ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

COMPRESSIBLE CODE VERIFICATION USING MMS

Jônatas Ferreira Lacerda
Leandro Franco de Souza
Institute of Mathematical and Computer Sciences, University of São Paulo. São Carlos-SP, Brazil.
jonatasflacerda@hotmail.com, lefraso@icmc.usp.br

Abstract. *With the recent advances in computers processing capabilities and methodologies developed for computational aeroacoustic, numerical simulations have become attractive tools to study numerically aeroacoustical problems. The present paper deals with verification of a spatial direct numerical simulation code for compressible flow simulations designed for direct aeroacoustic simulations, by applying the Method of Manufactured Solutions (MMS). Spatial derivatives are discretized by high order compact finite difference schemes; a fourth-order Runge-Kutta scheme is adopted for time integration; and a numerical compact filter is used to avoid spurious non-physical oscillations. The code is parallelized through domain decomposition in both directions, using the Message Passing Interface (MPI) libraries. This study is part of an ongoing research where the aim is to use the code to study Computational Aeroacoustics (CAA).*

Keywords: *Compressible flow, Computational Aeroacoustics, Verification and Validation, Method of Manufactured Solution*

1. INTRODUCTION

The sound can be described as pressure fluctuations around atmosphere pressure, which propagates at a certain speed and Aeroacoustics studies the sound generated by flow and its propagation. Sound can be produced due to: great mass variations; flow interactions with solid surfaces; and vortices pairing in turbulent flows, as occurs in shear layers. Jet noise generation was extensively studied (Freund, 2001; Boersma, 2004; Bailly and Bogey, 2004; Bogey and Bailly, 2006; Babucke, 2009) due to its application in aerospace field.

First studies on Aeroacoustics were performed by Sir James Lighthill (Lighthill, 1952; Lighthill, 1954) to study sound generation in turbulent jets. He proposed an acoustic analogy where there is a clear distinction between the flow field, where the sound is generated, and the the acoustic field where it propagates. The Lighthill's tensor calculated with flow variables is used in a non-homogeneous wave equation as source term to represent a sound source. This procedure is known as an hybrid methodology, where the sound propagation is decoupled from flow calculation, being it as some kind of flow post-processing. Nowadays, several works use this approach by applying acoustic analogies to temporal flow solutions obtained with Direct Numerical Simulations (DNS), Large Eddy Simulations (LES) or Unsteady Reynolds-averaged Navier-Stokes (U-RANS), from where the acoustic sources are extracted. The flow domain must not be the same as the acoustic domain, being the flow calculations performed in a restrict domain while the sound propagation to far-field is performed in larger domain.

Also is possible to use a direct approach where the sound generation and propagation are calculated together with the flow variables through solution of compressible flow equations. In this case, it can be used DNS or LES simulations. The main advantage is the absence of modeling for sound generation and propagation. However, the flow domain must include sound source region and part of the far-field for acoustic propagation, making it computational expensive, depending on the studied case.

Both approaches are part of Computational Aeroacoustics (CAA) area, which is recent compared to current state of Computational Fluid Dynamics (CFD). CAA differs from CFD because aeroacoustics problems must deal with:

- Wide range of frequencies involved;
- Different scale orders, because acoustic waves generally have low amplitudes compared to hydrodynamic quantities;
- Far-field propagation.

Such characteristics impose some challenges for the numerical code, which must: mandatory have a temporal advance scheme; use an adequate computational mesh if used LES or DNS; have numerical resolution of the higher frequencies which have smaller wavelengths and perform a numerical treatment at boundaries with non-reflective boundaries conditions to avoid domain solution contamination with spurious acoustics waves.

The satisfactory transport of all quantities is obtained with high order spacial and temporal numerical schemes, that have low numerical dispersion and dissipation, which tends to change the phase and amplitude of the traveling waves, respectively; two effects that are extremely undesired in aeracoustics simulations.

