Journal of Non-Newtonian Fluid Mechanics 167–168 (2012) 30–37

Contents lists available at SciVerse ScienceDirect

Journal of Non-Newtonian Fluid Mechanics

journal homepage: http://www.elsevier.com/locate/jnnfm

The *kernel-conformation* constitutive laws

A.M. Afonso [a,*], F.T. Pinho [b], M.A. Alves [a]

[a] Departamento de Engenharia Química, Centro de Estudos de Fenómenos de Transporte, Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

[b] Departamento de Engenharia Mecânica, Centro de Estudos de Fenómenos de Transporte, Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:
Received 12 January 2011
Received in revised form 28 September 2011
Accepted 29 September 2011
Available online 10 October 2011

Keywords:
Kernel-conformation tensor transformation
Computational rheology
Differential constitutive equations

ABSTRACT

This work presents a generic *kernel-conformation* tensor transformation for a large class of differential constitutive models, in which several matrix *kernel*-transformation families can be applied to the conformation tensor evolution equation. The discretization of this generic analytical framework generates numerical results consistent with the *standard* conformation tensor discretization at low Weissenberg numbers (*Wi*). The numerical efficiency of High Weissenberg Number Problems is ultimately related with both the specific *kernel* function used in the matrix transformation, but is also related to the existence of mathematical singularities introduced either by the physical description of the flow or by the characteristics of the adopted constitutive equation. We also show that the *log-conformation* [R. Fattal, R. Kupferman, J. Non-Newtonian Fluid Mech., 123(2–3) (2004) 281–285] and the *square-root-conformation* [N. Balci, B. Thomases, M. Renardy, C.R. Doering, J. Non-Newtonian Fluid Mech., 166(11) (2011) 546–553] approaches are two relevant particular cases of the *kernel-conformation* tensor transformation. Simulations for the benchmark confined cylinder flow of an Oldroyd-B fluid provide a preliminary assessment of the merits of some tested transformation functions.

1. Introduction

In computational rheology, the momentum and mass conservation equations are inherently coupled with constitutive equations, usually of macroscopic type, here represented by the class of differential equations. These rheological equations of state can be often formulated in terms of the extra-stress tensor, $\boldsymbol{\tau}$, or of the conformation tensor, $\mathbf{A}$. The conformation tensor is a variance-covariance, symmetric positive definite tensor (SPD), and should remain SPD when evolving in time. Therefore, it can always be diagonalized into

$$\mathbf{A} = \mathbf{O}\boldsymbol{\Lambda}\mathbf{O}^T \tag{1}$$

In a pictorial view, it represents the local, instantaneous polymeric "conformation" (or configuration) of an ensemble of model "molecules", in which its unitary eigenvectors (columns of the orthogonal matrix $\mathbf{O}$) represent the principal axes of the polymeric conformation and its eigenvalues (the diagonal elements of the diagonal matrix $\boldsymbol{\Lambda}$) represent the magnitude of the extension of the polymeric conformation along the eigenvector directions. The eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ of $\mathbf{A}$ must be real and positive because $\mathbf{A}$ is positive definite.

The time and space evolution of this conformation tensor obeys an evolution equation, here represented by differential constitutive models. Many of these constitutive models present the following generic form [1,2]:

$$\frac{\mathrm{D}\mathbf{A}}{\mathrm{D}t} = \frac{\partial \mathbf{A}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{A} = \mathbf{A}\nabla\mathbf{u} + \nabla\mathbf{u}^T\mathbf{A} + \frac{\mathcal{F}(\mathbf{A})}{Wi}\mathcal{H}(\mathbf{A}) \tag{2}$$

or as used by Fattal and Kupferman [1]

$$\frac{\mathrm{D}\mathbf{A}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbf{A} - \mathbf{A}\boldsymbol{\Omega} + 2\mathbf{B}\mathbf{A} + \frac{\mathcal{F}(\mathbf{A})}{Wi}\mathcal{H}(\mathbf{A}) \tag{3}$$

where $\mathbf{B}$ and $\boldsymbol{\Omega}$ are defined in Eq. (4) below, $\mathcal{F}(\mathbf{A})$ is a scalar (hereafter represented as $\mathcal{F}$) and $\mathcal{H}(\mathbf{A})$ a tensor, and both of these functions allow the definition of a specific constitutive model, such as the Oldroyd-B, PTT or FENE-type models. The Weissenberg number is defined as $Wi = \tau_p U/L$, where τ_p is the relaxation time of the fluid, and U and L are characteristic velocity and length scales, respectively. It is a non-dimensional parameter that corresponds to the ratio between the relaxation time of the fluid and a time scale of the flow (L/U) and measures the elasticity of the fluid in a given flow.

The advection and deformation induced by the flow field on the conformation tensor is here represented by the upper-convected time derivative. This derivative was reformulated by Fattal and Kupferman [1,2], by splitting the velocity gradient and its transpose into a symmetric traceless pure extension component (oriented with the axes of the polymeric conformation), the tensor $\mathbf{B}$

* Corresponding author.
E-mail addresses: aafonso@fe.up.pt (A.M. Afonso), fpinho@fe.up.pt (F.T. Pinho), mmalves@fe.up.pt (M.A. Alves).

0377-0257/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jnnfm.2011.09.008

that commutes with **A**, and an anti-symmetric pure rotation component, tensor $\mathbf{\Omega}$ (which in general is not the vorticity tensor). This local decomposition rule for the traceless second-order transpose velocity gradient tensor was obtained by setting:

$$\begin{aligned} &\mathbf{L} = \nabla\mathbf{u}^T = \mathbf{\Omega} + \mathbf{B} + \mathbf{N}\mathbf{A}^{-1} \\ &\widetilde{\mathbf{L}} = \mathbf{O}^T\nabla\mathbf{u}^T\mathbf{O} = \widetilde{\mathbf{\Omega}} + \widetilde{\mathbf{B}} + \widetilde{\mathbf{N}}\mathbf{\Lambda}^{-1} \end{aligned} \tag{4}$$

with the tensors $\widetilde{\mathbf{B}}$, $\widetilde{\mathbf{\Omega}}$ and $\widetilde{\mathbf{N}}$ representing the symmetric (diagonal) and the two anti-symmetric matrices, respectively, used originally by Fattal and Kuperfman [1,2]. As shown in that work this decomposition always exists and is unique. It can be represented for a two dimensional case as,

$$\mathbf{O}^T\nabla\mathbf{u}^T\mathbf{O} = \begin{bmatrix} \tilde{m}_{11} & \tilde{m}_{12} \\ \tilde{m}_{21} & \tilde{m}_{22} \end{bmatrix} \Rightarrow \begin{cases} \widetilde{\mathbf{B}} = \widetilde{B}_{ii} = \begin{bmatrix} \tilde{m}_{11} & 0 \\ 0 & \tilde{m}_{22} \end{bmatrix} \\ \widetilde{\mathbf{\Omega}} = \begin{bmatrix} 0 & \varpi \\ -\varpi & 0 \end{bmatrix} \\ \widetilde{\mathbf{N}} = \begin{bmatrix} 0 & n \\ -n & 0 \end{bmatrix} \end{cases} \tag{5}$$

