European Congress on Computational Methods
in Applied Sciences and Engineering (ECCOMAS 2012)
J. Eberhardsteiner et.al. (eds.)
Vienna, Austria, September 10-14, 2012

HIGHER ORDER IMPLICIT TIME INTEGRATION SCHEMES TO SOLVE INCOMPRESSIBLE NAVIER-STOKES ON CO-LOCATED GRIDS USING CONSISTENT UNSTEADY RHIE-CHOW

V. Kazemi-Kamyab *, A.H. van Zuijlen, and H. Bijl

Aerodynamics Section, Faculty of Aerospace Engineering, Delft University of Technology
P.O. Box 5058, 2600 GB Delft, The Netherlands.
* e-mail: v.kazemikamyab@tudelft.nl

Keywords: incompressible Navier-Stokes, co-located, Rhie-Chow interpolation, implicit time integration, ESDIRK, OpenFoam

Abstract. *Noting that time-accurate computations of the unsteady incompressible Navier-Stokes (INS) equations can be computationally expensive, a family of higher order implicit multi-stage time integration schemes (namely ESDIRK) is used for advancing the solution to the unsteady INS in time. The higher order time integration schemes have the potential to decrease the computational cost of obtaining engineering levels of accuracy relative to the traditionally used 2^{nd} order implicit schemes. The finite volume method is used for spatial discretization, and co-located arrangement of the primitive variables is considered. Furthermore, an iterated PISO algorithm is used to solve the incompressible Navier-Stokes equations. By using a temporally consistent Rhie-Chow interpolation, higher order temporal accuracy in solving the INS equations on co-located grids is achieved. For a two-dimensional lid driven cavity test case, the temporal convergence of the solution is investigated, with the third and fourth order ESDIRK schemes for time integration. The results demonstrate the temporal consistency and temporal order preservation of the algorithm.*

1 INTRODUCTION

The unsteady incompressible Navier-Stokes equations in primitive variables are given by:

$$\begin{aligned}\frac{\partial \boldsymbol{u}}{\partial t} &= -\nabla.(\boldsymbol{u}\boldsymbol{u}) + \nu\nabla^2\boldsymbol{u} - \nabla p = R(\boldsymbol{u}) - \nabla p = \mathcal{F}(\boldsymbol{u}, p, t), \\ \nabla.\boldsymbol{u} &= 0.\end{aligned} \tag{1}$$

where $\boldsymbol{u}$ is the velocity vector, p is the kinematic pressure and ν the kinematic viscosity. The stability and accuracy of a numerical approach to solve Eqn.(1) in primitive variables, are influenced by the method with which the velocity-pressure coupling is resolved; both in the solution algorithm and in the arrangement of the variable in discretization of Eqn.(1) [1].

In solving the steady form of Eqn.(1) on a co-located grid using the Finite Volume Method (FVM), straightforward discretization of the equations can result in spurious oscillations in the pressure field. One of the widely used methods in the literature and large scale CFD packages to tackle this issue, is the Rhie-Chow interpolation [2]. This interpolation technique relies on the determination of an expression for the cell face velocity (which appears in the continuity equation) in terms of the pressure gradient across the cell face, thus restoring the lost linkage between the velocity and pressure fields [1, 3].

In solving the unsteady INS equations (in which implicit time integration is used for advancing the solution in time), a class of segregated solution algorithms includes SIMPLE-like and PISO-like approaches. Extensive research has been conducted to extend the original Rhie-Chow interpolation (developed for steady-state computations) to unsteady flows. In Pascau [4], a survey of the literature on this topic is presented and it has been pointed out that by performing the interpolation incorrectly, the numerical scheme is temporally inconsistent.

In engineering applications, to advance the solution to the unsteady INS (Eqn.(1)) in time, typically an implicit time integration is preferred over an explicit one in order to circumvent time step restrictions due to probable stiffness in the problem. Stiffness in a system can, for example, arise due to the nature of the governing equations or due to the generated grid (such as clustering of nodes near the interface or walls of the domains [5]). In the literature, for the mentioned class of solution algorithms to solve the unsteady INS, mainly the first order Backward Euler (BDF1) and the second order BDF2 and Crank-Nicolson schemes have been considered.

