Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 420189, 15 pages
doi:10.1155/2012/420189

Research Article

Viscoelastic Flow through an Axisymmetric Contraction Using the Grid-by-Grid Inversion Method

H. M. Park

Department of Chemical and Biomolecular Engineering, Sogang University, Seoul 121-742, Republic of Korea

Correspondence should be addressed to H. M. Park, hmpark@sogang.ac.kr

Received 1 December 2011; Revised 5 January 2012; Accepted 15 January 2012

Academic Editor: M. F. El-Amin

The newly developed algorithm called the grid-by-grid inversion method is a very convenient method for converting an existing computer code for Newtonian flow simulations to that for viscoelastic flow simulations. In this method, the hyperbolic constitutive equation is split such that the term for the convective transport of stress tensor is treated as a source which is updated iteratively. This allows the stress tensors at each grid point to be expressed in terms of velocity gradient tensor at the same location, and the set of stress tensor components is found after inverting a small matrix at each grid point. To corroborate the robustness and accuracy of the grid-by-grid inversion method, we apply it to the $4:1$ axisymmetric contraction problem. This algorithm is found to be robust and yields accurate results as compared with other finite volume methods. Any commercial CFD packages for Newtonian flow simulations can be easily converted to those for viscoelastic fluids exploiting the grid-by-grid inversion method.

1. Introduction

Contrary to the techniques of computational fluid dynamics for Newtonian fluids, the numerical algorithms for viscoelastic flows are not so matured. The hyperbolic nature of the constitutive equation incurs peculiar flow phenomena such as rod-climbing and extrudate swell as well as causes difficulties in numerical simulation [1–3]. Since the momentum balance equation is elliptic in steady state and parabolic in unsteady state, the complete set for viscoelastic flows is a mixed type, hyperbolic-elliptic, or hyperbolic-parabolic. This situation is difficult to treat numerically since it is difficult to devise a numerical algorithm that works for mixed systems. Another difficulty associated with the hyperbolic constitutive equation is the choice of appropriate boundary conditions for the stress field at the boundary of computational domain. One can impose nonslip boundary condition on the walls for the velocity field but there exist no apparent or natural boundary conditions for the stress components at the wall. Various numerical techniques for solving viscoelastic flows, such as finite volume methods, finite element methods, and spectral methods, are well documented in the references cited [4–9].

In the present investigation, a newly developed algorithm called the grid-by-grid inversion method [10] is employed to solve the viscoelastic flows through an axisymmetric contraction. Figure 1 shows the flow geometry for the axisymmetric contraction. Viscoelastic fluid passes from one circular tube into the other tube of smaller radius and generates a complex flow having a strong shear near the walls and uniaxial extension along the centerline. The existence of strong shearing zones and uniaxial extension zone makes this flow geometry a good test bed for numerical algorithms of viscoelastic fluids. Many investigators of rheology adopt this flow geometry as an important benchmark problem, especially the 4 : 1 contraction geometry [11]. We shall solve the 4 : 1 axisymmetric contraction problem employing the newly developed "grid-by-grid inversion method" [10], which is implemented based on a finite volume method [10].

The hyperbolic constitutive equations of viscoelastic fluids have a nonlocal character because of the term representing convective transport of stress tensor. If this term is assumed to be known, the constitutive equation becomes local and the stress tensor is easily evaluated for a given velocity gradient tensor at the same location. The six stress tensor components for the cases of a three-dimensional flow are found after inverting a six by six matrix at each grid point and are substituted into the Navier-Stokes equation as a source term. In this way, the numerical solution of viscoelastic flows becomes as straightforward as that of Newtonian fluids. We call this algorithm the grid-by-grid inversion method since the viscoelastic stress tensor is obtained by the grid-by-grid inversion of a matrix equation at each grid point. This algorithm can easily be implemented using finite volume methods, finite element methods and spectral methods. When applied to the 4 : 1 axisymmetric contraction problem, it is found that the grid-by-grid inversion method yields accurate results efficiently in comparison with numerical results of traditional algorithms.

2. Governing Equations and the Grid-by-Grid Inversion Method

We consider incompressible isothermal flows of viscoelastic fluids. The governing equations may be written in dimensionless variables as follows:

$$\nabla \cdot \mathbf{v} = 0, \tag{2.1}$$

$$\mathrm{Re}\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \nabla \cdot \boldsymbol{\sigma}, \tag{2.2}$$

$$\boldsymbol{\sigma} = -P\mathbf{I} + 2\beta\mathbf{D} + \boldsymbol{\tau}, \tag{2.3}$$

$$\lambda\left(\frac{\partial \boldsymbol{\tau}}{\partial t} + \mathbf{v} \cdot \nabla \boldsymbol{\tau}\right) = 2(1-\beta)\mathbf{D} + \lambda\left((\nabla \mathbf{v})^T \cdot \boldsymbol{\tau} + \boldsymbol{\tau} \cdot \nabla \mathbf{v}\right) - \boldsymbol{\tau}. \tag{2.4}$$

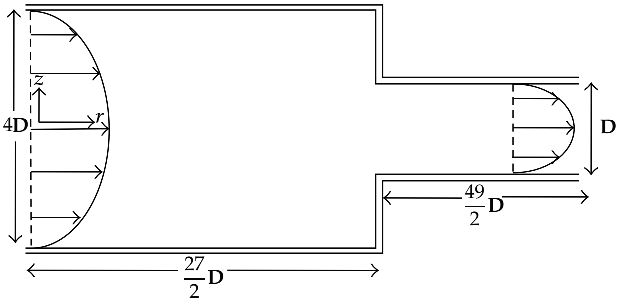


Figure 1: The 4 : 1 axisymmetric contraction.

In the above equations, P is pressure, $\mathbf{D}$ is the rate of deformation tensor, $\boldsymbol{\tau}$ is the viscoelastic part of the stress tensor. The parameter β is the ratio of the retardation and relaxation times of the fluid. Equation (2.4) is the constitutive equation of the Oldroyd-B model [9], λ is the dimensionless relaxation time or the Deborah number, and Re is the Reynolds number. The superscript T in (2.4) is the transpose. The Reynolds number and the dimensionless Deborah number are defined by

$$\text{Re} = \frac{\rho U L}{\eta}, \qquad \lambda = \frac{\lambda_1 U}{L}, \tag{2.5}$$

where ρ is the density, λ_1 the dimensional relaxation time, U the characteristic speed, and L is the characteristic length. The characteristic velocity U and the characteristic length L are taken as the average velocity in the downstream tube and the radius of the downstream tube, respectively. To compare with the results of other investigators [9, 11], we take the value of β to be $1/9$.