with $\varpi = \frac{\lambda_2\tilde{m}_{12}+\lambda_1\tilde{m}_{21}}{\lambda_2-\lambda_1}$ and $n = \frac{\tilde{m}_{12}+\tilde{m}_{21}}{\lambda_2^{-1}-\lambda_1^{-1}}$. Note that the term $\mathbf{N}\mathbf{A}^{-1}$ vanishes in the back substitution of Eq. (4) onto Eq. (2), leading to the final form of Eq. (3) [1]. The evolution equation for **A** can also be represented in a *eigendecomposed* version (see [3]), by using the chain derivative rule for $\mathbf{O}^T\mathbf{A}\mathbf{O} = \mathbf{\Lambda}$:

$$\frac{\mathrm{D}\mathbf{\Lambda}}{\mathrm{D}t} = \frac{\mathrm{D}\mathbf{O}^T}{\mathrm{D}t}\mathbf{A}\mathbf{O} + \mathbf{O}^T\frac{\mathrm{D}\mathbf{A}}{\mathrm{D}t}\mathbf{O} + \mathbf{O}^T\mathbf{A}\frac{\mathrm{D}\mathbf{O}}{\mathrm{D}t} \tag{6}$$

which using Eq. (3) leads to the following evolution equation for the diagonal tensor $\mathbf{\Lambda}$,

$$\frac{\mathrm{D}\mathbf{\Lambda}}{\mathrm{D}t} = \mathbf{\Lambda}\mathcal{O} - \mathcal{O}\mathbf{\Lambda} + \widetilde{\mathbf{\Omega}}\mathbf{\Lambda} - \mathbf{\Lambda}\widetilde{\mathbf{\Omega}} + 2\widetilde{\mathbf{B}}\mathbf{\Lambda} + \frac{\mathcal{F}}{Wi}\mathcal{H}(\mathbf{\Lambda}) \tag{7}$$

where $\mathcal{O} = \mathbf{O}^T\frac{\mathrm{D}\mathbf{O}}{\mathrm{D}t} = -\frac{\mathrm{D}\mathbf{O}^T}{\mathrm{D}t}\mathbf{O} = -\mathcal{O}^T$, and so a skew-symmetric tensor. Since $\mathcal{O}$ (and also $\widetilde{\mathbf{\Omega}}$) are anti-symmetric tensors (with all the diagonal elements being equal to zero), therefore they cannot contribute to the time evolution of $\mathbf{\Lambda}$, and Eq. (7) reduces to

$$\frac{\mathrm{D}\lambda_i}{\mathrm{D}t} = 2\widetilde{B}_{ii}\lambda_i + \frac{\mathcal{F}}{Wi}\mathcal{H}(\lambda_i) \tag{8}$$

for the diagonal components (no summation over *ii*). For the off-diagonal components ($i \neq j$) Eq. (7) reduces to

$$\mathcal{O}\mathbf{\Lambda} - \mathbf{\Lambda}\mathcal{O} = \widetilde{\mathbf{\Omega}}\mathbf{\Lambda} - \mathbf{\Lambda}\widetilde{\mathbf{\Omega}} \tag{9}$$

In computational rheology, the inclusion of a viscoelastic constitutive equation in the solution procedure not only increases the total number of degrees of freedom relative to the corresponding Generalized Newtonian fluid computation, but also modifies the mathematical type of the resulting system of governing equations. The well known High Weissenberg Number Problem (HWNP) can be related to the numerical discretization method, because the exact form of the constitutive equation can admit regular solutions at the continuous level. The evolutionary character of the hyperbolic constitutive equations is usually dependent on the positive definiteness of the conformation tensor, and numerical discretization errors could, eventually, lead to the loss of such positive definiteness, resulting in a loss of topological evolutionarity that can trigger Hadamard instabilities [4]. Nevertheless, in our view, the loss of the positive definiteness of the conformation tensor is not a *cause* of HWNP but a *consequence*. As pointed out by Fattal and Kupferman [1,2], the HWNP is a numerical instability caused by the failure to balance the exponential growth of the stresses (due to deformation) with their convection, which can be alleviated by means of a linearization of the conformation fields. Balci et al. [5] reported, recently, another method in which the square root of the conformation tensor is used to alleviate the stress imbalance for Oldroyd-B and FENE-P models. They derive an evolution equation for the positive-definite square root of the conformation tensor, by taking advantage of the fact that the positive definite symmetric polymer conformation tensor possesses a unique symmetric square root that satisfies a closed-form evolution equation. Balci et al. [5] claimed that their method can be easily implemented in numerical simulations, because it does not require the determination of eigenvectors and eigenvalues of the conformation tensor at every calculation step and location, resulting in significant practical advantages in terms of a reduction of computational cost, as well as in terms of enhanced accuracy and stability.

Based on these two works [1,5], a natural question arises: is it possible to obtain a generic transformation for the conformation tensor evolution equation, rather than the specific logarithm or square root transformations?

Here, we present a generic framework for a wide range of matrix transformations, allowing distinct regularizations of the spatial and temporal *exponential* profiles developed by the conformation tensor near critical points in the flow domain, such as re-entrant corners or stagnation flow points.

2. The *kernel-conformation* constitutive law – analytical framework

The *kernel-conformation* constitutive law is obtained by introducing the *kernel-conformation* tensor transformation, defined as

$$\mathbb{k}(\mathbf{A}) = \mathbf{O}\mathbb{k}(\mathbf{\Lambda})\mathbf{O}^T \tag{10}$$

where $\mathbb{k}()$ represents *any* continuous, invertible and differentiable matrix transformation function, and taking advantage of the fact that generally every *analytic function* of a diagonal matrix $\mathbf{d} = d_{ij}$ can be computed entrywise:

$$\mathbb{k}(\mathrm{diag}(d_{11}; d_{22}; \ldots; d_{nn})) = \mathrm{diag}(\mathbb{k}(d_{11}); \mathbb{k}(d_{22}); \ldots; \mathbb{k}(d_{nn})) \tag{11}$$

The *kernel* function will operate directly on the magnitude of the extension of the polymeric conformation, and not in the eigenvector directions of the deformations (and so the name "kernel"). We start by defining $\mathbb{k}()$ as a continuous, invertible and differentiable function of $\mathbf{\Lambda}$ defined in $\mathbb{R}$, and apply it directly to the diagonal eigendecomposed version of **A**, Eq. (8). In order to obtain an evolution equation for the *kernel-conformation* tensor, we use the following chain derivative rule on the diagonal matrix $\mathbb{k}(\mathbf{\Lambda})$:

$$\frac{\partial\mathbb{k}(\Lambda_{ii})}{\partial t} = \frac{\partial\Lambda_{ii}}{\partial t}\frac{\partial\mathbb{k}(\Lambda_{ii})}{\partial\Lambda_{ii}} = \frac{\partial\Lambda_{ii}}{\partial t}J_{ii} \tag{12}$$

for the diagonal element *ii* (no summation over *ii*), or in tensor form:

$$\frac{\partial\mathbb{k}(\mathbf{\Lambda})}{\partial t} = \frac{\partial\mathbf{\Lambda}}{\partial t}\mathbf{J} \tag{13}$$

where **J** is the gradient matrix, a diagonal matrix of the form

$$\mathbf{J} = \mathrm{diag}\left(\frac{\partial\mathbb{k}(\lambda_1)}{\partial\lambda_1}; \frac{\partial\mathbb{k}(\lambda_2)}{\partial\lambda_2}; \frac{\partial\mathbb{k}(\lambda_3)}{\partial\lambda_3}\right) \tag{14}$$

