COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0737
VERIFICATION OF A VISCOELASTIC LPTT FLUID FLOW CODE BY THE METHOD OF MANUFACTURED SOLUTIONS

Welton Costa Lavércio
Andreza Beatriz Jacinto da Silva
Juniormar Organista
Leandro Franco de Souza
Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos - SP, Brasil
welton.lavercio@usp.br;andrezabeatriz@usp.br;juniormarorganista@usp.br; lefraso@icmc.usp.br

Abstract. *The verification of in-house numerical codes used for simulating fluid flow is essential to ensure accurate and reliable results. This becomes particularly challenging when dealing with non-linear equations that lack analytical solutions. The Method of Manufactured Solution (MMS) offers a reliable approach to code verification by creating a hypothetical solution and allowing for a comprehensive examination of each code subroutine. In this study, we develop a numerical code to simulate viscoelastic fluid flow and verify its accuracy using the MMS. The mathematical model employed is based on the two-dimensional Navier-Stokes equations, incorporating a viscoelastic LPTT model. Spatial derivatives are discretized using finite difference schemes, with three codes implemented utilizing second, fourth, and sixth-order schemes. For time integration, a fourth-order Runge-Kutta scheme is employed, and a numerical compact filter is incorporated to suppress non-physical oscillations. The MMS is implemented using sine functions. The results demonstrate that the implemented code is error-free, and the order of accuracy associated with each code is verified. This study provides valuable insights into the verification of numerical codes for simulating viscoelastic fluid flows, showcasing the effectiveness of the MMS in ensuring accurate and reliable results.*

Keywords: *Viscoelastic fluids, LPTT Model, Methods of Manufactured Solutions, Runge-Kutta Method, Code Verification, Finite Difference Schemes*

1. INTRODUCTION

Industrial applications involving Non-Newtonian fluids, particularly viscoelastic ones, have stimulated significant research in Computational Fluid Dynamics (CFD) to achieve reliable results using numerical methods. In this context, Verification and Validation (V&V) emerge as fundamental steps to ensure the credibility of the obtained solutions.

This article focuses on Code Verification in CFD, with a specific focus on applications involving Non-Newtonian fluids modeled by Linear Phan-Thien-Tanner (LPTT) constitutive equations. Code Verification entails evaluating the order of accuracy of discretized equations and confirming the code's proper response to the implemented models. It is important to underscore that Verification and Validation are complementary and synergistic concepts, both playing crucial roles in the process of validating numerical results. This study follows the approach proposed by Roache (2012), with a particular emphasis on identifying and analyzing the five main sources of errors in CFD solutions: spatial discretization convergence, temporal discretization convergence, convergence in iterative procedures, machine rounding errors, and programming mistakes. The code verification aims to ensure that numerical procedures are free from programming errors that may affect the accuracy of the results and the interpretation of the studied phenomena.

To conduct the code verification tests, the adopted strategy involves simulating a problem with a known analytical solution similar to the case under study, using the Method of Manufactured Solutions (MMS). The MMS, proposed by Steinberg and Roache (1985) and Roache (2012), involves introducing source terms into the governing equations to create a problem with an analytical solution, allowing comparisons with the numerical solution and obtaining the order of accuracy of the calculations.

The objective of this article is to construct a Manufactured Solution for a Non-Newtonian fluid flow modeled by LPTT and test the solutions and source terms in a code used by the research group. At the conclusion of the tests, it is expected that the code will be adequately verified, enabling the removal or nullification of the source terms in the studied equations and the utilization of the original equations in other group research with greater confidence in the obtained results. By exploring the precision of numerical methods in applications involving Non-Newtonian fluids, this work contributes to the advancement and reliability of Computational Fluid Dynamics studies.

2. MATHEMATICAL FORMULATION

Considering that the flow is incompressible, isothermal, and of a non-Newtonian fluid. The conservation of mass and conservation of momentum equations governing the flow in the dimensionless form are given by

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

and

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = -\nabla p + \frac{\beta}{Re}\nabla^2 \mathbf{u} + \nabla \cdot \mathbf{T}, \tag{2}$$

where $\mathbf{u}$ denotes the velocity field, t is the time, p is the pressure, β is the dimensionless coefficient of the solvent viscosity, Re is the Reynolds number and $\mathbf{T}$ is the extra-stress tensor. The tensors are modeled by constitutive equations that allow the study of viscoelastic fluids. In this paper, the LPTT models were considered, according to the references Thien and Tanner (1977) and Phan-Thien (1978)

$$\left(1 + \varepsilon \frac{\mathrm{ReWi}}{(1-\beta)} tr(\mathbf{T})\right)\mathbf{T} + W_i \overset{\nabla}{\mathbf{T}} = 2\frac{(1-\beta)}{\mathrm{Re}}\mathbf{D} \tag{3}$$

where Wi represents the Weissenberg number and $\overset{\nabla}{\mathbf{T}}$ denotes the upper-convected derivative of $\mathbf{T}$, which is defined as follows:

$$\overset{\nabla}{\mathbf{T}} = \frac{D\mathbf{T}}{Dt} - \mathbf{T}(\nabla \mathbf{u} - \xi \mathbf{D}) - (\nabla \mathbf{u} - \xi \mathbf{D})^T \cdot \mathbf{T}$$

In the two-dimensional case, equations (1), (2), and (3) can be provided as follows:

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

$$\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\beta}{\mathrm{Re}}\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + \frac{\partial T^{xx}}{\partial x} + \frac{\partial T^{xy}}{\partial y}, \tag{5}$$

$$\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (vv)}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\beta}{\mathrm{Re}}\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right] + \frac{\partial T^{xy}}{\partial x} + \frac{\partial T^{yy}}{\partial y}, \tag{6}$$

$$\begin{aligned}
&\left(1 + \varepsilon \frac{\mathrm{ReWi}}{(1-\beta)} tr(\mathbf{T})\right) T^{xx} + Wi\left(\frac{\partial T^{xx}}{\partial t} + \frac{\partial (uT^{xx})}{\partial x} + \frac{\partial (vT^{xx})}{\partial y} - 2T^{xx}\frac{\partial u}{\partial x} - \right. \\
&\left. -2T^{xy}\frac{\partial u}{\partial y} + \xi\left[2T^{xx}\frac{\partial u}{\partial x} + T^{xy}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right]\right) = 2\frac{(1-\beta)}{\mathrm{Re}}\frac{\partial u}{\partial x},
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\left(1 + \varepsilon \frac{\mathrm{ReWi}}{(1-\beta)} tr(\mathbf{T})\right) T^{xy} + Wi\left(\frac{\partial T^{xy}}{\partial t} + \frac{\partial (uT^{xy})}{\partial x} + \frac{\partial (vT^{xy})}{\partial y} - T^{xx}\frac{\partial v}{\partial x} - \right. \\
&\left. -T^{yy}\frac{\partial u}{\partial y} + \xi\left[T^{xy}\frac{\partial u}{\partial x}\frac{1}{2}T^{yy}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + T^{xy}\frac{\partial v}{\partial y}\right]\right) = \frac{(1-\beta)}{\mathrm{Re}}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right),
\end{aligned} \tag{8}$$