Next, we consider the grid-by-grid inversion method as applied to the set of equations (2.1)–(2.4). Contrary to the Newtonian fluids where the stress field $\boldsymbol{\tau}$ depends on the velocity gradient tensor $\nabla\mathbf{v}$ locally, the stress depends on the velocity gradient tensor nonlocally due to the fact that constitutive equation is hyperbolic partial differential equation. The convective transport of the stress tensor, represented by $\mathbf{v}\cdot\nabla\boldsymbol{\tau}$ in (2.4), takes care of the memory effect of $\boldsymbol{\tau}$ in its dependence on $\nabla\mathbf{v}$ and causes the functional dependence of $\boldsymbol{\tau}$ on $\nabla\mathbf{v}$ to be nonlocal. In the grid-by-grid inversion method, $\mathbf{v}\cdot\nabla\boldsymbol{\tau}$ is assumed to be a known source term which shall be updated iteratively in the process. Then the constitutive equation (2.4) can be converted such that $\boldsymbol{\tau}$ is represented as a local function of $\nabla\mathbf{v}$ as in the case of Newtonian fluids. This arrangement renders the numerical solution of viscoelastic fluids as straightforward as that of Newtonian fluids. The steps for obtaining the velocity and the stress at time step $n+1$, $\mathbf{v}^{n+1}$ and $\boldsymbol{\tau}^{n+1}$, using known velocity and stress at time step $n, \mathbf{v}^{n}$ and $\boldsymbol{\tau}^{n}$, proceed in an iterative manner as follows. First, define

$$\mathbf{C} \equiv \mathbf{v}^{n+1(it)} \cdot \nabla\boldsymbol{\tau}^{n+1(it)}, \tag{2.6}$$

where the superscript $n+1(it)$ indicates variable at time step $n+1$ in the itth iteration. Discretizing (2.4) in time implicitly and representing $\mathbf{D}$ in terms of the velocity gradient $\nabla\mathbf{v}$, we find the following local matrix equations defined at each grid point.

$$\begin{aligned}&\left(\frac{\lambda}{\Delta t}+1.0\right)\boldsymbol{\tau}^{n+1(it+1)}-\lambda(\nabla\mathbf{v})^{T^{n+1(it)}}\cdot\boldsymbol{\tau}^{n+1(it+1)}-\lambda\boldsymbol{\tau}^{n+1(it+1)}\cdot(\nabla\mathbf{v})^{n+1(it)}\\&\quad=\frac{\lambda}{\Delta t}\boldsymbol{\tau}^{n}-\lambda\mathbf{C}+(1-\beta)\left[(\nabla\mathbf{v})^{n+1(it)}+(\nabla\mathbf{v})^{T^{n+1(it)}}\right].\end{aligned}\tag{2.7}$$

Once $\mathbf{C}$ is evaluated using variables at $n+1(it)$, (2.7) can be solved for $\boldsymbol{\tau}^{n+1(it+1)}$ by inverting a six-by-six matrix at each grid point for the six independent stress components for the case of three-dimensional flows. Equation (2.7) is also solved at the boundary grid points to find the boundary stress field. This suggests a natural method of imposing boundary conditions for the hyperbolic constitutive equations at all boundaries of the computational domain. This method had been employed to impose outflow boundary conditions by Papanastasiou et al. [12] and is called the open boundary condition. Using $\boldsymbol{\tau}^{n+1(it+1)}$ obtained from (2.7), the velocity at the $n+1$ th time step in the $it+1$ th iteration, $\mathbf{v}^{n+1(it+1)}$, is found by solving (2.1)–(2.3) as follows:

$$\nabla\cdot\mathbf{v}^{n+1(it+1)}=0,\tag{2.8}$$

$$\mathrm{Re}\left(\frac{\mathbf{v}^{n+1(it+1)}-\mathbf{v}^{n}}{\Delta t}+\mathbf{v}^{n+1(it+1)}\cdot\nabla\mathbf{v}^{n+1(it+1)}\right)=-\nabla P^{n+1(it+1)}+2\beta\nabla^{2}\mathbf{v}^{n+1(it+1)}+\nabla\cdot\boldsymbol{\tau}^{n+1(it+1)}.\tag{2.9}$$

Although (2.8)-(2.9) can be solved using any numerical methods for incompressible Navier-Stokes equations, we employ a finite volume method based on the SIMPLE algorithm [13, 14]. To stabilize the numerical scheme for large values of λ, (2.6) is evaluated using one of the various upwind schemes. In the present work, we adopt a higher-order upwind scheme, MinMod [14], to evaluate $\mathbf{C}$.

The case of $\beta=0$ requires a special deliberation before employing standard algorithms for incompressible Navier-Stokes equations in the grid-by-grid inversion technique. When $\beta=0$, we decompose the total stress $\boldsymbol{\sigma}$ into pure viscous and elastic parts $-P\mathbf{I}+2\mathbf{D}$ and $\boldsymbol{\Sigma}$, respectively, such that

$$\boldsymbol{\sigma}=-P\mathbf{I}+2\mathbf{D}+\boldsymbol{\Sigma},\tag{2.10}$$

where

$$\boldsymbol{\Sigma}\equiv\boldsymbol{\tau}-2\mathbf{D}.\tag{2.11}$$

Then the momentum and constitutive equations for the case of $\beta = 0$ may be written as

$$\nabla \cdot \mathbf{v} = 0, \tag{2.12}$$

$$\mathrm{Re}\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla P + \nabla^2 \mathbf{v} + \nabla \cdot \mathbf{\Sigma}, \tag{2.13}$$

$$\lambda\left(\frac{\partial \mathbf{\Sigma}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{\Sigma} - (\nabla \mathbf{v})^T \cdot \mathbf{\Sigma} - \mathbf{\Sigma} \cdot (\nabla \mathbf{v})\right) + \mathbf{\Sigma} = -2\lambda\left(\frac{\partial \mathbf{D}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{D} - (\nabla \mathbf{v})^T \cdot \mathbf{D} - \mathbf{D} \cdot \nabla \mathbf{v}\right). \tag{2.14}$$

This decomposition is called the EVSS formulation and was proposed by Perera and Walters [15]. The implementation of grid-by-grid inversion method to the set of (2.12)–(2.14) is straightforward. First, we evaluate

$$\mathbf{E} \equiv \mathbf{v}^{n+1(it)} \cdot \nabla \mathbf{\Sigma}^{n+1(it)}. \tag{2.15}$$