The evolution equation for the *kernel-conformation* tensor, in its eigenvalue formulation, can be expressed by

$$\frac{\mathrm{D}\mathbb{k}(\mathbf{\Lambda})}{\mathrm{D}t} = 2\widetilde{\mathbf{B}}\mathbf{\Lambda}\mathbf{J} + \frac{\mathcal{F}}{Wi}\mathcal{H}(\mathbf{\Lambda})\mathbf{J} \tag{15}$$

which is non-zero for the diagonal components of the tensor (no summation over *ii*):

$$\frac{\mathrm{D}\mathbb{k}(\lambda_i)}{\mathrm{D}t} = 2\widetilde{B}_{ii}\lambda_i J_{ii} + \frac{\mathcal{F}}{Wi}\mathcal{H}(\lambda_i)J_{ii} \tag{16}$$

Note that the orthogonality is maintained for the $\mathbb{k}(\boldsymbol{\Lambda})$ transfomation, since $\mathbf{A}$ is positive definite and symmetric, then its eigenvectors are real, mutually orthogonal and provide a basis for $\mathbb{R}^n$, and the eigenvectors of $\mathbb{k}(\mathbf{A})$ are the same as the eigenvectors of $\mathbf{A}$.

The evolution equation for $\mathbb{k}(\mathbf{A})$ can be obtained using the chain derivative rule for $\mathbb{k}(\mathbf{A}) = \mathbf{O}\mathbb{k}(\boldsymbol{\Lambda})\mathbf{O}^T$

$$\frac{\mathrm{D}\mathbb{k}(\mathbf{A})}{\mathrm{D}t} = \frac{\mathrm{D}\mathbf{O}}{\mathrm{D}t}\mathbb{k}(\boldsymbol{\Lambda})\mathbf{O}^T + \mathbf{O}\frac{\mathrm{D}\mathbb{k}(\boldsymbol{\Lambda})}{\mathrm{D}t}\mathbf{O}^T + \mathbf{O}\mathbb{k}(\boldsymbol{\Lambda})\frac{\mathrm{D}\mathbf{O}^T}{\mathrm{D}t} \tag{17}$$

which using Eq. (15) and after some manipulation leads to (note that from Eq. (9) we conclude that $\mathcal{O} = \widetilde{\boldsymbol{\Omega}}$):

$$\frac{\mathrm{D}\mathbb{k}(\mathbf{A})}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbb{k}(\mathbf{A}) - \mathbb{k}(\mathbf{A})\boldsymbol{\Omega} + 2\mathbb{B} + \frac{1}{Wi}\mathbb{H} \tag{18}$$

where $\mathbb{B}$ and $\mathbb{H}$ are symmetric tensors constructed by the orthogonalization of the diagonal tensors $\mathbf{D}_{\mathbb{B}}$ and $\mathbf{D}_{\mathbb{H}}$, respectively. These tensors can be constructed as

$$\begin{aligned} \mathbb{B} &= \mathbf{O}\mathbf{D}_{\mathbb{B}}\mathbf{O}^T = \mathbf{O}\widetilde{\mathbf{B}}\boldsymbol{\Lambda}\mathbf{J}\mathbf{O}^T \\ \mathbb{H} &= \mathbf{O}\mathbf{D}_{\mathbb{H}}\mathbf{O}^T = \mathcal{F}\mathbf{O}\mathcal{H}(\boldsymbol{\Lambda})\mathbf{J}\mathbf{O}^T \end{aligned} \tag{19}$$

3. Some examples of *kernel-conformation* transformations

The *kernel-conformation* transformation function can include *any* continuous, invertible and differentiable matrix transformation function. Here, and for the sake of exemplification, only three different types of functions $\mathbb{k}(\mathbf{A})$ are considered, namely the linear shifted $\mathbf{A}$, the $\log_a(\mathbf{A})$ and the $\text{root}^k(\mathbf{A})$. For simplicity, the Oldroyd-B differential equation is used in this representation, for which $\mathcal{F}(\mathbf{A}) = 1$ and $\mathcal{H}(\mathbf{A}) = \mathbf{I} - \mathbf{A}$.

3.1. The linear shifted kernel-conformation transformation

Defining the linear shifted *kernel-conformation* transformation function as

$$\mathbb{k}(\mathbf{A}) = \frac{1}{Wi}(\mathbf{A} - \mathbf{I}) = \frac{1}{Wi}\mathbf{O}(\boldsymbol{\Lambda} - \mathbf{I})\mathbf{O}^T \tag{20}$$

For this kernel function, tensors $\mathbb{B}$ and $\mathbb{H}$ can be constructed following Eq. (19), and we obtain:

$$\begin{aligned} \mathbb{B} &= \mathbf{O}\mathbf{D}_{\mathbb{B}}\mathbf{O}^T = \frac{1}{Wi}\mathbf{O}\widetilde{\mathbf{B}}\boldsymbol{\Lambda}\mathbf{O}^T = \frac{1}{Wi}\mathbf{B}\mathbf{A} = \mathbf{B}\left(\mathbb{k} + \frac{1}{Wi}\mathbf{I}\right) \\ \mathbb{H} &= \mathbf{O}\mathbf{D}_{\mathbb{H}}\mathbf{O}^T = \frac{1}{Wi}\mathbf{O}(\mathbf{I} - \boldsymbol{\Lambda})\mathbf{O}^T = \frac{1}{Wi}(\mathbf{I} - \mathbf{A}) = -\mathbb{k} \end{aligned} \tag{21}$$

Substituting tensors $\mathbb{B}$ and $\mathbb{H}$ in Eq. (18), leads to

$$\frac{\mathrm{D}\mathbb{k}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbb{k} - \mathbb{k}\boldsymbol{\Omega} + 2\mathbf{B}\left(\mathbb{k} + \frac{1}{Wi}\mathbf{I}\right) - \frac{1}{Wi}\mathbb{k} \tag{22}$$

After some manipulation and recalling Eq. (4), then Eq. (22) can be rearranged to obtain:

$$Wi\left(\frac{\mathrm{D}\boldsymbol{\tau}}{\mathrm{D}t} - \boldsymbol{\tau}\nabla\mathbf{u} - \nabla\mathbf{u}^T\boldsymbol{\tau}\right) = Wi\overset{\triangledown}{\boldsymbol{\tau}} = 2\mathbf{D} - \boldsymbol{\tau} \tag{23}$$

i.e., one arrives at the well known evolution equation for the Oldroyd-B model based in the extra stress tensor, $\boldsymbol{\tau}$, because in this case $\boldsymbol{\tau} = \mathbb{k} = \frac{1}{Wi}(\mathbf{A} - \mathbf{I})$ and $\mathbf{D} = (\mathbf{L} + \mathbf{L}^T)/2$.