$$\begin{aligned}
&\left(1 + \varepsilon \frac{\mathrm{ReWi}}{(1-\beta)} tr(\mathbf{T})\right) T^{yy} + Wi\left(\frac{\partial T^{yy}}{\partial t} + \frac{\partial (uT^{yy})}{\partial x} + \frac{\partial (vT^{yy})}{\partial y} - 2T^{xy}\frac{\partial v}{\partial x} - \right. \\
&\left. -2T^{yy}\frac{\partial v}{\partial y} + \xi\left[T^{xy}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} + 2T^{yy}\frac{\partial v}{\partial y}\right)\right]\right) = 2\frac{(1-\beta)}{\mathrm{Re}}\frac{\partial v}{\partial y},
\end{aligned} \tag{9}$$

for all $(x, y) \in \Omega$.

2.1 Stream Function-Vorticity Formulation

In order to obtain an accurate numerical solution for the equations, a Stream Function-Vorticity Formulation was chosen

$$\omega_z = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \tag{10}$$

Differentiating equation (5) with respect to y and subtracting it from equation (6) differentiated with respect to x, yields

$$\frac{\partial \omega_z}{\partial t} + \frac{\partial (u\omega_z)}{\partial x} + \frac{\partial (v\omega_z)}{\partial y} = \frac{\beta}{Re}\left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2}\right) + \frac{\partial^2 T_{xx}}{\partial x \partial y} + \frac{\partial^2 T_{xy}}{\partial y^2} - \frac{\partial^2 T_{xy}}{\partial x^2} - \frac{\partial^2 T_{yy}}{\partial x \partial y} \tag{11}$$

Considering the definition of vorticity and the continuity equation, we can derive a Poisson equation for the stream function Ψ:

$$\frac{\partial^2\Psi}{\partial x^2}+\frac{\partial^2\Psi}{\partial y^2}=\omega_z, \tag{12}$$

which is obtained by using the definitions:

$$u=\frac{\partial\Psi}{\partial y}\qquad v=-\frac{\partial\Psi}{\partial x}$$

Thus, equations (4), (5), and (6) of the system of equations are replaced by equations (11) and (12).

3. METHOD OF MANUFACTURED SOLUTION - MMS

The Method of Manufactured Solutions requires the provision of certain solutions, from which the remaining functions can be calculated. In this context, the given functions are denoted by $\overline{f}$, while the functions calculated from the given functions and the corresponding equations are represented by $\widetilde{f}$. This notation serves the purpose of distinguishing between the provided original functions and the derived functions obtained through mathematical operations. To illustrate the application of this notation, we consider the functions $\overline{u}$, $\overline{T}xx$, $\overline{T}xy$, and $\overline{T}_{yy}$ and subsequently calculate the corresponding derived functions using the given functions and the relevant equations.

$$\widetilde{v}=-\int\frac{\partial\overline{u}}{\partial x}dy,$$

and

$$\widetilde{\omega_z}=\frac{\partial\overline{u}}{\partial y}-\frac{\partial\widetilde{v}}{\partial x}.$$

Since the utilized solutions are not exact solutions of the equations for the presented models, it is necessary to incorporate source terms into equations (7), (8), (9), and (11), given by $\mathbf{tfT^{xx}}$, $\mathbf{tfT^{xy}}$, $\mathbf{tfT^{yy}}$ and $\mathbf{tf\omega_z}$, respectively. These additional terms account for the discrepancy between the exact solutions and the solutions employed in our analysis. In this way, the system of equations formed by equations (7), (8), (9), (11) and (12) becomes

$$\begin{aligned}
&\frac{\partial^2\Psi}{\partial x^2}+\frac{\partial^2\Psi}{\partial y^2}=\omega_z, \\
&\frac{\partial\omega_z}{\partial t}+\frac{\partial\left(u\omega_z\right)}{\partial x}+\frac{\partial\left(v\omega_z\right)}{\partial y}=\frac{\beta}{Re}\left(\frac{\partial^2\omega_z}{\partial x^2}+\frac{\partial^2\omega_z}{\partial y^2}\right)+\frac{\partial^2T_{xx}}{\partial x\partial y}+\frac{\partial^2T_{xy}}{\partial y^2}-\frac{\partial^2T_{xy}}{\partial x^2}-\frac{\partial^2T_{yy}}{\partial x\partial y}+\mathbf{tf\omega_z}, \\
&\left(1+\varepsilon\frac{\mathrm{ReWi}}{(1-\beta)}tr(\mathbf{T})\right)T^{xx}+Wi\left(\frac{\partial T^{xx}}{\partial t}+\frac{\partial\left(uT^{xx}\right)}{\partial x}+\frac{\partial\left(vT^{xx}\right)}{\partial y}-2T^{xx}(1-\xi)\frac{\partial u}{\partial x}-\right. \\
&\left.-2T^{xy}\left(1-\frac{1}{2}\xi\right)\frac{\partial u}{\partial y}+T^{xy}\xi\frac{\partial v}{\partial x}\right)=2\frac{(1-\beta)}{\mathrm{Re}}\frac{\partial u}{\partial x}+\mathbf{tfT^{xx}}, \\
&\left(1+\varepsilon\frac{\mathrm{ReWi}}{(1-\beta)}tr(\mathbf{T})\right)T^{xy}+Wi\left(\frac{\partial T^{xy}}{\partial t}+\frac{\partial\left(uT^{xy}\right)}{\partial x}+\frac{\partial\left(vT^{xy}\right)}{\partial y}-T^{xx}\left(1-\frac{1}{2}\xi\right)\frac{\partial v}{\partial x}+\right. \\
&\left.+T^{xx}\frac{\xi}{2}\frac{\partial u}{\partial y}-T^{yy}\left(1-\frac{1}{2}\xi\right)\frac{\partial u}{\partial y}+T^{yy}\frac{\xi}{2}\frac{\partial v}{\partial x}\right)=\frac{(1-\beta)}{\mathrm{Re}}\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)+\mathbf{tfT^{xy}}, \\
&\left(1+\varepsilon\frac{\mathrm{ReWi}}{(1-\beta)}tr(\mathbf{T})\right)T^{yy}+Wi\left(\frac{\partial T^{yy}}{\partial t}+\frac{\partial\left(uT^{yy}\right)}{\partial x}+\frac{\partial\left(vT^{yy}\right)}{\partial y}-2T^{yy}(1-\xi)\frac{\partial v}{\partial y}-\right. \\
&\left.-2T^{xy}\left(1-\frac{1}{2}\xi\right)\frac{\partial v}{\partial x}+T^{xy}\xi\frac{\partial u}{\partial y}\right)=2\frac{(1-\beta)}{\mathrm{Re}}\frac{\partial v}{\partial y}+\mathbf{tfT^{yy}}.
\end{aligned} \tag{13}$$

