A VERIFICATION TEST FOR A DIRECT NUMERICAL SIMULATION CODE THAT USES A HIGH ORDER DISCRETIZATION SCHEME

Homero Ghioti da Silva
Departamento de Engenharia de Materiais, Aeronáutica e Automobilística-Escola de Engenharia de São Carlos - Universidade de São Paulo, Brazil
ghioti@sc.usp.br

Marcello Augusto Faraco de Medeiros
Departamento de Engenharia de Materiais, Aeronáutica e Automobilística-Escola de Engenharia de São Carlos - Universidade de São Paulo, Brazil
marcello@sc.usp.br

Leandro Franco de Souza
Instituto de Ciências Matemáticas e de Computação-Departamento de Ciências de Computação e Estatísitica-Universidade de São Paulo, Brazil
lefraso@icmc.usp.br

Abstract
In the present paper a verification test of a direct numerical simulation code for the Navier/Stokes equations was carried out. In the code high order finite-difference schemes were used for the spatial derivative calculations and a 4th order Runge-Kutta method for the time integration. The verification method is based on the introduction of a source term in the Navier-Stokes equations creating a fictitious problem that has an analytical solution and that, in particular, satisfies the boundary conditions implemented in the code. The numerical solution was compared with the analytical solution providing a good way of verifying the code. A mesh refinement test gave the actual order of approximation used for the spatial derivative calculations. The results showed that the actual order of the spatial derivatives can be obtained and that this technique is an useful tool for debugging numerical codes.

Key Words: Navier/Stokes Equations; Numerical Simulation; Verification Test; High Order Finite Difference; Fictitious Problem.

1. Introduction

It is possible to find erroneous solution in a Direct Numeric Simulation (DNS) even if the governing differential equations have been selected correctly and it the chosen numerical scheme has been proved to be convergent. In general, users of numerical codes are concerned about truncation errors, round off error, convergence errors and errors related to the fitting of the mesh to the boundaries. If a numerical solution describes reasonably well the physical phenomenon studied, programming error are regard as negligible. However transitional and turbulent flows posse a wide range of space and time scales and for simulations, these types of flows require high accuracy of the numerical solution. Therefore, it is necessary to eliminate even the smallest programming error. According to the need above presented, a verification test utilizing a invented analytical solution method (or manufactured solution method) was carried out. The study of the convergence was made by using a mesh refinement test.

The central idea of the manufactured solution method is to introduce a source term in the Navier-Stokes equations creating a fictitious problem that satisfies the boundary conditions implemented in the code. Therefore an analytical solution exists for comparison with the computed solution (Roache, 1989; Shih, 1985). This method can reliably debug a computer program, but in any case, if the boundary conditions are changed, new programming error can occur. Also, when a boundary condition is modified, numerical instability problems can arise (Shih, 1985). Therefore the authors believe that the manufactured solution should satisfy the boundary conditions implemented in the code.

The method is useful also to quantify the numerical accuracy of the calculations in a code (Shih, 1985). The quantification of the numerical accuracy of code is little applied by programmers. But in 1986, the editors of the ASME Journal of Fluids Engineering (JFE) published a brief policy statement requiring that, in a paper, at least a minimal attention should be paid to the quantification of numerical accuracy (Roache & White, 1986). Since then, other journals have adopted similar explicit policies (ASME Editorial Board, 1994; AIAA, 1994; Gresho & Taylor, 1994). The JFE has expanded its original policy, including a prohibition of methods with first-order spatial accuracy (Freitas, 1993; Roache, 1994b). It is important to note that this policy statement was not adopted still by Journal of Brazilian Society of Mechanical Sciences and Engineering.

The numerical code used in this work was developed to simulate the Navier-Stokes equations written in vorticity-velocity formulation for the two-dimensional flow in a channel. Linear and non-linear evolution of disturbances in a laminar flow can be simulated with the code. This code was based on an existing code for spatial direct numerical simulation in boundary layers that was validated with results from linear stability theory (Souza, 2005).

For spatial derivatives it was used implicit methods with finite-difference schemes from 4th to 7th order of accuracy near and at the boundaries and, 4th and 6th order of accuracy for the interior of the domain. For the temporal integration a 4th order Runge-Kutta scheme was adopted. More details of these discretization schemes can be found in Souza (2005). Compared with the traditional 2nd order finite difference approximations the high order schemes provide a better representation of the shorter length scales and at a smaller computational cost (Lele, 1991; Souza, 2005).