Next, convert (2.14) to the following local matrix equations for $\mathbf{\Sigma}$ decoupled at each grid point:

$$\begin{aligned}\left(\frac{\lambda}{\Delta t} + 1.0\right)\mathbf{\Sigma}^{n+1(it+1)} - \lambda(\nabla \mathbf{v})^{T^{n+1(it)}} \cdot \mathbf{\Sigma}^{n+1(it+1)} - \lambda \mathbf{\Sigma}^{n+1(it+1)} \cdot (\nabla \mathbf{v})^{n+1(it)} \\ = \frac{\lambda}{\Delta t}\mathbf{\Sigma}^n - \lambda \mathbf{E} - 2\lambda\left(\frac{1}{\Delta t}\mathbf{D}^{n+1(it)} - \frac{1}{\Delta t}\mathbf{D}^n + \mathbf{v} \cdot \nabla \mathbf{D} - (\nabla \mathbf{v})^T \cdot \mathbf{D} - \mathbf{D} \cdot \nabla \mathbf{v}\right)^{n+1(it)}.\end{aligned} \tag{2.16}$$

Then, (2.12) and (2.13) are discretized as follows:

$$\nabla \cdot \mathbf{v}^{n+1(it+1)} \tag{2.17}$$

$$\mathrm{Re}\left(\frac{\mathbf{v}^{n+1(it+1)} - \mathbf{v}^n}{\Delta t} + \mathbf{v}^{n+1(it+1)} \cdot \nabla \mathbf{v}^{n+1(it+1)}\right) = -P^{n+1(it+1)} + \nabla^2 \mathbf{v}^{n+1(it+1)} + \nabla \cdot \mathbf{\Sigma}^{n+1(it+1)}. \tag{2.18}$$

The structure of (2.16) is the same as that of (2.7) and can be solved for $\mathbf{\Sigma}^{n+1(it+1)}$ by inverting a six-by-six matrix at each grid point for a three-dimensional flow once $(\nabla \mathbf{v})^{n+1(it)}$ is given. After obtaining $\mathbf{\Sigma}^{n+1(it+1)}$, $\mathbf{v}^{n+1(it+1)}$ is found by solving (2.17)-(2.18) using the SIMPLE algorithm [13]. The second-order derivative terms appearing in $\mathbf{v} \cdot \nabla \mathbf{D}$ of (2.14) are evaluated using a one-sided finite difference formula at the boundaries.

The overall solution procedure for the grid-by-grid inversion method may be summarized as follows.

(1) $\mathbf{v}^n$ and $\boldsymbol{\tau}^n$ have obtained in the previous time step n.

The procedure for the time step $n + 1$ begins as follows.

(2) Assume $\mathbf{v}^{n+1(it)}$ and $\boldsymbol{\tau}^{n+1(it)}$. For the first iteration ($it = 1$), $\mathbf{v}^{n+1(it)}$=$\mathbf{v}^n$ and $\boldsymbol{\tau}^{n+1(it)}$=$\boldsymbol{\tau}^n$.

(3) Evaluate $\mathbf{C}$ of (2.6) ($\beta \neq 0$) or $\mathbf{E}$ of (2.15) ($\beta = 0$) using an upwind scheme.

(4) Using $\mathbf{v}^{n+1(it)}$, solve (2.7) ($\beta \neq 0$) or (2.16) ($\beta = 0$) for $\boldsymbol{\tau}^{n+1(it+1)}$ by inverting a six-by-six matrix at each grid point including the boundary grids.

(5) Using $\boldsymbol{\tau}^{n+1(it+1)}$,

($\beta \neq 0$) solve the momentum equation, (2.9), and continuity equation to find $\mathbf{v}^{n+1(it+1)}$.
($\beta = 0$) solve the momentum equation, (2.18), and continuity equation to find $\mathbf{v}^{n+1(it+1)}$.

(6) Convergence check for $\boldsymbol{\tau}^{n+1}$ and $\mathbf{v}^{n+1}$. If not converged, go to step (2). Otherwise, update the time step and go to step (1).

Usually convergence is attained in two or three iterations. The novelty of the grid-by-grid inversion method is its easiness of numerical implementation. As noted in the above procedure, one can easily convert any existing computer code for Newtonian flow simulations to that for viscoelastic flow simulation by adding a subroutine that solves the viscoelastic constitutive equation using the grid-by-grid inversion method, summarized as steps (3)~(4), to evaluate the viscoelastic source terms in the Navier-Stokes equation, $\nabla \cdot \boldsymbol{\tau}$. Adding a source term to the Navier-Stokes equation is an easy procedure whether we employ a finite volume method or a finite element method. In the subroutine for the viscoelastic constitutive equation, one inverts a small matrix at each grid point, which can be performed cheaply. Therefore, one can easily convert any commercial CFD package for Newtonian fluid flows to that for viscoelastic fluids employing the grid-by-grid inversion method. Its robustness and accuracy are corroborated in the next section, where the grid-by-grid inversion method is employed to solve the 4 : 1 viscoelastic axisymmetric contraction problem. Although Phillips and Williams [16] solve the viscoelastic constitutive equation by converting small matrix equations in their semi-Lagrange method, it is very difficult to convert an existing Newtonian code to a viscoelastic code using the semi-Lagrange method since it treats the momentum and the constitutive equation simultaneously.

3. Viscoelastic Flow through an Axisymmetric Contraction

In this section, we solve the flow of an Oldroyd-B fluid through a 4 : 1 axisymmetric contraction geometry using the grid-by-grid inversion method. The schematic representation of the flow geometry is depicted in Figure 1. The viscoelastic flow through the circular contraction geometry has served as a standard benchmark problem for numerical algorithms for viscoelastic fluids. The presence of a singularity in the entry-flow geometry as well as regions of high shear and extension in the flow have been a significant challenge for a long time, and many investigators have struggled to develop accurate and robust algorithms for viscoelastic flows through contraction geometries [17, 18]. The contraction ratio 4 : 1 is the standard one in this benchmark problem. A good review of old works may be Trebotich et al. [18]. Recently, Phillips and Williams [11, 16] used a finite volume method to solve the flow of an Oldroyd-B fluid through planar and axisymmetric contractions. Among these two geometries, the axisymmetric contraction yields more dramatic viscoelastic effects. In the present investigation, we will solve the flow of Oldroyd-B fluid through the axisymmetric geometry using the grid-by-grid inversion method based on the SIMPLE algorithm to corroborate the accuracy and robustness of the grid-by-grid inversion method. When solving (2.8)-(2.9) or (2.17)-(2.18) using the SIMPLE algorithm, we adopt a collocated mesh and the pressure oscillation induced by the collocated mesh is eliminated using the Rhie-Chow method [19], and a higher-order upwind scheme, MinMod [14], is used to treat the inertia force term in the momentum equation and the convective transport of stress tensor in the constitutive equation. For the 4 : 1 axisymmetric contraction geometry depicted in Figure 1, the number of relevant components of stress tensor is four because the $\nabla \cdot \boldsymbol{\tau}$ term in (2.9) contains a circular stress component $\tau^{\theta\theta}$ as follows:

$$\begin{gathered}(\nabla \cdot \boldsymbol{\tau})_r = \frac{1}{r}\frac{\partial}{\partial r}(r\tau^{rr}) + \frac{\partial}{\partial z}\tau^{rz} - \frac{\tau^{\theta\theta}}{r}, \\ (\nabla \cdot \boldsymbol{\tau})_z = \frac{1}{r}\frac{\partial}{\partial r}(r\tau^{rz}) + \frac{\partial}{\partial z}\tau^{zz}.\end{gathered} \tag{3.1}$$

Although the relevant velocity components for axisymmetric flows are (v^r, v^z), it is necessary to find the four stress components, $\tau^{rr}, \tau^{rz}, \tau^{zz}$, and $\tau^{\theta\theta}$ which appear in (3.1). The constitutive equation (2.4) may be written for an axisymmetric geometry as follows:

$$\lambda\left(\frac{\partial \tau^{rr}}{\partial t} + v^r\frac{\partial \tau^{rr}}{\partial r} + v^z\frac{\partial \tau^{rr}}{\partial z}\right) = 2(1-\beta)\frac{\partial v^r}{\partial r} + \lambda\left(2\frac{\partial v^r}{\partial r}\tau^{rr} + 2\frac{\partial v^r}{\partial z}\tau^{rz}\right) - \tau^{rr}, \tag{3.2}$$

$$\begin{aligned}&\lambda\left(\frac{\partial \tau^{rz}}{\partial t} + v^r\frac{\partial \tau^{rz}}{\partial r} + v^z\frac{\partial \tau^{rz}}{\partial z}\right) \\ &\quad = (1-\beta)\left(\frac{\partial v^z}{\partial r} + \frac{\partial v^r}{\partial z}\right) + \lambda\left(\frac{\partial v^r}{\partial r}\tau^{rz} + \frac{\partial v^r}{\partial z}\tau^{zz} + \tau^{rr}\frac{\partial v^z}{\partial r} + \tau^{rz}\frac{\partial v^z}{\partial z}\right) - \tau^{rz},\end{aligned} \tag{3.3}$$

$$\lambda\left(\frac{\partial \tau^{zz}}{\partial t} + v^r\frac{\partial \tau^{zz}}{\partial r} + v^z\frac{\partial \tau^{zz}}{\partial z}\right) = 2(1-\beta)\frac{\partial v^z}{\partial z} + \lambda\left(2\frac{\partial v^z}{\partial r}\tau^{rz} + 2\frac{\partial v^z}{\partial z}\tau^{zz}\right) - \tau^{zz}, \tag{3.4}$$

$$\lambda\left(\frac{\partial \tau^{\theta\theta}}{\partial t} + v^r\frac{\partial \tau^{\theta\theta}}{\partial r} + v^z\frac{\partial \tau^{\theta\theta}}{\partial z}\right) = 2(1-\beta)\frac{v^r}{r} + \lambda\left(2\frac{v^r}{r}\tau^{\theta\theta}\right) - \tau^{\theta\theta}. \tag{3.5}$$

It is to be noted that equations for $(\tau^{rr}, \tau^{rz}, \tau^{zz})$ are coupled with each other, while that for $\tau^{\theta\theta}$ is decoupled. The relevant components of $\mathbf{C}$ of (2.6) are

$$C^{IJ} = v^r\frac{\partial \tau^{IJ}}{\partial r} + v^z\frac{\partial \tau^{IJ}}{\partial z}, \tag{3.6}$$

where $IJ = rr,\ rz,\ zz,\ \theta\theta$. Assuming known values of C^{IJ}, (3.2)–(3.4) may be represented as a set of matrix equations at each grid point including the boundary grids as follows:

$$\begin{bmatrix} \dfrac{\lambda}{\Delta t} + 1 - 2\lambda\dfrac{\partial v^r}{\partial r} & -2\lambda\dfrac{\partial v^r}{\partial z} & 0 \\ -\lambda\dfrac{\partial v^z}{\partial r} & \dfrac{\lambda}{\Delta t} + 1 - \lambda\left(\dfrac{\partial v^r}{\partial r} + \dfrac{\partial v^z}{\partial z}\right) & -\lambda\dfrac{\partial v^r}{\partial z} \\ 0 & -2\lambda\dfrac{\partial v^z}{\partial r} & \dfrac{\lambda}{\Delta t} + 1 - 2\lambda\dfrac{\partial v^z}{\partial z} \end{bmatrix}_{ik}^{n+1(it)} \begin{bmatrix} \tau^{rr} \\ \tau^{rz} \\ \tau^{zz} \end{bmatrix}_{ik}^{n+1(it+1)}$$

$$= \begin{bmatrix} \dfrac{\lambda\tau^{rr(n)}}{\Delta t} - \lambda C^{rr} + 2(1-\beta)\dfrac{\partial v^r}{\partial r} \\ \dfrac{\lambda\tau^{rz(n)}}{\Delta t} - \lambda C^{rz} + (1-\beta)\left(\dfrac{\partial v^z}{\partial r} + \dfrac{\partial v^r}{\partial z}\right) \\ \dfrac{\lambda}{\Delta t}\tau^{zz(n)} - \lambda C^{zz} + 2(1-\beta)\dfrac{\partial v^z}{\partial z} \end{bmatrix}_{ik}^{n+1(it)}. \tag{3.7}$$