In this paper, a family of higher order implicit multi-stage time integration schemes is used for advancing the solution to the unsteady incompressible Navier-Stokes (INS) in time. In particular the Explicit first stage, Singly Diagonally Implicit Runge-Kutta (ESDIRK) schemes are considered. The eventual goal of using the higher order time schemes is to decrease the computational cost to obtain a certain accuracy level (in particular engineering levels of accuracy) relative to traditionally used implicit first and second order time integration schemes.

Obtaining higher order temporal accuracy in solving the unsteady INS on co-located grids is not straight forward. The obtained solution may fail to preserve the design order of the applied higher order time integration scheme (and in particular the ESDIRK), mainly due to use of (1) a temporally inconsistent Rhie-Chow interpolation, and (2) a segregated solution algorithm (e.g.

PISO). In this paper, the first issue is tackled and the second issue is left to future work. In order to neglect the influence of the second issue on the temporal order preservation, an iterated PISO algorithm is considered. This class of higher order implicit time integration schemes has also been considered in [7] to solve the incompressible Navier-Stokes equations, however, the solution algorithm made use of the temporally inconsistent Rhie-chow interpolation of Choi [6, 4].

In what follow, first the semi-discrete form of Eqn.(1) is obtained by discretizing the spatial operators. After a small discussion on the higher order implicit ESDIRK schemes, the solution algorithm to advance the solution in time using the ESDIRK schemes is presented. The lid driven cavity is used to validate the temporal convergence of the schemes.

2 SPATIAL DISCRETIZATION AND SEMI-DISCRETE FORM

The method of lines is used to discretize Eqn.(1), with spatial operators being discretized first to obtain the semi-discrete form of the equations. The computational domain is subdivided into N finite volumes (or cells) where each cell is bounded by arbitrary number of cell faces N_f. Through out this paper, it is assumed that the grid does not vary with time. The volume integral form of Eqn.(1) is applied to each finite volume, and the Gauss divergence theorem is used to convert the volume integrals to surface integration over the closed boundary of the cell (see [8, 9, 10] for more details). Noting that each cell is bounded by discrete number of faces, the resultant discrete form of Eqn.(1) is give by:

$$V_P \frac{d\boldsymbol{U}_P}{dt} = -\sum_f^{N_f} \phi_f \boldsymbol{U}_f + \nu \sum_f^{N_f} (\nabla \boldsymbol{U})_f . \boldsymbol{n}_f S_f - \sum_f^{N_f} p_f \boldsymbol{n}_f S_f = \mathcal{F}_P, \tag{2}$$

$$\sum_f^{N_f} \phi_f = 0. \tag{3}$$

where V_P is the volume of the cell and $V_P(\nabla p)_P$ is approximated by $\sum_f^{N_f} p_f \boldsymbol{n}_f S_f$. For a given cell face f, $\boldsymbol{n}_f$ is the face normal vector, S_f is its surface area, and $\phi_f = \boldsymbol{U}_f . \boldsymbol{n}_f S_f$ is the flux through the face.

After using appropriate schemes to discretize the convective and diffusive fluxes, the following semi-discretized form of Eqn.(1) for each cell P surrounded by N_{nb} neighboring cells is obtained (where Eqn.(2) is divided by the cell volume):

$$\begin{gathered} \frac{d\boldsymbol{U}_P}{dt} + a_P \boldsymbol{U}_P - \sum_{nb} a_{nb} \boldsymbol{U}_{nb} = -(\nabla p)_P + \boldsymbol{r}_P, \\ \sum_f^{N_f} \phi_f = 0 \end{gathered} \tag{4}$$

where a_P and a_{nb} are the diagonal and off-diagonal coefficients of the (spatial) discretization matrix. $\boldsymbol{r}_P$ represents the explicit convection and diffusion terms as well as boundary contributions and $(\nabla p)_P = \frac{1}{V_P}\sum_f^{N_f} p_f \boldsymbol{n}_f S_f$. Using the following definition:

$$H_P(\boldsymbol{U}_P) = \sum_{nb} a_{nb}\boldsymbol{U}_{nb} + \boldsymbol{r}_P, \tag{5}$$

we can rewrite the first equation of Eqn.(4) as:

$$\frac{d\boldsymbol{U}_P}{dt} + a_P\boldsymbol{U}_P = H_P - (\nabla p)_P. \tag{6}$$

or in short,

$$\frac{d\boldsymbol{U}_P}{dt} = \mathcal{F}_P(\boldsymbol{U}_P, p, t). \tag{7}$$

The resultant system of equations, can be written by:

$$\frac{d\boldsymbol{U}}{dt} = \mathcal{F}(\boldsymbol{U}, p, t) = R\boldsymbol{U} - Gp = -C\boldsymbol{U} + B\boldsymbol{U} - Gp + \boldsymbol{r} \tag{8}$$

where R is sum of the convective C and diffusive B matrices and G the gradient operator.