The organization of the paper is as follows. Section 2 presents the formulation and the boundary conditions adopted in the current work. Section 3 gives the fictitious problem used to verify the numerical code and describes the verification method using a mesh refinement test. This quantifies the order of accuracy of the numerical scheme adopted. This method was used for the spatial derivative calculation. Section 3 also presents the results obtained in this work and the section 4 gives some final remarks.

2. Formulation

The governing equations for an incompressible, two-dimensional flow were written in a vorticity-velocity formulation. The spanwise vorticity transport equation is

$$\frac{\partial \omega_z}{\partial t} + \frac{\partial u\omega_z}{\partial x} + \frac{\partial v\omega_z}{\partial y} = \frac{1}{Re}(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2}),$$

and the continuity equation is

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0,$$

where u and v denote the streamwise ($x-$direction) and wall normal ($y-$direction) velocity components. With the continuity equation and the vorticity definition that is

$$\omega_z = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}, \tag{1}$$

one finds the following Poisson equation,

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = -\frac{\partial \omega_z}{\partial x}.$$

This allows the calculation of the wall normal velocity component. To solve the Poisson equation, a multigrid iterative method based on a line successive over relaxation scheme (LSOR) was used.

Taking the $y-$derivative of the ω_z-vorticity one obtains the equation

$$\frac{\partial^2 u}{\partial y^2} - \frac{\partial^2 v}{\partial x \partial y} = \frac{\partial \omega_z}{\partial y} \tag{2}$$

which is used to calculate the $u-$velocity.

The above equations were made dimensionless using the following reference parameters: U_{max} which is the velocity at the centerline of the channel and H wich is half channel height. These variables produce the following dimensionless parameters:

$$x = \frac{x^*}{H},\ y = \frac{y^*}{H},\ u = \frac{u^*}{U_{max}},\ v = \frac{v^*}{U_{max}},\ \omega_z = {\omega_z}^* \frac{H}{U_{max}},\ t = t^* \frac{U_{max}}{H},$$

where the terms with an asterisk are dimensional. The Reynolds number (Re) is $\frac{U_{max}H}{\nu}$, were ν, denotes the kinematic viscosity.

The governing equations were integrated in a channel described by the rectangular domain: $D = [x_0,\ x_1) \times [0\,, 2H]$, where x_0 is the inflow and $x_1 = \frac{2\pi}{\alpha}$ is the outflow positions, with α, the wavenumber.

2.1 Boundary Conditions

A periodic boundary condition was adopted in the streamwise direction ($x-$direction). At the walls ($y = 0$ and $y = 2H$) the non-slip and impermeability ($u = v = 0$) conditions were used.

A crucial problem of the vorticity-velocity formulation is that there are no boundary conditions for the vorticity at the wall. Rather, it must be computed from the velocity field to maintain consistency and ensure overall conservation of mass. In this work, the calculation of the wall vorticity used $\frac{\partial v}{\partial x} = 0$ in the equation (1), which resulted

$$\omega_z = \frac{\partial u}{\partial y}. \tag{3}$$

For the calculation of the wall vorticity an iterative procedure was used.

2.2 Numerical Methods

The original code used high order compact finite difference schemes in the spatial directions and for temporal integration a 4th order Runge-Kutta algorithim (Souza, 2005). That code had been developed for the simulation of the spatial evolution of perturbations in the channel Poiseuille flow. This code was then modified to simulate the temporal development of these perturbations in the Poiseuille flow in a channel.

In the $y-$direction the discretization used for the interior of domain was a centered compact finite difference scheme of 6th order of accuracy. For the points near the wall boundaries, an asymmetric scheme of 6th accuracy was used, except for the calculation of the $u-$velocity, where a 5th order of accuracy was adopted. An asymmetric scheme of 7th order of accuracy was used for the $v-$velocity component for points at the wall. In the $x-$direction a centered compact finite difference scheme of 6th order of accuracy was utilized, except for the $v-$velocity component, where 4th order of accuracy was adopted.

For the calculation of the ω_z-vorticity at the wall, an explicit asymmetrics finite difference scheme of the 6th order of accuracy was adopted.

