**HEFAT2014**
**10th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics**
**14 – 16 July 2014**
**Orlando, Florida**

# FINITE ELEMENT SOLUTION OF NAVIER-STOKES EQUATIONS USING KRYLOV SUBSPACE METHODS

Vujičić R. M.
Heatric, Division of Meggitt
Poole, BH16 6LT
United Kingdom
E-mail: mile.vujicic@meggitt.com

## ABSTRACT

This paper presents an analysis of different Krylov subspace methods used to solve non-symmetric, non-linear matrix equations obtained after finite element discretization of Navier-Stokes equations. Mixed velocity-pressure formulation, also known as the primitive variable formulation, which consists of two momentum equations and a zero-velocity-divergence constraint representing mass conservation is applied (2D problem). Matrix equations obtained are solved using following Krylov subspace methods: Least Squares Conjugate Gradient, Bi-Conjugate Gradient, Conjugate Gradient Squared, Bi-Conjugate Gradient Stabilized and Bi-Conjugate Gradient Stabilized (ell). Also, a comparison between these iterative methods and direct Gaussian elimination was made. Findings presented in this paper show that Least Square Conjugate Gradient method with its stability, which has been abandoned by many authors as the slowest, has became very fast when the 'element-by-element' method is applied.

Lid-driven cavity is chosen to be the test case, and results obtained for two different Reynolds numbers; Re = 400 and Re = 1000, and for two discretization schemes (10x10 and 48x48; uniform and non-uniform) are compared with the results presented in literature.

## INTRODUCTION

The finite element application in solving Navier-Stokes equations can be generally divided into two different approaches: primitive variables formulation [2] and stabilized finite element method including penalty method [3], pressure stabilized methods ([4], [5], [6], and [7]) and pressure projection methods [8] and [9]. The primitive variables formulation is the most straight forward finite element procedure for the solution of the non-linear Navier-Stokes equations. To avoid a singular matrix appearance frequently encountered (and satisfy Ladyzhenskaya-Babuška-Brezzi condition [10], [11]) when this approach is applied a reduced interpolation proposed by Taylor and Hood [12] is adopted. The basic idea proposed by Taylor and Hood is that interpolation functions for pressure are one order less than for velocity. Since then, there has been controversy such as this approach is mathematically expedient and it can be used only in cases when viscous effects dominate. However, the main goal of this paper is not to analyze different finite element techniques, but to analyze the behaviour of Krylov subspace methods used to solve matrix equations obtained when primitive variables formulation is applied.

## NOMENCLATURE

| | | |
|---|---|---|
| $u_i$ | [m/s] | Velocity component in $x_i$ direction |
| $v_i$ | [m/s] | Velocity component in $y_i$ direction |
| $P$ | [N/m$^2$] | Pressure |
| $t$ | [s] | Time |
| $g_i$ | [m/s$^2$] | Gravitational acceleration in $x_i$ direction |
| $\rho$ | [kg/m$^3$] | Fluid density |
| $\nabla^2$ | [-] | Laplacian operator in two dimensions |
| $N_i$ | [-] | Interpolation functions for velocities |
| $N_{pi}$ | [-] | Interpolation functions for pressure |
| $r$ | [-] | Number of nodes |
| $\boldsymbol{A}$ | [-] | Matrix |
| $\boldsymbol{A}^T$ | [-] | Transpose of **A** |

Special characters

| | | |
|---|---|---|
| $\psi^{(i)}, \phi(i)$ | [-] | Polynomials of degree i |
| $\omega$ | [-] | Scalar |
| $\varepsilon$ | [-] | Tolerance |
| $\lambda$ | [-] | Relaxation factor |

## GOVERNING EQUATIONS

Considering 2D incompressible flow of a fluid having constant properties, the governing differential equations are:

$$\rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \mu\nabla^2 u_i + \rho g_i \tag{1}$$

and

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2}$$

where i; j = 1; 2.

Applying Galerkin approach (weighting functions equal to the interpolation functions), on a general element within a two-dimensional flow, $u_i^{(e)}$ and $P^{(e)}$ are selected as nodal variables and interpolated:

$$u_i^{(e)} = \sum N_i\, u_i = \boldsymbol{Nu}$$
$$P^{(e)} = \sum N_{pi}\, P_i = \boldsymbol{Np} \qquad (3)$$

To avoid standard derivations, which are presented in many text books such as Lewis et al. [13], the final equations in the matrix form is given only:

$$\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{pmatrix} \dot{u}_{1,\dots,r} \\ \dot{v}_{1,\dots,r} \end{pmatrix} + \begin{bmatrix} K_{11}^{*} & K_{12} \\ K_{12}^{T} & K_{22}^{*} \end{bmatrix} \begin{pmatrix} u_{1,\dots,r} \\ v_{1,\dots,r} \end{pmatrix} = \begin{pmatrix} bu_{1,\dots,r} \\ bv_{1,\dots,r} \end{pmatrix} \qquad (4)$$

where

$$K_{11}^{*} = C + 2K_{11} + K_{22} \qquad (5)$$
$$K_{22}^{*} = C + K_{11} + 2K_{22}$$

Coefficients of convective matrices (equations 4 and 5) are given in the following text:

$$C_{ij} = \iint \rho N_i \left( \underline{u} \frac{\partial N_j}{\partial x} + \underline{v} \frac{\partial N_j}{\partial y} \right) dA$$
$$K_{11ij} = \iint \mu \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} dA$$
$$K_{22ij} = \iint \mu \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} dA$$
$$K_{12ij} = \iint \mu \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial x} dA \qquad (6)$$
$$bu_{1,\dots,r} = \iint \frac{\partial N_j}{\partial x} p dA$$
$$bv_{1,\dots,r} = \iint \frac{\partial N_j}{\partial y} p dA$$

where the velocity components u and v are specified in the element. An eight noded isoparametric element was used in the analysis presented in this paper. As it has been known diffusive terms $K_{11ij}$ and $K_{22ij}$ are symmetric in nature while the convective matrix C is asymmetric.