This work deals with the implementation and verification of a numerical two-dimensional code to solve the set of equations presented in section 3. In this code spatial derivatives are discretized by high-order compact finite difference

ISSN 2176-5480

scheme. A fourth-order Runge-Kutta scheme is adopted for time integration and a high order numerical compact filter is used to avoid spurious non-physical oscillations. Fourier analysis of the compact finite difference scheme used shows that this discretization scheme has lower error compared with explicit Dispersion-Relation-Preserving (DRP) schemes, diminishing dissipation effects, and also there is no dispersion errors because it is a centered scheme (Lele, 1992). The code is parallelized through domain decomposition in both directions, using the Message Passing Interface (MPI) libraries (Lusk *et al.*, 1996).

The main propose of this work is to show the verification of the implemented code described briefly earlier. This task was performed with aid of a formal procedure called Method of Manufactured Solutions (MMS), which is presented with more details lately.

2. CODE VERIFICATION AND VALIDATION - Code V & V

After code implementation, it is necessary to confirm if it was performed correctly, being the code free of programing and/or modeling mistakes. Such procedure is called as *code verification* and *validation*. According to Silva and Villar (2010), for a good representation of the studied problem, the realization of this analysis is essential.

Code verification is concerned in verify if the code is solving correctly the implemented equations to solve the proposed mathematical model used to represent a physical problem (Roache, 1998). In this step, implementation mistakes are searched and numerical error quantified. A code can be considered validated if it is shown that the equations are solved with the theoretical discretization method precision order. Such analysis is a purely mathematical exercise and no physical requirement must be followed at this phase.

On the other hand, the code validation is performed by comparing the numerical results obtained with the code for a given physical problem with experimental data (Roache, 1998). In this case, physical restrictions are highly necessary and acceptable agreement indicates that the implemented mathematical model is adequate to represent the physical problem in question. Thus, a verified and validated code gives coherent results with the physical problem, with known numerical order and are calculated safety in respect to implementation mistakes (Oberkampf and Trucano, 2002).

Software Quality Assurance (SQA) area deals with the formal procedure tests for codes quality and reliability, that can be divided in basically three parts according to Salari and Knupp (2000):

- **static verification**: to verify if the program is free of compilation errors (not necessary to run the program);
- **dynamic verification**: the code is executed and accessed if there is programing error;
- **formal verification**: any remaining error that do not affect code result is searched.

Among these procedures, dynamic verification is the more important and time consuming, because errors found on this step affect code precision, convergence and efficiency. There are several dynamic tests, since less rigorous, as tendency analysis, to a more demanding one, as is MMS. In the same way of tests classification, so is the acceptance criteria. A specialist subjective analysis of the results can be performed, but also a formal precision evaluation method. According to Salari and Knupp (2000), a more complete and rigorous evaluation method is that one that uses a rigorous test (e.g. MMS) together with a rigorous acceptance criteria (e.g. numerical scheme precision order). This is the procedure used in the present work.

2.1 Method of Manufactured Solutions - MMS

A good way to evaluate numerical results obtained with a code is to compare them with exact results. MMS is an easier way to obtain exact solutions to a set of partial differential equations (PDE's). Firstly are built solutions to the variables to be solved. Then, this solutions are applied to the set of PDE's, generating source terms. After that, the source terms are introduce into the code, creating an unrealistic problem, but with a analytical solution to be compared with. Besides, a mesh refinement test can provide the numerical scheme order, which will be compared with the formal numerical scheme one (Silva and Villar, 2010). If this criteria is fulfilled, the code is considered verified and it is enough just to remove the source terms added from MMS to let code ready for simulations.

Although not necessary physical demand on manufactured solutions, some recommendations are presented by Salari and Knupp (2000). According to the author, smooth functions as polynomials, trigonometric and exponentials are preferred. Also, the solution must be general, not showing strong dependence on space or time variables, and must have a certain number of derivatives to assure that constants or null values are not obtained when derivative are calculated.