where the source terms are

$$\mathbf{tf\omega_z}=\frac{\partial\widetilde{\omega_z}}{\partial t}+\frac{\partial(\overline{u}\widetilde{\omega_z})}{\partial x}+\frac{\partial(\widetilde{v}\widetilde{\omega_z})}{\partial y}-\frac{\beta}{Re}\left[\frac{\partial^2\widetilde{\omega_z}}{\partial x^2}+\frac{\partial^2\widetilde{\omega_z}}{\partial y^2}\right]-\frac{\partial^2\overline{T^{xx}}}{\partial x\partial y}-\frac{\partial^2\overline{T^{xy}}}{\partial y^2}+\frac{\partial^2\overline{T^{xy}}}{\partial x^2}+\frac{\partial^2\overline{T^{yy}}}{\partial x\partial y}$$

$$\begin{aligned}
\mathbf{tfT^{xx}}=&\left(1+\varepsilon\frac{\mathrm{ReWi}}{(1-\beta)}tr(\overline{\mathbf{T}})\right)\overline{T}^{xx}+Wi\left(\frac{\partial\overline{T^{xx}}}{\partial t}+\frac{\partial\left(\overline{u}\overline{T^{xx}}\right)}{\partial x}+\frac{\partial\left(\widetilde{v}\overline{T^{xx}}\right)}{\partial y}-2\overline{T^{xx}}(1-\xi)\frac{\partial\overline{u}}{\partial x}-\right. \\
&\left.-2\overline{T^{xy}}\left(1-\frac{1}{2}\xi\right)\frac{\partial\overline{u}}{\partial y}+\overline{T^{xy}}\xi\frac{\partial\widetilde{v}}{\partial x}\right)-2\frac{(1-\beta)}{\mathrm{Re}}\frac{\partial\overline{u}}{\partial x},
\end{aligned}$$

$$\mathbf{tfT^{xy}} = \left(1+\varepsilon\frac{\mathrm{ReWi}}{(1-\beta)}tr(\overline{\mathbf{T}})\right)\overline{T^{xy}} + Wi\left(\frac{\partial\overline{T^{xy}}}{\partial t} + \frac{\partial\left(\overline{u}\overline{T^{xy}}\right)}{\partial x} + \frac{\partial\left(\widetilde{v}\overline{T^{xy}}\right)}{\partial y} - T^{xx}\left(1-\frac{1}{2}\xi\right)\frac{\partial\widetilde{v}}{\partial x} + \right.$$
$$\left. + \overline{T^{xx}}\frac{\xi}{2}\frac{\partial\overline{u}}{\partial y} - \overline{T^{yy}}\left(1-\frac{1}{2}\xi\right)\frac{\partial\overline{u}}{\partial y} + \overline{T^{yy}}\frac{\xi}{2}\frac{\partial\widetilde{v}}{\partial x}\right) - \frac{(1-\beta)}{\mathrm{Re}}\left(\frac{\partial\widetilde{v}}{\partial x} - \frac{\partial\overline{u}}{\partial y}\right)$$

$$\mathbf{tfT^{yy}} = \left(1+\varepsilon\frac{\mathrm{ReWi}}{(1-\beta)}tr(\overline{\mathbf{T}})\right)\overline{T^{yy}} + Wi\left(\frac{\partial\overline{T^{yy}}}{\partial t} + \frac{\partial\left(\overline{u}\overline{T^{yy}}\right)}{\partial x} + \frac{\partial\left(\widetilde{v}\overline{T^{yy}}\right)}{\partial y} - 2\overline{T^{yy}}(1-\xi)\frac{\partial\widetilde{v}}{\partial y} - \right.$$
$$\left. - 2\overline{T^{xy}}\left(1-\frac{1}{2}\xi\right)\frac{\partial\widetilde{v}}{\partial x} + \overline{T^{xy}}\xi\frac{\partial\overline{u}}{\partial y}\right) - 2\frac{(1-\beta)}{\mathrm{Re}}\frac{\partial\widetilde{v}}{\partial y}$$

An essential aspect in solving the problem is carefully handling the boundary conditions, particularly due to the utilization of the vorticity-stream function formulation. In this context, the judicious adaptation of boundary conditions plays a crucial role in ensuring the consistency of the employed methods while preserving the existence of viable solutions. Thus, relying on the exact solutions obtained through the Method of Manufactured Solutions (MMS), we impose $u = \overline{u},\ v = \widetilde{v},\ \omega_z = \widetilde{\omega_z},\ \psi = \widetilde{\psi}, T^{xx} = \overline{T^{xx}},\ T^{xy} = \overline{T^{xy}}$, and $T^{yy} = \overline{T^{yy}}$ on $\partial\Omega$.

This approach aims to maintain coherence and integrity in the adopted solution process, preventing any undesirable or inappropriate deviations that could compromise the accuracy of the results. By enforcing the boundary conditions using the exact solutions obtained through MMS, we can obtain reliable and consistent responses, closely approximating the correct solutions for the analyzed problem. Moreover, employing this strategy of adapting the boundary conditions provides a robust foundation for conducting tests in cases where the values of ω_z and ψ at the boundary are not known a priori. This flexibility is particularly relevant in real-world problems, where obtaining precise information about the boundary conditions is not always feasible.

4. NUMERICAL METHODS AND CODE

In this study, we have employed a fourth-order Runge-Kutta scheme for time integration in fluid dynamics simulations, as cited in (Ferziger *et al.*, 2002). This numerical method is renowned for its accuracy and stability in solving time-dependent problems. To ensure accurate representations of spatial variations, we utilize fourth-order compact finite difference schemes, as described in (Souza, 2003), to discretize the spatial derivatives in our computational framework. These compact schemes offer high-order accuracy while requiring fewer grid points than traditional finite difference schemes.