One of the greatest problems in CFD is a solution of pressure field, and this problem was analyzed by Patankar and Spalding in early seventies [14], [15]. They proposed Semi-Implicit Method for Pressure-Linked Equations so-called SIMPLE, based on a guess-and-correct procedure. Later, Patankar improved SIMPLE by algorithm SIMPLER (SIMPLE Revised) in the early eighties. In this algorithm the discretized continuity equation is used to derive a discretized equation for pressure, instead of a pressure correction equation as in SIMPLE.

Results presented in this paper are obtained using algorithm SIMPLER. Generally speaking, there are few reasons for this decision. Firstly, the resulting pressure field corresponds to the velocity field. Therefore, the application of the correct velocity fields results in the correct pressure field.
Secondly, SIMPLER algorithm is often used as the default procedure in commercial CFD codes.
Except these two algorithms there are few more such as SIMPLEC, SIMPLEX and PISO in use recent years [16].

## ITERATIVE SOLVERS

As problem sizes grow, the storage requirement becomes a burden, even on a modern computer. For this reason, alternative solution strategies have been developed - iterative methods. This family includes the following methods: Jacobi iteration, Gauss-Seidel, Line Relaxation, Successive Over-relaxation, methods based on Conjugate Gradient [17], or Minimum Residual [18]. Also, Multigrid method ([19], [20]) based on a mesh hierarchy construction using one of the previous methods. Performance of Jacobi iteration, Gauss-Seidel, Line Relaxation, Successive Over-relaxation is highly dependent on the diagonal dominance of the coefficient matrices, the mesh size and the boundary conditions. For the SOR method, the estimate of the optimal over-relaxation parameter for general problems is still an open question.

A discretization (finite difference, boundary element, finite element or finite volume) of the Navier-Stokes equations gives a set of nonlinear, non-symmetric algebraic (matrix) equations:

$$\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b} \qquad (7)$$

Algorithms of methods presented in the following text (except BiCGStab(ell)) with and without preconditioning can be found in [21] where transient heat conduction was analyzed.

**Least Squares Conjugate Gradient Method**

Multiplying both sides of equation (7) by $\mathbf{A}^T$ gives:

$$\mathbf{A}^T \mathbf{r} = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x} \qquad (8)$$

The new matrix $\mathbf{A}^T\mathbf{A}$ is always symmetric.

**Bi-conjugate Gradient Method**

The main characteristic of the bi-conjugate gradient method is in replacing the orthogonal sequence of residuals (conjugate gradient method) by two mutually orthogonal sequences [22]. Instead, solving system equations (7), using this method involves solving the following system equations [23]:

$$\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b} \qquad (9)$$
$$\boldsymbol{A}^T \tilde{\boldsymbol{x}} = \boldsymbol{b}$$

As it was mentioned above two sequences of residuals are updated:

$$\boldsymbol{r}^{(i)} = \boldsymbol{r}^{(i-1)} - \alpha^{(i)} \boldsymbol{A}\boldsymbol{p}^{(i)}$$
$$\tilde{\boldsymbol{r}}^{(i)} = \tilde{\boldsymbol{r}}^{(i-1)} - \alpha^{(i)} \boldsymbol{A}^T \tilde{\boldsymbol{p}}^{(i)}$$

and two sequences of search directions:

$$\boldsymbol{p}^{(i)} = \boldsymbol{r}^{(i-1)} - \beta^{(i-1)} \boldsymbol{A}\boldsymbol{p}^{(i-1)}$$
$$\tilde{\boldsymbol{p}}^{(i)} = \tilde{\boldsymbol{r}}^{(i-1)} - \beta^{(i-1)} \boldsymbol{A}^T \tilde{\boldsymbol{p}}^{(i-1)}$$

For symmetric positive definite systems, this method gives the same results as the conjugate gradient method, but at twice the cost per iteration.

For non-symmetric matrices, it could be shown that convergence behaviour may be quite irregular, and the method may break down. The breakdown situation, which may occur when $\rho^{(i)} = \left(\tilde{r}^{(i-1)}, r^{(i-1)}\right) \approx 0$, can be avoided by so-called look-ahead strategies. Also, the problem can appear if chosen decomposition fails. However, this problem can repaired using another decomposition.

**Conjugate Gradient Squared Method**

Conjugate gradient squared method developed by **Sonneveld** [24] has been derived from the bi-conjugate gradient method to deal with non-symmetric, non-positive real systems of equations.

The main idea for introducing this method is to avoid using the transpose of A as it is in the BiCG and to gain faster convergence for roughly the same computation cost.
In this method the residual and direction vectors are chosen according to:

$$\boldsymbol{r}^{(i)} = \phi^{(i)2}(\boldsymbol{A})\boldsymbol{r}^{(0)}$$
$$p^{(i)} = \psi^{(i)2}(\boldsymbol{A})\boldsymbol{p}^{(0)}$$

Convergence is usually twice as fast as for the BiCG method, but it has a very often highly irregular behaviour. In cases where initial guess is very close to the solution this method will diverge. This is the main reason for taking initial guess equal to 11 during calculations in the case which has been analyzed and results presented in this paper.