The hardest task would be the mathematical manipulation for term source generation from manufactured solutions, when they are applied into the PDE's. However, this is easier performed with the aid of symbolic manipulation codes, such as Wolfram Mathematica®. Besides, this auxiliary codes can provide the source terms in the used programming language format.

The test itself is then performed by running the code using different meshes with progressive grid refinement between them, and carry out a mesh convergence test (Salari and Knupp, 2000), where the discretization error and solution

ISSN 2176-5480

precision order are evaluated. The discretization error is the difference between the numerical and analytical solution, respectively $(\mathbf{Q}_{num} - \mathbf{Q}_{an})$, which can be obtained point to point. On the other hand, there are global error measurements as l_2-norm, given by:

$$l_2 = \sqrt{\frac{\sum_n (\mathbf{Q}_{num} - \mathbf{Q}_{an})_n^{\ 2}}{N}} \tag{1}$$

where N is the total number of points and subscripts $_{num}$ e $_{an}$ correspond to numerical and analytical solution, respectively. The precision order o is given by comparing the results of two consecutive refined meshes:

$$o = \frac{\log\left(\frac{E_{2h}}{E_h}\right)}{\log(r)} \tag{2}$$

where E_{2h} and E_h are global errors (in this case l_2-norm) for coarse and refined mesh, respectively, and r is refinement ration between these two meshes.

3. FORMULATION

The set of equations implemented into the DNS numerical code is presented in this section. The equations are written in dimensionless form and Cartesian reference system (x,y) is used, with respective velocity components u and v. Considering a two-dimensional non-isothermal compressible flow in a transient regime, the solution is obtained by solving continuity, Navier-Stokes (N-S) and energy equations. The solution vector in conservative way in given by:

$$\mathbf{Q} = (\rho, \rho u, \rho v, E)^T \tag{3}$$

containing, as variables, the density ρ, mass fluxes ρu and ρv, and total energy by volume unit E, which is defined as:

$$E = \rho \int c_v dT + \frac{\rho}{2}\left(u^2 + v^2\right) \tag{4}$$

being T the fluid temperature and c_v the heat capacity at constant volume.

The governing equations can be described in vectorial notation as:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{5}$$

where the fluxes vectors are:

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ u(E+p) + q_x - u\tau_{xx} - v\tau_{xy} \end{bmatrix} \tag{6}$$

$$\mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ v(E+p) + q_y - u\tau_{xy} - v\tau_{yy} \end{bmatrix} \tag{7}$$

The pressure is denoted by p. Normal stresses are

$$\tau_{xx} = \frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}\right) \tag{8}$$

$$\tau_{yy} = \frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}\right) \tag{9}$$

shear stress are

$$\tau_{xy} = \frac{\mu}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{10}$$

and heat fluxes are

$$q_x = -\frac{\vartheta}{(\kappa - 1) RePrMa^2} \cdot \frac{\partial T}{\partial x} \tag{11}$$

$$q_y = -\frac{\vartheta}{(\kappa - 1) RePrMa^2} \cdot \frac{\partial T}{\partial y} \tag{12}$$

where μ is the dynamic viscosity, ϑ the thermal conductivity and κ heat capacity ratio. Finally Re, Pr and Ma are Reynolds, Prandtl and Mach numbers, respectively.