3.2. The $\log_a$ kernel-conformation transformation

In the $\log_a$ *kernel-conformation* transformation, the functional kernel relation is based on the logarithm of *any* base a of the conformation tensor:

$$\mathbb{k}(\mathbf{A}) = \log_a\mathbf{A} = \mathbf{O}\log_a(\boldsymbol{\Lambda})\mathbf{O}^T \tag{24}$$

The inverse *kernel* function is represented by

$$\mathbf{A} = a^{\mathbb{k}} \tag{25}$$

and the diagonal gradient matrix $\mathbf{J}$ can be computed as

$$\mathbf{J} = \mathrm{diag}\left(\frac{\partial\log_a(\lambda_1)}{\partial\lambda_1};\frac{\partial\log_a(\lambda_2)}{\partial\lambda_2};\frac{\partial\log_a(\lambda_3)}{\partial\lambda_3}\right) = \frac{\boldsymbol{\Lambda}^{-1}}{\ln(a)} \tag{26}$$

The evolution equation for the $\log_a$ *kernel-conformation* tensor is

$$\frac{\mathrm{D}\mathbb{k}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbb{k} - \mathbb{k}\boldsymbol{\Omega} + 2\frac{\mathbf{B}}{\ln(a)} + \frac{1}{\ln(a)Wi}(a^{-\mathbb{k}} - \mathbf{I}) \tag{27}$$

If the natural logarithm is used (meaning that, $a = e$, $\ln(a) = 1$ and $\mathbb{k}(\mathbf{A}) = \boldsymbol{\Psi} = \ln(\mathbf{A})$), the original *log-conformation* of Fattal and Kupferman [1] is recovered

$$\frac{\mathrm{D}\boldsymbol{\Psi}}{\mathrm{D}t} = \boldsymbol{\Omega}\boldsymbol{\Psi} - \boldsymbol{\Psi}\boldsymbol{\Omega} + 2\mathbf{B} + \frac{1}{Wi}(e^{-\boldsymbol{\Psi}} - \mathbf{I}) \tag{28}$$

3.3. The root^k kernel-conformation transformation

In the root^k *kernel-conformation* transformation family, the *kernel* relation is defined as:

$$\mathbb{k}(\mathbf{A}) = \mathbf{A}^{\frac{1}{k}} = \mathbf{O}\boldsymbol{\Lambda}^{\frac{1}{k}}\mathbf{O}^T \tag{29}$$

The inverse root^k *kernel* function is represented by

$$\mathbf{A} = \mathbb{k}^k \tag{30}$$

and the diagonal gradient matrix $\mathbf{J}$ is

$$\mathbf{J} = \mathrm{diag}\left(\frac{\partial\lambda_1^{\frac{1}{k}}}{\partial\lambda_1};\frac{\partial\lambda_2^{\frac{1}{k}}}{\partial\lambda_2};\frac{\partial\lambda_3^{\frac{1}{k}}}{\partial\lambda_3}\right) = \frac{\boldsymbol{\Lambda}^{\frac{1-k}{k}}}{k} \tag{31}$$

The evolution equation for the root^k *kernel-conformation*, can be expressed by

$$\frac{\mathrm{D}\mathbb{k}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbb{k} - \mathbb{k}\boldsymbol{\Omega} + \frac{2}{k}\mathbf{B}\mathbb{k} + \frac{1}{kWi}(\mathbb{k}^{1-k} - \mathbb{k}) \tag{32}$$

For $k = 1$ (meaning that $\mathbb{k} = \mathbf{A}$), this transformation results in the *standard conformation* formulation, represented by Eq. (3). For $k = -1$ (meaning that $\mathbb{k} = \mathbf{A}^{-1}$), it can be shown that the resulting equation

$$\frac{\mathrm{D}\mathbb{k}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbb{k} - \mathbb{k}\boldsymbol{\Omega} - 2\mathbf{B}\mathbb{k} - \frac{1}{Wi}(\mathbb{k}^2 - \mathbb{k}) \tag{33}$$

is the same as the evolution equation obtained by some matrix manipulation of Eq. (3) (note that $\frac{\mathrm{D}\mathbf{A}^{-1}}{\mathrm{D}t} = -\mathbf{A}^{-1}\frac{\mathrm{D}\mathbf{A}}{\mathrm{D}t}\mathbf{A}^{-1}$):

$$\frac{\mathrm{D}\mathbf{A}^{-1}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbf{A}^{-1} - \mathbf{A}^{-1}\boldsymbol{\Omega} - 2\mathbf{B}\mathbf{A}^{-1} - \frac{1}{Wi}((\mathbf{A}^{-1})^2 - \mathbf{A}^{-1}) \tag{34}$$

For $k = 2$, Eq. (32) reduces to

$$\frac{\mathrm{D}\mathbb{k}}{\mathrm{D}t} = \boldsymbol{\Omega}\mathbb{k} - \mathbb{k}\boldsymbol{\Omega} + \mathbf{B}\mathbb{k} + \frac{1}{2Wi}(\mathbb{k}^{-1} - \mathbb{k}), \tag{35}$$

and since $\mathbb{k} = \mathbf{b} = \mathbf{A}^{\frac{1}{2}}$, this transformation is equivalent to the *square-root-conformation* proposed recently by Balci et al. [5]

$$\frac{\mathrm{D}\mathbf{b}}{\mathrm{D}t} = \mathbf{a}\mathbf{b} + \mathbf{b}\nabla\mathbf{u} + \frac{1}{2Wi}((\mathbf{b}^T)^{-1} - \mathbf{b}) \tag{36}$$

where $\mathbf{a}$ is an anti-symmetric tensor and the tensor $\mathbf{b}$ is symmetric, leading to $(\mathbf{b}^T)^{-1} = \mathbf{b}^{-1}$.

Many other functions can be used for the matrix transformations. These are not explained here, but some were also tested numerically as mentioned in the next section.

4. Numerical analysis of the *kernel-conformation* transformations

The *analytical kernel-conformation* framework is generic, and by itself does not guarantee a well-posed numerical solution for high *Wi* numbers (indeed for the linear shifted **A** we recover the standard formulation based on the extra-stress tensor, which suffers from the HWNP). This will depend on the characteristic behavior of the selected *kernel* function (as also on the inverse function characteristics), and in particular on its ability to smooth the sharp conformation tensor field zones to better balance numerically the exponential growth and convection of the conformation tensor field. Naturally, it is also relevant if it helps maintain the positive definiteness of the conformation tensor to ensure realizability.

From a numerical point of view, two main questions arise naturally in relation to the *kernel-conformation* transformation: is *any* valid kernel function also suitable numerically? If so, which function(s) is(are) the most appropriate for high *Wi* simulations?

Related to the first question, we implemented the *kernel-conformation* in a Finite Volume Method code (see [6–9], for more details). The implementation followed Eqs. (19) and (18), meaning that tensors $\mathbf{D}_{\mathbb{B}}$ and $\mathbf{D}_{\mathbb{H}}$ were calculated first and then $\mathbb{B}$ and $\mathbb{H}$ were created by orthogonalization.