3. Verification Test and Quantification of the Order of Numerical Accuracy

According to Roache (1990) the key in such an approach is to choose a fictitious problem with enough structure in it to exercise all of the terms in the equations and all of the leading error terms in the discretization. The chosen manufactured solution for the current test satisfies the boundary conditions implemented in the code. But, for the current discretization schemes that utilizes for the points near and at the boundaries different order of accuracy that in the domain interior points, it it is hard to indicate actual order of the numerical code.

3.1 The Fictitious Problems

The governing equations for the chosen fictitious problem were

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{4}$$

$$\frac{\partial \omega_z}{\partial t} + \frac{\partial u\omega_z}{\partial x} + \frac{\partial v\omega_z}{\partial y} = \frac{1}{Re}(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2}) + F(x,y). \tag{5}$$

Note the introduction of the source term $F(x, y)$ in the vorticity transport equation.

The analytical solution chosen for the fictitious problem was

$$u(x,y) = \exp(y)y(y+1-\sqrt{5})(y+1+\sqrt{5})(y-2)\cos(\alpha x),$$
$$v(x,y) = \sin(\alpha x)\alpha\exp(y)(y^4 - 4y^3 + 4y^2),$$
$$\omega_z(x,y) = 4\exp(y)\cos(x)(y^4(\frac{1}{4}(1-\alpha^2)) + y^3(\alpha^2+1) - y^2(\alpha^2+2) - 2y + 2).$$

whose the source term was

$$\begin{aligned} F(x,y) \quad = \quad & -\exp(y)\cos(\alpha x)[y^6 8\alpha\exp(y)\sin(\alpha x)Re - y^5 24\alpha\exp(y)\sin(\alpha x)Re + y^4(-8\alpha\exp(y)\sin(\alpha x)Re \\ & +\alpha^4 - 2\alpha^2 + 1) + y^3(96\alpha\exp(y)\sin(\alpha x)Re - 4\alpha^4 - 8\alpha^2 + 12) + y^2(-128\alpha\exp(y)\sin(\alpha x)Re \\ & +4\alpha^4 + 16\alpha^2 + 28) + y(64\alpha\exp(y)\sin(\alpha x)Re + 16\alpha^2 - 16) - (24 + 16\alpha^2)]/Re. \end{aligned}$$

According to the manufactured solution above presented, when α is small the derivatives in $y-$direction is larger than the derivatives in $x-$direction. But when α is large the reverse situation occur. For the case α null, the $v-$velocity component is null and, therefore, it is impossible to obtain the actual order for the $v-$velocity component calculations. The discretization scheme for the manufactured solution has truncation error both in the x and $y-$directions. When α is small the truncation error in the $y-$direction is larger than the truncation error in the $x-$direction and vice-versa. This occur because the truncation error depends on the values of the high order derivatives of the manufactured solution. More details on truncation error can be found in (Lele, 1992; Kopal, 1961).

The numerical solutions of the fictitious problems were compared with the analytical solution. According to Shih (1985), if the program is free of programming errors, then one simply deletes F(x,y) of equation (4) and the numerical code is ready for simulating the problem desired.

To verify the order of spatial approximation, one has to assure that the round-off error and convergence errors (due to iterative methods), are negligible. The convergence errors for the iterative methods was set to 10^{-9} for the calculation of the $v-$velocity component and 10^{-12} for the iteractive procedure that involve the $u-$velocity components to calculate the $\omega_z$vorticity at the wall.

The initial conditions given to the code are the exact solutions. Here, the exact solutions created for verification do not depend on time. However these solutions have to migrate to the numerical solution. There is a short transient in the simulation where the truncation and other errors grow. Therefore, it is necessary to assure that the truncation error on time is also negligible. This is done by using very small time step. For the test, the times step (dt) was set to 10^{-6} and a hundred steps were used in time.

According to the above hypothesis, only the truncation errors and possible programming errors predominate in the absolute numerical error. This error was denoted for $E(x, y)$.

The results are presented by in plots in logarithmic scale. Seven different meshes were used. The number of intervals was doubled from one mesh to the next mesh. For the first mesh, the distance between two consecutive points were $dx = 0.7854$ and $dy = 0.25$ in the x and the $y-$directions, respectively. Note that the spatial domain was $\frac{2\pi}{\alpha}$ for the $x-$direction and $2H$ for the $y-$direction. This gave eight intervals both in the $x-$ and $y-$directions.