ISSN 2176-5480

4. NUMERICAL METHOD

4.1 Numerical Scheme

In this work is used a two-dimensional DNS code to solve the equations presented in Sec. 3. Spatial derivatives are being discretized by high order compact finite difference schemes (Lele, 1992). A fourth-order Runge-Kutta scheme is adopted for time integration and a numerical compact filter is used to avoid spurious non-physical oscillations. The code is parallelized through domain decomposition in both directions, using the MPI libraries. A grid transformation in $(x-y)$ plane is used by mapping physical grid in a equidistant $(\xi-\eta)$ computational grid, being necessary the use of metrics for derivative calculations. First derivatives are given by:

$$\frac{\partial}{\partial x} = \frac{1}{\left(\frac{\partial x}{\partial \xi}\right)}\frac{\partial}{\partial \xi} \tag{13}$$

$$\frac{\partial}{\partial y} = \frac{1}{\left(\frac{\partial y}{\partial \eta}\right)}\frac{\partial}{\partial \eta} \tag{14}$$

Second derivatives are:

$$\frac{\partial^2}{\partial x^2} = \frac{1}{\left(\frac{\partial x}{\partial \xi}\right)^2}\frac{\partial^2}{\partial \xi^2} - \frac{\frac{\partial^2 x}{\partial \xi^2}}{\left(\frac{\partial x}{\partial \xi}\right)^3}\frac{\partial}{\partial \xi} = \frac{\partial^2}{\partial \xi^2}\left(\frac{\partial \xi}{\partial x}\right)^2 - \frac{\partial}{\partial \xi}\frac{\partial^2 \xi}{\partial x^2} \tag{15}$$

$$\frac{\partial^2}{\partial y^2} = \frac{1}{\left(\frac{\partial y}{\partial \eta}\right)^2}\frac{\partial^2}{\partial \eta^2} - \frac{\frac{\partial^2 y}{\partial \eta^2}}{\left(\frac{\partial y}{\partial \eta}\right)^3}\frac{\partial}{\partial \eta} = \frac{\partial^2}{\partial \eta^2}\left(\frac{\partial \eta}{\partial y}\right)^2 - \frac{\partial}{\partial \eta}\frac{\partial^2 \eta}{\partial y^2} \tag{16}$$

being the metrics given by:

$$\frac{\partial^2 \xi}{\partial x^2} = -\frac{\partial^2 x}{\partial \xi^2}\left(\frac{\partial \xi}{\partial x}\right)^3 \tag{17}$$

$$\frac{\partial^2 \eta}{\partial y^2} = -\frac{\partial^2 y}{\partial \eta^2}\left(\frac{\partial \eta}{\partial y}\right)^3 \tag{18}$$

4.2 Case Setup for MMS usage

In this section is presented the case setup for MMS usage to verify the implemented numerical code through a dynamic verification. Equation 5 can be rewritten as:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} \tag{19}$$

where $\mathbf{S}$ is the source term obtained from manufactured solutions.

In the present work, it was used the following manufactured solutions:

$$\rho(x,y,t) = \rho_0\left[\mathrm{sen}\left(x^2+y^2+\omega t\right)+1,5\right] \tag{20}$$

$$u(x,y,t) = u_0\left[\mathrm{sen}\left(x^2+y^2+\omega t\right)+0.5\right] \tag{21}$$

$$v(x,y,t) = v_0\left[\cos\left(x^2+y^2+\omega t\right)+0.5\right] \tag{22}$$

$$E(x,y,t) = \rho_0 e_0\left[\cos\left(x^2+y^2+\omega t\right)+1.5\right] \tag{23}$$

where the constants were defined as $\rho_0 = 1.0$, $u_0 = 0.1$, $v_0 = 1.0$ and, $e_0 = 0.5$ to perform the calculations. The computational grid range was $x \in [-\pi+\pi/16; \pi+\pi/16]$ and $y \in [-\pi+\pi/16; \pi+\pi/16]$. Figure **??** presents the source terms in the entire domain. Prescribed values were used at all boundaries, by solving Eqs. 20 - 23 for each Runge-Kutta sub-step. Source terms were obtained by applying Eqs. 20 - 23 to Eq. 5 with help of Wolfram Mathematica®.

5. RESULTS

Using the source terms presented earlier, the code was executed using seven different meshes, as detached on Tab. 1, and them obtained the global errors and precision order considering l_2-norm.