The benchmark flow of an Oldroyd-B fluid past a confined circular cylinder [10], represented in Fig. 1, was selected as a test case. The ratio of channel half-height *h* to cylinder radius *R* is set equal to 2, which corresponds to the benchmark 50% blockage case [10]. The computational domain is 200*R* long, with 99*R* upstream and 99*R* downstream of the forward and rear stagnation points of the cylinder, respectively. The downstream length is sufficiently long for the flow to become fully-developed and to avoid any effect of the outflow boundary condition upon the flow in the vicinity of the cylinder. Vanishing streamwise gradients are applied to all variables at the outlet plane, except for the pressure which is linearly extrapolated to the outlet from the two nearest upstream cells. No-slip conditions are imposed at both the cylinder surface and the channel walls. The main characteristics of the meshes used here are identical to those used in Afonso et al. [9]. These two computational meshes map the complete flow domain (i.e. no symmetry boundary condition was imposed along the centerline). Mesh *M*30 is composed of 30 cells placed radially from the cylinder to the channel wall, leading to a minimum cell spacing normalized with the cylinder radius along the radial (Δr) and the azimuthal ($\Delta s = R\Delta\theta$) directions, of 0.008 and 0.0012, respectively. In mesh *M*60, those dimensions are reduced to 0.004 and 0.0006, respectively. All the calculations were carried out at a vanishing Reynolds number, $Re = \rho UR/\eta = 0$, i.e., assuming creeping flow conditions which are set numerically by dropping out the advective term in the momentum equation. The time step used was the same in all the simulations, $\Delta t = 10^{-3}$. The viscosity ratio, β, here defined as the ratio between the Newtonian solvent viscosity, η_s, and the total zero shear-rate viscosity, $\eta_0 = \eta_s + \eta_p$, is fixed at $\beta = 0.59$, as in previous works [11,12]. Results of computations include the dimensionless drag coefficient, *K*, here calculated as

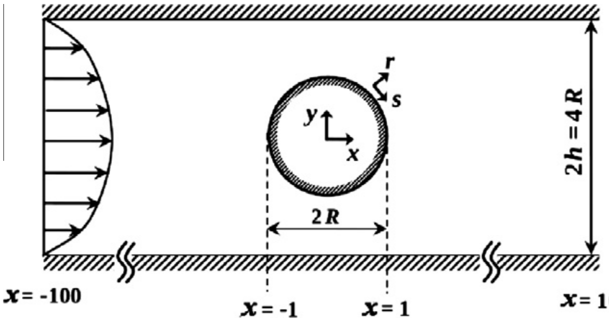

Fig. 1. Schematic representation of the flow past a confined cylinder geometry.

$$K = \frac{F_x}{\eta_0 UR} \tag{37}$$

where F_x is the drag force on the cylinder surface.

Several *kernel* functions were assessed at moderate $Wi = 0.6$ and compared with the solution obtained with the *standard* extra-stress tensor formulation (*StrT*), using both meshes. The functions tested were:

- $\Bbbk(\lambda_i) = \log_a(\lambda_i)$ with $a = e^{-1}$, 2, e, 10 and an adaptive version with $a^{(n)} = \sqrt{\lambda_{max}^{(n-1)}}, (n)$ representing the time level and λ_{max} the maximum eigenvalue of **A** in the computed domain;
- $\Bbbk(\lambda_i) = \lambda_i^{\frac{1}{k}}$ with $k = -16, -8, -4 - 2, 2$ and 4;
- $\Bbbk(\lambda_i) = \ln\left(\ln\left(\lambda_i^2 + 1\right)\right)$;
- $\Bbbk(\lambda_i) = \text{asinh}(\lambda_i)$.

where asinh (λ_i) represents the inverse hyperbolic sine function. For all the functions used (see also Table 1), the numerical solutions obtained with the coarse mesh *M*30 were consistent with the *StrT* discretization, as observed in the time evolution of the dimensionless drag coefficient, *K*, presented in Fig. 2 (also represented in Table 1). For the refined mesh *M*60, simulations for some kernel functions diverged, such as for some $\log_a$ family functions (for the highest logarithm base tested, $a = 10$, and for the adaptive version) and some root^k family functions (with $k = 4$). This failure can be related to numerical errors due to excessive sub-linearization in some zones (or even due to over-exponentiation on the inverse transformation). The introduction of zone-adaptive functions could be a solution, and will be addressed in a future work. For all the converged functions (see Fig. 3 and Table 1), the results obtained are in good agreement with the *StrT* formulation, as observed in the nondimensional τ_{xx} profiles presented in Fig. 3a. The contour maps of the determinant of the conformation tensor, $|\mathbf{A}|$ (cf. Fig. 3b), for the $\log_{1/e}(\lambda_i)$ and $\lambda_i^{\frac{1}{2}}$ kernel functions, show that $|\mathbf{A}|_{min} \geqslant 1$ in the whole domain. These results show that the *kernel-conformation* is also valid numerically, for valid kernel functions.

In order to obtain more insight into the question of which kernel function is numerically more efficient at high *Wi*, we first

Table 1
Dimensionless drag coefficient, *K*, obtained with different *kernel* functions and meshes.

		*M*30		*M*60	
		Wi = 0.6	*Wi* = 1.0	*Wi* = 0.6	*Wi* = 1.0
$\Bbbk(\lambda_i)$	Parameter	*K*	*K*	*K*	*K*
StrT	–	117.768	$*$	117.776	$*$
$\log_a(\lambda_i)$	$a = e^{-1}$	117.778	(118.578)	117.774	(118.733)
	$a = 2$	117.778	(118.578)	117.774	(118.733)
	$a = e$	117.778	(118.578)	117.774	(118.733)
	$a = 10$	117.778	(118.578)	$*$	$*$
	$a^{(n)} = \sqrt{\lambda_{max}^{(n-1)}}$	117.778	(118.580)	$*$	$*$
$\lambda_i^{\frac{1}{k}}$	$k = -16(-2^4)$	117.776	(118.604)	117.773	118.707
	$k = -8(-2^3)$	117.773	(118.535)	117.772	(118.682)
	$k = -4(-2^2)$	117.766	(118.487)	117.770	(118.644)
	$k = -2(-2^1)$	117.752	(118.184)	117.767	(118.461)
	$k = 2(2^1)$	117.792	$*$	117.779	$*$
	$k = 4(2^2)$	117.789	$*$	$*$	$*$
$\ln\left(\ln\left(\lambda_i^2 + 1\right)\right)$	-	117.760	(118.454)	117.768	$*$
$\text{asinh}(\lambda_i)$	–	117.781	$*$	117.775	118.782

$*$ – Diverged and () – mean.