Figures from 1 to 3 show the behavior of the average numerical errors and figures from 4 to 6 show the behavior of the maximum numerical errors for the u and $v-$velocities and ω_z-vorticity respectively.

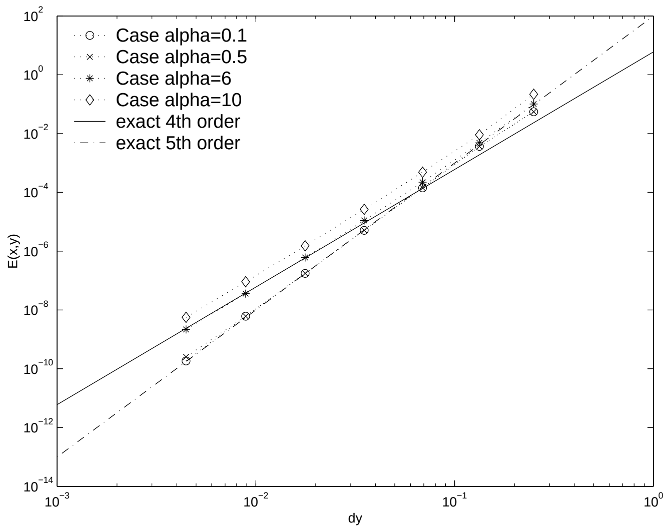

Figure 1. Behavior of the mean numerical errors $E(x, y)$ for $u-$velocity component. The straight lines represent the best fit to the data. When α is small, the straight line has slope of 5. When α is large, the straight line has slope of 4.

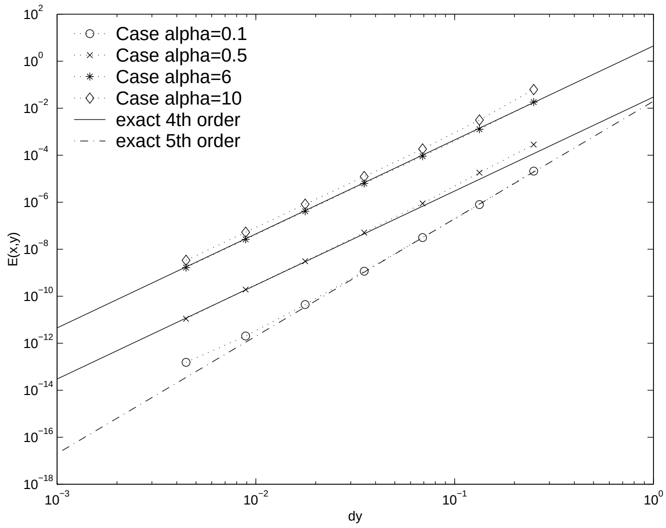

Figure 2. Behavior of the mean numerical errors $E(x,y)$ for $v-$velocity component. When α is small, the straight line has slope of 5, when α is large, the straight line has slope of 4.

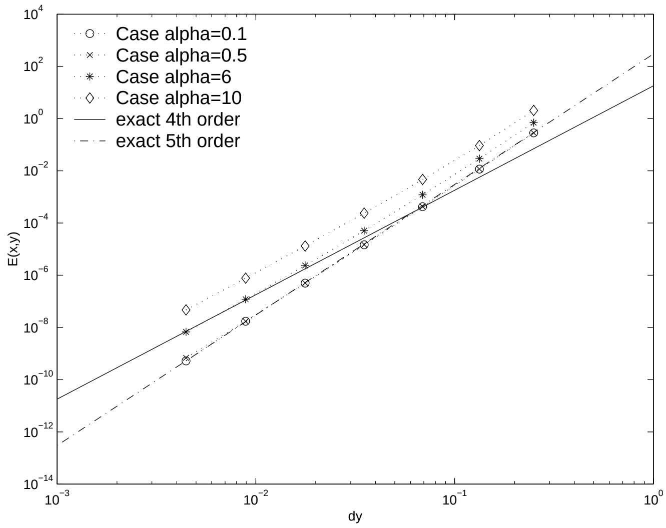

Figure 3. Behavior of the mean of the numerical errors $E(x,y)$ for ω_z-vorticity. The behavior of the mean error for this case is similar to the behavior of the errors presented in figures 1 and 2, but it has different interpretation. When α is small, the straight line has slope of 5, when α is large, the straight line has slope of 4.