For the resolution of the Poisson equation (13), we implement the Multigrid method of the Full Approximation Scheme (FAS) type. The Multigrid method is a powerful iterative technique that efficiently solves elliptic partial differential equations by recursively solving the problem on coarser grids and transferring the error back to finer grids. The Full Approximation Scheme (FAS) type of Multigrid is known for its robustness and effectiveness in handling complex and large-scale problems. Detailed discussions on this method can be found in the references (Rogenski, 2011), (Souza, 2003), and Brandt (1977).

In addition to these advanced techniques, we further enhance the overall numerical stability and precision of our computations by incorporating a fourth-order spatial filtering technique, inspired by the method introduced by Lele (1992). This filtering approach effectively suppresses high-frequency numerical oscillations, commonly associated with high-order numerical schemes. By selectively damping these oscillations, we achieve smoother and more accurate numerical solutions, particularly in regions of strong gradients or steep variations.

The combined utilization of the fourth-order Runge-Kutta scheme, fourth-order compact finite difference schemes, Multigrid FAS, and spatial filtering technique creates a robust and accurate computational framework for tackling complex fluid dynamics problems. By employing state-of-the-art numerical methods and techniques, we aim to ensure the reliability and validity of our simulations while providing valuable insights into the behavior of fluid flow involving non-Newtonian fluids modeled by the LPTT approach.

For the determination of essential source terms in the context of the MMS, we utilized the Mathematica software, developed by Wolfram Research, which is a powerful tool for computational algebra and symbolic computing. As for the implementation, we chose the Fortran compiler, an integral part of the GNU Compiler Collection (GCC). Fortran is widely recognized for its efficiency in processing numerical calculations, making it a suitable choice for the scope of this study. The code follows the following structure:

function MAIN(par.nn)
  Initial and boundary conditions for φ;
  Set the value of Er_∞; ▷ Maximum error to achieve steady-state
  while $t \leq tf$ **do**
    for i **do** 1 to 4
      Calculate the i-th step of Runge-Kutta for φ ;
      Calculate Ψ by solving the Poisson equation using the Multigrid method;
      Perform the Spatial Filtering as described for φ;
      Calculate the values of u and v using Ψ;
    end for
    Calculate $Er_\varphi = \frac{||\varphi^{n+1}-\varphi^n||_\infty}{||\varphi^{n+1}||_\infty}$; ▷ Relative error
    if $Er_\varphi \geq Er_\infty$ and $t \geq 0.01$ **then**
      Break
    end if
    $t \leftarrow t + dt$; ▷ Time step or time increment
  end while
  Calculate the errors between the numerical and analytical solutions;
end function

The variable φ simultaneously represents the variables ω_z, T_{xx}, T_{xy}, and T_{yy}. By employing the Method of Manufactured Solutions, it is possible to generate a known reference solution for comparison with the numerical solution obtained from the computational code. The discrete standards used are given by:

$$\|e_k\|_\infty = \max_{\substack{1\leq i\leq N_1 \\ 1\leq j\leq N_2}} |e_k|_{i,j} \tag{14}$$

and

$$\|e_k\|_q = \left(\Delta x \Delta y \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} |e_k|_{i,j}^q\right)^{1/q} \tag{15}$$

5. RESULTS

For the purpose of verification, the code will be executed with manufactured solutions provided for u, v, ω_z, ψ, T^{xx}, T^{xy}, and T^{yy}. The numerical solutions obtained by the code will be compared to these exact analytical solutions when the established convergence criterion is met, utilizing the function norms defined in equations (14) and (15) for error calculation. The convergence rates of the numerical solutions will be analyzed for different mesh sizes, and the error norms between the numerical and analytical solutions will be computed to quantify the code's accuracy. The order p will be determined using the expression (16) introduced by LeVeque (2007):

$$p_k \approx \log_2\left(\frac{e_k}{e_{k+1}}\right), \tag{16}$$

where e_1 represents the coarser mesh and e_2 the finer mesh.

To test the code, we consider the functions u and v given by:

$$\begin{aligned} \overline{u} &= 8e^{-a\cdot t}\cos(\pi x)\sin(\pi y) \\ \widetilde{v} &= -8e^{-a\cdot t}\sin(\pi x)\cos(\pi y). \end{aligned}$$

Consequently, we obtain:

$$\begin{aligned} \widetilde{\omega_z} &= 16\pi e^{-a\cdot t}\cos(\pi x)\cos(\pi y) \\ \widetilde{\psi} &= -\frac{8}{\pi}e^{-a\cdot t}\cos(\pi x)\cos(\pi y) \end{aligned}$$

Finally, the tensors are given by:

$$\begin{aligned} \overline{T^{xx}} &= (1-\beta)e^{-a\cdot t}\left(1+x+x^2\right)\left(1+y+y^2\right) \\ \overline{T^{xy}} &= (1-\beta)e^{-a\cdot t}\left(1+x+x^2\right)\left(1+y+y^2\right) \\ \overline{T^{yy}} &= (1-\beta)e^{-a\cdot t}\left(1+x+x^2\right)\left(1+y+y^2\right) \end{aligned}$$

The source terms $\mathbf{tf\omega_z}$, $\mathbf{tfT^{xx}}$, $\mathbf{tfT^{xy}}$, and $\mathbf{tfT^{yy}}$ are given by (17), (18), (19) and (21):

$$\mathbf{tf\omega_z} = (1-\beta)e^{-a\cdot t}\left[2\left(1+x+x^2\right)+(1+2x)(1+2y)-(1+2x)(1+2y)+2\left(1+y+y^2\right)\right]$$
$$+16\pi\frac{e^{-a\cdot t}\cos(\pi x)\cos(\pi y)}{Re}\left[ae^{-a\cdot t}-2\pi^2\beta\right] \quad (17)$$