ISSN 2176-5480

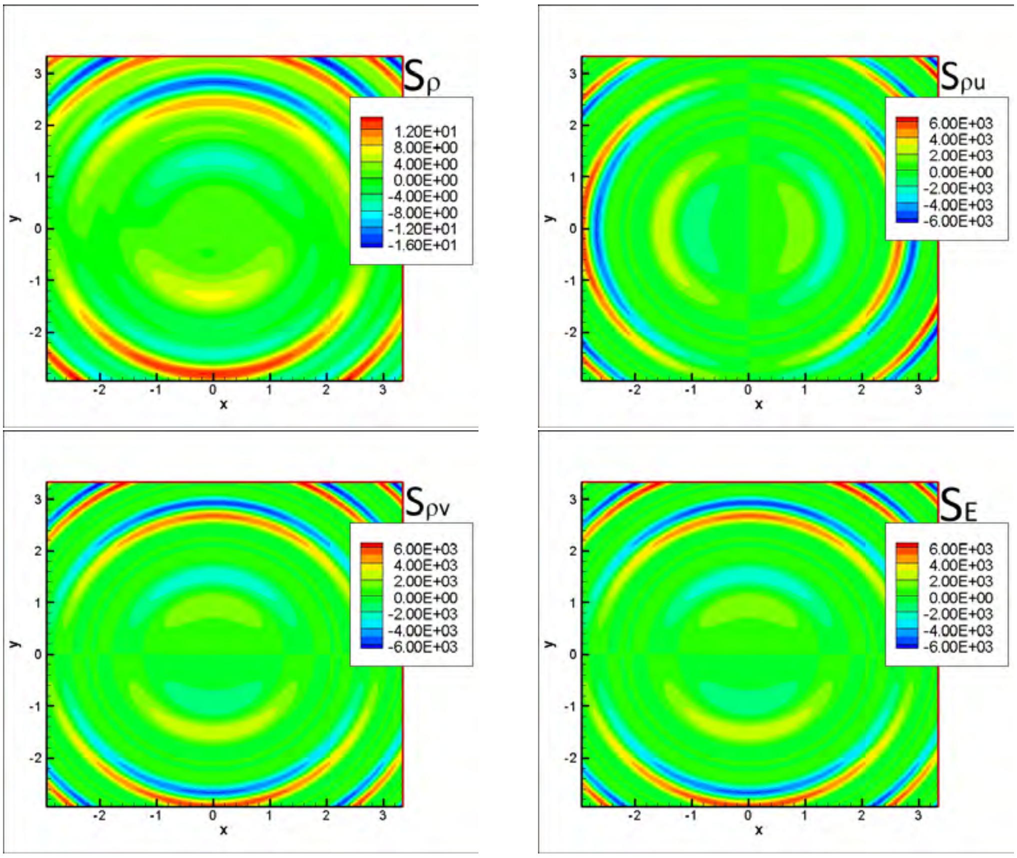

Figure 1. Source terms for MMS test

Table 1. Meshes used in MMS test

Mesh	N_x x N_y	Δx x Δy
1	16 x 16	4,00E-01 x 4,00E-01
2	31 x 31	2,00E-01 x 2,00E-01
3	61 x 61	1,00E-01 x 1,00E-01
4	121 x 121	5,00E-02 x 5,00E-02
5	241 x 241	2,50E-02 x 2,50E-02
6	481 x 481	1,25E-02 x 1,25E-02
7	961 x 961	1,25E-02 x 1,25E-02

A time increment of $\Delta t = 1 \cdot 10^{-8}$ was used and 500 time steps simulated. Other constants are $Re = 1000.0$, $Pr = 100.0$, $Ma = 0.1$, $c_v = 0.1$, $\vartheta = 1.0$, $\mu = 0.3$ e $\kappa = 1.0$.