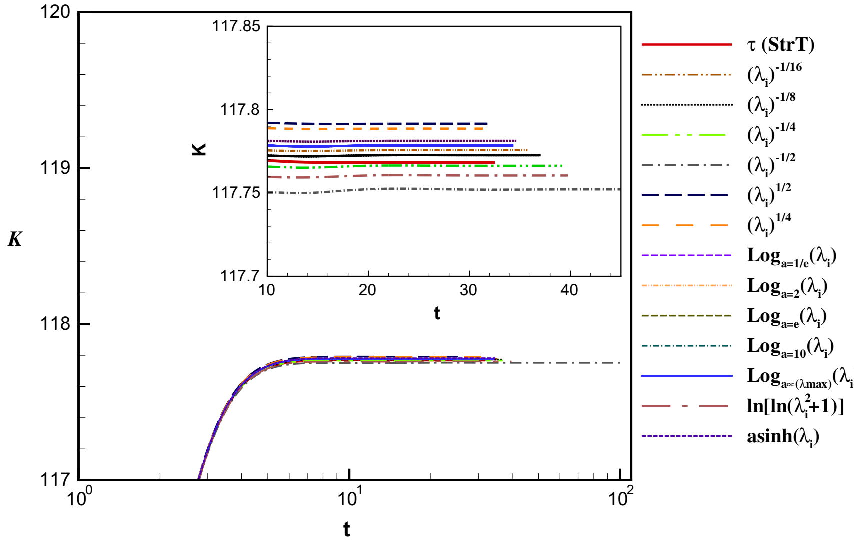

Fig. 2. Time evolution of the dimensionless drag coefficient, *K*, for the Oldroyd-B model, $\beta = 0.59$ at $Wi = 0.6$ and mesh *M*30.

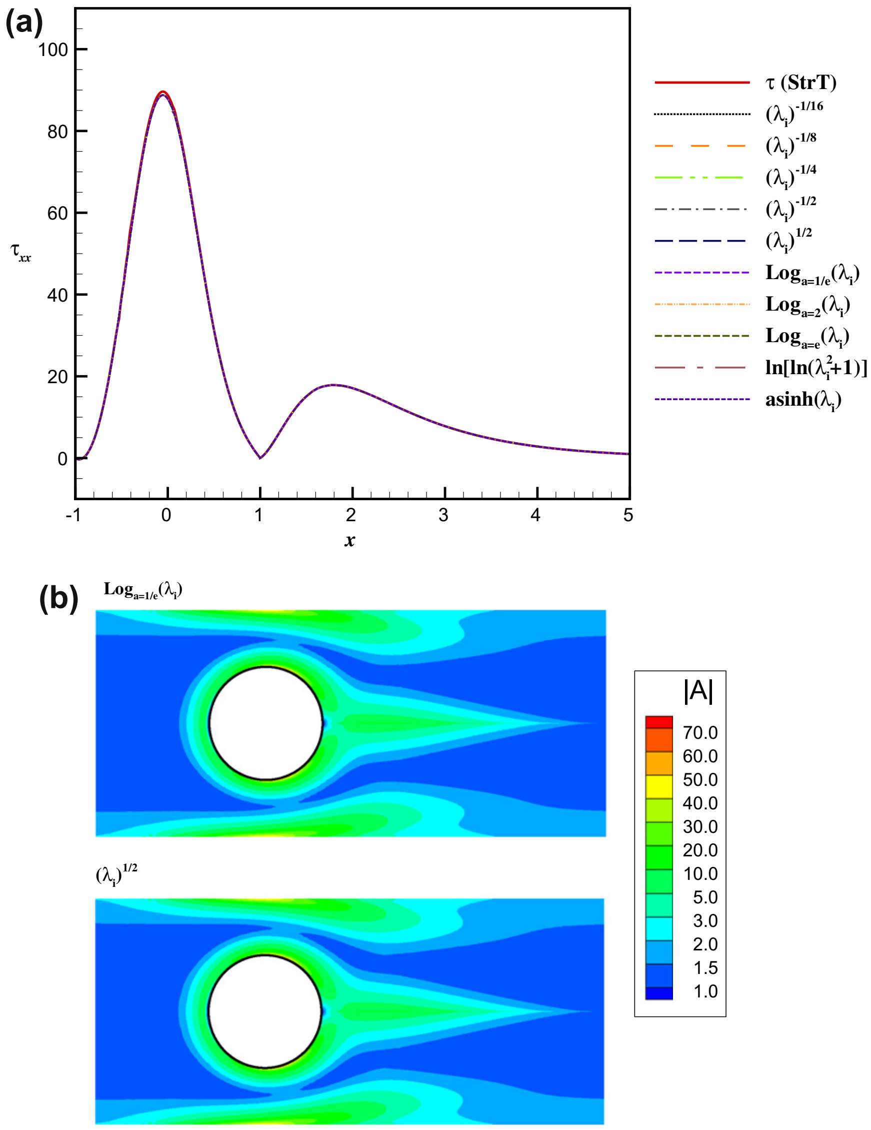

Fig. 3. (a) Normal stress profiles along the cylinder surface and the rear wake and (b) countour maps of the determinant of the conformation tensor, |**A**| (for $\log_{1/e}(\lambda_i)$ and $\lambda_i^{\frac{1}{2}}$), for the Oldroyd-B model, $\beta = 0.59$ at $Wi = 0.6$ and *M*60.

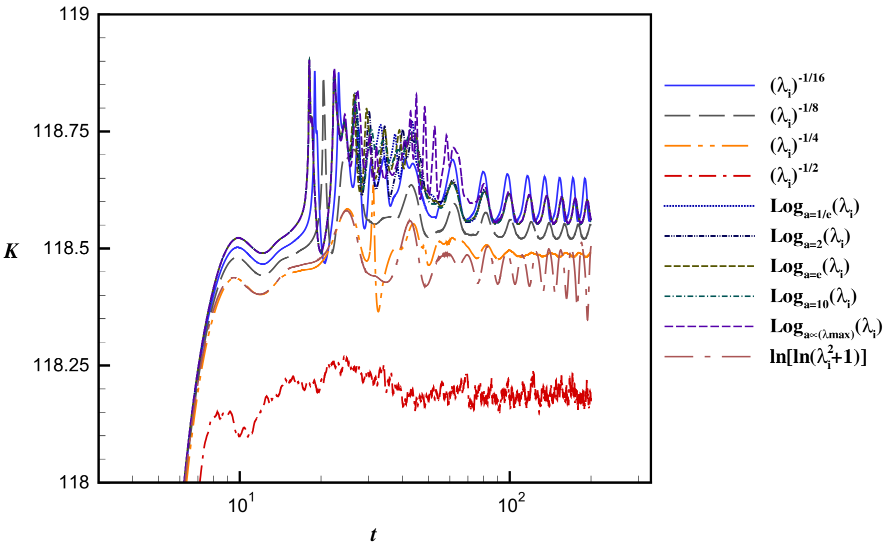

Fig. 4. Time evolution of the dimensionless drag coefficient, K, for the Oldroyd-B model, $\beta = 0.59$ at $Wi = 1$ and mesh $M30$.

follow the sequence of simplifications invoked in refs. [1,2,13], and use the following *toy model*: assume that the velocity field is fixed and replace the conformation tensor equation by a scalar equation of similar type in one space dimension. Here we will work with a one-dimensional *eigenvalue* scalar equation (see Eq. (8)),

$$\frac{\partial\lambda}{\partial t}+u\frac{\partial\lambda}{\partial x}=2L\lambda+\frac{1}{Wi}(1-\lambda)=(2L-Wi^{-1})\lambda+Wi^{-1}$$
$$=b\lambda+c \qquad (38)$$

where u is a convective velocity, assumed constant, and $(2L-Wi^{-1})$ is the exponential growth factor. As in [13], we will assume $\lambda_{|x,t=0} = 1$ as initial condition and $\lambda_{|x=0,t} = 1$ as inflow boundary condition. The general solution for this partial differential equation in x=[0,X] is given by

$$\lambda(x,t)=\begin{cases}\frac{(b+c)e^{bt}-c}{b}=\frac{2Le^{(2L-Wi^{-1})t}-Wi^{-1}}{(2L-Wi^{-1})} & \text{for } ut<x\leqslant X\\ \frac{(b+c)e^{\frac{b}{u}x}-c}{b}=\frac{2Le^{\frac{2L-Wi^{-1}}{u}x}-Wi^{-1}}{(2L-Wi^{-1})} & \text{for } x\leqslant ut\end{cases} \qquad (39)$$