In figure 1, when $\alpha = 0.1$ or $\alpha = 0.5$ the $E(x,y)$ converges with 5th order of accuracy but when $\alpha = 6$ or $\alpha = 10$ the $E(x,y)$ converges with 4th order of accuracy except for the coarse mesh, where it converges with 5th order of accuracy. One possible explanation for the resultant order in coarse mesh is the strong influence of the numerical error via the discretization near and in the boundaries, where the truncation error in the $y-$direction is predominant. The straight line that has inclination 5 is consistent with the chosen numerical schemes, because for the $u-$velocity calculation, the numerical scheme utilized an asymmetric discretization of 5th order of accuracy in the $y-$direction near the boundaries. The straight line that has inclination 4 is also consistent with the chosen numerical schemes, because the calculation of the $u-$velocity component depends of the calculation of the $v-$velocity component, that uses, for discretization in the $x-$direction, an explicit scheme of 4th order of accuracy.

In figure 2, the $E(x,y)$ has similar behavior as $E(x,y)$ presented in figure 1. When α is large, it decays with 4th order. But, when α is small the predominant error arises from the source term in the Poisson equation, because the calculation of this source term depends of the calculation of the ω_z-vorticity. The behavior of the error for the calculation of the ω_z-vorticity will be discussed via figures 3 and 6.

The results presented in figures 4 and 5 shows similar results if compared with figures 1 and 2 respectively and the $E(x,y)$ decays with equal order of accuracy.

In figure 3, the mean error has similar behavior as the mean errors presented in figures 1 and 2. But, according to figure 6, it has a different interpretation. In an attempt to understand the order of the mean error in the calculation of ω_z-vorticity, it is crucial to study the behavior of the error in each term of the vorticity transport equation, because they are used for the ω_z-vorticity calculation in the interior of the domain. When α is small, the mean error for the numerical solution of $u-$velocity decays with 5th order. This occurs because the numerical error of the wall ω_z-vorticity calculation is predominant, compared with the numerical error of the ω_z-vorticity calculation in the interior of the domain. These results are show in table 1. It shows the predominance of the numerical error at the wall when α is small or the mesh is 'coarse'. This does not occur if the α is large and the mesh is refined. This results are consistent with the numerical scheme because when α is small the numerical error, in equation (2), of the term $u\frac{\partial \omega_z}{\partial x}$ is larger than the numerical error of the term $v\frac{\partial \omega_z}{\partial y}$. But, when α is large the reverse situation occurs. And, consequently, the error of the 4th order is predominant in the interior of the domain.

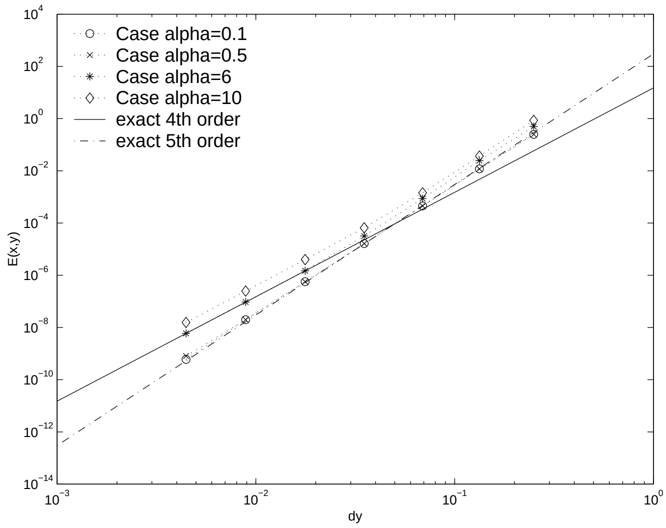

Figure 4. Behavior of the maximum errors $E(x, y)$ for $u-$velocity component. When α is small, the straight line has an agreement of 5, when α is large, the straight line has an agreement of 4.