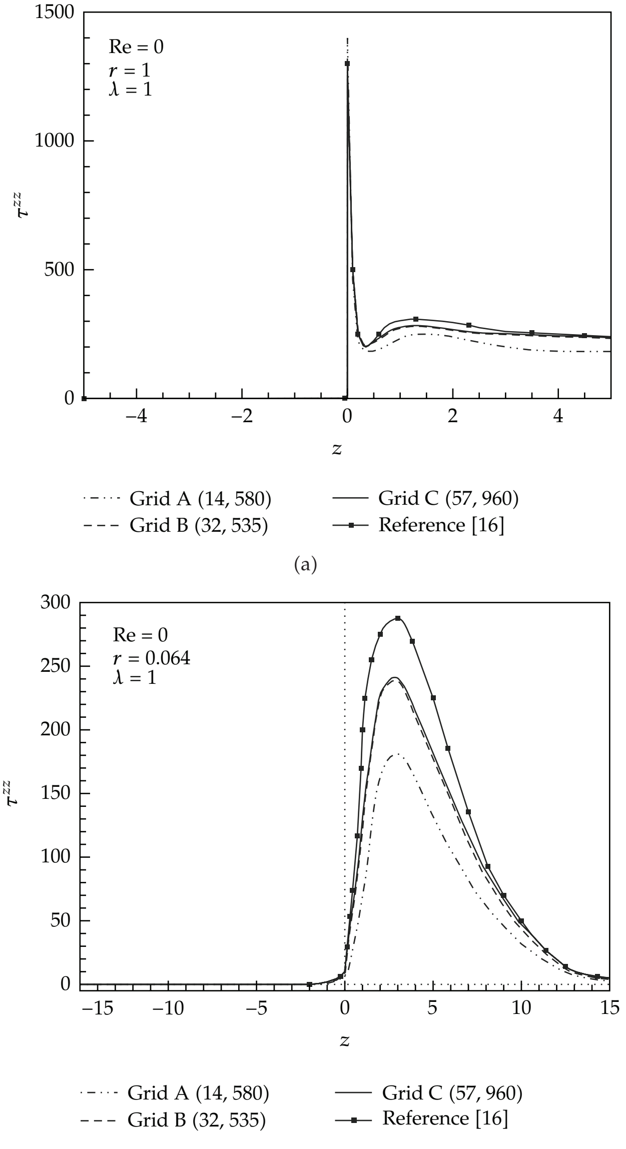
On the other hand, from (3.5), $\tau_{ik}^{\theta\theta}$ can be found at each grid point within the computational domain and on the boundary as follows:

$$\left(\tau^{\theta\theta}\right)_{ik}^{n+1(it+1)} = \frac{\left(\dfrac{\lambda}{\Delta t}\left(\tau^{\theta\theta}\right)^n - \lambda C^{\theta\theta} + 2(1-\beta)\dfrac{v^r}{r}\right)_{ik}^{n+1(it)}}{\left(\dfrac{\lambda}{\Delta t} + 1 - 2\lambda\dfrac{v^r}{r}\right)_{ik}^{n+1(it)}}. \tag{3.8}$$

Once $(\tau_{ik}^{rr}, \tau_{ik}^{rz}, \tau_{ik}^{zz}, \tau_{ik}^{\theta\theta})^{n+1(it+1)}$ are obtained from (3.7)-(3.8), they are used to solve (2.8)-(2.9) to find $(v^r, v^z)_{ik}^{n+1(it+1)}$ in the 4 : 1 axisymmetric contraction geometry.

4. Results

In this section, we compare the results from the grid-by-grid inversion method with those of Phillips and Williams [11] to corroborate the accuracy and robustness of the grid-by-grid inversion method. To make the comparison meaningful, we adopt the same flow geometry and inlet conditions as those employed by Phillips and Williams [11]. Namely, the contraction



(a)

(b)

Figure 2: Confirmation of grid convergence of the grid-by-grid inversion method when Re = 0.0 and $\lambda = 1.0$ (a) along $r = 1.0$ (b) along $r = 0.064$.

ratio is 4 : 1, the length of the larger channel is $27R$, and that of smaller channel is $49R$ when R is the radius of the smaller channel (cf. Figure 1). At the inlet, the parabolic Poiseuille flow is adopted as Phillips and Williams [11] have done.

(-) inlet velocity

$$v^z(r) = \frac{1}{64}\left(16 - r^2\right), \quad v^r = 0, \; 0 \le r \le 4, \tag{4.1}$$

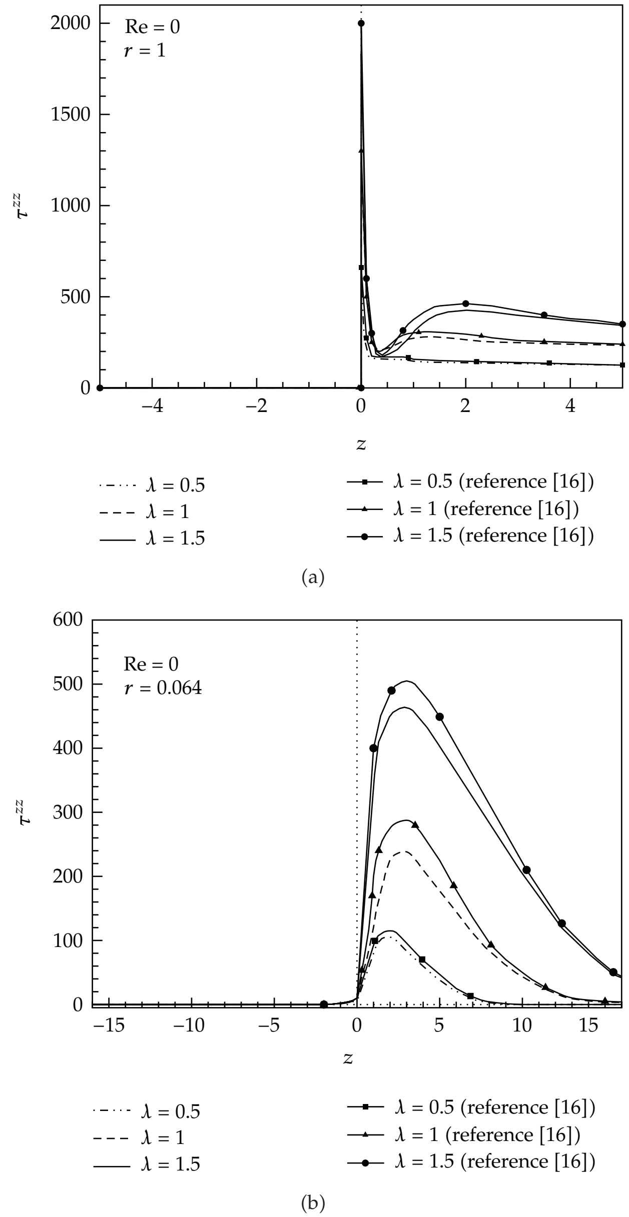


Figure 3: Stress overshoot at various λ when Re = 0.0 (a) along r = 1.0 (b) along r = 0.064.

where $r \equiv r'/R$ and r' is the dimensional radial distance. Contrary to Phillips and Williams [11], we do not impose inlet boundary conditions for the stress field, since the grid-by-grid inversion method does not require them and simply solve (3.7)-(3.8) at the inlet grid points to find the stress field at the inlet location. This is also an important merit of the grid-by-grid inversion method over other algorithms for the viscoelastic flows. Although Papanastasiou et al. [12] employed this method of imposing open outflow boundary conditions previously and named it open boundary condition method or free boundary condition method, it appears very naturally in the grid-by-grid inversion method.