Note that for the limit of high Wi, these solutions simplify to

$$\lambda(x,t)=\begin{cases}e^{2Lt} & \text{for } ut<x\leqslant X\\ e^{\frac{2L}{u}x} & \text{for } x\leqslant ut\end{cases} \qquad (40)$$

and the original Eq. (38) reduces to

$$\frac{\partial\lambda}{\partial t}+u\frac{\partial\lambda}{\partial x}=2L\lambda \qquad (41)$$

The inherent numerical difficulties of the discretization of such simple equation were elegantly stressed out by Kupferman and coworkers [1,2,13], especially the failure to balance the exponential growth presented in Eq. (40) by the convective forcing. This failure is not solved by the use of higher-order methods or implicit methods, and is found to exist whenever classical non-Lagrangian type of discretization are used (Finite element, finite differences or finite volume methods).

Now we focus on the kernel version of this *eigenvalue toy model*,

$$\frac{\partial\Bbbk(\lambda)}{\partial t}+u\frac{\partial\Bbbk(\lambda)}{\partial x}=2L\lambda\frac{\partial\Bbbk(\lambda)}{\partial\lambda}+\frac{1}{Wi}(1-\lambda)\frac{\partial\Bbbk(\lambda)}{\partial\lambda}=b\lambda J+cJ \qquad (42)$$

By design, given the inherent properties of the variable transformation of the kernel formalism and if the initial and inflow conditions can be assumed to be $\Bbbk(\lambda)_{|x,t=0}=\Bbbk(\lambda_{|x,t=0})$ and $\Bbbk(\lambda)_{|x=0,t}=\Bbbk(\lambda_{|x=0,t})$, then the final solution of $\Bbbk(x,t)$ is:

$$\Bbbk(x,t)=\Bbbk(\lambda(x,t)) \qquad (43)$$

meaning that the solution for the kernel variable can be obtained by applying the kernel function to the $\lambda(x,t)$ solution. Therefore, for instance the generic solution for the specific kernel function $\Bbbk()=\log_a(\lambda)$ is

$$\Bbbk(x,t)=\Bbbk(\lambda(x,t))=\begin{cases}\frac{\ln\left[\frac{2Le^{(2L-Wi^{-1})t}-Wi^{-1}}{(2L-Wi^{-1})}\right]}{\ln(a)} & \text{for } ut<x\leqslant X\\ \frac{\ln\left[\frac{2Le^{\frac{2L-Wi^{-1}}{u}x}-Wi^{-1}}{(2L-Wi^{-1})}\right]}{\ln(a)} & \text{for } x\leqslant ut\end{cases} \qquad (44)$$

which for high Wi simplifies to

$$\Bbbk(x,t)=\begin{cases}\frac{2L}{\ln(a)}t & \text{for } ut<x\leqslant X\\ \frac{2L}{u\ln(a)}x & \text{for } x\leqslant ut\end{cases}=\Bbbk(\lambda(x,t)) \qquad (45)$$

showing that the logarithm kernel linearizes the exponential growth of $\Bbbk$, as already discussed in [1].

For the kernel function $\Bbbk()=\lambda^{\frac{1}{k}}$ the solution is

$$\Bbbk(x,t)=\Bbbk(\lambda(x,t))=\begin{cases}\left[\frac{2Le^{(2L-Wi^{-1})t}-Wi^{-1}}{(2L-Wi^{-1})}\right]^{\frac{1}{k}} & \text{for } ut<x\leqslant X\\ \left[\frac{2Le^{\frac{2L-Wi^{-1}}{u}x}-Wi^{-1}}{(2L-Wi^{-1})}\right]^{\frac{1}{k}} & \text{for } x\leqslant ut\end{cases} \qquad (46)$$

For high Wi, these equations simplify to

$$\Bbbk(x,t)=\Bbbk(\lambda(x,t))=\begin{cases}e^{\frac{2L}{k}t} & \text{for } ut<x\leqslant X\\ e^{\frac{2L}{ku}x} & \text{for } x\leqslant ut\end{cases} \qquad (47)$$

The above solutions indicate that when the root function is used, the obtained exponential growth is not linearized but only normalized by a k factor. This could be an important fact in understanding the numerical behavior of this kernel family at high Wi.

In order to obtain more information of the "efficient" *kernel* function for high Wi, we carried out several simulations at Wi = 1 in both meshes, a situation in which the *standard* extra-stress tensor formulation diverges. For the coarse mesh $M30$ and all used functions, stabilized solutions were only obtained for *any* $\log_a$ and root^k (if $k\leqslant-2$) functions, whereas for all the other functions,