Despite this problem, the CGS method works quite well in many cases. The only problem is that the polynomials are squared and rounding errors tend to be more damaging than in the standard BiCG method.

**Bi-conjugate Gradient Stabilized Method**

The bi-conjugate gradient stabilized method was developed by **Van der Vorst** [22] to solve non-symmetric linear systems avoiding the often irregular convergence patterns of the conjugate gradient squared method. Instead of computing the **conjugate gradient squared method** sequence the following equations will be solved:

$$\boldsymbol{r}^{(i)} = \phi^{(i)}(\boldsymbol{A})\psi^{(i)}(\boldsymbol{A})\boldsymbol{r}^{(0)}$$
$$\psi^{(i)}(\boldsymbol{A}) = (\boldsymbol{I} - \omega^{(1)}\boldsymbol{A})(\boldsymbol{I} - \omega^{(2)}\boldsymbol{A}) \ldots (\boldsymbol{I} - \omega^{(i)}\boldsymbol{A})$$

where $\omega^{(i)}$ is chosen to minimize $r^{(i)}$.

**Bi-conjugate Gradient Stabilized Method (ell)**

The BiCGStab (ell) method is developed by **Sleijpen** and **Fokkema** [25], and it combines advantages of BiCG and GMRES(ell)- Generalized Minimum Residual [18].
As it was mentioned earlier the irregular convergence, breakdown situations are main disadvantages of BiCG method, while GMRES can be very expensive in computational cost as well as memory required.

Although the BiCGStab (ell) method performs well for a class of problems, it breaks down in the case where the coefficient matrix has eigenvalues close to the imaginary axis. Since 1993, the BiCGStab (ell) method has been used to solve non-symmetric linear (non-linear) equations obtained after discretization of different partial equations. Although for serial an extensive work has been done for parallel computers [26].

## NUMERICAL RESULTS AND DISCUSSION

The analysis considers a fluid of constant properties inside a square cavity, in laminar motion [27]. Lid driven cavity flow is very often used as a benchmark case for validations of numerical techniques in solution of either steady or unsteady flows [28].

As mentioned earlier, the cavity is divided by 10x10 and 48x48 uniform and non-uniform meshes. Geometry with 48x48 non-uniform mesh is given in **Figure 1**.

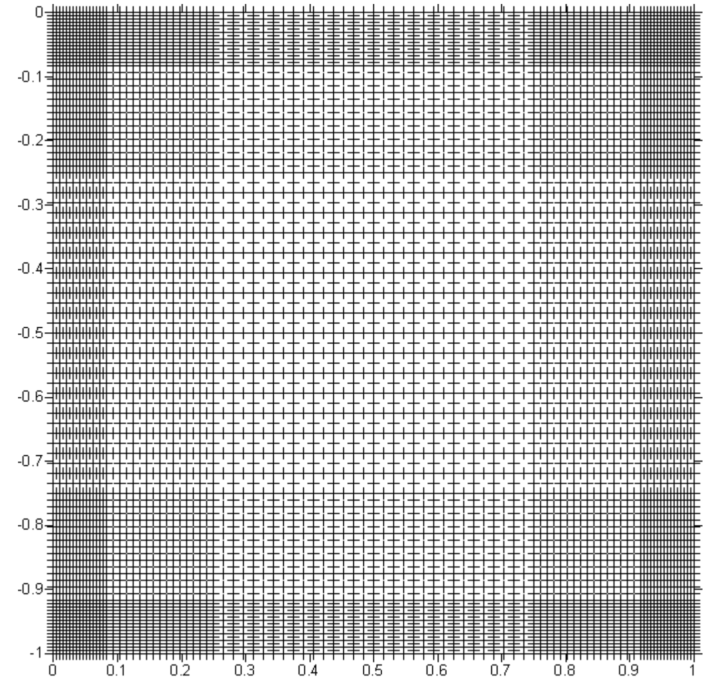


**Figure 1** Non-uniform mesh 48x48

Flow is defined by two Reynolds numbers Re=400 and Re=1000.
Following boundary conditions are applied:
1. $U$ = (0; 0) for $y$ = -1 or $x$ = 0 or $x$ = 1;
2. $U$ = (1; 0) for $y$ = 0;
3. $P$ = (0; 0) for $x$ = 0.

Non-linear solution of the equations analyzed is found using well-known *Picard* scheme:

$$x^{(n+1)} = \lambda x^{(n-1)} + (1-\lambda)x^{(n)} \tag{10}$$

where λ usually takes values between 0 and 1. In the analysis presented in this paper λ was taken to be 0.5. Tolerance $\varepsilon$, which is defined as:

$$\frac{x^{(n+1)} - x^{(n)}}{x^{(n)}} \leq \varepsilon \tag{11}$$

was set to $1.e^{-2}$ and $1.e^{-3}$. At the same time, the maximum error value for iterative solvers during calculations was set to $1.e^{-5}$.

Results obtained are compared with results published by Spalding et al. [14] where finite difference method based on the SIMPLE pressure-correction technique of Patankar and Spalding was applied, Leonard [29] where he used so-called QUICK (Quadratic Upstream Interpolation for Convective Kinematics) method based on a control volume integral formulation, Burgaff [30], and Perić et al. [31] where finite volume was applied (second order central difference scheme - CDS). Also results published by Ghia et al. [32] are used for the comparison.

Velocity and pressure distributions, as well as shear stress, drag and lift of the moving lid are the most desirable variables of almost every calculation.
Velocity distributions in x and y directions for Re=1000 and 48x48 mesh are presented in **Figures 2** and **3**.