Figure 2 shows the obtained error while Fig. 3 the precision order observed for each variable from $\mathbf{Q}$. As can be observed, the error observed for ρ is lower than for other variables. In respect to precision order, it was observed an asymptotic point among meshes 3 to 6, presenting approximately fifth and sixth order for all variables.

Looking carefully Fig. 3 it is noticeable that precision order is lower between meshes 2/1 and 3/2, indicating that could have some programing mistake affecting result. However, it occurred because meshes 1, 2 and 3 do not have good resolution to represent high order frequency waves. Figure 4 shows a comparison of the source term for each conservative variable for meshes 1 and 5, where is possible to verify that not all data is captured, damaging, in a certain degree the numerical solution, as shown in Fig 5.

Therefore, according to tests presented in this section it is possible to say that the code is verified through a dynamic verification procedure and free of programing mistakes that could affect its numerical precision order.

6. CONCLUSIONS

In this work is presented a verification procedure for the implementation of a numerical two-dimensional DNS code which will be used to solve compressible flow equations and to study aeroacoustic problems. MMS was used as a formal procedure for the code verification. The results showed that the formal precision order of the compact finite difference

ISSN 2176-5480

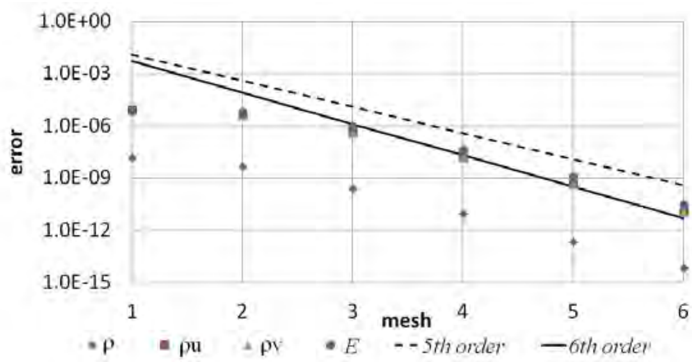