$$\mathbf{tfT^{xx}} = e^{-a.t}\left[(1-\beta)(1+x+x^2)(1+y+y^2)\left(1+\left(\epsilon\,\mathrm{ReWi}(1-\beta)\left(1+x+x^2\right)\left(1+y+y^2\right)\right]\right.$$
$$+e^{-a.t}\left[(1-\beta)^2\left(1+x+x^2\right)\left(1+y+y^2\right)e^{-a.t}+\frac{\pi}{\mathrm{Re}}16(1-\beta)\sin(\pi x)\sin(\pi y)\right]$$
$$-(1-\beta)e^{-a.t}\left[a\cdot Wi\left(1+x+x^2\right)\left(1+y+y^2\right)+\right.$$
$$16\pi e^{-a.t}\left(1+x+x^2\right)\left(1+y+y^2\right)\cos(\pi x)\cos(\pi y)\Big]$$
$$-8e^{-2a.t}(1-\beta)\left[\left(1+x+x^2\right)(1+2y)\cos(\pi y)\sin(\pi x)\right)+(1+2x)\left(1+y+y^2\right)\cos(\pi x)\sin(\pi y)\Big)\Big]$$
$$+16\pi e^{-2a.t}\sin(\pi x)\sin(\pi y)(1-\beta)\left[\left(1+x+x^2\right)\left(1+y+y^2\right)-\xi\left(1+x+x^2\right)\left(1+y+y^2\right)\right], \quad (18)$$

$$\mathbf{tfT^{xy}} = (1-\beta)e^{-2a.t}\left[\left(1+x+x^2\right)\left(1+x+x^2\right)\left(1+y+y^2\right)\left(1+y+y^2\right)\right]+$$
$$(1-\beta)e^{-a.t}\left[\left(1+x+x^2\right)\left(1+y+y^2\right)\left(1+\epsilon\,\mathrm{ReWi}(1-\beta)\left(1+x+x^2\right)\left(1+y+y^2\right)\right]\right.$$
$$+e^{-a.t}\left[\left(1+x+x^2\right)\left(1+y+y^2\right)-a\cdot Wi(1-\beta)\left(1+x+x^2\right)\left(1+y+y^2\right)\right]$$
$$+8\pi(1-\beta)e^{-2a.t}\cos(\pi x)\cos(\pi y)\left[\left(\left(1+x+x^2\right)\left(1+y+y^2\right)\right)-\left(\left(1+x+x^2\right)\left(1+y+y^2\right)\right]\right.$$
$$8(1-\beta)e^{-2a.t}\left[-\left(1+x+x^2\right)(1+2y)\cos(\pi y)\sin(\pi x)+(1+2x)\left(1+y+y^2\right)\cos(\pi x)\sin(\pi y)\right], \quad (19)$$

and

$$\mathbf{tfT^{yy}} = (1-\beta)e^{-2a\cdot t}\left(1+x+x^2\right)^2\left(1+y+y^2\right)^2$$
$$+(1-\beta)e^{-a.t}\left(1+x+x^2\right)\left(1+y+y^2\right)\left(1+\left(\epsilon\,\mathrm{ReWi}(1-\beta)\left(1+x+x^2\right)\left(1+y+y^2\right)\right)\right.$$
$$\left.+e^{-2a.t}\left(1+x+x^2\right)\left(1+y+y^2\right)-\frac{16\pi e^{-a.t}}{\mathrm{Re}}(1-\beta)\sin(\pi x)\sin(\pi y)\right)+$$
$$Wie^{-a.t}\left[(1-\beta)(-a\cdot\left(1+x+x^2\right)\left(1+y+y^2\right)+\right. \quad (20)$$
$$16\pi e^{-a.t}\left(1+x+x^2\right)\left(1+y+y^2\right)\cos(\pi x)\cos(\pi y))\Big]$$
$$-8(1-\beta)e^{-2a.t}\left[\left(\left(1+x+x^2\right)(1+2y)\cos(\pi y)\sin(\pi x)+(1+2x)\left(1+y+y^2\right)\cos(\pi x)\sin(\pi y)\right]\right.$$
$$-16\pi(1-\beta)e^{-2a.t}\sin(\pi x)\sin(\pi y)\left[\left(1+x+x^2\right)\left(1+y+y^2\right)+\left(1+x+x^2\right)\xi\left(1+y+y^2\right)\right] \quad (21)$$

These manufactured solutions will serve as the benchmark for validating the code's numerical accuracy, ensuring its reliability for studying fluid dynamics problems involving non-Newtonian fluids modeled by the LPTT approach. By rigorously verifying the code against these exact solutions under various mesh resolutions and boundary conditions, we aim to establish the code's robustness and suitability for accurate numerical simulations in this complex and important field of research.

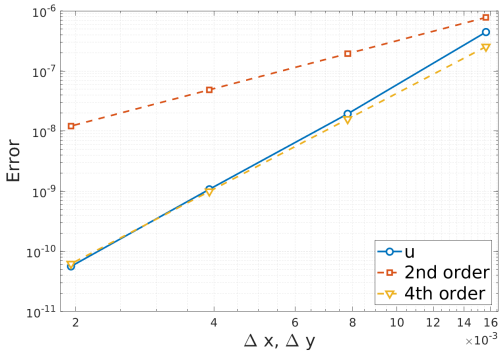
For the initial test, we set the parameter a to zero, leading to steady-state solutions that remain unchanged with time t. The chosen parameters for this case are as follows: $Re = 1000$, $\beta = 0.5$, $\xi = 0.25$, $\epsilon = 0.5$, and $Wi = 1.0$.

The results of the tests are summarized in Table 1. By analyzing the calculated values using equation (16), it was found that the order varies between 3.3805 and 4.5808, representing, respectively, the minimum and maximum values obtained among all the results. These findings consistently validate the successful application of the Method of Manufactured Solutions (MMS), demonstrating the high precision and reliability of the results in this study. Additionally, it is worth noting that the relatively narrow range of variation suggests result consistency, which is a positive indicator of the quality of the simulation method adopted.

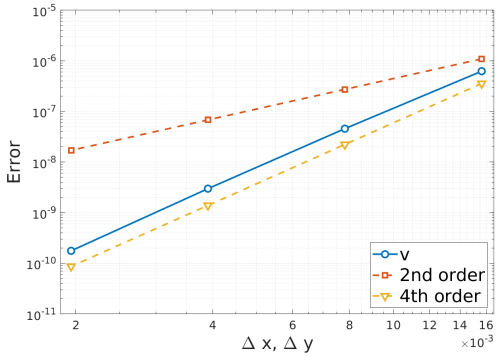
Figures 1 and 2 graphically present the results from Table 1 with the $||\cdot||_2$ norm on the log-log scale in Matlab. In all the figures, the red dashed line with squares represents a second-order slope, the yellow dashed line with triangles represents a fourth-order slope, and the blue solid line with circles represents the numerical error throughout mesh refinement. In these figures, it is noticeable that the numerical errors closely follow the fourth-order slope. The alignment between the numerical error and the fourth-order line is evident, indicating a strong correlation and confirming the accuracy and reliability of the numerical method employed. These graphical representations provide valuable visual insights into the convergence behavior of the numerical method with mesh refinement. The proximity between the numerical errors and the fourth-order line further reinforces the successful application of the Method of Manufactured Solutions (MMS) in validating the precision of the numerical model.