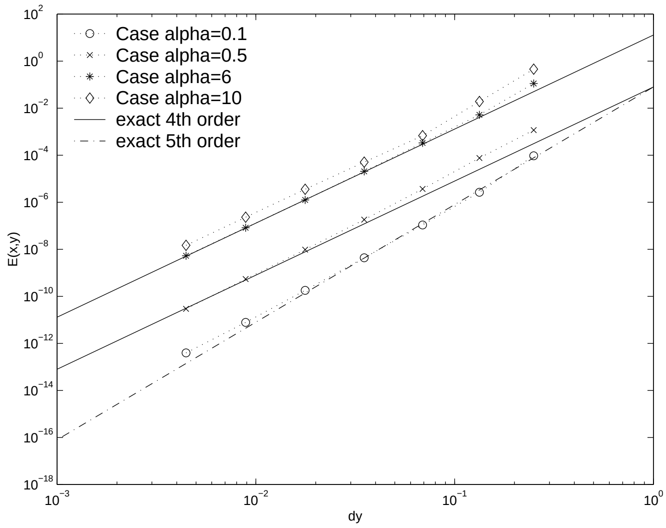

Figure 5. Behavior of the maximum errors $E(x, y)$ for $v-$velocity components.When α is small, the straight line has an agreement of 5, when α is large, the straight line has an agreement of 4.

The figure 6 shows the behavior of the maximum error in $E(x, y)$ for the ω_z-vorticity. Note that the error decays with 4th order, independent of the chosen α. This is consistent with the discretization scheme, because the wall ω_z-vorticity calculation depends of the $y-$derivative of $u-$velocity calculation via equation (3). Therefore, while the numerical error for $u-$velocity calculation decays with 5th order, the $y-$derivative of u decays with 4th order.

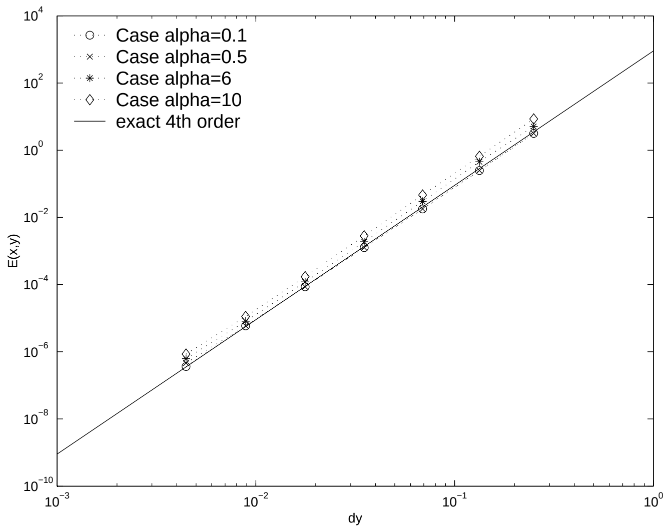

Figure 6. Behavior of the maximum errors $E(x, y)$ for ω_z-vorticity. The straight line has inclination 4 for every chosen α.

Table 1. Comparison of the mean error of the ω_z-vorticity calculation with the error rate of the wall ω_z-vorticity calculation by number of points utilized in each mesh respectively. The columns 2 and 3 show the numerical results with $\alpha = 0.1$ and the columns 4 and 5 show the numerical results with $\alpha = 6$.

Mesh	Mean Error	Distributed Wall Error $E(x,0) + E(x,2H)$	Mean Error	Distributed Wall Error $E(x,0) + E(x,2H)$
1	0.2821	0.2381	2.052	0.2390
2	0.0116	0.0104	0.0931	0.0183
3	$4.2275e^{-4}$	$3.8991e^{-4}$	0.0047	$7.3151e^{-4}$
4	$1.4710e^{-5}$	$1.4036e^{-5}$	$2.3828e^{-4}$	$2.6619e^{-5}$
5	$4.9873e^{-7}$	$4.8478e^{-5}$	$1.3220e^{-5}$	$9.1196e^{-7}$
6	$1.7207e^{-8}$	$1.6777e^{-8}$	$7.7485e^{-7}$	$3.1321e^{-8}$
7	$5.2038e^{-10}$	$2.2634e^{-11}$	$4.6971e^{-8}$	1.2079^{-9}

4. Final Remarks

In the current work, a verification method for DNS was presented and the order of accuracy of a numerical code that uses high order discretization scheme for the spatial derivatives was investigated. The discretization scheme used in this work has different orders of accuracy depending on the direction, position in the grid and equation. This fact requires a lot of attention in the analyze of the results.