3 TIME INTEGRATION

In this paper a family of multi-stage implicit Runge-Kutta schemes (IRK), namely the Explicit first stage, Singly Diagonally Implicit Runge-Kutta (ESDIRK) schemes, is considered for time integration which can be made of arbitrary higher order while retaining the L-stability property. While BDF1 and BDF2 are L-stable, the Crank-Nicolson scheme is not. It is possible to construct higher order BDF methods, but they are only $L(\alpha)$ stable and their stability region rapidly drops as the order of the scheme is increased [5]. For a coupled ODE system of the form $\frac{d\boldsymbol{U}}{dt} = \boldsymbol{\mathcal{F}}(\boldsymbol{U}, t)$, the solution at each stage of an ESDIRK scheme can be written as:

$$\boldsymbol{U}^{(k)} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{k} a_{ki}^I \boldsymbol{\mathcal{F}}(t^n + c_i\Delta t, \boldsymbol{U}^{(i)}) = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{k} a_{ki}^I \boldsymbol{\mathcal{F}}^{(i)}, \tag{9}$$

where a_{ki}^I are the coefficients of the corresponding stage and $c_i = \sum_j a_{ij}^I$ is the location (quadrature node) of the stage solution at $t^{(i)} = t^n + c_i\Delta t$. High order solution at the next time level can be achieved by the weighted sum of the stage residuals such that the lower order errors cancel out:

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{\mathcal{F}}^{(i)}, \tag{10}$$

where b_i are the weight factors with $\sum_i b_i = 1$, and s is the number of stages. In this paper, the stiffly accurate ESDIRK schemes presented in [11] are considered where $a_{si}^I = b_i$ and thus the solution of the last stage is equal to the solution of the time-level, $\boldsymbol{U}^{n+1} = \boldsymbol{U}^s$. The coefficients and weights are usually arranged in Butcher tableau (see Table.(1)). For the ESDIRK schemes, as the name implies, the diagonal coefficients are equal ($a_{kk} = \gamma$). Furthermore, it is possible to incorporate the ESDIRK schemes into solvers which already include Backward Euler, since from an implementation view point, the solution at each stage of the ESDIRK scheme resembles the BDF1 scheme with a source term.

Table 1: A four stage ESDIRKscheme.

$$\begin{array}{c|cccc} c_1 & 0 & 0 & 0 & 0 \\ c_2 & a^I_{21} & a^I_{22} & 0 & 0 \\ c_3 & a^I_{31} & a^I_{32} & a^I_{33} & 0 \\ c_4 & a^I_{41} & a^I_{42} & a^I_{43} & a^I_{44} \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array}$$

4 SOLUTION ALGORITHM

In this section the ESDIRK schemes introduced in the previous section are applied to the semi-discrete form of the problem given by Eqn.(8) to advance the solution of the INS in time. In order to obtain the solution field at each stage, an iterated PISO algorithm is employed (no under-relaxation is considered in this paper). Furthermore, due to the use of a co-located grid, an appropriate Rhie-Chow interpolation is used to avoid a check-board pressure field. For an implicit stage of the ESDIRK schemes, at each iteration j the solution algorithm begins by solving an intermediate velocity field (predictor step) given by:

$$\tilde{a}_P \boldsymbol{U}^*_P - \sum_N a_N \boldsymbol{U}^*_N = \frac{1}{\Delta t a_{k,k}} \boldsymbol{U}^n_P - (\nabla p^{j-1})_P + \boldsymbol{r}_P + \frac{1}{a_{k,k}} \sum_{i=1}^{k-1} \mathcal{F}^{(i)}_P, \tag{11}$$

where,

$$\tilde{a}_P = \frac{1}{\Delta t a_{k,k}} + a_P. \tag{12}$$

Using the earlier definition of H_P (Eqn.(5)), the above equation can be rewritten as:

$$\boldsymbol{U}^*_P = \frac{1}{\Delta t a_{k,k}} \frac{\boldsymbol{U}^n_P}{\tilde{a}_P} + \frac{H_P(\boldsymbol{U}^*)}{\tilde{a}_P} - \frac{(\nabla p^{j-1})_P}{\tilde{a}_P} + \frac{1}{a_{k,k}} \frac{\sum_{i=1}^{k-1} \mathcal{F}^{(i)}_P}{\tilde{a}_P}. \tag{13}$$