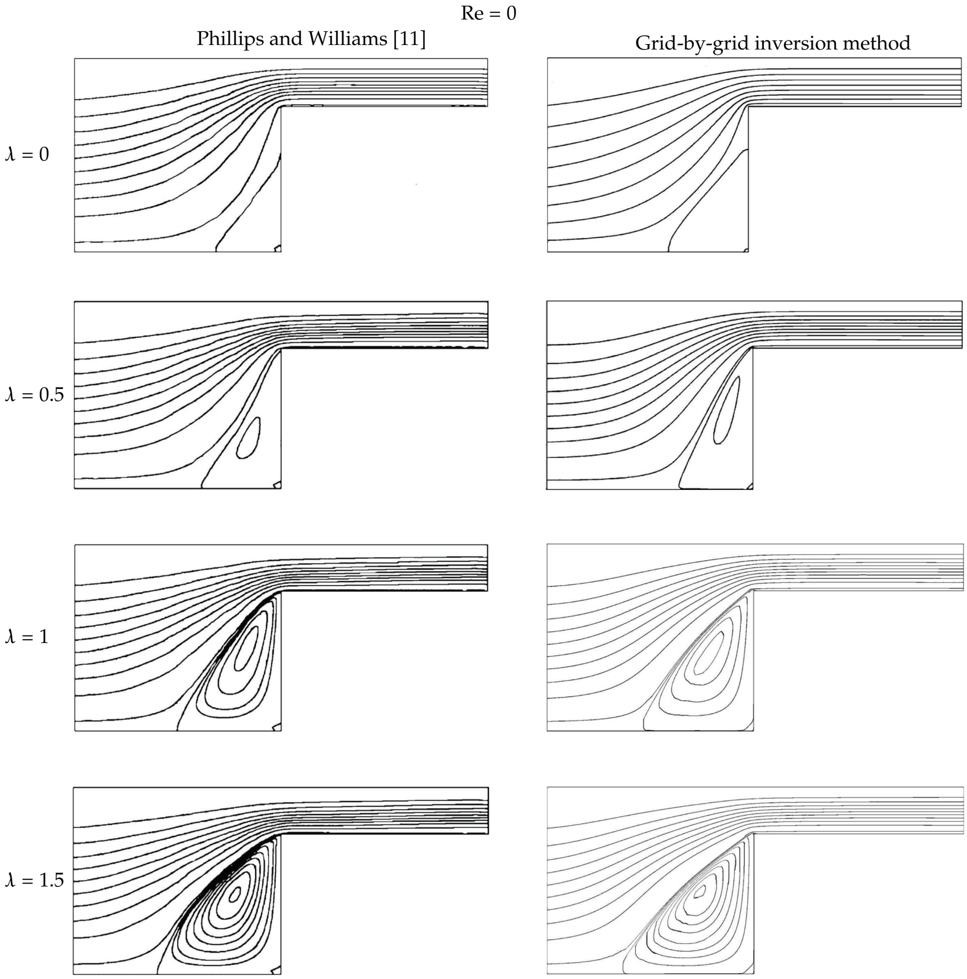


Figure 4: Comparison of streamlines for $\lambda = 0.0$, $\lambda = 0.5$, $\lambda = 1.0$, and $\lambda = 1.5$ when Re = 0.0.

We have employed three sets of grid system to ensure grid convergence of numerical results. For the three sets of grids, that is, 14,580 (Grid A), 32,535 (Grid B), and 57,960 (Grid C), the extrastress component τ^{zz} is plotted when $\lambda = 1.0$ for Re $= 0.0$ along $r = 1$ in Figure 2(a), and along $r = 0.064$ in Figure 2(b). For there three mesh systems, the numbers of cells along r in the small tube are 30, 45, and 60, respectively. For Grid A, the minimum Δr is 0.005 at the wall and it grows gradually until it becomes 5.1×10^{-2} at the center. For Grid B, $(\Delta r)_{\text{min}}$ is 0.003 and $(\Delta r)_{\text{max}}$ is 3.4×10^{-2} at the center, and for Grid C, $(\Delta r)_{\text{min}} = 0.002$ at the wall and $(\Delta r)_{\text{max}} = 2.3 \times 10^{-2}$ at the center. The results of the present work are also compared with those of Phillips and Williams [11]. Though the grid number increases from 32,535 to 57,960, there is not appreciable change in τ^{zz}. Therefore, we adopt Grid B in the subsequent computations. Figure 3 shows the extrastress component τ^{zz} at Re $= 0.0$ for $\lambda = 0.5, 1.0$, and 1.5 along $r = 1.0$ (Figure 2(a)) and along $r = 0.064$ (Figure 2(b)). The grid-converged data for τ^{zz} obtained from the grid-by-grid inversion method are somewhat smaller than the benchmark data of Phillips and Williams [11]. At the corner of the contractor, τ^{zz} has a sharp overshoot, which increases with respect to λ, and it settles down to a downstream

