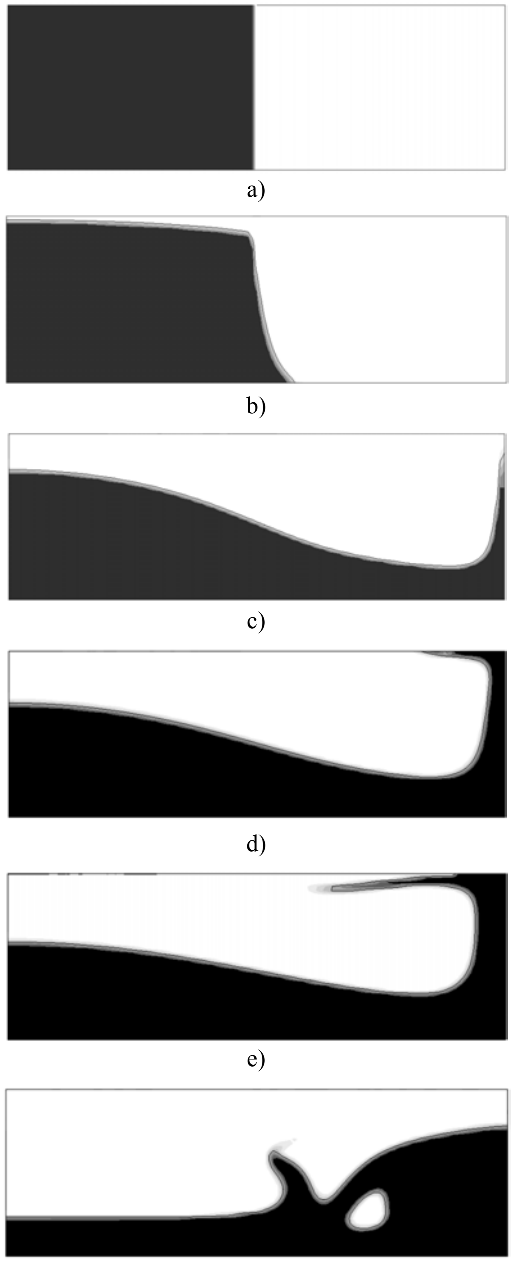
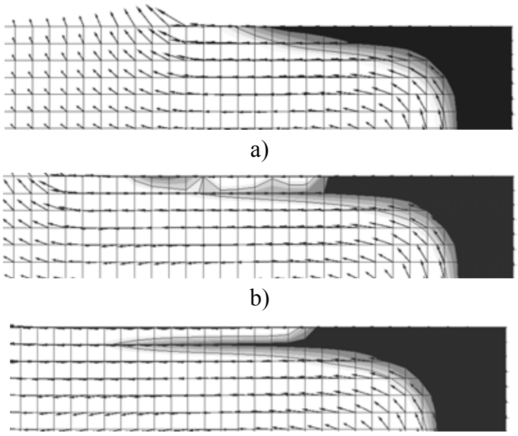


Figure 5. Time evolution of the dam break flow, the pseudo-concentration function: (a) t=0.000s; (b) t=0.025s; (c) t=0.100s; (d) t=0.125s; (e) t=0.150s; (f) t=0.300s

if (u_2>0 and φ>*phi1*) then
    set up (5)
elseif ($n_i \sigma_{ij} n_j$ <*push* and φ>*phi2*) then
    set up (5)
else
    set up (7-8)
endif

Figure 6. Algorithm for the stress dependent boundary conditions



a)

b)

c)

Figure 7. Water separation from the upper wall:
(a) water propagation in the tangent direction of the wall;
(b) undesirable numerical oscillations, *phi1*=*phi2*=0.6;
(c) proper handling of separation, *phi1*=0.7, *phi2*=0.8

6. Conclusions

In this paper, the development of FEM code for solution of coupled problems, including moving interfaces and breaking waves, has been described. The implementation issues of stress dependent boundary conditions have been covered. The implementation of stress dependent boundary conditions requires significant modification of the conventional routines of FEM codes, because of dynamic changes in the structure of the global finite element matrix. The appropriate handling of numerical parameters helps to avoid the oscillations caused by the dynamic change of the type of boundary conditions. The presented numerical solution of the dam break problem proves that the developed code is capable of simulating fluid separation from walls, air bubbles and permeable boundaries.

Acknowledgement

The work is partly supported by the Lithuanian State Science and Studies Foundation in the frame of the project GridTechno (the contract number B-07043). The computational resources provided by the LitGRID are also gratefully acknowledged.