As expected, better results are obtained when the finer mesh 48x48 is used in comparison to the coarser 10x10 mesh (results obtained for the 10x10 mesh are not included in this paper). Also, it can be noticed that moving lid creates a strong vortex which position depends on Re number, and two secondary vortices in the lower corners.

A comparison with results presented in literature for Re=400 is presented in **Figure 4**, where velocity component in x-direction was analyzed. A good agreement of results obtained using FEM with results presented by Burgaff can be noticed in the **Figure 4**. Also, it can be clearly seen that better results are obtained using finer grid. A further comparison with results published by Leonard [29] and Perić et al. [31] (results noted as FVM) is presented in the **Figure 5** where velocity of moving lid was defined by Re=1000. An excellent agreement with results published by Perić et al. and Ghia et al. is shown in the **Figure 5**. **Figure 6** illustrates a comparison between results obtained using FEM and FVM [31] and FDM ([29]). In this case velocity in y-direction across the horizontal central plane was chosen. As in the previous example an excellent agreement with results published by [31] is obvious.

From the **Figure 2** to the **Figure 6** is shown that FEM using mixed velocity-pressure interpolation in comparison with other numerical methods, for this particular case, delivers very reliable results.

An analysis of the behaviour of Krylov subspace methods is presented in the following text. Residual norm, error, number of inner and outer iterations, and CPU time are chosen to be relevant criterions in the analysis.
**Figure 7** illustrates trends of residual norm when Least squares conjugate gradient (LSCG) method is applied. Residual norms for first, second, third and the last outer iterations are presented. The number of inner iterations was limited to 500. In the **Figure 7** a monotonically decreasing trend of residual norms can be noticed.

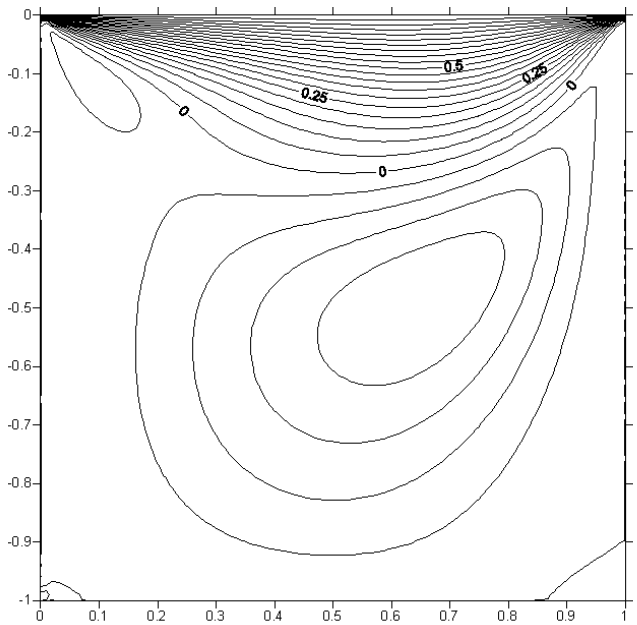


**Figure 2** Non-uniform mesh 48x48. Velocity component in x-direction (U)

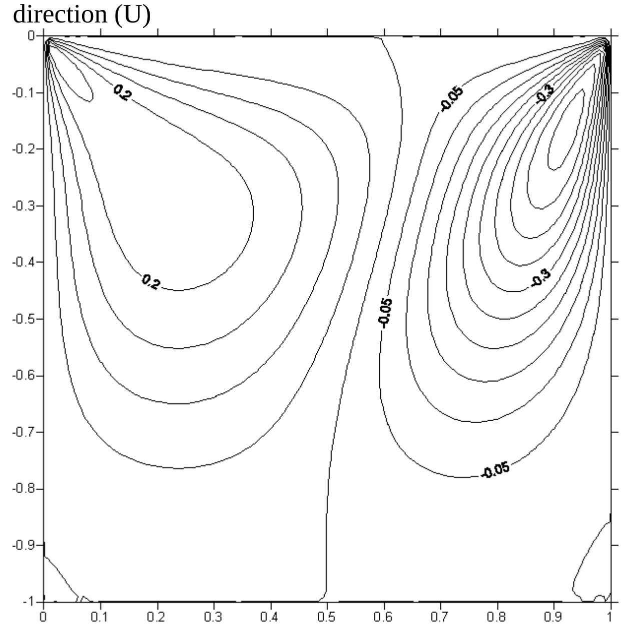


**Figure 3** Non-uniform mesh 48x48. Velocity component in y-direction (V)

At the same time, **Figure 8** presents trends of residual norms for Bi-conjugate gradient (BiCG) method. From the **Figure 8** it can be seen that trends of the residual norm decrease from one outer iteration to another.

At the same time trends of residual norm during inner iterations within an outer iteration is oscillatory. The number of inner iterations was limited to 200.