Table 1. Numerical error in different norms and order testing.

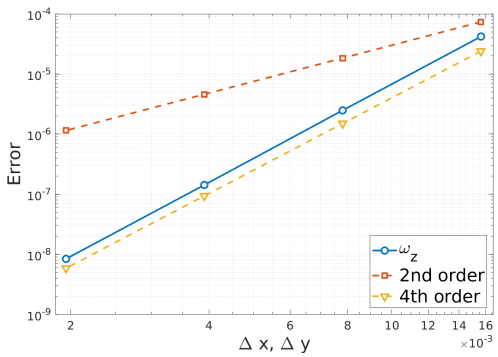
	Grid	fourth-order					
		$\|\|\cdot\|\|_\infty$		$\|\|\cdot\|\|_1$		$\|\|\cdot\|\|_2$	
		Error	Order	Error	Order	Error	Order
ω_z	65	$2.3206e^{-04}$	—	$1.6668e^{-05}$	—	$4.1962e^{-05}$	—
	129	$1.7033e^{-05}$	3.7681	$8.1963e^{-07}$	4.3459	$2.4802e^{-06}$	4.0806
	257	$1.0225e^{-06}$	4.0582	$4.2903e^{-08}$	4.2558	$1.4243e^{-07}$	4.1221
	513	$5.6219e^{-08}$	4.1849	$2.4394e^{-09}$	4.1365	$8.3867e^{-09}$	4.0860
Ψ	65	$1.3481e^{-07}$	—	$2.1447e^{-08}$	—	$3.5438e^{-08}$	—
	129	$9.6157e^{-09}$	3.8094	$1.6347e^{-09}$	3.7137	$2.6974e^{-09}$	3.7157
	257	$5.9179e^{-10}$	4.0222	$1.0178e^{-10}$	4.0055	$1.6654e^{-10}$	4.0176
	513	$3.2878e^{-11}$	4.1699	$4.2531e^{-12}$	4.5808	$8.1109e^{-12}$	4.3598
T^{xx}	65	$1.0007e^{-07}$	—	$7.3175e^{-09}$	—	$1.5914e^{-08}$	—
	129	$5.9939e^{-09}$	4.0613	$3.4034e^{-10}$	4.4263	$8.4219e^{-10}$	4.2400
	257	$3.8795e^{-10}$	3.9495	$1.7904e^{-11}$	4.2486	$14.6198e^{-11}$	4.1882
	513	$2.3190e^{-11}$	4.0643	$1.0770e^{-12}$	4.0552	$2.7150e^{-12}$	4.0888
T^{xy}	65	$2.6726e^{-08}$	—	$2.7691e^{-09}$	—	$5.8192e^{-09}$	—
	129	$2.5662e^{-09}$	3.3805	$1.4486e^{-10}$	4.2567	$3.6794e^{-10}$	3.9833
	257	$1.8373e^{-10}$	3.8040	$8.4130e^{-12}$	4.1059	$2.2851e^{-11}$	4.0091
	513	$1.6254e^{-11}$	3.4987	$5.2447e^{-13}$	4.0037	$1.4844e^{-12}$	3.9443
T^{yy}	65	$1.1106e^{-07}$	—	$8.8105e^{-09}$	—	$2.0281e^{-08}$	—
	129	$1.0171e^{-08}$	3.4488	$4.8370e^{-10}$	4.1870	$1.3411e^{-09}$	3.9187
	257	$7.2190e^{-10}$	3.8165	$2.9281e^{-11}$	4.0461	$8.4797e^{-11}$	3.9832
	513	$5.1728e^{-11}$	3.8028	$1.9048e^{-12}$	3.9423	$5.4411e^{-12}$	3.9620



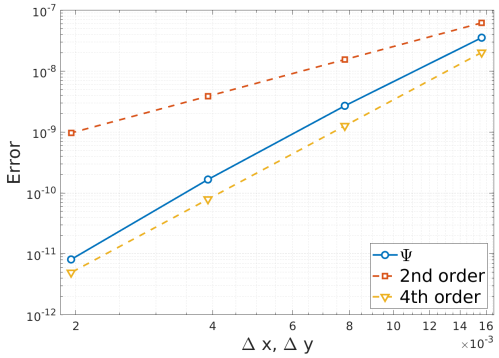
(a) u.



(b) v.

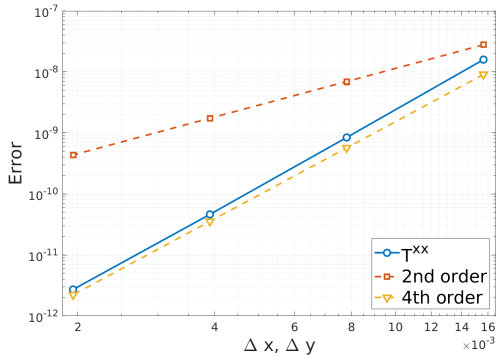


(c) ω_z.

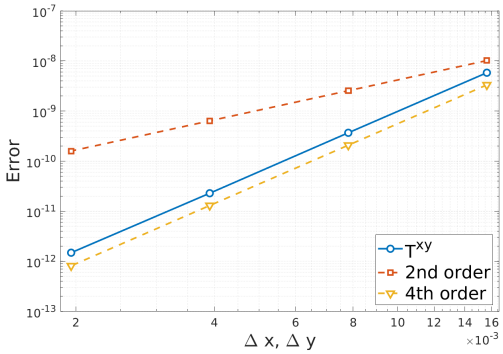


(d) ψ.

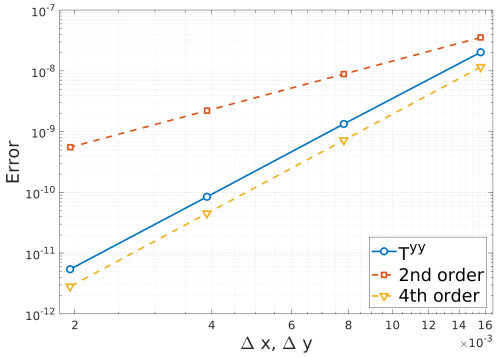
Figure 1. Errors of u, v, ω_z and ψ with the norm: $||\cdot||_2$: $\xi = 0.25$, $\epsilon = 0.5$, $Re = 1000$, $\beta = 0.5$ and $Wi = 1$.