References

[1] **R. Codina, O. Soto.** Finite Element Simulation of the Filling of Thin Moulds. *International Journal Numerical Methods in Engineering*, *Vol.*46(9), 1999, 1559-1573.

[2] **F. Del Pin, S. Idelsohn, E. Oñate, R. Aubry.** The ALE/Lagrangian Particle Finite Element Method: a New Approach to Computation of Free-surface Flows and Fluid–Object Interactions. *Computers & Fluids*, 2007, *Vol.*36(1), 27–38.

[3] **R. Dieterlen, V. Maronnier, M. Picasso, J. Rappaz.** Numerical Simulation of Free Surface Flows. *Computational Mechanics. New Trends and Applications, Eds. S. Idelson, E. Onate, E. Dvorkin, Barcelona*, 1998, 13.

[4] **D.M. Gao.** A Three-Dimensional Hybrid Finite Element-Volume Tracking Model for Mould Filling in Processes. *International Journal for Numerical Methods in Fluids*, 1999, *Vol.*29(7), 877-895.

[5] **K.K. Gupta, J.L. Meek.** Finite Element Multidisciplinary Analysis. *AIAA, New York*, 2000.

[6] **K. Hameyer, J. Driesen, H. De Gersem, R. Belmans.** The Classification of Coupled Field Problems, *IEEE Transactions on Magnetics*, 1999, *Vol.*35(3), 1618-1621.

[7] **F.H. Harlow, J.E. Welch.** Numerical Calculation of Time - Dependent Viscous Incompressible Flow. *Journal of Physics of Fluids*, 1965, *Vol.*8(12), 2182–2189.

[8] **C.W. Hirt, B.D. Nichols.** Volume Of Fluid (VOF) Method for the Dynamics of Free Boundaries. *Journal of Computational Physics*, 1981, *Vol.*39(1), 201–225.

[9] **A. Kačeniauskas.** Two-phase Flow Modelling by the Level Set Method and Finite Elements. *Power Engineering*, 2000, *No.*4, 22-31.

[10] **A. Kačeniauskas, P. Rutschmann.** Parallel FEM Software for CFD Problems. *Informatica*, 2004, *Vol.* 3(15), 363-378.

[11] **A. Kačeniauskas.** Dam Break Flow Simulation by the Pseudo-Concentration Method. *Mechanika*, 2005, *Vol.* 51(1), 39-43.

[12] **A. Kačeniauskas, R. Kačianauskas, E. Stupak, V. Šnirpūnas, S. Balevičius, J. Novickij, N. Žurauskienė.** Coupled magneto-thermo-mechanical problems investigating high magnetic fields. *Computational Methods for Coupled Problems in Science and Engineering* (Eds. *M. Papadrakakis, E. Oñate and B. Schrefler), CIMNE, Barcelona*, 2005, 1-12.

[13] **A. Kačeniauskas.** Development of Efficient Interface Sharpening Procedure for Viscous Incompressible Flows. *Submitted to Informatica*, 2008.

[14] **S. Koshizuka, H. Tamako, Y. Oka.** Aparticle Method for Incompressible Viscous Flow with Fluid Fragmentation. *Journal of Computational Fluid Dynamics*, 1995, *Vol.*4(1), 29-46.

[15] **A. Masud., T.J.R. Hughes.** A Space - Time Galerkin/Least - Squares Finite Element Formulation of The Navier-Stokes Equations for Moving Domain Problems. *Computational Methods in Applied Mechanics and Engineering*, 1997, *Vol.*146(1-2), 91-126.

[16] **S. Osher, J.A. Sethian.** Fronts Propagating with Curvature - Dependent Speed: Algorithms Based On Hamilton-Jacobi Formulation. *Journal of Computational Physics*, 1988, *Vol.*79(1), 12-49.

[17] **S. Reese, B. Svendsen, M. Stiemer, J. Unger, M. Schwarze, H. Blum.** On a New Finite Element Technology for Electromagnetic Metal Forming Processes. *Archive of Applied Mechanics*, 2005, *Vol.* 74(11-12), 834-845.

[18] **T.E. Tezduyar.** Finite Elements in Fluids: Special Methods and Enhanced Solution Techniques. *Computers & Fluids*, 2007, *Vol.*36(2), 207–223.

[19] **E. Thompson.** Use of Pseudo-Concentration to Follow Creeping Viscous Flow During Transient Analysis. *International Journal for Numerical Methods in Fluids*, 1986, *Vol.*7(2), 749-761.

Received November 2007.

DOI: 10.5755/j01.itc.37.1.11922