The results suggest that the actual order of the code has strong influence from the numerical error of the discretization in $y-$direction, near and at the boundaries. At these locations it was adopted an asymmetric finite difference schemes of lower order of accuracy than in the interior of the domain. The results also suggested a strong influence of the discretization of 4th order of accuracy in the $x-$direction for the Poisson equation. The actual order of the code depends on the coefficients of the truncation errors embedded in the chosen approximations. It also depends on the chosen fictitious problem, in particular, on the values of the high order derivatives of the manufactured solution. It is important to observe that in the case of the ω_z-vorticity calculation the use of the average errors is not a good technique for the study the actual order. Because when α is small the order of accuracy of the maximum error is lower than the order of accuracy of the average error.

The authors believe that the programming errors were eliminated from the code and that the actual order of the numerical code is 4 for the ω_z-vorticity calculation. And, for the $u-$ and $v-$velocity components the actual order is at least 4, but it depends on α, consequently, on the chosen fictitious problem.

The numerical code, presented in this work, will be used to simulate the temporal development of the perturbations (normal modes) in the Poiseuille flow. The validation of the numerical code will be done by using Linear Stability Theory (Schimid & Henninson, 2000). The verification tests carried out on this numerical code is an important requirement for the objective of the simulating transitional flows.

5. Acknowledgments

The financial support of the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and FAPESP(Fundação de Amparo à Pesquisa do Estado de São Paulo) are gratefully aknowledged.

6. Responsibility notice

The authors are the only responsible for the printed material included in this paper.

7. References

AIAA, 1994, "Policy Statement on Numerical Accuracy and experimental uncertainty", AIAA Journal, Vol.32, n.1, pp.3 editorial.

ASME Editorial Board, 1994, "Journal of Heat Transfer Editorial Policy Statment on Numerical Accuracy", ASME Journal of Heat Transfer, Vol.116, n.11, pp.797-798.

Freitas, C. J., 1993a, "Editorial Policy Statement on the Control of Numerical Accuracy", ASME Journal of Fluids Engineering, Vol.115, n.3, pp.339-340.

Gresho, P. M.& Taylor, C., 1994 "",Int. J. Num. Meth. Fluids Vol.19, n.3, editorial.

Kopal, Z., 1961, "Numerical Analysis", Chapman & Hall Ltd, second edition.

Lele S. K., 1992, "Compact Finite Difference Schemes with Spectral-like Resolution", Journal of Computational Physics Vol.103, pp.16-42.

Roache, P. J. & White, F. M., 1986 "Editorial Policy Statement on the Control of Numerical Accuracy", ASME Journal of Fluids Engineering, Vol.108, n.1, pp.2.

Roache, P. J., 1989 "Need for Control of numerical Accuracy", AIAA Thermophysics Conference, Buffalo, New York, June 12-14.

Roache, P. J. 1994b, "Response: To the Comments by Drs. W. Shyy and M. Sindir", ASME Journal of Fluids Engineering, Vol.116, n.2, pp.198-199.

Schimid, P. J. & Henningson D. S. 2000, "Stability and Transition in Shear Flows", Applied Mathematical Sciences, Vol.142, Springer-Verlag, New York, USA.

Souza, L. F. & M. T. Mendonça & M. A. F. Medeiros 2005, "The advantages of using high-order finite differences schemes in laminar-turbulent transition studies", International Journal for Numerical Methods in Fluids, Vol.48, n.5, pp 565-582.

Steinberg, S.& Roache, P. J., 1985 "Symbolic Manipulation and Computational Fluid Dynamics", Journal of Computational Physics, Vol.57, n.30, pp.251-283.

Shih, T. M., Tan C. H. & Hwang, B. C., 1989 "Effects of Grid Staggering on Numerical Schemes, International Journal for Numerical Methods in Fluids, Vol.9, n.2, pp.193-212.

Shih, T. M, 1985 "A Procedure to Debug Computer Programs, International Journal for Numerical Methods in Engineering Vol. 21, pp.1027-1037.