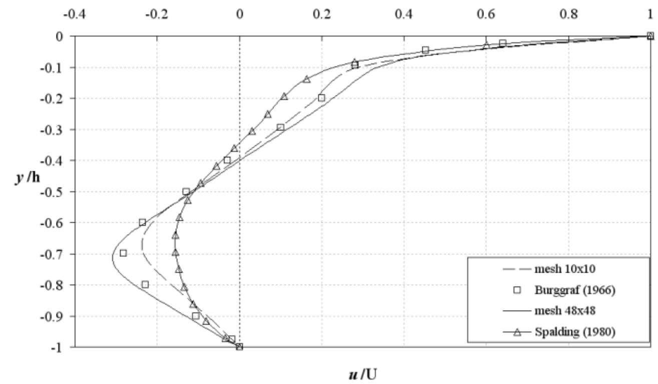


**Figure 4** Velocity component in x-direction (U) for Re=400 compared to results published in literature

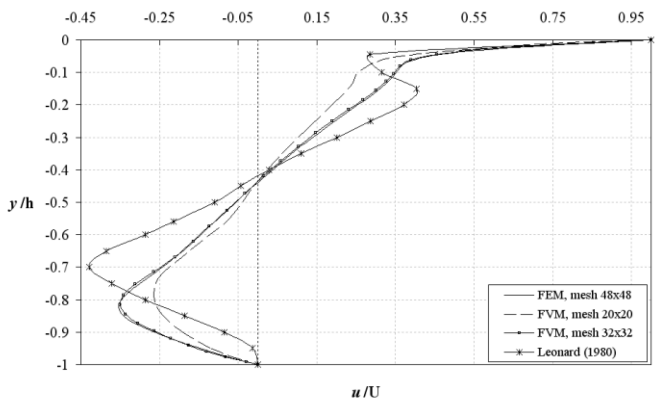


**Figure 5** Velocity component in x-direction (U) for Re=1000 compared to results published in literature

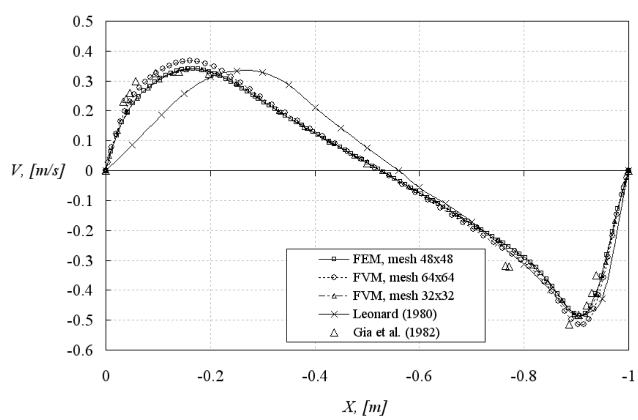


**Figure 6** Velocity component in y-direction (V) for Re=1000 compared to results published in literature

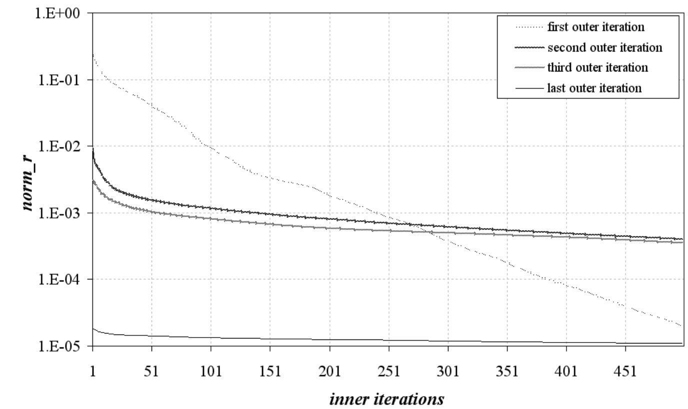


**Figure 7** Residual norm for the LSCG method

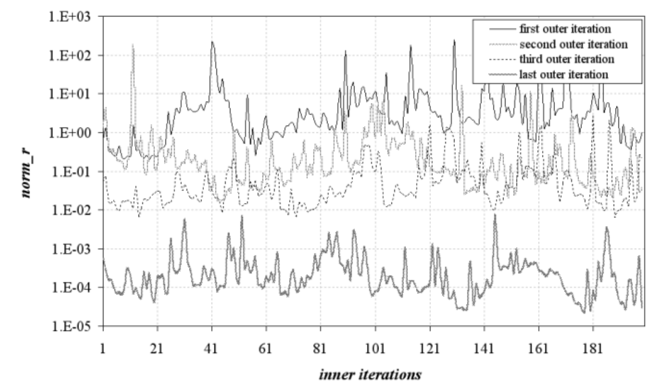


**Figure 8** Residual norm for the BiCG method.

Finally, **Figure 9** (BiCGStab(ell=2)) present residual norms for BCGStab(ell) method. From figure it can be seen that trend of residual norm is more stable and decrease during both inner and outer iterations than for BiCG method. In this case the number of inner iterations was limited to 100.

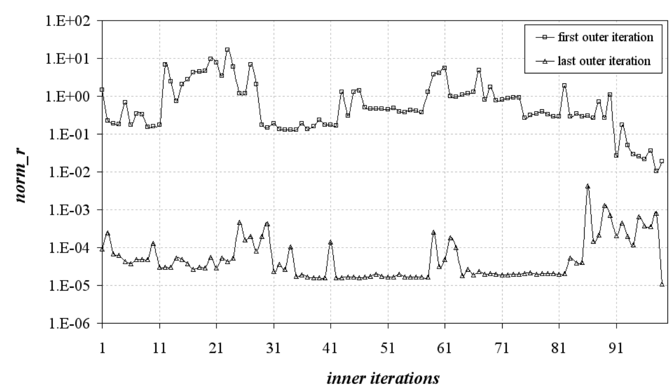


**Figure 9** Residual norm for the BiCGStab (ell=2) method

Error analysis for chosen Krylov subspace methods is presented in **Figures 10** and **11**. The error is defined by the following relation:

$$error = \frac{\|r\|}{\|r_0\|}$$

From **Figures 10** and **11** it can be seen that error trends follow residual trends for methods chosen. Error trends of least squares conjugate gradient and Gaussian elimination methods are monotonically decreasing, while on the other hand for bi-conjugate gradient and conjugate gradient squared (CGS specially ) methods are rapidly oscillatory.