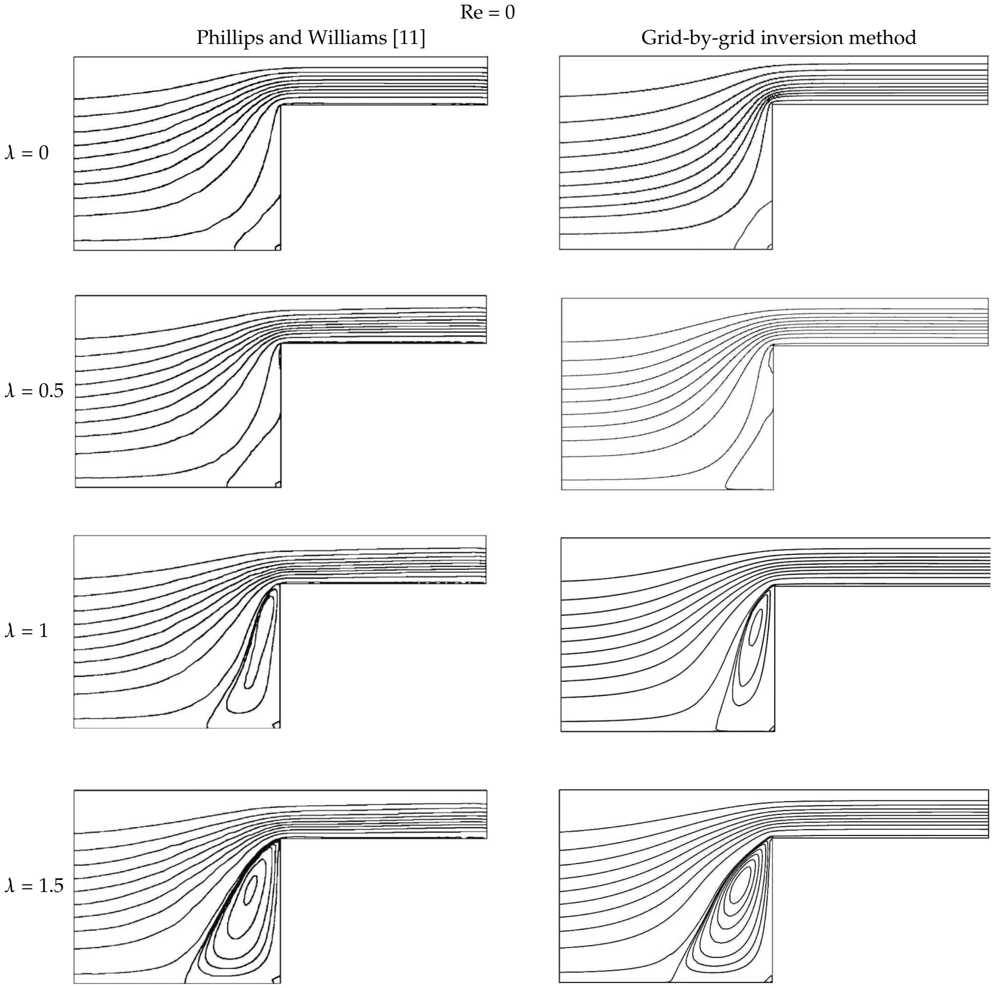


Figure 5: Comparison of streamlines for λ = 0.0, λ = 0.5, λ = 1.0, and λ = 1.5 when Re = 1.0.

value rapidly. Near the center of the contracted channel (r = 0.064), a smaller overshoot of τ^{zz} appears, which increases as λ increases, and it settles down to a downstream value much slowly as compared to that at the wall (r = 1.0).

Figure 4 shows comparison of streamlines for λ = 0.0, 0.5, 1.0, and 1.5 obtained from the grid-by-grid inversion method with those from the Phillips and Williams [11] when Re = 0.0. The grid-by-grid inversion method yields accurate results as compared with the benchmark data for the range of λ values considered. The size of the corner vortex increases as λ increases. Figure 5 depicts comparison of streamlines at various values of λ when Re = 1.0. The predictions of the grid-by-grid inversion method are also in good agreement with those of the benchmark data [11]. Comparing with the streamlines of Re = 0.0, it is found that the vortex sizes are diminished due to the inertia effect [11]. The size of the corner vortex can be measured using a parameter L_1, as suggested by Phillips and Williams [11] and Oliveira et al.[8], which is the distance of the upstream separation point from the salient corner. Figure 6 shows the variation of L_1 with respect to λ for Re = 0.0 and Re = 1.0.

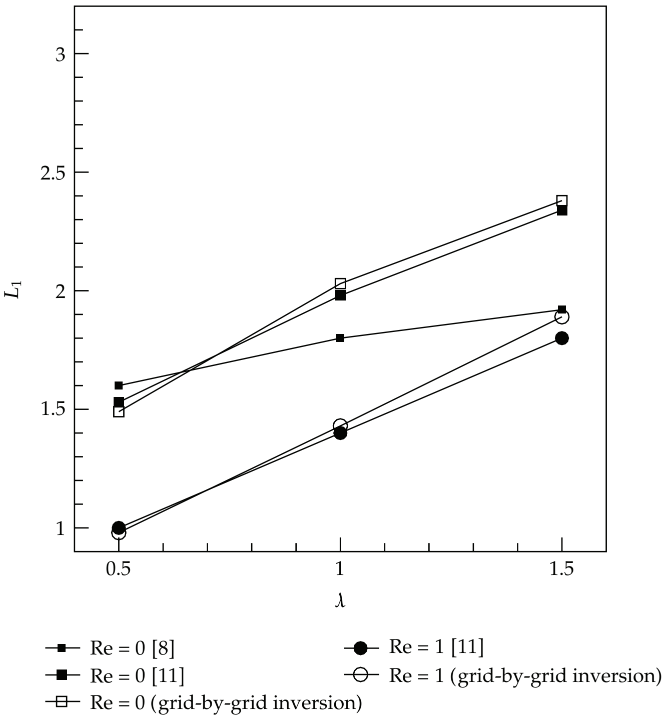


Figure 6: Variation of the size of corner vortex L_1 with respect to λ when Re = 0.0 and Re = 1.0.

For Re = 1.0, the grid-by-grid inversion method is found to yield results similar to those of the benchmark data of Phillips and Williams [11]. For Re = 0.0, the results of the grid-by-grid inversion method are compared with those of Oliveira et al. [8] as well as those of Phillips and Williams [11]. It is shown that both the grid-by-grid inversion and Phillips and Williams [11] yield L_1 somewhat larger than those of Oliveira et al. [8]. Since Oliveira et al. [8] do not consider the case Re = 1.0, we cannot compare the grid-by-grid inversion method with Oliveira et al. when Re = 1.0. To corroborate the accuracy of the grid-by-grid inversion method further, we compare the results for the vortex intensity Ψ_R, which is defined as the flow rate in recirculation divided by inlet flow rate, in Figure 7 when Re = 0.0. As in previous comparison, the grid-by-grid inversion method yields results similar to those of Phillips and Williams [11], while Oliveira et al. [8] predicts results somewhat smaller than the grid-by-grid inversion method. The grid-by-grid inversion method yields results for λ larger than 1.5 and Re larger than 1.0, which demonstrates the robustness of the grid-by-grid inversion method as compared to other finite volume methods or finite element methods [9, 11, 16, 18].