(a) T^{xx}.



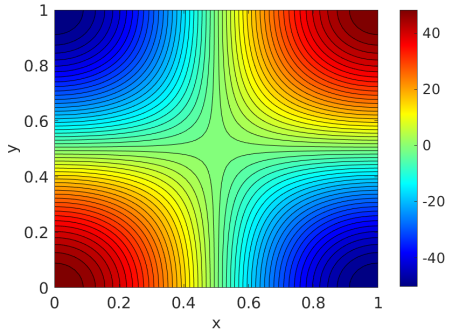
(b) T^{xy}.



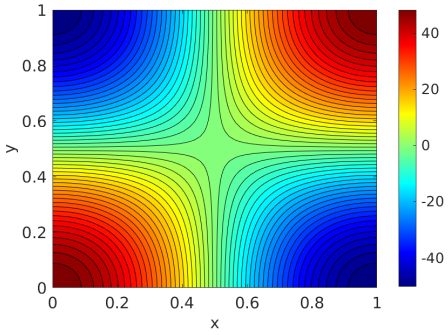
(c) T^{yy}.

Figure 2. Errors of the tensors with the norm $||\cdot||_2$: $\xi = 0.25$, $\epsilon = 0.5$, $Re = 1000$, $\beta = 0.5$ and $Wi = 1$.

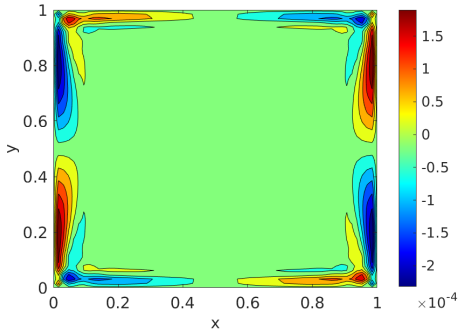
Figures 3(a), 4(a), and 5(a) display the map of exact solutions, while Figures 3(b), 4(b), and 5(b) illustrate the numerical solutions of ω_z, ψ, and T^{xy}, respectively.



(a) ω_z: Exact soluction.

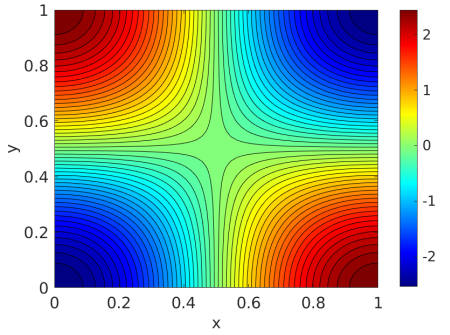


(b) ω_z: Numerical soluction.

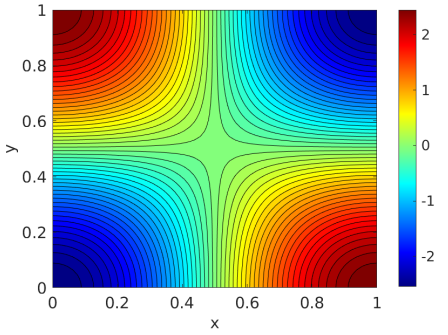


(c) ω_z: Difference between numerical and exact.

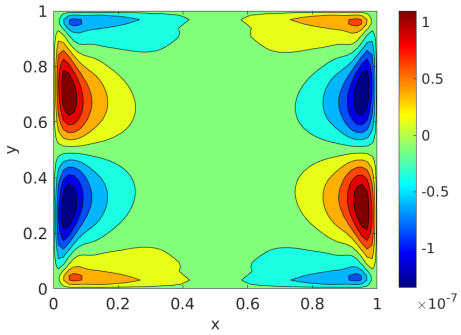
Figure 3. Numerical and exact solutions ω_z: $\xi = 0.25$, $\epsilon = 0.5$, $Re = 1000$, $\beta = 0.5$ and $Wi = 1$.



(a) ψ: Exact soluction.

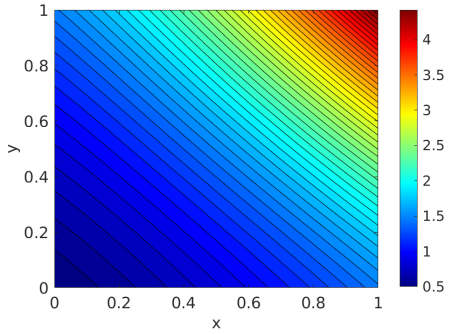


(b) ψ: Numerical soluction.

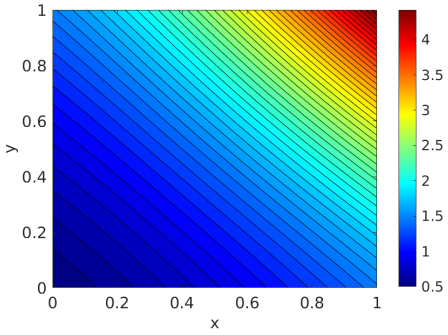


(c) ψ: Difference between numerical and exact.

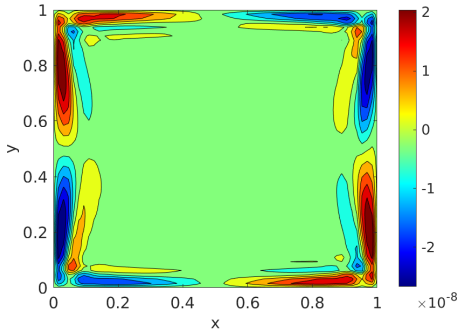
Figure 4. Numerical and exact solutions ψ: $\xi = 0.25$, $\epsilon = 0.5$, $Re = 1000$, $\beta = 0.5$ and $Wi = 1$.



(a) T^{xy}: Exact soluction.



(b) T^{xy}: Numerical soluction.



(c) T^{xy}: Difference between numerical and exact.

Figure 5. Numerical and exact solutions T^{xy}: $\xi = 0.25$, $\epsilon = 0.5$, $Re = 1000$, $\beta = 0.5$ and $Wi = 1$.

Additionally, Figures 3(c), 4(c), and 5(c) illustrate the difference between the numerical and exact solutions of ω_z, ψ, and T^{xy}, respectively. The higher errors observed in the regions near the boundary conditions, while not significantly affecting the overall order of accuracy in the test, indicate the need for improvements in the code to enhance accuracy and minimize errors in these critical areas. These findings provide valuable guidance for refining the numerical model and ensuring its robustness and precision in real-world applications involving non-Newtonian fluid dynamics modeled by the LPTT approach.