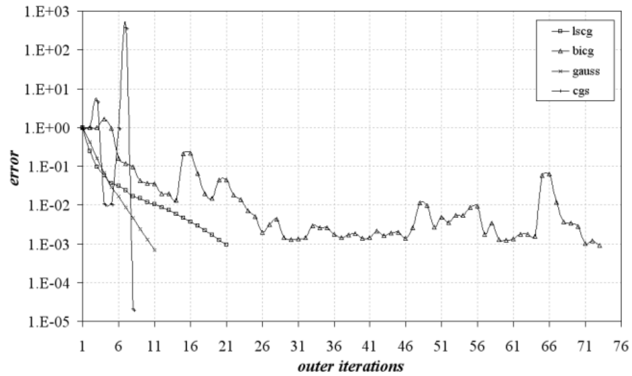


**Figure 10** Error analysis (BiCG, CGS, LSCG)

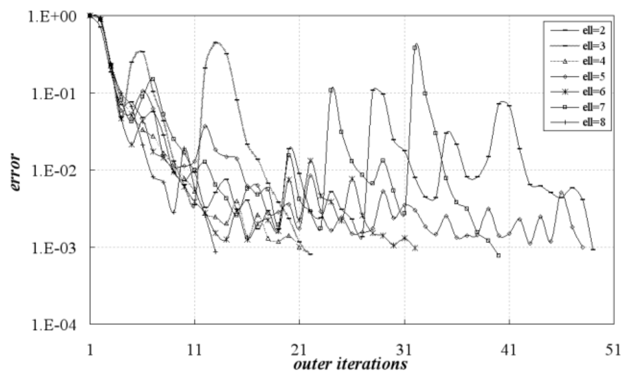


**Figure 11** Error analysis of BiCGStab (ell) methods

CPU time with number of inner and outer iterations used for calculations are presented in **Figures 12** and **13** and **Table 1** for Re=400 and non-uniform 48x48 mesh, **Table 2** for Re=1000 and $\varepsilon= 10^{-2}$, **Table 3** for Re=1000 and $\varepsilon= 10^{-3}$ using LSCG method, **Table 4** for Re=1000 and $\varepsilon= 10^{-3}$ using BiCGStab(ell=2) method, and **Table 5** for Re=1000 and $\varepsilon= 10^{-3}$ using BiCGStab(ell=4) method.

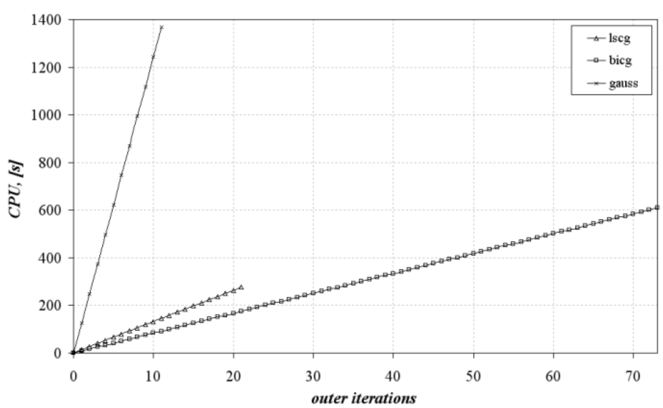


**Figure 12** CPU time for the Gaussian elimination, LSCG, and BiCG methods

It should be expected that an increase of ell values for BiCGStab (ell) method will cause a decrease of CPU time (faster convergence) and a decrease of outer iterations. This conclusion has been drawn by some researchers when they used this method to solve linear non-symmetric matrix equations [33]. Unfortunately, matrix equations obtained after finite element discretization of Navier-Stokes equations are non-linear and non-symmetric, so this conclusion is not valid anymore.

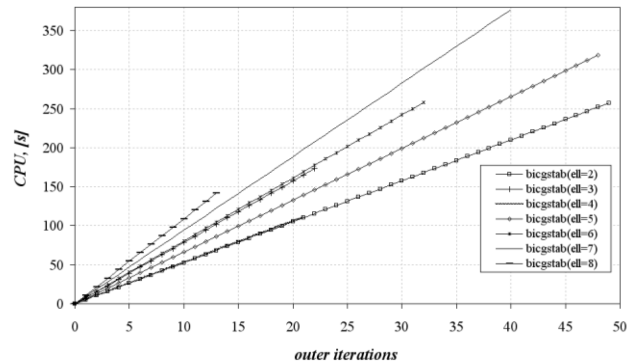


**Figure 13** CPU time for BiCGStab (ell) methods

**Table 1** presents CPU time for Re=400 and tolerance $\varepsilon= 10^{-2}$. The fastest but irregular convergence was achieved by CGS method. To achieve results presented in diagrams and **Table 1** an initial value of order 11 for the CGS method had to be taken.

| Solver | CPU [s] | No. of outer iterations | No. of inner iterations |
|---|---|---|---|
| BiCGStab (ell=2) | 257.3 | 49 | 100 |
| BiCGStab (ell=3) | 173.3 | 22 | 100 |
| BiCGStab (ell=4) | 111.5 | 21 | 50 |
| BiCGStab (ell=5) | 318.3 | 48 | 50 |
| BiCGStab (ell=6) | 258 | 32 | 50 |
| BiCGStab (ell=7) | 377 | 40 | 50 |
| BiCGStab (ell=8) | 141.7 | 13 | 50 |
| LSCG | 113 | 24 | 100 |
| BiCG | 609.4 | 73 | 200 |
| CGS | 40 | 8 | 300 |
| Gauss | 1366.6 | 11 | |

**Table 1** CPU time, number of inner and outer iterations for Re=400

As it has been known the behaviour of CGS method is highly irregular when initial guess is very close to the solution. Also this method does not converge for any number of inner iterations. The fastest convergence of BiCGStab (ell) method was gained when ell was 4.