Figure 2. Global errors obtained in MMS test

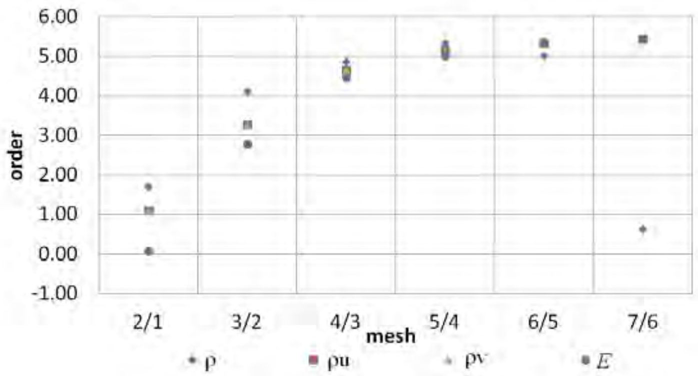

Figure 3. Precision order obtained in MMS test

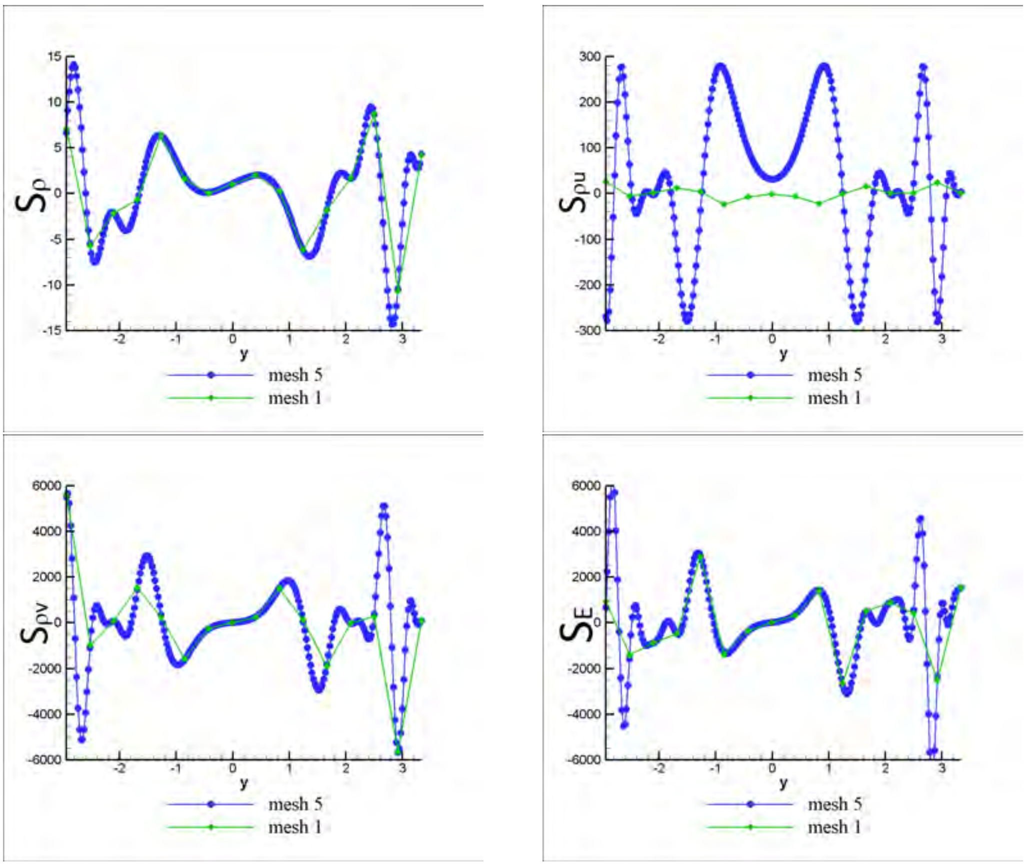

Figure 4. Source terms representation for meshs 1 and 5

ISSN 2176-5480

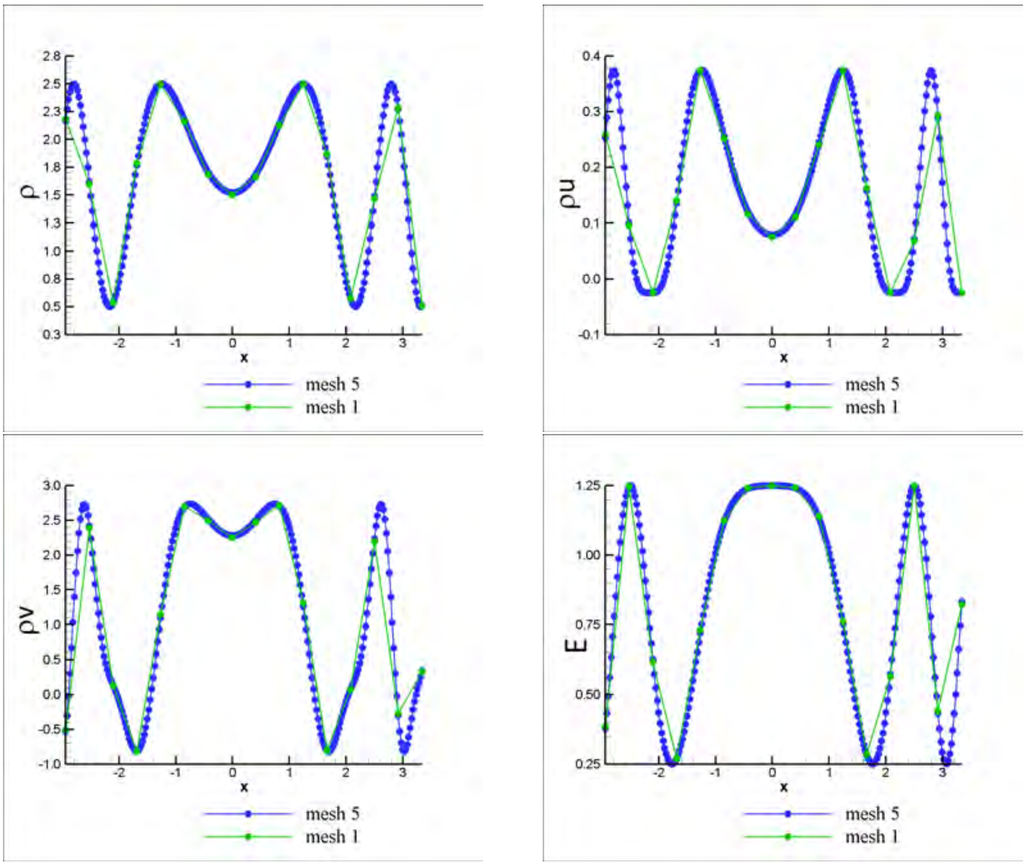

Figure 5. Variables representation for meshs 1 and 5

scheme used for spatial derivatives calculations was obtained, assuring that the code is free of programing mistakes that could affect its numerical precision order. Further works will aim to validate the code by comparing numerical results with experimental one for a benchmark problem, e.g. mixing layer flows.

7. ACKNOWLEDGEMENTS

To Tecumseh Products Company for the financial support and ICMC-USP for offered infrastructure .

8. REFERENCES

Babucke, A., 2009. *Direct Numerical Simulation of Noise-Generation Mechanisms in the Mixing Layer of a Jet*. Ph.D. thesis, University of Stuttgart.

Bailly, C. and Bogey, C., 2004. "Contributions of computational aeroacoustics to jet noise research and prediction". *International Journal of Computational Fluid Dynamics*, Vol. 18, No. 6, pp. 481–491.

Boersma, B.J., 2004. "Numerical simulation of the noise generated by a low mach number, low reynolds number jet". *Fluid Dynamics Research*, Vol. 35, No. 6, pp. 425 – 447.

Bogey, C. and Bailly, C., 2006. "Computation of a high reynolds number jet and its radiated noise using large eddy simulation based on explicit filtering". *Computers & Fluids*, Vol. 35, No. 10, pp. 1344 – 1358.