6. Tabelas

TABLE I Results for u-velocity along Vertical Line through Geometric Center of Cavity

129−		Re						
grid pt. no.	y	100	400	1000	3200	5000	7500	10,000
129	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
126	0.9766	0.84123	0.75837	0.65928	0.53236	0.48223	0.47244	0.47221
125	0.9688	0.78871	0.68439	0.57492	0.48296	0.46120	0.47048	0.47783
124	0.9609	0.73722	0.61756	0.51117	0.46547	0.45992	0.47323	0.48070
123	0.9531	0.68717	0.55892	0.46604	0.46101	0.46036	0.47167	0.47804
110	0.8516	0.23151	0.29093	0.33304	0.34682	0.33556	0.34228	0.34635
95	0.7344	0.00332	0.16256	0.18719	0.19791	0.20087	0.20591	0.20673
80	0.6172	-0.13641	0.02135	0.05702	0.07156	0.08183	0.08342	0.08344
65	0.5000	-0.20581	-0.11477	-0.06080	-0.04272	-0.03039	-0.03800	0.03111
59	0.4531	-0.21090	-0.17119	-0.10648	-0.86636	-0.07404	-0.07503	-0.07540
37	0.2813	-0.15662	-0.32726	-0.27805	-0.24427	-0.22855	-0.23176	-0.23186
23	0.1719	-0.10150	-0.24299	-0.38289	-0.34323	-0.33050	-0.32393	-0.32709
14	0.1016	-0.06434	-0.14612	-0.29730	-0.41933	-0.40435	-0.38324	-0.38000
10	0.0703	-0.04775	-0.10338	-0.22220	-0.37827	-0.43643	-0.43025	-0.41657
9	0.0625	-0.04192	-0.09266	-0.20196	-0.35344	-0.42901	-0.43590	-0.42537
8	0.0547	-0.03717	-0.08186	-0.18109	-0.32407	-0.41165	-0.43154	-0.42735
1	0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

TABLE II Results for v-Velocity along Horizontal Line through Geometric Center of Cavity

129−		Re						
grid pt. no.	x	100	400	1000	3200	5000	7500	10,000
129	1.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
125	0.9688	-0.05906	-0.12146	-0.21388	-0.39017	-0.49774	-0.53858	-0.54302
124	0.9609	-0.07391	-0.15663	-0.27669	-0.47425	-0.55069	-0.55216	-0.52987
123	0.9531	-0.08864	-0.19254	-0.33714	-0.52357	-0.55408	-0.52347	-0.49099
122	0.9453	-0.10313	-0.22847	-0.39188	-0.54053	$\overline{-0.52876}$	-0.48590	-0.45863
117	0.9063	-0.16914	-0.23827	-0.51550	$\overline{-0.44307}$	-0.41442	-0.41050	-0.41496
111	0.8594	-0.22445	-0.44993	-0.42665	-0.37401	-0.36214	-0.36213	-0.36737
104	0.8047	-0.24533	-0.38598	-0.31966	-0.31184	-0.30018	-0.30448	-0.30719
65	0.5000	0.05454	0.05186°	0.02526	0.00999	0.00945	0.00824	0.00831
31	0.2344	$\underline{0.17527}$	0.30174	0.32235	0.28188	0.27280	0.27348	0.27224
30	0.2266	$\overline{0.17507}$	0.30203	0.33075	0.29030	0.28066	0.28117	0.28003
21	0.1563	0.16077	0.28124	0.37095	0.37119	0.35368	0.35060	0.35070
13	0.0938	0.12317	0.22965	0.32627	0.42768	0.42951	0.41824	0.41487
11	0.0781	0.10890	0.20920	0.30353	0.41906	0.43648	0.43564	0.43124
10	0.0703	0.10091	0.19713	0.29012	0.40917	0.43329	$\underline{0.44030}$	0.43733
9	0.0625	0.09233	0.18360	0.27485	0.39560	0.42447	0.43979	0.43983
1	0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

7. CONCLUSIONS

In conclusion, the development and testing of the numerical code for simulating non-Newtonian fluid flows modeled by the LPTT method have been successful. Verification tests have revealed that the code achieves an order of accuracy between 3.8 and 4.5, showcasing its capability to accurately capture spatial and temporal variations in the studied flows.

The utilization of the Method of Manufactured Solutions (MMS) for code verification provided robust validation, ensuring that the numerical solutions are in agreement with exact analytical solutions. The next step, validation, will involve comparing the code with experimental data or results from known simulations in more complex scenarios, allowing for an assessment of its performance in real-world situations.

8. ACKNOWLEDGEMENTS

The authors thank the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) for the research grant received during the development of this study. The research was carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI), funded by FAPESP (grant 2013/07375-0).

9. REFERENCES

Brandt, A., 1977. "Multi-level adaptive solutions to boundary-value problems". *Mathematics of computation*, Vol. 31, No. 138, pp. 333–390.

Ferziger, J.H., Perić, M. and Street, R.L., 2002. *Computational methods for fluid dynamics*, Vol. 3. Springer.

Lele, S.K., 1992. "Compact finite difference schemes with spectral-like resolution". *Journal of computational physics*, Vol. 103, No. 1, pp. 16–42.

LeVeque, R.J., 2007. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM.

Phan-Thien, N., 1978. "A nonlinear network viscoelastic model". *Journal of Rheology*, Vol. 22, No. 3, pp. 259–283.

Roache, P.J., 2012. "Verification of codes and calculations." *AIAA Journal*, Vol. 36, No. 2, pp. 696–702.

Rogenski, J.K., 2011. *Desenvolvimento e otimização de um código paralelizado para simulação de escoamentos incompressíveis*. Ph.D. thesis, Universidade de São Paulo.

Souza, L.d., 2003. "Instabilidade centrífuga e transição para turbulência em escoamentos laminares sobre superfícies côncavas". *Instituto Tecnológico de Aeronáutica, São Paulo, Brasil*.

Steinberg, S. and Roache, P.J., 1985. "Symbolic manipulation and computational fluid dynamics". *Journal of Computational Physics*, Vol. 57, No. 2, pp. 251–284.

Thien, N.P. and Tanner, R.I., 1977. "A new constitutive equation derived from network theory". *Journal of Non-Newtonian Fluid Mechanics*, Vol. 2, No. 4, pp. 353–365.

10. RESPONSIBILITY NOTICE

The following text, properly adapted to the number of authors, must be included in the last section of the paper: The author(s) is (are) solely responsible for the printed material included in this paper.