It is noted that the *current available values* of the solution fields are used in evaluating the pressure gradient, the linearized convective flux ϕ_f (picard iterations are used for linearization of the nonlinear convection term), and $\boldsymbol{r}_P$ (if it contains explicit diffusion and convection terms). The evaluation of $\mathcal{F}_P$ is discussed in section 4.1. The obtained velocity field is not divergence free since the pressure field was treated explicitly. To satisfy the incompressibly constrain, the intermediate velocity field needs to be corrected by an amount which is obtained by solving for the pressure field. The expression for $\boldsymbol{U}^j_P$ (denoted as the divergence free velocity field) with p^j as its corresponding pressure field is given by:

$$\boldsymbol{U}^j_P = \frac{1}{\Delta t a_{k,k}} \frac{\boldsymbol{U}^n_P}{\tilde{a}_P} + \frac{H_P(\boldsymbol{U}^*)}{\tilde{a}_P} - \frac{(\nabla p^{j})_P}{\tilde{a}_P} + \frac{1}{a_{k,k}} \frac{\sum_{i=1}^{k-1} \mathcal{F}^{(i)}_P}{\tilde{a}_P}. \tag{14}$$

As pointed out earlier, the Rhie-Chow interpolation is based on the determination of an expression for the cell face velocity. An equivalent relationship as the one for the centered velocity Eqn.(14) can be defined for the face (convective) velocity (which is also divergence-free):

$$\boldsymbol{U}^j_f = \frac{1}{\Delta t a_{k,k}} \frac{\boldsymbol{U}^n_f}{\tilde{a}_f} + \frac{H_f(\boldsymbol{U}^*_f)}{\tilde{a}_f} - \frac{(\nabla p^{j})_f}{\tilde{a}_f} + \frac{1}{a_{k,k}} \frac{\sum_{i=1}^{k-1} \mathcal{F}^{(i)}_f}{\tilde{a}_f}, \tag{15}$$

where, following [12], in order to have a consistent scheme, the following interpolations are used:

$$\frac{H_f(\boldsymbol{U}^*_f)}{\tilde{a}_f} = \overline{\frac{H_P(\boldsymbol{U}^*)}{\tilde{a}_P}}, \tag{16}$$

and

$$\frac{1}{\tilde{a}_f} = \overline{\frac{1}{\tilde{a}_P}}. \tag{17}$$

It is noted that as Eqn.(5) shows, H_P only contains spatial contribution as a result of the discretization of the spatial operators [4]. It does not contain the previous time-step solution, nor previous iteration solution (if under-relaxation is used), nor the contribution of the previous stages (which appears as sum of previous stage residuals). The evaluation of $\mathcal{F}_f$ is also discussed in section 4.1.

The discrete continuity equation is given by:

$$\sum_f \boldsymbol{U}^j_f.\boldsymbol{n}_f S_f = 0. \tag{18}$$

Substituting Eqn.(15) into Eqn.(18):

$$\sum_f \left[\frac{1}{\tilde{a}_f}(\nabla p^j)_f\right].S_f = \sum_f \left(\frac{1}{\Delta t a_{k,k}}\frac{\boldsymbol{U}^n_f}{\tilde{a}_f} + \frac{H_f(\boldsymbol{U}^*_f)}{\tilde{a}_f} + \frac{1}{a_{k,k}}\frac{\sum_{i=1}^{k-1}\mathcal{F}^{(i)}_f}{\tilde{a}_f}\right).S_f \tag{19}$$

which is the equation for obtaining the pressure field that results in a divergence free velocity field.