Freund, J.B., 2001. "Noise sources in a low-reynolds-number turbulent jet at mach 0.9". *Journal of Fluid Mechanics*, Vol. 438, pp. 277–305.

Lele, S.K., 1992. "Compact finite difference schemes with spectral-like resolution". *Journal of Computational Physics*, Vol. 103, pp. 16–42.

Lighthill, M.J., 1954. "On sound generated aerodynamically. ii. turbulence as a source of sound". *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 222, No. 1148, pp. pp. 1–32.

Lighthill, M.J., 1952. "On sound generated aerodynamically. i. general theory". *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, Vol. 211, No. 1107, pp. 564–587.

Lusk, E., Doss, N. and Skjellum, A., 1996. "A high-performance, portable implementation of the mpi message passing interface standard". *Parallel Computing*, Vol. 22, pp. 789–828.

Oberkampf, W.L. and Trucano, T.G., 2002. "Verification and validation in computational fluid dynamics". Technical

ISSN 2176-5480

report, Sandia National Laboratories.
Roache, P., 1998. *Verification and validation in computational science and engineering*. Hermosa Publishers, Albuquerque, N.M.
Salari, K. and Knupp, P., 2000. "Code verification by the method of manufactured solutions". Technical report, Sandia National Laboratories.
Silva, H.G. and Villar, M.M., 2010. "Verificação e validação de códigos computacionais". In *7a Escola de Primavera de Transição e Turbulência – EPTT 2010*.

9. RESPONSIBILITY NOTICE

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