Also, a very fast convergence of LSCG can be noticed (only 2 [s] slower than BiCGStab (4)).

Results obtained for Re=1000 are presented in **Table 2** where the fastest convergence was gained by LSCG method (except CGS method). Also, it has to be mentioned that LSCG method converged for any number of inner iterations.

A tolerance decrease to the value of $10^{-3}$ caused an increase of CPU time and results obtained are presented in **Tables 3**, **4** and **5**. In these tables a very fast convergence of LSCG method can be noticed. Also, BiCGStab (ell) method does not converge for any number of inner iterations for $2 \leq ell \leq 8$.

| Solver | CPU [s] | No. of outer iterations | No. of inner iterations |
|---|---|---|---|
| BiCGStab (ell=2) | 1169.2 | 27 | 150 |
| BiCGStab (ell=3) | 779.3 | 12 | 150 |
| BiCGStab (ell=4) | 1124.23 | 39 | 50 |
| BiCGStab (ell=5) | 762.4 | 21 | 50 |
| BiCGStab (ell=6) | 1046.7 | 24 | 50 |
| BiCGStab (ell=7) | 872.2 | 17 | 50 |
| BiCGStab (ell=8) | 791.3 | 13 | 50 |
| LSCG | 110.3 | 25 | 100 |
| BiCG | 952.4 | 73 | 200 |
| CGS | 26 | 4 | 300 |
| Gauss | 3476.9 | 16 | |

**Table 2** CPU time, number of inner and outer iterations for Re=1000

| CPU [s] | No. of outer iterations | No. of inner iterations |
|---|---|---|
| 1045.9 | 459 | 50 |
| 4470.14 | 1001 | 100 |
| 3512.5 | 487 | 150 |
| 2353.9 | 270 | 200 |
| 1540.4 | 120 | 300 |
| 1159.7 | 67 | 400 |
| 944.82 | 44 | 500 |
| 957.66 | 37 | 600 |

**Table 3** LSCG - CPU time, number of inner and outer iterations for Re=1000 and $\varepsilon = 10^{-3}$

| CPU [s] | No. of outer iterations | No. of inner iterations |
|---|---|---|
| 1534.2 | 52 | 100 |
| 2579.1 | 60 | 150 |
| 4767.2 | 80 | 200 |
| 1484 | 17 | 300 |
| 5319.5 | 3 | 400 |

**Table 4** BiCGStab(ell=2)- CPU time, number of inner and outer iterations for Re=1000 and $\varepsilon = 10^{-3}$

| CPU [s] | No. of outer iterations | No. of inner iterations |
|---|---|---|
| 1523.67 | 52 | 50 |
| 1683.9 | 29 | 100 |
| 3564.3 | 40 | 150 |
| 12106 | 103 | 200 |

**Table 5** BiCGStab(ell=4)- CPU time, number of inner and outer iterations for Re=1000 and $\varepsilon = 10^{-3}$

## CONCLUSION

In this paper Krylov subspace methods employed to solve non-linear non-symmetric equations obtained after finite element discretization of Navier-Stokes equations where analyzed. Also, as it was mentioned earlier, mixed velocity-pressure formulation as one of finite element techniques was applied.

Results obtained for two values of Reynolds number are compared against results already published using other numerical methods such as finite difference and finite volume methods. From Figure 6 to Figure 8 an excellent agreement with other numerical methods (finite volume method specially) can be noticed. Obviously, an increase of mesh density (from **Figure 2** to **Figure 5**) will give more accurate velocity distributions.

Behaviour of Krylov subspace methods using residual norms, error trends with numbers of inner and outer iterations, and CPU times as the main goal of this paper is analyzed. All trends shown from **Figure 9** are expected and agree with previous published results. As it can be noticed a different number of inner iterations was used in calculations. There is a strong evidence that some methods diverge when the number of inner iterations is too small.

At the same time the LSCG method converged for any rational number of inner iterations.

As it was shown earlier a very fast convergence of LSCG is achieved when element-by-element technique is applied. Also, it can be noticed that the LSCG method became even faster than BiCGStab (ell) method for the case analyzed.

## REFERENCES


[1] Zienkiewicz O. C. and Morgan K., *Finite Elements and Approximation, Wiley*, 1983.

[2] Rao R. R. Baer T. A. Subia S. R. Schunk P. R., Heroux M. A. and Sun A. C., Iterative Solvers and Preconditioners for Fully-coupled Finite Element Formulations of Incompressible Fluid Mechanics and Related Transport Problems, *Sandia National Laboratories Albuquerque*, *New Mexico*, 2001.

[3] Donéa J. and Huerta A., *Finite Element Method for Flow Problems*, *Wiley*, 2003.

[4] Engelman M. S. and Hasbani I., Matrix-free solution algorithms in a finite element context, *Report 88-1*, *Fluid Dynamics International*, Evanston, Illinois, 1988.