4.1 Evaluation of the cell and face momentum residuals

After the computation of the solution field at each stage, the cell-centered and face residual vectors ($\mathcal{F}_p$ and $\mathcal{F}_f$) must be evaluated. The evaluation $\mathcal{F}_P$ is given by :

$$\mathcal{F}_P = -a_P\boldsymbol{U}_P + H_P - (\nabla p)_P, \tag{20}$$

Note that a_P is the diagonal coefficient matrix as a result of spatial discretization. The resultant residual vector for the whole domain can be expressed by:

$$\mathcal{F} = -C(\phi)\boldsymbol{U} - B\boldsymbol{U} - Gp + \boldsymbol{r}. \tag{21}$$

The evaluation of $\mathcal{F}_f$ is not as clear as $\mathcal{F}_P$, since the discretized operators at the face are not available. Here F_f is computed based on the fully-discretized form of $\frac{d\boldsymbol{U}_f}{dt} = \mathcal{F}_f$:

$$\frac{\boldsymbol{U}^{(k)}_f - \boldsymbol{U}^n_f}{\Delta t a_{k,k}} = \mathcal{F}^{(k)}_f + \frac{1}{a_{k,k}}\sum_{i=1}^{k-1}\mathcal{F}^{(i)}_f. \tag{22}$$

Noting that at the end of a stage, $\boldsymbol{U}^{(k)}_f$ is known, it is possible to evaluate $\mathcal{F}_f$ using:

$$\mathcal{F}^{(k)}_f = \frac{\boldsymbol{U}^{(k)}_f - \boldsymbol{U}^n_f}{\Delta t a_{k,k}} - \frac{1}{a_{k,k}}\sum_{i=1}^{k-1}\mathcal{F}^{(i)}_f \tag{23}$$

A similar approach was used in [13] to compute the residual vector for a different (spatial and) temporal discretization scheme.

5 RESULTS

The presented solution algorithm for advancing the solution to the transient INS in time using the ESDIRK schemes for time integration was implemented into OpenFoam v1.7.1 (an open source CFD package). The solver has BDF1 as one of the available time integration schemes and as mentioned earlier can be used as a basis to construct the higher order ESDIRK schemes. The temporal consistency and accuracy of the algorithm are investigated using a two-dimensional lid driven cavity with $Re = 10$ as a test case. Prior to discussing the result several definitions are given. The temporally exact solution $\boldsymbol{U}_{exact}$ is defined as one acquired by solving the problem using a fine time step. The time integration error ϵ_t is defined as the difference between the temporally exact solution and the solution obtained using a coarse time step relative to the temporally exact solution ($\boldsymbol{\epsilon}_t = \boldsymbol{U}_{exact} - \boldsymbol{U}$).

Computations are performed using the iterated PISO algorithm with a fixed number of (outer) iterations. The selected number of iterations was sufficient to neglect the influence of the splitting error on temporal accuracy of the solution algorithms for the considered accuracy levels. The temporally exact solution was obtained by solving the problem using a time step of $\Delta t_{fine} = 10^{-6}$. Computations were carried out to $t_{final} = 0.01$. Other model parameters are shown in Table.(2).

Table 2: Set of parameters used in analyzing temporal accuracy of the various solution algorithms.

ν	L_x	L_y	N_x	N_y	PISO loops	Outer Iter	solver Tol
0.01	0.1	0.1	50	50	2	20	10^{-15}

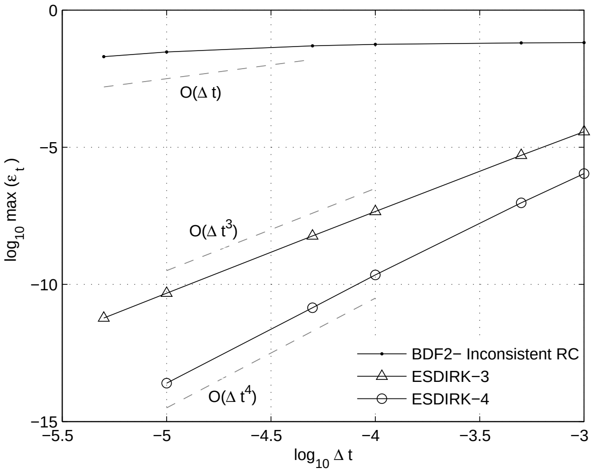

Figure 1: Temporal error convergence of two solution algorithms: 1) An inconsistent Rhie-Chow interpolation using BDF2 for time integration.2) A consistent Rhie-Chow using the higher order esdirk schemes for time integration.