5. Conclusion

The grid-by-grid inversion method [10] has been applied to the 4 : 1 circular contraction problem in the present investigation. Viscoelastic flows through the contraction generate complex flows exhibiting strong shear and uniaxial extension, which is a good test bed for the robustness and accuracy of a new numerical algorithms. In the grid-by-grid inversion

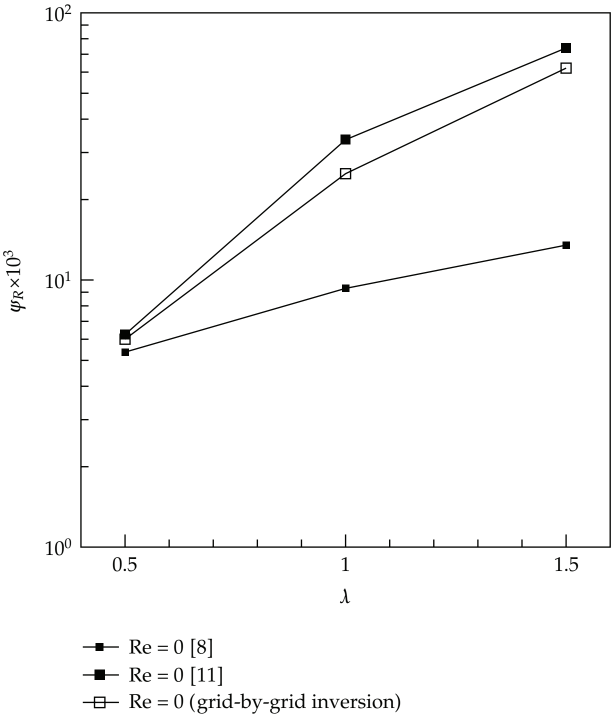


Figure 7: Variation of the dimensionless vortex intensity Ψ_R with respect to λ when Re = 0.0.

method [10], the hyperbolic constitutive equation is split such that the term for the convective transport of stress tensor is treated as a source, which is updated iteratively. This allows the stress tensor at each grid point to be expressed in terms of velocity gradient tensor at the same location, and the set of stress tensor components is found after inverting a small matrix at each grid point. The grid-by-grid inversion method can be implemented easily in any commercial CFD packages for Newtonian fluid flows to convert them to be used for viscoelastic fluid flows. The grid-by-grid inversion method is found to yield accurate results as compared with the benchmark data of Phillips and Williams [11]. It predicts accurately the variation of corner vortex size and the τ^{zz} overshoot with respect to λ at Re = 0.0 and Re = 1.0.

Acknowledgment

This paper was supported by the Human Resources Development of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) Grant funded by the Korea Government Ministry of Knowledge and Economy (no. 20114010203090).

References

[1] R. B. Bird, R. C. Amstrong, and O. Hassager, *Dynamics of Polymeric Liquids*, vol. 1, Wiley, New York, NY, USA, 1987.

[2] R. R. Huilgol and N. Phan-Thien, *Fluid Mechanics of Viscoelasticity*, Elsevier, Amsterdam, The Netherlands, 1997.

[3] D. D. Joseph, *Fluid Dynamics of Viscoelastic Liquids*, vol. 84 of *Applied Mathematical Sciences*, Springer, 1990.

[4] M. A. Alves, F. T. Pinho, and P. J. Oliveira, "The flow of viscoelastic fluids past a cylinder: finite-volume high-resolutions method," *Journal of Non-Newtonian Fluid Mechanics*, vol. 97, no. 2-3, pp. 207–232, 2001.

[5] M. J. Crochet, A. R. Davis, and K. Walters, *Numerical Simulation of Non-Newtonian Flow*, Elsevier, Amsterdam, The Netherlands, 1984.

[6] F. Yurun and M. J. Crochet, "High-order finite element methods for steady viscoelastic flows," *Journal of Non-Newtonian Fluid Mechanics*, vol. 57, no. 2-3, pp. 283–311, 1995.

[7] B. Khomami, K. K. Talwar, and H. K. Ganpule, "A comparative study of higher- and lower-order finite element techniques for computation of viscoelastic flows," *Journal of Rheology*, vol. 38, no. 2, pp. 255–289, 1994.

[8] M. S. N. Oliveira, P. J. Oliveira, F. T. Pinho, and M. A. Alves, "Effect of contraction ratio upon viscoelastic flow in contractions: the axisymmetric case," *Journal of Non-Newtonian Fluid Mechanics*, vol. 147, no. 1-2, pp. 92–108, 2007.

[9] R. G. Owens and T. N. Phillips, *Computational Rheology*, Imperial College Press, London, UK, 2002.

[10] H. M. Park and J. Y. Lim, "A new numerical algorithm for viscoelastic fluid flows: the grid-by-grid inversion method," *Journal of Non-Newtonian Fluid Mechanics*, vol. 165, no. 5-6, pp. 238–246, 2010.

[11] T. N. Phillips and A. J. Williams, "Comparison of creeping and inertial flow of an Oldroyd B fluid through planar and axisymmetric contractions," *Journal of Non-Newtonian Fluid Mechanics*, vol. 108, no. 1–3, pp. 25–47, 2002.

[12] T. C. Papanastasiou, N. Malamataris, and K. Ellwood, "A new outflow boundary condition," *International Journal for Numerical Methods in Fluids*, vol. 14, no. 5, pp. 587–608, 1992.

[13] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, NY, USA, 1980.

[14] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics*, Pearson Education Limited, Essex, UK, 2007.

[15] M. G. N. Perera and K. Walters, "Long-range memory effects in flows involving abrupt changes in geometry: part1. Flows associated with L-shaped and T-shaped geometries," *Journal of Non-Newtonian Fluid Mechanics*, vol. 2, no. 2, pp. 191–204, 1977.

[16] T. N. Phillips and A. J. Williams, "Viscoelastic flow through a planar contraction using a semi-Lagrangian finite volume method," *Journal of Non-Newtonian Fluid Mechanics*, vol. 87, no. 2-3, pp. 215–246, 1999.

[17] R. Keunings and M. J. Crochet, "Numerical simulation of the flow of a viscoelastic fluid through an abrupt contraction," *Journal of Non-Newtonian Fluid Mechanics*, vol. 14, pp. 279–299, 1984.

[18] D. Trebotich, P. Colella, and G. H. Miller, "A stable and convergent scheme for viscoelastic flow in contraction channels," *Journal of Computational Physics*, vol. 205, no. 1, pp. 315–342, 2005.

[19] C. M. Rhie and W. L. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation," *AIAA Journal*, vol. 21, no. 11, pp. 1525–1532, 1983.