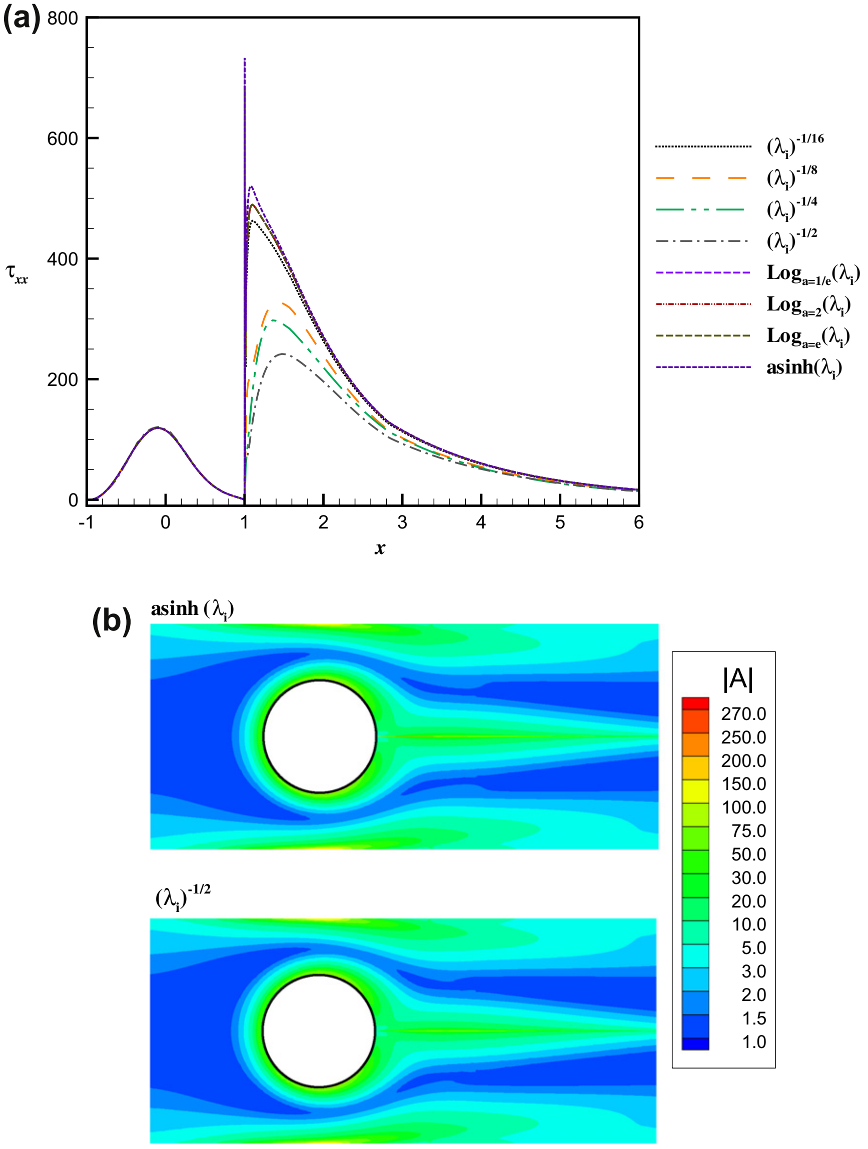

Fig. 5. (a) Normal stress profiles along the cylinder surface and the rear wake and (b) contour maps of the determinant of the conformation tensor, |**A**| (for asinh(λ_i) and $\lambda_i^{-\frac{1}{2}}$), for the Oldroyd-B model, $\beta = 0.59$ at $Wi = 1$ and $M60$.

such as the $\ln\left(\ln\left(\lambda_i^2+1\right)\right)$ transformation, the solution diverged, as observed in the time evolution of K presented in Fig. 4, and also reported in Table 1. The effect of mesh refinement (using $M60$) introduces again some restrictions on the parameters used in the kernel functions. Stable solutions (but unsteady) could be obtained for the $\log_a$ family with *low* logarithm base ($a < 10$), while for the root^k family, k was restricted to $-16 \leqslant k \leqslant -2$ (with $k \in \mathbb{Z}$). Since at $Wi = 1$ the numerical solution becomes unsteady [9,13], the τ_{xx} profiles presented in Fig. 5a were obtained at a fixed time of $t = 200$ for all the simulations, while the K values presented in Table 1 are mean values along time. The contour maps of the determinant of the conformation tensor, |**A**|, presented in Fig. 5b, also obtained at $t = 200$, show that $|\mathbf{A}|_{min} \geqslant 1$ everywhere, as they should [13].

These results can be a good guide to estimate the "efficient" *kernel* functions for high Wi simulations: the $\log_a$ functions with *small* logarithm base ($a < 10$) and the root^k with $-16 \leqslant k \leqslant -2$ are, so far, the best candidates. Further analyses on the accuracy of the numerical versions of these (or other) functions will be assessed field in future work.

5. Conclusions

A generic *Kernel-conformation* tensor transformation for a large class of differential constitutive models is presented in this work. This *analytical kernel-conformation* framework, in which several matrix *kernel*-transformation(s) families can be applied to the conformation tensor evolution equation, is generic and by itself does not guarantee a well-posed numerical solution for high Wi numbers.

The numerical discretization of this generic analytical framework is consistent with the standard conformation tensor discretization at low Weissenberg numbers, and the numerical efficiency for High Weissenberg Number Problems is ultimately related with the kernel function used in the matrix transformation. This will depend on the characteristic behavior of the selected *kernel* function (as also on the inverse function characteristics), and in particular on its ability to smooth the sharp conformation field zones to better balance numerically the exponential growth and convection of the conformation tensor field. Naturally, it is also relevant if it preserves the positive definiteness of the conformation tensor. The numerical tests conducted so far have shown that the best candidates for an “efficient” *kernel* function for high *Wi* simulations are the $\log_a$ functions with *small* logarithm base ($a < 10$) and the root^k with $-16 \leqslant k \leqslant -2$, but future work is required to assess their behavior in different flows.

Acknowledgements

The authors acknowledge funding from FEDER and Fundação para a Ciência e a Tecnologia (FCT), Portugal, through projects PTDC/EQU-FTT/70727/2006 and PTDC/EQU-FTT/113811/2009. A.M. Afonso would also like to thank FCT for financial support through the scholarship SFRH/BPD/75436/2010. M.A. Alves acknowledges the Chemical Engineering Department of FEUP for conceding a sabbatical leave.

References

[1] R. Fattal, R. Kupferman, Constitutive laws for the matrix-logarithm the conformation tensor, J. Non-Newton. Fluid Mech. 123 (2–3) (2004) 281–285.

[2] R. Fattal, R. Kupferman, Time-dependent simulation of viscoelastic flows at high Weissenberg number using the log-conformation representation, J. Non-Newton. Fluid Mech. 126 (1) (2005) 23–37.

[3] T. Vaithianathan, L.R. Collins, Numerical approach to simulating turbulent flow of a viscoelastic polymer solution, J. Comput. Phys. 187 (2003) 1–21.

[4] D.D. Joseph, Fluid Dynamics of Viscoelastic Liquids, Springer-Verlag, New York, 1990.

[5] N. Balci, B. Thomases, M. Renardy, C.R. Doering, Symmetric factorization of the conformation tensor in viscoelastic fluid models, J. Non-Newton. Fluid Mech. 166 (11) (2011) 546–553.

[6] P.J. Oliveira, F.T. Pinho, G.A. Pinto, Numerical simulation of non-linear elastic flows with a general collocated finite-volume method, J. Non-Newton. Fluid Mech. 79 (1998) 1–43.

[7] M.A. Alves, F.T. Pinho, P.J. Oliveira, Effect of a high-resolution differencing scheme on finite-volume predictions of viscoelastic flows, J. Non-Newton. Fluid Mech. 93 (2000) 287–314.

[8] M.A. Alves, P.J. Oliveira, F.T. Pinho, A convergent and universally bounded interpolation scheme for the treatment of advection, Int. J. Numer. Methods Fluids 41 (2003) 47–75.

[9] A. Afonso, P.J. Oliveira, F.T. Pinho, M.A. Alves, The log-conformation tensor approach in the finite-volume method framework, J. Non-Newton. Fluid Mech. 157 (2009) 55–65.

[10] R.A. Brown, G.H. McKinley, Report on the VIIIth international workshop on numerical methods in viscoelastic flows, J. Non-Newton. Fluid Mech. 52 (3) (1994) 407–413.

[11] Y. Fan, R.I. Tanner, N. Phan-Thien, Galerkin/least-square finite element methods for steady viscoelastic flows, J. Non-Newton. Fluid Mech. 84 (1999) 233–256.

[12] M.A. Alves, F.T. Pinho, P.J. Oliveira, The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods, J. Non- Newton. Fluid Mech. 97 (2001) 207–232.

[13] M.A. Hulsen, R. Fattal, R. Kupferman, Flow of viscoelastic fluids past a cylinder at high Weissenberg number: stabilized simulations using matrix logarithms, J. Non-Newton. Fluid Mech. 127 (2005) 27–39.