By investigating the temporal convergence of an algorithm it is possible to detect mistakes in the implementation that would be otherwise difficult to find. For example, in order to demonstrate the influence of using an inconsistent Rhie-Chow interpolation on the temporal convergence of a scheme, computations are performed using the standard INS solver of OpenFoam. It has been pointed out in [9] that the solver uses an inconsistent Rhie-Chow interpolation and to remove this inconsistency, the temporally consistent Rhie-Chow interpolation of [12] is applied. The temporally inconsistent Rhie-Chow interpolation in the standard INS solver of OpenFoam has similarities to the inconsistent Rhie-Chow interpolation of Choi [6]. The second order $BDF2$ is used for time integration. In Fig.(1), the temporal convergence of the time integration error corresponding to the u-component of the velocity is shown (denoted in the figure as $BDF2 - InconsistentRC$). As the figure shows, the temporal error in the solution does not seem to reduce with a specific order (aside from the fact that it does not show second order accuracy (Δt^2)). The results are irrespective of the used time integration scheme.

In Fig.(1), temporal convergence of the time integration error for the third and fourth order ESDIRK schemes using the solution algorithm proposed in section 4 is demonstrated. As the figure demonstrates the design order of the time integration schemes are preserved, validating the temporal consistency and accuracy of the algorithm.

6 CONCLUSIONS

A family of higher order implicit multi-stage time integration schemes (namely ESDIRK) has been used for advancing the solution to the unsteady INS in time. The eventual goal of using the higher order time schemes is to decrease the computational cost to obtain a certain accuracy level (in particular engineering levels of accuracy) relative to traditionally used implicit first and second order time integration schemes. The finite volume method was used for spatial discretization, and co-located arrangement of the primitive variables was considered. Furthermore, an iterated PISO algorithm was used to solve the incompressible Navier-Stokes equations. By using a temporally consistent Rhie-Chow interpolation, higher order temporal accuracy in solving the INS equations on co-located grids has been achieved. For a two-dimensional lid driven cavity test case, the temporal convergence of the solution was investigated, with the third and fourth order ESDIRK schemes for time integration. The results demonstrated the temporal consistency and temporal order preservation of the algorithm. Using a temporally inconsistent Rhie-Chow interpolation which maybe encountered in some of the literature and engineering CFD codes, such as OpenFoam, will deteriorate the temporal accuracy and potential computational efficiency of the higher order time integration schemes.

In the next step, in order to increase the computational efficiency of the algorithm, the iterated-PISO algorithm will be replaced by the PISO algorithm.

7 ACKNOWLEDGEMENTS

The financial support provided by Netherlands Technology Foundation STW is greatly appreciated.

REFERENCES

[1] T.F. Miller, F.W. Schmidt: Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a nonstaggered grid system. *Numerical Heat Transfer*, 14 (1988), 213–233.

[2] C.M. Rhie, W.L. Chow: Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21 (1983), 1525–1532.

[3] H. Versteeg , W. Malalasekera: An Introduction to Computational Fluid Dynamics: The Finite Volume Method. *Prentice Hall; 2nd edition*, (2007).

[4] A. Pascau: Cell face velocity alternatives in a structured colocated grid for the unsteady Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 65 (2011), 812–833.

[5] A. H. van Zuijlen , H. Bijl: Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations. *Computers and Structures*, 83 (2005), 93–105.

[6] S. K. Choi: Note on the Use of Momentum Interpolation Method for Unsteady Flows. *Numer. Heat Transfer A*, 36 (1999),545-550.

[7] M. Ijaz, N. K. Anand: Co-Located Variables Approach Using Implicit Runge-Kutta Methods for Unsteady Incompressible Flow Simulation. *Numerical Heat Transfer,Part B: Fundamentals*, 54 (2008) , 291–313.

[8] H. Jasak: Error Analysis and Estimation for Finite Volume Method with Applications to Fluid Flows. *Ph.D. Thesis, Imperial College, University of London*, (1996).

[9] Z. Tukovic, H. Jasak: A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. *Computers* & *Fluids* 55 (2012),70–84.

[10] M. Auvinen, J. Ala-Juusela, N. Pedersen,T. Siikonen :Time-accurate turbomachinery simulations with open-source cfd; flow analysis of a single-channel pump with OpenFOAM. *V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010*, J.C.F. Pereira and A. Sequeira (Eds).

[11] C. A. Kennedy, M. H. Carpenter: Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, 44 (2003), 139–181.

[12] B. Yu, W.Q. Tao, J.J. Wei: Discussion on momentum interpolation method for colocated grids of incompressible flow. *Numerical Heat Transfer Part B*, 42 (2002), 141–166.

[13] P. M. Gresho, R. L. Lee, S. T. Chan, R. L. Sani: Solution of the time-dependent incompressible Navier-Stokes and Boussinesq equations using the Galerkin finite element method. *Lecture Notes in Mathematics*, 771 (1980).