[5] Brooks A.N. and Hughes T.J.R., Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, *Computer Methods in Applied Mechanics and Engineering*, 32:199-259, 1982.

[6] Franca L. P. Hughes T. J. R. and Balestra M., A New Finite Element Formulation for Computational Fluid Dynamics: V Circumventing the Babuška-Brezzi Condition: A Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-order Interpolations, *Computer Methods in Applied Mechanics and Engineering*, 59:85-99, 1986.

[7] Mittal S. Tezduyar T. E., Shih R. and Ray S. E., Incompressible Flow Computations with Stabilized Bilinear and Linear Equal Order Interpolation Velocity-Pressure Elements, *Research Report UMSI 901165*, *University of Minnesota*, 1990.

[8] Blasco J. and Codina R., Stabilized Finite Element Method for the Transient Navier-Stokes Equations Based on a Pressure Gradient

Projection, *Computer Methods in Applied Mechanics and Engineering*, 182:277-300, 2000.
[9] Dohrmann C. R. and Bochev P. B., A Stabilized Finite Element Method for the Stokes Problem Based on Polynomial Pressure Projections, *International Journal for Numerical Methods in Fluids*, 46:183-201, 2000.
[10] Babuška I., Error Bounds for Finite Element Method. *Numerical Mathematics*, 16:322-333, 1971.
[11]Brezzi F., On the Existence, Uniqueness and Approximation of Saddle-point Problems Arising from Lagrange Multiplies, *RAIRO Anal. Numer*. (R-2), pages 129-151, 1974.
[12] Hood P. and Taylor C., Navier-Stokes Equations Using Mixed Interpolation, *International Symposium on Finite Element Methods in Flow Problems*, *University of Wales Swansea*, UK, 1974.
[13] Thomas H. R. Lewis R. W., Morgan K. and Seetharamu K. N., *The Finite Element Method in Heat Transfer Analysis*, *Wiley*, 1996.
[14] Pun W. M. Al-Sanea S. A. and Spalding D. B., Computation of Two-dimensional Elliptic Flows Including Heat Transfer, *Computer Methods in Fluids*, pages 217-256, 1980.
[15] Patankar S.V., *Numerical Heat Transfer and Fluid Flow*, *Hemisphere Publishing Corporation*, *Taylor & Francis Group,* 1980.
[16] Versteeg H. K. and Malalasekera W. *An Introduction to Computational Fluid Dynamics, Prentice Hall*, 1997.
[17] Fletcher R., Conjugate Gradient Methods for Indefinite Systems. *Springer*, pages 73-89, 1976.
[18] Saad Y. and Schultz M., GMRES: A General Minimal Residual Algorithm for Solving Non-symmetric Linear System, *SIAM Journal of Sci. Stat.Comp.,* 7:856-869, 1986.
[19] Wesseling P., *An Introduction to Multigrid Method, Wiley,* 1992.
[20] Lavery N.P., Iterative Methods for the Finite Element Solution of Turbulent and Incompressible Flow, *Ph.D. Thesis, School of Engineering, University of Wales Swansea*, 1996.
[21] Vujičić M. R., Finite Element Solution of Transient Heat Conduction Using Iterative Solvers, *Engineering Computations*, 23:408-431, 2006.
[22] Van der Vorst H., BiCGStab, A Fast and Smoothly Converging Variant of BiCG for the Solution of Non-symmetric Linear Systems. *SIAM Journal of Sci. Stat.Comp.,* 18:631-634, 1992.
[23] Tong C. H. and Ye Q., Analysis of the Finite Precision Biconjugate Gradient Algorithm for Non-symmetric Linear Systems, *Report SCCM 95-11, Stanford University*, 1995.
[24] Sonneveld P., CGS, A Fast Lancosz-type Solver for Non-symmetric Linear Systems. *SIAM Journal of Sci. Stat.Comp*., 10:36-52, 1989.
[25] Sleijpen G. L.G. and Fokkema D.R., BICGSTAB (L) for Linear Equations Involving Un-symmetric Matrices with Complex Spectrum, *Electronic Transactions on Numerical Analysis*, 1:11-32, 1993.
[26] Smith I. M. and Griffits G. V., *Finite Element Method Programming, Wiley*, 2004.
[27] Koseff J. R. and Street R. L., Visualization Studies of a Shear Driven Three-dimensional Incompressible Navier-Stokes equations, *Journal of Fluids Engineering,* 106:21-29, 1984.
[28] Lé T.-H. Deville M. and Morchoisne Y., Numerical simulation of 3-d Incompressible Unsteady Viscous Laminar Flows, *Notes on Numerical Fluid Meshanics, Vieweg, Wiesbaden*, 1992.
[29] Leonard B. P. The Quick Algorithm, A Uniformly Third-Order Finite Difference Method for Highly Convective flows, *Computer Methods in Fluids*, pages 159-195, 1980.
[30] Burgaff O. R., Analytical and numerical studies of the structures of steady separated flows. *Journal of Fluids Mechanics*, 24:113-151, 1966.
[31] Ferziger J. H. and Perić M., *Computational Methods for Fluid Dynamics*, *Springer,* 2002.
[32] Ghia K.N. Ghia U. and Shin C.T., High-re Solution for Incompressible flow Using the Navier-Stokes Equations and a Multigrid Method, *Journal of Computational Physics*, 48:387-411, 1982.
[33] Sertel K. Topsakal E., Kindt R. and Volakis J. L., Evolution of the BICGSTAB (l) Algorithm for the Finite-Element/Boundary-Integral Method, *IEEE Antennas and Propagation Magazine*, 43:124-131, 2001.